



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

**Facultad de Ingeniería en Mecánica y Ciencias de la
Producción**

“Modelización, simulación y diseño del control de un mecanismo de péndulo invertido simple, aplicado a su auto-levantamiento y mantenimiento de posición vertical mediante el uso de herramientas gráficas de programación”.

**TRABAJO FINAL DE GRADUACIÓN:
PROYECTO DE GRADUACIÓN**

Previa a la obtención del título de:

INGENIERO MECÁNICO

Presentado por:

Geancarlos Zamora Olea

GUAYAQUIL – ECUADOR

Año: 2016

AGRADECIMIENTO

A todos por su apoyo y paciencia, especialmente paciencia.

A mi familia, en especial a mis padres, por su confianza y por ser mi sostén anímico durante todo el proceso.

A ambos, mi director y vocal, Ing. Eduardo Orcés e Ing. Jorge Roca, por sus consejos, sabia asistencia, y guía en este trabajo.

Al Ing. Manuel Helguero y el personal de CAMPRO, por su inmensa ayuda en la construcción del equipo.

A mis amigos y compañeros de clases.

DEDICATORIA

A MIS PADRES

A MI HERMANA Y

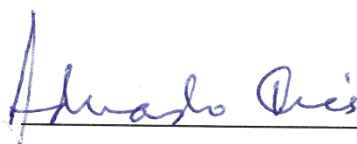
SOBRINOS

A MIS AMIGOS

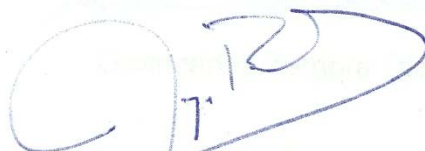
TRIBUNAL DE SUSTENTACIÓN



M.Sc. Jorge Duque R.
DECANO DE LA FIMCP
PRESIDENTE



Ing. Eduardo Orcés P.
DIRECTOR



Ing. Jorge Roca G.
VOCAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este trabajo final de graduación me corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Graduación de la ESPOL)



Geancarlos Zamora Olea

RESUMEN

El péndulo invertido simple es un problema clásico en el campo de la ingeniería de control, puesto que varias de sus características como el ser naturalmente inestable, altamente no lineal, tener una entrada, y dos salidas en el caso del péndulo invertido en móvil o carro, son retos que lo vuelven una interesante banco de pruebas para el análisis y aplicación de diversas estrategias de control, si se suma la relativa sencillez del equipo puede entenderse aún más su importancia en el campo académico de los sistemas de control. Los resultados obtenidos a partir de su estudio, sin embargo, tienen aplicaciones a problemas reales como la estabilización de misiles, barcos y cohetes, vehículos de dos ruedas con control de auto-balanceo, y modelización y reproducción en robots de la caminata bípeda.

El enfoque de buena parte de los controladores ha sido la estabilización o mantenimiento de posición vertical invertida del péndulo, donde el usuario es quien tiene que manualmente levantar al péndulo hasta la zona de control. No obstante, un problema adicional para el cual se han venido ideando diferentes estrategias de control es lograr que el péndulo se levante sin la intervención manual del usuario, y más bien lo haga por acción del movimiento del móvil en el que el péndulo está acoplado, esto es conocido como “swing-up” o auto-levantamiento.

Con tal contexto, en este proyecto de graduación se ha documentado el desarrollo, diseño e implementación y pruebas del sistema de control para un equipo de péndulo invertido simple aplicado a su auto-levantamiento y mantenimiento de posición vertical.

La metodología seguida fue identificar y analizar los principios físicos bajo los cuales el comportamiento del equipo está regido y con ello hacer una representación del mismo mediante modelos matemáticos que son usados para simulación, luego seleccionar una estrategia de control con ciertos requerimientos, diseñar y simular dicho control en los modelos simulados, y por último implementar el sistema de control en el equipo para su posterior análisis y ajuste. Software matemático y herramientas gráficas de programación fueron utilizados tanto para las simulaciones como para el control implementado.

Como resultado final se obtuvo un control satisfactorio del equipo tanto para su auto-levantamiento como para su estabilización, aunque varios ajustes fueron necesarios dadas ciertas incompatibilidades entre el modelo simulado y el equipo real, las posibles razones de ello son explicadas en el penúltimo capítulo.

ÍNDICE GENERAL

RESUMEN	ii
ÍNDICE GENERAL.....	iv
ABREVIATURAS	vi
SIMBOLOGÍA	vii
ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE TABLAS	xvii
INTRODUCCIÓN	1
CAPÍTULO 1	
1. DESCRIPCIÓN GENERAL.....	3
1.1. Antecedentes.....	3
1.2. Descripción del equipo.....	13
1.3. Alcance y Aplicaciones del Proyecto	37
CAPÍTULO 2	
2. MODELO MATEMÁTICO DEL SISTEMA.....	39
2.1. Obtención de ecuaciones diferenciales del sistema	40
2.2. Obtención de funciones de transferencia del sistema	69
CAPITULO 3	
3. SIMULACIÓN DEL MODELO MATEMÁTICO	77
3.1 Representación del sistema en una plataforma matemática de programación	77
3.2 Diagrama de bloques del sistema en una plataforma gráfica de simulación	95
3.3 Análisis de respuesta temporal del sistema simulado y ajustes al modelo	117

CAPÍTULO 4

4. DISEÑO DEL CONTROLADOR	123
4.1 Selección del tipo de controlador	123
4.2 Selección de parámetros de desempeño.....	132
4.3 Proceso de diseño del controlador	133
4.4 Simulación del controlador.....	178

CAPÍTULO 5

5. IMPLEMENTACIÓN DEL CONTROL EN UN ENTORNO GRÁFICO DE PROGRAMACIÓN	186
5.1 Introducción a la plataforma gráfica de programación.	186
5.2 Programación e implementación del control	205
5.3 Interfaz gráfica de usuario.	232
5.4 Adquisición y envío de señales en plataforma matemática de programación.	232

CAPÍTULO 6

6. ANÁLISIS DE DATOS EXPERIMENTALES	240
6.1 Análisis del desempeño del controlador en el modelo real.	240
6.2 Ajustes finales del controlador.	250

CAPÍTULO 7

7. CONCLUSIONES Y RECOMENDACIONES.....	263
--	-----

APÉNDICES**BIBLIOGRAFÍA**

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
FIMCP	Facultad de Ingeniería Mecánica y Ciencias de la Producción
CAMPRO	Campo Avanzado de Mecanización y Prototipado Rápido
PID	Proporcional, Integral y Derivativo
BLDCM	Brushless Direct Current Motor (Motor de Corriente Directa sin Escobillas)
FCEM	Fuerza Contra Electro-Motriz
RPM	Revoluciones Por Minuto
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor (transistor de efecto de campo metal-óxido-semiconductor)
PWM	Pulse Width Modulation (Modulación de Ancho de Pulso)
NEMA	National Electrical Manufacturers Association
SISO	Single Input Single Output (Una Entrada, Una Salida)
SIMO	Single Input Multiple Output (Una Entrada, Múltiples Salidas)
MIMO	Multiple Input Multiple Output (Múltiples Entradas, Múltiples Salidas)
LGR	Lugar Geométrico de las Raíces

SIMBOLOGÍA

Kg	Kilogramo (unidad de masa)
m	Metro (unidad de longitud)
oz	Onza (unidad de masa)
pulg	Pulgada (unidad de longitud)
V, Vdc	Voltios de corriente directa (unidad de voltaje)
s	Segundo (unidad de tiempo)
π	Pi
rad	Radianes (unidad de posición angular)
Θ	Teta (ángulo, posición angular del péndulo)
ω	Omega (velocidad angular)
α	Alfa (aceleración angular)
Ω	Ohmios (unidad de resistencia eléctrica)
A	Amperios (unidad de corriente eléctrica)
N	Newtons (unidad de fuerza)

ÍNDICE DE FIGURAS

Figura 1.1.	Diagrama simplificado del péndulo de furuta [1].....	6
Figura 1.2.	Diagrama simplificado del péndulo invertido simple lineal o en un carro [1].....	7
Figura 1.3.	Segway pt [2].....	10
Figura 1.4.	Estabilidad de cohetes [3].....	11
Figura 1.5.	Primeros robots bípedos (de izquierda a derecha) wl-1, wl-3, wap-1 y wap-2 [4]	13
Figura 1.6.	Fuente de poder.	16
Figura 1.7.	Driver del motor.	16
Figura 1.8.	Motor sin escobillas.	17
Figura 1.9.	Reductora 3:1.	18
Figura 1.10.	Polea conductora.....	19
Figura 1.11.	Polea conducida, con eje y rodamientos.	20
Figura 1.12.	Banda dentada.	21
Figura 1.13.	Carro o bloque móvil.....	21
Figura 1.14.	Péndulo con peso en el extremo libre.....	22
Figura 1.15.	Encoder óptico incremental.	23
Figura 1.16.	Tarjeta de adquisición de datos ni pci-6221.....	24
Figura 1.17.	Bornera y cable.....	25
Figura 1.18.	Diagrama del sistema de péndulo invertido.....	26
Figura 1.19.	Diagrama de conexiones disponibles en el driver del motor....	29
Figura 1.20.	Ensamblaje de driver, motor, reductora y polea conductora....	30

Figura 1.21. Detalle del móvil, banda, guías de deslizamiento y péndulo....	32
Figura 1.22. Diagrama de un encoder óptico incremental [6].	33
Figura 1.23. Canales de salida a y b en un encoder de cuadratura [6].....	34
Figura 1.24. Detalle de encoders instalados para medición de posición lineal (izq.) Y posición angular (der.).....	34
Figura 1.25. Detalle de la caja de conexiones, borneras y tarjetas de adquisición de datos	35
Figura 1.26. Sistema de péndulo invertido con detalle de caja de conexiones	37
Figura 2.1. Estructura interna de un motor dc sin escobillas	41
Figura 2.2. Diagramas de configuración “wye” o “estrella [7]”	42
Figura 2.3. Diagramas de configuración delta [7]	43
Figura 2.4. Curva de velocidad angular vs. Tiempo en simulación de modelo de motor dc [8]	45
Figura 2.5. Curva de velocidad angular vs. Tiempo en simulación de modelo de motor bldc [8]	45
Figura 2.6. Curva de torque vs. Tiempo en simulación de modelo de motor dc [8].....	46
Figura 2.7. Curva de torque vs. Tiempo en simulación de modelo de motor bldc [8]	46
Figura 2.8. Diagrama básico de un motor dc [9]	48
Figura 2.9. Diagrama de inercias de carga reflejadas a la armadura.....	53
Figura 2.10. Esquema simplificado del péndulo invertido	58
Figura 2.11. Diagrama de cuerpo libre del carro o móvil	61
Figura 2.12. Diagrama de cuerpo libre del péndulo	62
Figura 3.1. Entorno o espacio de trabajo de trabajo de matlab	79

Figura 3.2.	Respuesta del sistema a entrada impulso, en lazo abierto.....	94
Figura 3.3.	Respuesta del sistema a entrada paso, en lazo abierto	95
Figura 3.4.	Entorno de trabajo de Simulink.....	97
Figura 3.5.	Diagrama de bloques para graficar función seno	98
Figura 3.6.	Diagrama de bloques para graficar función seno con ganancia 10.....	99
Figura 3.7.	Bloque función de transferencia y su cuadro de diálogo.....	101
Figura 3.8.	Ingreso de numeradores y denominadores en bloques de funciones de transferencia.....	102
Figura 3.9.	Diagrama de bloques preliminar del sistema no simplificado.	104
Figura 3.10.	Diagrama de bloques del sistema no simplificado en lazo abierto.....	105
Figura 3.11.	Diagrama de bloques del sistema simplificado preliminar.	106
Figura 3.12.	Configuración de estilos de líneas del bloque de visualización.	107
Figura 3.13.	Diagrama de bloques del sistema simplificado en lazo abierto.	108
Figura 3.14.	Configuración de la duración y tiempo de muestro de la simulación.....	109
Figura 3.15.	Respuesta escalón del sistema en lazo abierto.....	109
Figura 3.16.	Disposición inicial de bloques para simulación no lineal [13].	111
Figura 3.17.	Configuración de bloques de funciones.....	112
Figura 3.18.	Configuración de conectores mux.	113
Figura 3.19.	Conexiones de los bloques del sistema no lineal.	114
Figura 3.20.	Conexiones de los bloques del sistema no lineal.	115
Figura 3.21.	Diagrama simplificado del sistema no lineal.	116

Figura 3.22. Respuesta del sistema no lineal en reposo a una entrada tipo paso.....	116
Figura 3.23. Efectos de los polos de un sistema en su respuesta escalón [19].....	118
Figura 3.24. Respuestas al impulso según ubicaciones de polos [20].....	120
Figura 3.25. Voltajes en las 3 fases para un voltaje de referencia aproximado a 1.62 v.	122
Figura 4.1. Diagrama de bloques de un sistema en lazo cerrado.....	125
Figura 4.2. Diagrama de bloques de un sistema con controladores en cascada.	131
Figura 4.3. Diagrama de bloques del lazo interno.....	133
Figura 4.4. Arquitectura de diagrama de bloques en sisotool.....	137
Figura 4.5. Importación de modelo para el bloque g.....	139
Figura 4.6. Selección de tipo de gráfica de respuesta en sisotool.	140
Figura 4.7. Respuesta a perturbación tipo impulso con retroalimentación unitaria.	141
Figura 4.8. Ventana gráfica de diseño de sisotool.	142
Figura 4.9. Detalle de los polos dominantes.	143
Figura 4.10. Ingreso de parámetros de diseño	144
Figura 4.11. Gráfica del lgr con la restricción de diseño.	145
Figura 4.12. Ingreso de restricciones de error de estado estable.....	146
Figura 4.13. Restricciones de error de estado estable en gráfica de respuesta a perturbación impulso.....	147
Figura 4.14. Modificación del lgr con $c=-1$	148
Figura 4.15. Respuesta a perturbación impulso con $c=-8$	149
Figura 4.16. Gráfica del lgr con un integrador (polo en el origen).....	150

Figura 4.17. Gráfica del lgr con un cero real añadido.	151
Figura 4.18. Respuesta a perturbación con un cero real añadido.	152
Figura 4.19. Lgr y respuesta al impulso con cero en -0.824 y ganancia 10.597.....	153
Figura 4.20. Lgr para $c=-2.3103*(1 + 4.3s / s)$	154
Figura 4.21. Respuesta al impulso para $c=-2.3103*(1 + 4.3s / s)$	154
Figura 4.22. Diagrama de bloques de planta linealizada con control en cascada	157
Figura 4.23. Respuesta a la perturbación con control interno ($p= -10.0448$, $i=$ -2.3103).	158
Figura 4.24. Respuesta a la perturbación con el control interno: con acción proporcional duplicada (superior) y sin modificar (inferior). ...	159
Figura 4.25. Respuesta a la perturbación con acción proporcional $p=-80$ en el control interno	160
Figura 4.26. Respuesta a la perturbación con $p=-80$, $i=-40$ y $d=-20$ en el control interno	161
Figura 4.27. Bloque de la función de auto-levantamiento.....	165
Figura 4.28. Configuración de entradas y salidas del bloque de la función de auto-levantamiento.	166
Figura 4.29. Configuración de entradas tipo parámetro de la máscara del bloque de auto-levantamiento.....	167
Figura 4.30. Introducción de título y texto de la máscara del bloque de auto- levantamiento.	168
Figura 4.31. Vista de la máscara del bloque de la función de auto- levantamiento.	169
Figura 4.32. Diagrama de bloques completo del sistema no lineal para el auto-levantamiento.	170
Figura 4.33. Auto-levantamiento para $ng=0.1$ y $k=10$	172

Figura 4.34. Auto-levantamiento para $n_g=0.5$ y $k=10$.	173
Figura 4.35. Auto-levantamiento para $n_g=1$ y $k=10$.	174
Figura 4.36. Auto-levantamiento para $n_g=1.25$ y $k=10$.	175
Figura 4.37. Detalle del auto-levantamiento para $n_g=1.25$ y $k=10$.	176
Figura 4.38. Auto-levantamiento para $n_g=1.25$ y $k=2$.	177
Figura 4.39. Auto-levantamiento para $n_g=1.25$ y $k=1.5$.	178
Figura 4.40. Estructura interna de bloque selector de control.	181
Figura 4.41. Diagrama de bloques de planta no lineal con control de auto-levantamiento y estabilización.	182
Figura 4.42. Gráficas de respuesta del auto-levantamiento, transición y estabilización en el modelo no lineal.	183
Figura 4.43. Gráficas de respuesta del auto-levantamiento, transición y estabilización en el modelo no lineal con acción integral y derivativa en pid externo.	184
Figura 5.1. Bloques de la librería real-time windows target (simulink® desktop real-time™).	187
Figura 5.2. Ícono del bloque de entrada de encoder.	188
Figura 5.3. Ícono del bloque de salida digital.	193
Figura 5.4. Ícono del bloque de salida analógica.	196
Figura 5.5. Diagrama de ejecución del simulink desktop real-time en modo externo.	201
Figura 5.6. Configuración de la ventana de generación de código.	203
Figura 5.7. Datos del procesador en ventana de implementación de hardware.	204
Figura 5.8. Configuración de bloque entrada de encoder para medición de posición angular del péndulo.	208

Figura 5.9. Configuración de la tarjeta de adquisición de datos para la entrada de encoder.....	209
Figura 5.10. Configuración de ganancia de conversión de pulsos a radianes	211
Figura 5.11. Inicialización de la medición de posición angular del péndulo.	212
Figura 5.12. Diagrama de bloques de medición de la posición angular.....	213
Figura 5.13. Diagrama de bloques de medición de la velocidad angular (sombreado en celeste).	214
Figura 5.14. Diagrama de bloques de medición de la posición lineal del carro.....	215
Figura 5.15. Configuración del bloque de salida analógica de voltaje.	217
Figura 5.16. Diagrama de bloques de la salida analógica de voltaje.	218
Figura 5.17. Configuración del bloque de salida digital de dirección.	219
Figura 5.18. Diagrama de bloques de salida digital de dirección.....	220
Figura 5.19. Configuración del bloque de la salida digital de parada/arranque.	221
Figura 5.20. Diagrama de bloques de la lógica de protección de extremos.	223
Figura 5.21. Sub-sistema de protección de extremos.....	214
Figura 5.22. Diagrama de bloques de manejo de paradas.	225
Figura 5.23. Diagrama de bloques completo de la planta.....	226
Figura 5.24. Bloque sub-sistema de la planta.....	226
Figura 5.25. Bloque selector de control.	227
Figura 5.26. Bloques de control.	227
Figura 5.27. Configuración del bloque de la relación no lineal de $e(s)$ vs v_{ref}	229

Figura 5.28. Conexión del selector de parada con el bloque planta.	230
Figura 5.29. Bloque de visualización de señales.....	231
Figura 5.30. Diagrama de bloques completo de la implementación del control.....	231
Figura 5.31. Gráfica de adquisición de datos obtenida con matlab.	239
Figura 6.1. Inicio de auto-levantamiento en modelo real ($n_g=1.25$, $k=3$).	241
Figura 6.2. Oscilación sostenida del auto-levantamiento en modelo real ($n_g=1.25$, $k=3$).	242
Figura 6.3. Error de estado estable del control diseñado en el modelo real.	243
Figura 6.4. Error de estado estable del control diseñado en el modelo real amplificado.....	244
Figura 6.5. Respuesta a la perturbación del control diseñado en el modelo real.....	245
Figura 6.6. Respuesta a la perturbación con caída y recuperación del control diseñado en el modelo real.	246
Figura 6.7. Diagrama de bloques de sistema linealizado con visualización de torque motor y fuerza.....	247
Figura 6.8. Fuerza sobre el carro y torque del motor para una entrada escalón unitario de voltaje.	248
Figura 6.9. Fuerza sobre el carro y torque del motor para una entrada unitaria con perfil rampa de 500 ms.....	249
Figura 6.10. Respuesta a la perturbación del sistema con control pid interno ($p=-100$, $i=-50$, $d=-20$).....	252
Figura 6.11. Posición del carro con control externo con a) $p=0.1$ b) $p=0.2$ c) $p=0.5$ y d) $p=1$	253
Figura 6.12. Posición del carro con control externo con a) $d=0.05$ b) $d=0.1$ y c) $d=0.2$	254

Figura 6.13. Posición del carro con control externo con $d=0.2$ (superior) $d=0.3$ (inferior).	255
Figura 6.14. Posición del carro con control externo con: a) $i=0$ b) $i=0.01$ c) $i=0.02$ y d) $i=0.04$	256
Figura 6.15. Error de estado estable del control corregido.	256
Figura 6.16. Error de estado estable del control corregido amplificado.	257
Figura 6.17. Respuesta a la perturbación del control corregido.....	258
Figura 6.18. Auto-levantamiento fallido con $n_g=2.5$	259
Figura 6.19. Auto-levantamiento exitoso con $n_g=8$	260
Figura 6.20. Auto-levantamiento fallido con $n_g=8$ y $K=2$	261
Figura 6.21 Auto-levantamiento exitoso y mejorado con $n_g=8$ y $K=4$	262

ÍNDICE DE TABLAS

Tabla 1.1.	Conexiones entre dispositivos y bornera de tarjeta de adquisición de datos.	36
Tabla 2.1.	Tabla de teoremas de transformada de Laplace [12].	70
Tabla 3.1.	Operaciones aritméticas en matlab.	81
Tabla 3.2.	Funciones trigonométricas básicas en matlab.	81
Tabla 3.3.	Parámetros físicos del sistema de péndulo invertido.	86

INTRODUCCIÓN

Este proyecto de graduación está formado por siete capítulos, cada uno dividido en diferentes subtemas según convenga, aunque el diseño de un controlador tiene un componente de proceso iterativo, los capítulos están organizados de manera secuencial y lineal. A continuación se describe brevemente con qué lidia cada uno:

En el capítulo 1 se dieron los antecedentes del péndulo invertido, trabajos anteriores, sistemas y conceptos similares, historia y aplicaciones. Se mencionaron los objetivos que buscan alcanzarse con este proyecto y por último, una descripción del equipo y sus componentes, funcionamiento general y por partes.

En el capítulo 2 se obtuvieron las funciones de transferencia que describen el comportamiento del sistema, para ello primero fueron determinadas las ecuaciones matemáticas en base a los principios físicos básicos. Luego se aplicó la Transformada de Laplace, y se reordenaron los términos de tal forma que se relacionara una entrada con una salida del sistema.

El capítulo 3 se describió el proceso de la simulación del sistema usando en la primera fase una plataforma matemática de programación. Los

“scripts” o programas obtenidos en la fase anterior fueron usados para la siguiente fase donde la simulación se llevó a cabo en una plataforma gráfica de programación usando diagramas de bloques interconectados.

En el capítulo 4 se detalló el proceso de diseño del controlador, empezando por la definición de los parámetros de desempeño, seguido de la selección del tipo de controlador o estrategia de control a aplicar y el desarrollo del controlador en sí.

En el capítulo 5 se describió la implementación del controlador usando la misma plataforma gráfica de la segunda fase del capítulo 3, esta vez enfocada a la generación y adquisición de datos en tiempo real.

En el capítulo 6 se realizó un análisis de la respuesta del sistema, comparando la simulación del modelo, con los resultados del controlador aplicado al sistema en tiempo real, y se realizaron correcciones al controlador.

Finalmente en el capítulo 7 se presentaron las conclusiones y las recomendaciones.

CAPÍTULO 1

1. DESCRIPCIÓN GENERAL

En este capítulo se hizo una breve reseña y repaso de los antecedentes históricos del concepto del péndulo invertido y otros similares. Se presentaron además las diversas aplicaciones fuera del ámbito meramente académico como justificación de la importancia de su estudio. Luego se detalló una descripción de los componentes, clasificados de acuerdo a la función que cumplen, y el funcionamiento básico de los componentes principales. Con ese conocimiento previo, se procedió a establecer los alcances y aplicaciones del proyecto.

1.1. Antecedentes

Gran parte de la optimización de procesos se debe a la automatización e implementación de sistemas de control cada vez más eficaces y que necesitan menos supervisión humana,

previniendo los errores de los mismos, logrando mayor eficiencia, rapidez y precisión.

Muchos de los procesos que deben controlarse son sistemas complejos que tienden a la inestabilidad, y que salvo simulaciones o visitas técnicas no pueden ser constatados por los estudiantes de ingeniería mecánica con formación en sistemas de control.

Un ejemplo sencillo y dinámico de este tipo de sistemas es el péndulo invertido simple, que consta de una barra articulada sobre un móvil que se mueve horizontalmente en el mismo plano de la barra cuya posición vertical la cual es inestable, es mantenida mediante la aplicación de fuerzas en el móvil por un actuador. En este proyecto de graduación se ha documentado el desarrollo e implementación de un sistema de control para la auto-elevación y mantenimiento en posición vertical de un equipo de péndulo invertido simple. Se ha dejado abierta la posibilidad de mejoras futuras, tanto en su parte física como en la implementación y prueba de otras estrategias o algoritmos de control.

Diversos sistemas han sido usados a lo largo de la relativamente reciente historia de la ingeniería de control, como bancos de prueba

para la experimentación y desarrollo de diferentes estrategias de control, entre ellos están el sistema barra-esfera, el Pendubot, el péndulo de Furuta y el péndulo invertido en un carro o móvil. Ante la pregunta de por qué estos sistemas en especial se han convertido en bancos de prueba en el diseño de sistemas de control habría que observar un par de características comunes en estos equipos: la relativa simpleza de sus mecanismos y el ser sistemas no lineales e inestables.

Como bancos de prueba, estos equipos son mayormente utilizados en el campo de la investigación y la enseñanza, y como tales se busca en primer lugar que su construcción sea sencilla, en especial comparada con la de los sistemas o aplicaciones reales donde se implementarán los resultados y en segundo lugar que presenten verdaderos retos para el diseño del control, algo que un sistema naturalmente inestable ofrece en mejor manera que aquellos que presentan comportamientos que tienden a la estabilidad en lazo abierto.

El péndulo invertido, es quizás de los sistemas mencionados en el primer párrafo, uno de los más conocidos, al punto que existen diversas versiones del mismo, como de tipo rotacional de dos

brazos conocido normalmente como el péndulo de Furuta, concebido en 1992 en el Instituto de Tecnología de Tokio y llamado así debido a su co-inventor Katsuhisa Furuta (Figura 1.1), el de rueda de reacción, el rotacional de un solo brazo y el de tipo lineal o de carro en pista el cual fue objeto de estudio de este proyecto de graduación.

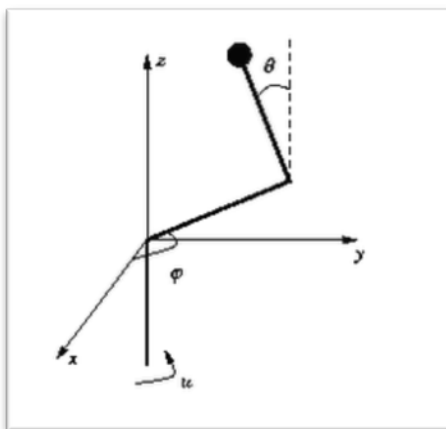
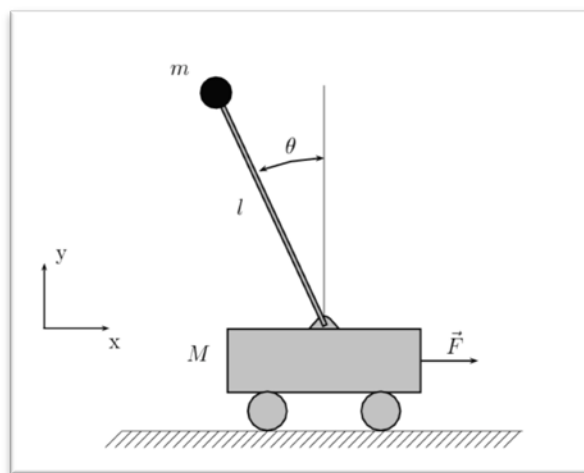


FIGURA 1.27. DIAGRAMA SIMPLIFICADO DEL PÉNDULO DE FURUTA [1]

El péndulo invertido lineal o en un carro, es un sistema no-lineal, sub-actuado y naturalmente inestable. Consiste principalmente y en su forma más sencilla en un carro o móvil de masa M en el cual esta pivoteado el péndulo considerado en este caso como una varilla de masa despreciable y longitud l con una masa m en su extremo, este péndulo gira libremente en el plano xy perpendicular

al eje de rotación (Figura 1.2), el carro a su vez, se desplaza de forma lineal en una pista o riel de longitud finita, el control de la posición del péndulo en su modo invertido se logra aplicando una fuerza F sobre el móvil lo cual provoca su desplazamiento y éste a su vez afecta la posición angular θ del péndulo.



**FIGURA 1.28. DIAGRAMA SIMPLIFICADO DEL PÉNDULO
INVERTIDO SIMPLE LINEAL O EN UN CARRO [1]**

La historia del péndulo invertido lineal se remonta a los años cincuenta, sirviendo desde su concepción como medio de enseñanza de teorías de control. La primera solución a la estabilización del péndulo invertido usando control lineal fue dada por J. K. Robergeen 1960 en su tesis de graduación "The Mechanical Seal".

Con el desarrollo, mayor disponibilidad y mejoras en capacidad de procesamiento de las computadoras, además del apareamiento de software dedicado, se hizo posible también simular el sistema de péndulo invertido en tiempo real. A pesar de no brindar todas las particularidades del sistema real, la simulación tiene otras ventajas, pues se elimina el costo de obtención del equipo real, lo cual tiene repercusión en la educación, pues los estudiantes tienen así libertad para probar sus algoritmos de control sin tener que contar con el equipo; y si ya se cuenta con el equipo da mayor holgura para la organización de prácticas de laboratorio. También por cuestiones de seguridad, es deseable tener la posibilidad de probar un programa en una simulación para poder detectar a tiempo errores que podrían derivar en daños tanto en el equipo como en el estudiante.

El desarrollo de las computadoras también trajo otra ventaja consigo: el control digital, pues donde antes se realizaba el control enteramente con componentes electrónicos (control analógico puro), ahora con las computadoras se podía implementar en tiempo real usando programas e interfaces de adquisición de datos (control digital), con las ventajas de que el control pueda ser modular,

adaptable a los requerimientos y relativamente más sencillo de modificar y expandir.

De esta forma el péndulo invertido se ha ido adaptando a los tiempos, y su vigencia se puede comprobar por el número de menciones en libros académicos, tesis y artículos sobre el uso del sistema para comprobar el desempeño de todo tipo de estrategias de control, sean éstas PID, control difuso, neuronal, predictivo, robusto, etc.

Los sistemas donde se aplica el principio del péndulo invertido pueden ser muy complejos como es el caso de: vehículos auto-balanceables de dos ruedas, estabilización de cohetes y embarcaciones, y la reproducción de la caminata humana en robots bípedos.

El ejemplo más notable de un vehículo con auto-balanceo y que ha llegado a ser un producto comercializable es el Segway (Figura 1.3)



FIGURA 1.29. SEGWAY PT [2]

El uso del modelo del péndulo invertido también se ha aplicado al control de vuelo de cohetes y guía de misiles. Dado que la fuerza de empuje en un cohete está por lo general ubicada en su parte inferior y por tanto debajo del centro de masa del cohete, sin un control adecuado de dirección en la base el cohete tenderá a inclinarse por el par que se forma entre el empuje y el peso en lugar de permanecer vertical, se volverá inestable y dará vueltas hasta su eventual caída (Figura 1.4)

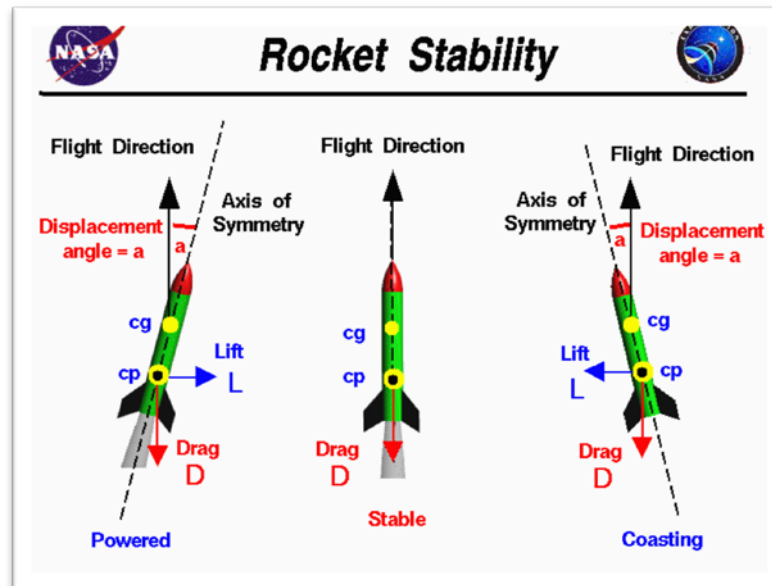


FIGURA 1.30. ESTABILIDAD DE COHETES [3]

Lo anterior se puede aplicar embarcaciones, en especial cuando se desea mantener una trayectoria recta, dado que los propulsores están en la parte trasera de la embarcación, sin un control adecuado de dirección en la parte posterior, la dirección rectilínea será inestable.

En algunos casos el patrón de la caminata humana puede modelarse a partir del concepto del péndulo invertido simple en casos de simplificación máxima o doble péndulo invertido si se requiere mayor detalle. Esto también es aplicable al análisis de cómo logramos mantenernos de pie incluso ante perturbaciones.

En cada pisada del patrón de caminata las extremidades inferiores actúan como péndulo pudiendo el resto del cuerpo ser simplificado como una masa puntual en el extremo y donde el punto de pivote es el tobillo. Aunque la locomoción humana es aún más complicada el análisis ha permitido un entendimiento básico del proceso, al punto de ser replicable en robots bípedos.

En cuanto a los robots bípedos, éstos se remontan a los años sesenta, con los estudios del profesor Ichiro Kato (1925-1994) que dieron como fruto al primer prototipo en 1967, el WL-1, seguido por el WL-3 en 1969 el cual tenía servo-actuadores de tipo electro-hidráulico, ese mismo año otro robot antropomórfico era construido bajo el nombre WAP-1, esta vez usando actuadores neumáticos, y el año siguiente el WAP-2 fue construido, el cual en lugar de actuadores poseía unos músculos artificiales y controlaba su postura gracias a unos sensores de presión colocados en las suelas de las extremidades (Figura 1.5).

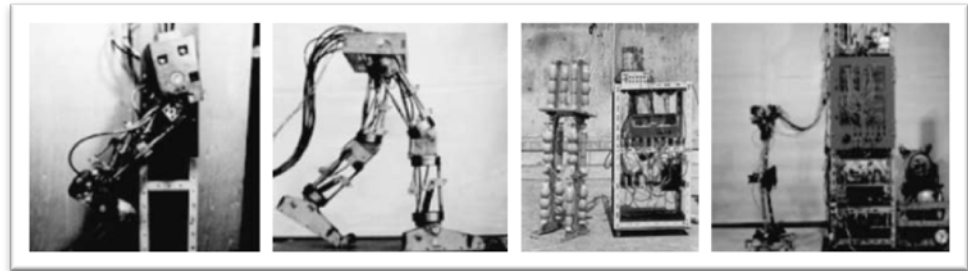


FIGURA 1.31. PRIMEROS ROBOTS BÍPEDOS (DE IZQUIERDA A DERECHA) WL-1, WL-3, WAP-1 Y WAP-2 [4]

Todos estos robots fueron diseñados y construidos en la Universidad Waseda de Japón, donde se siguieron haciendo investigaciones y avances en el área, lanzando más prototipos mejorados en los años siguientes.

1.2. Descripción del equipo

El equipo del péndulo invertido fue en su mayoría diseñado, construido y ensamblado en el Laboratorio de Control y Mecatrónica en conjunto con CAMPRO (Campo Avanzado de Mecanización y Prototipado), ambos de la Facultad de Ingeniería Mecánica y Ciencias de la Producción de la ESPOL.

Dado que este proyecto de graduación está enfocado en el diseño del control, no está documentado el diseño mecánico y

construcción, no obstante se hará a continuación una breve reseña del proceso para obtener el equipo final, para dar un contexto y como punto de partida de la descripción de los aspectos técnicos del equipo.

Las especificaciones para el sistema fueron las siguientes:

- El tipo de péndulo es el de un “carro” o móvil desplazándose sobre una guía horizontal, cuya longitud debe estar entre 1 y 1.5 m.
- Motor eléctrico de corriente continua sin escobillas (tipo BLDC, Brushless Direct Current), con o sin la necesidad de un reductor de velocidad.
- Construcción robusta para que pueda ser usado como equipo demostrativo en el laboratorio.

1.2.1. Descripción técnica de los componentes del equipo

Para facilitar la descripción se separaran los componentes de acuerdo a la función que cumplen, las categorías son las siguientes: parte actuadora, sistema mecánico o planta, e instrumentación.

Parte actuadora

Está conformada por los componentes que hacen posible generar un efecto sobre el sistema de acuerdo a la señal de control, en el caso del péndulo invertido generar el desplazamiento del móvil o carro e indirectamente controlar la posición del péndulo. Sus elementos son: la fuente de poder, el driver, el motor, y la reductora.

- Fuente de poder (Figura 1.6)

Voltaje de salida nominal: 48 Vdc

Potencia: 600 W

Marca: TRC Electronics

Modelo: SE-600-48

Para mayor detalle ver Anexo A



FIGURA 1.32. FUENTE DE PODER.

- Driver (Figura 1.7)

Alimentación (voltaje de entrada): 20-50 Vdc

Tipo de operación: 2 cuadrantes

Controles: velocidad, sentido de giro, parada

Marca: Anaheim Automation

Modelo: MDC150-050301

Más detalles se encuentran en el Anexo B



FIGURA 1.33. DRIVER DEL MOTOR.

- Motor (Figura 1.8)

Tipo: BLDC (Brushless Direct Current) o motor sin escobillas de corriente directa.

Factor de forma: NEMA 34

Potencia nominal: 440 W

Voltaje nominal: 48 Vdc

Torque nominal: 198 oz-in

Torque pico: 600 oz-in

Velocidad Nominal: 3200 RPM

Marca: Anaheim Automation

Modelo: BLY343S-48V-3200

Más detalles están incluidos en el Anexo C



FIGURA 1.34. MOTOR SIN ESCOBILLAS.

- Reductora (Figura 1.9)

Etapas: 1

Relación de reducción: 3:1

Eficiencia: 95%

Marca: Anaheim Automation

Modelo: GBPH-0901-NS-003

Más detalles en el Anexo D



FIGURA 1.35. REDUCTORA 3:1.

Sistema mecánico o planta

Aquí se listan los componentes principales que forman buena parte de la planta o sistema físico, conformado por la polea dentada conductora, polea dentada conducida, la banda dentada, el carro o bloque móvil (conformado a su vez por otros componentes que se describirán más adelante), y el péndulo.

- Polea dentada conductora (Figura 1.10)

Esta polea está montada en el eje de salida (que tiene un agujero roscado M6) de la reductora con la ayuda de un anillo de aluminio, anillo de presión y perno M6

Material: Duraluminio

Número de dientes: 20

Diámetro de paso: 79.48 mm

Diámetro de agujero: 20 mm

Diámetro de anillos guía o bordes (a ambos lados de la polea): 120 mm

Más detalles en el Anexo E

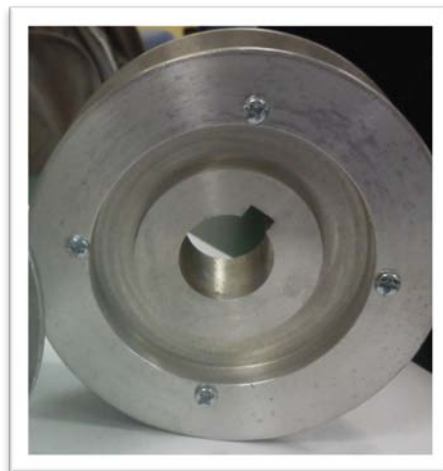


FIGURA 1.36. POLEA CONDUCTORA.

- Polea dentada conducida (Figura 1.11)

Mismas características que la polea dentada conductora, salvo por las dimensiones del chavetero. Está montada en un eje de acero inoxidable a presión y asegurada con anillos de presión a cada lado.



FIGURA 1.37. POLEA CONDUCTIDA, CON EJE Y RODAMIENTOS.

- Banda dentada (Figura 1.12)

Número de dientes: 220

Perfil de diente: trapezoidal tipo H

Paso: ½ pulgada

Perímetro de paso: 110 pulgadas (2.794 metros)

Modelo: 1100H100



FIGURA 1.38. BANDA DENTADA.

- Carro o bloque móvil (Figura 1.13)

Conformado por varias piezas maquinadas en Duraluminio. El cuerpo principal donde van los rodamientos lineales, el agarre de la banda, placa aseguradora, los ángulos para montaje de encoder y de apoyo del eje (ambos con rodamientos), eje de acero inoxidable y acople entre eje y péndulo.



FIGURA 1.39. CARRO O BLOQUE MÓVIL.

- Péndulo (Figura 1.14)

Conformado por un tubo de aluminio y un peso de acero inoxidable

Diámetro exterior del tubo: 9.6 mm.

Longitud total del tubo: 0.45 m.

Masa del tubo: 34.6 g.

Masa del peso: 130 g.



FIGURA 1.40. PÉNDULO CON PESO EN EL EXTREMO LIBRE.

Instrumentación

Conformada por los componentes que se encargan de la medición, medición y comunicación de datos, esencialmente los Encoders para medición de posición y las interfaces o tarjetas de adquisición de datos.

- Encoders (Figura 1.15)

Para el péndulo invertido simple se usan un par de encoders, uno para medir la posición angular del péndulo y el otro para medir la posición lineal del móvil. Ambos tienen las mismas especificaciones técnicas salvo por el diámetro interno del disco para adaptarse a los distintos ejes donde van instalados.

Tipo: Óptico incremental

Resolución: 2500 PPR (Pulses Per Revolution/Pulsos Por Revolución)

Voltaje de alimentación nominal: 5 Vdc

Tipo de salida: dos canales (A y B) de tipo cuadratura TTL

Más detalles técnicos se encuentran en el Anexo F



FIGURA 1.41. ENCODER ÓPTICO INCREMENTAL.

- Tarjetas de adquisición de datos (Figura 1.16)

Tipo: PCI

Número de entradas y salidas en general: 68

Modelo: NI PCI6221 (68 pines)

Fabricante: National Instruments

Más detalles y especificaciones técnicas en el Anexo G



FIGURA 1.42. TARJETA DE ADQUISICIÓN DE DATOS NI PCI-6221.

- Bloques de borneras y cables de conexión (Figura 1.17)

Modelo (Bornera): NI CB-68LP

Terminales: 68 de tipo tornillo

Modelo (Cable): NI SHC68-68-EPM

Fabricante: National Instruments

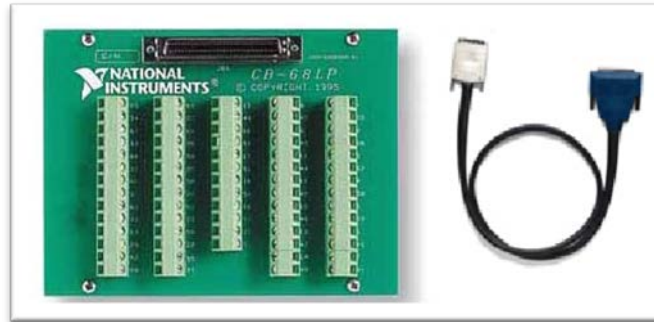
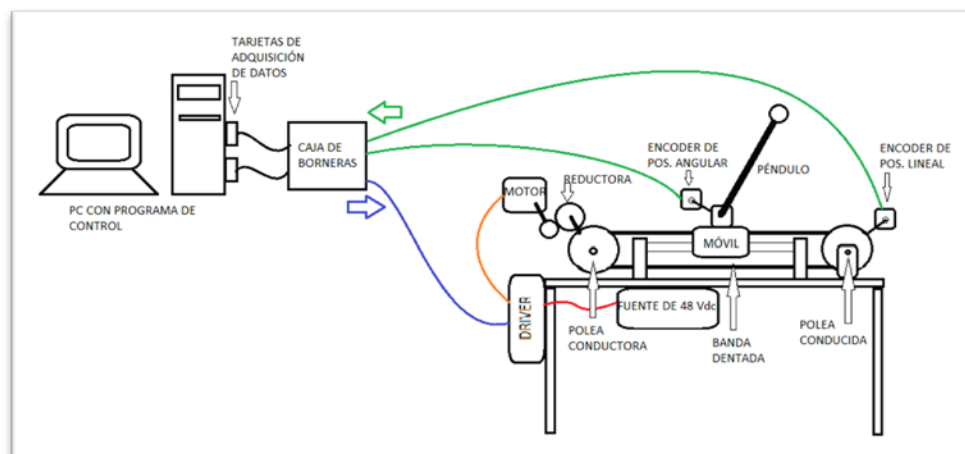


FIGURA 1.43. BORNERA Y CABLE.

1.2.2. Funcionamiento

El equipo o sistema de péndulo invertido es controlado vía software, donde la comunicación entre el programa instalado en la computadora y el equipo es llevada a cabo por las tarjetas de adquisición de datos (Figura 1.18).

En el diagrama las señales de entrada de los sensores de posición están en color verde, las de control en azul, la alimentación al driver en rojo y la alimentación al motor en naranja.



**FIGURA 1.44. DIAGRAMA DEL SISTEMA DE PÉNDULO
INVERTIDO**

El motor no es comandado directamente, su driver (Figura 1.19) sirve de intermediario y es el que recibe las señales de control, y de acuerdo a ello hace actuar al motor.

Para explicar su funcionamiento se puede aplicar una analogía con los motores DC, la velocidad de un motor DC es directamente proporcional al voltaje de alimentación, así que si el voltaje se reduce, también lo hará la velocidad, y viceversa.

Por ejemplo, tomando en cuenta la información de la página del fabricante del motor y driver, si se tiene una fuente de 12 Vdc y se

reduce a 6 Vdc, entonces la velocidad será un 50% más lenta que corriendo a 12 Vdc.

El controlador de velocidad trabaja al variar el voltaje promedio enviado al motor, podría ajustar el voltaje directamente, pero esto es ineficiente. Una manera mucho más eficiente es, interrumpir y pasar el voltaje de alimentación de forma rápida. Cuando ese cambio de estado es lo suficientemente rápido el motor solo reconocerá el efecto promedio. La velocidad promedio del motor incrementa, de acuerdo al incremento de la cantidad de tiempo en que el voltaje pasa comparada con la cantidad de tiempo en que es interrumpida.

Ese cambio de paso e interrupción es ejecutado por un elemento llamado MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor). Los Mosfets son dispositivos que pueden hacer pasar o interrumpir Corrientes muy grandes bajo el control de una señal de bajo nivel de voltaje [5].

El driver es alimentado por los 48 voltios proporcionados por la fuente de poder en las terminales Vin y Gnd.

Las señales de entrada de control principales que maneja el driver son Speed (Velocidad), Run/Stop (Arranque/Parada) y Direction (Dirección).

La señal de entrada Velocidad está dada por un voltaje analógico admisible de 1 a 4 voltios, que controla el ciclo de trabajo del PWM mediante el cual el driver alimenta voltaje a las fases del motor.

La señal de entrada Arranque/Parada es de tipo lógica, es decir que solo admite dos estados lógicos ("1" ó "0") donde "0" está dada por un señal de 0 a 0.8 Vdc y "1" de 4.5 a 5 Vdc. En este caso "1" corresponde a parada y "0" a arranque o habilitación del motor.

La señal de entrada Dirección también es de tipo lógica, donde "1" es giro en el sentido horario y "0" anti-horario, viéndolo desde la parte trasera del motor, en el sistema final esto quiere decir que "1" hacer mover al móvil hacia la izquierda y "0" hacia la derecha.

El driver también tiene como entrada las señales de los sensores Hall del motor (Sensor A, B, y C), pero no cuentan como señales de control pues no son manejables por el usuario y solo sirven como referencia de la posición del rotor y así que el driver sepa qué fases

alimentar de acuerdo al tipo de movimiento que se requiere de acuerdo a las señales de entrada de control. El mismo driver se encarga de alimentar con voltaje a los sensores Hall en los bornes nombrados como Hall Power y Hall Gnd. El driver alimenta con pulsos de 48 Vdc a las tres fases del motor en los bornes de salida (Phase A, B, y C), la alimentación sigue un patrón sincronizado de acuerdo a los requerimientos de las señales de control y las señales de los sensores Hall.

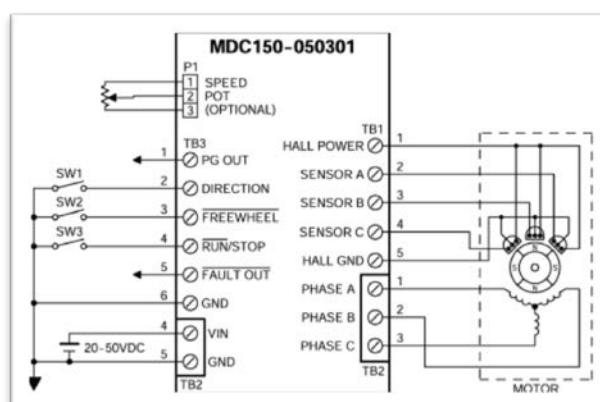


FIGURA 1.45. DIAGRAMA DE CONEXIONES DISPONIBLES EN EL DRIVER DEL MOTOR.

El motor es de tipo BLDCM (Brushless Direct Current Motor), Motor de Corriente Directa sin Escobillas, tiene una velocidad nominal de 3200 RPM, su torque nominal y pico es 1.4 Nm y 4.24 Nm respectivamente.

Una descripción más detallada del funcionamiento del motor DC sin escobillas puede encontrarse en el capítulo 2 correspondiente a la obtención del modelo matemático del sistema.

La salida del motor está conectada la reductora de 3:1, reduciendo por lo tanto la velocidad de salida en una razón de tres y aproximadamente triplicando el torque dada la eficiencia de la reductora que es del 95% (Figura 1.20)

En el eje de salida de la reductora va la polea conductora, la que a su vez mueve a la banda dentada, y ésta última mueve a la polea conducida y al bloque móvil o carro.



FIGURA 1.46. ENSAMBLAJE DE DRIVER, MOTOR, REDUCTORA Y POLEA CONDUCTORA

La polea conducida cumple la función de transmisión de movimiento y ayuda en la medición de la posición del móvil puesto que en su eje va instalado el disco del encoder de posición lineal, y despreciando la elasticidad de la banda se puede inferir que el movimiento de la polea conducida es análogo al de la conductora, y que la posición lineal del móvil se obtiene por simple geometría. Sabiendo el diámetro de paso de las poleas ($D = 79.48 \text{ mm}$), se puede calcular el perímetro de paso de la polea y esa longitud será el avance lineal del carro por cada revolución de la polea (Ec. 1.1).

$$P = \pi D = \pi * 79.48(\text{mm}) = 249.69 \text{ mm} \quad (1.1)$$

El móvil se desliza sobre dos guías conformadas por barras de sección circular de acero con la ayuda de un par de rodamientos lineales. Las guías están sostenidas por un par de soportes en sus extremos, lo cual nos da una carrera total de 1.05 m.

La banda va sujeta al móvil entre el bloque principal inferior y los soportes superiores. En la parte superior del móvil hay dos soportes para el eje del péndulo, en el soporte posterior va montado el cuerpo del encoder de posición angular y el soporte frontal sirve de apoyo extra, ambos tienen rodamientos. A la salida del eje del

péndulo va montado un bloque que sirve de conexión entre el eje y el péndulo. El péndulo va sujeto a este bloque con un perno prisionero (Figura 1.21).



FIGURA 1.47. DETALLE DEL MÓVIL, BANDA, GUÍAS DE DESLIZAMIENTO Y PÉNDULO.

La posición angular del péndulo y la posición lineal del móvil son medidas usando encoders ópticos incrementales. Un encoder es un dispositivo electro-mecánico que convierte desplazamiento lineal o rotacional en señales digitales o de pulso. El más popular es el del tipo óptico, el cual consiste en un disco rotatorio, una fuente de luz, y un foto-detector (sensor de luz). El disco, que está montado en el eje rotatorio, tiene patrones de sectores opacos y transparentes codificados en el disco (Figura 1.22) al rotar el disco, estos patrones

interrumpen la luz emitida hacia el foto-detector, generando una señal de salida digital o de pulso.

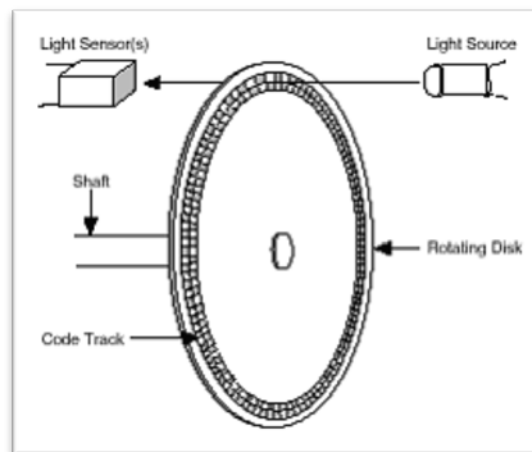


FIGURA 1.48. DIAGRAMA DE UN ENCODER ÓPTICO INCREMENTAL [6].

Siendo más específico, un encoder incremental genera un pulso por cada paso incremental. Aunque el encoder incremental no muestra posición absoluta, provee mayor resolución a un menor precio.

Los encoders incrementales más comunes usan dos canales de salida (A y B) para medir posición. Usando dos pistas de código con los sectores posicionados 90° fuera de fase (Figura 1.23), los dos canales de salida del encoder de cuadratura indican tanto posición como dirección de rotación. Si A lidera a B, por ejemplo, el disco está rotando en dirección horaria y viceversa.

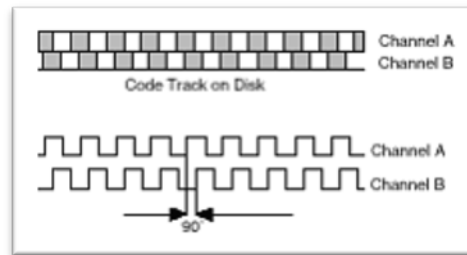


FIGURA 1.49. CANALES DE SALIDA A Y B EN UN ENCODER DE CUADRATURA [6].

Los encoders necesitan de un eje saliente y una superficie perpendicular al eje de rotación para ser instalados, pues se instala primero el disco codificado y luego la carcasa con el emisor-lector empernado a la superficie, la transmisión de datos se dan por un cable de cinco terminales de fácil instalación (Figura 1.24).



FIGURA 1.50. DETALLE DE ENCODERS INSTALADOS PARA MEDICIÓN DE POSICIÓN LINEAL (IZQ.) Y POSICIÓN ANGULAR (DER.)

Todas las señales de entrada y salida viajan por cables hasta una caja que contiene las borneras para fácil conexión con las terminales de las tarjetas de adquisición, las borneras están conectadas a las tarjetas (instaladas en las ranuras PCI de la computadora) mediante el cable especificado en la sección anterior de componentes del equipo (Figura 1.25).



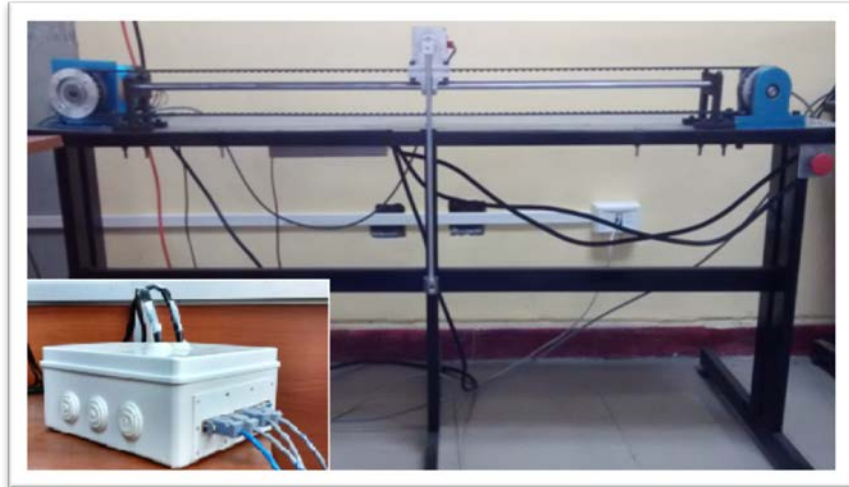
FIGURA 1.51. DETALLE DE LA CAJA DE CONEXIONES, BORNERAS Y TARJETAS DE ADQUISICIÓN DE DATOS

A continuación se presenta una tabla donde se indican las conexiones de las diferentes entradas y salidas.

TABLA 1.1. CONEXIONES ENTRE DISPOSITIVOS Y BORNERA DE TARJETA DE ADQUISICIÓN DE DATOS.

DISPOSITIVO	CONEXIONES/CABLES DEL DISPOSITIVO	NÚMERO DE BORNERA
ENCODER DE POSICIÓN ANGULAR	Canal A	42
	Canal B	46
	Alimentación	8
	Tierra	9
ENCODER DE POSICIÓN LINEAL	Canal A	37
	Canal B	45
	Alimentación	14
	Tierra	7
DRIVER	Voltaje de referencia/velocidad	22
	Señal "Run/Stop"	17
	Señal de control de giro	52
	Tierra	53

El equipo final puede observarse en la figura 1.26, la fuente no es visible pues está instalada debajo de la mesa, por razones de seguridad en la esquina derecha se ha instalado un botón de emergencia que al presionarlo corta la alimentación a la fuente de voltaje.



**FIGURA 1.52. SISTEMA DE PÉNDULO INVERTIDO CON
DETALLE DE CAJA DE CONEXIONES**

1.3. Alcance y Aplicaciones del Proyecto

El proyecto tiene un enfoque principalmente académico, en consecuencia los objetivos a cumplir por el mismo van por esa misma línea, esto es:

- Aplicar los conocimientos de sistemas de control para crear un modelo matemático cuya simulación tenga un comportamiento aceptablemente similar al real (el péndulo invertido).

- Comprobar la teoría de control aprendida aplicándola a un sistema real, diseñando el control en base a la simulación del sistema anteriormente obtenida.
- Estudio de algunas de las aplicaciones reales del control de un péndulo invertido simple ya han sido mencionadas de forma superficial, por ejemplo la caminata bípeda de robots y control de posición y estabilidad de cohetes, misiles y barcos.

CAPÍTULO 2

2. MODELO MATEMÁTICO DEL SISTEMA

Para una mayor facilidad en la obtención del modelo de la planta, se separó al sistema en dos partes principales, la etapa de actuación conformada por el conjunto: driver, motor, reductora, poleas, y banda; y la planta de péndulo invertido en sí o péndulo invertido básico, esto es, el carro o móvil, los rieles o guías de desplazamiento y el péndulo.

Utilizando las expresiones físicas básicas que describen a los sistemas se obtuvieron las ecuaciones con las cuales se obtienen las funciones de transferencia.

2.1. Obtención de ecuaciones diferenciales del sistema

Etapa de actuación

La etapa de actuación está conformada por el driver, el motor, la reductora, las poleas y la banda, donde se comanda un voltaje de entrada para obtener una fuerza que actúa como señal de control sobre el sistema de carro-péndulo invertido.

Fueron necesarias varias asunciones iniciales para hacer más sencillo el análisis de estos componentes. La dinámica del driver, es decir de sus componentes electrónicos y el “backlash” o efecto de pequeñas holguras entre engranajes se consideraron despreciables.

Sobre el funcionamiento de los motores DC sin escobilla, la página del fabricante indica que en términos generales, los motores sin escobillas son motores eléctricos síncronos alimentados por corriente directa. Estos motores operan bajo un sistema de conmutación electrónica de seis pasos. A diferencia de los motores DC con escobillas, éstos no contienen ninguna escobilla de carbón. En su lugar, los electroimanes dentro del motor permanecen

estacionarios junto con la armadura, mientras que los imanes permanentes rotan, generando torque. (Figura 2.1)

La conmutación de seis pasos es una manera costo-efectiva de conmutación electrónica, en este método, solo dos de los tres bobinados del motor son usados a la vez. Los pasos son equivalentes a 60 grados eléctricos, así, los seis pasos forman un rotación completa de 360 grados.

Los motores DC sin escobillas consisten en un rotor giratorio, imanes de Neodimio, y un estator. Son considerados como una versión de “adentro hacia afuera” de un motor DC con escobillas; donde no hay conmutador ni escobillas, y los bobinados están localizados en el exterior, conectados al controlador [7].

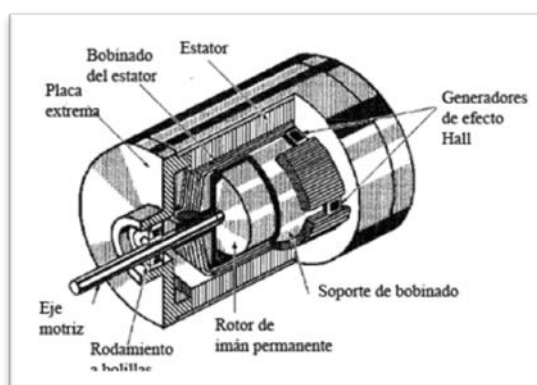
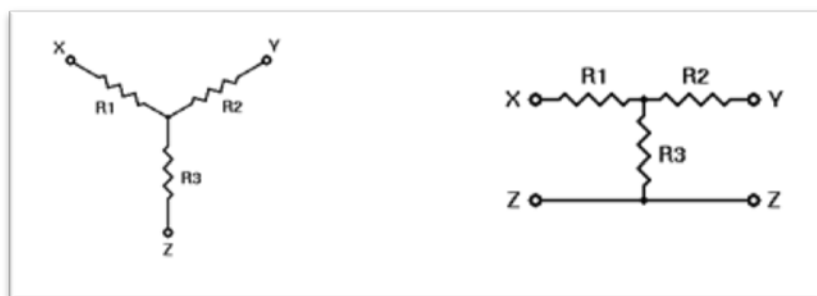


FIGURA 2.1. ESTRUCTURA INTERNA DE UN MOTOR DC SIN ESCOBILLAS

Internamente, los bobinados en un motor de 3 fases puede venir con una de las siguientes configuraciones: “Wye”, o “Delta”.

La principal ventaja de la configuración “Wye” o “Estrella”, es el voltaje de fase a neutro es igual en los tres terminales o ramales. El arreglo es básicamente un circuito paralelo con la forma de la letra Y, donde todos los bobinados están conectados a un punto central, y la energía se aplica a los terminales restantes de los bobinados.

El motor usado en el equipó del péndulo invertido viene con la configuración estrella por defecto y no ha sido cambiada (Figura 2.2).



**FIGURA 2.2. DIAGRAMAS DE CONFIGURACIÓN “WYE” O
“ESTRELLA [7]”**

La configuración Delta por el otro lado, crea una figura triangular, haciéndolo un circuito en serie. En esta configuración se aplica energía en cada conexión o terminales de los bobinados (Figura 2.3).



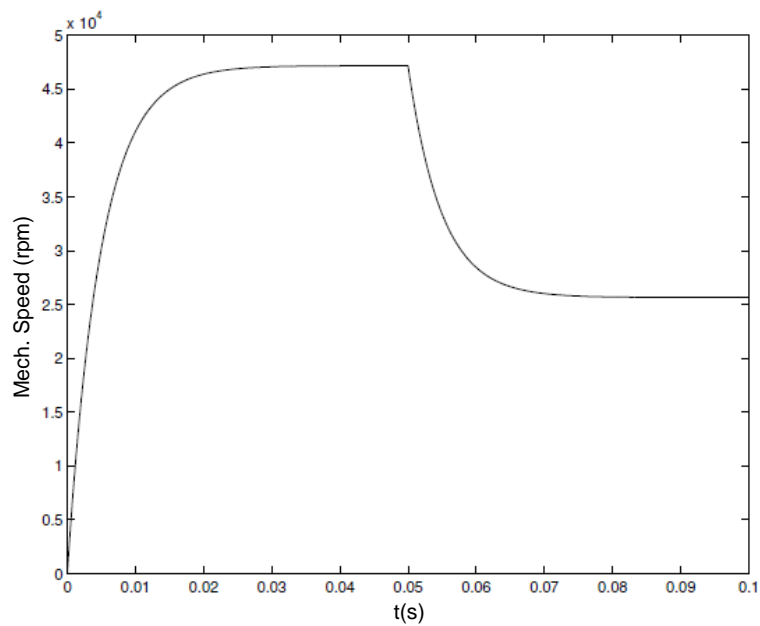
FIGURA 2.3. DIAGRAMAS DE CONFIGURACIÓN DELTA [7]

La complejidad en el desarrollo del modelo matemático de un motor BLDC depende de los objetivos, por ejemplo si se busca una simulación a nivel profundo con miras a analizar el comportamiento y establecer estrategias de control en el driver, conviene un modelo matemático que tome en cuenta las tres fases del motor, las formas trapezoidales de la FCEM, trabajar con las constantes de fase en lugar de línea a línea, etc. Por otro lado, si no se busca o no existe forma de establecer un control al nivel del driver y simplemente se quiere establecer la relación entre el voltaje que ingresa al motor y la velocidad o torque promedio en el tiempo, entonces puede aproximarse el comportamiento del motor BLDC al de su par DC

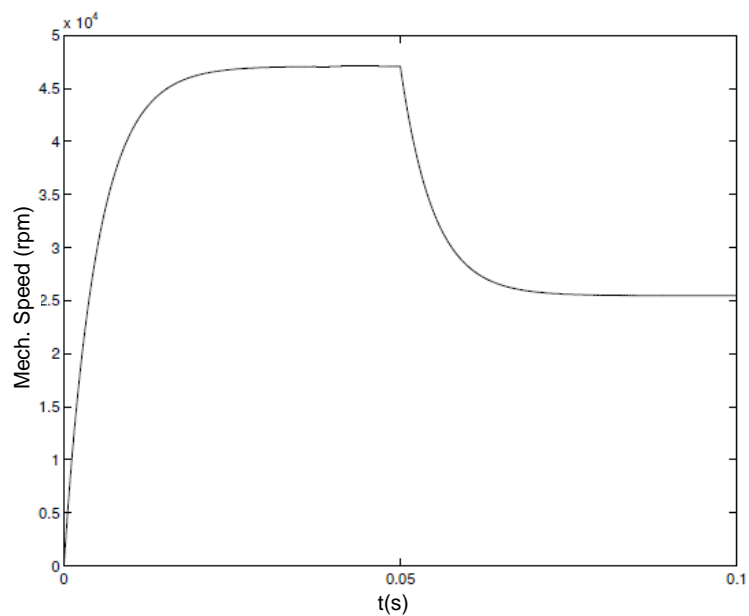
con escobillas, como ha quedado demostrado en el trabajo: “BLDC Motor Modelling and Control – A Matlab®/Simulink® Implementation” de Stefán Baldursson [8].

En dicho trabajo, se derivó un modelo matemático sencillo basado en el motor DC, y uno más complejo basado en un motor BLDC con conexión en estrella incluyendo la conmutación electrónica del driver. Con los datos de la hoja de especificaciones de un motor BLDC comercial, se llevaron a cabo las simulaciones para cada modelo, obteniendo gráficas de velocidad angular-tiempo, torque-tiempo, corriente-tiempo, y torque-velocidad angular. Los perfiles de las curvas fueron prácticamente iguales, salvo que en el caso del modelo del motor BLDC se presenta el fenómeno conocido como “Torque Ripple”, que en resumen es una variación periódica del torque lo cual le da mayor realismo y permite analizar los efectos de estrategias de control al nivel del driver.

A continuación se presentan las comparaciones de las curvas de velocidad angular-tiempo y torque-tiempo para cada modelo:



**FIGURA 2.4. CURVA DE VELOCIDAD ANGULAR VS. TIEMPO
EN SIMULACIÓN DE MODELO DE MOTOR DC [8]**



**FIGURA 2.5. CURVA DE VELOCIDAD ANGULAR VS. TIEMPO
EN SIMULACIÓN DE MODELO DE MOTOR BLDC [8]**

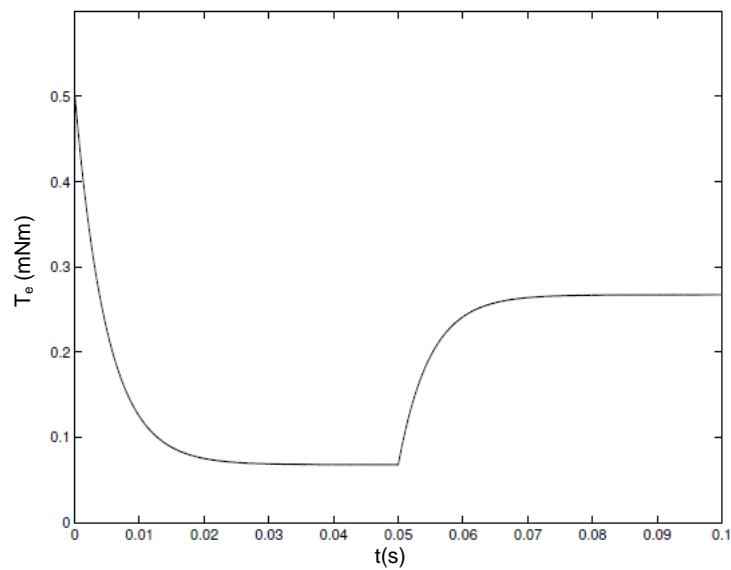


FIGURA 2.6. CURVA DE TORQUE VS. TIEMPO EN SIMULACIÓN DE MODELO DE MOTOR DC [8]

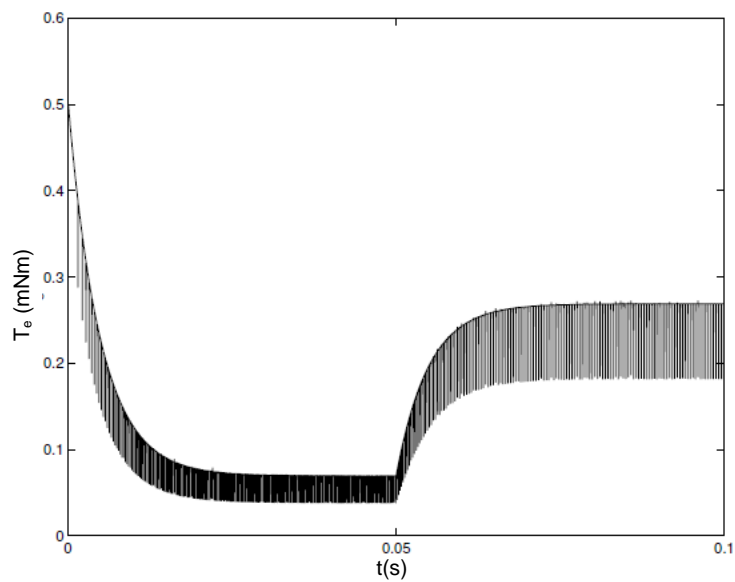


FIGURA 2.7. CURVA DE TORQUE VS. TIEMPO EN SIMULACIÓN DE MODELO DE MOTOR BLDC [8]

Las curvas mostradas en las gráficas 2.4 y 2.5 son casi las mismas y en el caso de las gráficas 2.6 y 2.7 los perfiles de las curvas son bastante semejantes, en especial si se decide despreciar las caídas o variaciones (“Torque Ripple”).

Como se había explicado en párrafos anteriores, tener un modelo meticuloso y detallista sirve si es que se piensa analizar y diseñar el control, modificando parámetros en el driver o la forma en cómo es ingresado el voltaje, como la estrategia usando Modulación de Ancho de Pulso propuesta también en el trabajo referenciado. Sin embargo, ese no fue el caso en el sistema del péndulo invertido, pues solo se podía controlar la magnitud del voltaje en las fases a través del voltaje de referencia, ello sumado a la semejanza de resultados y al hecho de que un modelo menos complejo facilita el análisis y diseño del control, llevó a elegir utilizar el modelo matemático del motor DC para describir el comportamiento del motor BLDC en la etapa de actuación.

Si bien el trabajo anterior usó espacios de estado MIMO para el modelo matemático y simulación, es preferible trabajar con un sistema SISO, puesto que las señales que interesan son el voltaje como entrada y el torque generado por el motor como salida, dicha

relación es representada primero por ecuaciones diferenciales para en el siguiente capítulo obtener una función de transferencia SISO.

Tomando como contexto lo anterior, se necesita hacer una introducción a la teoría del funcionamiento del motor de corriente continua.

Sea el siguiente diagrama representando una versión simplificada del motor DC:

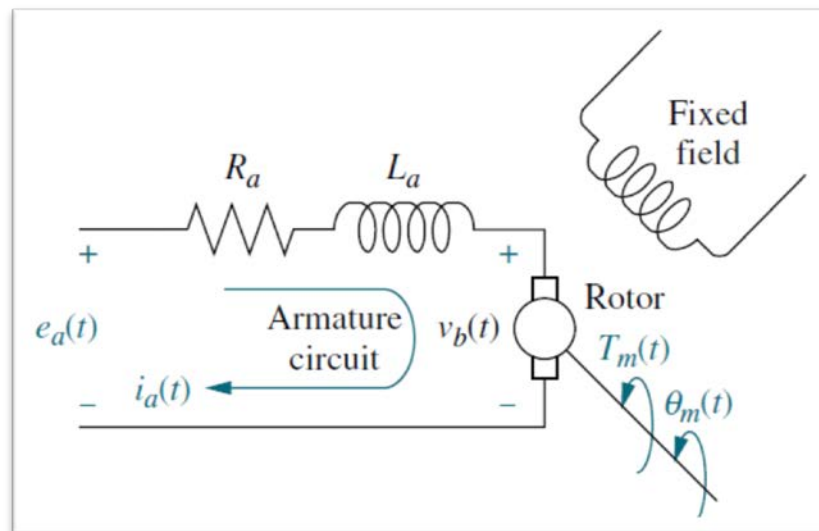


FIGURA 2.8. DIAGRAMA BÁSICO DE UN MOTOR DC [9]

El comportamiento de un motor DC, al ser un sistema electromecánico está regido por ecuaciones eléctricas y mecánicas (ecuaciones 2.1 a 2.4) [10].

$$e_a(t) = L_a \frac{di_a(t)}{dt} + R_a i_a(t) + v_b(t) \quad \text{Ec. 2.1}$$

Dónde: e_a es el voltaje aplicado en los terminales del motor, L_a la inductancia de la armadura, i_a la corriente de armadura, R_a la resistencia de armadura y v_b la FCEM generada.

$$T_m(t) = I_{rotor} \frac{d^2 \theta_m(t)}{dt^2} + T_l(t) + T_f(t) \quad \text{Ec. 2.2}$$

Dónde: T_m es el torque motor generado, I_{rotor} la inercia del rotor, θ_m la posición angular del rotor, T_l el torque de la carga y T_f el torque generado por la fricción.

Los motores DC presentan además un comportamiento aproximadamente lineal en las relaciones entre torque y corriente, y voltaje y velocidad, lo cual se expresa en las siguientes ecuaciones:

$$T_m(t) = k_t i_a(t) \quad \text{Ec. 2.3}$$

$$v_b(t) = k_b \frac{d\theta_m(t)}{dt} \text{ Ec.2.4}$$

Donde K_t es la constante de torque y k_b la constante de FCEM.

Dado que el valor de la inductancia es pequeño, éste puede considerarse despreciable, por lo tanto la ecuación 2.1 queda de la siguiente forma:

$$e_a(t) = R_a i_a(t) + v_b(t) \text{ Ec. 2.5}$$

Para facilitar el trabajo posterior de la obtención de una función de transferencia del motor que relacione el voltaje de entrada con el torque de salida, se introducen las expresiones 2.3 (reordenada) y 2.4 en la 2.5, obteniendo de esta manera la ecuación 2.6.

$$e_a(t) = R_a \frac{T_m(t)}{k_t} + k_b \frac{d\theta_m(t)}{dt} \text{ Ec. 2.6}$$

El torque de fricción está dado por un sencillo modelo de fricción viscosa (ecuación 2.7), simplificación que se hace para no introducir no linealidades como las dadas en el modelo de fricción de Coulomb, donde B_m no solo es la constante de amortiguación

viscosa del rotor, sino una equivalente que incluyen los efectos de fricción viscosa de otros elementos [9]:

$$T_f(t) = B_m \frac{d\theta_m(t)}{dt} \quad \text{Ec. 2.7}$$

De acuerdo al concepto de inercias equivalentes en la armadura observado [9], puede verse que el torque de carga puede ser representado por una suma de inercias equivalentes (I_m), que incluye a las inercias reflejadas en la armadura más la inercia del rotor, multiplicada por la aceleración angular del motor (ecuación 2.8).

$$I_{rotor} \frac{d^2\theta_m(t)}{dt^2} + T_l(t) = I_m \frac{d^2\theta_m(t)}{dt^2} \quad \text{Ec. 2.8}$$

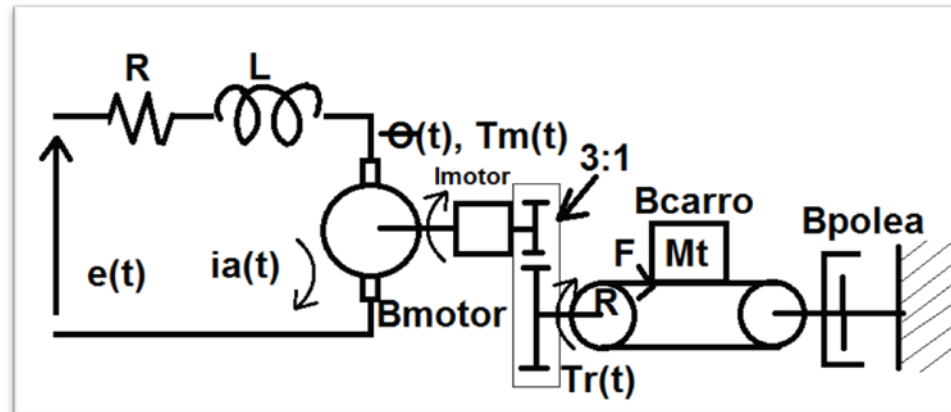
Con las consideraciones anteriores, la ecuación 2.2 queda expresada de la siguiente forma:

$$T_m(t) = I_m \frac{d^2\theta_m(t)}{dt^2} + B_m \frac{d\theta_m(t)}{dt} \quad \text{Ec. 2.9}$$

En la ecuación 2.9 se debe determinar la sumatoria de inercias o inercia equivalente y la sumatoria de amortiguación viscosa o coeficiente de fricción viscosa equivalente.

Como se mencionó en párrafos anteriores, la inercia equivalente es la inercia del rotor más las inercias reflejadas a la armadura, tomando como base la recopilación de cálculos de inercias reflejadas por John Mazurkiewicz en “The Basics of Motion Control – Part 1” [11].

Las inercias de carga están dadas por las inercias de las dos poleas consideradas iguales, más la inercia del sistema carro-péndulo, considerado en este caso como una masa concentrada en un sistema de banda transportadora, y éstas a su vez son disminuidas por la acción de la reductora de razón de reducción N y eficiencia η (ver figura 2.9).



**FIGURA 2.9. DIAGRAMA DE INERCIAS DE CARGA
REFLEJADAS A LA ARMADURA.**

Despreciando la inercia de la reductora y de la banda, se obtiene la siguiente expresión básica para la inercia equivalente:

$$I_m = I_{rotor} + \frac{I_{sist\ c-p} + 2I_{polea}}{N^2\eta} \quad \text{Ec. 2.10}$$

Dónde: $I_{sist\ c-p}$ es la inercia del sistema carro-péndulo, e I_{polea} es la inercia de una polea.

$I_{sist\ c-p}$ está considerada como una carga tangencial o de masa concentrada en una banda transportadora con poleas de radio R. en este caso la masa concentrada es la masa total del sistema m_T que es básicamente la masa del péndulo más la masa del carro; el radio

considerado es el radio de paso de las poleas dentadas R_{paso} (ecuación 2.11)

$$I_{sist\ c-p} = m_T (R_{paso})^2 \quad \text{Ec. 2.11}$$

La inercia de la polea dentada se aproxima al de un cilindro hueco de masa conocida m_{polea} , radio exterior R_{ext} y radio interior R_{int} , mediante la siguiente expresión:

$$I_{polea} = \frac{m_T}{2} (R_{ext}^2 + R_{int}^2) \quad \text{Ec. 2.12}$$

Introduciendo las ecuaciones 2.11 y 2.11 en la ecuación 2.10, se obtiene una expresión más detallada para la inercia equivalente:

$$I_m = I_{rotor} + \frac{m_T (R_{paso})^2 + m_T (R_{ext}^2 + R_{int}^2)}{N^2 \eta} \quad \text{Ec. 2.13}$$

De forma parecida se procede con los coeficientes de fricción viscosa donde se tienen los coeficientes del carro y de una de sus poleas, disminuidas por acción de la reductora, más el coeficiente del rotor. En el caso del coeficiente del carro al ser para movimiento lineal está dado por unidades de Fuerza x Tiempo/longitud, en

Sistema Internacional esto es N-s/m (Newtons-Segundos/metros), lo cual se traduce a coeficiente de tipo Torque x Tiempo (N-m-s) multiplicando por el radio de paso de la polea al cuadrado, la expresión queda así:

$$B_m = B_{rotor} + \frac{B_{carro}R_{paso}^2 + B_{polea}}{N^2\eta} \quad \text{Ec. 2.14}$$

Las expresiones 2.13 y 2.14, como puede verse están dadas por constantes, éstas fueron evaluadas más adelante, una vez obtenida una función de transferencia que relacione el voltaje de entrada con el torque de salida del motor.

El torque generado por el motor es incrementado con la ayuda de la reductora de acuerdo a la ecuación 2.15, donde T_r es el torque resultante generado por la reductora.

$$T_r(t) = T_m(t)N\eta \quad \text{Ec. 2.15}$$

Por último ese torque es transmitido mediante el sistema de bandas y poleas dentadas al conjunto carro-péndulo resultando en la fuerza F aplicada sobre éste (ecuación 2.16).

$$F(t) = \frac{T_r(t)}{R_{paso}} \quad \text{Ec. 2.16}$$

El elemento que proporciona voltaje al motor es el driver que tiene una entrada de voltaje de referencia para controlar la velocidad, en realidad ese voltaje controla el ciclo de trabajo del PWM del voltaje en las fases. La entrada de referencia es de 1 a 4 voltios de acuerdo a la hoja de datos. Al despreciar la dinámica de los componentes electrónicos se puede considerar al driver como un simple amplificador con zona muerta, donde a un voltaje de referencia de 1 le correspondería 0 V a las fases, y a 4 V le correspondería 48 V, análogamente para representar el cambio de sentido de giro se puede decir que se cumple la misma relación en valores negativos, es decir que a un voltaje de referencia de -1 le correspondería 0 V a las fases, y a -4 V le correspondería -48 V, aunque como ya ha sido mencionado el cambio de sentido de giro se produce activando una entrada correspondiente que provoca un cambio el orden de alimentación de las fases, pero para resultados prácticos el efecto es el mismo.

De esta forma, la relación quedó como sigue:

$$e_a = \begin{cases} 16(v_{ref} - 1); v_{ref} = (1, 4] \\ 16(v_{ref} + 1); v_{ref} = [-4, -1) \\ 0; v_{ref} = [-1, 1] \end{cases} \text{ Ec. 2.17}$$

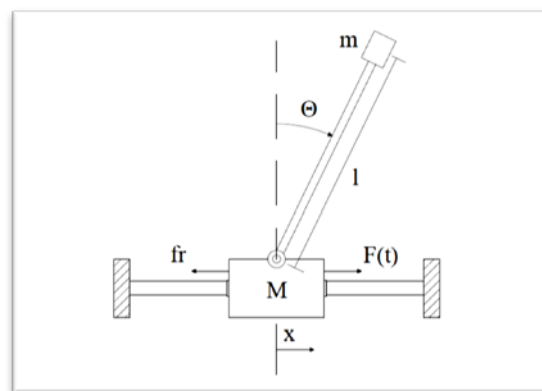
Donde e_a es el voltaje de alimentación a las fases del motor, y v_{ref} es el voltaje de referencia a la entrada del driver.

El problema con esta relación entre el voltaje de referencia recibido por el driver y el que éste alimenta a las fases, es que introducía una no linealidad adicional y mayor a la esperada zona muerta del motor, y la posterior linealización, que es necesaria para el análisis y diseño del control, resultaría bastante imprecisa y alejada de lo que se pretende representar, además de ser dependiente del punto de linealización sobre el cual existe incertidumbre previa; por lo tanto dicha relación se deja fuera del análisis, aunque fue luego aplicada durante la implementación del control de forma inversa.

Planta de péndulo invertido

Como se mencionó al inicio del capítulo, la planta del péndulo invertido se refiere a las partes mecánicas del sistema que no están incluidas en la etapa de actuación.

Sus componentes principales son el carro o móvil de masa “M”, los rieles o guías de desplazamiento, y el péndulo que a su vez tiene una masa “m” en su extremo libre cuyo centro de masa está a una distancia “l” del pivote o eje de rotación.(Fig. 2.10).



**FIGURA 2.10. ESQUEMA SIMPLIFICADO DEL PÉNDULO
INVERTIDO**

Para controlar la posición del péndulo -sea para mantenerlo en su posición vertical inversa o para inyectarle energía y lograr el auto-levantamiento- se tiene una señal de control que varía en el tiempo

$u(t)$ representada en la figura por “ $F(t)$ ” dado que esta señal es la fuerza que se aplica sobre el carro para lograr que se mueva de forma horizontal en el eje x (positivo hacia la derecha), el movimiento del carro tiene un efecto en el péndulo al estar éste último pivoteado en el carro, permitiéndole al péndulo girar libremente, formando un ángulo “ Θ ” con respecto al eje vertical positivo en la dirección horaria, la referencia vertical para el ángulo se encuentra en la parte superior de tal modo que cuando el péndulo cuelga hacia abajo en su posición de equilibrio estable el ángulo sea 180° o π radianes, y cuando está en la posición vertical invertida el ángulo sea 0° o 0 radianes.

El carro en el equipo real se mueve en un par de guías (varillas de acero de sección redonda) usando un par de rodamientos lineales, aunque el propósito de usarlos es disminuir la fricción en el movimiento del carro, ésta no queda eliminada y es tomada en consideración como una fuerza “ f_r ” que se opone a la señal de control (la fuerza aplicada $F(t)$), dicha fricción para propósitos de poder obtener un modelo lineal se considera solo de tipo viscosa y proporcional al producto de un coeficiente de fricción viscosa entre los rodamientos lineales del carro o móvil y las guías en las que se desplaza conocida como “ B_{carro} ” y la velocidad lineal del carro.

Cabe recalcar también las asunciones que se hacen para el péndulo, en el equipo real se tiene un tubo de aluminio de sección redonda y delgada, con un peso acoplado en su extremo libre hecho de acero inoxidable que es aproximadamente más de tres veces el peso del tubo, para simplificar el análisis se tomó al conjunto como un péndulo simple, esto es despreciando la masa y geometría del tubo considerado ahora como un hilo inextensible y sin peso y que tiene suspendida una masa concentrada.

Estabilización

Para la parte de estabilización fue necesario determinar las ecuaciones que rigen el movimiento del sistema, tomando como base la figura 2.10, se desarrollaron los diagramas de cuerpo libre de las dos partes principales del sistema: el carro y el péndulo (ver figuras 2.11 y 2.12).

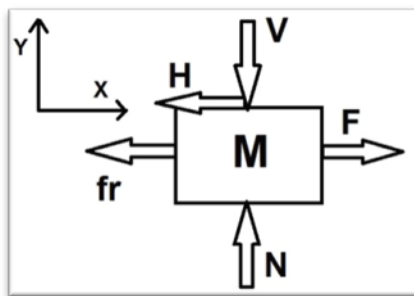


FIGURA 2.11. DIAGRAMA DE CUERPO LIBRE DEL CARRO O MÓVIL

De acuerdo a la figura 2.11, y tomando como referencia que en el eje horizontal o x , la dirección hacia la derecha es la positiva y que para el eje vertical o y la dirección hacia arriba es la positiva, se realizó una sumatoria de fuerzas aplicando de la Segunda Ley de Newton, representada en la ecuación 2.18:

$$\sum \vec{F} = m * \vec{a} \quad \text{Ec. 2.18}$$

Haciendo sumatoria de fuerzas en el eje x para el carro se obtuvo lo siguiente:

$$F - H - fr = M \frac{d^2x}{dt^2} \quad \text{Ec. 2.19}$$

$$F - H - B_{carro} \frac{dx}{dt} = M \frac{d^2x}{dt^2} \quad \text{Ec. 2.20}$$

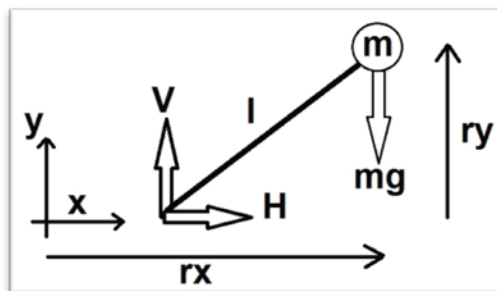


FIGURA 2.12. DIAGRAMA DE CUERPO LIBRE DEL PÉNDULO

De acuerdo a la figura 2.12, se tiene que la distancia en el eje x del centro de masa del péndulo (el centro de masa del peso en el extremo pues se desprecia al cuerpo del péndulo como un péndulo simple ideal) es r_x con la siguiente expresión:

$$r_x = x + l \sin(\theta) \quad \text{Ec. 2.21}$$

La sumatoria de fuerzas en x para el péndulo está dada por:

$$H = m \left(\frac{d^2 r_x}{dt^2} \right) \quad \text{Ec. 2.22}$$

Introduciendo la expresión para r_x :

$$H = m \left(\frac{d^2 (x + l \sin(\theta))}{dt^2} \right) \quad \text{Ec. 2.23}$$

En el eje vertical, la distancia del centro de masa del péndulo está dada por la expresión de r_y :

$$r_y = l \cos(\theta) \quad \text{Ec. 2.24}$$

Realizando la sumatoria de fuerzas en el eje y para el péndulo se tiene:

$$V - mg = m \frac{d^2 r_y}{dt^2} \quad \text{Ec. 2.25}$$

Introduciendo la expresión para r_y :

$$V - mg = m \left(\frac{d^2 (l \cos(\theta))}{dt^2} \right) \quad \text{Ec. 2.26}$$

La Segunda Ley de Newton para movimiento rotacional está descrita por la ecuación 2.27:

$$\sum \vec{M} = I * \vec{\alpha} \quad \text{Ec. 2.27}$$

Aplicando sumatoria de momentos alrededor del centro de masa y tomando la dirección horaria como positiva se obtuvo:

$$V(l \sin(\theta)) - H(l \cos(\theta)) = I\alpha \quad \text{Ec. 2.28}$$

$$V \sin(\theta) - H \cos(\theta) = \frac{I\alpha}{l} \quad \text{Ec. 2.29}$$

De esta forma se obtuvo un conjunto de cuatro ecuaciones (2.20, 2.23, 2.26, y 2.29) que describen el comportamiento del péndulo invertido básico, sin embargo las tres últimas ecuaciones son no lineales puesto que contienen las funciones trigonométricas seno y coseno, y dado que el controlador a diseñar estaba en el campo del control lineal se necesitó linealizar las ecuaciones alrededor de un punto de operación.

Considerando que esta parte del análisis corresponde a la estabilización donde lo que interesa en primer lugar es que el ángulo del péndulo se mantenga lo más cercano a 0, es razonable que el punto de operación alrededor del cual se deben linealizar las ecuaciones sea $\theta=0$ radianes.

Como consecuencia de la linealización se tienen las siguientes identidades:

$$\theta \approx 0 \quad \text{Ec. 2.30}$$

$$\cos(\theta) \approx 1 \quad \text{Ec. 2.31}$$

$$\text{sen}(\theta) \approx \theta \quad \text{Ec. 2.32}$$

Para simplificar el proceso del desarrollo de las ecuaciones, se desarrollaron aparte todas las posibles derivadas y dobles derivadas presentes, acompañadas del término que sirve como denominación de éstas.

$$\frac{dx}{dt} = \dot{x} \quad \text{Ec. 2.33}$$

$$\frac{d^2x}{dt^2} = \frac{d}{dt} \left(\frac{dx}{dt} \right) = \frac{d}{dt} (\dot{x}) = \ddot{x} \quad \text{Ec. 2.34}$$

$$\frac{d\theta}{dt} = \dot{\theta} \quad \text{Ec. 2.35}$$

$$\frac{d^2\theta}{dt^2} = \frac{d}{dt} \left(\frac{d\theta}{dt} \right) = \frac{d}{dt} (\dot{\theta}) = \ddot{\theta} \quad \text{Ec. 2.36}$$

La ecuación 2.20 no necesita ser linealizada pero se pueden aplicar las expresiones 2.33 y 2.34, de esta forma se obtiene:

$$F = M\ddot{x} + B_{\text{carro}}\dot{x} + H \quad \text{Ec. 2.37}$$

Se linealizó la ecuación 2.23 aplicando la identidad 2.32:

$$H = m \left(\frac{d^2(x+l\theta)}{dt^2} \right) \quad \text{Ec. 2.38}$$

Al desarrollar la ecuación 2.38 aplicando las expresiones 2.34 y 2.35 se obtuvo:

$$H = m \left(\frac{d^2x}{dt^2} + \frac{d^2l\theta}{dt^2} \right) \quad \text{Ec. 2.39}$$

$$H = m \left(\frac{d^2x}{dt^2} + l \frac{d^2\theta}{dt^2} \right) \quad \text{Ec. 2.40}$$

$$H = m(\ddot{x} + l\ddot{\theta}) \quad \text{Ec. 2.41}$$

Al linealizar la ecuación 2.26 aplicando la identidad 2.31 se obtuvo:

$$V - mg = m \left(\frac{d^2(l)}{dt^2} \right) \quad \text{Ec. 2.42}$$

Desarrollando la ecuación 2.42 se obtuvo lo siguiente:

$$V - mg = ml \left(\frac{d^2(\theta)}{dt^2} \right) \quad \text{Ec. 2.43}$$

$$V - mg = ml(0) \quad \text{Ec. 2.44}$$

$$V = mg \quad \text{Ec. 2.45}$$

Por último se linealizó la ecuación 2.29 haciendo uso de las identidades 2.31 y 2.32:

$$V\theta - H = \frac{I\alpha}{l} \quad \text{Ec. 2.46}$$

La aceleración angular se puede expresar como la segunda derivada del ángulo, y por tanto usar la expresión 2.36:

$$V\theta - H = \frac{l\ddot{\theta}}{l} \quad \text{Ec. 2.47}$$

Las cuatro ecuaciones diferenciales lineales obtenidas que describen el movimiento del péndulo invertido básico (2.37, 2.41, 2.45, y 2.47), son luego relacionadas para reducirlas a dos.

Introduciendo la ecuación 2.41 en la 2.37 se obtuvo:

$$F = M\ddot{x} + b\dot{x} + m(\ddot{x} + l\ddot{\theta}) \quad \text{Ec. 2.48}$$

$$F = (M + m)\ddot{x} + B_{carrro}\dot{x} + ml\ddot{\theta} \quad \text{Ec. 2.49}$$

Introduciendo las ecuaciones 2.41 y 2.45 en la 2.47 se obtuvo:

$$mg\theta - (m(\ddot{x} + l\ddot{\theta})) = \frac{l\ddot{\theta}}{l} \quad \text{Ec. 2.50}$$

$$mg\theta - m\ddot{x} - ml\ddot{\theta} = \frac{l\ddot{\theta}}{l} \quad \text{Ec. 2.51}$$

Reordenando se obtuvieron las dos ecuaciones simplificadas y linealizadas que rigen el movimiento del péndulo invertido básico:

$$F = (M + m)\ddot{x} + B_{carro}\dot{x} + ml\ddot{\theta} \quad \text{Ec. 2.52}$$

$$(I + ml^2)\ddot{\theta} - mgl\theta = -m\ddot{x}l \quad \text{Ec. 2.53}$$

Auto-levantamiento

El auto-levantamiento presentó varios retos, para diseñar el controlador de esta etapa se necesitaba conocer el modelo que rige el comportamiento del sistema, sin embargo, a diferencia del problema de la estabilización, aquí no era posible linealizar alrededor de un punto de operación, puesto que el péndulo debía balancearse moviendo el carro para sacarlo de su estado de equilibrio estable. El rango de movimiento es amplio, pues se parte de un ángulo teta igual a pi y el péndulo debe balancearse hasta llegar a un ángulo cercano a cero.

Como consecuencia, en esta sección se trabajó con el modelo no lineal del sistema, el mismo que está descrito por las ecuaciones 2.20, 2.23, 2.26, y 2.29. Luego de resolver las dobles derivadas y reubicar términos, se obtuvo el siguiente sistema de ecuaciones:

$$\ddot{x} = \frac{1}{M}(F - H - B_{carro}\dot{x}) \quad \text{Ec. 2.54}$$

$$\ddot{\theta} = \frac{l}{I} (V * \text{Sen}\theta - H * \text{Cos}\theta) \quad \text{Ec. 2.55}$$

$$H = m(\ddot{x} - l\dot{\theta}^2 \text{Sen}\theta + l\ddot{\theta} \text{Cos}\theta) \quad \text{Ec. 2.56}$$

$$V = m[g - l(\ddot{\theta} \text{Sen}\theta + \dot{\theta}^2 \text{Cos}\theta)] \quad \text{Ec. 2.57}$$

2.2. Obtención de funciones de transferencia del sistema

A partir de las ecuaciones lineales obtenidas para la actuación y estabilización del sistema de péndulo invertido en parte de la sección 2.1 y que rigen la dinámica del sistema (ecuaciones 2.6, 2.9, 2.15, 2.16, 2.52 y 2.53), se procedió a obtener las funciones de transferencia del sistema haciendo uso de la Transformada de Laplace, (Tabla 2.1).

TABLA 2.1. TABLA DE TEOREMAS DE TRANSFORMADA DE LAPLACE [12].

Propiedad	Señal	Transformada
	$f(t)$	$F(s)$
	$g(t)$	$G(s)$
Linealidad	$Af(t) + Bg(t)$	$AF(s) + BG(s)$
Diferenciación en t	$f'(t)$	$sF(s) - f(0)$
Diferenciación en t	$f^{(n)}(t)$	$s^n F(s) - s^{n-1}f(0) - s^{n-2}f'(0)\dots - f^{(n-1)}(0)$
Desplazamiento en s	$e^{at} f(t)$	$F(s - a)$
Diferenciación en s	$t^n f(t)$	$(-1)^n F^{(n)}(s)$
Escalamiento en t	$f(at)$	$(1/a)F(s/a)$
	$\begin{cases} f(t - a) & \text{si } t > a \\ 0 & \text{si } t < a \end{cases}$	$e^{-as} F(s)$
Convolución	$f(t)$ periódica (P)	$\frac{\int_0^P e^{-st} f(t) dt}{1 - e^{-sP}}$
	$f(t) * g(t)$	$F(s) G(s)$
Integración en t	$\int_0^t f(u) du$	$F(s)/s$

Se reescribieron y agruparon las ecuaciones mencionadas en el párrafo anterior:

$$e_a(t) = R_a \frac{T_m(t)}{k_t} + k_b \frac{d\theta_m(t)}{dt} \quad \text{Ec. 2.58}$$

$$T_m(t) = I_m \frac{d^2\theta_m(t)}{dt^2} + B_m \frac{d\theta_m(t)}{dt} \quad \text{Ec. 2.59}$$

$$T_r(t) = T_m(t) N \eta \quad \text{Ec. 2.60}$$

$$F(t) = \frac{T_r(t)}{R_{paso}} \quad \text{Ec. 2.61}$$

$$F = (M + m)\ddot{x} + B_{carro}\dot{x} + ml\ddot{\theta} \quad \text{Ec. 2.62}$$

$$(I + ml^2)\ddot{\theta} - mgl\theta = -ml\ddot{x} \quad \text{Ec. 2.63}$$

Tomando la Transformada de Laplace de las ecuaciones anteriores se obtuvo:

$$E_a(S) = R_a \frac{T_m(S)}{k_t} + k_b S \theta_m(S) \quad \text{Ec. 2.64}$$

$$T_m(S) = I_m S^2 \theta_m(S) + B_m S \theta_m(S) \quad \text{Ec. 2.65}$$

$$T_r(S) = T_m(S) N \eta \quad \text{Ec. 2.66}$$

$$F(S) = \frac{T_r(S)}{R_{paso}} \quad \text{Ec. 2.67}$$

$$F(S) = (M + m)X(S)S^2 + B_{carro}X(S)S + ml\theta(S)S^2 \quad \text{Ec. 2.68}$$

$$(I + ml^2)\theta(S)S^2 - mgl\theta(S) = -mlX(S)S^2 \quad \text{Ec. 2.69}$$

Primero se obtuvo una función de transferencia para el motor, donde como entrada se tiene el voltaje alimentado a las fases E_a y como salida el torque del motor T_m .

Para ello se reordena la ecuación 2.65 obteniendo una expresión para $\theta_m(S)$:

$$\theta_m(S) = \frac{T_m(S)}{I_m S^2 + B_m S} \quad \text{Ec. 2.70}$$

Introduciendo la expresión anterior en la ecuación 2.64 se obtuvo:

$$R_a \frac{T_m(S)}{k_t} + k_b S \frac{T_m(S)}{I_m S^2 + B_m S} = E_a(S) \quad \text{Ec. 2.71}$$

Reagrupando, resolviendo y reordenando términos se obtuvo la función de transferencia del motor cuya expresión es la siguiente:

$$T_{motor}(S) = \frac{T_m(S)}{E_a(S)} = \frac{k_t I_m S + k_t B_m}{R_a I_m S + (B_m R_a + k_b k_t)} \quad \text{Ec. 2.72}$$

De la ecuación 2.66 se obtuvo la función de transferencia de la reductora:

$$T_{red}(S) = \frac{T_r(S)}{T_m(S)} = N\eta \quad \text{Ec. 2.73}$$

De la ecuación 2.67 se obtuvo la función de transferencia del sistema de poleas y banda dentada:

$$F_{p-b}(S) = \frac{F(S)}{T_r(S)} = \frac{1}{R_{paso}} \quad \text{Ec. 2.74}$$

Para obtener las funciones de transferencia del sistema o conjunto carro-péndulo invertido, fue necesario reordenar las ecuaciones 2.68 y 2.69.

El sistema del péndulo invertido es del tipo SIMO (Single Input, Multiple Output) donde se tiene una sola entrada que es la señal de control, que es la fuerza si se desprecia la etapa de actuación (driver y moto-reductor), y dos salidas a controlar: el ángulo del péndulo y la posición lineal del carro. Por lo tanto se necesitaron dos funciones de transferencia donde se relacione la señal de control común con cada una de las salidas mencionadas así: Angulo vs. Fuerza y Posición Lineal vs. Fuerza.

De la ecuación 2.69, se reordenaron los términos de tal forma que $X(S)$ quede en un solo lado de la expresión:

$$X(S) = \left(\frac{mgl - (I + ml^2)S^2}{mIS^2} \right) \theta(S) \quad \text{Ec. 2.75}$$

La ecuación 2.75 se introdujo en la ecuación 2.68, de tal modo que ésta última quede en función de $\theta(S)$

$$F(S) = (M + m) \left(\frac{mgl - (I + ml^2)S^2}{mlS^2} \right) \theta(S)S^2 + B_{carro} \left(\frac{mgl - (I + ml^2)S^2}{mlS^2} \right) \theta(S)S + ml\theta(S)S^2 \quad \text{Ec. 2.76}$$

Reordenando los términos de la ecuación 2.76 para relacionar la entrada $F(S)$ con $\theta(S)$ se obtuvo:

$$\frac{\theta(S)}{F(S)} = \frac{mlS^2}{[(ml)^2 - (M+m)(I+ml^2)]S^4 - B_{carro}(I+ml^2)S^3 + (M+m)mglS^2 + B_{carro}mglS}$$

Ec. 2.77

En la ecuación 2.77, puesto que hay tanto un polo como un cero en el origen, éstos se cancelan; y para dejar a S^4 con coeficiente uno se reagruparon términos obteniendo de esta forma la función de transferencia de la posición angular del péndulo (ecuación 2.78).

$$P_{pend}(S) = \frac{\theta(S)}{F(S)} = \frac{\frac{mlS}{q}}{S^3 - \frac{B_{carro}(I+ml^2)}{q}S^2 + \frac{(M+m)mgl}{q}S + \frac{B_{carro}mgl}{q}} \left[\frac{rad}{N} \right] \quad \text{Ec. 2.78}$$

Dónde:

$$q = [(ml)^2 - (M + m)(I + ml^2)] \quad \text{Ec. 2.79}$$

De forma similar se obtuvo la función de transferencia que relaciona la entrada de fuerza versus la salida de posición lineal del móvil. Para ello se reagruparon términos de la ecuación 2.69 como se muestra a continuación:

$$\theta(S) = \frac{mIX(S)S^2}{mgl - (I+ml^2)S^2} \quad \text{Ec. 2.80}$$

La expresión anterior se introdujo en la ecuación 2.68 obteniendo:

$$F(S) = (M + m)X(S)S^2 + B_{carro}X(S)S + ml \left[\frac{mIX(S)S^2}{mgl - (I+ml^2)S^2} \right] S^2$$

Ec. 2.81

Reordenando los términos de la ecuación 2.81 para relacionar la entrada $F(S)$ con $X(S)$ se obtuvo:

$$\frac{X(S)}{F(S)} = \frac{-(I+ml^2)S^2 + mgl}{[(ml)^2 - (M+m)(I+ml^2)]S^4 - B_{carro}(I+ml^2)S^3 + (M+m)mglS^2 + B_{carro}mglS}$$

Ec. 2.82

Reagrupando términos para para dejar a S^4 con coeficiente uno se obtuvo la función de transferencia de la posición lineal del móvil

(ecuación 2.83), donde la expresión de q es la misma de la ecuación 2.79.

$$P_{carro}(S) = \frac{X(S)}{F(S)} = \frac{\frac{-(1+ml^2)S^2+mgI}{q}}{S^4 - \frac{B_{carro}(1+ml^2)}{q}S^3 + \frac{(M+m)mgI}{q}S^2 + \frac{B_{carro}mgI}{q}S} \left[\frac{m}{N} \right] \text{ Ec. 2.83}$$

De esta forma, las funciones de transferencia finales obtenidas fueron las ecuaciones 2.72, 2.73, 2.74, 2.78, y 2.83 (con éstas dos últimas compartiendo la expresión auxiliar 2.79).

En cuanto a la etapa de auto levantamiento, dado que se trabajó con ecuaciones no lineales, no era posible obtener funciones de transferencia, y la resolución de éstas se trabajó mediante simulación en el siguiente capítulo.

CAPITULO 3

3. SIMULACIÓN DEL MODELO MATEMÁTICO

En este capítulo se detallan los pasos para la simulación del modelo matemático, la cual se llevó a cabo en dos etapas. La primera fue una representación usando una plataforma matemática de programación usando programación lineal y por comandos. La segunda se realizó en una plataforma gráfica de simulación, donde la programación se hizo insertando bloques con funciones específicas e interconectándolos.

3.1 Representación del sistema en una plataforma matemática de programación

Para esta sección se necesitó de un programa que permita introducir y manipular expresiones matemáticas. Dado su amplio uso en la academia, ingeniería e investigación además de su disponibilidad en la universidad, se eligió trabajar con Matlab, una plataforma matemática de programación.

3.1.1 Breve introducción a la plataforma matemática de programación

MATLAB® es un lenguaje de alto nivel y entorno interactivo usado por millones de ingenieros y científicos alrededor del mundo. Permite explorar y visualizar ideas y colaborar en todas las disciplinas incluyendo procesamiento de señales e imágenes, comunicaciones, sistemas de control, y finanzas computacionales [13].

El entorno de trabajo de Matlab, se puede clasificar en cinco partes, la barra de pestañas de funciones (1), el árbol de trabajo donde se ve la estructura de carpetas y archivos (2), el espacio de trabajo donde se almacenan los datos generados por los programas ejecutados (3), la ventana de edición donde se escribe el programa (4), y la ventana de comandos donde se muestran resultados y también se pueden escribir comandos de ejecución directa (5).

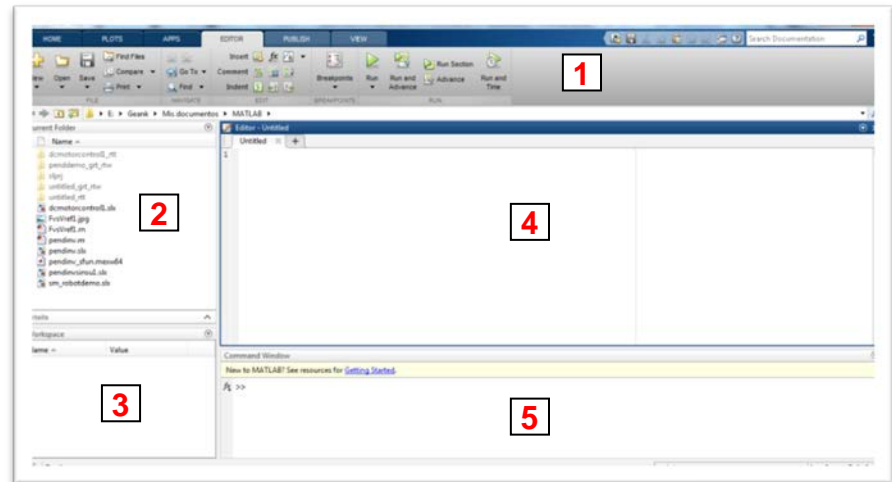


FIGURA 3.1. ENTORNO O ESPACIO DE TRABAJO DE TRABAJO DE MATLAB

La programación se realiza escribiendo líneas de comandos que se ejecutan por lo general de forma secuencial, se pueden añadir comentarios que ayudan a la comprensión de la estructura del programa, los comentarios son precedidos por el símbolo %, aparecen en color verde y son ignorados en la ejecución del programa.

Diversos tutoriales en el uso del programa pueden ser encontrados en el Internet, con diferentes niveles de detalles y dificultad, desde lo más básico e introductorio a lo más avanzado y específico. Sin embargo, para el propósito general del capítulo y de este subtema solo interesan temas básicos como estructura del “script” o programa, declaración de constantes y variables, uso de operaciones aritméticas, funciones matemáticas, definición de polinomios y de funciones de transferencia [14] y [15].

Por lo general, la estructura está formada por un título y descripción breve del programa en forma de comentario, seguido de la declaración de constantes y variables, luego los cálculos o procesamiento de datos, y finalmente la presentación o visualización.

La declaración de constantes se lleva a cabo escribiendo la letra o nombre con el cual se identificará ésta, seguida del signo igual “=”, y el valor que se le quiere asignar; por ejemplo para definir el valor de la gravedad se escribiría: $g = 9.8$

Las variables pueden declararse primero como las constantes para que tengan un valor inicial, pero más adelante en el programa se le asigna la expresión con el cual se calculará su nuevo valor, dicha expresión por lo general hace uso de operaciones aritméticas y funciones trigonométricas básicas como las de las tablas 3.1 y 3.2.

TABLA 3.1. OPERACIONES ARITMÉTICAS EN MATLAB.

Suma	+	$6+2=8$
Diferencia	-	$6-2=4$
Multiplicación	*	$6*2=12$
División	/	$6/2=3$
Elevado a la potencia	^	$6^2=36$

TABLA 3.2. FUNCIONES TRIGONOMÉTRICAS BÁSICAS EN MATLAB.

sin(x) sind(x)	Calcula el seno de x, x en radianes (o grados)
cos(x) cosd(x)	Calcula el coseno de x, x en radianes (o grados)
tan(x) tand(x)	Calcula la tangente de x, x en radianes (o grados)

Para poder definir las funciones de transferencia [15], éstas son manejadas mediante polinomios. Un polinomio se presenta mediante un vector que contiene los coeficientes del polinomio, donde el primer componente es el coeficiente de mayor potencia de s , y el último es el coeficiente de orden 0.

Por ejemplo la expresión: $S^2 + 3S + 5$, se representa por: $p = [1 \ 3 \ 5]$.

Los polinomios del numerador y del denominador de la función de transferencia se mantienen separados. Así dada una función de transferencia $G(S) = 1/S^2 + 3S + 5$ se definen dos polinomios: **num = [1]** y **den = [1 3 5]**.

Existen varias formas de definir una función de transferencia en Matlab, una de ellas es usando la forma **g=tf(num,den)** una vez que se han declarado los polinomios del numerador y denominador como se describe en los párrafos anteriores.

Sin embargo, la usada esta ocasión es aquella que utiliza la declaración **s=tf('s')**, luego se define la función de transferencia escribiendo directamente sus polinomios en función de S, usando como ejemplo la función definida en el párrafo anterior se ingresaría directamente: $G=1/(S^2 + 3S + 5)$, después de haber escrito la instrucción **s=tf('s')**.

Otras instrucciones que sirven para trabajar con funciones de transferencia en Matlab son: la declaración de entradas y salidas, configuraciones de propiedades de objetos y visualización de la respuesta en el tiempo del sistema a diferentes tipos de entradas típicas.

Se declaran las entradas y salidas de forma similar a las variables o constantes pero con la adición de corchetes y comillas simples, así: **input = {'Fuerza'}**.

Para configurar propiedades de objetos, se utiliza el comando **set**, con el formato:
set(H,'PropertyName',PropertyValue,...)

Para visualizar la respuesta en el tiempo del sistema definido por las funciones de transferencia, existen comandos dedicados como el **step** y el **impulse**, con los siguientes formatos: **impulse(sys,t)** y **step(sys,t)**. Donde **sys** es el sistema definido por sus funciones de transferencia y **t** es el intervalo de tiempo, el cual se define de acuerdo a la siguiente instrucción: $t=t_{in}:paso:t_{final}$, donde t_{in} es el tiempo inicial, t_{final} el tiempo final, y **paso** los incrementos en los que el tiempo se divide.

Otros usos de instrucciones y comandos se irán delineando en el transcurso del documento según sea necesario.

3.1.2 Elaboración del programa

Para propósitos de orden, y facilidad en la determinación y corrección de errores, se siguió una estructura en la elaboración del programa, usando comentarios para darle un título y una muy breve descripción general, seguido de la declaración de constantes y variables y por último la definición de las funciones de transferencia.

Las constantes básicas que deben declararse aparecen en la tabla 3.3 además de las conversiones de unidades y otras constantes calculadas a partir de las anteriores como la inercia y coeficiente de fricción viscosa equivalentes, inercia del péndulo, y expresión auxiliar "q".

En el caso de los coeficientes de fricción viscosa se utilizaron aproximaciones teóricas.

**TABLA 3.3. PARÁMETROS FÍSICOS DEL SISTEMA DE
PÉNDULO INVERTIDO.**

PARÁMETRO	VALOR
R_a	0.18 Ω
K_e	9.31 V/KRPM
K_t	17 oz-pulg/A
I_{rotor}	0.02266 oz-pulg-s ²
R_{paso}	0.03974 m
$R_{exterior}$	0.039 m
$R_{interior}$	0.01 m
m_{polea}	0.34 Kg
N	3
η	0.95
B_{rotor}	0.001 N-m-s
B_{carro}	0.005 N-s/m
B_{polea}	0.01 N-m-s
M	1 Kg
m	0.1646 Kg
m_{total}	1.1646 Kg
l	0.45 m

Adicionalmente, es útil disponer de los coeficientes calculados del numerador y denominador de las funciones de transferencia, para esto se utiliza el comando **tfdata** que extrae y almacena los coeficientes en unas variables asignadas, la sintaxis es como sigue:

```
[num,den]=tfdata(FDT,'v');
```

Donde **num** y **den** son los nombres que uno asigna a donde se desea almacenar los vectores y **FDT** es el nombre de la función de transferencia de la cual se desean extraer el numerador y denominador. Se escribió esta línea de comandos luego de cada definición de función de transferencia en el programa.

Excepciones para lo anterior fueron las funciones de transferencia de la reductora y del sistema de polea y banda, dado que éstas prácticamente eran ganancias. Adicionalmente, para poder analizar la respuesta del sistema, se necesitaban únicamente dos funciones de transferencia que relacione directamente la entrada de voltaje en las fases con la posición lineal del carro por un lado, y la posición angular del péndulo en el otro.

Si se observan las entradas y salidas de las funciones de transferencia puede verse que éstas están relacionadas de forma serial, la del motor recibe voltaje de fase y entrega torque de motor, la de la reductora recibe el torque de motor anterior y entrega torque de salida de la reductora, la del sistema polea-banda, recibe la señal anterior de torque de reductora y entrega fuerza, por último esta fuerza la reciben las funciones de transferencia de posición del carro y posición del péndulo entregando respectivamente los datos de posición lineal del carro y la posición angular del péndulo.

Las funciones de transferencia finales se obtienen de multiplicar las correspondientes a los procesos en serie, nombrándolas **P_carro2** y **P_pend2** respectivamente.

Cabe recalcar que dado que la simulación del sistema para el auto-levantamiento estaba expresada por ecuaciones no lineales, éstas fueron tratadas en la siguiente sección donde se trabajó con la plataforma gráfica de simulación.

Tomando como referencia el desarrollo de la simulación y control de un péndulo invertido en la página “Control Tutorials for Matlab & Simulink” [16]. La secuencia de comandos básica se ve de la siguiente forma:

```
% SIMULACIÓN DEL SISTEMA DE PÉNDULO INVERTIDO EN
PLATAFORMA MATEMÁTICA

%PARÁMETROS DEL MOTOR BLDC

Ke_d=9.31; %Valor del cte de velocidad en V/KRPM (hoja
de datos)
Ke=(Ke_d*60)/(1000*2*pi); %Constante de voltaje cem en
V-s/rad
Kt_d=17; %Valor cte de torque en oz-in/A (hoja de
datos)
Kt=Kt_d*0.0070; %Constante de torque en N-m/A
R=0.18; %Resistencia línea a línea en ohmios

%PARÁMETROS DE INERCIA

N=3; %Tasa de reducción
n=0.95; %Eficiencia de la reductora
```

```

Irotor_d=0.02266; %Inercia del rotor en oz-pulg-s2
(hoja de datos)
Irotor=Irotor_d*0.00706154; %Inercia del rotor en Kg-
m^2
M_polea=0.34; %Masa de las poleas en Kg
Rext_polea=0.039; %Radio exterior de las poleas en m
Rint_polea=0.01; %Radio interior de las poleas en m
Rpaso_polea=0.03974; %Radio de paso de la polea
Ipolea=0.5*M_polea*((Rext_polea^2)+(Rint_polea^2));
%Inercia de una polea en Kg-m^2
M_carro=1; %Masa del carro en Kg
M_pendulo=0.1646; %Masa del péndulo en Kg
Icarga=(M_carro+M_pendulo)*Rpaso_polea^2+2*Ipolea;
%Inercia de la carga
Im=Irotor+(Icarga/(n*(N^2))); %Inercia equivalente

%PARAMETROS DE FRICCIÓN VISCOSA

B_motor=0.001; %Coef. de fricción viscosa del rotor
[N-m-s]
B_carro=0.005; %Coef. de fricción viscosa entre carro
y guías [N-s/m]
B_polea=0.01; %Coef. de fricción viscosa de polea
conducida [N-m-s]
Bm=B_motor+(((B_carro*(Rext_polea^2))+B_polea))/(n*(N^
2)); %Coef. de fricción viscosa equivalente [N-m-s]

%PARÁMETROS ADICIONALES DEL CONJUNTO CARRO-PÉNDULO

g = 9.8; %Gravedad en m/s^2
l = 0.45; %Longitud del péndulo en metros
I = M_pendulo*l^2; %inercia del péndulo en Kg.m^2
q = (M_carro+M_pendulo)*l^2-
(M_carro+M_pendulo)*(I+M_pendulo*l^2); %definición de
expresión auxiliar
s = tf('s');

%FUNCIONES DE TRANSFERENCIA

%Función de Transferencia del motor (Voltaje vs Torque
Motor)

T_motor=(((Kt*Im*s)+Kt*Bm)/(((R*Im*s)+(R*Bm+Kt*Ke))));
[num1,den1]=tfdata(T_motor,'v');

%Función de Transferencia de la reductora (torque
Motor vs Torque de Reductora)

T_red=(N*n)/l;

%Función de Transferencia del Sistema de Polea-Banda
Dentada (Torque de Reductora vs Fuerza)

```

```

F_pb=(1/Rpaso_polea);

%Función de transferencia de la posición lineal del
carro vs fuerza
P_carro = ((-(I+M_pendolo*l^2)/q)*s^2 +
(M_pendolo*g*l/q))/(s^4 - (B_carro*(I +
M_pendolo*l^2))*s^3/q + ((M_carro +
M_pendolo)*M_pendolo*g*l)*s^2/q +
B_carro*M_pendolo*g*l*s/q);
[num2,den2]=tfdata(P_carro,'v');

%Función de transferencia de la posición angular del
péndulo vs fuerza
P_pend = (M_pendolo*l*s/q)/(s^3 - (B_carro*(I +
M_pendolo*l^2))*s^2/q +((M_carro +
M_pendolo)*M_pendolo*g*l)*s/q -
B_carro*M_pendolo*g*l/q);
[num3,den3]=tfdata(P_pend,'v');

%Multiplicación de las FdT en serie
%FdT que relaciona entrada de voltaje y posición
lineal del carro

P_carro2=T_motor*T_red*F_pb*P_carro;
[num4,den4]=tfdata(P_carro2,'v');

%FdT que relaciona entrada de voltaje y posición
angular del péndulo

P_pend2=T_motor*T_red*F_pb*P_pend;
[num5,den5]=tfdata(P_pend2,'v');

```


Para analizar las respuestas del sistema se usaron las dos señales de entrada más usadas para éste propósito, la señal impulso y el escalón unitario.

Primeramente se definió el sistema **sist_ft** dado por las funciones de transferencia finales creadas en el programa anterior, luego se definieron las entradas y salidas del sistema, donde la entrada E de voltaje de fases es común para ambas funciones de transferencia, y las salidas con X para la posición lineal del carro y Teta para la posición angular del péndulo. Por último usando el comando set se configuraron las propiedades de los objetos (nombres de entradas y salidas del sistema). Estas líneas de comandos se colocaron a continuación del conjunto anterior de línea de comandos o programa.

```
%Sistema definido por sus funciones de transferencia
sist_ft = [P_carro2 ; P_pend2];

%Definición de entradas y salidas
inputs = {'E'};
outputs = {'X'; 'Teta'};

%Configuración de propiedades de objetos
set(sist_ft, 'InputName', inputs)
set(sist_ft, 'OutputName', outputs)
```

Para observar las respuestas del sistema a las entradas impulso o paso unitario, se añadieron líneas de código adicionales, primero definiendo un vector de tiempo y luego usando los comandos **impulse** y **step** respectivamente, que grafican la respuesta temporal del sistema **sist_ft** en el vector de tiempo **t** dado.

Para las respuestas a la entrada impulso se tienen las siguientes líneas de código y la gráfica correspondiente.

```
%Respuesta del sistema en Lazo Abierto a entrada tipo
Impulso

t=0:0.01:1;
impulse(sist_ft,t);
title('Respuesta del Sistema en Lazo Abierto a entrada
tipo Impulso')
grid on
```

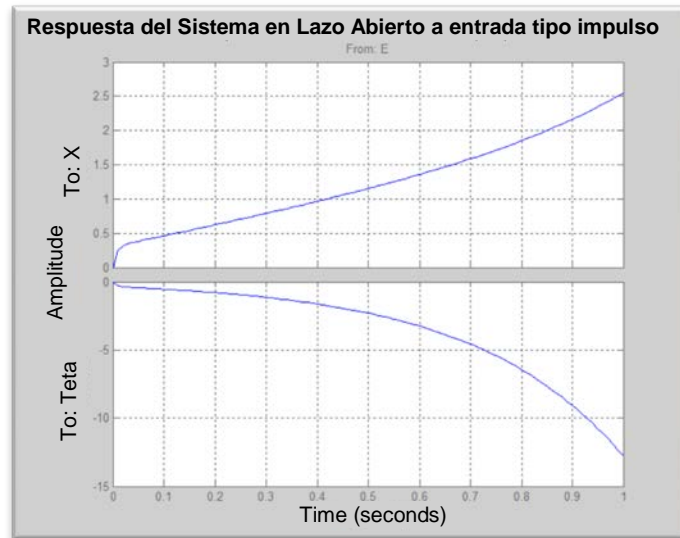


FIGURA 3.2. RESPUESTA DEL SISTEMA A ENTRADA IMPULSO, EN LAZO ABIERTO

Y para observar la respuesta al paso unitario, las líneas de código y la gráfica generada son presentadas a continuación.

```
%Respuesta del sistema en Lazo Abierto a entrada tipo
Paso
t=0:0.01:1;
step(sist_ft,t);
title('Respuesta del Sistema en Lazo Abierto a entrada
tipo Paso')
grid on
```

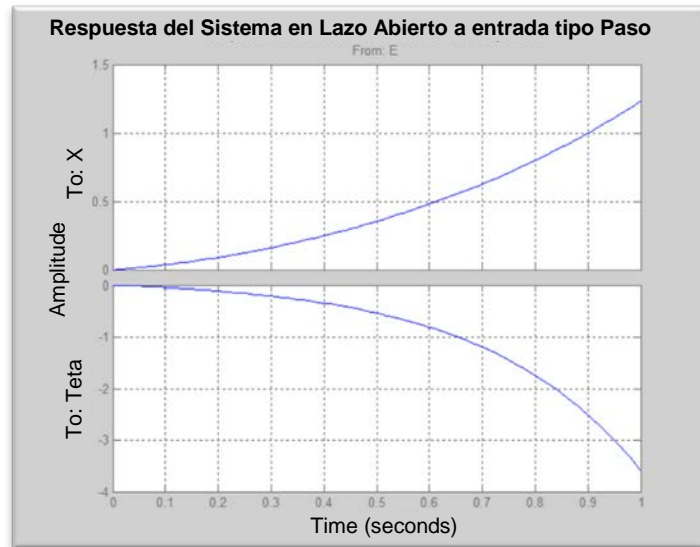


FIGURA 3.3. RESPUESTA DEL SISTEMA A ENTRADA PASO, EN LAZO ABIERTO

3.2 Diagrama de bloques del sistema en una plataforma grafica de simulación

Para poder llevar a cabo la siguiente fase de la simulación, es necesario primero hacer una breve introducción a ciertos aspectos de la plataforma, la cual en este caso es parte de Matlab, lo cual facilita la interacción entre ambos entornos.

3.2.1 Introducción a la plataforma grafica de simulación

Simulink® es un entorno de diagrama de bloques para simulación multi-dominio y diseño basado en modelos. Soporta simulación, generación automática de código, y análisis continuo y verificación de sistemas embebidos.

Simulink provee un editor gráfico, librerías de bloques personalizadas, y resolvedores para modelización y simulación de sistemas dinámicos. Está integrado con MATLAB®, permitiendo incorporar algoritmos de MATLAB en los modelos y exportar los resultados de la simulación a MATLAB para un posterior análisis.

Capacidades: Construcción del modelo – Subsistemas jerárquicos del modelo con librerías de bloques predefinidos.

Simulación del modelo – Simula el comportamiento dinámico del sistema y permite visualizar los resultados mientras la simulación se está ejecutando. Análisis de Resultados de la

Simulación – Se observan los resultados de la simulación y se depura la simulación.

Manejo de Proyectos – Fácil manejo de archivos, componentes, y grandes cantidades de datos para el proyecto. Conexión con Hardware – Conexión del modelo a hardware para pruebas en tiempo real y despliegue de sistemas embebidos [17].

El espacio de trabajo por una ventana principal (1) donde se arman los diagramas de bloques, la barra de menús (2), y la ventana de las librerías de bloques (3) que muestra un árbol de clasificaciones o categorías a la izquierda y los bloques que contiene a la derecha (figura 3.4).

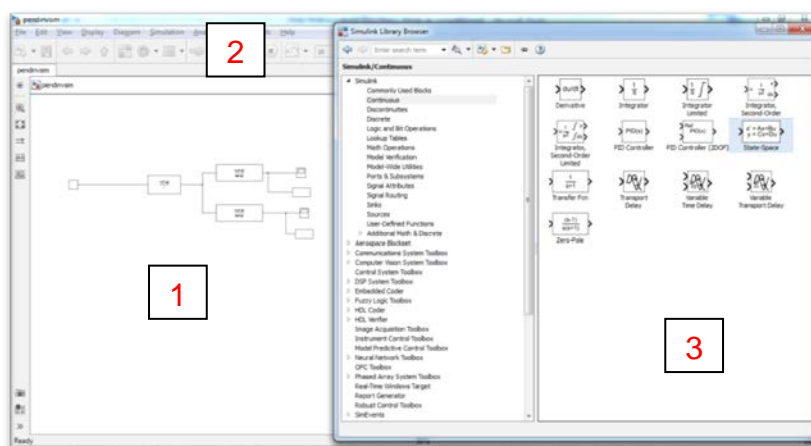
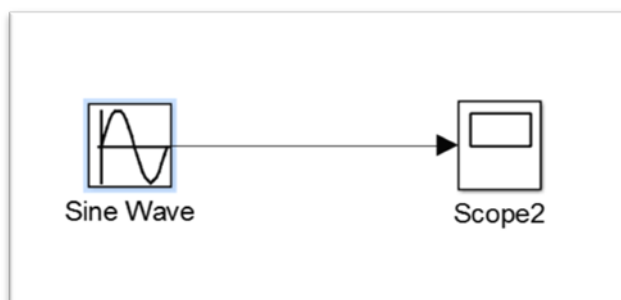


FIGURA 3.4. ENTORNO DE TRABAJO DE SIMULINK

Para elaborar un diagrama de bloques se selecciona el bloque que se desea y se lo arrastra al espacio de trabajo principal, las flechas indican la dirección de los datos y como deben conectarse, de esta forma, por ejemplo, no se puede conectar la salida de un bloque a la salida de otro.

Un ejemplo sencillo sería obtener la gráfica de una función, como el seno, cuyo diagrama de bloques se muestra en la figura 3.5



**FIGURA 3.5. DIAGRAMA DE BLOQUES PARA GRAFICAR
FUNCIÓN SENO**

Existen otros bloques que contienen tanto una entrada como una salida, son bloques que procesan la entrada y entregan como salida la señal de entrada modificada de acuerdo a la función del bloque, un ejemplo clásico de ello es el bloque de ganancia que multiplica la

entrada que recibe por una ganancia ingresada por el usuario (dando doble click sobre el bloque se ingresa a su cuadro de diálogo) y entrega la salida modificada, si se aplica al ejemplo anterior se tiene lo que muestra la figura 3.6:

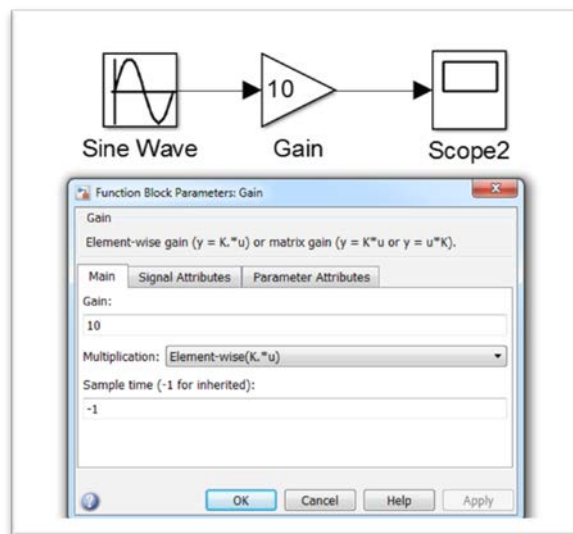


FIGURA 3.6. DIAGRAMA DE BLOQUES PARA GRAFICAR FUNCIÓN SENO CON GANANCIA 10

Otras funciones y procedimientos específicos para trabajar en Simulink son descritos más adelante conforme su aplicación.

3.2.2 Elaboración y simulación del programa de diagrama de bloques

Tutoriales para el uso del programa pueden encontrarse en clasificados por categorías o lo que se quiere lograr, desde lo más básico como el estilo de programación por bloques y las descripciones de estos hasta configuraciones especiales para tareas más específicas [18]. Los bloques más importantes para elaborar estos diagramas son los que representan a las funciones de transferencia, el bloque base para ello se encuentra en la categoría “continuous” de la librería de bloques.

El bloque fue arrastrado y soltado en la ventana principal, y se le dio doble click para acceder a su cuadro de diálogo como se muestra en la figura 3.7.

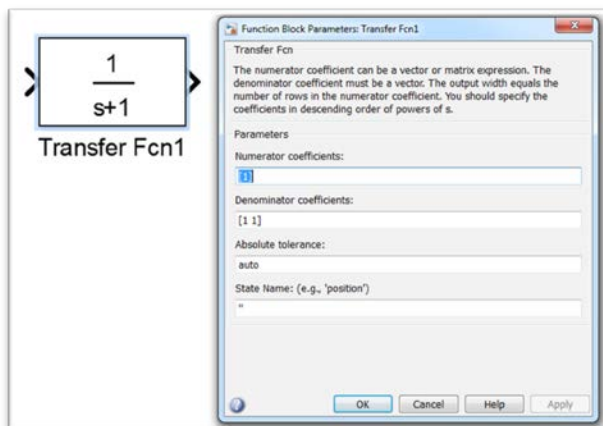


FIGURA 3.7. BLOQUE FUNCIÓN DE TRANSFERENCIA Y SU CUADRO DE DIÁLOGO.

Se puede observar que el bloque de función de transferencia posee tanto una entrada o una salida, lo cual es consecuente con su estructura de relacionar una entrada con una salida. Por ejemplo la entrada sería de fuerza y a la salida el correspondiente ángulo del péndulo.

En su cuadro de diálogo se puede notar dos entradas de parámetros importantes: los coeficientes del numerador y denominador, los cuales pueden ingresarse directamente en forma de vectores, o como en el caso de este proyecto obtenerse del programa en Matlab escrito anteriormente.

Para ello, se necesita añadir los nombres de las variables correspondientes en las cuales se almacenaron los coeficientes extraídos con el comando **tfdata**. Por ejemplo, para la primera función de transferencia que relaciona entrada de voltaje en las fases con salida de torque de motor, los nombres de las variables fueron num1 y den1 (figura 3.8).

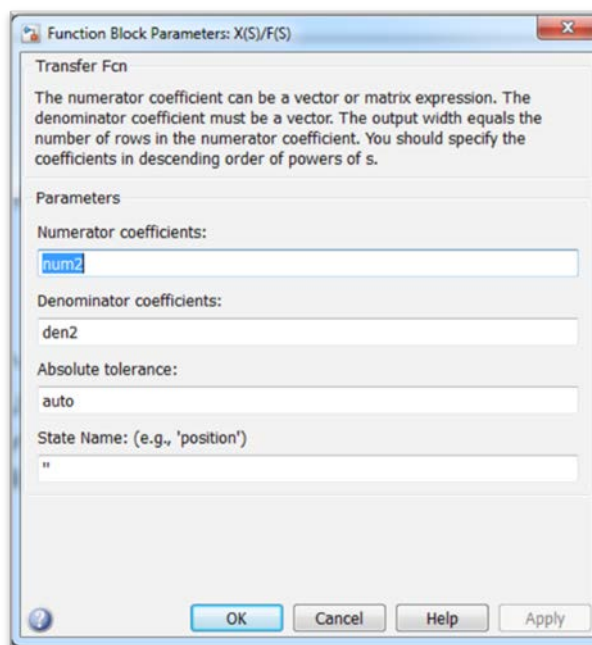


FIGURA 3.8. INGRESO DE NUMERADORES Y DENOMINADORES EN BLOQUES DE FUNCIONES DE TRANSFERENCIA.

Dado que se deseaba mostrar primero como está estructurado el diagrama de bloques por partes o etapas, se trabajó con las funciones de transferencia básicas, la de posición lineal versus fuerza (num2, den2) y posición angular versus fuerza (num3, den3). Además, las relaciones entre el torque del motor de entrada y el torque de reductora a la salida, y la de torque de reductora y fuerza, son ambas una constante multiplicada, por tanto se usó el bloque de ganancias para su representación, ingresando en sus respectivos cuadros del diálogo no una cantidad numérica sino el nombre de función de transferencia asignado en el programa de Matlab.

Los tres primeros bloques se ubicaron y conectaron de forma secuencial, luego dado que la fuerza es la señal de entrada para las dos últimas funciones de transferencia, se realizó una conexión bifurcada, haciendo doble click sobre las líneas se pudo etiquetar debidamente el tipo de señal correspondiente (figura 3.9).

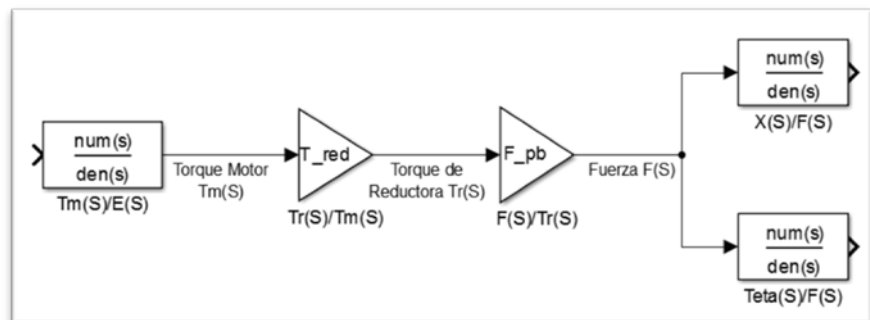


FIGURA 3.9. DIAGRAMA DE BLOQUES PRELIMINAR DEL SISTEMA NO SIMPLIFICADO.

Se ingresó la señal de entrada dada por un bloque de señal tipo escalón unitario, con un retraso de 1 segundo amplitud de 1, este bloque se encuentra en la librería de Fuentes o “Sources”, para poder observar las gráficas de posición generadas se introducen dos bloques de visualización (“Scope”, de la librería “Sinks”) y se conectan a las salidas de los bloques de funciones de transferencia $X(S)/F(S)$ y $Teta(S)/F(S)$ respectivamente, por último se etiquetan los bloques y líneas de conexión creados (figura 3.10).

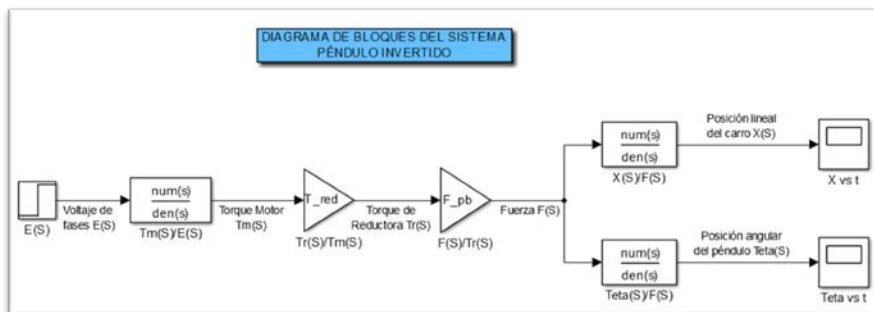


FIGURA 3.10. DIAGRAMA DE BLOQUES DEL SISTEMA NO SIMPLIFICADO EN LAZO ABIERTO.

En la figura 3.10 se puede observar los procesos en serie, los cuales de acuerdo al álgebra de diagrama de bloques pueden simplificarse en un solo bloque cuya función de transferencia es el producto de las correspondientes a los procesos en serie, esto ya se hizo previamente en el programa de Matlab del capítulo 2 donde los numeradores y denominadores de las funciones de transferencia resultantes se nombraron num4, den4 y num5, den5. De esta forma los bloques después de la entrada de paso y antes de los bloques de visualización fueron reemplazados por un par de bloques de funciones de transferencia con los numeradores y denominadores nombrados en líneas anteriores (figura 3.11).

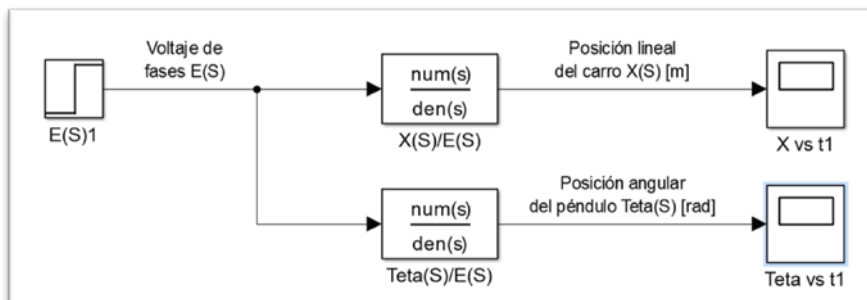


FIGURA 3.11. DIAGRAMA DE BLOQUES DEL SISTEMA SIMPLIFICADO PRELIMINAR.

Otra modificación conveniente es en la forma de la visualización, donde es mejor poder observar las señales de salida y la entrada en una sola ventana con gráficas separadas, para ello se necesitó un solo bloque de visualización con el número de ejes modificados, esto se logró dando click sobre el ícono de configuración (engranaje) e ingresando el número de señales deseadas en el campo "Number of axes", en este caso es 3, uno para la entrada y dos para las salidas. También se modificaron, accediendo a la pestaña "Style", los colores de fondo y ejes, y el grosor y color de línea (figura 3.12).

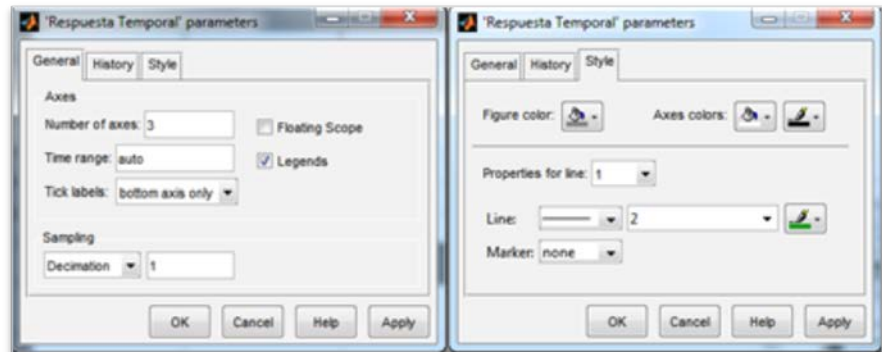


FIGURA 3.12. CONFIGURACIÓN DE ESTILOS DE LÍNEAS DEL BLOQUE DE VISUALIZACIÓN.

El bloque de visualización modificado presentaba tres entradas, a éstas se conectaron las salidas de los dos bloques del sistema y el de la entrada de voltaje de fases. Para dejar al sistema como un solo bloque, se creó un subsistema, seleccionando los dos bloques de las funciones de transferencia del sistema, dando click derecho y eligiendo la opción “Create subsystem from selection”, luego se nombró al bloque resultante como “Planta de sistema de Péndulo Invertido” (figura 3.13)

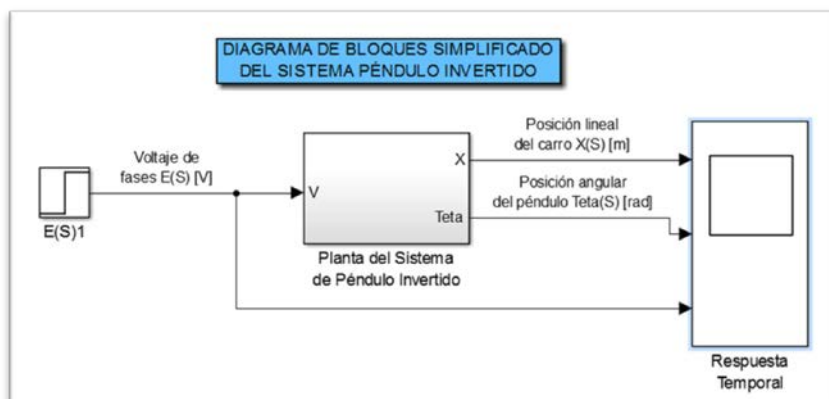


FIGURA 3.13. DIAGRAMA DE BLOQUES DEL SISTEMA SIMPLIFICADO EN LAZO ABIERTO.

Por último, se configuraron parámetros básicos de la simulación, como el tiempo de muestreo y la duración de la simulación, para ello se da click en la pestaña “Simulation” de la barra de funciones y luego en “model configuration parameters”, se introducen 0.0 y 2.0 como tiempos de inicio y final de simulación, tipo de muestreo de paso fijo o “fixed step”, con tiempo de muestreo de 0.001, la ventana de configuración debe verse como en la figura 3.14.

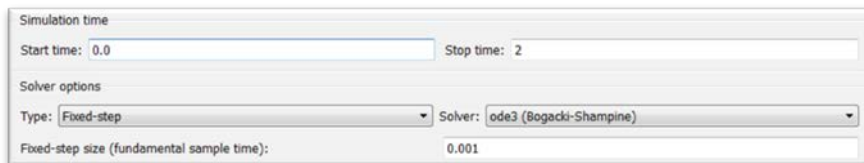


FIGURA 3.14. CONFIGURACIÓN DE LA DURACIÓN Y TIEMPO DE MUESTRO DE LA SIMULACIÓN.

Se ejecutó la simulación dando click al botón “Run” (botón de reproducción verde), se esperó hasta que ésta finalice y se dio doble click sobre los bloques de visualización para observar las gráficas obtenidas (figura 3.15).

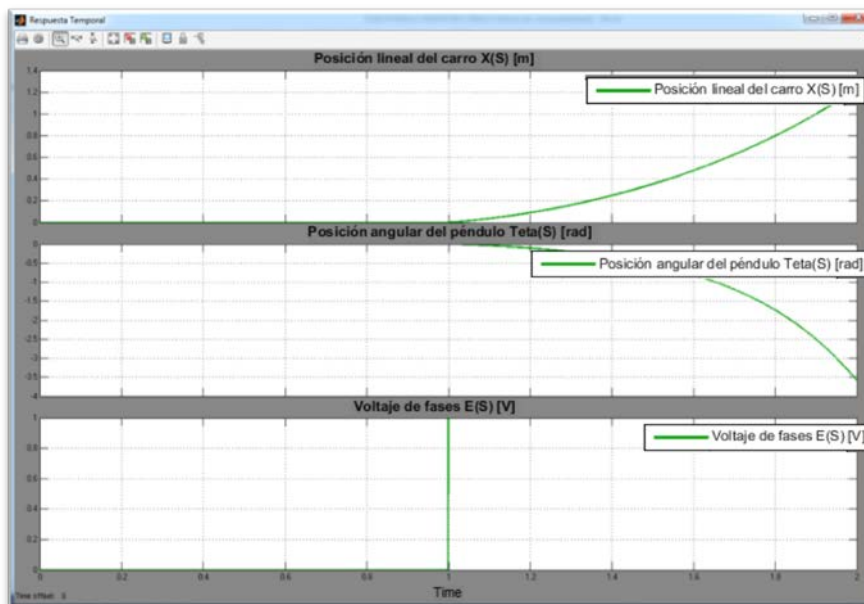


FIGURA 3.15. RESPUESTA ESCALÓN DEL SISTEMA EN LAZO ABIERTO.

Como se mencionó en la sección anterior, para modelar la etapa de auto-levantamiento se tuvo que usar ecuaciones no lineales, un método para simularlas usando bloques comunes de Simulink se puede consultar en la página de “Control Tutorials for Matlab and Simulink”, sección de “Modeling” [13].

Los bloques principales para la simulación del sistema no lineal son: “Fcn” de la librería “User Defined Functions”, el bloque “Integrator” o integrador de la librería “Continuous”, los bloques “In1” y “Out” de las librerías “Sources” y “Sinks” respectivamente, y “Mux” de la librería “Signal Routing” para las conexiones múltiples.

Primero se arrastraron y soltaron los bloques de funciones (se necesitaron cuatro, uno por cada ecuación no lineal de la 2.54 a la 2.57), cada uno con un conector Mux, y en el caso de las dos primeras (las ecuaciones de \ddot{x} y $\ddot{\theta}$) se conectaron un par de integradores en serie con las salidas de los bloques, parecido a como se muestra en la figura 3.12, se nombró cada bloque acorde a lo que representa, la entrada de fuerza se dejó desconectada.

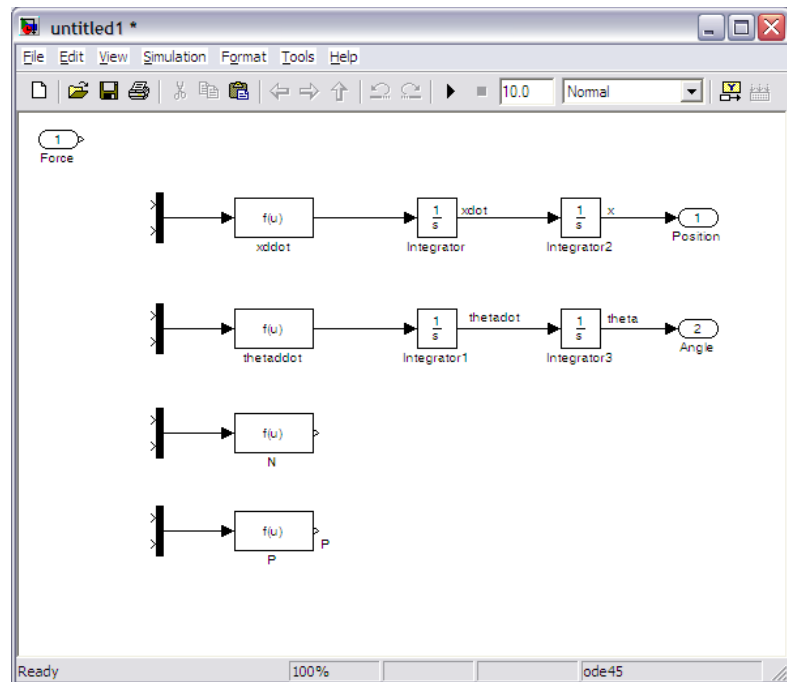


FIGURA 3.16. DISPOSICIÓN INICIAL DE BLOQUES PARA SIMULACIÓN NO LINEAL [13].

También, en los bloques integradores se introdujeron las condiciones iniciales, que es cero en todos los casos con la excepción del integrador donde se obtiene el ángulo del péndulo, pues de acuerdo al sistema de referencia elegido, cuando el péndulo está colgando en su posición de equilibrio estable, éste se encuentra en pi radianes.

Al dar doble click sobre los bloques de funciones aparece una ventana de dialogo con una entrada para "expression", aquí es donde se introdujo la expresión deseada. Por ejemplo, en la figura 3.17 se muestra la expresión ingresada representando a la ecuación 2.54, donde las variables F, reacción horizontal H y \dot{x} , se ingresan en la forma u(n), donde n es el orden en que se conectan al Mux de arriba hacia abajo.

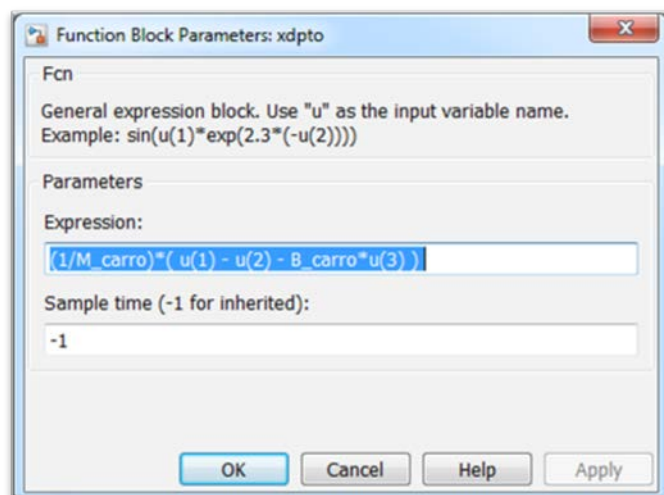


FIGURA 3.17. CONFIGURACIÓN DE BLOQUES DE FUNCIONES.

Se ingresaron las expresiones en los demás bloques, tomando en cuenta que para la ecuación 2.55 las variables fueron H , V y θ ; para la ecuación 2.56 \ddot{x} , θ , $\dot{\theta}$ y $\ddot{\theta}$; y finalmente para la ecuación 2.57 las variables fueron θ , $\dot{\theta}$ y $\ddot{\theta}$.

Los conectores Mux vienen por defecto con dos entradas, del párrafo anterior se puede ver que en algunos casos eran necesarias tres o cuatro entradas, lo cual se puede configurar al da doble click sobre el conector y en el campo "Number of inputs" ingresar el número correspondiente, como se muestra en la figura 3.18.

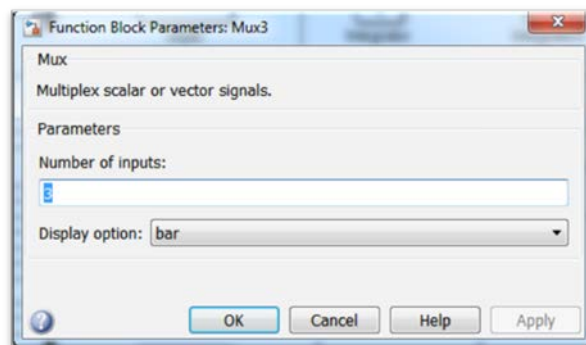


FIGURA 3.18. CONFIGURACIÓN DE CONECTORES MUX.

Por último se conectaron todos los bloques como se indica en la figura 3.19, etiquetando los bloques de acuerdo a lo que representaban.

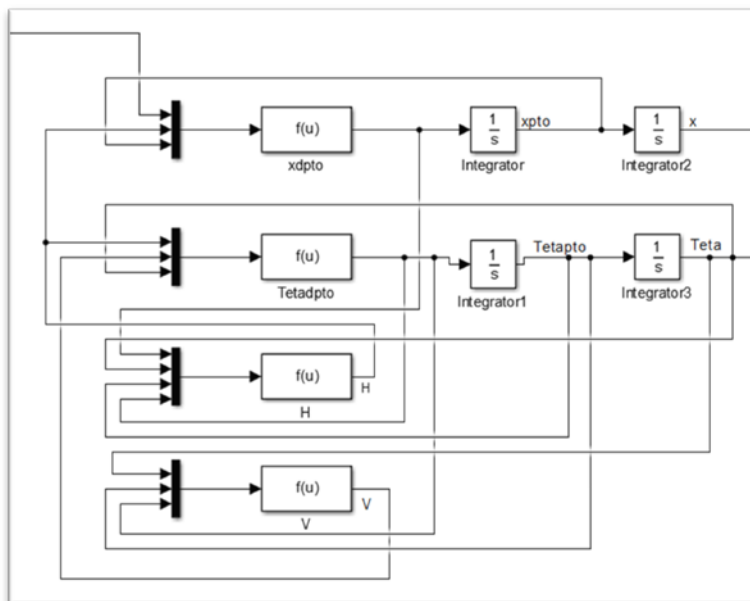


FIGURA 3.19. CONEXIONES DE LOS BLOQUES DEL SISTEMA NO LINEAL.

Los bloques de las ecuaciones no lineales esperaban fuerza como señal de entrada, y la entrada principal del sistema completo es de voltaje de fases, por lo tanto se necesitaba introducir la etapa que realiza la conversión entre estas dos señales, dicha etapa estaba representada por las funciones de transferencia del motor, reductora y sistema de polea-banda, es decir los tres primeros bloques previamente configurados en la figura 3.9, con esa adición, y el bloque de entrada escalón y el bloque de visualización, el diagrama de bloques quedó como se observa en la figura 3.20.

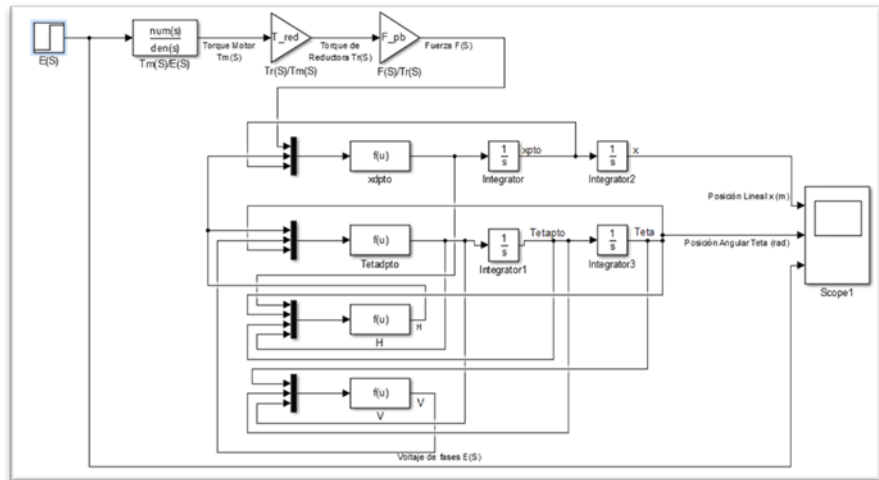


FIGURA 3.20. CONEXIONES DE LOS BLOQUES DEL SISTEMA NO LINEAL.

Dada la cantidad de bloques y conexiones, todo lo anterior se ingresó en un solo bloque o sub-sistema, usando el mismo procedimiento al indicado en el diagrama de bloques del sistema lineal. Se etiquetó el bloque planta y se añadió un título para identificar propiamente al diagrama (figura 3.21).

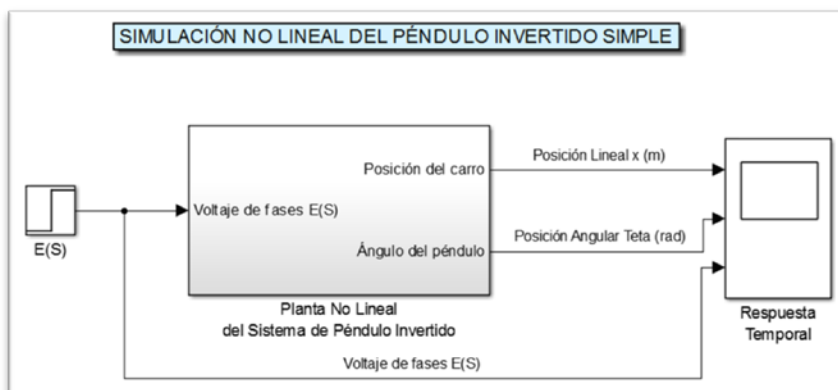


FIGURA 3.21. DIAGRAMA SIMPLIFICADO DEL SISTEMA NO LINEAL.

Se ejecutó la simulación, esta vez con una duración de 10 segundos para observar parte de la oscilación del péndulo, obteniendo la siguiente gráfica (figura 3.22).

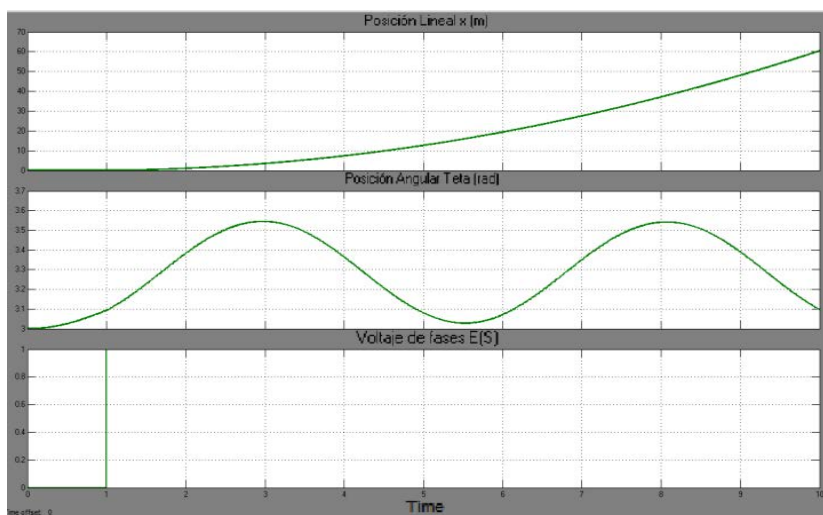


FIGURA 3.22. RESPUESTA DEL SISTEMA NO LINEAL EN REPOSO A UNA ENTRADA TIPO PASO.

3.3 Análisis de respuesta temporal del sistema simulado y ajustes al modelo

Observando las respuestas a la entrada escalón, tanto para el sistema linealizado alrededor del punto de estabilización invertido (figura 3.15), como para el sistema no lineal que parte de la posición de equilibrio estable (figura 3.22) se pudo ver que en ambos casos se vuelven altamente inestables, con la posición lineal y la posición angular creciendo sin control, a excepción de la posición angular en el sistema no lineal donde el péndulo parte de su posición de reposo y oscila.

Por lo tanto, al menos en el caso de la posición lineal y la posición angular, los sistemas no se acercaron a alguno de 1er o 2do orden.

A simple vista parece ser el efecto de un polo real positivo que ante una entrada escalón, presenta una respuesta inestable y creciente como se muestra en la figura 3.23.

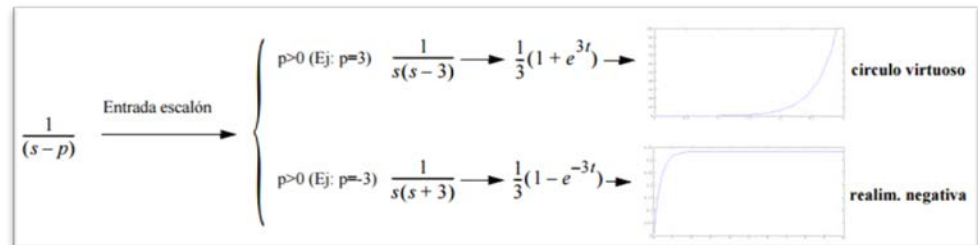


FIGURA 3.23. EFECTOS DE LOS POLOS DE UN SISTEMA EN SU RESPUESTA ESCALÓN [19].

Para ello se analizaron los polos y ceros de las funciones de transferencia, esto obviamente solo se pudo realizar en el caso del modelo lineal para la estabilización.

Se utilizó el comando **tf2zp**, el cual para un numerador y denominador dados de una función de transferencia, regresa los polos, ceros y ganancias de la misma. Las funciones de transferencia a analizar son la 4 (posición del carro versus voltaje) y la 5 (posición del péndulo versus voltaje), se añadieron las siguientes líneas de comando al programa principal.

```
[z4,p4,k4]=tf2zp(num4,den4)
[z5,p5,k5]=tf2zp(num5,den5)
```

A continuación se presentan los resultados obtenidos para los ceros (z), polos (p) y ganancias (k) de las funciones de transferencia del sistema linealizado.

z4 =

3.2998
-4.9375
-3.2998

p4 =

0
-138.6431
3.4228
-3.4232
-0.0043

k4 =

43.8070

z5 =

0
-4.9375

p5 =

-138.6431
-3.4274
3.4185
0.0043

k5 =

-48.6745

En ambos casos puede verse que al menos ambas funciones de transferencia tenían un polo positivo, es decir en el semiplano real ($p_4=3.4228$ y $p_5=3.4185$), lo cual los confirmó como sistemas inestables

Además tomando como referencia el diagrama de la figura 3.24, se pudo ver que al no tener los polos componentes imaginarias, no se presentaron comportamientos oscilatorios. De esta forma las respuestas a las señales impulso y escalón o paso, fueron crecimientos no controlados en la salida.

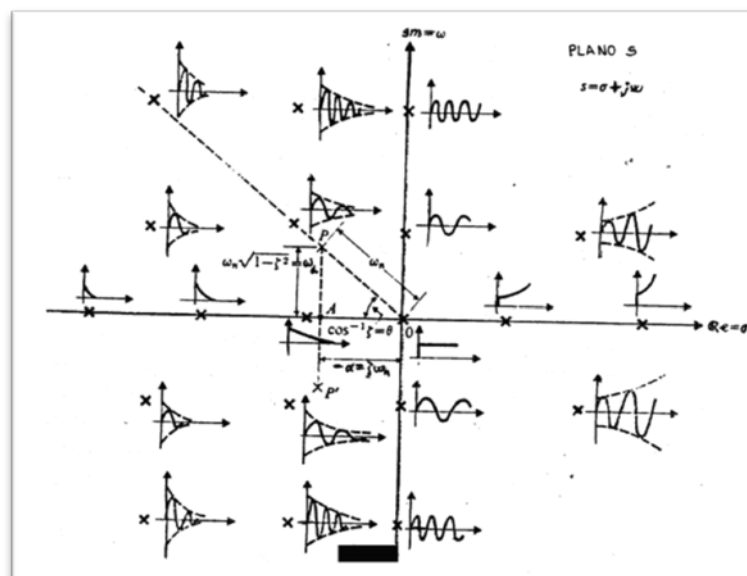


FIGURA 3.24. RESPUESTAS AL IMPULSO SEGÚN UBICACIONES DE POLOS [20].

En cuanto a los ajustes al modelo, pruebas preliminares del driver y motor usando pulsadores, generador de funciones y osciloscopio, revelaron dos curiosidades, la primera es que el voltaje de referencia justo antes de que se produzca movimiento es de aproximadamente 1.62 Voltios, también muy cerca de este valor recién se observa alimentación de voltaje a las fases en lo que se aprecia como el mínimo ciclo de trabajo del PWM (figura 3.25); y la segunda, es que dada la capacidad de la fuente de poder y las exigencias de corriente en los repentinos cambios de dirección, la fuente no era capaz de proporcionar corriente suficiente en cambios de dirección súbitos cuando la entrada de voltaje de referencia es mayor a 2.4 Voltios. En resumen, la entrada de voltaje de referencia utilizable se estableció en el rango de 1.62 a 2.4 Voltios.

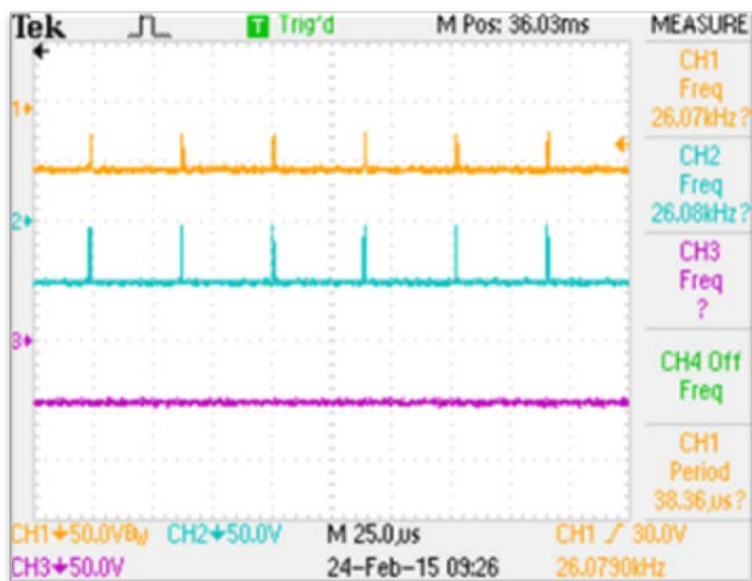


FIGURA 3.25. VOLTAJES EN LAS 3 FASES PARA UN VOLTAJE DE REFERENCIA APROXIMADO A 1.62 V.

De esta forma, se modificó la ecuación 2.17, quedando la expresión 3.1, la misma que aunque no estuvo presente en el análisis del sistema, fue de importancia en la implementación del control en tiempo real.

$$e_a = \begin{cases} 20.17v_{ref} - 32.68; v_{ref} = (1.62, 2.4) \\ 20.17v_{ref} + 32.68; v_{ref} = [-2.4, -1.62] \\ 0; v_{ref} = [-1.62, 1.62] \end{cases} \quad \text{Ec. 3.1}$$

CAPÍTULO 4

4. DISEÑO DEL CONTROLADOR

En este capítulo se detalla el proceso de diseño del controlador, empezando por la selección del tipo de controlador, los parámetros de desempeño que fueron usados luego para determinar la efectividad del mismo y si logró o no los objetivos, el diseño en sí y su simulación en la plataforma de Simulink usando diagramas de bloques.

4.1 Selección del tipo de controlador

Una de las formas de clasificar a los controladores es de acuerdo al tipo de sistema que controla (o enfoque que se aplique), así, de esta forma se tienen controladores lineales, y no lineales.

Las funciones de transferencia obtenidas en el capítulo dos respondían a un sistema linealizado, por lo tanto podía utilizarse

una técnica de control también lineal, además de ser concordante con el tipo de control aprendido a nivel de pregrado.

Uno de los controladores más utilizados y con aplicaciones en la industria es el Proporcional-Integral-Derivativo (PID), o sus diversas combinaciones como proporcional puro (P), proporcional integrativo (PI), y proporcional derivativo (PD).

Aunque los parámetros de desempeño fueron analizados en la siguiente sección, vale mencionar ciertos aspectos que se buscan, al menos de manera relativa. Por ejemplo, dada la dinámica del sistema se deseaba que la acción del controlador sea lo suficientemente rápida, para el ángulo del péndulo el error en estado estable debía ser muy bajo, para la posición del carro se podía ser más tolerante.

El problema de tener que controlar tanto la posición del péndulo como la del carro fue tratado en el proceso de diseño del controlador.

De acuerdo a la teoría de control [21] se tienen las siguientes características en las acciones del controlador P, I y D.

Considerando un sistema de lazo cerrado de control, tipo SISO, como el que se muestra en la figura 4.1.

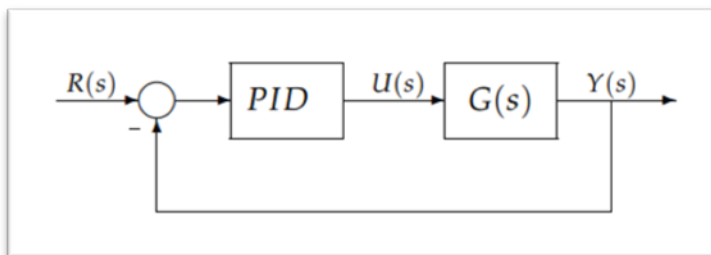


FIGURA 4.1. DIAGRAMA DE BLOQUES DE UN SISTEMA EN LAZO CERRADO.

P: acción de control proporcional, da una salida del controlador que es proporcional al error, (ecuación 4.1) Que descrita desde su función transferencia queda como se muestra en la ecuación 4.2:

$$u(t) = K_p e(t) \quad \text{Ec. 4.1}$$

$$C_p(S) = K_p \quad \text{Ec. 4.2}$$

Donde K_p es una ganancia proporcional ajustable. Un controlador proporcional puede controlar cualquier planta estable, pero posee un desempeño limitado y error en régimen permanente (off-set).

I: acción de control integral, da una salida del controlador que es proporcional al error acumulado, lo que implica que es un modo de controlar lento.

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad \text{Ec. 4.3}$$

$$C_i(S) = \frac{K_i}{S} \quad \text{Ec. 4.4}$$

La señal de control $u(t)$ tiene un valor diferente de cero cuando la señal de error $e(t)$ es cero. Por lo que se concluye que dada una referencia constante, o perturbaciones, el error en régimen permanente es cero.

PI: acción de control proporcional-integral, se define mediante:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad \text{Ec. 4.5}$$

Donde T_i se denomina tiempo integral y es quien ajusta la acción integral. La función de transferencia resulta:

$$C_{pi}(S) = K_p \left(1 + \frac{1}{T_i S} \right) \quad \text{Ec. 4.6}$$

Con un control proporcional, es necesario que exista error para tener una acción de control distinta de cero. Con acción integral, un error pequeño positivo siempre nos dará una acción de control creciente, y si el error fuese negativo la señal de control sería decreciente. Este razonamiento sencillo muestra que el error en régimen permanente será siempre cero.

Muchos controladores industriales tienen solo acción PI. Se puede demostrar que un control PI es adecuado para todos los procesos donde la dinámica es esencialmente de primer orden. Lo que puede demostrarse en forma sencilla, por ejemplo, mediante un ensayo al escalón.

PD: acción de control proporcional-derivativa, se define mediante:

$$u(t) = K_p e(t) + K_p P_d \frac{d e(t)}{dt} \quad \text{Ec. 4.7}$$

Donde T_d es una constante denominada tiempo derivativo. Esta acción tiene carácter de previsión, lo que hace más rápida la acción de control, aunque tiene la desventaja importante de amplificar las señales de ruido y puede provocar saturación en el actuador. La

acción de control derivativa nunca se utiliza por sí sola, debido a que solo es eficaz durante periodos transitorios. La función transferencia de un controlador PD resulta:

$$C_{pd}(S) = K_p + SK_p P_d \quad \text{Ec. 4.8}$$

Cuando una acción de control derivativa se agrega a un controlador proporcional, permite obtener un controlador de alta sensibilidad, es decir que responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande. Aunque el control derivativo no afecta en forma directa al error en estado estacionario, añade amortiguamiento al sistema y, por tanto, permite un valor más grande que la ganancia K , lo cual provoca una mejora en la precisión en estado estable.

PID: acción de control proporcional-integral-derivativa, esta acción combinada reúne las ventajas de cada una de las tres acciones de control individuales. La ecuación de un controlador con esta acción combinada se obtiene mediante:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p P_d \frac{d e(t)}{dt} \quad \text{Ec. 4.9}$$

Y su función transferencia resulta:

$$C_{pid}(S) = K_p \left(1 + \frac{1}{T_i S} + T_d S \right) \quad \text{Ec. 4.10}$$

Sin embargo los controladores PID son útiles para sistemas SISO, de una entrada y una salida, el sistema de péndulo invertido por el contrario es SIMO, donde la única entrada es la de control que es el voltaje de fases, y se tienen dos salidas (posición angular del péndulo y posición lineal del carro), y se necesita poder controlar ambas variables.

El problema del péndulo invertido en los sistemas con carrera finita es que en lo posible el carro debe moverse dentro de los límites para mantener al péndulo en su posición invertida y al carro en su posición inicial de cero. Esto es diferente por ejemplo en el péndulo de Furuta donde no hay límite de movimiento pues es rotatorio, y cualquier perturbación sobre el ángulo del péndulo puede ser satisfactoriamente controlada con un solo PID sin restricciones en la posición angular de la base.

En el caso descrito en este proyecto es necesario que el carro se mueva dentro de los límites de los rieles guía ante perturbaciones en el péndulo.

Dentro del campo de controladores PID, una solución para estos casos es usar dos controladores PID en cascada, donde la señal de control del primer controlador varía el “set point” o punto de ajuste del segundo. Es la señal de control del segundo la que actúa directamente sobre el sistema.

De acuerdo a la teoría de control en cascada [22] son dos las razones por las que se elige control en cascada: la primera es cuando el control realimentado simple no provee un desempeño satisfactorio a lazo cerrado (en el caso del péndulo invertido al no tener control sobre la posición lineal del carro), y la segunda es que la medida de la variable esté disponible (la posición del carro se mide directamente con el encoder en la polea conducida).

Para que la aplicación del control en cascada sea viable la variable secundaria debe satisfacer los siguientes criterios: indicar la ocurrencia de una importante perturbación, y que exista una relación causal entre la variable manipulada y la segunda variable.

Además, la variable secundaria debe tener una dinámica más rápida que la variable primaria. Típicamente t_p (tiempo pico) debe ser mayor que $3t_s$ (constante de tiempo del proceso secundario). Dado que el sistema en lazo abierto es inestable, la constante de tiempo no puede apreciarse.

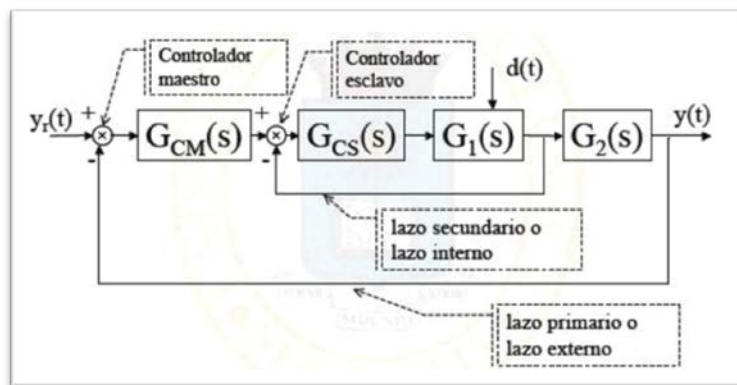


FIGURA 4.2. DIAGRAMA DE BLOQUES DE UN SISTEMA CON CONTROLADORES EN CASCADA.

En la figura 4.2, se tienen dos controladores en cascada G_{CM} y G_{CS} , al controlador externo se le llama controlador primario o maestro, al interno se le llama controlador secundario o esclavo.

4.2 Selección de parámetros de desempeño

Para el auto-levantamiento, el requerimiento principal era que el tiempo en que se ejecuta, es decir desde el pequeño impulso en la posición de reposo hasta que se produzca el cambio de estrategia de control, sea menor que 30 segundos.

Para la etapa de estabilización se requería que, ante una perturbación impulso en el ángulo del péndulo:

Para el péndulo:

Que no se salga del rango de estabilización (depende de amplitud de perturbación).

El tiempo de estabilización sea menor que 5 segundos.

Error en estado estable dentro del rango de ± 0.03 radianes.

Para el carro:

Que no se pegue en los extremos, sobresalto dentro del rango de ± 0.45 m.

El tiempo de estabilización debe ser menor que 5 segundos.

El error en estado estable debe estar dentro del rango de ± 0.02 m.

4.3 Proceso de diseño del controlador

Habiendo elegido el método de control de PID en cascada, el método de sintonización de sus parámetros era primero encontrar las ganancias del lazo interno o PID esclavo despreciando el PID externo o maestro, es decir al principio se hace la consideración de un sistema SISO como indica la figura 4.3.

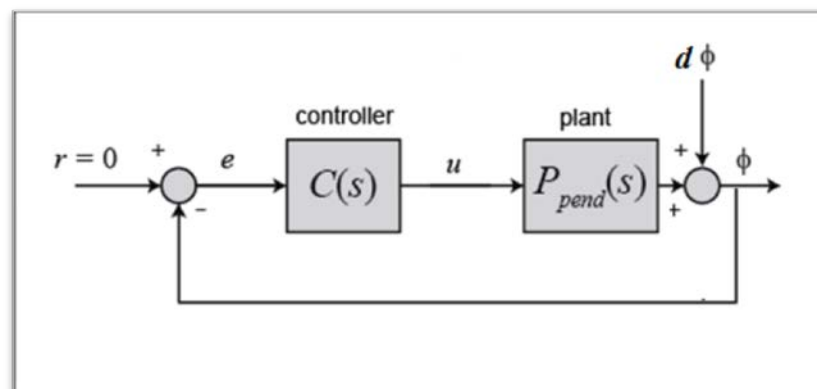


FIGURA 4.3. DIAGRAMA DE BLOQUES DEL LAZO INTERNO.

Lo anterior se explica con mayor detalle de acuerdo a la teoría relacionada al diseño de controladores en cascada [22], donde una consideración importante para la implementación de control en cascada es determinar cómo el proceso puede ser mejor dividido.

La selección de la variable controlada secundaria es tan importante en un sistema de control en cascada que es muy útil formalizar algunas reglas que ayuden a la selección:

Regla 1.- Diseñar el lazo secundario de manera que contenga las perturbaciones más serias, en el caso del sistema de péndulo invertido la perturbación más crítica es la de posición angular, pues este debe ser capaz de mantener su posición y no perder el equilibrio a pesar de las perturbaciones, además físicamente es más fácil perturbar la posición angular del péndulo que la posición lineal del carro que es más robusta.

Regla 2.- Hacer el lazo secundario tan rápido como sea posible incluyendo solamente los menores retrasos del sistema completo de control.

Regla 3.- Seleccionar una variable secundaria cuyos valores estén definitivamente y fácilmente relacionados a los valores de la variable primaria.

Regla 4.- Incluir en el lazo secundario tantas perturbaciones como sea posible, manteniéndolo al mismo tiempo, relativamente rápido.

Regla 5.- Escoger una variable secundaria de control que permita al controlador secundario operar a la ganancia más alta posible (la más baja banda proporcional). Esto es difícil de predecir.

El procedimiento para la sintonización los controladores en una configuración cascada se efectúa, igual que en controladores en configuración simple pero en dos etapas.

Primero se sintoniza el controlador secundario o esclavo sin tomar en cuenta al control primario que debe estar desactivado, para lo cual se deben seguir dos pasos: obtener un modelo de la parte del proceso incluida en el secundario (modelo de conocimiento o modelo experimental) lo cual ya fue desarrollado en los capítulos 2 y 3, y sintonizar el controlador secundario por cualquiera de los métodos conocidos (normalmente se utiliza un PI ya que el secundario debe ser un bucle rápido)

Luego se sintoniza el control primario o maestro, al igual que con el primario ayuda haber desarrollado un modelo matemático del sistema, con el control secundario ya trabajando se sintoniza el control primario hasta obtener la respuesta deseada.

Para el diseño del controlador del lazo interno se hizo uso de una herramienta de diseño de sistemas de control de Matlab conocida como Sisotool.

Sisotool, es una herramienta con interfaz gráfica para el análisis y diseño de compensadores en sistemas SISO y que es parte del “Control Systems Toolbox” de Matlab

Para acceder a esta herramienta se debe escribir **sisotool**, en la ventana de comandos de Matlab, y luego pulsar el botón Enter, de esta forma se abre la ventana de configuración.

La arquitectura mostrada por defecto (figura 4.4) es la elegida para el análisis.

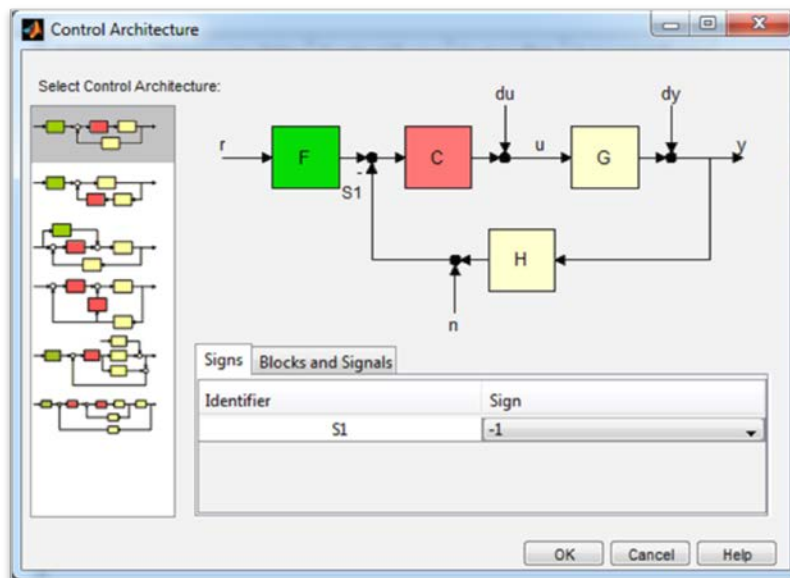


FIGURA 4.4. ARQUITECTURA DE DIAGRAMA DE BLOQUES EN SISOTOOL.

En el caso del lazo a analizar, solo importaban los bloques **C** del controlador a diseñar, y **G** que es el bloque de la planta del lazo interno, de esta forma los bloques **F** y **H** fueron considerados como ganancias unitarias que no afectan a las señales, así, **r** viene a ser la entrada de señal de ajuste (setpoint) de la posición angular del péndulo, **u** es la señal de control que en este caso es el voltaje a las fases, **y** es la señal de salida y variable medida que es la posición angular del péndulo, por último **dy** se consideró como una perturbación a la salida del sistema, y ésta fue utilizada como referencia en el diseño, pues no se buscaba que el

péndulo sea capaz de seguir una señal de referencia, sino que dado que el punto de control era constante durante la estabilización (alrededor de 0 radianes) éste pueda recuperarse después de una perturbación, esto también es conocido como un “problema de regulación”.

En cuanto a los datos del sistema, se ingresaron la función de transferencia correspondiente al bloque G de la planta, como se estaba analizando el lazo interno la función correspondiente era la que relacionaba voltaje en las fases con la posición angular del péndulo (P_pend2). Se dio click en “System Data” para abrir su ventana de configuración, luego se dio click en “Browse” para abrir la ventana que permite importar modelos, por defecto ya está definido que el modelo es para el bloque G, de entre la lista se eligió la función de transferencia mencionada líneas atrás, click al botón “Import”, luego “Close” y por último “Ok” en la ventana de datos del sistema (figura 4.5).

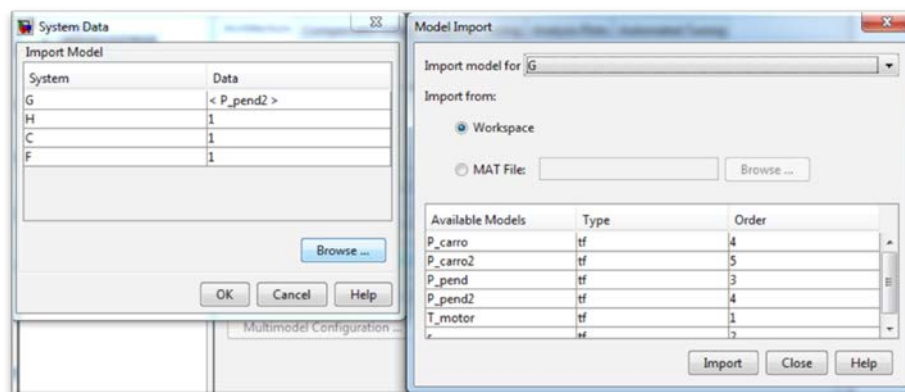


FIGURA 4.5. IMPORTACIÓN DE MODELO PARA EL BLOQUE G.

En la ventana principal de sisotool, se seleccionó la pestaña “Analysis Plots”, como ya se había mencionado, el diseño del controlador del lazo interno estuvo enfocado en el mantenimiento de la posición invertida del péndulo luego de una perturbación, por lo tanto en los tipos de gráficas (plot types) se eligió la opción “Impulse” y luego en la lista de respuestas se activó la casilla correspondiente en la respuesta en lazo cerrado de dy a y , la ventana debe verse como en la figura 4.6.

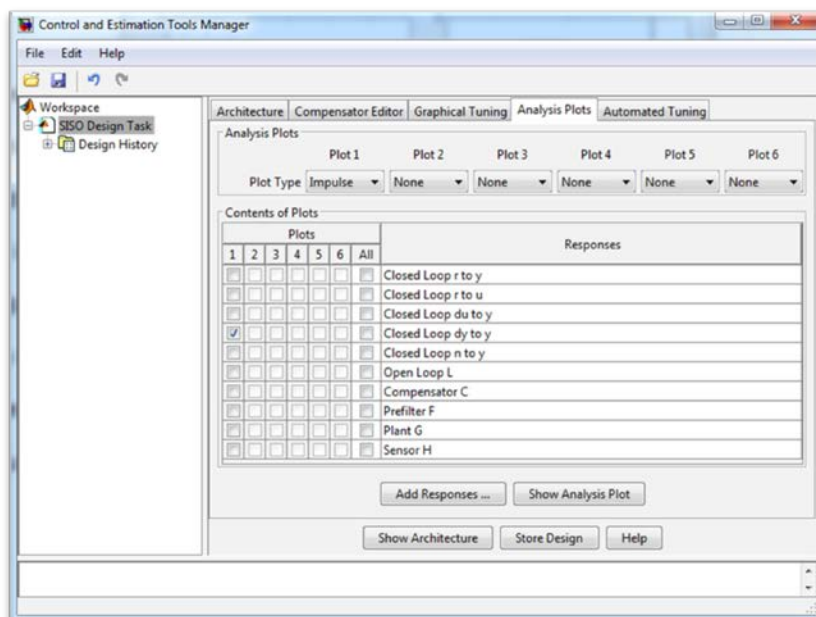
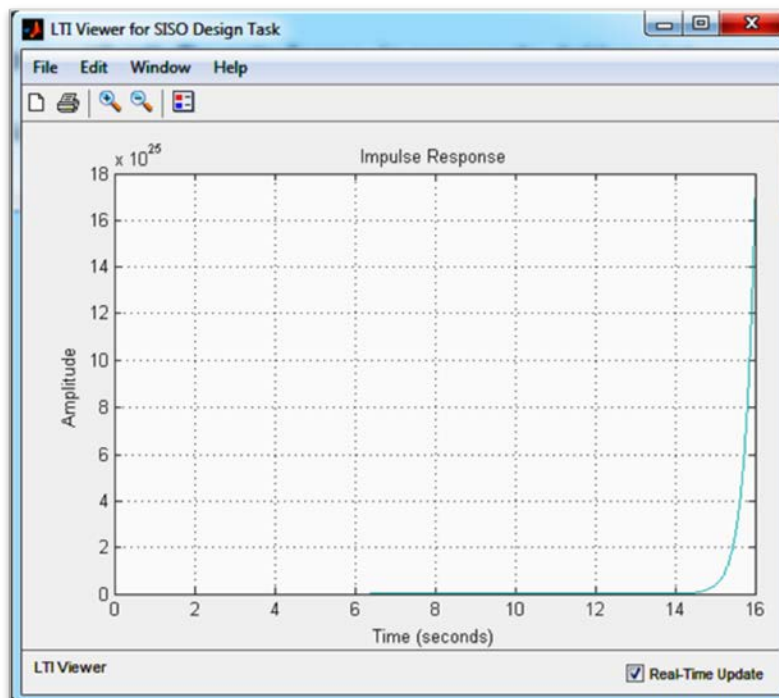


FIGURA 4.6. SELECCIÓN DE TIPO DE GRÁFICA DE RESPUESTA EN SISOTOOL.

Automáticamente se muestra la gráfica correspondiente, pero si por alguna razón ésta se cerrara, se puede volver a mostrar dando click en “Show Analysis Plot”, para una mejor visualización, se da click derecho en la gráfica y se activan las líneas de separación eligiendo la opción “Grid”. De esta forma, la respuesta del lazo interno ante una perturbación de tipo impulso en la posición angular del péndulo, con retroalimentación unitaria queda como se muestra en la figura 4.7.



**FIGURA 4.7. RESPUESTA A PERTURBACIÓN TIPO IMPULSO
CON RETROALIMENTACIÓN UNITARIA.**

La ventana de gráficos para el diseño del controlador (figura 4.8) también se abre de forma automática al ejecutar sisotool, sin embargo, si ésta se cierra, puede volver a abrirse dando click en "Show Design Plot" bajo la pestaña "Graphical Tuning".

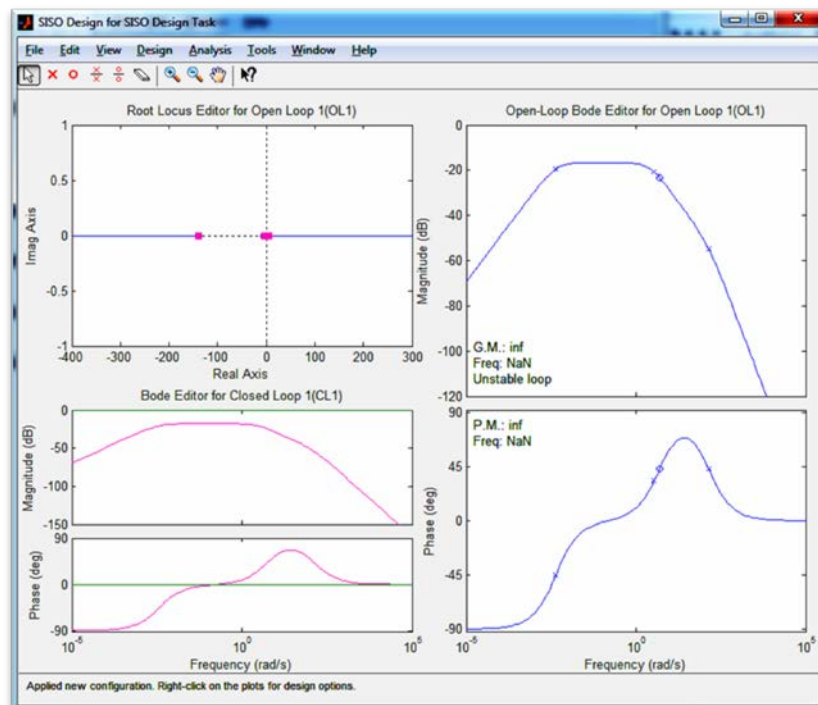


FIGURA 4.8. VENTANA GRÁFICA DE DISEÑO DE SISOTOOL.

El proceso de diseño del controlador estuvo basado en la modificación del lugar geométrico de las raíces (LGR), representado por la línea azul, y los puntos rosados representando a los polos, desplazables por la línea azul. Ingresando polos y ceros en diversas configuraciones cambia el LGR, idealmente éste debe pasar por las intersecciones de las rectas dadas por los parámetros de diseño para luego poder desplazar los polos dominantes a dichas intersecciones.

En la figura 4.8 puede verse un polo bastante alejado del origen, entre -100 y -200 del eje real. Haciendo zoom en el origen se podía ver con mayor detalle cómo están distribuidos los polos dominantes (figura 4.9), un polo en el origen más un polo positivo y uno negativo alejados de forma casi simétrica respecto al origen.

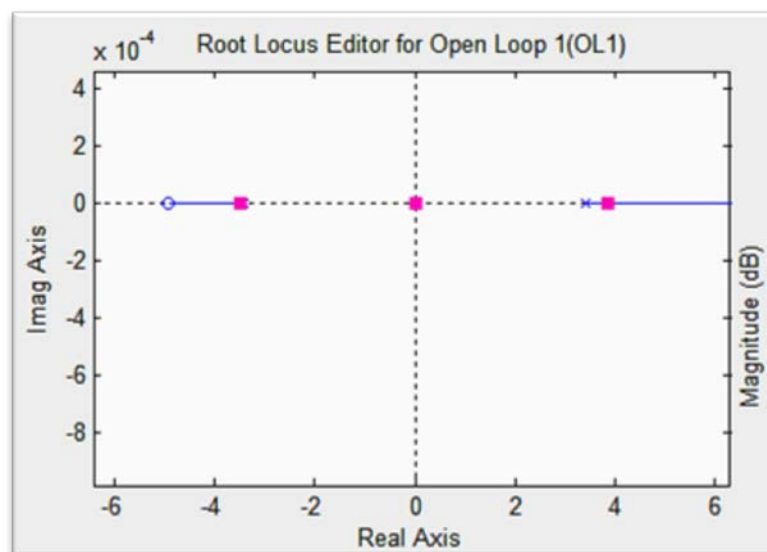


FIGURA 4.9. DETALLE DE LOS POLOS DOMINANTES.

Antes de modificar el LGR se debe tener una referencia, ésta es dada por los parámetros de diseño, se tomó en cuenta el tiempo de estabilización (menos de 5 segundos).

Para ingresar los parámetros de diseño hay que dirigirse a la ventana gráfica de diseño, dar click derecho en un área libre, elegir

“Design Requirements” (requerimientos de diseño) y luego “New”. En la pequeña ventana que aparece se eligió el tipo de requerimiento y se ingresó el valor del parámetro elegido (Settling Time) con un valor de 5 segundos como se muestra en la figura 4.10. Para terminar se dio click en Ok y Cerrar.

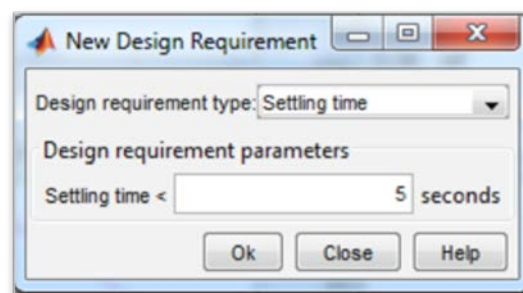


FIGURA 4.10. INGRESO DE PARÁMETROS DE DISEÑO

Después de haber ingresado el parámetro de diseño, pudo verse en la gráfica del LGR, una recta que separa el área en dos zonas una blanca y una amarilla; ésta representa el tiempo de estabilización, en la figura 4.11 puede verse lo descrito en detalle amplificado.

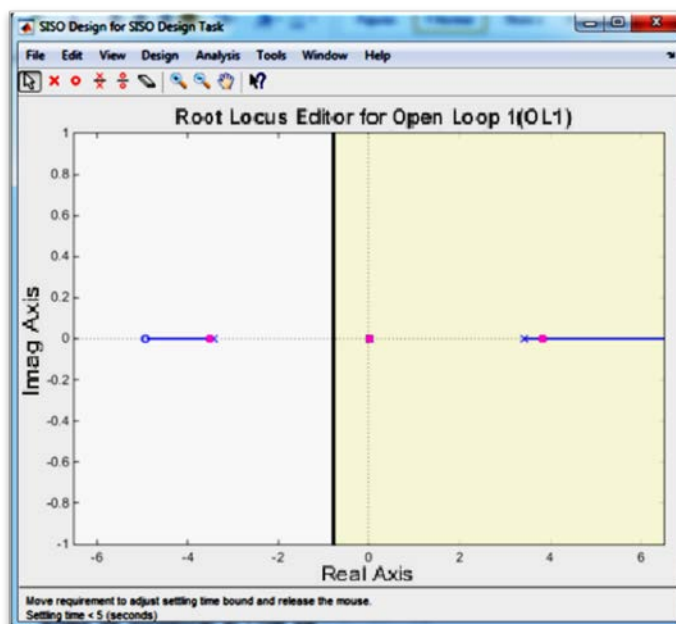


FIGURA 4.11. GRÁFICA DEL LGR CON LA RESTRICCIÓN DE DISEÑO.

En la gráfica de respuesta al perturbación impulso a la salida (figura 4.7) se ingresaron las restricciones de diseño para el error de estado estable de la señal, de la misma forma que en la gráfica de LGR, se da click derecho, se elige Design Requirements y New, la ventana que aparece es diferente y solo permite configurar rectas que definen rangos. Para la parte de estado estable (después de 5 segundos) se requería que la señal de salida esté dentro del rango ± 0.03 radianes, la configuración del límite superior puede verse en la figura 4.12, el límite inferior es similar, solo que las amplitudes son negativas.

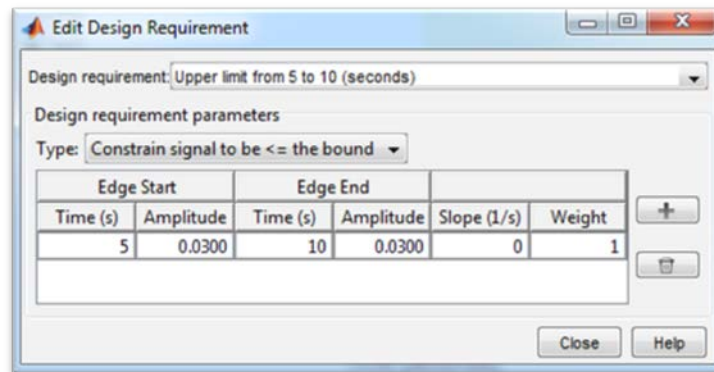


FIGURA 4.12. INGRESO DE RESTRICCIONES DE ERROR DE ESTADO ESTABLE.

Con tales restricciones, la gráfica de respuesta a la perturbación de impulso a la salida se ve como se muestra en la figura 4.13 (con vista ampliada).

Las líneas sirven de referencia, de tal forma que la señal debe estar confinada en esos límites.

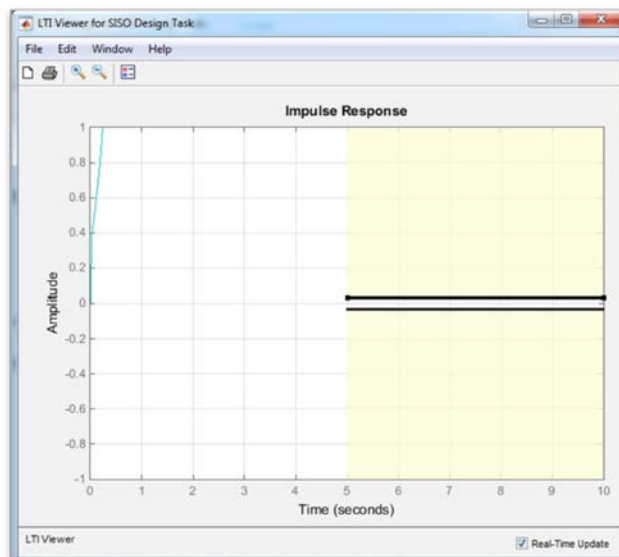


FIGURA 4.13. RESTRICCIONES DE ERROR DE ESTADO ESTABLE EN GRÁFICA DE RESPUESTA A PERTURBACIÓN IMPULSO.

Un primer paso para mejorar la respuesta temporal fue modificar la ganancia, pero ingresar valores de ganancia positivos en la pestaña del Editor de Compensador resultó inefectivo, pues la señal de volvió aún más inestable y el LGR no cambió. Al probar con ganancias negativas la respuesta mejoró y se pudo ver una modificación en el LGR de tal forma que permitía mover los polos dominantes hacia la izquierda. En la figura 4.14 puede verse el nuevo LGR para una ganancia del compensador C igual a -1 . A partir de este punto se puede apreciarse de forma más visible que la ganancia está ligada a la ubicación de los polos en el LGR.

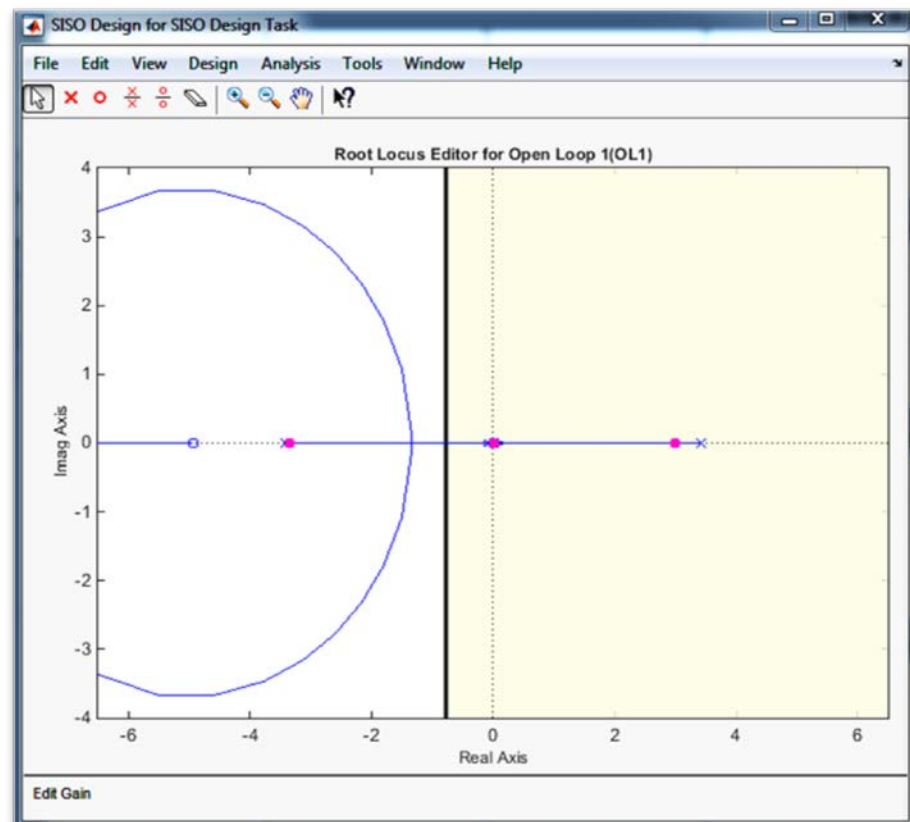


FIGURA 4.14. MODIFICACIÓN DEL LGR CON $C=-1$.

A partir de $C=-7.5$ aproximadamente, la respuesta se volvió estable, en la figura 4.15 se presenta la respuesta para una ganancia de $C=-8$, se puede ver que la señal se estabilizó pero fuera de los límites dispuestos para la parte estable.

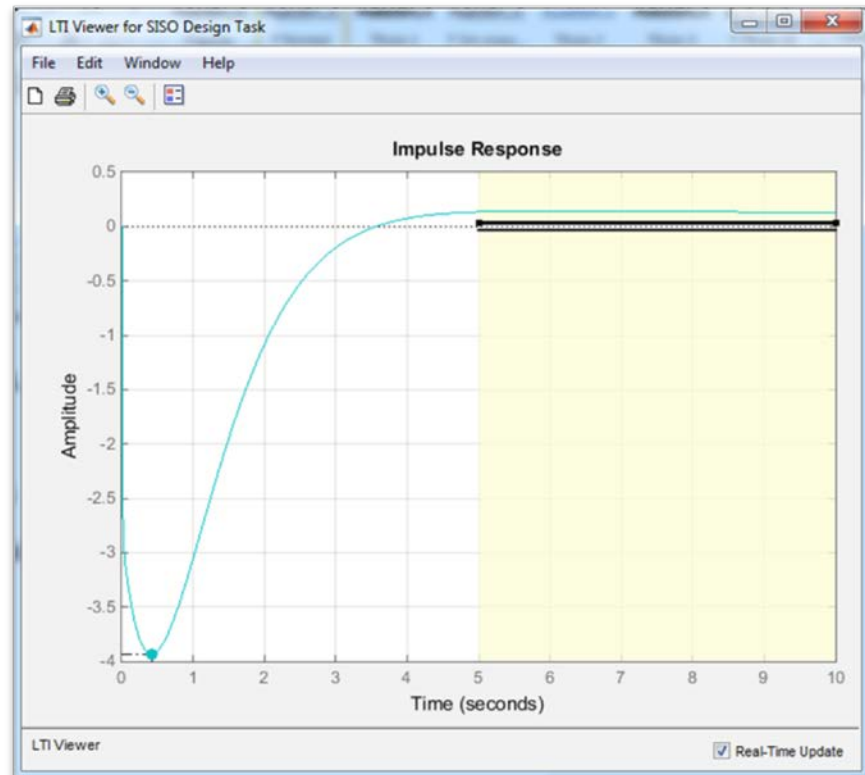
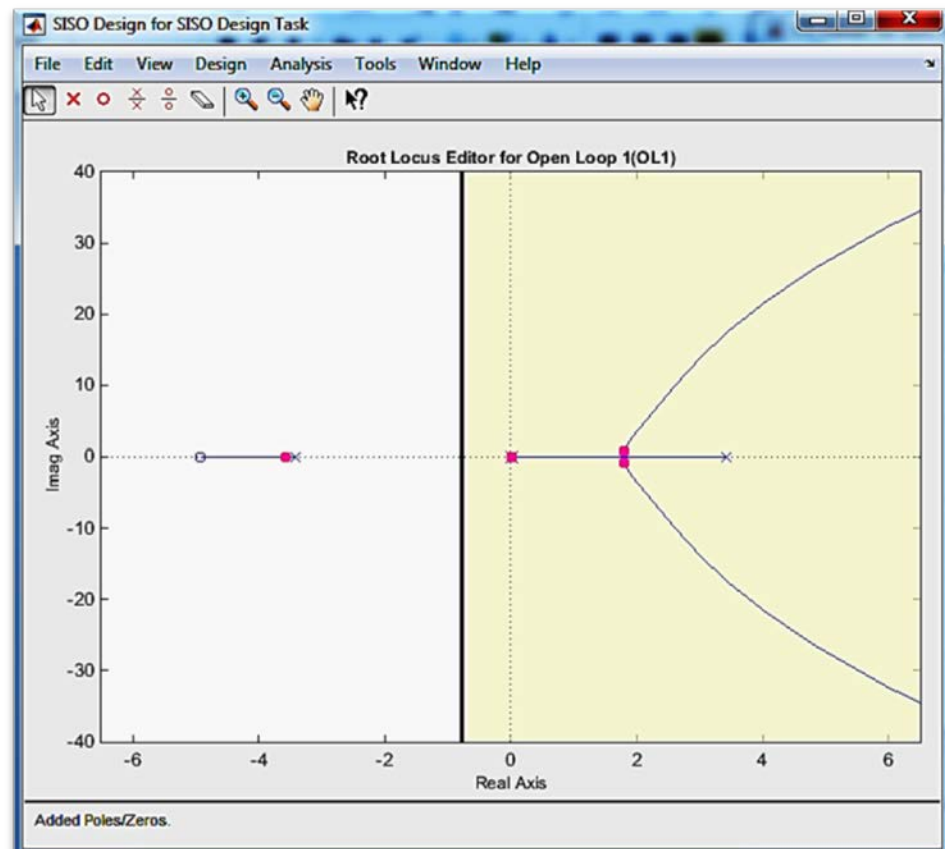


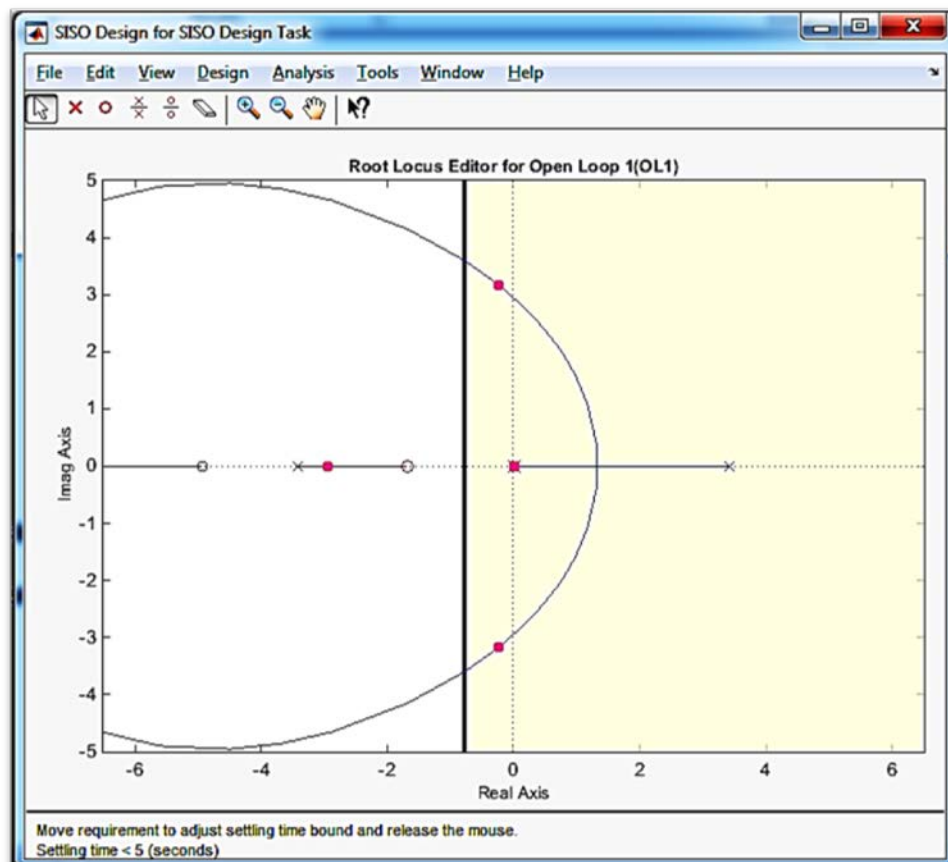
FIGURA 4.15. RESPUESTA A PERTURBACIÓN IMPULSO CON $C=-8$.

Para contrarrestar ese error de estado estable se necesitaba de acción integral, para introducir un integrador (polo en el origen) se dio click derecho ya sea en el área de "Poles/Zeros" de la pestaña de editor del compensador o en la gráfica de LGR, se eligió añadir polos o ceros y luego integrador, pero con esta adición el sistema se volvió inestable otra vez, es más el LGR se ubicaba ahora en el semiplano derecho como se ve en la figura 4.16.



**FIGURA 4.16. GRÁFICA DEL LGR CON UN INTEGRADOR
(POLO EN EL ORIGEN).**

Para atraer el LGR al semiplano izquierdo se necesitaba introducir un cero real, siguiendo el mismo procedimiento de la adición del integrador. Efectivamente el LGR estaba ahora mayormente ubicado en el semiplano izquierdo (figura 4.17), aunque la trayectoria pudo ser aún modificada desplazando el cero añadido.



**FIGURA 4.17. GRÁFICA DEL LGR CON UN CERO REAL
AÑADIDO.**

El efecto de la adición del cero real observado en la respuesta temporal se muestra en la figura 4.18, el sistema volvió a ser estable aunque oscilatorio y con tiempo de estabilización fuera de la restricción de los 5 segundos.

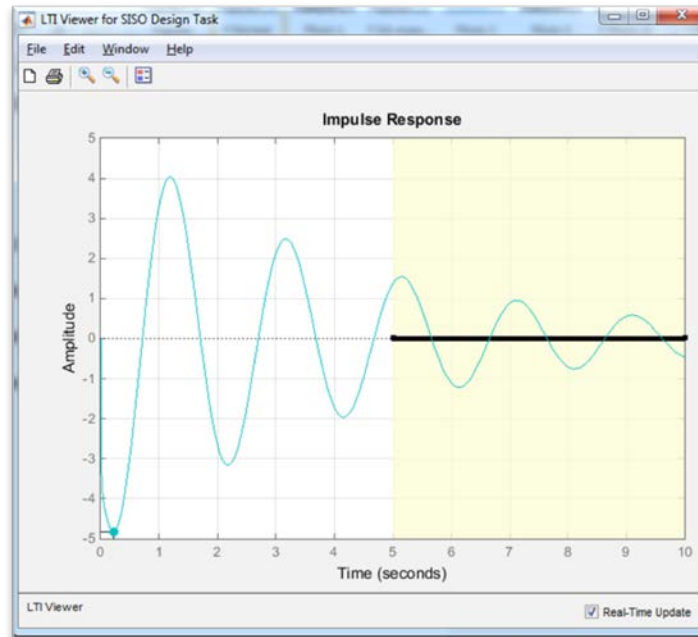


FIGURA 4.18. RESPUESTA A PERTURBACIÓN CON UN CERO REAL AÑADIDO.

Moviendo el cero hacia la derecha, se logró un LGR tal que se necesitaba menos ganancia para cumplir al menos con el criterio de error de estado estable, por ejemplo con el cero ubicado en -0.824 , el LGR queda de tal forma que al mover los polos dominantes lo suficiente para la respuesta temporal esté dentro de los límites de error de estado estable, la ganancia queda en -10.597 . La figura 4.19 muestra tanto el LGR como la respuesta temporal con esta configuración. En comparación con el LGR de la figura 4.17, con el cero ubicado en -1.68 , se necesitó una ganancia de -23.837 para lograr limitar el error de estado estable.

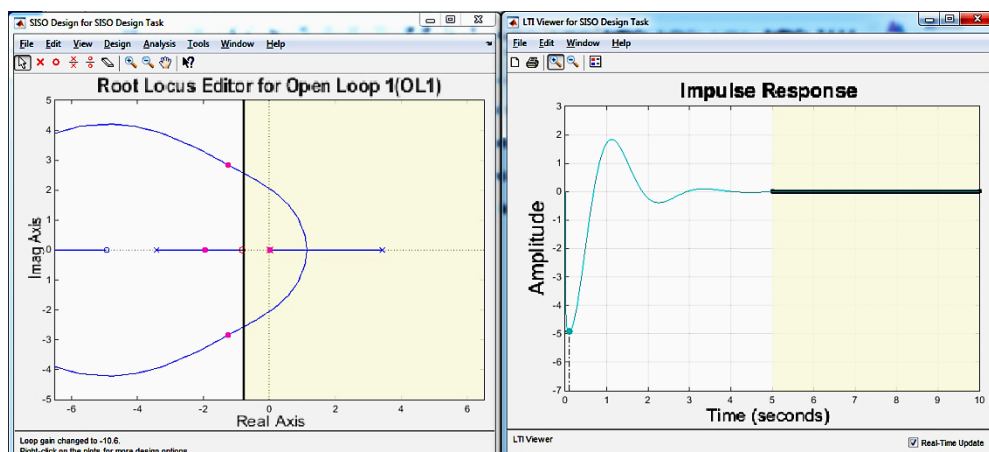


FIGURA 4.19. LGR Y RESPUESTA AL IMPULSO CON CERO EN -0.824 Y GANANCIA 10.597.

La magnitud de la ganancia está relacionada con el esfuerzo de control. Una ganancia demasiado alta, incluso si ayuda a mejorar la respuesta en un modelo puede que genere señales de control de una amplitud tal que pueda estar bastante fuera de rango de lo que pueda manejar el controlador real, el cual es limitado.

Luego de varios ajustes se obtuvo un controlador que presenta la forma $C = -2.3103 \cdot (1 + 4.3S / S)$, con un LGR como el mostrado en la figura 4.20 y una respuesta temporal ante una perturbación impulso como la mostrada en la figura 4.21.

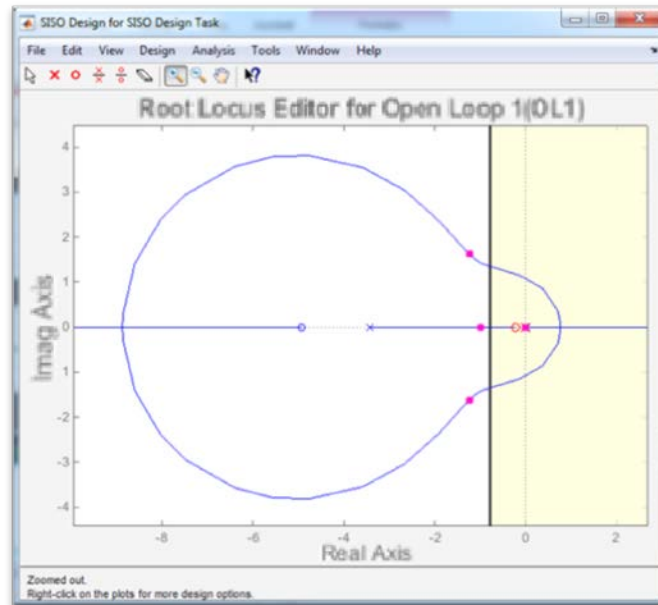


FIGURA 4.20. LGR PARA $C=-2.3103*(1 + 4.3S / S)$.

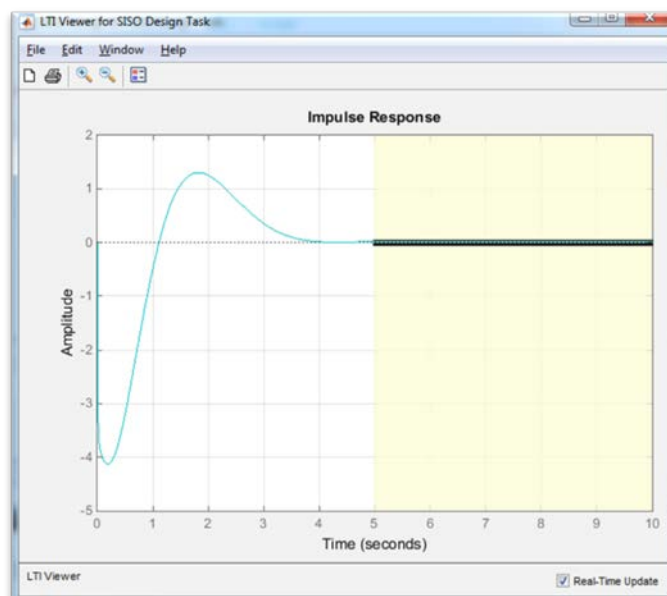


FIGURA 4.21. RESPUESTA AL IMPULSO PARA $C=-2.3103*(1 + 4.3S / S)$.

Una vez diseñado el lazo interno, se sintonizaron los valores del lazo externo, para cumplir con los parámetros de diseño del movimiento del carro.

El lazo externo cambia el punto de ajuste del segundo ante las perturbaciones para mantener tanto la posición lineal como angular en cero. Dado el diseño del lazo interno, los cambios de punto de ajuste no podían sobrepasar valores tales que provoquen la caída del péndulo, estos límites a su vez estaban relacionados con el cambio de control entre el auto-levantamiento y la estabilización. Por seguridad se eligió un valor pequeño (± 0.15 radianes) para el rango en el cual debe estar la salida del PID maestro, si esto no se lograba a través de ajustes de las ganancias, se podía aplicar saturación de la señal posteriormente.

El controlador interno obtenido con la ayuda de Sisotool estaba como una función de S, si se requería expresarlo de la forma estándar como ganancias K_p , K_i o K_d , se necesitaba exportar el controlador. Para ello, ubicado en la ventana principal de Sisotool, se dio click en Archivo, Exportar, elegir el Compensador C, y Exportar a Espacio de Trabajo. En Matlab se escribió lo siguiente en la ventana de comandos:


```
[Kp,Ki,Kd,Tf] = piddata(C)
```

Al ejecutar la línea de comandos anterior, el resultado fueron las ganancias del controlador PID de configuración paralela, puede notarse que el controlador resultante fue del tipo PI:

Kp =

-10.0448

Ki =

-2.3103

Kd =

0

Tf =

0

Dado que ya se contaba con las ganancias del lazo interno, el ajuste de las ganancias del lazo externo, podía llevarse a cabo en el modelo linealizado del sistema, en el entorno de Simulink.

Para ello se introdujeron dos bloques PID de la librería Continuos, y un par de bloques de suma-resta, al diagrama de bloques de la simulación del sistema linealizado de la figura 3.13. Además se introdujo un bloque de funciones que simula la perturbación impulso en la posición angular del péndulo. El diagrama de bloques con el control en cascada puede verse en la figura 4.22:

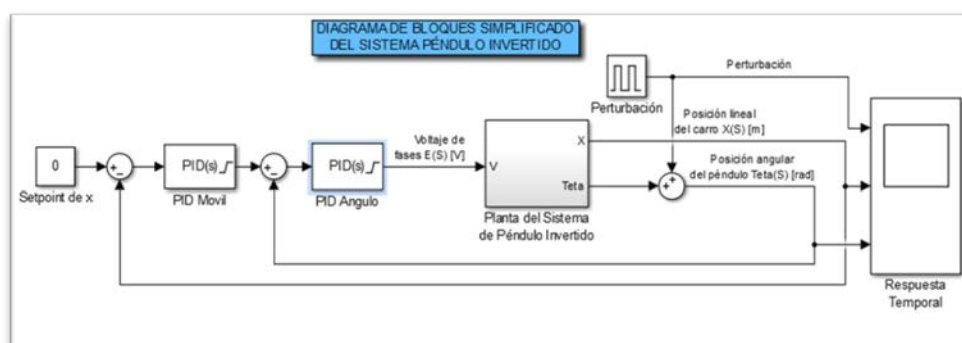


FIGURA 4.22. DIAGRAMA DE BLOQUES DE PLANTA LINEALIZADA CON CONTROL EN CASCADA

Se introdujeron las ganancias para al bloque PID interno (PID Ángulo) dejando en cero a las ganancias del bloque PID externo (PID Móvil) y se ejecutó la simulación para comprobar lo obtenido

mediante Sisotool, las gráficas obtenidas pueden verse en la figura 4.23.

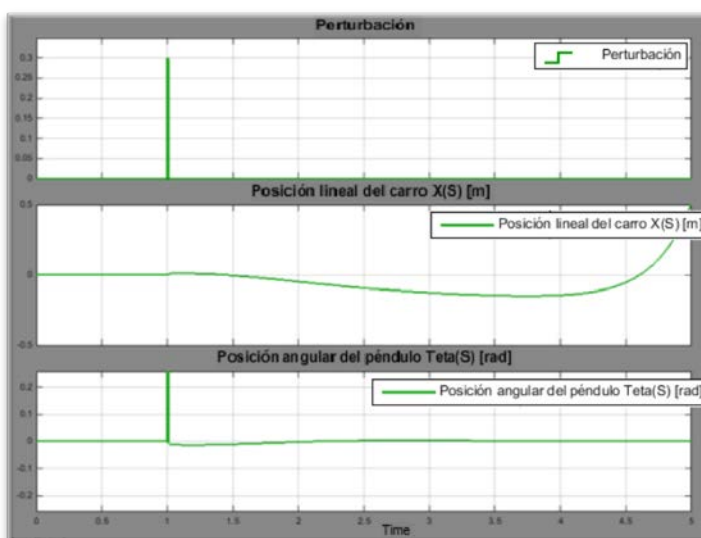


FIGURA 4.23. RESPUESTA A LA PERTURBACIÓN CON CONTROL INTERNO (P= -10.0448, I= -2.3103).

La posición angular, se recuperó de la perturbación como era de esperarse, la posición lineal se mostró inestable pues aún no se había tomado en cuenta el control de ésta última.

Para acercar el modelo simulado al real, se introdujeron ciertas restricciones, como el hecho de que de acuerdo a la ecuación 3.1 no se podía utilizar todo el rango de voltaje de referencia, lo cual en voltaje de fases se traduce a un rango utilizable de ± 15.73 Vdc.

Estos límites de saturación se introdujeron en el bloque PID bajo la pestaña PID avanzado, activando la casilla de limitar la señal e ingresando los valores de límite superior e inferior.

Por lo general, era de esperarse el tener que volver a ajustar los valores del PID interno al habilitar el PID externo. Se añadió primero acción proporcional en el controlador maestro sin modificar los valores del controlador esclavo, y se lo comparó con el resultado de duplicar la acción proporcional para el lazo interno (figura 4.24). Adicionalmente como fue mencionado en párrafos anteriores, el rango de salida del PID maestro fue de ± 0.15 radianes.

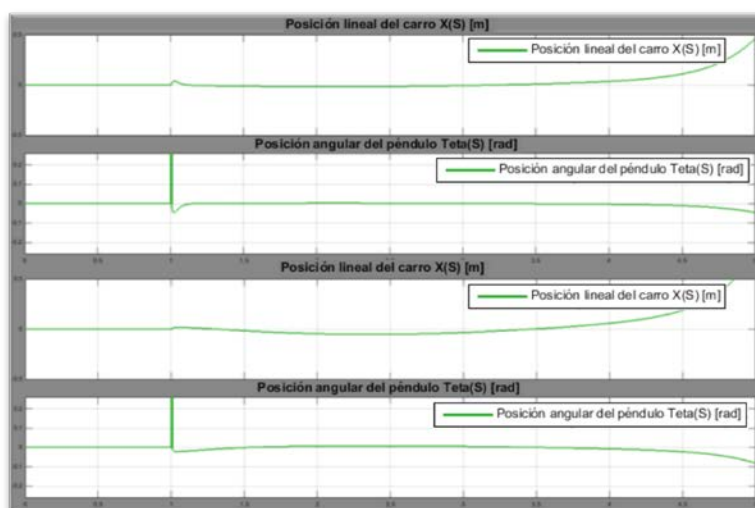


FIGURA 4.24. RESPUESTA A LA PERTURBACIÓN CON EL CONTROL INTERNO: CON ACCIÓN PROPORCIONAL DUPLICADA (SUPERIOR) Y SIN MODIFICAR (INFERIOR).

Aumentando la acción proporcional del control interno disminuyó un poco la desviación de la posición lineal y angular que se presenta hacia los 5 segundos hasta aproximadamente la mitad, en la figura 4.25 se ve el resultado de únicamente cuadruplicar la acción proporcional del control interno hasta $P=-80$.

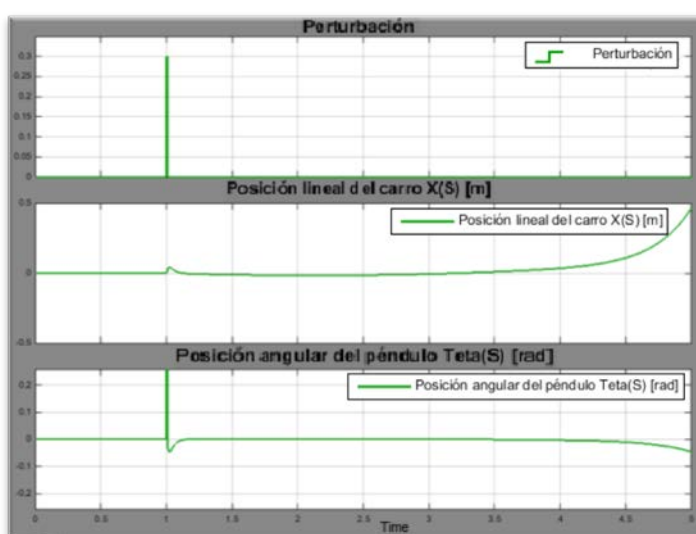
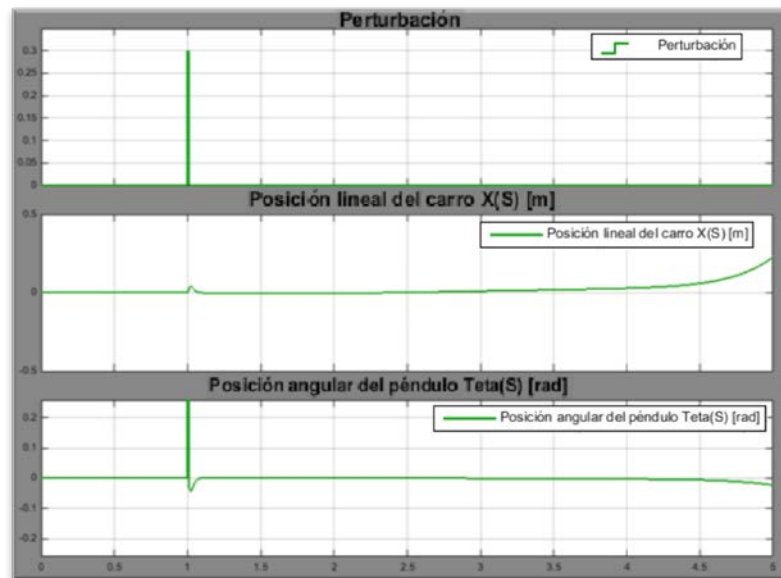


FIGURA 4.25. RESPUESTA A LA PERTURBACIÓN CON ACCIÓN PROPORCIONAL $P=-80$ EN EL CONTROL INTERNO

Como puede verse en esta etapa la sintonización de las ganancias de los controladores PID se hizo de manera empírica sobre la simulación linealizada de la planta del sistema de péndulo invertido. Por un proceso de prueba y error se llegó a que una relación de $I=P/2$ y $D=P/4$ genera una respuesta como la mostrada en la figura

4.26 con menor desviación tanto en la posición lineal como en la angular.



**FIGURA 4.26. RESPUESTA A LA PERTURBACIÓN CON $P=-80$,
 $I=-40$ Y $D=-20$ EN EL CONTROL INTERNO**

De esta forma se obtuvieron las siguientes ganancias:

PID interno: $P = -80$, $I = -40$, $D = -20$

PID externo: $P = 0.1$, $I = D = 0$

Estos valores luego fueron puestos a prueba junto con el auto-levantamiento en la simulación no lineal de la planta.

DISEÑO DEL CONTROLADOR DE AUTO-LEVANTAMIENTO:

Para el auto-levantamiento se tomó como referencia el trabajo de Anström y Furuta [23], donde el auto-levantamiento se realizó por el método de inyección de energía. Dado que esta ley de control fue diseñada para el tipo de péndulo invertido del mismo Furuta, no se tomó en cuenta el desplazamiento lineal, pues en el equipo mencionado el movimiento es circular y prácticamente infinito, en el péndulo invertido en un carro, la carrera es finita, por lo tanto en lo posible se debían ajustar los parámetros de esta ley para que el valor de x no sobrepase los límites físicos de los apoyos de los rieles. La ley de control es la siguiente:

$$u = \text{sat}_{ng}(k(E - E_0)\text{sign}(\dot{\theta}\text{Cos}\theta)) \quad \text{Ec. 4.11}$$

Donde $(E - E_0)$ calcula el error entre la energía medida versus la energía del punto de ajuste o a donde se quiere llegar, que es la energía cuando el péndulo está en su posición vertical invertida, la energía que se toma en cuenta es la del péndulo.

La función $\text{sign}(\dot{\theta}\text{Cos}\theta)$ se encarga de realizar el cambio de dirección de la señal de control aplicada, su argumento está en

función de la velocidad angular y el coseno de la posición angular del péndulo, por lo tanto cuando el péndulo está en su posición de equilibrio estable, el valor de esta función es cero, para iniciar la rutina por tanto se necesita de alguna excitación inicial.

La función sat_{ng} satura la salida del controlador entre los valores $\pm ng$, y está definida por las restricciones del sistema, la máxima energía que el sistema está físicamente habilitado a dar, en este caso es el voltaje máximo con el que se puede trabajar.

La constante k define las zonas en las que el control se comporta de forma lineal o no lineal, cuando el error es alto la amplitud de la variación de la función es alta, cuando está cercana al punto de ajuste (posición invertida) se comporta como un controlador lineal, de esta forma al estar saturada la señal, mientras más alto sea k , la región en la que se comporte similar a un controlador “bang-bang” u “on-off” será mayor.

La energía del péndulo medida estaba en función de constantes como inercia, masa, gravedad y longitud del péndulo, y variables como velocidad angular y posición angular del péndulo, las cuales eran las entradas al controlador, tanto para calcular la energía,

como para calcular el argumento de la función signo descrita anteriormente. Está expresada en la siguiente ecuación:

$$E = 0.5I(\dot{\theta})^2 + mgl(\cos\theta - 1) \quad \text{Ec. 4.12}$$

El punto de ajuste o energía a alcanzar (cuando el péndulo está en su posición vertical invertida) está dado por:

$$E_0 = mgl \quad \text{Ec. 4.13}$$

Este control era más fácil de analizar en Simulink, introducido en forma de bloque de funciones de Matlab, dentro de la librería de Funciones Definidas por el Usuario.

Al introducir el bloque se le dio doble click, abriendo el editor de funciones de Matlab. Siguiendo las directrices del procedimiento descrito en Mathworks [24], se creó el siguiente código para el bloque de función de auto-levantamiento:

```
function u = auto_lev_f(teta,w,ng,K)
%#codegen

m=0.13; %masa puntual del péndulo en Kg
g=9.81; %gravedad en m/s^2
l=0.43; %longitud del péndulo en metros
I=m*(l^2); %Inercia del péndulo en Kg x m^2
```

```

%Energía del péndulo
E=0.5*I*(w^2)+m*g*l*(cos(teta)-1);

%Energía en la posición invertida
E0=m*g*l;

%Ecuación o ley de control del auto-levantamiento (método de
Furuta)
c=K*((E-E0)*sign(w*cos(teta)));

%saturación de la señal de control en el rango +ng -ng
u= min(ng, max(-ng, c));

```

Una vez guardada está función, el bloque mostró la apariencia que ilustra la figura 4.27:

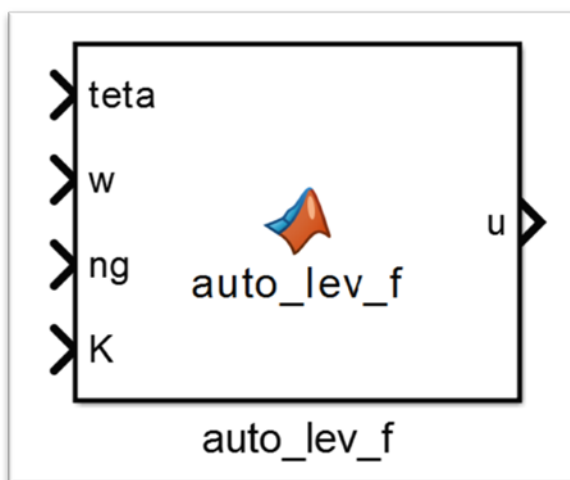


FIGURA 4.27. BLOQUE DE LA FUNCIÓN DE AUTO-LEVANTAMIENTO.

Aunque la función necesitaba 4 entradas, dos de ellas eran señales en tiempo real durante la simulación mientras que las otras dos

restantes eran parámetros ingresados por usuario, esta distinción se estableció seleccionando “Edit Data” en el editor de funciones de Matlab, en la ventana que aparece se podía ver una lista de las entradas y salidas del bloque función, se seleccionaron las que se deseaba dejar como parámetros, y a la derecha dentro del menú desplegable de “Scope” se seleccionó “Parameter” (figura 4.28).

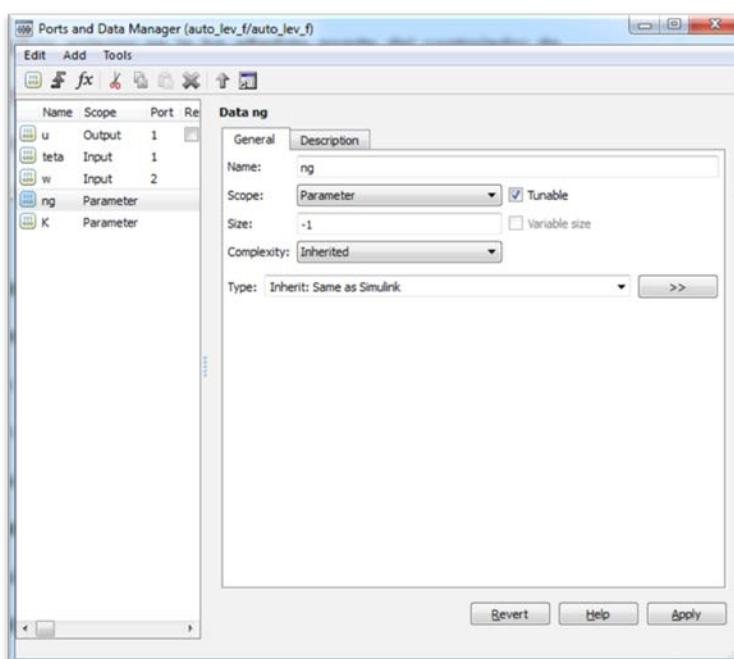


FIGURA 4.28. CONFIGURACIÓN DE ENTRADAS Y SALIDAS DEL BLOQUE DE LA FUNCIÓN DE AUTO-LEVANTAMIENTO.

Con los cambios anteriores el bloque se vio como en la figura 4.27, pero sin las entradas de bloque para ng y K. Para poder ingresar los parámetros como ng y k en el bloque de auto-levantamiento se creó

una máscara, para ello se dio click derecho sobre el bloque y en las opciones desplegadas de Mask se eligió “Create Mask”, esto abrió una ventana de configuración de la máscara del bloque. En la pestaña de parámetros y diálogo se dio click en la opción “Edit” de la lista de “Controls” dos veces, con las entradas de edición ya visibles en la parte central de la ventana de configuración, se dio click en sus recuadros correspondientes a “Prompt” y “Name” para darles sus nombres respectivos como se observa en la figura 4.29:

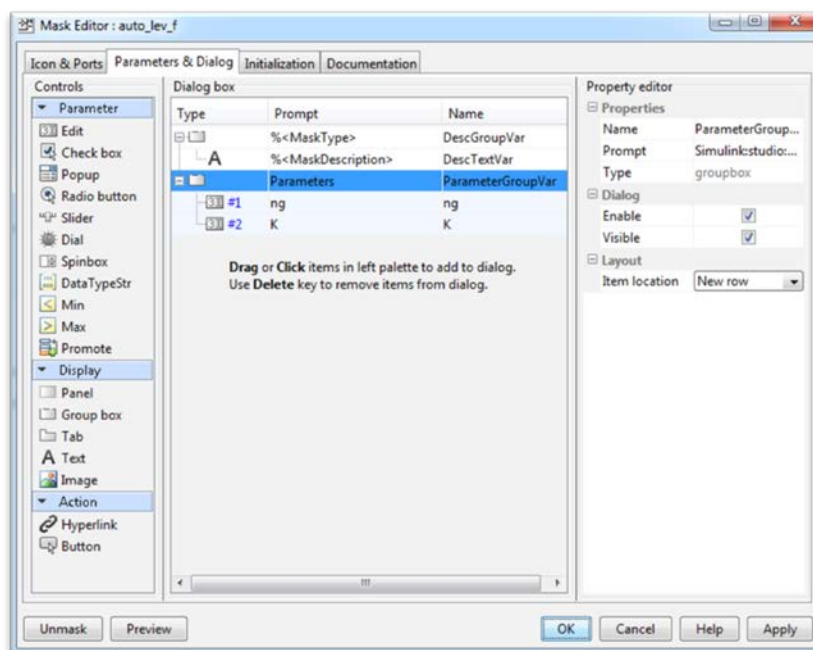


FIGURA 4.29. CONFIGURACIÓN DE ENTRADAS TIPO PARÁMETRO DE LA MÁSCARA DEL BLOQUE DE AUTO-LEVANTAMIENTO.

Por último, en la pestaña de documentación, se introdujo un título y un breve texto explicativo (ver figura 4.30):

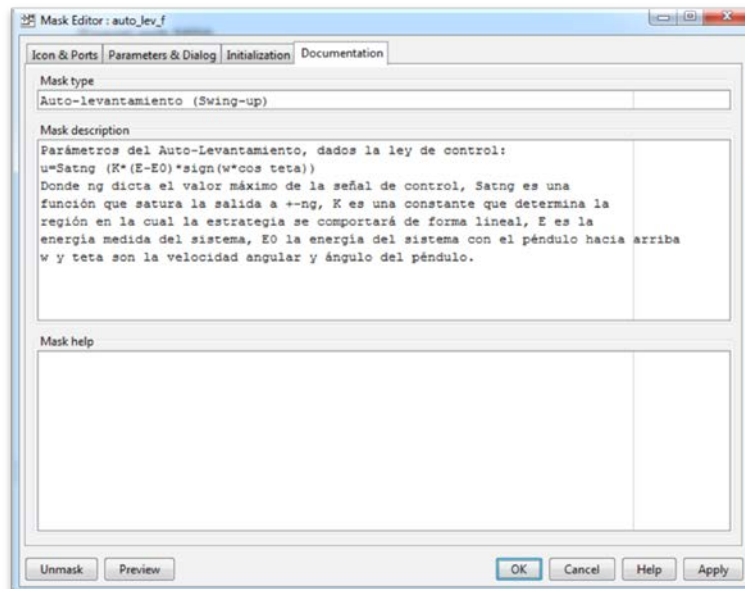


FIGURA 4.30. INTRODUCCIÓN DE TÍTULO Y TEXTO DE LA MÁSCARA DEL BLOQUE DE AUTO-LEVANTAMIENTO.

Finalmente se aplicaron los cambios y se presionó Ok, de esta manera el bloque función de auto-levantamiento quedó configurado de tal forma que al dar doble click sobre éste, aparezca la ventana de diálogo donde se deben introducir los valores de los parámetros ng y K, como se observa en la figura 4.31:

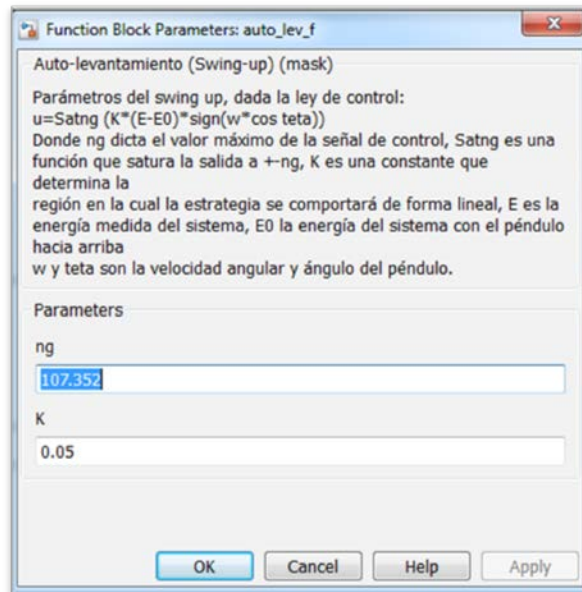


FIGURA 4.31. VISTA DE LA MÁSCARA DEL BLOQUE DE LA FUNCIÓN DE AUTO-LEVANTAMIENTO.

El control de auto-levantamiento se probó con el modelo no lineal. Para que la señal de control no se quede en cero, se introdujo un ángulo ligeramente menor a pi (3 radianes) en las condiciones iniciales del bloque integrador correspondiente al ángulo teta. El diagrama de bloques armado se ve en la figura 4.32.

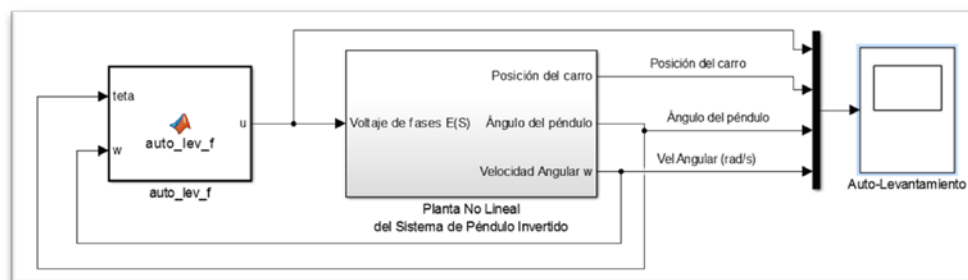


FIGURA 4.32. DIAGRAMA DE BLOQUES COMPLETO DEL SISTEMA NO LINEAL PARA EL AUTO-LEVANTAMIENTO.

Como puede verse en la imagen anterior, al modelo no lineal armado en el capítulo anterior se añadió aparte del controlador de auto-levantamiento, una salida extra que es la de velocidad angular, esto se logró entrando al subsistema (doble click sobre el bloque) e introduciendo un bloque de salida de la librería “Sinks”, se conectó ese bloque a la salida nombrada como “Tetapto”, y se etiquetó la salida, por último ya fuera del bloque subsistema se conectó al bloque de visualización a través del “Mux”.

Al final del capítulo 3 se mencionó la particularidad de que pasado los 2.4 V en la entrada de voltaje de diferencia no se podían hacer cambios de dirección repentinos, eso sumado a la zona muerta extendida del driver reducía el voltaje de fases utilizable. La restricción de voltaje de fases máximo dictaba el valor máximo de θ . Tomando la ecuación 3.1, para un voltaje de referencia de 2.4

V , el voltaje de fases máximo y por ende valor máximo de n_g fue calculado en aproximadamente 15.73 V.

Para la control de auto-levantamiento debían darse dos condiciones: un valor mínimo del ángulo, es decir cercano a cero y que la velocidad angular en ese mismo punto del tiempo sea cero, lo cual garantizaría un cambio suave entre las estrategias de control cuando el péndulo entre en la zona de equilibrio (la zona en la que se activa el control cascada de mantenimiento de la posición invertida). Ambas condiciones debían darse dentro del tiempo acordado en los requerimientos de diseño (menos de 30 segundos).

El procedimiento para encontrar los valores ideales de n_g y K , fue de prueba y error corriendo simulaciones y analizando las curvas obtenidas, interesaba principalmente observar las curvas de posición angular y velocidad angular del péndulo. Se partió con valores bajos de n_g y altos de K para tener un control de tipo “bang-bang”, una vez encontrado un valor de n_g para el cual se cumplían las condiciones descritas en el párrafo anterior, se probó disminuyendo K para ver si mejoraba la acción del control cuando el péndulo estaba cerca de la zona de equilibrio.

Por ejemplo sea $K=10$ y $ng=0.1$, al ejecutar la simulación para un tiempo de 40 segundos se obtuvo la gráfica de la figura 4.33. Tomar en cuenta que de aquí en adelante la curva celeste representa posición angular del péndulo, la roja la velocidad angular del péndulo, la purpura la posición del carro y la verde la señal de control.

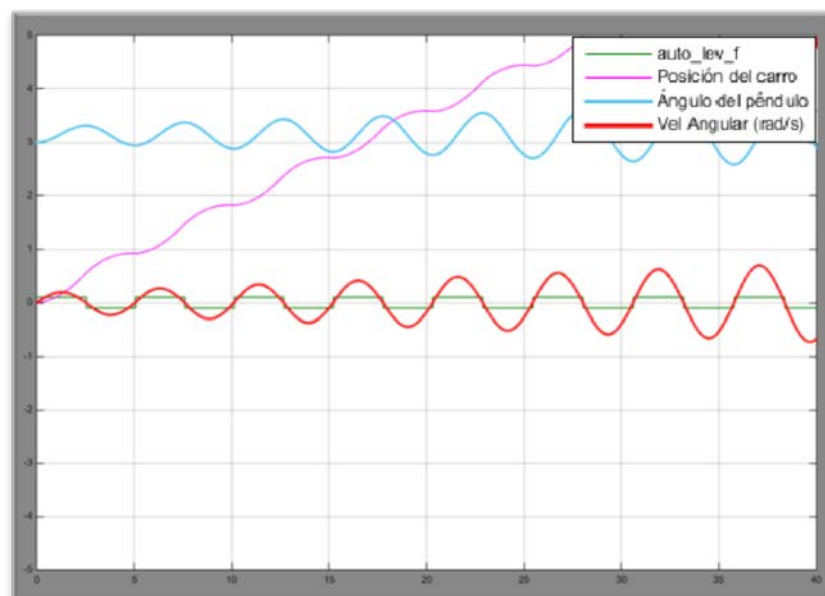


FIGURA 4.33. AUTO-LEVANTAMIENTO PARA $NG=0.1$ Y $K=10$.

En la imagen anterior se puede ver la oscilación creciente de la posición angular, pero no con la suficiente rapidez como para alcanzar valores cercanos a cero en menos de 30 s, se aumentó

entonces el valor de n_g , se eligió un valor de 0.5, generando la siguiente gráfica:

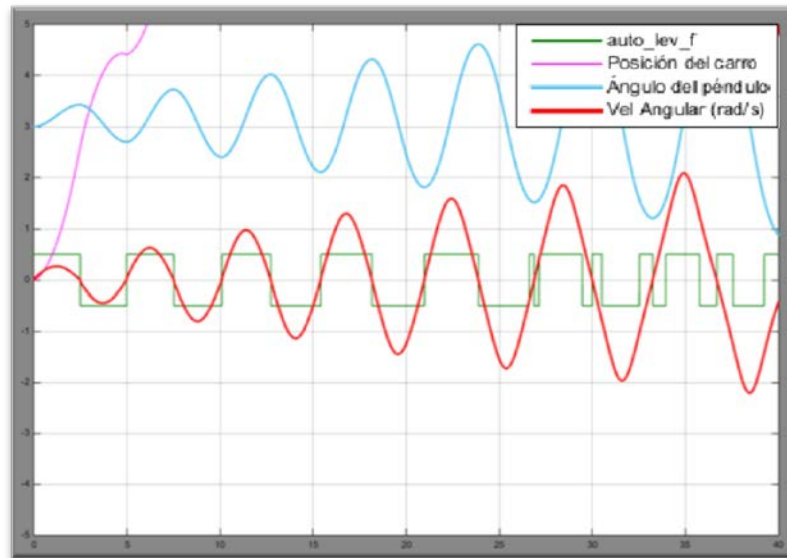


FIGURA 4.34. AUTO-LEVANTAMIENTO PARA $n_g=0.5$ Y $K=10$.

La tasa de crecimiento de la amplitud de las oscilaciones aumentó, pero no aun lo suficiente para llegar a las condiciones, aunque éstas parecían darse casi al final a los 40 s. Se probó con un valor de $n_g=1$, obteniendo el resultado mostrado en la figura 4.35.

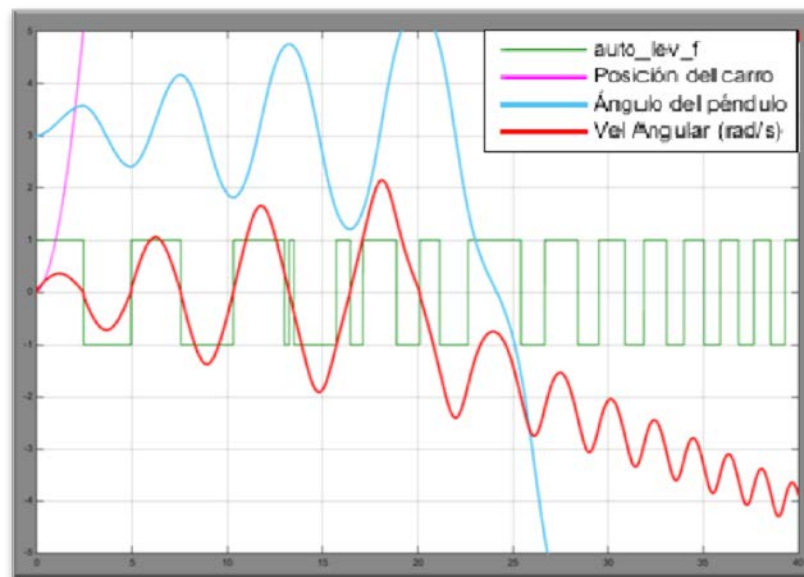


FIGURA 4.35. AUTO-LEVANTAMIENTO PARA NG=1 Y K=10.

Esta vez, se llegó en un punto a tener ángulo cero antes de los 30 s, pero no con velocidad cero, pasaría por la zona de equilibrio pero con una velocidad angular de aproximadamente -0.75 rad/s, lo cual dificultaría el cambio de estrategia de control como se había mencionado anteriormente.

Se aumentó n_g en menores incrementos para tratar de cumplir con las condiciones en la tercera oscilación entre los 15 s y 20 s, y después de algunas pruebas más, se lo logró con un $n_g=1.25$, a continuación se presenta la gráfica resultante (figura 4.36).

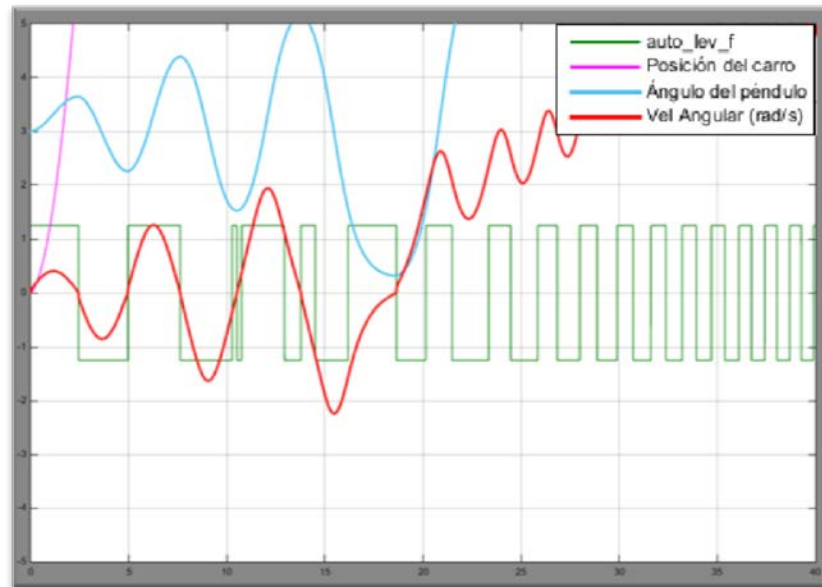


FIGURA 4.36. AUTO-LEVANTAMIENTO PARA $NG=1.25$ Y $K=10$.

Amplificando la imagen anterior cerca de la zona donde se cumplen las condiciones, puede verse que en un poco más de 18.6 s, el péndulo alcanzó velocidad angular cero con una posición angular mínima de aproximadamente 0.32 radianes.

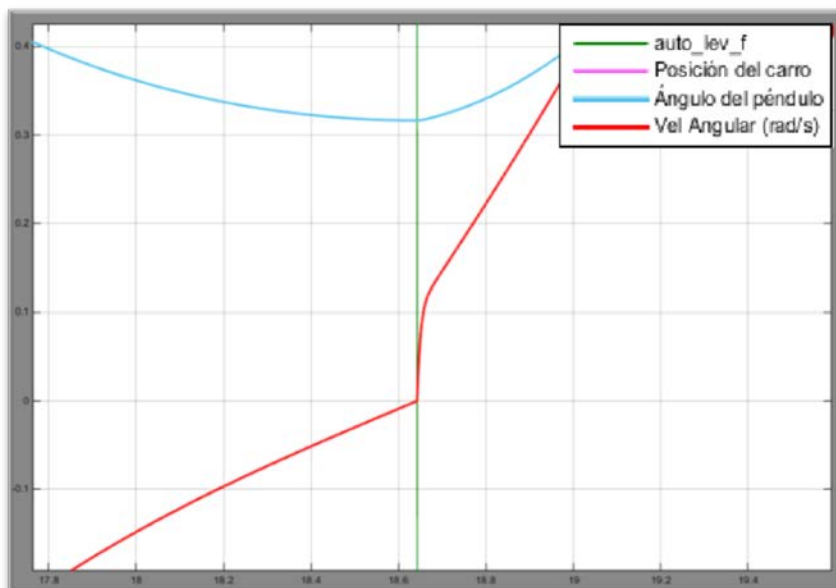


FIGURA 4.37. DETALLE DEL AUTO-LEVANTAMIENTO PARA NG=1.25 Y K=10.

Se probó con disminuir el valor de K, para mejorar las condiciones en la transición, sin embargo no es sino hasta $K=2$ que recién se observaron zonas en la señal de control no afectadas por la saturación de ng , una vista ampliada en el punto de transición (ver figura 4.38) deja ver que aumentó ligeramente la posición angular (0.33 radianes), es decir que el péndulo llegó más alejado de la posición de control de equilibrio, lo cual es negativo para el cambio de control.

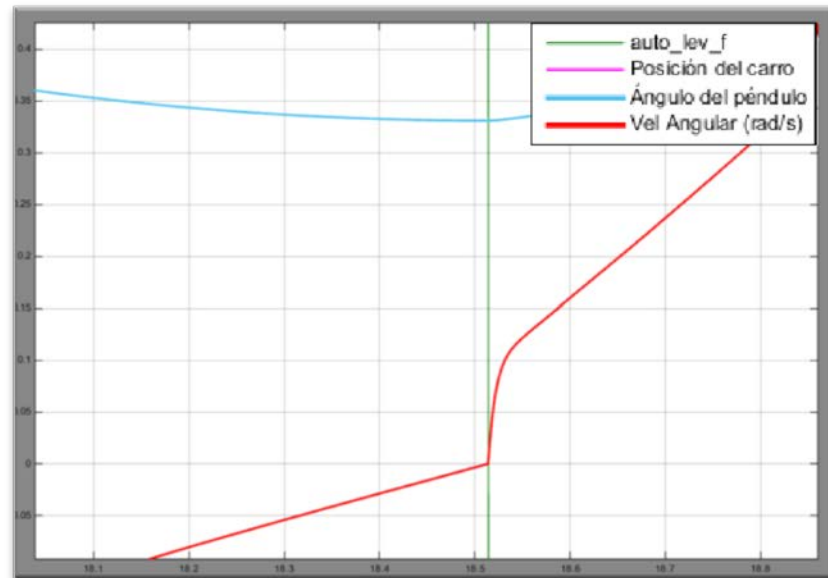


FIGURA 4.38. AUTO-LEVANTAMIENTO PARA NG=1.25 Y K=2.

Disminuyendo aún más la ganancia K hasta 1.5, aumentó el valor del ángulo (0.42 radianes) con el cual el péndulo llegaba a velocidad cero (ver figura 4.39).

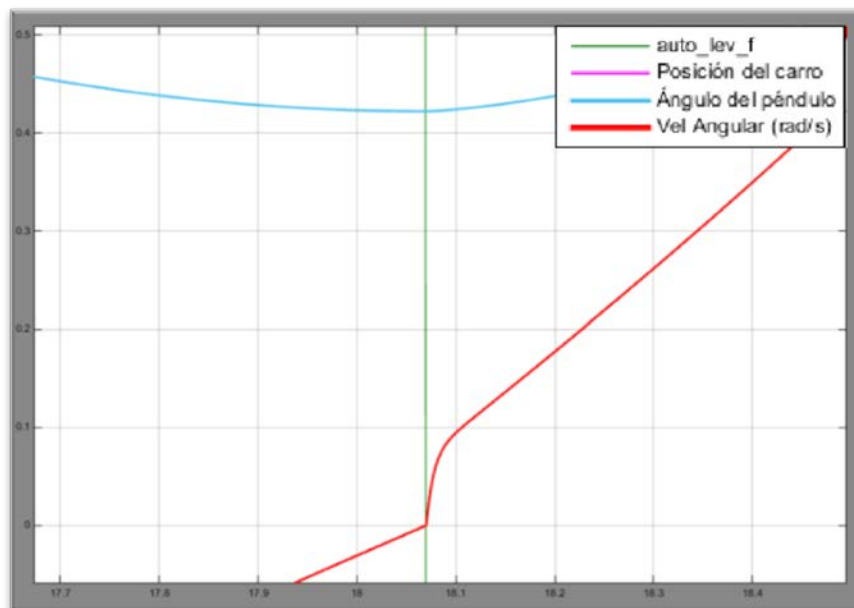


FIGURA 4.39. AUTO-LEVANTAMIENTO PARA $NG=1.25$ Y $K=1.5$.

De lo anterior se concluyó que era preferible mantener un K lo suficientemente alto de tal manera que todo el control de auto-levantamiento sea de tipo “bang-bang”, lo cual se cumplió para $K=3$.

4.4 Simulación del controlador

La simulación o validación de los datos obtenidos del proceso de diseño del controlador (esto es el conjunto formado por el control de auto-levantamiento y el control de estabilización) se la llevó a cabo en el modelo no lineal, desde la etapa de estabilización, pasando por la transición y la estabilización del péndulo. Luego se realizaron

ajustes de ser necesarios y las ganancias así obtenidas luego fueron aplicadas en el modelo real obtenido en la implementación del capítulo 5 y sus resultados analizados en el capítulo 6.

En toda la rutina se tuvo dos estrategias de control, la del auto-levantamiento y la de mantenimiento de posición o equilibrio, pero estas estrategias eran exclusivas la una de la otra y la señal de control solo debía provenir de una de las dos dependiendo de las condiciones del sistema. Mientras el péndulo estuviera colgando hacia abajo la señal de control debía ser la dictaminada por la función de auto-levantamiento, mientras que cuando el péndulo estuviera dentro de un cierto rango alrededor de la posición invertida (0 radianes) se debía producir un cambio para que la señal de control sea la del control en cascada. Con los resultados de la simulación del auto-levantamiento, se estableció que el mejor momento para realizar el cambio de control debía ser cuando se alcanzaba un ángulo algo mayor al obtenido en la simulación del auto-levantamiento (± 0.32 radianes). Se decidió realizar la transición o cambio de control cuando el ángulo estuviera dentro del rango de ± 0.4 radianes.

Se armó un subsistema que tuviera como entrada las dos señales de control (la de estabilización y la de auto-levantamiento), y el ángulo para saber en qué momento hacer el cambio, lo cual fue evaluado con un bloque de intervalo (librería "Logic and Bit Operations"), si el ángulo estaba fuera del intervalo de estabilización (± 0.4 radianes) el bloque emitía señal falsa (valor 0), caso contrario emitía una señal de valor verdadero (valor 1). La señal booleana del bloque de intervalo comandaba la acción de un selector de dos posiciones (librería "Signal Routing") que dejaba pasar a una de las dos señales conectadas a sus terminales en los extremos de acuerdo al resultado de comparar con cero la señal proveniente del bloque de intervalo, por defecto si la señal booleana era mayor que cero o verdadera se dejaba pasar la señal conectada en el terminal superior, caso contrario pasaba la señal del terminal inferior.

Bajo dicha lógica, la señal de control de estabilización era la que debía ser conectada en el terminal superior, y la de auto-levantamiento en el terminal inferior. Para prevenir que la acción integral de los bloques PID no se acumulara cuando aún no entraba en actuación el control de estabilización, se activó la opción de "Reset", de tal manera que se encere el integrador mientras no se lo

necesite. Para ello se requería una señal que se mantenga mayor que cero mientras el péndulo no esté en el rango de estabilización, esto se consiguió fácilmente negando la señal del bloque de intervalo.

Por pulcritud se creó un subsistema de estos elementos, etiquetando cada señal, entrada y salida debidamente, la estructura interna del bloque subsistema Selector de Control puede verse en la figura 4.40:

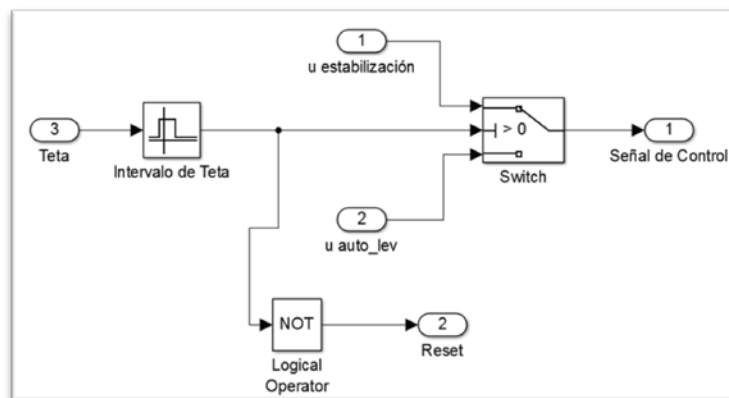


FIGURA 4.40. ESTRUCTURA INTERNA DE BLOQUE SELECTOR DE CONTROL.

El diagrama completo puede observarse en la figura 4.41:

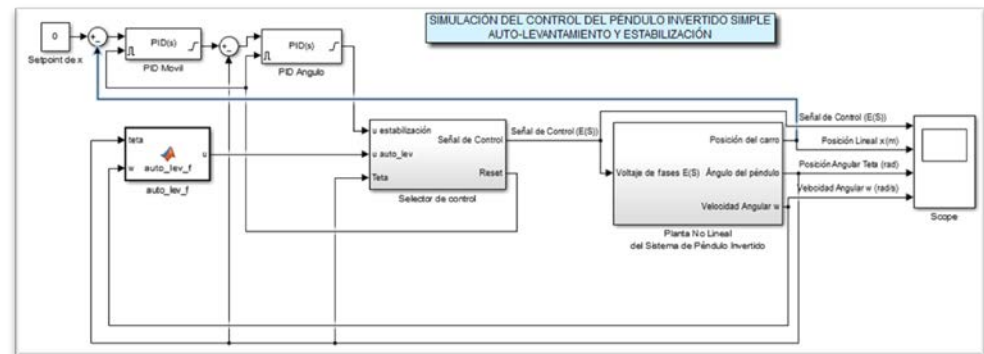


FIGURA 4.41. DIAGRAMA DE BLOQUES DE PLANTA NO LINEAL CON CONTROL DE AUTO-LEVANTAMIENTO Y ESTABILIZACIÓN.

La gráfica para la señal de control, posición lineal, posición angular y velocidad angular obtenida al ejecutar el modelo no lineal controlado bajo los parámetros obtenidos en el proceso de diseño se presenta en la figura 4.42.

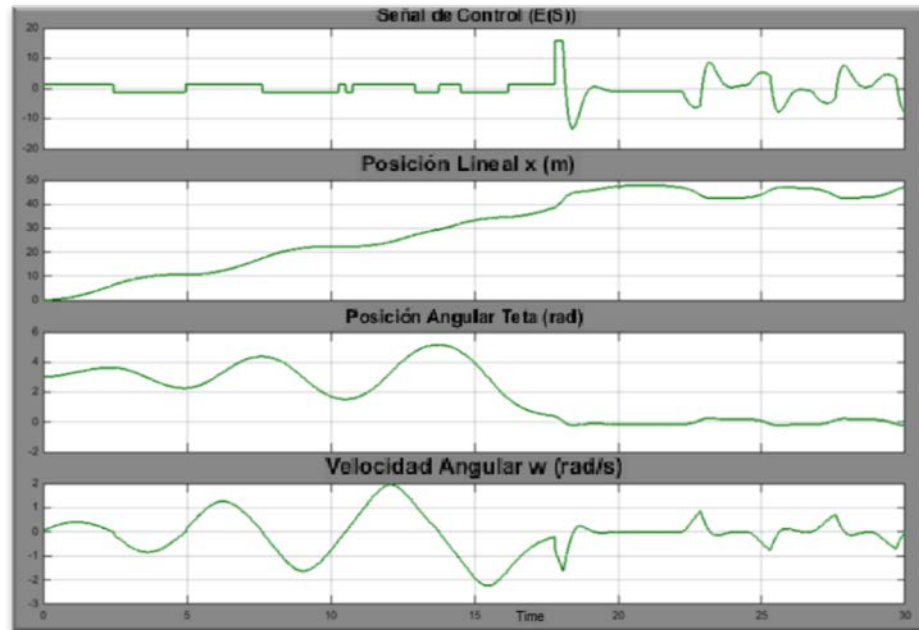


FIGURA 4.42. GRÁFICAS DE RESPUESTA DEL AUTO-LEVANTAMIENTO, TRANSICIÓN Y ESTABILIZACIÓN EN EL MODELO NO LINEAL.

Como puede observarse en la figura anterior, se logró balancear el péndulo desde su posición estable con el control de auto-levantamiento, produciéndose la transición alrededor de los 16 y 17 segundos, luego de lo cual toma el mando el control en cascada para la estabilización, aunque éste presentó cierto nivel de oscilación, lo cual se corrigió añadiendo las acciones integral y derivativas en el bloque PID externo que solo tenía acción proporcional ($P=0.1$).

Después de un proceso de prueba y error, con $I = 0.05$ y $D = 0.2$, la señal en estado estable de la posición angular mejoró, cumpliendo con las restricciones de diseño.

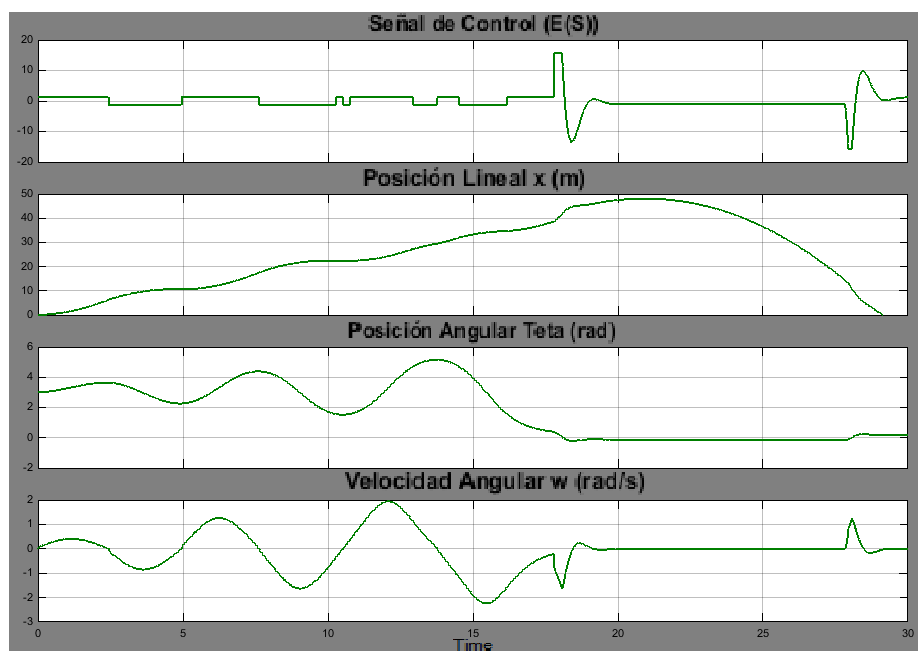


FIGURA 4.43. GRÁFICAS DE RESPUESTA DEL AUTO-LEVANTAMIENTO, TRANSICIÓN Y ESTABILIZACIÓN EN EL MODELO NO LINEAL CON ACCIÓN INTEGRAL Y DERIVATIVA EN PID EXTERNO.

Una discordancia notable con lo que debiera ser el control luego aplicado al modelo real fue el comportamiento de la posición lineal del carro o móvil, que está completamente fuera de rango, al punto que en la figura 4.43 puede verse que durante la etapa de auto-

levantamiento el carro estaba tan alejado (aproximadamente 50 m) que cuando se activaba el control de estabilización éste trataba a partir de los 20 segundos regresar al centro, en teoría esto volvería inestable la aplicación de éste control en el modelo real que tiene un límite físico de ± 0.5 m, lo cual fue examinado y explicado posteriormente en el análisis de resultados y conclusiones.

CAPÍTULO 5

5. IMPLEMENTACIÓN DEL CONTROL EN UN ENTORNO GRÁFICO DE PROGRAMACIÓN

Una vez obtenidos y simulados los controladores y sus parámetros en el capítulo anterior, estos fueron aplicados al sistema o equipo real utilizando el entorno gráfico de programación, que es el mismo Simulink pero con una librería de bloques especiales para comunicación en tiempo real. En este capítulo se detalló cómo fue llevada a cabo la implementación del sistema de control, previa descripción de la librería de bloques.

5.1 Introducción a la plataforma gráfica de programación.

Aparte de los bloques básicos de programación en Simulink, para poder comunicarse con el hardware (el equipo del péndulo invertido) a través de las tarjetas de adquisición, se hizo uso de los

bloques de la librería Real-Time Windows Target, también conocida en una forma general como Simulink® Desktop Real-Time™.

Esta librería ofrece diferentes bloques de entradas y salidas analógicas, digitales, y genéricas (fig. 5.1).

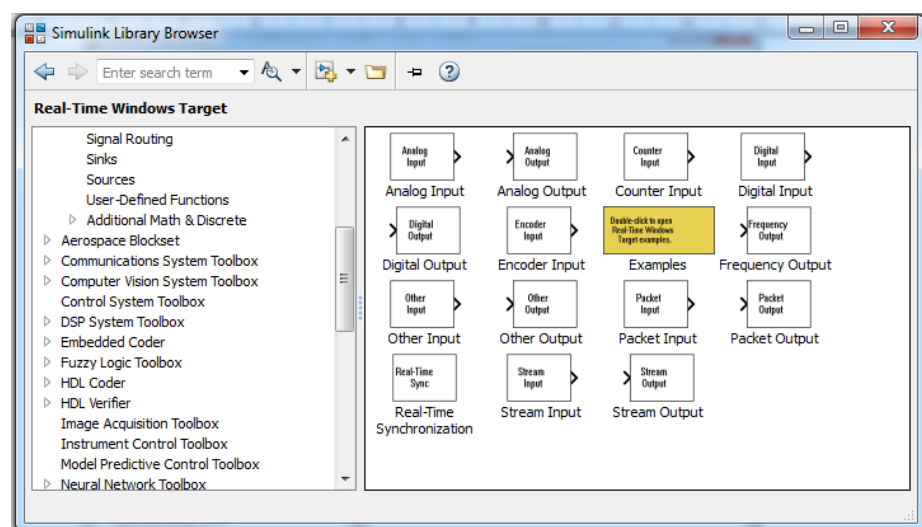


FIGURA 5.1. BLOQUES DE LA LIBRERÍA REAL-TIME WINDOWS TARGET (SIMULINK® DESKTOP REAL-TIME™.)

A continuación se describen los bloques más importantes de esta librería, junto a sus respectivos parámetros de configuración:

Encoder Input (Entrada de Encoder)

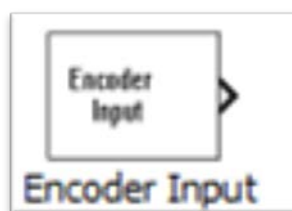


FIGURA 5.2. ÍCONO DEL BLOQUE DE ENTRADA DE ENCODER.

Se conecta a un encoder y lee las entradas de sus canales.

Librería: Simulink® Desktop Real-Time™.

Descripción: Se conecta a un encoder y lee entradas de canales especificadas comunicando las lecturas al modelo de Simulink. Después de que se ha añadido el bloque al modelo, se pueden ingresar los parámetros para las entradas/salidas del driver o tarjeta correspondiente.

Parámetros

Install new board: Hacer click en este botón para registrar una tarjeta en el Simulink Desktop Real-Time.

Cuando se instala una nueva tarjeta, el programa despliega una lista de fabricantes de tarjetas soportadas. Cuando se selecciona un fabricante, el programa despliega una lista de tarjetas disponibles de ese fabricante. Cuando se selecciona una tarjeta, el programa añade la tarjeta a la lista de tarjetas registradas y la vuelve la tarjeta en uso. Por defecto, la selección inicial en la lista de tarjetas registradas es <no boardselected> (ninguna tarjeta seleccionada).

Boardsetup (configuración de tarjeta): Dar click a este botón para configurar la tarjeta. Un cuadro de dialogo específico de la tarjeta se abre para poder configurarla. Para mayor información acerca de la configuración de la tarjeta, ver la documentación del fabricante.

Sample time (tiempo o periodo de muestreo): Se ingresa un valor representando que tan frecuentemente se desea que el bloque se ejecute e interactúe con la tarjeta. El bloque también sincroniza el modelo con el reloj de tiempo real a esta tasa de muestreo.

Tip: si se está usando un solucionador o resolvidor de paso fijo (fixed-stepsolver), se debe ingresar el mismo valor que se ingresó en la casilla de Tamaño de paso (Fixedstepsize) del cuadro de dialogo de Parámetros de Configuración, o un entero múltiple de ese valor.

Maximum missed ticks (marcas perdidas máximas): En Modo Externo (External Mode), el programa ignora este valor.

Yield CPU whenwaiting (Ceder CPU cuando se esté en espera): En Modo Externo, el programa ignora este valor.

Input Channels (Canales de Entrada): Se ingresa un vector de canales que seleccione los canales de entrada del encoder que se está utilizando en la tarjeta correspondiente. El vector puede ser cualquier forma de válida dentro de MATLAB. Por ejemplo, para

seleccionar los primeros cuatro canales de entrada de un encoder, se ingresa:

[1,2,3,4] o [1:4]

Quadrature mode (Modo de Cuadratura): Este parámetro especifica qué esquinas de las franjas del encoder deben ser contadas. De la lista se puede seleccionar: Single (Simple) — Cuenta las esquinas crecientes de un juego de franjas. Double (Doble) — Cuenta las esquinas crecientes de ambos juegos de franjas. Quadruple (default) (Cuádruple, por defecto) — Cuenta las esquinas crecientes y decrecientes de ambos juegos de franjas.

Reset input function (función de entrada de reinicio): El chip de interfaz del encoder tiene un conector de reinicio adicionalmente a las entradas del encoder. Este parámetro especifica la función de este conector. De la lista se puede seleccionar: Gate (puerta) — Habilita conteo del encoder. Reset (reinicio) — reinicio de nivel del conteo del encoder. Rising edge index (índice de esquina creciente) — reinicia el conteo del encoder en la esquina creciente. Falling edge index (índice de esquina decreciente) — Reinicia el conteo del encoder en la esquina decreciente.

Input filter clock frequency (Reloj de la frecuencia del filtro de entrada): El chip de interfaz de un encoder tiene un filtro pasa-bajo incluido que intenta filtrar las altas frecuencias, que son interpretadas como ruido.

Este parámetro es la frecuencia de corte (Hz) de este filtro. La frecuencia de corte que se especifique es redondeada a la frecuencia más cercana soportada por el chip.

Output data type (tipo de dato de salida): De la lista, se selecciona el tipo de dato que se desea que el bloque emita como salida.

Digital Output (Salida Digital)



FIGURA 5.3. ÍCONO DEL BLOQUE DE SALIDA DIGITAL.

Se conecta a líneas o canales de salidas digitales y escribe en éstas.

Librería: Simulink® Desktop Real-Time™

Descripción: Se conecta a líneas o canales digitales específicos y comunica datos desde el modelo de Simulink. Después de haber añadido un bloque de Salida Digital al modelo, se pueden ingresar los parámetros para su driver o tarjeta.

Parámetros

Install new board (Instalar nueva tarjeta): Igual que en el bloque de entrada de encoder.

Sample time (tiempo o periodo de muestreo): Igual que en el bloque de entrada de encoder.

Maximum missed ticks (marcas perdidas máximas): Igual que en el bloque de entrada de encoder.

Yield CPU when waiting (Ceder CPU cuando se esté en espera): Igual que en bloque de entrada de encoder.

Output channels (Canales de Salida): Ingrese un vector de canales que seleccione los canales de salida digital que se están usando en la tarjeta registrada. El vector puede ser de cualquier tipo válido de MATLAB. Por ejemplo, para seleccionar los primeros ocho canales de salida digitales, ingrese: [1,2,3,4,5,6,7,8] o [1:8]. Si se quiere usar las primeras cuatro líneas de salidas digitales, ingrese: [1,2,3,4].

Si se tiene un canal digital de 8-bits, ingrese [1]. Si se tiene dos canales de 8-bits, ingrese [1 9], y de la lista de Channel mode (modo de canal) escoja Byte.

Channel mode (modo de canal): De la lista, escoja entre las siguientes: Bit — Se espera un valor de tipo 0 o 1. Byte — Se espera un valor de 0 a 255 que es convertido en un canal digital de ocho líneas digitales.

Initial value (valor inicial): Ingrese los valores iniciales para cada línea o canal de salida digital que haya sido ingresada en la casilla de “Output Channels” (Canales de Salida).

Por ejemplo, si se ingresó [1,2,3,4] en la casilla de Canales de Salida, y se requieren valores iniciales de 0 y 1, entonces se debe ingresar: [0,0,1,1].

Si se escogió “Byte” en la lista desplegable de “Channel Mode” (Modo de Canal), ingrese un valor entre 0 y 255 por cada canal de salida digital. Por ejemplo, para un byte (ocho líneas digitales) con un valor inicial de 25 y 50, ingrese [25 50].

Final Value (Valor Final): Ingrese un valor final por cada canal de salida digital que se haya ingresado en la casilla de Output Channels (Canales de Salida).

Por ejemplo, si se ingresó [1,2,3,4] en la casilla de canales de salida, y se requieren valores finales de 0, se de ingresar: [0,0,0,0]

Si se escogió Byte de la lista desplegable de Channel Mode (Modos de Canal), ingrese un valor entre 0 y 255 por cada canal de salida digital.

Analog Output (Salida Analógica)

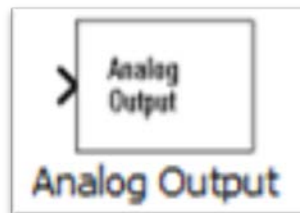


FIGURA 5.4. ÍCONO DEL BLOQUE DE SALIDA ANALÓGICA.

Se conecta y pasa/escrbe datos a los canales de salida analógicos.

Librería: Simulink® Desktop Real-Time™

Descripción: Se conecta a canales de salida analógica específicos y comunica datos desde el modelo de Simulink. Después de que se añade un bloque de Salida Analógica al modelo, se pueden ingresar los parámetros para su tarjeta correspondiente.

Parámetros

Install new board (Instalar nueva tarjeta): Igual que en el bloque de entrada de encoder.

Sample time (tiempo o periodo de muestreo): Igual que en el bloque de entrada de encoder.

Maximum missed ticks (marcas perdidas máximas): Igual que en el bloque de entrada de encoder.

Yield CPU when waiting (Ceder CPU cuando se esté en espera): Igual que en el bloque de entrada de encoder.

Output channels (Canales de Salida): Se ingresa un vector de canales que seleccione los canales de salida analógicos que se estén usando en la tarjeta. El vector puede ser cualquier vector

válido de MATLAB. Por ejemplo, para seleccionar los dos primeros canales de salida analógicos, ingrese:

[1,2] o [1:2]

Output range (Rango de Salida): De la lista, se escoge el rango de salida para los canales de salida analógica que se ingresaron en la casilla de Canales de Salida. Por ejemplo, si la tarjeta tiene un rango de salida de ± 5 V, escoja -5 a 5 V.

Si se requiere un rango de salida diferente para diferentes canales analógicos, añada un bloque de salidas por cada rango de salida diferente.

Block input signal (Señal de Entrada del Bloque): De la lista, escoja entre las siguientes opciones: Volts (Voltios) — Se espera un valor igual al de la salida de voltaje analógico. Normalized bipolar (Bipolar Normalizada) — Se espera un valor entre -1 y +1 que es convertido al rango completo de la salida de voltaje sin importar el rango de voltaje de salida. Normalized unipolar (Normalizada Unipolar) — Espera un valor entre 0 y +1 que es convertido al rango completo de la salida de voltaje sin importar el rango de voltaje de salida. Por ejemplo, rangos de salida analógicos de 0 a +5 voltios y de -5 a +5

voltios son convertidos desde valores de 0 a +1. Raw (Puro) — Espera un valor de 0 a $2^n - 1$. Por ejemplo, un conversor A/D de 12 bits esperaría un valor entre 0 y $2^{12} - 1$ (0 a 4095).

La ventaja de este método es que el valor esperado es un entero sin errores de redondeo.

Initial Value (Valor Inicial): Ingrese el valor inicial para cada canal de salida analógica ingresado en la casilla de Canales de Salida. Por ejemplo, si se ingresó [1,2] en la casilla de Canales de Salida, y se requiere un valor inicial de 0 voltios, ingrese [0,0].

Final Value (Valor Final): Ingrese un valor final para canal analógico ingresado en la casilla de Canales de Salida. Por ejemplo, si se ingresó [1,2] en la casilla de Canales de Salida, y se requiere un valor final de 0 voltios, ingrese [0,0].

Configuración de parámetros para simulación y adquisición - generación de datos en tiempo real con Windows Real Target

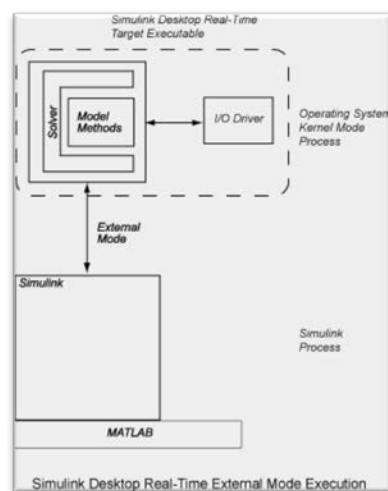
Tomando como referencia los tutoriales en línea en la página de MathWorks [25], se describe a continuación el proceso de configuración de Simulink para trabajar en tiempo real.

Una alternativa de más alto rendimiento que el Modo Normal es la ejecución en tiempo real en Modo Externo. En este modo, se usa Simulink® Coder™ para enlazar dinámicamente el código del algoritmo generado con el código del driver de entradas y salidas generado por los bloques de entradas y salidas. El ejecutable resultante corre en el modo de kernel del sistema operativo en la computadora anfitriona e intercambia datos de parámetros con Simulink vía una interfaz de memoria compartida.

Adquisición de señales — Se pueden capturar y visualizar señales desde la aplicación de tiempo real mientras está corriendo. La información de las señales es recuperada de la aplicación de tiempo real y visualizada en los mismos bloques de Visualización (Scope) de Simulink que se usaron para simular el modelo.

Ajuste de parámetros — Se pueden cambiar los parámetros en el diagrama de bloques de Simulink y pasar los nuevos parámetros automáticamente a la aplicación de tiempo real. El modo externo de Simulink cambia los parámetros en la aplicación de tiempo real mientras está corriendo en tiempo real.

El ejecutable del modo externo está totalmente sincronizado con el reloj de tiempo real. El principal rol de Simulink es leer y desplegar los resultados de la simulación retornados desde el ejecutable (Fig. 5.5). Nota: se debe usar un resolvidor de paso fijo en el modo externo.



**FIGURA 5.5. DIAGRAMA DE EJECUCIÓN DEL SIMULINK
DESKTOP REAL-TIME EN MODO EXTERNO**

Después de haber creado el modelo en Simulink, se pueden ingresar los parámetros de simulación para uso del programa de generación de código SimulinkCoder™ para crear código C y construir la aplicación de tiempo real.

1. En la ventana de Simulink, y desde el menú de Simulación, hacer click en Model Configuration Parameters (Configuración de Parámetros del Modelo).
2. Hacer click en el nodo de Code Generation (Generación de Código). El panel de Generación de Código se abre.
3. En la sección de selección del Target (Destino), hacer click en el botón Browse (Buscar) en la lista de archivos destino del Sistema. El buscador de archivos destino del sistema se abre.
4. Seleccione el archivo destino del sistema para construir una aplicación en el Simulink Desktop Real-Time™, rtwin.tlc, y hacer click en OK.

El dialogo automáticamente ingresa el archivo destino del sistema rtwin.tlc, la plantilla de makefile rtwin.tmf, y el comando makemake_rtw en el panel de Generación de Código.

Aunque no está visible en el panel de Generación de Código, cuando se hace click en OK también se configura el archivo MEXrtwinext de la interfaz destino externa. Esto permite al modo externo pasar los nuevos parámetros a la aplicación de tiempo real y retornar la información de señales desde la aplicación de tiempo real. La información es desplegada en los bloques de visualización (Scope) o almacenados con registro de señales. El panel de Generación de Código debe lucir como se muestra en la figura 5.6:

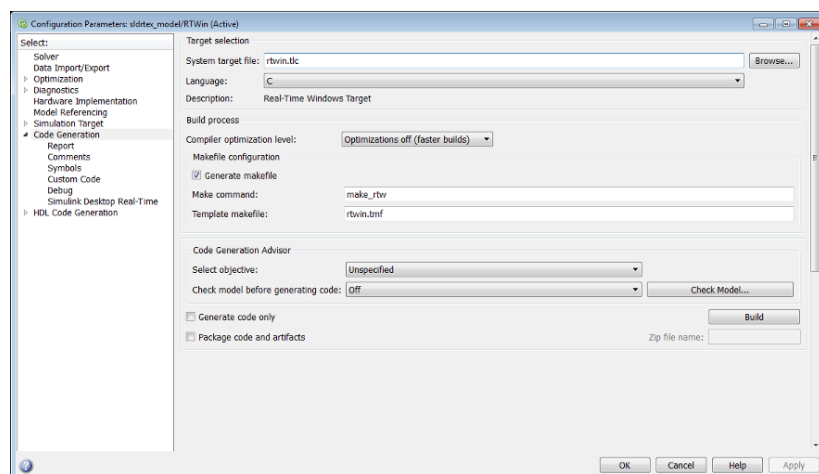


FIGURA 5.6. CONFIGURACIÓN DE LA VENTANA DE GENERACIÓN DE CÓDIGO.

No seleccionar Inline parameters (Parámetros en línea) en el nodo de Signals and Parameters (Señales y Parámetros) bajo Optimization (Optimización).

5. Hacer click en el nodo de Hardware Implementation (Implementación de Hardware). Los valores por defecto son derivados de la arquitectura de la computadora anfitriona. Por ejemplo (Fig. 5.7), para una máquina Intel® de 64 bits, éstos son:

- a. Vendedor del dispositivo — Intel
- b. Tipo de dispositivo — x86-64
- c. Hardware de prueba es el mismo que el de producción — Seleccionado

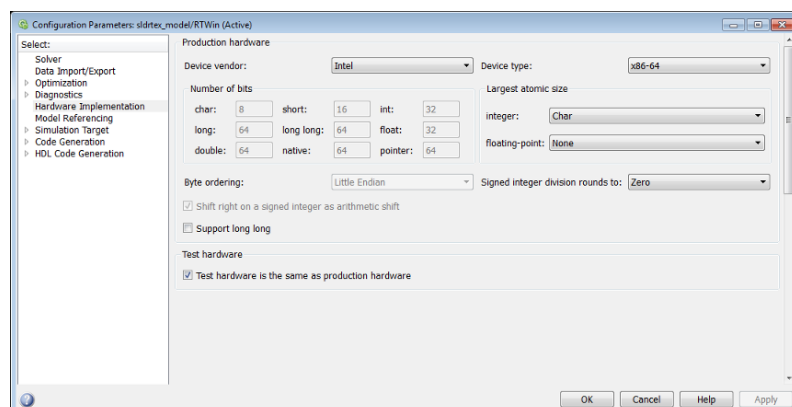


FIGURA 5.7. DATOS DEL PROCESADOR EN VENTANA DE IMPLEMENTACIÓN DE HARDWARE.

Hacer click en OK para aplicar los cambios al modelo y cerrar el cuadro de dialogo.

5.2 Programación e implementación del control

Dada la forma en que se pueden programar y agrupar los bloques en Simulink, el objetivo fue obtener una interfaz final con pocos bloques para la fácil visualización, que los parámetros de control estén disponibles para su modificación, además de contar con alguna forma de control de arranque o parada por parte del usuario y algún medio de visualización de las señales.

En cuanto a la organización de los bloques, se quiso mantener una diagramación lo más parecida posible a la de un diagrama de bloques básico, de tal modo que se tengan cuatro categorías principales: los bloques de control, el bloque que realiza el cambio entre auto-levantamiento y estabilización, el bloque que representa la planta y bloques adicionales de interfaz de usuario.

Dentro de la categoría de bloques de control se tuvo a los dos bloques PID que controlaban la estabilización del péndulo, y el bloque encargado del auto-levantamiento. El bloque de cambio de

señal de control contenía la lógica de comparación para controlar un selector y re-direccionar la señal de control requerida. Dentro del bloque planta se encontraba el manejo de señales de entrada y salida, en otras palabras el bloque que comunicaba el programa de Simulink con la planta real a través de las tarjetas de adquisición.

Entre los bloques adicionales de interfaz de usuario estaban un selector o “switch” de parada manual, y un bloque tipo Scope para visualización de varias señales.

A continuación se detallan la formación y configuración de las partes del programa mencionadas en el párrafo anterior.

Bloque “Planta”: en primer lugar se establecieron los bloques de entrada y salida de señales. Como entradas se tuvo a las señales de los sensores de posición que están dadas por los encoders incrementales, y como salida, el control digital de sentido de giro del motor, el control analógico de voltaje del driver del motor y el control digital de arranque/parada del motor.

Los bloques necesarios pueden ser encontrados en la librería del Real-Time Windows Target.

Se arrastraron hacia el espacio de trabajo dos bloques de “Encoder Input”, para configurar la entrada de encoder que mide la posición angular se hizo doble click sobre el bloque para abrir la ventana de configuración. La primera vez que se realiza esta acción se deben instalar los drivers de la tarjeta de adquisición para que sea reconocida por Simulink, para ello se dio un click sobre “install new board” y en la lista desplegable se eligió National Instruments (2), donde se encuentra el modelo de la tarjeta que se está usando, la PCI-6221 (no la de 38 conectores), posteriormente se hizo una configuración de la tarjeta en particular.

En “Sample time” se ingresó el tiempo o periodo de muestreo, en este caso es de 0.001 segundos; y en “Maximum missed ticks” basta con un valor de 1000.

Como canal de entrada se eligió el 2 que corresponde al conjunto de conectores físicos destinados al encoder que mide la posición angular del péndulo.

En el modo de cuadratura se eligió trabajar con la resolución nativa del encoder (2500 PPR) en otras palabras, el modo single o simple.

En general el cuadro de dialogo de la entrada de encoder debe verse como indica la figura 5.8.

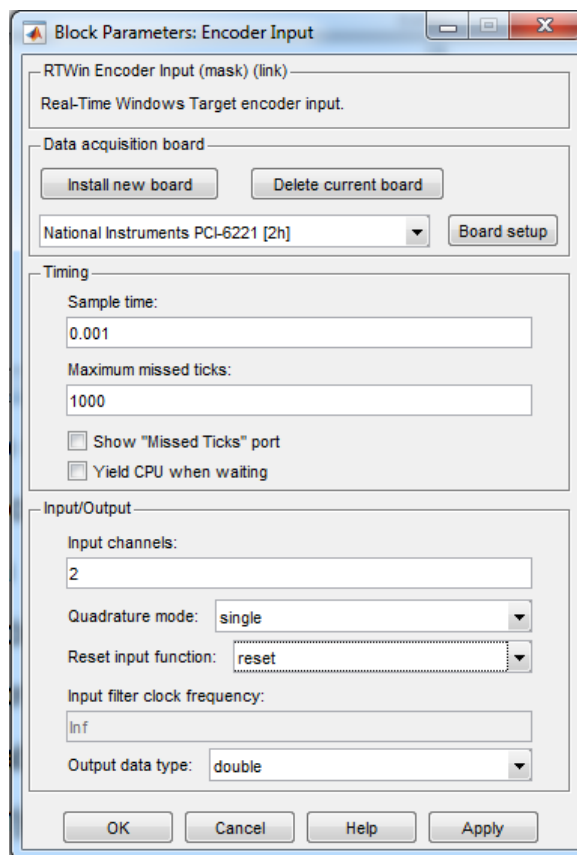


FIGURA 5.8. CONFIGURACIÓN DE BLOQUE ENTRADA DE ENCODER PARA MEDICIÓN DE POSICIÓN ANGULAR DEL PÉNDULO

Solo restando configurar la tarjeta en particular, se dio click en “board setup”, abriendo una ventana de diálogo. Existen diferentes campos a llenar, por ejemplo en “device order” se ingresó el número

con el cual Simulink reconoce a la tarjeta, éste número está relacionado con su instalación física y el slot PCI que ocupa, y que es 2, este número es el mismo para las posteriores configuraciones pues las entradas y salidas usan la misma tarjeta.

En las dos entradas para modos de operación del contador “Counter 0/1 mode”, se eligió “Quadrature encoder”, la configuración al final debe lucir como se aprecia en la figura 5.9

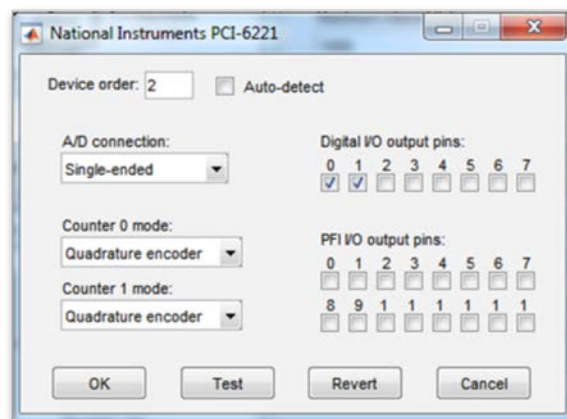


FIGURA 5.9. CONFIGURACIÓN DE LA TARJETA DE ADQUISICIÓN DE DATOS PARA LA ENTRADA DE ENCODER

Para confirmar si la configuración está bien hecha, se puede hacer click en “Test”, si todo está bien, debe aparecer un visto verde, finalmente se dio click en OK para cerrar la ventana de configuración de entrada de encoder.

La configuración para la entrada de encoder utilizada para la posición lineal, fue igual a la de posición angular, con la salvedad de que se usó el canal 1 en lugar del 2 en el campo de "Input Channels".

Ambos bloques leen correctamente las señales de los encoders, pero los datos que emiten están en forma de pulsos, éstos necesitaban ser transformados a radianes en el caso del encargado de medir la posición angular del péndulo, y a metros para el que mide la posición lineal del carro.

Ambos encoders tienen la misma resolución de 2500 PPR, y en las configuraciones de sus bloques de lectura se escogió la lectura simple, por lo tanto se emiten 2500 pulsos en una vuelta o 2π radianes. Además, puesto que ambos encoders están instalados en la parte trasera, los conteos disminuyen con el movimiento horario y viceversa. En el sistema referencial elegido, el ángulo aumenta en el sentido horario, con todo lo anterior en mente se configura un bloque de ganancia que transforme los pulsos a radianes y cambie el sentido con la expresión $-2\pi/2500$, como se muestra en la figura 5.10

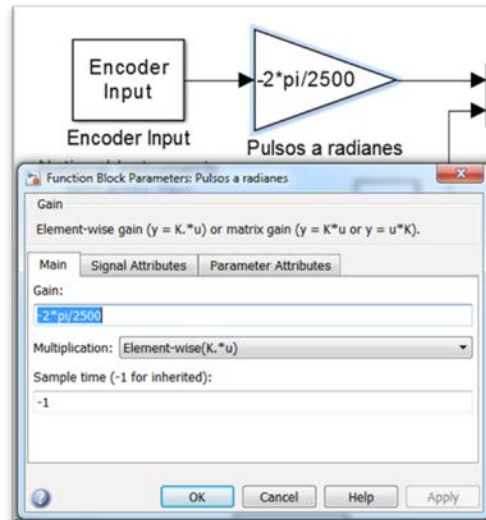


FIGURA 5.10. CONFIGURACIÓN DE GANANCIA DE CONVERSIÓN DE PULSOS A RADIANES

Adicionalmente, el contador parte de cero, por lo tanto cuando se ejecuta el programa esto se interpreta como si el péndulo parte de 0 radianes en su posición de equilibrio estable, sin embargo, en el análisis del modelo en realidad parte de π radianes. Esto se arregló insertando un bloque de constante de la librería de “Sources” o fuentes y un bloque de Suma de la librería de “Math Operations” u operaciones matemáticas. En el bloque de constante se ingresó “pi”, el bloque de suma tiene dos entradas se conecta la salida del bloque de conversión y la constante en cada una, como se muestra en la figura 5.11.

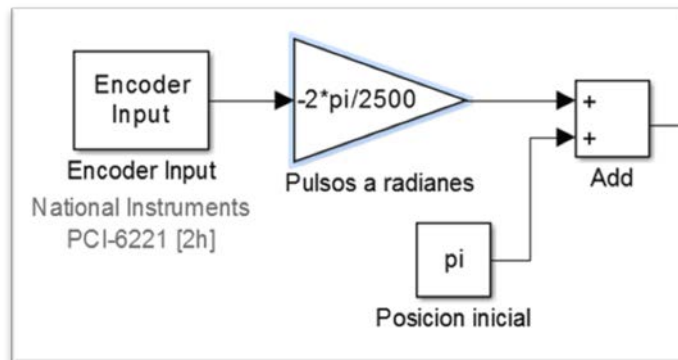


FIGURA 5.11. INICIALIZACIÓN DE LA MEDICIÓN DE POSICIÓN ANGULAR DEL PÉNDULO.

A diferencia de la simulación, en la implementación del control en la planta se requería una acción para cautelar que el ángulo del péndulo este dentro de un cierto rango en el cual el control puede funcionar, esto es especialmente importante si de alguna forma falla el control de auto-levantamiento y el péndulo da una vuelta en uno u otro sentido, también si durante el control de estabilización se le perturba de tal forma que caiga y se salga del rango de $\pm 2\pi$. Se necesitó de una función discontinua que reciba valores infinitos positivos o negativos de ángulos múltiplos de π y a la salida entregue solo valores en el rango $(-\pi, \pi]$ por ejemplo para una entrada de 3π a 5π la salida debe ser $-\pi, \pi$. Una función tipo diente de sierra cumplía con los requisitos, se ingresó un bloque de función de Matlab con el siguiente código:

```
function y = rango(u)
%#codegen
y=2*pi*(mod(((u-pi)/(2*pi))-0.0001,1)-0.4999);
```

La salida de encoder ya transformada a radianes se conectó al bloque de función etiquetado como “rango” y la salida de éste al bloque de tipo salida “Teta”, como se puede ver en la figura 5.12.

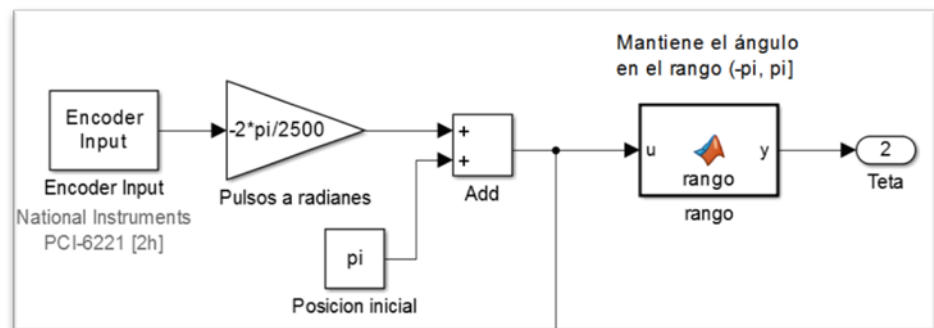


FIGURA 5.12. DIAGRAMA DE BLOQUES DE MEDICIÓN DE LA POSICIÓN ANGULAR.

Puesto que el bloque de control del auto-levantamiento necesitaba una retroalimentación de velocidad angular, ésta necesitaba ser medida a partir de la información del encoder, a diferencia de los modelos simulados donde se obtiene por ejemplo usando un bloque de derivada o tomando la primera integral de la aceleración angular; en el caso de una medición real se enfrentaba el problema del ruido.

Para medir la velocidad angular con el dato de posición angular, en teoría bastaba con tomar el valor actual y restarle el anterior y ese resultado dividirlo para el tiempo de toma de cada valor, que es constante y es el tiempo de muestreo (0.001 segundos), usando para ello el bloque “Difference” de la librería “Discrete” y un bloque de ganancia. Sin embargo, la señal así obtenida era altamente ruidosa, lo cual se arregló usando un bloque de promedio “Mean”. La señal que entrega se conectó a un bloque de salida con el nombre de “w”, como se muestra en la figura 5.13.

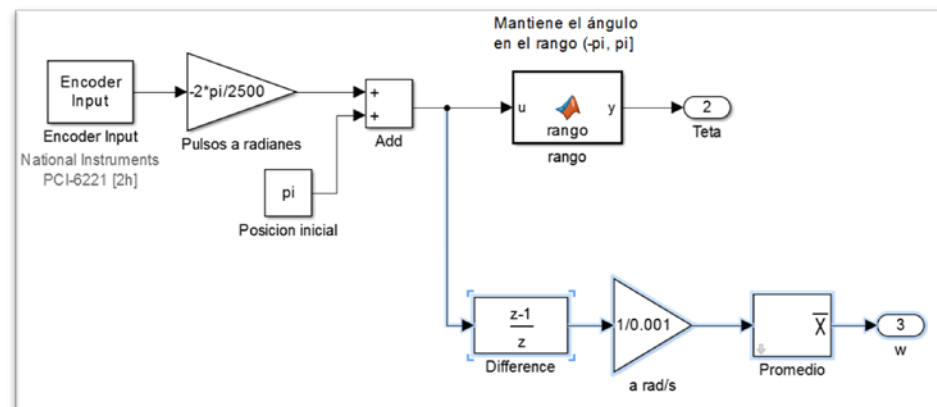


FIGURA 5.13. DIAGRAMA DE BLOQUES DE MEDICIÓN DE LA VELOCIDAD ANGULAR (SOMBREADO EN CELESTE).

Una vez completamente configurada la medición de posición y velocidad angular, se aplicó un procedimiento similar para la posición lineal. Usando la información de la ecuación 1.1, se sabía

que en una vuelta completa de la polea en cuyo eje está montado el encoder, se producía un desplazamiento lineal equivalente a la longitud de una circunferencia con diámetro de paso es 0.07948 m.

Conectando la salida del bloque de conversión a un bloque de salida x, se obtuvo un diagrama de bloques como el mostrado en la figura 5.14.

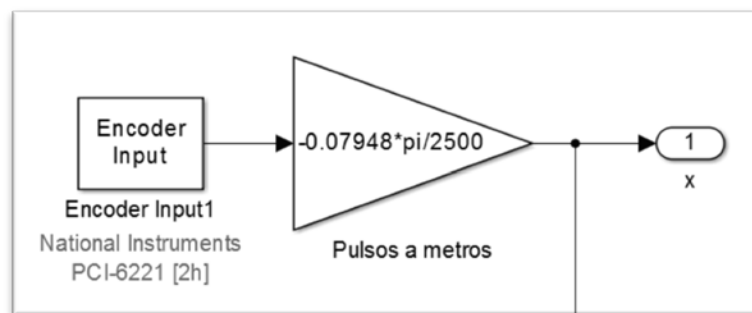


FIGURA 5.14. DIAGRAMA DE BLOQUES DE MEDICIÓN DE LA POSICIÓN LINEAL DEL CARRO.

En cuanto a las salidas, se necesitaban tres bloques, uno que controle la velocidad usando un voltaje analógico con un rango máximo de 1 a 4 voltios, la dirección dada por una señal digital, y el arranque o parada del motor también usando una señal digital, en estas dos últimas la señal digital era de tipo “booleana”.

Para el manejo de la velocidad se ingresó un bloque de salida analógica, con tiempo de muestreo igual al bloque de entrada de encoder y al de la simulación, varias ejecuciones mostraron que un valor de 10 en marcas perdidas máximas era suficiente.

La selección de canal fue 1, correspondiente a donde se tenía físicamente conectada la entrada de voltaje del driver con la bornera de la tarjeta de adquisición.

El rango disponible es el nativo de la tarjeta (-10 a +10 voltios), por seguridad se establecieron los valores iniciales y finales como 0. Esto garantizaba que si ocurre algún error o parada repentina del programa no se siga emitiendo voltaje en salida analógica, es una forma de resetear la señal al inicio y final de la ejecución. La ventana de configuración debe verse como se muestra en la figura 5.15.

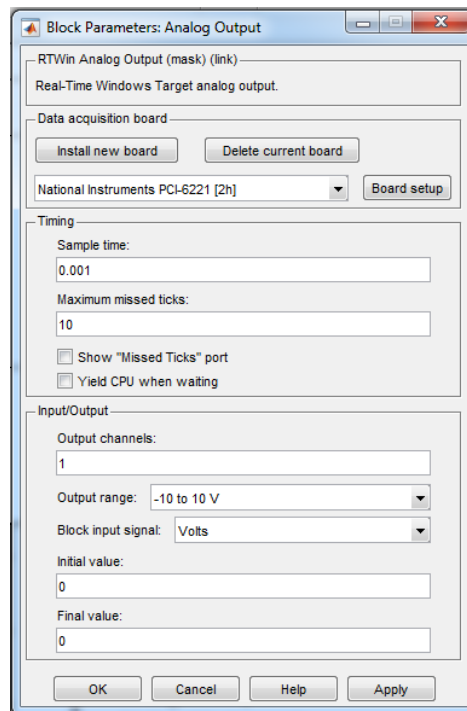


FIGURA 5.15. CONFIGURACIÓN DEL BLOQUE DE SALIDA ANALÓGICA DE VOLTAJE.

En cuanto a la configuración de la tarjeta en Board Setup, la ventana es la misma que se muestra en la figura 5.6, en el campo de “A/D Connection” se eligió “Single Ended”, que es para el tipo de conexión no diferencial, pues solo se conecta un cable que lleva la señal y un cable de tierra).

La señal de control es un valor que puede variar entre valores positivos o negativos, pero la entrada de voltaje referencial en el driver del motor solo acepta valores positivos entre 1 y 4 voltios y

determina la dirección con la entrada digital. Por lo tanto, la solución fue bifurcar la señal de control de tal forma que al bloque de salida analógica ingrese su valor absoluto, para ello se introdujo el bloque “Abs” de la librería “Math Operations”, quedando como se muestra en la figura 5.16.

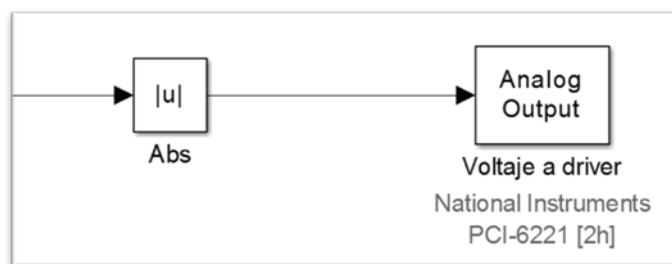


FIGURA 5.16. DIAGRAMA DE BLOQUES DE LA SALIDA ANALÓGICA DE VOLTAJE.

Para la salida digital de la señal de dirección la configuración es similar a los otros bloques, en el canal de salida se eligió 1 que es el correspondiente a donde se ha hecho la conexión física, el tipo de dato o modo de canal es Bit (1 ó 0), y como valores iniciales y finales se elige 0 (movimiento horario o hacia la derecha). La ventana de configuración debe verse como en la figura 5.17. en “Board setup”, se activan las casillas de entradas/salidas digitales 0 y 1, correspondientes a los canales 1 y 2, dejando de esta forma ya

activada la salida digital del bloque de “Run/Stop” descrito más adelante.

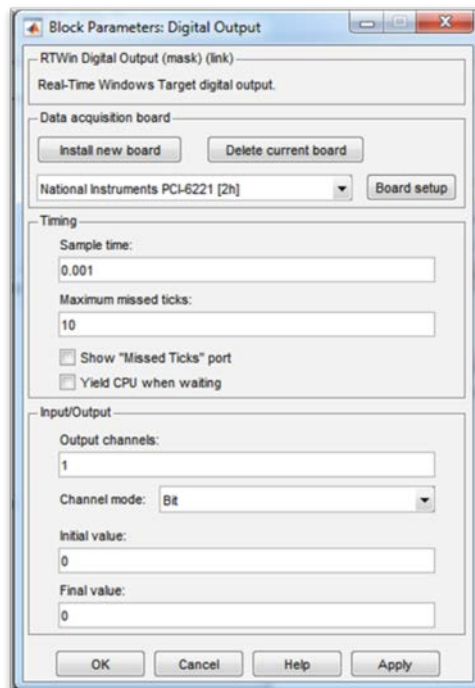


FIGURA 5.17. CONFIGURACIÓN DEL BLOQUE DE SALIDA DIGITAL DE DIRECCIÓN.

En la configuración de la salida analógica se mencionó que la señal de control necesitaba bifurcarse, teniendo por un lado el valor absoluto para el voltaje de referencia; para manejar el problema de la dirección el otro ramal de la bifurcación de la señal de control se compara con cero.

Se introdujo el bloque de comparación con cero de la librería de operación lógicas, cambiando el operador a menor que (<). Si la señal de control era negativa, lo cual se interpretaba como movimiento a la izquierda, el bloque de comparación emitía una señal de verdadero de tipo “booleana” o bit 1, caso contrario emitía bit 0. La conexión se muestra en la figura 5.18.

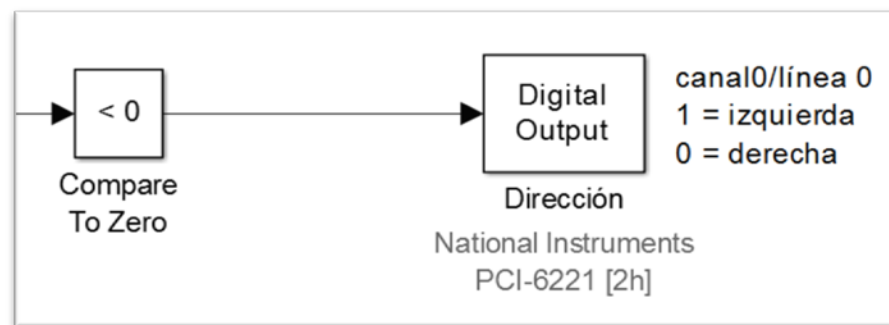


FIGURA 5.18. DIAGRAMA DE BLOQUES DE SALIDA DIGITAL DE DIRECCIÓN.

Para la salida digital de “Run/Stop” o de control de parada y habilitación del motor, la configuración de su bloque fue similar al de dirección y se muestra en la figura 5.19, con la diferencia de que el canal de salida utilizado fue el número 2, y los valores iniciales y finales fueron en ambos casos 1, lo cual se hace por seguridad, pues la señal 1 activa la parada inmediata del motor, con esto se

aseguró que al inicio y al finalizar la ejecución del programa el motor estará en parada.

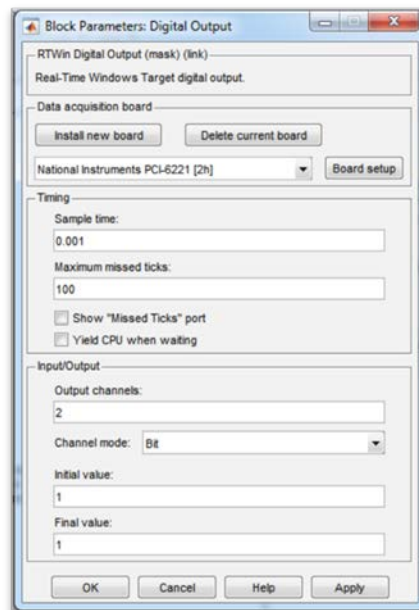


FIGURA 5.19. CONFIGURACIÓN DEL BLOQUE DE LA SALIDA DIGITAL DE PARADA/ARRANQUE.

Esta señal de parada bien podría ser activada únicamente de forma manual, pero se dejaba de lado un problema que no se había solucionado completamente en las simulaciones: la longitud finita de los rieles guía. Sin nada que le indique al sistema que ha llegado a los límites de desplazamiento, podrían ocurrir choques del carro con los extremos.

Puesto que no se cuenta con finales de carrera, la protección de extremos se llevó a cabo usando lógica. Asumiendo las condiciones iniciales de que el carro está ubicado en el centro y conociendo que la longitud del riel es de cerca de 1 metro, lo más sencillo sería comparar la posición lineal x con los extremos, si el carro ha llegado a alguno de los dos extremos la comparación se vuelve verdadera, emitiendo la señal booleana 1 que se puede direccionar para activar la parada. Sin embargo, si solo se compara la posición, una vez que el carro ha llegado a un extremo y se ha parado, no existiría nada que lo permita volver a la zona de control, por lo tanto hace falta una comparación adicional.

La propuesta fue incluir una comparación de la señal de control, de tal modo que las paradas solo se activen si además de haber llegado a un posición límite, la señal de control seguía ordenando moverse en la dirección del límite; así, si la señal de control ordenaba alejarse del límite la parada se desactivaba permitiendo al sistema seguir ejecutando el control.

Esto en la práctica, por ejemplo para el límite a la derecha es comparar la señal de control para establecer si es mayor a cero (moverse hacia la derecha) y si la posición lineal x ha sobrepasado

el límite de 0.45 m desde el centro (cerca de 2 cm de distancia entre el carro y el apoyo derecho de los rieles guía). Dado que ambas condiciones deben cumplirse para activar la parada, las salidas de ambos comparadores se introducen en un bloque de lógica “AND”, el cual solo será verdadero (bit 1) cuando las dos señales que tiene de entrada sean verdaderas.

Se aplicó lo mismo para el límite a la izquierda, únicamente cambiando el signo del límite de posición (-0.45 metros) y los comparadores por menor que (<). Las salidas de los bloques “AND”, se nombraron apropiadamente. La estructura de la lógica de protección de extremos se ilustra en la figura 5.20.

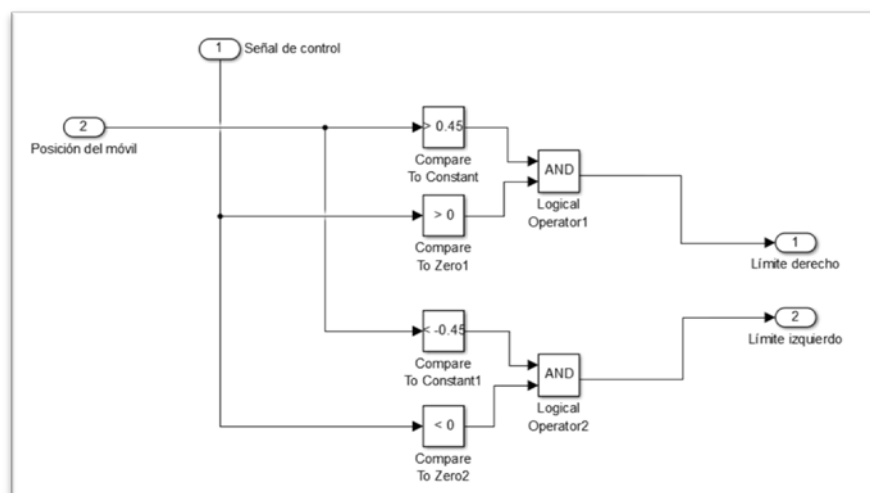


FIGURA 5.20. DIAGRAMA DE BLOQUES DE LA LÓGICA DE PROTECCIÓN DE EXTREMOS.

Se seleccionaron todos los componentes de la lógica de protección de extremos y se colocaron en un sub-sistema como se muestra en la figura 5.21.

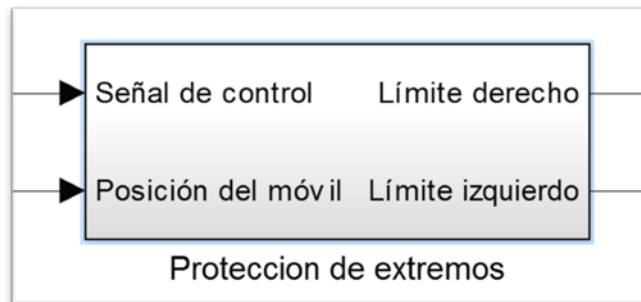


FIGURA 5.21. SUB-SISTEMA DE PROTECCIÓN DE EXTREMOS.

Con lo anterior se tenían tres posibilidades de activar la parada: de forma manual, activación de límite derecho, y activación de límite izquierdo. Basta con que una de estas esté activa para disparar la parada, lo cual se logra introduciendo las tres señales en un bloque de lógica "OR", éste es obtenido modificando el bloque "Logical Operator" u operador lógico de la librería de "Logic and Bit Operations", una vez ingresado al espacio de trabajo, se da doble click sobre éste y se cambia el número de puertos de entrada a 3, y el operador lógico a OR (tiene a AND por defecto).

Finalmente el diagrama de bloques completo para manejar las paradas se muestra en la figura 5.22.

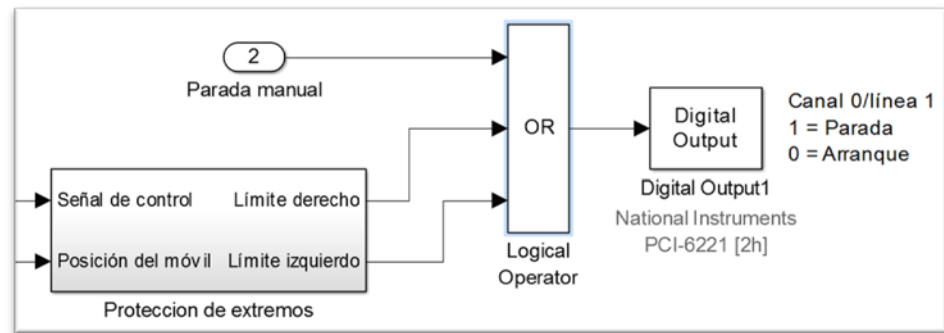


FIGURA 5.22. DIAGRAMA DE BLOQUES DE MANEJO DE PARADAS.

Los diagramas de adquisición y generación de datos de la planta finalmente quedaron establecidos, una vista completa del mismo puede observarse en la figura 5.23.

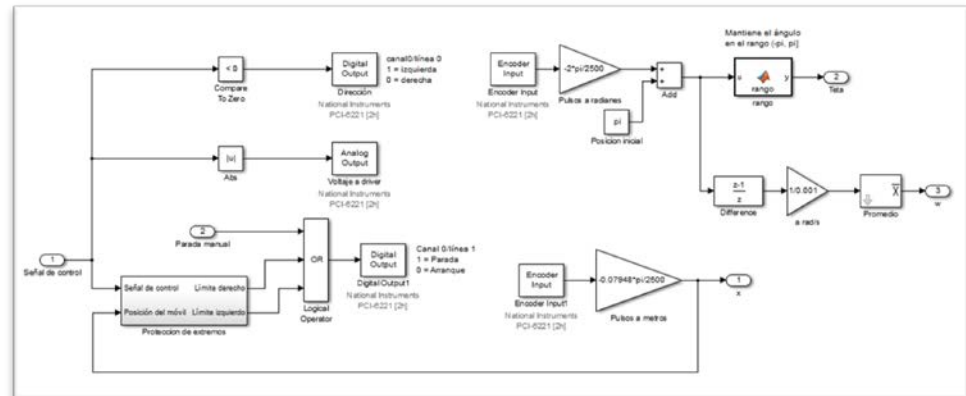


FIGURA 5.23. DIAGRAMA DE BLOQUES COMPLETO DE LA PLANTA.

Se seleccionaron los componentes que conforman la planta, se dio click derecho y se eligió crear subsistema, así la planta quedaba representada por un solo bloque (figura 5.24), si se deseaba acceder a su contenido bastaba con dar doble click sobre éste.

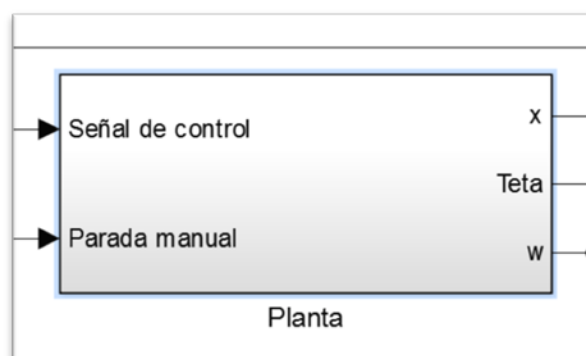


FIGURA 5.24. BLOQUE SUB-SISTEMA DE LA PLANTA.

Bloque Selector de Control: prácticamente igual al descrito en la simulación del controlador (figura 5.25).

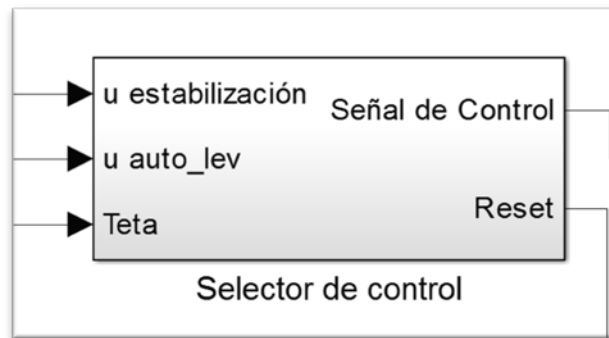


FIGURA 5.25. BLOQUE SELECTOR DE CONTROL.

Bloques de Control: también tienen la misma estructura de la simulación del controlador, con los dos bloques PID en cascada, y el bloque función de Matlab, "auto_lev_f" que se encarga del auto-levantamiento (figura 5.26).

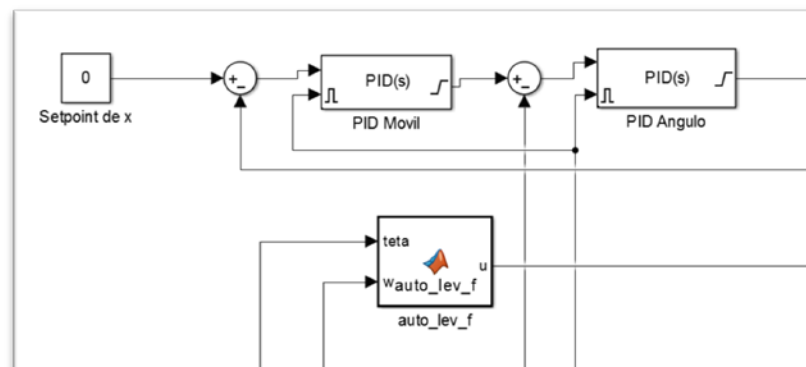


FIGURA 5.26. BLOQUES DE CONTROL.

Bloque de la relación no lineal Voltaje de Referencia – Voltaje

de Fases: Con este bloque se introdujo la relación no lineal que se dejó fuera del análisis por su alta no linealidad y que responde a la ecuación 3.1, aunque de forma inversa, pues la entrada es la señal de control de voltaje de fases y ésta necesita ser transformada a voltaje de referencia para poder comunicarlo al bloque de salida analógica. La relación inversa estaría expresada como sigue:

$$v_{ref} = \begin{cases} 0.05v_{ref} + 1.62; e_a = (0, 15.73] \\ 0.05v_{ref} - 1.62; e_a = [-15.73, 0) \\ 0; e_a = 0 \end{cases} \quad \text{Ec. 5.1}$$

Un bloque que ayudó a implementar este tipo de relación aunque no sea su propósito inicial, es el bloque de “Coulomb & Viscous Friction” de la librería de Discontinuidades, cuya configuración de acuerdo a los datos de la ecuación 5.1 debe verse como en la figura 5.27.

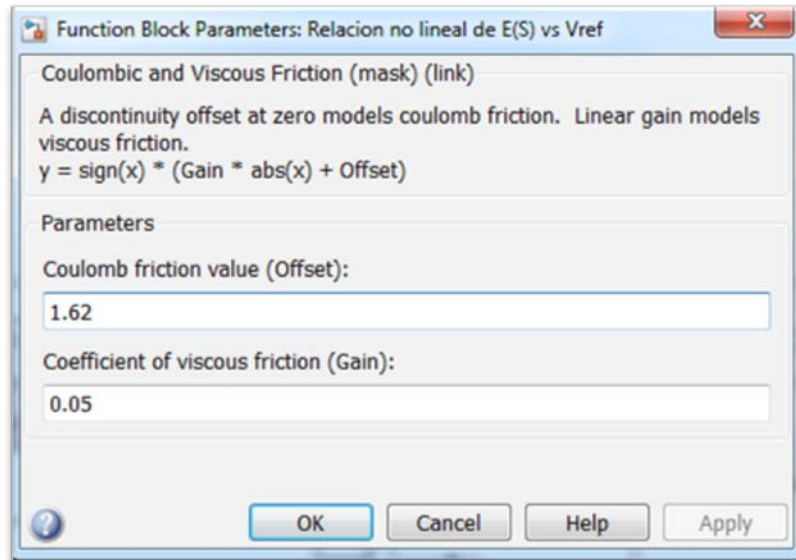


FIGURA 5.27. CONFIGURACIÓN DEL BLOQUE DE LA RELACIÓN NO LINEAL DE E(S) VS V_{REF} .

Bloques adicionales de interfaz de usuario: Básicamente comprende el bloque de visualización de señales y el de parada/arranque manual.

Para la parada manual se introduce un bloque selector de la librería “Signal Routing”, con dos bloques de constantes, 0 para el arranque y 1 para la parada, la salida del selector se introduce a la entrada de parada manual del bloque planta como se observa en la figura 5.28.

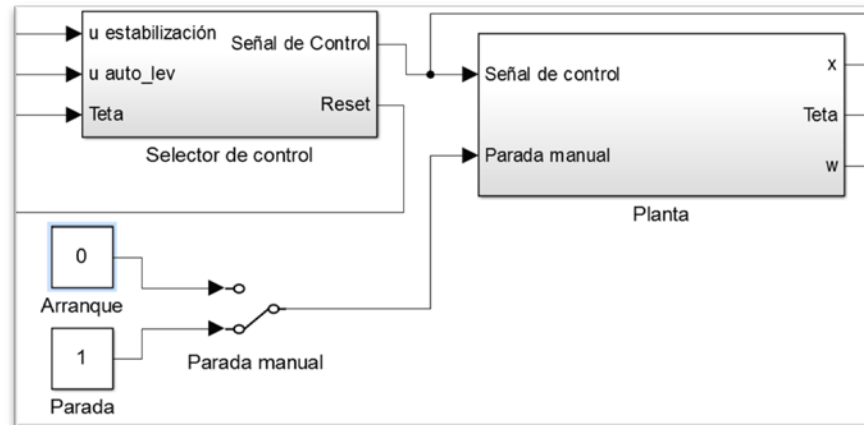


FIGURA 5.28. CONEXIÓN DEL SELECTOR DE PARADA CON EL BLOQUE PLANTA.

Para la visualización de las señales se usó un bloque de tipo “Scope”, también usado en las simulaciones, configurado para que admita cuatro entradas (de esta forma se muestran por separado), también se activa la casilla “scroll”, para la visualización continua en barrido. Se cambia el color de fondo y grosores y colores de línea para una mejor visualización también. En la figura 5.29 se muestra el bloque y como se ve al abrirlo.

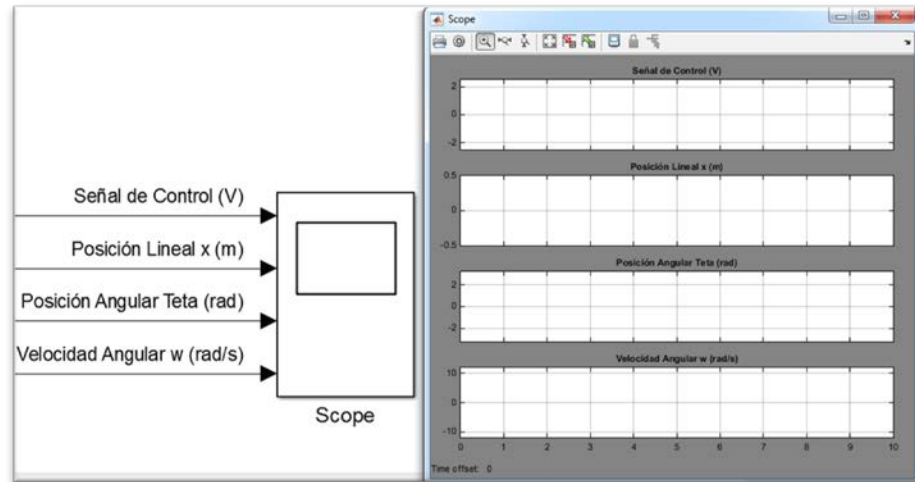


FIGURA 5.29. BLOQUE DE VISUALIZACIÓN DE SEÑALES.

La estructura o diagrama de bloques y conexiones del programa completo pueden observarse en la figura 5.30.

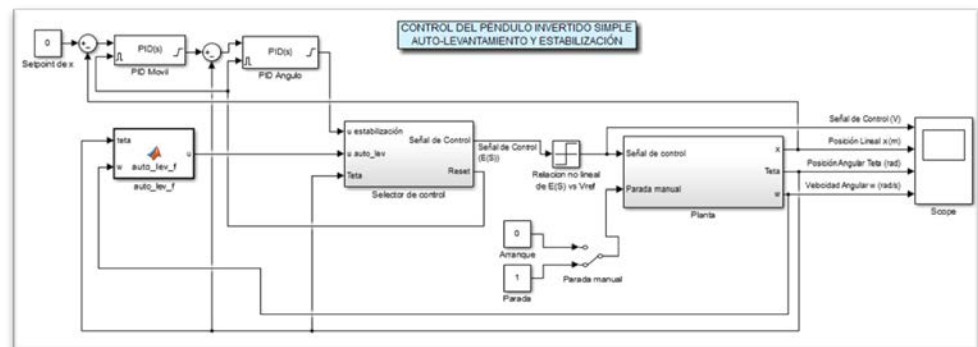


FIGURA 5.30. DIAGRAMA DE BLOQUES COMPLETO DE LA IMPLEMENTACIÓN DEL CONTROL.

5.3 Interfaz gráfica de usuario.

La interfaz gráfica en el programa de Simulink fue bastante sobria, consistiendo básicamente de los bloques de control, planta, cambio de control ya descritos en el apartado anterior, siendo de esta forma las únicas interacciones con el usuario, los cuadros de diálogo de los bloques PID, los parámetros de control del bloque del auto-levantamiento, el control manual de arranque/parada y el “scope” o visualizador de señales múltiple.

5.4 Adquisición y envío de señales en plataforma matemática de programación.

Para análisis específicos de señales se puede realizar la adquisición y procesamiento directamente en Matlab usando el estilo de programación lineal y textual del mismo, ésto como método alternativo o adicional a usar un bloque de Simulink (To Workspace) que recoja datos y los almacene en un objeto de Matlab.

Antes de escribir el programa en la ventana de edición era necesario conocer ciertos datos de la tarjeta, para ello se ingresó y ejecutó el comando `d = daq.getDevices` en la ventana de

comandos, lo que como resultado dio un número de índice, vendedor, nombre con el cual es identificada la tarjeta en Matlab, y la descripción o modelo de la tarjeta.

```
Data acquisition devices:
index Vendor Device ID      Description
-----
1      ni      Dev1      National Instruments PCI-6221
```

Ejecutando $d(n)$, donde n es el número de índice de la tarjeta de interés, se obtuvieron datos más técnicos, como tasas de muestreo, nombres de canales de medición, y tipos de medición. Los dos últimos datos sirvieron para la programación posterior.

```
ni: National Instruments PCI-6221 (Device ID: 'Dev1')
  Analog input subsystem supports:
    4 ranges supported
    Rates from 0.1 to 250000.0 scans/sec
    16 channels ('ai0' - 'ai15')
    'Voltage' measurement type

  Analog output subsystem supports:
    -10 to +10 Volts range
    Rates from 0.1 to 833333.3 scans/sec
    2 channels ('ao0','ao1')
    'Voltage' measurement type

  Digital subsystem supports:
    Rates from 0.1 to 1000000.0 scans/sec
    24 channels ('port0/line0' - 'port2/line7')
    'InputOnly','OutputOnly','Bidirectional' measurement
types

  Counter input subsystem supports:
    Rates from 0.1 to 80000000.0 scans/sec
    2 channels ('ctr0','ctrl')
```

```
'EdgeCount','PulseWidth','Frequency','Position'
measurement types
```

```
Counter output subsystem supports:
Rates from 0.1 to 80000000.0 scans/sec
2 channels ('ctr0','ctrl')
'PulseGeneration' measurement type
```

Tomando como guía el tutorial “Measure Angular Position with an Incremental Rotary Encoder”, de la página de MathWorks [26], se necesitaba primero crear una sesión con el comando `daq.createSession`

```
s = daq.createSession('ni');
```

Luego se configuraron los canales a utilizar, primeramente los contadores para los encoders siguiendo la sintaxis: `ch = addCounterInputChannel(s, 'nombre', n, medición);` donde 'nombre' es como Matlab reconoce a la tarjeta (Device ID), n el número del contador o canal y medición es el nombre del tipo de medición a hacer, de acuerdo a lo obtenido en la información usando `d(n)`.

```
ch1 = addCounterInputChannel(s, 'Dev1', 0, 'Position');
ch2 = addCounterInputChannel(s, 'Dev1', 1, 'Position');
```

Se acuerdo al tutorial para trabajar con contadores por hardware se necesita abrir un canal de entrada analógica.

```
addAnalogInputChannel(s, 'Dev1', 1, 'Voltage');
```

Luego se determinó la tasa de muestreo en número de muestras por segundo y la duración de la sesión.

```
s.Rate = 10000;  
Duración = 0.6;  
s.DurationInSeconds = Duración;
```

Se abrieron dos líneas de canales digitales, especificando que son únicamente de salida.

```
ch4 =  
addDigitalChannel(s, 'Dev1', 'Port0/Line0:1', 'OutputOnly');
```

Se habilitó un canal para salida analógica de voltaje.

```
ch5 = addAnalogOutputChannel(s, 'Dev1', 'ao0', 'Voltage');
```

Se prepararon ciertos datos para la generación, por ejemplo el tiempo, empieza en cero, con incrementos dictados por la razón de muestreo, y tiempo final dado por la duración de la sesión.

```
Tiempo = 0:(1/s.Rate):Duracion;
```


Se crearon las funciones para los canales de salida, **y1** y **y2** para las salidas digitales de dirección y habilitación del motor respectivamente. Y la función **y** que en este caso es un pulso de voltaje analógico de 2.4 Vdc con una duración de 0.6 segundos y un retraso de 1 segundo.

```
y1 = (1*(heaviside(Tiempo)));
y2 = (0*(heaviside(Tiempo)));
y = (2.4*(heaviside(Tiempo-0)-heaviside(Tiempo-0.6)));
```

Se prepararon los datos a generar.

```
queueOutputData(s,[y1.',y2.',y.']);
```

Con el siguiente comando se empezaba la adquisición/generación, almacenando los datos en el arreglo [PulsosContados, MarcasDeTiempo].

```
[PulsosContados, MarcasDeTiempo] = startForeground(s);
```

Se introdujeron las siguientes líneas de comando para establecer límites en caso de que se sature el contador y se guardan los datos de pulsos contados del encoder de posición angular y el de posición lineal en variables distintas.

```

counterNBits = 32;
signedThreshold = 2^(counterNBits-1);
signedData = PulsosContados(:,1);
signedData(signedData > signedThreshold) =
signedData(signedData > signedThreshold) - 2^counterNBits;

counterNBits = 32;
signedThreshold = 2^(counterNBits-1);
signedDatapendl = PulsosContados(:,2);
signedDatapendl(signedDatapendl > signedThreshold) =
signedDatapendl(signedDatapendl > signedThreshold) -
2^counterNBits;

```

Se efectuó la conversión de pulsos a posición lineal en mm, y de pulsos a grados para la posición angular.

```

PPR = 2500;
PosicionGrados = signedData * 360/PPR;
PerimetroPolea = 249.69;
AvanceLineal = PerimetroPolea * PosicionGrados/360;
PosicionGradospendl = signedDatapendl * 360/PPR;

```

Se prepararon los datos a graficar, y posteriormente se introdujeron los comandos para graficar el escalón de voltaje, y la posición angular y lineal.

```

PosicionAngular = PosicionGradospendl.';
figure
subplot (4,1,1);
plot(Tiempo,y);
hold on;
title ('Señal escalón de velocidad');
xlabel('Tiempo (s)');
ylabel('Voltaje (Vdc)');
ylim([-1 3]);
grid on
grid minor

```

```

subplot (4,1,2);
plot(MarcasDeTiempo, AvanceLineal,'g');
title ('Respuesta de la posición del móvil a la señal
escalón de velocidad');
xlabel('Tiempo (s)');
ylabel('Posición del móvil (mm)');
ylim([-1200 1200]);
grid on
grid minor

subplot (4,1,3);
plot(MarcasDeTiempo, PosicionGradospend1,'r');
title ('Respuesta de la posición del péndulo 1 a la señal
escalón de velocidad');
xlabel('Tiempo (s)');
ylabel('Posición del péndulo 1 (grados)');
grid on
grid minor

hold off

```

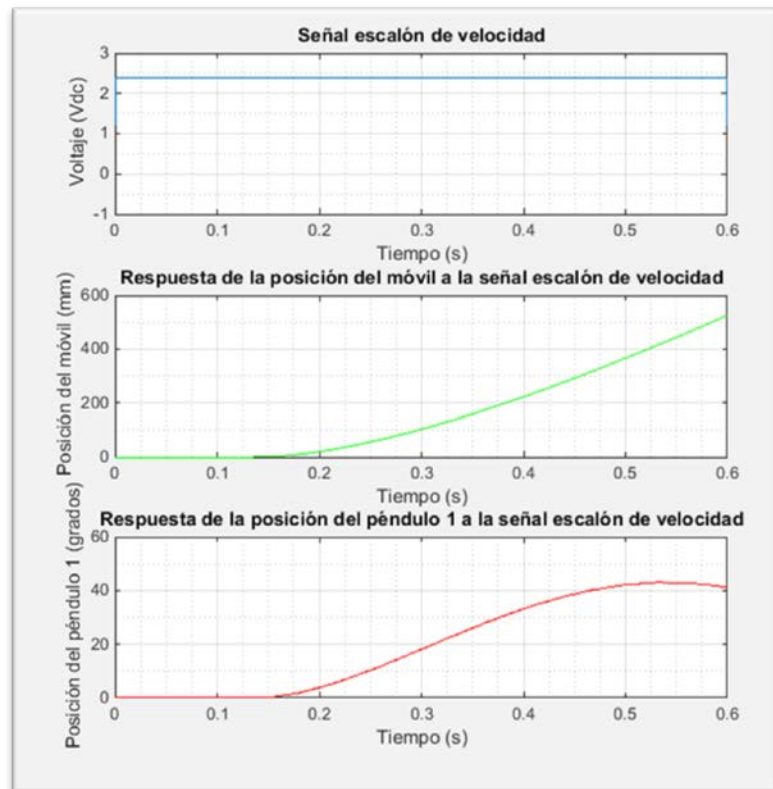
Por último, se emitieron datos a las salidas digitales y analógica para finalizar el programa, esto se realizó especialmente para terminar el programa con una señal digital de parada y evitar luego de terminada la ejecución el motor se siga moviendo.

```

queueOutputData(s,[0.',0.',1.']);
startForeground(s);

```

A continuación se muestra una gráfica obtenida siguiendo los pasos anteriores, el programa completo puede consultarse en el Anexo H.



**FIGURA 5.31. GRÁFICA DE ADQUISICIÓN DE DATOS
OBTENIDA CON MATLAB.**

CAPÍTULO 6

6. ANÁLISIS DE DATOS EXPERIMENTALES

6.1 Análisis del desempeño del controlador en el modelo real.

Sean los valores de parámetros de control usados en la simulación del modelo no lineal:

Auto-levantamiento: $n_g=1.25$, $K=3$

PID interno: $P = -80$, $I = -40$, $D = -20$ (saturado en ± 15.73)

PID externo: $P = 0.1$, $I = 0.05$, $D = 0.2$ (saturado en ± 0.15)

Como resultado se obtuvo que el equipo se quedó estancado en la etapa de auto-levantamiento, en la figura 6.1 pueden observarse las señales de control, posición lineal, posición angular y velocidad angular, al inicio existió un ligero crecimiento de la amplitud de oscilación del péndulo

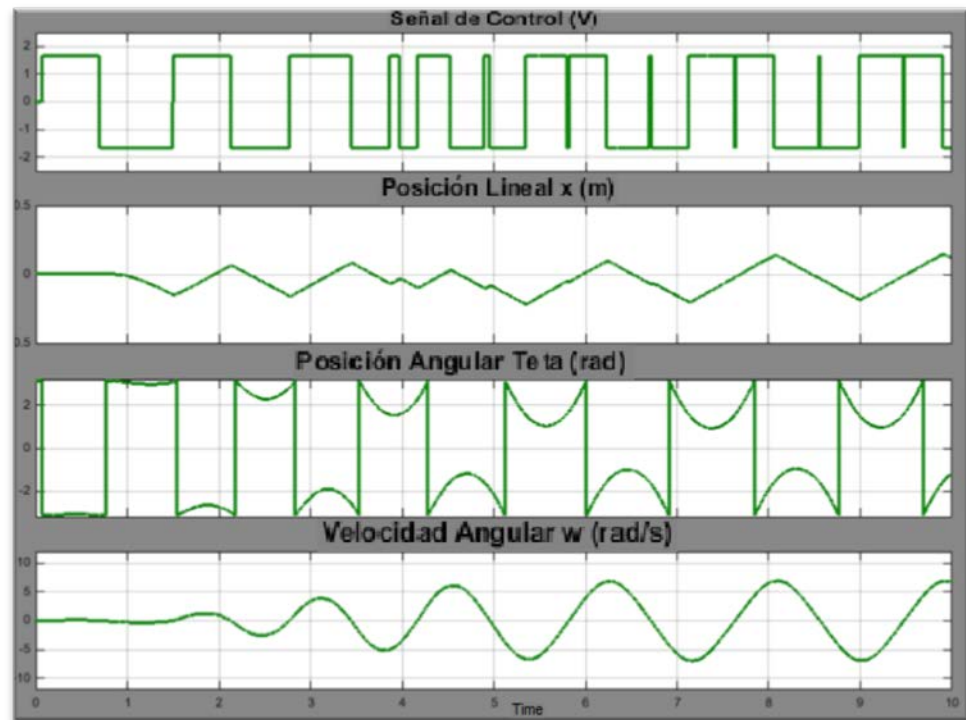


FIGURA 6.1. INICIO DE AUTO-LEVANTAMIENTO EN MODELO REAL (NG=1.25, K=3).

Sin embargo, pasados ya los 30 segundos dentro de los cuales se supone debía ocurrir el auto-levantamiento, tanto el carro como el péndulo entraron en oscilación sostenida con un periodo algo menor de 2 s y con una amplitud de alrededor de 0.15 m para la posición lineal y un rango de oscilación de ± 2 radianes, como puede observarse en la figura 6.2.

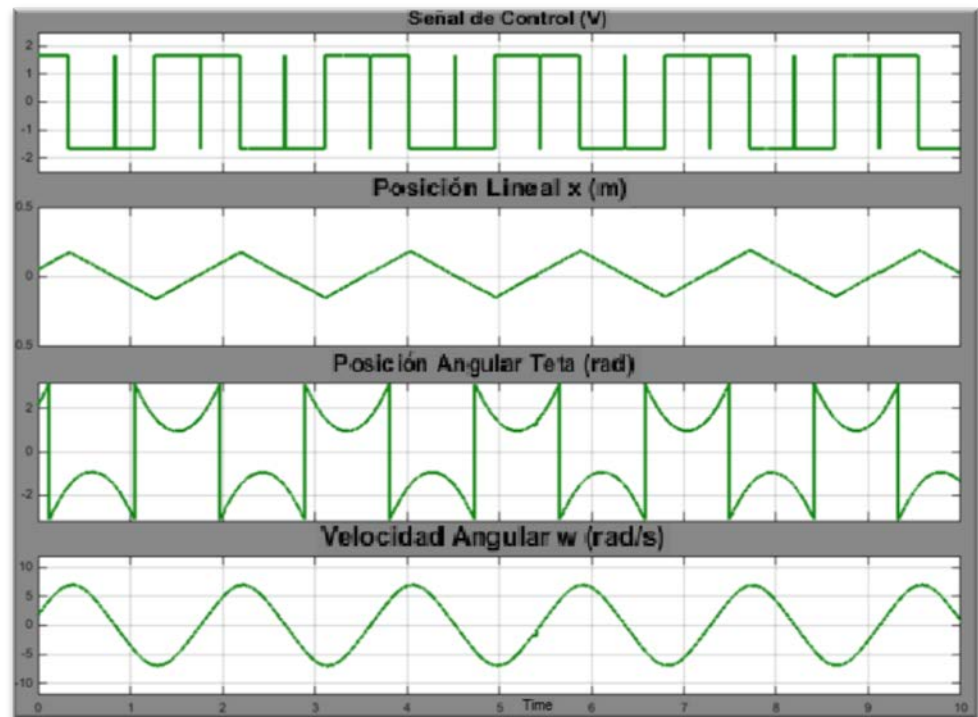
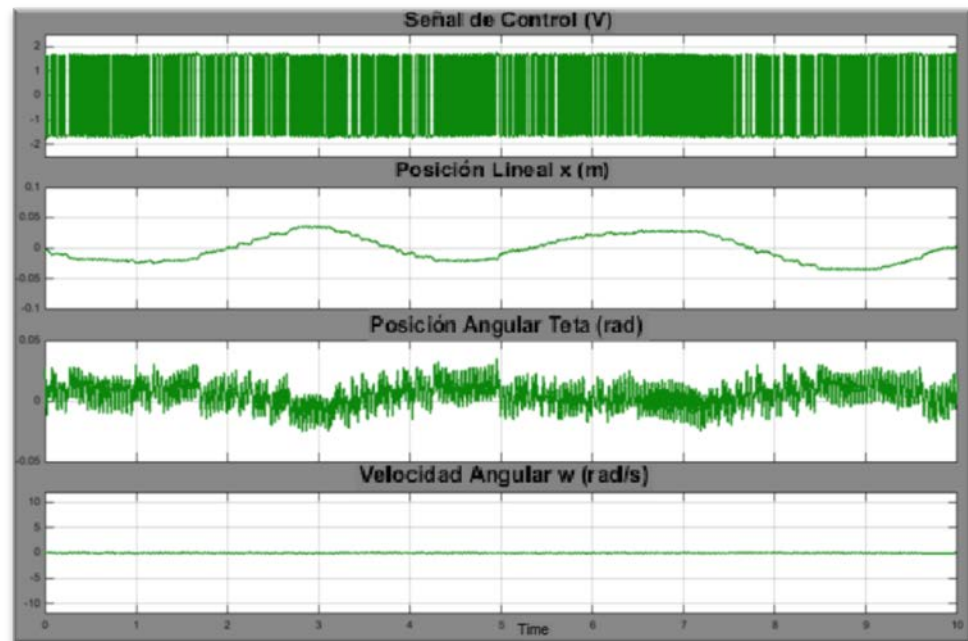


FIGURA 6.2. OSCILACIÓN SOSTENIDA DEL AUTO-LEVANTAMIENTO EN MODELO REAL ($NG=1.25$, $K=3$).

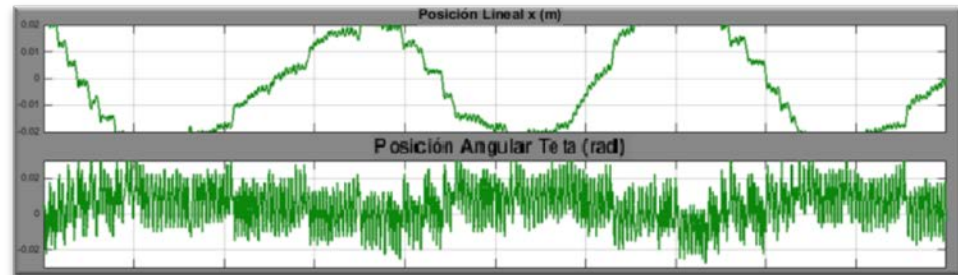
Los parámetros del bloque de auto-levantamiento fueron ajustados en la segunda sección de este capítulo, para poder evaluar al control de estabilización, se necesitó deshabilitar temporalmente al auto-levantamiento, lo cual se logró encerrando la ganancia ($K=0$), se lleva el péndulo manualmente a la zona de control (± 0.45 radianes) y se evalúa el desempeño.

En la figura 6.3 puede verse el error de estado estable para la posición lineal y la posición angular, junto con la señal de control y la velocidad angular.



**FIGURA 6.3. ERROR DE ESTADO ESTABLE DEL CONTROL
DISEÑADO EN EL MODELO REAL.**

Detalle del error de estado estable mostrado en la figura 6.3 con los límites de las gráficas siendo los límites de las restricciones de diseño (figura 6.4).



**FIGURA 6.4. ERROR DE ESTADO ESTABLE DEL CONTROL
DISEÑADO EN EL MODELO REAL AMPLIFICADO.**

De las figuras 6.3 y 6.4 puede notarse que el error de estado estable para la posición lineal sobrepasó al límite de diseño llegando a valores en el rango de ± 0.03 m, un 50% por encima del límite de ± 0.02 m.

En cuanto a la posición angular del péndulo, parece estar ajustadamente dentro de los límites impuestos por los requerimiento de diseño ± 0.03 radianes.

La figura 6.5 muestra cómo reaccionó el sistema ante una perturbación en el ángulo del péndulo, donde a la posición lineal le tomó alrededor de 7 segundos estabilizarse, 40% por encima del límite de 5 segundos.

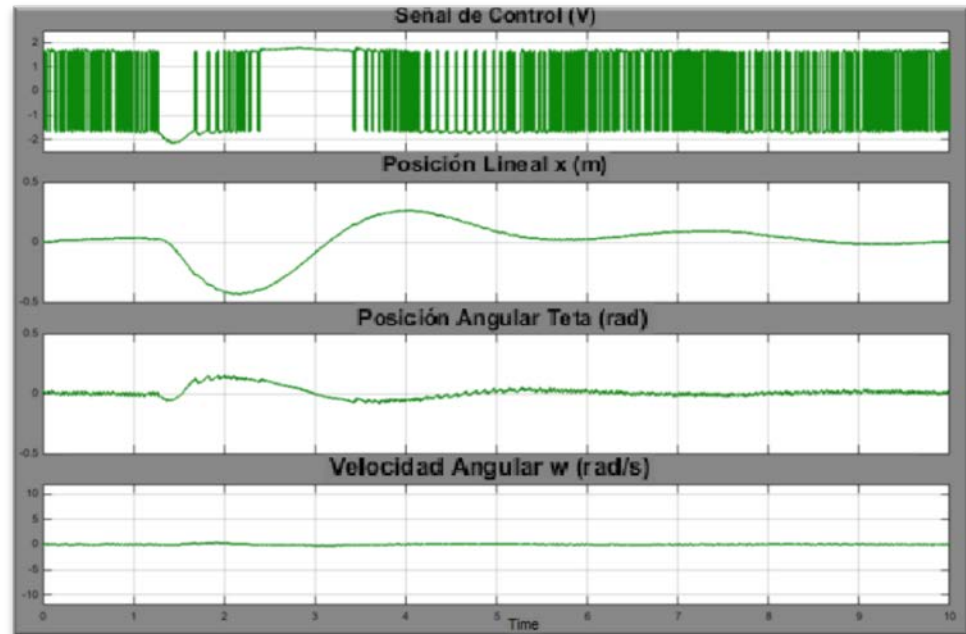


FIGURA 6.5. RESPUESTA A LA PERTURBACIÓN DEL CONTROL DISEÑADO EN EL MODELO REAL.

Si la perturbación llegaba a cierto límite el péndulo caía, aunque era recuperado después de una vuelta, lo que indica falta de rapidez de respuesta para vencer a las perturbaciones (figura 6.6).

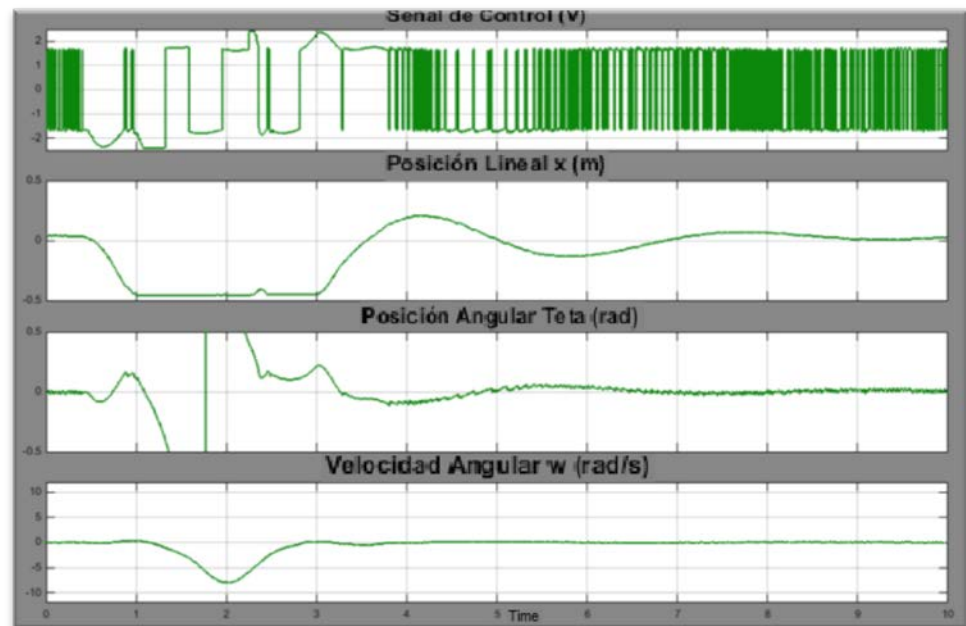


FIGURA 6.6. RESPUESTA A LA PERTURBACIÓN CON CAÍDA Y RECUPERACIÓN DEL CONTROL DISEÑADO EN EL MODELO REAL.

Cabe analizar ahora, la divergencia entre la posición del carro de acuerdo al modelo simulado y la posición real, y la gran diferencia entre los ganancias del control PID interno obtenidas usando Sisotool y las obtenidas por prueba y error en la simulación.

Factores como las asunciones y simplificaciones hechas para obtener un modelo, están casi siempre presentes, especialmente en modelos lineales, por ejemplo considerar a la fricción como puramente viscosa en lugar de considerar las aportaciones de la

fricción de Coulomb y la fricción estática, esto con el propósito de obtener un modelo linealizado.

También existieron ciertas consideraciones sobre elementos del sistema que tienen restricciones en la realidad que les hace comportarse diferente de sus modelos matemáticos.

Por ejemplo un motor DC, al recibir una entrada tipo paso unitario de voltaje de entrada, presenta un pico de torque de salida, cuyo valor máximo puede ser de alrededor de 6 veces el torque de salida estable. Para poder observar ese comportamiento se parte del modelo de la figura 3.10, y se añade un bloque de visualización para las señales de torque a la salida del motor y la fuerza, como se muestra en la figura 6.7.

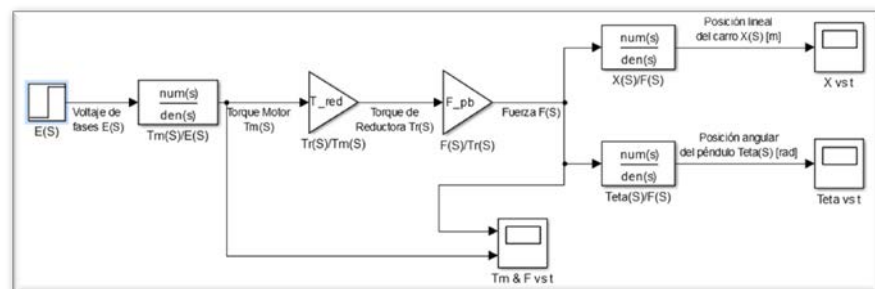


FIGURA 6.7. DIAGRAMA DE BLOQUES DE SISTEMA LINEALIZADO CON VISUALIZACIÓN DE TORQUE MOTOR Y FUERZA.

Luego de ejecutar la simulación del diagrama anterior para una duración de 2 segundos se obtuvieron las gráficas de torque y fuerza para una entrada escalón unitario de voltaje (figura 6.8).

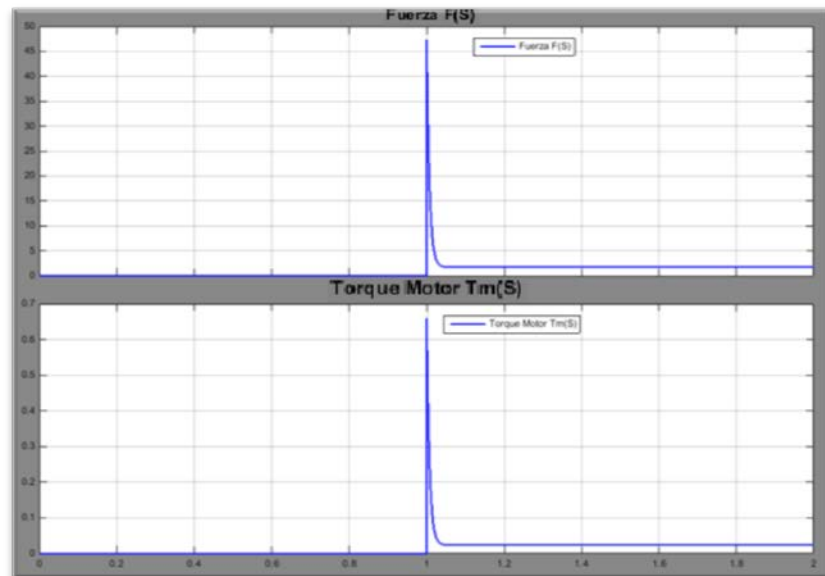


FIGURA 6.8. FUERZA SOBRE EL CARRO Y TORQUE DEL MOTOR PARA UNA ENTRADA ESCALÓN UNITARIO DE VOLTAJE.

Estos picos 0.65 Nm del torque de salida del motor y consecuentemente los 47 N de fuerza sobre el carro están varias órdenes de magnitud por encima de los valores estables, en los modelos teóricos simulados estos picos se terminan manifestando en la posición lineal del carro como el efecto de una gran empuje inicial. En el equipo real esto no sucedía pues tanto la fuente de

voltaje y el driver del motor tienen limitadores de corriente (que está directamente relacionada con el torque por la constante de torque).

Adicionalmente, el driver del motor permitía elegir entre varios perfiles de rampa, siendo el más corto y en el que está configurado el de 500 ms. Esto ayuda a atenuar el pico formado como se puede ver en la figura 6.9 donde la entrada tiene una amplitud de 1 pero con un perfil inicial tipo rampa de 500 ms.

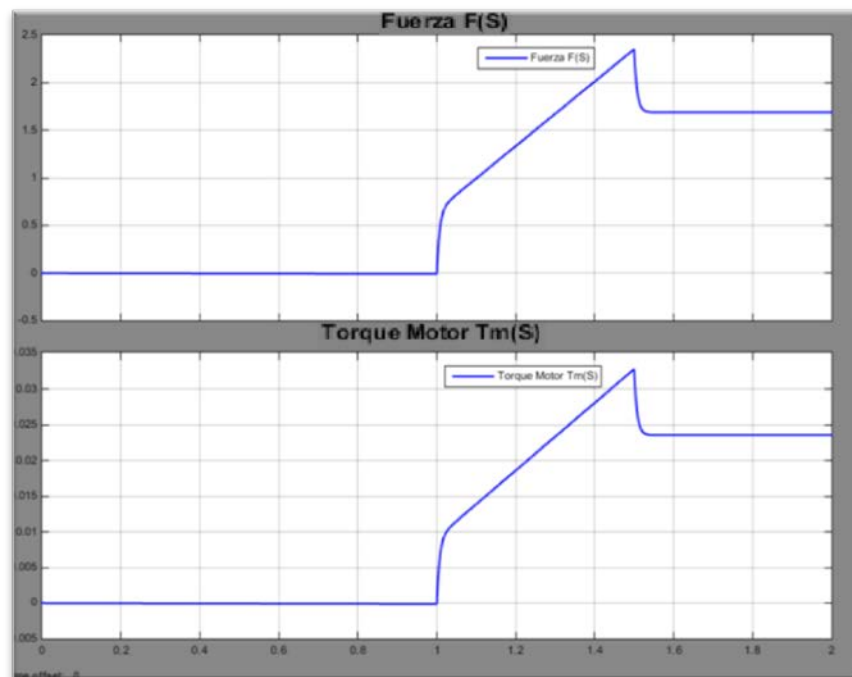


FIGURA 6.9. FUERZA SOBRE EL CARRO Y TORQUE DEL MOTOR PARA UNA ENTRADA UNITARIA CON PERFIL RAMPA DE 500 MS.

Con el perfil rampa de 500 ms, se logró atenuar los picos a 0.032 Nm en el torque de salida del motor y 2.4 N en la fuerza sobre el carro.

La presencia de esos picos de fuerza podría ser la causante de los altos desplazamientos en la posición lineal (inyecciones bruscas y altas de fuerza empujando el carro) y que las ganancias de los controladores hayan resultado más bajas de lo que se necesitó en el equipo real, es decir de alguna manera los picos compensaban y hacían aparentemente innecesarias mayores ganancias en los controladores

6.2 Ajustes finales del controlador.

Primero se realizó el ajuste final del control de estabilización, una vez lograda una respuesta satisfactoria del sistema que cumpla con los requerimientos de diseño para error de estado estable y rechazo a la perturbación se procedió a ajustar los parámetros del control de auto-levantamiento.

El ajuste de la estabilización se lo hizo en forma parecida al proceso delineado en el capítulo acerca del diseño de los controladores, se

deshabilitó el control del lazo externo para poder ajustar el control del lazo interno, luego se ajustaron los parámetros del control maestro con el control esclavo -previamente ajustado- habilitado.

Se probó a aumentar la acción proporcional para mejorar el tiempo de respuesta ante perturbaciones, en conjunto también se aumentó la acción integral para reducir el error de estado estable provocado por la mayor ganancia proporcional.

La ganancia proporcional pasó de -80 a -100, mientras que la ganancia integral pasó de -40 a -50, la ganancia derivativa se mantuvo en -20.

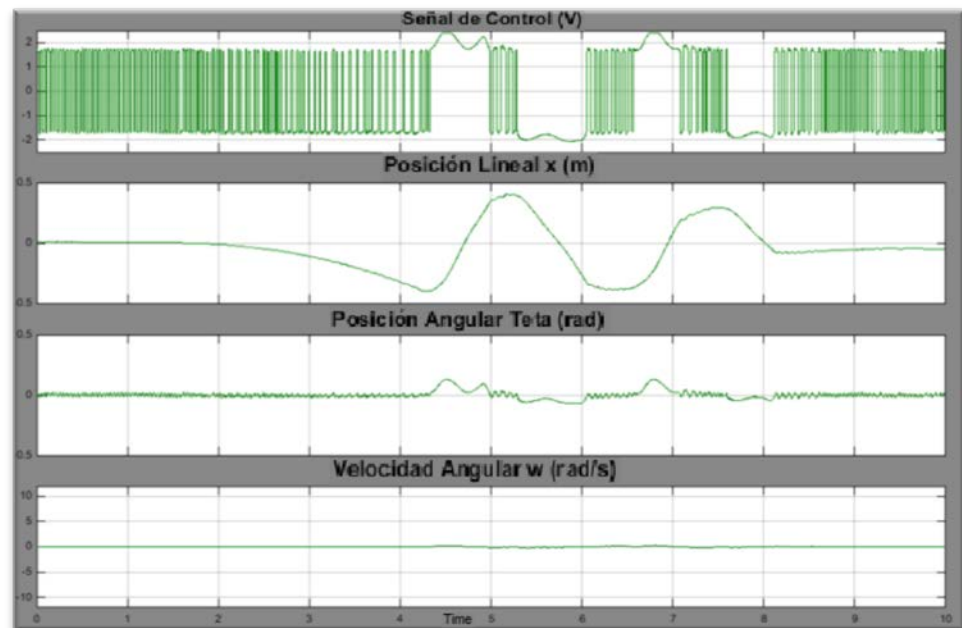


FIGURA 6.10. RESPUESTA A LA PERTURBACIÓN DEL SISTEMA CON CONTROL PID INTERNO ($P=-100$, $I=-50$, $D=-20$).

Con el control interno ajustado, se procedió a sintonizar las ganancias del PID externo, prefiriendo empezar de cero y una vez alcanzada una respuesta satisfactoria del sistema, dichas ganancias fueron comparadas por las obtenidas en el capítulo 4.

Se empezó por la acción proporcional, hallando una tendencia clara entre el aumento de ésta y una oscilación creciente de la posición lineal y en consecuencia también de la posición angular del péndulo, que aumentaban. A mayor valor de P , mayor era la tasa de crecimiento de la oscilación y a mayor frecuencia cada vez, en la

figura 6.11 puede observarse la respuesta de la posición lineal para diferentes valores de P.

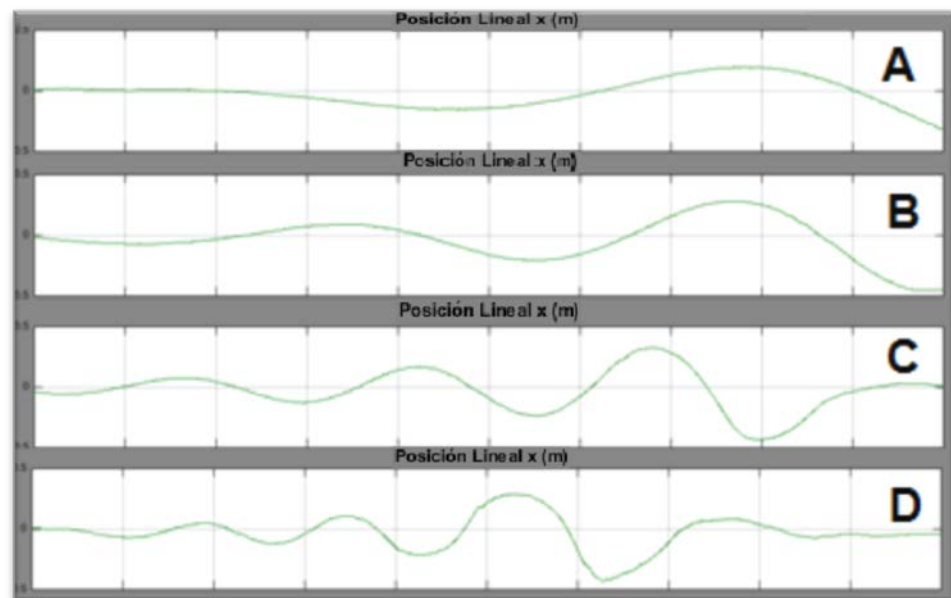


FIGURA 6.11. POSICIÓN DEL CARRO CON CONTROL EXTERNO CON A) $P=0.1$ B) $P=0.2$ C) $P=0.5$ Y D) $P=1$.

Se tomó como base el valor de $P=0.2$. Para mejorar la estabilidad del sistema se introdujo acción derivativa, donde también se producía una tendencia, a mayor valor de D, más disminución de las oscilaciones. Sin embargo un valor demasiado alto de D haría que el sistema se comporte de forma errática, especialmente cuando la señal tiene ruido, lo cual en parte se alivió con el uso del filtro que el mismo bloque PID tiene, a menor valor de N, menos

ruidosa la señal en la parte derivativa, se eligió seguir trabajando con $N=2$.

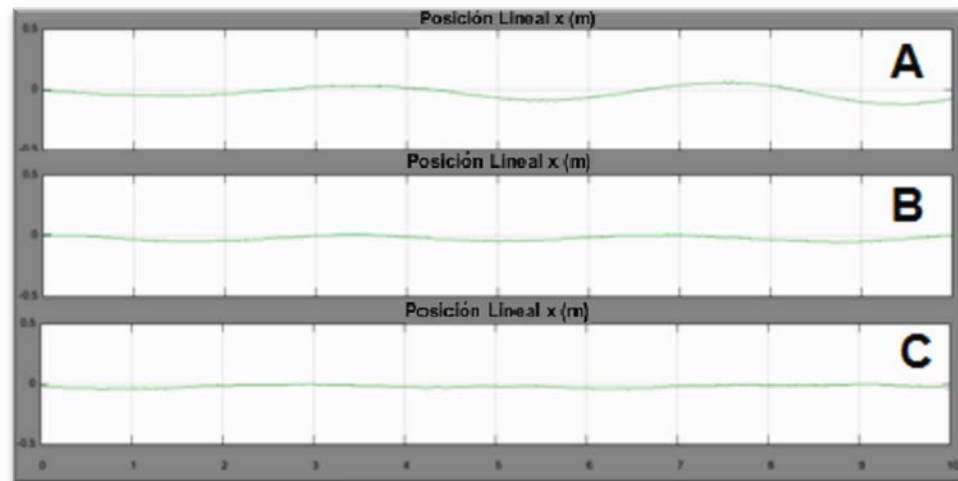


FIGURA 6.12. POSICIÓN DEL CARRO CON CONTROL EXTERNO CON A) $D=0.05$ B) $D=0.1$ Y C) $D=0.2$.

Si bien el sistema se volvió más estable, se presentó un error de estado estable (el carro tendía no a centrarse sino a moverse hacia la izquierda) que es más apreciable amplificando la vista en la gráfica, esto es, configurando los ejes para que muestren la señal en el rango de ± 0.1 m. Dicho error disminuyó al aumentar aún más la acción derivativa hasta $D=0.3$ (ver figura 6.13).

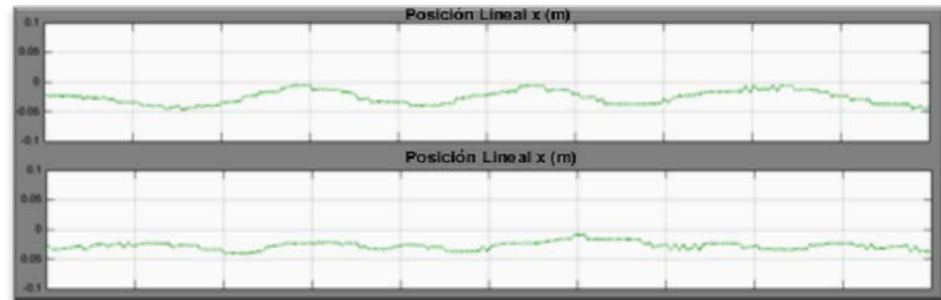


FIGURA 6.13. POSICIÓN DEL CARRO CON CONTROL EXTERNO CON $D=0.2$ (SUPERIOR) $D=0.3$ (INFERIOR).

Por último se introduce la acción integral para reducir el error de estado estable, el efecto de la acción integral era más notorio cuando el sistema era perturbado, en la figura 6.14 se puede ver el efecto de diferentes valores de la ganancia integral sobre cómo actuó el sistema –enfocado en la posición lineal para este análisis– ante una perturbación.

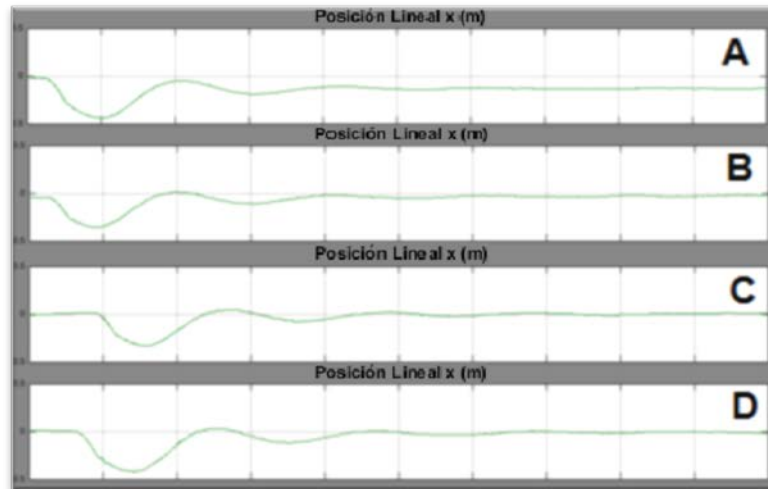


FIGURA 6.14. POSICIÓN DEL CARRO CON CONTROL EXTERNO CON: A) $I=0$ B) $I=0.01$ C) $I=0.02$ Y D) $I=0.04$.

En la figura 6.15 se muestra el comportamiento del sistema en estado estable bajo el control corregido.

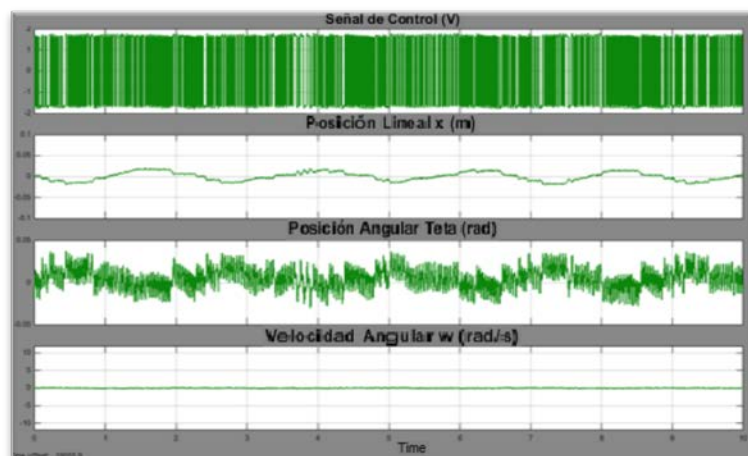


FIGURA 6.15. ERROR DE ESTADO ESTABLE DEL CONTROL CORREGIDO.

En la figura 6.16 se muestra el detalle amplificado de la posición lineal y angular donde los límites de visualización son las restricciones de diseño, a diferencia de lo visto en la figura 6.4, esta vez las señales se mantuvieron dentro sus rangos, especialmente la posición lineal.

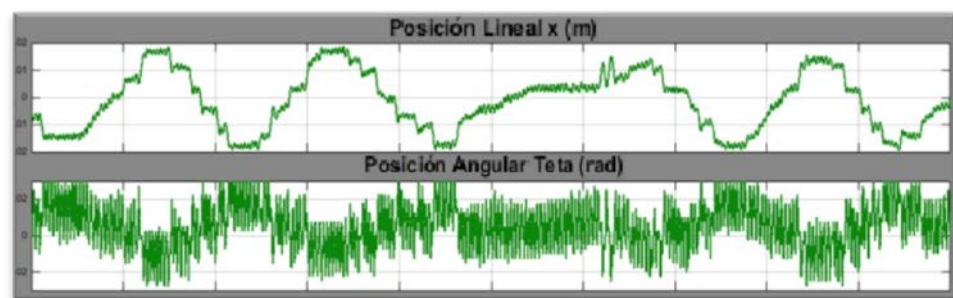


FIGURA 6.16. ERROR DE ESTADO ESTABLE DEL CONTROL CORREGIDO AMPLIFICADO.

La respuesta a la perturbación también mejoró, con tiempo de estabilización apenas mayor de 5 segundos (figura 6.17)

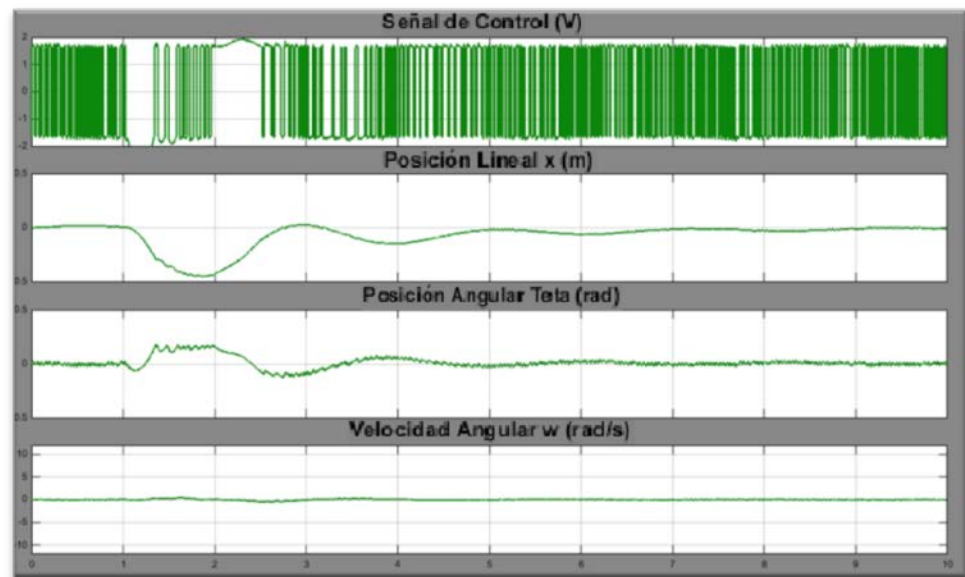


FIGURA 6.17. RESPUESTA A LA PERTURBACIÓN DEL CONTROL CORREGIDO.

Con el control en cascada para la estabilización ya sintonizado, se ajustaron los valores de los parámetros del control de auto-levantamiento, en las primeras pruebas se vio que el sistema entraba en una oscilación sostenida, por tanto se necesitaba aumentar la salida para inyectar más energía al sistema en menos tiempo. Los límites en la salida del control del auto-levantamiento estaban dados por el valor de n_g , en la figura 6.18 se ve el efecto de usar $n_g=2.5$.

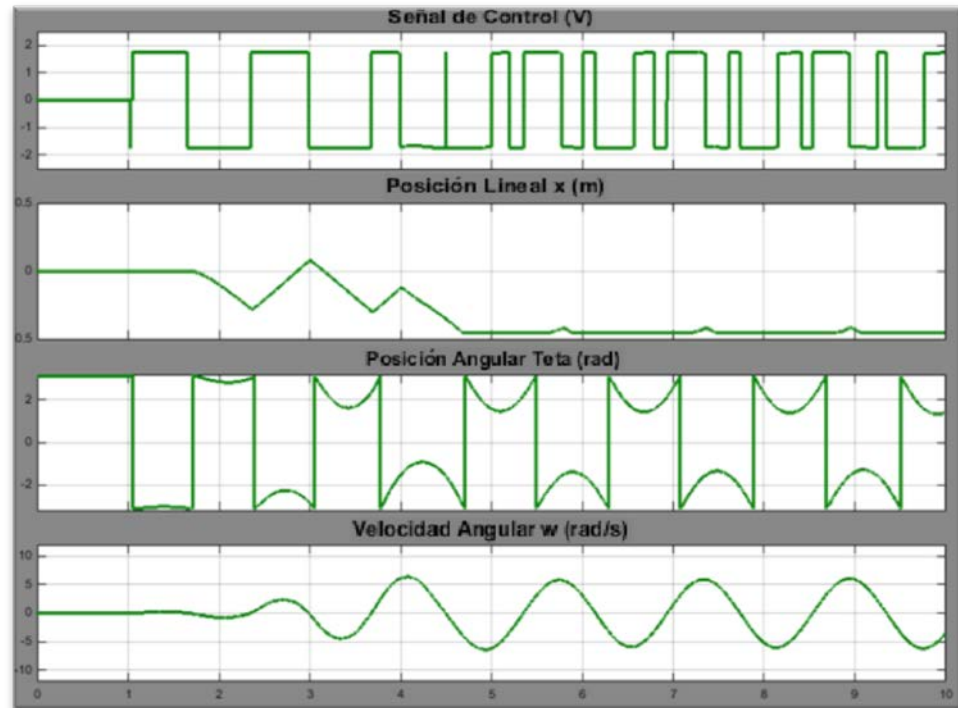


FIGURA 6.18. AUTO-LEVANTAMIENTO FALLIDO CON $NG=2.5$.

Como puede verse en la imagen anterior, el carro se movió unas cinco veces tendiendo hacia la izquierda, hasta que llegó al tope un poco antes de 4 s contados desde el inicio (1 s) del intento de auto-levantamiento. El mismo tipo de comportamiento se notó para valores de ng entre 3 y 7.

Pero es con $ng=8$ que se produjo un auto-levantamiento exitoso, en corto tiempo (alrededor de 5 s), sin que el carro llegue a sus límites laterales (ver figura 6.19).

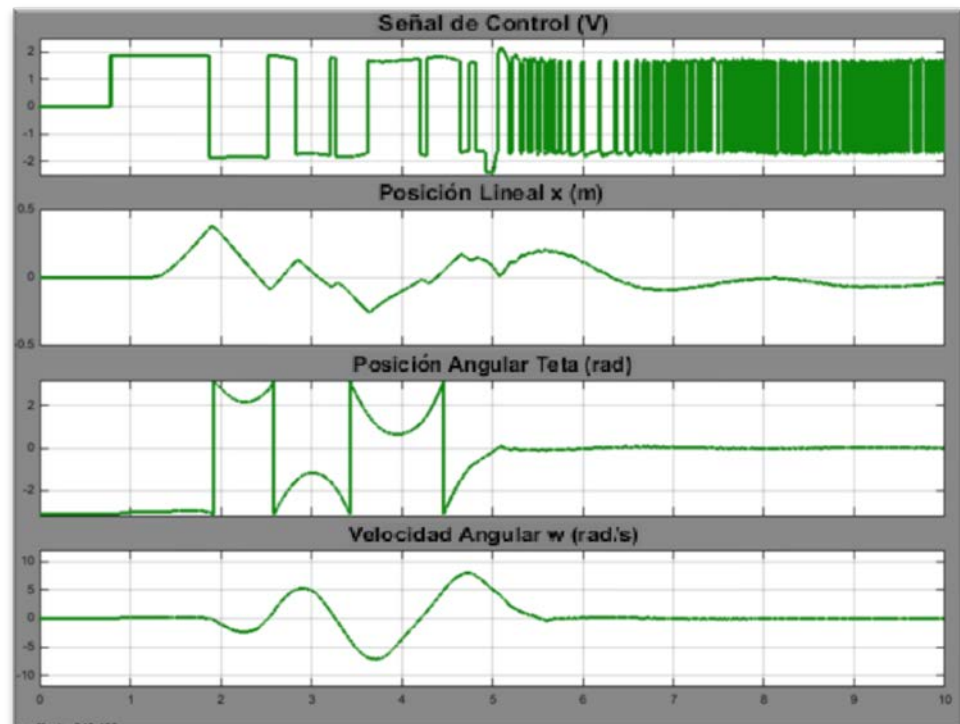


FIGURA 6.19. AUTO-LEVANTAMIENTO EXITOSO CON NG=8.

Luego, a pesar de haberse logrado el auto-levantamiento, se modificó el valor de la ganancia K que se había mantenido en 3 dejando como constante $ng=8$, en esperas de alguna mejora adicional.

Al probar con valores menores a 3, se pudo notar que se presentaba una tendencia del móvil o carro a arrinconarse en uno de los extremos, al parecer causado por no inyectar la energía suficiente de tal modo que el péndulo se balanceara de tal forma que los cambios de dirección del movimiento lineal se dieran dentro

de los límites del sistema, esto puede verse en la figura 6.20 para $K=2$.

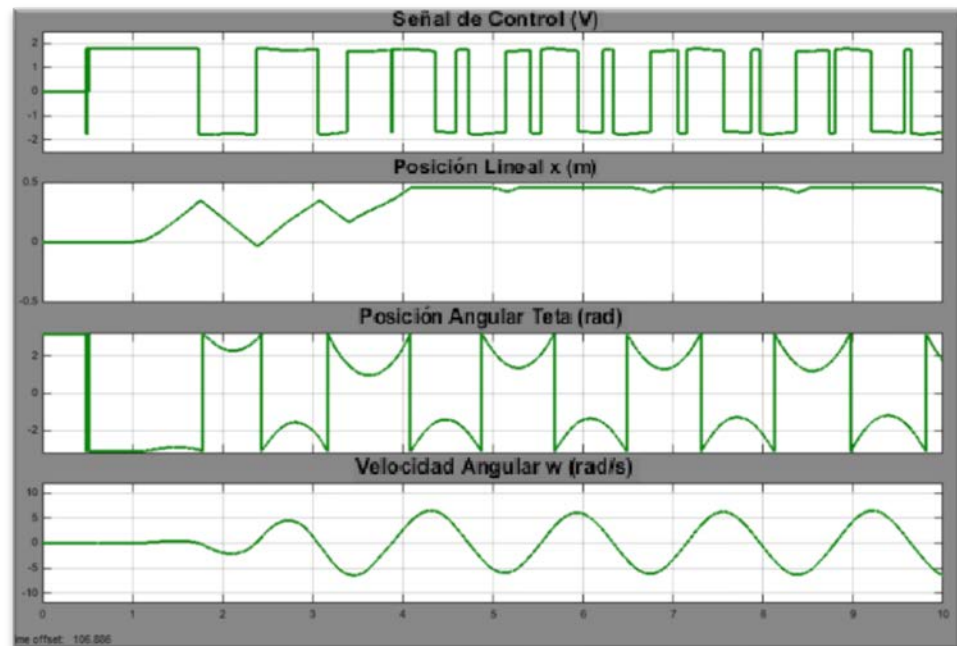
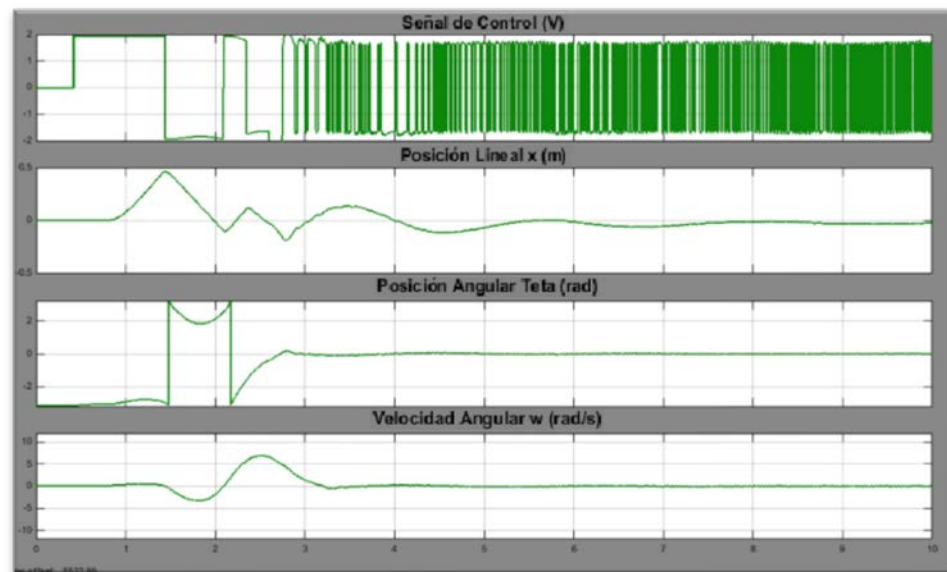


FIGURA 6.20. AUTO-LEVANTAMIENTO FALLIDO CON $NG=8$ Y $K=2$.

Por otro lado, al usar un valor de K mayor a 3, el auto-levantamiento se dio mucho más rápido y empleando menos oscilaciones del péndulo, en un tiempo de alrededor de 2.5 s, reduciendo a la mitad su tiempo de ejecución en comparación con el mostrado en la figura 6.19. En la figura 6.21 puede observarse la ejecución del auto-levantamiento con valor de $K=4$.



**FIGURA 6.21. AUTO-LEVANTAMIENTO EXITOSO Y MEJORADO
CON $NG=8$ Y $K=4$.**

Algo de notar es que al ser el control de auto-levantamiento un proceso con parámetros calibrados, éste mostró ser un poco dependiente de las condiciones iniciales, al necesitar el sistema una excitación inicial por medio de un leve golpe al péndulo, si el golpe era fuerte se afectaba el desempeño de éste, produciendo un efecto parecido a usar un valor de $K=2$ como en la figura 6.20, con el carro arrinconándose en uno de los extremos.

Control PID interno: $P = -100$, $I = -50$, $D = -20$, (saturado en ± 15.73)

Control PID externo: $P = 0.2$, $I = 0.04$, $D = 0.3$, (saturado en ± 0.15)

Control de auto-levantamiento: $ng = 8$, $K = 4$.

CAPÍTULO 7

7. CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

Con las ganancias de controladores PID ajustadas de acuerdo al modelo no lineal simulado (sección 4.4, valores resumidos al inicio de sección 6.1) se pudo llevar a cabo la estabilización del péndulo (sección 6.1, figuras 6.3, 6.4 y 6.5), aunque no con los requerimientos de diseño (sección 4.2), por lo cual necesitó un posterior proceso de ajuste (sección 6.2).

Se logró de manera exitosa usar un algoritmo de control de auto-levantamiento originalmente diseñado para un péndulo invertido rotatorio aunque los valores de los parámetros para el auto-levantamiento hallados y probados en la simulación (sección 4.3) fueron insuficientes en el modelo real (sección 6.1, figuras 6.1 y 6.2), necesitando posterior

ajuste (sección 6.2, figura 6.21), siendo además el algoritmo usado un poco sensible a las condiciones iniciales del sistema (sección 6.2).

El modelo matemático necesita ser revisado, especialmente la etapa de actuación y el efecto de los picos de fuerza actuante sobre el carro (sección 6.1, figuras 6.8 y 6.9) y consecuentemente su posición lineal.

Recomendaciones:

Diseñar un filtro tipo pasa bajos para la señal de posición angular antes de ser retroalimentada como una forma de reducir la cantidad de los pequeños movimientos rápidos que colaboran en parte con las vibraciones del sistema.

Usar una estrategia de control de auto-levantamiento que tome en cuenta el desplazamiento lineal del carro en su retroalimentación, de tal forma que se tenga control sobre ésta y se use la distancia disponible sin llegar a sobrepasar los límites de los extremos.

Entre las recomendaciones para mejorar el desempeño del equipo están: introducir un par de sensores para banda ubicados en la base para reducir ruido y vibraciones de ésta, estabilizar los ejes guías usando al

menos tres apoyos puntuales para reducir el pandeo que hace mover al carro hacia atrás o adelante, también se necesita ajustar la plancha o base de la mesa para reducir el ruido, por último es muy recomendable usar un sistema porta-cables flexible tipo "oruga" para el cable del encoder de posición angular del péndulo de tal forma que éste no quede colgando y roce en el filo de la mesa.

APÉNDICES

APÉNDICE A

HOJA DE DATOS DE LA FUENTE



AC/DC Single Output Power Supply, 600W
Mean Well SE-600 Series



- Features :
- AC input active surge current limiting
 - AC input range selected by switch
 - Protections: Short circuit / Overload / Over voltage / Over temperature
 - Forced air cooling by built-in DC fan
 - High power density 4.9w/inch³
 - Built-in remote sense function
 - UL approved
 - Low cost
 - 2 years warranty



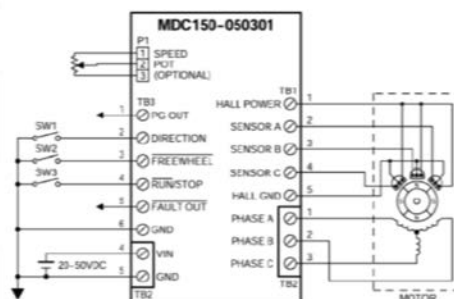
SPECIFICATION

MODEL	SE-600-5	SE-600-12	SE-600-15	SE-600-24	SE-600-27	SE-600-36	SE-600-48	
OUTPUT	DC VOLTAGE	5V	12V	15V	24V	27V	36V	48V
	RATED CURRENT	100A	50A	40A	25A	22.2A	16.6A	12.5A
	CURRENT RANGE	0 ~ 100A	0 ~ 50A	0 ~ 40A	0 ~ 25A	0 ~ 22.2A	0 ~ 16.6A	0 ~ 12.5A
	RATED POWER	500W	600W	600W	600W	599.4W	597.6W	600W
	RIPPLE & NOISE (max.) Note.2	150mVp-p	150mVp-p	150mVp-p	150mVp-p	150mVp-p	200mVp-p	200mVp-p
	VOLTA GE ADJ. RANGE	4.75 ~ 5.5V	10 ~ 13.5V	13.5 ~ 16.5V	22 ~ 26.4V	24 ~ 30V	32 ~ 40V	43 ~ 56V
	VOLTAGE TOLERANCE Note.3	±2.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%
	LINE REGULATION	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	LOAD REGULATION	±2.0%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	SETUP, RISE TIME	100ms, 50ms/230VAC 100ms, 50ms/115VAC at full load						
HOLD UP TIME (Typ.)	20ms/230VAC 18ms/115VAC at full load							
INPUT	VOLTA GE RANGE	90 ~ 132VAC / 180 ~ 264VAC selected by switch			254 ~ 370VDC			
	FREQUENCY RANGE	47 ~ 63Hz						
	EFFICIENCY (Typ.)	78%	83%	84%	87%	87%	87%	88%
	AC CURRENT (Typ.)	12A/115VAC 7.5A/230VAC						
	INRUSH CURRENT (Typ.)	30A/115VAC 60A/230VAC						
	LEAKAGE CURRENT	<2.0mA/ 240VAC						
PROTECTION	OVERLOAD	105 ~ 125% rated output power Protection type : Shut down o/p voltage, re-power on to recover						
	OVER VOLTAGE	5.75 ~ 6.3V	13.8 ~ 16.2V	18 ~ 21V	27.6 ~ 32.4V	31 ~ 36.5V	42 ~ 50V	57.6 ~ 67.2V
	OVER TEMPERATURE	85°C ±5°C (TSW) detect on heatsink of power transistor Protection type : Shut down o/p voltage, recovers automatically after temperature goes down						
ENVIRONMENT	WORKING TEMP.	-20 ~ +60°C (Refer to output load derating curve)						
	WORKING HUMIDITY	20 ~ 90% RH non-condensing						
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH						
	TEMP. COEFFICIENT	±0.05%/°C (0 ~ 50°C)						
SAFETY & EMC	VIBRATION	10 ~ 500Hz, 2G 10min/1cycle, 60min. each along X, Y, Z axes						
	SAFETY STANDARDS	UL60950-1 approved						
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC	I/P-FG:1.5KVAC	O/P-FG:0.5KVAC				
OTHERS	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms/500VDC						
	MTBF	197K hrs min. MIL-HDBK-217F (25°C)						
	DIMENSION	247*127*63.5mm (L*W*H)						
NOTE	PACKING	2.1Kg: 6pcs/13.4Kg/1.03CUFT						
		1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uF & 4.7uF parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation.						

APÉNDICE B

EXTRACTOS DE LA HOJA DE DATOS DEL DRIVER

Power Requirements:	20 - 50 VDC
Output Current Range:	10.0 - 30.0 Amps (Peak) 1.0 - 5.0 Amps (Continuous)
Hall Sensor Power Output:	6.25V @ 30mA (Max.)
Control Inputs:	(TB3, Pin2-4) TTL-CMOS Compatible Logic "0" = 0 - 0.8VDC Logic "1" = Open
Freewheel:	(TB3, Pin 3) Logic "1" (open) - Motor is Enabled Logic "0" = Motor is de-energized and will coast
Fault Output:	(TB3, Pin 5) Logic "1" (5V out) - normal operation Logic "0" - 1 of 3 fault conditions in 'Fault Protection' section
Run/Stop:	Logic "1" (open) - Motor will not run and will decelerate if running Logic "0" - Motor will run and accelerate according to ramp dip switch setting
Operating Temperature:	0°C to +70° C
Driver Type:	Bipolar, Compatible with 4, 6, and 8 Lead Motors. Series or Parallel connection.

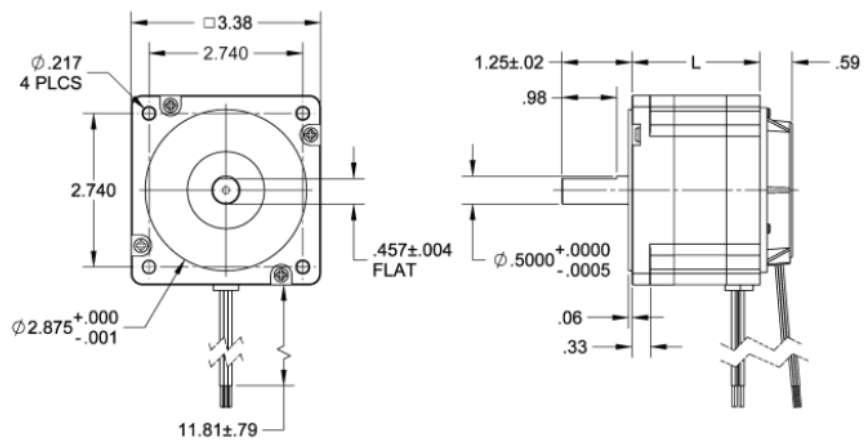


APÉNDICE C

EXTRACTOS DE LA HOJA DE DATOS DEL MOTOR

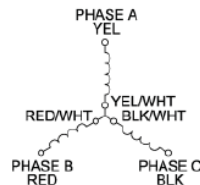
Model #	Rated Voltage (V)	Rated Speed (RPM)	Rated Power (W)	Line to Line Resistance (ohms)	Line to Line Inductance (mH)	Torque Constant (oz-in/A)	Motor Efficiency	Back EMF Voltage (V/kRPM)	Rotor Inertia (oz-in-sec ²)	Weight (lbs)	"L" Length (in)
BLY343S-48V-3200	48	3200	440	0.18	0.61	17.0	86.3%	9.31	0.02266	5.73	3.81

Single Shaft:

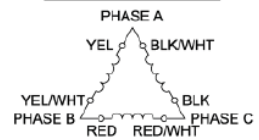


Description	Motor Wire Color	Cable Adder Color
Hall Supply	Red	Red/White
Hall A	Blue	Orange/White
Hall B	Green	Orange
Hall C	White	Yellow/White
Hall Ground	Black	Black/White
Phase A	Yellow, Yel/Wht	Yellow
Phase B	Red, Red/Wht	Red
Phase C	Black, Blk/Wht	Black

STAR CONFIGURATION



DELTA CONFIGURATION



NOTE:

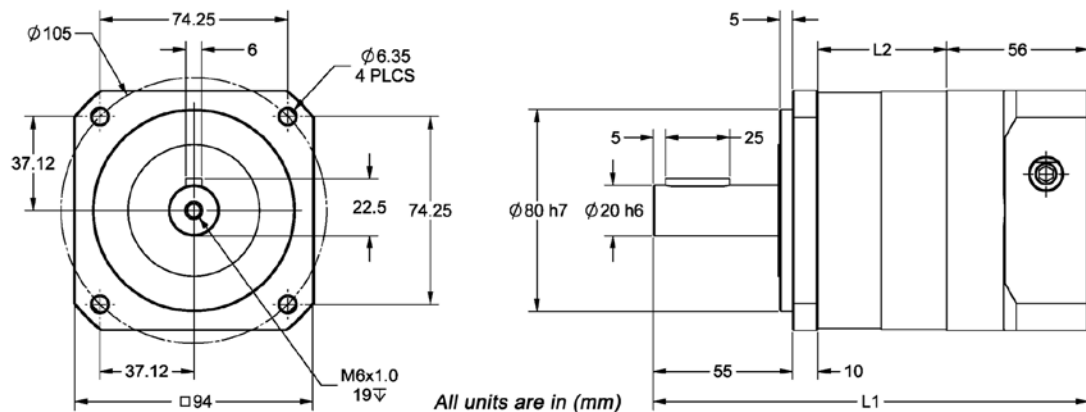
1. STAR CONFIGURATION IS DEFAULT.
2. FOR DELTA CONFIGURATION:

REMOVE PLASTIC CAP,
REMOVE 3 SCREWS HOLDING PCB,
ROTATE HALL PCB TO LINE UP WITH
RED MARKER, REPLACE SCREWS.
REPLACE PLASTIC CAP.

Hall Effect Angle:	120° Electrical Angle	Insulation Class	Class B
Shaft Run Out:	0.05mm	Dielectric Strength	500VDC for one minute
Radial Play:	0.02mm@450g	Insulation Resistance:	100MOhms Min, 500VDC
End Play:	0.08mm@450g	Number of Poles	8
Max. Radial Force:	220N@20mm from the flange	Number of Phase:	3
Max. Axial Force:	60N	Temperature Rise:	70°C
Typical Efficiency:	80%	Life Expectancy:	Over 20,000 hours

APÉNDICE D

EXTRACTOS DE LA HOJA DE DATOS DE LA REDUCTORA



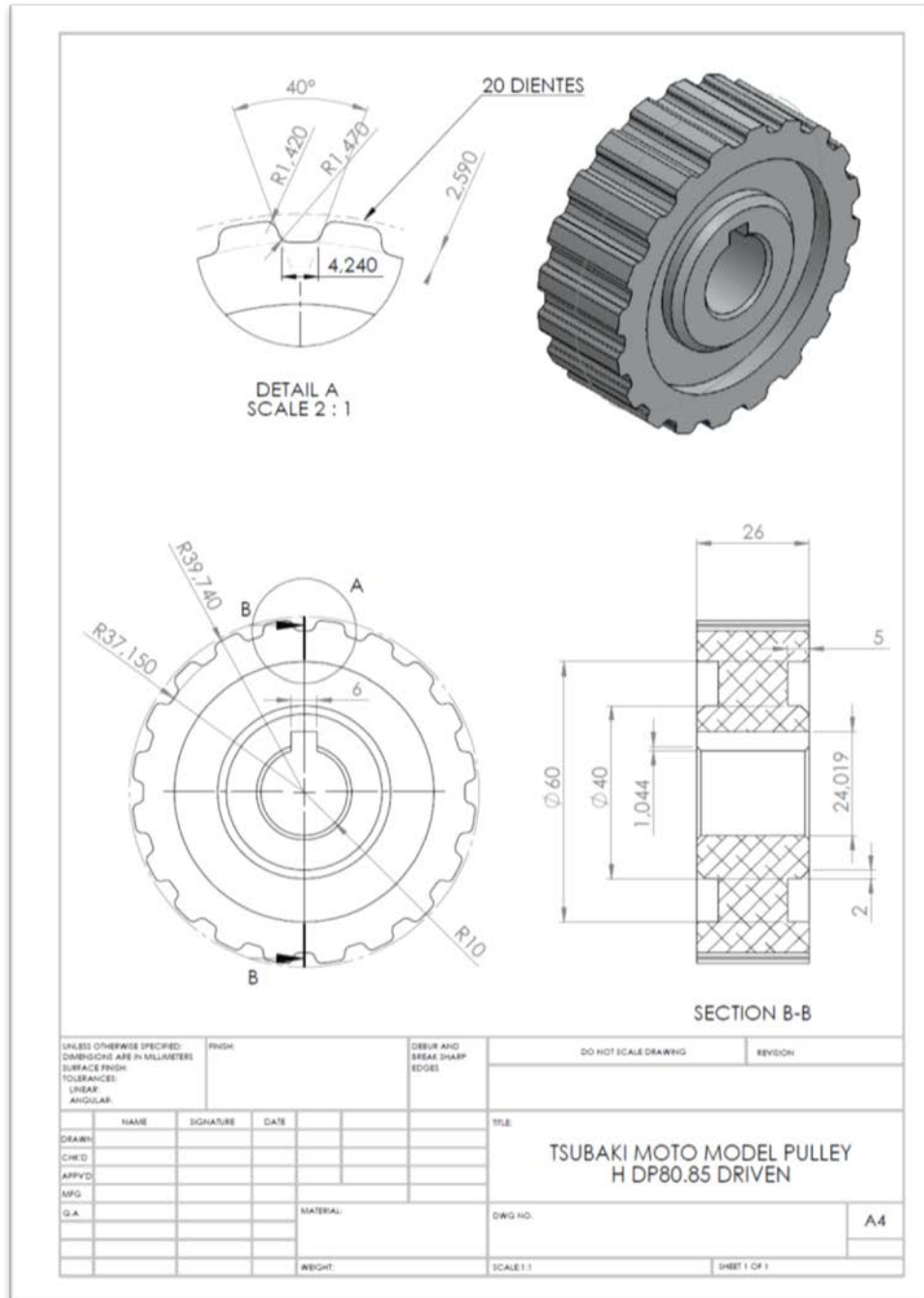
Parameter	Range
Bolt Hole Circle Diameter	70mm to 100mm
Input Shaft Diameter	14mm to 19mm
Input Shaft Length	40mm Max
Pilot Thickness	8mm Max
Pilot Diameter	50mm to 80mm
Hole Diameter	Any Size

Model #	Gear Ratio (X:1)	Rated Output Torque (oz-in)	Max Output Torque (oz-in)	Maximum Input Speed (RPM)	Stages	Backlash Maximum (arcmin)	L1 Length (mm)	L2 Length (mm)	Weight (lbs)
GBPH-0901-NS-003-xxxx-yyy	3	23791	71372	4000	1	11	172	51	7.72

Radial Load:	189 lbs - Force	Axial Load:	136 lbs - Force
Operating Temperature:	15° to 194°C	Noise Level:	68 dB
Efficiency:	95%	IP Rating:	IP65

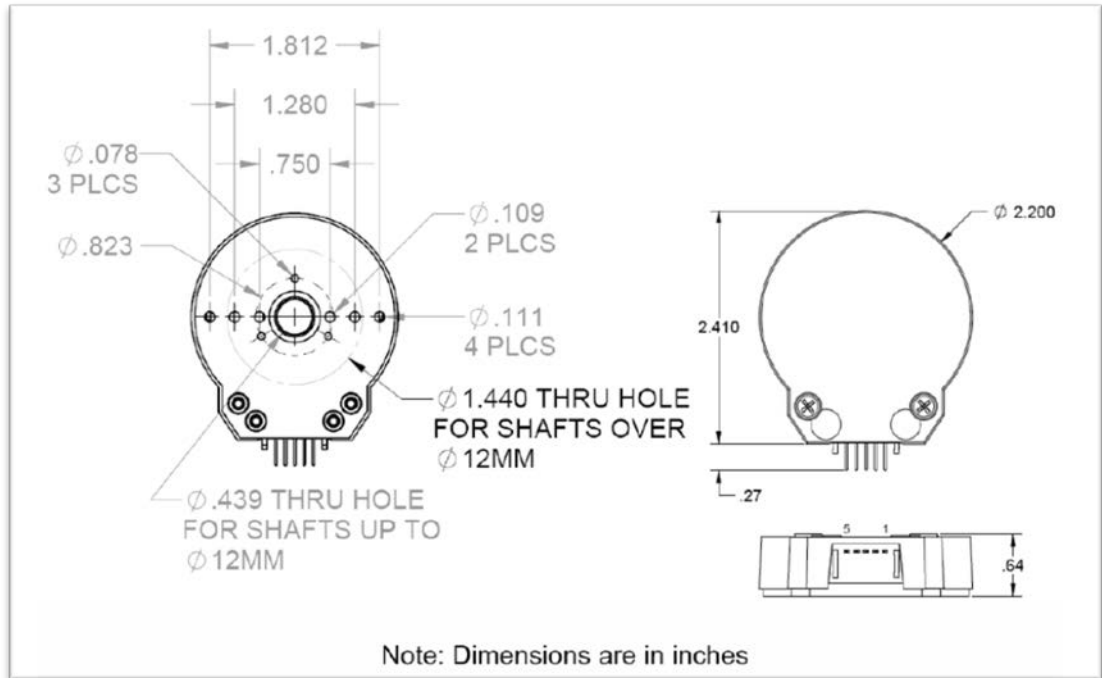
APÉNDICE E

PLANO DE LA POLEA CONDUCTORA

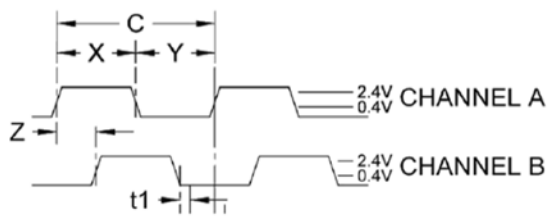


APÉNDICE F

EXTRACTOS DE LA HOJA DE DATOS DEL ENCODER



SINGLE-END ENCODER TIMING DIAGRAMS



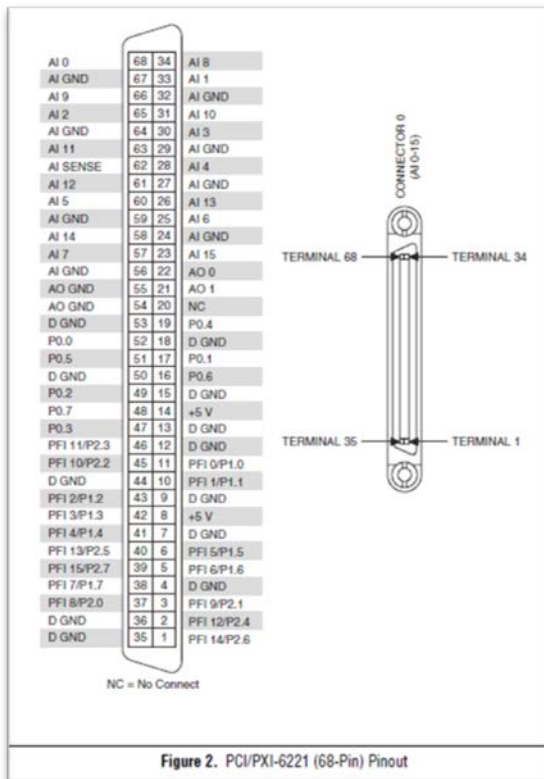
ROTATION:
CW - A LEADS B, CCW - B LEADS A

SINGLE-ENDED ENCODER PINOUT TOP OF ENCODER FACING PLUG

Pin #	Function
1	Ground
2	No Connection
3	Channel A
4	+5VDC Input
5	Channel B

APÉNDICE G

EXTRACTOS DE LA HOJA DE DATOS DE LA TARJETA DE ADQUISICIÓN DE DATOS NI PCI 6221 (68 PINES)



Default NI-DAQmx Counter Terminals		
Counter/Timer Signal	Default Pin Number	Signal Name
CTR 0 SRC	37	PFI 8
CTR 0 GATE	3	PFI 9
CTR 0 AUX	45	PFI 10
CTR 0 OUT	2	PFI 12
CTR 0 A	37	PFI 8
CTR 0 Z	3	PFI 9
CTR 0 B	45	PFI 10
CTR 1 SRC	42	PFI 3
CTR 1 GATE	41	PFI 4
CTR 1 AUX	46	PFI 11
CTR 1 OUT	40	PFI 13
CTR 1 A	42	PFI 3
CTR 1 Z	41	PFI 4
CTR 1 B	46	PFI 11
FREQ OUT	1	PFI 14

APÉNDICE H

CÓDIGO PARA ADQUISICIÓN DE DATOS EN MATLAB

```
%LECTURA DE POSICIÓN DE MÓVIL RESPECTO A SEÑAL PULSO%
%
% Se limpian datos y sesiones anteriores
clear;
clc;
close;

%Se crea una sesión "s" para NI
s = daq.createSession('ni');

%---CONFIGURACIÓN DE ADQUISICIÓN---

%Se añade la entrada de contador ctrl1 de la tarjeta PCI 6221, con
ID: dev2
%y tipo de medición: Position al canal ch1 para posición del móvil%
ch1 = addCounterInputChannel(s, 'Dev1', 0, 'Position');
%Se añade la entrada de contador ctr0 de la tarjeta PCI 6221, con
ID: dev2
%y tipo de medición: Position al canal ch2 para posición del
péndulo%
ch2 = addCounterInputChannel(s, 'Dev1', 1, 'Position');

%para trabajar con contadores por hardware se debe abrir un canal
%analógico en la tarjeta 1%
addAnalogInputChannel(s, 'Dev1', 1, 'Voltage');

%Configuración de la tasa de muestreo (en muestras/s) y duración de
la
%adquisición en segundos
s.Rate = 10000;
Duracion = 0.6;
s.DurationInSeconds = Duracion;

% Se crea el canal de comunicación para entradas y salidas digitales
en la sesión s, la
% tarjeta PCI 6221 con ID: Dev1, el puerto 0 y la línea 1, y tipo de
% señal solamente de salida
ch4 = addDigitalChannel(s, 'Dev1', 'Port0/Line0:1', 'OutputOnly');

%--- CONFIGURACIÓN DE GENERACIÓN ---

%Se crea una salida analógica en el canal "ch5" en la sesión "s",
tarjeta
%PCI 6221 con ID: Dev1, salida "ao0" y tipo de señal voltaje dc
ch5 = addAnalogOutputChannel(s, 'Dev1', 'ao0', 'Voltage');
```

```

%--- PREPARACIÓN DE DATOS A GENERAR ---

%Se establece el tiempo en función de la tasa de
%muestreo y duración de la adquisición/generación

Tiempo = 0:(1/s.Rate):Duracion;

%Se crea una función escalón 0 "y1" y otra función "y" de un pulso
de 0.6 segs

y1 = (1*(heaviside(Tiempo)));
y2 = (0*(heaviside(Tiempo)));

y = (2.4*(heaviside(Tiempo-0)-heaviside(Tiempo-0.6)));

%Se preparan los datos a generar en las salidas digital y analógica
%respectivamente

queueOutputData(s,[y1.',y2.',y.']);

% --- EJECUCIÓN DE LA ADQUISICIÓN ---

%inicia la adquisición/generación de datos y los guarda en un
arreglo de
%pulsos contados y marcas de tiempo
[PulsosContados, MarcasDeTiempo] = startForeground(s);

%para contadores de 32 bits (como el de la pci 6221) se recomienda
usar un
%umbral (threshold) de 2^31, válido para valores entre +- 2^31%
counterNBits = 32;
signedThreshold = 2^(counterNBits-1);
signedData = PulsosContados(:,1);
signedData(signedData > signedThreshold) = signedData(signedData >
signedThreshold) - 2^counterNBits;

%para contadores de 32 bits (como el de la pci 6221) se recomienda
usar un
%umbral (threshold) de 2^31, válido para valores entre +- 2^31%
counterNBits = 32;
signedThreshold = 2^(counterNBits-1);
signedDatapend1 = PulsosContados(:,2);
signedDatapend1(signedDatapend1 > signedThreshold) =
signedDatapend1(signedDatapend1 > signedThreshold) - 2^counterNBits;

%se hace la conversión de pulsos a grados y luego a avance lineal,
los ppr se ingresan a la
%variable PPR%
PPR = 2500;
PosicionGrados = signedData * 360/PPR;
PerimetroPolea = 249.69;

```



```

AvanceLineal = PerimetroPolea * PosicionGrados/360;

%se hace la conversión de pulsos a grados para la posición del
péndulo 1
PosicionGradospend1 = signedDatapend1 * 360/PPR;

% ---- PREPARACIÓN DE DATOS PARA IDENTIFICACIÓN DE SISTEMAS ----

PosicionAngular = PosicionGradospend1.';

% ---- GRAFICACIÓN ----

%Se grafica la función pulso generada%

figure
subplot (3,1,1);
plot(Tiempo,y);
hold on;
title ('Señal escalón de velocidad');
xlabel('Tiempo (s)');
ylabel('Voltaje (Vdc)');
ylim([-1 3]);
grid on
grid minor

%Cambio de posición lineal del m'vil en mm

subplot (3,1,2);
plot(MarcasDeTiempo, AvanceLineal,'g');
title ('Respuesta de la posición del móvil a la señal escalón de
velocidad');
xlabel('Tiempo (s)');
ylabel('Posición del móvil (mm)');
%ylim([-1200 1200]);
grid on
grid minor

%Cambio de posición angular del péndulo 1 en grados

subplot (3,1,3);
plot(MarcasDeTiempo, PosicionGradospend1,'r');
title ('Respuesta de la posición del péndulo 1 a la señal escalón de
velocidad');
xlabel('Tiempo (s)');
ylabel('Posición del péndulo 1 (grados)');
grid on
grid minor

hold off

```

```
% --- SEÑAL DE PARADA (V=0) ---

%Se prepara la parada a ser enviada, 1 para la señal analógica de
velocidad
%y 1 para la señal digital de run/stop
queueOutputData(s,[0.',0.',1.']);
%Se inicia el envío de la señal de parada
startForeground(s);
```

BIBLIOGRAFÍA

1. BOUBAKER OLFA, "The Inverted Pendulum: A fundamental Benchmark in Control Theory and Robotics", Tunisia, National Institute of Applied Sciences and Technology
2. <http://www.segwaymexico.com.mx/>
3. <http://exploration.grc.nasa.gov/education/rocket/rktstab.html>
4. Hunk-ok L., Takanishi A. 2006, "Biped walking robots created at Waseda University: WL and WABIAN family"
5. <http://www.anaheimautomation.com/manuals/forms/brushless-driver-and-controller-guide.php#sthash.TahXOf9g.dpbs>
6. <http://www.ni.com/tutorial/7109/es/>
7. <http://www.anaheimautomation.com/manuals/forms/brushless-dc-motor-guide.php#sthash.LYGfhlz7.dpbs>
8. Baldursson S. (2005). "BLDC Motor Modelling and Control – A Matlab®/Simulink® Implementation", Tesis de Maestría, Institutionen för Energi och Miljö.
9. Nise N. S. (2011), "Control Systems Engineering", 6ta edición, Estados Unidos, John Wiley & Sons, Inc.
10. Monasterio-Huelin F. & Gutiérrez A. (2012). "Modelo lineal de un motor de corriente continua".

11. Mazurkiewicz J. (1995). "The basics of motion control – Part 1", Power Transmission Design, Baldor Electric Co.
12. <http://cacheme.org/lesson/transformada-de-laplace-y-funcion-transferencia/#!/prettyPhoto>
13. <http://www.mathworks.com/products/matlab/>
14. <http://www.sc.ehu.es/sbweb/energias-renovables/MATLAB/intro.html>
15. <http://ira.unileon.es/es/book/export/html/92>
16. <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemAnalysis>
17. <http://www.mathworks.com/products/simulink/>
18. http://www.mathworks.com/products/simulink/features.html#building_the_model
19. http://www.uhu.es/fernando.gomez/sdynamic_archivos/transparencias/transp_tema4.pdf
20. http://isa.uniovi.es/docencia/ra_marina/UCLM_TEMA5.PDF
21. <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>
22. <https://dinamicaycontrol.files.wordpress.com/2012/03/resumen-control-cascada.pdf>
23. Aström K. J., Furuta K. (1999). "Swinging up a pendulum by energy control", Elsevier Science Ltd.
24. <http://www.mathworks.com/help/simulink/ug/creating-an-example-model-that-uses-a-matlab-function-block.html#f6-115902>

25. <http://www.mathworks.com/help/rtwin/ug/set-external-mode-code-generation-parameters.html>

26. http://www.mathworks.com/help/daq/examples/measure-angular-position-with-an-incremental-rotary-encoder.html?s_tid=gn_loc_drop