



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación



**“ANÁLISIS DE ESTABILIDAD DE UN PÉNDULO INVERTIDO ROTATIVO
UTILIZANDO TÉCNICAS DE CONTROL NO LINEAL”**

TRABAJO DE TITULACIÓN

Previo a la obtención del Título de:

MAGISTER EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

Presentado por:

Ing. Byron Xavier Lima Cedillo

Guayaquil – Ecuador

2016

AGRADECIMIENTO

Agradezco a Dios, a mis padres y a mi esposa Verónica, que me han brindado su apoyo incondicional hasta conseguir la meta.

También la Universidad Politécnica Salesiana-Sede Guayaquil por permitir el uso de sus laboratorios.

DEDICATORIA

Dedicado a todas las personas que me brindaron su ayuda y consejo desinteresadamente en el transcurso de mi vida.

En especial a mi madre Lupe, mi padre José y a mi esposa Verónica, por ser un pilar fundamental en esta etapa que culmina.

TRIBUNAL DE SUSTENTACIÓN

Ing. LENIN FREIRE C.

SUB-DECANO FIEC



Ph.D. WILTON AGILA

DIRECTOR

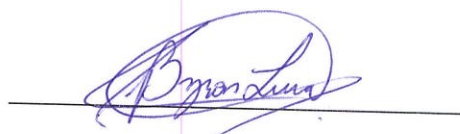


PHD. DOUGLAS PLAZA G.

VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”.



Ing. Byron Xavier Lima Cedillo

RESUMEN

El presente trabajo se basa en el análisis de sistemas sub-actuados, concretamente el Péndulo Invertido Rotativo (PIR), para observar su dinámica y notar la dificultad en la estabilización de dicho sistema. Se escogió el sistema PIR, debido a que es un sistema de fase no mínima y altamente no lineal. En este contexto, se analiza el comportamiento del sistema con la finalidad de obtener un modelo matemático no lineal, seguido a esto, se procede a linealizar dicho modelo en torno a un punto de operación determinado. Cabe mencionar que ambos modelos han sido validados experimentalmente. En base al modelo lineal, se procede a diseñar el controlador PID lineal para estabilizar el péndulo en la posición vertical arriba y luego se diseña el controlador PI lineal para regular la posición angular del brazo. Utilizando el procedimiento de linealización extendida, se procede a diseñar el controlador PI no lineal para regular el recorrido del brazo. Finalmente, se analiza el comportamiento de los controladores, PI lineal y no lineal, implementados en el sistema PIR a través de las gráficas: señal de control, posición angular del brazo y posición angular del péndulo.

Palabras claves: Control lineal, Control no Lineal, PID, PIR, Modelo matemático, Linealización extendida, sub-actuado.

ÍNDICE GENERAL

AGRADECIMIENTO.....	II
DEDICATORIA.....	III
TRIBUNAL DE SUSTENTACIÓN.....	IV
DECLARACIÓN EXPRESA.....	V
RESUMEN	VI
ÍNDICE GENERAL	VII
ABREVIATURAS y SIMBOLOGÍA	XI
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABLAS	XXI
INTRODUCCIÓN	XXII
CAPÍTULO 1	1
1. ANÁLISIS DEL PROBLEMA	1
1.1 <i>Planteamiento del problema</i>	1
1.2 <i>Solución Propuesta</i>	3
1.3 <i>Delimitación del problema</i>	4
1.4 <i>Objetivos</i>	4
1.4.1 Objetivo General	4
1.4.2 Objetivos Específicos.....	4
1.5 <i>Metodología</i>	5
CAPÍTULO 2	6
2. ANÁLISIS DE HERRAMIENTAS Y CONOCIMIENTOS DISPONIBLES	6
2.1 <i>Antecedentes</i>	6

2.2	<i>Sistemas sub-actuados</i>	8
2.3	<i>Control de procesos</i>	9
2.3.1	El lazo de control.....	10
2.3.2	Variables de proceso	11
2.4	<i>Codificadores incrementales (Encoders)</i>	12
2.5	<i>Sistemas de control en lazo abierto</i>	13
2.6	<i>Sistemas de control en lazo cerrado</i>	14
2.7	<i>Modelos matemáticos</i>	15
2.8	<i>Actuadores</i>	15
2.9	<i>Computadora interfaz, software y hardware</i>	16
2.10	<i>Sistemas de Control en tiempo continuo y discreto</i>	17
2.11	<i>Control No Lineal de Sistemas No Lineales</i>	19
2.11.1	El Método de la Linealización Extendida	20
2.11.2	Diseño de reguladores no lineales del tipo P, PI y PID mediante linealización extendida.....	20
CAPÍTULO 3		23
3. MODELADO MATEMÁTICO		23
3.1	<i>Descripción del escenario</i>	23
3.1.1	Péndulo Invertido Rotativo	25
3.1.2	Hardware y Software utilizado	26
3.2	<i>Modelamiento matemático del sistema PIR</i>	28
3.2.1	Modelo analítico del motor DC.....	28
3.2.1.1	Modelado del circuito eléctrico de la armadura	28
3.2.1.2	Descripción matemática del sistema mecánico	30
3.2.1.3	Análisis electromecánico del motor DC.....	32
3.2.2	Modelo analítico del brazo y del péndulo.....	36

3.2.3	Modelo no lineal del sistema PIR.....	40
3.2.4	Modelo lineal del sistema PIR.....	41
3.2.1.4	Péndulo en posición vertical arriba ($\alpha \approx 0 \text{ rad}$)	41
3.2.1.5	Péndulo en posición vertical abajo ($\alpha \approx \pi \text{ rad}$).....	44
3.2.5	Parámetros experimentales de la planta	46
3.2.5.1	Constantes del Motor DC	46
3.2.5.2	Parámetros constructivos del Péndulo.....	56
3.3	<i>Validación del modelo matemático del PIR</i>	66
3.3.1	Pruebas de validación del motor DC	66
3.3.2	Pruebas de validación del sistema PIR	70
CAPÍTULO 4		74
4. DISEÑO DE CONTROLADORES LINEALES.....		74
4.1	<i>Diseño del controlador PID Lineal para el control del péndulo</i>	74
4.2	<i>Diseño del controlador PI Lineal para el control del brazo</i>	84
CAPÍTULO 5		93
5. DISEÑO DEL CONTROLADOR NO LINEAL		93
5.1	<i>Diseño del controlador PI No lineal para el control del brazo.</i>	93
5.2	<i>Validación del controlador PI no lineal mediante simulación.</i>	104
CAPÍTULO 6		108
6. IMPLEMENTACIÓN DE PERIFÉRICOS.....		108
6.1	<i>Acondicionamiento de señales</i>	108
6.2	<i>Adquisición de datos</i>	109
6.3	<i>Interfaz de potencia</i>	113
CAPÍTULO 7		116
7. ANÁLISIS DE RESULTADOS		116

7.1	<i>Prueba de controladores en la planta real</i>	116
7.1.1	Evaluación del controlador PID lineal para el péndulo	116
7.1.2	Evaluación del controlador PI lineal para el brazo.....	121
7.1.3	Evaluación del controlador PI no lineal para el brazo.....	127
7.2	<i>Análisis y comparación de resultados</i>	132
	CONCLUSIONES Y RECOMENDACIONES	135
	BIBLIOGRAFÍA	138
	ANEXOS	140

ABREVIATURAS y SIMBOLOGÍA

CM	Centro de masa
clc	Limpia ventana de comandos
clear all	Borra variables del espacio de trabajo
DAQ	Tarjeta de adquisición de datos
DC	Corriente Directa
SISO	Sistema con una entrada y una salida
t_s	Tiempo de asentamiento
e(t)	Error
g	Aceleración de la gravedad
Hz	Hertz
P.O.	Porcentaje de sobrepaso
I/O	Entradas / Salidas
.mdl	Tipo de archivo para guardar datos en Simulink
MIMO	Sistema con múltiples entradas y múltiples salida
rtwintgt	Real time windows target
PCB	Placa de circuito impreso
PCI	Interconexión de componentes periféricos
PC	Computador personal
PID	Proporcional-Integral - Derivativo
PIR	Péndulo invertido rotativo

Set point	Valor de referencia para el controlador
Alpha	Variable de salida para ángulo del péndulo
Theta	Variable de salida para ángulo del brazo
α	Posición angular del péndulo
θ	Posición angular del brazo
u	Señal de control para el motor
τ_m	Torque del motor DC
J_p	Momento de inercia del péndulo
$C_1(s)$	Controlador PID lineal para equilibrar el péndulo
$C_2(s)$	Controlador PI lineal para posicionar el brazo
$C_{2NL}(s)$	Controlador PI no lineal para posicionar el brazo
$G_U(s)$	Función de transferencia del sistema
$z(t)$	Estado del compensador dinámico
y	Salida

ÍNDICE DE FIGURAS

Figura 1.1 Módulo de péndulo invertido	2
Figura 2.1 Lazo de control [5].....	10
Figura 2.2 Error [5]	11
Figura 2.3 Principio del transductor de rejilla [6].....	13
Figura 2.4 Sistemas en tiempo continuo [10]	17
Figura 2.5 Diagrama de bloque de un sistema de control digital [11].....	18
Figura 3.1 Descripción del PIR.....	24
Figura 3.2 Interfaz de comunicación del PIR.....	25
Figura 3.3 Encoder incremental serie 50S	26
Figura 3.4 Matlab 2010a.....	27
Figura 3.5 Esquema de un motor DC controlado por armadura [14].....	29
Figura 3.6 Diagrama de bloques del motor DC.	34
Figura 3.7 Diagrama de bloques simplificado del motor DC.....	35
Figura 3.8 Diagrama de cuerpo libre del sistema PIR	36

Figura 3.9 Diagrama de cuerpo libre del PIR en diferentes vistas.....	37
Figura 3.10 Acoplamiento entre motor y encoder incremental	46
Figura 3.11 Medición de kb utilizando módulo NI ELVIS II+.....	49
Figura 3.12 Medición de frecuencia utilizando osciloscopio PeakTeach	50
Figura 3.13 Medición de constante mecánica del motor.	55
Figura 3.14 Diseño del péndulo en SolidWorks.....	57
Figura 3.15 Parámetros del péndulo en obtenidos en SolidWorks.....	58
Figura 3.16 Medición de masa del péndulo.....	58
Figura 3.17 Obtención de centro de masa	60
Figura 3.18 Configuración del péndulo para oscilación amortiguada	61
Figura 3.19 Respuesta oscilatoria del péndulo.....	62
Figura 3.20 Gráfico de envolventes del péndulo	63
Figura 3.21 Relación entrada – salida de circuito de potencia	64
Figura 3.22 Línea de tendencia para el circuito de potencia	64
Figura 3.23 Simulación de modelo analítico del motor DC.....	66
Figura 3.24 Interface de entrada y salidas de datos del motor DC.....	67

Figura 3.25 Señal cuadrada aplicada al motor.	67
Figura 3.26 Comparación entre la respuesta implementada y simulada del motor DC.	68
Figura 3.27 Señal sinusoidal aplicada al motor.	68
Figura 3.28 Comparación entre la respuesta implementada y simulada del motor DC.	69
Figura 3.29 Señales aplicadas al sistema PIR en lazo abierto.....	70
Figura 3.30 Modelo simulado del sistema PIR en lazo abierto.....	70
Figura 3.31 Implementación del sistema PIR en lazo abierto.....	71
Figura 3.32 Comparación de señales de control aplicadas.....	71
Figura 3.33 Comparación de valores angulares obtenidos en el péndulo	72
Figura 3.34 Comparación de valores angulares obtenidos en el brazo.....	72
Figura 4.1 Respuesta del ángulo del péndulo ante una entrada paso	75
Figura 4.2 Respuesta del ángulo del brazo ante una entrada paso	76
Figura 4.3 Diagrama de bloques para control regulatorio del péndulo	77
Figura 4.4 Bloques simplificados para control regulatorio del péndulo.....	77

Figura 4.5 Estructura de control para equilibrio del péndulo	78
Figura 4.6 Lugar de las raíces del sistema PIR con controlador PID	79
Figura 4.7 Controlador PID para el péndulo	79
Figura 4.8 Rechazo ante una perturbación tipo impulso en el péndulo	80
Figura 4.9 Rechazo ante una perturbación tipo impulso en el péndulo	81
Figura 4.10 Control PID lineal del péndulo en Simulink	82
Figura 4.11 Respuesta del control PID del péndulo ante condiciones iniciales (Ángulo del brazo)	82
Figura 4.12 Respuesta del control PID del péndulo ante condiciones iniciales (Ángulo del péndulo)	83
Figura 4.13 Respuesta del control PID del péndulo ante condiciones iniciales (Señal de control)	83
Figura 4.14 Esquema de bloques para control de posición.....	85
Figura 4.15 Reducción de bloques para control de posición.....	85
Figura 4.16 Control PI de posición del brazo.....	86
Figura 4.17 Herramienta SISO para diseño de controlador PI del brazo.	87
Figura 4.18 Controlador PI ajustado para posicionamiento del brazo.....	87

Figura 4.19 Lugar de las raíces del controlador PI lineal del brazo.	88
Figura 4.20 Respuesta estimada del controlador PI lineal.	88
Figura 4.21 Respuesta estimada del controlador PI lineal desde Matlab.	90
Figura 4.22 Diagrama de los controladores lineales en Simulink.....	91
Figura 4.23 Respuesta del ángulo del péndulo ante cambios de referencia en el brazo.....	91
Figura 4.24 Respuesta del ángulo del brazo ante cambios de referencia. ...	92
Figura 4.25 Señal de referencia ante cambios de referencia en el brazo. ...	92
Figura 5.1 Diagrama de bloques para el control No Lineal.....	98
Figura 5.2 Esquema de control No Lineal para posición del brazo	99
Figura 5.3 Esquema de control No Lineal para posición del brazo en Simulink	104
Figura 5.4 Detalle del controlador PI No lineal realizado en Simulink	105
Figura 5.5 Respuesta del brazo utilizando controladores lineales y no lineales	106
Figura 5.6 Respuesta del péndulo utilizando controladores lineales y no lineales	106

Figura 5.7 Señal de control utilizada en controladores lineales y no lineales	107
Figura 6.1 Tarjeta de acondicionamiento de señales	109
Figura 6.2 Conexión de tarjeta de acondicionamiento de señales	109
Figura 6.3 Tarjeta Arduino Mega 2560	110
Figura 6.4 Conexiones de tarjeta Arduino Mega	110
Figura 6.5 Método X2 para lectura de codificadores ópticos [18].....	111
Figura 6.6 Bloque Conector CB-68LP	112
Figura 6.7 Bloque para lectura de encoders en Simulink	112
Figura 6.8 Algoritmo X2 para lectura de encoders en simulink	113
Figura 6.9 Bloque para interfaz con Saberthoot en simulink	114
Figura 6.10 Acondicionamiento de señales para interface de potencia	114
Figura 7.1 Implementación de control PID para el péndulo.....	117
Figura 7.2 Respuesta del péndulo ante controlador PID lineal implementado	117
Figura 7.3 Respuesta del brazo para control PID lineal en el péndulo.....	118

Figura 7.4 Salida del controlador PID lineal implementado en el péndulo invertido.....	118
Figura 7.5 Posición angular del péndulo, Real e Implementación.....	119
Figura 7.6 Posición angular del Brazo, Real e Implementación.	120
Figura 7.7 Salida del controlador PID lineal, Real e Implementación.....	120
Figura 7.8 Simulación del control PI para el brazo desde Simulink.....	121
Figura 7.9 Implementación del control PI para el brazo desde Simulink. ...	122
Figura 7.10 Comparación entre simulación e implementación ante una señal cuadrada de 2.5° de amplitud.....	122
Figura 7.11 Comparación entre simulación e implementación ante una señal cuadrada de 10° de amplitud.....	123
Figura 7.11 Comparación entre simulación e implementación ante una señal cuadrada de 10° de amplitud.....	124
Figura 7.12 Comparación entre simulación e implementación ante una señal cuadrada de 20° de amplitud.....	125
Figura 7.13 Caída del péndulo ante cambios de referencia excesivos	125
Figura 7.14 Simulación de Control PI no Lineal en simulink.....	127

Figura 7.15 Interface para Control PI no Lineal en simulink.....	128
Figura 7.16 Comparación entre simulación e implementación ante una señal cuadrada de 2.5° de amplitud.....	128
Figura 7.17 Comparación entre simulación e implementación ante una señal cuadrada de 10° de amplitud.....	129
Figura 7.18 Comparación entre simulación e implementación ante una señal cuadrada de 20° de amplitud.....	130
Figura 7.19 Comparación entre simulación e implementación ante una señal cuadrada de 27.5° de amplitud.....	131
Figura 7.20 Comparación entre controladores lineal y no lineal ante una señal cuadrada de 47.5° de amplitud.....	133
Figura 7.26 Comparación entre controladores lineal y no lineal ante una señal cuadrada de 71° de amplitud.....	134

ÍNDICE DE TABLAS

Tabla 1: Estimación experimental de resistencia R_a	47
Tabla 2: Estimación experimental de inductancia L_a	48
Tabla 3: Estimación experimental de constante k_b	51
Tabla 4: Estimación experimental de constante mecánica del motor	55
Tabla 5: Comparación de constantes de tiempo	56
Tabla 6: Parámetros del sistema PIR.....	65
Tabla 7: Cálculo del error absoluto y relativo RMS para el motor DC	69
Tabla 9: Cálculo del error absoluto y relativo RMS para el motor DC	73
Tabla 8: Parámetros K_1 y K_2 para controlador PI del brazo utilizando método de Ziegler-Nichols [12]	103
Tabla10: Respuesta del sistema a diferentes referencias del ángulo del brazo aplicando control PI lineal.....	126
Tabla11: Respuesta del sistema a diferentes referencias de amplitud en el brazo aplicando control PI no lineal.....	132

INTRODUCCIÓN

El Péndulo Invertido Rotativo (PIR) funciona como un sub-actuado que permite controlar dos grados de libertad rotacionales llamados brazo y péndulo, utilizando solamente un actuador. El movimiento del brazo se realiza en un plano horizontal girando alrededor de un eje perpendicular al plano, el péndulo se encuentra ubicado en un extremo del brazo y su eje giro es colineal al eje axial del brazo y su movimiento se realiza en un plano perpendicular al de éste último.

En algunas ocasiones, la subactuación puede ser causada por fallas en los actuadores. El desarrollo de mecanismos que pueden realizar tareas complejas con un número reducido de actuadores es un asunto de gran interés, puesto que implica reducción de peso y de costos.

El sistema PIR es del tipo lazo abierto inestable pues se necesita un tipo de control para mantener al péndulo en la posición vertical arriba. Cabe mencionar que este tipo de planta es conocido como un típico sistema no lineal y ha sido estudiado ampliamente debido a sus características, como son: no linealidad, inestabilidad, fase no mínima, etc., lo cual hace de tal sistema muy conveniente para la prueba de técnicas y esquemas de control.

Los controladores PID son suficientes para resolver el problema de control de muchas aplicaciones en la industria, particularmente cuando la dinámica del proceso lo permite y los requerimientos de desempeño son modestos.

La eficiencia de un controlador PID lineal depende del ajuste de los parámetros del controlador para que trabajen alrededor de un punto de operación, pero esto podría mejorar al ampliar su rango de acción como es el enfoque de los controladores de tipo no lineal.

Considerando lo antes mencionado, el presente trabajo propone diseñar y evaluar dos algoritmos de control, como son: PI lineal y PI no lineal, en el sistema PIR.

Este trabajo escrito está compuesto por siete capítulos: El capítulo I analiza el estado actual de la problemática en la estabilización de sistemas tipo péndulo, se plantea una solución y objetivos a alcanzar a través de métodos y técnicas de investigación a aplicar.

El capítulo II presenta un estudio bibliográfico sobre trabajos previamente realizados, continuando con la descripción de las principales herramientas matemáticas y programas que se emplearan para resolver el problema de estabilización.

El capítulo III corresponde al análisis del sistema con la finalidad de obtener el modelo matemático del mismo, seguido a esto podemos encontrar el

procedimiento experimental que se utilizó para obtener los parámetros constructivos de la planta, culminando con las pruebas de validación respectivas.

Luego se realiza el diseño de los controladores PI lineal y PI no lineal en los Capitulo IV y V respectivamente que serán aplicados al sistema PIR en base a especificaciones de funcionamiento previamente analizadas. En el capítulo VI se da a conocer el proceso realizado para la implementación del mismo considerando el hardware y firmware empleado.

Seguidamente, en el capítulo VII se evalúa el desempeño de los controladores sobre el sistema PIR. Finalmente, se presentan las conclusiones, recomendaciones y futuras trabajos que podrían realizarse en la planta.

CAPÍTULO 1

1. ANÁLISIS DEL PROBLEMA

1.1 Planteamiento del problema

Actualmente la Universidad Politécnica Salesiana sede Guayaquil se encuentra en proceso de fortalecimiento de sus laboratorios, uno de estos es el laboratorio de Control que posee módulos comerciales adquiridos en su debido tiempo. Dichos módulos incorporan controladores clásicos diseñados para prácticas de pregrado.



Figura 1.1 Módulo de péndulo invertido

Esta tesis se basa en diseñar e implementar un prototipo de péndulo invertido rotativo para evaluar un algoritmo de control aplicando técnicas no lineales mediante el modelado del sistema y así poder utilizar el mismo para fines didácticos e investigativos.

Este tipo de sistemas tiene aplicaciones académicas, pues en el mismo se pueden implementar diversos controladores y verificar distintos controladores, además en los últimos años, existe una creciente atención en los sistemas mecánicos subactuados, (sistemas mecánicos que poseen menos actuadores que sus grados de libertad).

El control de tales sistemas es difícil, pero usando técnicas adecuadas se permite la reducción del número de actuadores, el costo y el peso de los sistemas. De ahí que este problema es un tema de investigación interesante en la industria para fines de diseño.

El sistema de péndulo invertido rotativo es un sistema robótico subactuado de dos eslabones, cuyo primer eslabón (brazo) es accionado por un motor con el objetivo de equilibrar el segundo eslabón (péndulo) en la posición invertida.

Otro enfoque de interés en este problema puede asociarse a la inherente dificultad de controlar un proceso altamente no lineal como éste, lo que constituye un reto importante para la comunidad científica.

1.2 Solución Propuesta

Se analizará el modelo matemático y de forma experimental se obtendrán los parámetros constructivos del mismo. Para controlar el sistema, se propone como técnicas de control, un controlador PID lineal para estabilizar el péndulo junto a un controlador PI lineal y PI no lineal para regular la posición del brazo.

Cabe mencionar que se realizará énfasis en mantener la posición vertical del péndulo verificando su efectividad ante perturbaciones externas.

1.3 Delimitación del problema

Este trabajo se enfoca en el diseño, implementación y evaluación mediante simulación y operación real de los algoritmos de control PI lineal y PI no lineal para el péndulo invertido rotativo existente en la Universidad Politécnica Salesiana sede Guayaquil ubicado en el laboratorio de control automático.

1.4 Objetivos

1.4.1 Objetivo General

Estabilizar el péndulo invertido mediante la implementación de un controlador PID lineal, aplicando un controlador PI lineal y no lineal para la regulación de posición del brazo.

1.4.2 Objetivos Específicos

1. Determinar el modelo matemático del sistema por modelación.
2. Diseñar el controlador PID lineal para estabilizar el péndulo.
3. Diseñar el controlador PI no lineal para controlar el movimiento del brazo.
4. Evaluar el comportamiento del controlador no lineal frente a un PI lineal mediante la simulación y experimentación.

1.5 Metodología

Para la realización del tema propuesto se plantea utilizar en su análisis teórico los siguientes métodos:

- **Análisis- Síntesis:** Este permitirá descomponer el proceso o sistema en sus partes básicas a ser estudiadas, para luego encontrar las características totales del sistema.
- **Hipotético-Deductivo:** Debido a que se está empleando la estructura de investigación, se tiene lo siguiente: Planteamiento del problema, solución, delimitación etc.

Y en su fase experimental se empleará:

- **Observación:** Permite analizar la respuesta y comportamiento temporal del péndulo invertido frente a los algoritmos aplicados.
- **Medición:** Porque se tomará información de datos numéricos a través de técnicas, instrumentos y condiciones iniciales.
- **Experimentación:** Permitirá evaluar el desempeño del sistema ante los algoritmos de control aplicados a través de la medición de sus variables medibles que posea el mismo.

CAPÍTULO 2

2. ANÁLISIS DE HERRAMIENTAS Y CONOCIMIENTOS DISPONIBLES

2.1 Antecedentes

El análisis de estabilidad del péndulo invertido es un problema clásico que se puede encontrar en diversas fuentes bibliográficas y en la actualidad existen diversos trabajos relacionados a este tema, los cuales muestran resultados a través de simulaciones o de implementación.

Entre las técnicas comúnmente aplicadas se podrían mencionar controladores clásicos y modernos [1], control óptimo [2], etc.

Muchos trabajos de investigación comparan el controlador PID lineal con otros tipos de controladores considerando la linealización en un solo punto de operación, pero pocos se enfocan en la aplicación de controladores no lineales.

Una de las características más notable que presenta este tipo de sistema es la poca linealidad en el modelo matemático, esto ha llevado a que se pueda aplicar diversas técnicas de control para verificar su efectividad y así analizar los resultados obtenidos.

El objetivo principal de este proyecto es mantener en equilibrado el péndulo invertido rotativo en su posición vertical superior considerando como actuador un motor DC y a su vez realizar una comparación entre la respuesta de un controlador lineal y no lineal en la misma planta.

Los péndulos invertidos constituyen un banco de pruebas completo e interesante para la ingeniería de control. Uno de los más estudiados de esta familia de artefactos es el denominado péndulo invertido sobre un vehículo. Desde los años 70 se han realizado varios proyectos con péndulos invertidos.

Un investigador líder en esta área es el Profesor Furuta, quien desde entonces ha realizado notables aportes teóricos y experimentales concernientes a este problema de control [3].

A partir de ahora y durante todo el presente trabajo se usará el término PIR para referirnos a la planta mencionada.

Los controladores bajo estudio se han seleccionado teniendo en cuenta que la naturaleza del sistema PIR es del tipo no lineal e inestable. Esto implica que para utilizar métodos de control lineal se hace necesario linealizar el sistema en torno a un punto de trabajo, mientras que los métodos de control no lineal pueden trabajar directamente con sistemas no lineales. En efecto, la teoría indica que para desviaciones grandes respecto al punto de operación el controlador lineal pierde su efectividad, mientras que el controlador no lineal debe trabajar con la misma efectividad independientemente del punto de trabajo.

2.2 Sistemas sub-actuados

Los sistemas subactuados son sistemas mecánicos con un menor número de variables de control de los necesarios en la configuración. En los últimos años ha existido gran interés entre los investigadores en el control de los sistemas mecánicos subactuados debido a sus amplias aplicaciones. Muchos sistemas de control de la vida real, incluyendo

aviones, naves espaciales, helicópteros, vehículos submarinos, buques de superficie, robots móviles, robots caminantes, y robots articulados son ejemplos de sistemas subactuados. [4]

En el caso del péndulo invertido rotativo se controlará dos variables angulares a través de un voltaje aplicado a un motor DC, lo cual implica tener un sistema sobreactuado que minimiza la cantidad de actuadores a utilizar.

2.3 Control de procesos

El término proceso utilizado en “control de procesos” o “procesos industriales”, se refiere a cambiar o refinar materias primas para lograr un producto final.

La materia prima, que puede o no cambiar de estado físico durante el proceso, es transferida, medida, mezclada, calentada, enfriada, filtrada, almacenada o manipulada de alguna manera para producir el producto final. [5]

Ejemplos de procesos industriales

- Industrias químicas
- Petróleo y gas
- Alimentos y bebidas
- Productos farmacéuticos

- Tratamientos de agua
- Generación de energía, etc.

El controlar un proceso, se refiere a como se regulan, variables inherentes al mismo, para:

- Reducir la variabilidad del producto final
- Incrementar la eficiencia
- Reducir impacto ambiental
- Mantener el proceso dentro de los límites de seguridad que corresponda.

2.3.1 El lazo de control

Un lazo de control requiere la ocurrencia de tres tareas:

- Medida
- Comparación
- Ajuste

Estas tareas se indican en la Figura.2.1.

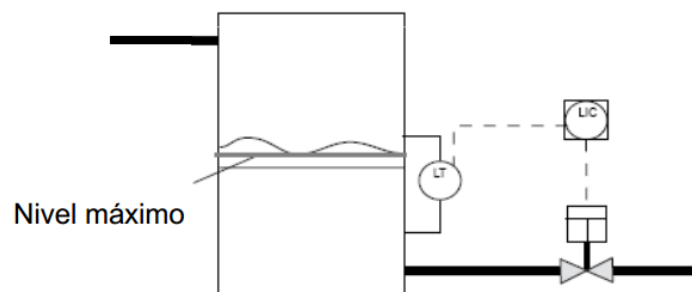


Figura 2.1 Lazo de control [5]

2.3.2 Variables de proceso

Son aquellas que pueden cambiar las condiciones de un proceso

- **Setpoint**

Valor al que se desea mantener una variable de proceso

- **Variable medida**

Es aquella que se desea mantener estable

- **Variable manipulada**

Es aquella que varía para mantener constante la variable medida

- **Error**

Es la diferencia entre la variable medida y el setpoint. Puede ser +/-.

Está compuesto por: Magnitud, duración y velocidad de variación como se indica en la Figura.2.2.

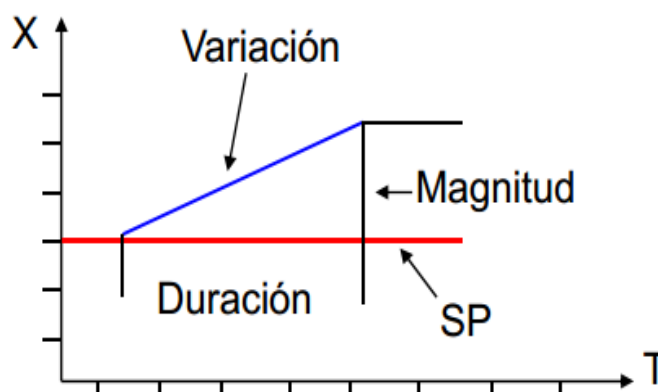


Figura 2.2 Error [5]

- **Offset**

Es una variación constante de la variable de **proceso**, respecto del setpoint.

- **Variación de carga**

Es un cambio no deseado en algún factor que pueda afectar la variable de proceso.

- **Algoritmo de control**

Es la expresión matemática de una función de control

2.4 Codificadores incrementales (Encoders)

Los codificadores incrementales se utilizan para la medición angular y constan de un eje de detección unido a un disco que se divide en un número igual de sectores en la circunferencia. En el tipo lineal de codificadores, hay segmentos iguales a lo largo de la longitud del recorrido. Las lecturas son detectadas por el contacto eléctrico directo con una escobilla u ópticamente usando rendijas o rejillas ópticas. Mientras más rejillas, más alta es la resolución. Esta especificación se expresa como pulsos por revolución, lo cual es un factor importante en la selección de codificador.

Los codificadores incrementales son muy útiles para medir la rotación del eje y consisten principalmente de tres componentes: una fuente de luz, una rueda codificada, y un sensor fotoeléctrico.

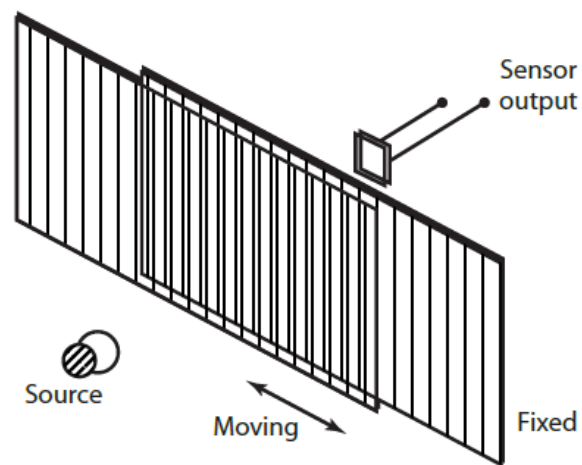


Figura 2.3 Principio del transductor de rejilla [6]

La figura 2.3 muestra un sistema de medición con encoder que utiliza rejillas de transmisión. Como la rejilla móvil se desplace con respecto a una rejilla fija, los impulsos se cuentan para proporcionar información de posición. [6]

2.5 Sistemas de control en lazo abierto

Los sistemas en los cuales la salida no tiene efecto sobre la acción de control se denominan sistemas de control en lazo abierto. En otras palabras, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es una lavadora. El remojo, el lavado y el centrifugado en la lavadora operan con una base de tiempo. La máquina no mide la señal de salida, que es la limpieza de la ropa.

En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Así, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se usa si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas. Es evidente que estos sistemas no son de control realimentado.

Obsérvese que cualquier sistema de control que opere con una base de tiempo está en lazo abierto. Por ejemplo, el control de tráfico mediante señales operadas con una base de tiempo es otro ejemplo de control en lazo abierto. [7]

2.6 Sistemas de control en lazo cerrado

Los sistemas de control realimentados se denominan también sistemas de control en lazo cerrado. En la práctica, los términos control realimentado y control en lazo cerrado se usan indistintamente.

En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales),

con el fin de reducir el error y llevar la salida del sistema a un valor deseado. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema. [7]

2.7 Modelos matemáticos

Los modelos matemáticos pueden adoptar muchas formas distintas.

Dependiendo del sistema del que se trate y de las circunstancias específicas, un modelo matemático puede ser más conveniente que otros. Por ejemplo, en problemas de control óptimo, es provechoso usar representaciones en el espacio de estados. En cambio, para los análisis de la respuesta transitoria o de la respuesta en frecuencia de sistemas lineales con una entrada y una salida invariantes en el tiempo, la representación mediante la función de transferencia puede ser más conveniente que cualquier otra.

Una vez obtenido un modelo matemático de un sistema, se usan diversos recursos analíticos, así como computadoras para estudiarlo y sintetizarlo. [7]

2.8 Actuadores

Son dispositivos de potencia que producen la entrada para la planta de acuerdo con la señal de control, a fin de que la señal de salida se aproxime a la señal de entrada de referencia. [7]

2.9 Computadora interfaz, software y hardware

En la actualidad el diseño de los sistemas de control se realiza con la asistencia de computadoras con el objetivo de alcanzar el desempeño óptimo y disminuir el tiempo empleado antes de la puesta en marcha.

Entre las ventajas más notorias se podría mencionar:

- Menor consumo de energía
- Capacidad de realizar simulaciones
- Minimización de costos
- Mayor eficiencia y eficacia

Debido a que el diseño electrónico del computador no permite la conexión directa con los diversos actuadores que tiene la planta, se vuelve necesario el uso de tarjetas electrónicas que permitan realizar el acoplamiento o transferencia de datos desde el sistema de control – PC y viceversa.

Entre las tarjetas de adquisición de datos (DAQ) comerciales que permiten esta funcionalidad tenemos:

- Tarjetas de adquisición PCI -6220, PCI -6221, PCI-1711, etc.
- Tarjetas de adquisición PCIe -6321, PCIe -6323, etc.
- Tarjetas de adquisición NI USB-6000, NI USB-6008, NI-UB-6009, etc.

La selección adecuada de la DAQ dependerá de ciertos parámetros tales como: números de entradas/salidas, tipo de entrada/salida, frecuencia de muestreo, resolución, niveles de voltaje, etc.

Estos modelos de DAQ comerciales requieren de algún software para funcionar y dependiendo del tipo de tarjetas estas pueden ser utilizadas con LabVIEW, MATLAB o con ambos. Si se utilizan con LabVIEW se requiere el driver NI-DAQmx [8], en el caso de MATLAB se debe instalar el Data Acquisition Toolbox [9].

2.10 Sistemas de Control en tiempo continuo y discreto

Cuando una señal es procesada en un sistema en intervalos continuos de tiempo, el sistema es llamado Tiempo Continuo, donde se tiene una entrada $u(t)$ y una salida $y(t)$ como se indica en la Figura.2.4.

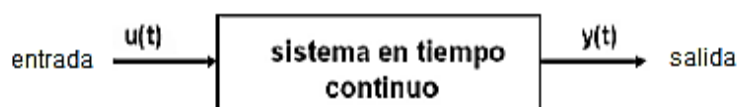


Figura 2.4 Sistemas en tiempo continuo [10]

El sistema de mayor importancia práctica es el lineal invariante en el tiempo LTI por sus siglas en inglés. Los sistemas pueden funcionar con controles de lazo abierto o cerrado.

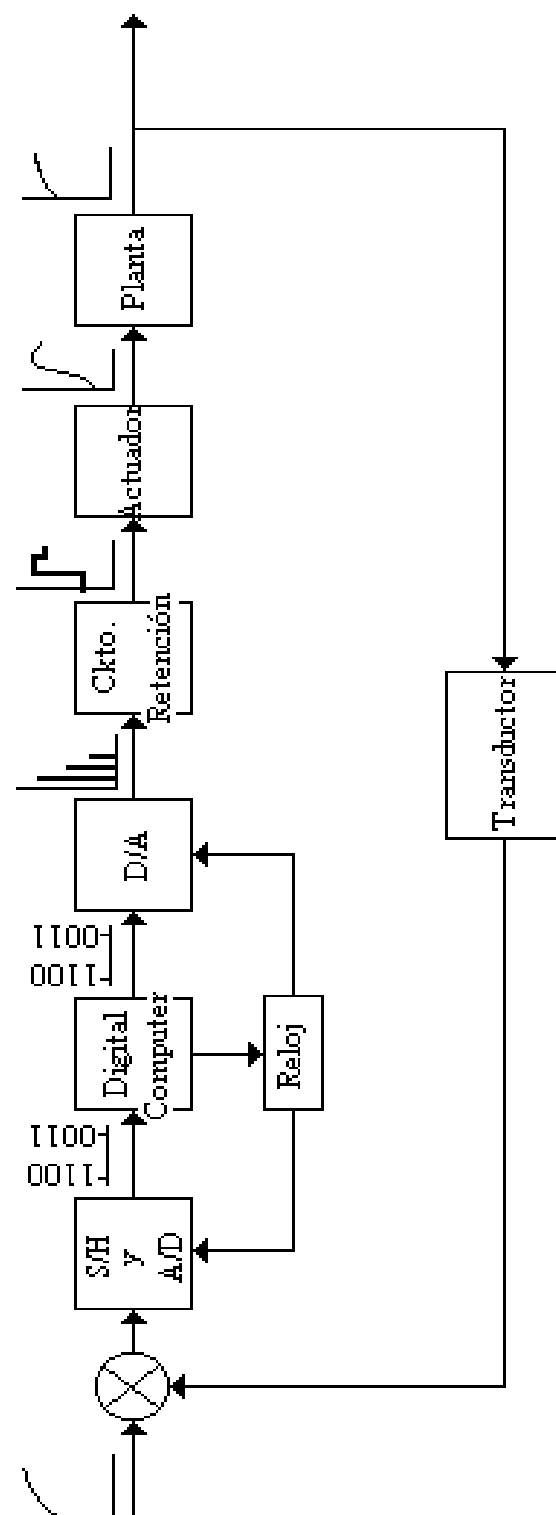


Figura 2.5 Diagrama de bloque de un sistema de control digital [11]

Según en [11] las señales de un sistema de control en tiempo discreto están en forma de datos muestreados o en forma digital y se describen mediante ecuaciones en diferencias. La mayoría de los procesos o plantas involucran señales de tiempo continuo. Si los sistemas incluyen controladores digitales, se hace necesaria la conversión de señales de analógico al digital y de digital a analógico como se muestra en la Figura 2.5.

2.11 Control No Lineal de Sistemas No Lineales

Desde los años 40, el uso de la técnica de linealización aproximada ha permitido resolver una gran diversidad de problemas relacionados con el control de sistemas reales. Sin embargo, esta estrategia presenta algunas restricciones fundamentales en su aplicación. Para resolver estos inconvenientes, se han propuesto diferentes extensiones del método, estas extensiones incluyen aspectos teóricos fundamentales relacionados con diversas áreas de la matemática, tales como la teoría de operadores, el análisis funcional, el álgebra y la geometría diferencial, por ejemplo.

Básicamente, existen dos métodos para ilustrar el diseño de controladores no lineales para sistemas no lineales: el método de la linealización extendida, propuesta por Wilson Rugh y sus colaboradores, y el método de la linealización exacta, propuesto, entre otros, por A.

Isidori, cuyos orígenes se pueden encontrar en los trabajos de H. J. Sussmann y V. J. Jurdjevic, R. Hermann y A. J. Krener, etc. [12]

2.11.1 El Método de la Linealización Extendida

Este método utiliza la linealización aproximada tradicional como herramienta fundamental. El modelo utilizado para generar el diseño del controlador es un modelo obtenido sobre la base de los primeros términos (de carácter lineal) de una expansión en serie de Taylor de un sistema no lineal que opera en la vecindad de un punto de operación constante. En este caso, la diferencia radica en que en lugar de obtener un modelo linealizado para un punto en particular, se procede a obtener el modelo parametrizado de carácter lineal para un punto de operación genérico operación genérico $(X(U), Y(U), U)$ de naturaleza arbitraria [12]

2.11.2 Diseño de reguladores no lineales del tipo P, PI y PID mediante linealización extendida.

Recordemos la forma del regulador PID:

$$u_{\delta} = K_1(U)e_{\delta} + K_2(U)\int_0^t e_{\delta}(\sigma)d\sigma + K_3(U)\frac{de_{\delta}}{dt} \quad (2.1)$$

Siguiendo el método de la linealización extendida, se propone a partir de la expresión (2.1) un *compensador no lineal general del tipo PID*. Aplicaremos la linealización aproximada a este controlador, con el objeto de identificarlo al diseño lineal llevado a cabo precisamente sobre la

familia de sistemas linealizados. En este caso, la generalización no lineal del controlador PID asume la forma siguiente presentada en la ecuación (2.2):

$$\begin{aligned}\dot{z}(t) &= k_2(z(t))e(t) \\ u(t) &= z(t) + k_1(z(t))e(t) + k_3(z(t))\frac{de(t)}{dt}\end{aligned}\quad (2.2)$$

Donde la variable $z(t)$ juega el papel de “estado” del compensador dinámico no lineal. Veamos. La señal de control sintetizada por el compensador anterior adopta entonces la forma general.

$$u(t) = k_1(z(t))e(t) + \int_0^t k_2(z(\sigma))e(\sigma)d\sigma + k_3(z(t))\frac{de(t)}{dt}\quad (2.3)$$

La cual en efecto representa una generalización, de tipo no lineal, del controlador clásico PID. El error $e(t)$ está dado por $e = y_{ref} - y = 0 - y = -y$. Note que el punto de equilibrio de la estrategia de control no lineal propuesta está dado, evidentemente, por $e = 0$ y $z(U) = U$.

De esta forma, la linealización del regulador no lineal (2.1), alrededor de su punto de equilibrio, resulta en:

$$\begin{aligned}\dot{z}_\delta &= k_2(U)e_\delta \\ u_\delta &= z_\delta + k_1(U)e_\delta(t) + k_3(U)\frac{de_\delta(t)}{dt}\end{aligned}\quad (2.4)$$

Se desprende inmediatamente, de la comparación de este compensador linealizado con el compensador lineal PID diseñado anteriormente, que:

$$\begin{aligned}k_1(z)|_{z=U} &= K_1(U) \\k_2(z)|_{z=U} &= K_2(U) \\k_3(z)|_{z=U} &= K_3(U)\end{aligned}\tag{2.5}$$

En este caso, las ganancias no lineales se obtienen directamente, de manera prácticamente trivial, a partir de las siguientes relaciones:

$$\begin{aligned}k_1(z) &= K_1(z) \\k_2(z) &= K_2(z) \\k_3(z) &= K_3(z)\end{aligned}\tag{2.6}$$

Lo cual facilita enormemente la búsqueda del controlador no lineal [12].

CAPÍTULO 3

3. MODELADO MATEMÁTICO

3.1 Descripción del escenario

El péndulo invertido rotativo (PIR) es un dispositivo físico que consiste en una barra cilíndrica con libertad de oscilar alrededor de un pivote fijo. Este pivote es montado sobre un brazo siguiendo un movimiento angular.

El propósito final es mantener al péndulo en una posición perpendicular equilibrada de forma autónoma. La fuerza aplicada como acción de control es generada por el par que el motor transmite al brazo utilizando

un controlador para modificar el voltaje de armadura y tomando lectura del ángulo de inclinación del péndulo.

Como se puede observar en la Figura. 3.1 la planta puede ser representada como un sistema tipo SIMO, donde la entrada corresponde al voltaje de armadura del motor DC (V_a) y las salidas corresponden a las posiciones angulares del brazo y del péndulo (θ, α) respectivamente. En la figura mostrada se indica la nomenclatura de las variables a utilizar

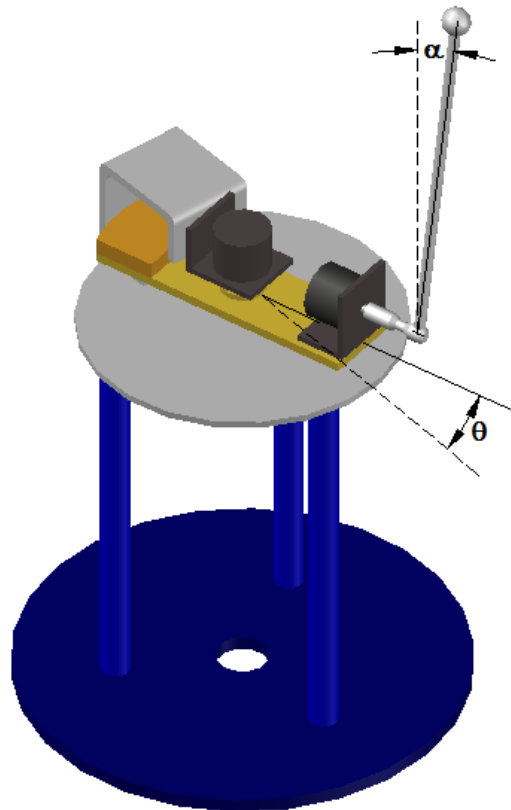


Figura 3.1 Descripción del PIR

3.1.1 Péndulo Invertido Rotativo

El sistema PIR es controlado desde una PC, por tanto, el sistema contiene el hardware y software que se puede montar e instalar fácilmente en un laboratorio.

Con respecto al hardware, el sistema está compuesto por: panel de marcha y paro de emergencia, fuente de alimentación, tarjeta de adquisición de datos Arduino Mega (ver ANEXO 1), controlador de motor Saberthoot 2x32 (ver ANEXO 2) e interfaz a un PC a través de la tarjeta PCI-1711 (ver ANEXO 3) y bloque conector CB-68LP (ver ANEXO 4) como se indica en la Figura. 3.2.

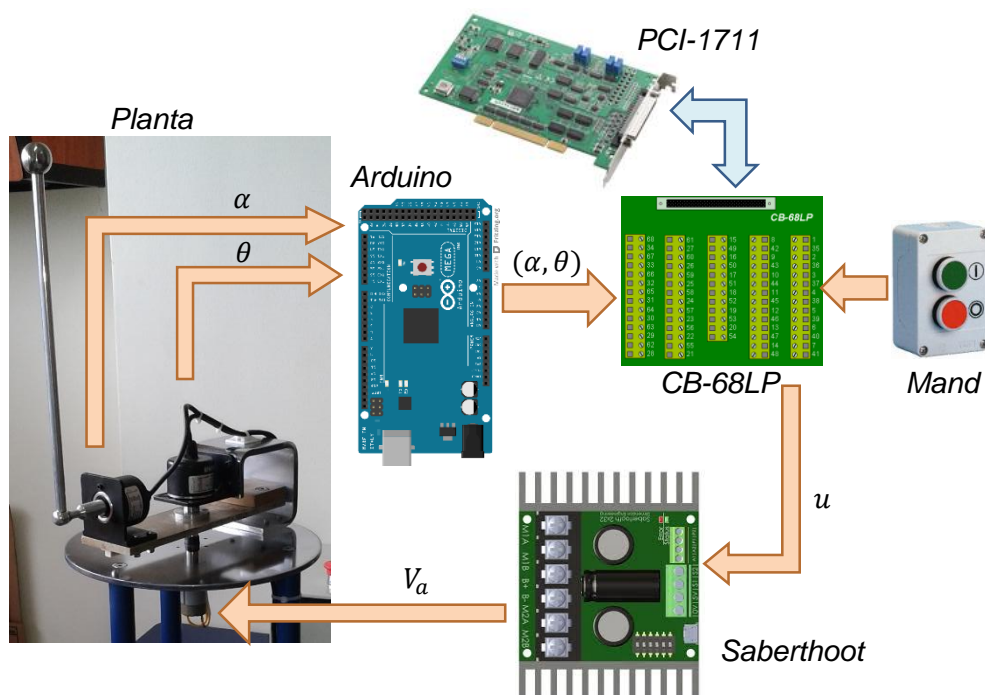


Figura 3.2 Interfaz de comunicación del PIR

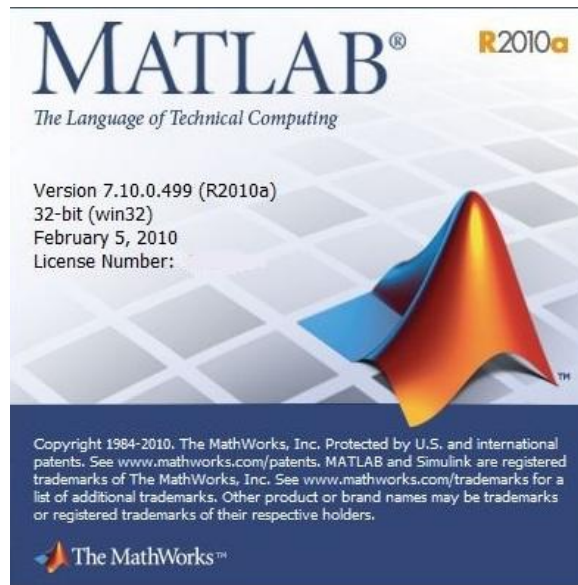


Figura 3.4 Matlab 2010a

- Para llevar a cabo el manejo del PIR se requieren el toolbox correspondiente a Sistemas de Control.
- Activar Real-Time Windows Target para esto se debe abrir Matlab e instalar el Kernel escribiendo en la ventana de comandos [13]:

`>> rtwintgt -install` (note el espacio antes del "-")

- Pulse la tecla "enter" para ejecutar el comando anterior. Del mismo modo, si se han instalado varios compiladores de C, deben ser deseleccionados escribiendo:

`>> mex -setup` (note el espacio antes del "-")

- Pulse la tecla "enter" para ejecutar el comando anterior.

- Al siguiente aviso seleccionar el "0". Esto asegura que el sistema sólo intente compilar modelos con el compilador por defecto.

3.2 Modelamiento matemático del sistema PIR

Para el desarrollo de este proyecto es importante conocer el modelo de planta, ya que a través de este, se puede describir el funcionamiento del sistema ante ciertas condiciones de operación. Esto permitirá analizar la dinámica y diseñar controladores en un tiempo menor ya que no se tiene que realizar ninguna implementación hasta haber realizado la mayor cantidad de pruebas.

Para tener un mayor grado de comprensión en el desarrollo, se empezará con el modelado del motor DC y posterior a esto se analizará el diseño mecánico.

3.2.1 Modelo analítico del motor DC

El modelo matemático del motor se ha obtenido a través de formulación matemática y considerando varios experimentos para conocer los diversos parámetros que lo conforman.

3.2.1.1 Modelado del circuito eléctrico de la armadura

La Figura 3.5 muestra el esquema de un motor de DC, donde se puede apreciar que se trata de un circuito eléctrico constituido por una resistencia y un inductor, los cuales son llamados R_a y L_a

respectivamente, por lo tanto el funcionamiento del mismo se puede determinar por medio de un circuito RL de corriente directa.

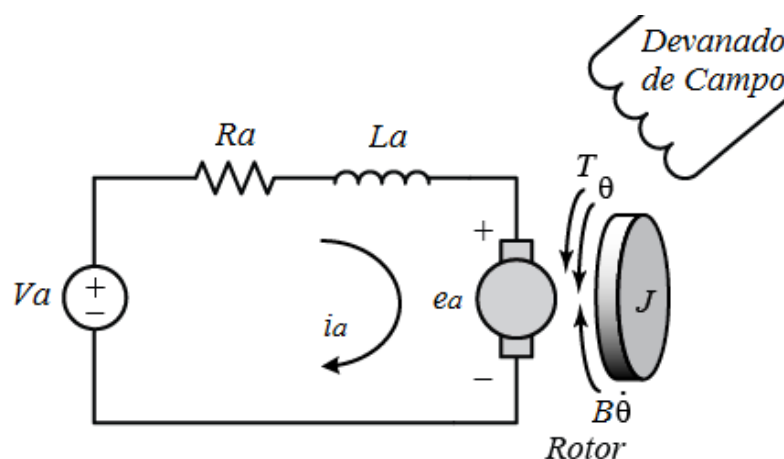


Figura 3.5 Esquema de un motor DC controlado por armadura [14].

El comportamiento del circuito RL se encuentra desde el arranque del motor hasta que el motor alcanza su estado estacionario, en este transcurso es donde se realiza el análisis transitorio del sistema.

Mediante la Ley de voltajes de Kirchhoff y teoría de circuitos eléctricos, las ecuaciones para el equivalente de un motor DC son representadas como:

$$V_a = V_{R_a} + V_{L_a} + e_a$$

$$V_a = L_a \frac{di_a}{dt} + i_a R_a + e_a \quad (3.1)$$

Donde:

V_a : Voltaje en las terminales de la armadura.

$L_a \frac{di_a}{dt}$: Caída de voltaje en la inductancia.

$i_a R_a$: Caída de voltaje en la resistencia.

e_a : Fuerza contra electromotriz desarrollada en la armadura (FCEM).

La ecuación (3.1) relaciona la FCEM, puesto que al proporcionar corriente en el circuito produce una FCEM en el inducido a causa del giro de la máquina y los cortes de líneas de flujo magnético. La polaridad de la FCEM es contraria al voltaje del inducido.

3.2.1.2 Descripción matemática del sistema mecánico

Cuando una máquina rotatoria se utiliza como motor, proporciona energía eléctrica a sus bobinados y las fuerzas electromagnéticas desarrollan un par mecánico.

Utilizando las leyes de Newton se puede describir este par mecánico, sabiendo que el par equivalente que actúa sobre un eje siempre es cero, interpretando todas las componentes, de acuerdo a una dirección de referencia común.

“Entonces el equilibrio o balance del par de la armadura que acelera una carga en la dirección del giro del rotor, sometida a la pérdida de par

debido al rozamiento mecánico, por la fricción del aire y la pérdida en el hierro, es el siguiente" [15].

$$\sum T = T_{mec} - T_{acel} - T_{perd} - T_L = 0$$

$$T_{mec} = T_{acel} + T_{perd} + T_L$$

$$T_{mec} = J \frac{d\omega}{dt} + B\omega + T_L \quad (3.2)$$

Donde:

$J \frac{d\omega}{dt}$: Par mecánico producido por el momento de inercia J , éste término

desaparece cuando la aceleración es cero y se presenta en condiciones de estado permanente.

$B\omega$: Par mecánico producido por el coeficiente de fricción viscosa B desarrollado en los valeros o chumaceras, así como en cualquier otra zona donde existe algún rozamiento, este término permanece constante si ω lo es también.

T_L : Par mecánico producido por la carga externa aplicada al rotor de la máquina.

El momento de inercia J está compuesto por el momento de inercia J_m al hacer girar el inducido del motor y el momento de inercia de la carga J_L , ya que la carga gira a la misma velocidad que el motor.

$$J = J_m + J_L \quad (3.3)$$

De la misma manera el coeficiente de fricción viscosa B es el resultado del rozamiento de las partes mecánicas B_m y el rozamiento de la carga conectada al eje del motor B_L .

$$B = B_m + B_L \quad (3.4)$$

3.2.1.3 Análisis electromecánico del motor DC

En base a las relaciones antes descritas, se pueden realizar las ecuaciones que describen el modelado del sistema como:

$$T_{elec} = k_t i_a \quad (3.5)$$

$$e_a = k_b \omega \quad (3.6)$$

$$V_a = L_a \frac{di_a}{dt} + i_a R_a + e_a \quad (3.7)$$

$$T_{mec} = J \frac{d\omega}{dt} + B\omega + T_L \quad (3.8)$$

Donde:

T_{elec} : Par electromagnético generado (N.m)

i_a : Corriente de armadura(A).

e_a : Fuerza contraelectromotriz inducida (V).

ω : Velocidad angular del motor (rad/s).

V_a : Voltaje aplicado en terminales de la armadura (V).

T_{mec} : Par mecánico producido en el motor (N.m).

T_L : Par de carga (N.m).

R_a : Resistencia de devanado de armadura (Ω).

Por la ley de Newton, se sabe que en cualquier máquina giratoria se presenta un balance del par electromagnético T_{elec} con el par mecánico T_{mec} con lo que se obtiene la igualdad:

$$T_{elec} = T_{mec} = T_m \quad (3.9)$$

Las ecuaciones desarrolladas para el comportamiento del sistema, se encuentran en función del tiempo para el análisis del sistema dinámico. Cabe señalar que los valores de la constante de par electromagnético k_t y la constante de la FCEM k_b , se encuentran estrechamente relacionados. Una manera de demostrar esta relación es sabiendo que la potencia en estado permanente [15] generada en la armadura es:

$$P_{elec} = e_a i_a \quad (3.10)$$

La potencia mecánica en estado estacionario es descrita por:

$$P_{mec} = T_{elec} \omega \quad (3.11)$$

Ya que en las máquinas eléctricas debe haber un equilibrio entre la potencia eléctrica y la potencia mecánica:

$$P_{elec} = P_{mec} \quad (3.12)$$

$$e_a i_a = T_{elec} \omega \quad (3.13)$$

Sustituyendo el valor de las ecuaciones (3.5) y (3.6) en la ecuación (3.13).

$$k_b \omega i_a = k_t i_a \omega$$

De donde se obtiene:

$$k_b = k_t \quad (3.14)$$

Para llegar a obtener una función de transferencia del motor, debemos llevar las ecuaciones previas al dominio de la variable compleja "s" utilizando la transformada de Laplace.

$$T_m(s) = k_t I_a(s) \quad (3.15)$$

$$E_a(s) = k_b \Omega(s) \quad (3.16)$$

$$V_a(s) = L_a s I_a(s) + R_a I_a(s) + E_a(s) \quad (3.17)$$

$$T_m(s) = J s \Omega(s) + B \Omega(s) + T_L(s) \quad (3.18)$$

Dadas las expresiones (3.15), (3.16), (3.17) y (3.18) se obtiene el diagrama de bloques que se muestra a continuación:

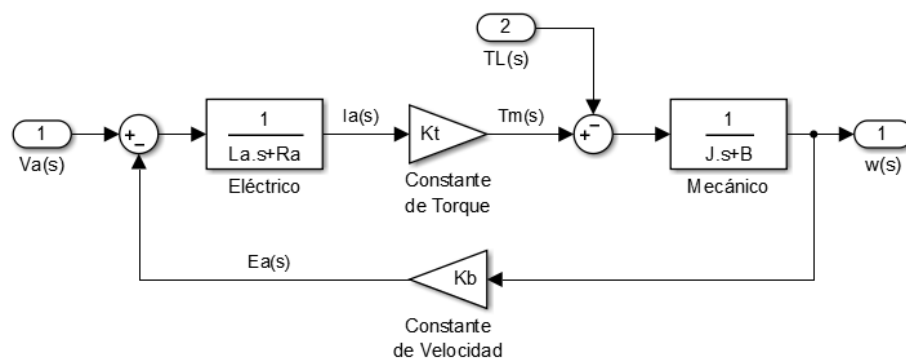


Figura 3.6 Diagrama de bloques del motor DC.

En la mayoría de los casos se puede apreciar que la constante de tiempo del circuito eléctrico de armadura es muy pequeña comparada con la constante de tiempo del sistema mecánico, lo cual permite omitir la inductancia con fines de reducción de términos en el modelo matemático.

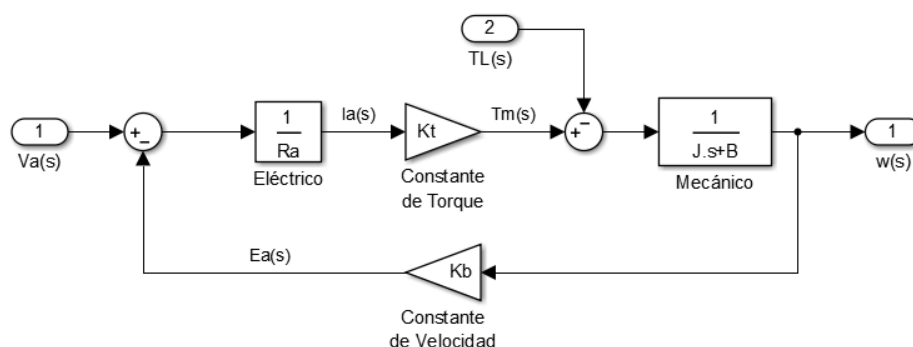


Figura 3.7 Diagrama de bloques simplificado del motor DC.

Utilizando álgebra de bloques se puede obtener la siguiente función de transferencia correspondiente a la velocidad angular $\Omega(s)$:

$$\frac{\Omega(s)}{V_a(s)} = \frac{k_t}{R_a J s + R_a B + k_t k_b}$$

$$\frac{\Omega(s)}{V_a(s)} = \frac{\frac{k_t}{R_a J}}{s + \left(\frac{B}{J} + \frac{k_t k_b}{R_a J} \right)} \quad (3.19)$$

Para calcular el torque del motor se considerará que el par de carga $T_L = 0$ puesto que no se está realizando una acción contraria al movimiento del rotor, lo cual representa:

$$T_m(s) = \frac{k_t}{R_a} (V_a(s) - k_b \Omega(s)) \quad (3.20)$$

$$\frac{T_m(s)}{V_a(s)} = \frac{\frac{k_t}{R_a} \left(s + \frac{B}{J} \right)}{s + \left(\frac{B}{J} + \frac{k_t k_b}{R_a J} \right)} \quad (3.21)$$

3.2.2 Modelo analítico del brazo y del péndulo

Para el modelamiento matemático del brazo y el péndulo se plantea el diagrama de cuerpo libre que se muestra a continuación:

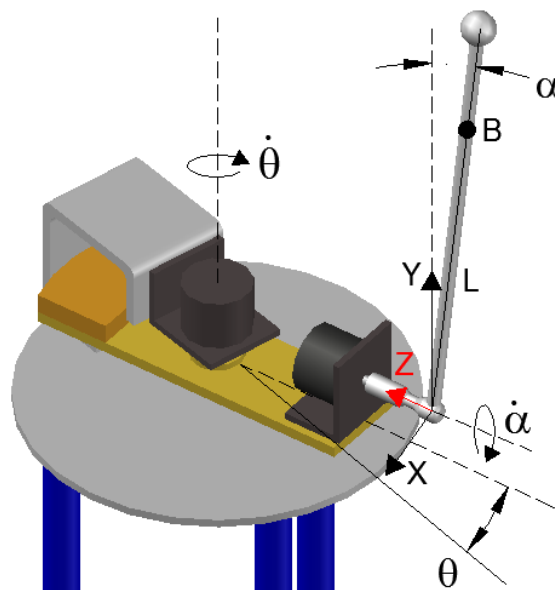


Figura 3.8 Diagrama de cuerpo libre del sistema PIR

Hay que tener en cuenta que el brazo gira solamente en el plano horizontal (plano XZ) y que el péndulo gira solamente en el plano vertical (plano XY). De la gráfica mostrada podemos notar lo siguiente:

$$x_{BA} = -L \sin \alpha \quad (3.22)$$

$$y_{BA} = L \cos \alpha \quad (3.23)$$

Derivando las ecuaciones (3.22) y (3.23), se puede observar que la velocidad del péndulo en el punto B con relación al punto A es:

$$\dot{x}_{BA} = -L(\cos \alpha) \dot{\alpha} \quad (3.24)$$

$$\dot{y}_{BA} = -L(\sin \alpha) \dot{\alpha} \quad (3.25)$$

En la Figura 3.9 se muestran las diferentes vistas de los diagramas de cuerpo libre y así notar las diferentes variables que intervienen en el movimiento.

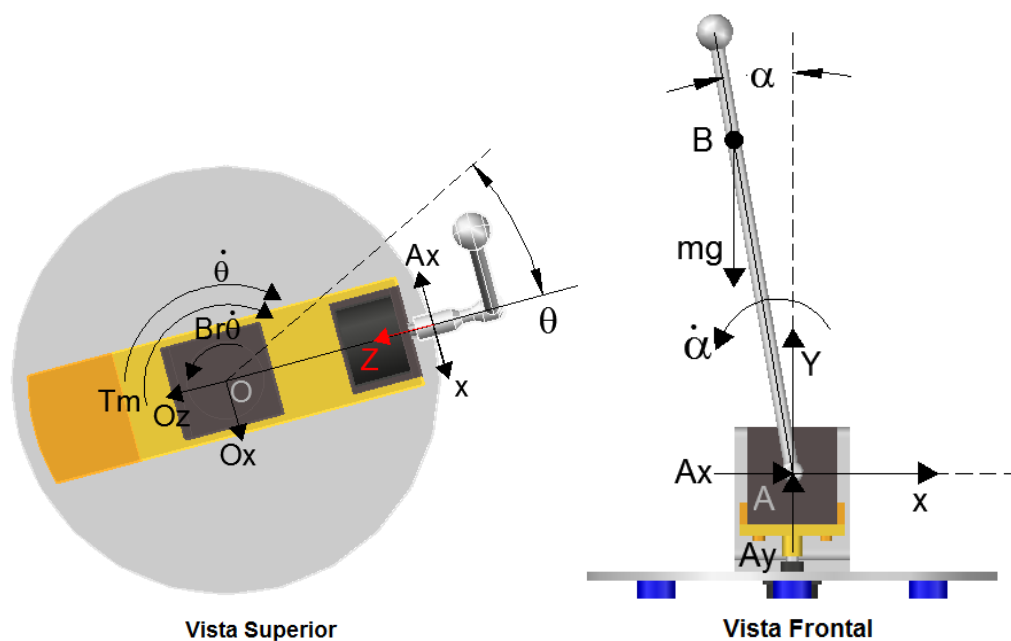


Figura 3.9 Diagrama de cuerpo libre del PIR en diferentes vistas

Se conoce que el péndulo se mueve a través del brazo con una velocidad lineal $r\dot{\theta}$, por tanto la velocidad absoluta del péndulo a través en el punto B se calcula a través de:

$$\dot{x}_B = r\dot{\theta} - L(\cos \alpha)\dot{\alpha} \quad (3.26)$$

$$\dot{y}_B = -L(\sin \alpha)\dot{\alpha} \quad (3.27)$$

Derivando las expresiones (3.26) y (3.27) respecto al tiempo se puede obtener la aceleración en el punto B:

$$\ddot{x}_B = r\ddot{\theta} + (L \sin \alpha)(\dot{\alpha})^2 - (L \cos \alpha)\ddot{\alpha} \quad (3.28)$$

$$\ddot{y}_B = -(L \cos \alpha)(\dot{\alpha})^2 - (L \sin \alpha)\ddot{\alpha} \quad (3.29)$$

Aplicando la segunda ley de Newton para el péndulo en el eje X:

$$\sum F_x = m_p \ddot{x}_B$$

$$m_p r \ddot{\theta} + (m_p L \sin \alpha)(\dot{\alpha})^2 - (m_p L \cos \alpha)\ddot{\alpha} = A_x \quad (3.30)$$

Aplicando la segunda ley de Newton para el péndulo en el eje y:

$$\sum F_y = m_p \ddot{y}_B$$

$$-(m_p L \cos \alpha)(\dot{\alpha})^2 - (m_p L \sin \alpha)\ddot{\alpha} + m_p g = A_y \quad (3.31)$$

Aplicando la ecuación de Euler para el movimiento rotacional en el péndulo alrededor del punto B, obtenemos:

$$\sum M_B = J_P \ddot{\alpha}$$

$$J_P \ddot{\alpha} = A_x L \cos \alpha + A_y L \sin \alpha - B_p \dot{\alpha} \quad (3.32)$$

Aplicando la ecuación de Euler para el movimiento rotacional en el brazo alrededor del punto O, obtenemos:

$$\sum M_O = J_O \ddot{\theta}$$

$$J_r \ddot{\theta} = T_m - B_r \dot{\theta} - A_x r \quad (3.33)$$

Reemplazando las ecuaciones (3.30) y (3.31) en (3.32):

$$\begin{aligned} J_P \ddot{\alpha} &= (m_p r \ddot{\theta} + (m_p L \sin \alpha)(\dot{\alpha})^2 - (m_p L \cos \alpha) \ddot{\alpha}) L \cos \alpha \\ &\quad + (-(m_p L \cos \alpha)(\dot{\alpha})^2 - (m_p L \sin \alpha) \ddot{\alpha} + m_p g) L \sin \alpha - B_p \dot{\alpha} \\ (J_P + m_p L^2) \ddot{\alpha} &- (m_p r L \cos \alpha) \ddot{\theta} - m_p g L \sin \alpha = -B_p \dot{\alpha} \quad (3.34) \end{aligned}$$

Reemplazando la ecuación (3.30) en (3.33) se obtiene:

$$\begin{aligned} (J_r + J_m) \ddot{\theta} &= T_m - B_r \dot{\theta} - (m_p r \ddot{\theta} + (m_p L \sin \alpha)(\dot{\alpha})^2 - (m_p L \cos \alpha) \ddot{\alpha}) r \\ -(m_p r L \cos \alpha) \ddot{\alpha} &+ (J_r + J_m + m_p r^2) \ddot{\theta} + (m_p r L \sin \alpha)(\dot{\alpha})^2 = T_m - B_r \dot{\theta} \quad (3.35) \end{aligned}$$

Finalmente el modelo no lineal del movimiento del péndulo y el brazo está representado por:

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} (J_p + m_p L^2) & -(m_p r L \cos \alpha) \\ -(m_p r L \cos \alpha) & (J_r + m_p r^2) \end{bmatrix}^{-1} \begin{bmatrix} m_p g L \sin \alpha - B_p \dot{\alpha} \\ T_m - B_r \dot{\theta} - (m_p r L \sin \alpha) (\dot{\alpha})^2 \end{bmatrix} \quad (3.36)$$

3.2.3 Modelo no lineal del sistema PIR

Considerando el diagrama de bloques del motor DC mostrado en la Figura 3.7, la relación entre el torque T_m , el voltaje de armadura V_a y la velocidad angular $\dot{\theta}$ se puede determinar a través de la expresión:

$$T_m = \frac{k_t}{R_a} (V_a - k_b \dot{\theta}) \quad (3.37)$$

Si se reemplaza (3.37) en (3.35) se obtiene:

$$\begin{aligned} -(m_p r L \cos \alpha) \ddot{\alpha} + (J_r + m_p r^2) \ddot{\theta} + (m_p r L \sin \alpha) (\dot{\alpha})^2 &= \frac{k_t}{R_a} (V_a - k_b \dot{\theta}) - B_r \dot{\theta} \\ -(m_p r L \cos \alpha) \ddot{\alpha} + (J_r + J_m + m_p r^2) \ddot{\theta} + (m_p r L \sin \alpha) (\dot{\alpha})^2 &= \frac{k_t}{R_a} V_a - \left(\frac{k_t k_b}{R_a} + B_r \right) \dot{\theta} \end{aligned} \quad (3.38)$$

Finalmente el modelo no lineal del sistema PIR queda definido a través de las ecuaciones (3.34) y (3.38):

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} (J_p + m_p L^2) & -(m_p r L \cos \alpha) \\ -(m_p r L \cos \alpha) & (J_r + m_p r^2) \end{bmatrix}^{-1} \begin{bmatrix} m_p g L \sin \alpha - B_p \dot{\alpha} \\ \frac{k_r}{R_a} V_a - \left(\frac{k_r k_b}{R_a} + B_r \right) \dot{\theta} - (m_p r L \sin \alpha) (\dot{\alpha})^2 \end{bmatrix}$$

3.2.4 Modelo lineal del sistema PIR

Para obtener el modelo lineal del sistema PIR se debe considerar un punto de equilibrio en torno al cual el sistema trabajará con normalidad. Para el sistema PIR se considerarán dos puntos de operación en particular: $\alpha = 0^\circ$ (En equilibrio) y $\alpha = 180^\circ$ (En reposo).

Se debe recordar que el método de la linealización aproximada será útil en la vecindad de su punto de equilibrio, siempre y cuando las perturbaciones que afectan la evolución del sistema sean suficientemente pequeñas [12].

3.2.1.4 Péndulo en posición vertical arriba ($\alpha \approx 0 \text{ rad}$)

Para mantener el equilibrio del péndulo en la posición vertical arriba, se debe considerar la función de transferencia relacionada a dicho punto de operación. Debido a la cantidad de términos que presenta el modelo, se

realizarán cambios de variables para minimizar errores matemáticos, utilizando lo siguiente:

$$\begin{aligned}
 k_1 &= J_p + m_p L^2 \\
 k_2 &= m_p r L \\
 k_3 &= m_p g L \\
 k_4 &= m_p r L \\
 k_5 &= J_r + J_m + m_p r^2 \\
 k_6 &= m_p r L
 \end{aligned} \tag{3.39}$$

$$k_1 \ddot{\alpha} - k_2 (\cos \alpha) \ddot{\theta} - k_3 (\sin \alpha) = -B_p \dot{\alpha} \tag{3.40}$$

$$-k_4 (\cos \alpha) \ddot{\alpha} + k_5 \ddot{\theta} + k_6 (\sin \alpha) (\dot{\alpha})^2 = \frac{k_t}{R_a} V_a - \left(\frac{k_t k_b}{R_a} + B_r \right) \dot{\theta} \tag{3.41}$$

Para linealizar la expresión (3.40) y (3.41) se consideran las siguientes aproximaciones:

$$\begin{aligned}
 \alpha &\approx 0 \\
 (\dot{\alpha})^2 &\approx 0 \\
 \sin \alpha &\approx \alpha \\
 \cos \alpha &\approx 1
 \end{aligned}$$

Luego de reemplazar los valores mencionados previamente en las expresiones (3.40) y (3.41), se obtiene:

$$k_1 \ddot{\alpha} - k_2 \ddot{\theta} - k_3 \alpha = -B_p \dot{\alpha} \tag{3.42}$$

$$-k_4 \ddot{\alpha} + k_5 \ddot{\theta} = \frac{k_t}{R_a} V_a - \left(\frac{k_t k_b}{R_a} + B_r \right) \dot{\theta} \tag{3.43}$$

Para llevar las expresiones (3.42) y (3.43) al dominio de la variable compleja “s” utilizamos la transformada de Laplace.

$$\frac{\theta(s)}{\alpha(s)} = \frac{k_1 s^2 + B_p s - k_3}{k_2 s^2} \quad (3.44)$$

$$\alpha(s) = \frac{k_2 s^2}{k_1 s^2 + B_p s - k_3} \theta(s) \quad (3.45)$$

Finalmente al combinar las expresiones (3.43) y (3.45) en términos de “s” se puede encontrar las funciones de transferencia mostradas a continuación:

$$\frac{\alpha(s)}{V_a(s)} = \frac{d_5 s}{R_a d_1 s^3 + (R_a d_2 + d_6) s^2 + (R_a d_3 + d_7) s + (R_a d_4 + d_8)} \quad (3.46)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{d_5 k_1 s^2 + d_5 B_p s - k_3 d_5}{s(R_a k_2 d_1 s^3 + (R_a k_2 d_2 + k_2 d_6) s^2 + (R_a k_2 d_3 + k_2 d_7) s + (R_a k_2 d_4 + k_2 d_8))} \quad (3.47)$$

Para simplificar las expresiones (3.46) y (3.47) se ha utilizado los siguientes cambios de variable:

$$\begin{aligned}
d_1 &= (k_1 k_5 - k_4 k_2) \\
d_2 &= (k_5 B_p + B_r k_1) \\
d_3 &= (-k_5 k_3 + B_r B_p) \\
d_4 &= -B_r k_3 \\
d_5 &= k_t k_2 \\
d_6 &= k_b k_1 k_t \\
d_7 &= k_b B_p k_t \\
d_8 &= -k_b k_3 k_t
\end{aligned}$$

3.2.1.5 Péndulo en posición vertical abajo ($\alpha \approx \pi \text{ rad}$)

Cuando el péndulo se encuentra en reposo, su dinámica debe ser representada por una función de transferencia acorde al movimiento que desarrollará. En este caso se considera un punto de equilibrio distinto y tendrá en cuenta las siguientes aproximaciones:

$$\begin{aligned}
(\alpha + \pi) &\approx \pi \\
(\dot{\alpha})^2 &\approx 0 \\
\sin(\alpha + \pi) &\approx -\alpha \\
\cos \alpha &\approx -1
\end{aligned}$$

Luego de reemplazar los valores mencionados previamente en las expresiones (3.40) y (3.41), se obtiene:

$$k_1 \ddot{\alpha} + k_2 \ddot{\theta} + k_3 \alpha = -B_p \dot{\alpha} \quad (3.48)$$

$$k_4 \ddot{\alpha} + k_5 \ddot{\theta} = \frac{k_t}{R_a} V_a - \left(\frac{k_t k_b}{R_a} + B_r \right) \dot{\theta} \quad (3.49)$$

Para llevar las expresiones (3.48) y (3.49) al dominio de la variable compleja “s” utilizamos la transformada de Laplace.

$$\frac{\theta(s)}{\alpha(s)} = \frac{-k_1 s^2 - B_p s - k_3}{k_2 s^2} \quad (3.50)$$

$$\alpha(s) = \frac{k_2 s^2}{-k_1 s^2 - B_p s - k_3} \theta(s) \quad (3.51)$$

Finalmente al combinar las expresiones (3.49) y (3.51) en términos de “s” se puede encontrar las funciones de transferencia mostradas a continuación:

$$\frac{\alpha(s)}{V_a(s)} = - \frac{k_t k_1 k_2 s^3 + k_t B_p k_2 s^2 + k_t k_2 k_3 s}{k_1 d_9 s^5 + d_{13} s^4 + d_{14} s^3 + d_{15} s^2 + d_{16} s + d_{12} k_3} \quad (3.52)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{k_t (k_1 s^2 + B_p s + k_3)}{s (d_9 s^3 + d_{10} s^2 + d_{11} s + d_{12})} \quad (3.53)$$

Para simplificar las expresiones (3.46) y (3.47) se ha utilizado los siguientes cambios de variable:

$$\begin{aligned}
 d_9 &= k_1 k_5 R_a - k_4 k_2 R_a \\
 d_{10} &= k_5 B_p R_a + B_r k_1 R_a + k_b k_1 k_t \\
 d_{11} &= k_5 k_3 R_a + B_r B_p R_a + k_b B_p k_t \\
 d_{12} &= B_r k_3 R_a + k_b k_3 k_t \\
 d_{13} &= d_9 B_p + d_{10} \\
 d_{14} &= d_9 k_3 + d_{10} B_p + d_{11} k_1 \\
 d_{15} &= d_{10} k_3 + d_{11} B_p + d_{12} k_1 \\
 d_{16} &= d_{11} k_3 + d_{12} B_p
 \end{aligned}$$

3.2.5 Parámetros experimentales de la planta

3.2.5.1 Constantes del Motor DC

Para medir los parámetros del modelo matemático del motor DC se desacopla el mismo del péndulo y se lo analiza de forma independiente utilizando el encoder incremental. La Figura 3.10 muestra el acoplamiento mecánico utilizado.

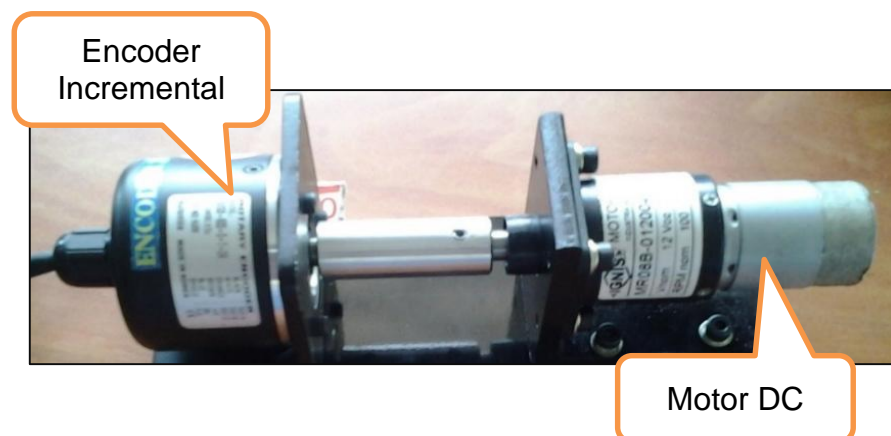


Figura 3.10 Acoplamiento entre motor y encoder incremental

✓ **Medición de la resistencia de armadura (R_a):**

Utilizando un multímetro digital Proskit MT-1232 y conectando los terminales del motor se realiza la medición de la resistencia de armadura. El valor de R_a varía respecto a la posición del rotor por lo que se debe girar lentamente el eje de forma manual, con esto obtenemos un cierto número de muestras y se realizó un promedio de las mismas. (Tabla1)

Tabla 1: Estimación experimental de resistencia Ra

Muestra	Ra (Ω)
1	5.414
2	5.49
3	5.426
4	5.626
5	4.76
6	5.53
7	5.8
Promedio:	5.435

Finalmente se obtiene el valor de $R_a = 5.435\Omega$.

✓ **Medición de la inductancia de armadura (L_a):**

Para obtener el valor de inductancia se utilizó el módulo NI ELVIS II+ de National Instruments conectando de forma directa los terminales del motor al mismo.

Se deben realizar diversas mediciones para obtener un promedio de las mismas, lo cual se puede evidenciar en la tabla 2.

Tabla 2: Estimación experimental de inductancia L_a

Muestra	L_a (H)
1	0.00196
2	0.00195
3	0.00195
4	0.00195
5	0.00185
6	0.00185
7	0.00188
8	0.00179
9	0.00179
10	0.00178
11	0.00178
12	0.00196
13	0.00198
14	0.00179
15	0.00198
16	0.00177
17	0.00200
Promedio:	0.00188

Finalmente se obtiene el valor de $L_a = 0.00188H$.

✓ **Constante de la fuerza contra electromotriz (K_b)**

Para determinar esta constante se realizó la medición de la velocidad angular (ω) del motor sin carga a diferentes valores de entrada de voltaje.

Para obtener las mediciones de la velocidad angular utilizó el módulo NI ELVIS II+ de National Instruments en combinación con un encoder rotativo tipo incremental de 600 pulsos por revolución instalado en el eje del motor.



Figura 3.11 Medición de k_b utilizando módulo NI ELVIS II+

En términos generales, el encoder rotativo transforma un movimiento angular en una serie de impulsos digitales, lo cual implica que se tiene una relación entre la frecuencia medida en sus terminales y la velocidad angular del eje:

$$\omega = 2\pi f \quad (3.54)$$

Donde f representa la frecuencia en Hertz y ω la velocidad angular del eje en rad/seg. De forma adicional se utilizó un osciloscopio PeakTeach 1200 para verificar la frecuencia medida por el módulo de NI Elvis II+ y se logró verificar la veracidad de los datos muestreados.

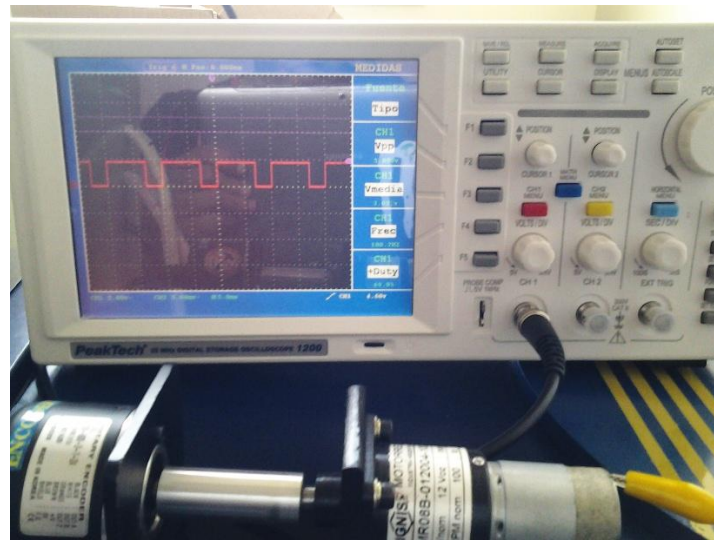


Figura 3.12 Medición de frecuencia utilizando osciloscopio PeakTech

Considerando que se ha medido la velocidad angular del motor en estado estacionario, se podría calcular la constante de velocidad sustituyendo las ecuaciones (3.6) y (3.54) en (3.1) y despejando k_b se obtiene:

$$k_b = \frac{V_a - R_a i_a}{\omega} \quad (3.55)$$

Realizando el cálculo de los valores de velocidad angular y midiendo la corriente de armadura para distintos valores de voltaje de alimentación (considerando el motor sin carga) se obtuvo el valor de k_b , promediando las mediciones como se indica en la Tabla 3.

Tabla 3: Estimación experimental de constante kb

V [V]	Ra [Ω]	Ia [A]	f [Hz]	ω [rad/s]	Kb [V.seg/rad]
2.5	5.435	0.117	125	1.309	1.424
3	5.435	0.126	170	1.780	1.300
4	5.435	0.135	270	2.827	1.155
5	5.435	0.133	370	3.875	1.104
6	5.435	0.14	460	4.817	1.088
7	5.435	0.143	570	5.969	1.043
8	5.435	0.154	680	7.121	1.006
9	5.435	0.151	770	8.063	1.014
10	5.435	0.154	850	8.901	1.029
11	5.435	0.156	950	9.948	1.020
12	5.435	0.157	1040	10.891	1.023
PROMEDIO:					1.110

Finalmente se obtiene el valor de $k_b = 1.11 \frac{V \cdot \text{seg}}{\text{rad}}$.

✓ **Constante del par electromagnético (Kt)**

Utilizando la relación de la ecuación (3.14) y expresando k_t en sus respectivas unidades:

$$k_t = 1.11 \frac{N \cdot m}{A}$$

✓ **Coefficiente de fricción viscosa (B)**

Para la obtención del coeficiente de fricción viscosa del motor, es necesario utilizar la ecuación diferencial del motor sin carga ($T_L = 0$):

$$T_m = J_m \frac{d\omega}{dt} + B\omega \quad (3.56)$$

Donde $T_m = k_b \cdot i_a$ es el torque en el motor. Al energizar el motor con un voltaje constante durante cierto tiempo, hasta que llegue a una velocidad constante, la aceleración angular desaparece obteniendo directamente la relación del coeficiente de fricción viscosa. Al eliminar el término de la derivada de la velocidad angular y despejando el coeficiente de fricción viscosa, tenemos:

$$B = \frac{k_b i_a}{\omega}$$

(3.57)

Considerando los valores promedios de k_b e i_a mostrados previamente, finalmente obtenemos:

$$B = 0.04166 \frac{N \cdot m \cdot seg}{rad}$$

✓ **Momento de inercia equivalente del rotor (Jm)**

Utilizando la ecuación (3.56) y resolviendo la misma es posible encontrar el valor del momento de inercia. Sabiendo que la solución de esta ecuación diferencial puede ser encontrada como la suma de una solución particular más una respuesta transitoria.

$$y = y_p + y_t$$

Para encontrar la solución particular de la ecuación, el término derivativo de la ecuación desaparece con lo cual la solución particular puede ser calculada directamente como:

$$y_p = \omega = \frac{T_m}{B} \quad (3.58)$$

Una vez obtenida la solución particular del sistema es necesario calcular la respuesta transitoria, para ello se realiza la igualdad de $\omega = e^{mt}$, sustituyendo e igualando a cero:

$$J_m \frac{de^{mt}}{dt} + Be^{mt} = 0$$

Realizando la derivada de e^{mt} en el primer término de la ecuación y reordenando:

$$e^{mt} (mJ_m + B) = 0$$

Para conocer el valor de la variable m en la exponencial se realiza el cálculo de la ecuación anterior para un tiempo igual a cero, siendo el valor de la exponencial igual a uno. Despejando m se encuentra su valor como:

$$m = -\frac{B}{J_m}$$

Obtenido el valor de la constante de tiempo en la exponencial la respuesta transitoria está dada por:

$$y_t = \frac{T_m}{B} e^{-\frac{B}{J_m}t} \quad (3.59)$$

Sumando la respuesta particular y la respuesta transitoria

$$\omega(t) = \frac{T_m}{B} \left(1 - e^{-\frac{B}{J_m}t} \right)$$

Donde la constante de tiempo de esta ecuación es igual a

$$\tau = \frac{J_m}{B} \quad (3.60)$$

Esta constante es conocida como la constante mecánica del motor, la cual por medio de experimentación se calcula como un tercio del tiempo transcurrido entre la desconexión del motor sin carga a una velocidad constante y su frenado.

Para conocer el tiempo que transcurría entre la desconexión y el frenado del motor se utilizó el osciloscopio PeakTech 1200 y una fuente de voltaje DC variable GPS-3303, como se muestra a continuación:



Figura 3.13 Medición de constante mecánica del motor.

Tabla 4: Estimación experimental de constante mecánica del motor

Muestra	Tiempo de Frenado (seg)
1	0.036
2	0.044
3	0.047
4	0.064
5	0.065
6	0.066
7	0.067
8	0.069
9	0.067
10	0.068
11	0.068
Promedio:	0.060

Realizando varias mediciones y obteniendo un promedio de las mismas se obtiene un tiempo de frenado del motor de 60 ms, lo cual se puede evidenciar a través de la tabla 4.

Con esto en la ecuación $\tau = \frac{J_m}{B}$, sustituyendo los valores de la constante de tiempo t , y la constante de fricción equivalente a un tercio de la constante mecánica del motor se obtiene el momento de inercia del motor como:

$$J_m = \tau B = (0.020 \text{ seg}) (0.04166 \frac{N \cdot m \cdot \text{seg}}{\text{rad}})$$

$$J_m = 0.00070926 \text{ Kg} \cdot \text{m}^2$$

Para demostrar lo mencionado en el análisis teórico acerca de las constantes de tiempo del circuito eléctrico y el circuito mecánico, se procede a comparar las mismas a través de la tabla 5.

Tabla 5: Comparación de constantes de tiempo

La [H]	Ra [Ω]	Jm [kgm ²]	B [Nms/rad]	τ_{mec} [s]	τ_{elec} [s]	$\frac{\tau_{mec}}{\tau_{elec}}$
0.00188	5.435	0.00070926	0.04166	0.01702	0.0003459	49.2184

Como se puede evidenciar en la tabla, la constante de tiempo eléctrica es 49.21 veces más rápida que la constante de tiempo mecánica, lo cual permite eliminar el efecto de la inductancia La y así reducir el modelo matemático del motor.

3.2.5.2 Parámetros constructivos del Péndulo

✓ Momento de Inercia, J_p

Para obtener el momento de inercia del péndulo, se ha realizado el diseño del mismo en el software SolidWorks 2014, considerando las densidades de los materiales que lo conforman: Aluminio en la barra y Acero Inoxidable en la esfera.

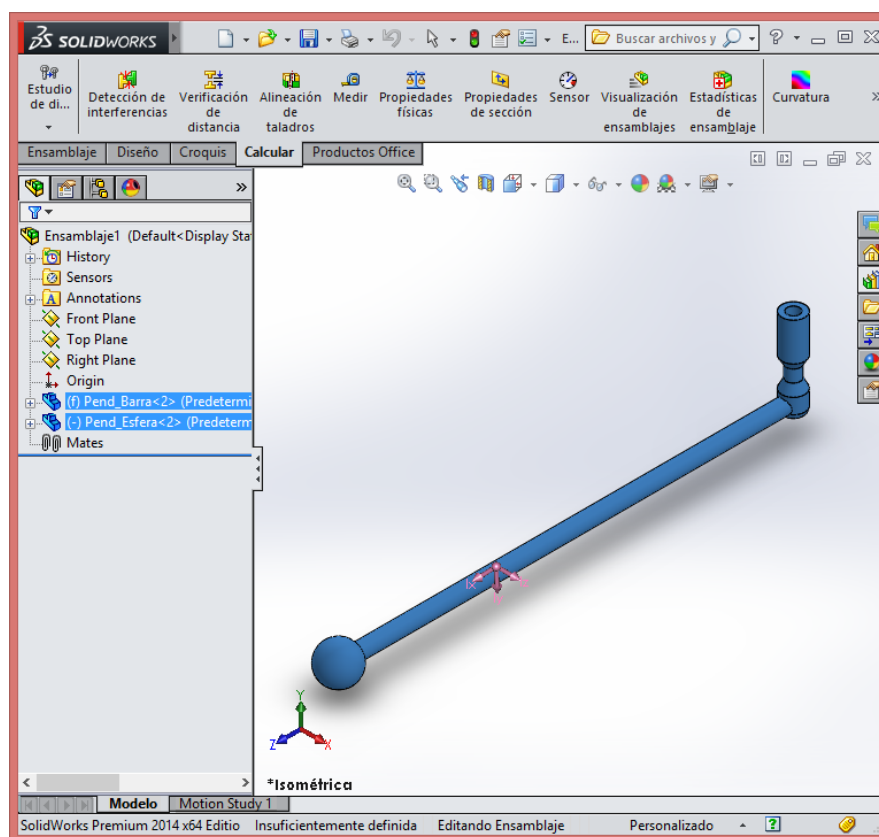


Figura 3.14 Diseño del péndulo en SolidWorks.

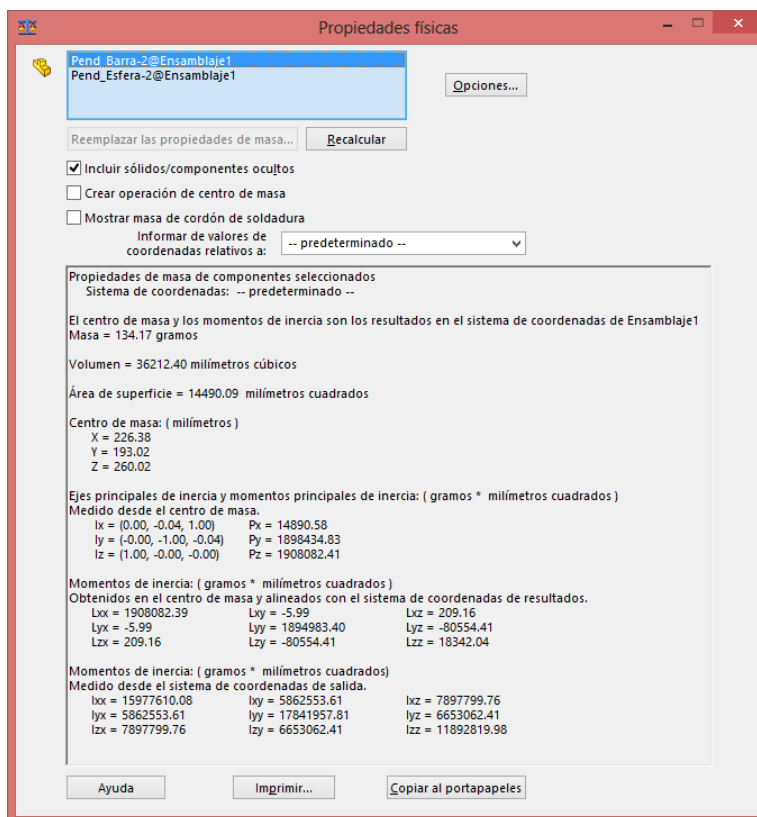


Figura 3.15 Parámetros del péndulo en obtenidos en SolidWorks.



Figura 3.16 Medición de masa del péndulo

Para verificar la aproximación entre los datos estimados por el programa y los valores reales del péndulo, se procede a realizar la medición de la masa de cada una de las partes que conforman al péndulo y la sumatoria de estas es muy próxima a los valores mostrados por el CAD ($m_p = 134.17\text{gr}$). En la Figura 3.16 se puede apreciar el detalle de las mediciones realizadas.

De los resultados obtenidos en el CAD, podemos mencionar que el momento de inercia del péndulo relativo a la rotación en el centro de masa, cuyo eje de giro es paralelo al eje del encoder, es $J_{pCM} = 1898434.83\text{ gr} \cdot \text{mm}^2$.

$$J_{pCM} = 0.00189843483\text{ kgm}^2$$

En este caso es importante calcular el momento de inercia del péndulo alrededor del eje del encoder. Para ello se utilizará el teorema de ejes paralelos de Huygens-Steiner [16], el cual se plantea de la siguiente forma:

$$J_p = J_{pCM} + m_p L^2 \quad (3.61)$$

Donde:

J_p es el momento de inercia del péndulo según el eje que no pasa a través de su centro de masas.

J_{pCM} es el momento de inercia del péndulo según un eje que pasa a través de su centro de masas

m_p es la masa del péndulo

L es la distancia perpendicular entre los dos ejes.

A través del software se ha podido obtener el valor de la masa y de la distancia entre los ejes.

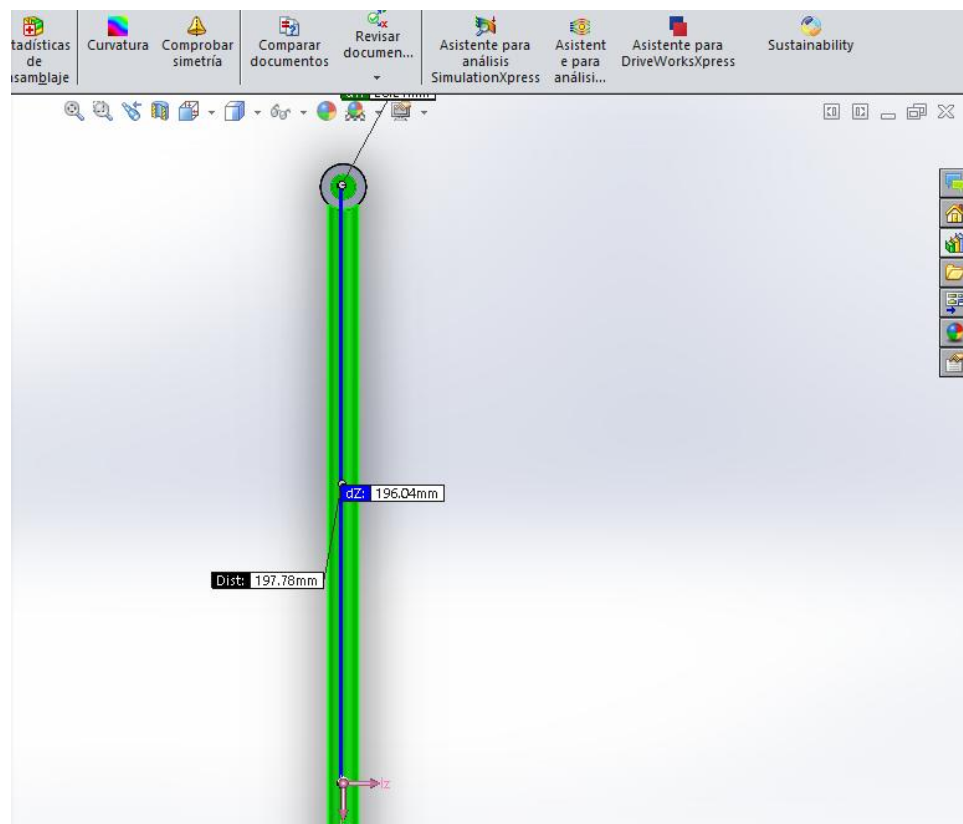


Figura 3.17 Obtención de centro de masa

Reemplazando los valores medidos a través de software en la expresión (3.61), tenemos:

$$J_p = 0.00189843483 + (0.13417)(0.19604)^2$$

$$J_p = 0.00705481355 \text{ kgm}^2$$

✓ Fricción Viscosa, B_p

Para obtener el coeficiente de fricción viscosa en el péndulo, se ha procedido a observar la dinámica del mismo cuando oscila durante la caída libre.

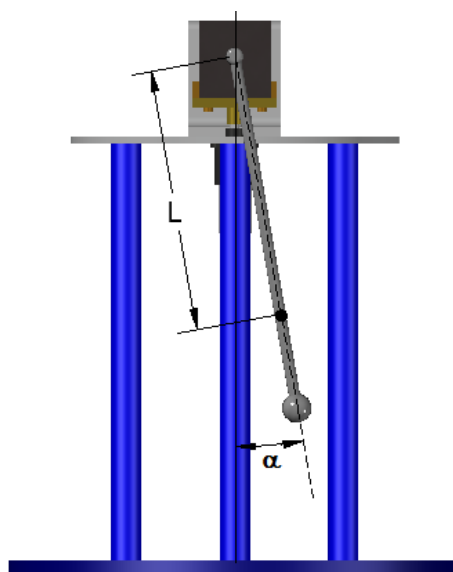


Figura 3.18 Configuración del péndulo para oscilación amortiguada

La ecuación diferencial del movimiento oscilatorio amortiguado mostrado en la Figura 3.18, está dada por:

$$J_p \frac{d^2\alpha}{dt^2} + B_p \frac{d\alpha}{dt} + m_p g L \alpha = 0 \quad (3.62)$$

Donde $B_p \frac{d\theta}{dt}$ representa el amortiguamiento dado por la fricción entre el péndulo y el aire. La solución a esta ecuación corresponde a la siguiente expresión:

$$\alpha = \alpha_0 e^{-\frac{B_p t}{2J_p}} \cos(\omega_0 t + \varphi) \quad (3.63)$$

Donde ω_0 es la frecuencia de movimiento, φ es la fase inicial del movimiento, α_0 es la amplitud de la oscilación y $\gamma = B_p/2J_p$ corresponde a la constante de amortiguamiento.

Para obtener experimentalmente esta constante se posiciona el péndulo con un ángulo inicial de 9.3° y se deja caer hasta que se detenga.

Para la captura de datos se consideró un tiempo de muestreo de 2ms y se recolectaron 4164 muestras, lo cual se muestra en la figura 3.19.

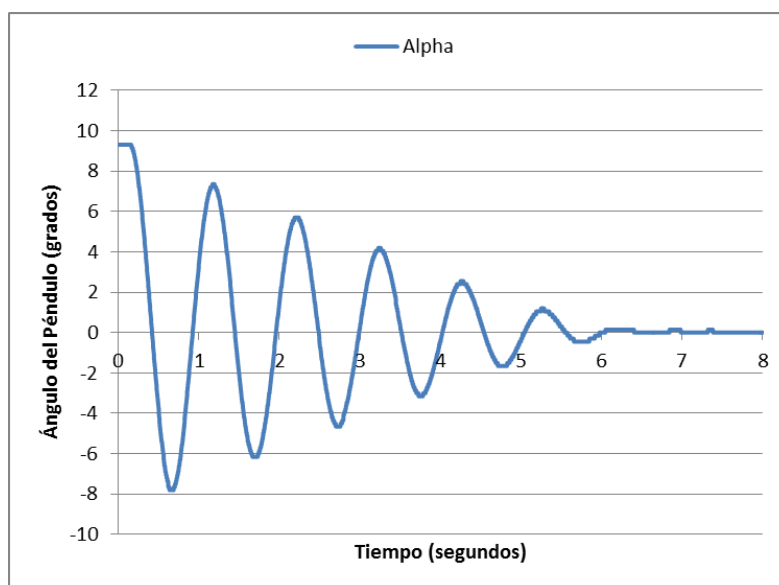


Figura 3.19 Respuesta oscilatoria del péndulo

Utilizando herramientas informáticas, como Microsoft Excel, la envolvente de dicha señal se puede aproximar tal como se visualiza en la figura 3.20.

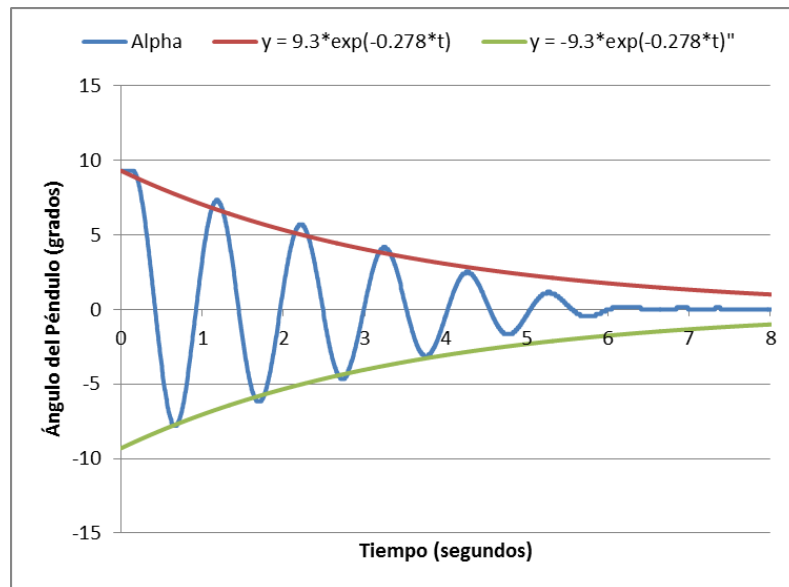


Figura 3.20 Gráfico de envolventes del péndulo

Dado que la representación mostrada se asemeja a la señal medida podemos calcular el coeficiente de fricción viscosa B_p de la siguiente forma:

$$\frac{-B_p}{2J_p} = -0.278$$

$$B_p = 0.278(2)(J_p)$$

$$B_p = 0.278(2)(0.02820112)$$

$$B_p = 0.015679822 \text{ Nms}$$

✓ **Ganancia de la interface de potencia, G_drive**

Hay que recordar que la interfaz de potencia corresponde a un bloque en el diagrama de control, por ende deberá ser parametrizado. Dado que el

tiempo de respuesta del circuito es pequeño, el bloque se considerará como una ganancia estática.

Para obtener el valor de ganancia, se aplican diversos valores de voltaje en la entrada y se observará el comportamiento de la salida, como se muestra en la figura 3.21.

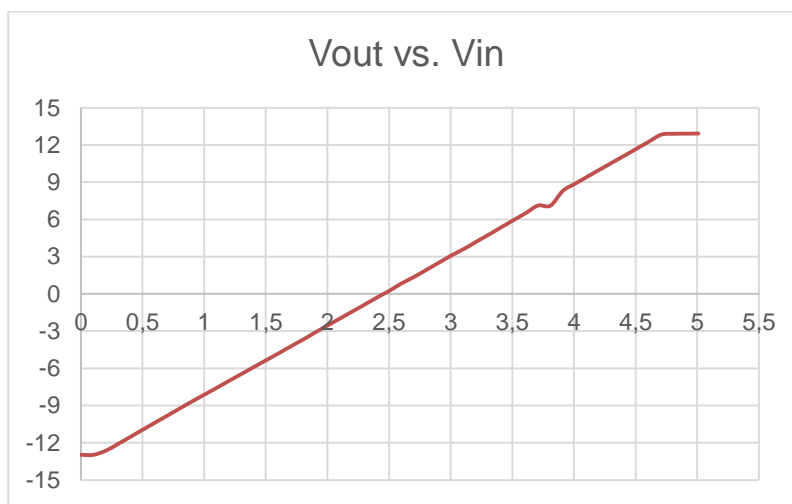


Figura 3.21 Relación entrada – salida de circuito de potencia

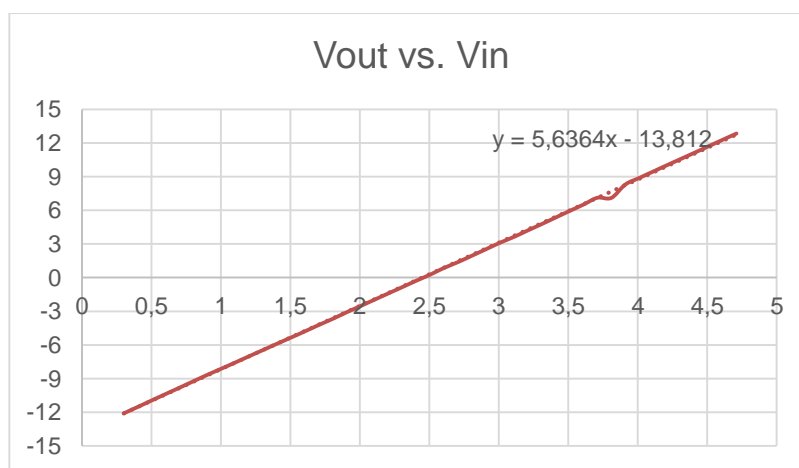


Figura 3.22 Línea de tendencia para el circuito de potencia

En la figura 3.21, se muestra puede notar que la relación entrada-salida es una recta en cuyos extremos existe una saturación. Si se descartan los puntos donde se satura la señal, puede encontrar la ganancia correspondiente al circuito, lo cual se muestra en la figura 3.22.

Para este caso, la ganancia es igual a la pendiente de la recta mostrada lo que significa que $G_{drive} = 5.635$.

✓ Resumen de parámetros

Finalmente en la tabla1 se puede apreciar los parámetros del modelo a utilizar para el diseño de controladores. Cabe mencionar que durante la validación del modelo se han afinado algunos valores para tener una mejor respuesta.

Tabla 6: Parámetros del sistema PIR

Parámetros	Valores
R_a – resistencia de armadura	5.4351 Ω
L_a – inductancia de armadura	0.00188 H
k_b – constante de fuerza contraelectromotriz	0.91098 Vs/rad
k_t – constante de par electromagnético	1.1098 Nm/A
J_m – momento de inercia equivalente del motor	0.00071 $kg \cdot m^2$
B_r – coeficiente de fricción viscosa del motor	0.04166 Nms/rad
g – aceleración de la gravedad	9.8 m/s^2
m_p – masa del péndulo	0.13417 kg
J_r – momento de inercia equivalente del brazo	0.0078 $kg \cdot m^2$
r – radio de giro del brazo	0.125 m
L – Longitud del eje de giro hasta el CM del péndulo	0.19604 m
J_p – momento de inercia del péndulo	0.001882 $kg \cdot m^2$
B_p – coeficiente de fricción viscosa del péndulo	$4.56798 \cdot 10^{-4}$ Nms/rad

3.3 Validación del modelo matemático del PIR

Para validar el modelo matemático obtenido se realizarán varias pruebas con la finalidad de observar la cercanía que se tiene con el modelo real.

3.3.1 Pruebas de validación del motor DC

Para validar el modelo matemático del motor DC se procede a generar distintas señales de prueba con el objetivo de observar que tanto se aproxima con la respuesta Real; cabe mencionar que para esta prueba se ha considerado desconectar al motor del resto del sistema.

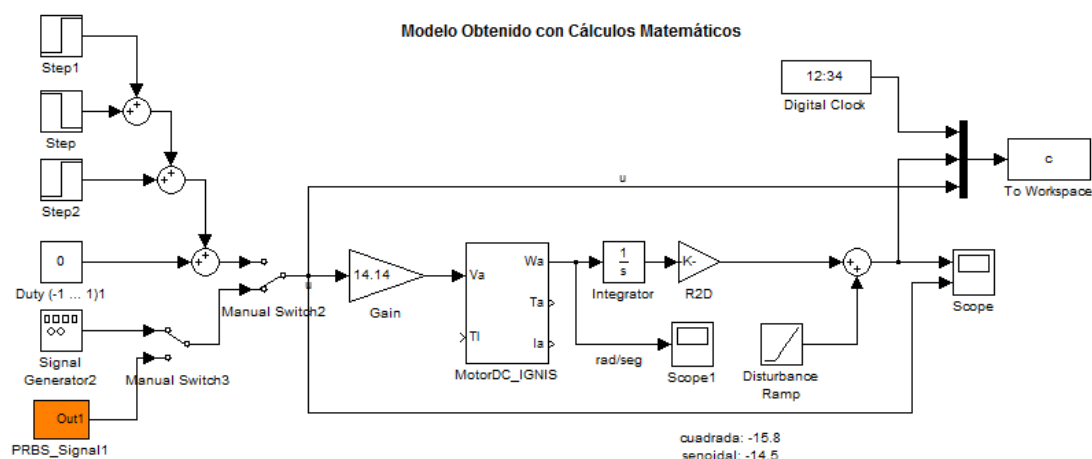


Figura 3.23 Simulación de modelo analítico del motor DC.

Para las pruebas se ha utilizado diversas señales de voltaje, tales como onda cuadrada o senoide, lo cual se puede apreciar en los archivos realizados en SimuLink mostrados en las Figuras 3.23 y 3.24.

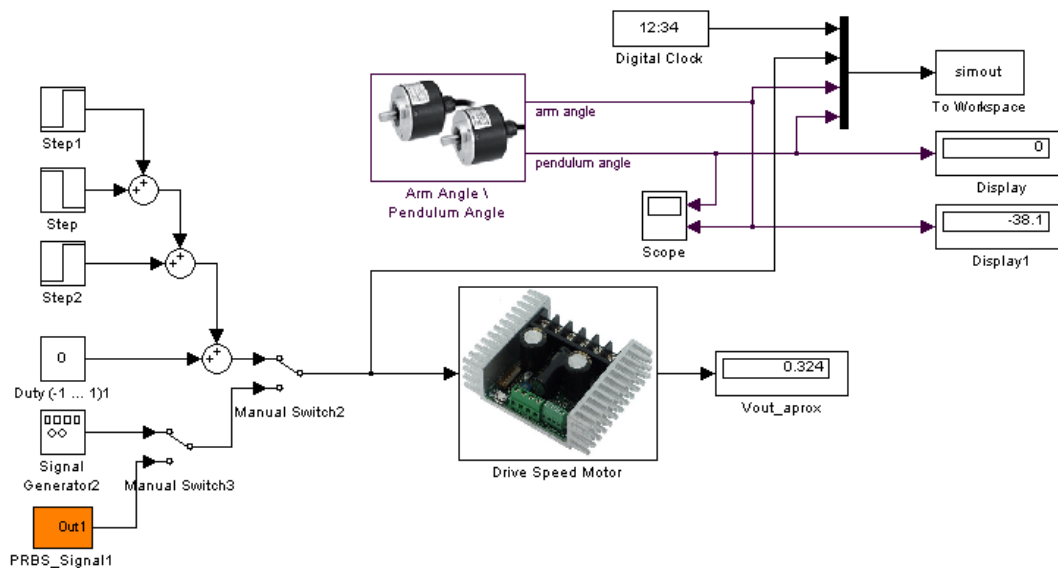


Figura 3.24 Interface de entrada y salidas de datos del motor DC.

En primer lugar se aplica una señal cuadrada con las características mostradas en la figura 3.25 y se verifica la respuesta del sistema en la figura 3.26.

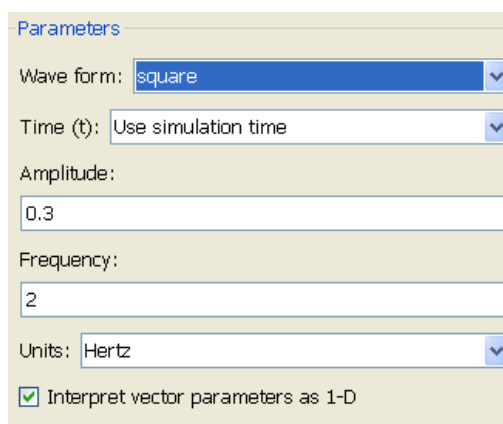


Figura 3.25 Señal cuadrada aplicada al motor.

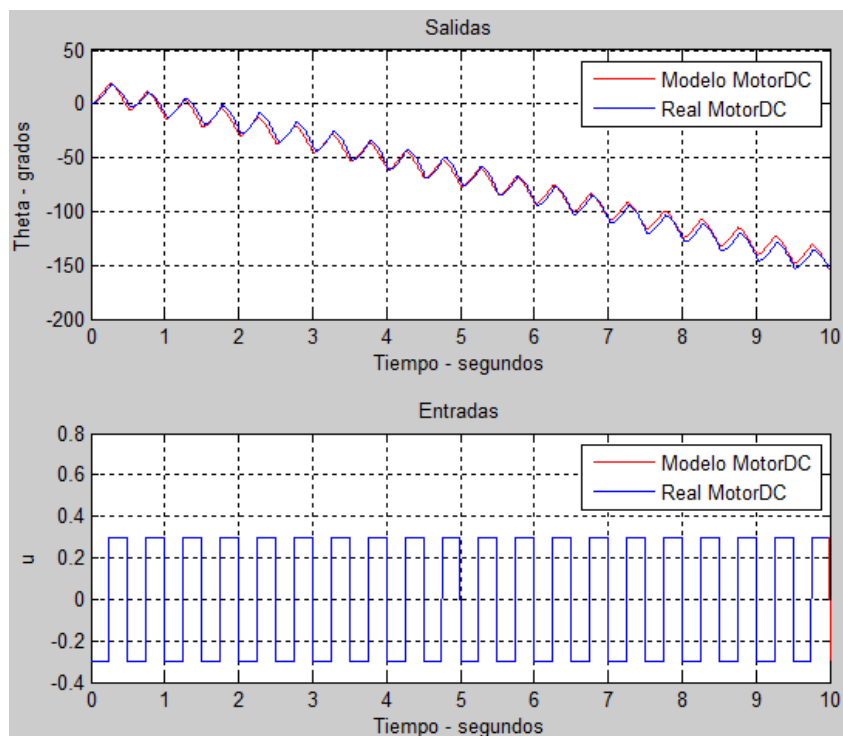


Figura 3.26 Comparación entre la respuesta implementada y simulada del motor DC.

En segundo lugar se aplica una señal sinusoidal con las características mostradas en la figura 3.27 y se verifica la respuesta del sistema en la figura 3.28.

Parameters

Wave form: sine ▼

Time (t): Use simulation time ▼

Amplitude: 0.3

Frequency: 2

Units: Hertz ▼

Interpret vector parameters as 1-D

Figura 3.27 Señal sinusoidal aplicada al motor.

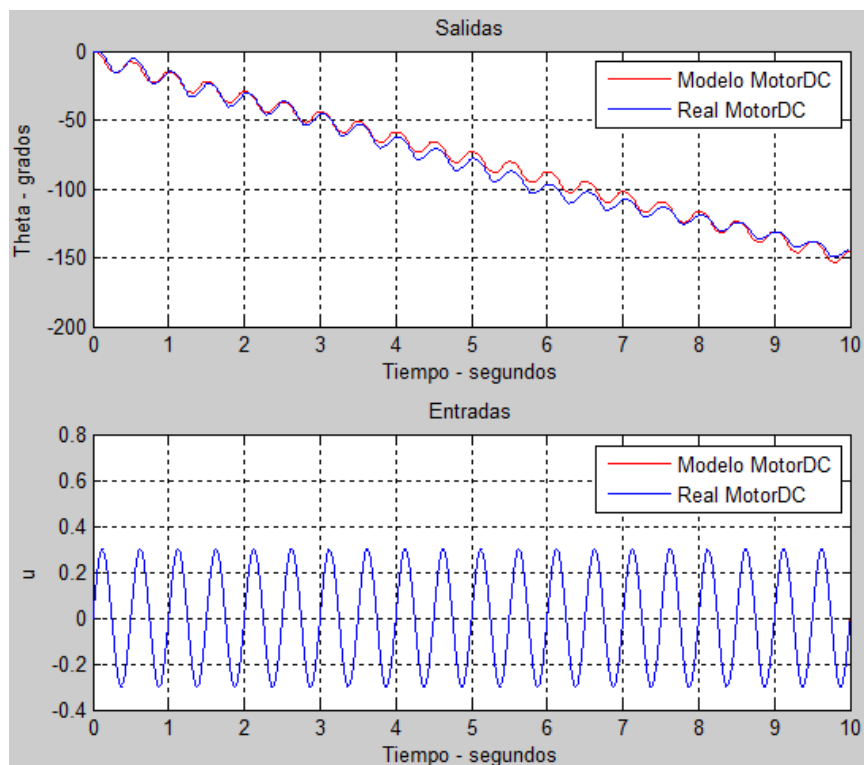


Figura 3.28 Comparación entre la respuesta implementada y simulada del motor DC.

Para verificar que el modelo matemático del motor DC corresponde al elemento real, se procede a calcular el error RMS y el porcentaje correspondiente mostrada en la Tabla 7. El código relacionado a esta prueba se puede encontrar en el Anexo 12.

Tabla 7: Cálculo del error absoluto y relativo RMS para el motor DC

Señales de Prueba	Error RMS	Error Relativo RMS (%)
Onda Cuadrada	3.7649 °	4.5994 %
Onda Senoidal	3.9999 °	4.4113 %

3.3.2 Pruebas de validación del sistema PIR

Para comprobar que los valores calculados experimentalmente son correctos, se ha considerado aplicar una sumatoria de señales sinusoidales como se muestra en la figura 3.29. Cabe mencionar que las pruebas han sido realizadas con el péndulo inicialmente en reposo ($\alpha = 180^\circ$).

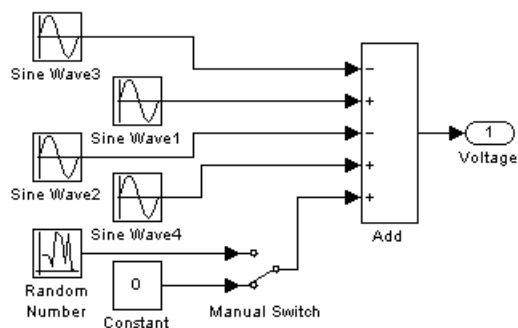


Figura 3.29 Señales aplicadas al sistema PIR en lazo abierto

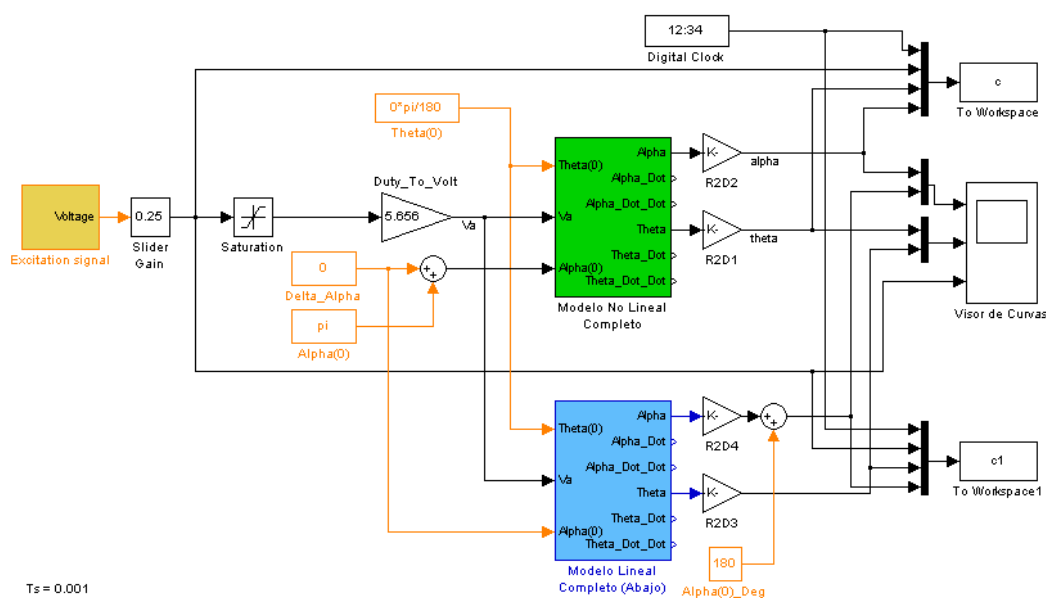


Figura 3.30 Modelo simulado del sistema PIR en lazo abierto

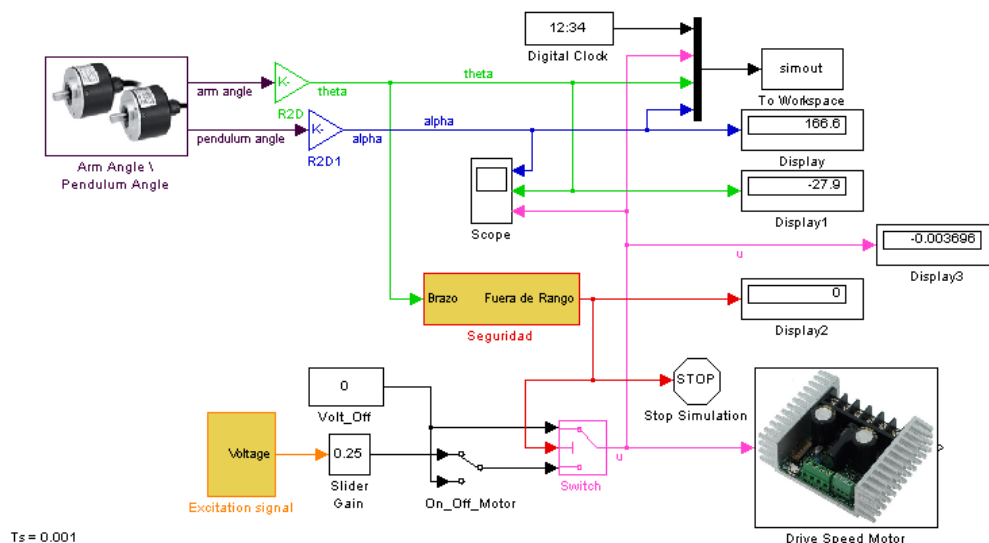


Figura 3.31 Implementación del sistema PIR en lazo abierto

Al simular los modelos, lineal y no lineal, mostrados en la Figura 3.30 y la implementación mostrada en la Figura 3.31, obteniendo las gráficas mostradas a continuación.

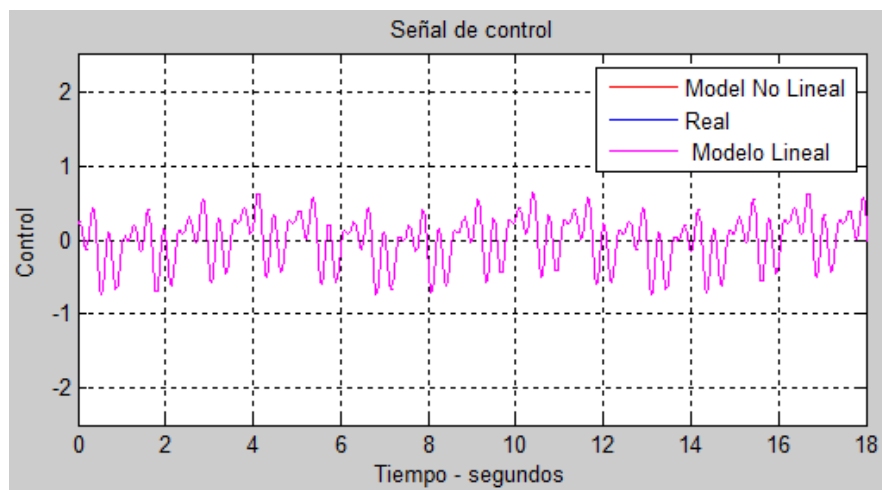


Figura 3.32 Comparación de señales de control aplicadas

En la figura 3.32 se muestra la señal aplicada al motor DC lo cual es el resultado de la sumatoria de señales sinusoidales. En la figura 3.33 se visualiza el comportamiento del péndulo tanto en la implementación como en simulación del modelo lineal y no lineal.

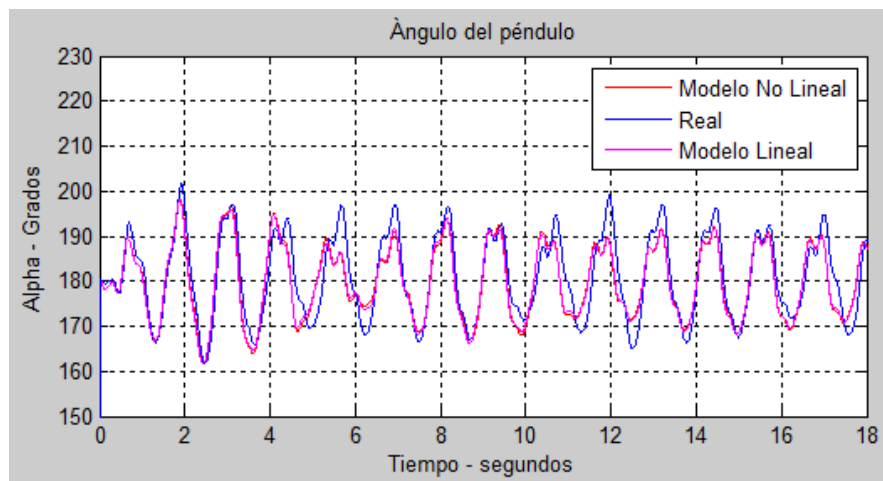


Figura 3.33 Comparación de valores angulares obtenidos en el péndulo

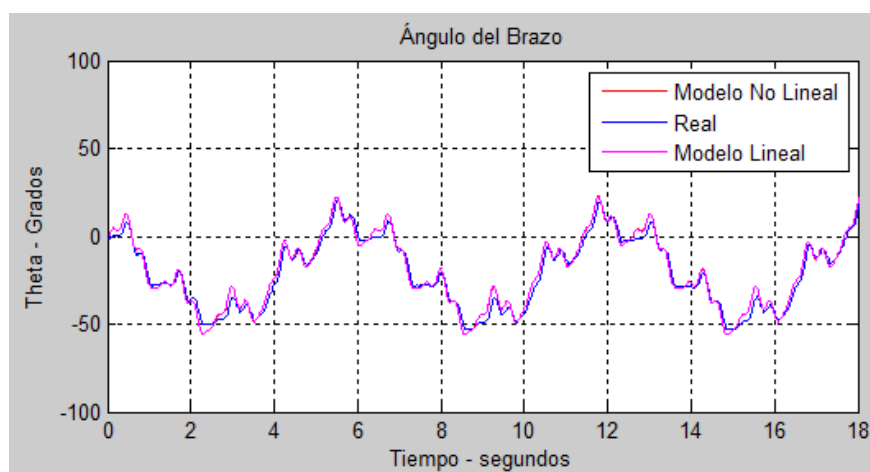


Figura 3.34 Comparación de valores angulares obtenidos en el brazo

La figura 3.34 muestra el comportamiento del péndulo, simulado e implementado, se puede apreciar la similitud entre la respuesta real y la simulación.

Se puede apreciar que la dinámica del modelo estimado presenta una aproximación considerable respecto a la respuesta real del sistema PIR, lo cual se puede apreciar en la Tabla 9, por tanto se utilizará el mismo para el diseño de controladores.

Tabla 9: Cálculo del error absoluto y relativo RMS para el motor DC

Parámetros	Error RMS
θ	3.2034 °
α	4.2377 °

CAPÍTULO 4

4. DISEÑO DE CONTROLADORES LINEALES

En esta sección se realiza el diseño del controlador PID lineal para mantener equilibrado el péndulo en posición vertical hacia arriba. Además se realiza el dimensionamiento del controlador PI lineal para el control del brazo durante el equilibrio del péndulo.

4.1 Diseño del controlador PID Lineal para el control del péndulo

Para el diseño del controlador lineal se considerará la función de transferencia mostrada en (3.46), la cual es ingresada en Matlab utilizado

el código mostrado en el Anexo 6. Considerando el punto de operación en $\alpha = 0^\circ$ y $\theta = 0^\circ$ y los parámetros mostrados en la tabla 1, se obtienen las funciones de transferencia en:

$$G_1(s) = \frac{10.53s}{s^3 + 25.2s^2 - 41.24s - 921.2} \quad (4.1)$$

$$G_2(s) = \frac{22.54s^2 + 1.463s - 826.2}{s(s^3 + 25.2s^2 - 41.24s - 921.2)} \quad (4.2)$$

Se procede a verificar la dinámica del sistema PIR ante una señal tipo escalón unitario como se indica en las figuras 4.1 y 4.2.

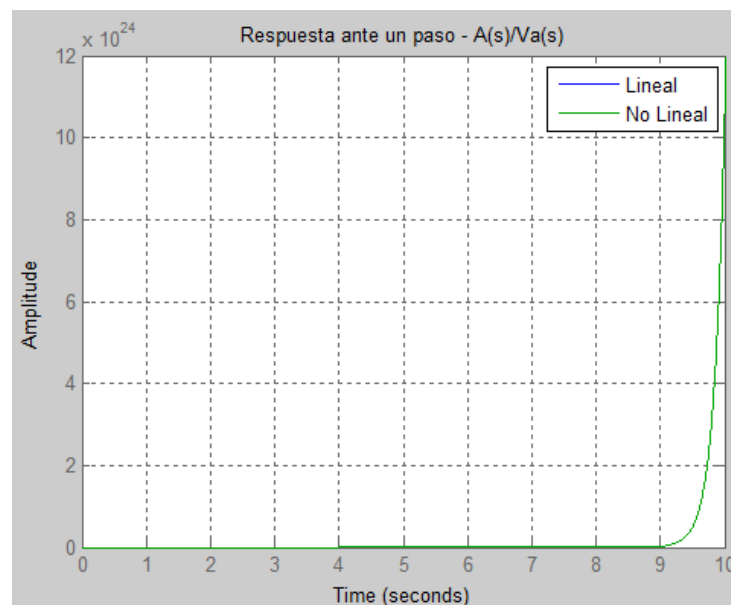


Figura 4.1 Respuesta del ángulo del péndulo ante una entrada paso

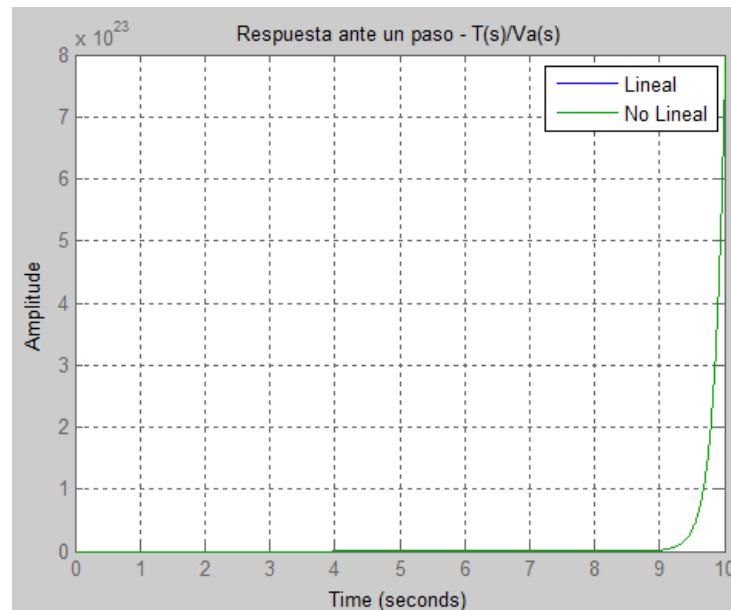


Figura 4.2 Respuesta del ángulo del brazo ante una entrada paso

Como se puede notar en las gráficas previas, el sistema en lazo abierto es inestable. Posterior a esto se procede a diseñar el controlador empleando la herramienta SISOTOOL, es importante considerar que el circuito de potencia será representado por una ganancia estática y se utilizará el siguiente código:

```

%% Análisis en lazo cerrado del controlador PID
Drive_Gain = 5.656; %Ganancia aproximada en la tarjeta de
    %potencia
sisotool(Drive_Gain*G1);

```

La estructura de control para este caso es un tanto diferente al sistema realimentado usual, debido a se quiere controlar la posición vertical del péndulo en en $\alpha = 0^\circ$ rechazando el efecto de las perturbaciones.

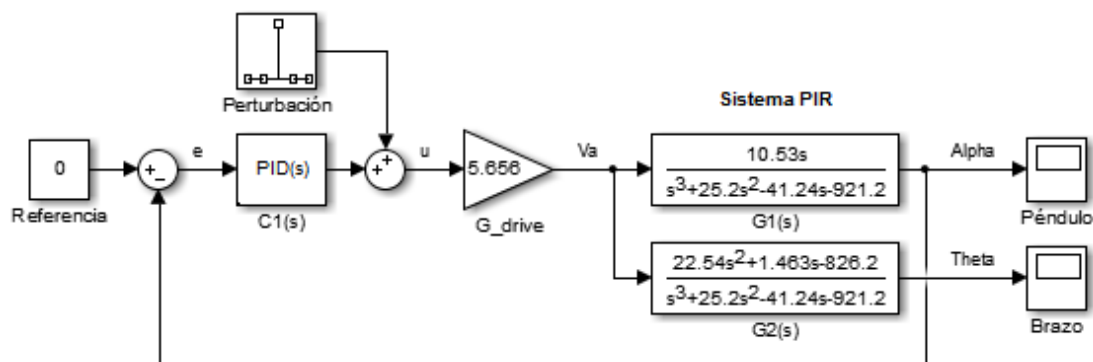


Figura 4.3 Diagrama de bloques para control regulatorio del péndulo

Este tipo de situación hace referencia a un problema de regulación, donde la fuerza externa podría ser considerada un disturbio impulsivo.

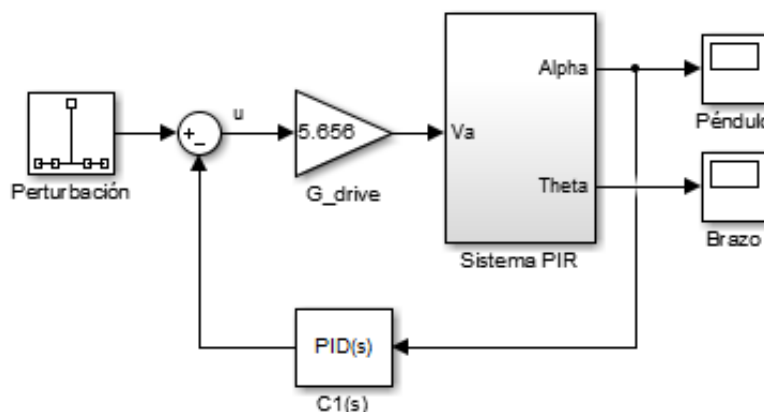


Figura 4.4 Bloques simplificados para control regulatorio del péndulo

La estructura de control a utilizar en SISOTOOL se muestra en la figura 4.5, haciendo enfoque en el rechazo a las perturbaciones.

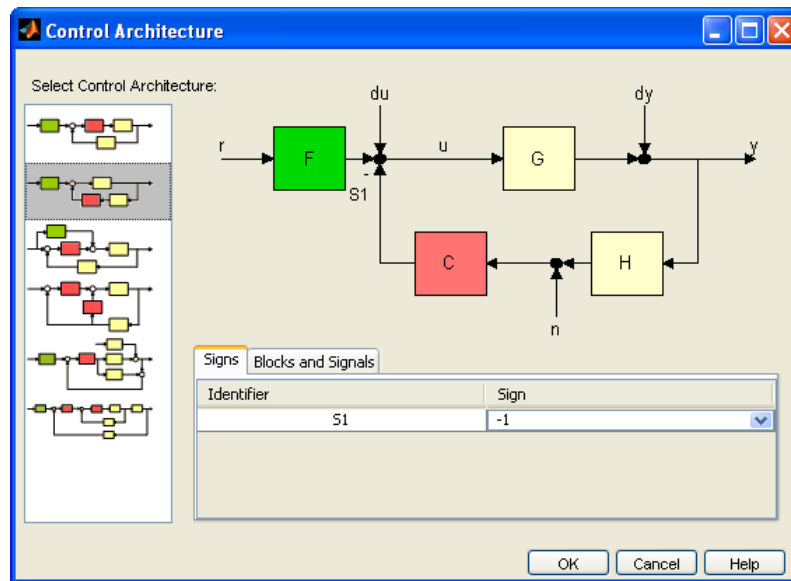


Figura 4.5 Estructura de control para equilibrio del péndulo

Para el ajuste se tiene en cuenta requerimientos de diseño, tales como, tiempo de asentamiento de 0.2 segundos y un porcentaje de sobrepaso del 2%.

Cabe mencionar que el ajuste del controlador debe procurar que la salida del mismo no se sature, pues en ese caso no se garantiza que el control sea efectivo.

El ajuste del controlador está basado en el análisis del lugar de las raíces considerando los requerimientos mencionados previamente, lo cual se puede apreciar en las Figuras 4.6, 4.7 y 4.8.

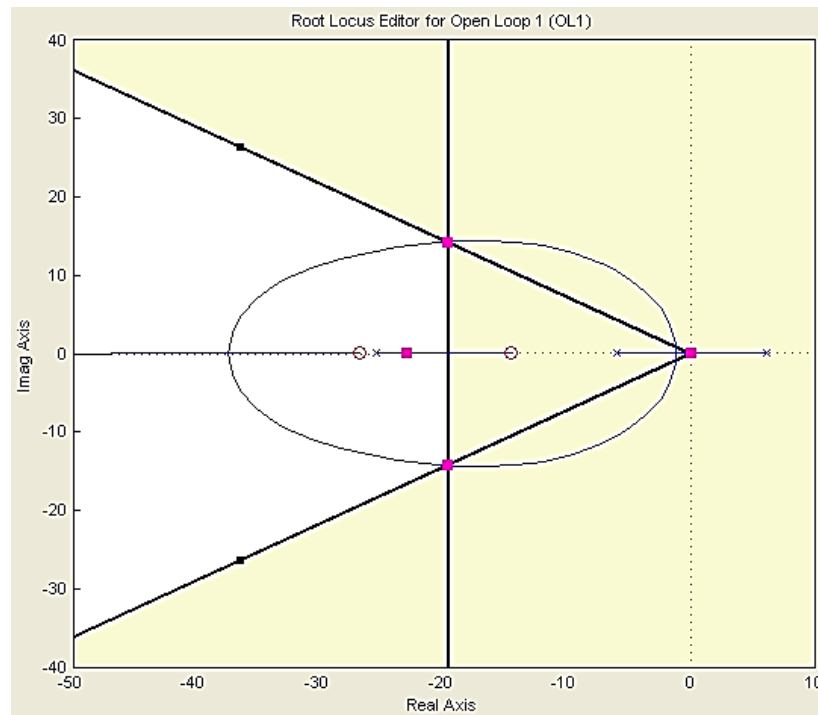


Figura 4.6 Lugar de las raíces del sistema PIR con controlador PID

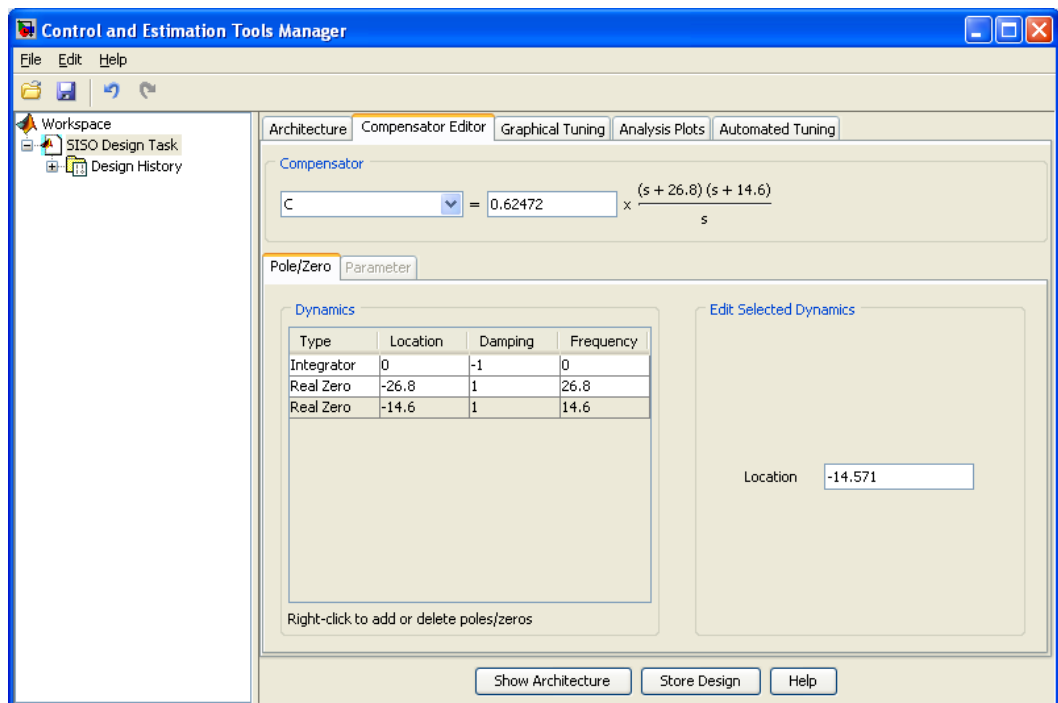


Figura 4.7 Controlador PID para el péndulo

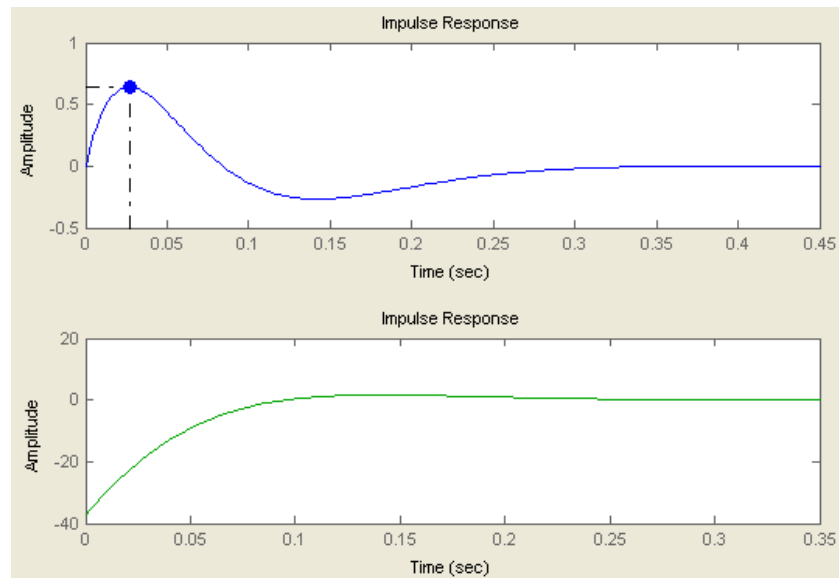


Figura 4.8 Rechazo ante una perturbación tipo impulso en el péndulo

Considerando el compensador mostrado en la Figura 4.7 se procede a calcular los parámetros del controlador PID, k_p , k_i y k_d , como se muestra a continuación:

$$C_1(s) = \frac{k_{d1}s^2 + k_{p1}s + k_{i1}}{s} \quad (4.3)$$

$$C_1(s) = k_{d1} \frac{s^2 + (k_{p1}/k_{d1})s + (k_{i1}/k_{d1})}{s}$$

$$C_1(s) = k_{d1} \frac{(s + z_1)(s + z_2)}{s} \quad (4.4)$$

Donde:

$$k_{p1} = k_{d1}(z_1 + z_2) \quad (4.5)$$

$$k_{i1} = k_{d1}z_1z_2 \quad (4.6)$$

Aplicando las expresiones (4.4), (4.5) y (4.6) se obtiene los siguientes valores:

$$k_{d1} = 0.6247$$

$$k_{p1} = 25.8634$$

$$k_{i1} = 244.4404$$

Utilizando MATLAB se procede a graficar la respuesta simulada, utilizando el siguiente código:

```
figure();
kd = 0.62472; z1 = 26.8; z2 = 14.6; kp= kd*(z1+z2); ki =
kd*z1*z2; %PID
impulse((8*pi/180)*feedback(tf([kd kp ki],[1
0])*Drive_Gain*G1,1))
```

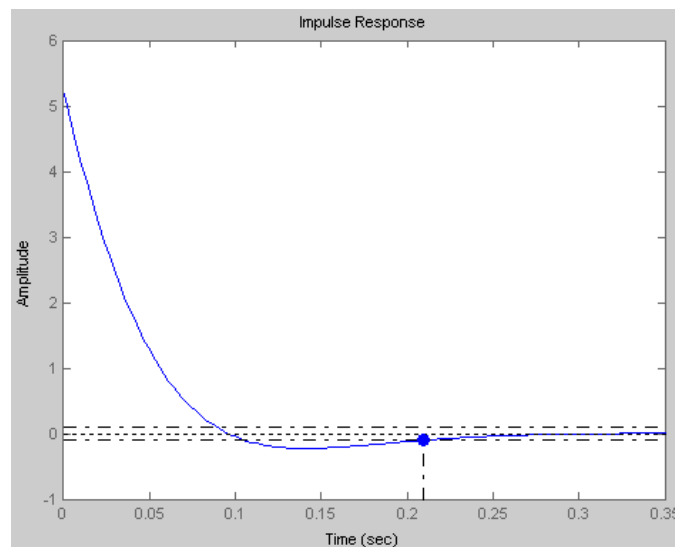


Figura 4.9 Rechazo ante una perturbación tipo impulso en el péndulo

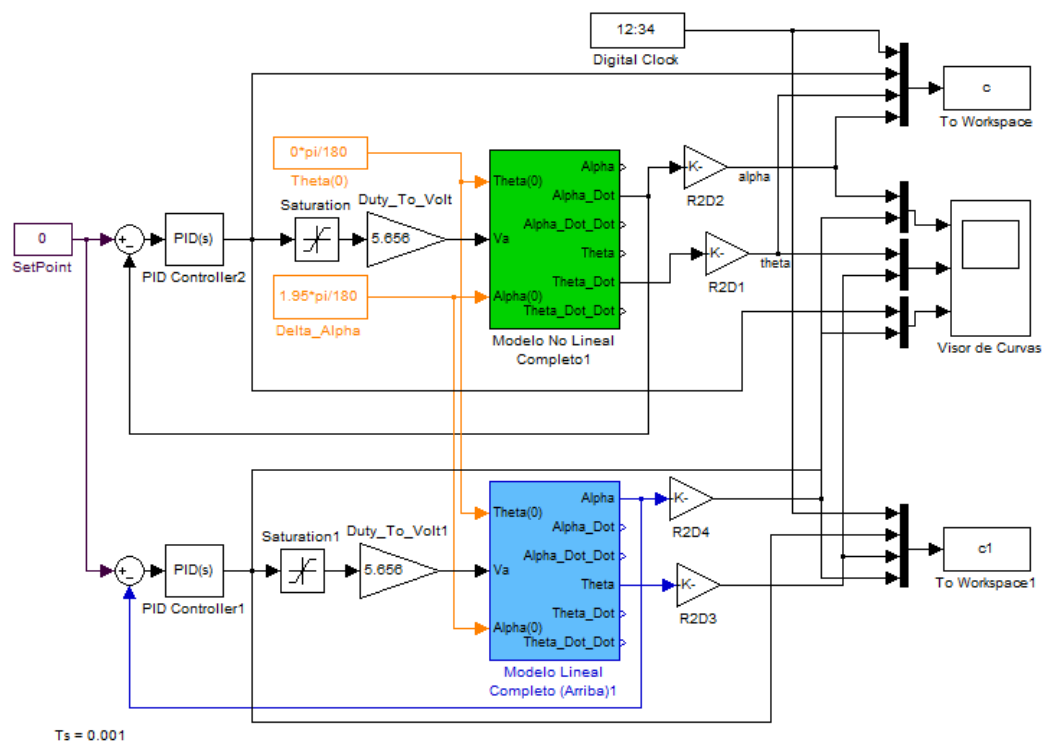


Figura 4.10 Control PID lineal del péndulo en Simulink

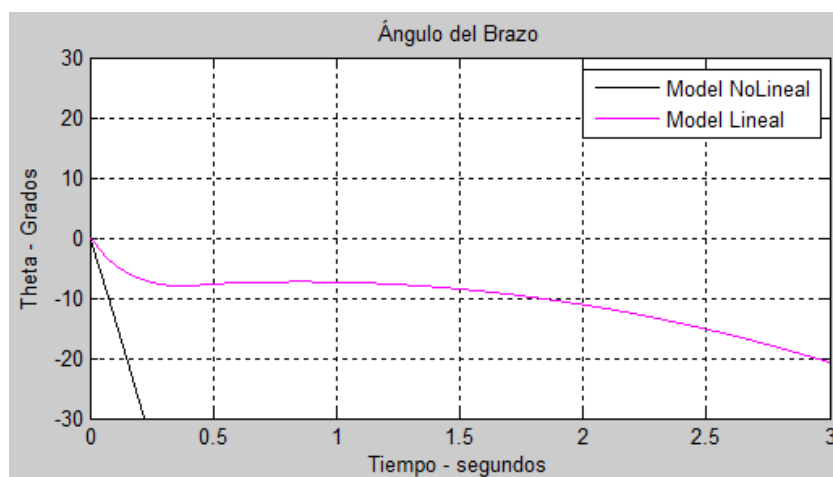


Figura 4.11 Respuesta del control PID del péndulo ante condiciones iniciales (Ángulo del brazo)

Para verificar la dinámica del péndulo y la respuesta del controlador PID lineal diseñado en sisotool, se procede a realizar el esquema en Simulink tal como se muestra en la figura 4.10.

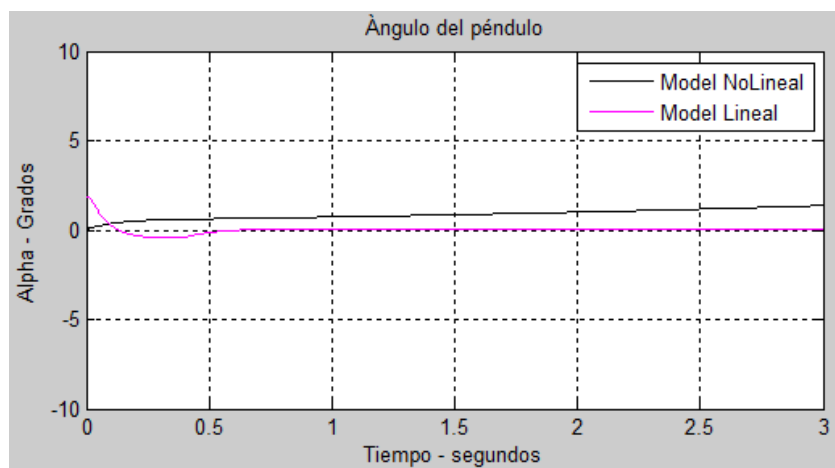


Figura 4.12 Respuesta del control PID del péndulo ante condiciones iniciales (Ángulo del péndulo)

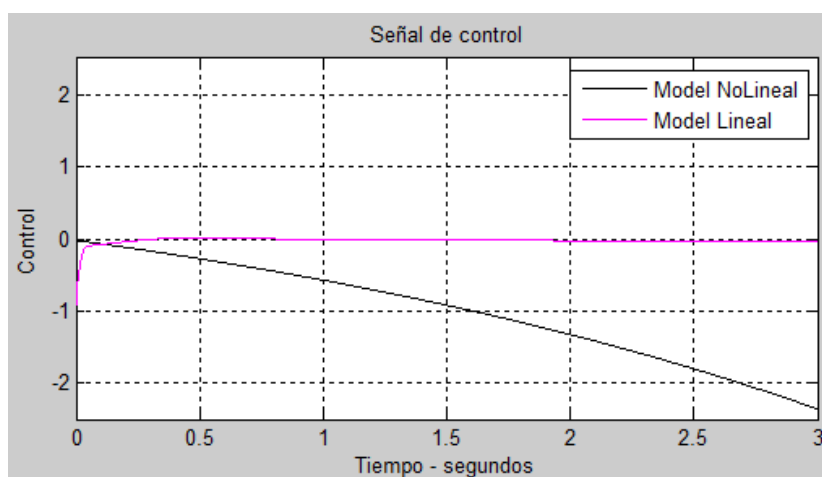


Figura 4.13 Respuesta del control PID del péndulo ante condiciones iniciales (Señal de control)

En la Figura 4.12 se puede apreciar que el ángulo del péndulo se mantiene en 0° a pesar de iniciar con un valor distinto, esto deja en claro que el controlador PID se encuentra operativo.

En cuanto al ángulo del brazo, se puede notar que al pasar del tiempo su valor incrementa sin control, figura 4.13, esto implica que el brazo girará hasta llegar al límite y se detendrá antes de colisionar con la estructura.

Dado esta situación se hace relevante diseñar un controlador para mantener el brazo en una posición fija y así el péndulo está funcionando según los requerimientos del usuario.

Los valores del controlados ajustados previamente permiten operar en un rango de $\pm 5^\circ$ en α sin que la señal de control se sature, caso contrario el controlador no será efectivo.

4.2 Diseño del controlador PI Lineal para el control del brazo

Dado que el objetivo principal del proyecto consiste en mantener el péndulo en posición vertical arriba, se diseñarán dos controladores tipo PID para el péndulo y PI para el brazo. Para el desarrollo se consideran los puntos de operación $\alpha = 0^\circ$ y $\theta = 0^\circ$, esto representa trabajar con el modelo linealizado en dicho punto, por ende si se quisiera que el brazo esté en otro punto de operación se necesitaría linealizar y analizar nuevamente el sistema.

Para controlar las posiciones angulares del brazo y el péndulo, se considerará el siguiente esquema de bloques:

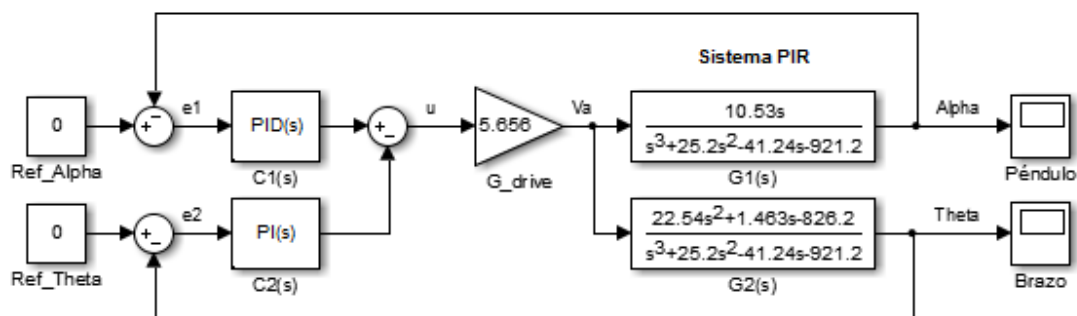


Figura 4.14 Esquema de bloques para control de posición.

Dado que el problema del péndulo es de tipo regulatorio, ya que la referencia siempre va a ser 0 (posición vertical arriba), se considerará cómo valor de referencia el valor deseado para la posición del brazo. Para hacer que el brazo se mantenga en el centro se trabajará en torno a $\beta = 0^\circ$, y se redibujará el diagrama de bloques tal como se muestra en la figura 4.15.

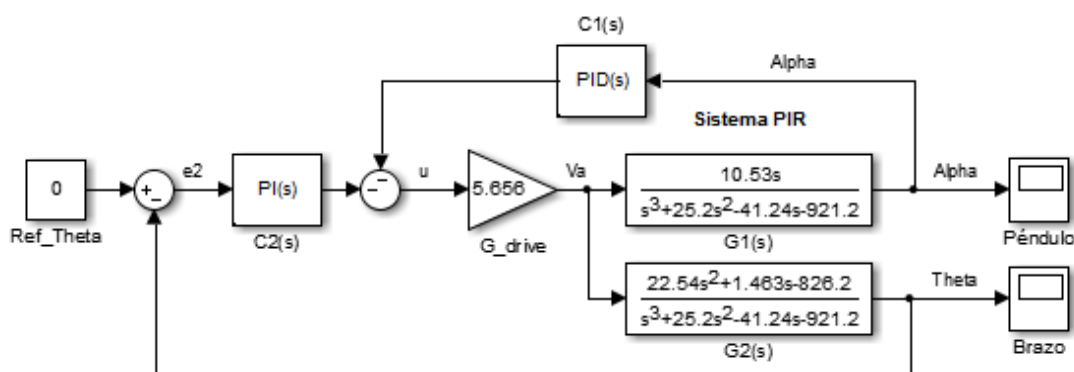


Figura 4.15 Reducción de bloques para control de posición.

Seguidamente se procede a encontrar la planta equivalente para el diseño del controlador PI de posición del brazo.

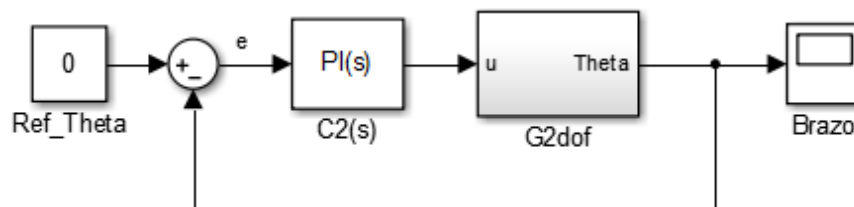


Figura 4.16 Control PI de posición del brazo.

Para obtener el bloque G_{2dof} se utiliza álgebra de bloques como se indica a continuación:

$$G_{2dof}(s) = -\left(\frac{G_{drive}}{1 + G_1 C_1 G_{drive}}\right) G_2 \quad (4.7)$$

$$G_{2dof}(s) = \frac{-127.5s^2 - 8.273s + 4673}{s^4 + 62.4s^3 + 1499s^2 + 13630s} \quad (4.8)$$

Se procede a ajustar el controlador para la posición del brazo utilizando la herramienta SISOTOOL:

```

%% Análisis en lazo cerrado del controlador PID doble
Drive_Gain = 5.656; %Ganancia aproximada en la tarjeta de
potencia
kd = 0.62472; z1 = 26.8; z2 = 14.6; kp= kd*(z1+z2); ki = kd*z1*z2;
%PID
C1 = tf([kd kp ki],[1 0]); %Controlador PID del ángulo del péndulo
G2dof=minreal((-feedback(Drive_Gain,C1*G1))*G2)
sisotool(G2dof);

```

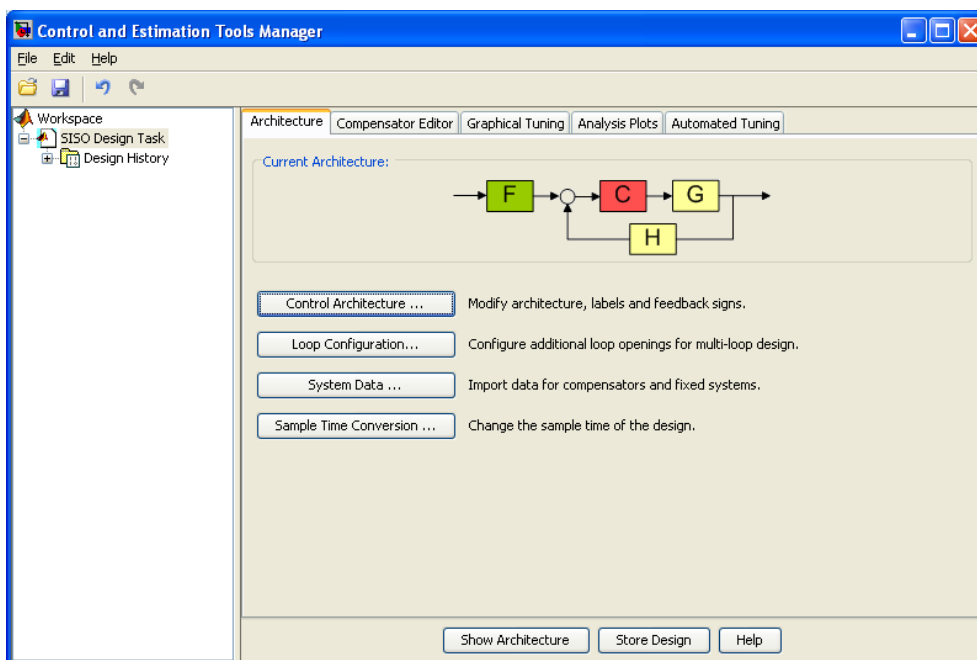


Figura 4.17 Herramienta SISO para diseño de controlador PI del brazo.

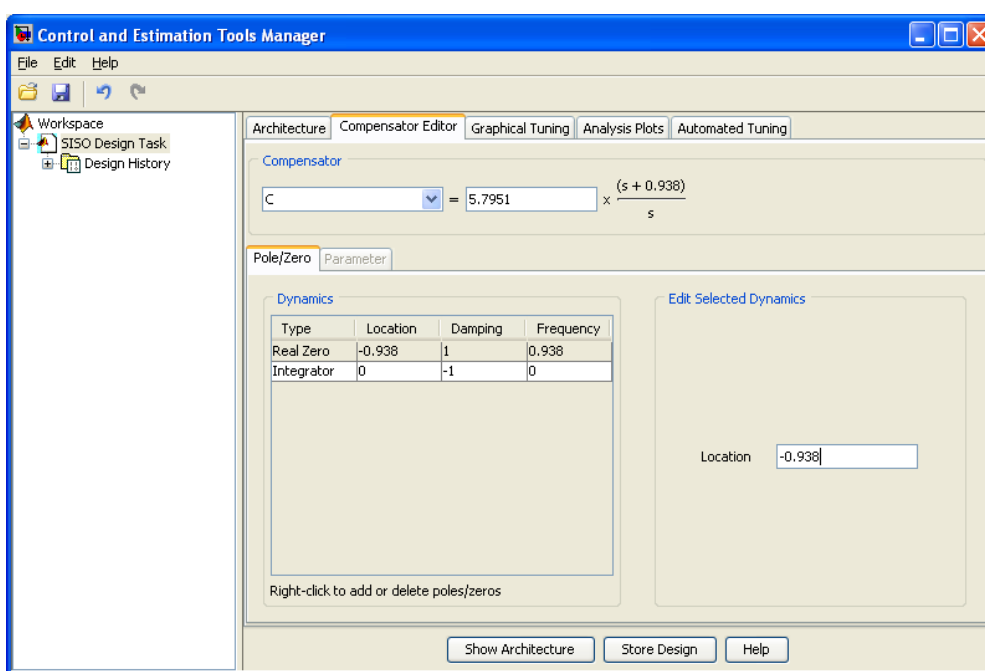


Figura 4.18 Controlador PI ajustado para posicionamiento del brazo.

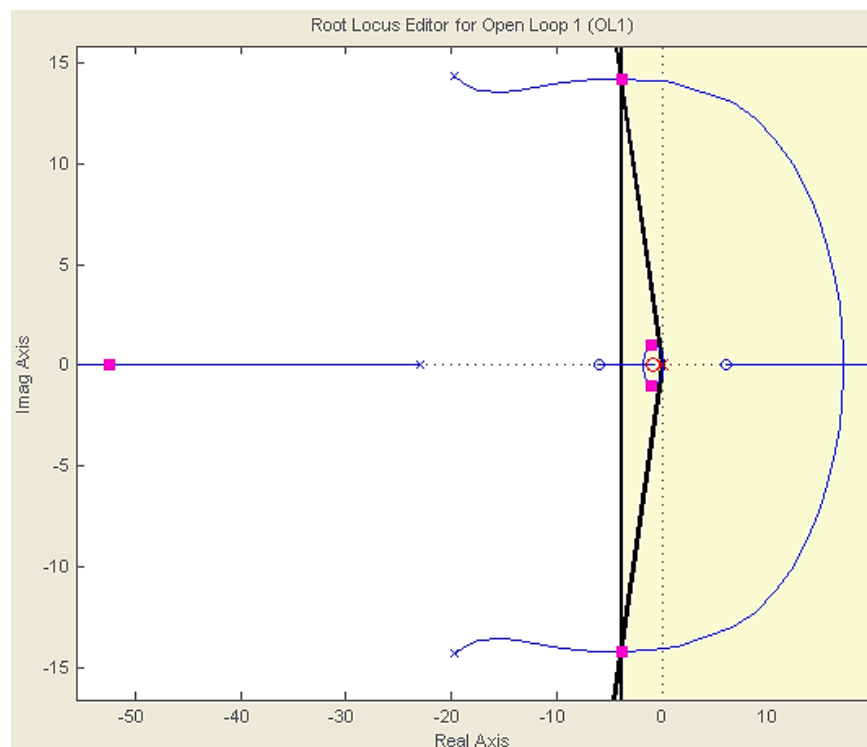


Figura 4.19 Lugar de las raíces del controlador PI lineal del brazo.

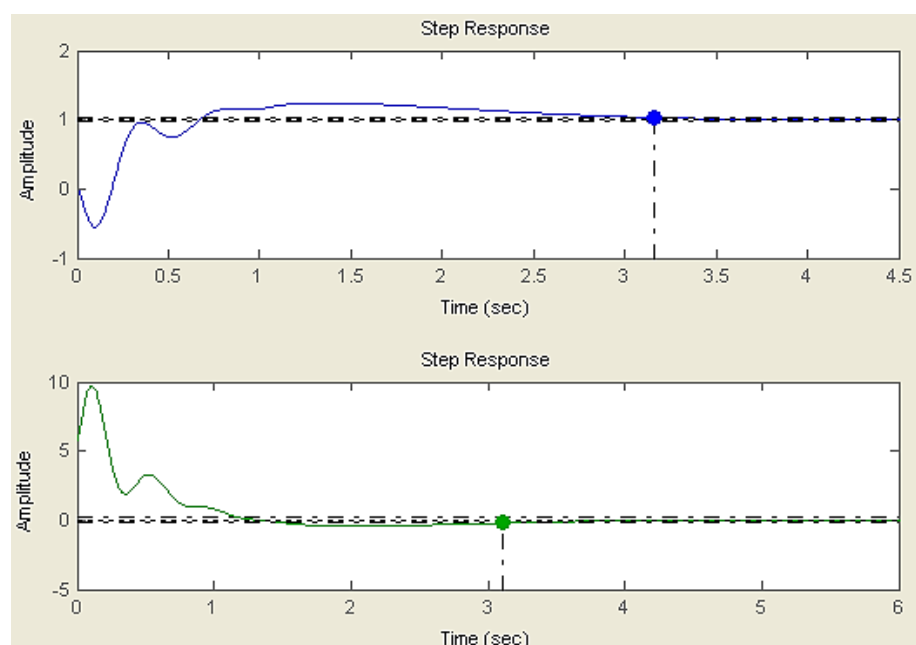


Figura 4.20 Respuesta estimada del controlador PI lineal.

Considerando el compensador mostrado en la Figura 4.18 se procede a calcular los parámetros del controlador PI, k_p y k_i , como se muestra a continuación:

$$C_2(s) = \frac{k_{p2}s + k_{i2}}{s} \quad (4.9)$$

$$C_2(s) = k_{p2} \frac{s + (k_{i2} / k_{p2})}{s}$$

$$C_2(s) = k_{p2} \frac{(s + z)}{s} \quad (4.10)$$

Donde:

$$k_{i2} = k_{p2}z \quad (4.11)$$

Aplicando las expresiones (4.10) y (4.11) se obtiene los siguientes valores:

$$k_{p2} = 5.7951$$

$$k_{i2} = 5.4358$$

Considerando el siguiente código se procede a verificar la respuesta del controlador desde la ventana de comando de MatLab:

```
kp=5.7951;
ki=5.4358;
C2 = tf([kp ki],[1 0]); %Controlador PI del ángulo del brazo
step(2*pi/180*feedback(C2*G2dof,1))
```

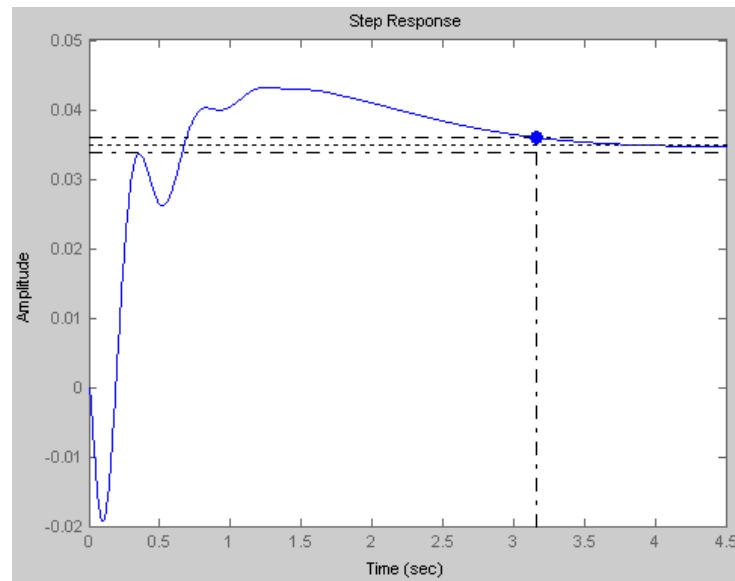


Figura 4.21 Respuesta estimada del controlador PI lineal desde Matlab.

Luego de ajustar los controladores, PID y PI, se procede a comprobar su efectividad en Simulink realizando una simulación de la respuesta del modelo lineal y no lineal de la planta utilizando la estructura mostrada en la Figura 4.22. Para simular los modelos matemáticos, hay que tener en cuenta los controladores ajustados:

$$C_1(s) = \frac{0.6247s^2 + 25.86s + 244.4}{s} \quad (4.12)$$

$$C_2(s) = \frac{5.795s + 5.436}{s} \quad (4.13)$$

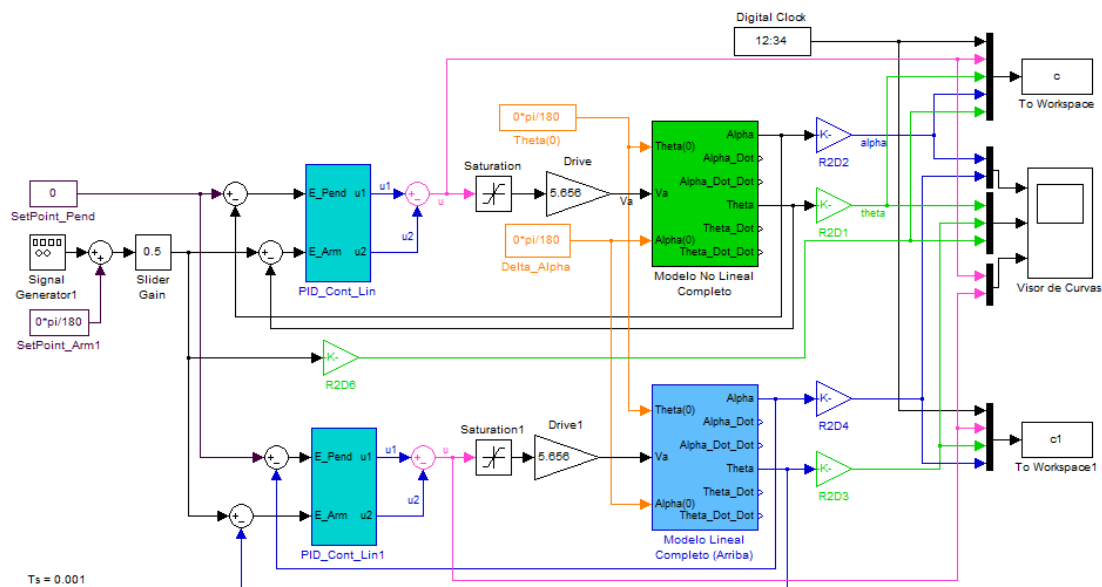


Figura 4.22 Diagrama de los controladores lineales en Simulink

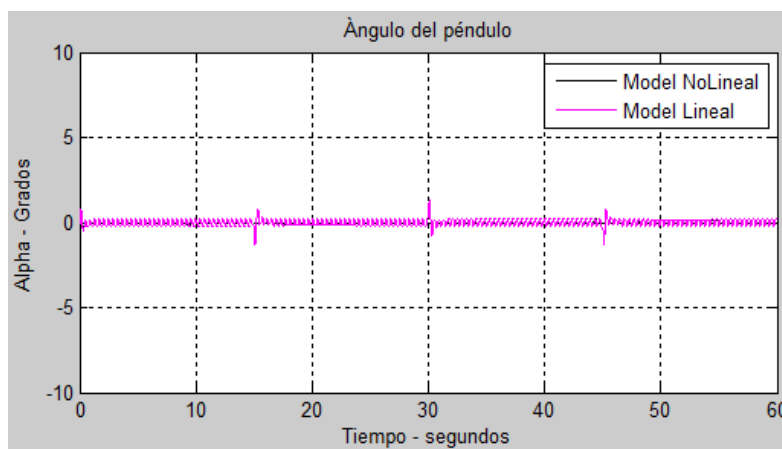


Figura 4.23 Respuesta del ángulo del péndulo ante cambios de referencia en el brazo.

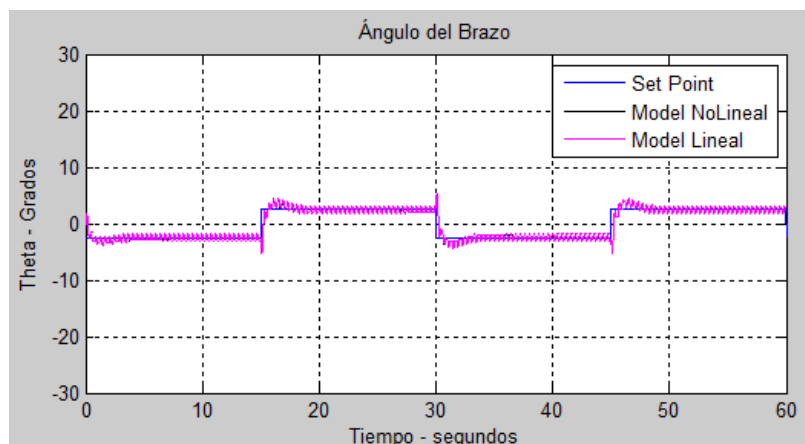


Figura 4.24 Respuesta del ángulo del brazo ante cambios de referencia.

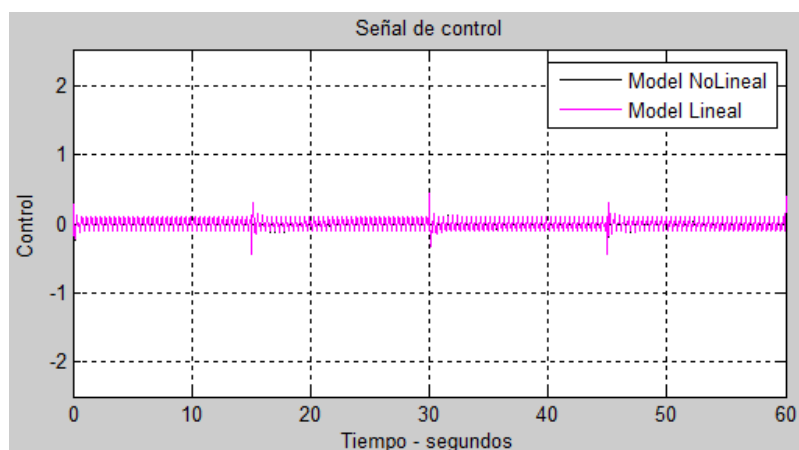


Figura 4.25 Señal de referencia ante cambios de referencia en el brazo.

Como se puede apreciar en las gráficas 4.23, 4.24 y 4.25, la dinámica del modelo lineal y del modelo no lineal se controla de forma adecuada al aplicar los controladores lineales PID y PI. Cabe mencionar que este tipo de controladores son adecuados para valores de referencia cercanos al punto de operación utilizado en el diseño.

CAPÍTULO 5

5. DISEÑO DEL CONTROLADOR NO LINEAL

En esta sección se realiza el diseño del controlador PI no lineal para regular la posición angular del brazo y se mantendrá el mismo controlador PID lineal para equilibrar el péndulo.

5.1 Diseño del controlador PI No lineal para el control del brazo.

Para el diseño del controlador no lineal se parte del diseño basado en el sistema linealizado y a través del método de Ziegler-Nichols en régimen frecuencial para el ajuste de los parámetros del compensador

$(K_1(U), K_2(U))$, nótese que estos parámetros se encuentran parametrizados con respecto a la señal de entrada U , para aquello debemos encontrar el punto de equilibrio para la entrada u y salida θ .

Primero se asigna a cada señal física del sistema PIR como son $[\alpha \ \dot{\alpha} \ \theta \ \dot{\theta}]$ a variables de estado $[x_1 \ x_2 \ x_3 \ x_4]$ respectivamente, entonces la representación en espacio de estado del sistema PIR se muestra en (5.1). Considere el cambio de variables mostrado en (3.39) para minimizar la expresión.

$$\begin{aligned}
 \frac{d}{dt}x_1 &= x_2 \\
 \frac{d}{dt}x_2 &= \frac{k_t k_2 \cos x_1}{R_a (k_1 k_5 - k_2 k_4 \cos^2 x_1)} u - \frac{k_2 \cos x_1 (k_b k_t + R_a B_r)}{R_a (k_1 k_5 - k_2 k_4 \cos^2 x_1)} x_4 \\
 &\quad - \frac{k_2 k_6 (\sin x_1) (\cos x_1) x_2^2}{k_1 k_5 - k_1 k_4 \cos^2 x_1} - \frac{k_5 (B_p x_2 - k_3 \sin x_1)}{k_1 k_5 - k_2 k_4 \cos^2 x_1} \\
 \frac{d}{dt}x_3 &= x_4 \tag{5.1} \\
 \frac{d}{dt}x_4 &= \frac{B_p x_2 - k_3 \sin x_1 + k_1 \left(\frac{d}{dt}x_2 \right)}{k_2 \cos x_1}
 \end{aligned}$$

$$y = [x_1 \ x_3]^T$$

Con la representación (5.1) se calcula el punto de equilibrio, para lo cual se cumple que la derivada de los estados $\dot{x} = 0$ y parametrizando en función de la señal de control $u = U$ se tiene:

$$u = U \quad x_1 = X_1(U) = \tan^{-1}\left(-\frac{k_t k_2}{k_5 k_3 R_a} U\right) \quad x_2 = 0 \quad x_4 = 0 \quad (5.2)$$

Con el punto de equilibrio encontrado (5.2) se procede a linealizar el sistema como se indica a continuación.

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{X, U} = \begin{bmatrix} \frac{\partial f_1(x, u)}{\partial x_1} & \frac{\partial f_1(x, u)}{\partial x_2} & \frac{\partial f_1(x, u)}{\partial x_3} & \frac{\partial f_1(x, u)}{\partial x_4} \\ \frac{\partial f_2(x, u)}{\partial x_1} & \frac{\partial f_2(x, u)}{\partial x_2} & \frac{\partial f_2(x, u)}{\partial x_3} & \frac{\partial f_2(x, u)}{\partial x_4} \\ \frac{\partial f_3(x, u)}{\partial x_1} & \frac{\partial f_3(x, u)}{\partial x_2} & \frac{\partial f_3(x, u)}{\partial x_3} & \frac{\partial f_3(x, u)}{\partial x_4} \\ \frac{\partial f_4(x, u)}{\partial x_1} & \frac{\partial f_4(x, u)}{\partial x_2} & \frac{\partial f_4(x, u)}{\partial x_3} & \frac{\partial f_4(x, u)}{\partial x_4} \end{bmatrix}_{X_1(U), X_2(U), X_3(U), X_4(U), U}$$

$$B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{X, U} = \begin{bmatrix} \frac{\partial f_1(x, u)}{\partial u} \\ \frac{\partial f_2(x, u)}{\partial u} \\ \frac{\partial f_3(x, u)}{\partial u} \\ \frac{\partial f_4(x, u)}{\partial u} \end{bmatrix}_{X_1(U), X_2(U), X_3(U), X_4(U), U}$$

$$\begin{aligned}
 A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ a & b & 0 & c \\ 0 & 0 & 0 & 1 \\ d & e & 0 & f \end{bmatrix} \\
 B &= \begin{bmatrix} 0 \\ h \\ 0 \\ i \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\
 D &= 0
 \end{aligned} \tag{5.3}$$

Para representar las matrices (A, B, C, D) en la expresión (5.3) se ha utilizado el siguiente cambio de variables como se indica a continuación:

$$\begin{aligned}
 a_1 &= k_b k_t + B_r R_a \\
 a_2 &= k_1 k_5 - \frac{k_2 k_4}{a_3} \\
 a_3 &= \left(\frac{a_4}{R_a^2 k_3^2 k_5^2} + 1 \right) \\
 a_4 &= U^2 k_2^2 k_t^2 \\
 a &= \frac{\frac{R_a k_3 k_5}{\sqrt{a_3}} + \frac{a_4}{R_a k_3 k_5 \sqrt{a_3}}}{R_a a_2} \\
 b &= -\frac{B_p k_5}{a_2} \\
 c &= -\frac{k_2 a_1}{R_a \sqrt{a_3} a_2}
 \end{aligned}$$

$$d = - \frac{\frac{R_a k_2 k_3 k_4^2}{a_3} + R_a k_1 k_3 k_4 k_5 - \frac{2R_a k_1 k_3 k_4 k_5}{a_3} - U k_1 k_2 k_4 k_t \sin\left(2 \tan^{-1}\left(\frac{U k_2 k_t}{R_a k_3 k_5}\right)\right)}{R_a a_2^2}$$

$$e = - \frac{B_p k_4}{\sqrt{a_3 a_2}} \quad (5.4)$$

$$f = - \frac{k_1 a_1}{R_a a_2}$$

$$b_1 = k_1 k_5 - \frac{k_2 k_4}{b_2}$$

$$b_2 = \left(\frac{U^2 k_2^2 k_t^2}{R_a^2 k_3^2 k_5^2} + 1 \right)$$

$$h = \frac{k_2 k_t}{R_a \sqrt{b_2 b_1}}$$

$$i = \frac{k_1 k_t}{R_a b_1}$$

El método de Ziegler-Nichols de respuesta crítica en régimen frecuencial consiste en determinar los parámetros $P_0(U)$ $K_0(U)$, analíticamente estos valores se calculan a partir de la función de transferencia $G_U(s)$ por lo cual se debe transformar la representación de espacio de estado dada en (5.3) a función de transferencia a través de la ecuación (5.5).

$$G_U(s) = C(U)[sI - A(U)]^{-1}B(U) \quad (5.5)$$

$$G_U(s) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left(\begin{bmatrix} s & -1 & 0 & 0 \\ -a & s-b & 0 & -c \\ 0 & 0 & s & -1 \\ -d & -e & 0 & s-f \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ h \\ 0 \\ i \end{bmatrix}$$

Realizando los cálculos a partir de la ecuación (5.5) se obtiene:

$$G_U(s) = \begin{bmatrix} \frac{hs + ci - fh}{s^3 + (-b-f)s^2 + (bf - a - ce)s + (af - cd)} \\ \frac{is^2 + (eh - bi)s - ai + dh}{s(s^3 + (-b-f)s^2 + (bf - a - ce)s + (af - cd))} \end{bmatrix} \quad (5.6)$$

Debido a que el péndulo debe estar siempre en posición vertical, el controlador mantendrá los mismos valores que se calcularon en la sección 4.1. Esto implica considerar un esquema de bloques que regule la posición del péndulo y a la vez que regule la posición del brazo en una referencia determinada, lo cual se muestra en la Figura 4.25.

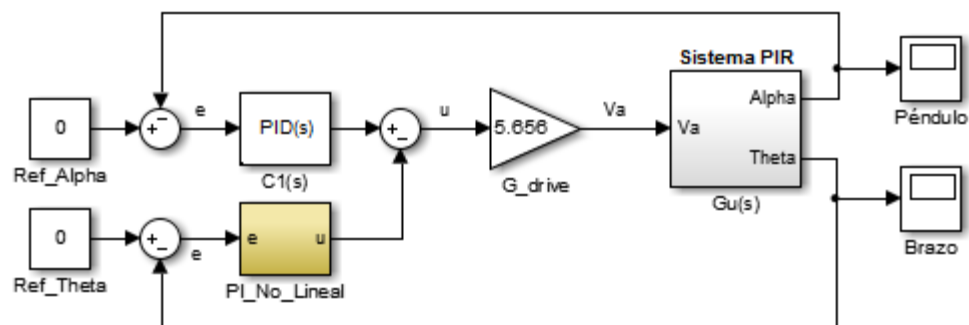


Figura 5.1 Diagrama de bloques para el control No Lineal

Antes de proceder a diseñar el controlador, se reducen los bloques del sistema, de tal manera que se disponga de un esquema realimentado simple. En la figura 4.26 se muestra el resultado de la simplificación de los bloques mencionados.

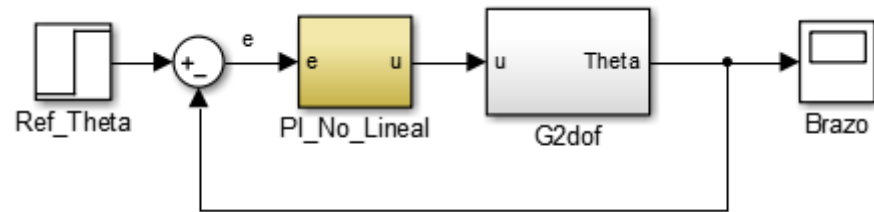


Figura 5.2 Esquema de control No Lineal para posición del brazo

Considerando el diagrama mostrado en la figura 5.2, se debe tener en cuenta que el bloque llamado $G_{2dof}(s)$ representará la nueva planta para el diseño.

$$G_{2dof}(s) = -\frac{c_1s^2 + c_2s + c_3}{s^4 + c_4s^3 + c_5s^2 + c_6s + c_7} \quad (5.7)$$

Donde:

$$c_1 = ik_{drive}$$

$$c_2 = (eh - bi)k_{drive}$$

$$c_3 = (-ai + dh)k_{drive}$$

$$c_4 = (-b - f) + hk_{d1}k_{drive}$$

$$c_5 = (bf - a - ce) + (hk_{p1} + k_{d1}(ci - fh))k_{drive}$$

$$c_6 = (af - cd) + (hk_{i1} + k_{p1}(ci - fh))k_{drive}$$

$$c_7 = k_{i1}(ci - fh)k_{drive}$$

En la expresión 5.7 se puede notar que los coeficientes de la función de transferencia no son valores constantes, en tal caso se encuentran dependientes del valor del punto de operación. Esto muestra una diferencia ante el diseño del controlador lineal del brazo mostrado en el capítulo previo.

Con la ecuación (5.7) se calculan los valores $K_0(U)$ y $\omega_0(U)$ para esto se utilizan las siguientes condiciones:

$$\begin{aligned} \arg |G_{2dof}(j\omega_0(U))| &= -\pi \\ K_0(U) &= \frac{1}{|G_{2dof}(j\omega_0(U))|} \end{aligned} \quad (5.8)$$

Una vez calculados los valores indicados en (5.8) se obtienen los parámetros dados en (5.9).

$$\begin{aligned} P_0(U) &= \frac{2\pi}{\omega_0(U)} = \frac{1}{f_0(U)} \\ K_0(U) &= \frac{1}{|G_{2dof}(j\omega_0(U))|} \end{aligned} \quad (5.9)$$

Primero se reemplaza la variable s por $j\omega$ y empleando la primera condición dada en (5.8) se obtiene:

$$G_{2dof}(s) = -\frac{c_1(j\omega)^2 + c_2(j\omega) + c_3}{(j\omega)^4 + c_4(j\omega)^3 + c_5(j\omega)^2 + c_6(j\omega) + c_7}$$

$$G_{2dof}(s) = -\frac{(-c_1\omega^2 + c_3) + c_2\omega j}{(\omega^4 - c_5\omega^2 + c_7) + (-c_4\omega^3 + c_6\omega)j}$$

$$G_{2dof}(s) = -\left[\frac{(-c_1\omega^2 + c_3)(\omega^4 - c_5\omega^2 + c_7) + c_2\omega(-c_4\omega^3 + c_6\omega)}{(\omega^4 - c_5\omega^2 + c_7)^2 + (-c_4\omega^3 + c_6\omega)^2} + \frac{c_2\omega(\omega^4 - c_5\omega^2 + c_7) - (-c_1\omega^2 + c_3)(-c_4\omega^3 + c_6\omega)}{(\omega^4 - c_5\omega^2 + c_7)^2 + (-c_4\omega^3 + c_6\omega)^2} j \right]$$

$$\tan^{-1} \left| \frac{\text{Im } g(G_{2dof})}{\text{Re } g(G_{2dof})} \right| = -\pi$$

$$\omega = \sqrt{\frac{-(-c_2c_5 + c_1c_6 + c_3c_4) - \sqrt{(-c_2c_5 + c_1c_6 + c_3c_4)^2 - 4(c_2 - c_1c_4)(c_2c_7 - c_3c_6)}}{2(c_2 - c_1c_4)}}$$

(5.10)

Cabe recalcar que este método no es aplicable a sistemas de primer y segundo orden ya que las únicas frecuencias que satisfacen la condición (5.8) son $\omega = 0$ y $\omega = \infty$ las cuales no sirven para calcular los parámetros indicados en (5.9), esta es la razón por la cual se desecha la solución de $\omega = 0$, en cambio la solución negativa de ω tampoco sirve debido a que todo sistema real opera con frecuencia positivas por lo tanto:

$$\omega_0(U) = \sqrt{\frac{-(-c_2c_5 + c_1c_6 + c_3c_4) - \sqrt{(-c_2c_5 + c_1c_6 + c_3c_4)^2 - 4(c_2 - c_1c_4)(c_2c_7 - c_3c_6)}}{2(c_2 - c_1c_4)}}$$

$$P_0(U) = \frac{2\pi}{\omega_0(U)}$$

(5.11)

De igual forma utilizando la segunda condición dada en (5.8) tenemos:

$$|G_{2dof}(\omega_0(U))| = \frac{\sqrt{(-c_1\omega^2 + c_3)^2 + (c_2\omega)^2}}{\sqrt{(\omega^4 - c_5\omega^2 + c_7)^2 + (-c_4\omega^3 + c_6\omega)^2}}$$

(5.12)

$$K_0(U) = \left| \sqrt{\frac{(\omega^4 - c_5\omega^2 + c_7)^2 + (-c_4\omega^3 + c_6\omega)^2}{(-c_1\omega^2 + c_3)^2 + (c_2\omega)^2}} \right|$$

Entonces el controlador lineal parametrizado en función de U esta dado por:

$$G_{2NL}(s) = K_1(s) + \frac{K_2(s)}{s} \quad (5.13)$$

La relación entre los parámetros indicados en (5.9) y los del compensador se describen en la tabla 8.

Tabla 8: Parámetros K1 y K2 para controlador PI del brazo utilizando método de Ziegler-Nichols [12]

Tipo de controlador	Parámetro K1	Parámetro K2	Parámetro K3
P	$0.5K_0(U_1)$	0	0
PI	$0.45K_0(U_1)$	$0.54 \frac{K_0(U_1)}{P_0(U_1)}$	0
PID	$0.5K_0(U_1)$	$1.2 \frac{K_0(U_1)}{P_0(U_1)}$	$0.075K_0(U_1)P_0(U_1)$

Se puede deducir, de la comparación de este compensador linealizado y con ayuda de las expresiones 2.5 y 2.6 se obtiene que las ganancias del controlador PI no lineal están dada por:

$$k_1(z) = K_1(z(t)) = 0.45K_0(z(t)) \quad (5.14)$$

$$k_2(z) = K_2(z(t)) = 0.54 \frac{K_0(z(t))}{P_0(z(t))}$$

En este caso, la generalización no lineal del controlador PI asume la forma siguiente:

$$\begin{aligned} \dot{z}(t) &= k_2(z(t))e(t) \\ u(t) &= z(t) + k_1(z(t))e(t) \end{aligned} \quad (5.15)$$

Donde el error está dado por:

$$e(t) = \theta_{ref} - \theta(t) \quad (5.16)$$

Por lo tanto la ecuación 5.15 se reescribe como:

$$\dot{z}(t) = 0.54 \frac{K_0(z(t))}{P_0(z(t))} (\theta_{ref} - \theta(t)) \quad (5.17)$$

$$u(t) = z(t) + 0.45K_0(z(t))(\theta_{ref} - \theta(t))$$

5.2 Validación del controlador PI no lineal mediante simulación.

Para notar la efectividad del controlador no lineal diseñado en este capítulo, se procede a simular y comparar con el controlador lineal el comportamiento del sistema ante ligeros cambios de la referencia del ángulo del brazo, figura 5.3.

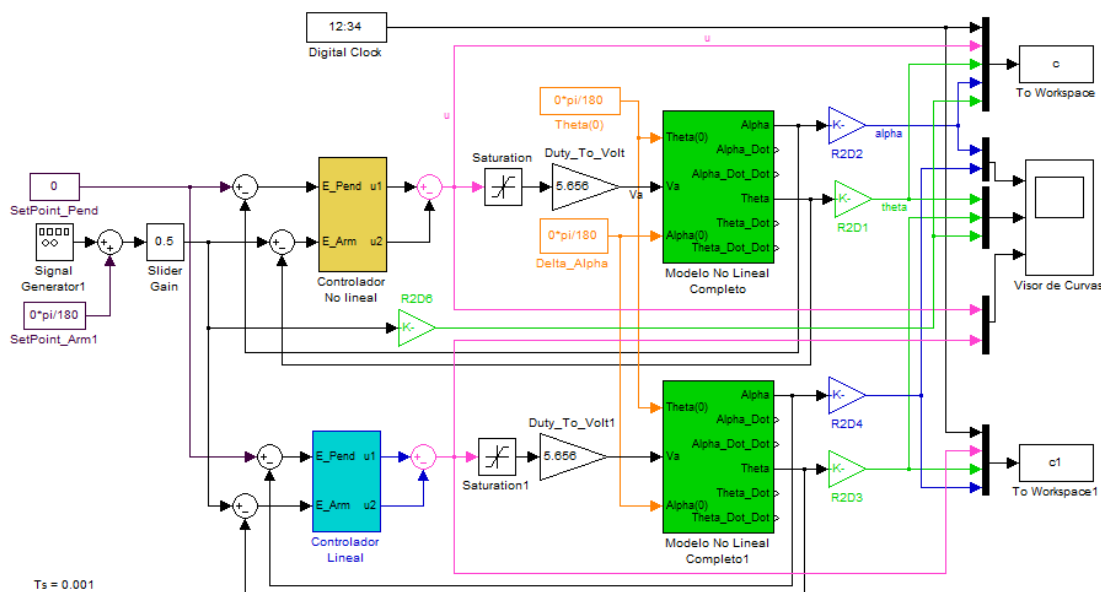


Figura 5.3 Esquema de control No Lineal para posición del brazo en Simulink

Para ver el detalle de los subsistemas utilizados para el cambio de variables en el controlador no lineal para la posición del brazo presentados en la figura 5.4 se puede consultar en los Anexos 13, 14 y 15.

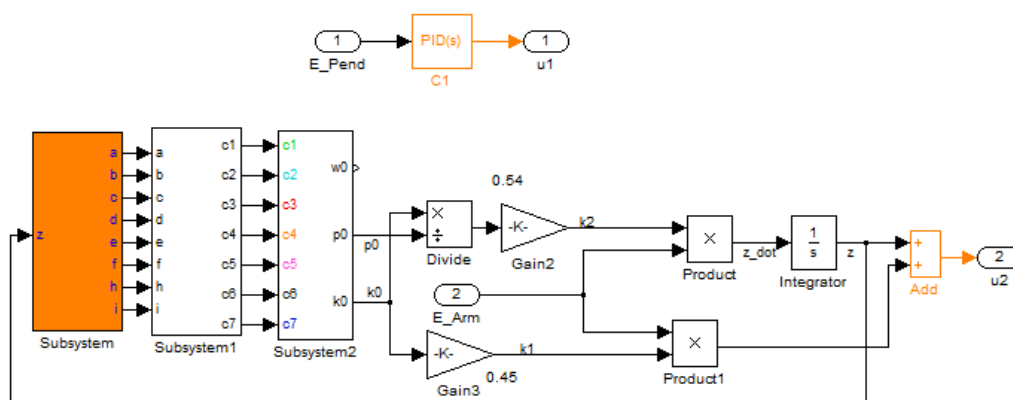


Figura 5.4 Detalle del controlador PI No lineal realizado en Simulink

Para la simulación se emplea como señal de referencia para el brazo una onda rectangular de 2.5° de amplitud y una frecuencia de 0.033 Hz con la finalidad de notar la dinámica del sistema ante cambios instantáneos en la entrada.

A través de la gráfica 5.5 se puede notar que ambos controladores realizan el seguimiento de la referencia angular del brazo mostrando un tiempo de estabilización aproximadamente de 4seg.

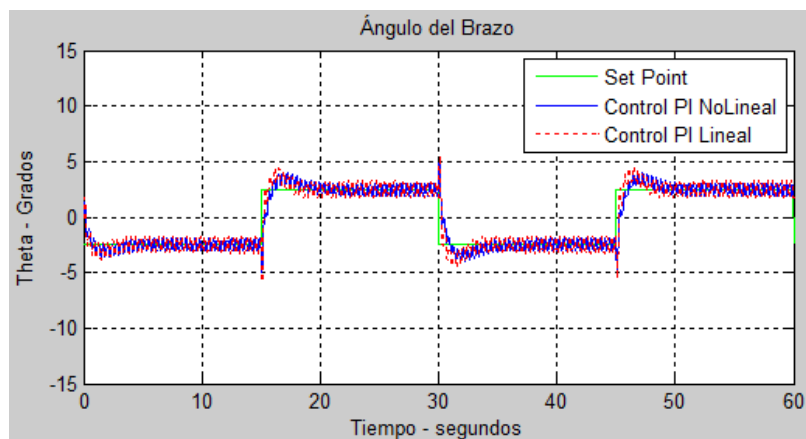


Figura 5.5 Respuesta del brazo utilizando controladores lineales y no lineales

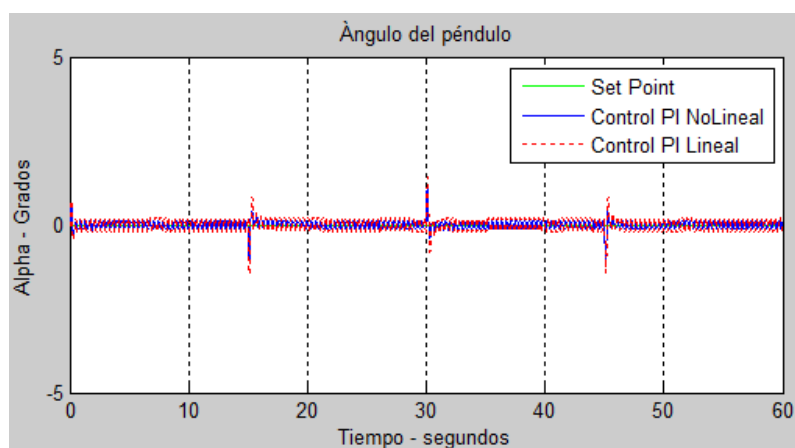


Figura 5.6 Respuesta del péndulo utilizando controladores lineales y no lineales

En la figura 5.6 se aprecia las variaciones que tiene el péndulo ante los cambios de referencia en el brazo; se puede apreciar que el control lineal produce impulsos con una amplitud mayor que en el control no lineal.

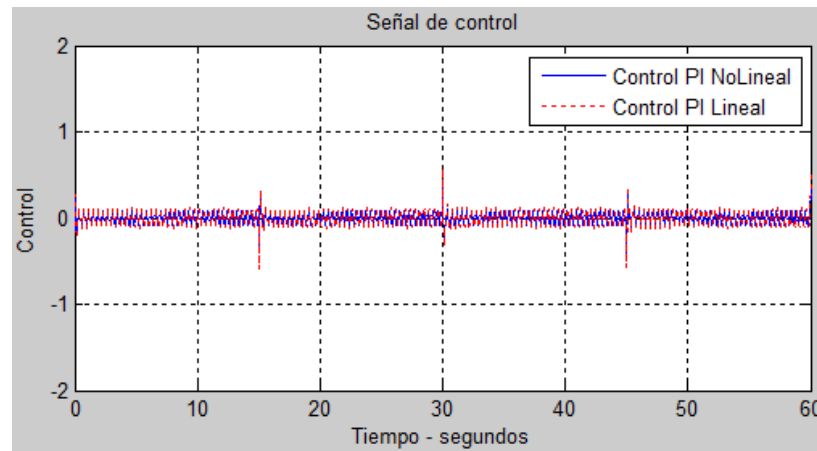


Figura 5.7 Señal de control utilizada en controladores lineales y no lineales

La gráfica 5.7 muestra la existencia de impulsos en la salida del controlador en el momento del cambio de la referencia, lo cual es más notorio en el controlador no lineal. El incremento de estos picos permite ver que el controlador lineal consume más energía que el control no lineal al realizar la regulación de posición.

CAPÍTULO 6

6. IMPLEMENTACIÓN DE PERIFÉRICOS

6.1 Acondicionamiento de señales

Dado que los encoders incrementales ubicados en la planta utilizan niveles de voltaje mayores a 5Vdc, se debe acondicionar las señales digitales obtenidas para luego poder conectarlas a la tarjeta PCI. Para ello se ha diseñado una tarjeta basada en opto acopladores, con la finalidad de aislar las fuentes de alimentación y así mantener menor efecto de ruido en el controlador. Los detalles del circuito electrónico se encuentran en el Anexo 16.

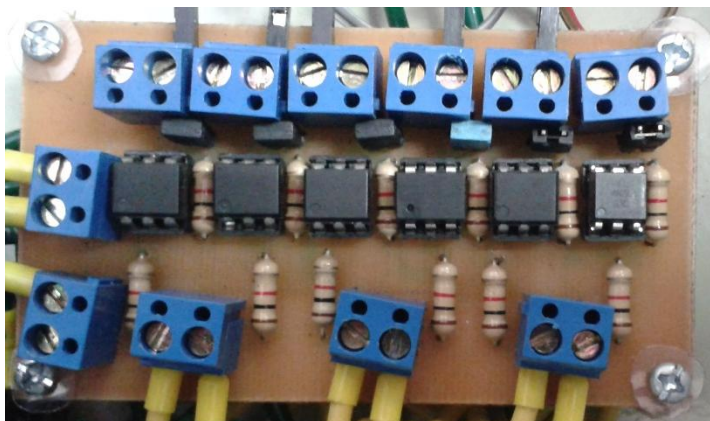


Figura 6.1 Tarjeta de acondicionamiento de señales

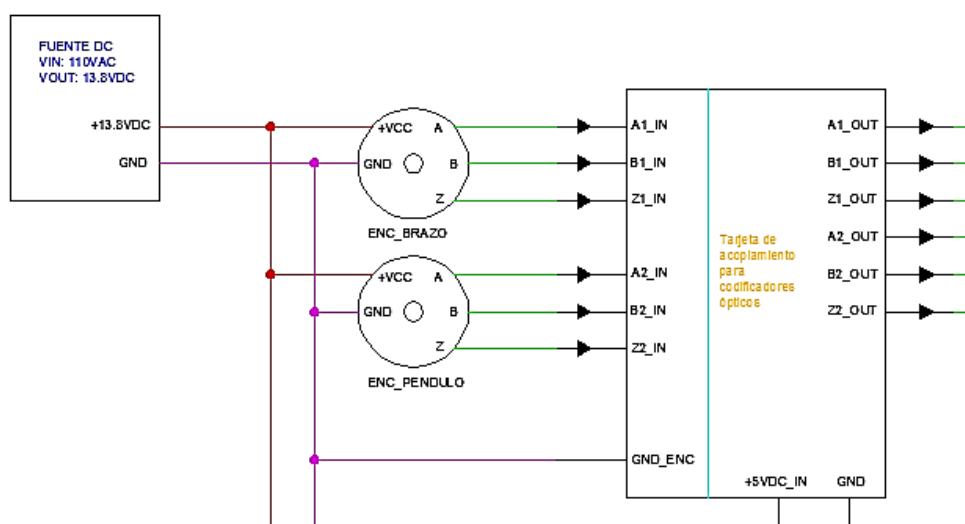


Figura 6.2 Conexión de tarjeta de acondicionamiento de señales

6.2 Adquisición de datos

Luego de acondicionar las respectivas señales del sensor, se procede a contabilizar los pulsos dados y enviarlos hacia la tarjeta PCI con la mayor precisión posible.

Para ello se ha utilizado una tarjeta de entrenamiento basada en microcontroladores llamada Arduino Mega 2560 mostrada en la figura 6.3.

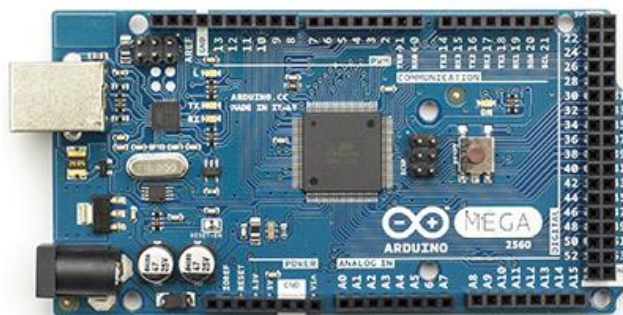


Figura 6.3 Tarjeta Arduino Mega 2560

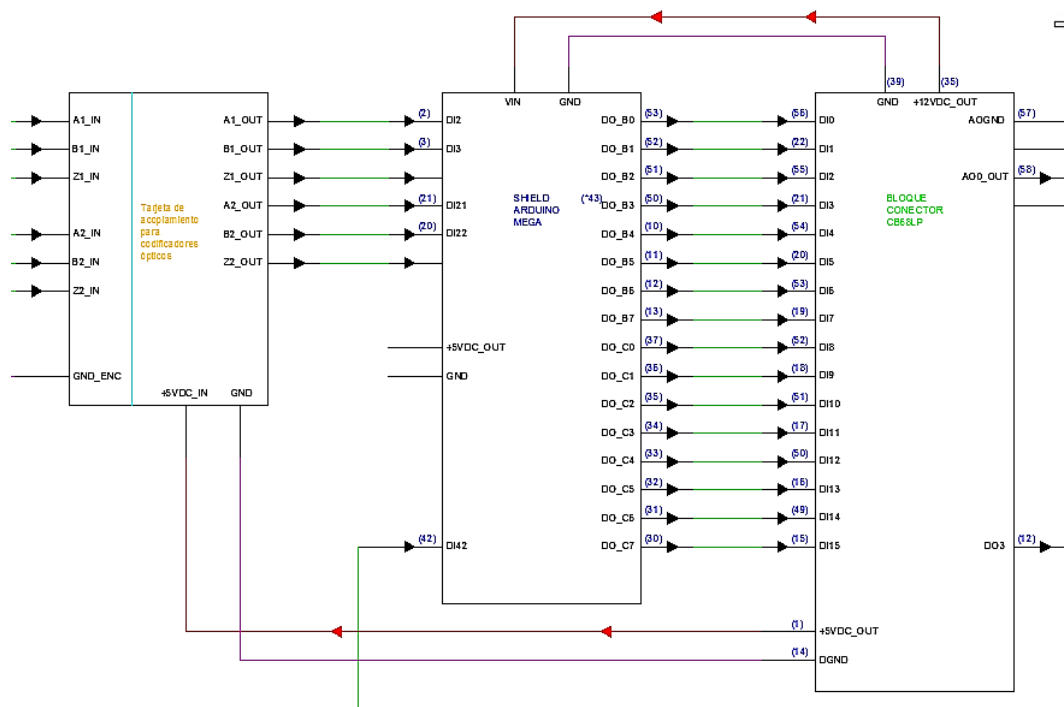


Figura 6.4 Conexiones de tarjeta Arduino Mega

Se seleccionó esta tarjeta debido a sus características de diseño; debido a que se cuenta con 6 interrupciones externas hace posible la conexión de los encoders utilizados y así realizar el conteo de pulsos de forma precisa. Cabe recalcar que al tener 54 puertos de entradas/salidas digitales se permite la conexión de conteo de pulsos con una resolución de 16 bits para cada codificador óptico. [17]

En relación al conteo de pulsos de los sensores se puede mencionar que se utilizó en método X2 para decodificación [18], lo cual se puede apreciar en la figura 6.5.

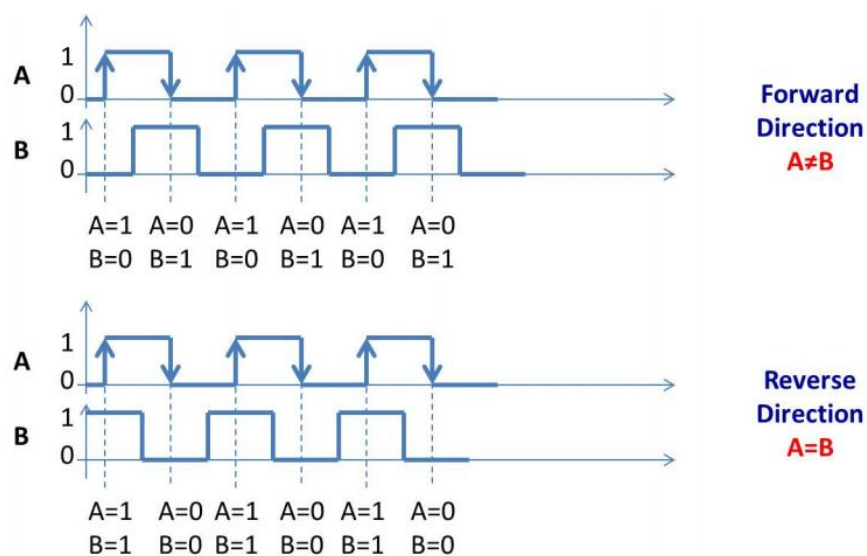


Figura 6.5 Método X2 para lectura de codificadores ópticos [18]

Con respecto al código utilizado en la placa Arduino se puede ver el detalle en el Anexo 11.

Cabe mencionar que se utilizó el bloque conector CB-68LP de National Instruments con la finalidad de realizar el cableado con facilidad, ya que se cuenta con borneras de tornillo, se reduce la posibilidad de hacer cortocircuito al manipular directamente los pines de la tarjeta PCI. Para más detalle del bloque se puede consultar en el Anexo 4.



Figura 6.6 Bloque Conector CB-68LP

Luego de realizar las conexiones mostradas en los diagramas anteriores se procede a crear un bloque en Simulink para medir los ángulos del brazo y el péndulo respectivamente. El detalle del bloque mencionado se muestra en la figura 6.8.



Arm Angle \
Pendulum Angle

Figura 6.7 Bloque para lectura de encoders en Simulink

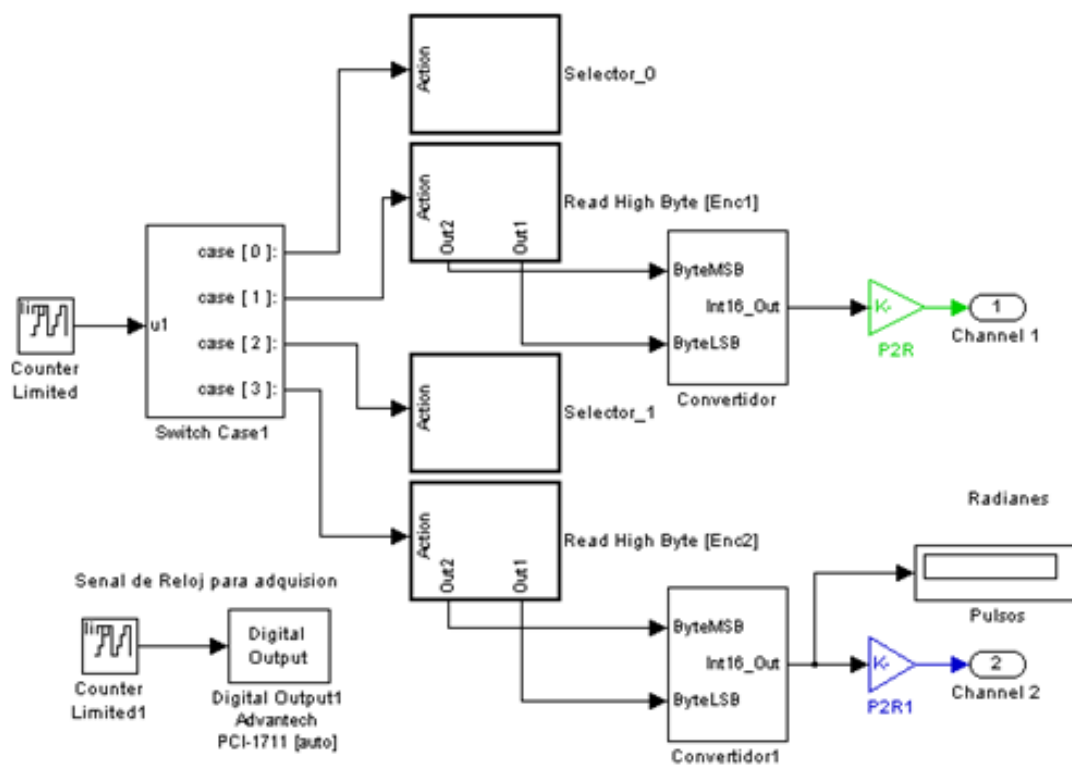


Figura 6.8 Algoritmo X2 para lectura de encoders en simulink

6.3 Interfaz de potencia

El control del motor DC modelado previamente se realiza a través un driver Saberthoot 2x32, en el cual se necesita especificar un voltaje analógico que se encuentra entre 0 y 5Vdc. Para ello se ha construido un bloque en simulink para poder enviar los valores adecuados considerando la curva mostrada en el capítulo 3.

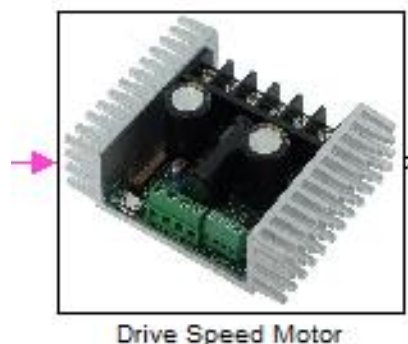


Figura 6.9 Bloque para interfaz con Saberthoot en simulink

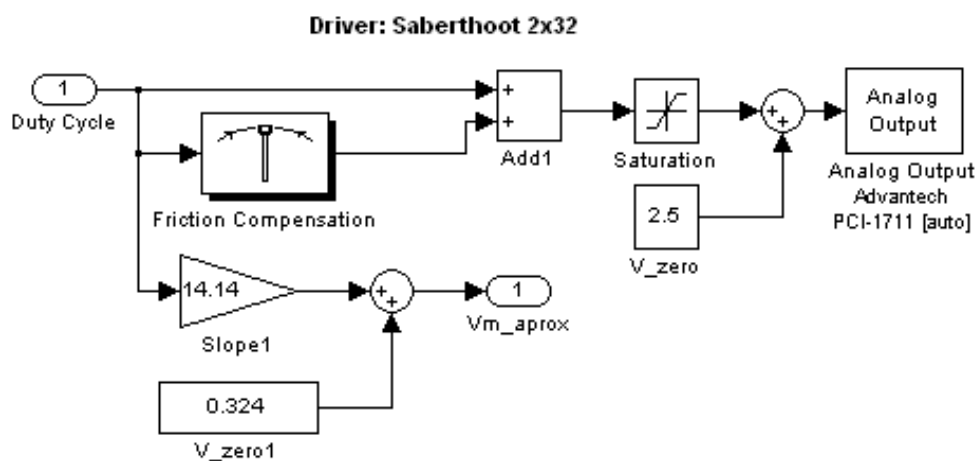


Figura 6.10 Acondicionamiento de señales para interface de potencia

La señal de control u para el motor está definida para un rango de operación de $[-2.5 V \dots + 2.5V]$, dado que controlador del motor funciona bajo esos niveles de voltaje.

Para eliminar los efectos negativos que podrían presentarse por motivo de la fricción estática, se considera una compensación de voltaje que suma valores pequeños a la salida analógica de la tarjeta PCI.

Esto permite mantener una simetría relativa en los dos sentidos de giro del motor, pero lastimosamente al momento de trabajar con señales pequeñas se tendrán voltaje de desplazamiento.

CAPÍTULO 7

7. ANÁLISIS DE RESULTADOS

7.1 Prueba de controladores en la planta real

7.1.1 Evaluación del controlador PID lineal para el péndulo

Para comprobar los diseños realizados en los capítulos 5 y 6, se procede a implementar los controladores en el sistema PIR con la finalidad de recopilar datos y comparar con la simulación. En la figura 7.1 se puede apreciar el control PID lineal implantado sobre la planta.

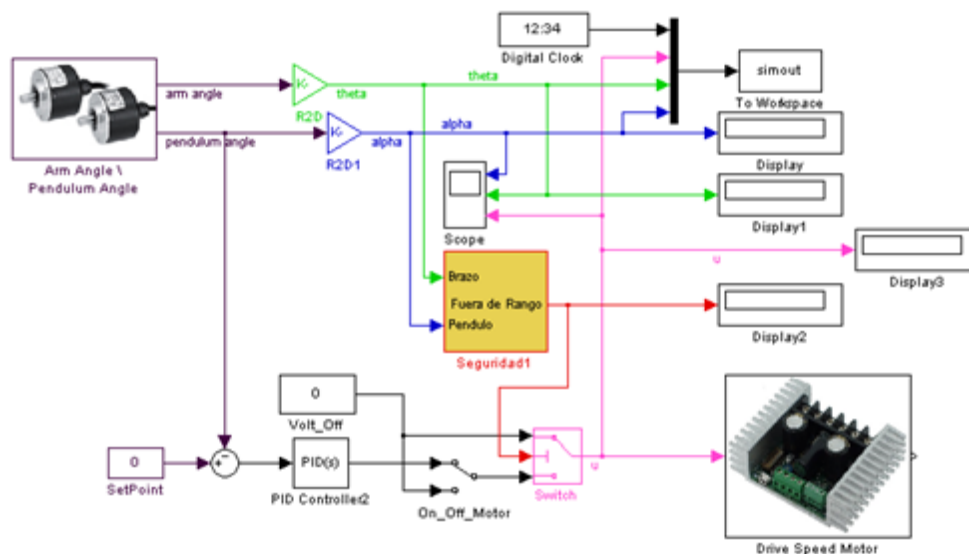


Figura 7.1 Implementación de control PID para el péndulo

Dado que el sistema a controlar podría realizar movimientos rápidos y bruscos, es necesario ubicar bloques de seguridad que limitarán el desplazamiento angular del brazo. Esto ayuda a prevenir colisiones y así conversar el buen estado del equipo.

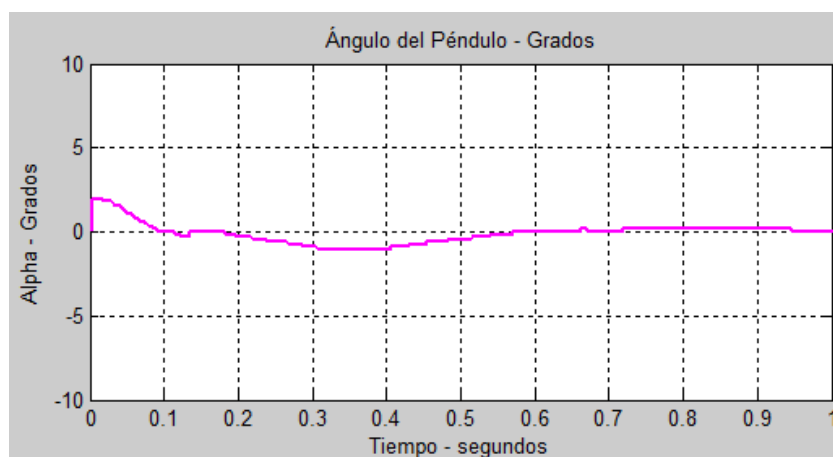


Figura 7.2 Respuesta del péndulo ante controlador PID lineal implementado

En la gráfica 7.2 se puede apreciar la regulación del ángulo del péndulo en el valor de referencia igual a 0° , cabe mencionar que el péndulo inició con una ligera inclinación de 1.5° .

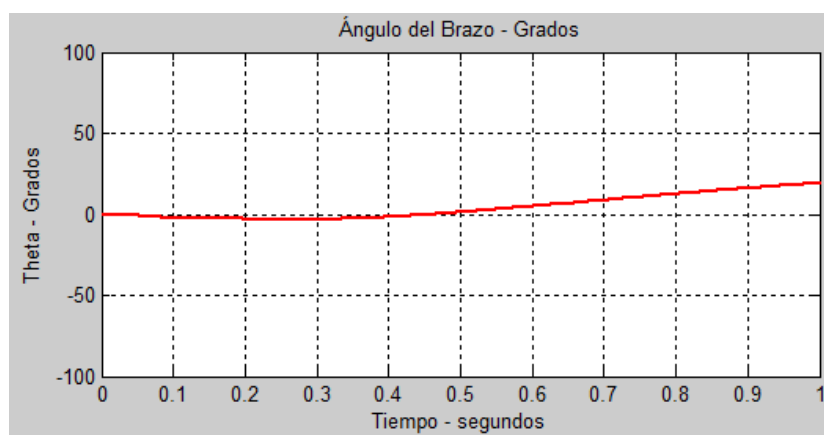


Figura 7.3 Respuesta del brazo para control PID lineal en el péndulo

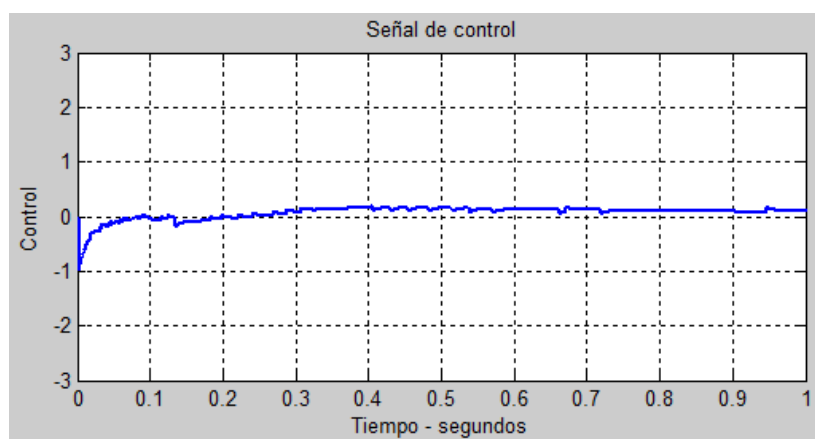


Figura 7.4 Salida del controlador PID lineal implementado en el péndulo invertido

En la gráfica 7.3 se muestra el comportamiento dinámico del brazo cuando el péndulo procede a regular su posición, en ella se puede evidenciar que el brazo se desplaza de forma no controlada.

En la gráfica 7.4 se observa que la señal de control decrementa su valor conforme la salida se acerca a la consigna deseada.

Para fines de comparación se realiza la simulación del modelo lineal tal como se muestra en la figura 4.10. Seguido a esto, se procede a mostrar los resultados de la implementación y la simulación en la misma figura para notar diferencias, figura 7.5, 7.6 y 7.7.

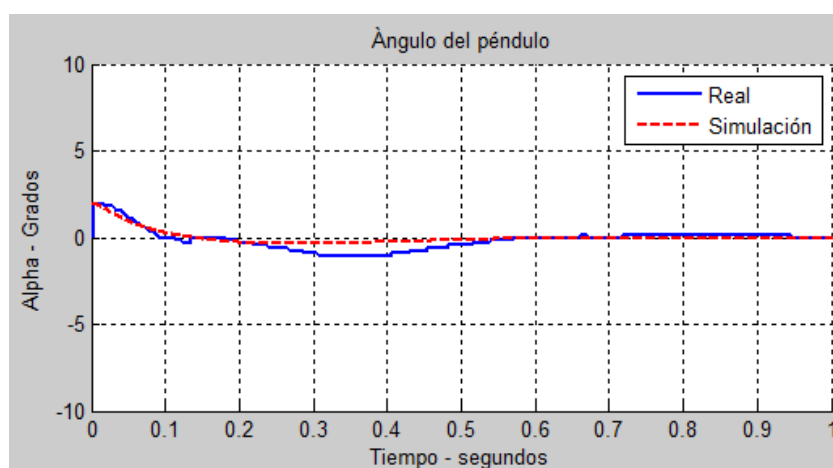


Figura 7.5 Posición angular del péndulo, real y simulación.

En la figura 7.5 se observa la regulación del ángulo del péndulo en 0° gracias al controlador PID lineal. Además en la figura 7.6 se muestra que, tanto en la simulación como en la implementación, el brazo

comienza a girar sin control para tratar de regular la posición del péndulo en el valor de consigna.

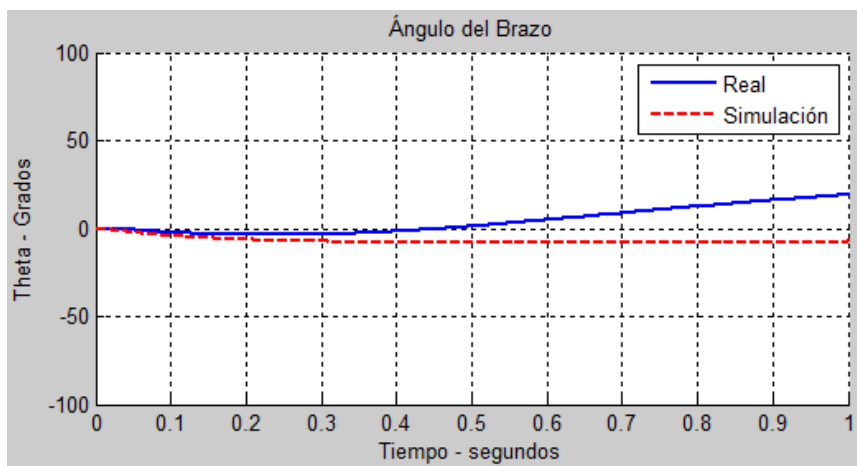


Figura 7.6 Posición angular del Brazo, real y simulación.

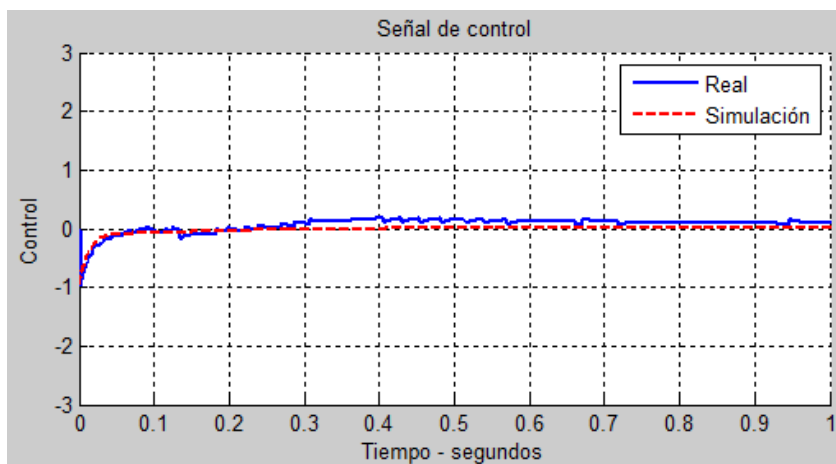


Figura 7.7 Salida del controlador PID lineal, real y simulación.

Según lo observado en la implementación y la simulación, se vuelve necesario la implementación de un controlador de posición para el brazo, dado que sin el mismo se presentarían colisiones con la estructura.

7.1.2 Evaluación del controlador PI lineal para el brazo

En esta sección se procede a evaluar el comportamiento del controlador lineal tipo Proporcional Integral, cabe mencionar que se debe tener en cuenta las expresiones 4.12 y 4.13, las cuales indican los valores ajustados para los controladores PID y PI lineal.

Se consideran diferentes valores de referencia para describir de mejor forma las características del sistema controlado, lo cual se puede apreciar en las figuras 7.8 y 7.9.

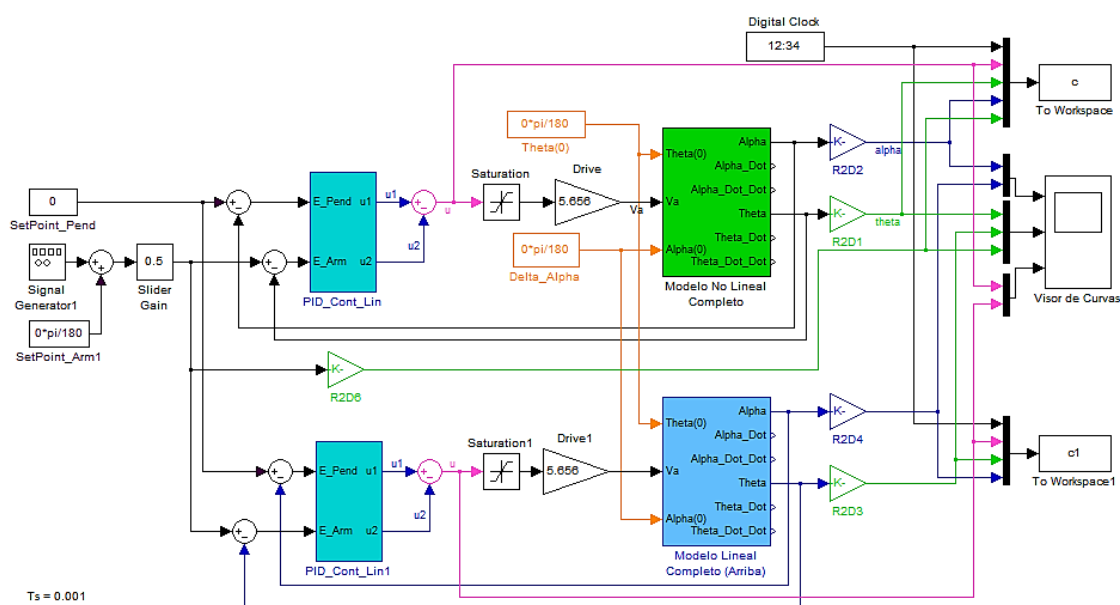


Figura 7.8 Simulación del control PI para el brazo desde Simulink.

Para realizar la simulación y comparación se considerarán los códigos mostrados en los Anexos 9 y 10, considerando varios puntos de operación tal como se muestran en las figuras 7.10 al 7.11.

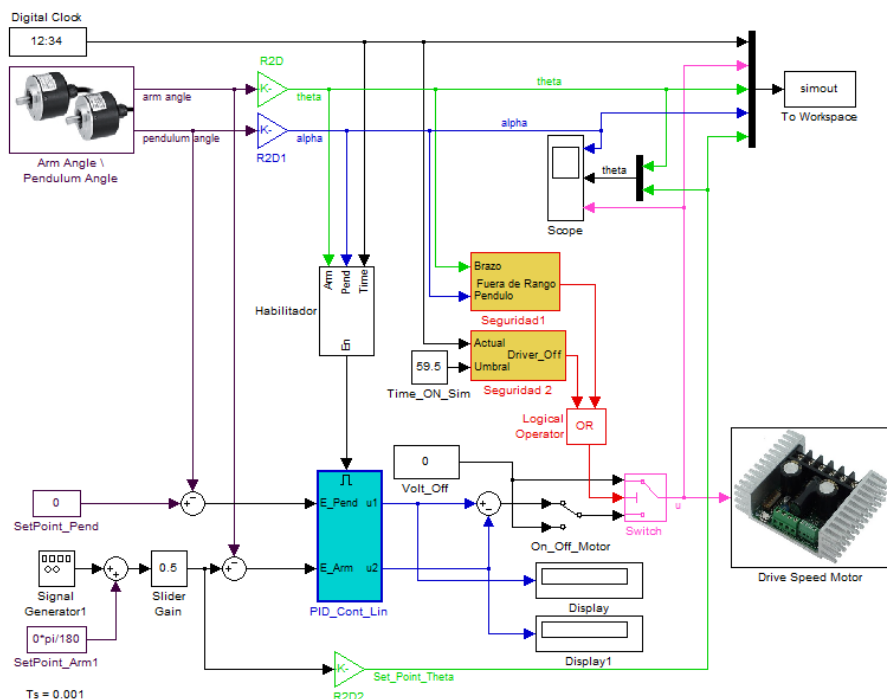


Figura 7.9 Implementación del control PI para el brazo desde Simulink.

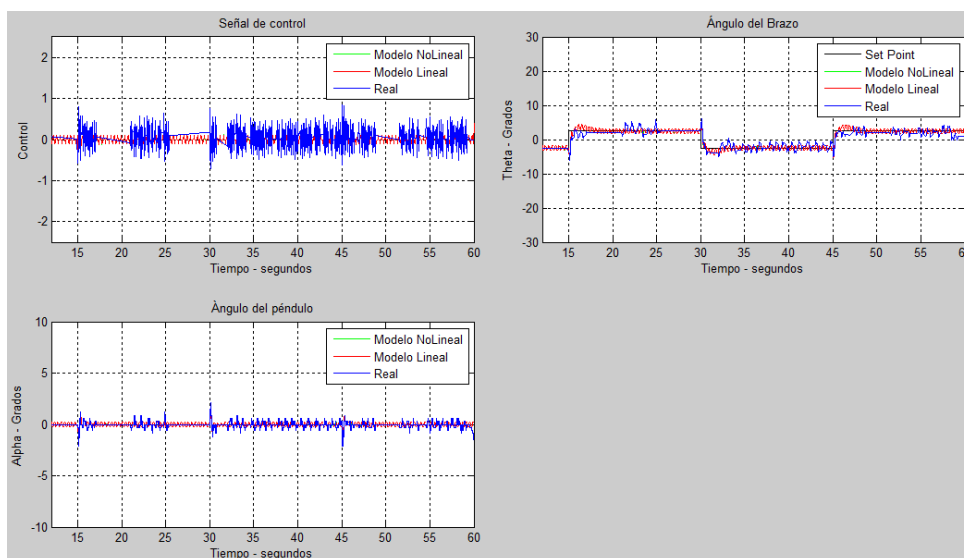


Figura 7.10 Comparación entre simulación e implementación ante una señal cuadrada de 2.5° de amplitud

En la figura 7.10 se observa que el movimiento angular del brazo sigue la referencia deseada, con un tiempo de asentamiento corto y valores de sobrepaso adecuados para permitir que el péndulo no pierda el equilibrio. En la implementación, se debe considerar la existencia de pequeños picos en la señal de control debido a perturbaciones externas como el viento.

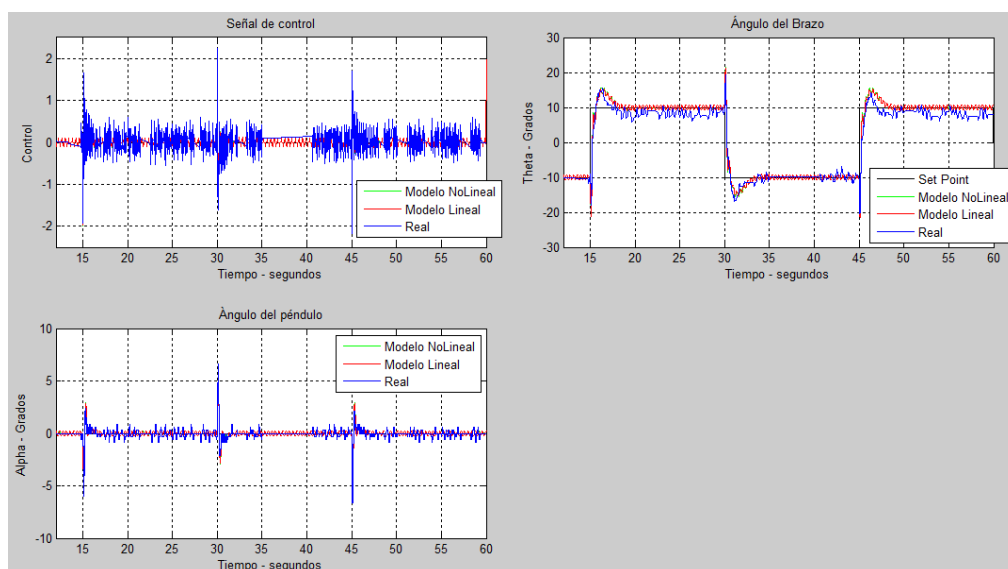


Figura 7.11 Comparación entre simulación e implementación ante una señal cuadrada de 10° de amplitud

Para verificar si el controlador trabaja de forma correcta, se procede a aumentar gradualmente la amplitud de la señal de referencia y así notar la robustez del controlador ante cambios bruscos en el valor deseado.

Al incrementar la amplitud del valor angular deseado en el brazo, como se muestra en la figura 7.11, se puede notar que el sistema aún mantiene el péndulo equilibrado. El valor pico que se genera en la respuesta angular del brazo no es suficientemente elevado para desestabilizar el péndulo y hacerlo caer.

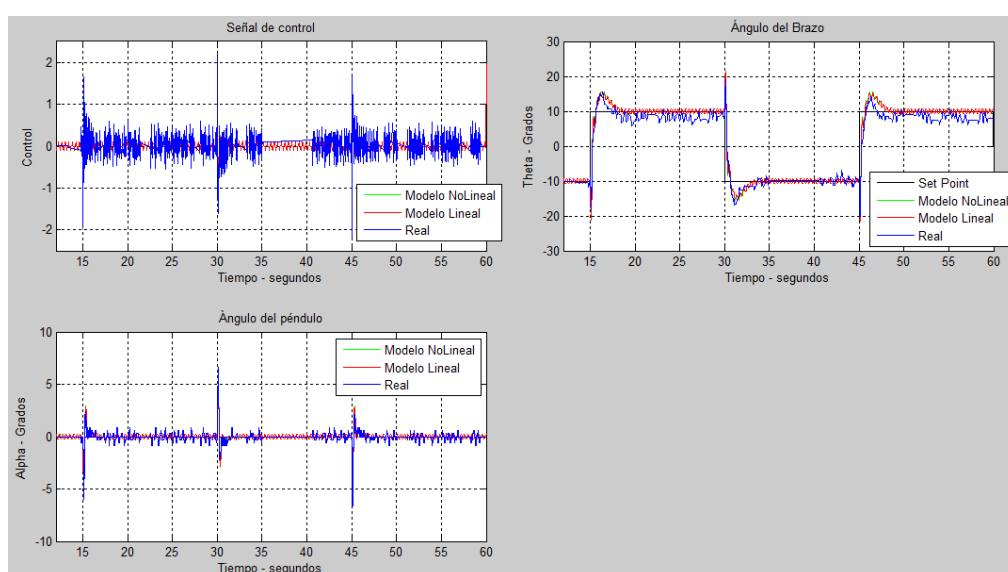


Figura 7.11 Comparación entre simulación e implementación ante una señal cuadrada de 10° de amplitud

En la figura 7.12 se visualiza la respuesta del sistema PIR ante cambios de mayor amplitud en el valor referencial del brazo; la regulación de posición angular se conserva y el péndulo aún se mantiene en equilibrio, logrando aún el objetivo propuesto.

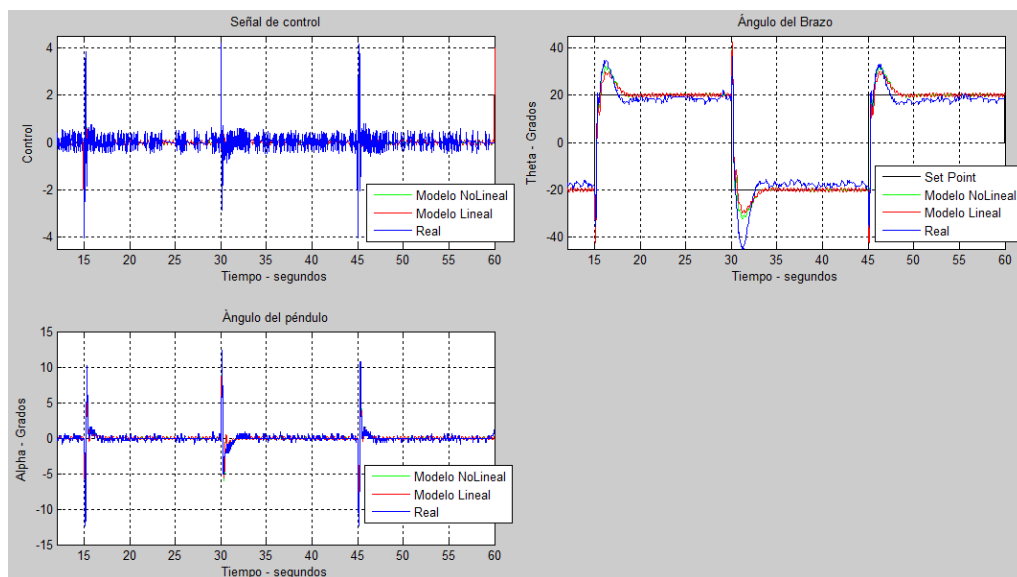


Figura 7.12 Comparación entre simulación e implementación ante una señal cuadrada de 20° de amplitud

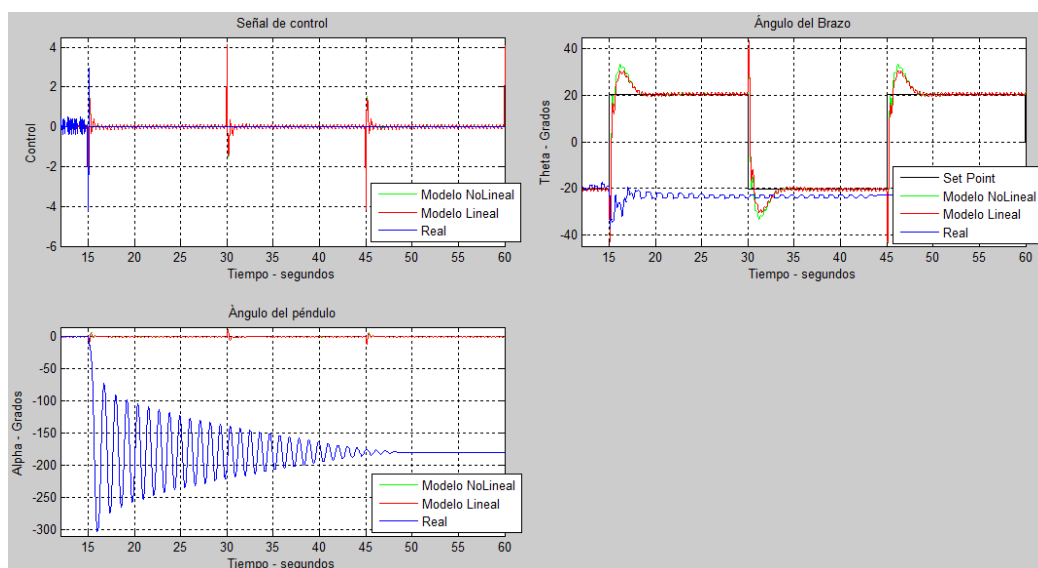


Figura 7.13 Caída del péndulo ante cambios de referencia excesivos

De forma experimental se puede comprobar que al aplicar una señal de referencial cuadrada con amplitud mayor a 20° , se genera una

señal de control de gran amplitud, un movimiento brusco en el brazo y por ende el péndulo pierde estabilidad, figura 7.13.

La tabla 10 muestra las características de tiempo de asentamiento y porcentaje de sobrepaso para diferentes referencias del ángulo del brazo (Amplitud de señal), se puede ver una semejanza durante la fase de estabilización del sistema.

Sin embargo la señal de control presenta un incremento en su valor pico tan pronto como exista un incremento en la amplitud de la referencia.

Tabla10: Respuesta del sistema a diferentes referencias del ángulo del brazo aplicando control PI lineal

Amplitud de señal [°]	θ [°]		u
	Tiempo de asentamiento [seg.]	Sobrepaso [%]	Valor pico [Volts.]
2.5	2.65 seg.	26	0.6
5	2.7 seg.	24	1.09
10	3.25 seg.	22.5	2.1
15	3.3 seg.	30	3.2 (sat.)
20	3.3 seg.	27.5	4.1 (sat.)

A través de dicha tabla se puede apreciar que la señal de control presenta una saturación al momento que la referencia tiene una amplitud mayor a 10°.

7.1.3 Evaluación del controlador PI no lineal para el brazo

Al igual que en el controlador lineal, se procede a verificar la efectividad del controlador no lineal aplicando una señal cuadrada con una frecuencia de 0.033Hz y amplitud variable. Para este caso se compara el modelo simulado con la respuesta real en la planta tal como se muestra en las gráficas 7.16 y 7.17.

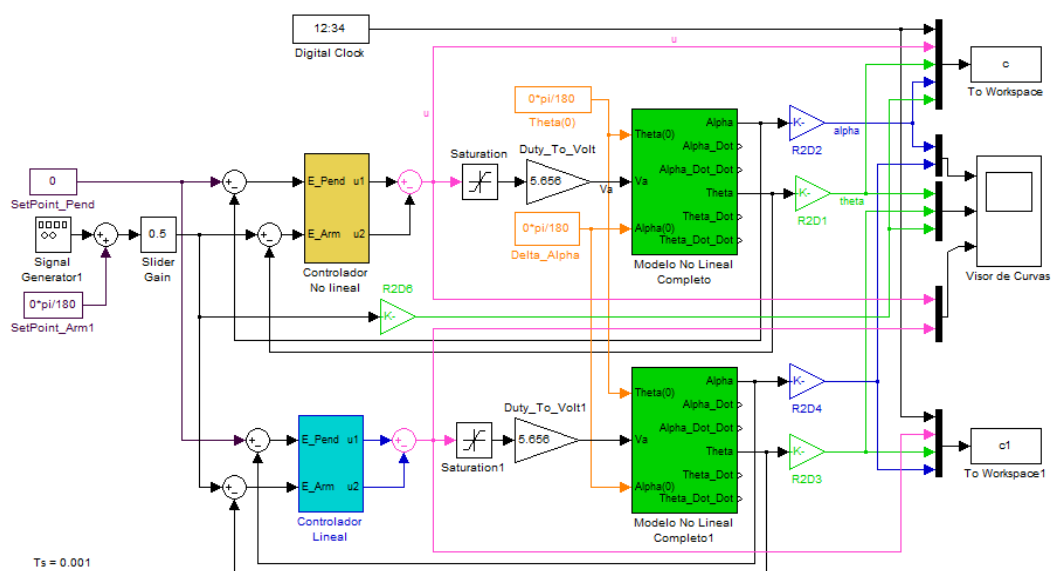


Figura 7.14 Simulación de Control PI no Lineal en simulink

A continuación se muestran las gráficas obtenidas de la experimentación del control no lineal en la planta.

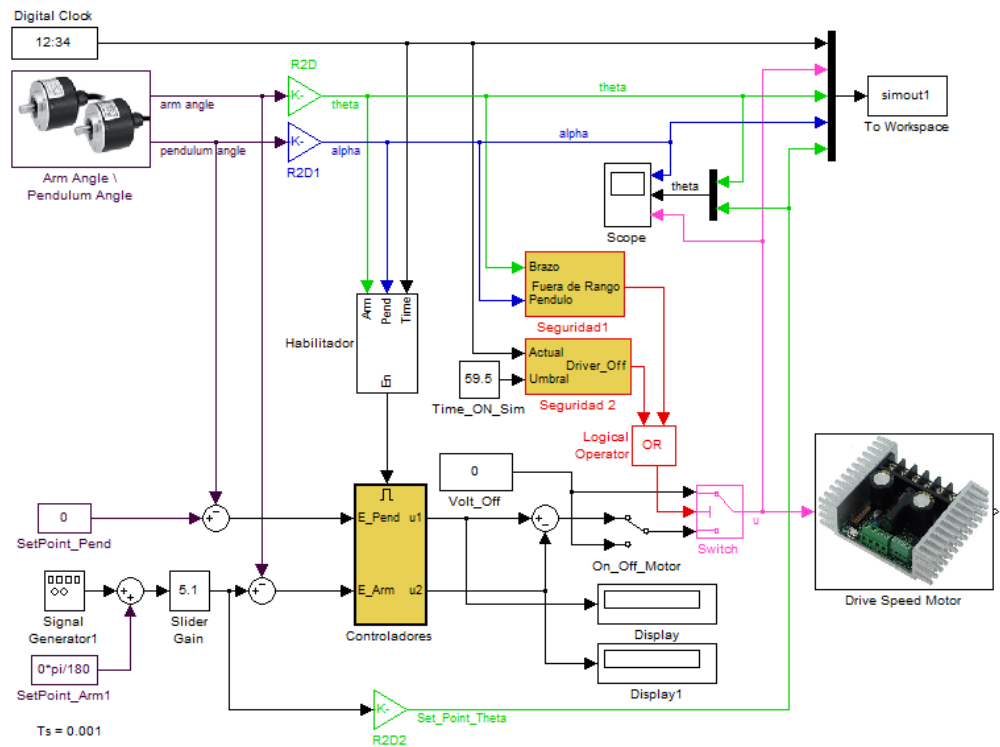


Figura 7.15 Interface para Control PI no Lineal en simulink

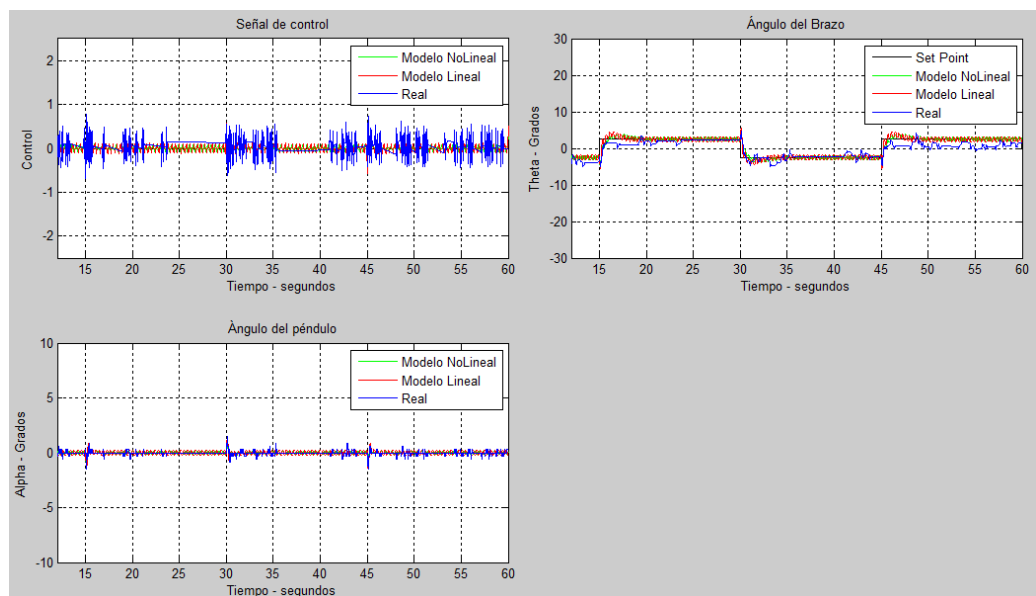


Figura 7.16 Comparación entre simulación e implementación ante una señal cuadrada de 2.5° de amplitud

Al aplicar una señal de pequeña amplitud, como se muestra en la figura 7.16, el control no lineal permite regular la posición del brazo en el valor de referencia deseado y a su vez mantiene equilibrado el péndulo de forma vertical.

Se puede notar que en los cambios de referencia angular del brazo, la señal de control genera reacciona de forma suave.

Al igual que en las pruebas del controlador PI lineal del brazo, se incrementa de forma gradual la amplitud de la señal de referencia para conocer su robustez ante cambios bruscos en la consigna deseada.

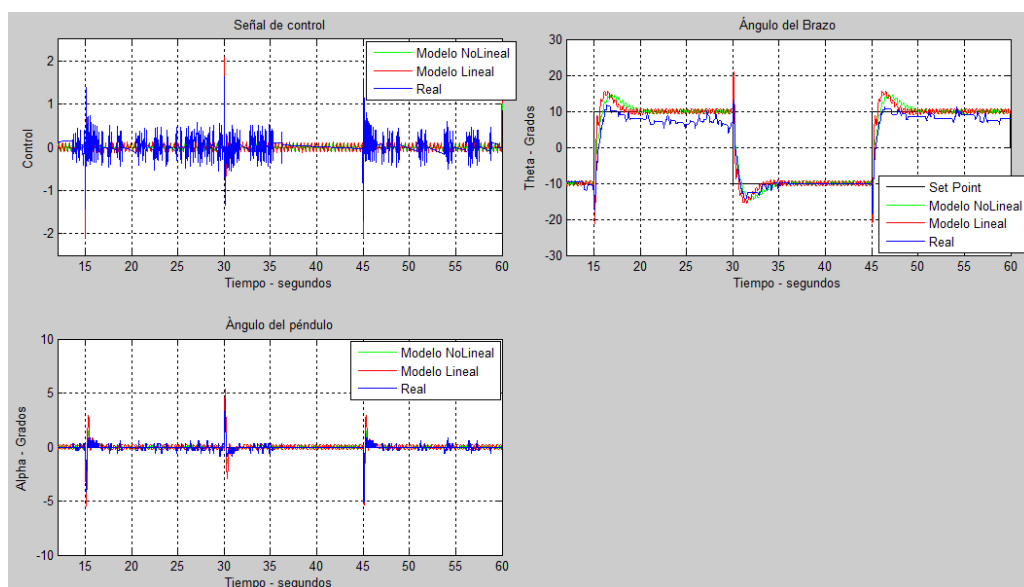


Figura 7.17 Comparación entre simulación e implementación ante una señal cuadrada de 10° de amplitud

Para ello se aplica una señal cuadrada con una amplitud de 10° , como se muestra en la figura 7.17. La posición angular del brazo sigue regulándose, con un tiempo de estabilización y valor pico adecuado para no hacer caer al péndulo.

La señal de control presenta picos en los momentos de transición del valor deseado, pero en comparación a la señal del controlador lineal, tiene una amplitud menor.

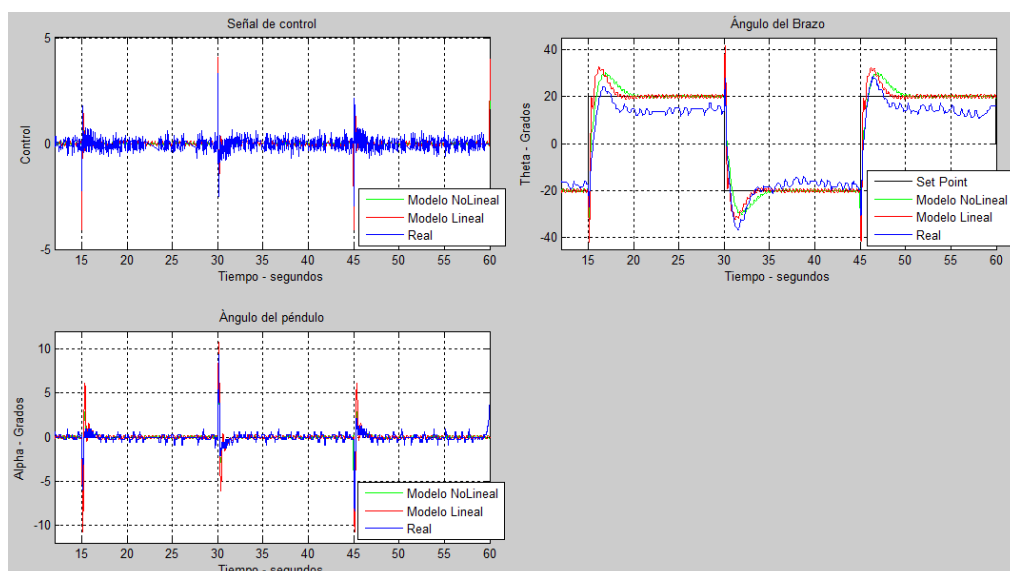


Figura 7.18 Comparación entre simulación e implementación ante una señal cuadrada de 20° de amplitud

Al incrementar la señal de referencia a 20° de amplitud, se puede notar que la salida del controlador presenta picos menores a los del controlador lineal en los momentos transitorios. Además se puede apreciar que péndulo tiene un mayor movimiento lo que podría provocar que pierda el equilibrio.

Para comprobar la capacidad límite del controlador no lineal en para regular la posición del brazo y mantener en equilibrio del péndulo, se incrementa nuevamente la amplitud de la señal cuadrada de referencia llegando a un valor de 27.5° , figura 7.19.

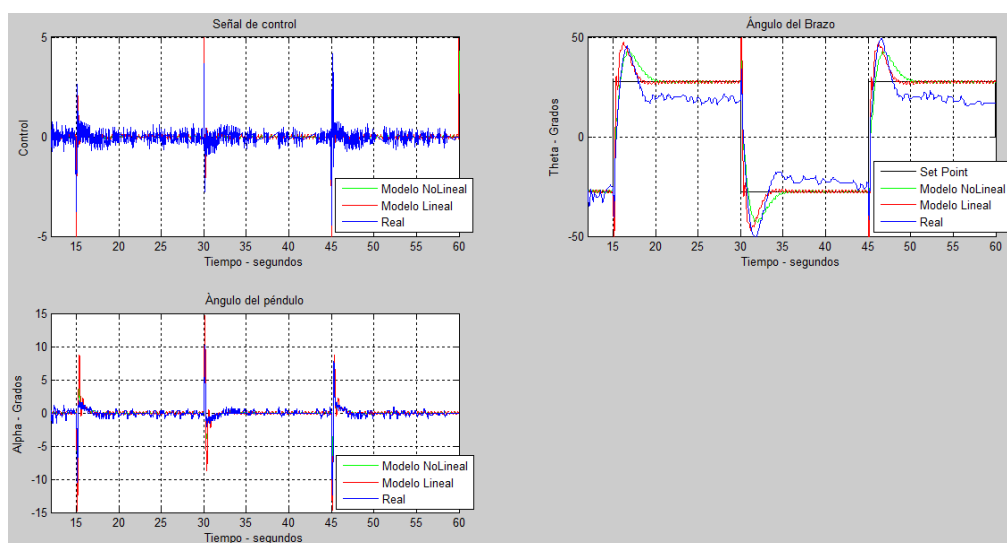


Figura 7.19 Comparación entre simulación e implementación ante una señal cuadrada de 27.5° de amplitud

Esto quiere decir que el método de linealización extendida para el diseño del controlador no lineal, ha permitido mantener equilibrado el péndulo considerando un rango más amplio de acción para la posición del brazo.

A manera de resumen, se recopilan las características de las pruebas realizadas, lo cual se muestra en la tabla 11.

Tabla11: Respuesta del sistema a diferentes referencias de amplitud en el brazo aplicando control PI no lineal

Amplitud de señal [°]	θ [°]		u
	Tiempo de asentamiento [seg.]	Sobrepaso [%]	Valor pico [Volts.]
2.5	2.6 seg.	22	0.7
5	2.7 seg.	25	0.85
10	2.7 seg.	25	1.5
15	2.8 seg.	17	2.2
20	3.1 seg.	20	2.65
25	4.8 seg.	44	2.7
27.5	3.4 seg.	37	2.72

Se puede notar que controlador no lineal del brazo cumple su trabajo.

Considerando que mientras más alejada esté la referencia del punto de operación, se puede notar un error de estado estacionario.

7.2 Análisis y comparación de resultados

A través de simulación se procede a verificar los límites de cada controlador para comprobar la hipótesis planteada.

Como se puede apreciar en la figura 7.20, en caso de tener un incremento notable en la amplitud de la referencia, el regulador PI lineal genera una señal de control muy grande que causa un movimiento brusco en el brazo y por ende la caída del péndulo. En el caso del regulador PI no lineal se mantiene operando con un tiempo de respuesta similar a los casos previos.

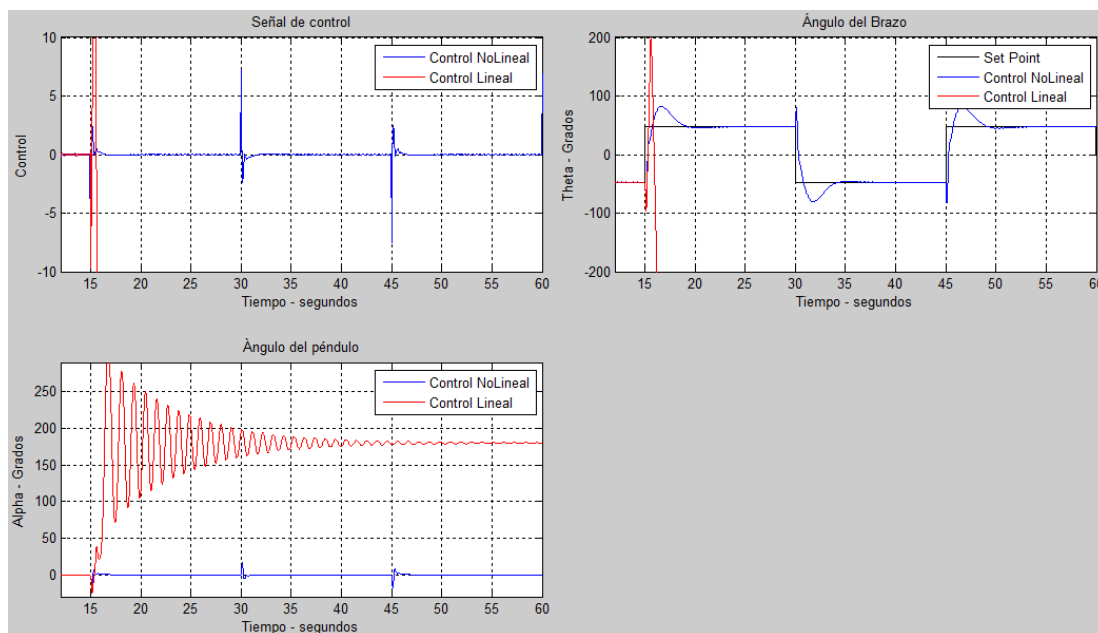


Figura 7.20 Comparación entre controladores lineal y no lineal ante una señal cuadrada de 47.5° de amplitud

A continuación se procede a seguir incrementando la amplitud de la señal de referencia para ver el límite del controlador no lineal como se muestra en la figura 7.21.

Al momento de incrementar la señal de control hasta 71° aproximadamente se produce la caída del péndulo aplicando el controlador PI no lineal a causa de un valor excesivo de la señal de control.

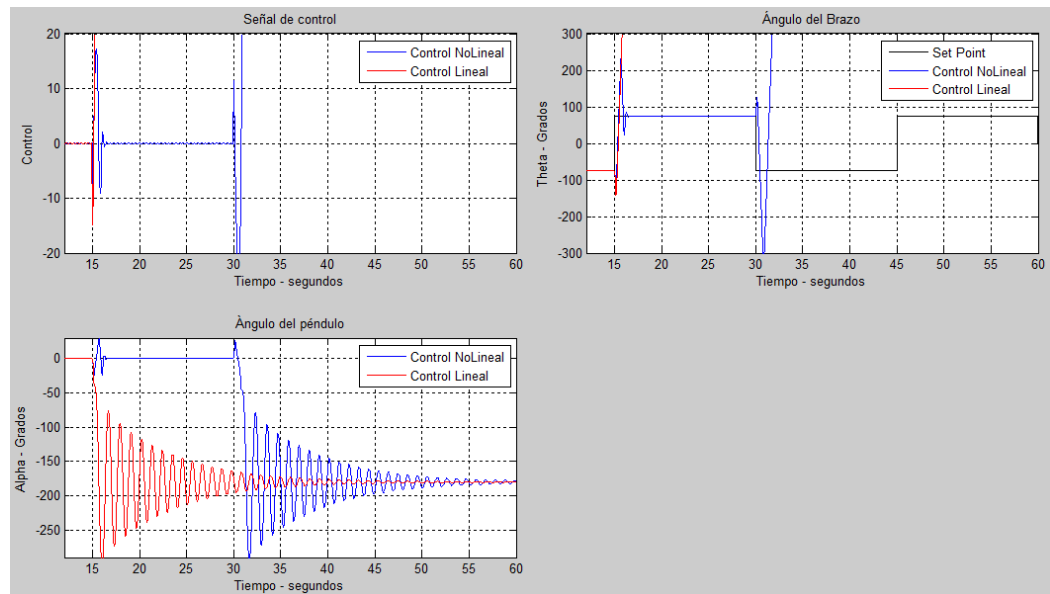


Figura 7.26 Comparación entre controladores lineal y no lineal ante una señal cuadrada de 71° de amplitud

Para cuantificar la mejora obtenida por el controlador no lineal se calcula, en porcentaje, la diferencia de los valores límites de la señal de referencia previos a que el péndulo pierda estabilidad.

$$\frac{(71 - 47.5)}{47.5} \times 100\% = 49\%$$

El rango de referencia de amplitud del controlador no lineal es un 49% mayor respecto al rango de referencia del controlador lineal.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El controlador PID lineal implementado es suficiente para estabilizar el péndulo en la posición vertical ante perturbaciones pequeñas sin importar la posición angular del brazo, lo que generaría colisiones con los extremos de estructura.
2. El controlador PI lineal implementado para mantener la posición del brazo logra seguir la referencia indicada dentro de un rango específico de posición angular evitando el choque con los extremos, sin embargo genera picos que se incrementan cuando nos alejamos del punto de operación.

3. El controlador PI no lineal desarrollado para regular la posición del brazo consigue mantener la referencia de posición angular en un rango de operación mayor que el regulador PI lineal, lográndose reducir los picos, su consumo de energía fue menor y evitando los choques en los extremos ante perturbaciones.
4. En relación al diseño de los controladores, se utilizaron diversas técnicas y herramientas de software que permitieron implementar los mismos.
5. Se pudo notar que el controlador no lineal exige una capacidad computacional mayor debido a la cantidad de cálculos y manejo de variables tipo real (float).

RECOMENDACIONES Y TRABAJOS FUTUROS

1. Para implementar sistemas en tiempo real utilizando tarjetas tipo PCI, se debe cumplir con las exigencias de software requeridas por el fabricante para su correcto funcionamiento.
2. El tiempo de muestreo seleccionado debe ser acorde al tipo de sistema a controlar, dado que el regulador debe ser más rápido que la respuesta de la planta, para cumplir la consigna y mantener estable el sistema.
3. La instrumentación utilizada para recopilar datos al momento de obtener los parámetros constructivos del modelo matemático, deben de manejar la mayor precisión posible.

4. Al momento de utilizar controladores no lineales se recomienda verificar que existan pocas inexactitudes mecánicas en la planta pues esto hace que los resultados simulados difieran con los datos de la planta.
5. Antes de la puesta en marcha de los controladores se sugiere diseñar las seguridades del caso, pues por tratarse de un sistema mecánico con gran volumen de trabajo se necesita evitar colisiones.
6. Como trabajo futuro se pretende implementar controladores difusos y controladores en espacio de estados para fortalecer la cátedra de las asignaturas relacionadas a la Teoría de Control.

BIBLIOGRAFÍA

- [1] Oltean, S. E., Swing-up and stabilization of the rotational inverted pendulum usign PD and fuzzy-PD controllers, *Procedia Technology*, vol. 12, 2013.
- [2] M. P. Viroch Sukontanakarn, Real-Time Optimal Control for Rotary Inverted Pendulum, *Science Publications*, 2009.
- [3] Furuta, K., Nishihara,H. y Mori, S., Control of Unstable Mechanical Systems: Control of Pendulum, *International Journal of Control*, vol. 23, pp. 673-692, 1976.
- [4] Olfati-Saber, R., Normal Forms for Underactuated Mechanical Systems with Symmetry, *IEEE Transactions on Automatic Control*, vol. 47, nº 2, 2002.
- [5] Sanchez, J. A., Control Avanzado de Procesos, Madrid: Díaz de Santos, 2003.
- [6] Shetty, D. y Kolk, R., Mechatonics System Design, 2 ed., Cengage Learning, 2010.
- [7] Ogata, K., Ingeniería de Control Moderna, 5 ed., Madrid: Pearson Educación, 2010, p. 904.
- [8] National Instruments, <http://www.ni.com/data-acquisition/esa/>, fecha de consulta 5 Mayo 2015.
- [9] Mathworks Inc., <http://www.mathworks.com/products/daq/>, fecha de consulta 5 Mayo 2015.
- [10] Garcini, H., Sistemas de Control en Tiempo Continuo, Maestría en Automatización y Control Industrial, Cuba, 2011.

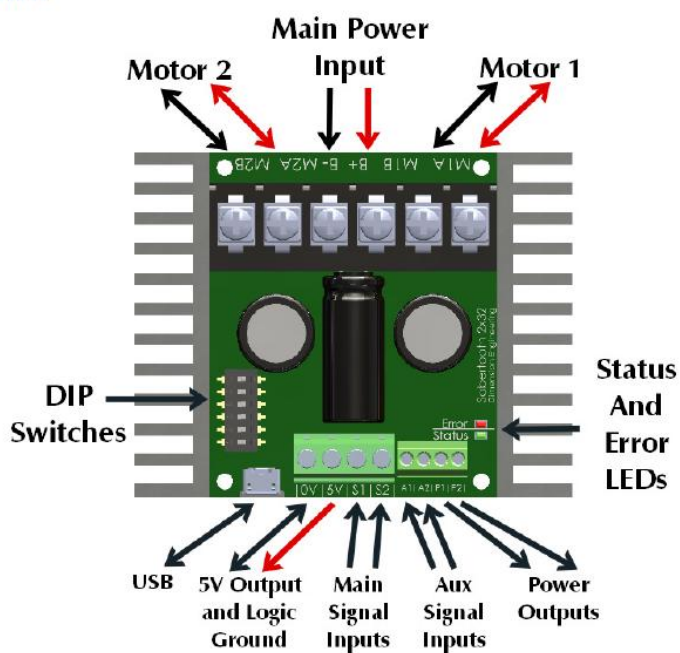
- [11] Ogata, K., *Sistemas de Control en Tiempo Discreto*, 2 ed., Prentice Hall, 2000.
- [12] Ramírez, H. S., *Control de sistemas no lineales: linealización aproximada, extendida, exacta*, Madrid: Pearson Educación, 2005, pp. 32, 336.
- [13] The MathWorks, *Real-Time Windows Target User's Guide*.
- [14] Dorf, R. C. y Bishop, R. H., *Modern Control Systems*, 12 ed., New Jersey: Prentice Hall, 2011.
- [15] Gray, C. B., *Máquinas Eléctricas y Sistemas Accionadores*, México: Alfaomega, 1997, p. 231.
- [16] Raymond, S. y Robert, B., *Física para ciencias e ingenierías*, 5 ed., vol. 1, Mexico: McGrawHill, 2002, p. 304.
- [17] Arduino, <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>, fecha de consulta 2 de Agosto 2015.
- [18] Da Silva, A. y Craig, K., *Optical Encoder and the Arduino*, 2012, p. 16.
- [19] Creus Solé, A., *Instrumentación industrial*, México: Marcombo, 2012.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
USB Host Chip	MAX3421E
Length	101.52 mm
Width	53.3 mm
Weight	36 g

ANEXO 1. Conexiones y características de la placa Arduino Mega

Overview



Main Power Input: Connect to a 6V-33.6V Battery or Power Supply.

Motor 1 and Motor 2: Connect Motor 1 to the M1A and M1B. Connect Motor 2 to M2A and M2B.

DIP Switches: These are used to set the operating mode and options. Can be changed while operating.

USB: A standard Micro USB port. Connect to a PC or other USB host to control, monitor or modify.

Logic Ground: The 0V logic ground is connected internally to B-.

5V Output: 5V is a regulated 5 volt output. You can use it to power additional circuitry up to 1 amp.

Main Signal Inputs: Connect your main analog, R/C or serial signals here.

Aux Signal Inputs: These may be used for additional control. Optional in most modes.

Power Outputs: These connect to voltage clamp resistors, electromagnetic brakes, field windings, or other moderate power loads. 8A max current per channel.

Status and Error LEDs: These glow and flash to indicate the status of the Sabertooth 2x32.

Specifications

	<i>Mechanical specifications</i>		
Dimensions	2.75 x 3.5 x 1.0 inches (70mm x 90mm x 26mm)		
Weight	4.5 ounces (125 grams)		
	Minimum	Typical	Maximum
Wire size, battery	16 gauge	10 gauge	10 gauge
Wire size, motors	16 gauge	12 gauge	10 gauge
Wire size, signal	28 gauge	24 gauge	18 gauge
Operating temperature	0F (-20C)	70F (25C)	160F (70C) ¹
	<i>Electrical Characteristics</i>		
	Minimum	Typical	Maximum
Input voltage, B+ and B-	6.0 Volts	12 or 24 Volts	33.6 Volts
Continuous output current, M1 and M2	-	-	32 amps ¹
Peak output current, M1 and M2	-	-	64 amps ²
Output voltage, M1 and M2	-95% of input voltage (average)	-	95% of input voltage (average)
Voltage, P1 and P2	0V	-	Input voltage +.3V
Output current, P1 and P2			8 amps, sink only
Output voltage, 5V	4.85	5.0	5.15
Output Current, 5V	-	-	1A
Input voltage, S1 and S2	-.3V	0V to 5V	12V ³
Input voltage, A1 and A2	-.3V	0V to 5V	12V ³
Output Voltage, S2 and A2	0V		3.5V

¹ Maximum continuous output current derates linearly above 40C ambient. At 70C ambient the maximum continuous output current is 10 amps per channel.

ANEXO 2. Conexiones y características de la placa Sabertooth 2x32

PCI-1711/1711L

100 KS/s, 12-bit, 16-ch
S.E. Inputs Low-cost
Multifunction Card



PCI-1711/1711L

PCI-1710



Features

- 16 single-ended analog inputs
- 12-bit A/D converter, with up to 100 kHz sampling rate
- Programmable gain for each input channel
- Automatic channel/gain scanning
- On-board 1K samples FIFO buffer
- Two 12-bit analog output channels (Only for PCI-1711)
- 16 digital inputs and 16 digital outputs
- Programmable timer/counter

Introduction

The PCI-1711/1711L is a powerful but low-cost multifunction card for the PCI bus. The PCI-1711 comes with 2 analog output channels, while the PCI-1711L doesn't. Thus, you can choose between PCI-1711 and PCI-1711L according to what you really need for your application, with no extra costs.

Specifications

Analog Input

- Channels 16 Single-Ended
- Resolution 12-bit
- FIFO Size 1K samples
- Sampling Rate* 100 KS/s max.
- Conversion Time 10 μ s

Input range and Gain List	Gain	1	2	4	8	16
	Input	± 10 V	± 5 V	± 2.5 V	± 1.25 V	± 0.625 V
Drift (ppm/°C)	Gain	1	2	4	8	16
	Zero	15	15	15	15	15
Small Signal Bandwidth for PGA	Gain	25	25	25	30	40
	Gain	1	2	4	8	16
Bandwidth	Bandwidth	4.0 MHz	2.0 MHz	1.5 MHz	0.65 MHz	0.35 MHz

- Max. Input Overvoltage 10 μ s
- Input Protect 30 Vp-p
- Input Impedance 2 M Ω /5 pF
- Trigger Mode Software, On-board Programmable Pacer or external

Accuracy	DC	INL: ± 0.5 LSB
		AC
		Offset error: Adjustable to zero
		Gain error: 0.005% FSR (Gain=1)
		SNR: 68 dB
		ENOB: 11 bits

Programmable Counter / Timer

- Channels 1
- Resolution 16-bit
- Compatibility TTL level
- Base Clock 10 MHz
- Max. Input Frequency 10 MHz

Note:

The sampling rate and throughput depends on the computer hardware architecture and software environment. The rates may vary due to programming language, code efficiency, CPU utilization and so on.

Analog Output (only for PCI-1711)

- Channels 2
- Resolution 12-bit

Output Range (Internal & External Reference)	Internal Reference	0 - +5 V, 0 - +10 V
	External Reference	0 - +x V @ -x V (-10 \leq x \leq 10)
Accuracy	Relative	$\pm 1/2$ LSB
	Differential Non-linearity	$\pm 1/2$ LSB

- Gain Error Adjustable to zero
- Slew Rate 11 V/ μ s
- Drift 40 ppm/°C
- Driving Capability 3 mA
- Throughput 38 KS/s (min.)
- Output Impedance 0.81 Ω
- Settling Time 26 μ s (to $\pm 1/2$ LSB of FSR)
- Reference Voltage Internal -5 or -10 V
External -10 or +10 V

Digital Input / Output

Input Channels	16	
Input Voltage	Low	0.6 V max.
	High	2.0 V max.
Output Channels	16	
Output Voltage	Low	0.8 V max. @ 0.8 mA (sink)
	High	2.0 V min. @ -0.4 mA (source)

General

IO Connector Type	68-pin SCSI-II female	
Dimensions	175 mm x 100 mm (6.9" x 3.9")	
Power Consumption	typical	PCI-1711 +5 V @ 850 mA PCI-1711L +5 V @ 700 mA
	Max.	+5 V @ 1.0 A
Temperature	Operation	0 - +60° C (32 - 140° F) (refer to IEC 68-2-1, 2)
	Storage	-20 - +70° C (-4 - 158° F)
Relative Humidity	5% - 95% RH non-condensing (refer to IEC 68-2-3)	

Ordering Information

- **PCI-1711** 100 KS/s, 12-bit, 16-ch S.E. Inputs Low-cost Multifunction Card, user's manual and driver CD-ROM. (cable not included)
- **PCI-1711L** 100 KS/s, 12-bit, 16-ch S.E. Inputs Low-cost Multifunction Card w/o analog output, user's manual and driver CD-ROM. (cable not included)
- **PCLD-8710** Industrial Wiring Terminal Board with CJC circuit for DIN-rail mounting (cable not included)
- **PCL-10168** 68-pin SCSI-II cable with male connectors on both ends and special shielding for noise reduction, 1 and 2 m
- **ADAM-3968** 68-pin SCSI-II Wiring Terminal Board for DIN-rail Mounting

Pin Assignments

A0	68	34	A11
A2	67	33	A3
A4	66	32	A5
A6	65	31	A7
A8	64	30	A9
A10	63	29	A11
A12	62	28	A13
A14	61	27	A15
AGND	60	26	AGND
ADD_REF	59	25	AOL_REF
ADD_OUT	58	24	AOL_OUT
AOGND	57	23	AOGND
D0	56	22	D11
D2	56	21	D3
D4	54	20	D5
D6	53	19	D7
D8	52	18	D9
D10	51	17	D11
D12	50	16	D13
D14	48	15	D15
DGND	48	14	DGND
DO0	47	13	DO1
DO2	46	12	DO3
DO4	45	11	DO5
DO6	44	10	DO7
DO8	43	9	DO9
DO10	42	8	DO11
DO12	41	7	DO13
DO14	40	6	DO15
DGND	39	5	DGND
CNT0_CLK	38	4	PACER_OUT
CNT0_OUT	37	3	TRG_GATE
CNT0_GATE	36	2	EXT_TRG
+12V	35	1	+5V

*: Pins 23-25 and pins 57-59 are not defined for PCI-1711L

Feature Details

Plug-and-play Function

The PCI-1711/1711L fully complies with the PCI Specification Rev 2.1, and thus is a Plug-and-Play device. During card installation, it is virtually unnecessary to set any jumpers or DIP switches. Instead, all bus-related configurations such as base I/O address and interrupts are conveniently taken care of by the Plug-and-Play function.

Flexible Input Types and Range Settings

The PCI-1711/1711L features an automatic channel/gain scanning circuit. This circuit design controls multiplexer switching during sampling. You can set different gain values for each channel according to your needs for the corresponding range of input voltages. The gain values thus selected are stored in the SRAM. This flexible design enables multi-channel and high-speed sampling for high-performance data acquisition (up to 100 KS/s).

Onboard FIFO Memory

The PCI-1711/1711L provides an onboard FIFO (First In First Out) memory buffer, storing up to 1 K A/D samplings. You can either enable or disable the interrupt request feature of the FIFO buffer. While the interrupt request for FIFO is enabled, you can further specify whether the interrupt request will be sent whenever one sampling takes place or when the FIFO buffer is half saturated. This feature enables a continuous high-speed data transfer with more predictable performance on Windows systems.

Onboard Programmable Counter

The PCI-1711/1711L is equipped with a programmable counter, which can serve as a pacer trigger for A/D conversions. The counter chip is an 82C54 or equivalent, which incorporates three 16-bit counters on a 10 MHz clock. One of the three counters is used as an event counter for input channels. The other two are cascaded into a 32-bit timer for pacer triggering.

Applications

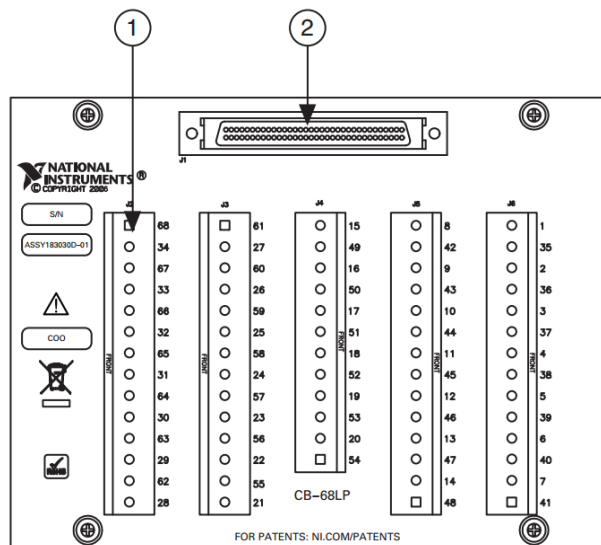
- Process monitoring and control
- Transducer and sensor measurement
- Multi-channel DC voltage measurement

USER GUIDE

NI CB-68LP/CB-68LPR

Unshielded 68-Pin Screw Terminal Connector Block Accessories

The National Instruments CB-68LP and CB-68LPR are unshielded I/O accessories with 68 screw terminals for easy signal connection to a National Instruments 68-pin DAQ device.



1 Screw Terminals 2 68-Pin I/O Connector

Physical Characteristics

Dimensions

CB-68LP 143.5 mm × 107.4 mm × 30.0 mm
(5.65 in. × 4.23 in. × 1.18 in.)

CB-68LPR 167.6 mm × 76.2 mm × 29.2 mm
(6.60 in. × 3.00 in. × 1.15 in.)

Weight

CB-68LP 155 g (5.4 oz)

CB-68LPR 162 g (5.7 oz)

I/O connector One 68-pin male SCSI connector

Screw terminals 68, all I/O signals are available at screw terminals

Wire gauge 0.0509 to 3.31 mm² (12 to 30 AWG)

Torque 0.5 N · m (4.4 lb · in.)

Clean the hardware with a soft, nonmetallic brush.

Autonics

ROTARY ENCODER(INCREMENTAL TYPE)

E50S/E50SP SERIES

M A N U A L



E50S



E50SP



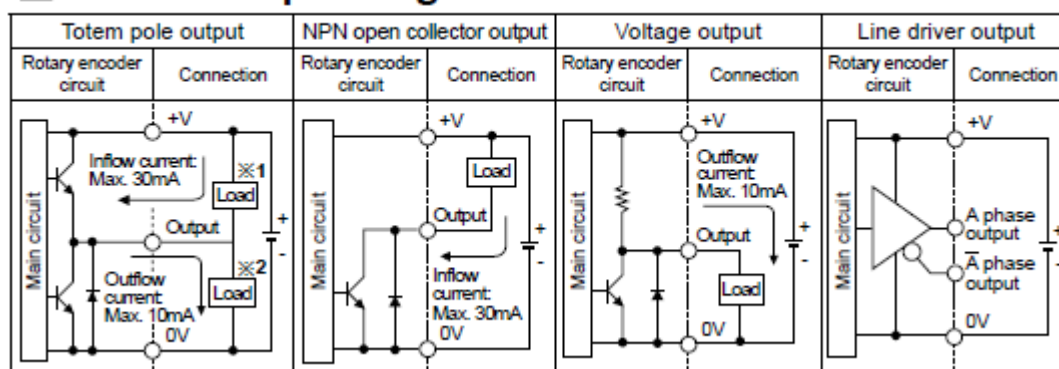
Ordering information

Series	Shaft diameter	Pulse/1Revolution	Output phase	Output	Power supply	Cable
E50S	8	8000	3	N	24	
Diameter ø50mm, shaft type	ø8mm	See resolution	2: A, B 3: A, B, Z 4: A, \bar{A} , B, \bar{B} 6: A, \bar{A} , B, \bar{B} , Z, \bar{Z}	T: Totem pole output N: NPN open collector output V: Voltage output L: Line driver output	5: 5VDC \pm 5% 24: 12-24VDC \pm 5%	No mark: Normal type C: Cable outgoing connector type(※) CR: Rear connector integrated type CS: Side connector integrated type

※Standard: E50S8-[PULSE]-3-N-24

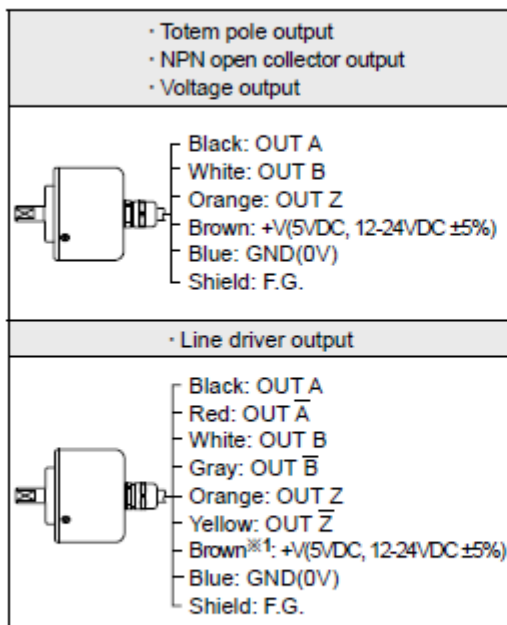
※Cable length: 250mm

Control output diagram



■ Connections

• Normal type



- ※1: E50SP series is only for 5VDC ± 5%.
- ※ Unused wires must be insulated.
- ※ The shield cable and metal case of encoder must be grounded(F.G.).

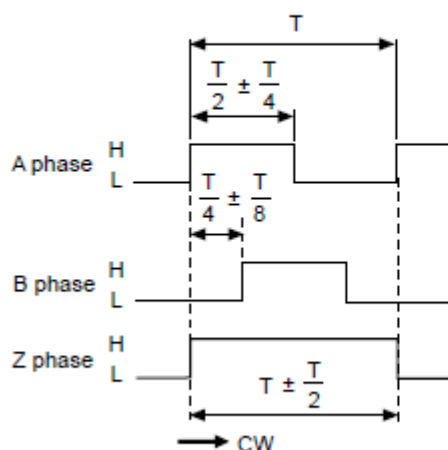
• Cable outgoing connector type/
• Connector integrated type

• Totem pole output • NPN open collector output • Voltage output			• Line driver output		
Pin No.	Cable color	Function	Pin No.	Cable color	Function
1	Black	OUT A	1	Black	OUT A
2	White	OUT B	2	Red	OUT \bar{A}
3	Orange	OUT Z	3	Brown	+V
4	Brown	+V	4	Blue	GND
5	Blue	GND	5	White	OUT B
6	Shield	F.G.	6	Gray	OUT \bar{B}
			7	Orange	OUT Z
			8	Yellow	OUT \bar{Z}
			9	Shield	F.G.

- ※ It must be isolation grounded.

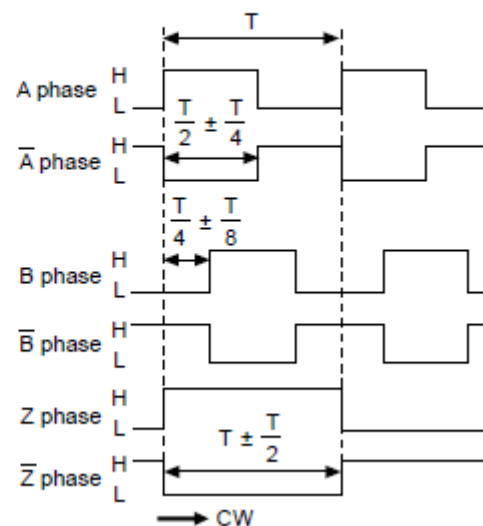
■ Output waveform

• Totem pole output / NPN open collector output /
• Voltage output



- ※ Clockwise(CW): Right turn as from the shaft

• Line driver output



```

%Inicialización
clc; close all; clear all;
%% Variables simbólicas
syms abcdefhis
%% *****
A = [0 1 0 0; a b 0 c; 0 0 0 1; d e 0 f]
B = [0; h; 0; i]
C = [1 0 0 0; 0 0 1 0]
I = eye(4);
G = C*(inv(s*I-A))*B
G = collect(G,s)
pretty(G)
%Parámetros y constantes del sistema *****
Ra=5.4351;      %Resistencia - Ohms
La=0.00188;    %Inductancia - Henrios
kb=0.91098;    %Constante Eléctrica - V*s/rad
kt=1.1098;     %Constante Torque - N*m/A
Jm=0.00071;    %Inercia del motor sin carga - kg*(m^2)
Br=0.04166;%0.04166; %Constante de fricción viscosa - Nms/rad
g=9.81;        %Aceleración de la gravedad - m/(s^2)
mp=0.13417;    %Masa del péndulo - kg
Jr=0.0085-Jm;  %0.00429; %Momento de inercia del brazo -
r=0.125;       %Longitud/2 del brazo - m
L=0.19604;     %Longitud desde el eje de giro hasta el CM del
péndulo - m
Jp=0.001882;   %Momento de inercia del péndulo -
%Bp=0.00156798; %Constante de fricción viscosa - Nms/rad
(simulación)
Bp=0.000456798;
k1=Jp+mp*(L^2);
k2=mp*r*L;
k3=mp*g*L;
k4=mp*r*L;
k5=Jr+Jm+mp*(r^2);
k6=mp*r*L;
%Punto de operación
x1=0;%0*pi/180;
x2=0; x4=0; x3=0;
u=- (Ra*k3*k5*tan(x1))/k2*kt;
%cambio de variables
a = (- Ra*k2*k6*(2*cos(x1)^2 - 1)*x2^2 + Ra*k3*k5*cos(x1) -
k2*kt*u*sin(x1) + k2*kb*kt*x4*sin(x1) +
Br*Ra*k2*x4*sin(x1))/(Ra*(k1*k5 - k2*k4*cos(x1)^2)) +
(2*k2*k4*cos(x1)*sin(x1)*(k5*(Bp*x2 - k3*sin(x1)) -
(k2*kt*u*cos(x1))/Ra + (k2*x4*cos(x1)*(kb*kt + Br*Ra))/Ra +
k2*k6*x2^2*cos(x1)*sin(x1)))/(k1*k5 - k2*k4*cos(x1)^2)^2;
b = -(Bp*k5 + k2*k6*x2*sin(2*x1))/(k1*k5 - k2*k4*cos(x1)^2);
c = -(k2*cos(x1)*(kb*kt + Br*Ra))/(Ra*(k1*k5 - k2*k4*cos(x1)^2));
d = (Bp*Ra*k2*k4^2*x2*(sin(x1) - sin(x1)^3) -
Ra*k2*k3*k4^2*cos(x1)^2 - Ra*k1^2*k5*k6*x2^2*cos(x1) -
Ra*k1*k3*k4*k5 + 2*Ra*k1*k3*k4*k5*cos(x1)^2 -
k1*k2*k4*kt*u*sin(2*x1) + 2*Ra*k1*k2*k4*k6*x2^2*cos(x1) +
k1*k2*k4*kb*kt*x4*sin(2*x1) - Ra*k1*k2*k4*k6*x2^2*cos(x1)^3 +

```

```

Bp*Ra*k1*k4*k5*x2*sin(x1) + Br*Ra*k1*k2*k4*x4*sin(2*x1))/(Ra*(k1*k5
- k2*k4*cos(x1)^2)^2);
e = -(Bp*k4*cos(x1) + 2*k1*k6*x2*sin(x1))/(k1*k5 - k2*k4*cos(x1)^2);
f = -(k1*(kb*kt + Br*Ra))/(Ra*(k1*k5 - k2*k4*cos(x1)^2));
h = (k2*kt*cos(x1))/(Ra*(k1*k5 - k2*k4*cos(x1)^2));
i = (k1*kt)/(Ra*(k1*k5 - k2*k4*cos(x1)^2));
%Funcion de transferencia no lineal A(s)/Va(s)
G1 = tf([h (c*i-f*h)], [1 (-b-f) (b*f-a-c*e) (a*f-c*d)])
figure(1);step(G1);
%Funcion de transferencia no lineal T(s)/Va(s)
G2 = tf([i (e*h-b*i) (-a*i+d*h)], [1 (-b-f) (b*f-a-c*e) (a*f-c*d) 0])
figure(2); step(G2);
%% -----
% %Cambios de variables
d1=k1*k5-k4*k2;
d2=k5*Bp+Br*k1;
d3=-k5*k3+Br*Bp;
d4=-Br*k3;
d5=kt*k2;
d6=kb*k1*kt;
d7=kb*Bp*kt;
d8=-kb*k3*kt;
%Función de transferencia linealizada A(s)/Va(s) - Péndulo arriba
G1_lin = tf([d5 0],[Ra*d1 (Ra*d2+d6) (Ra*d3+d7) (Ra*d4+d8)]);
G1_lin = tf(zpk(G1_lin))
figure(1);hold on;step(G1_lin); hold off;
legend('Lineal','No Lineal');grid;title('Respuesta ante un paso -
A(s)/Va(s)');
%Función de transferencia linealizada T(s)/Va(s) - Péndulo arriba
G2_lin = tf([d5*k1 d5*Bp -k3*d5],[Ra*d1*k2 k2*(Ra*d2+d6)
k2*(Ra*d3+d7) ...
k2*(Ra*d4+d8) 0]);
G2_lin = tf(zpk(G2_lin))
figure(2);hold on;step(G2_lin); hold off;
legend('Lineal','No Lineal');grid;title('Respuesta ante un paso -
T(s)/Va(s)');

```

ANEXO 6. Código para obtener y evaluar las funciones de transferencia de la planta

```

%Constantes del motor *****
Ra=5.4351;      %Resistencia - Ohms
La=0.00188;    %Inductancia - Henrios
kb=0.91098;    %Constante Eléctrica - V*s/rad
kt=1.1098;     %Constante Torque - N*m/A
Jm=0.00071;    %Inercia del motor sin carga - kg*(m^2)
Br=0.04166;%0.04166; %Constante de fricción viscosa - Nms/rad
% *****
%Constantes de la estructura
g=9.81;        %Aceleración de la gravedad - m/(s^2)
mp=0.13417;    %Masa del péndulo - kg
Jr=0.0085-Jm; %0.00429; %Momento de inercia del brazo -
r=0.125;       %Longitud/2 del brazo - m

L=0.19604;     %Longitud desde el eje de giro hasta el CM del
péndulo - m
Jp=0.001882;   %Momento de inercia del péndulo -
%Bp=0.000456798; %Constante de fricción viscosa - Nms/rad
Bp=0.00156798;
%Compensación debido a la fricción estática en el péndulo
Fs_dir_neg=0.00035; %Dirección negativa
Fs_dir_pos=0.00035; %Dirección positiva

%Constantes de los modelos
k1=Jp+mp*(L^2);
k2=mp*r*L;
k3=mp*g*L;
k4=mp*r*L;
k5=Jr+Jm+mp*(r^2);
k6=mp*r*L;

run('Test_PID_Pend_Compare');

```

ANEXO 7. Código para simular el control PID lineal del péndulo

```

%% Para graficar solamente la señal real
figure(); tsim =1;
%hold on
subplot(2,2,1);
plot(simout(:,1),simout(:,2),'b');
title('Señal de control'); axis([0 tsim -2.5 2.5]);
subplot(2,2,2);
plot(simout(:,1),simout(:,3),'r');
title('Ángulo del Brazo - Grados'); axis([0 tsim -150 150]);
subplot(2,2,3);
plot(simout(:,1),simout(:,4),'m');
title('Ángulo del Péndulo - Grados'); axis([0 tsim -20 20]);
%hold off

%% Para graficar simulación y real
sim('Test_PID_Pend_Simu');
figure();
subplot(2,2,1); %Señales de control
plot(c(:,1),c(:,2),'r')
hold on;
plot(simout(:,1),simout(:,2),'b')
plot(c1(:,1),c1(:,2),'m')
hold off; xlabel('Tiempo - segundos'); ylabel('Control');
title('Señal de control'); legend('No Lineal','Real','Lineal');
axis([0 tsim -3.5 3.5]);

subplot(2,2,2); %Ángulo Brazo
plot(c(:,1),c(:,3),'r')
hold on;
plot(simout(:,1),simout(:,3),'b')
plot(c1(:,1),c1(:,3),'m')
hold off; xlabel('Tiempo - segundos'); ylabel('Theta - Grados');
title('Ángulo del Brazo'); legend('No Lineal','Real','Lineal');
axis([0 tsim -150 150]);

subplot(2,2,3); %Ángulo Péndulo
plot(c(:,1),c(:,4),'r')
hold on;
plot(simout(:,1),simout(:,4),'b')
plot(c1(:,1),c1(:,4),'m')
hold off; xlabel('Tiempo - segundos'); ylabel('Alpha - Grados');
title('Ángulo del péndulo'); legend('No Lineal','Real','Lineal');
axis([0 tsim -20 20]);

```

ANEXO 8. Código para comparar la simulación e implementación del control PID lineal del péndulo

```

%Constantes del motor *****
Ra=5.4351;      %Resistencia - Ohms
La=0.00188;    %Inductancia - Henrios
kb=0.91098;    %Constante Eléctrica - V*s/rad
kt=1.1098;     %Constante Torque - N*m/A
Jm=0.00071;    %Inercia del motor sin carga - kg*(m^2)
Br=0.04166;%0.04166; %Constante de fricción viscosa - Nms/rad
% *****
%Constantes de la estructura
g=9.81;        %Aceleración de la gravedad - m/(s^2)
mp=0.13417;    %Masa del péndulo - kg
Jr=0.0085-Jm; %0.00429; %Momento de inercia del brazo -
r=0.125;       %Longitud/2 del brazo - m

L=0.19604;     %Longitud desde el eje de giro hasta el CM del
péndulo - m
Jp=0.001882;   %Momento de inercia del péndulo -
%Bp=0.000456798; %Constante de fricción viscosa - Nms/rad
Bp=0.00156798;
%Compensación debido a la fricción estática en el péndulo
Fs_dir_neg=0.00035; %Dirección negativa
Fs_dir_pos=0.00035; %Dirección positiva

%Constantes de los modelos
k1=Jp+mp*(L^2);
k2=mp*r*L;
k3=mp*g*L;
k4=mp*r*L;
k5=Jr+Jm+mp*(r^2);
k6=mp*r*L;

run('Test_PID_Pend_Arm_Compare');

```

ANEXO 9. Código para simular el control PI lineal del brazo


```

%% Para graficar solamente la señal real
figure(); tsim =60;
%hold on
subplot(2,2,1);
plot(simout(:,1),simout(:,2),'b');
title('Señal de control'); axis([0 tsim -2.5 2.5]);
subplot(2,2,2);
plot(simout(:,1),simout(:,3),'r');
title('Ángulo del Brazo - Grados'); axis([0 tsim -150 150]);
subplot(2,2,3);
plot(simout(:,1),simout(:,4),'m');
title('Ángulo del Péndulo - Grados'); axis([0 tsim -20 20]);
%hold off

%% Para graficar simulación y real
sim('Test_PID_Pend_Arm_Simu');
figure();
subplot(2,2,1); %Señales de control
plot(c(:,1),c(:,2),'r')
hold on;
plot(simout(:,1),simout(:,2),'b')
plot(c1(:,1),c1(:,2),'m')
hold off; xlabel('Tiempo - segundos'); ylabel('Control');
title('Señal de control'); legend('No Lineal','Real','Lineal');
axis([0 tsim -2.5 2.5]);

subplot(2,2,2); %Ángulo Brazo
plot(c(:,1),c(:,3),'r')
hold on;
plot(simout(:,1),simout(:,3),'b')
plot(c1(:,1),c1(:,3),'m')
hold off; xlabel('Tiempo - segundos'); ylabel('Theta - Grados');
title('Ángulo del Brazo'); legend('No Lineal','Real','Lineal');
axis([0 tsim -150 150]);

subplot(2,2,3); %Ángulo Péndulo
plot(c(:,1),c(:,4),'r')
hold on;
plot(simout(:,1),simout(:,4),'b')
plot(c1(:,1),c1(:,4),'m')
hold off; xlabel('Tiempo - segundos'); ylabel('Alpha - Grados');
title('Ángulo del péndulo'); legend('No Lineal','Real','Lineal');
axis([0 tsim -20 20]);

```

ANEXO 10. Código para comparar la simulación e implementación del control PI lineal del brazo

```

// Lectura de Encoders para péndulo invertido rotativo
// Octubre2015 - UPS_G - Algoritmo X2 para lectura

// Encoder ubicado en el brazo *****
#define encoder0PinA 2
#define encoder0PinB 3 // Normal DI
#define encoder0PinZ 18 //Cruce por cero - envío de datos por flanco
// Encoder ubicado en el péndulo *****
#define encoder1PinA 21
#define encoder1PinB 20 //Normal DI
#define encoder1PinZ 19 //Cruce por cero

volatile unsigned int encoder0Pos = 0;
volatile unsigned int encoder1Pos = 600; //1200;
unsigned int temporal = 0;
int inPin = 42;
int ClockPin = 43;
boolean selector = 0;
boolean toggle = 0;

void setup() {
// Configuración de pines para interrupciones (entradas)
pinMode(encoder0PinA, INPUT); //Para el brazo
pinMode(encoder0PinB, INPUT);
pinMode(encoder0PinZ, INPUT); //Usado para envío por flancos
// pinMode(encoder0PinZ, INPUT);
pinMode(encoder1PinA, INPUT); //Para el péndulo
pinMode(encoder1PinB, INPUT);
pinMode(encoder1PinZ, INPUT);
// Configuración de pines para selección de datos
pinMode(inPin, INPUT);
pinMode(ClockPin, OUTPUT);
// Encoder pin on interrupt 0 (pin 2)
attachInterrupt(0, doEncoder0A, CHANGE);
// Encoder pin on interrupt 1 (pin 3)
//attachInterrupt(1, doEncoder0B, CHANGE);
// Encoder pin on interrupt 2 (pin 21)
attachInterrupt(2, doEncoder1A, CHANGE);
// Encoder pin on interrupt 3 (pin 20)
//attachInterrupt(3, doEncoder1B, CHANGE);
// Encoder pin on interrupt 4 (pin 19)
attachInterrupt(4, doEncoder1Z, CHANGE);
// Encoder pin on interrupt 5 (pin 18)
attachInterrupt(5, doEncoder0Z, FALLING); //Usado para envío por flancos

// Serial.begin (9600); //Opcional
// Digital Ports for MatLab
DDRC = B11111111; // set PORTC (digital 7~0) to outputs MSB

```

```

DDR_B = B11111111; // set PORTB (digital 7~0) to outputs LSB
}
void loop(){ //Do stuff here
// For test: Monitor Serial
  //Serial.println (encoder0Pos, DEC);
  //Serial.println (encoder1Pos, DEC);
}
// ***** Encoder 0 *****
void doEncoder0A(){
  // look for a low-to-high on channel A
  if (digitalRead(encoder0PinA) == HIGH) {
    // check channel B to see which way encoder is turning
    if (digitalRead(encoder0PinB) == LOW) {
      encoder0Pos = encoder0Pos + 1;    // CW
    }
    else {
      encoder0Pos = encoder0Pos - 1;    // CCW
    }
  }
  else // must be a high-to-low edge on channel A
  {
    // check channel B to see which way encoder is turning
    if (digitalRead(encoder0PinB) == HIGH) {
      encoder0Pos = encoder0Pos + 1;    // CW
    }
    else {
      encoder0Pos = encoder0Pos - 1;    // CCW
    }
  }
}
// ***** Encoder 1 *****
void doEncoder1A(){
  // look for a low-to-high on channel A
  if (digitalRead(encoder1PinA) == HIGH) {
    // check channel B to see which way encoder is turning
    if (digitalRead(encoder1PinB) == LOW) {
      encoder1Pos = encoder1Pos + 1;    // CW
    }
    else {
      encoder1Pos = encoder1Pos - 1;    // CCW
    }
  }
  else // must be a high-to-low edge on channel A
  {
    // check channel B to see which way encoder is turning
    if (digitalRead(encoder1PinB) == HIGH) {
      encoder1Pos = encoder1Pos + 1;    // CW
    }
  }
}

```

```

else {
  encoder1Pos = encoder1Pos - 1;    // CCW
}
}
}

void doEncoder1Z(){
  if ((encoder1Pos>900)&&(encoder1Pos<1800))
    encoder1Pos = 1200; //Reset por cruce por cero
  if ((encoder1Pos>2100)&&(encoder1Pos<3000))
    encoder1Pos = 2400; //Reset por cruce por cero
}
//***** Envío de Datos *****
void doEncoder0Z(){
  //encoder0Pos = 0; //Reset por cruce por cero
  delayMicroseconds(100);
  selector = digitalRead(inPin); // read the input pin
  if (selector==0){ //Cambio para lectura adecuada
  // For MatLab Measure (Encoder Arm)
    temporal = encoder0Pos << 8;
    PORTB = temporal >> 8;
    digitalWrite(43,bitRead(PORTB,3));
    PORTC = encoder0Pos >> 8;
    //digitalWrite(43,bitRead(temporal >> 8,3));
  }
  else {
  // For MatLab Measure (Encoder Pendulum)
    temporal = encoder1Pos << 8;
    PORTB = temporal >> 8;
    digitalWrite(43,bitRead(PORTB,3));
    PORTC = encoder1Pos >> 8;
    //digitalWrite(43,bitRead(temporal >> 8,3));
  }
  //digitalWrite(43,toggle); //señal pulsande de test
  //toggle = ~toggle;
}
}

```

ANEXO 11. Código empleado en la placa Arduino Mega

```

Ra = 5.4351;      %Resistencia de armadura - Ohms
La = 0.00188;    %Inductancia de armadura - H
kb = 0.91098;    %Constante de la fuerza FCEM - V*s/rad
kt = 1.1098;     %Constante del par electromagnético - N*m/A
b = 0.04166;     %0.04166 constante de fricción viscosa - N*m*s/rad
J = 0.0085;      %0.005 Momento de inercia equivalente - kg*m^2 ;0.0021
;J = 0.00070926

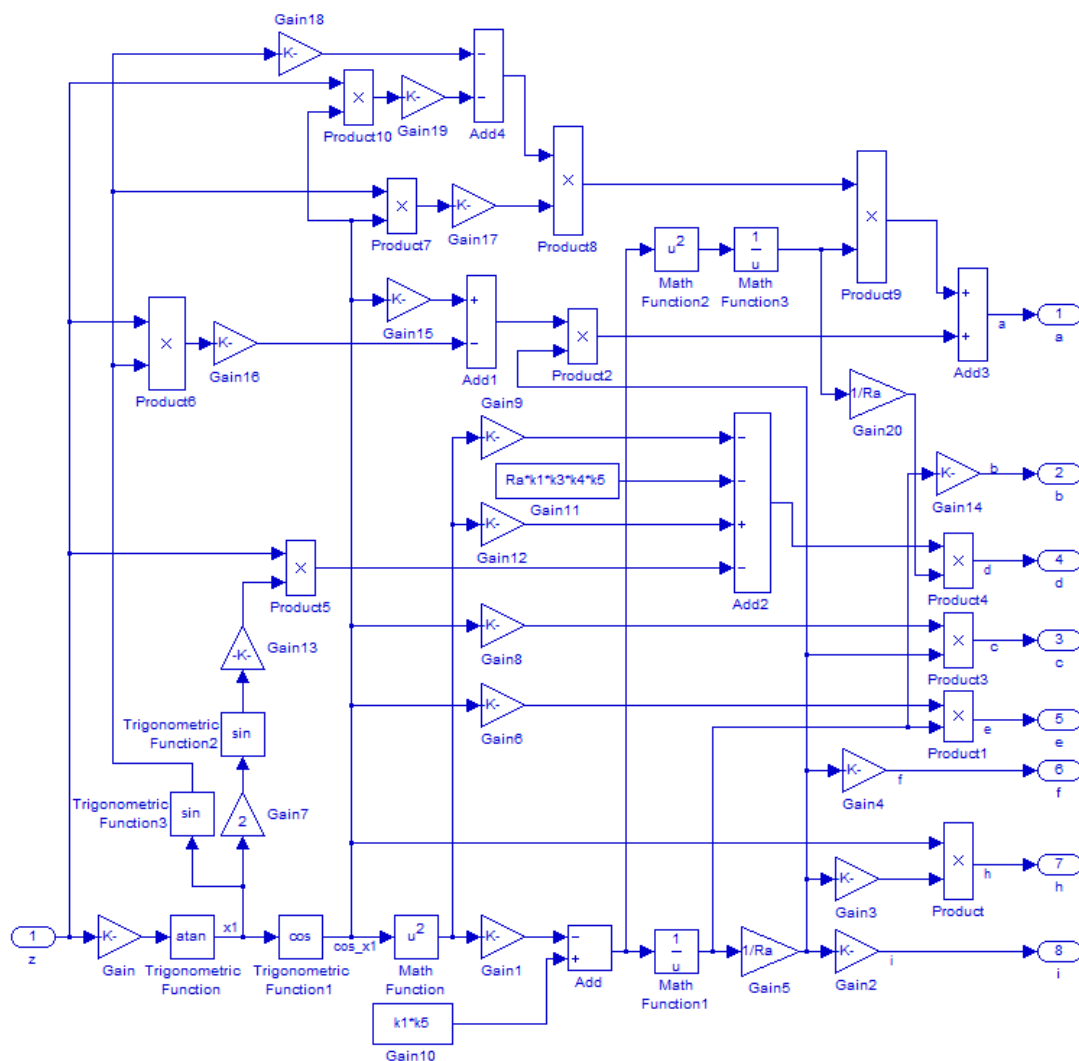
%%
% OJO: Antes de ejecutar hay importar los datos del motor, ejecutar
el mdl
% del modelo matemático
sim('Experimental_Model');
figure();
subplot(2,1,1);
plot(c(:,1),c(:,2),'r') %con tendencias
hold on;
plot(simout(:,1),simout(:,3),'b')
hold off; xlabel('Tiempo - segundos'); ylabel('Theta - grados');
title('Salidas'); legend('Modelo MotorDC','Real MotorDC'); grid;

subplot(2,1,2);
plot(c(:,1),c(:,3),'r')
hold on;
plot(simout(:,1),simout(:,2),'b')
hold off; xlabel('Tiempo - segundos'); ylabel('u');
title('Entradas'); legend('Modelo MotorDC','Real MotorDC'); grid;

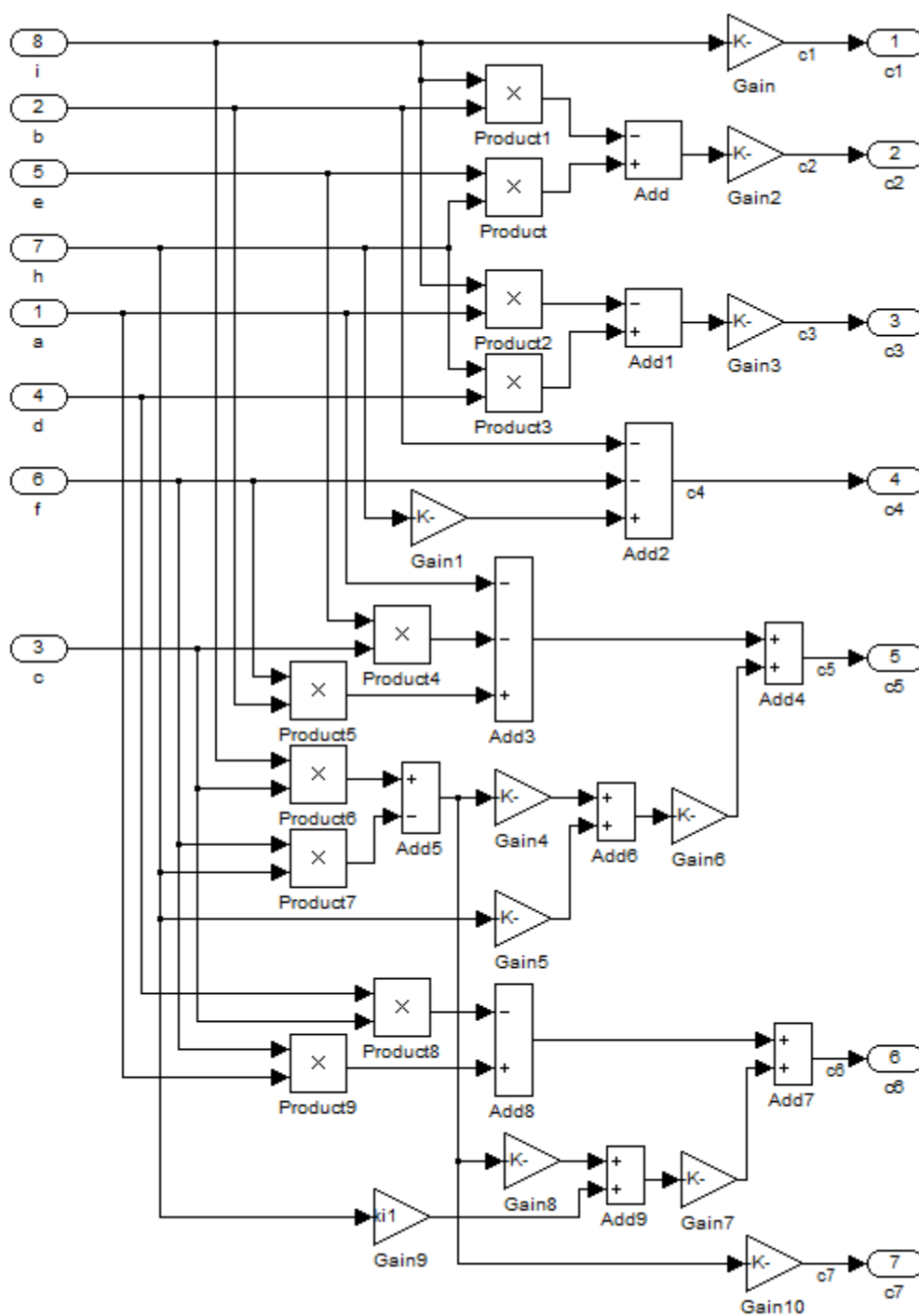
%%Error RMS (roots mean square)
RMSE_error_theta = sqrt(mean((simout(:,3)-
c(1:length(simout),2)).^2))
Rel_error_theta=(RMSE_error_theta/sqrt(mean(simout(:,3).^2)))*100

```

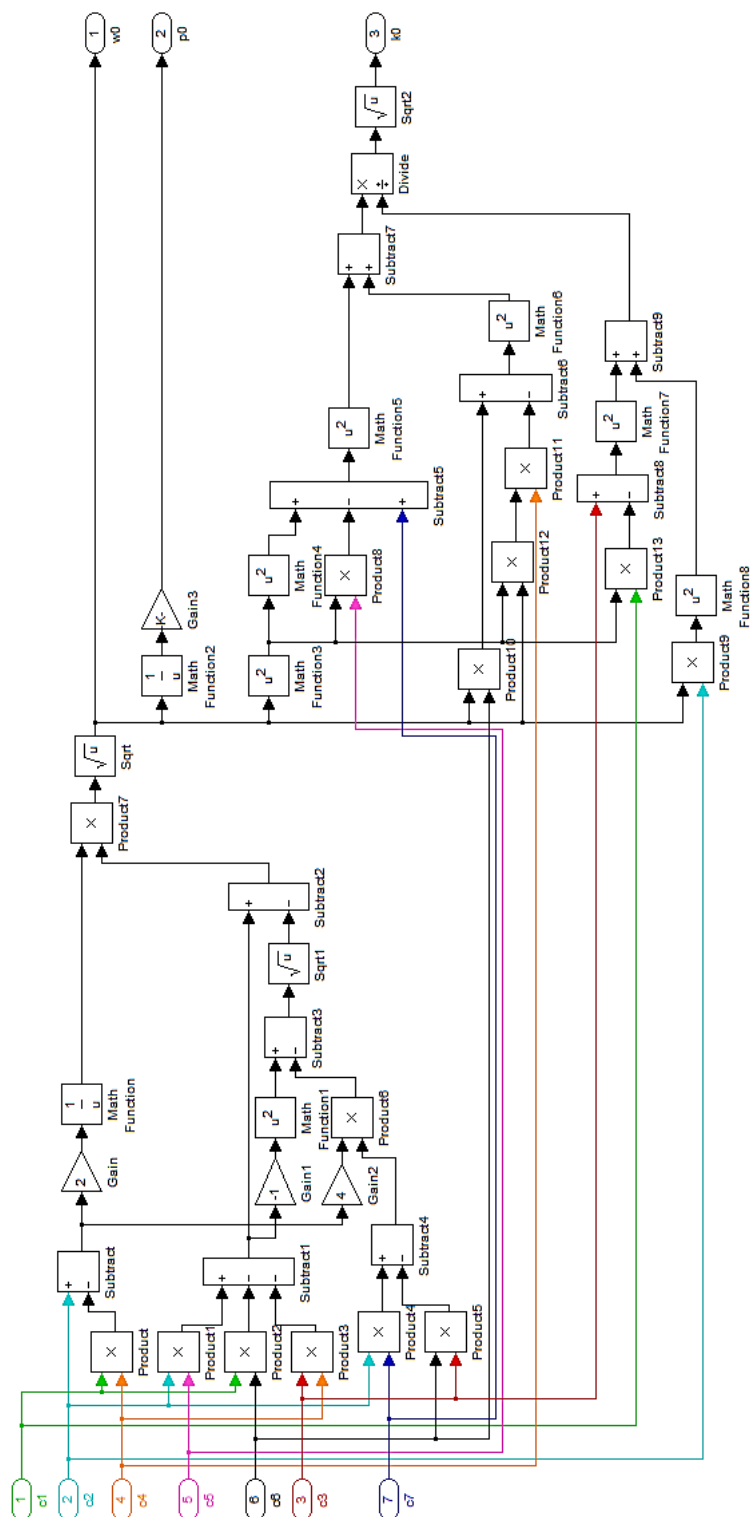
ANEXO 12. Código para validar el modelo matemático del motor DC



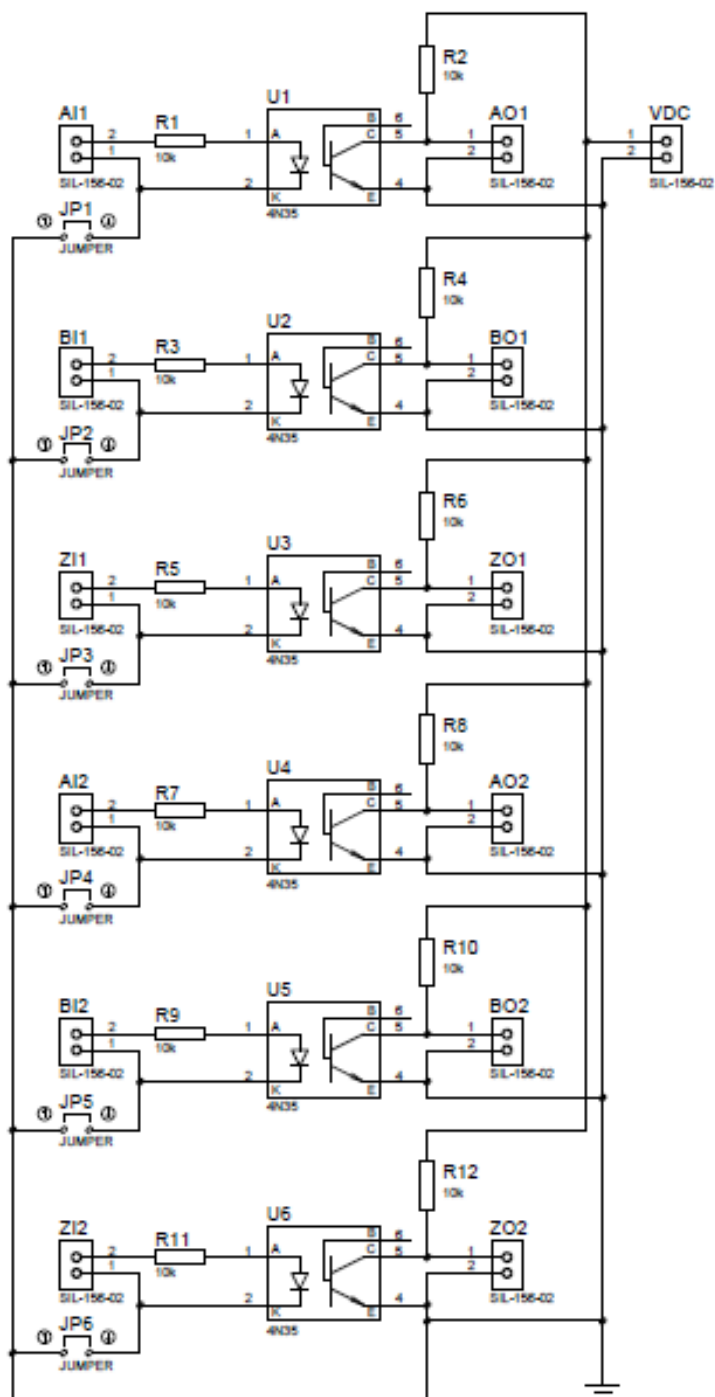
ANEXO 13. Detalle del bloque Subsystem utilizado en el controlador no lineal



ANEXO 14. Detalle del bloque Subsystem1 utilizado en el controlador no lineal



ANEXO 15. Detalle del bloque Subsystem2 utilizado en el controlador no lineal



ANEXO 16. Diagrama electrónico para tarjeta de acoplamiento de sensores