



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

---

---

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA  
OPCIÓN EN AUTOMATIZACIÓN

DESARROLLO DE UN SISTEMA MECATRÓNICO TIPO  
PÉNDULO INVERTIDO SOBRE BASE MÓVIL.

TESIS PRESENTADA PARA OBTENER EL TÍTULO DE:  
MAESTRO EN CIENCIAS DE LA ELECTRÓNICA

PRESENTA:

LIC. MARTÍN RAMOS ORTEGA

ASESORES:

DR. SERGIO VERGARA LIMON

DR. FERNANDO REYES CORTÉS

PUEBLA, PUE., NOVIEMBRE DE 2015

## Resumen.

Este trabajo presenta, el desarrollo e implementación de un sistema mecatrónico tipo péndulo invertido sobre base móvil, el cual está constituido por la parte mecánica, hardware, firmware y software de aplicación, así como el estudio cinemático y dinámico del sistema. El objetivo del sistema es el de llevar a una posición vertical el péndulo, a partir de su posición de reposo, por medio del movimiento de un carro montado sobre un riel. Este sistema es subatuado debido a que está compuesto por dos grados de libertad pero solo un actuador.

La parte mecánica está construida en su mayoría por material de aluminio debido a que es ligero, resistente, de fácil manipulación y su costo no es muy elevado.

El hardware consiste en el diseño de un puente H con transistores MOSFET (*Metal-oxide-semiconductor Field-effect transistor*) para poder manipular el motor, se desarrolló una fuente de alimentación capaz de suministrar suficiente corriente al motor, encoders incrementales, y una tarjeta de adquisición de datos PCI (*Peripheral Component Interconnect*).

El firmware está programado en lenguaje AHDL (*Altera Hardware Description Language*), en un FPGA Cyclone II de Altera, utilizando para su programación el software Quartus II, y se utiliza para la lectura de los encoders y la manipulación del motor, esto a través del protocolo PCI.

El software se desarrolló utilizando el modelo de ciclo de vida de cascada realimentado y se implementó en lenguaje gráfico LabVIEW 10.0. El software tiene una interfaz gráfica que se utiliza para: configurar los parámetros de funcionamiento, procesar los datos obtenidos por el firmware, mostrar los resultados y almacenar los datos y gráficas en archivos.

Como resultado, el sistema es capaz de manejar, hasta el momento, tres tipos diferentes de controladores para su estudio, comparación y análisis de comportamiento, los controladores son: *Tanh*, *P*, y *PID*.

Concluyendo que con este sistema se mejoró el diseño del péndulo debido a los cambios realizados como son el cambio del motor para incrementar el torque, así como el mejoramiento en la parte mecánica y electrónica de potencia, esto con respecto al diseño anterior realizado en la FCE-BUAP.

Como trabajo a futuro para el sistema es: implementar más controladores para su estudio y realizar mejoras en firmware y software para hacer más flexible los algoritmos de control, cambiar la base del péndulo para poder dejarlo fijo en un área específica del laboratorio de robótica de la FCE.

## **Agradecimientos.**

Agradezco a mi esposa Jessica Belem Silva Montes por haberme impulsado a realizar esta maestría y por todo su amor, tiempo, comprensión y apoyo incondicional.

A mis padres, Miguel Ramos Báez, en especial a mi madre Alberta Ortega Huerta quien es un ejemplo de amor, lucha, entrega y comprensión, gracias por darme todo de ti, gracias a ti he llegado hasta estas instancias.

A mis sobrinos-hermanos: Mtro. Ricardo Ramos Aguilar y Mtro. Eric Ramos Aguilar por formar parte de mi vida y ser un ejemplo a seguir, gracias por su apoyo y cariño.

A todos mis familiares y amigos que de alguna forma me han apoyado y han estado al pendiente de este trabajo de Tesis.

A mi asesor de Tesis, Dr. Sergio Vergara Limon, por darme la oportunidad de tener un lugar en la maestría, ya que desde el primer momento de conocerlo me ofreció su apoyo incondicional, así como por su asesoría durante todo el transcurso de la maestría, y especialmente durante el desarrollo de este trabajo de tesis, gracias por no perder la confianza en mí.

A mi asesor Dr. Fernando Reyes Cortés por sus asesorías brindadas.

A mis sinodales: Dra. Amparo Palomino Merino, Dra. Monserrat Morín Castillo, Dra. Ma. Aurora D. Vargas Treviño, por todas las observaciones y correcciones hechas para que este trabajo de Tesis llegara a buen término.

A mis Profesores de la Maestría en Ciencias de la Electrónica por sus conocimientos aportados a mi preparación académica a lo largo de mis estudios de posgrado.

A todos mis compañeros de generación, con quienes compartí momentos buenos y malos tanto académicos como de esparcimiento y quienes me apoyaron durante mi estancia en la maestría.

A la empresa donde laboro, ya que sin sus permisos no habría sido posible realizar los estudios de posgrado.

A todos ellos por ser parte importante de esta etapa de mi vida.

Muchas Gracias.  
Martín Ramos Ortega.

## **Dedicatoria.**

A mi esposa, Jessica Belem Silva Montes, por su amor, apoyo y comprensión a lo largo de este proyecto de vida.

A mis padres Alberta Ortega Huerta y Miguel Ramos Báez, quienes han sido parte fundamental en mi desarrollo humano y profesional.

Martín

## Índice general

Índice de figuras .....	VII
Índice de tablas.....	X
Introducción .....	XI
<b>Capítulo 1</b> .....	1
Descripción del sistema de péndulo invertido sobre base móvil. ....	1
1.1 Características de robot propuesto. ....	1
1.2 Método de control del péndulo invertido.....	4
1.3 Diseño mecánico del péndulo invertido sobre base móvil. ....	7
1.3.1 Especificaciones de diseño.....	7
1.3.2 Tornillo sinfín y corona. ....	12
1.3.3 Poleas.....	14
1.3.4 Bandas .....	15
1.3.5 Sistema transmisor de velocidad .....	16
Conclusiones. ....	16
<b>Capítulo 2</b> .....	17
Cinemática y Dinámica. ....	17
2.1 Modelo cinemático del péndulo invertido. ....	17
2.2 Modelo Dinámico del péndulo invertido. ....	22
2.3 Simulación del sistema de control del péndulo invertido. ....	27
2.3.1 Verificación experimental y resultados de los algoritmos de control. ....	28
Conclusiones .....	30
<b>Capítulo 3</b> .....	31
Diseño del Hardware .....	31
3.1 Hardware. ....	31
3.2 Fuente de alimentación de potencia. ....	32
3.2.1 Cálculo del capacitor de filtrado. ....	33
3.2.2 Fuente de alimentación de control.....	35
3.3 Diseño del Puente H. ....	35
3.3.1 Activación de MOSFET's.....	35
3.3.2 Circulación de corriente regenerativa. ....	37
3.3.3 Lógica de control de dirección de rotación del motor y activación de MOSFET's. ....	37
3.3.4 Protección contra Shoot-Through. ....	39
3.4 Caracterización del motor. ....	40
3.4.1 Torque. ....	41
3.4.2 Desarrollo de torquímetro. ....	41
3.4.3 Procedimiento de caracterización de motor.....	42
3.5 Sensor de corriente. ....	49
3.5.1 Descripción del sensor .....	49
3.6 Tarjeta PCI. ....	50
3.7 Sensores para el sistema del péndulo invertido. ....	50
3.7.1 Microswitch.....	51

3.7.2 Encoders.....	52
3.7.3 Encoder incremental.....	53
3.7.4 Encoder incremental para motor.....	54
3.7.5 Encoder incremental para péndulo.....	55
Conclusiones.....	56
<b>Capítulo 4.....</b>	<b>57</b>
Diseño del firmware.....	57
4.1 Firmware.....	57
4.2 Protocolo PCI.....	58
4.2.1 Especificaciones de PCI.....	58
4.3 Módulo para generar PWM.....	60
4.3.1 Cálculos para el contador de 500 Hz.....	63
4.4 Metodología de lectura para un encoder incremental.....	65
4.4.1 Implementación del encoder.....	69
4.5 Módulo de control de giro.....	70
4.6 Módulo de filtrado.....	71
4.7 Módulo multiplexor.....	71
4.8 Módulo de Sensores de final de carrera.....	73
Conclusiones.....	74
<b>Capítulo 5.....</b>	<b>75</b>
Diseño del Software.....	75
5.1 Software de interfaz.....	75
5.1.1 Pantalla de usuario.....	75
5.1.2 Posicionamiento en el origen del carro.....	77
5.2 Software del sistema.....	78
5.2.1 Posicionamiento en el origen.....	79
5.2.2 Posición de inicio del carro.....	82
5.2.3 Control de levantamiento y estabilización del péndulo.....	85
Conclusiones.....	88
<b>Capítulo 6.....</b>	<b>89</b>
Controlador para el sistema mecatrónico.....	89
6.1 Control de posición.....	89
6.1.1 Controladores para el péndulo invertido.....	89
6.2 Controlador de estabilización.....	94
6.2.1 Acción de control Proporcional.....	94
6.2.2 Acción de control proporcional integral derivativa PID.....	95
6.2.3 Acción de control Tangente Hiperbólica.....	98
Conclusiones.....	100
<b>Capítulo 7.....</b>	<b>101</b>
Resultados experimentales.....	101
7.1 Controlador P.....	102
7.2 Controlador PID.....	105
7.3 Controlador tanh.....	107
7.4 Evaluación del mejor controlador.....	111
7.5 Comparativa entre el sistema desarrollado y sistemas comerciales.....	112

7.6 Ventajas y desventajas en el desarrollo del sistema mecatrónico. ....	113
Conclusiones .....	114
<b>Conclusiones generales</b> .....	115
Conclusiones .....	115
Trabajo a futuro .....	116
Referencias .....	117
Apéndice A – Constancia Congreso CIINDET 2015 .....	119
Artículo publicado .....	120
Apéndice B – Diseño de sistema en SolidWorks .....	121
Apéndice C – Circuito eléctrico completo Puente H .....	125
Apéndice D – Diagrama de conexión eléctrica de Firmware .....	126

## Índice de figuras

<b>Figura 1.1:</b> Diagrama general del funcionamiento del robot péndulo .....	1
<b>Figura 1.2:</b> Modelo de un péndulo invertido .....	3
<b>Figura 1.3:</b> Diagrama a bloques del sistema .....	5
<b>Figura 1.4:</b> Sistema de control de péndulo invertido sobre base móvil .....	5
<b>Figura 1.5:</b> Zonas de control para el péndulo invertido .....	6
<b>Figura 1.6:</b> Diagrama de flujo para conmutación de controladores del sistema .....	7
<b>Figura 1.7:</b> Estructura del péndulo invertido .....	8
<b>Figura 1.8:</b> Perfil del aluminio usado en el péndulo invertido .....	8
<b>Figura 1.9:</b> Base del motor para movimiento del péndulo .....	9
<b>Figura 1.10:</b> Interior de la caja para el motor y el mecanismo de transmisión .....	10
<b>Figura 1.11:</b> Cambio de posición de encoder de Motor .....	10
<b>Figura 1.12:</b> Las dos poleas con las que cuenta el péndulo invertido .....	11
<b>Figura 1.13:</b> Vista lateral del carro completo .....	12
<b>Figura 1.14:</b> Acoplamiento del tornillo sin fin y la corona .....	13
<b>Figura 1.15:</b> Mecanismo de transmisión de tornillo sin fin y corona .....	14
<b>Figura 1.16:</b> Vista lateral del juego de poleas y banda dentada .....	15
<b>Figura 1.17:</b> Banda dentada .....	15
<b>Figura 1.18:</b> Sistema de transmisión de velocidad .....	16
<b>Figura 2.1:</b> Diagrama del sistema mecatrónico tipo péndulo invertido .....	19
<b>Figura 2.2:</b> Diagrama del péndulo invertido .....	24
<b>Figura 2.3:</b> Modelo de masa puntual del péndulo invertido .....	24
<b>Figura 2.4:</b> Levantamiento y estabilización del péndulo con $\theta = 180^\circ$ final .....	28
<b>Figura 2.5:</b> Posición del carro sobre el riel .....	29
<b>Figura 2.6:</b> Par aplicado al motor para movimiento del péndulo invertido .....	29
<b>Figura 3.1:</b> Diagrama a bloques para la implementación de la electrónica .....	32
<b>Figura 3.2:</b> Diagrama a bloques para la implementación de la electrónica .....	32
<b>Figura 3.3:</b> Circuito eléctrico de la fuente de alimentación .....	33
<b>Figura 3.4:</b> Fuente de alimentación de 24VDC construida .....	34

<b>Figura 3.5:</b> Fuente de alimentación atx de computadora.....	35
<b>Figura 3.6:</b> Circuito general de un Puente H.....	36
<b>Figura 3.7:</b> Circuito driver IR2110 en configuración bomba de carga.....	36
<b>Figura 3.8:</b> Configuración Puente H dual MOSFET.....	37
<b>Figura 3.9:</b> Lógica de control de activación de transistores MOSFET.....	38
<b>Figura 3.10</b> Modo de operación del circuito "Tiempo muerto".....	39
<b>Figura 3.11:</b> Circuito electrónico "Tiempo muerto".....	40
<b>Figura 3.12:</b> Enrutamiento para la placa PCB (izquierda) y tarjeta terminada.....	40
<b>Figura 3.13:</b> Construcción final del torquímetro.....	42
<b>Figura 3.14:</b> Montaje del motor en el torquímetro.....	43
<b>Figura 3.15:</b> Gráfica de comportamiento del motor al incrementar voltaje de alimentación.....	44
<b>Figura 3.16:</b> Diagrama a bloques para la caracterización del motor.....	46
<b>Figura 3.17:</b> Gráficas de comportamiento del motor al aplicarle PWM y variar el ciclo de trabajo.....	48
<b>Figura 3.18:</b> Pines y diagrama de conexión, sensor de corriente Honeywell.....	49
<b>Figura 3.19:</b> Tarjeta PCI FOTON_1.....	50
<b>Figura 3.20:</b> Diagrama interno de un Microswitch.....	51
<b>Figura 3.21:</b> Sensor de final de carrera atornillado al riel.....	52
<b>Figura 3.22:</b> Conexión para los sensores de final de carrera.....	52
<b>Figura 3.23:</b> Salidas de encoder incremental A y B.....	53
<b>Figura 3.24:</b> Disco codificado y generación de pulsos A y B.....	54
<b>Figura 3.25:</b> Encoder modelo E50S6-2500-6-I-5 y señales de salida A y B.....	54
<b>Figura 3.26:</b> Encoder modelo DFS60- BHC02000.....	55
<b>Figura 3.27:</b> Señales A y B obtenidas del encoder.....	56
<b>Figura 4.1:</b> Diagrama a bloques del Firmware.....	58
<b>Figura 4.2:</b> Símbolo del bloque protocolo PCI.....	59
<b>Figura 4.3:</b> Bloque de protocolo PCI en software Quartus.....	59
<b>Figura 4.4:</b> Simulación de protocolo PCI en software Quartus.....	60
<b>Figura 4.5:</b> Aplicación de la modulación por ancho de pulso.....	61
<b>Figura 4.6:</b> Diagrama de flujo para generar frecuencia de 500 Hz a partir del reloj del FPGA.....	62
<b>Figura 4.7:</b> Bloque para generar la frecuencia de 500 Hz.....	63
<b>Figura 4.8:</b> Generación de una frecuencia de 500 Hz.....	63
<b>Figura 4.9:</b> Bloque del PWM en Quartus.....	64
<b>Figura 4.10:</b> Diagrama de flujo para generar el PWM.....	65
<b>Figura 4.11:</b> Combinaciones lógicas de las señales A y B.....	66
<b>Figura 4.12:</b> Circuito de muestreo de señales.....	66
<b>Figura 4.13:</b> Diagrama de tiempos de las señales A y B.....	67
<b>Figura 4.14:</b> Retraso de las señales (A o B).....	67
<b>Figura 4.15:</b> Reloj generado por las señales A y B.....	68
<b>Figura 4.16:</b> Diagrama de Flujo de la lectura del encoder.....	70
<b>Figura 4.17:</b> Bloque de conteo de las señales del encoder.....	70
<b>Figura 4.18:</b> Bloque de control de giro.....	71

<b>Figura 4.19:</b> Módulo de filtro digital.....	71
<b>Figura 4.20:</b> Bloque del multiplexor.....	72
<b>Figura 4.21:</b> Diagrama a bloques del multiplexor.....	73
<b>Figura 4.22:</b> Bloque para los sensores de final de carrera.....	74
<b>Figura 5.1:</b> Pantalla de usuario.....	76
<b>Figura 5.2:</b> Diagrama a bloques del software del péndulo invertido.....	79
<b>Figura 5.3:</b> Diagrama de flujo del posicionamiento en el origen.....	80
<b>Figura 5.4:</b> Escritura en los bits 8 y 9 para indicar giro de motor a la derecha y a la izquierda respectivamente.....	81
<b>Figura 5.5:</b> Escritura de H'3' al PCI en los bits 10 y 11 para leer los valores de S1 y S2.....	82
<b>Figura 5.6:</b> Lectura de PCI de los valores de S1 y S2.....	82
<b>Figura 5.7:</b> Petición de lectura de encoder del motor por medio del bit 10 y recepción del valor del encoder del motor.....	83
<b>Figura 5.8:</b> Calculo de la posición y la velocidad del carro.....	84
<b>Figura 5.9:</b> Diagrama de flujo del control 1 del péndulo invertido.....	85
<b>Figura 5.10:</b> Diagrama de flujo para el control 2 del péndulo invertido.....	86
<b>Figura 5.11:</b> Control 3 del péndulo invertido.....	87
<b>Figura 6.1:</b> Funcionamiento del control para el péndulo invertido.....	90
<b>Figura 6.2:</b> Función senoidal.....	91
<b>Figura 6.3:</b> Fórmula para función senoidal implementada en LabVIEW.....	93
<b>Figura 6.4:</b> Control TANH para el controlador senoidal.....	93
<b>Figura 6.5:</b> Bloque de control P y acción de control proporcional.....	95
<b>Figura 6.6:</b> Bloque para controlador PID.....	97
<b>Figura 6.7:</b> Acción de control proporcional integral derivativo.....	98
<b>Figura 6.8:</b> Gráfica de comportamiento de la función TANH.....	99
<b>Figura 6.9:</b> Región de trabajo del controlador TANH.....	99
<b>Figura 6.10:</b> Controlador TANH en LabVIEW.....	100
<b>Figura 7.1:</b> Posición del péndulo, controlador tipo P.....	103
<b>Figura 7.2:</b> Recorrido de las varillas del péndulo invertido.....	104
<b>Figura 7.3:</b> Tau resultante del control P.....	104
<b>Figura 7.4:</b> Posición del péndulo, controlador PID.....	106
<b>Figura 7.5:</b> Tau resultante del control PID.....	107
<b>Figura 7.6:</b> Posición del péndulo.....	108
<b>Figura 7.7:</b> Posición del péndulo, simulación.....	109
<b>Figura 7.8:</b> Tau resultante del controlador tanh.....	109
<b>Figura 7.9:</b> Tau, simulación.....	110
<b>Figura 7.10:</b> Resultados de una prueba con controlador tanh.....	110
<b>Figura A.1:</b> Constancia Congreso CIINDET 2015.....	119
<b>Figura A.2:</b> Artículo publicado.....	120
<b>Figura B.1:</b> Perfil del aluminio usado en estructura del péndulo invertido.....	121

<b>Figura B.2:</b> Soportes del péndulo invertido.....	121
<b>Figura B.3:</b> Vista lateral e inferior de las poleas del péndulo invertido.....	121
<b>Figura B.4:</b> Riel del péndulo invertido.....	122
<b>Figura B.5:</b> Vista lateral y superior del carro.....	122
<b>Figura B.6:</b> Perfil de las llantas de los carritos .....	123
<b>Figura B.7:</b> Medidas de las varillas del péndulo.....	123
<b>Figura B.8:</b> Poleas para el péndulo invertido.....	123
<b>Figura B.9:</b> Diseño de torquímetro en SolidWorks.....	124
<b>Figura C.1:</b> Diagrama de conexión eléctrico de Puente H.....	125
<b>Figura D.1:</b> Diagrama de conexión eléctrico del Firmware parte 1.....	126
<b>Figura D.2:</b> Diagrama de conexión eléctrico del Firmware parte 2.....	127

## Índice de tablas

<b>Tabla 2.4:</b> Parámetros del péndulo invertido.....	27
<b>Tabla 3.1:</b> Características de la fuente de alimentación.....	33
<b>Tabla 3.2:</b> Señales de control de dirección y activación de transistores MOSFET .....	38
<b>Tabla 3.3:</b> Torque obtenido del motor al incrementar voltaje de alimentación...	43
<b>Tabla 3.4:</b> Datos obtenidos al aplicar diferentes frecuencias.....	46
<b>Tabla 3.5:</b> Valores obtenidos variando el ciclo de trabajo.....	47
<b>Tabla 3.6:</b> Características tarjeta PCI FOTON_1.....	50
<b>Tabla 4.1:</b> Tabla de verdad para los códigos de los datos de los encoders.....	72
<b>Tabla 4.2:</b> Salidas de los sensores de final de carrera.....	73
<b>Tabla 5.1:</b> Datos de entrada para el FPGA.....	80
<b>Tabla 5.2:</b> Descripción de datos para el bloque PWM y selección de datos a leer.....	81
<b>Tabla 6.1:</b> Efecto de las constantes sobre el control PID.....	96
<b>Tabla 7.1:</b> Comparación entre sistema desarrollado y sistemas comerciales....	112

## Introducción

Los péndulos invertidos son una familia de artefactos que constituyen un banco de pruebas muy completo para la ingeniería de control lineal. El más estudiado de los miembros de esta familia es el denominado control invertido sobre un vehículo, al que corrientemente se denomina como carro. Consiste en un péndulo o varilla que gira libremente por uno de sus extremos mediante una articulación situada sobre un vehículo o sistema móvil, que se mueve horizontalmente bajo la acción de una fuerza  $F$ . Esta fuerza es la acción de control con la que se pretende actuar sobre la posición de la varilla [1].

El modelo del péndulo invertido es un ejemplo clásico en la literatura de control y tiene múltiples aplicaciones, desde el control de misiles hasta el análisis de la biomecánica de la marcha, balance y postura humana. El péndulo invertido es un sistema inestable, ya que puede caer en cualquier momento, a menos de que se aplique una fuerza de control adecuada [2].

El interés del problema local del péndulo invertido reside en que se trata de estabilizar una posición inestable en lazo abierto. En sistemas lineales, la estabilidad en lazo cerrado de un punto inestable en lazo abierto, no ofrece particulares problemas, estos aparecen cuando el sistema es no lineal.

El principio del péndulo fue descubierto por el físico y astrónomo italiano Galileo, quien estableció que el periodo de la oscilación de un péndulo de una longitud dada puede considerarse independiente de su amplitud (la amplitud es la distancia máxima que se puede alejar el péndulo de la posición de equilibrio vertical). Este descubrimiento lo hizo observando las oscilaciones de una lámpara colgante en la Catedral de Pisa.

En 1851 el científico francés León Foucault realizó un experimento en el panteón de París que puso de manifiesto la rotación terrestre, esto usando el principio del péndulo normal.

Posteriormente Katsuhisa Furuta construyó su péndulo conocido como el *Péndulo de Furuta*. Consiste en un motor de eje vertical al que es solidario un brazo de cuyo extremo cuelga la varilla del péndulo. Es decir, el extremo del brazo juega el mismo papel que el de un carro en otros péndulos [3].

El sistema mecatrónico tipo péndulo invertido es un sistema formado por un motor cuyo eje de giro se encuentra acoplado a un sistema mecánico que permite el movimiento de una estructura con un péndulo que puede oscilar libremente, en ausencia de excitación del motor el péndulo pende hacia abajo, el problema consiste en actuar sobre el motor de modo que se haga que el péndulo invierta su posición quedando levantado y se mantenga en esa nueva posición, esto quiere

decir que tiene que oponer a su estado natural así como a factores externos como fuerzas que detengan el movimiento de oscilación llamadas amortiguadoras como la del rozamiento.

El péndulo invertido es un sistema electromecánico bastante simple sin embargo el comportamiento que queremos obtener de él es relativamente complejo, en realidad se presentan dos problemas, el de balanceo hasta conseguir que el péndulo se acerque hasta la posición vertical invertida conocido normalmente como swing-up y el de mantener el péndulo en la posición invertida vertical una vez alcanzada. Para el problema del balanceo se tienen que aplicar impulsos de modo que se va aumentando progresivamente la amplitud de las oscilaciones hasta alcanzar la horizontal o incluso sobrepasarla, es decir se quiere inyectar impulsos de energía en los momentos adecuados para ir aumentando la amplitud de las oscilaciones hasta sobrepasar la línea horizontal, una vez que se logra esto la estrategia de control se modifica, en este caso no se trata de aumentar las oscilaciones sino de mantener el péndulo erguido este problema se conoce como el de la estabilización local entorno al equilibrio deseado.

Para conseguir el objetivo del péndulo invertido se requiere medir mediante un sensor, que usualmente es un encoder, el ángulo que forma el péndulo con la vertical y a partir de esta información decidir qué actuación debe tener el motor, además se quiere tener la posición del carro para lo que se usa un segundo encoder, el péndulo invertido es entonces una estructura retroalimentada que mediante la información proporcionada por los sensores procesa un algoritmo en una computadora para decidir qué acción es la conveniente a llevar a cabo, en este caso para que el motor ejerza sobre el péndulo la fuerza necesaria para llevarlo a la posición horizontal vertical. El interés en este proceso radica en que estamos trabajando con un sistema subactuado en el cual el motor actúa sobre la base de la que pende el péndulo y no sobre el péndulo cuya posición es la que se quiere controlar, esto da lugar a interesantes problemas de estabilidad que han hecho que el péndulo invertido sea objeto de estudio durante los últimos años. El péndulo invertido está compuesto por una serie de elementos como lo son una estructura mecánica, un sistema sensorial, un sistema de adquisición de datos y una computadora, además de esto tiene que contar con un software que gestione el área sensorial y el movimiento de la estructura mecánica, para conservar el péndulo perpendicular incluso ante la presencia de perturbaciones.

Actualmente podemos encontrar el sistema del péndulo invertido en el mercado gracias a varias empresas que se dedican a hacer kits didácticos, algunas de ellas como Gunt HAMBURG incluso los desarrollan en cooperación con universidades, con los prototipos desarrollados por esta empresa se pueden probar diversos métodos de control como el difuso [4].

La empresa FeedBack tiene entre sus equipos educativos modelos en los que se pueden realizar actividades como control digital en tiempo real, modelación e identificación lineal, modelación no lineal y representación en el espacio de estado, control óptimo cuadrático y control difuso, el equipo incluye manuales, software e interfaz para el usuario [5].

Con esto podemos ver la importancia del desarrollo de este tipo de proyectos hoy en día, ya que ser capaces de producir equipo didáctico evidentemente resulta menos costoso que adquirirlo, además los péndulos invertidos son muy populares debido a que son útiles para el análisis de control en el ámbito educativo, prueba de esto es que el primer péndulo invertido fue diseñado en los años setenta y aún continúa usándose como objeto de estudio principalmente porque presenta dos problemas de control interesantes, el levantamiento del péndulo y el cómo mantenerlo en esa posición.

En resumen el modelo del péndulo invertido es un ejemplo clásico de control con múltiples aplicaciones, principalmente en el control automático, con este se pueden comprender los principios básicos del control y de la robótica y además el modelo del péndulo invertido puede usarse como base para otros sistemas, gracias a esta versatilidad es muy usado tanto en prácticas como en proyectos de investigación, el sistema es bien conocido por su posibilidad de analogía para el diseño de diversos controladores y aplicaciones experimentales.

## **Objetivo general**

*“Desarrollar un sistema mecatrónico tipo péndulo invertido sobre base móvil”.*

## **Objetivos específicos**

- Adecuar un sistema mecatrónico del robot tipo péndulo invertido sobre base móvil.
- Estudiar el análisis de cinemática y dinámica del sistema mecatrónico.
- Instrumentar el sistema de control del robot mecatrónico.
- Desarrollar la interfaz de usuario para configurar el controlador deseado.
- Pruebas del sistema diseñado.
- Publicación de los resultados.

En los últimos años el uso de sistemas mecatrónicos se ha generalizado y ha dado origen a avances tecnológicos importantes para la humanidad de tal manera que actualmente es estratégico para un país contar con recursos humanos especializados en el área. El sistema que se está proponiendo permitirá a los estudiantes de la facultad probar experimentalmente diferentes propuestas de controladores tanto en simulación como en un sistema real, lo cual pretende contribuir al mejoramiento del aprendizaje de estas técnicas.

Lo que se pretende con este trabajo es que los alumnos puedan probar sus algoritmos de control de forma experimental y dinámica y de esta manera complementar sus conocimientos teóricos. Ya que el péndulo invertido tiene similitud con otros sistemas, podemos mudar fácilmente un control previamente probado en el péndulo invertido a un sistema más complejo por ejemplo en el estudio de los robots con extremidades donde lo más importante es el balanceo de estas, el péndulo invertido puede ayudar dentro del estudio y el desarrollo de modelos que capturen una respuesta muy parecida a la humana. En el posicionamiento de un satélite con respecto a la tierra, en este caso el satélite está en movimiento y las antenas que se encuentran en la tierra no pueden dejar que el satélite se mueva demasiado con respecto a ellas ya que si no se saldría del rango de comunicación.

Dado que el control automático se ha convertido en parte importante de los procesos industriales, tener un sistema que permita la evaluación y comparación de diferentes algoritmos de control es una herramienta que nos da una gran ventaja con respecto a las simulaciones comunes con software, es por esto que se desarrolló este proyecto, para que se pudiera contar con un sistema real con el cual los estudiantes pudieran poner en práctica sus conocimientos.

A continuación se describe la estructura del presente documento de tesis.

El Capítulo 1 se habla de un poco de historia y de las aplicaciones más importantes de este sistema mecatrónico, como el diseño con el que se contó, así como las modificaciones que se hicieron para mejorar el desempeño, se habla de la estructura móvil y fija del sistema.

El Capítulo 2 presentan los fundamentos teóricos básicos necesarios para comprender el funcionamiento del péndulo invertido en base a su modelo cinemático y dinámico.

El Capítulo 3 muestra el desarrollo de la electrónica de potencia para controlar el motor y la electrónica necesaria para el funcionamiento del sistema.

El Capítulo 4 describe el diseño y desarrollo de cada uno de los módulos que conforman el firmware, este se desarrolla con el software Quartus II.

El Capítulo 5 describe el desarrollo del software, la metodología utilizada y la implementación de los módulos individuales que lo componen, esto se realiza con el software LabVIEW.

El Capítulo 6 describe el software de control, tanto del levantamiento como el de estabilización del péndulo, al igual se desarrolla en LabVIEW.

El Capítulo 7 presenta los procedimientos que se emplean para realizar las pruebas y los resultados experimentales que se obtienen. Se muestra la comparación de las pruebas realizadas en el péndulo invertido con los distintos controladores, llevando a cabo un análisis de los resultados y una comparación entre los controladores, evaluando el desempeño de cada uno de ellos.

Por último se presentan las conclusiones, referencias y anexos; en el anexo A, el artículo publicado y la constancia obtenida de la presentación del artículo, en el apéndice B se muestra el diseño mecánico del sistema en el software SolidWorks.

# Capítulo 1.

## Descripción del sistema de péndulo invertido sobre base móvil.

El péndulo digital es una versión moderna de un problema de control clásico, el carro sobre el riel es controlado digitalmente para balancear el péndulo y llegar a su posición vertical y mantener su posición. El riel del carro es una longitud limitada, lo cual impone restricciones en el algoritmo de control.

En este capítulo se explica la metodología para controlar en sistema así como la estructura mecánica del péndulo invertido, tanto en la forma y diseño así como en los materiales. Se describirá cada una de las partes que conforma al sistema y los cambios que se realizaron a lo largo del proyecto. Para este trabajo se contó con la estructura del péndulo invertido sobre base móvil, la cual se puso a punto con algunas modificaciones para mejorar su desempeño.

### 1.1 Características de robot propuesto.

Prototipo propuesto se toma como base el mostrado en la Figura 1.1.

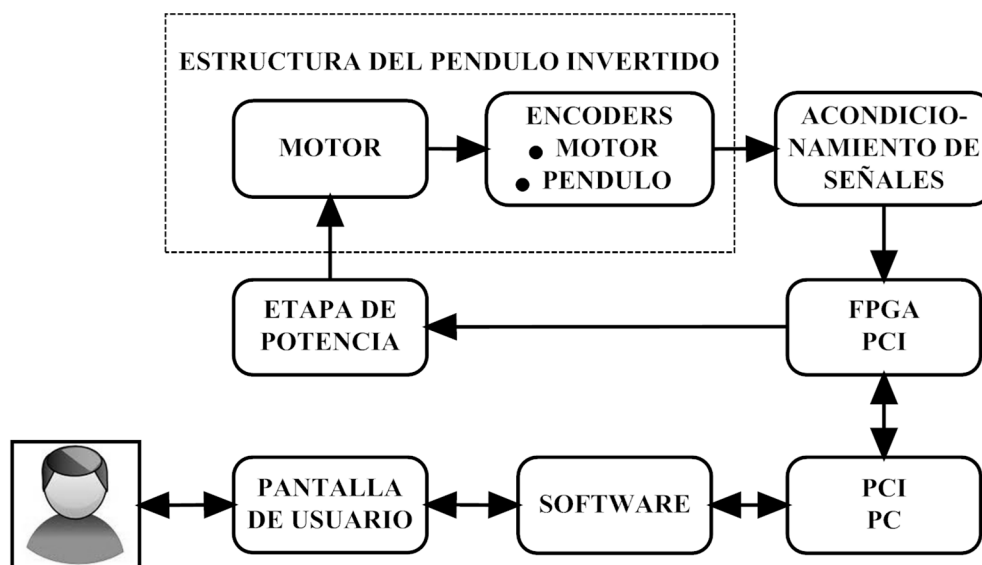


Figura 1.1: Diagrama general del funcionamiento del robot péndulo.

En la Figura 1.1 se puede observar el diagrama de funcionamiento de nuestro robot, el cual cuenta con 4 etapas para el correcto funcionamiento, las cuales son la interfaz gráfica, la etapa de adquisición de datos, etapa de potencia, y el acondicionamiento de señales, cada una de estas realiza tareas específicas para la realización de los trabajos que tiene que realizar el sistema.

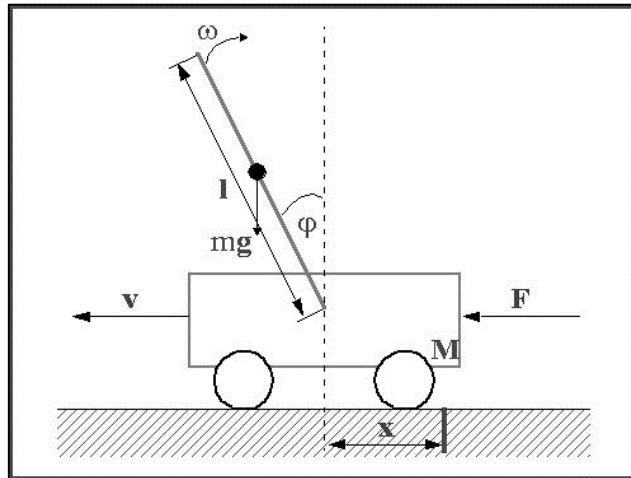
**Etapa de interfaz gráfica:** En esta parte del sistema se realiza el interactuado del usuario con la máquina, y en base a los datos proporcionados por la persona el sistema responde, el ambiente grafico esta realizado en el software LABVIEW, el programa creado en está diseñado para que sea de fácil uso e intuitivo para la persona que trabajará con el robot, una vez que el usuario manda los datos para el movimiento del robot estos son capturados, en la etapa de adquisición de datos.

**Etapa de adquisición de datos:** Aquí los datos enviados por el usuario se reciben y se procesan por medio de un FPGA de altera, dentro de este se desarrolló previamente un firmware para poder realizar la comunicación de la computadora a nuestro sistema, que según sean los datos enviados este responderá enviando las señales a la etapa de potencia del sistema.

**Etapa de Potencia:** En esta parte del sistema las señales enviadas por el FPGA los recibe una tarjeta de etapa de potencia la cual consta de un puente H esto para poder controlar el giro del motor, tomando las decisiones por en base al control de sistema que está cargado en el software de este, una vez que el actuador está en movimiento los encoders que están acoplados al motor y al péndulo, envían señales digitales las cuales pasan por la etapa de acondicionamiento.

**Acondicionamiento de señal:** Por ultimo las señales generadas por el encoders incrementales vienen con ruido por eso es necesario meterlos a una etapa de acondicionamiento de señales para poder limpiar las impurezas de estas señales y retroalimentarlas al FPGA, esto con el propósito de que dichas señales no presenten problemas de ruido.

El péndulo invertido por su parte se compone de una barra sujeta a un carro, el carro se encuentra sobre un riel de forma que se puede mover horizontalmente de un lado a otro, Figura 1.2.



**Figura 1.2:** Modelo de un péndulo invertido

El sistema mecatrónico del péndulo invertido sobre base móvil cumple las siguientes condiciones:

- La barra es articulada y puede girar libremente alrededor de la articulación en el carro.
- El carro se encuentra sujeto al riel mediante una banda.
- Un sistema de actuación se encarga de proporcionar la energía al carro a través de la banda.
- Cuenta con un conjunto de sensores para determinar la posición del carro y de la barra.
- El objetivo es el de llevar el brazo a la posición de equilibrio superior mediante la aplicación de una fuerza de control al carro.

Debido a que el movimiento del brazo no puede ser controlado directamente el péndulo invertido es un sistema subactuado.

Podemos definir como sistema subactuado como aquel sistema que carece de al menos un actuador en uno de sus grados de libertad, es decir, un sistema subactuado será aquel que posee un número de grados de libertad mayor que de actuadores [6].

La definición de sistema subactuado incluye no sólo el fallo en los actuadores, sino también la ausencia de los mismos provocada por consideraciones de diseño del sistema físico, falta de espacio, exceso de peso, por cuestiones de presupuesto etc.

Con la finalidad de entender el comportamiento del sistema y de poder controlarlo es necesario obtener modelos matemáticos que los representen. Los modelos representan de forma matemática al sistema mecatrónico, y así se puede simular su comportamiento con la ayuda de software especializado, además de que nos ayuda en la elaboración de un algoritmo de control apropiado. Para la realización de estos modelos se hace uso de los siguientes conceptos básicos.

La *posición*, se utiliza para describir el movimiento de un cuerpo. Hay dos tipos de posición, la lineal es el lugar en el espacio que ocupa un punto característico del cuerpo y la angular es la orientación en el espacio de ese cuerpo. A un cambio de posición lineal se le llama traslación y a un cambio de posición angular una rotación.

La *velocidad*, es el flujo de posición, es decir el proceso que genera movimiento, como tal siempre existe en un intervalo de tiempo, cuando este intervalo es pequeño hablamos de velocidad instantánea y cuando es finito de velocidad promedio. Que sea una función del tiempo quiere decir que puede asumir valores que pueden ser diferentes en cada intervalo de tiempo.

La *aceleración*, es la tasa de cambio de velocidad, matemáticamente es la primera derivada de la velocidad con respecto al tiempo, la aceleración nos sirve como indicador del modo en que cambia la velocidad.

Finalmente el *tiempo*, es la variable independiente en función de la cual se describe el movimiento, es un atributo externo al sistema.

## **1.2 Método de control del péndulo invertido.**

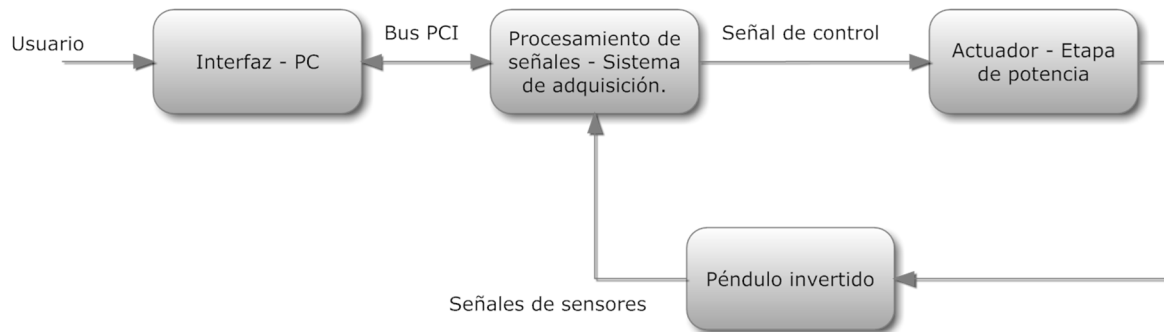
El controlador podría ser continuo, usando amplificadores operacionales por mencionar algún ejemplo; o discreto, usando un micro controlador, un procesador digital de señales (DSP) o una computadora personal (PC). El diseño de un controlador continuo es más sencillo que el de uno discreto, pero un controlador discreto es mucho más flexible. Es por esto que es necesario contar con un péndulo invertido que incluya una tarjeta de adquisición de datos diseñada para incorporarse a una PC, un módulo que contenga los acondicionadores de señal para los sensores y el actuador para el motor. El uso de una PC para implementar el controlador que es la manera más rápida.

La flexibilidad con que cuentan las PC ofrece también la opción más didáctica.

El sistema del péndulo invertido sobre base móvil utiliza un sistema de retroalimentación, en el cual, la señal de control es proporcionada por un algoritmo implementado vía software, que parte de la computadora mediante

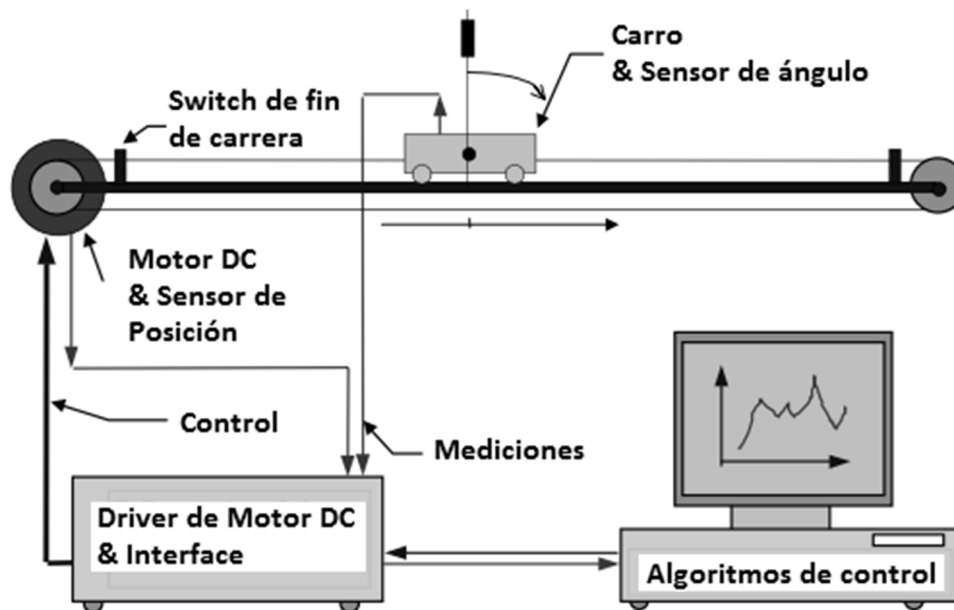
una tarjeta de adquisición de datos, la cual es capaz de enviar y recibir señales analógicas y digitales.

El control se realiza en la PC, donde se podrá configurar el control a usar mediante la interfaz de usuario, las señales se envían y reciben a través de una tarjeta de adquisición de datos con un FPGA que se comunica a la PC mediante el bus PCI. Figura 1.3



**Figura 1.3:** Diagrama a bloques del sistema

En seguida se muestra en la Figura 1.4 el sistema de control de péndulo invertido sobre base móvil.

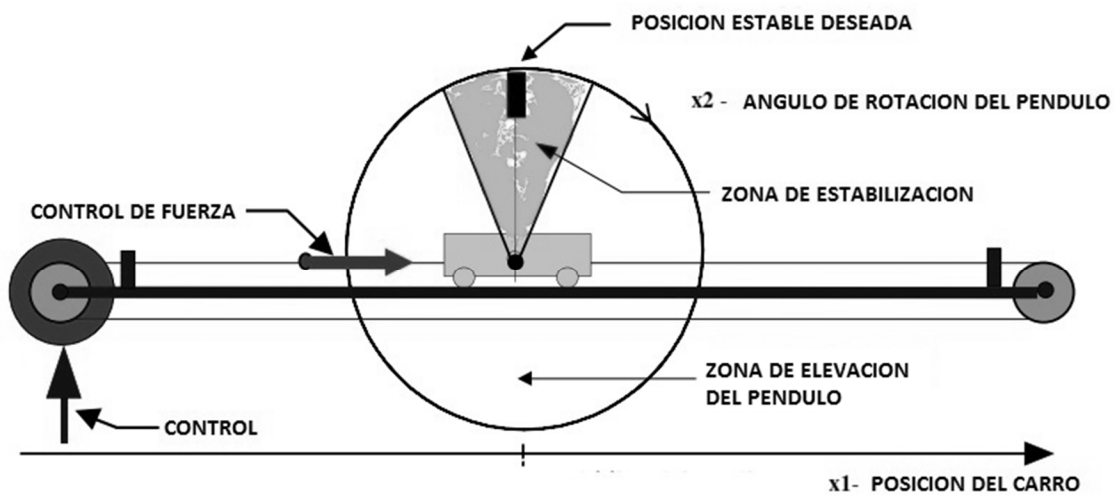


**Figura 1.4:** Sistema de control de péndulo invertido sobre base móvil.

Para el propósito de funcionamiento del péndulo es necesario usar dos algoritmos de control independientes que se aplicarán para:

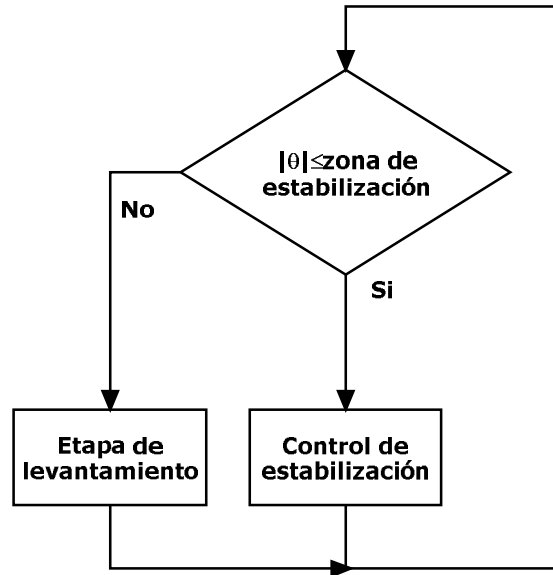
- Un algoritmo de control de elevación (swing up).
- Un algoritmo de control de estabilización.

Solo uno de los dos algoritmos de control es activado en cada zona de control como se muestra en la Figura 1.5, de tal manera que los controladores conmutaran de acuerdo a la posición del péndulo con respecto a la vertical, siempre y cuando el ángulo  $|\theta| \leq \text{rango de zona de estabilización}$ .



**Figura 1.5:** Zonas de control para el péndulo invertido.

Como el sistema requiere dos sistemas de control para el levantamiento y para la estabilización se tiene el siguiente diagrama de flujo para la conmutación de controladores Figura 1.6.



**Figura 1.6:** Diagrama de flujo para conmutación de controladores del sistema.

### 1.3 Diseño mecánico del péndulo invertido sobre base móvil.

#### 1.3.1 Especificaciones de diseño.

Para el desarrollo de este trabajo se cuenta con la estructura del péndulo invertido sobre base móvil, en la cual se realizaron modificaciones para mejorar su desempeño.

Dentro de las modificaciones que se realizaron se encuentra la del cambio del motor con un torque más grande, el cual mediante un mecanismo de transmisión, proporciona la fuerza de control al carro, el carro está unido al mecanismo de transmisión mediante un par de poleas y una banda.

El carro tiene dos barras de aluminio, una a cada lado con el propósito de mantener el equilibrio, que pueden girar libremente alrededor del eje que las une al carro. Al final de cada barra se encuentra un peso de bronce para darles más masa a las barras.

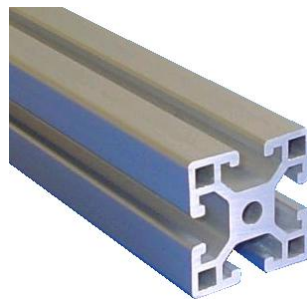
El carro se mueve sobre un riel que se encuentra sujeto en sus extremos a dos patas, como se aprecia en la Figura 1.7.



**Figura 1.7:** Estructura del péndulo invertido

A continuación se describen cada uno de los elementos que forman parte del péndulo invertido, comenzando por la estructura y después los elementos que se encuentran en ella.

El péndulo está hecho en perfil de aluminio #21223 de 40x40 mm 6051 TC, este se puede observar en la Figura 1.8 y Figura B.1 en el apéndice B.



**Figura 1.8:** Perfil de aluminio usado en el péndulo invertido.

El péndulo consta en general de dos partes:

- La parte fija, que se encarga de darle soporte al sistema.
- Y la parte móvil, la cual consta del mecanismo del carro con el péndulo.

La parte fija consta del riel con sus dos soportes en los cuales se fijan las poleas para la banda del carro, así como de una caja de aluminio en la que se monta el motor, el encoder y el mecanismo de transmisión.

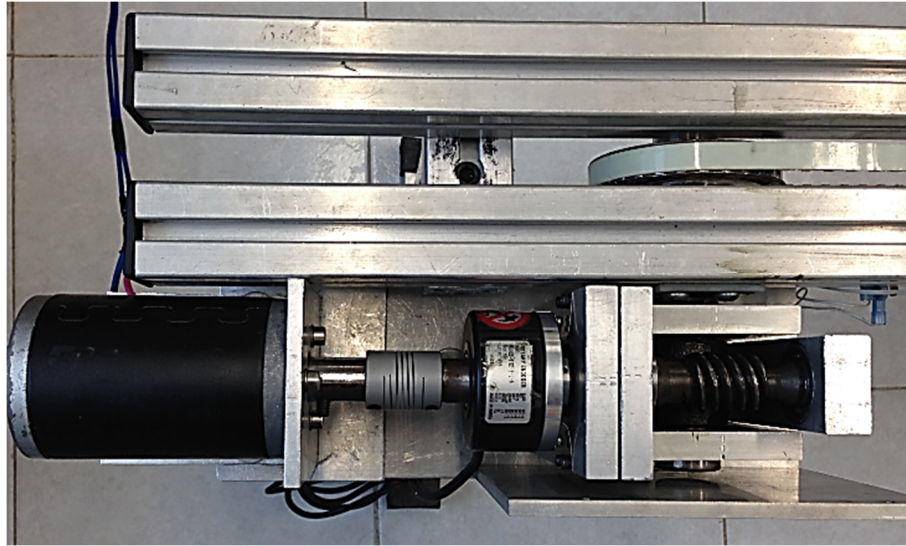
En la Figura B.2 (apéndice B), se muestran los soportes con sus respectivas medidas, las medidas se encuentran en milímetros y al igual que en la Figura B.3 (apéndice B), se usa el software SolidWorks.



**Figura 1.9:** Base del motor para movimiento del péndulo.

En la Figura 1.9 se observa el soporte para el motor y su mecanismo de transmisión, Figura 1.10, la caja está conformada por una placa de aluminio y un conjunto de paredes de aluminio y de Nylamid, que se encargan de cargar al motor, al encoder y al mecanismo de tornillo sin fin y corona, estas placas están hechas a la medida de cada una de estos elementos según su diámetro, esta caja se encuentra cerrada por seguridad, debido a la alta velocidad a la que gira el motor y para no tener expuesto el tornillo sin fin.

Una de las mejoras que se hizo al péndulo invertido fue la de alinear todo este mecanismo, desmontándolo y montándolo de nuevo parte por parte cuidando que el eje del motor no se encontrara desalineado. En la Figura 1.10 se muestra esta caja.



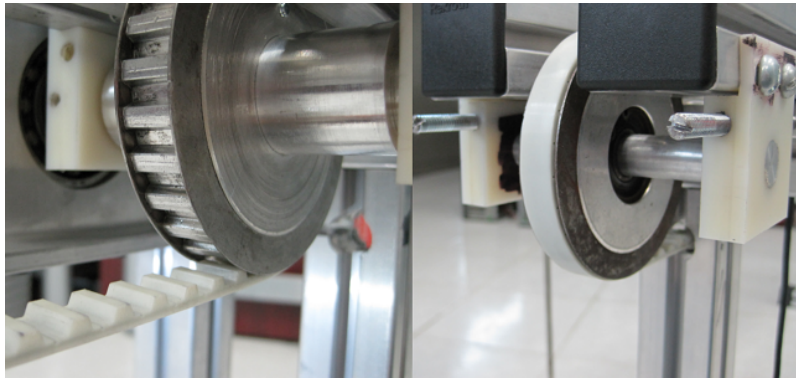
**Figura 1.10:** Interior de la caja para el motor y el mecanismo de transmisión.

También se realizó el cambio de posición del encoder del motor debido a que se dañó debido a la alta velocidad que presentó el motor al momento de hacer las pruebas, para esto se fabricó una base para colocarlo al eje de la corona, como se muestra en la Figura 1.11.



**Figura 1.11:** Cambio de posición de encoder de Motor.

En la Figura B.3 (apéndice B) y Figura 1.12, se muestra cómo es que se fijan las poleas a las cuales se sujeta la banda del carro.



**Figura 1.12:** Las dos poleas con las que cuenta el péndulo invertido.

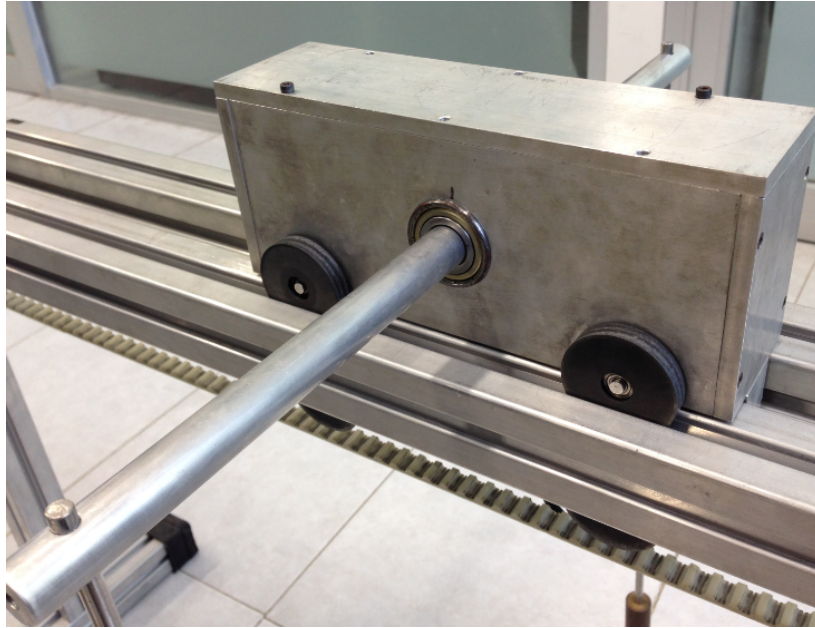
El último elemento de la parte fija es el riel, sobre este se desplaza el carro y se aprovecha el perfil de aluminio para crearlo con dos barras.

Aun cuando el riel mide más de un metro de largo, no es esta toda la longitud que tiene el carro para recorrer, como se ve en la Figura B.4 (apéndice B), la longitud del riel al lado de la caja no puede ser recorrida por el carro pues las varillas chocarían con la caja además de que el péndulo cuenta con un par de sensores de final de carrera colocados por debajo del riel, con lo cual la longitud para recorrer por el carro disminuye hasta unos 64 cm.

La parte móvil consta del carro y las varillas que forman el péndulo, el carro está formado por una caja hueca con cuatro llantas de Nylamid SL.

Otra de las mejoras que se hizo al péndulo invertido es la ajustar al carro al riel por ambos lados tanto superior como inferior, pero después de algunas pruebas se observó que el carro tenía demasiado juego y las varillas perdían potencia. Por lo que se acoplaron unas lanas para fijar el carrito inferior a la banda y al carrito superior, esto debido a que presentaba juego entre estos componentes. En la Figura B.5 (apéndice B) se muestra a detalle el diseño del carro.

En la Figura B.6 (apéndice B) se observa el diseño de las llantas las cuales están fabricadas en Nylamid debido a que es un material resistente pero además esta lubricado de manera que resbala con facilidad, esto para que las llantas no causaran fricción al desplazarse sobre el riel. En la Figura 1.13 se observan las ruedas montadas en el carro.



**Figura 1.13:** Vista lateral del carro completo.

Las dos varillas (Figura B.7 apéndice B) son de aluminio y tienen en su extremo inferior un peso de bronce, estas se pueden ajustar a distintas alturas, se cambió su altura original de 35 cm a una de 42 cm, pues el péndulo respondía mejor mientras mayor fuera la distancia entre el origen y el peso.

### **1.3.2 Tornillo sinfín y corona.**

Un mecanismo transforma un movimiento y una fuerza de entrada en un movimiento y una fuerza de salida. El problema básico de las máquinas es el de reducir la alta velocidad de los motores a una velocidad utilizable por los equipos, además de reducir se deben contemplar las posiciones de los ejes de entrada y salida y la potencia mecánica a transmitir.

En nuestro caso contamos con un motor de 3600 rpm, esta velocidad es más de la que necesitamos en nuestro péndulo invertido, además de que requerimos de una potencia mayor para poder desplazar el carro.

Para potencias mayores se usan equipos reductores, el reductor básico está formado por un mecanismo de tornillo sinfín y corona, el cual es el que se emplea en nuestro sistema.

Al emplear reductores de velocidad se obtienen una serie de beneficios sobre otras formas de reducción, tales como:

- Una regularidad perfecta tanto en la velocidad como en la potencia transmitida.
- Mayor eficiencia en la transmisión de la potencia suministrada por el motor.
- Mayor seguridad en la transmisión, reduciendo los costos de mantenimiento.
- Menor espacio requerido y mayor rigidez en el montaje.

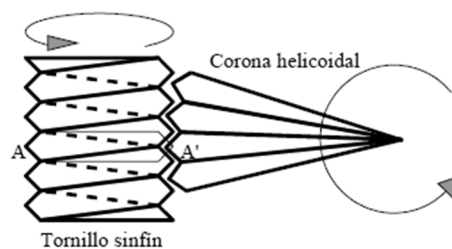
Es el mecanismo que usaremos para transmitir el movimiento, se usa para transmitir, además de movimiento, potencia entre ejes que no se cruzan, generalmente forman 90° entre sí.

La velocidad con la que gira la rueda siempre es menor que con la que gira el tornillo, su relación de transmisión es reducida, pero la de potencia es alta.

Se usa en mecanismos que necesiten una gran reducción de velocidad y un aumento importante en la ganancia mecánica.

Este mecanismo además transmite el movimiento circular y de forma silenciosa.

El tornillo sinfín (Figura 1.14) que actúa como eje motor acoplado con una rueda dentada que actúa como eje conducido, los dientes de la rueda helicoidal están ligeramente inclinados para que se adapten a la rosca del tornillo, cada vuelta que del tornillo hace que la rueda gire el ángulo equivalente a un diente, así se consiguen reducciones de velocidad en un espacio reducido.



**Figura 1.14:** Acoplamiento del tornillo sin fin y la corona.

Como en toda transmisión de ruedas dentadas se cumple:

$$N1 \times D1 = N2 \times D2 \quad (1.1)$$

$N1$  es la velocidad del eje motriz

$N2$  es la velocidad del eje conducido

$D1$  es el número de dientes de la rueda conductora

$D2$  es el número de dientes de la rueda conducida

En este caso  $N1=1$ , pues el sinfín solamente tiene un diente, pero enrollado helicoidalmente, por lo que la velocidad del eje conducido será:

$$N2 = D1/D2 \quad (1.2)$$

Es decir la velocidad del eje conducido es la del eje conductor dividido por el número de dientes del piñón.

Por lo tanto mientras mayor sea el número de dientes del piñón menor será la velocidad que obtendremos en el eje conducido.

En resumen este tipo de transmisión:

- Ofrece una gran reducción de velocidad y un aumento importante de la ganancia mecánica.
- Esa gran ganancia mecánica y la posición relativa de los dientes hace que el mecanismo sea irreversible, no es posible hacerlo funcionar si conectamos el piñón al árbol motriz y el sinfín al conducido.
- El espacio que ocupa es mínimo en relación a otras opciones como multiplicador de velocidad, tren de engranajes, etc.

En nuestro caso contamos con una corona de 31 dientes (Figura 1.15), por lo que este será el factor de reducción de nuestro mecanismo.



**Figura 1.15:** Mecanismo de transmisión de tornillo sin fin y corona

### 1.3.3 Poleas

Las poleas son ruedas que tienen el perímetro exterior diseñado especialmente para facilitar el contacto con bandas.

Las poleas dentadas como las que se usan en este sistema, no dependen de la fricción para transmitir potencia, permitiendo una tensión mínima en las bandas, además al no haber deslizamiento entre la polea y la banda, hay sincronización entre los ejes.

Se cuenta con dos poleas (Figura B.8 apéndice B), una está unida a la salida de la corona de nuestro mecanismo de transmisión, a esta se le sujeta una banda dentada que se apoya en el otro extremo a una segunda polea de igual diámetro que la primera (Figura 1.16).



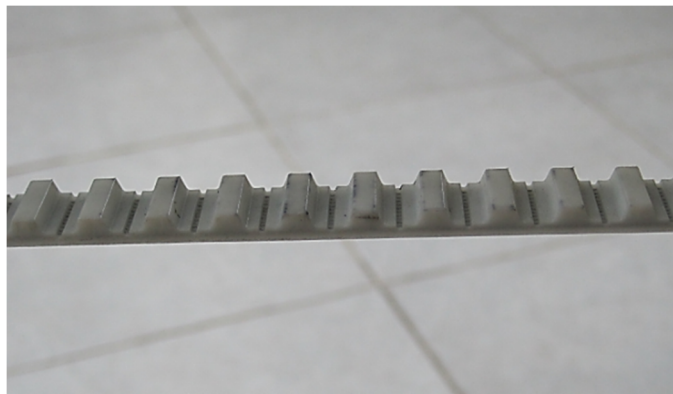
**Figura 1.16:** Vista lateral del juego de poleas y banda dentada

### 1.3.4 Bandas

El sistema de poleas y banda es una forma simple, barata y efectiva de transmitir movimiento entre dos ejes. Este sistema no requieren lubricación y la elasticidad de la banda amortigua cargas y vibraciones.

Los ejes son paralelos y giran en el mismo sentido. Se emplea una banda y poleas dentadas para evitar un deslizamiento no deseado del carro.

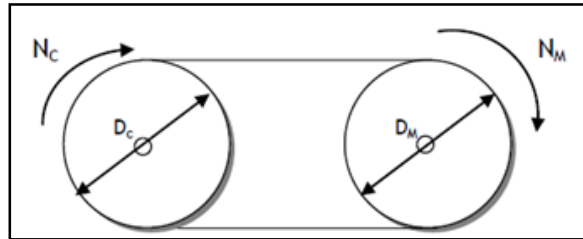
La banda usada es de caucho y cuenta con una sección trapezoidal, esta tiene 12 mm de ancho, la banda se observa en la Figura 1.17.



**Figura 1.17:** Banda dentada

### 1.3.5 Sistema transmisor de velocidad

Este se caracteriza por que el diámetro de la polea motriz es igual al de la conducida (Figura 1.18), por lo tanto la velocidad es la misma. Se usa simplemente para transmitir el movimiento y es el que se tiene implementado en el péndulo invertido



**Figura 1.18:** Sistema de transmisión de velocidad

$$D_M N_M = D_C N_C \quad (1.3)$$

$D_M$  es el diámetro de la polea motriz.

$D_C$  es el diámetro de la polea conducida.

$N_M$  es la velocidad de la polea motriz.

$N_C$  es la velocidad de la polea conducida.

$D$  se expresa en unidades de longitud, metros, y  $N$  se expresa en rpm, es decir el número de vueltas que gira la polea en un minuto.

### Conclusiones.

En el capítulo se presentó la revisión de la estructura mecánica con la que se contó al inicio de esta tesis y de las modificaciones que se le realizaron para poder mejorar el funcionamiento del péndulo invertido, con la revisión y modificación de la estructura se pudo conocer a profundidad el funcionamiento del sistema mecánico, lo cual es indispensable para realizar el estudio e implementación del control del mismo. De esta manera se obtiene una mejor eficiencia mecánica de dicho sistema.

## Capítulo 2.

### Cinemática y Dinámica.

En este capítulo se presenta el modelado cinemático y dinámico del péndulo invertido sobre base móvil para comprender su funcionamiento y esto para saber cómo resolver el problema de su levantamiento y estabilidad, puesto que se requiere de dos tipos de controladores aplicados en diferentes partes del funcionamiento del sistema.

#### 2.1 Modelo cinemático del péndulo invertido.

El problema básico que debe de resolverse, es el de obtener un modelo geométrico de la estructura, el cual nos permita el estudio del movimiento del sistema, esto se conoce como problema cinemático directo y para sistemas con pocos grados de libertad tiene una solución sencilla. El modelo cinemático estudia el movimiento con respecto a un sistema de referencia, se interesa por la descripción analítica del movimiento como una función del tiempo y por las relaciones entre la posición y orientación del extremo final y los valores que toman sus coordenadas articulares.

Entre las aplicaciones que tiene la cinemática está el uso como modelo matemático, que sirva de punto de partida para el diseño del controlador del sistema y la simulación del comportamiento cinemático. Con el modelo cinemático se puede:

- Conocer las órdenes de mando para los accionadores de cada articulación del sistema para que el extremo alcance una determinada posición.
- Saber dónde está el extremo a partir de la lectura de los sensores de posición de cada articulación.
- Relacionar las velocidades del extremo y los actuadores.

En el péndulo invertido sobre base móvil, podemos identificar claramente dos tipos de movimientos. El primer movimiento identificado es el ejercido por el carro sobre el eje x, este mismo es efectuado por la fuerza ejercida por el actuador sobre la banda. El segundo movimiento es el del péndulo el cual es producido por

los movimientos lineales del carro, esto genera que el péndulo comience a oscilar con un movimiento rotacional.

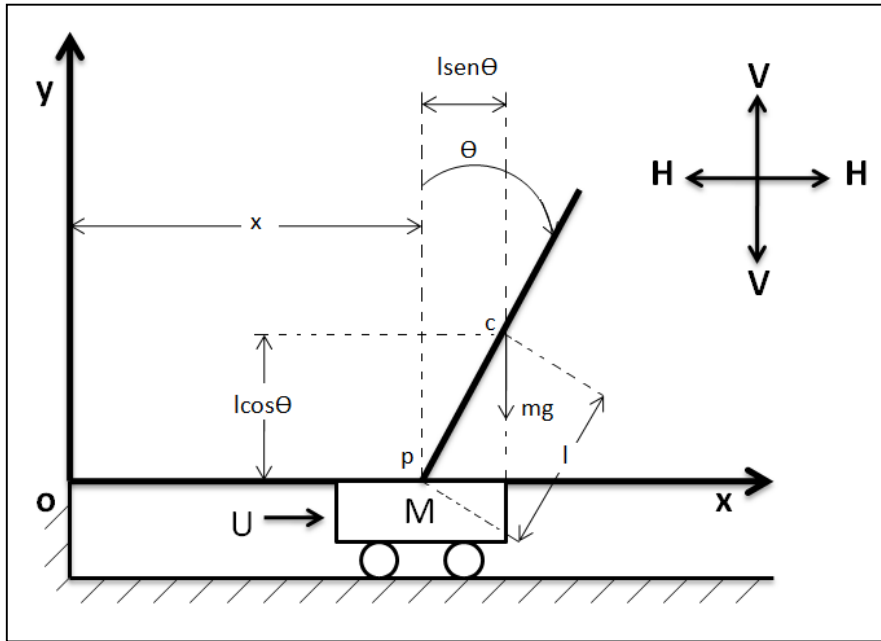
Los procedimientos para obtener la cinemática de un sistema pueden ser:

- Basados en relaciones geométricas, esto mediante trigonometría, no es sistemático y solo es válido en sistemas que tienen pocos grados de libertad.
- Basados en el uso de matrices de transformación homogénea, es sistemático y válido para robots de muchos grados de libertad.
- Cuando se tiene un sistema de pocos grados de libertad generalmente se usan procedimientos basados en relaciones geométricas, trigonometría.

Existen tantos grados de libertad como articulaciones en el sistema, para nuestro caso el péndulo invertido cuenta con una sola articulación, la barra solo puede moverse sobre un eje, por lo tanto se considera un sistema con un solo grado de libertad y su modelo se obtiene mediante relaciones geométricas. Para nuestro sistema se consideran las siguientes limitaciones:

- El carro se mueve sobre una superficie plana.
- No existen elementos flexibles en la estructura.
- Las ruedas no poseen eje de direccionamiento, de manera que siempre se encuentran de forma perpendicular al suelo.
- No se consideran ningún tipo de fricciones en elementos móviles del vehículo, o contra el suelo.

Para comenzar a determinar el modelo cinemático de nuestro péndulo invertido nos ayudamos de la siguiente Figura 2.1.



**Figura 2.1:** Diagrama del sistema mecatrónico tipo péndulo invertido.

- Donde  $M$  representa al carro y su masa, el cual se puede desplazar sobre el eje  $X$ .
- $U$  es la fuerza de control que se aplica al carro.
- $m$  es la masa de la barra.
- El punto  $c$  es el centro de masa de la barra y  $l$  la distancia desde el inicio de la barra al centro de masa de la misma.
- $x$  es la distancia desde el origen hasta la barra.
- $g$  es la fuerza de gravedad.
- $\theta$  es el ángulo de apertura de la barra con respecto al eje  $y$  positivo
- Finalmente  $V$  y  $H$  son el movimiento vertical y horizontal del centro de gravedad de la masa.

A partir de la Figura 2.1 se obtienen las siguientes relaciones.

Las coordenadas para el punto  $c$  son

$$x_c = x + l \sin \theta \quad (2.1)$$

$$y_c = l \cos \theta \quad (2.2)$$

El movimiento rotacional de la barra alrededor de su centro de gravedad se describe como

$$I \ddot{\theta} = V l \sin \theta - H l \cos \theta \quad (2.3)$$

En donde  $I$ , es el momento de inercia de la barra alrededor de su centro de gravedad.

La inercia es la propiedad de la sección transversal de un cuerpo de resistir a cualquier cambio de movimiento. El momento de inercia es similar a la inercia pero aplicable a la rotación en lugar del movimiento lineal.

El movimiento horizontal del centro de gravedad de la barra es

$$H = m \frac{d^2}{dt^2} (x + l \sin \theta) \quad (2.4)$$

El movimiento vertical del centro de gravedad de la barra es

$$V - mg = m \frac{d^2}{dt^2} (l \cos \theta) \quad (2.5)$$

El movimiento horizontal del carro es

$$U - H = M \frac{d^2 x}{dt^2} \quad (2.6)$$

Si  $\theta$  se supone muy pequeño, las ecuaciones se linealizan en torno al punto de equilibrio y quedan de la siguiente manera

$$I \ddot{\theta} = Vl \theta - Hl \quad (2.7)$$

$$H = m(\ddot{x} + l \ddot{\theta}) \quad (2.8)$$

$$V - mg = 0 \quad (2.9)$$

$$U - H = M \ddot{x} \quad (2.10)$$

De las ecuaciones 2.8 y 2.10 tenemos que

$$U - M \ddot{x} = H \quad (2.11)$$

$$U - M \ddot{x} = m(\ddot{x} + l \ddot{\theta}) \quad (2.12)$$

$$U = m(\ddot{x} + l \ddot{\theta}) + M \ddot{x} \quad (2.13)$$

$$U = m \ddot{x} + ml \ddot{\theta} + M \ddot{x} \quad (2.14)$$

$$U = (m + M) \ddot{x} + ml \ddot{\theta} \quad (2.15)$$

Y de las ecuaciones 2.7 y 2.9:

$$I \ddot{\theta} = mgl \theta - Hl \quad (2.16)$$

$$I \ddot{\theta} = mgl \theta - ml(\ddot{x} + l \ddot{\theta}) \quad (2.17)$$

$$I \ddot{\theta} = mgl \theta - ml \ddot{x} + ml^2 \ddot{\theta} \quad (2.18)$$

$$I\ddot{\theta} + ml\ddot{x} - ml^2\ddot{\theta} = mgl\theta \quad (2.19)$$

$$\ddot{\theta}(I - ml^2) + ml\ddot{x} = mgl\theta \quad (2.20)$$

Las ecuaciones 2.15 y 2.20 describen el movimiento del sistema del péndulo invertido en el carro y constituyen un modelo matemático del sistema, entonces

$$(M + m)\ddot{x} + ml\ddot{\theta} = U \quad (2.21)$$

$$(I + ml^2)\ddot{\theta} + ml\ddot{x} = mgl\theta \quad (2.22)$$

$I$ , el momento de inercia de la barra con respecto a su centro de gravedad se supone cero, así tenemos las siguientes ecuaciones, recordando que siempre y cuando  $\theta$  sea pequeña

$$(M + m)\ddot{x} + ml\ddot{\theta} = U \quad (2.23)$$

$$ml^2\ddot{\theta} + ml\ddot{x} = mgl\theta \quad (2.24)$$

Eliminando  $\ddot{x}$  de la ecuación 2.23 tenemos

$$(M + m)\left(\frac{mgl\theta - ml\ddot{\theta}}{ml}\right) + ml\ddot{\theta} = U \quad (2.25)$$

$$(M + m)(g\theta - l\ddot{\theta}) + ml\ddot{\theta} = U \quad (2.26)$$

$$Mg\theta + mg\theta - Ml\ddot{\theta} - ml\ddot{\theta} + ml\ddot{\theta} = U \quad (2.27)$$

$$-Ml\ddot{\theta} = U - Mg\theta - mg\theta \quad (2.28)$$

$$Ml\ddot{\theta} = Mg\theta - mg\theta - U \quad (2.29)$$

$$Ml\ddot{\theta} = (M + m)g\theta - U \quad (2.30)$$

Eliminando  $\ddot{\theta}$

$$(M + m)\ddot{x} + ml\left(\frac{mgl\theta - ml\ddot{x}}{ml^2}\right) = U \quad (2.31)$$

$$(M + m)\ddot{x} + m(g\theta - \ddot{x}) = U \quad (2.32)$$

$$M\ddot{x} + m\ddot{x} + mg\theta - m\ddot{x} = U \quad (2.33)$$

$$M\ddot{x} = U - mg\theta \quad (2.34)$$

Entonces finalmente obtenemos las ecuaciones que describen el movimiento de nuestro sistema de las ecuaciones 2.30 y 2.34 [7]:

$$Ml\ddot{\theta} = (M + m)g\theta - U$$

$$M\ddot{x} = U - mg\theta$$

Con estas ecuaciones se puede obtener por ejemplo una función de transferencia en el dominio  $s$ , se aplica la transformada de Laplace tomando condiciones iniciales nulas a las ecuaciones y nos queda entonces:

$$\frac{\theta(s)}{u(s)} = \frac{1}{-Mls^2 + (M+m)g} \quad (2.35)$$

## 2.2 Modelo Dinámico del péndulo invertido.

Desde el punto de vista de la robótica el sistema del péndulo invertido se visualiza como partes mecánicas en movimiento, el planteamiento cinemático no es válido cuando se pretende manipular objetos en movimiento, con esto surge la necesidad de representar la dinámica de dichas partes en forma matemática, es necesario entonces plantear modelos dinámicos donde intervenga el tiempo. Esto nos permite describir las trayectorias de los movimientos del sistema y la posición final de las partes.

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina como resultado de las mismas. El modelo dinámico es de extrema importancia para la simulación de movimientos, el análisis mecánico de la estructura, la elaboración de algoritmos de control y la programación de movimientos. Con él se puede determinar la resistencia mecánica de los componentes, torques y fuerzas que deben ser producidos por el accionamiento para efectuar un movimiento deseado.

Existen varios métodos para modelar un sistema, dentro de los más conocidos se encuentran los métodos de Newton y de Euler-Lagrange. El método de Newton se basa en la segunda ley de Newton y el método de Euler-Lagrange está basado en consideraciones energéticas, usa las ecuaciones de energía del sistema.

En sistemas complejos el método de Newton puede ser difícil de implementar debido a la interacción de varios subsistemas dentro del sistema y a la cantidad de variables que pueden tener. El método de Euler-Lagrange es útil para este tipo de sistemas ya que basta con determinar la energía del sistema y aplicar derivadas.

Aun cuando nuestro sistema no es complejo se va a obtener el modelo dinámico del sistema mediante el método de Euler-Lagrange. Ya que el procedimiento de este, se basa en el balance de fuerzas que actúan sobre un cuerpo, se tiene que obtener la energía cinética y la energía potencial.

La energía cinética del sistema es la que surge en el fenómeno de movimiento de un cuerpo, es el trabajo necesario para acelerar un cuerpo con una masa dada,

desde su posición de equilibrio hasta una cierta velocidad, una vez conseguida esta energía el cuerpo la mantiene sin importar cambios en su rapidez. La energía potencial es la energía que actúa sobre un cuerpo rígido y causa su movimiento.

Sabemos también que el modelo dinámico de un robot de  $n$  grados de libertad, en ausencia de perturbaciones externas, está definido por la siguiente estructura [8]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (2.36)$$

Donde

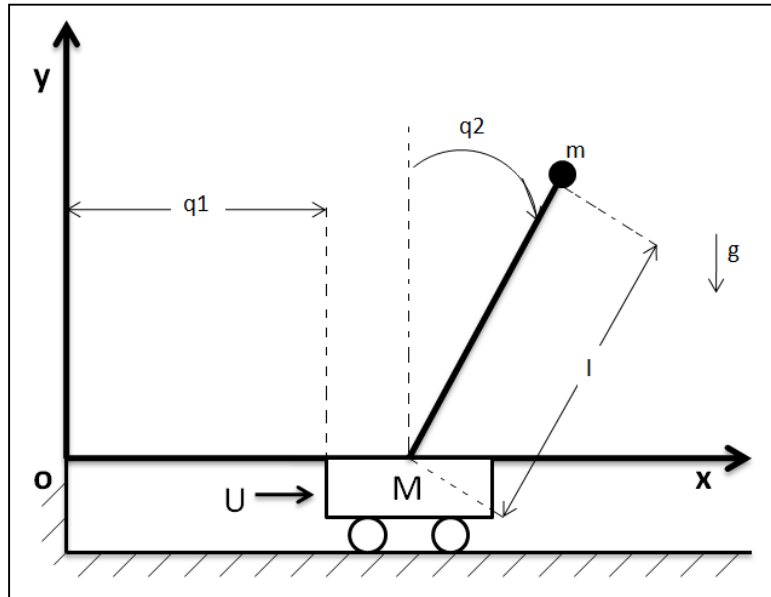
- $\ddot{q}$  es la aceleración articular.
- $\dot{q}$  la velocidad articular.
- $q$  el vector de posición del robot.
- $M(q)$  es la matriz de inercia.
- $C(q, \dot{q})$  es la matriz de Coriolis
- $g(q)$  es el par gravitacional.
- Y  $\tau$  es la fuerza aplicada.

Para llevar a cabo el procedimiento de Euler-Lagrange, se siguen los siguientes pasos:

- Cálculo de la energía cinética:  $K(q(t), \dot{q}(t))$ .
- Cálculo de la energía potencial:  $U(q(t))$ .
- Cálculo del Lagrangiano:  $L(q(t), \dot{q}(t))$ .

Desarrollo de las ecuaciones de Lagrange.

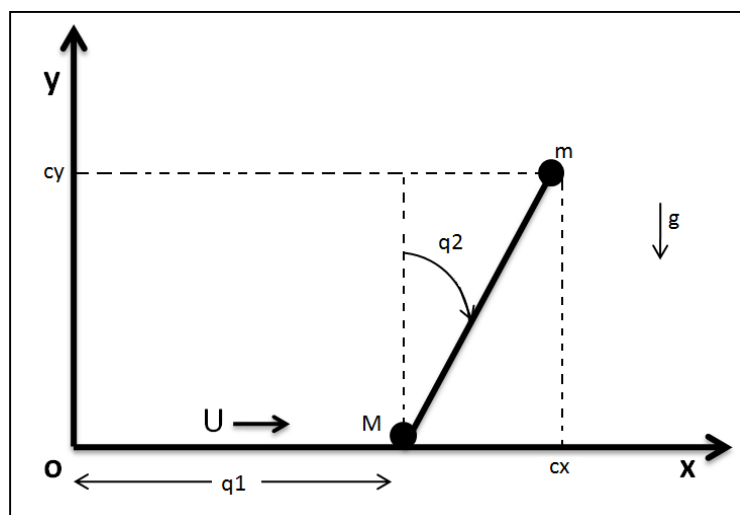
La Figura 2.2 muestra el diagrama a utilizar.



**Figura 2.2:** Diagrama del péndulo invertido.

Donde tenemos que

- $M$  es la masa del carro
- $m$  es la masa del péndulo
- $l$  la longitud de la barra
- $q1$  la posición articular 1
- $q2$  la posición articular 2
- $U$  la fuerza aplicada al carro



**Figura 2.3:** Modelo de masa puntual del péndulo invertido

Primero se calcula la energía cinética, la cual depende de la masa y de la velocidad

$$E_c = \frac{1}{2}mv^2 \quad (2.37)$$

$$K_M = \frac{1}{2}M\dot{q}_1^2 \quad (2.38)$$

$$K_m = \frac{1}{2}m[Vx^2 + Vy^2] \quad (2.39)$$

Luego tenemos que la energía potencial está en términos de la gravedad, se divide en la energía potencial para el carro y para el péndulo

$$Ep = mgh \quad (2.40)$$

$$U = U_M + U_m \quad (2.41)$$

$$U_M = 0 \quad (2.42)$$

$$U_m = mgl\cos(q_2) \quad (2.43)$$

Para el carro, tenemos que

$$\dot{q}_1 = \frac{d}{dt}q_1 \quad (2.44)$$

Esta última es la velocidad del carro. Y para el péndulo tenemos

$$[Vx^2 + Vy^2] = [Cx^2 + Cy^2] \quad (2.45)$$

Donde

$$Cx = \dot{q}_1 + l\sin(q_2)\dot{q}_2 \quad (2.46)$$

$$Cy = l\cos(q_2)\dot{q}_2 \quad (2.47)$$

$$\dot{C}x = \dot{q}_1 + l\cos(q_2)\dot{q}_2^2 \quad (2.48)$$

$$\dot{C}y = -l\sin(q_2)\dot{q}_2^2 \quad (2.49)$$

Entonces podemos sustituir para la energía cinética

$$K_m = \frac{1}{2}m \left[ (\dot{q}_1 + l\sin(q_2)\dot{q}_2)^2 + (-l\sin(q_2)\dot{q}_2)^2 \right] \quad (2.50)$$

$$K_m = \frac{1}{2}m \left[ \dot{q}_1^2 + 2\dot{q}_1\dot{q}_2l\cos(q_2) + l^2\dot{q}_2^2\cos^2(q_2) + l^2\dot{q}_2^2\sin^2(q_2) \right] \quad (2.51)$$

$$K_m = \frac{1}{2}m \left[ \dot{q}_1^2 + 2l\cos(q_2)\dot{q}_1\dot{q}_2 + l^2\dot{q}_2^2(\cos^2(q_2) + \sin^2(q_2)) \right] \quad (2.52)$$

Como  $\sin^2(q_2) + \cos^2(q_2) = 1$

$$K = K_M + K_m = \frac{1}{2}M\dot{q}_1^2 + \frac{1}{2}m[l^2\dot{q}_2^2 + 2l\cos(q_2)\dot{q}_1\dot{q}_2 + \dot{q}_1^2] \quad (2.52)$$

Luego calculamos el Lagrangeano, este es expresado como la diferencia entre la energía cinética y la energía potencial, matemáticamente se describe como

$$L(q(t), \dot{q}(t)) = K(q(t), \dot{q}(t)) - U(q(t)) \quad (2.53)$$

$$L = \frac{1}{2}M\dot{q}_1^2 + \frac{1}{2}m[l^2\dot{q}_2^2 + 2l\cos(q_2)\dot{q}_1\dot{q}_2 + \dot{q}_1^2] - mgl\cos(q_2) \quad (2.54)$$

Después derivamos  $L$ , así tenemos la ecuación de movimiento de Euler-Lagrange

$$\tau = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} \quad (2.55)$$

$$\frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} \frac{\partial L}{\partial \dot{q}_1} \\ \frac{\partial L}{\partial \dot{q}_2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}M2\dot{q}_1 + \frac{1}{2}m[0 + 2l\cos(q_2)\dot{q}_2 + 2\dot{q}_1] - 0 \\ 0 + \frac{1}{2}m[l^22\dot{q}_2 + 2l\cos(q_2)\dot{q}_1 + 0 - 0] \end{bmatrix} \quad (2.56)$$

$$= \begin{bmatrix} M\dot{q}_1 + \frac{1}{2}m[2l\cos(q_2)\dot{q}_2 + 2\dot{q}_1] \\ \frac{1}{2}m[l^22\dot{q}_2 + 2l\cos(q_2)\dot{q}_1] \end{bmatrix} \quad (2.57)$$

$$= \begin{bmatrix} M\dot{q}_1 + ml\cos(q_2)\dot{q}_2 + m\dot{q}_1 \\ ml^2\dot{q}_2 + ml\cos(q_2)\dot{q}_1 \end{bmatrix} \quad (2.58)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) = \begin{bmatrix} M\ddot{q}_1 + ml\cos(q_2)\ddot{q}_2 + (-ml\sin(q_2)\dot{q}_2\dot{q}_2) + m\ddot{q}_1 \\ ml^2\ddot{q}_2 + ml\cos(q_2)\ddot{q}_1 + (-ml\sin(q_2)\dot{q}_1\dot{q}_2) \end{bmatrix} \quad (2.59)$$

$$= \begin{bmatrix} M\ddot{q}_1 + ml\cos(q_2)\ddot{q}_2 - ml\sin(q_2)\dot{q}_2^2 + m\ddot{q}_1 \\ ml^2\ddot{q}_2 + ml\cos(q_2)\ddot{q}_1 - ml\sin(q_2)\dot{q}_1\dot{q}_2 \end{bmatrix} \quad (2.60)$$

$$\frac{\partial L}{\partial q} = \begin{bmatrix} \frac{\partial L}{\partial q_1} \\ \frac{\partial L}{\partial q_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 + \frac{1}{2}m[0 + 2l\dot{q}_1\dot{q}_2\sin(q_2) + 0] + mgl\sin(q_2) \end{bmatrix} \quad (2.61)$$

$$= \begin{bmatrix} 0 \\ -ml\sin(q_2)\dot{q}_1\dot{q}_2 + mgl\sin(q_2) \end{bmatrix} \quad (2.62)$$

$$\tau = \begin{bmatrix} U \\ 0 \end{bmatrix} =$$

$$= \begin{bmatrix} M\ddot{q}_1 + ml\cos(q_2)\ddot{q}_2 - ml\sin(q_2)\dot{q}_2^2 + m\ddot{q}_1 \\ ml^2\ddot{q}_2 + ml\cos(q_2)\ddot{q}_1 - ml\sin(q_2)\dot{q}_1\dot{q}_2 + ml\sin(q_2)\dot{q}_1\dot{q}_2 - mg\sin(q_2) \end{bmatrix} \quad (2.63)$$

$$= \begin{bmatrix} (M + m)\ddot{q}_1 + ml\cos(q_2)\ddot{q}_2 - ml\sin(q_2)\dot{q}_2^2 \\ ml^2\ddot{q}_2 + ml\cos(q_2)\ddot{q}_1 - mg\sin(q_2) \end{bmatrix} \quad (2.64)$$

Como

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2.65)$$

Tenemos entonces

$$\begin{bmatrix} M + m & ml\cos(q_2) \\ ml\cos(q_2) & ml^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} 0 & -ml\sin(q_2)\dot{q}_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -mg\sin(q_2) \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \quad (2.66)$$

Donde  $\tau$  corresponde al factor de potencia que nos da el mecanismo de transmisión y es la energía que se aplica al motor para controlar el sistema.

### 2.3 Simulación del sistema de control del péndulo invertido.

Un software de simulación es una herramienta capaz de reproducir el comportamiento de un sistema mediante las ecuaciones matemáticas que lo describen. Además de las ecuaciones que describen la dinámica del sistema, una simulación requiere los parámetros físicos que se describen en la Tabla 2.4.

**Tabla 2.4:** Parámetros del péndulo invertido.

Parámetro	Símbolo	Valor
Masa del carro	$m_c$	1.53 Kg
Masa del péndulo	$m_p$	0.44 Kg
Distancia al centro de masa	$l$	0.35 mts

Estos parámetros se obtuvieron pesando, con la ayuda de un dinamómetro, los componentes por partes (masa del carro y del péndulo), la distancia al centro de

masa se obtuvo encontrando el punto de equilibrio de las varillas y midiendo con un flexómetro. Cabe hacer mención que los parámetros de momento de inercia y fricción no se toman en cuenta debido a que no se tienen, además de que son despreciables debido a la composición mecánica del carro ya que cuenta con llantas para evitar dicha fricción.

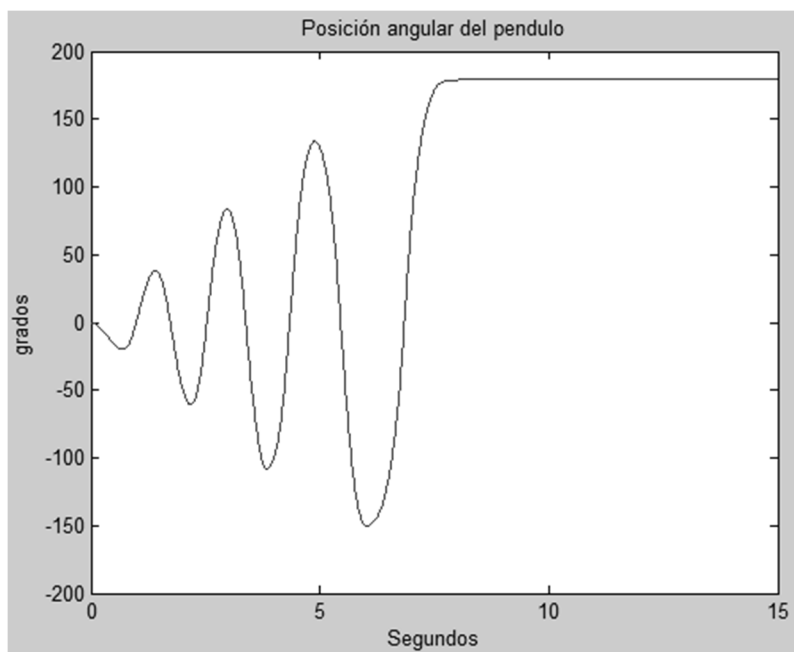
La simulación se realiza con el software MATLAB que es un programa de cálculo numérico orientado a matrices. Por tanto, será más eficiente si se diseñan los algoritmos en términos de matrices y vectores [9].

MATLAB (abreviatura de *MATrix LABORatory*, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

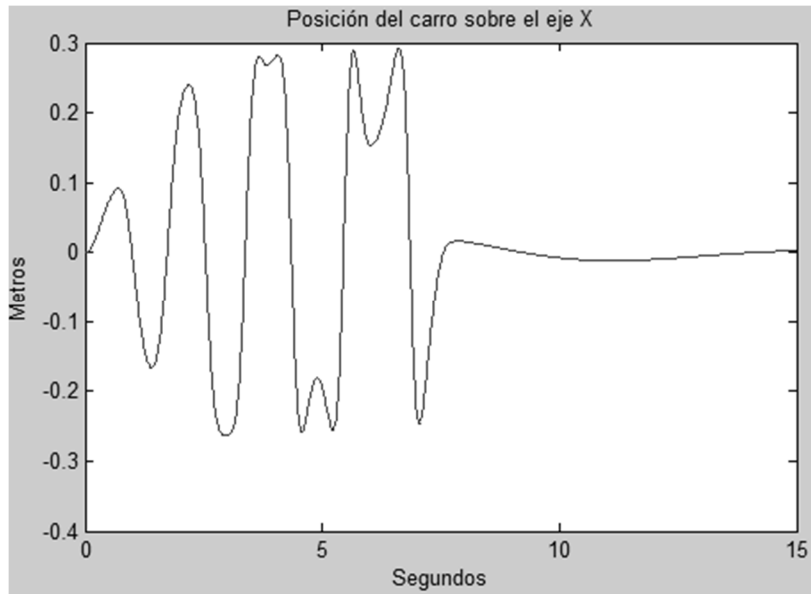
### 2.3.1 Verificación experimental y resultados de los algoritmos de control.

Para verificar los algoritmos diseñados para levantamiento y estabilización del péndulo se realizó lo siguiente: El péndulo está inicialmente situado en la posición de equilibrio inferior, cuando el control inicia su funcionamiento, después de unas cuantas oscilaciones para el levantamiento, el péndulo es colocado cerca de la posición de equilibrio inestable.

Se obtienen las gráficas de comportamiento del péndulo invertido sobre base móvil,

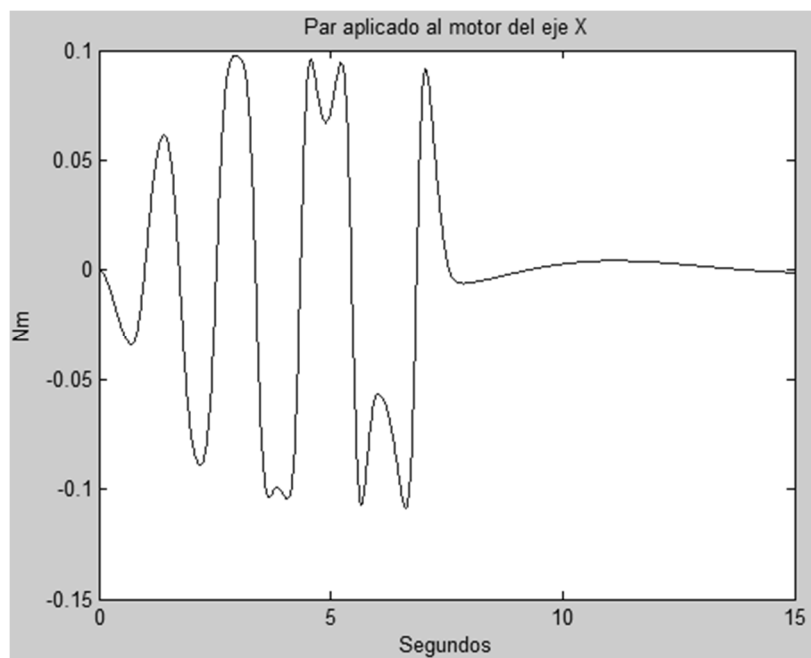


**Figura 2.4:** Levantamiento y estabilización del péndulo con  $\theta = 180^\circ$  final.



**Figura 2.5:** Posición del carro sobre el riel.

Como se puede observar a partir de las formas de onda de las gráficas de las Figuras 2.4 y 2.5, la estrategia de control diseñada es capaz de girar hacia la parte superior para que el sistema entre en el rango de estabilización y así el péndulo llegue a la posición de equilibrio superior incluso cuando se aplica la acción de control durante la secuencia de oscilación se corta de manera significativa por fuera de los límites de los parámetros del motor.



**Figura 2.6:** Par aplicado al motor para movimiento del péndulo invertido

En la figura 2.6 se observa que el par máximo para el movimiento del péndulo se encuentra entre  $\pm 0.1$  Nm, lo cual no sobrepasa las características del motor. Por consiguiente el motor seleccionado puede ser usado para nuestro propósito.

## **Conclusiones**

En este capítulo se presentan conceptos importantes para poder entender y realizar el control del péndulo invertido, se emplea la metodología Euler-Lagrange, para poder obtener el modelo dinámico del sistema, esto con el fin de poder conocer el comportamiento de nuestro robot y obtener una idea más clara de cómo actuará aplicándole el controlador propuesto.

La utilidad del modelo cinemático y dinámico en los robots, es fundamental para propósitos de simulación, diseño y construcción del sistema mecánico, así como en el análisis y diseño de algoritmos de control, es fundamental conocer el modelo dinámico, sobre todo cuando la técnica de diseño se basa en esta estructura del modelo, en este caso las propiedades matemáticas del modelo dinámico son explotadas para facilitar el análisis y hacer propuestas de nuevas estrategias de control.

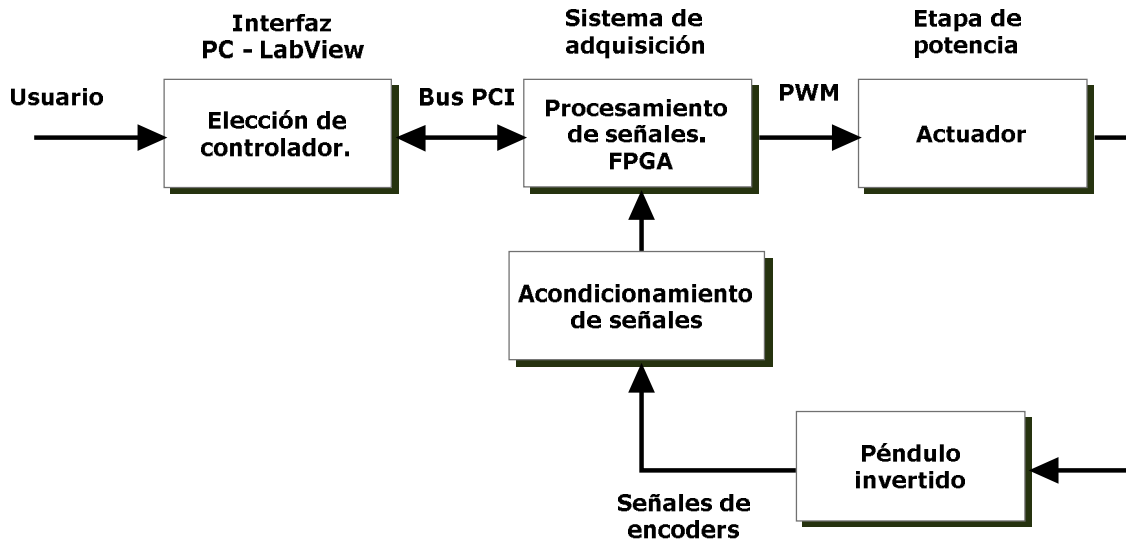
## Capítulo 3

### Diseño del Hardware

En este capítulo se describen las partes del hardware que conforman el sistema del robot péndulo, que comprende la instrumentación de nuestro sistema, como lo es el motor, la etapa de potencia que nos brindará la energía para que el motor se mueva en este caso se utiliza un puente H para controlar el giro del motor, así como la fuente de alimentación, los sensores de posición y su acondicionamiento, también se describe la tarjeta de adquisición de datos utilizada.

#### 3.1 Hardware.

En la Figura 3.1 se observa el diagrama a bloques para la implementación de la electrónica del sistema que se requiere para el prototipo propuesto, el actuador en este caso el motor se encuentra acoplado a un encoder, así como el encoder del péndulo que en este caso es la planta a controlar, el encoder cuenta con dos señales que hay que manipular para nuestro sistema (señales A y B), las cuales deben de ser acopladas de manera correcta al FPGA, ya que cualquier problema de voltaje podría dañar a este último, este FPGA a su vez esta acoplada a una tarjeta PCI, esta tarjeta es para poder enviar y recibir datos de la computadora al FPGA y así poder hacer el proceso de lectura de los encoder, del motor y del péndulo, hacia la computadora, una vez procesadas la señales con el ordenador, con la ayuda del software LABVIEW, se controlarán todos esas señales y se podrán visualizar en la pantalla del usuario para que este pueda darle órdenes al robot péndulo invertido y así poder realizar el control del sistema, de tal manera que pueda ser entendible para el usuario del sistema.

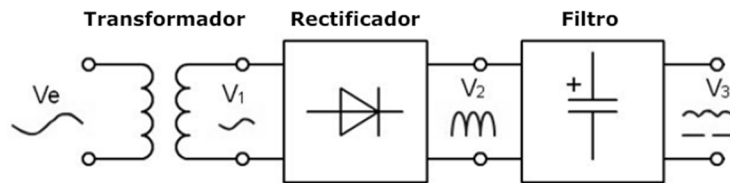


**Figura 3.1:** Diagrama a bloques para la implementación de la electrónica.

### 3.2 Fuente de alimentación de potencia.

Para que el carro del péndulo invertido se pueda desplazar sobre el riel es necesario suministrar al motor la energía suficiente para ello, por lo cual se requiere de una fuente de alimentación la cual a continuación se describe.

Una fuente de alimentación o de poder convierte la entrada de C.A. (corriente alterna) de la línea de 60 Hz a una salida de voltaje de C.D. (corriente directa). Las partes básicas de una fuente básica se muestran en la Figura 3.2.



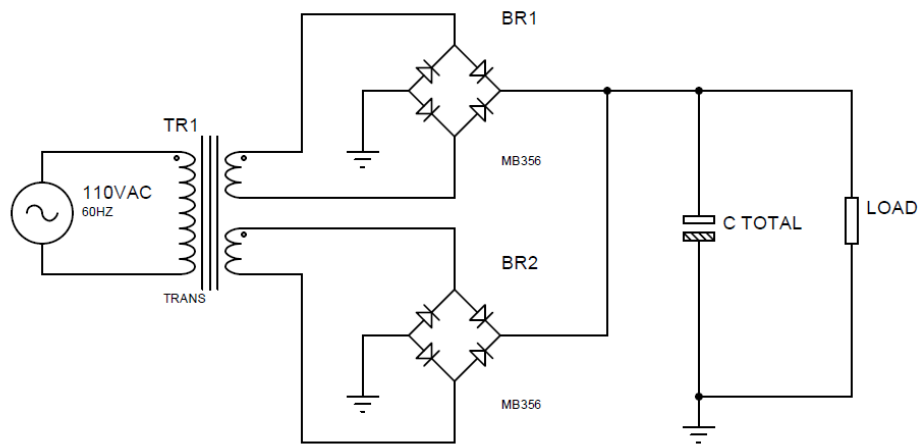
**Figura 3.2:** Diagrama a bloques para la implementación de la electrónica.

Debido a que el motor de corriente directa necesita una gran cantidad de corriente para poder realizar su trabajo es importante contar con una fuente de alimentación la cual proporcione dicha energía, por ello se construyó una fuente con las características de la Tabla 3.1.

**Tabla 3.1:** Características de la fuente de alimentación.

Característica	Valor
Voltaje de alimentación	110 Vca 60 Hz
Voltaje de salida	24 Vcd
Potencia	360 W
Corriente de salida.	15 A.

Se tiene el siguiente diagrama de la fuente de alimentación, Figura 3.3.



**Figura 3.3:** Circuito eléctrico de la fuente de alimentación.

En donde tenemos que:

TR1 = Voltaje primario 110 Vca, 60 Hz, 2 secundarios de 17 Vca de 7.5 A c/u.

BR1, BR2 = puente rectificador MB356, 600V, 35A.

C\_TOTAL = 27 x 4700 $\mu$ F 63V.

### 3.2.1 Cálculo del capacitor de filtrado.

Una formula práctica muy aceptada para calcular el valor de la capacidad del capacitor, para un cierto valor de tensión de rizado y de corriente de salida, y se obtiene de la siguiente expresión [10]:

$$V_{rizo} = \frac{V_p}{2fCR} = \frac{1}{2fC} \frac{V_p}{R} = \frac{I_{carga}}{2fC} \quad (3.1)$$

Despejamos  $C$  de la ecuación (1), entonces tenemos:

$$C = \frac{I_{carga}}{2fV_{rizo}} \quad (3.2)$$

- De la ecuación (2) tenemos:

- $C$  = Capacitancia (F)
- $I_{carga}$  = Corriente de salida (A)
- $f$  = Frecuencia de la onda.
- $V_{rizo}$  = Voltaje de rizo (V)

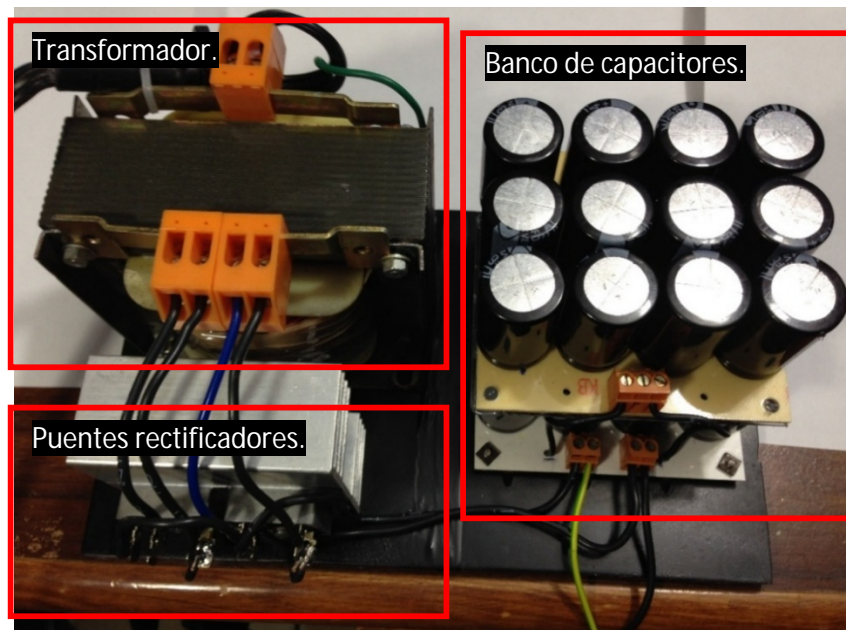
Si dividimos el resultado de la ecuación (2) por  $10^{-6}$ , se obtiene el valor del capacitor en microfaradios.

Tenemos que el motor consume 20 A, pero el transformador solo genera 15A, a un voltaje de 24 VDC, y tomando que un voltaje de rizo aceptable es de 1 V, sustituyendo los valores anteriores la ecuación (2), tenemos:

$$C = \frac{15A}{2(60\text{ Hz})(1V)} = 125000\mu F \quad (3.3)$$

Como se observa el resultado de la ecuación 3.3, este valor es muy grande para capacitores electrolíticos comerciales, y los existentes son demasiado grandes físicamente y también son muy caros, es por eso que se opta por construirlo con varios capacitores en paralelo, para esto se utilizan capacitores de  $4700\mu F$  a 63V, entonces dividiendo el valor obtenido entre el valor del capacitor comercial, tenemos el número de capacitores a utilizar, que en este caso es de  $26.59 \approx 27$ .

Como observación se tiene que, si el capacitor es grande significa menos rizado, pero aun cumpliéndose esta condición, el rizado podría ser grande si la resistencia de carga es muy pequeña (corriente en la carga es grande). En la Figura 3.4 observamos la construcción de la fuente de alimentación con sus componentes.



**Figura 3.4:** Fuente de alimentación de 24VDC construida.

### 3.2.2 Fuente de alimentación de control.

Para la etapa de control del Puente H, se ocupa una fuente ATX de computadora, Figura 3.5, ya que esta proporciona voltajes de 3.3Vdc, 5Vdc y +/-12V, (dichos voltajes son seguros para el uso del sistema debido a que son derivados de una fuente conmutada, estos valores no presentaran cambios al haber una fluctuación en la alimentación de dicha fuente) los cuales ocupamos en la alimentación de los encoders, puente H, sensores de corriente así como en el FPGA que maneja voltaje de 3.3V.



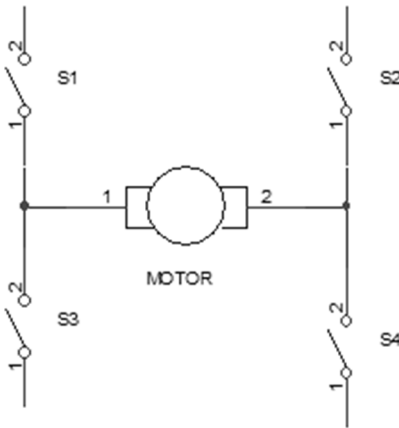
**Figura 3.5:** Fuente de alimentación atx de computadora.

## 3.3 Diseño del Puente H.

Como una mejora para el funcionamiento del Puente H, se implementó este dispositivo con transistores MOSFET, esto con el fin de incrementar el manejo de corriente, así como un circuito de seguridad para evitar que los transistores queden en corto y se destruyan.

### 3.3.1 Activación de MOSFET's.

Los interruptores mostrados en la Figura 3.6 nombrados como S1, S2, S3 y S4 son transistores MOSFET de potencia que tienen ciertas propiedades que les permiten manejar altas corrientes basadas en la activación de una señal de entrada. Los MOSFET son usados en dos regiones de operación; modo de corte y saturación, los cuales corresponden a los modos de apagado y encendido respectivamente.

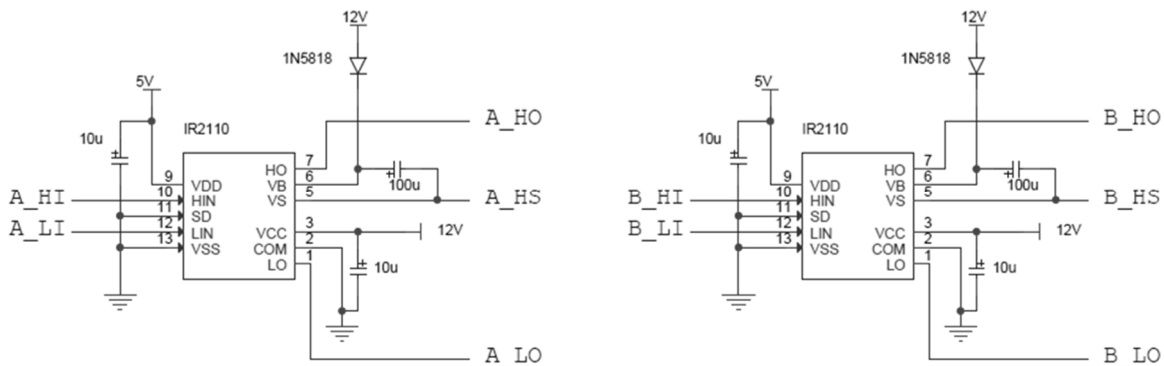


**Figura 3.6:** Circuito general de un Puente H.

En el caso del puente H, para poner un MOSFET en modo de corte, la señal de entrada (Voltaje de puerta) al MOSFET debe estar aterrizada. Sin embargo, para encender el MOSFET y ponerlo en modo de saturación requiere un proceso más complicado.

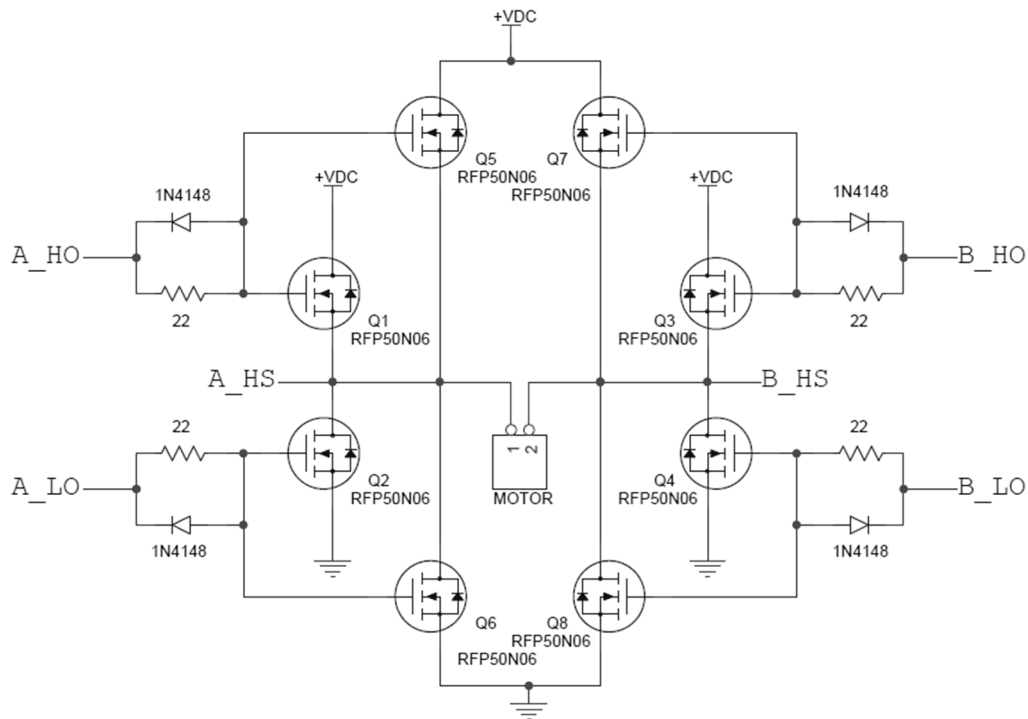
Para tener el MOSFET en modo de saturación el voltaje en la terminal Gate debe ser aproximadamente 12 Volts más alto que el voltaje de la terminal Source.

Como la fuente de alimentación solo nos proporciona 24VDC, entonces se tiene el inconveniente para activar la terminal Gate con 36 Volts ( $24V + 12V$ ), para ello se hace uso del chip driver MOSFET IR2110, [11], que soluciona este problema usando una bomba de carga, Figura 3.7.



**Figura 3.7:** Circuito driver IR2110 en configuración bomba de carga.

Una de las características de los transistores MOSFET es que se pueden ser colocados en paralelo, aprovechando este hecho se colocan en el puente H y queda construido de la siguiente manera, Figura 3.8.



**Figura 3.8:** Configuración Puentes H dual MOSFET.

### 3.3.2 Circulación de corriente regenerativa.

Una ley de la electrónica básica establece que la corriente no puede cambiar instantáneamente a través de un inductor. La bobina principal del motor es un inductor grande, la corriente que corre a través del motor solo puede cambiar gradualmente. Los cambios abruptos provocan picos de voltaje de realimentación. Una técnica para evitar este hecho, es la de circulación de corriente regenerativa (RCC), es implementada redirigiendo la corriente no utilizada hacia el banco de capacitores de la fuente de alimentación, es decir, se activan los transistores inferiores S3 y S4 de la Figura 3.6 o en su defecto se activan los MOSFET con las señales A\_LO y B\_LO que se observan en la Figura 3.8.

### 3.3.3 Lógica de control de dirección de rotación del motor y activación de MOSFET's.

Para realizar el control de dirección del motor y evitar que los transistores MOSFET se activen al mismo tiempo, lo que provocaría que el puente H quedara en corto y destruir los MOSFET, se realiza una Tabla 3.2 de activación de acuerdo a la conmutación de los transistores de la Figura 3.6, con las siguientes variables.

- bits de entrada: Izq., Der y PWM
- bits de salida: S1, S2, S3 y S4.

**Tabla 3.2.** Señales de control de dirección y activación de transistores MOSFET.

Estado de motor	Bits de dirección		Bit	salidas			
	Izq	Der	PWM	S1	S2	S3	S4
Detenido	0	0	0	0	0	1	1
	0	0	1	0	0	1	1
Derecha	0	1	0	0	0	1	1
	0	1	1	1	0	0	1
Izquierda	1	0	0	0	0	1	1
	1	0	1	0	1	1	0
Detenido	1	1	0	0	0	1	1
	1	1	1	0	0	1	1

De la Tabla 3.2 y resolviendo por mapas de Karnaugh [23] se obtiene las siguientes funciones:

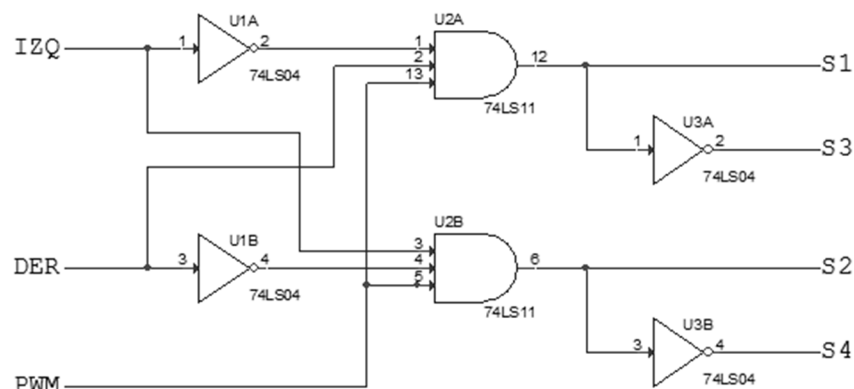
$$S1 = \overline{Izq} \cdot Der \cdot PWM \quad (3.4)$$

$$S2 = Izq \cdot \overline{Der} \cdot PWM \quad (3.5)$$

$$S3 = \overline{S1} \quad (3.6)$$

$$S4 = \overline{S2} \quad (3.7)$$

Pasando las ecuaciones 3.4, 3.5, 3.6, y a lógica digital tenemos el diagrama de la Figura 3.9.

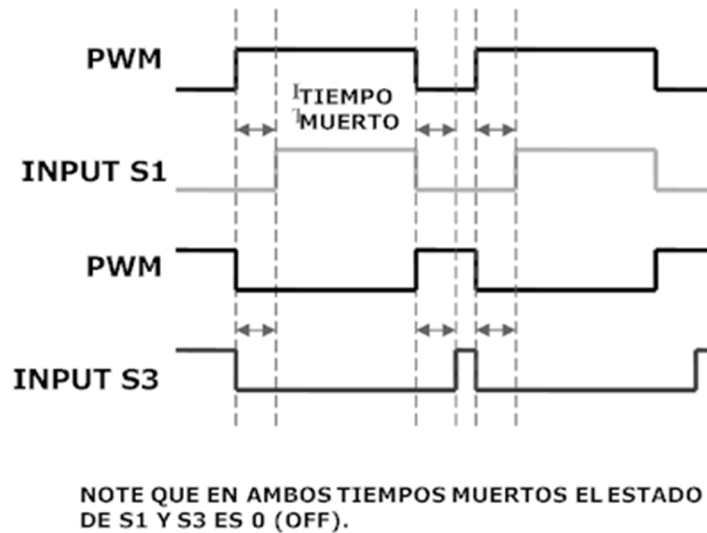


**Figura 3.9:** Lógica de control de activación de transistores MOSFET.

Debido a la implementación del RCC, existe un peligro inherente; si el interruptor S1 y S3 de la Figura 3.6 se activan al mismo tiempo, aun por una pequeña cantidad de tiempo, la fuente de alimentación estará en corto y el puente H literalmente volará. Esto es llamado "Shoot Through" (disparo a través de), para evitar este factor, se incorpora al circuito del puente H una protección.

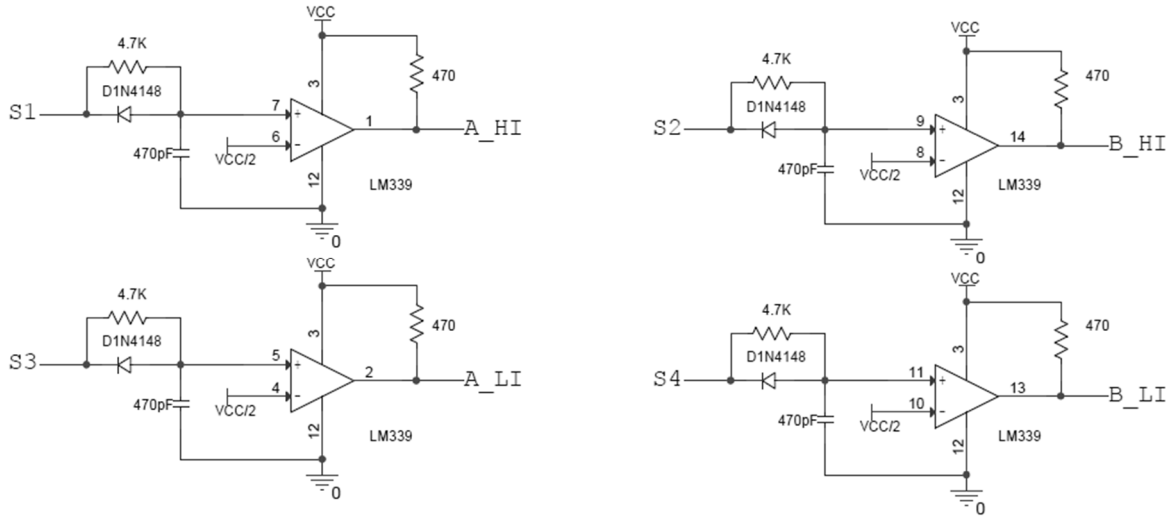
### 3.3.4 Protección contra Shoot-Through.

Para prevenir la condición que causa el Shoot-Through, un circuito denominado "tiempo muerto" es introducido, en la Figura 3.10 se muestra su modo de operación.



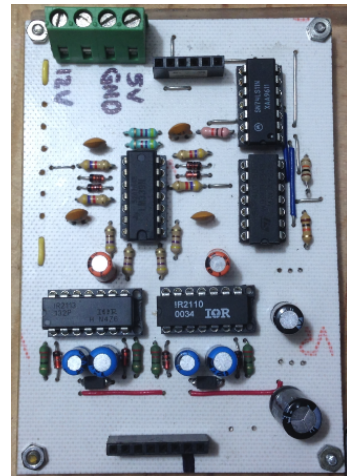
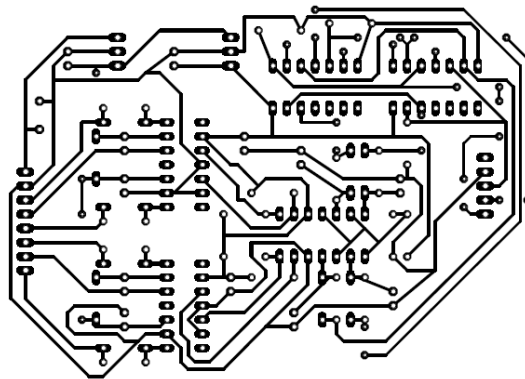
**Figura 3.10** Modo de operación del circuito "Tiempo muerto".

El tiempo muerto es realizado demorando solamente el flanco de subida del PWM como se muestra en la Figura 3.10. El flanco de bajada pasa a través del circuito de tiempo muerto y no es afectado. El driver IR2110 adjunta una pequeña cantidad de tiempo muerto, sin embargo, se incorpora un circuito de tiempo muerto para asegurar este estado, como se muestra en la Figura 3.11. El circuito de tiempo muerto proporciona aproximadamente un retardo de 1 $\mu$ s a los flancos de subida del PWM, lo cual asegura que los MOSFET nunca encenderán al mismo tiempo.



**Figura 3.11:** Circuito electrónico "Tiempo muerto".

Una vez que se probó el funcionamiento del circuito se trasladó a una tarjeta de circuito impreso para su fácil manipulación, en la Figura 3.12, se muestra el enrutamiento y su construcción final.



**Figura 3.12:** Enrutamiento para la placa PCB (izquierda) y tarjeta terminada.

En la Figura C.1 del apéndice C, se muestra el diagrama de conexión eléctrico del puente H.

### 3.4 Caracterización del motor.

La caracterización del motor se lleva a cabo por medio de la prueba de rotor bloqueado, que consiste en encontrar una frecuencia a la cual nuestro actuador trabaje de una manera óptima, lo que se busca es que el motor trabaje en su región de operación lineal y la respuesta sea en base al ciclo de trabajo de la

aplicación de un PWM, de esta forma se puede manipular por medio del firmware y obtener un mejor control del sistema.

### 3.4.1 Torque.

El torque o par motor es el momento de fuerza que ejerce un motor sobre el eje de transmisión de potencia [12].

Si un cuerpo es capaz de rotar sobre un eje, el resultado de la fuerza es una combinación de la fuerza aplicada y la distancia al eje rotacional.

A este resultado se le conoce como par o torque y es el producto vectorial de multiplicar la fuerza  $\vec{F}$  por la distancia perpendicular al eje de rotación  $\vec{r}$ , donde  $\vec{F}$  y  $\vec{r}$  son vectores.

El torque este representado por la siguiente formula 3.8 [13]:

$$\vec{\tau} = \vec{F} \times \vec{r} \quad (3.8)$$

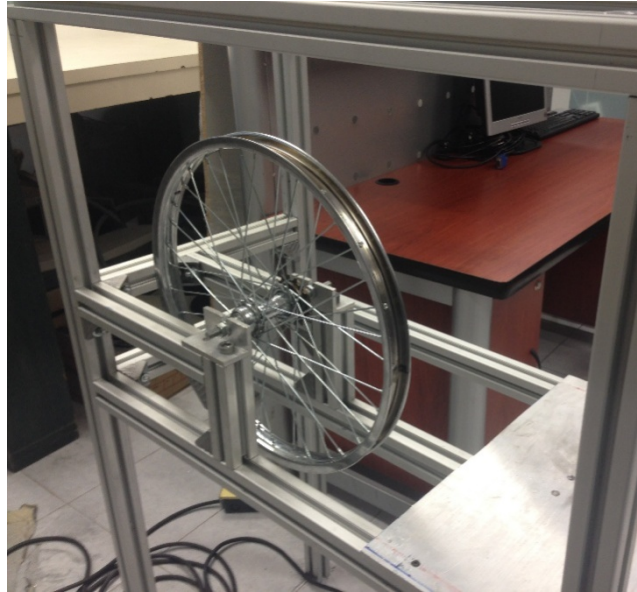
Donde:

- $\vec{\tau}$  = Torque (N-m)
- $\vec{F}$  = Fuerza (Kg)
- $\vec{r}$  = radio del eje motor (m)

### 3.4.2 Desarrollo de torquímetro.

Debido a que la velocidad y la fuerza ejercida sobre el eje del motor son demasiado altas y para no dañar el dinamómetro digital que es utilizado para hallar el torque máximo, se construyó, en conjunto con compañeros de la maestría, un torquímetro para atenuar tanto la velocidad y fuerza ejercida por el motor, así también nos permite tener mediciones más estables.

El torquímetro se construye a base de perfil de aluminio de 40 x 40 mm para la estructura principal, una placa de aluminio para soporte del motor que estará sometido a la caracterización, se utilizan también rines, sprokets y cadena de bicicleta. En la siguiente Figura 3.13 se observa la estructura del torquímetro y en la Figura B.9 (apéndice B), se muestra el diseño en SolidWorks.



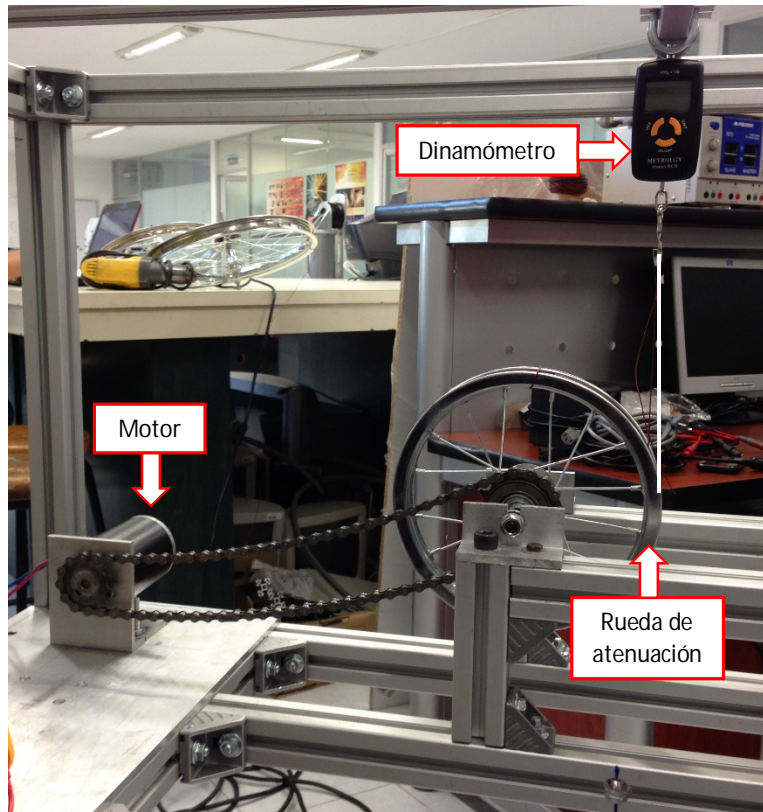
**Figura 3.13:** Construcción final del torquímetro.

### **3.4.3 Procedimiento de caracterización de motor.**

Es de vital importancia conocer algunas características del motor para tener un buen control sobre su comportamiento al aplicar el PWM, es decir, conocer en que parte de su funcionamiento se comporta de forma lineal y así aplicar las condiciones necesarias para tener un buen control. Como dato único, el valor que se conoce del motor es el voltaje nominal de trabajo que es de 24VDC.

Para obtener el torque nominal del motor se realiza el siguiente procedimiento:

Se coloca el motor en el torquímetro, como se muestra en la siguiente Figura 3.14.

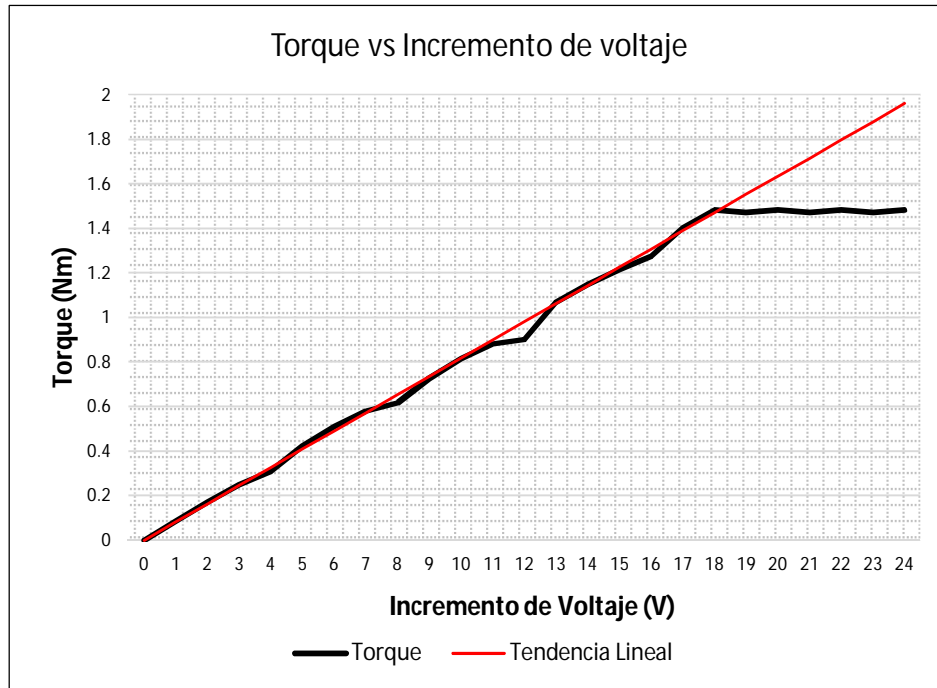


**Figura 3.14:** Montaje del motor en el torquímetro.

Con la ayuda de una fuente variable se procede a inducir incrementos de voltaje al motor y medir el valor obtenido en el dinamómetro y en el sensor de corriente. Obtenemos los valores mostrados en la Tabla 3.3.

**Tabla 3.3:** Torque obtenido del motor al incrementar voltaje de alimentación.

$\Delta V(\text{DC})$	Torque (Nm)	$\Delta V(\text{DC})$	Torque (Nm)
0	0.00	13	1.07
1	0.08	14	1.15
2	0.17	15	1.22
3	0.25	16	1.28
4	0.31	17	1.40
5	0.42	18	1.48
6	0.51	19	1.47
7	0.58	20	1.48
8	0.62	21	1.47
9	0.73	22	1.48
10	0.81	23	1.47
11	0.88	24	1.48
12	0.90		



**Figura 3.15:** Gráfica de comportamiento del motor al incrementar voltaje de alimentación.

De los datos obtenidos de la Tabla 3.3 y a partir de la gráfica de la Figura 3.15 se observa que en el rango de 0-18V el comportamiento del motor con respecto a su torque es cercano al comportamiento lineal pero a partir de los 18V el torque se mantiene a un máximo de 1.48 Nm esto debido a que la fuente de alimentación solo puede generar 15 amperes a 24V, al incrementar el voltaje por arriba de 18V este cae hasta este voltaje y no alcanza a llegar al voltaje nominal de 24V y por consecuencia el torque se mantiene hasta 1.48 Nm.

Debido a que no se alcanza a obtener el torque máximo aplicando el voltaje y corriente necesaria al motor se tiene que linealizar la respuesta del torque y solo así se puede visualizar que el torque máximo es 1.96 Nm, este valor nos sirve para realizar el siguiente paso que es el de obtener la frecuencia óptima para el funcionamiento del PWM.

Como usaremos una frecuencia para su excitación, y al modificar el ciclo de trabajo obtenemos el mismo par y modificamos la velocidad del motor, esto es lo que se requiere para realizar el control del sistema.

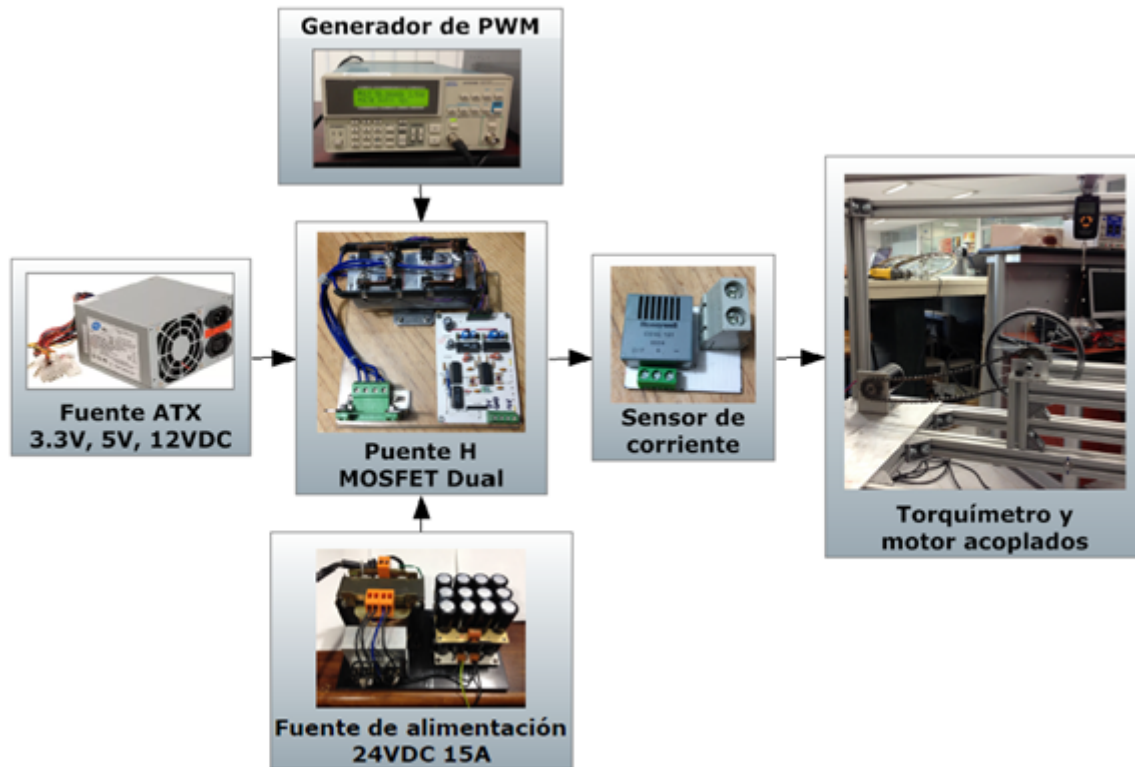
La caracterización consiste en someter al motor a un rango de frecuencias de 50 Hz a 50 KHz, con un incremento de 50 Hz hasta el valor de 1 KHz, después de esta frecuencia los incrementos de frecuencia se llevan de 1 KHz hasta 10 KHz,

estas frecuencias se manejan a un ciclo de trabajo del 50%, y seleccionamos aquellas frecuencias en las que se aproxima a la mitad del torque máximo, además a cada una de estas frecuencias se le hizo una modulación de ancho de pulso de 10% hasta al 100%, en incrementos de 10%, obteniendo así los datos donde encontraremos la frecuencia ideal para obtener el mejor desempeño de nuestro motor.

Para poder realizar los cambios de frecuencia y el ancho de pulso, se ocupa un puente H con transistores MOSFET, excitando dicho puente H con un generador funciones, aplicando una amplitud de 3.3V, que es el voltaje que enviará el FPGA para generar las distintas frecuencias y cambiar el ciclo de trabajo.

Para obtener la fuerza a la que trabaja el motor, se hizo un acoplamiento del motor a un dinamómetro, aplicando las frecuencias y ejerciendo una fuerza opuesta al giro del motor, es decir bloquear el rotor.

En la Figura 3.16 se observa el diagrama a bloques del sistema utilizado para realizar la caracterización, se utiliza un generador de funciones para aplicar el PWM y hacer la modificación de ancho de pulso y así poder observar cómo funciona con distintas frecuencias, la fuente de alimentación para proporcionar la potencia requerida por el motor, el sensor de corriente para saber el consumo del motor y el dinamómetro para saber la fuerza que se genera al activar el motor sobre un punto fijo.



**Figura 3.16:** Diagrama para la caracterización del motor.

En la Figura 3.16 podemos observar el montaje realizado para caracterizar nuestro motor, después de haber realizado las pruebas para las distintas frecuencias, obtuvimos los valores de la Tabla 3.4, utilizando de voltaje nominal 24V, y el barrido del ciclo de trabajo del PWM.

**Tabla 3.4:** Datos obtenidos al aplicar diferentes frecuencias.

Frecuencia (Hz)	I(A)	Torque(Nm)	Frecuencia (Hz)	I(A)	Torque(Nm)
100	8	1.14777	900	6.8	1.11834
150	7.8	1.1772	950	6.7	1.12815
200	7.6	1.21644	1000	6.8	1.13796
250	7.3	1.19682	2000	6.5	1.09872
300	7.2	1.18701	3000	6.6	1.12815
350	7.2	1.1772	4000	6.6	1.12815
400	7.1	1.16739	5000	6.4	1.10853
450	7	1.16739	6000	6.3	1.10853
500	7	1.13796	7000	6.2	1.08891
550	7.1	1.16739	8000	6.1	1.10853
600	7	1.16739	9000	6.2	1.10853

650	6.9	1.16739	10000	6.2	1.09872
700	7	1.1772	20000	6	1.08891
750	6.9	1.15758	30000	5.6	1.04967
800	6.8	1.14777	40000	6.2	1.09872
850	6.8	1.12815	50000	6.3	1.10853

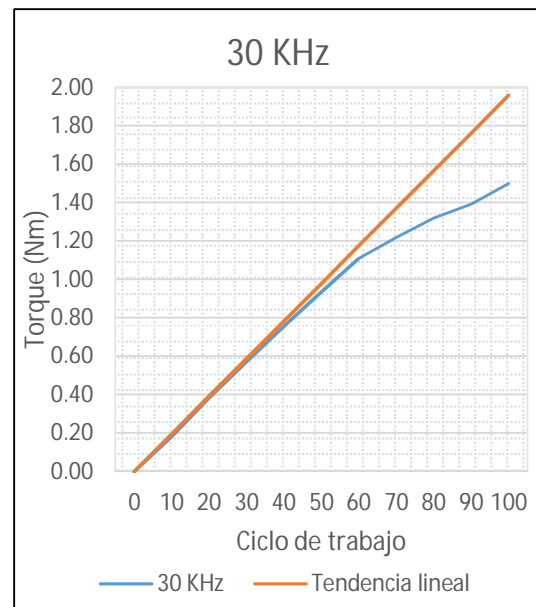
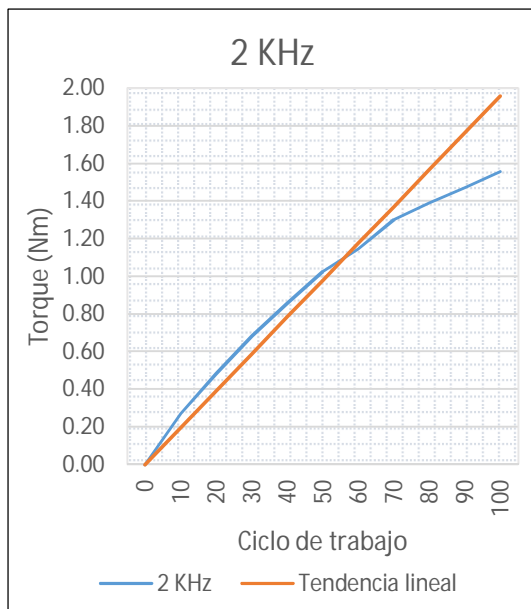
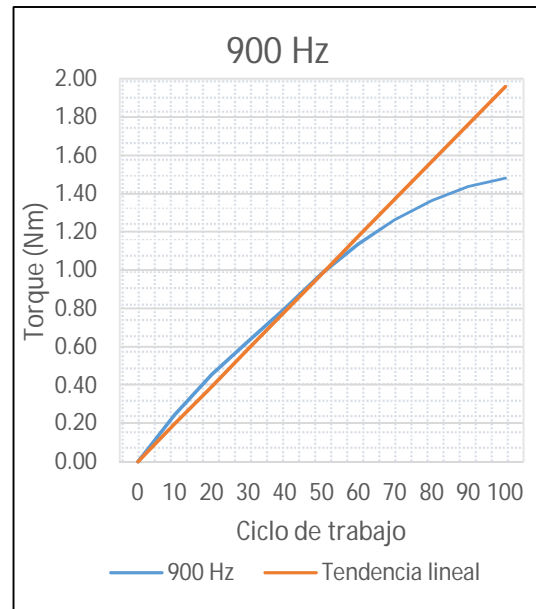
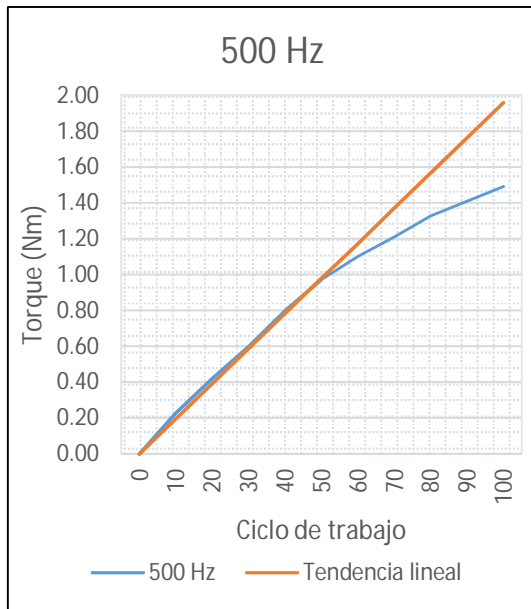
De la Tabla 3.4 se observa que para diferentes frecuencias el 50% del torque total se encuentra alrededor de 1 Nm, por lo que al no poder hallar el torque máximo debido a que la fuente de alimentación no genera suficiente corriente, podemos definir que el máximo torque del motor es de aproximadamente 2 Nm.

Se identifican las frecuencias en las que se obtiene una aproximación a la mitad del torque máximo que son: 500, 900, 2000 y 30000 Hz. A estas frecuencias se les aplica un barrido de ciclo de trabajo de 10% al 100% con incrementos de 10%. Se obtiene las siguientes tablas y de ahí se selecciona la que mejor linealidad presente.

**Tabla 3.5:** Valores obtenidos variando el ciclo de trabajo.

Duty cycle	Torque (Nm) obtenido a:				Tendencia lineal (Nm)
	500 Hz	900 Hz	2 KHz	30 KHz	
0	0.00	0.00	0.00	0.00	0.00
10	0.23	0.25	0.27	0.18	0.20
20	0.42	0.46	0.48	0.38	0.39
30	0.60	0.63	0.68	0.57	0.59
40	0.80	0.80	0.86	0.76	0.78
50	0.97	0.98	1.03	0.94	0.98
60	1.10	1.14	1.15	1.11	1.18
70	1.21	1.26	1.30	1.22	1.37
80	1.33	1.37	1.39	1.32	1.57
90	1.41	1.44	1.47	1.39	1.76
100	1.49	1.48	1.56	1.50	1.96

De los datos obtenidos mostrados en la Tabla 3.5, procedemos a graficarlos para observar el comportamiento del motor al aplicarle la energía en forma de PWM.



**Figura 3.17:** Gráficas de comportamiento del motor al aplicarle PWM y variar el ciclo de trabajo.

De los resultados graficados, Figura 3.17, se observa que el mejor comportamiento lo tenemos en las frecuencias para PWM de 500 Hz y 30 KHz, ya que esta presenta una linealidad en la región de 0 a 60% de ciclo de trabajo,

del 60 al 100% de ciclo de trabajo, no alcanza a llegar al máximo torque, esto debido a que la fuente de alimentación no proporciona suficiente corriente para alimentar al motor, sin embargo con la parte lineal se puede trabajar ya que es un torque de 1.5 Nm que es suficiente para poder mover el motor y este a su vez el carro del péndulo. Se decide trabajar con la frecuencia de 500 Hz para evitar que se genere más ruido eléctrico con la frecuencia de 30 KHz.

### 3.5 Sensor de corriente.

Para poder realizar la caracterización del motor es necesario tener un sensor capaz de medir corrientes superiores a 10A ya que nuestro sistema consume esta cantidad de corriente, para esto se utiliza un sensor de la marca Honeywell modelo CSNE 151-005, el cual es capaz de medir corriente directa y/o alterna, en nuestro caso corriente directa del orden de 36A, y en salida nos proporciona milivolts, por ejemplo, si el motor consume 5 amperes, nuestra salida será de 500mV.

#### 3.5.1 Descripción del sensor

El CSNE 151-005 es un sensor de corriente multi-rango basado en el principio de efecto Hall, en donde pueden ser medidas corrientes hasta de 36A, figura 3.18.

Datos eléctricos [14]:

Corriente nominal primaria  $I_{pn}: 25A$

Rango de medición:  $I_p = 0 a \pm 36A$

Voltaje de alimentación:  $\pm 15V (\pm 5\%)$

Resistencia de carga:  $R_{m\ min}(100\Omega), R_{m\ max}(190\Omega)$

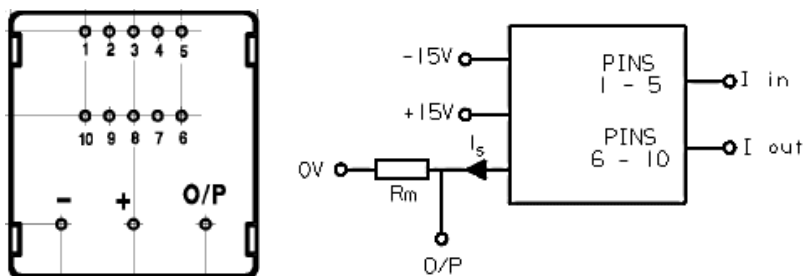


Figura 3.18: Pines y diagrama de conexión, sensor de corriente Honeywell.

### 3.6 Tarjeta PCI.

Como parte principal del hardware del sistema se utiliza para la comunicación de la PC a la instrumentación del sistema una tarjeta PCI como se puede observar en la Figura 3.19, esta tarjeta fue desarrollada por la facultad de ciencias de la electrónica (BUAP), esta se encarga de la comunicación FPGA con el software LABVIEW, para poder hacer la comunicación se desarrolló un firmware de comunicación con la PCI de la PC, esta tarjeta cuenta con 32 bits entrada-salida, para poder realizar la adquisición de datos de nuestro sistema, esta tarjeta contiene una tarjeta hija donde va montado el FPGA, el cual realiza todos los procesos, la transferencia de datos se realiza a 132 Mbytes/seg, con una velocidad en bus de 33Mhz, en la Tabla 3.6 se describen algunas de las características de esta tarjeta.



Figura 3.19: Tarjeta PCI FOTON\_1.

Tabla 3.6: Características tarjeta PCI FOTON\_1

Características tarjeta PCI FOTON_1	
1 conector header de 25 pines I/O	Banco 2
1 Puerto PCI 32 bits	Banco 3 y 4
6 canales de I/O (disponibles)	Banco 1
Voltaje de trabajo	5 y 3.3 VDC

### 3.7 Sensores para el sistema del péndulo invertido.

Un sensor es un dispositivo que detecta y mide una condición de cambio, con frecuencia se trata de la presencia o ausencia de un objeto a esto se le llama detección discreta, también puede medirse un cambio de distancia, tamaño o

color esto es detección analógica. De esta forma se puede monitorear y controlar un proceso.

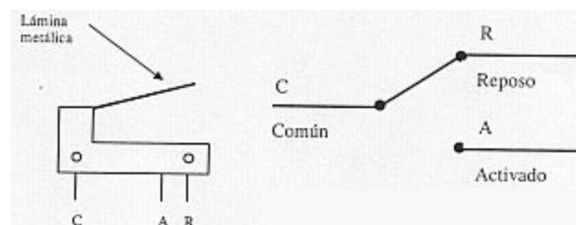
### 3.7.1 Microswitch

El sistema cuenta con sensores de fin de carrera, estos se encuentran sujetos al riel y se colocaron por seguridad para evitar que en caso de que el control presente algún problema el carro choque y se dañe, debido a que el carro alcanza una gran velocidad esto puede ser muy peligroso.

Para los sensores de fin de carrera se usaron sensores de contacto micro switch (Figura 3.20) que son dispositivos electromecánicos que detectan cambios a través del contacto físico, de esta forma se usan como paradas de emergencia.

Constan de un accionador unido mecánicamente a una serie de contactos, cuando el objeto entra en contacto con el accionador el dispositivo activa los contactos para establecer o interrumpir la conexión eléctrica.

Son resistentes, de fácil instalación y confiables. Los términos normalmente abierto y normalmente cerrado describen el estado de cada par de contactos cuando el interruptor se encuentra en la posición de accionamiento o de reposo.



**Figura 3.20:** Diagrama interno de un Microswitch.

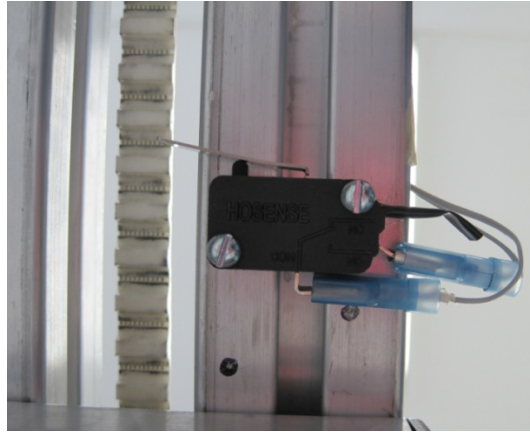
Al presionar la lámina, el borne común C pasa a conectarse con el borne activado A, de otra forma permanece unido con R, reposo. R se identifica como NC o normalmente cerrado y A como NO o normalmente abierto.

Tiene las siguientes ventajas:

- Facilidad de uso
- Operación sencilla
- Capacidad de conmutar cargas elevadas
- Inmunidad a la interferencia de ruido eléctrico y frecuencias de radio
- Ausencia de corriente de fuga
- Pérdidas de voltaje mínimas

Y como desventaja, se tiene que debido al contacto, la vida de los contactos es más breve dependiendo del uso.

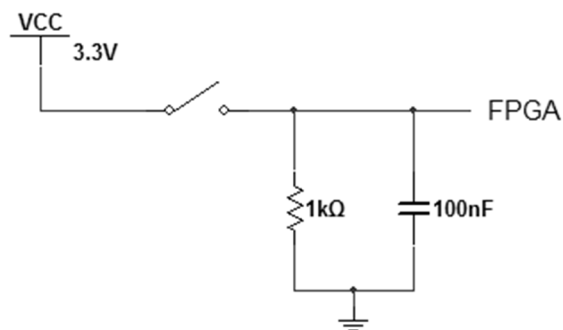
En la Figura 3.21 se muestra el montaje de estos sensores en el riel.



**Figura 3.21:** Sensor de final de carrera atornillado al riel.

El péndulo invertido cuenta con dos, uno a cada extremo del riel.

Para los interruptores de final de carrera se realizó la conexión mostrada en la Figura 3.22, el switch abierto representa al sensor, cuando el carro entra en contacto con este se cierra y deja pasar el voltaje al FPGA, la resistencia se coloca a tierra para evitar que al encontrarse el circuito abierto ocurran flujos indeseados de corriente. Se elaboró una placa para realizar estas conexiones más fácilmente.



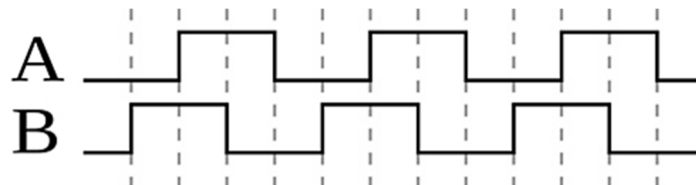
**Figura 3.22:** Conexión para los sensores de final de carrera.

### 3.7.2 Encoders.

Los encoders son sensores que generan señales digitales en respuesta al movimiento. Están disponibles en dos tipos, uno que responde a la rotación y otro que responde al movimiento lineal. Cuando son usados en conjunto con

dispositivos mecánicos tales como engranes, ruedas de medición o flechas de motores, estos pueden ser utilizados para medir movimiento lineal, velocidad o posición, en nuestro caso utilizaremos encoders del tipo incremental.

Teóricamente se puede obtener de los encoders, señales de la siguiente forma, como se puede observar en la Figura 3.23.



**Figura 3.23:** Salidas de encoder incremental A y B.

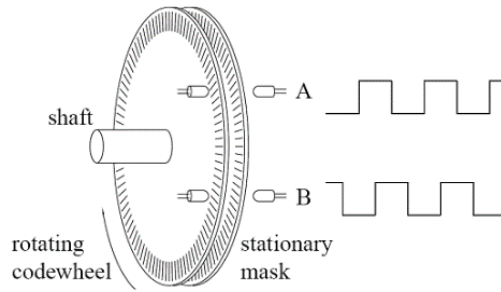
Haciendo el conteo de los flancos (subida y bajada) de ambas salidas se puede obtener, por ejemplo 64 conteos por cada revolución del motor. Si requerimos menos precisión en el sistema se puede solamente contar uno de los dos flancos y obtener una salida de 16 conteos por revolución.

### 3.7.3 Encoder incremental.

Este tipo de encoder se caracteriza porque determina su posición, contando el número de impulsos que se generan cuando un rayo de luz, es atravesado por marcas opacas en la superficie de un disco unido al eje.

Los encoders incrementales son probablemente el tipo más común de encoder utilizado en la industria, por la gran variedad de aplicaciones en las que puede ser usado. Cada encoder incremental tiene en su interior un disco, marcado con una serie de líneas uniformes a través de una única pista alrededor de su perímetro, las líneas impermeables a luz de anchura igual a los huecos transparentes, trabajando como una unidad emisora de luz y una unidad captadora de la misma, al girar el disco, generan unas señales que debidamente tratadas proporcionan las señales de salida de un encoder incremental [15].

En la Figura 3.24 se observa un encoder incremental en donde tenemos dos fotodiodos A y B en conjunto con un disco codificado con perforaciones o ranuras por donde pasa el haz de luz emitido por los fotodiodos y así generar las señales que se observan en la misma figura.



**Figura 3.24:** Disco codificado y generación de pulsos A y B.

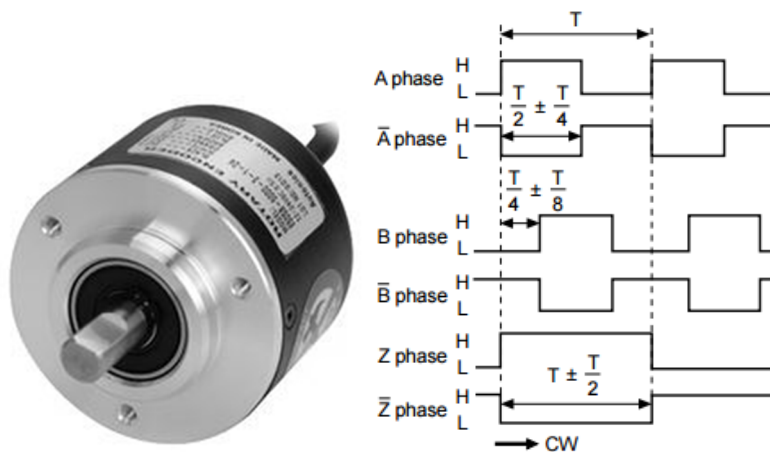
Las señales de salida de un encoder pueden ser un tren de pulsos, en forma de señal cuadrada, donde el número de impulsos que genera en una vuelta coincidirá con el número de impulsos del disco en el interior del encoder, nos referimos a encoders de un solo canal (señal A).

Una segunda señal B suministrara un tren de pulsos idénticos al que suministra la señal A pero desfasada 90° respecto de esta, se refiere a un encoder de dos canales [15].

### 3.7.4 Encoder incremental para motor.

Para nuestro sistema se utilizan dos encoders, uno para el motor y el otro para la posición del péndulo, el encoder para el motor es de la marca Autonics modelo E50S6-2500-6-I-5 ver Figura 3.25, el cual cuenta con las siguientes características [16]:

- 5Vdc Voltaje de alimentación.
- 6 Salidas  $A, \bar{A}, B, \bar{B}, Z, \bar{Z}$ .
- 2500 pulsos por revolución.
- Salida Line Driver



**Figura 3.25:** Encoder modelo E50S6-2500-6-I-5 y señales de salida A y B.

### 3.7.5 Encoder incremental para péndulo.

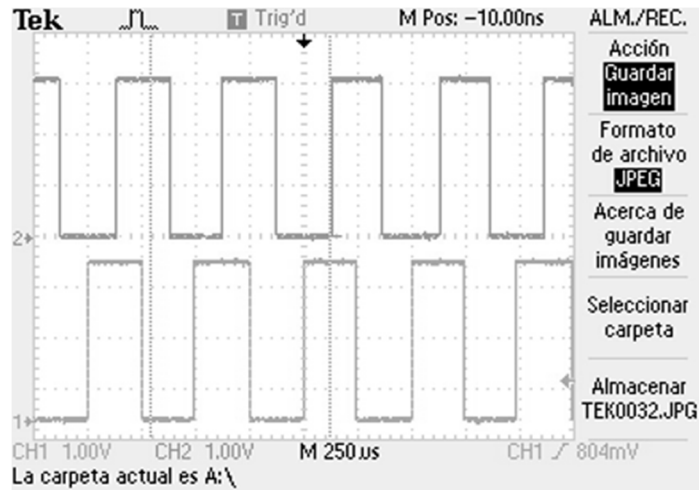
Para el péndulo se utilizó un encoder de eje hueco de la empresa SICK modelo DFS60E-BHC02000 (ver Figura 3.26) con las siguientes características:

- Voltaje de alimentación 5 VDC.
- 6 Salidas  $A, \bar{A}, B, \bar{B}, Z, \bar{Z}$ .
- 2000 pulsos por revolución.
- Salida Line Driver



**Figura 3.26:** Encoder modelo DFS60- BHC02000.

Se realiza una prueba con el encoder para poder observar las señales que nos proporcionan, observar el comportamiento de las señales, aplicando 3.3V a la alimentación del encoder esto para corroborar que las salidas de los sensores también sea de 3.3V y poder introducirla sin problemas al FPGA, ya que el FPGA solo soporta voltaje de 0 a 3.3V pasando ese régimen de voltaje el FPGA se puede dañar una vez que se obtienen las señales de los encoder se le agrega un capacitor de 10 nf en paralelo con la señal y tierra del sistema, esto con el fin de que las señales obtenidas eviten tener ruido, picos de voltaje y así evitar dañar el FPGA.



**Figura 3.27:** Señales A y B obtenidas del encoder.

En la Figura 3.27, se puede observar las señales obtenidas por los encoders, con los filtros capacitivos para evitar el ruido de la señal, se puede observar que las señales están desfasadas  $90^\circ$  tal y como se tenía previsto la señales están dentro del voltaje requerido por el FPGA de 3.3 V de esta manera no existe riesgo de que este se dañe.

Una vez obtenida la señal y acondiciona, se necesita adquirirla en el FPGA, para eso se diseñara un firmware que tendrá la capacidad de realizar conteos por cada flanco que subida que detecte, lo cual se describirá con más detalle en el capítulo de firmware.

## Conclusiones

En este capítulo se profundiza en la etapa de potencia ya que es una de las partes fundamentales del sistema la cual se encarga de suministrar energía de forma controlada al péndulo invertido, también se construye un torquímetro para realizar la caracterización del motor empleado para poder trabajar en su región lineal, se describe también los encoders para determinar la posición y velocidad tanto del carro como del péndulo.

# Capítulo 4

## Diseño del firmware

En este capítulo se describe el diseño del firmware, el cual se encuentra constituido por los módulos siguientes: el protocolo PCI, filtro digital, lector de encoders, multiplexor de datos, generador de frecuencia de PWM, selección de giro.

### 4.1 Firmware

La comunicación entre el firmware y el software se lleva a cabo mediante el bus PCI, el cual es una arquitectura síncrona, debido a que todas sus transferencias de datos se realizan con base a un sistema síncrono.

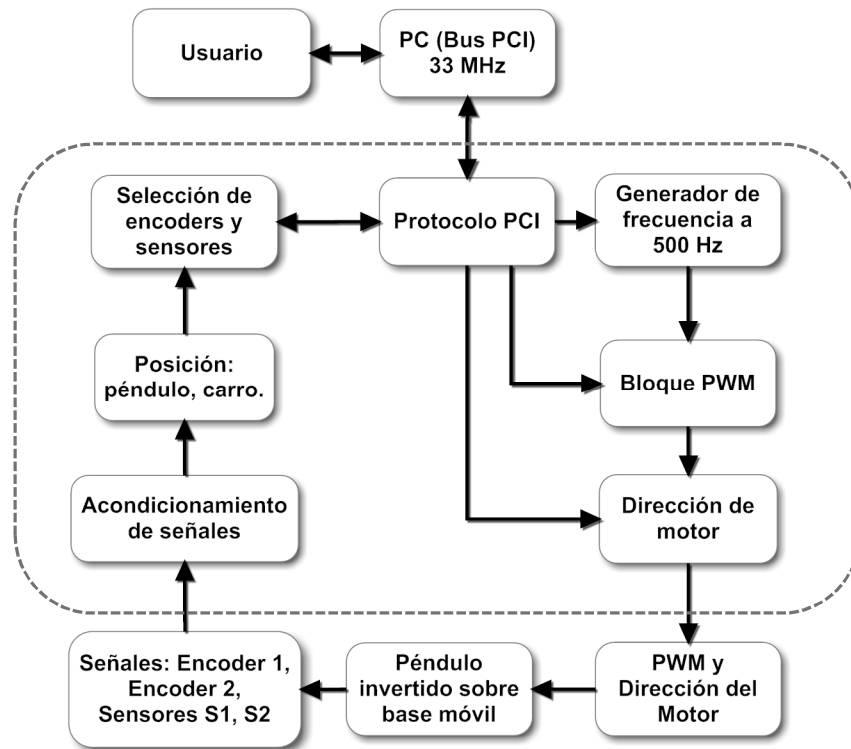
Se describe también el desarrollo para manipular las señales requeridas por el sistema, como las señales de los encoders, las señales para poder controlar el motor y todas las demás señales de instrumentación, todo esto mediante los bloques que contiene el FPGA, lo anterior es desarrollado mediante en software Quartus 9.

Se realiza el desarrollo y programación del firmware del FPGA, en este caso se utiliza un Cyclone II, matrícula EP2C8Q208C7, aquí se descarga el firmware implementado para hacer el control del robot mecatrónico, en este firmware se integra un protocolo PCI, que se encarga de la comunicación del FPGA con el PCI de la computadora, es decir el envío y recepción de datos por medio de LABVIEW, en el firmware de la comunicación PCI acoplaremos la parte del acondicionamiento de la señales de los encoders, y en base a eso poder modular el ancho de pulso que necesitara nuestro motor a través del PWM.

Haciendo una comparación, el firmware es como un sistema operativo que se encarga de dar instrucciones detalladas al FPGA para realizar diversas tareas, todo esto se realiza mediante el lenguaje de bajo nivel que es un código de programación AHDL, el cual se encargará de controlar todos los circuitos electrónicos del sistema.

En el firmware se integra un protocolo PCI, que se encarga de la comunicación del FPGA con el PCI de la computadora, es decir el envío y recepción de datos por medio de LABVIEW, en el firmware de la comunicación PCI acoplaremos la parte del acondicionamiento de la señales de los encoders, y en base a eso poder

modular el ancho de pulso que necesitara nuestro motor a través del PWM así como su dirección. El firmware está desarrollado en el software Quartus 9 de la empresa Altera y está desarrollado en varios bloques que lo conforman, Figura 4.1.



**Figura 4.1:** Diagrama a bloques del Firmware.

## 4.2 Protocolo PCI.

Un PCI (Peripheral Component Interconnect) es un bus de ordenador estándar para conectar dispositivos periféricos directamente a su placa base. Estos dispositivos pueden ser circuitos integrados ajustados en ésta o tarjetas de expansión que se ajustan en conectores. Es común en PC, donde ha desplazado al ISA como bus estándar, pero también se emplea en otro tipo de ordenadores [17].

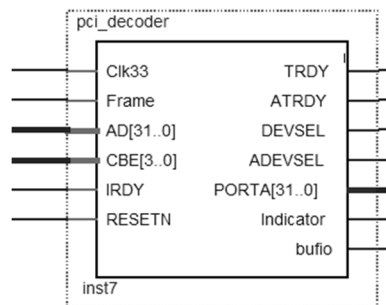
### 4.2.1 Especificaciones de PCI.

Estas especificaciones representan a la versión de PCI más comúnmente usada en los PC [15].

- Reloj de 33 MHz con transferencias síncronas
- Ancho de bus de 32 bits o 64 bits

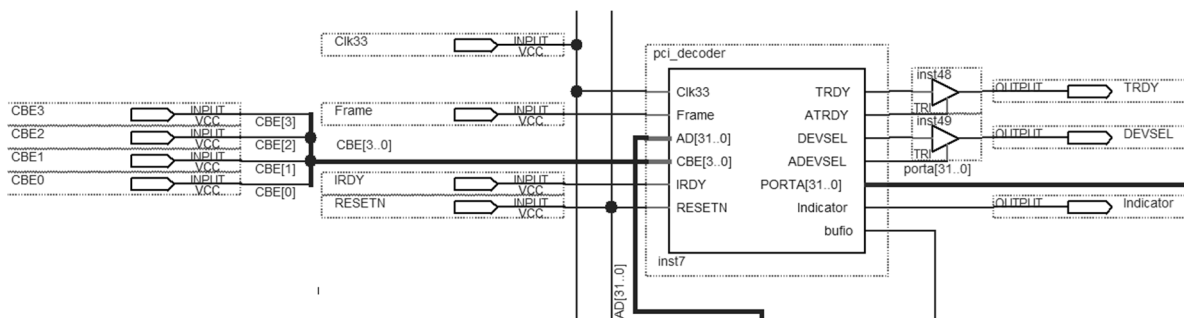
- Tasa de transferencia máxima de 133 MB por segundo en el bus de 32 bits ( $33,33 \text{ MHz} \times 32 \text{ bits} \div 8 \text{ bits/byte} = 133 \text{ MB/s}$ )
- Tasa de transferencia máxima de 266 MB/s en el bus de 64 bits.
- Espacio de puertos I/O de 32 bits.-
- 256 bytes de espacio de configuración.
- 3,3 V o 5 V, dependiendo del dispositivo.

El bloque que se muestra en la Figura 4.2, se desarrolló con anterioridad en la FCE-BUAP, tomando las especificaciones del protocolo PCI de 32 bits, el cual se puede consultar en [18]. Este protocolo toma los datos del bus PCI, los procesa y los envía a través de los diferentes bloques del firmware, el cual comparte con el protocolo PCI el reloj de 33 MHz, esto para sincronizar los datos.



**Figura 4.2:** Símbolo del bloque protocolo PCI.

El firmware PCI se encarga de la comunicación de la tarjeta hija al puerto PCI de la computadora, se utiliza este tipo de comunicación ya que es en tiempo real y así evitamos la pérdida de datos y por ende se presenta una mejor manera de realizar la comunicación entre los dispositivos.



**Figura 4.3:** Bloque de protocolo PCI en software Quartus.

En la Figura 4.3 podemos observar el bloque del firmware del PCI, este es el encargado de la comunicación entre el puerto de comunicación PCI y la tarjeta hija del FPGA, se requiere este tipo de comunicación debido a que las señales tienen que ser tratadas, el problema fundamental es el control del actuador en este caso, el motor, para ello se requieren varias entradas, tanto de entrada como de salida las cuales deben de ser procesadas por el FPGA y este a su vez debe de enviarlas al puerto PCI de la PC para poder interpretarlas correctamente, hacer cálculos y enviarlas de regreso al FPGA y así poder cerrar el lazo de control del sistema.

El código se realiza en Quartus II, para la escritura y lectura de los datos del FPGA, para la comunicación PCI para poder generar el bloque del protocolo PCI como se puede observar en la simulación de la Figura 4.4.

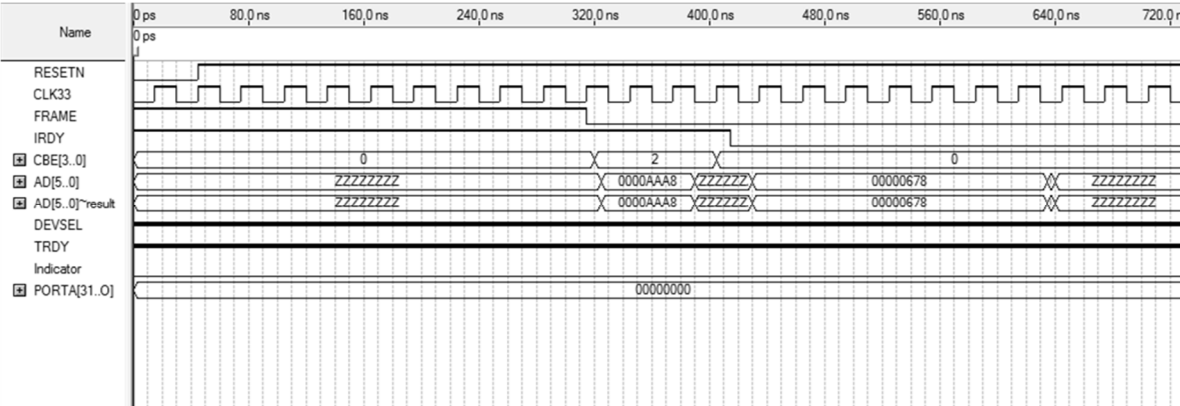


Figura 4.4: Simulación de protocolo PCI en software Quartus

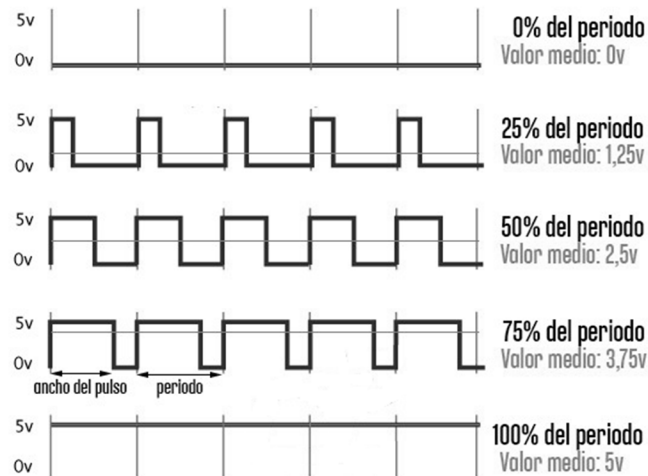
### 4.3 Modulo para generar PWM.

El PWM o modulación por ancho de pulsos es una técnica utilizada para regular la velocidad de giro de los motores eléctricos de inducción o asíncronos. Mantiene el par motor constante y no supone un desaprovechamiento de la energía eléctrica. Se utiliza tanto en corriente continua como en alterna, como su nombre lo indica, al controlar: un momento alto (encendido o alimentado) y un momento bajo (apagado o desconectado).

Se eligió esta técnica de regulación de velocidad ya que otros modifican la tensión eléctrica, con lo que disminuye el par motor; o interponen una resistencia eléctrica, con lo que se pierde energía en forma de calor en esta resistencia.

La Regulación por Ancho de Pulso de un motor de DC está basada en el hecho de que si se recorta la DC de alimentación en forma de una onda cuadrada, la

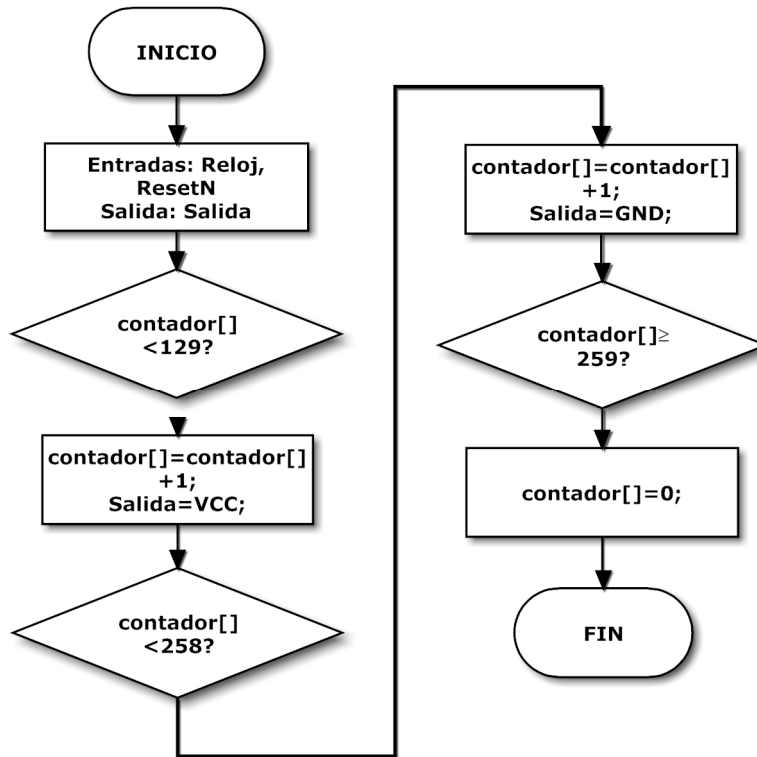
energía que recibe el motor disminuirá de manera proporcional a la relación entre la parte alta (habilita corriente) y baja (cero corriente) del ciclo de la onda cuadrada. Controlando esta relación se logra variar la velocidad del motor de una manera bastante aceptable (Figura 4.5).



**Figura 4.5:** Aplicación de la modulación por ancho de pulso

Para utilizar la técnica de control de velocidad mediante, modulación de ancho de pulso se necesita trabajar con ciertas frecuencias, ya que cada motor trabaja con ciertos tipos de frecuencias, por eso se realizó la caracterización del motor, explicada en un capítulo anterior, en donde la frecuencia encontrada fue de 500 Hz, para realizar el control del motor con el FPGA necesitamos generar esta frecuencia, ya que nuestro FPGA trabaja con un reloj interno de 33 MHz, necesitamos realizar un retraso en las salida de las señales para poder lograr la frecuencia deseada.

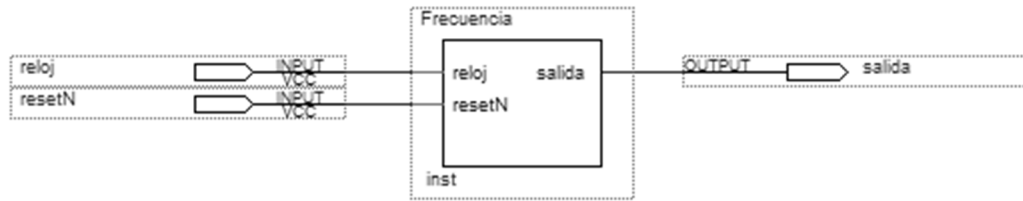
En el diagrama de flujo de la Figura 4.6 se muestra la lógica para generar la frecuencia de 500 Hz.



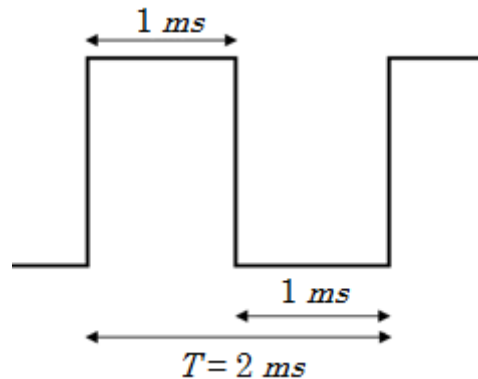
**Figura 4.6:** Diagrama de flujo para generar frecuencia de 500 Hz a partir del reloj del FPGA.

Para generar el PWM se requiere la frecuencia de trabajo del motor en este caso es de 500 Hz, para obtener dicha frecuencia se hace uso del reloj que trae el FPGA que es de 33MHz, necesitamos realizar el retraso de la salida de la señal, para que se obtenga la frecuencia requerida, esto se realiza generando un contador de 258 eventos, esta última cantidad corresponde al periodo de la frecuencia de 500 Hz con respecto a la frecuencia de reloj del FPGA.

Para poder saber el periodo para la generación de 500 Hz, y como se trabaja con 255 bits, esto nos indica que nuestra frecuencia debe de ser 255 veces más rápida, esto nos genera un valor de 127.5 KHz esta frecuencia es la que se requiere, obtenemos el periodo de esta frecuencia y nos da un valor de 7.843µs, como ya se conoce el periodo con el que se va a trabajar, ahora utilizamos el valor del reloj del FPGA que para este caso es de 33 MHz, obtenemos el periodo de esta frecuencia que es de 30.30 ns y la dividimos por el periodo con el que se va a trabajar y este nos da el valor de 258 y este el valor que tomara nuestro contador para generar la frecuencia de 500 Hz, se realiza la programación en Quartus y generamos el bloque de la frecuencia (Figura 4.7).



**Figura 4.7:** Bloque para generar la frecuencia de 500 Hz.



**Figura 4.8:** Generación de una frecuencia de 500 Hz.

#### 4.3.1 Cálculos para el contador de 500 Hz.

$$T = \frac{1}{500\text{Hz}} = 2\text{ ms} \quad (4.1)$$

$$F = (500\text{ Hz})(255) = 127.5\text{ KHz} \quad (4.2)$$

$$T = \frac{1}{127.5\text{ KHz}} = 7.843\mu\text{s} \quad (4.3)$$

$$\frac{7.843\mu\text{s}}{30.3\text{ns}} \approx 258 \text{ cuenta maxima} \quad (4.4)$$

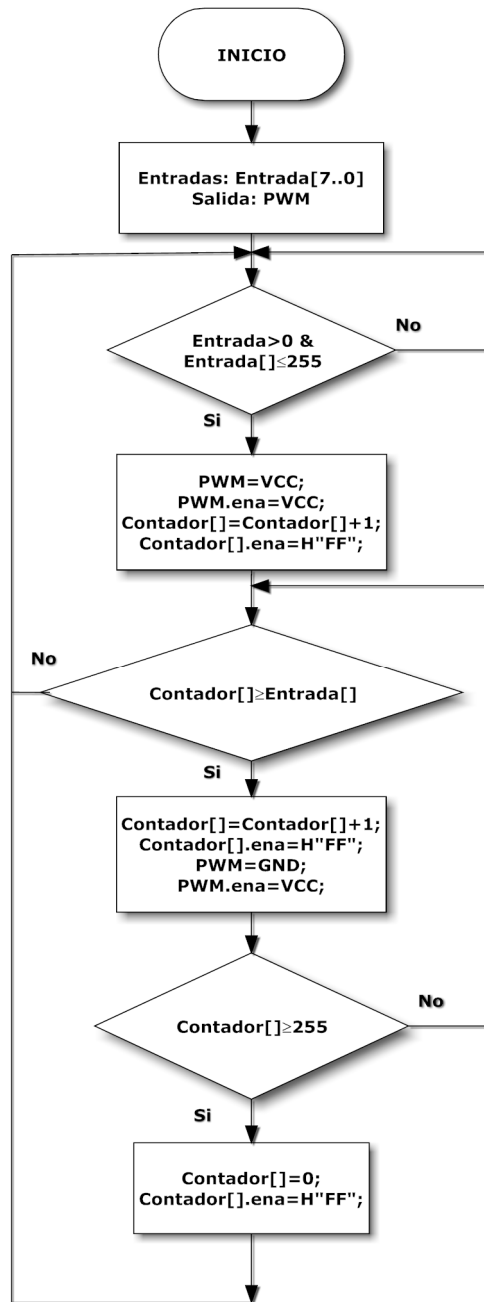
Una vez obtenida nuestra frecuencia de 500 Hz (Figura 4.8) podemos implementar el PWM, el bloque del PWM cuenta con una entrada que es nuestro dato y una salida que es la señal del PWM generado, en nuestro diagrama de flujo se puede observar que se necesita checar el dato que hay en la entrada si el dato es cero, el PWM tiene que estar en el valor en bajo y el contador se debe de inicializar en 1 esto es para evitar que cuando el contador este en cero este en valor en alto una vez que se ha inicializado, se hace una comparación entre la entrada y el contador si este contador es menor que la entrada nuestro PWM debe estar en valor en alto y nuestro contador debe de empezar a contar, si no se hace otra comparación si el contador es mayor a la entrada el PWM debe de estar

en valor en alto y nuestro contador debe de seguir contando, para cualquier otro estado el PWM debe de estar en bajo. El bloque de programación del PWM queda como se muestra en la Figura 4.9



**Figura 4.9:** Bloque del PWM en Quartus.

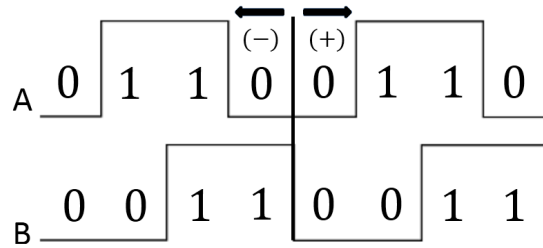
Usando el diagrama de flujo de la Figura 4.10, se describe la lógica para programar el PWM, se requiere conocer cuántas entradas y salidas debe tener el programa, el bloque del PWM únicamente tiene una entrada donde se le dará el dato, esta entrada siempre estará en alto y ahí se le dará el valor que se requiere como es una entrada de 0 a 255 bits, si se requiere que el ciclo de trabajo sea del 50%, a la entrada del bloque PWM se le proporcionara el dato de 128 y si se requiere un 25% se le proporciona 64 y así sucesivamente para generar el valor entregado por el PWM.



**Figura 4.10:** Diagrama de flujo para generar el PWM.

#### 4.4 Metodología de lectura para un encoder incremental

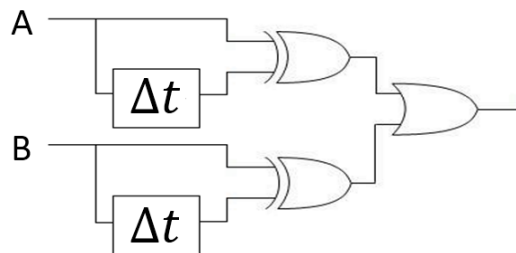
Con la siguiente metodología podemos nosotros saber la posición de un motor de manera incremental, ya que siempre se tiene un punto de referencia.



**Figura 4.11:** Combinaciones lógicas de las señales A y B.

La manera de lograr esto es leyendo las señales A y B que nos entrega el encoder de manera digital, como lo es este caso e ir viendo si incrementa o decrementa su valor tomando como referencia el primer valor muestreado, y con respecto a este valor cualquier cambio que haga el encoder nos genera un cambio en las combinaciones lógicas de nuestras señales y dependiendo como sea este cambio, sabremos si el giro del motor es en sentido horario o anti horario, las señales generadas por el encoder se muestran en la Figura 4.11.

De la Figura 4.11 observamos que las señales generadas por el encoder contienen toda la información que necesitamos para saber la posición. Como ejemplo podemos ver que si nos encontramos en la combinaciones de las señales A=0 y B=0, y tenemos un cambio de A=1 y B=0, tendremos un giro o un cambio en sentido horario, caso contrario si tenemos un cambio A=0 y B=1, tendremos un giro o un cambio en sentido anti horario, en caso que no haya cambio en las señales quiere decir que no ha ocurrido nada [19].

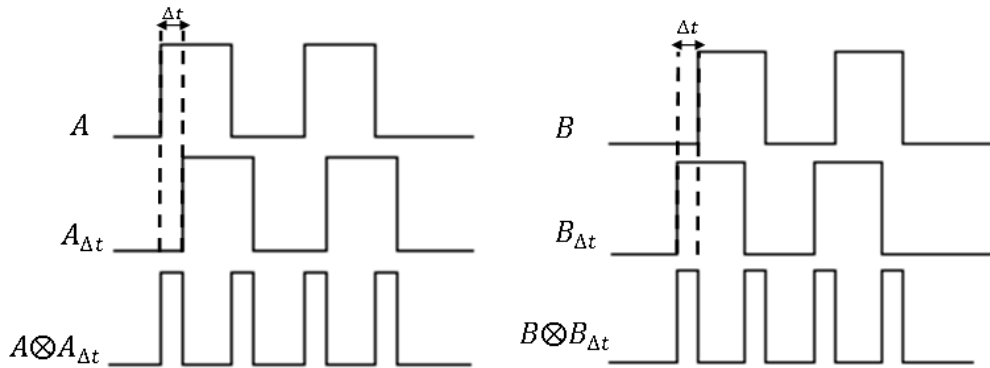


**Figura 4.12:** Circuito de muestreo de señales.

Uno de los problemas que aparecen, es el de muestrear correctamente las señales que son generadas por el encoder, para ello es necesario generar una señal de reloj que esté basado en las misma señales y para el firmware se implementa el circuito de la Figura 4.12.

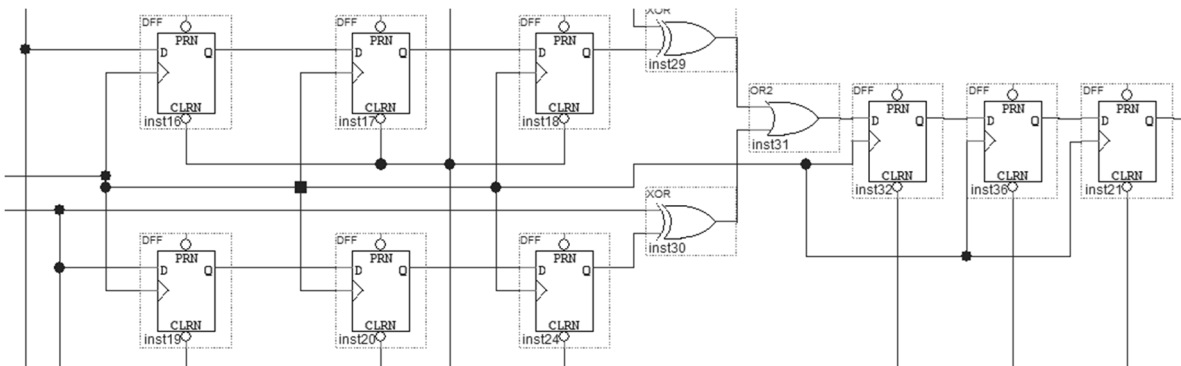
La creación de este reloj en base a las señales nos asegura que no se tendrá ninguna pérdida de muestra que sea generada por nuestros encoders, a

continuación se explicara de manera más detallada como se genera este reloj [19].



**Figura 4.13:** Diagrama de tiempos de las señales A y B.

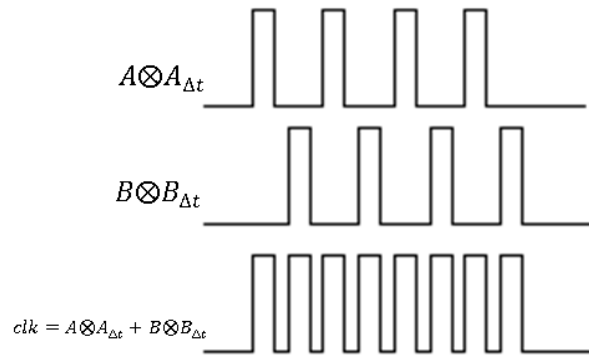
Las señales originales de nuestros encoders están desfasadas  $90^\circ$  una de la otra (Figura 4.13), para poder generar ese reloj que necesitamos para no perder ningún tipo de información en el momento que se está ejecutando el firmware, por medio de un arreglo de tres flip-flops tipo "D" generamos un retraso de la señal original, el retraso que necesitamos generar es de un octavo de la señal original y con este arreglo lo generamos, ya sea la señal A o la señal B, el arreglo de los flip-flops se puede observar en la siguiente Figura 4.14.



**Figura 4.14:** Retraso de las señales (A o B).

Una vez realizado el retraso de la señal como se tenía previsto, realizamos la operación XOR entre nuestra señal original (señal A o B) y nuestra señal retrasada ( $A_{\Delta t}$  o  $B_{\Delta t}$ ), el resultado de esta operación nos genera señales cuadradas donde las señales (A y  $A_{\Delta t}$  o B y  $B_{\Delta t}$ ) sean diferentes, una vez obtenidos nuestros dos ciclos de reloj A y B generamos una OR entre las dos para

obtener el ciclo completo de reloj como se puede observar en la siguiente Figura 4.15.



**Figura 4.15:** Reloj generado por las señales A y B.

*Lectura de los encoders.* La señal de los encoders es un tren de pulsos que no pueden ser tomados por el controlador sin antes ser organizados, el encoder incremental proporciona dos formas de ondas cuadradas y desfasadas entre sí en  $90^\circ$  a los cuales llamamos canal A y canal B, así podemos conocer la velocidad de rotación y el sentido. Podemos leer los flancos ya sea de subida o de bajada de los canales, al leer los dos canales duplicamos la resolución por 2 pues contamos el doble de pulsos, también podemos leer tanto los flancos de subida como los de bajada de los canales para multiplicar la resolución por 4, este es el método que usaremos y como tenemos un encoder de 2500 pulsos por revolución tenemos una resolución de 10000.

Si detectamos un flanco de subida o de bajada en A, después verificamos el estado de B, y dependiendo de este se procederá a sumar o restar pulsos a la derecha o a la izquierda según sea el caso. En el FPGA implementaremos esto con una máquina de estados, después mandamos estos datos a la computadora donde el software de control se encargará de calcular la posición con uso de contadores.

A continuación se definirán términos que nos permiten elaborar el diseño de nuestra máquina de estados, la respuesta máxima en frecuencia a la cual el encoder responde eléctricamente está relacionada con la velocidad de giro del eje del encoder y con el número de impulsos del mismo.

$$\text{Frecuencia (Hz)} = \frac{\text{N}^\circ \text{ vueltas del eje por minuto}}{60} \text{ N}^\circ \text{ impulsos del encoder} \quad (4.5)$$

Si se tiene un encoder de 2500 pulsos por revolución y lo acoplamos a un motor que puede ir hasta 3600 rpm entonces tenemos que la máxima respuesta en frecuencia es

$$\text{Frecuencia (Hz)} = \frac{3600 \text{ rpm}}{60} \cdot 2500 \text{ pulsos por revolución} \quad (4.6)$$

$$\text{Frecuencia (Hz)} = 150 \text{ KHz} \quad (4.7)$$

*Resolución.* Es la medida de cambio más pequeño de entrada que el encoder puede detectar, se mide por el número de pulsos de la salida por cada revolución del eje, cuantas más ranuras tenga mayor será la resolución.

$$\text{Resolución} = \frac{360^\circ}{\text{N}^\circ \text{ impulsos del encoder}} \quad (4.8)$$

$$\text{Resolución} = \frac{360^\circ}{2500 \text{ pulsos por revolución}} = 0.144^\circ \quad (4.9)$$

*Reloj para la máquina de estados.* Se creó un bloque que genera una frecuencia mucho menor a la del reloj del FPGA, el reloj necesario para que la máquina de estados pueda contar correctamente los pulsos del encoder, viene determinado por la siguiente operación lógica,  $\text{CLK}=(A \text{ XOR } A')\text{OR}(B \text{ XOR } B')$ , A' y B' son las señales A y B desfasadas un tiempo  $\Delta t$  de ellas mismas, para calcular este  $\Delta t$  sabemos que la frecuencia máxima a la que puede responder el encoder es de 150 KHz aproximadamente con esto podemos entonces determinar el periodo mínimo que tendrá la señal

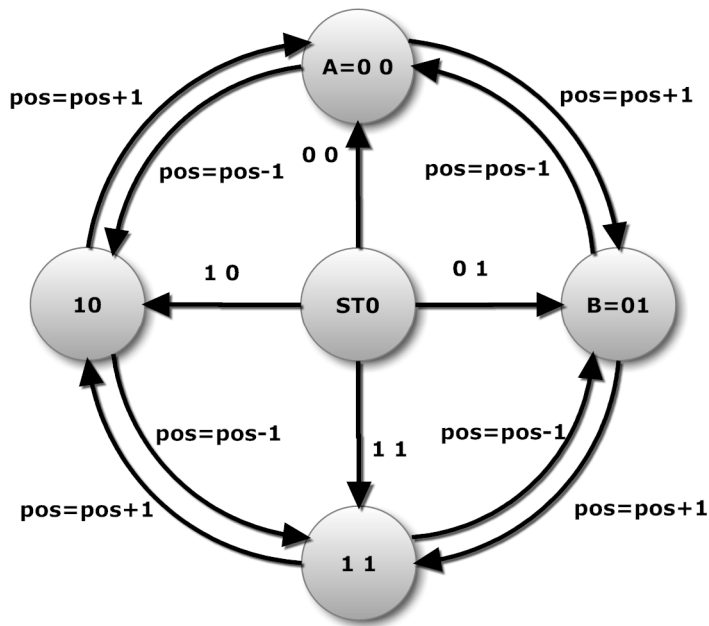
$$\Delta t = \frac{T_{min}}{8} \quad (4.10)$$

$$T = \frac{1}{150 \text{ KHz}} = 6.66 \text{ us} \quad (4.11)$$

$$\Delta t = \frac{6.66 \text{ us}}{8} = 833.3 \text{ ns} \quad (4.12)$$

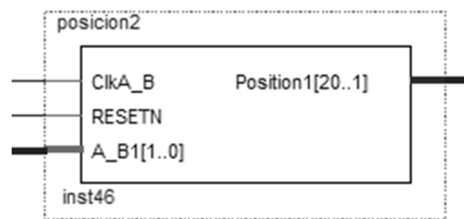
#### 4.4.1 Implementación del encoder.

Una vez que ya se ha visto la metodología para hacer un correcto muestreo de las señales del encoder y se ha entendido todo el proceso por el cual deben de pasar las señales A y B, se lleva a nivel programación, se diseña el diagrama de flujo en base a lo visto con anterioridad ver figura 4.16 [20].



**Figura 4.16:** Diagrama de Flujo de la lectura del encoder.

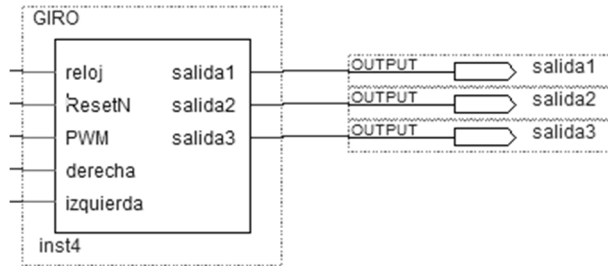
Como resultado del diagrama de flujo de la lectura del encoder generamos nuestro código en Quartus y obtenemos el siguiente bloque de firmware, con este bloque ya se puede hacer el conteo de las señales del encoder con toda precisión sin perder cuentas, garantizando el máximo desempeño de nuestro sistema (Figura 4.17).



**Figura 4.17:** Bloque de conteo de las señales del encoder.

## 4.5 Módulo de control de giro.

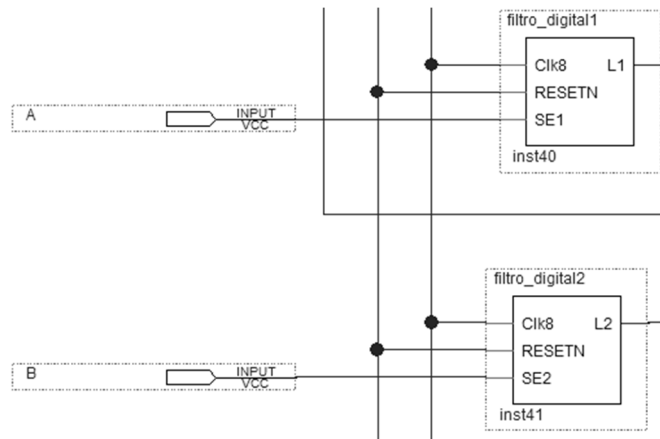
Para poder realizar el control del giro de los actuadores es necesario realizar un bloque de giro en el firmware, este bloque cuenta con dos entradas en estas entradas se le manda el valor que necesita para poder realizar el giro de nuestro actuador (Figura 4.18).



**Figura 4.18:** Bloque de control de giro.

#### 4.6 Módulo de filtrado.

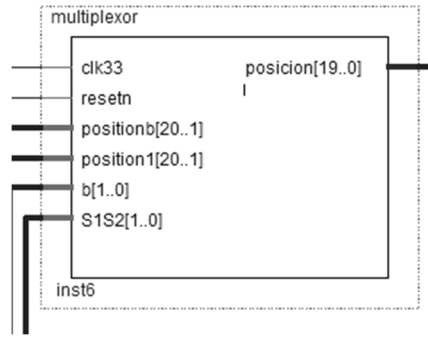
Impide o permite el paso de una cierta gama de frecuencias, este es digital y tiene las ventajas de que es programable, son estables con respecto al tiempo y la temperatura y tienen alta inmunidad al ruido, este código se desarrolló anteriormente, las señales del encoder entran a este filtro, Figura 4.19, y la salida es la señal que se encuentra lista para los demás bloques.



**Figura 4.19:** Módulo de filtro digital.

#### 4.7 Módulo multiplexor.

También se realizó un bloque para poder elegir que encoder se necesita leer, ya sea el del motor o el del péndulo, este bloque tiene una habilitación donde desde LabVIEW se le manda a escribir los datos y así se elige el encoder que se desea leer (Figura 4.20).

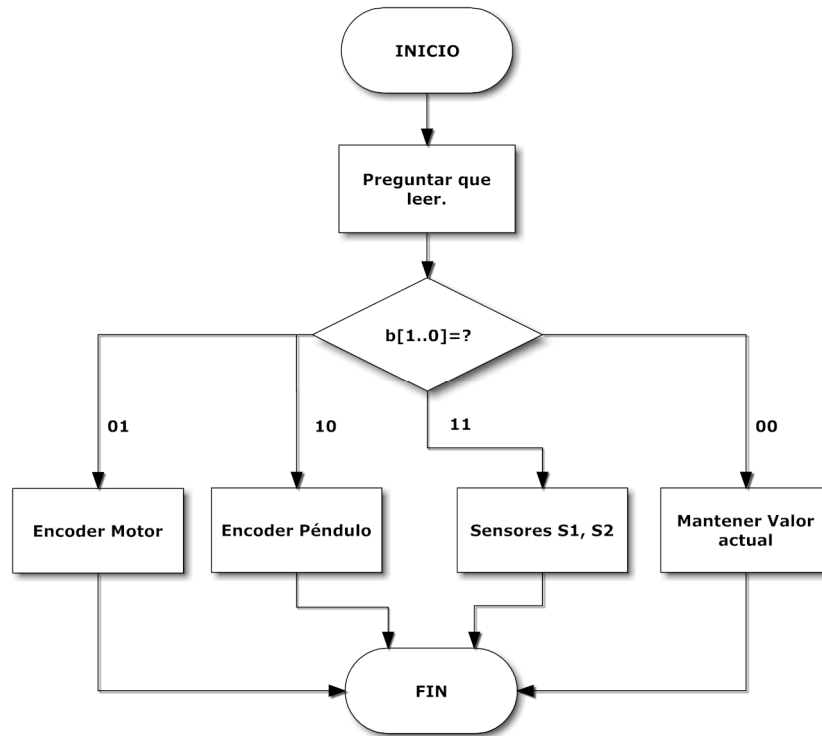


**Figura 4.20:** Bloque del multiplexor

Debido a que tenemos que leer dos encoders y sensores de fin de carrera y para enviar a la computadora los resultados de estos tres bloques se utiliza un multiplexor, esto se realiza de esta manera para tener una mayor resolución. Para asegurarnos de no confundir los valores de los dos encoders se tienen los siguientes métodos, en el software pide un dato enviado un código al FPGA cuando el multiplexor detecta el código deja pasar el dato deseado. En la Figura 4.21, tenemos el diagrama a bloques para el desarrollo del multiplexor.

**Tabla 4.1:** Tabla de verdad para los códigos de los datos de los encoders.

<b>b[1..0] selección de dato a leer.</b>	<b>posición[19..0] valor de salida del bloque multiplexor según el valor enviado en b[1..0]</b>
11	S1S2[1..0] valor de sensores de fin de carrera
10	posicion1[20..1] valor del encoder del péndulo.
01	posicionb[20..1] valor del encoder del motor
00	Mantener el valor



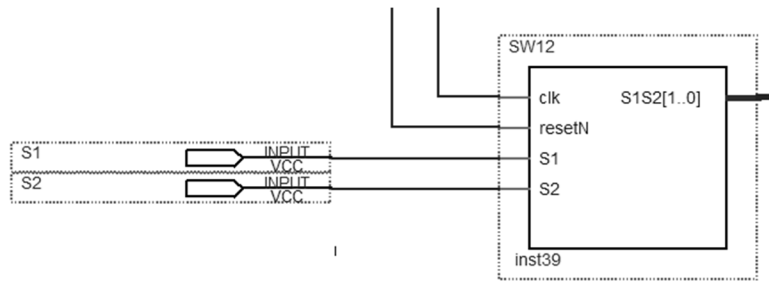
**Figura 4.21:** Diagrama a bloques del multiplexor.

#### 4.8 Módulo de Sensores de final de carrera.

Este detecta el cambio de estado de las entradas de los sensores de final de carrera y luego envía la información a la computadora para poder realizar el paro y el regreso al origen del carro, los bits seleccionados para este propósito se muestran en la Tabla 4.2 y el módulo en la Figura 4.22.

**Tabla 4.2:** Salidas de los sensores de final de carrera.

SWITCH	S1S2[1..0]	
	S1S2[1]	S1S2[0]
S1	0	1
S2	1	0



**Figura 4.22:** Bloque para los sensores de final de carrera.

En la Figura D.1 y D.2 (apéndice D) se muestra el diagrama completo de conexión eléctrica del Firmware.

### Conclusiones.

En este capítulo se desarrolla la parte del firmware que es fundamental para el manejo de los datos tanto de entrada como de salida, se muestran los módulos que implementan el firmware como el protocolo PCI, la metodología para la correcta lectura de los encoders y evitar la pérdida de datos, también se realizó la generación de un PWM con una frecuencia determinada a partir de la frecuencia de reloj del PCI. La programación del firmware es flexible y se pueden realizar cambios dentro de este sin ningún problema, para poder corregir o mejorar el funcionamiento.

## Capítulo 5.

### Diseño del Software.

En este capítulo se describe el desarrollo de la herramienta de comunicación entre el usuario y la parte electrónica, para ello se hace uso del software LABVIEW, el cual hace la función de interfaz, en conjunto con la computadora en donde se encuentra instalada la tarjeta PCI y el software LabVIEW.

#### 5.1 Software de interfaz.

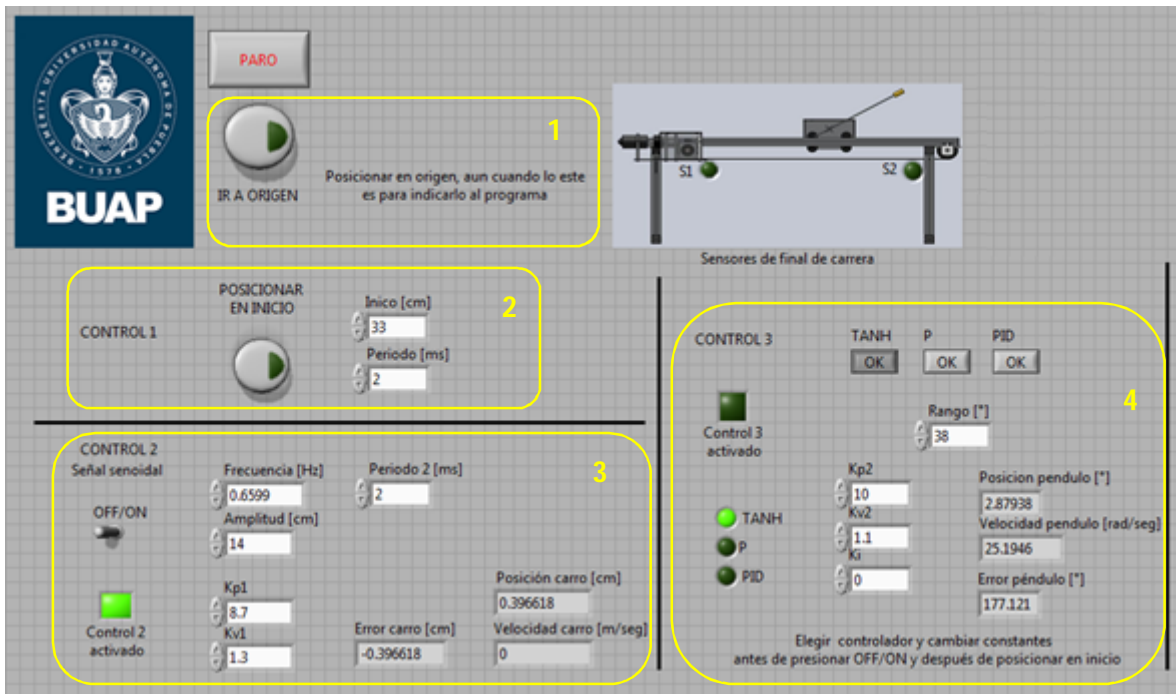
LabVIEW es usado principalmente por ingenieros y científicos para realizar tareas como: adquisición de datos, control de instrumentos, automatización industrial, control y supervisión de procesos y robótica, entre otras. Su principal característica es la facilidad de uso, pues los programas no se escriben sino se dibujan. En LabVIEW, un programa se divide en "panel frontal" y "diagrama de bloques". El panel Frontal, es la interfaz de usuario, en él se definen los controles e indicadores; en el diagrama a bloques, se define su funcionalidad, aquí es en donde se colocan y se interconectan los instrumentos virtuales.

##### 5.1.1 Pantalla de usuario

Una vez que se hace la adquisición de los datos en el FPGA se mandan a la computadora a través del puerto PCI, el software es el encargado de recibir estos datos, interpretarlos y procesarlos de forma que emita las señales de control en respuesta a los sensores y a las órdenes del usuario, además de que sirve como puente entre el usuario y el péndulo invertido LabVIEW tiene un entorno amigable tanto para programar como para usarlo, cuenta con herramientas gráficas ideales para crear una interfaz dinámica. Un programa en LabVIEW consta básicamente de un panel frontal y un diagrama de bloques, en el panel frontal, Figura 5.1, se diseña la interfaz con el usuario, en él se ven los datos y se controlan, en el diagrama de bloques esta la estructura del programa, su función y algoritmo en lenguaje G, es decir, gráfico. Otra de las ventajas de LabVIEW es que tiene la capacidad de trabajar en tiempo real, ya que el FPGA trabaja a una velocidad alta la sincronización puede ser difícil y si no se lleva a cabo correctamente puede ocasionar errores, aunque el desarrollo de sistemas en tiempo real requiere típicamente conocimientos de programación a nivel de

registros y la instalación de hardware a la medida con LabVIEW puede llevarse a cabo de una manera fácil usando estructuras ya programadas y que solo se tienen que configurar a nuestras necesidades, hay que mencionar que tiempo real es un término usado en la industria pero su definición es ambigua, la mayoría de los ingenieros está de acuerdo con que tiempo real significa con retrasos aceptables, el término se usa para definir un sistema que debe ejecutarse sin falla y cumplir con los requerimientos en todo momento.

La estructura Timed Loop de LabVIEW [21] es la que se utiliza para la interfaz, puede realizar tareas de tiempo-critico, retroalimentarse en cada lazo y tiene un reloj de 1 MHz, para la ejecución del programa usamos un periodo de 2 ms, aunque puede ajustarse, esta es una estructura temporizada es decir es una función ciclada que se ejecuta cuando ocurre un evento en la fuente de temporización. A continuación se muestra la pantalla de usuario de nuestro programa para controlar al péndulo invertido.



**Figura 5.1:** Pantalla de usuario.

En la Figura 5.1 se observa el panel frontal de LabVIEW, desde aquí se modifica los parámetros para el control del péndulo invertido y se visualizan los resultados y las alarmas del sistema. El programa se divide en cuatro etapas:

### 5.1.2 Posicionamiento en el origen del carro.

En la Figura 5.1 vemos que en la parte de Control 1, es donde se lleva al carro a una posición inicial para comenzar el control 2, esta se lleva a cabo en un Timed Loop.

Control 2, en este control se busca que las varillas del péndulo se eleven lo suficiente para que entre el control 3, esto se hace suministrando energía al motor en forma de una onda senoidal, el carro comienza a describir esta onda es decir se mueve de un lado a otro, este control se lleva a cabo en una segunda estructura Timed Loop.

Control 3, se encuentra dentro del segundo Timed Loop, una vez que las varillas alcanzaron una posición lo suficientemente alta el control 2 conmuta al control 3, donde se implementa ya sea un controlador proporcional, proporcional integral derivativo o tangente hiperbólica para llevar las varillas a la posición deseada.

A continuación se describen las diversas partes que conforman el panel frontal comenzando por los controles.

- *Botón paro*, este detiene la ejecución de todo el programa y limpia todas las variables.
- *Botón origen*, lleva al carro a su posición de origen es decir la posición extrema del riel cerca del motor.
- *Botón posicionamiento en el inicio*, con este se manda al carro a la posición desde la cual comenzara a moverse de un lado a otro.
- *Palanca ON/OFF*, con esta una vez que el carro se encuentra en la posición de inicio se le indica que puede comenzar a moverse.
- *Botones TANH, P y PID*, con estos le indicamos al programa que tipo de controlador usar.

Los parámetros son las variables que se pueden modificar en el programa.

- *Inicio*, aquí se indica en cm, la posición en la que se quiere colocar al carro para comenzar a moverse de un lado a otro.
- *Periodo y Periodo 2*, son los tiempos en ms para la duración de las dos estructuras Timed Loop.
- *Frecuencia*, es a la cual se pondrá la onda senoidal que controla el movimiento del carro.
- *Amplitud*, de la onda senoidal será también la distancia que recorrerá el carro de un lado a otro.
- *Kp1, kv1*, son las ganancias del segundo controlador es decir de la función senoidal.

- *Rango*, este es para el control tres y determina en donde se cambiara del control senoidal al control p, pid o tanh, se encuentra en grados y son por ambos lados con respecto de la posición deseada.
- *Kp2, kv2 y ki*, son los ganancias para el tercer controlador.

Los resultados se visualizan mientras se ejecuta el programa además de que se guardan en un archivo que después se puede graficar para su mejor análisis.

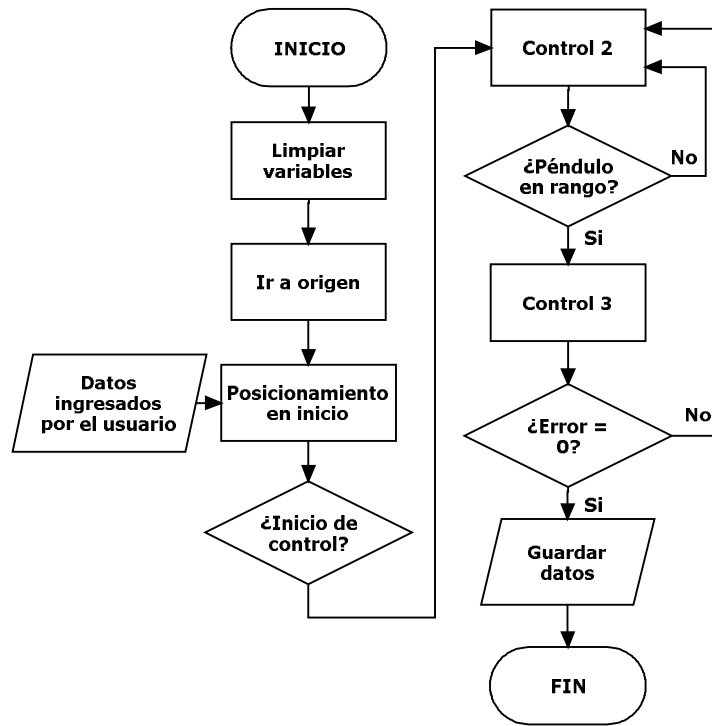
- *Posición carro*, se encuentra en centímetros y es la distancia que ha recorrido desde su posición inicial.
- *Velocidad carro*, en metros sobre segundo es la velocidad instantánea del carro.
- *Error carro*, es el error del controlador 2.
- *Posición péndulo*, en grados es la posición en la que se encuentran las varillas.
- *Velocidad péndulo*, en radianes sobre segundo es la velocidad que llevan las varillas.
- *Error péndulo*, es el error del controlador 3 ya sea el p, pid o tanh.

Las alarmas son indicadores tipo led que nos avisan de algún evento.

- *S1 y S2*, son los indicadores de que el carro alcanzó alguna de sus posiciones extremas y alguno de los switch se activó, el programa se para cuando esto sucede.
- *Control 2*, se ilumina cuando el control dos se activa.
- *Control 3*, cuando el control 2 conmuta al control 3 el indicador control 2 se apaga y se enciende el indicador control 3.
- *Indicadores p, pid y tanh*, se enciende alguno de estos tres cuando control 3 se activa.

## 5.2 Software del sistema

El objetivo del control es encontrar la fuerza que se ha de aplicar al carro para que el péndulo llegue a su posición vertical de  $180^\circ$ , para esto es necesario recibir, almacenar y procesar los datos provenientes de la tarjeta de adquisición, el control implementado en el software responderá con las señales apropiadas para este objetivo. A continuación se describe el software para el control del péndulo invertido.



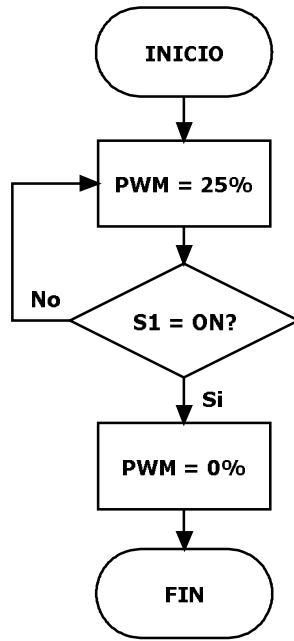
**Figura 5.2:** Diagrama a bloques del software del péndulo invertido.

El diagrama a bloques del programa, Figura 5.2, se divide en cuatro etapas que se encuentran dentro de un While Loop que ejecutará el programa a menos que se presione el botón de paro. Las etapas se encuentran implementadas en una estructura Stacked Sequence Structure, que son subdiagramas o frames como se llaman en LabVIEW, que se ejecutan secuencialmente uno después de otro según el número que tengan.

### 5.2.1 Posicionamiento en el origen

Reinicio de variables, aquí se limpian todas las variables que guardaron los valores de los diversos parámetros que se usaron y se mostraron en el panel de control mientras se ejecutaba el programa, este es un tipo de reset para que el programa comience con condiciones iniciales la próxima vez que se ejecute.

Posicionamiento en el origen, Figura 5.3, si esta función se activa desde el panel de control se realiza lo siguiente. Ya que la posición de origen se encuentra en el extremo donde está un sensor de final de carrera, es decir S1, el programa manda al FPGA un comando que mueve el carro con una velocidad baja hacia esa dirección y una vez que se detecta que S1 se encuentra activado se manda un comando al FPGA para que el carro se detenga.



**Figura 5.3:** Diagrama de flujo del posicionamiento en el origen.

El posicionamiento en el origen se encuentra dentro de un Case Structure, para saber si es que el carro se encuentra ya en la posición de origen se revisa el dato de entrada de la dirección AAA8, esta es la dirección del puerto PCI donde se encuentra la tarjeta de adquisición de datos, la Tabla 5.1, muestra los datos que se reciben desde el FPGA donde para los microswitch tenemos que S1 es el bit 0 y S2 el bit 1.

**Tabla 5.1:** Datos de entrada para el FPGA.

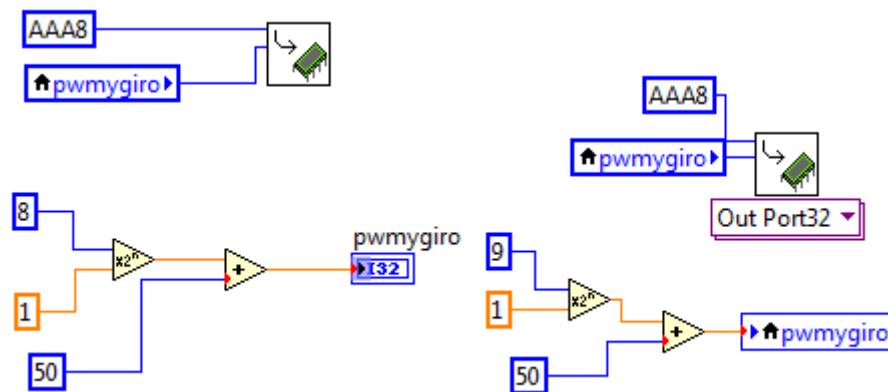
<b>Entrada</b>	<b>Información contenida</b>	<b>Multiplexión de datos por medio de b[1..0] y salida en posicion[19..0]</b>
positionb[20..1]	Encoder del motor	01
position1[20..1]	Encoder del péndulo	10
S1S2[1..0]	Microswitch S1 y S2	11

La variable *pwmgiro* se encarga de guardar la información que se escribirá en el puerto, tanto de la velocidad con la que queremos que se mueva el carro como de la dirección según la Tabla 5.2.

**Tabla 5.2:** Descripción de datos para el bloque PWM y selección de datos a leer.

Bits de salida	Información contenida
porta[7..0]	Valor para el PWM
port[8]	Giro a la derecha
port[9]	Giro a la izquierda
port[11..10]	Selección de dato a leer b[1..0]
Bits 31 – 12	No se usan

El valor para los bits del sentido de giro está determinado por port[9..8] puede ser 0 y 1, para activar el bit sumo a la variable pwmygiro el valor de 50 o cualquier otro valor dentro del rango ya que es el peso que tienen estos bits y para desactivarlo, ver Figura 5.4.



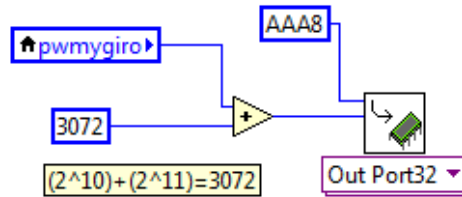
**Figura 5.4:** Escritura en los bits 8 y 9 para indicar giro de motor a la derecha y a la izquierda respectivamente.

Un 0 indica que el motor girará a la izquierda, es decir en dirección a S1 y un 1 indica que girará al derecha, es decir hacia S2.

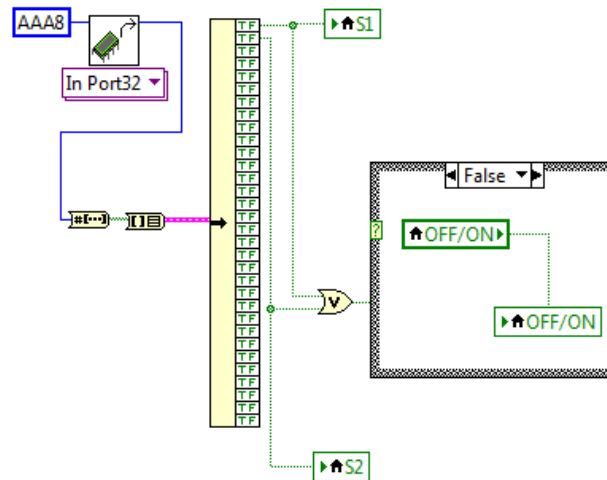
El bloque PWM en el FPGA funciona con un contador desde 0 hasta 258, podemos escribir en los primeros 8 bits desde el 0 hasta el 255, donde mientras mayor sea el número que se escribe mayor será la velocidad que tenga el motor.

En todas las etapas se revisa el estado de los sensores S1 y S2, para que en cualquier momento que se active uno de estos se detenga el programa.

Primero se manda a escribir un H'3' a b[1..0] que son los bits 10 y 11 (Figura 5.5) para que mande a leer el valor de S1 y S2 almacenados en el bloque multiplexor, Figura 5.6.



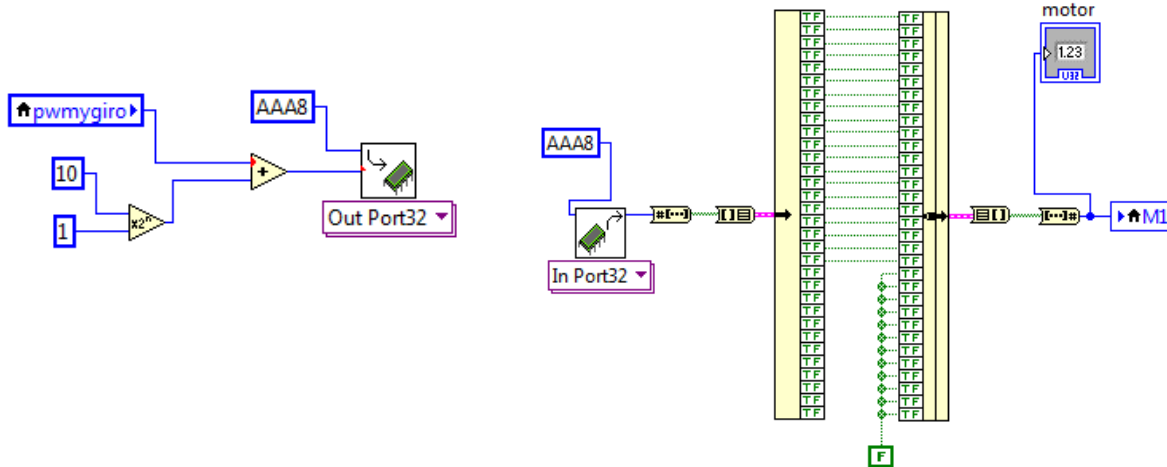
**Figura 5.5:** Escritura de H'3' al PCI en los bits 10 y 11 para leer los valores de S1 y S2.



**Figura 5.6:** Lectura de PCI de los valores de S1 y S2.

### 5.2.2 Posición de inicio del carro.

Esta se encuentra dentro del primer Timed Loop y se encarga de llevar a cabo el control 1, llevar al carro a su posición de inicio para poder comenzar las pruebas. Primero se lleva a cabo la adquisición del FPGA mediante una sincronización entre la PC y el FPGA para asegurarnos de que estemos recibiendo el dato que queremos, primero se pide al FPGA el valor del encoder del motor activando la bandera en el multiplexor, según la tabla de datos de salida estas banderas se encuentran en los bits 10 y 11 donde el bit 10 es para el encoder del péndulo y el bit 11 para el encoder del motor, como queremos el valor del motor activamos este bit sumando a la variable *pwmgiro* el peso de este es decir 1024, si quisiéramos en valor del péndulo activamos el bit 11 sumando 2048. Además de almacenar el valor leído del encoder en la variable M1, Figura 5.7.



**Figura 5.7:** Petición de lectura de encoder del motor por medio del bit 10 y recepción del valor del encoder del motor.

El valor se escribe junto con el que se encuentre almacenado en la variable PWM, esto para que mientras se pidan los valores de los encoders no se deje de enviar la información que necesita el FPGA para moverse. Después de que se pidió el dato se revisan en el puerto de entrada los valores de la posición de los encoders.

El mismo procedimiento se realiza pero ahora para leer el encoder del péndulo, ahora escribiendo en el bit 11, pero ahora se almacena el valor leído en otra variable M2.

Para determinar la posición del carro en base a los pulsos del encoder se realiza el siguiente esquema de la Figura 5.8, Se calcula una delta con respecto al desplazamiento de una revolución de la polea de la banda del carro y una revolución del encoder acoplado a dicha polea quedando de la siguiente manera:

Primero se calcula el diámetro de la polea que mueve el carro, dicho diámetro es el desplazamiento del carro sobre el riel.

$$diámetro = 2\pi * radio \quad (5.1)$$

$$diámetro = 2\pi(4.456 \text{ cm}) = 27.99 \text{ cm} \quad (5.2)$$

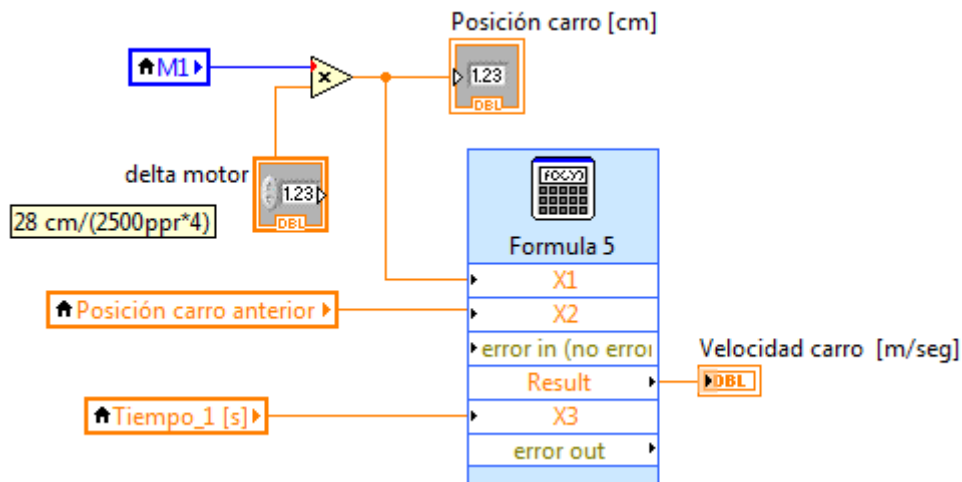
$$\Delta x = \frac{\text{diámetro de la polea}}{\text{pulsos por revolución del encoder} * 4} \quad (5.3)$$

$$\Delta x = \frac{28 \text{ cm}}{(2500\text{ppr})(4)} = 0.0028 \quad (5.4)$$

Por ejemplo, si el encoder avanza media vuelta, es decir, obtiene una cuenta de 5000 pulsos, el carro avanza solo 14 cm., ya que al multiplicar 5000 x 0.0028 obtenemos esa cantidad.

Si queremos que el carro recorra toda la longitud permitida por el riel, que es de aproximadamente 66 cm, necesitamos que el encoder sea capaz de contar aproximadamente hasta 23572.

De esta cuenta podemos deducir que necesitamos un contador de 15 bits, pero el sistema se desarrolló para un contador de 20 bits.



**Figura 5.8:** Cálculo de la posición y la velocidad del carro.

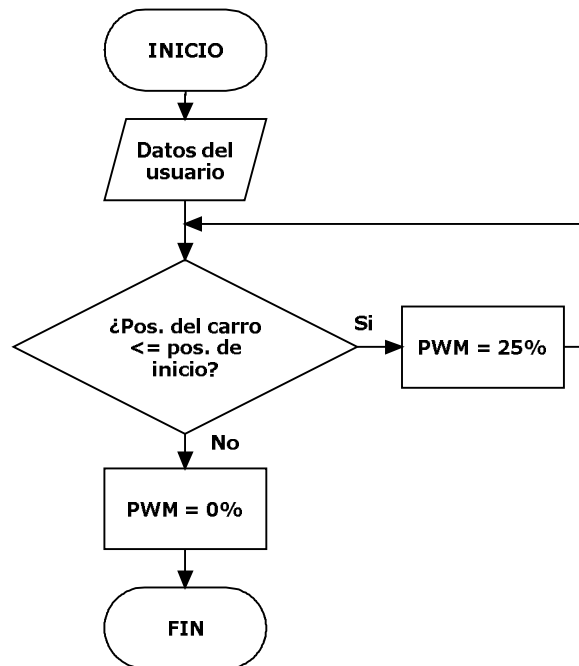
Para el cálculo de la posición y la velocidad del motor se realizan las siguientes operaciones:

$$\text{Posición} = \text{Lectura de encoder} * \Delta x \quad (5.5)$$

Se multiplica el valor leído del encoder por la  $\Delta x$  de esta manera sabemos la distancia que recorre el carro por cada vuelta del motor, para obtener la velocidad del carro es necesario guardar la posición del carro actual y la anterior, la fórmula es la siguiente

$$\text{velocidad} = \left( \frac{\text{posición actual} - \text{posición anterior}}{\text{tiempo de muestreo}} \right) \quad (5.6)$$

Donde el tiempo de muestreo es de 0.002 seg., que es el tiempo en segundos entre el cual se realiza el cambio de posición, es decir el tiempo de muestreo del Timed Loop. Una vez que se cuentan con todos los valores necesarios se puede llevar a cabo entonces el primer control, en este control se mueve el carro hacia la derecha hasta que haya alcanzado la posición introducida en el panel de control, esto lo hacemos comparando la posición del carro con la posición a la que queremos llegar mientras no sea la misma el carro se seguirá acercando lentamente.



**Figura 5.9:** Diagrama de flujo del control 1 del péndulo invertido.

Mientras no se ha llegado a la posición deseada, se manda al FPGA a activar el bit 8, que significa sentido de giro a la derecha y PWM de 50 cuando se ha llegado a la posición deseada se detiene el carro enviando un 0, se limpian las variables usadas para que cuando se inicie el control 2 y además se indica con una bandera que el carro ha llegado a la posición inicial. En la Figura 5.9 se observa el diagrama a bloques de flujo de control 1.

### 5.2.3 Control de levantamiento y estabilización del péndulo.

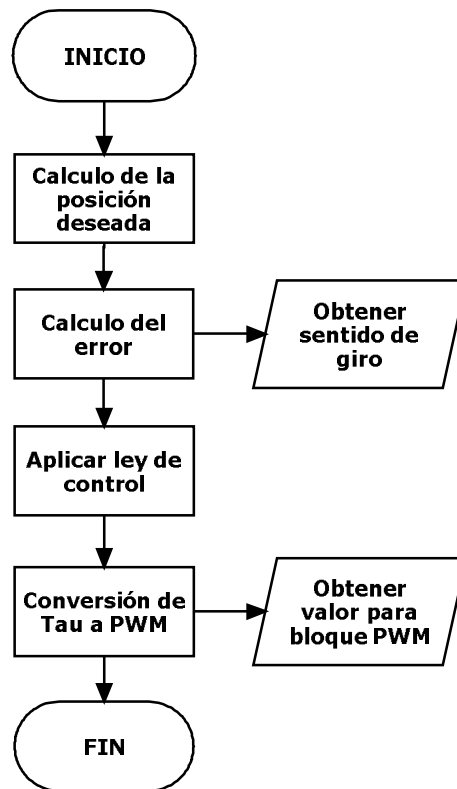
En esta se realizan los controles 2 y 3 que conmutan entre sí según la posición de las varillas del péndulo, estos controles se encuentran dentro del segundo Timed Loop. Primero se lleva a cabo la adquisición de los datos de los encoders, esto como se describió en la segunda etapa, se almacenan los datos para los valores actual, anterior y ante anterior tanto del motor como el péndulo, de la manera descrita también en la segunda etapa, luego se calculan las posiciones del encoder del motor como del péndulo, para el péndulo se hace de manera similar a la descrita para el motor pero tiene la siguiente resolución.

$$\Delta y = \frac{360^\circ}{2000ppr*4} = 0.045 \quad (5.7)$$

Se multiplica la velocidad por 0.0175, para obtenerla en radianes

$$velocidad\ del\ péndulo = \left( \frac{(posición\ actual - posición\ anterior)}{tiempo\ de\ muestreo} \times 0.0175 \right) \quad (5.8)$$

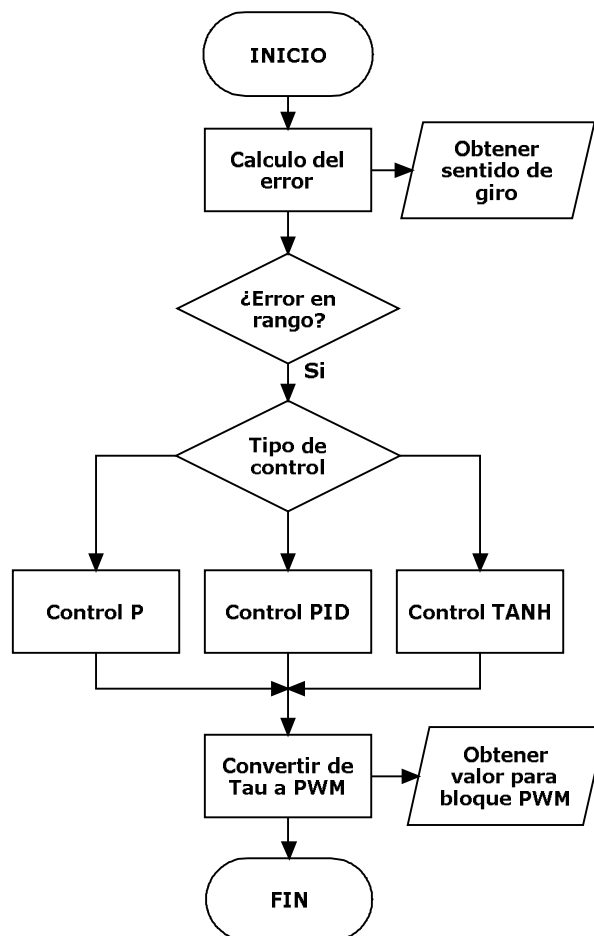
Para saber qué control es que tiene que actuar se toma en cuenta la posición de las varillas del péndulo y el rango introducido en el panel de control, este rango indica cuantos grados alrededor de la posición deseada actuara el control 3, primero se calcula el error, tanto de carro como del péndulo, una vez que los tenemos preguntamos si el error del péndulo es menor que el rango de acción del control 3, dependiendo de esta respuesta se efectuara el control 2, Figura 5.10, o el control 3, Figura 5.11.



**Figura 5.10:** Diagrama de flujo para el control 2 del péndulo invertido.

En este control se introduce una onda senoidal donde tanto la frecuencia como la amplitud se pueden controlar desde el panel de control, la salida de esta onda senoidal describe el movimiento del carro este se moverá de un lado a otro conforme la onda avance a lo largo del tiempo, para esto la salida de esta señal se convierte en la posición deseada para el carro, esta posición se introduce en un controlador *tanh*, los parámetros para el controlador se pueden manejar desde el panel de control, el sentido de giro se decide con el signo del error, de

esta manera podemos invertir el giro del motor según el carro este más lejos de la posición deseada por un extremo o por el otro. La salida del controlador es un valor en torque llamado Tau 1, este debe de convertirse en un valor PWM para nuestro bloque en el FPGA, lo dividimos entre 0.017275, dicho valor es obtenido de la pendiente de la gráfica del torque que produce el motor a una frecuencia de 500 Hz, para conseguir el valor en porcentaje luego haciendo una regla de tres, se multiplica por 1.53 para obtener solo el 60 % de torque del motor, con relación a los 255 bits del PWM.



**Figura 5.11:** Control 3 del péndulo invertido.

Dentro de este control existen tres controladores diferentes, un  $p$ , un  $pid$  y un  $tanh$ , solo uno de ellos se ejecuta a la vez, los parámetros para cada controlador pueden ser cambiados desde el panel de control. El control  $P$ , este es el más sencillo de los tres controladores, el bloque  $kp$  realiza multiplica el error del péndulo por la constante  $kp$  para sacar una acción de control proporcional. El

control *PID*, realiza la suma de los términos *kp*, *ki* y *kd*. Finalmente el controlador *TANH* se introduce dentro de un Formula node para así se meter directamente la ecuación de control. Al final el *Tau* resultante de los controladores recibe la misma transformación en su valor equivalente en PWM. El sentido de giro se decide igualmente mediante el signo del error.

Por último dentro de esta etapa se realiza el almacenamiento de los resultados del programa, esto guardando el valor de las variables dentro de un archivo que con uso de otro programa en LabVIEW se podrá graficar. Al finalizar el funcionamiento del programa se detiene el movimiento del carro y se limpia el valor escrito en el puerto.

## **Conclusiones**

En este capítulo se llevó a cabo la descripción de la interfaz de usuario, la comunicación de la tarjeta de adquisición con la computadora es una parte importante para evitar la pérdida de datos o la lectura incorrecta de estos, LabVIEW se sincronizo con la tarjeta de adquisición mediante el uso de banderas en la petición y recepción de los datos, con esto y con la estructura Timed Loop de LabVIEW se logró una sincronización que nos permitió adquirir datos con la mayor velocidad posible, también se explica la interfaz con el usuario la cual se diseñó de manera que fuera fácil de usar y ya que LabVIEW es un software con el que los estudiantes de la facultad están muy familiarizados aumenta la posibilidad de que cualquiera sea capaz de manejarlo fácilmente.

## Capítulo 6.

### Controlador para el sistema mecatrónico.

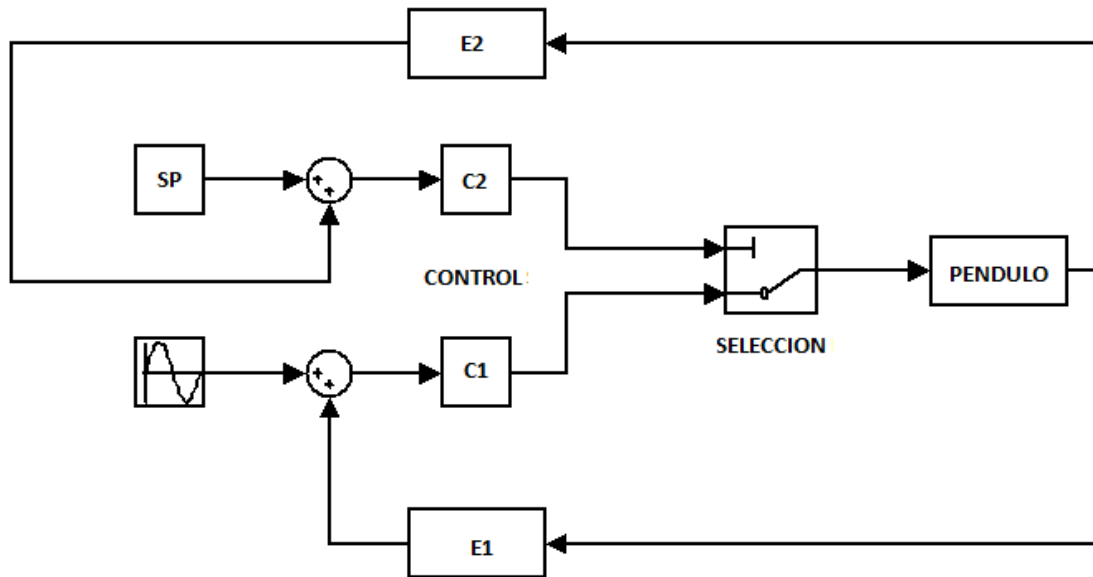
En este capítulo se describen los controladores usados tanto como para el funcionamiento del péndulo invertido como los que se programaron para que los alumnos de la facultad puedan realizar pruebas de los diferentes controladores.

#### 6.1 Control de posición

Para el péndulo invertido se implementó un control de posición basado en la lectura de los encoders, se calcula la posición en la que se encuentra tanto el carro como las varillas del péndulo, entonces se calcula la diferencia entre estas posiciones actuales y las deseadas, de esta manera se logra aplicar al motor el torque necesario para llevar las varillas a la posición deseada. Un controlador corrige el error entre un valor medido y el valor al que se quiere llegar. Los diseños de controladores que operen de manera adecuada para sistemas subactuados como lo es el péndulo invertido han sido objeto de estudio desde hace varios años, es por esto que se implementaron tres controladores para observar la respuesta del péndulo y poder comparar el funcionamiento de cada uno, se escogieron un control  $P$ , un  $PID$  y un  $TANH$ .

##### 6.1.1 Controladores para el péndulo invertido

Para el funcionamiento del péndulo invertido, se realiza en dos controladores Figura 6.1, el primero es un controlador senoidal que se encarga de mover el carro de un lado a otro con el objetivo de dar impulso a las varillas para que comiencen a levantarse y se acerquen a la región de estabilización y posición deseada, este controlador se ejecuta en cuanto el usuario activa el botón de encendido después de haber seleccionado todos los parámetros para los demás controles, después de esto el funcionamiento del péndulo es completamente automático y uno no tiene más que observar los resultados y detener la ejecución del programa si es necesario o si se quiere cambiar las especificaciones de los controles. El segundo controlador puede ser  $P$ ,  $PID$  o  $TANH$ , y tienen como objetivo una vez que las varillas están lo suficientemente cerca de la posición deseada de llevarlas y mantenerlas en esta posición. En el siguiente diagrama se muestra como se conmuta entre estos dos controladores.

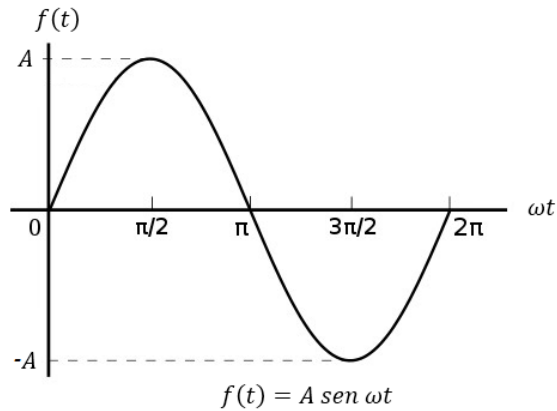


**Figura 6.1:** Funcionamiento del control para el péndulo invertido.

De la Figura 6.1, C1 es el primer controlador el senoidal, E1 es el error del carro y la posición deseada está en función de la onda senoidal. C2 es el segundo controlador cuyo error E2, es el de la posición de las varillas y SP es la posición deseada de estas, es decir 180°.

### 6.1.1.1 Control senoidal para levantamiento del péndulo

En la ingeniería eléctrica las funciones senoidales son importantes por su relativa sencillez de construcción además de sus propiedades matemáticas, pues es la única función periódica cuya derivada e integral son de la misma forma, las ondas senoidales abundan en la naturaleza, como en el rebote de una pelota, las vibraciones de una cuerda y también en el movimiento de un péndulo. Mediante este controlador se pretende hacer entrar al péndulo en resonancia, se quiere que el carro describa una onda senoidal a lo largo del tiempo, es decir cada valor de la onda senoidal introducida será una nueva posición deseada para el carro, la señal del error puede ser positiva o negativa, según en qué parte del ciclo de la onda senoidal se encuentre, esto determina también la dirección del motor. La función senoidal, es periódica con se muestra en la Figura 6.2 [22].



**Figura 6.2:** Función senoidal.

Es una función que se repite cada  $T = 2\pi/\omega$  o cada  $2\pi rad$  y viene determinada por la siguiente función

$$f(t) = A \text{ sen } \omega t \quad (6.1)$$

- $f(t)$  es el valor instantáneo en un tiempo  $t$
- $A$  el valor máximo o amplitud
- $\omega$  la frecuencia angular

$$\omega = \frac{2\pi}{T} = 2\pi f \text{ [rad/seg]} \quad (6.2)$$

$$f(t) = A \text{ sen} 2\pi f t \quad (6.3)$$

El usuario podrá variar la amplitud para determinar la distancia que recorrerá el carro hacia un lado o hacia el otro y también se podrá cambiar la frecuencia para cambiar la velocidad con la que el carro se mueve. El funcionamiento del controlador se basa en que la posición de un péndulo en movimiento describe una onda senoidal a lo largo del tiempo por lo tanto se quiere poner al péndulo en movimiento para que las varillas comiencen a alzarse. La señal senoidal se convierte en la posición deseada del carro y se usa uno de la forma TANH, de esta manera nuestra ecuación de control tiene la forma

$$\tau = k_p \tanh(\varepsilon) \quad (6.4)$$

- $\tau$  es la energía aplicada al motor y que por lo tanto se tendrá que convertir a su equivalente en ciclo de trabajo para el bloque de pwm en el FPGA.
- $\varepsilon$  es el error, que calculamos en cm con la fórmula  $\varepsilon = q_d - q'$ .
- $k_p$  es una variable ingresada por el usuario, en la que se podrá escoger el porcentaje de acción del controlador.

La posición deseada vendrá determinada por la función senoidal, en la que podremos variar los parámetros de la amplitud y la frecuencia

$$q_d = A \text{sen}(\omega t) \quad (6.5)$$

- $A$  es la amplitud de la señal, o lo que sería lo mismo los centímetros hacia la derecha y hacia la izquierda que el carro recorrerá.
- $\omega t$  está en función de la frecuencia, es la velocidad con la que se quiera mover al péndulo, una vez que se ingresa la frecuencia podemos calcular  $\omega$  con la expresión  $\omega = 2\pi f$

El tiempo será el periodo en que se ejecute la estructura Timed Loop, multiplicado por el muestreo en el que nos encontremos es decir  $i$ , entonces viene determinado por  $t = i\Delta t$

La ecuación de control nos da como resultado un valor  $\tau$  que corresponde al torque que se debe aplicar al motor, tenemos que convertir este valor a un valor representado en PWM para el bloque del FPGA, usamos la expresión que obtenemos de las gráficas resultantes de la caracterización del motor, partiendo de nuestra ecuación 6.4.

El torque se puede expresar también como  $(PWM)(m)$ , entonces

$$(PWM)(m) = k_p \tanh(\varepsilon) \quad (6.6)$$

$$PWM = \frac{k_p}{m} \tanh(\varepsilon) \quad (6.7)$$

$m$  es la pendiente de la curva del motor, que podemos calcular con ayuda de las gráficas obtenidas para la caracterización del motor y de la ecuación de una recta  $y = mx + b$ , así describimos a la recta resultante de la caracterización y la pendiente se calcula escogiendo dos puntos de la gráfica y sustituyendo en

$$m = \frac{y_1 - y_2}{x_1 - x_2} \quad (6.8)$$

Por otro lado tenemos que él torque es esta pendiente por el porcentaje de ciclo de trabajo y podemos despejar el valor PWM

$$\tau = m \text{ PWM} \quad (6.9)$$

$$PWM = \frac{\tau}{m} \quad (6.10)$$

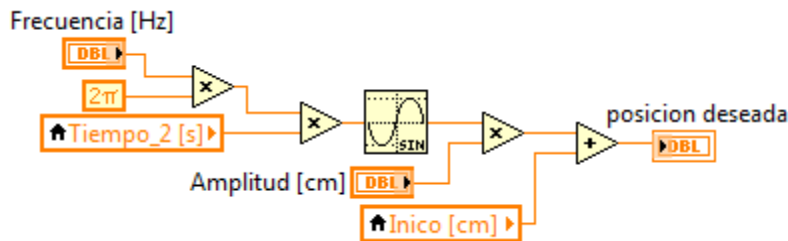
Con estas ecuaciones podemos convertir el resultado de la ecuación de control en un valor para nuestro bloque de PWM.

$$PWM\% = \frac{\tau}{0.017275} \quad (6.11)$$

Hay que tener en cuenta que la constante  $k_p$  tiene que cumplir la condición de la ecuación 6.12.

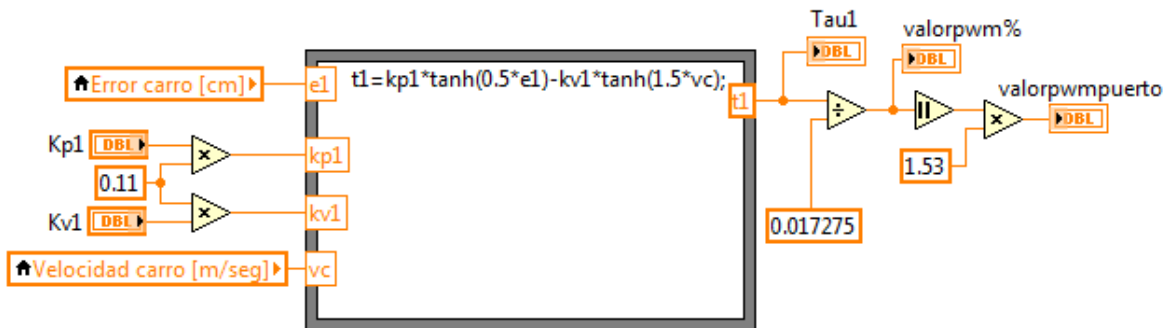
$$k_p < \tau_{max} < 1.1 \quad (6.12)$$

Esto usando el PWM desde el 0 hasta el 60 %, el torque máximo es 1.1 Nm, donde como vimos en la caracterización es la región donde el motor se comporta de forma lineal. La implementación de este controlador en LabVIEW se describe a continuación, primero se calcula el error del péndulo, dependiendo del resultado se ejecutará el control senoidal, si el valor se encuentra fuera del rango establecido por el usuario, es decir las varillas no se encuentran lo suficientemente arriba, el control senoidal entrara en acción, Figura 6.3.



**Figura 6.3:** Fórmula para función senoidal implementada en LabVIEW.

Con la posición deseada calculada solo resta calcular el valor para el bloque PWM en el FPGA. Las constantes  $kp1$  y  $kv1$  estas normalizadas a 10 para hacer más fácil su manipulación, esto se realiza como se muestra en la Figura 6.4.



**Figura 6.4:** Control TANH para el controlador de la forma senoidal.

La variable  $Tau1$  entregada por la ecuación de control pasa por las fórmulas descritas anteriormente para calcular su equivalente en PWM, el sentido de giro

se calcula en base a la posición del carro, si el error es positivo significa que el carro deberá continuar con la dirección actual en cambio si es negativo significa que la función senoidal se encuentra en el cuadrante opuesto y por lo tanto el carro deberá ahora cambiar el sentido de giro para tratar de alcanzar la posición deseada.

## 6.2 Controlador de estabilización.

Cuando las varillas han alcanzado una altura suficiente debido al movimiento del carro, entra en acción este controlador donde se ejecuta uno de los tres disponibles, P, PID o TANH, según se haya escogido. Este controlador entonces se activa cuando las varillas entran en el rango de acción indicado por el usuario. El funcionamiento de cada uno de estos controladores se describe detalladamente en las siguientes secciones.

### 6.2.1 Acción de control Proporcional

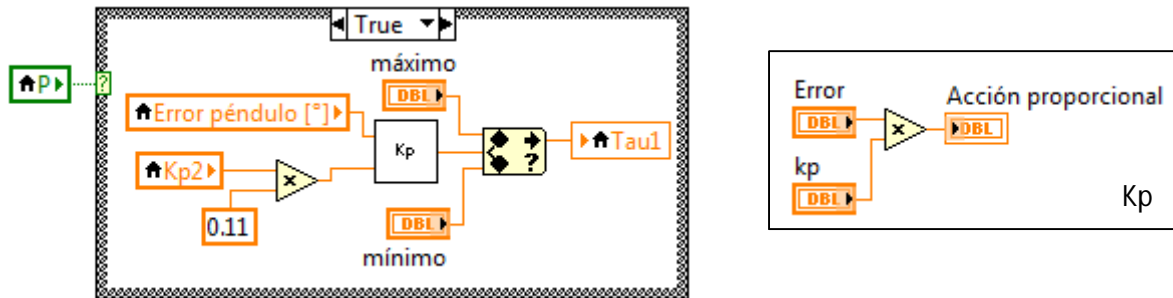
Este es el más sencillo de los tres controladores que se implementaron, el motivo por el que se escogió es que al principio no sabíamos cómo iba a responder el péndulo invertido no solo a este controlador si no a la estructura en general, al cambio entre el controlador dos y el tres y su ejecución en tiempo real, al ser la primera vez que se ponía a funcionar el péndulo invertido con objeto de tener una sintonización sencilla se decidió comenzar con lo más básico para observar el funcionamiento del sistema. Proporcional significa que el cambio presente en la salida del controlador es algún múltiplo del porcentaje de cambio en la medición, el término proporcional es el encargado de indicar que tan lejos se encuentra actualmente el sistema con respecto al valor deseado, el controlador es en general un amplificador con ganancia ajustable donde la relación entre la salida del controlador y la señal del error es

$$u(t) = k_p e(t) \quad (6.13)$$

- $k_p$  es la ganancia proporcional
- $e(t)$  es el error del sistema
- $u(t)$  es la acción de control resultante

Cuanto mayor es la ganancia del control proporcional mayor es la señal de control generada para un mismo valor de señal de error, por lo que un aumento en la ganancia del controlador reduce el error en estado estacionario hasta cierto límite, pero hace al sistema más sensible y menos estable. Este control tiene las

propiedades de reducir el tiempo de subida, incrementar el sobretiro y reducir el error de estado estable. Esto quiere decir que mientras más alta sea la constante menor será el tiempo que tarde en reducir el error del sistema pero tenderá a oscilaciones sub-amortiguadas alrededor del valor deseado, en cambio si la constante es pequeña aunque el sistema no tenga oscilaciones nunca llegara al valor deseado. La constante se introduce a la ecuación de control normalizada a 10, el usuario puede introducir fácilmente un valor entre el 1 y el 10 y el programa lo ajustará automáticamente al valor real para la ecuación de control, Figura 6.5.



**Figura 6.5:** Bloque de control P y acción de control proporcional dentro de  $K_p$ .

### 6.2.2 Acción de control proporcional integral derivativa PID

Este es el algoritmo de control más empleado en la industria, ya que proporciona una gran flexibilidad y es la base teórica de todos los métodos de sintonía automática más habituales en los reguladores industriales.

El valor proporcional como se había mencionado antes determina la relación del error actual.

El valor integral almacena información sobre las actividades pasadas del sistema, genera una corrección proporcional a la integral del error, le brinda al sistema de control una memoria de cómo fue evolucionando el error del mismo.

El valor derivativo hace una comparación del error pasado con el actual con lo que obtiene una noción de cómo seguirá respondiendo el sistema, determina la reacción de control del tiempo en el que el error se produce.

La suma de las tres constantes es entonces usada para ajustar al controlador, existen diversas versiones del algoritmo de control PID, de entre las cuales se implementó el algoritmo paralelo, en esta estructura las tres acciones de control actúan directamente sobre la señal de error, cada acción de control puede ser variada por separado sin que ello pueda inducir a cambios en las demás acciones de control, la variación de cada constante tiene un efecto diferente que se puede ver en la Tabla 6.1

**Tabla 6.1:** Efecto de las constantes sobre el control PID.

Acción de control	Tiempo de subida	Sobre impulso	Tiempo de estabilización	Error
Kp	Baja	Sube	Casi no varia	Baja
Ki	Baja	Sube	Sube	Elimina
Kd	Casi no varia	Baja	Baja	Casi no varia

La acción de control integral es la llamada también reset, la acción correctora se efectúa mediante la integral del error, el control entrega una señal que es función de la historia del error pues la integral es una función acumulativa del tiempo permitiendo así obtener una señal de control diferente de cero aun cuando el error lo sea. Así como la acción de control proporcional responde al tamaño del error y la integral al tamaño y duración del error, la acción de control derivativa responde a cuán rápido cambia el error, genera una señal de control proporcional a la derivada del error, tiene un efecto de anticipación que es tanto importante como peligroso pues puede presentar la acción correctiva antes de que la señal de error sea excesiva sin embargo si la acción de control puede ser demasiado severa si la constante no se calcula con cuidado y además si el error es constante su derivada será cero y el control adoptará una actitud pasiva. Por estas razones que el ajuste del controlador PID debe hacerse con cuidado y es recomendable seguir los siguientes pasos:

- Poner  $k_d$  y  $k_i$  en 0 y  $k_p$  en 1
- Incrementar  $k_p$  lentamente y observar la respuesta, se continúa incrementando hasta que el sistema oscile, es decir se mueva alrededor del punto deseado.

Ahora podemos empezar a incrementar  $k_d$ , en algún punto se producirá un paro y el robot se detendrá cerca del punto deseado ya que  $k_d$  es un término de velocidad y domina la función de salida cuando el sistema está cerca de su posición deseada, y ya que es restado causa que la cantidad de energía suministrada descienda hasta parar al motor en la posición deseada.

Si  $k_d$  es muy grande, ocurrirán oscilaciones realmente malas, por lo tanto es necesario incrementar lentamente.

$k_i$  es la constante más difícil de sintonizar, hacerla muy grande pondrá al motor a oscilar, esta se debe de modificar lentamente y se tiene que observar la respuesta para determinar cuánto de esta constante es necesario en el controlador.

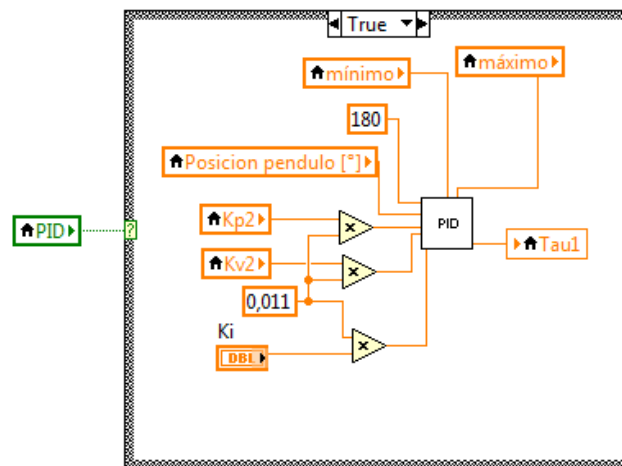
En resumen un control proporcional mejora el tiempo de subida, un control derivativo para mejorar el sobre impulso y un control integral elimina el error de estado estacionario. Como al implementar este controlador ya se tenía experiencia con el controlador P, se sabía de antemano que valores para esta constante funcionaban mejor, de esta forma la sintonización de este controlador fue un poco más sencilla pues nos concentramos en las otras dos constantes. La función de transferencia del controlador PID es la siguiente

$$k_p + \frac{k_I}{s} + k_D s = \frac{k_D s^2 + k_p s + k_I}{s} \quad (6.14)$$

Entonces con el controlador PID en lazo cerrado, la señal de salida  $u$ , ecuación 6.15.

$$u = k_p e + k_I \int e dt + k_D \frac{de}{dt} \quad (6.15)$$

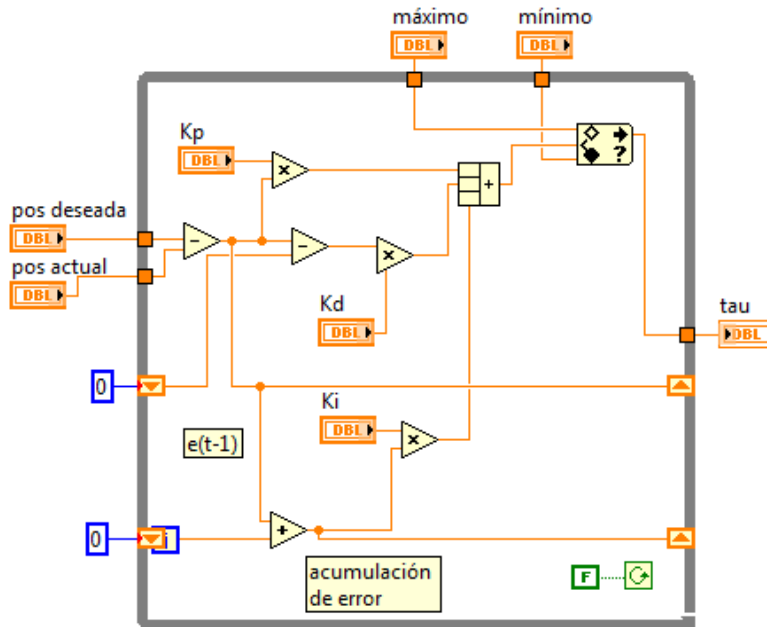
La variable  $e$  representa el error, esta señal de error se envía al controlador PID, y se calcula la integral y la derivada de la señal de error para poder aplicar la ecuación anterior, esta señal será enviada al FPGA y se obtendrá una nueva salida, de esta nueva salida se obtendrá una nueva señal de error, la cual se procesará de nuevo por el controlador y así sucesivamente. La ecuación se implementó en LabVIEW de la siguiente manera Figura 6.6.



**Figura 6.6:** Bloque para controlador PID.

Tanto para el control P como PID se crearon sub bloques ".vi", es decir programas en LabVIEW a los cuales se les creó su símbolo para mandarlos a llamar y así evitar el desorden en el programa principal, simplemente hay que tener en el mismo directorio todos los programas juntos y no cambiar el nombre de los programas entonces LabVIEW los reconocerá automáticamente. La secuencia se encuentra dentro de un CASE STRUCTURE, que se ejecutará dependiendo de

su respectivo controlador, dentro del bloque PID se encuentra la función de transferencia del controlador, Figura 6.7.

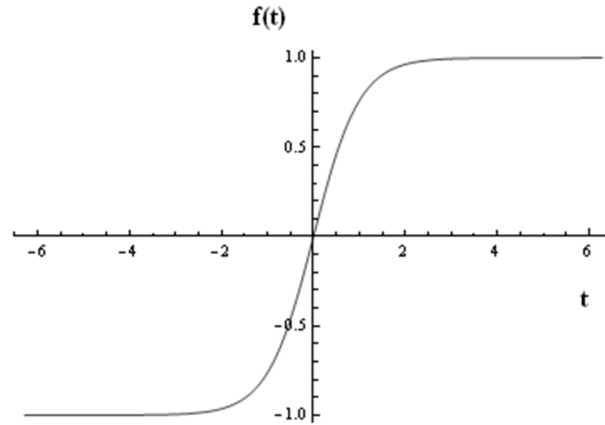


**Figura 6.7:** Acción de control proporcional integral derivativo.

### 6.2.3 Acción de control Tangente Hiperbólica

En años recientes se han propuesto controladores que resuelvan el problema de saturación de los actuadores de los sistemas, para esto la estructura del controlador está basada en funciones trigonométricas, entre las más importantes se encuentran el controlador tangente hiperbólica, TANH. Este controlador fue introducido para reducir el error de posición debido al fenómeno de fricción articular, esto gracias a que se satura de forma natural pues la función. El controlador TANH tiene la misma estructura que el PD, lo que cambia es el error de posición y de velocidad se les aplica una función tangente hiperbólica cuyos límites superior e inferior es de  $\pm 1$ . La ley de control está dada por la ecuación 6.16 y su gráfica de comportamiento se muestra en la Figura 6.8 [8]

$$\tau = K_p \tanh(\tilde{q}) - K_v \dot{q} + g(q) \quad (6.16)$$



**Figura 6.8:** Gráfica de comportamiento de la función TANH.

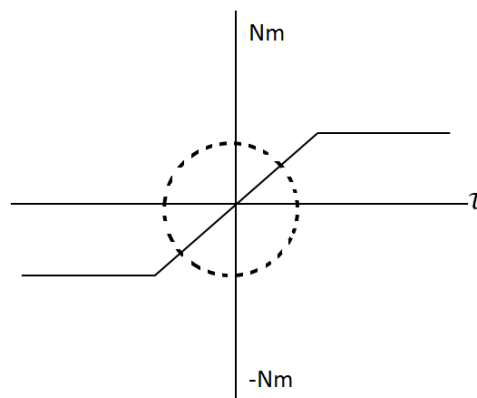
Al que igual que en los demás controladores, para el ajuste de las constantes hay que tener en cuenta la siguiente relación entre las constantes y el torque máximo para el motor

$$Kp < \frac{\tau_{max}}{qd} \times 0.6 \quad (6.17)$$

Con esto estamos trabajando al 60 % del sistema, de forma que tenemos un margen de seguridad para trabajar y nos aseguramos que nunca se le enviará al motor un voltaje demasiado alto y también de que se trabaje el torque en el área lineal, para la constante  $Kv$  tenemos que

$$Kv = 0.1Kp \quad (6.18)$$

De esta manera tenemos una guía para sintonizar las constantes. En la Figura 6.9 podemos ver el área de trabajo del controlador



**Figura 6.9:** Región de trabajo del controlador TANH.

La implementación de este control en LabVIEW la podemos observar en la Figura 6.10, hay que recordar que para todos los controladores, el  $\tau$  obtenido debe pasar por una ecuación para poder obtener el valor que se introducirá en el bloque PWM del FPGA.

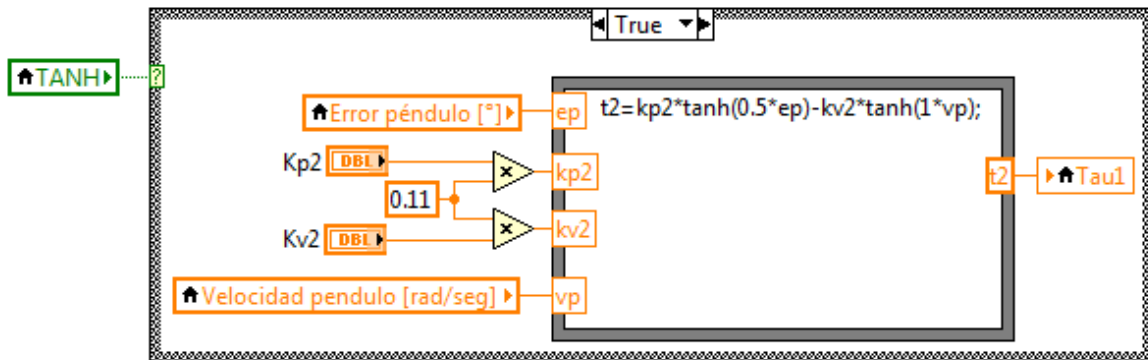


Figura 6.10: Controlador TANH en LabVIEW.

## Conclusiones

Se explicó cómo es que funciona el péndulo invertido para llevar a cabo los movimientos solicitados por el usuario como lo son el retorno a casa o el posicionamiento en el inicio. El control para el funcionamiento del péndulo invertido se dividió en dos partes, el primer control proporciona impulso a las varillas para que estas lleguen a una posición cercana a la deseada y el segundo se encarga de llevarlo a esta posición y de mantenerlo, ambos controles se pueden configurar por el usuario y el control final tiene la opción de ser P, PID o TANH de manera que el estudiante puede experimentar de muchas formas, estos controles se implementaron en el software de LabVIEW mediante estructuras de ecuaciones y sub bloques .vi sin embargo la facilidad de programación de LabVIEW permite que el software se pueda robustecer de manera que un estudiante con los conocimientos necesarios puede generar sus propios bloques de control y añadirlos al programa principal, con esto cabe la posibilidad de en un futuro aumentar las posibilidades de simulación de algoritmos del péndulo invertido.

## Capítulo 7

### Resultados experimentales

En este capítulo se muestran los resultados obtenidos con los controladores usados en el péndulo invertido, para poder comparar el desempeño entre ellos, también serán comparados con los resultados obtenidos de la simulación del sistema.

El modelo y simulación es indispensable cuando hacemos diseño y análisis de sistemas mecatrónicos, nos permite predecir el comportamiento del sistema y también su respuesta a un determinado controlador, por esto es que se realizaron simulaciones del sistema del péndulo invertido y de su control para tener una idea de que esperar de las pruebas y para poder comparar con los resultados obtenidos, se usó Matlab que es capaz de simular el comportamiento de sistemas continuos y sistemas discretos representados mediante ecuaciones diferenciales. Al realizar las pruebas el programa en LabVIEW se encargaba de almacenar los datos obtenidos, estos se puede abrir ya sea en LabVIEW, Matlab, o cualquier otro software para poder graficar y analizar los resultados de las pruebas. En el archivo los datos se almacenan en forma de columnas, cada una corresponde a una variable calculada en el programa, las variables se muestran a continuación en el orden en el que se encuentran guardadas en el archivo.

- Tiempo [s]
- Posición del carro [cm]
- Posición del péndulo [°]
- Error del carro
- Error del péndulo
- Tau1 (0-0.1) [Nm]
- Velocidad del carro [cm/seg]
- Velocidad del péndulo [°/seg]

## 7.1 Controlador P

Para este controlador se implementó la siguiente ecuación:

$$\tau = k_p e \quad (7.1)$$

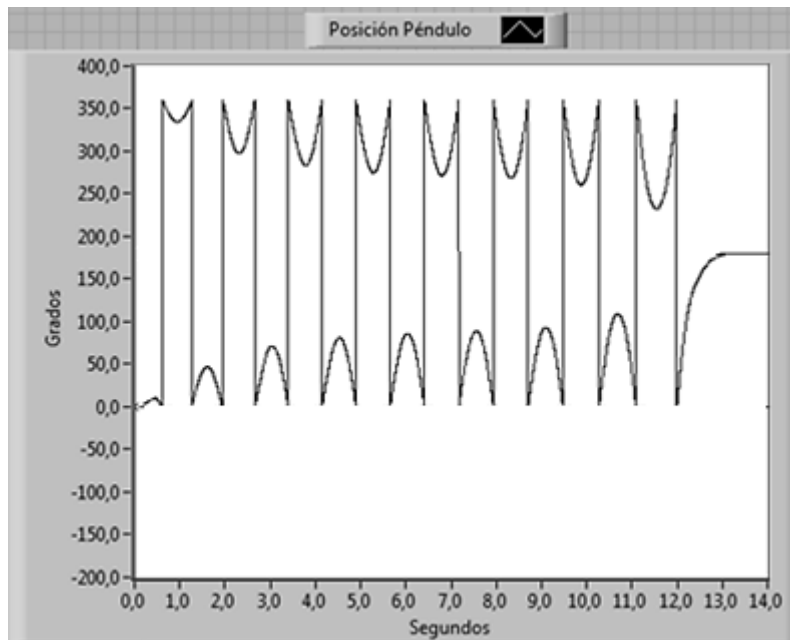
Donde:

- $\tau$  es la acción de control.
- $k_p$  es la constante proporcional.
- $e$  es el error.

Con el controlador proporcional se buscaba principalmente realizar una primera calibración del péndulo invertido, determinar los parámetros tales como el periodo de muestreo de LabVIEW, la posición de inicio para el carro, la frecuencia y la amplitud del control senoidal, y el rango de acción para el segundo controlador. Conocer los parámetros que hicieran que el péndulo oscilara más rápidamente pero al mismo tiempo de manera suave para que el segundo controlador tuviera tiempo de accionarse fue la primera tarea que se llevó a cabo para así poder después concentrarse en los controladores encargados de mantener al péndulo en la posición deseada.

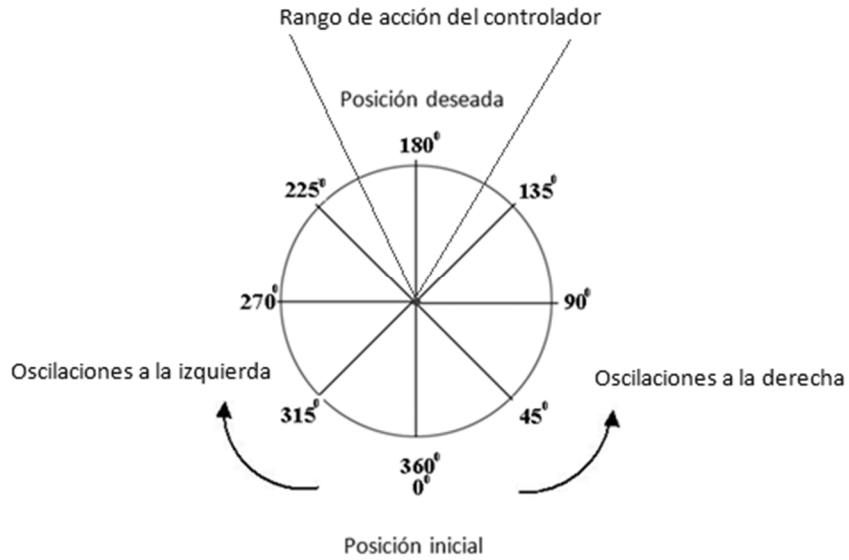
Al realizar las pruebas se observó que cuando se usaba una frecuencia de 0.6758 Hz es decir, se ponía a oscilar al carro con un periodo de aproximadamente 1.51 segundos el control senoidal del péndulo respondía bastante bien este valor fue constante para todas las pruebas, la amplitud de las oscilaciones varió de entre 13 y 18 cm con esta distancia el carro tenía un recorrido bastante amplio y no estaba cerca de chocar con los extremos, el periodo para el Timed Loop de LabVIEW fue de 2 ms pues eran suficientes para realizar la adquisición de los datos, su interpretación, la aplicación de la ecuación de control y el almacenamiento de los resultados, el rango de acción del segundo controlador se determinó entre  $\pm 10^\circ$  y  $\pm 40^\circ$  alrededor de la posición deseada así el segundo controlador entraba en acción no tan cerca de  $180^\circ$  y tenía tiempo para actuar, además también se calibraron las constantes de controlador senoidal, los valores que hacían oscilar al péndulo rápida pero suavemente fueron 8 y 9 para  $k_{p1}$  y para  $k_{v1}$  de 1 y 1.5, el controlador proporcional fue de bastante ayuda en estas calibraciones por ser el más sencillo de los tres, se buscó que ya no se modificaran estos parámetros durante las pruebas para poder actuar solamente sobre las constantes de los controladores.

En las siguientes figuras se observan los resultados de una de las pruebas del control P con una constante  $k_{p2}$  de 8.99976, se manejaron valores con hasta 6 décimas, en la Figura 7.1 se observan los resultados de la posición de las varillas del péndulo.



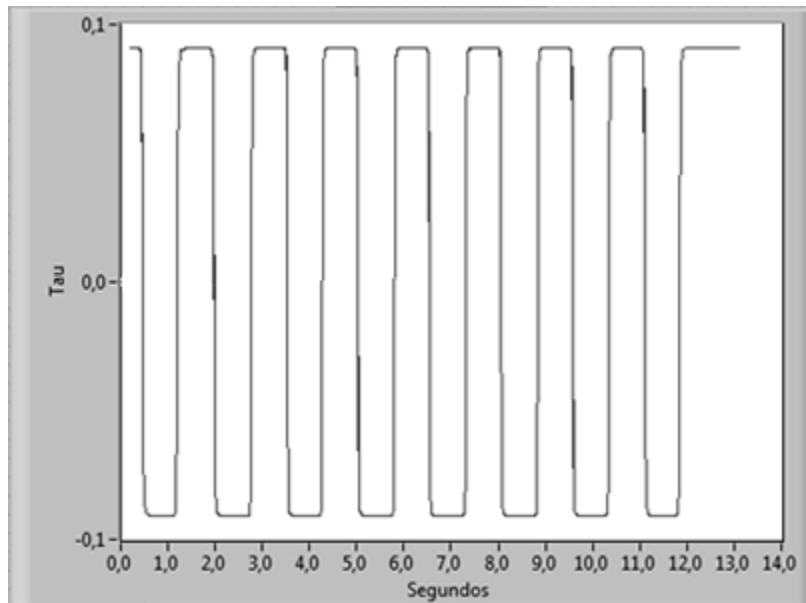
**Figura 7.1:** Posición del péndulo, controlador tipo P.

El recorrido de las varillas hasta llegar a la posición deseada duró poco más de 12 segundos después de haber comenzado la prueba, en la Figura 7.2 se muestra las diferentes posiciones que pueden tomar las varillas al oscilar, con esto se puede explicar cómo varían los valores de la posición en la Figura 7.1, al comienzo de la prueba las varillas se movieron hacia la derecha luego ya que el impulso aún es muy poco en menos de un segundo regresan, pasan por la posición inicial ( $0^\circ$ ) y ahora se mueven hacia la izquierda, con forme pasa el tiempo la amplitud de las oscilaciones es cada vez mayor, por ejemplo hacia la derecha la primera llegó apenas a unos  $10^\circ$  luego la segunda hacia la derecha a unos  $50^\circ$ , la tercera a  $70^\circ$ , la cuarta a unos  $80^\circ$  hasta que en la novena llega a alcanzar la posición deseada.



**Figura 7.2:** Recorrido de las varillas del péndulo invertido.

En la Figura 7.3 se ve el Tau a la salida del control.



**Figura 7.3:** Tau resultante del control P.

El Tau que se observa es en realidad generado por el controlador senoidal, pues al inicio de la prueba no se ha llegado al rango de acción del segundo controlador, ya que el controlador senoidal es proporcional al error, este al ser grande al comenzar la prueba hace que se inyecte una cantidad grande de Tau también,

en la prueba llega casi a 0.1 que es su valor máximo, los valores negativos o positivos indican el sentido de giro del motor, cuando cerca de los 12 segundos después de iniciar la prueba las varillas alcanzan el rango del segundo controlador, el encargado de generar el Tau es el controlador proporcional, al ser menor el error el Tau también lo es y en poco tiempo el controlador hace que Tau ya no varíe, es por esto que después de los 12 segundos el valor es constante, las varillas entonces llegan a la posición deseada suavemente y se mantienen hay por poco tiempo hasta que el peso de las mismas las obliga a bajar, el controlador no tiene tiempo de reaccionar y no las puede mantener en la posición deseada, el control es un poco rudo debido a que la energía crece o disminuye con rapidez según el error y por lo tanto no es efectivo para mantener las varillas en la posición deseada, si la constante  $k_p$  no es escogida con cuidado podría no tener ningún efecto o por el contrario suministrar demasiada energía al sistema.

## 7.2 Controlador PID

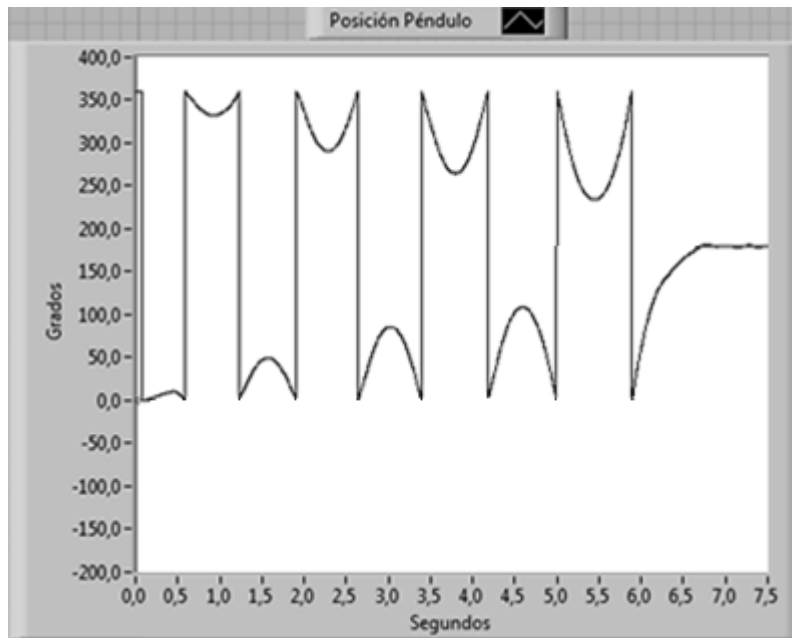
Este control tiene la forma:

$$\tau = k_p e + k_i \int e dt + k_v \frac{de}{dt} \quad (7.2)$$

Donde:

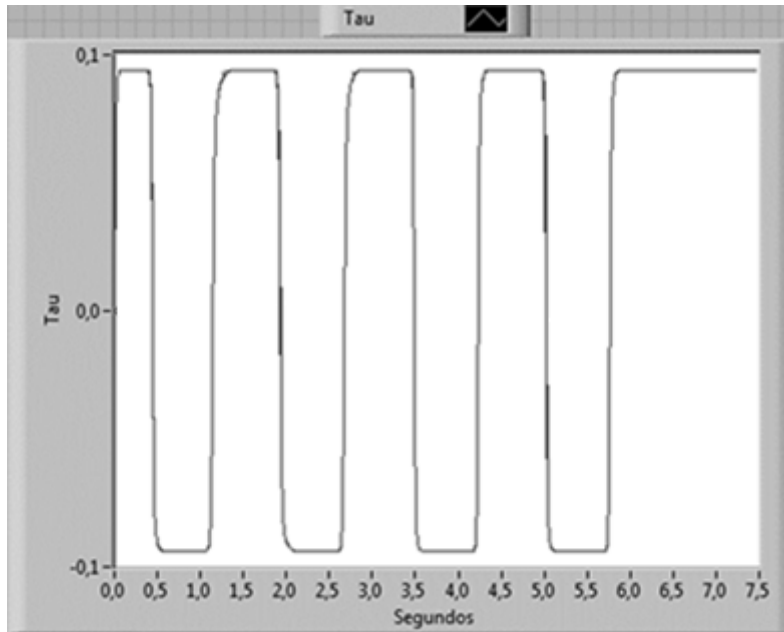
- $k_i$  es la constante de la parte integral
- $k_v$  es la constante de la parte derivativa

A continuación se muestran los resultados para una de las pruebas del controlador PDI, en este caso las variables  $k_p$ ,  $k_v$  y  $k_i$  son 10, 5 y 2 respectivamente.



**Figura 7.4:** Posición del péndulo, controlador PID.

En la Figura 7.4 se observa rápidamente que el controlador logró llevar a las varillas a la posición deseada en mucho menos tiempo que el controlador P, prácticamente en la mitad del tiempo, esto se debe a que la amplitud de las oscilaciones incrementa rápidamente y la acción del control PID actúa más efectivamente que la acción de control P y por lo tanto en esta prueba al cabo de 6 segundos se logra llegar a la posición deseada, en la Figura 7.5 se observa el Tau inyectado al péndulo durante la prueba, este es muy parecido al de la prueba con el controlador proporcional, con la diferencia de que debido a la participación de las otras dos constantes (integral y derivativa), el control ahora es capaz de mantener a las varillas en la posición deseada, al principio igual que en el controlador proporcional el Tau proviene del controlador senoidal y después, poco antes de los 6 segundos, se cambia al control PID, este es más suave y constante logrando así mantener a las varillas en la posición vertical un par de segundos.



**Figura 7.5:** Tau resultante del control PID.

### 7.3 Controlador tanh

La ecuación de este control es de la forma:

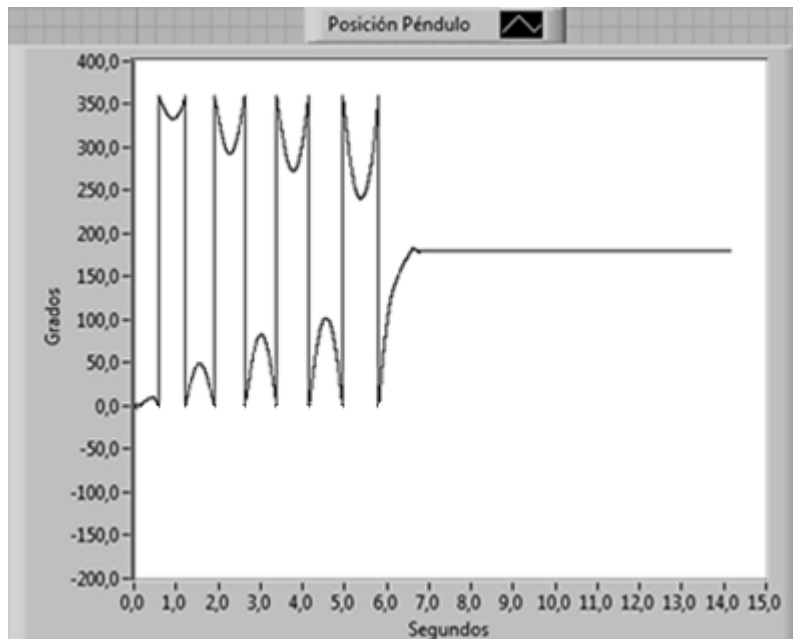
$$\tau = K_p \tanh(\tilde{q}) - K_v \dot{q} \quad (7.3)$$

Donde:

- $\tilde{q}$  es el error
- $\dot{q}$  es la velocidad

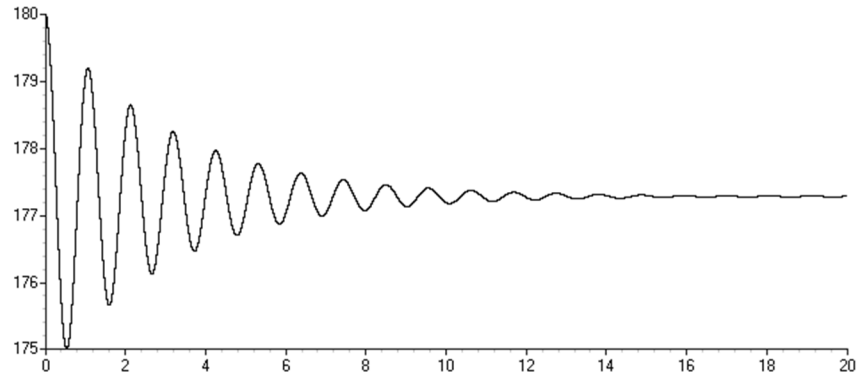
Para este control el valor de  $k_p$  que se usó es de 10 mientras que  $k_v$  fue de 1.5. Este control fue el que mejor se desempeñó de los tres ya que como se muestra en la Figura 7.6, tardó 6 segundos al igual que el controlador PID en estabilizarse, además este controlador fue capaz de mantener esta posición, el control PID al llegar las varillas del péndulo a la posición vertical, no pasaba mucho tiempo para que estas cayeran y volver a subirlas era más complicado que la primera vez que lo hizo, sin embargo el control tanh las mantenía en esta posición hasta que una fuerza externa (un golpe al carro) se aplicaba, y aún si caían el control tanh logró volver a subir las varillas después de esto. Otra gran diferencia es que los controladores P y PID no eran muy rápidos, es decir se tenía que dosificar la fuerza al sistema para que las varillas no tuvieran demasiado impulso y los controladores pudieran reaccionar, el controlador tanh podía

reaccionar más fácilmente aun cuando las varillas llevaban una mayor velocidad. A continuación se muestran los resultados de una de las pruebas.



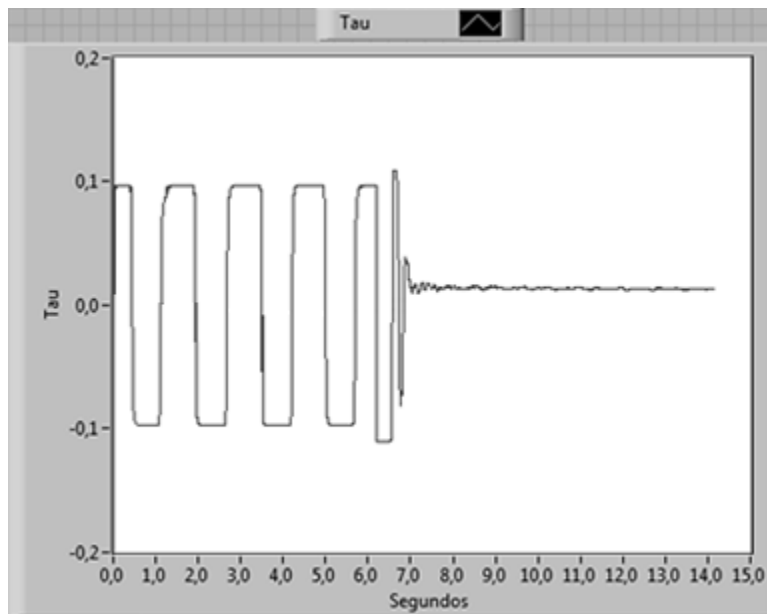
**Figura 7.6:** Posición del péndulo.

En la Figura 7.6 se observa que la amplitud de las oscilaciones aumenta rápidamente debido al control senoidal, cuando se entra en el rango de acción el control tanh la posición se estabiliza y se mantiene debido a la dosificación del Tau aplicado, los resultados de esta prueba se comparan con una simulación realizada, en la Figura 7.7 se observa la posición de las varillas del péndulo, los valores para las constantes  $k_{p2}$  y  $k_{v2}$  introducidos en la simulación fueron los mismos con las que se obtuvieron los mejores resultados en la práctica, cuando comienza a actuar el controlador tanh la posición de las varillas está cerca de los  $180^\circ$  y no tarda mucho en estabilizarse, esto lo hace alrededor de  $177^\circ$  lo cual es un error de  $3^\circ$ , que es aceptable.

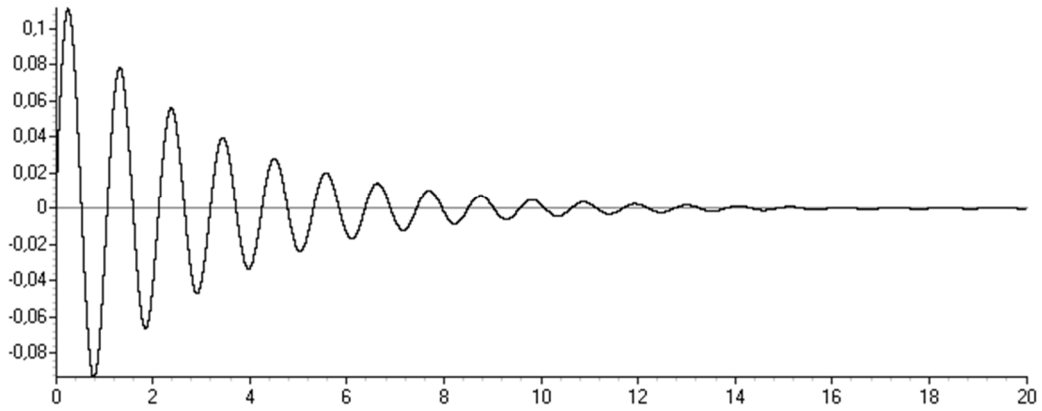


**Figura 7.7:** Posición del péndulo, simulación.

Cabe mencionar que aun cuando en la Figura 7.6 se muestran 16 seg, en realidad en la posición se mantuvo por algunos minutos. El Tau resultante de esta acción de control es diferente en que fue más suave que los otros controladores, en la Figura 7.8, el Tau de los primeros 6 segundos corresponde a la acción de control senoidal, de ahí en adelante el Tau fue determinado por el control tanh, esto muestra claramente cuando se cambió de un controlador a otro a diferencia de los controladores P y PID donde no es tan obvio el cambio de controlador. En las Figuras 7.8 y 7.9 se puede ver la comparación entre los resultados del Tau obtenido en la práctica y en la simulación, estos no son muy diferentes, ambos comienzan en su valor máximo de  $\pm 0.1$  y disminuyen gradualmente en el caos de la simulación hasta cero y en la práctica a un valor muy cercano a cero.

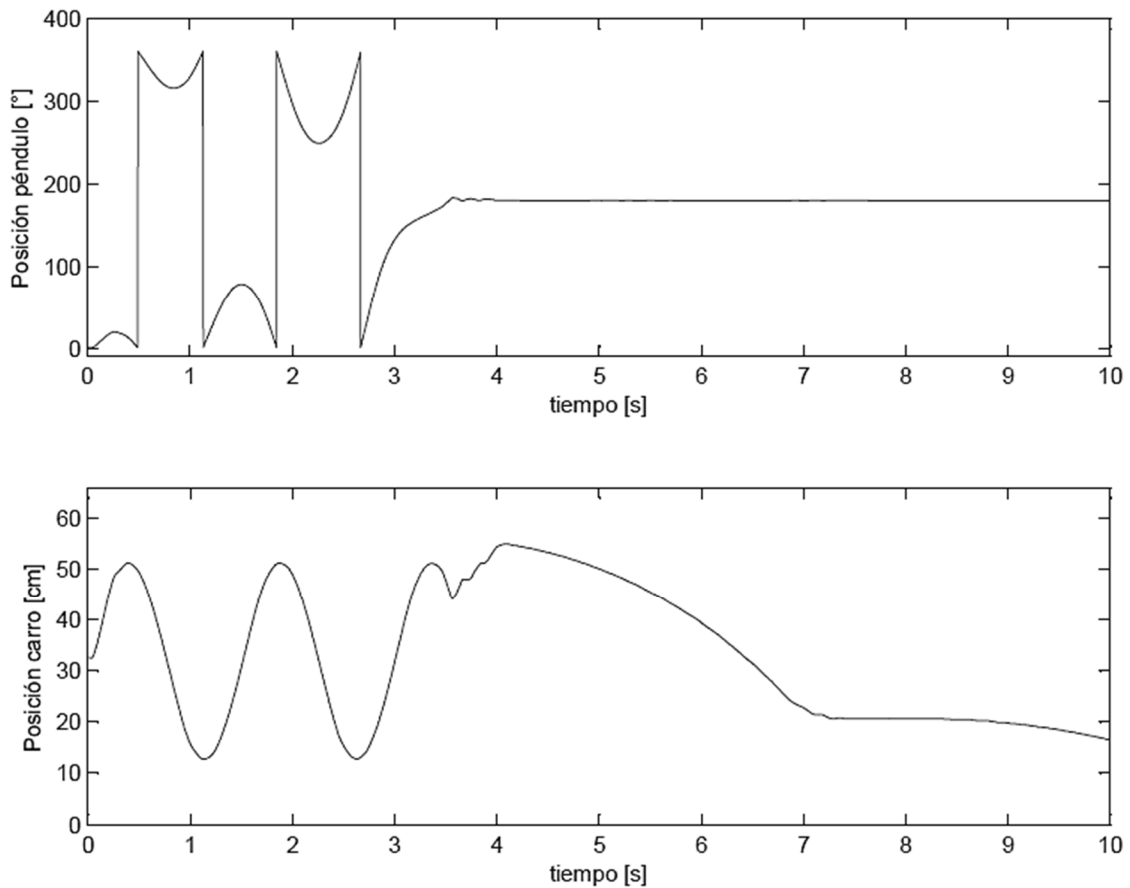


**Figura 7.8:** Tau resultante del controlador tanh.



**Figura 7.9:** Tau, simulación.

Como se mencionó antes la ventaja de este controlador es su capacidad de mantener a las varillas en su posición deseada, Figura 7.10 se observan dos parámetros obtenidos de la prueba, la posición del péndulo y del carro.



**Figura 7.10:** Resultados de una prueba con controlador tanh.

## 7.4 Evaluación del mejor controlador

Cada uno de los controladores tiene importantes ventajas, sin embargo es claro que el controlador *tanh* fue con el que mejores resultados se obtuvieron.

El control P fue especialmente importante ya que nos permitió llevar a cabo la calibración en general del sistema, y fue un punto de partida ideal para poder tener bases sobre las cuales trabajar para los controladores PID y *tanh*. Este control fue capaz de llevar al péndulo a la posición deseada, aun cuando tardó más que los controles PID y *tanh* y a pesar de no ser capaz de conservar esta posición, con los resultados y la práctica que se obtuvieron de las pruebas con el control P, fue mucho más sencillo implementar los otros controladores.

También se pudo comprobar porque el control PID es tan popular en la industria, pues este control después de que se sintoniza correctamente dio resultados bastante buenos con respecto al tiempo de estabilización, además no es un control tan brusco en comparación al P, la sintonización del control se llevó a cabo con los pasos descritos en el capítulo seis y como ya se tenían valores que funcionaron para la constante proporcional fue aún más sencillo.



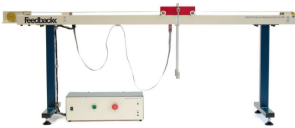
El controlador *tanh* fue el que mejor funcionó con el péndulo invertido, pues no solo estabilizó el sistema rápidamente si no que lo mantuvo en la posición deseada e incluso después de caer las varillas volvió a recuperar esta posición, la respuesta es mucho más suave y por lo tanto el consumo de corriente no es tan elevado, el hecho de que este control fue el último en probarse influyó en la sintonización pues como ya se conocía a profundidad el sistema, escoger las variables fue sencillo.

Comparando los resultados de la simulación con los de la práctica se puede observar que el comportamiento deseado del sistema no es muy diferente al obtenido, con lo cual se pueden llevar a cabo pruebas de control en el simulador antes que en el robot péndulo para poder estar seguros que los valores introducidos para las constantes de control no generarán resultados indeseados y poder tener una idea de que esperar al llevar a cabo las pruebas prácticas.

## 7.5 Comparativa entre el sistema desarrollado y sistemas comerciales.

Se realiza la cotización de dos sistemas comerciales y se comparan las características entre los comercializados y el sistema desarrollado en este trabajo de tesis, dando como resultado un sistema muy flexible en cuanto a material de construcción, hardware, firmware y software, además de un costo más bajo con respecto a los sistemas comerciales.

**Tabla 7.1:** Comparación entre sistema desarrollado y sistemas comerciales.

Diseño			
<b>Desarrollado por:</b>	FCE-BUAP. Maestría en Ciencias de la Electrónica	Latin Tech Inc IP-01	FEEDBACK 33-005-PCI
<b>Software de aplicación</b>	Labview	Matlab- simulink	Matlab- simulink
<b>Tarjeta de adquisición</b>	PCI- Fotón con FPGA Cyclone II o III	PTS-F1616 de la misma empresa	Tarjeta de interfaz PCI
<b>Interfaz</b>	Computadora Personal con puerto PCI	Abierto a usar algún microcontrolador, PLC, PC	Computadora Personal con puerto PCI
<b>Tipo de controlador</b>	P, PID, TANH, propuesto a implementar otros.	No proporciona información	Solo especifica PID, no proporciona más datos.
<b>Arquitectura abierta</b>	Si	No proporciona información	No proporciona información
<b>Material de fabricación</b>	Aluminio, bronce y nylamid	No proporciona información	No proporciona información
<b>Dimensiones (cms.)</b>	135x65x41	120x35x35	171x65x41
<b>Peso (kgs.)</b>	25	10	23
<b>Precio</b>	\$30,000.00	\$168,014.40	\$344,915.48

## **7.6 Ventajas y desventajas en el desarrollo del sistema mecatrónico.**

### **Ventajas**

- No se requirió de un presupuesto muy elevado para el desarrollo del sistema.
- Se tuvo la opción de seleccionar los materiales para su construcción.
- La tarjeta de adquisición de datos es solo para el propósito de adquisición de datos.
- El control se lleva a cabo a nivel software, por medio de la computadora.
- Se pueden cambiar los sensores por otros de las mismas características.
- Al realizar el cambio de algún sensor se puede hacer cambios a nivel firmware y software sin llegar a modificarlos en demasía.
- Los algoritmos de control pueden cambiar con facilidad, así como modificar sus parámetros de funcionamiento.
- El sistema es más flexible en la parte de firmware y software ya que se pueden realizar ajustes sin modificar el sistema mecánico.
- El sistema cuenta con un modo de almacenamiento de datos de funcionamiento para su análisis posterior con la ayuda de otro software como por ejemplo, Matlab.

### **Desventajas**

- El sistema no puede ser transportado con facilidad debido al peso, podría implementarse un sistema de transporte por medio de unas llantas con freno para mantener la posición estable al piso.
- El consumo de corriente del motor es elevado provocando que su manejo sea peligroso, además de que puede provocar ruido eléctrico a otros sistemas.
- Si se requiere migrar a otro tipo de FPGA, este tiene que ser con las mismas características físicas y de conexión, pues la tarjeta FOTON tiene un zócalo diseñado solo para este tipo de FPGA.
- El espacio que el carro tiene para moverse es limitada y puede no ser adecuada para el funcionamiento al probar otros controladores.

## Conclusiones

En este capítulo se compararon los diferentes resultados obtenidos del control del péndulo invertido, aun cuando los tres fueron capaces de lograr el objetivo de control, el controlador *tanh* sobresalió por sus resultados, en este capítulo solo se mostraron los resultados de una prueba por controlador pero se llevaron a cabo muchas pruebas más y se escogió la más representativa de cada uno. Además de comprobar el desempeño del controlador *tanh* para llevar las varillas hasta la posición deseada, se observó que este control es más fácil de programar en comparación con el PID que en LabVIEW requirió de un sub bloque y de varias variables a diferencia del *tanh* del cual solo se tiene que escribir su fórmula, sin duda el control *tanh* fue del que mejores resultados se obtuvieron.

# Conclusiones generales

## Conclusiones

Con el desarrollo de este trabajo de Tesis se logró la implementación del sistema mecatrónico tipo péndulo invertido sobre base móvil a nivel mecánico, hardware, firmware y software. En la parte mecánica se logró hacer más eficiente el sistema pues se encontraba con diferentes problemas, en la parte del hardware se implementan dos tipos de puentes H, uno con transistores BJT en configuración Darlington y otro con transistores MOSFET siendo este último el que mejor resultado brindó ya que presenta una pérdida de voltaje más baja que con transistores BJT, es decir que si se aplica un voltaje de 24V al motor solo tendrá una caída de 0.8V en el motor, sin en cambio sí se trabaja con BJT's la caída de voltaje es del doble y esto repercute en el desempeño del motor.

Por facilidad, para la fuente de alimentación que se encarga de la parte del control, se utiliza la fuente de alimentación de la computadora, tipo ATX, la cual proporciona voltajes de 3.3Vdc que son los que requiere el FPGA como voltaje nominal de trabajo, +/- 12V que se utilizan en el circuito de control del puente H para manejo de los transistores MOSFET, así como para alimentar el sensor de corriente. La fuente de alimentación de 24Vdc a 15 amperes se realiza con un transformador de estas características, y un puente de diodos capaz de manejar hasta 32A, se implementa un banco de capacitores para evitar que la fuente de alimentación tenga caídas de voltaje debido a la demanda de corriente del motor.

En la parte del firmware se desarrollaron los bloques involucrados para poder manipular los datos necesarios para el control del sistema, como son la lectura de las posiciones del carro y del péndulo así como la escritura del PWM y la dirección del motor, esto se realiza en un FPGA Cyclone II y Cyclone III, con esto se puede decir que el sistema se puede realizar en otros tipos de FPGA siempre y cuando se puedan adaptar a la tarjeta FOTON, debido a su configuración del PCI.

Se realiza la interfaz por medio de LabVIEW el cual es un software de tipo gráfico y demasiado intuitivo para su uso. Con este software se pueden realizar infinidad de instrucciones y el usuario puede agregar a quitar opciones de funcionamiento así como su manipulación del sistema. La interfaz quedo realizada en una sola pantalla lo que la hace muy fácil de manejar. Se pueden realizar modificaciones en cuanto a parámetros de funcionamiento para los controladores tipo P, PID y TANH.

Como aportaciones generadas con la realización de este trabajo de tesis tenemos: un sistema mecatrónico de tipo experimental en el cuál los estudiantes de la

maestría podrán realizar el estudio de comportamiento de diferentes tipos de controladores de este sistema y compararlos entre sí, esto ayudará a los estudiantes a comprender y comprobar la parte teórica del sistema, representándolo físicamente, ya que con la obtención del modelo cinemático, dinámico y su simulación solo lo representa en su forma teórica.

El desarrollo de este sistema es mucho más barato comparado con los que se comercializan, haciendo que este se pueda reproducir con poco presupuesto y poder tener más herramientas de estudio en el laboratorio de robótica.

Con el estudio de este tipo de controladores es posible aplicar el diseño de control de este sistema en otros como son, equilibrio de robots, vehículos, grúas, robots bípedos, aeronaves.

## **Trabajo a futuro**

A partir de este trabajo de tesis, se pueden realizar las siguientes actividades:

- Rediseñar el software para mejorar la flexibilidad del sistema en cuanto a las estrategias de control que requieran probarse.
- Implementar otros controladores para la fase de levantamiento del péndulo y así se tendrían dos modelos de estudio en vez de solo uno.
- Construir una base más alta para el péndulo invertido ya que la que tiene se encuentra a nivel de piso.
- Reproducir dicho sistema para tener al menos un sistema más para el laboratorio de robótica.

## Referencias

- [1] Hernández S., Gutiérrez J. E., Flores J. E., Morín M. M., *Diseño de un Controlador Óptimo Discreto para un Péndulo Invertido*, Facultad de Ciencias de la Electrónica, BUAP, Puebla, México, 2011
- [2] Aracil J., Gordillo F. *El Péndulo Invertido. Un Desafío para el Control no Lineal*. Revista Interamericana de Automática e Informática Industrial, Vol. 2, 2005.
- [3] Santamaria S. *Análisis y control del sistema dinámico péndulo invertido*. Tesis de Licenciatura. Facultad de ciencias de la electrónica, BUAP. Puebla, México, 2007.
- [4] Gunt HAMBURG, *Equipos para la educación en ingeniería, Mecatrónica, Automatización*.  
[http://www.gunt.de/static/s43\\_3.php?p1=&p2=&pN=actSetID;61](http://www.gunt.de/static/s43_3.php?p1=&p2=&pN=actSetID;61). Consultado en abril de 2015
- [5] Digital Pendulum 33-005-PCI, *FeedBack Datasheet*, [http://www.feedback-instruments.com/products/education/control\\_instrumentation/digital\\_pendulum](http://www.feedback-instruments.com/products/education/control_instrumentation/digital_pendulum). Consultado en enero de 2015
- [6] Spong M.W. "*The control of underactuated mechanical systems*" Int. conf. Mechatron, 1994.
- [7] Ogata K. "*Sistemas de control en tiempo discreto*", Prentice Hall. 1996.
- [8] Reyes F, "*Robótica: control de robots manipuladores*", Alfaomega, Primera edición. México. 2011.
- [9] Reyes F. "*Matlab: aplicado a robótica y mecatrónica*", Alfaomega, Primera edición. México 2012.
- [10] Hermosa A., "*Principios de Electricidad y Electrónica II* ", Capítulo 7: Rectificación de la corriente alterna, 1999.
- [11] International rectifier, *IR*, Datasheet IR2110 High and low side Driver, <http://www.irf.com/product-info/datasheets/data/ir2110.pdf>. Consultado en marzo de 2014.
- [12] Maloney, Tymothy J., *Electrónica industrial moderna*, Editorial Pearson Educación, 5ª. Edición, México, 2006.

- [13] Vargas Valencia J., Ramírez Velázquez I., Pérez Walton S., "*Física mecánica conceptos básicos y problemas*", Editorial ITM, Primera edición, 2008.
- [14] Honeywell, Datasheet *CSNE 151-005 sensor de corriente*, [http://sensing.honeywell.com/index.php?ci\\_id=49795](http://sensing.honeywell.com/index.php?ci_id=49795). Consultado en enero de 2015.
- [15] Balcells J., Romeral J. L., "*Autómatas Programables*", marcombo, 1997.
- [16] Autonics, Datasheet *encoder incremental series E50S* <http://www.farnell.com/datasheets/1673624.pdf>. Consultado en febrero de 2015.
- [17] Ginzburg M. C., "*Introducción general a la informática: la PC por dentro, arquitectura y funcionamiento de computadoras*". Ediciones. ISBN 987-43-7737-2, 2004.
- [18] López Aparicio J., "*Automatización del experimento de correlación de pares de fotones de conversión paramétrica descendente*" Tesis de Maestría en Ciencias de la electrónica opción Automatización, FCE-BUAP, 2013.
- [19] Mandado Pérez E., Mandado Rodríguez Y., "*Sistemas Electrónicos Digitales*", marcombo, 9ª edición.
- [20] Frank A. Scarpino, Ph.D., Frank Scarpino, "*Vhdl and Ahdl Digital System Implementation*", Prentice Hall International, 1998.
- [21] Ronald W. Larsen, "*Labview for Engineers*", Prentice Hall/Pearson, 2011.
- [22] Charles A. Alexander, Matthew N. O. Sadiku, "*Fundamentos de circuitos eléctricos*", McGraw Hill interamericana, 3ra. Edición 2006.
- [23] Prosser Franklin P., Winkel David E., "*The art of digital design*", Prentice-Hall, 1987.

## Apéndice A – Constancia Congreso CIINDET 2015





## Desarrollo de un sistema mecatrónico tipo péndulo invertido sobre base móvil.

M. Ramos Ortega, S. Vergara Limon, F. Reyes Cortes

**Resumen:** El péndulo invertido es un sistema que se puede utilizar para el estudio de la teoría de control, diseño y pruebas de diferentes controladores. En el desarrollo de este artículo se describe su estructura física, la caracterización del actuador, el modelado matemático del sistema, el diseño de la electrónica de potencia, la metodología del sistema de control, la implementación de una tarjeta de adquisición con un FPGA y su integración con el bus PCI, la interfaz de usuario mediante la cual se podrán manipular los parámetros del controlador.

Palabras Clave: Péndulo invertido, control, sub-actuado, motor, FPGA.

**Abstract:** The inverted pendulum is a system that can be used for the study of control theory, design and testing different drivers. In developing this article, it describes its physical structure, characterization actuator, mathematical modeling of the system, the design of power electronics, the methodology of the control system, the implementation of an acquisition card with FPGA and described integration with the PCI bus, the user interface through which you can manipulate the parameters of the controller.

Keywords: Inverted pendulum, control, sub-actuated, motor, FPGA.

### Introducción.

El péndulo invertido es uno de los ejemplos más conocidos de sistemas para estabilizar, es ampliamente estudiado por la comunidad científica y actualmente es un importante problema de control, ya que existen sistemas que se pueden aproximar a este modelo. El

interés de este trabajo es el desarrollar el sistema para uso del laboratorio de robótica y poder realizar pruebas de diferentes controladores. El péndulo invertido está constituido por un carro que tiene acoplado una barra (péndulo), que oscila libre y este a su vez se mueve sobre un eje (riel). El principal objetivo de este sistema es, a través de una ley de control, pasar de una posición vertical inferior de inicio ( $180^\circ$ ), a una posición vertical superior ( $0^\circ$ ) y mantenerse en esta última posición aun con la presencia de perturbaciones aplicadas al sistema. El sistema de péndulo invertido, como se mencionó anteriormente, tiene analogía con otros sistemas, por tal motivo se puede tomar como base en el diseño o pruebas de controladores como son: PID, algoritmos genéticos, lógica difusa, etc. Estos controladores se pueden utilizar en diferentes aplicaciones, algunas de ellas son: posicionamiento de cohetes y satélites, direccionamiento de grúas, el balance de articulaciones robóticas, como apoyo en el estudio de conceptos del control automático.

### Descripción del sistema.

El sistema del péndulo invertido está constituido por un carro el cual al aplicarle una fuerza que proviene de un motor se moverá de un lado hacia otro, sobre un riel, hasta que el par de varillas montadas en el carro se balanceen y logren estabilizarse en la posición deseada, en este caso  $0^\circ$ , esta estructura se muestra en la figura 1.

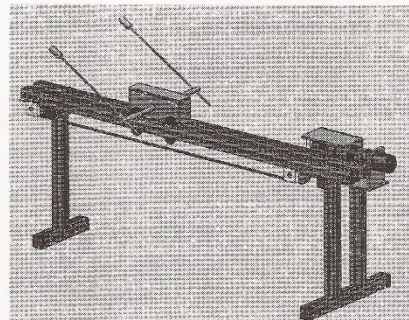


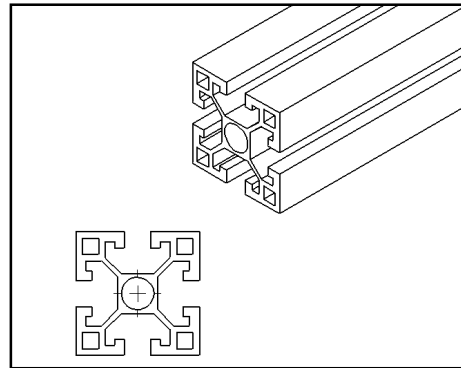
Fig. 1 Sistema péndulo invertido sobre base móvil.

Martin Ramos Ortega, Benemérita Universidad Autónoma de Puebla,  
Av. San Claudio y 18 Sur, Ciudad Universitaria, México,  
martin29ro@hotmail.com

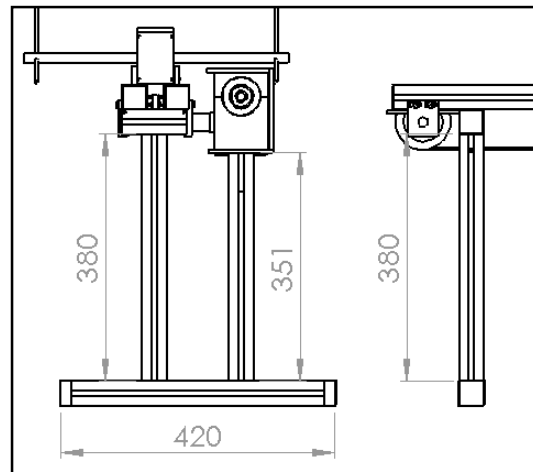
Sergio Vergara Limón, Benemérita Universidad Autónoma de Puebla,  
Av. san Claudio y 18 Sur, Ciudad Universitaria, México,  
svergara@ece.buap.mx

Fernando Reyes Cortez, Benemérita Universidad Autónoma de Puebla,  
Av. san Claudio y 18 Sur, Ciudad Universitaria, México,  
recf62@gmail.com

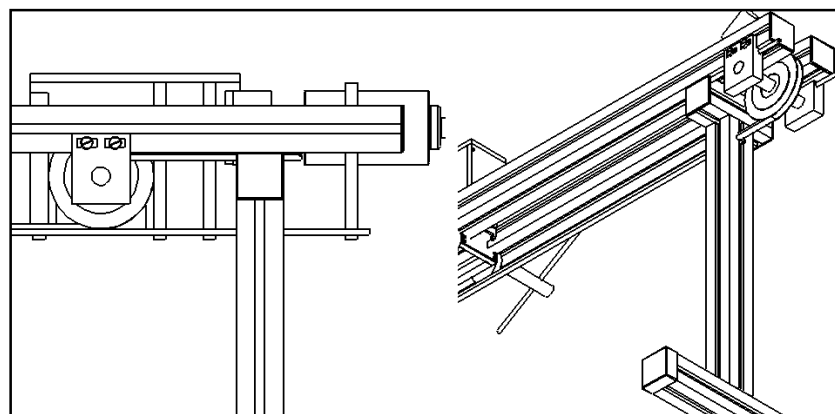
## Apéndice B – Diseño de sistema en SolidWorks.



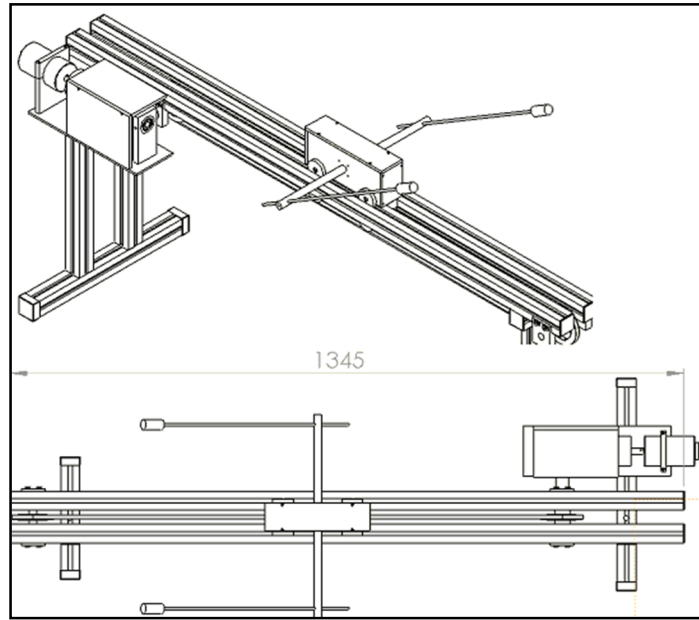
**Figura B.1:** Perfil del aluminio usado en la estructura del péndulo invertido



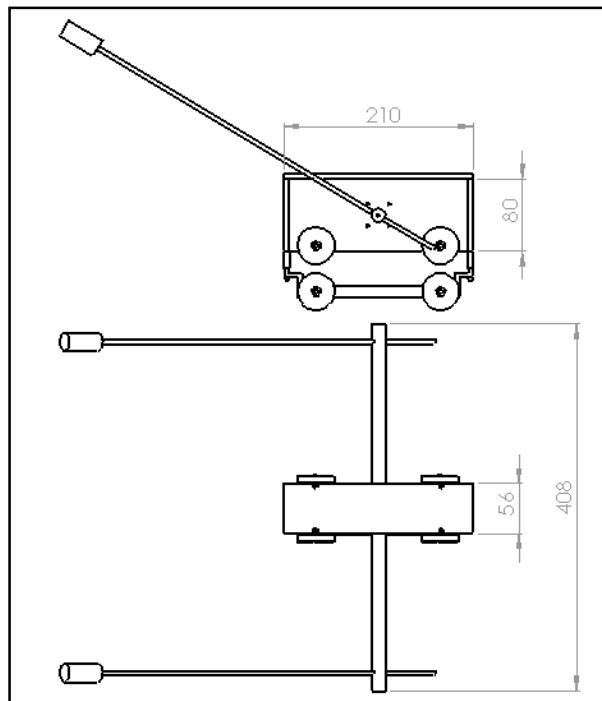
**Figura B.2:** Soportes del péndulo invertido.



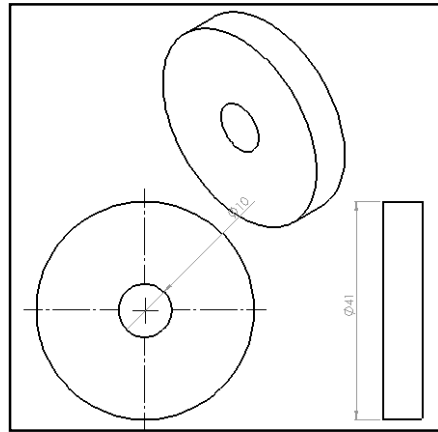
**Figura B.3:** Vista lateral e inferior de las poleas del péndulo invertido



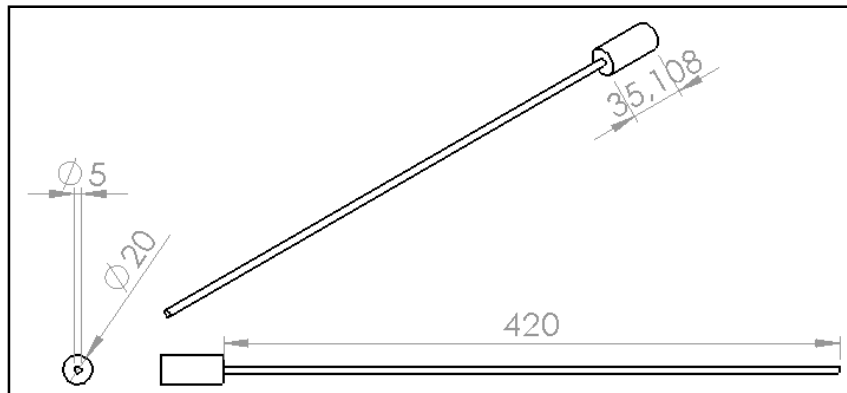
**Figura B.4:** Riel del péndulo invertido



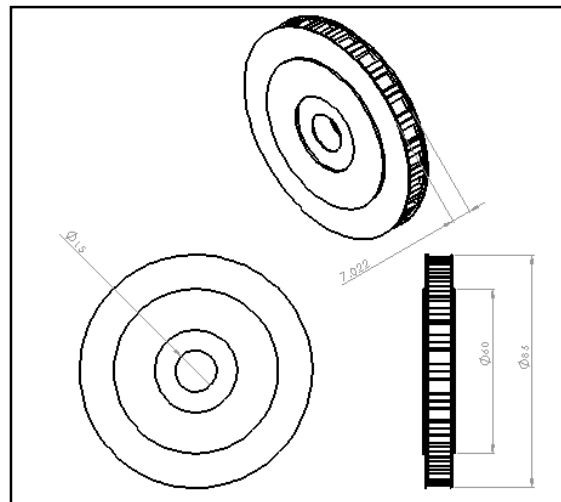
**Figura B.5:** Vista lateral y superior del carro



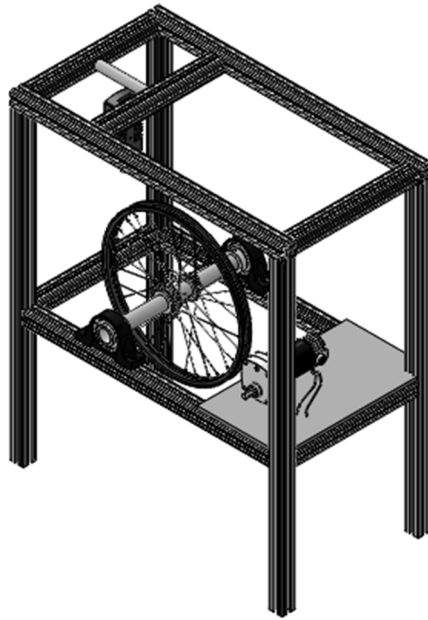
**Figura B.6:** Perfil de las llantas de los carritos



**Figura B.7:** Medidas de las varillas del péndulo



**Figura B.8:** Poleas para el péndulo invertido



**Figura B.9:** Diseño de torquímetro en SolidWorks.

# Apéndice C – Circuito eléctrico completo Puente H

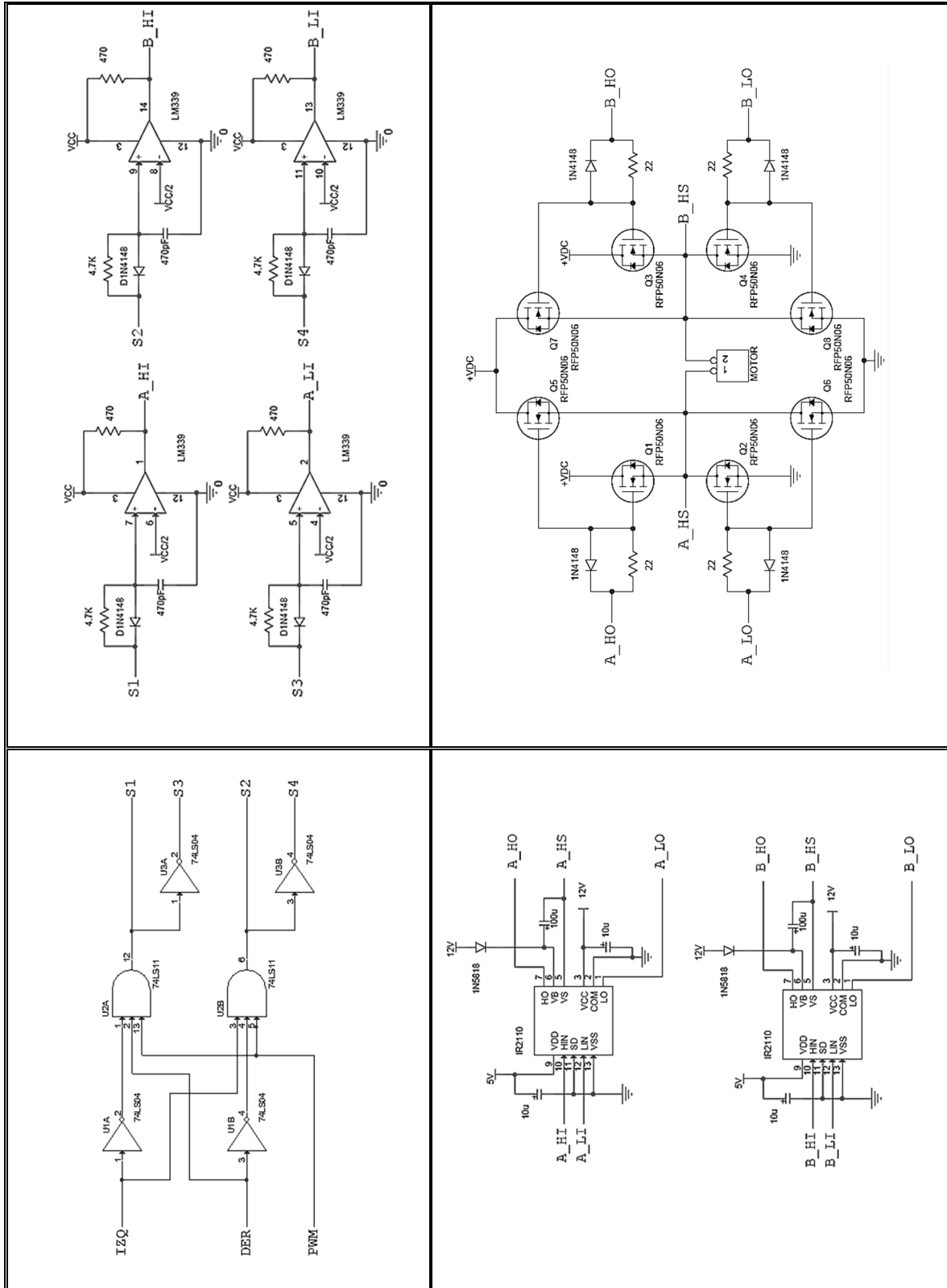


Figura C.1: Diagrama de conexión eléctrica de Puente H.

# Apéndice D – Diagrama de conexión eléctrica de Firmware.

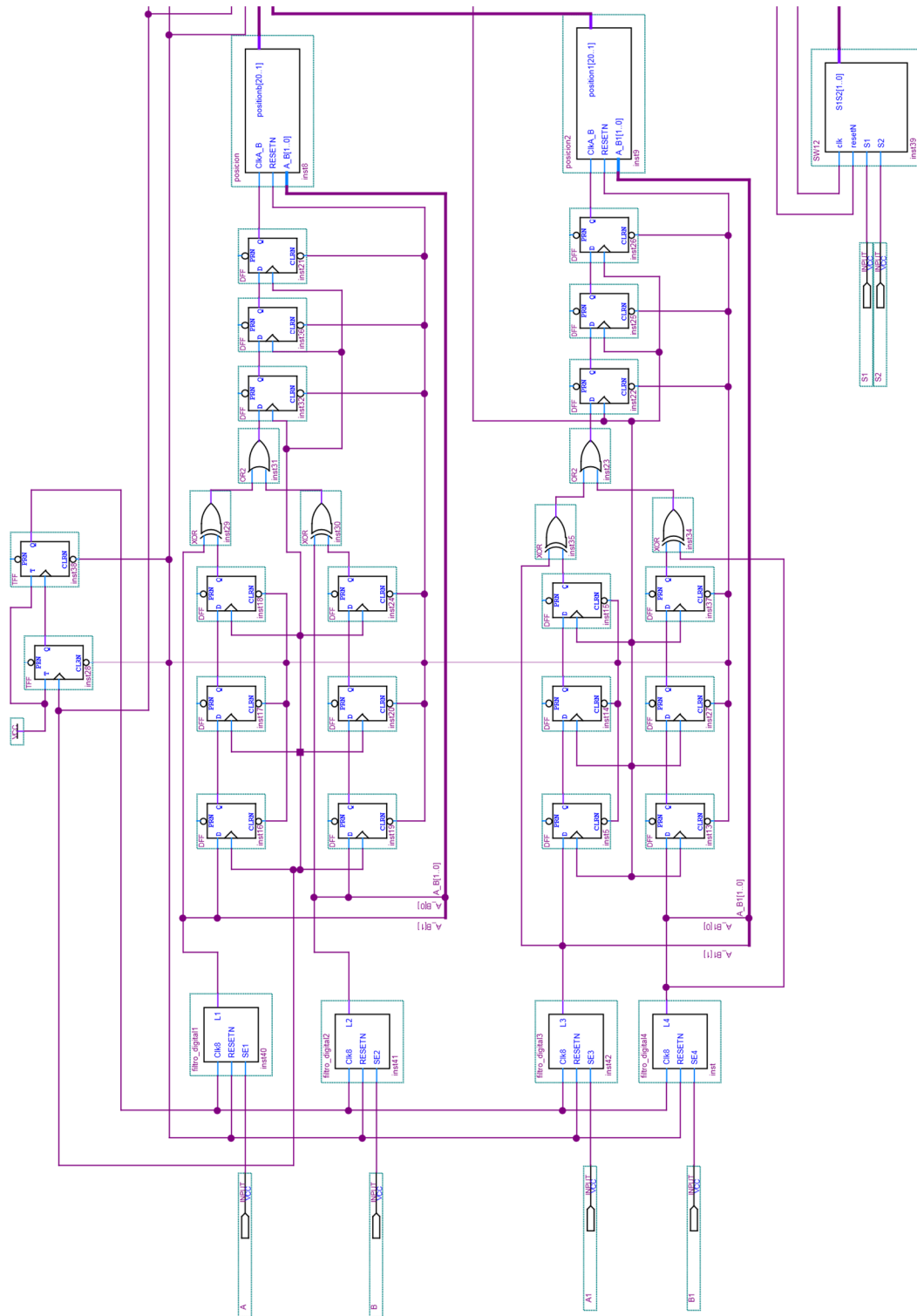
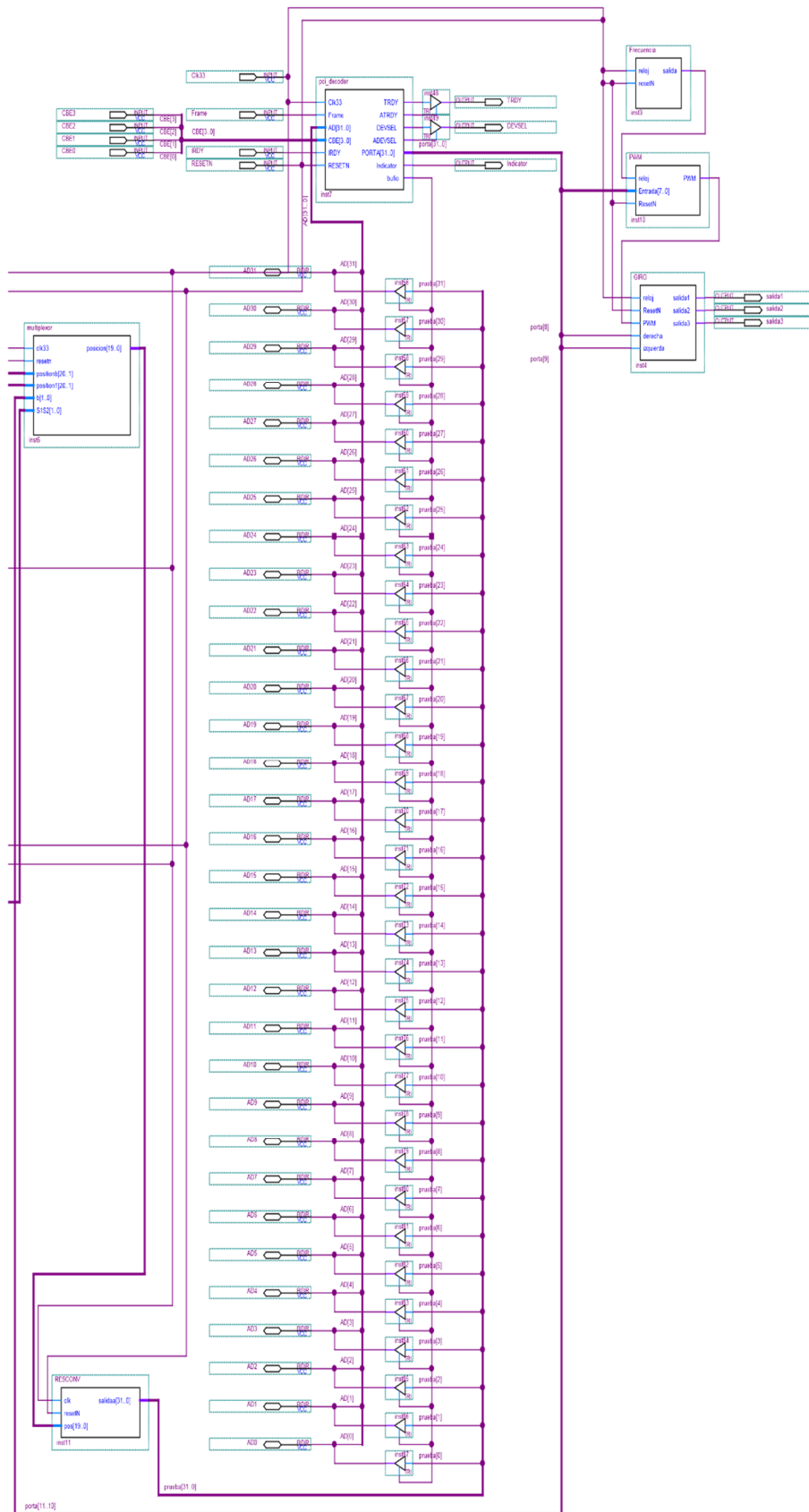


Figura D.1: Diagrama de conexión eléctrico del Firmware parte 1.



**Figura D.2:** Diagrama de conexión eléctrica del Firmware parte 2.