

DISEÑO Y CONSTRUCCIÓN DE UN PÉNDULO
INVERTIDO ROTATIVO COMO HERRAMIENTA DE
INVESTIGACIÓN Y DESARROLLO PARA EL
LABORATORIO DE SISTEMAS DINÁMICOS E
INGENIERÍA DE CONTROL

JHAIR OSWALDO MARTÍNEZ VILLAMIZAR



UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2012

DISEÑO Y CONSTRUCCIÓN DE UN PÉNDULO
INVERTIDO ROTATIVO COMO HERRAMIENTA DE
INVESTIGACIÓN Y DESARROLLO PARA EL
LABORATORIO DE SISTEMAS DINÁMICOS E
INGENIERÍA DE CONTROL

JHAIR OSWALDO MARTÍNEZ VILLAMIZAR

Trabajo de grado para optar al título de
Ingeniero Mecánico

Director

CARLOS BORRÁS PINILLA
Ingeniero Mecánico

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2012

DEDICATORIA

A Dios por darme el regalo de la vida, por iluminar mi camino en sendas opacas, por levantarme de mis tropiezos y reveses, por su amor y misericordia.

A la Virgen María por su maternal amor y protección y por ser mi intercesora ante mi señor Jesucristo.

A la memoria de mi padre Ciro Alfonso y mi abuelita María Praxedes, a quienes dedico esta meta y q comparten desde el cielo mi más profunda alegría por este logro.

A mi madre Myriam y a mis hermanos Uriel ylsleini por su apoyo, comprensión y acompañamiento durante estos años de formación profesional.

JHAIR OSWALDO MARTÍNEZ VILLAMIZAR

AGRADECIMIENTOS

A la Escuela de Ingeniería Mecánica y a la Universidad Industrial de Santander por todos estos años de enseñanza ética y profesional.

Al profesor Carlos Borrás, director del proyecto, por su asesoría y orientación.

Al ingeniero Raúl Hernando Cadena y el grupo de Electrónica y Robótica Aplicada (ERA) por su valiosa y constante asesoría.

A todos aquellos que de una u otra forma aportaron con su consejo u observación durante el desarrollo de este proyecto.

TABLA DE CONTENIDO

INTRODUCCION	17
ANTECEDENTES.....	19
1. MARCO TEORICO.....	21
1.1 MOTOR DC [11]	21
1.1.1 Rotor.....	22
1.1.2 Estator	24
1.2 PUENTE H	25
1.3 MODULACION POR ANCHO DE PULSO (PWM).....	27
1.4 MICROCONTROLADOR	28
1.4.1 Estructura de una microcomputadora.....	29
1.4.2 Buses	30
1.4.3 CPU.....	31
1.4.4 Memoria	32
1.4.5 ROM.....	33
1.4.6 PROM.....	33
1.4.7 EPROM	34
1.4.8 EEPROM.....	34
1.4.9 RAM	34
1.4.10 ENTRADA/SALIDA.....	35
2. FORMULACION DEL PROBLEMA.....	36
2.1 IDENTIFICACION DEL PROBLEMA.....	36
2.2 JUSTIFICACION PARA LA SOLUCION DEL PROBLEMA	37
2.3 OBJETIVOS DEL TRABAJO DE GRADO	38
2.3.1 Objetivo General	38
2.3.2 Objetivos específicos.....	39
3. DESCRIPCION DEL PENDULO INVERTIDO ROTATIVO COMO SISTEMA DE CONTROL	41
3.1 REQUERIMIENTOS DE DISEÑO	43
3.2 PARTE OPERATIVA	43
3.2.1 Operación “swing up”	45

3.2.2 Operación de balanceo	45
3.2.3 Planta Dinámica	46
3.2.4 Sensores	47
3.2.5 Actuador	53
3.3 PARTE DE CONTROL	55
3.3.1 Controlador.....	56
3.3.2 Pre-actuador.....	57
3.3.3 Integrado LMD18200T.....	58
4. MODELO MATEMATICO DEL PENDULO INVERTIDO ROTATIVO.....	61
4.1 ECUACIONES DE MOVIMIENTO DEL PENDULO INVERTIDO ROTATIVO	61
4.1.1 Ecuaciones de movimiento de Lagrange.....	62
4.1.2 Obtención del Lagrangiano L	64
4.1.3 Modelo matemático del motor	67
4.1.4 Linealización del modelo	68
5. CONTROL PID	70
5.1 ACCIONES DE CONTROL	70
5.1.1 Controlador proporcional.....	70
5.1.2 Controlador Integral.....	71
5.1.3 Controlador Proporcional-Integral (PI).....	72
5.1.4 Controlador Proporcional-Derivativo (PD)	73
5.1.5 Controlador Proporcional-Integral—Derivativo	74
6. IMPLEMENTACION.....	76
6.1 PARÁMETROS DE CONTROL	76
6.2 IMPLEMENTACION DE LOS SENSORES	76
6.2.1 Ubicación.....	76
6.2.2 Interface de sensores	78
6.3 IMPLEMENTACIÓN PUENTE H	80
6.3.1 Etapa de potencia	80
7. DISEÑO ESTRUCTURAL DEL PENDULO INVERTIDO ROTATIVO.....	83
7.1 Estructura soporte motor	83
7.2 Sistema de acople motor-brazo	86

7.3 Sistema de acople encoder-péndulo	88
8. SISTEMA DE CONTROL.....	91
8.1 SISTEMA DE CONTROL EMBEBIDO.....	91
8.1.1 Principales características de los sistemas embebidos.....	91
8.2 MICROCONTROLADOR ATMEGA 2560 DE ATMEL	92
8.3 TARJETA ARDUINO MEGA 2560	93
8.3.1 Alimentación del Arduino.....	94
8.3.2 Descripción de pines	95
8.4 PROGRAMACIÓN DEL ARDUINO	97
8.5 PROGRAMACIÓN DEL MICROCONTROLADOR	97
8.5.1 Entorno de programación de Arduino (IDE)	97
8.5.2 Funciones básicas.....	98
8.5.3 Estructura del programa	99
8.5.4 Registros de control.....	99
8.6 DISEÑO DEL ALGORITMO DE CONTROL	100
8.6.1 Lectura del encoder.....	100
8.6.2 Control del puente H LMD18200	103
8.7 PROGRAMACIÓN DEL CONTROLADOR PID BASADO EN ISR ().....	106
8.7.1 Configuración de la ISR ()	106
8.8 MODO SWING UP	108
8.9 MODO DE ESTABILIZACIÓN	108
8.9.1 Controlador PID.....	108
8.9.2 Interfaz con Simulink	110
8.10 PANEL DE CONTROL	114
9. PRUEBAS DEL CONTROLADOR PID	116
10. MANUAL DE OPERACIÓN.....	125
11. CONCLUSIONES	131
12. RECOMENDACIONES.....	133
BIBLIOGRAFIA.....	134
ANEXOS.....	136

LISTA DE FIGURAS

Figura 1. Configuración de péndulos invertidos: (a) horizontal y (b) Vertical. ...	19
Figura 2. Péndulo invertido fabricado por Quanser.....	20
Figura 3. (a) Rotor, (b) Estator	21
Figura 4. Rotor.	22
Figura 5. Rotor	23
Figura 6. Estator	24
Figura 7. Esquema básico de un circuito puente H.....	25
Figura 8. Rotación en sentido horario del motor.	26
Figura 9. Rotación en sentido anti-horario del motor.	26
Figura 10. Señal de onda cuadrada de amplitud acotada ($y_{max} - y_{min}$) mostrando el ciclo de trabajo.	27
Figura 11. Diagrama de bloques de un microcontrolador.	29
Figura 12. Configuración general de una computadora	30
Figura 13. Dispositivos de una CPU	32
Figura 14. Componentes básicos del péndulo: a) motor, b) brazo, c) péndulo.	41
Figura 15. Ubicación de los sensores de medición.....	42
Figura 16. Diagrama de bloque de un sistema de control en lazo cerrado.	42
Figura 17. Disposición de los elementos	44
Figura 18. Posición vertical del péndulo.	46
Figura 19. Planta dinámica de pruebas.	46
Figura 20. Esquema básico de un encoder óptico.	48
Figura 21. Encoder en cuadratura.	49
Figura 22. Encoder incremental TRD-S2500-BD	51
Figura 23. Encoder incremental HEDS-9100	52
Figura 24. Actuador	55
Figura 25. Componente de control.....	55
Figura 26. Arduino Mega2560 y microcontrolador ATMEGA2560	56
Figura 27. Esquema básico de un puente H.....	57
Figura 28. Puente H LMD18200	58
Figura 29. Pines de control del LMD18200T.....	59
Figura 30. Modelo simplificado	63
Figura 31. Diagrama cinemático del péndulo.....	65
Figura 32. Diagrama de polos y ceros.	69
Figura 33. Diagrama de bloques de un control proporcional	71
Figura 34. Diagrama de bloque de un controlador integral	72
Figura 35. (a) Diagrama de bloques de un controlador PI; (b) Diagrama que muestra una entrada escalón; (c) salida del controlador	73

Figura 36. (a) Diagrama de bloque de un controlador PD; (b) y (c) diagramas que muestran una entrada rampa unitaria y la salida del controlador.....	74
Figura 37. (a) Diagrama de bloques de un controlador PID; (b) y (c) diagramas que muestran una entrada de rampa unitaria y la salida del controlador PID...	75
Figura 38. Ubicación del sensor en el motor.....	77
Figura 39. Acoplamiento del eje del motor con al brazo	77
Figura 40. Ubicación del sensor del péndulo	78
Figura 41. Encoder PCB	78
Figura 42. (a) Esquemático (b) negativo.....	79
Figura 43. (a) Señal encoder motor o brazo, (b) señal encoder péndulo	80
Figura 44. Puente H PCB	81
Figura 45. (a) Esquema eléctrico tarjeta puente H, (b) negativo PCB	81
Figura 46. Planta dinámica Péndulo Invertido Rotativo	83
Figura 47. Subconjuntos del péndulo invertido rotativo.	84
Figura 48. (a) Estructura base, (b) Espigo, (c) Disco	85
Figura 49. Acabado final de la plataforma	86
Figura 50. Espigo roscado	86
Figura 51. Vista en corte sistema de acople motor-brazo.....	87
Figura 52. Acople motor- brazo.....	87
Figura 53. Sistema de acople encoder-péndulo	88
Figura 54. Eje péndulo.....	89
Figura 55. Diagramas de carga, cortante y momento del eje	89
Figura 56. Distribución de esfuerzos del eje	90
Figura 57. Tarjeta Arduino Mega 2560.	94
Figura 58. Diagrama explicativo para leer encoder.....	102
Figura 59. Frecuencia de la ISR	107
Figura 60. Zonas de trabajo.....	109
Figura 61. Interfaz SIMULINK.....	110
Figura 62. Contenido de la ISR.....	112
Figura 63. Doble lazo de control	113
Figura 64. Tareas en el loop del arduino	113
Figura 65. Conexión a la tarjeta.....	114
Figura 66. Caja de control.....	115
Figura 67. Gráficas prueba No.1.....	117
Figura 68. Prueba No. 2.....	118
Figura 69. Prueba No. 3.....	119
Figura 70. Prueba No. 4.....	120
Figura 71. Prueba No. 5.....	121
Figura 72. Prueba No. 6.....	122
Figura 73. Prueba No. 7.....	123
Figura 74. Prueba No. 8.....	124
Figura 75. Conexión eléctrica	126

Figura 76. Conexión del arduino	127
Figura 77. Apertura del programa de control PID	127
Figura 78. Cargado del programa a la tarjeta	128
Figura 79. Apertura de la interfaz Arduino-Simulink.....	129
Figura 80. Configuración de la interfaz	129
Figura 81. Encendido de la caja de control.....	130

LISTA DE TABLAS

Tabla 1. Longitud de un bus de datos de cuatro bits	30
Tabla 2. Características del encoder incremental TRD-S2500-BD	52
Tabla 3. Características del encoder incremental HEDS-9100	53
Tabla 4. Características del motor.	54
Tabla 5. Características Arduino MEGA	56
Tabla 6. Combinación pines de control del LMD18200.....	59
Tabla 7. Parámetros del modelo	68

LISTA DE ANEXOS

ANEXO A. Resumen hoja de datos encoder incremental	137
ANEXO B. Resumen hoja de datos Puente H LMD18200	140
ANEXO C. Hoja de Datos inversor 74LS04	142
ANEXO D. Optoaislador 4N25	144
ANEXO E. Optoaislador 6N137	148
ANEXO F. Configuración Timer 3 y 4 (hoja de datos)	151
ANEXO G. Código Fuente en Arduino.	159
ANEXO H. Planos mecánicos	175

RESUMEN

TITULO: DISEÑO Y CONSTRUCCIÓN DE UN PÉNDULO INVERTIDI ROTATIVO COMO HERRAMIENTA DE INVESTIGACIÓN Y DESARROOLLO PARA EL LABORATORIO DE SISTEMAS DINÁMICOS E INGENIERÍA DE CONTROL*

AUTOR: JHAIR OSWALDO MARTINEZ VILLAMIZAR**

PALABRAS CLAVES: motor de corriente continua, encoder óptico incremental, microcontrolador, auto-elevamiento, control PID, Simulink.

DESCRIPCIÓN:

El péndulo invertido rotativo es un sistema altamente no lineal e inestable, ha sido estudiado ampliamente para probar técnicas de control tanto clásicas como modernas. También se conoce como péndulo de Furuta. K. Furuta fue el inventor del mismo. Es una buena herramienta de enseñanza que permite comprender mejor las técnicas de control usadas en la industria. Es la primera vez que se diseña ésta clase de péndulo en la Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander, con el fin de contribuir a una mejor formación del estudiante en el área de la robótica. Es un sistema sub-actuado de dos grados de libertad. Está compuesto por un motor de corriente continua, dos encoders ópticos incrementales y una tarjeta de desarrollo llamada Arduino, basada en un microcontrolador ATmega2560. Se implementó un doble lazo de control PID que permite controlar la posición del brazo y el péndulo en su posición invertida. El sistema es capaz de resistir perturbaciones. Adicionalmente se diseñó un control de auto-elevamiento para el péndulo. Se realizaron pruebas para el controlador PID con el fin de evidenciar el efecto de las ganancias sobre la respuesta en el tiempo del péndulo invertido rotativo en su posición invertida. Cuenta con un módulo LCD que permite visualizar el valor del ángulo del brazo y del péndulo. Se implementó una interfaz con Simulink para ver en tiempo real la posición del brazo y el péndulo. También se diseño un manual de operación para llevar a cabo el experimento.

* Trabajo de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería Mecánica. Director: Ing. Carlos Bórras

SUMMARY

TITLE: DESIGN AND CONSTRUCTION OF A ROTARY INVERTED PENDULUM AS A RESEARCHING AND DEVELOPING TOOL FOR DYNAMIC SYSTEMS AND CONTROL ENGINEERING LABORATORIES*

AUTHOR: JHAIR OSWALDO MARTINEZ VILLAMIZAR**

KEY WORDS: DC motor, optical incremental encoder, microcontroller, swing up, PID control, Simulink.

DESCRIPTION:

The rotary inverted pendulum is a highly nonlinear and unstable system. It has been widely studied for testing classical and modern control techniques. It is also known as Furuta pendulum. K. Furuta was who invented. It is a good teaching tool which allows to get a better understanding about control techniques used at industry. This is the first time this kind of pendulum is designed and built in Mechanical Engineering School in Universidad Industrial de Santander. The purpose is to contribute to students training at robotic field. It is an underactuated system of two degrees of freedom. The inverted pendulum is made up by a DC motor, two optical incremental encoders and an open source board based on an Atmega2560 microcontroller. A double PID control loop was implemented, which allows to control arm position and the pendulum on its inverted position. The system is able to overcome external disturbances. In addition, a swing up control was designed. Controller tests were made with the purpose of analyzing the time response effects of control gains over the pendulum on its upright inverted position. An LCD module allows visualize arm and pendulum angle values. A Simulink interface was implemented to watch real time arm and pendulum position. A user guide also was made to carry out the experiment.

* Senior thesis

** Faculty of Physical-Mechanical Engineering, School of Mechanical Engineering. Director: Eng. Carlos Bórras

INTRODUCCION

El control de sistemas mecánicos sub-actuados [1] ha atraído el interés de la comunidad de los ingenieros de control por muchos años. Más aun, el control de dichos sistemas es actualmente un campo activo de investigación debido a su amplia aplicación en robótica, vehículos marinos y aeroespaciales. Algunos ejemplos de estos sistemas incluyen robots móviles, de plataforma, autos, sistemas de locomoción, aeronaves, helicópteros, satélites, vehículos subacuáticos.

Uno de los sistemas mecánicos sub-actuados más conocidos es el péndulo invertido, el cuál es un excelente punto de referencia para muchos problemas de control, atrayendo la atención de muchos investigadores durante años. El péndulo invertido rotativo es conocido como el péndulo de Furuta. K. Furuta fue el inventor y su péndulo ha sido estudiado por muchos autores. El péndulo de Furuta es un sistema mecánico sub-actuado no lineal el cual es inestable en la posición vertical deseada.

En efecto el péndulo muestra, principalmente, dos diferentes y muy interesantes problemas de control. El primero, es hacer oscilar el péndulo hacia una posición cercana a la vertical (arriba). Una vez que el péndulo está cerca de la posición vertical deseada, a baja velocidad, aparece el segundo problema de control, mantener el péndulo en la posición vertical (arriba), el cual se resuelve mediante alguna estrategia de estabilización. Para el presente trabajo de grado se implementará una estrategia de control PID (Proporcional Integral y Derivativa) ya que es el primer paso para introducirse en el universo de las técnicas y métodos de control, y teniendo en cuenta que el control PID es el más ampliamente utilizado en la industria moderna, controlando más del 95% de los procesos industriales en lazo cerrado, además dicho controlador también se destaca por la simplicidad de su estructura, celeridad en la respuesta y por ser uno de los métodos más precisos de la regulación automática.

Se recurrió a la tarjeta de desarrollo Arduino y a su entorno de programación

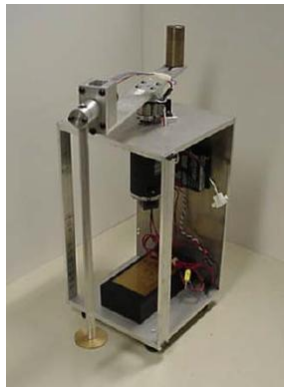
por su sencillez a la hora de programar y ejecutar algoritmos, así como su facilidad para la obtención de datos vía sensor. Una serie de pruebas serán realizadas con el fin de evidenciar el efecto de las acciones del controlador PID en el control de la planta.

ANTECEDENTES

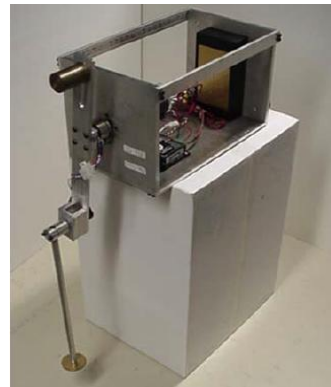
El péndulo de Furuta es una excelente plataforma de experimentación para la enseñanza de las técnicas de control. Muchas universidades en cuyo programa de formación de ingeniería mecánica hay una asignatura de control, cuentan con un banco de experimentación del péndulo de Furuta. El péndulo invertido rotativo es una popular aplicación mecatrónica, existen diferentes formas de configuración del sistema [6] (ver figura 1), pero todas tienen el objetivo en común de balancear un péndulo en el extremo usando alguna técnica de control en lazo cerrado. El péndulo invertido ha sido tema de estudio, no solo en programas de formación de pregrado, sino también de maestrías [2].

Figura 1. Configuración de péndulos invertidos: (a) horizontal y (b) Vertical.

(a)



(b)



Fuente. [6]

Existen varias estrategias de control para auto-elevar y balancear el péndulo las más ampliamente utilizadas como en [2,7,8] son el control en el espacio de estados y LQR (Linear Quadratic Regulator), y para el swing up, el uso de estrategias de control de energía [9] es el método más comúnmente utilizado.

Quanser [10] es el líder mundial en el diseño y fabricación e implementación de sistemas avanzados de control en tiempo real, posee un módulo de péndulo de Furuta como el de la figura 2.

Figura 2. Péndulo invertido fabricado por Quanser.



Fuente: (www.quanser.com)

En este, la estrategia de control de “swing up” implementada es un control PD (proporcional y derivativo) mientras que el control de balance del péndulo se diseña un controlador en el espacio de estados usando el regulador cuadrático lineal (LQR) y el modelo lineal del sistema.

1. MARCO TEORICO

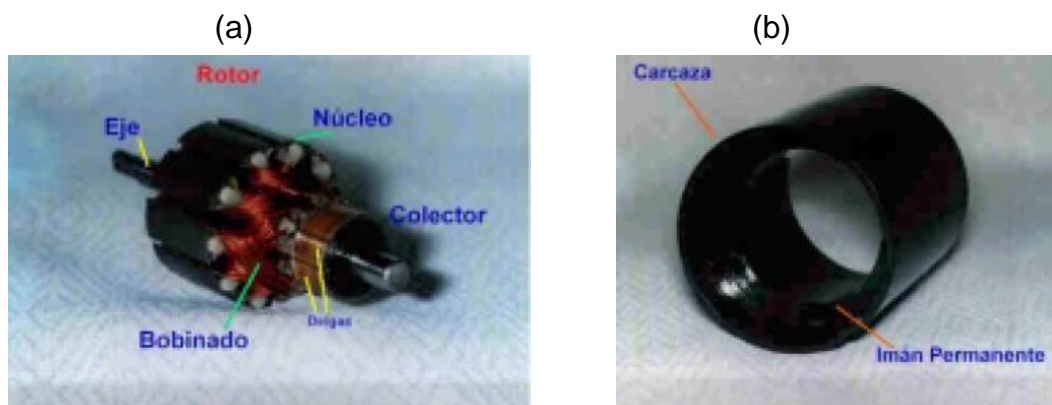
1.1 MOTOR DC [11]

El motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica, principalmente por movimiento rotativo. Esta máquina de corriente continua es una de las más versátiles en la industria y especialmente en el campo de la robótica. Su fácil control de posición, de paro y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos.

Accionar un motor DC (por sus siglas en inglés, Direct Current) es muy simple y solo es necesario aplicar la tensión de alimentación entre sus bornes. Para invertir el sentido de giro basta con invertir la alimentación y el motor comenzará a girar en sentido opuesto.

El motor DC está compuesto de dos piezas fundamentales las cuales se muestran en la Figura 3:

Figura 3. (a) Rotor, (b) Estator



Fuente. www.todorobot.com.ar

Dentro de éstas se ubican los demás componentes como:

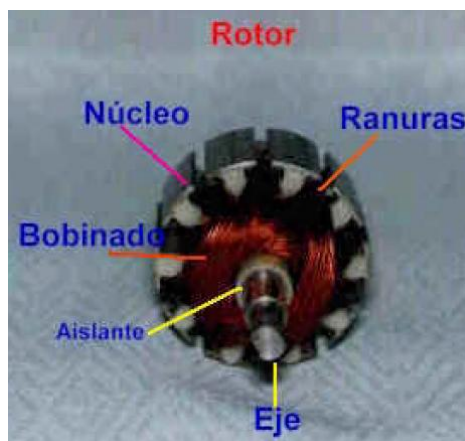
- Escobillas y portaescobillas.

- Colector.
- Eje.
- Núcleo y devanado del rotor.
- Imán permanente.
- Armazón.
- Tapas o campanas.

1.1.1 Rotor

Constituye la parte móvil del motor, proporciona el torque para mover la carga.
Ver figura 4.

Figura 4. Rotor.



Fuente: www.todorobot.com.ar

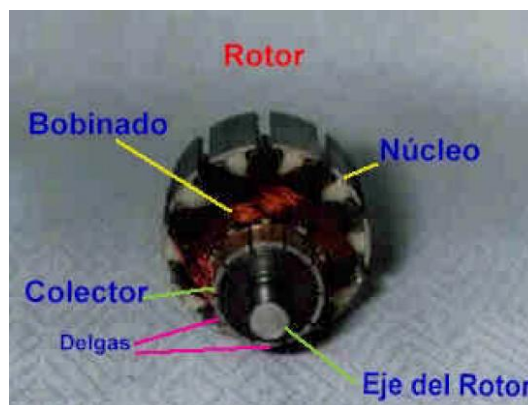
Está formado por:

- **Eje:** Formado por una barra de acero fresada. Imparte la rotación al núcleo, devanado y al colector.
- **Núcleo:** Se localiza sobre el eje. Fabricado con capas de láminas de acero, su función es proporcionar un trayecto magnético entre los polos para que el flujo magnético del devanado circule.

Las laminaciones tienen por objeto reducir las corrientes parásitas en el núcleo. El acero del núcleo debe ser capaz de mantener bajas las pérdidas por histéresis. Este núcleo laminado contiene ranuras a lo largo de su superficie para albergar el devanado de la armadura (bobinado).

- **Devanado:** Consta de bobinas aisladas entre sí y entre el núcleo de la armadura. Estas bobinas están alojadas en las ranuras, y están conectadas eléctricamente con el colector, el cual debido a su movimiento rotatorio, proporciona un camino de conducción conmutado.
- **Colector:** Denominado también conmutador, está constituido de láminas de material conductor (delgas), separadas entre sí y del centro del eje por un material aislante, para evitar cortocircuito con dichos elementos. El colector se encuentra sobre uno de los extremos del eje del rotor, de modo que gira con este y está en contacto con las escobillas. La función del colector es recoger la tensión producida por el devanado inducido, transmitiéndola al circuito por medio de las escobillas (llamadas también cepillos), ver figura 5.

Figura 5. Rotor



Fuente: www.todorobot.com.ar

1.1.2 Estator

Constituye la parte fija de la máquina. Su función es suministrar el flujo magnético que será usado por el bobinado del rotor para realizar su movimiento giratorio.

Está formado por:

- **Armazón:** Denominado también yugo, tiene dos funciones primordiales: servir como soporte y proporcionar una trayectoria de retorno al flujo magnético del rotor y del imán permanente, para completar el circuito magnético. Ver figura 6.
- **Imán permanente:** Compuesto del material ferromagnético altamente remanente, se encuentra fijado al armazón o carcasa del estator. Su función es proporcionar un campo magnético uniforme al devanado del rotor o armadura, de modo que interactúe con el campo formado por el bobinado, y se origine el movimiento del rotor como el resultado de la interacción de estos campos. Ver figura 6.

Figura 6. Estator



Fuente. www.todorobot.com.ar

- **Escobillas:** Las escobillas están fabricadas en carbón, y poseen una dureza menor que la del colector, para evitar que este se desgaste rápidamente. Se encuentran albergadas por los portaescobillas. Ambos,

escobillas y portaescobillas, se encuentran en una de las tapas del estator.

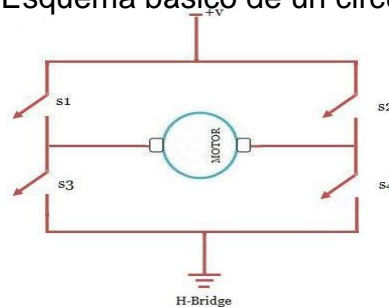
La función de las escobillas es transmitir la tensión y la corriente de la fuente de alimentación hacia el colector y, por consiguiente, al bobinado del rotor.

La función del portaescobillas es mantener a las escobillas en su posición de contacto firme con los segmentos del colector. Esta función la realiza por medio de resortes, los cuales hacen una presión moderada sobre las escobillas contra el colector. Esta presión debe mantenerse en un nivel intermedio pues, de ser excesiva, la fricción desgastaría tanto a las escobillas como al colector; por otro lado, de ser mínima esta presión, se produciría lo que se denomina “chisporroteo”, que cuando aparecen chispas entre las superficies del colector y las escobillas, debido a que no existe un buen contacto.

1.2 PUENTE H

El puente H es llamado así porque posee cuatro elementos conmutables o interruptores en las esquinas de una H, y el motor forma la línea transversal de la H. en la figura 7 se aprecia esta clase de circuito. Es un hecho conocido que la letra H no tiene la parte superior ni la inferior juntas, con un poco de surte al observar la figura 7 se puede apreciar dicha forma.

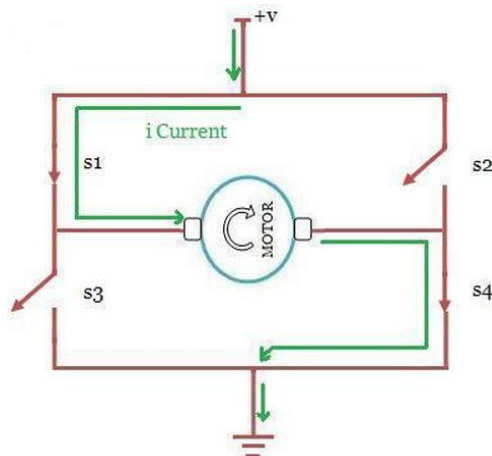
Figura 7. Esquema básico de un circuito puente H.



Fuente: www.botskool.com/tutorials/electronics/general-electronics/motor-controller-h-bridge.

Cuando todos los interruptores están abiertos, es decir que no están haciendo contacto, el motor no recibe voltaje ni corriente y por lo tanto este no girará. En la figura 8, cuando los interruptores S2 y S3 están abiertos, y S1 y S4 están cerrados, a través de los últimos se crea una diferencia de potencial a través del motor y simultáneamente corriente fluye por el circuito la cual hace girar el eje del motor, digamos que en sentido horario como se aprecia en la figura 8.

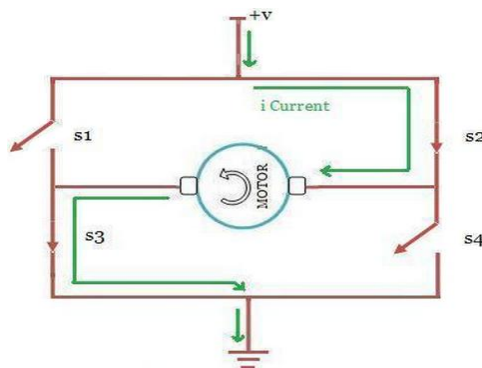
Figura 8. Rotación en sentido horario del motor.



Fuente: www.botskool.com/tutorials/electronics/general-electronics/motor-controller-h-bridge

Similarmente, cuando los interruptores S1 y S4 están abiertos, y S2 y S3 están cerrados el flujo de corriente hará rotar el motor en sentido anti-horario. Ver figura 9.

Figura 9. Rotación en sentido anti-horario del motor.



Fuente. www.botskool.com/tutorials/electronics/general-electronics/motor-controller-h-bridge

Existen muchas configuraciones de circuitos puente H, pero el concepto de funcionamiento es el mismo. Es el caso del IC (Integrated Circuit, por sus siglas en inglés) LMD18200 que será explicado en mayor detalle en el capítulo 2.

1.3 MODULACION POR ANCHO DE PULSO (PWM)

La modulación por ancho de pulso o PWM por sus siglas en inglés (Pulse Width Modulation) es una poderosa técnica en la que se modifica el ciclo de trabajo de una señal periódica (por ejemplo senoidal o cuadrada) ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el periodo. Ver figura 10. Expresado matemáticamente:

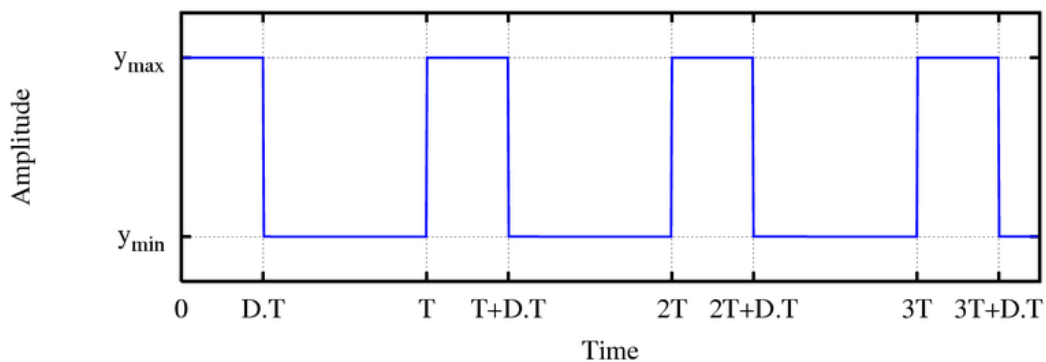
$$D = \frac{\tau}{T}$$

D es el ciclo de trabajo

τ es el tiempo en el que la función es positiva (ancho de pulso)

T es el periodo de la función

Figura 10. Señal de onda cuadrada de amplitud acotada (y_{\max} - y_{\min}) mostrando el ciclo de trabajo.



La modulación por ancho de pulso 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 alimentación) y un momento bajo (apagado o desconectado), controlado normalmente por relevadores (baja frecuencia) o MOSFET [4] (alta frecuencia). Otra forma de regular el giro del motor es variado el tiempo entre pulsos de duración constante, lo que se conoce como modulación por frecuencia de pulsos.

El PWM también se usa para controlar servomotores, los cuales modifican su posición de acuerdo al ancho del pulso enviado cada un cierto periodo que depende de cada servomotor. Esta información puede ser enviada utilizando un microprocesador o un microcontrolador.

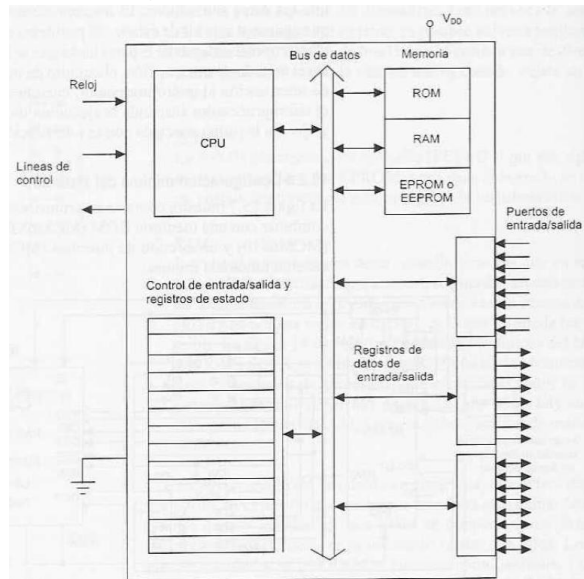
La principal desventaja que presentan los circuitos PWM es la posibilidad de que haya interferencias generadas por radiofrecuencias. Éstas pueden minimizarse ubicando el controlador cerca de la carga y realizando un filtrado cerca de la fuente de alimentación.

1.4 MICROCONTROLADOR

Para que un microprocesador pueda funcionar como un sistema de microcomputadora aplicando al control, son necesarios chips adicionales, por ejemplo, dispositivos de memoria para almacenar programas y datos, así como puertos de entrada/salida para permitir que se comunique con el mundo externo y reciba señales desde éste.

El microcontrolador consiste en la integración en un chip de un microprocesador con memoria, interfases de entrada/salida y otros dispositivos periféricos como temporizadores. En la figura 11 muestra un diagrama de bloques general de un microcontrolador.

Figura 11. Diagrama de bloques de un microcontrolador.



Fuente. [3]

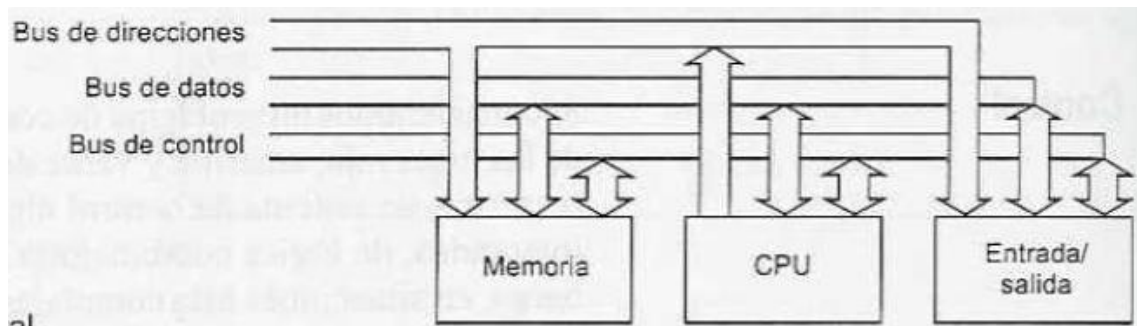
Un microcontrolador común tiene terminales para la conexión externa de entradas y salidas, alimentación eléctrica y señales de reloj y de control. Las conexiones de entradas y salidas se agrupan en unidades denominadas puertos de entrada/salida. Por lo general, estos puertos tienen ocho líneas para poder transportar una palabra de datos de 8 bits. Para una palabra de 16 bits se utilizan dos puertos, uno para transmitir los 8 bits inferiores, y el otro para los 8 bits superiores. Los puertos pueden ser sólo de entrada o de salida, o bidireccionales para que funcionen tanto de entrada como de salida.

1.4.1 Estructura de una microcomputadora

Las computadoras constan de tres secciones: la unidad central de proceso (CPU, por sus siglas en inglés), la cual reconoce y ejecuta las instrucciones de un programa; los circuitos de interfase de entrada y salida, los cuales controlan las comunicaciones entre la computadora y el mundo exterior, y la memoria, donde se almacenan las instrucciones y datos de un programa. Las señales digitales se desplazan de una sección a otra a través de vías llamadas buses. En sentido físico, el bus consta de varios conductores a través de los cuales se transportan diversas señales eléctricas. Éstos pueden ser las pistas de una tarjeta de circuito impreso, o los alambres de un cable plano. Los datos

necesarios para las instrucciones de procesamiento de la CPU se transportan a través del bus de datos, la dirección de una localidad de memoria específica para acceder a los datos almacenados se transporta por el bus de dirección y las señales de las acciones de control viajan por el bus de control. La figura 12 ilustra la configuración general.

Figura 12. Configuración general de una computadora



Fuente. [3]

1.4.2 Buses

El bus de datos se utiliza para transportar palabras a o desde la CPU, la memoria o las interfaces de entrada/salida. La longitud de las palabras puede ser 4, 8, 16, 32 o 64 bits. En cada línea del bus viaja una señal binaria, es decir, un 0 o un 1. Así, en un bus de cuatro líneas se podría transportar la palabra 1010; en cada cable se transporta un bit, es decir:

Tabla 1. Longitud de un bus de datos de cuatro bits

Palabra	Línea del bus
0 (bit menos significativo)	Primer línea del bus de datos
1	Segunda línea del bus de datos
0	Tercer línea del bus de datos
1 (bit más significativo)	Cuarta línea del bus de datos

Fuente: Autor

Entre más líneas tenga el bus de datos, más larga podrá ser la palabra que se utilice. El rango de valores que puede adoptar un elemento de datos está restringido al espacio correspondiente a una determinada longitud de palabra.

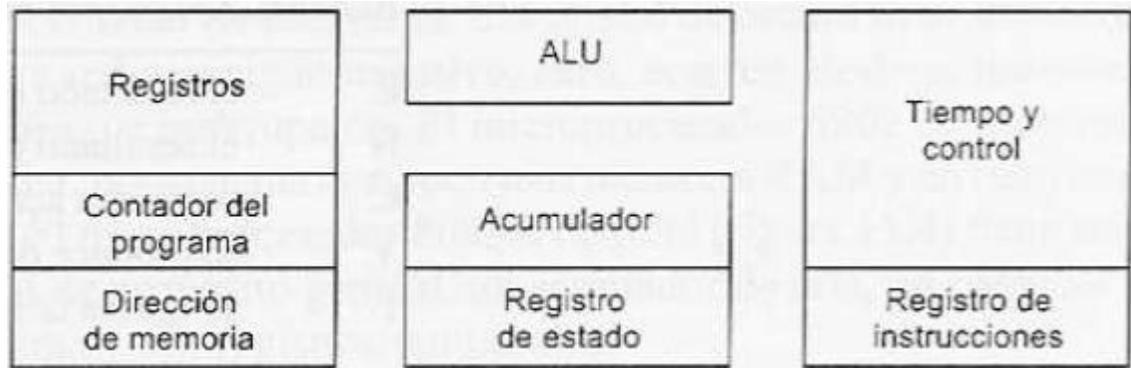
El bus de dirección transporta señales que indican donde se pueden encontrar los datos mediante la selección de alguna localidad de memoria o los puertos de entrada y salida. Cuando una determinada dirección es seleccionada, colocándola en el bus de dirección, dicha dirección será la única que estará abierta a la comunicación que se envía desde la CPU. Es decir, la CPU sólo puede comunicarse con una localidad a la vez.

El bus de control es el medio a través del cual se envían las señales que sincronizan cada uno de los elementos. Las señales del reloj del sistema se transportan en el bus de control. Estas señales generan intervalos durante los cuales se realizan las operaciones del sistema. La CPU envía señales de control a otros elementos para indicar qué tipo de operación se va a realizar, por ejemplo, si es necesario leer una señal (READ) o escribirla (WRITE).

1.4.3 CPU

La CPU es la sección del procesador en la que se procesan los datos, se traen instrucciones de la memoria que se decodifican y se ejecutan. Se puede decir que consta de una unidad de control, una unidad lógica y aritmética (ALU, por sus siglas en inglés) y de registros (ver figura 13) Es la parte que constituye el microprocesador.

Figura 13. Dispositivos de una CPU



Fuente. [3]

La unidad de control define la duración y secuencia de las operaciones. Produce las señales de reloj utilizadas para traer de la memoria una instrucción del programa y ejecutarla. Las operaciones pertenecientes al microprocesador se reconocen por la cantidad de ciclos que requiere su ejecución. La unidad aritmética y lógica se ocupa de las operaciones con los datos. Los datos internos que en un momento dado utiliza la CPU se guardan en forma temporal en un grupo de registros mientras se ejecutan las instrucciones. Existen diversos tipos de registros; la cantidad, dimensión y tipo de éstos varía de un microprocesador a otro.

1.4.4 Memoria

En la unidad de memoria se guardan datos de tipo binario; físicamente está formada por uno o varios circuitos integrados. Los datos pueden ser códigos de instrucciones de un programa, o números con los que se realizan operaciones. El tamaño de la memoria dependerá de la cantidad de líneas del bus de dirección. Los elementos de la unidad de memoria están formados en esencia por grandes cantidades de celdas de memoria, cada una de las cuales guarda un bit 0 o 1. Las celdas de memoria se agrupan por localidades, y cada una de ellas tiene capacidad para guardar una palabra. Para acceder a la palabra almacenada, se identifica cada localidad mediante una dirección específica. De

esta manera, en un bus de dirección de 4 bits se pueden identificar 16 direcciones diferentes, cada una de las cuales puede guardar un byte, es decir, un grupo de ocho bits. La capacidad de la unidad de memoria se especifica por la cantidad de localidades de memoria disponibles; 1K es $2^{10}=1024$ localidades; una memoria de 4K tiene 4096 localidades. Existen varios tipos de unidad de memoria:

1.4.5 ROM

Cuando se guardan datos en forma permanente, se utiliza un dispositivo de memoria conocido como memoria de sólo lectura (ROM por sus siglas en inglés). Las memorias ROM se programan con el contenido que se requiere durante la fabricación del circuito integrado. Mientras el chip de memoria esté en la computadora no es posible escribirle ningún dato, sólo se permite la lectura, y se utiliza para programas que no se van a modificar, como el sistema de arranque o “boot” de una computadora y programas para aplicaciones específicas en las que se utilizan microprocesadores.

1.4.6 PROM

El término ROM programable (PROM por sus siglas en inglés) se refiere a las memorias ROM que puede programar el usuario. En un principio, las celdas de memoria tienen un fusible como eslabón que mantiene su memoria en 0. Al hacer pasar una corriente a través del fusible, éste se abre de manera permanente, y así el valor cambia de 0 a 1. Una vez que el eslabón a quedado abierto, los datos se guardan en forma permanente en la memoria y ya no es posible modificarlos.

1.4.7 EPROM

El término ROM borrable y programable (EPROM por sus siglas en inglés) se refiere a las memorias ROM que es posible programar y modificar. Un chip EPORM típico contiene una serie de pequeños circuitos electrónicos, celdas, donde se almacena una carga. Para almacenar el programa se aplican voltajes a las terminales del circuito integrado y se produce una configuración de celdas cargadas y no cargadas. Esta configuración queda guardada de manera permanente en el chip hasta que la borra un haz de luz ultravioleta que pasa por una ventana de cuarzo ubicada en la parte superior del dispositivo. Esto provoca la descarga de todas las celdas. Por lo tanto, es posible volver a programar el chip.

1.4.8 EEPROM

La PROM eléctricamente borrable (EEPROM por sus siglas en inglés) es similar a las EPROM, pero para el borrado se utiliza un voltaje relativamente alto, en vez de luz ultravioleta.

1.4.9 RAM

Los datos temporales, es decir, aquellos con los que en un momento dado se realizan operaciones, se guardan en una memoria de lectura/escritura conocida como memoria de acceso aleatorio (RAM por sus siglas en inglés), es la que se puede leer y escribir. Cuando en una ROM se guarda un programa, éste estará disponible y listo en el mismo momento que se activa el sistema. Los programas que se guardan en una ROM se conocen como firmware (microprogramas). Siempre es necesario contar con ellos. Los programas guardados en una RAM se conocen como software. Cuando el sistema se activa, el software se puede cargar en la RAM desde el equipo periférico, como el teclado, el disco duro o un disco flexible.

1.4.10 ENTRADA/SALIDA

La operación de entrada/salida se define como la transferencia de datos entre el microprocesador y el mundo exterior. El término dispositivos periféricos se refiere a las piezas de equipo que intercambian datos con una arquitectura de microprocesador. Dado que las velocidades y características de los dispositivos periféricos pueden ser muy distintas a las del microprocesador, se conectan a través de circuitos de interfase. Una de las funciones más importantes de uno de estos circuitos es sincronizar la transferencia de datos entre el microprocesador y el dispositivo periférico. En las operaciones de entrada, el dispositivo de entrada coloca los datos en el registro de datos del circuito de interfase; estos datos permanecen ahí hasta que los lee el microprocesador. En las operaciones de salida, el microprocesador coloca los datos en el registro hasta que los lee el dispositivo periférico.

Para que el microprocesador pueda introducir datos válidos en un dispositivo de entrada necesita estar seguro de que el circuito de interfase ha retenido en forma correcta los datos de entrada. Para ello realiza un muestreo o una interrupción. En el primer caso, el chip de interfase recurre a un bit de estado definido como 1 para indicar que los datos son válidos. El microprocesador tiene que esperar a que aparezca este bit de estado. El problema con este método es que el microprocesador debe esperar hasta que se muestre el bit de datos. En el método de interrupción, el circuito de interfase envía una señal de interrupción al microprocesador, cuando contiene datos válidos; el microprocesador suspende la ejecución de su programa principal y ejecuta la rutina asociada con la interrupción para leer los datos.

2. FORMULACION DEL PROBLEMA

2.1 IDENTIFICACION DEL PROBLEMA

Las diferentes técnicas de control clásicas y modernas y su aplicación en la industria, han venido planteando retos importantes tanto a los educadores como a los estudiantes de pregrado y postgrado, razón por la cual su estudio y aplicación se ha venido incluyendo con más interés en los programas académicos de carreras afines con este campo de la ingeniería.

La enseñanza de la dinámica y el control desde el punto de vista pedagógico y científico, está llamada a realizar enormes cambios en su estructura académica. Por una parte, el estudiante que inicia tiene una gran expectativa acerca de la aplicabilidad de la teoría de control en la ingeniería, pero por otra, rápidamente se encuentra inmerso en planteamientos matemáticos y teorías con un elevado nivel de abstracción; es por esto que muchas universidades en el mundo actualmente poseen en sus laboratorios de Sistemas Dinámicos, Ingeniería del Control y Robótica varios bancos de trabajo, los cuales muestran a escala la aplicación de la teoría del control clásicas y modernas de manera sencilla y didáctica de diferentes sistemas y procesos industriales del mundo actual.

La Escuela de Ingeniería Mecánica de la Universidad industrial de Santander, comprometida en la misión de formar Ingenieros Mecánicos que respondan a los retos tecnológicos planteados por el dinamismo industrial, no cuenta en la actualidad con herramientas didácticas de Investigación y Desarrollo tecnológico que permitan aplicar los conocimientos adquiridos en las cátedras de Ingeniería del Control y Sistemas Dinámicos.

Para los docentes, no es sencillo explicar y dar a entender un tema que necesita mucha aplicación para ser asimilado por parte del estudiante; en la

escuela se han hecho diferentes intentos para dar explicación de los temas a los estudiantes de una manera didáctica, pero estos equipos tienen problemas de funcionamiento y no reflejan con precisión el contenido de la temática, es por esta razón, que el Grupo de Investigación en Dinámica, Control y Robótica (DICBOT) se ha propuesto como meta en este trabajo de grado, diseñar y construir un modelo para el control de un péndulo invertido rotativo, como forma de contribuir a la innovación en cuanto a la infraestructura de laboratorios se refiere, ayudando a la formación de los ingenieros emergentes y de alguna forma devolver a la universidad todo lo que hemos recibido durante estos años.

2.2 JUSTIFICACION PARA LA SOLUCION DEL PROBLEMA

El control automático es una de las disciplinas que ha tenido un importante y dinámico desarrollo, constituyéndose hoy por hoy en la piedra angular de lo que algunos autores han denominado la segunda revolución industrial, su imponente avance es consecuencia del uso generalizado de las técnicas de medición y control asociadas a la industria que en su incesante carrera por reducir costos, mejorar la calidad de los productos, acelerar los procesos productivos permitiendo estandarización y exactitud, reduciendo la mano de obra pasiva y remplazándola por trabajo especializado, ha encontrado en el control automático una solución necesaria y oportuna, base esencial y soporte teórico para otras áreas de interés tecnológico y científico, como es la Robótica, Ingeniería Aeronáutica, Ingeniería Aeroespacial, la Automatización, la Instrumentación, Ingeniería del Control, los Microsistemas y sus sistemas del Control.

Con todo esto la importancia de la construcción de equipos orientados a la enseñanza del control automático de sistemas dinámicos en la Escuela de Ingeniería Mecánica, que favorezcan un aprendizaje aplicativo, interactivo,

didáctico e investigativo y que permitan explorar las universalmente reconocidas ventajas del control automático se hace evidente. El péndulo invertido rotativo es una planta dinámica inestable muy útil y valiosa que puede ser utilizado para explicar y demostrar en forma práctica diferentes métodos de modelado, análisis y técnicas de control modernas y clásicas.

El diseño y construcción de este equipo se prefiere sobre la compra de el mismo en el mercado por 3 principales razones: sus altos costos de adquisición, la dependencia de proveedores foráneos para su operación y mantenimiento, y el limitado acomodo que presentan los equipos del mercado con los requerimientos que exige la formación en control del estudiante de Ingeniería Mecánica, inconvenientes que serán solucionados con la ejecución de este proyecto lo que justifica su realización.

La implementación de este banco de laboratorio servirá como apoyo a las asignaturas de Ingeniería de Control y Sistemas Dinámicos como una manera pedagógica y directa de fortalecer los conocimientos adquiridos en su contenido, ya que el estudiante de pregrado tendrá la oportunidad de practicar y aplicar lo que en un futuro podrá experimentar como profesional en el ejercicio de la Ingeniería Mecánica.

2.3 OBJETIVOS DEL TRABAJO DE GRADO

2.3.1 Objetivo General

Contribuir a la misión educativa impartida por la Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander, mediante el desarrollo de una herramienta pedagógica e investigativa, que ofrezca a los estudiantes de la escuela de Ingeniería Mecánica de la Universidad Industrial de Santander, una mejor forma de ilustrar y asimilar muchos de los conceptos y dificultades

asociadas a la solución de problemas de control en la industria moderna, y favorezca la posterior investigación en el *know how* en este campo de la ingeniería, y el compromiso del Grupo de Investigación DICBOT en invertir en el desarrollo y crecimiento investigativo de la Escuela de Ingeniería Mecánica.

2.3.2 Objetivos específicos

Diseño, selección y construcción de la plataforma mecánica del péndulo invertido rotativo. El proceso de diseño y construcción comprenderá:

- Seleccionar y adquirir los diferentes elementos que constituyen el hardware del péndulo invertido rotativo, como son:
 - Encoders ópticos incrementales (de 2500 pulsos/revolución, diámetro < 40mm).
 - Motor DC (12V) con encoder incremental y caja reductora con una potencia < 200 Watts.
 - Seleccionar y adaptar una tarjeta de control embebida, basada en el microcontrolador ATMEGA 2560 (Arduino) y/o MOTOROLA 68HC12 para realizar el control del péndulo invertido rotativo.

- Diseño y construcción de la estructura dinámica del péndulo invertido rotativo. Para este diseño se utilizarán herramientas de apoyo CAD, tal como SolidWork. Este diseño comprenderá:
 - ✓ Diseño y construcción de la plataforma de experimentación (péndulo invertido rotativo) con un espacio de trabajo con dimensiones en cm:
35 < largo < 45

35 < ancho < 45

55 < alto < 65

- ✓ Diseño y construcción de una mesa de soporte de la planta de experimentación del péndulo invertido rotativo con dimensiones en cm:

$60 < \text{ancho} < 70$

$60 < \text{largo} < 70$

$70 < \text{alto} < 80$

- Implementar el control del péndulo invertido rotativo con un filtro PID para mostrar la funcionalidad de la plataforma.
- Realizar las pruebas y ajustes del controlador para diferentes escenarios de desempeño y estabilidad.
- Realizar y documentar pruebas del sistema dinámico con el fin de analizar el tiempo de respuesta y el comportamiento del péndulo invertido rotativo al someterlo a señales de perturbación.
- Diseñar un manual didáctico y de operación para conducir el experimento.

3. DESCRIPCION DEL PENDULO INVERTIDO ROTATIVO COMO SISTEMA DE CONTROL

El Péndulo Invertido Rotativo como sistema dinámico es un sistema mecánico sub-actuado que posee dos grados de libertad rotacionales y un solo actuador. El sistema sub-actuado se puede definir como aquel sistema que carece al menos de un actuador en uno de sus grados de libertad, es decir que en dicho sistema será aquel que posea un número de grados de libertad mayor que el de actuadores [1].

Los componentes básicos de la estructura del péndulo se aprecian en la figura 14:

Figura 14. Componentes básicos del péndulo: a) motor, b) brazo, c) péndulo

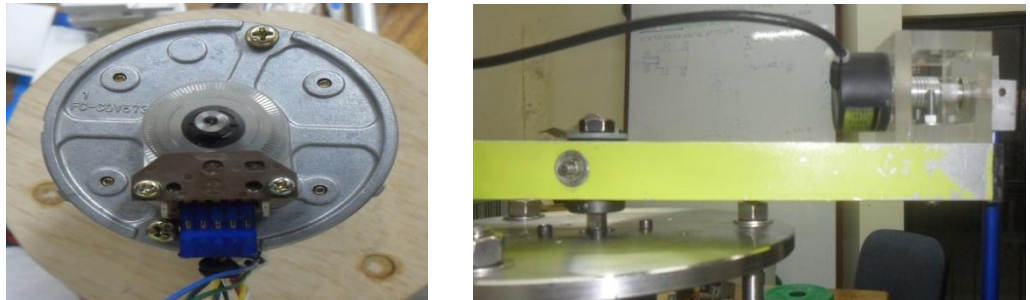


Fuente. Autor

El enfoque de ésta tesis, por ser el primero en su tipo en ser construido e implementado en la Escuela de Ingeniería Mecánica , está direccionado a la enseñanza de las técnicas de control clásico, en este caso PID (Proporcional, Integral y Derivativo) del cual se hará descripción más adelante. Se busca diseñar e implementar un algoritmo de control que permita realizar acciones correctivas de control sobre los componentes principales del péndulo invertido rotativo, mediante el uso de las señales de retroalimentación proporcionadas por los dispositivos de medición angular (encoders ópticos incrementales) que se encuentran adjuntos al péndulo y al brazo, en este último el dispositivo de medición se encuentra acoplado al eje del motor que a su vez se acopla con el

eje rotacional del brazo, por lo tanto al sensor la posición del eje del motor, implícitamente se está sensando la posición del brazo, como se aprecia en la figura 15.

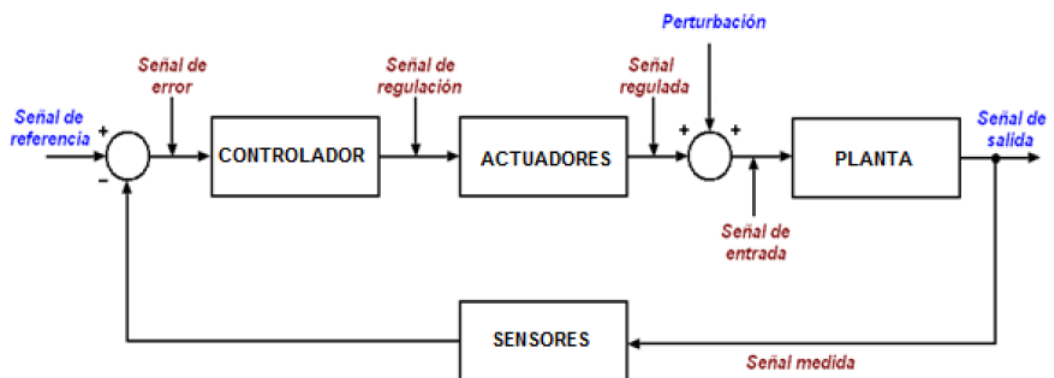
Figura 15. Ubicación de los sensores de medición.



Fuente. Laboratorio DICBOT

El sistema de control del Péndulo Invertido Rotativo se puede representar mediante diagramas de bloque, como un sistema de control en lazo cerrado como el mostrado en la figura 16.

Figura 16. Diagrama de bloque de un sistema de control en lazo cerrado.



Fuente: [12]

En capítulos posteriores se hará una descripción completa de los componentes que conforman el diagrama de bloques anteriormente mencionado.

3.1 REQUERIMIENTOS DE DISEÑO

Debido al comportamiento dinámico del péndulo de Furuta, en el que la principal variable a controlar es la posición angular del péndulo y la posición del brazo, para lo cual se requiere:

- Leer constantemente la información proporcionada por los encoders acerca de la posición del brazo y del péndulo.

- Relacionar dicha información con el voltaje que será enviado al motor de manera que a mayor error el motor responda más rápidamente y viceversa.

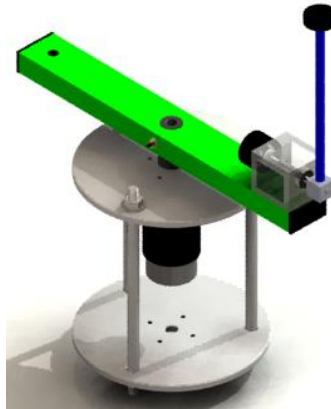
- El motor debe tener alto torque, para poder responder oportunamente, más no porque deba vencer altas cargas de inercia.

Para comprender en detalle los elementos que conforman el sistema dinámico y de control, se divide en dos etapas, operación y control.

3.2 PARTE OPERATIVA

La parte operativa de un sistema dinámico controlado, está compuesta por el conjunto (máquina, sensores y actuadores), los cuales están dispuestos de manera adecuada para dar soporte y permitir el correcto funcionamiento de todo el conjunto.

Figura 17. Disposición de los elementos



Fuente: Autor

La figura 17 muestra la disposición de dichos elementos, en este caso, a mencionar:

- Motoreductor de corriente directa con encoder incorporado.
- Base del motor.
- Brazo.
- Péndulo y encoder acoplado.

La forma en que opera el sistema dinámico se describe a continuación:

El sensor acoplado al motor envía la información en tiempo real acerca de la posición del brazo adjunto al eje del actuador (motor DC), pero ésta información, el sensor la envía como señales eléctricas o pulsos, una vez que el microcontrolador recibe la información, o lo que es lo mismo, las señales eléctricas, dichas señales se almacenan en una variable ingresada por software y el valor de la variable aumentará o disminuirá en un valor de uno (1) dependiendo del orden en que los pulsos o señales lleguen al microcontrolador.

El segundo encoder, acoplado al péndulo, opera de la misma manera, tan solo que los pulsos eléctricos enviados por este, se almacenan en otra variable. Una vez que el microcontrolador ha recibido los pulsos provenientes de ambos sensores, se procede mediante el software o el ambiente de programación de arduino, a convertir dichos pulsos en grados mecánicos, proceso que se lleva a cabo al multiplicar la variable donde se almacenan los pulsos por un factor de

equivalencia, obteniéndose así el valor de la posición en grados.

Ahora el microcontrolador, dependiendo de la posición del péndulo, ejecutará una de las dos tareas principales para la que se ha programado, ejecutar el control de “swing up” o la del control de balanceo.

3.2.1 Operación “swing up”

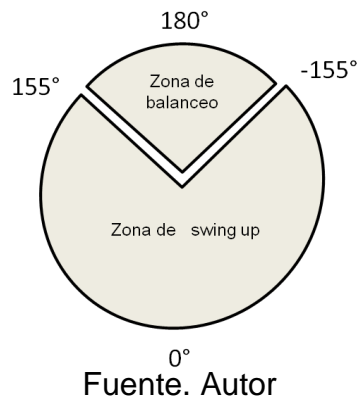
Se ejecutará si la posición del péndulo está entre los 0° y 155° , o entre los 0° y -155° . Este lazo de control consiste en que pondrá al motor a oscilar manteniendo un chequeo de la posición del péndulo hasta que el mismo, alcance una energía determinada correspondiente a una cercana a la posición vertical, una vez allí, el microcontrolador entrará a ejecutar el control de balanceo.

3.2.2 Operación de balanceo

Una vez el péndulo alcanza la posición mencionada arriba, un segundo lazo de control entra en juego, cuyo objetivo es mantener el péndulo en esa posición vertical, aún cuando haya perturbaciones en el sistema.

Para este lazo de control, el encoder adjunto al péndulo envía la información acerca de la posición hacia el microcontrolador. Una vez el microcontrolador ha leído la posición del brazo y péndulo procede a calcular el error que introduce en el algoritmo PID y dicho algoritmo entrega un valor el cual será enviado al motor como un voltaje en función de la posición, permitiendo así mantener el péndulo en la posición cercana a la vertical y responder ante eventuales perturbaciones. En la figura 18 se observan las zonas de trabajo de cada operación.

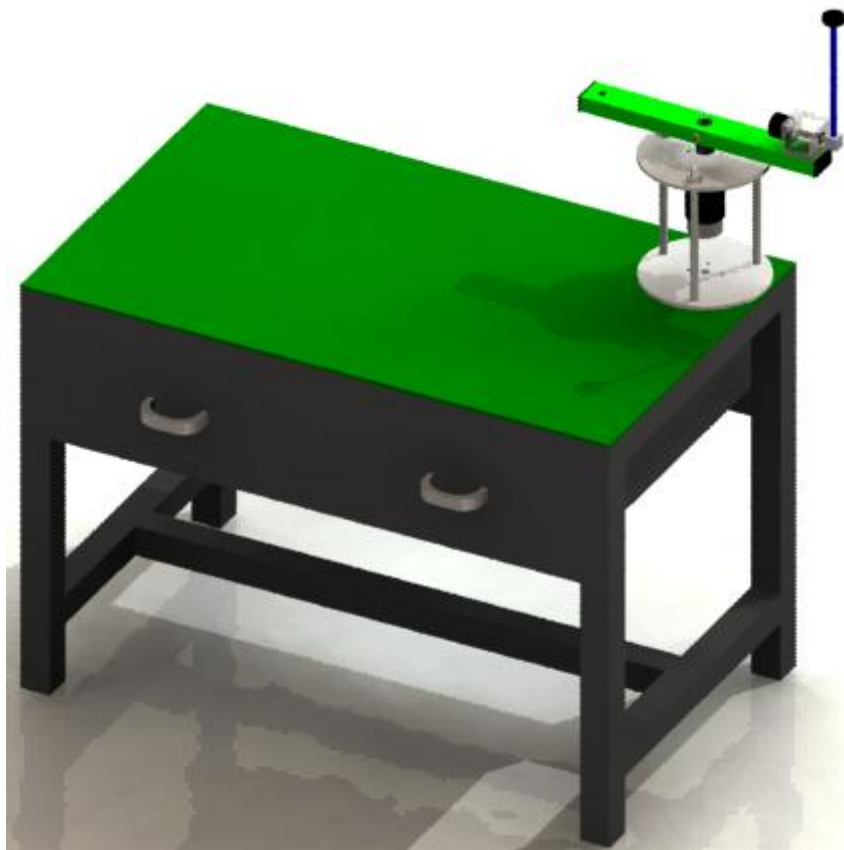
Figura 18. Posición vertical del péndulo.



3.2.3 Planta Dinámica

La máquina en este caso es el péndulo invertido rotativo, como se aprecia en la figura 19.

Figura 19. Planta dinámica de pruebas.



Fuente. Autor.

Como se aprecia en la selección de alternativas estipulado en el plan del proyecto, este sistema nos proporciona características pedagógicas para enseñar de manera didáctica la familia de controladores PID en primera instancia, ya que es un sistema inestable no lineal, el primero en construirse en el laboratorio de control y sistemas dinámicos de la Escuela de Ingeniería Mecánica, y que servirá como prototipo para realizar pruebas de otras técnicas de control, sin necesidad de realizar cambios físicos a la planta, solamente implementando modificaciones al algoritmo de control mediante el uso de otro tipo de controladores como por ejemplo un control en el espacio de estados, lógica fuzzy u otra técnica de control robusto o control adaptativo.

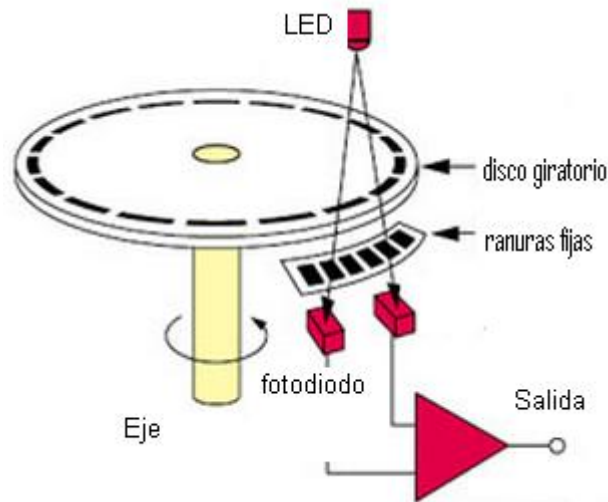
3.2.4 Sensores

Los sensores son dispositivos usados para proporcionar señales de retroalimentación. Dichas señales de retroalimentación pueden proporcionar información acerca de la dirección o rotación, posición y velocidad entre otros. En el presente trabajo de grado se hizo uso de dos encoders ópticos incrementales.

Un encoder óptico es un dispositivo electromecánico que posee una salida eléctrica en forma digital, proporcional al eje de salida. Un encoder óptico es un sensor de posición angular. Para obtener la información acerca de la posición angular tanto del brazo como del péndulo se hace necesario entender el funcionamiento del encoder.

Este posee un eje mecánicamente acoplado a otro que lo controla, el cual gira un disco rígido fijado a este. En la superficie del disco hay una serie de segmentos transparentes y opacos marcados. Ver figura 20.

Figura 20. Esquema básico de un encoder óptico.



Fuente. http://www.khi.co.jp/tb_e/07_robot/07_robot.html

La luz de los diodos emisores infrarrojos llega a los receptores infrarrojos a través de las ranuras transparentes del disco giratorio. Luego, electrónicamente la señal es amplificada y convertida a un formato digital. Esta señal es entonces transmitida al procesador de datos.

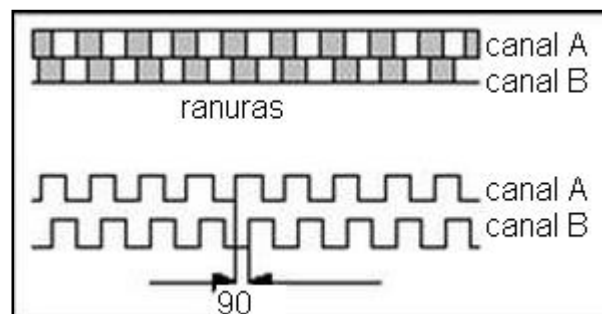
Este dispositivo es llamado encoder incremental porque los pulsos que produce son contados uno a la vez. Si las ranuras tienen una separación de un grado una de la otra, el encoder producirá 360 pulsos cada vez que de una revolución o vuelta. Si por ejemplo el eje gira 360 grados el encoder producirá 360 pulsos. Los encoders poseen uno, dos o tres canales de lectura o pistas de ranura. Una forma de mejorar el desempeño de los encoders incrementales que emplean franjas reflectivas es a través de los encoders de cuadratura. La mejora consiste en dos aspectos:

- Primero proporciona la dirección de rotación o sentido de giro del movimiento, es decir, indica si el desplazamiento rotacional tiene dirección negativa (en el sentido del giro de las manecillas del reloj) o positiva (en sentido contrario al de las manecillas del reloj); esto es importante para detectar el signo del error de posición.

- La segunda mejora se encuentra en duplicar la resolución, puesto que dos estados son generados por cada marca del encoder, por lo tanto se dispondrá de mayor exactitud en el posicionamiento.

Los encoders de cuadratura también pueden usar dos sensores arreglados sobre el disco giratorio, de tal forma que cuando un sensor detecta el límite blanco-negro de la franja causará un cambio de estado, el otro sensor estará exactamente en medio de la otra marca, ver figura 20, lo que asegura que responderá con un estado estable conforme el primer sensor cambia de estado. La salida resultante de cada sensor estará fuera de fase 90°. Ver figura 21.

Figura 21. Encoder en cuadratura.



Fuente. <http://www.ni.com/cms/images/devzone/tut/lcoktysa3498.jpg>

En la figura 21 se ejemplifica el arreglo de encoder incremental de cuadratura y las formas de onda de los sensores A y B. una ranura negra produce un pulso alto en la salida del sensor (una transición de bajo hacia alto). Una ranura blanca produce una salida en bajo (transición de alto hacia bajo). Conforme el disco se mueva en dirección de las manecillas del reloj, si B cambia de alto a bajo, después A cambiará de alto hacia abajo, o si B cambia de bajo hacia alto, después A cambiará de bajo hacia alto.

Deben considerarse las siguientes características para usar un encoder incremental:

- Es necesario tomar precauciones sobre el ruido de la señal del encoder, ya que se acumula el ruido en el contador.
- Cuando la fuente de alimentación se desconecta, es imposible mostrar la posición previa, aun si inmediatamente se conecta la fuente.

Debido a que los encoders incrementales tienen dos canales A y B que están fuera de fase 90 grados, de esta forma es posible determinar la dirección de giro. Existe también un canal opcional llamado Z o I, cuya salida proporcionará un pulso por cada 360 grados. La utilidad de este canal consiste en detectar el número de vueltas que realiza; además de que también es usado como índice de referencia para ajustar en el caso de los servomotores la posición cero (o home).

Aspectos importantes de los encoders [ibro en el lab]

La selección de encoders ópticos depende de la aplicación específica. Sin embargo, es conveniente tomar en cuenta las siguientes consideraciones: resolución y precisión de la posición; tamaño del encoder para propósitos de acondicionamiento mecánico, diferencia de fase de salida, sensibilidad al medio ambiente, cambios climáticos, polvo, condensación de agua (lugares húmedos o cálidos). Cuando el nivel de salida es compatible con TTL, *lógica transistor a transistor* (transistor-transistor logic). La fuente de voltaje para los encoders es generalmente de cinco voltios, por lo tanto la salida es fácilmente afectada por ruido ya que son contaminados en lugares donde hay ruido ambiental; en este caso; en este caso se deberá incorporar circuitos aisladores de ruido. En acoplamientos mecánicos las flechas del encoder y del rotor deben estar bien alineadas, la flecha del rotor no debe estar desbalanceada, ya que degrada la exactitud y acorta la vida del encoder.

Los encoders incrementales son de bajo precio; no registran la posición

absoluta después de que la fuente de alimentación ha sido desconectada, por lo tanto pueden utilizarse donde la pérdida de energía no es importante. Los encoders incrementales son adecuados para la mayoría de las aplicaciones ya que pueden rotar y producir pulsos como sea necesario, siempre y cuando tengan la instrumentación electrónica para almacenar y contabilizar números muy grandes. En contraste, los encoder absolutos con código binario o código gray pueden leer la posición dentro de una rotación y por lo tanto no son adecuados para múltiples rotaciones. El encoder óptico incremental genera pulsos por rotar el disco giratorio entre la luz que emiten los diodos LED y la luz que reciben los elementos fotosensibles.

Hay que tener cuidado de seleccionar un número alto de pulsos por revolución, ya que esto significa alta exactitud y resolución. No siempre es la mejor opción la selección de encoder con un número alto de pulsos por revolución, puesto que esto implica un mayor precio y altas frecuencias en las salidas para una velocidad dada del rotor al que este acoplado el encoder.

El sensor utilizado para medir la posición angular del péndulo es un encoder óptico incremental de automation direct de tres canales de lectura y sus características técnicas se aprecian en la tabla 2:

Figura 22. Encoder incremental TRD-S2500-BD



Fuente: (<http://www.aotomationdirect.com/statics/specs/encoderId.pdf>)

Tabla 2. Características del encoder incremental TRD-S2500-BD

Característica	Valor
Voltaje de alimentación	10.8-26.4 VDC
Consumo de corriente	50mA max.
Forma de la señal	Dos fases+posición inicial
Frecuencia máxima de respuesta	200 KHz
Resolución	2500 ppr

El encoder adjunto en la parte trasera del motor DC es uno de tipo óptico incremental HEDS-9100 Agilent Technologies de dos canales de lectura.

Figura 23. Encoder incremental HEDS-9100



Fuente: (<http://datasheet.octopart.com/HEDS-9100%23I00-Avago-datasheet-10307691.pdf>)

Tabla 3. Características del encoder incremental HEDS-9100

Característica	Valor
Voltaje de alimentación	5 VDC
Consumo de corriente	5mA
Forma de la señal	Dos fases compatibles TTL
Frecuencia máxima de respuesta	100 KHz
Resolución	100 ppr

3.2.5 Actuador

En este caso, es un motor de corriente continua o DC (Direct Current) el elemento que interactúa entre la señal de control y la planta. Debido a que los requerimientos eléctricos del sistema son de baja potencia y consumo de corriente y voltaje, el motor DC es una muy buena elección.

Es necesario que el motor DC a usar tenga un buen torque y unas rpm no muy bajas, ya que cuando el péndulo está en la etapa de elevación el motor debe vencer muy fácilmente las inercias, tanto del mismo motor como la inercia del brazo y el péndulo adjuntado al mismo, pero además debe responder rápidamente ante cualquier perturbación que trate de llevar el sistema a una posición fuera del equilibrio.

Para seleccionar el motor que se ajuste a las necesidades del proyecto como en [5]. En el laboratorio del grupo DICBOT se encuentran 3 motores DC que se ajustan perfectamente a las necesidades de velocidad, torque y potencia para esta aplicación, por lo tanto se usa uno de ellos como actuador en el péndulo de Furuta, y cuyas características se citan en la siguiente tabla.

Tabla 4. Características del motor.

MOTORREDUCTOR DC DE IMAN PERMANENTE CON ENCODER	
Diámetro	69 mm
Largo del motor	131 mm
Largo del encoder	31 mm
Largo del reductor	40 mm
Largo del eje	35 mm
Diámetro del eje	10 mm
Marca	Matsushita
Modelo	GMX-7MC019B
RPM	230
Alimentación	22 Volts
Consumo sin carga	0.5 Amp.
Consumo con carga	2 Amp. máx

Una imagen del actuador se aprecia en la figura 25.

Figura 24. Actuador

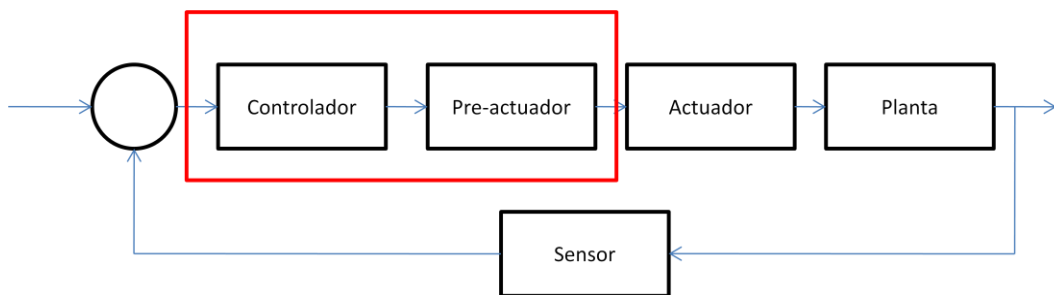


Fuente: Laboratorio DICBOT

3.3 PARTE DE CONTROL

El componente de control del péndulo de Furuta está constituido por el pre-actuador y el controlador, adicionalmente las comunicaciones entre los diferentes elementos que componen la planta dinámica del péndulo, en la figura 25 se aprecia la parte de control.

Figura 25. Componente de control

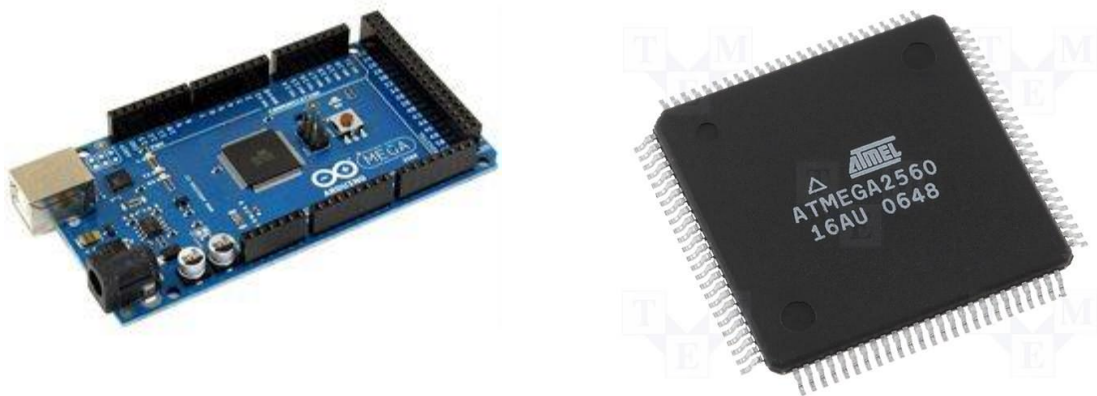


Fuente: Autor

3.3.1 Controlador

Se hizo uso de una tarjeta de desarrollo llamada Arduino Mega la cual se aprecia en la figura 26, esta tarjeta posee todos los componentes electrónicos necesarios para integrar un chip microcontrolador Atmega2560 del fabricante Atmel, mediante el cual se diseñó un controlador embebido para el péndulo de Furuta, haciendo uso del ambiente de programación de Arduino el cual está basado en C++.

Figura 26. Arduino Mega2560 y microcontrolador ATMEGA2560



Fuente. <http://www.robotshop.com/ProductPictureViewer.aspx?id=28678>

Tabla 5. Características Arduino MEGA

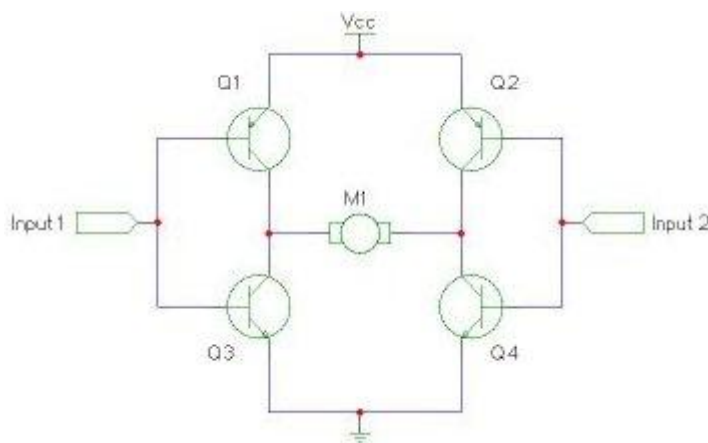
Principales características de la board Arduino MEGA	
Microcontrolador	Atmega2560
Voltaje de operación	5V
Voltaje de alimentación (recomendado)	7-12V
Voltaje de alimentación (límites)	6-20V
Pines digitales I/O	54 (14 proporcionan salidas PWM)
Pines de entrada analógicos	16
Corriente DC por pin I/O	40mA

Corriente DC para el pin de 3.3V	50mA
Memoria flash	256KB
SRAM	8KB
EEPROM	4KB
Velocidad de reloj	16MHz

3.3.2 Pre-actuador

Para el control y manejo del motor DC es indispensable implementar un dispositivo conocido como puente H, capaz de controlar la dirección y velocidad del actuador, dependiendo de las señales de control que entregue el actuador. Este dispositivo es el enlace entre la parte de control y la de potencia, y su importancia es clara en el funcionamiento del péndulo, un arreglo básico del puente H se observa en la figura 27.

Figura 27. Esquema básico de un puente H.



Fuente. <http://jorgefloresvergaray.blogspot.com/2009/03/puente-en-h-con-transistores.html>

El circuito puente H está basado en un arreglo de transistores que soportan los voltajes y corrientes para alimentar el motor. La alimentación se hará a 20 V proveniente de una fuente de voltaje regulable disponible en el laboratorio del

grupo DICBOT.

El control de voltaje entregado al motor se realizará mediante el envío de señales PWM por el microcontrolador hacia el motor, permitiendo el control de velocidad y dirección del motor.

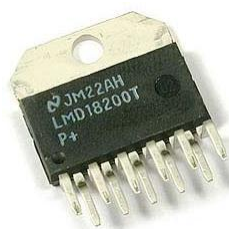
3.3.3 Integrado LMD18200T

El integrado LMD182200 fabricado por la National Semiconductor, es un puente H de tres amperios (3A) diseñado para aplicaciones de control de movimiento. Este dispositivo es construido utilizando multi-tecnología que combina circuitería de control bipolar y CMOS () con dispositivos de potencia DMOS () en la misma estructura monolítica. Algunas de las aplicaciones de este integrado son:

- Control de motores DC y paso a paso.
- Servomecanismos de posición y velocidad.
- Robots manipuladores.
- Máquinas de control numérico.
- Impresoras y plotters.

En la figura 28, se aprecia una imagen del chip

Figura 28. Puente H LMD18200

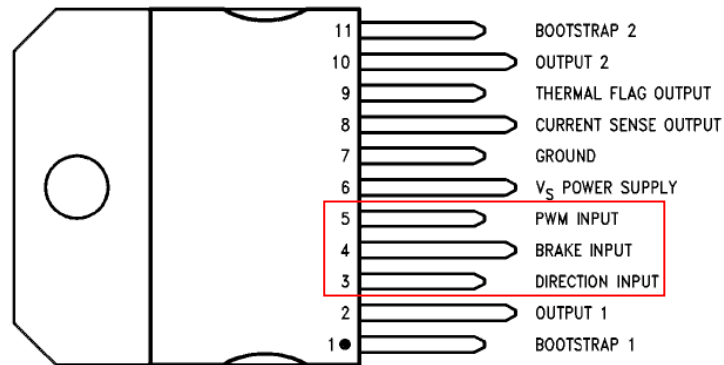


Fuente. http://catatansaad.files.wordpress.com/2009/08/17360_pv.jpg

Las señales de control enviadas desde el microcontrolador hacia el pre-actuador se envían por tres pines a través del integrado, los cuales son dirección (DIRECTION INPUT), pin de freno (BREAK INPUT) y el pin de pwm

(PWM INPUT), como se aprecia en la figura 29.

Figura 29. Pines de control del LMD18200T



Fuente. Anexo B.

Mediante la asignación de un estado lógico determinado a estos tres pines, se consigue hacer girar el motor ya sea en un sentido o en el otro con una señal PWM de magnitud determinada por la salida del algoritmo de control. La combinación de los estados lógicos se muestra en la tabla 6. En la tabla se observa la letra H (HIGH) que es equivalente a un estado lógico de uno (1), igualmente aparece la letra L (LOW) que representa un estado lógico de cero (0).

Tabla 6. Combinación pines de control del LMD18200

PWM	Dirección	Freno	Sentido de giro
H	H	L	Horario
H	L	L	Anti-horario
L	L	H	Frenado rápido

Fuente. Autor

Se observa en la tabla que cuando el estado de los pines es H-H-L respectivamente el puente H permite al motor girar en sentido horario, mientras

que si el estado de los pines es H-L-L el chip permite al motor girar en sentido anti-horario. Finalmente, al asignar a los pines el estado representado por L-L-H, la acción que ejecutará el motor será la de detenerse bruscamente permitiendo así un frenado rápido, evitando el efecto de la inercia de frenar lentamente.

4. MODELO MATEMATICO DEL PENDULO INVERTIDO ROTATIVO

Los modelos matemáticos de los sistemas físicos (mecánicos, eléctricos, etc.) pueden derivarse de consideraciones de energía. El modelo matemático es el primer paso en el diseño de un lazo de control. Cuanto mayor sea el conocimiento del sistema o proceso, mejores son los resultados obtenidos, ya que el diseño del control estará basado en el modelo matemático del sistema. Para conocer la dinámica del proceso, se puede deducir su comportamiento a partir de las leyes físicas que lo rigen como, eléctricas, mecánicas, térmicas, etc.

En este capítulo se hará una detallada descripción de la obtención del modelo matemático del péndulo invertido rotativo, a partir de las leyes físicas que lo gobiernan.

4.1 ECUACIONES DE MOVIMIENTO DEL PENDULO INVERTIDO ROTATIVO

Para realizar el modelo matemático del péndulo de Furuta se recurre a la aplicación la ecuación de Euler-Lagrange [13] la cual está basada en consideraciones de energía.

Para derivar las ecuaciones de movimiento de Lagrange, es necesario definir las coordenadas generalizadas y el Lagrangiano.

Coordenadas generalizadas: las coordenadas generalizadas de un sistema son un conjunto de coordenadas independientes que se necesita para describir completamente el movimiento del sistema. El número de coordenadas generalizadas necesario para describir el movimiento del sistema el igual número de grados de libertad.

Si un sistema requiere n coordenadas generalizadas q_1, q_2, \dots, q_n necesitamos considerar n coordenadas generalizadas como coordenadas de un sistema coordinado n -dimensional.

El péndulo de Furuta posee dos grados de libertad, ambos rotacionales, por lo tanto se necesitan dos coordenadas generalizadas para definir completamente el movimiento del péndulo invertido rotativo. Entonces llamemos q_1 y q_2 las coordenadas generalizadas para describir el movimiento del sistema.

Lagrangiano: El lagrangiano L de un sistema se define por:

$$L=T- U$$

En donde T es la energía cinética y U es la energía potencial del sistema. El lagrangiano L en forma general es una función de las coordenadas generalizadas, en este caso, de q_1 y q_2 y del tiempo t , o bien

$$L=L(q_1, q_2, t)$$

Para una descripción más detallada de las ecuaciones ver [13].

4.1.1 Ecuaciones de movimiento de Lagrange

La siguiente ecuación se conoce como la *ecuación de Euler-Lagrange*.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = 0$$

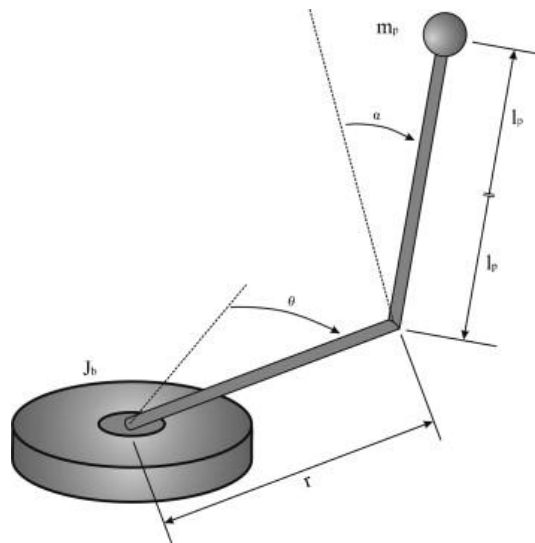
Que se reduce a la ecuación de movimiento del sistema que pudiera obtenerse al usar la segunda ley de Newton. Por consiguiente, también se le conoce como *ecuación de movimiento de Lagrange*. Existen n ecuaciones de Lagrange por cada n coordenadas generalizadas, es decir que se necesitan dos ecuaciones para describir el sistema dinámico. Por lo tanto:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_1} \right) - \frac{\partial L}{\partial q_1} = 0$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_2} \right) - \frac{\partial L}{\partial q_2} = 0$$

En donde q_1 y q_2 son las coordenadas generalizadas las cuales, basados en la figura 30 corresponden a α y θ respectivamente.

Figura 30. Modelo simplificado



Fuente. <http://ars.sciencedirect.com/content/image/1-s2.0-S0378475409002419-gr2.jpg>

Por lo tanto:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} = T \quad (1)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0 \quad (2)$$

α (alpha) es el ángulo formado por el brazo en un plano horizontal, y θ (theta) es el ángulo formado por el péndulo en un plano vertical.

4.1.2 Obtención del Lagrangiano L

De la cinemática del cuerpo rígido [14] se pueden obtener las expresiones para la energía cinética y potencial del péndulo de Furuta.

Para obtención de la ecuación de energía cinética T , se asume que a ésta contribuye la energía del brazo y del péndulo, es decir que la energía cinética del sistema es la suma de la energía cinética del brazo y del péndulo:

$$T_{\text{total}} = T_b + T_p$$

En donde:

T_b = energía cinética del brazo;

T_p = energía cinética del péndulo;

Para el brazo se tiene entonces que:

$$T_b = \frac{1}{2} m_b l_b^2 v_b^2 + \frac{1}{2} I_b \dot{\alpha}^2$$

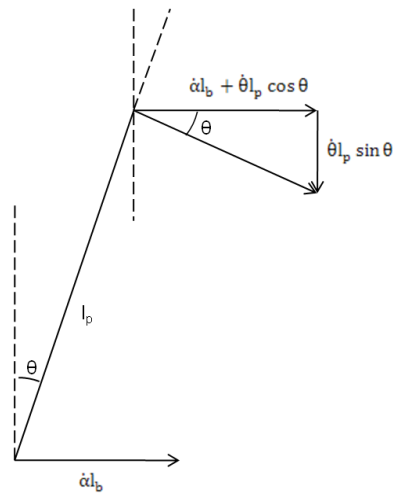
En la ecuación anterior ya que el brazo posee solo movimiento rotacional ($v_b = 0$), la ecuación anterior se reduce a:

$$T_b = \frac{1}{2} I_b \dot{\alpha}^2$$

Para el péndulo se tiene que la energía cinética corresponde a un movimiento en el plano general, por lo tanto tendrá componente traslacional y rotacional, basado en el diagrama de la figura 31, la energía cinética del péndulo es:

$$T_p = \frac{1}{2} I_p \dot{\theta}^2 + \frac{1}{2} m_p [(\dot{\alpha} l_b + \dot{\theta} l_p \cos \theta)^2 + (\dot{\theta} l_p \sin \theta)^2]$$

Figura 31. Diagrama cinemático del péndulo



Fuente. Autor

Ahora, para la energía potencial se tiene que, es la suma de las energías potenciales del brazo y del péndulo, es decir que:

$$U_{\text{total}} = U_b + U_p$$

En donde:

U_b = energía cinética del brazo;

U_p = energía cinética del péndulo;

Tomando como nivel de referencia el brazo se tiene que tiene que $U_b=0$;

Para el péndulo: $U_p = m_p g l_p \cos \theta$

Entonces se tiene que el Lagrangiano es:

$$L = T_b + T_p - U_p$$

Reemplazando los términos en la ecuación del Lagrangiano, resulta:

$$L = \frac{1}{2} I_b \dot{\alpha}^2 + \frac{1}{2} I_p \dot{\theta}^2 + \frac{1}{2} m_p [(\dot{\alpha} l_b + \dot{\theta} l_p \cos \theta)^2 + (\dot{\theta} l_p \sin \theta)^2] - m_p g l_p \cos \theta$$

Ahora se aplican las derivadas parciales al Lagrangiano con respecto a las coordenadas generalizadas α y θ respectivamente, luego con respecto a las velocidades $\dot{\alpha}$ y $\dot{\theta}$, finalmente con respecto al tiempo t , resultando:

La derivada parcial del Lagrangiano con respecto a la posición del brazo:

$$\frac{\partial L}{\partial \alpha} = 0$$

La derivada parcial del Lagrangiano con respecto a la velocidad angular del brazo:

$$\frac{\partial L}{\partial \dot{\alpha}} = I_b \dot{\alpha} + m_p \dot{\alpha} l_b^2 + m_p l_p l_b \dot{\theta} \cos \theta$$

La derivada con respecto al tiempo de la anterior es:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) = I_b \ddot{\alpha} + m_p l_b^2 \ddot{\alpha} + m_p l_p l_b \ddot{\theta} \cos \theta - m_p l_p l_b \dot{\theta}^2 \sin \theta$$

Ahora, reemplazando en (1) se obtiene la primera ecuación diferencial:

$$(I_b + m_p l_b^2) \ddot{\alpha} + (m_p l_b l_p \cos \theta) \ddot{\theta} - (m_p l_b l_p \sin \theta) \dot{\theta}^2 = T_m \quad (3)$$

Para obtener la segunda ecuación, primero, se deriva parcialmente el Lagrangiano con respecto a la posición del péndulo:

$$\frac{\partial L}{\partial \theta} = m_p l_p \dot{\theta} g \sin \theta - m_p l_p l_b \dot{\alpha} \sin \theta$$

La derivada parcial del Lagrangiano con respecto a la velocidad angular del péndulo:

$$\frac{\partial L}{\partial \dot{\theta}} = I_p \dot{\theta} + m_p l_b l_p \dot{\alpha} \cos \theta + m_p l_p^2 \dot{\theta}$$

Ahora se deriva con respecto al tiempo y resulta:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = I_p \ddot{\theta} + m_p l_p l_b (\ddot{\alpha} \cos \theta - \dot{\alpha} \dot{\theta} \sin \theta) + m_p l_p^2 \ddot{\theta}$$

Reemplazando en (2) resulta entonces:

$$(m_p l_b l_p \cos \theta) \ddot{\alpha} + (I_p + m_p l_p^2) \ddot{\theta} + (m_p l_p g \sin \theta) = 0 \quad (4)$$

Como se puede deducir de las ecuaciones, se evidencia la no linealidad del péndulo de Furuta, su comportamiento está definido por ecuaciones diferenciales de segundo orden y por productos entre las variables

4.1.3 Modelo matemático del motor

Un motor DC de imán permanente como el que se usa en esta aplicación, genera una fuerza electromagnética (E_a) contraria al sentido de rotación del motor, y que es proporcional a la velocidad de rotación del rotor:

$$E_a = K_v \dot{\alpha}$$

El torque electromagnético producido es directamente proporcional a la corriente del motor:

$$T_m = K_t I_a = K_v I_a$$

En estado estable, la ecuación que describe el motor es:

$$V_a = R_a I_a + E_a$$

Al despejar la corriente y reemplazarla en la ecuación de torque se obtiene la relación entre el torque y el voltaje aplicado al motor:

$$T_m = \frac{K_t V_a}{R_a} - \frac{K_t K_v \dot{\alpha}}{R_a}$$

En donde V_a es el voltaje aplicado al motor. Se introducen una serie de parámetros para que el modelo matemático sea abreviado (ver tabla 7):

$$C_1 \ddot{\alpha} + (C_2 \cos \theta) \ddot{\theta} - (C_2 \sin \theta) \dot{\theta}^2 = C_3 V_a - C_4 \dot{\alpha} \quad (5)$$

$$(C_2 \cos \theta) \ddot{\alpha} + C_5 \ddot{\theta} + C_6 \sin \theta = 0 \quad (6)$$

Tabla 7. Parámetros del modelo

Constante	Valor
C_1	$I_b + m_p l_b^2$
C_2	$m_p l_b l_p$
C_3	$\frac{K_t}{R_a}$
C_4	$\frac{K_t K_v}{R_a}$
C_5	$I_p + m_p l_p^2$
C_6	$m_p l_p g$

4.1.4 Linealización del modelo

Para linealizar el modelo es necesario escoger un punto de equilibrio alrededor del cual el péndulo tendría un comportamiento ligeramente estable, así que se introducen las siguientes aproximaciones:

$$\cos \theta \approx 1 ;$$

$$\sin \theta \approx \theta ;$$

$$\dot{\theta}^2 \approx 0$$

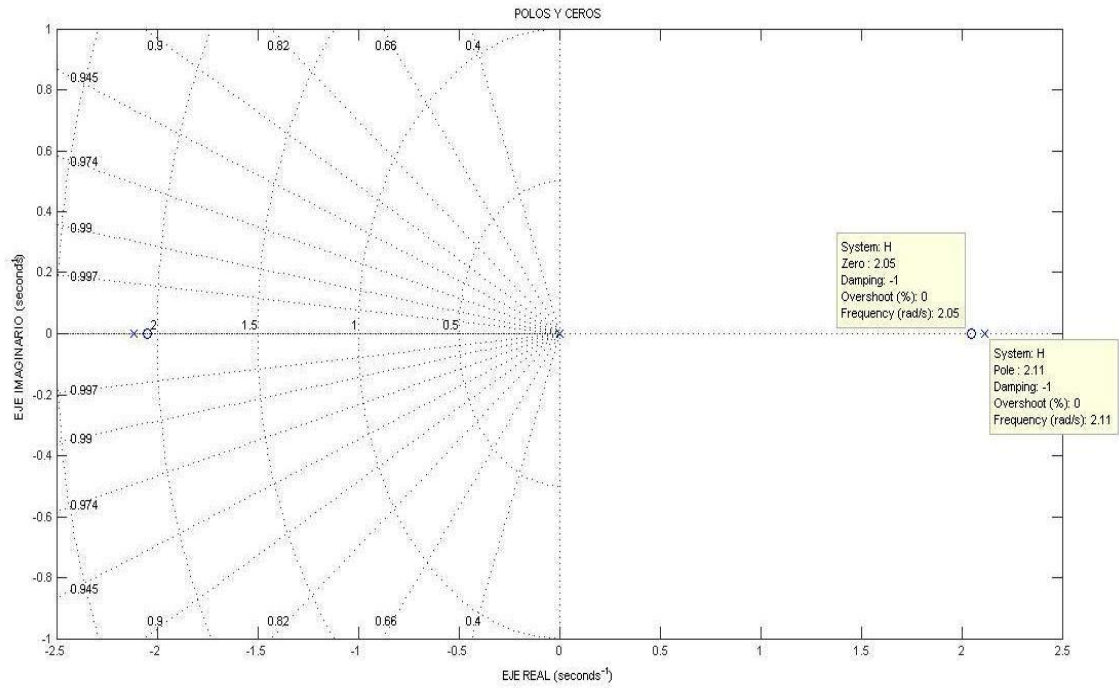
Entonces las ecuaciones del péndulo de Furuta linealizadas son:

$$C_1 \ddot{\alpha} + C_2 \ddot{\theta} + C_4 \dot{\alpha} = C_3 V_a \quad (7)$$

$$C_2 \ddot{\alpha} + C_3 \ddot{\theta} + C_6 \theta = 0 \quad (8)$$

Mediante el uso del software MATLAB, se obtiene los polos y ceros del sistema con el fin de observar la no linealidad del péndulo al poseer polos y ceros en el lado derecho del plano S como se observa en la figura 32.

Figura 32. Diagrama de polos y ceros.



Como se observa arriba en la figura, el sistema posee un cero ubicado en 2.05 y un polo ubicado en 2.11.

5. CONTROL PID

El controlador PID (Proporcional, Integral, Derivativo) es un controlador realimentado cuyo propósito es hacer que el error entre el la señal de referencia y la señal de salida de la planta, sea cero.

Un ejemplo muy sencillo que ilustra la funcionalidad básica de un PID, es cuando una persona entra a una ducha. Inicialmente abre la llave de agua caliente para aumentar la temperatura hasta un valor aceptable (también llamado “setpoint”). El problema es que pueda llegar el momento en que la temperatura del agua sobrepase este valor así que la persona tiene que abrir un poco la llave del agua fría para contrarrestar el calor y mantener el balance. El agua fría es ajustada hasta llegar a la temperatura deseada. En este caso, el humano es el que está ejerciendo el control sobre el lazo de control, y es el que toma las decisiones de abrir o cerrar alguna de las llaves.

Los controladores PID fueron inventados para simplificar las labores de los operadores y ejercer un mejor control sobre las operaciones.

5.1 ACCIONES DE CONTROL

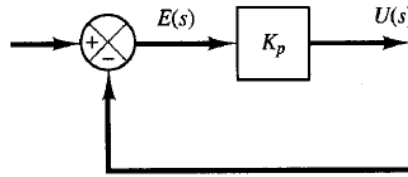
Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación y produce una señal de control que reducirá la desviación a cero o a un valor pequeño. La manera en la que un controlador automático produce la señal de control se denomina acción de control.

5.1.1 Controlador proporcional

Para un controlador con acción de control proporcional, como la que se aprecia en la figura 33, la relación entre la salida, del controlador $u(t)$ y la señal de error $e(t)$ es :

$$u(t) = K_p e(t)$$

Figura 33. Diagrama de bloques de un control proporcional



Fuente. [15]

O bien, en cantidades transformadas por el método de Laplace,

$$\frac{U(s)}{E(s)} = K_p$$

En donde K_p es la ganancia proporcional del controlador. El controlador proporcional es en esencia un amplificador con ganancia ajustable.

5.1.2 Controlador Integral

En un controlador con acción de control integral, el valor de la salida del controlador $u(t)$ se cambia a una razón proporcional a la señal de error $e(t)$:

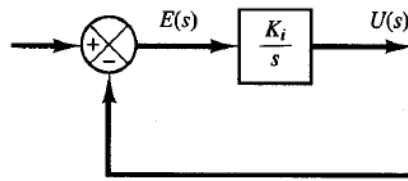
$$\frac{du(t)}{dt} = K_i e(t)$$
$$u(t) = K_i \int_0^t e(t) dt$$

En donde K_i es una constante ajustable. La función de transferencia del controlador integral es:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

Si se duplica el valor de $e(t)$ el valor $u(t)$ varía dos veces más rápido. Para un error de cero, el valor de $u(t)$ permanece estacionario. En ocasiones, la acción de control integral se denomina control de reajuste. En la figura 34 se observa un diagrama de bloques de un controlador integral.

Figura 34. Diagrama de bloque de un controlador integral



Fuente. [15]

5.1.3 Controlador Proporcional-Integral (PI)

La acción de control de un controlador PI se define como:

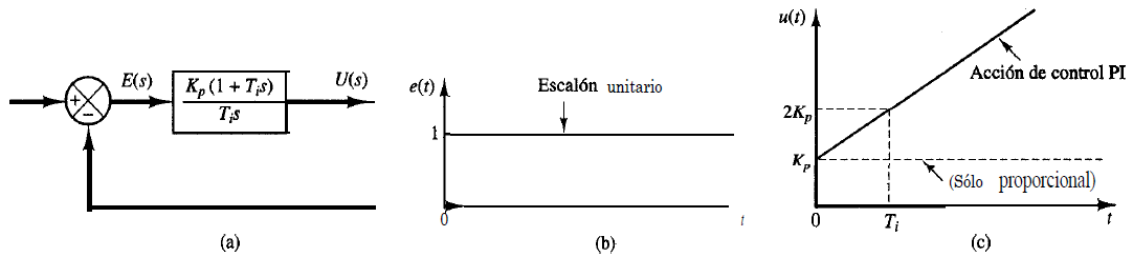
$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt$$

O la función de transferencia del controlador es:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right)$$

En donde K_p es la ganancia proporcional y T_i se denomina tiempo integral. Tanto K_p como T_i son ajustables. El tiempo integral ajusta la acción de control integral, mientras que un cambio en el valor de K_p afecta las partes integral y proporcional de la acción de control. El inverso del tiempo integral T_i se denomina velocidad de reajuste. La velocidad de reajuste es la cantidad de veces por minuto que se duplica la parte proporcional de la acción de control. La velocidad de reajuste se mide en términos de las repeticiones por minuto. La figura 35a muestra un diagrama de bloque de un controlador proporcional más integral. Si la señal de error $e(t)$ es una función escalón unitario, como se ve en la figura 35b, la salida del controlador $u(t)$ se convierte en lo que se observa en la figura 35c.

Figura 35. (a) Diagrama de bloques de un controlador PI; (b) Diagrama que muestra una entrada escalón; (c) salida del controlador



Fuente. [15]

5.1.4 Controlador Proporcional-Derivativo (PD)

La acción de control de un controlador proporcional-derivativo se define mediante

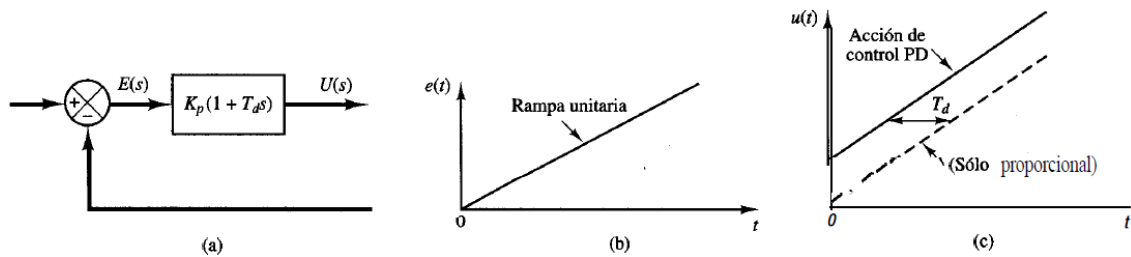
$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

Y la función de transferencia es:

$$\frac{U(s)}{E(s)} = K_p (1 + T_d s)$$

En donde K_p es la ganancia proporcional y T_d es una constante denominada tiempo derivativo. Tanto K_p como T_d son ajustables. La acción de control derivativa, en ocasiones denominada control de velocidad, ocurre donde la magnitud de la salida del controlador es proporcional a la velocidad de cambio de la señal de error. El tiempo derivativo T_d es el intervalo de tiempo durante el cual la acción de la velocidad hace avanzar el efecto de la acción de control proporcional. La figura 36a muestra un diagrama de bloques de un controlador proporcional-derivativo. Si la señal de error $e(t)$ es una función rampa unitaria como se aprecia en la figura 36b, la salida del controlador $u(t)$ se convierte en la que se muestra en la figura 36c. La acción de control derivativa tiene un carácter de previsión. Sin embargo, es obvio que una acción de control derivativa nunca prevé una acción que nunca ha ocurrido.

Figura 36. (a) Diagrama de bloque de un controlador PD; **(b)** y **(c)** diagramas que muestran una entrada rampa unitaria y la salida del controlador.



Fuente. [15]

Aunque la acción de control derivativa tiene la ventaja de ser de previsión, tiene las desventajas de que amplifica las señales de ruido y puede provocar un efecto de saturación en el actuador.

Hay que tener en cuenta que un control derivativo nunca se usa solo, debido a que sólo es eficaz durante periodos transitorios.

5.1.5 Controlador Proporcional-Integral—Derivativo

La combinación de una acción de control proporcional, una acción de control integral y una acción de control derivativa se denomina acción de control proporcional-integral-derivativa (PID). Esta acción combinada tiene las ventajas de cada una de las tres acciones de control individuales. La ecuación de un controlador con esta acción combinada se obtiene mediante:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

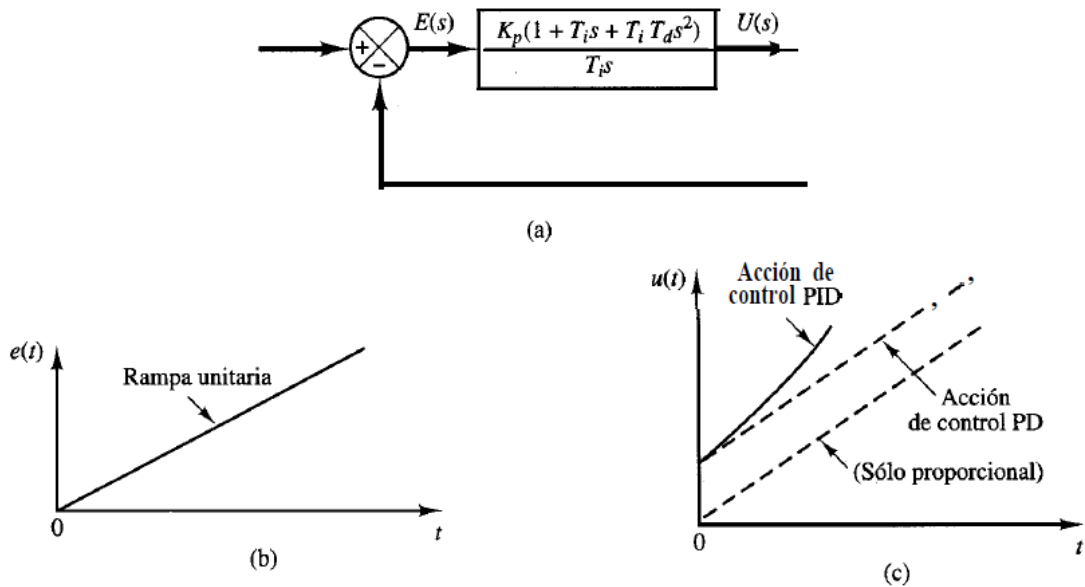
O la función de transferencia:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

En donde K_p es la ganancia proporcional, T_i es el tiempo integral y T_d es el tiempo derivativo. El diagrama de bloques de un controlador PID aparece en la

figura 37a. Si $e(t)$ es una función rampa unitaria, como la que se observa en la figura 37b, la salida del controlador $u(t)$ se convierte en la de la figura 37c.

Figura 37. (a) Diagrama de bloques de un controlador PID; (b) y (c) diagramas que muestran una entrada de rampa unitaria y la salida del controlador PID.



Fuente. [15]

6. IMPLEMENTACION

6.1 PARÁMETROS DE CONTROL

Angulo del brazo: es el ángulo medido por el encoder adjunto al motor.

Angulo del péndulo: Es el ángulo medido por el encoder adjunto al punto donde pivota el péndulo.

Error: Es la diferencia entre la variable medida y el setpoint. En este caso hace referencia a cuán lejos o cerca está el péndulo del setpoint o punto de equilibrio.

Setpoint: Es el punto donde el sistema se mantiene en equilibrio momentáneo, es decir colineal con el eje vertical perpendicular al plano de movimiento del brazo.

6.2 IMPLEMENTACION DE LOS SENSORES

6.2.1 Ubicación

Los dos sensores utilizados por el péndulo de Furuta se encuentran ubicados de la siguiente forma:

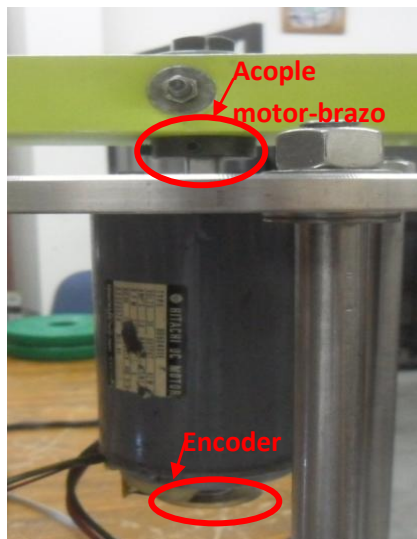
El primero se encuentra adjunto al eje de rotación del motor en la parte posterior del motor tal como se aprecia en la figura 38. Debido al hecho de que el brazo se encuentra adjunto al eje del motor, al sensar la posición del motor implícitamente se está sensando la posición del brazo, figura 39.

Figura 38. Ubicación del sensor en el motor.



Fuente. Autor

Figura 39. Acoplamiento del eje del motor con al brazo



Fuente: Autor

El segundo sensor se encuentra montado sobre un extremo del brazo y posee un sistema de acoplamiento mecánico que le permite conectarse con la barra del péndulo, tal como se observa en la figura 40. El punto donde el segundo sensor se acopla al péndulo, es aquel en el que pivota.

Figura 40. Ubicación del sensor del péndulo

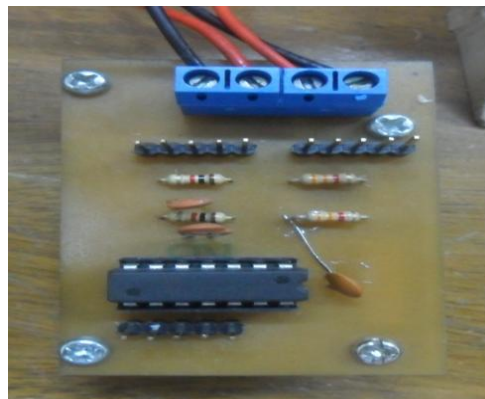


Fuente. Autor

6.2.2 Interface de sensores

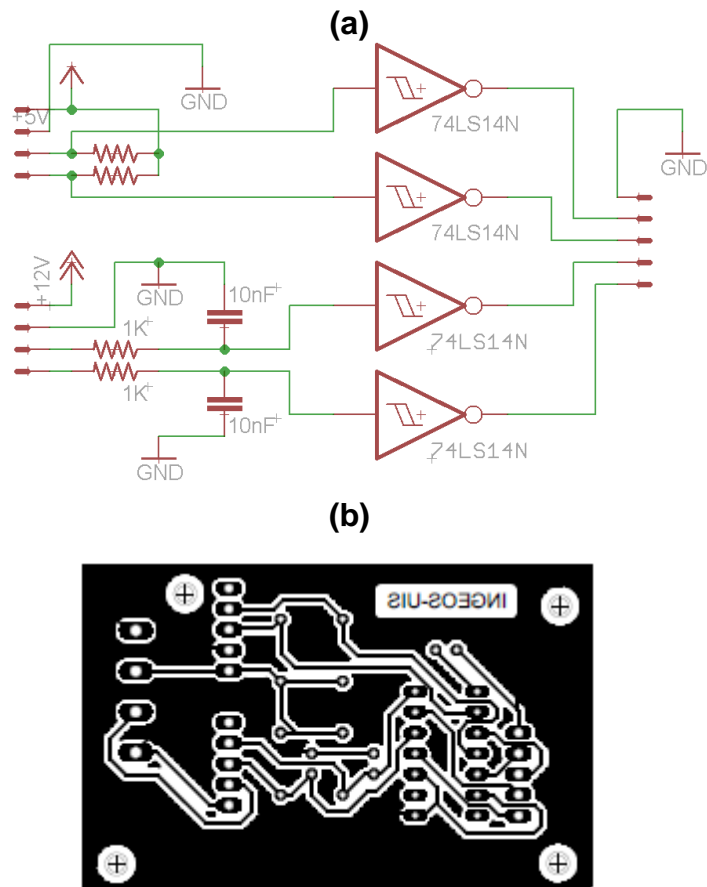
Para obtener los datos entregados por los sensores se hace necesaria la implementación de circuitos electrónicos adicionales con la finalidad de obtener señales de buena calidad y con el mínimo de ruido eléctrico. El circuito diseñado para tratar las señales de los encoder se compone de resistencias, capacitores y un integrado Schmitt trigger, como se aprecia en la figura 41. En la figura 42, se observa el diagrama esquemático de la tarjeta de encoder y el negativo de la PCB (Printed Circuit Board).

Figura 41. Encoder PCB



Fuente: Autor.

Figura 42. (a) Esquemático (b) negativo.

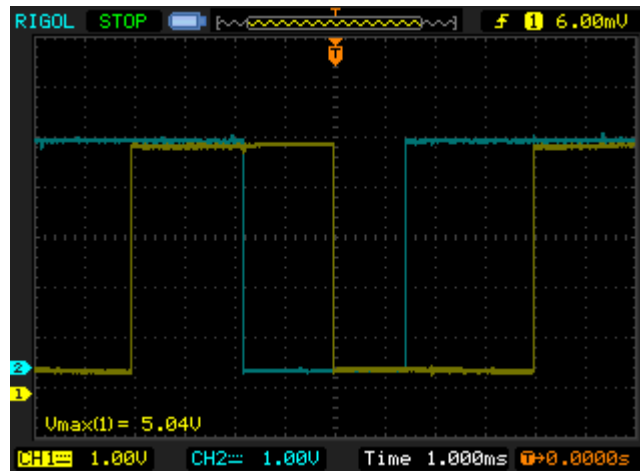


Fuente: Autor

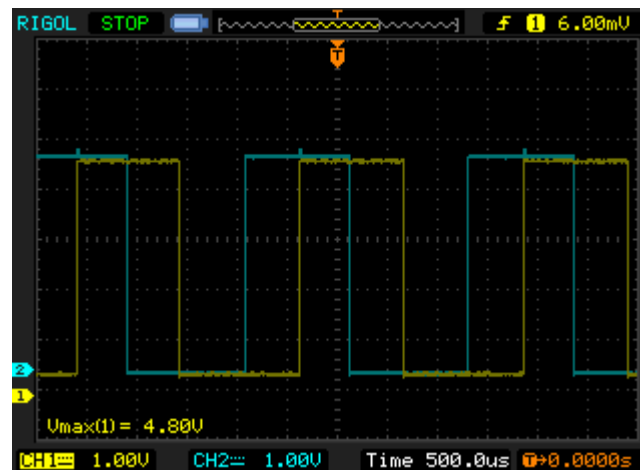
Las señales que provienen de los encoders, pasan a través de un filtro RC y luego a través de una compuerta Schmitt trigger (74LS14) (ver figura 42) que permite tener un pulso más cuadrado, con los flancos ascendente y descendente más marcados perpendicularmente, posteriormente éstas señales a la salida del Schmitt, llegan a los puertos de interrupción de Arduino donde mediante programación se hace la lectura de las señales provenientes de los encoders. Las señales del encoder del motor y péndulo se aprecian en la figura 43.

Figura 43. (a) Señal encoder motor o brazo, (b) señal encoder péndulo

(a)



(b)



Fuente.

6.3 IMPLEMENTACIÓN PUENTE H

6.3.1 Etapa de potencia

La etapa de potencia se compone principalmente por el pre-actuador, es decir el chip de puente H (LMD18200). Dicho circuito implementado se observa en la figura 44. En él, se aprecia la incorporación de los circuitos integrados optoaisladores, los cuáles permitieron corregir en gran medida el ruido eléctrico introducido por el motor DC a través de las fuentes de alimentación.

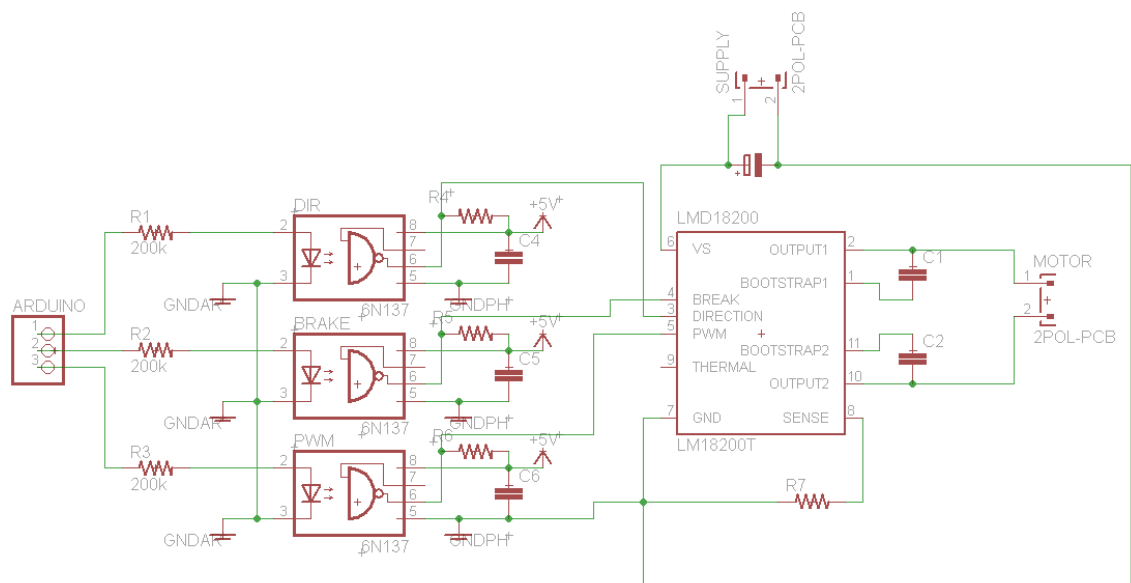
Figura 44. Puente H PCB

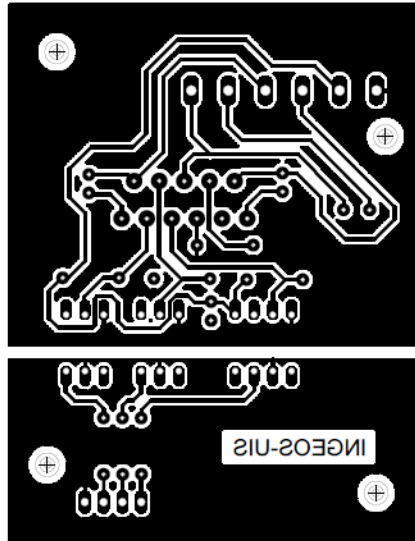


Fuente: Autor

En la figura 45 se observa el esquema eléctrico de conexión del puente H y el negativo de la board.

Figura 45. (a) Esquema eléctrico tarjeta puente H, (b) negativo PCB





Fuente: Autor

Las señales son enviadas desde el arduino hacia los optoaisladores los cuales invierten el valor de entrada, es decir, si el valor de entrada de un optoaislador es cero (0), a la salida entregará un uno (1), lo que significa que el optoaislador funciona como un inversor de señal. Hay uno por cada pin necesario para manejar el LMD18200, es decir que hay tres optoaisladores, uno para el pin de dirección (DIR), uno para el pin de freno (BRAKE) y uno para el pin de pwm (PWM). Las señales entregadas por los optoaisladores son recibidas por los pines de comando del LDM18200, y la salida del puente H es enviada a los terminales del motor.

7. DISEÑO ESTRUCTURAL DEL PENDULO INVERTIDO ROTATIVO

El diseño final de la planta dinámica de control resulta en un banco que se ajusta a las necesidades y exigencias del laboratorio, como son: fácil manejo, modular, de fácil transporte y pedagógico, el aspecto de la planta se observa en la figura 46.

Figura 46. Planta dinámica Péndulo Invertido Rotativo



Fuente: Autor

El péndulo invertido rotativo está compuesto por una serie de subconjuntos (ver figura 47), a mencionar:

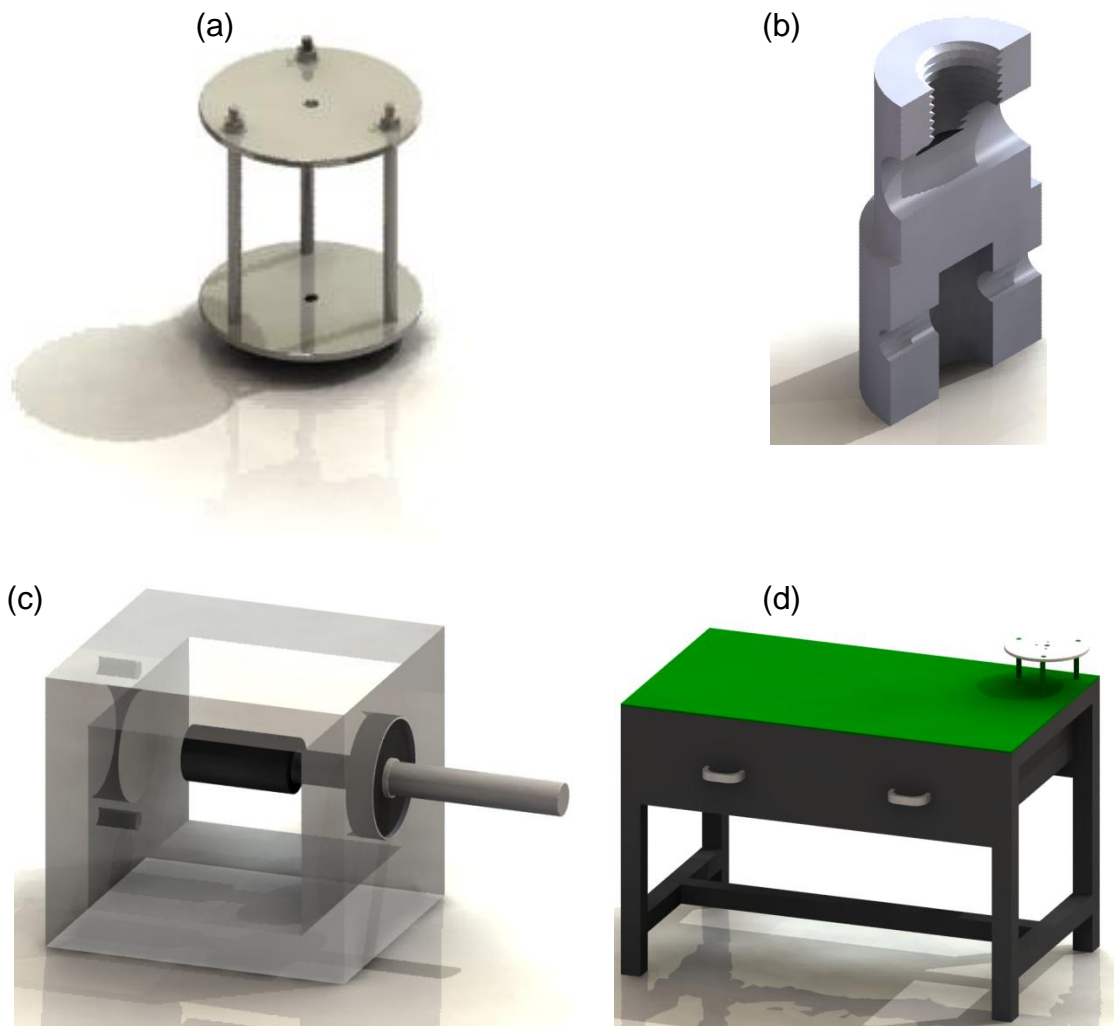
- Estructura soporte de motor.
- Acople motor brazo.
- Acople péndulo.
- Mesa de soporte de la planta.

7.1 Estructura soporte motor

La estructura que soporta el motor debe ser rígida, para soportar el peso del motor principalmente, pesada para resistir los cambios de dirección del motor, y

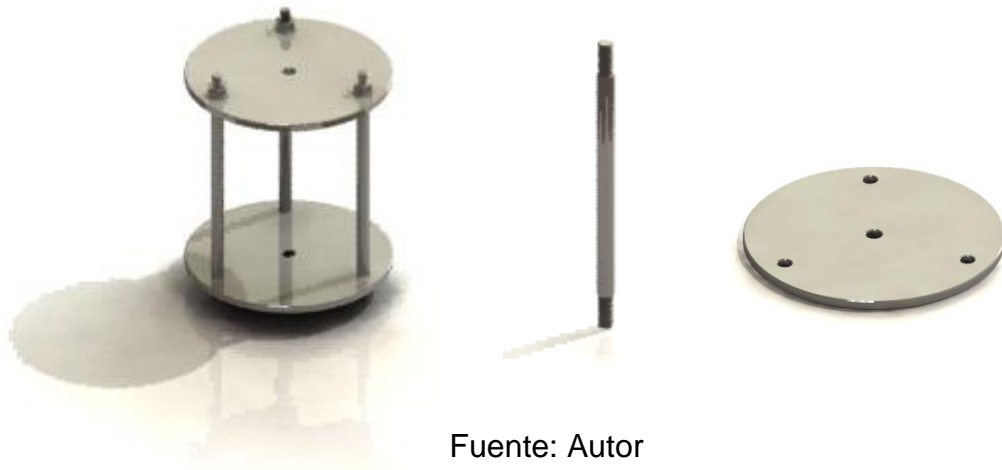
el torque transmitido a la estructura con el fin de evitar volcadura. La estructura está compuesta por tres espigas ubicados a 120° uno del otro y que sujetan dos discos, uno de los cuales soporta el motor, tuercas de seguridad permiten asegurar los discos a los espigas por ambos extremos, el material de la estructura es de acero inoxidable (ver figura 48).

Figura 47. Subconjuntos del péndulo invertido rotativo.



Fuente: Autor

Figura 48. (a) Estructura base, (b) Espigo, (c) Disco



Fuente: Autor

El acero inoxidable de los discos fue previamente cortado con plasma para obtener una forma similar a un círculo. El diseño circular de los discos se debe al movimiento circular que realiza el brazo adjunto al motor, debido al acabado del corte y la apariencia inicial del material, este fue sometido a un proceso de mecanizado, el cual consistió en refrentar ambas caras de los discos y un posterior cilindrado sobre la periferia de los mismos, dando así la apariencia final que se observa en la figura 49.

Los espigos provienen de una barra de acero inoxidable de media pulgada de diámetro (1/2 in), la cual fue cortada en tres secciones de igual longitud, dicha longitud fue considerada al tener en cuenta la distancia entre los discos, de manera que el motor DC se pudiera montar y desmontar fácilmente, ya que este se encuentra ubicado entre un disco y el otro. Los extremos de los espigos fueron sometidos a un proceso de roscado (ver figura 50) con el objetivo de juntar los discos a los espigos mediante tuercas de seguridad

Figura 49. Acabado final de la plataforma



Fuente: Autor

Figura 50. Espigo roscado

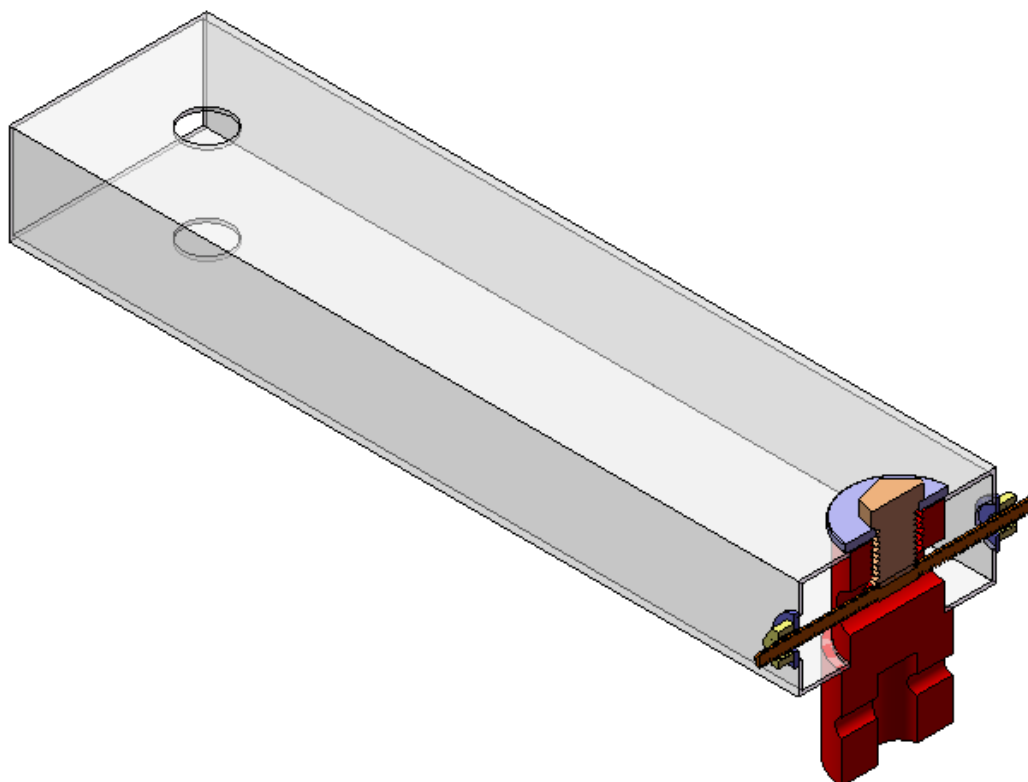


Fuente: Autor

7.2 Sistema de acople motor-brazo

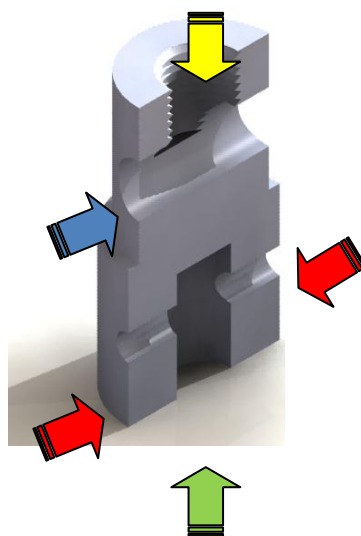
El acople entre el motor y el brazo cumple la función de transmitir el torque y el movimiento del motor hacia el brazo, de manera que el acople debe ser lo más rígido posible sin dejar holguras o espacio a tolerancias, y que transmita la mayor cantidad de energía mecánica hacia el brazo sin disiparla en el proceso, en la figura 51 se observa el diseño del acople y la ubicación.

Figura 51. Vista en corte sistema de acople motor-brazo



Fuente: Autor

Figura 52. Acople motor- brazo



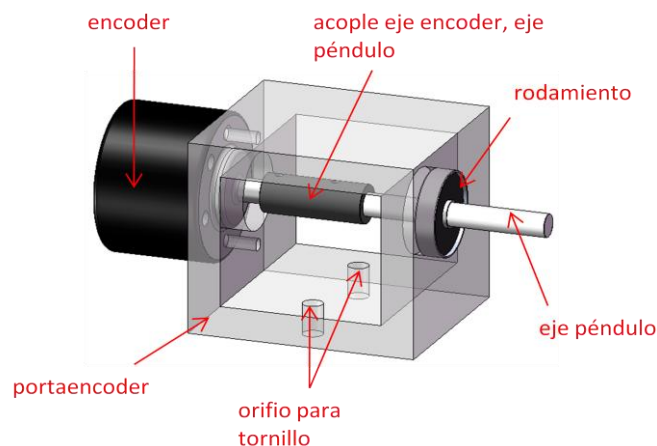
Fuente: Autor

En la figura 52 se observa la ubicación en la que se acoplan los componentes como el eje del motor (flecha verde), tornillo prisionero (flecha roja) el permite la sujeción del acople al eje del motor, espárrago (flecha azul) es el que permite acoplar en dirección horizontal al brazo, tuerca (amarilla) permite acoplar en forma vertical el brazo.

7.3 Sistema de acople encoder-péndulo

Es el conjunto de elementos que permiten acoplar el eje del encoder al punto de pivote del péndulo, permitiendo así medir la posición angular del mismo. El arreglo del sistema de acople se observa en la figura 53.

Figura 53. Sistema de acople encoder-péndulo

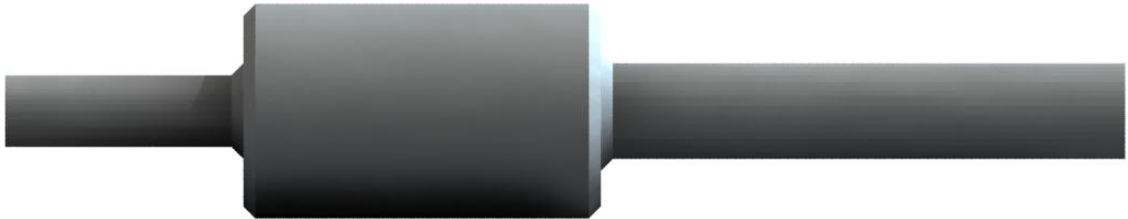


Fuente: Autor

El encoder es atornillado con tornillos de cabeza bristol de 4 mm a una estructura de perfil cuadrado hecha en acrílico a la que se le ha denominado como portaencoder, este último se encuentra montado en un extremo del brazo mediante tornillos de sujeción. Un acople flexible permite unir el eje del encoder con el eje en que pivota el péndulo, el cual permite que al tiempo en que el eje del péndulo gire, el del encoder también lo haga. El eje del péndulo reposa sobre un rodamiento radial en el que puede girar libremente. El eje se muestra en la figura 54, y se diseñó asumiendo que se comporta como una viga

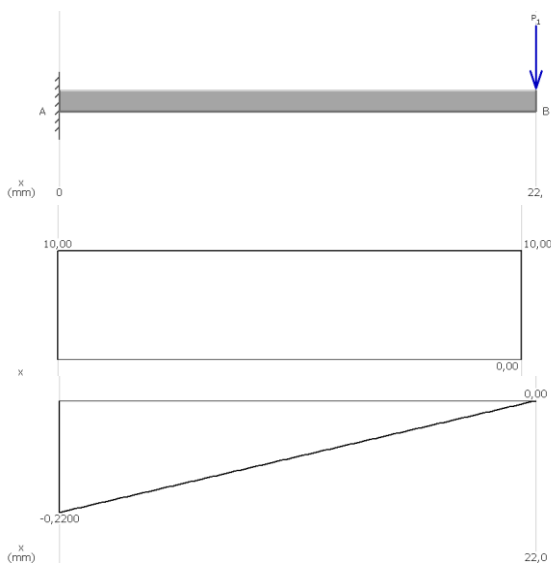
empotrada. El diagrama de carga, cortante y momentos se aprecian en la figura 55. Adicionalmente se realizó una simulación de los esfuerzos en COSMOSXpress donde se aprecia la distribución de la tensión al someter el eje a una carga estática de 10 Newton (10N) en el extremo derecho del mismo, en la figura 56; se aprecia la distribución de los esfuerzos.

Figura 54. Eje péndulo



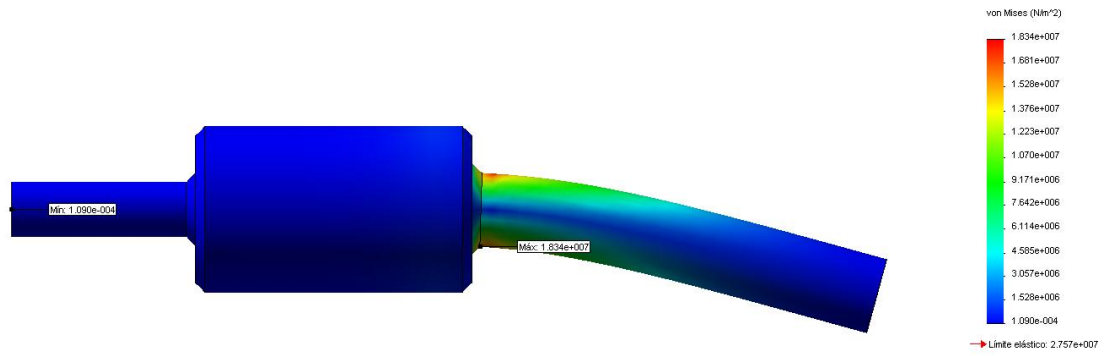
Fuente: SolidWorks 2007

Figura 55. Diagramas de carga, cortante y momento del eje



Fuente: MDSolids

Figura 56. Distribución de esfuerzos del eje



Fuente: COSMOSXpress

8. SISTEMA DE CONTROL

Este capítulo se explica cómo se implementó el sistema de control embebido para el péndulo invertido rotativo, y que encierra características, funciones de la tarjeta y sus fundamentos operativos, además se detalla cómo se realizó la organización del código fuente del microcontrolador, una descripción de cómo se integra el hardware y software, es decir, el controlador con los sensores y actuadores por medio del software, se explica la programación del Arduino y las funciones que se utilizaron.

8.1 SISTEMA DE CONTROL EMBEBIDO

Se conoce como sistema de control embebido a un circuito electrónico computarizado que está diseñado para cumplir una función específica en un producto. La inteligencia artificial, secuencias y algoritmos de un sistema embebido, están residentes en la memoria de una pequeña computadora denominada microcontrolador [16].

8.1.1 Principales características de los sistemas embebidos

- Los sistemas embebidos son diseñados para una aplicación específica, es decir, estos sistemas realizan un grupo de funciones previamente definidas y una vez el sistema es diseñado, no se puede cambiar su funcionalidad. Por ejemplo, el control de un ascensor siempre realizará las mismas acciones durante su vida útil.

- Debido a su interacción con el entorno los ES (Embebed Systems, por sus siglas en inglés) deben cumplir estrictamente restricciones temporales. El término Sistemas de Tiempo Real es utilizado para enfatizar este aspecto.

- Los ES son heterogéneos, es decir, están compuestos por componentes como Hardware y Software. Los componentes Hardware, como ASICs y Dispositivos Lógicos Programables (PLD) proporcionan la velocidad de ejecución y el consumo de potencia necesarios en algunas aplicaciones.
- Los ES tienen grandes requerimientos en términos de confiabilidad. Errores en aplicaciones como la aviación y el automovilismo, pueden tener consecuencias desastrosas.

8.2 MICROCONTROLADOR ATMEGA 2560 DE ATMEL

Como se definió en un principio en los objetivos del proyecto, se hace uso de una tarjeta de desarrollo basada en la familia de microcontroladores ATMEGA. Se decide utilizar una tarjeta de desarrollo basada en la familia de microcontroladores de Atmel [18], un chip Atmega de arquitectura AVR de 8 bits. Se hace uso de la tarjeta de desarrollo Arduino MEGA 2560, que integra un microcontrolador ATMEGA 2560, la cual contiene las siguientes características técnicas de operación:

Memoria flash: 256 kbytes

Máxima frecuencia de operación: 16 Mhz

CPU: 8-bit AVR

Pines I/O máx: 86

Interrupciones externas: 32

SPI (Serial Peripheral Interface): 5

TWI (Two Wire Interface): 1

UART (Universal Asynchronous serial Receiver and Transmitter): 4

Canales de conversión Analógica-Digital (ADC): 16

Resolución ADC (bits): 10

SRAM (Kbytes): 8

EEPROM (Kbytes): 4

Rango de temperatura (°C): -40 a 85

Voltaje de operación (Vcc): 1.8 a 5.5

Timers: 6

Canales PWM: 15

En el Anexo X, se muestra la hoja de datos del microcontrolador utilizado, en este podemos ver la asignación de pines para los encapsulados, así como las características térmicas, eléctricas y planos del microcontrolador ATMEGA 2560.

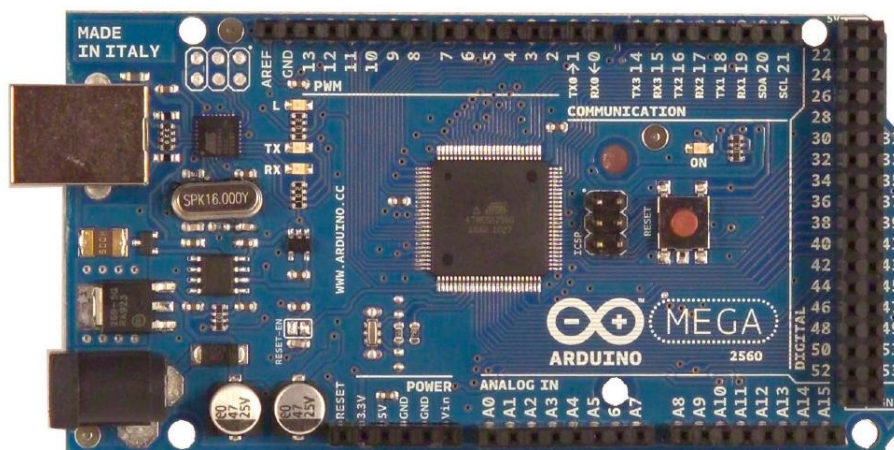
En este trabajo de grado se implementó la tarjeta de desarrollo Arduino MEGA 2560, que incorpora un microcontrolador Atmega 2560, la cual posee los componentes y dispositivos necesarios para su operación, haciendo más fácil la aplicación del microcontrolador, en vista de que la tarjeta Arduino fue diseñada y construida junto con su software, en el cual se incluyen gran cantidad de librerías que permiten utilizar el microcontrolador Atmega 2560 con una programación de alto nivel basado en C++ y cuyo ambiente de programación está basado en processing [17]. En el **Anexo C** se aprecia un esquemático del ATmega2560.

8.3 TARJETA ARDUINO MEGA 2560

8.3.1 Alimentación del Arduino

El Arduino Mega puede ser alimentado por conexión USB o por una alimentación externa como se aprecia en la figura 57. La alimentación es seleccionada automáticamente.

Figura 57. Tarjeta Arduino Mega 2560.



Fuente: <http://arduino.cc/en/Main/ArduinoBoardMega2560>

La alimentación externa puede venir de un adaptador de AC a DC, o de una batería. El adaptador puede ser conectado insertando el conector en el jack de alimentación. La batería puede ser conectada a los pines de tierra (GND) y voltaje de entrada (Vin) que se encuentran sobre la tarjeta.

La tarjeta puede operar con una alimentación externa de 6 a 20 voltios. Si la tarjeta es alimentada con menos de 7v, sin embargo, con el pin de 5v puede suministrar menos de cinco voltios y la tarjeta puede comportarse de manera inestable. Si se usan más de 12v en la alimentación, el regulador de voltaje puede sobrecalentarse y dañar la tarjeta. El rango de alimentación recomendado es de 7 a 12 voltios.

La tarjeta Mega 2560 difiere de las tarjetas predecesoras en que no usa un controlador serial a FTDI USB. En vez de eso, se caracteriza por tener un chip ATmega8U2 programado como convertidor serial a USB.

8.3.2 Descripción de pines

Vin: El voltaje de entrada a la tarjeta Arduino cuando está usando una fuente de alimentación externa (diferente a los 5v de una conexión USB u otra fuente de alimentación regulada). Se puede suministrar voltaje a través de este pin.

5v: es un pin de alimentación regulada usada para alimentar el microcontrolador y otros componentes en la tarjeta.

3.3v: Fuente de voltaje generada por el regulador incorporado en la tarjeta. Maneja una corriente máxima de 50 mA.

GND: Pines de tierra.

Memoria

Tiene 256 KB de memoria flash para almacenamiento de código (de los cuales 8 KB son usados por el gestor de arranque), 8 KB de SRAM y 4 KB de EEPROM.

Entradas y Salidas

Cada uno de los 54 pines digitales en el Mega pueden ser usados como entradas o salidas, usando las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Todos los pines operan a 5v. Cada pin puede proporcionar o recibir una corriente máxima de 40mA y tiene una resistencia interna de pull-up (desconectada por defecto) de 20-50 KOhms. Además algunos pines tienen funciones especializadas:

Serial: 0 (RX) y 1 (TX); Serial 1:19 (RX) y 18 (TX); Serial 2:17 (RX) y 16 (TX); Serial 3:15 (RX) y 14 (TX). Usados para recibir (RX) y transmitir (TX) información serial de tipo TTL (Transistor-Transistor Logic). Los pines 0 y 1 también están conectados a los pines correspondientes del chip serial ATmega8U2 USB a TTL.

Interrupciones Externas: 2 (interrupción 0), 3 (interrupción 1), 18 (interrupción 5), 19 (interrupción 4), 20 (interrupción 3) y 21 (interrupción 2). Estos pines pueden ser configurados para activar una interrupción en un valor bajo (LOW),

en un flanco ascendente o descendente, o en un cambio de valor, con la función `attachInterrupt()`.

PWM: Proporciona una de salida de PWM con una resolución de 8-bit con la función `analogWrite()` en los pines 0 a 13.

SPI: 50 (MISO), 51(MOSI), 52 (SCK), 53 (SS). Estos pines soportan comunicación SPI (Serial Peripheral Interface) usando la librería SPI.

LED: pin 13. Hay un led conectado al pin digital 13. Cuando el pin está en valor alto (HIGH), el LED se enciende, cuando está en valor bajo (LOW), el LED se apaga.

La tarjeta Mega2560 tiene 16 entradas análogas, cada una de las cuales proporcionan una resolución de 10-bit. Por defecto estos pines miden desde tierra a 5v, aunque es posible cambiar el límite superior de este rango usando el pin AREF y la función `analogReference()`.

Otros pines en la tarjeta son:

AREF: Voltaje de referencia para las entradas análogas.

Reset: Resetea el microcontrolador. Generalmente se usa para adicionar un botón de reset a tarjetas que bloquean el botón que se encuentra sobre la tarjeta.

Comunicación

El Arduino Mega2560 posee un número de facilidades para comunicarse con un computador, otro Arduino, u otros microcontroladores. El ATmega2560 proporciona cuatro puertos UARTs para comunicación serial. El software de Arduino posee una ventana que permite visualizar información serial en formato de texto que es enviada o recibida de la tarjeta. Los LEDs RX y TX en la tarjeta parpadearan cuando información es transmitida a través del chip ATmega8U2 y la conexión USB al computador (pero no para comunicación serial en los pines 0 y 1).

8.4 PROGRAMACIÓN DEL ARDUINO

El Arduino Mega puede ser programado con el software de Arduino. El ATmega2560 viene prequemado con el gestor de arranque que permite cargar nuevo código a la tarjeta sin necesidad de usar un programador externo. Se comunica usando el protocolo STK500 original.




8.5 PROGRAMACIÓN DEL MICROCONTROLADOR


8.5.1 Entorno de programación de Arduino (IDE)


El entorno de programación de Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.


El software escrito usando Arduino es denominado “sketch” (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/reemplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y el muestreo por serial.

En la barra de herramientas se encuentran los siguientes comandos:

-  **Verify/Compilar:** Revisa el código en busca de errores.
-  **Upload/Cargar:** Compila el código y lo carga a los pines de E/S de la tarjeta Arduino.
-  **New/Nuevo:** Crea un nuevo programa.

 **Open/Abrir:** Presenta un menú de todos los programas guardados en la librería de programas, al hacer click sobre uno de estos programas el software lo abrirá en la ventana actual.

 **Save/Guardar:** Guarda el programa actual.

 **Serial monitor:** Abre el muestreo en pantalla del serial.

Existen otros comandos adicionales que se encuentran en los menús: file/archivo, edit/editar, sketch/programa, tools/herramientas, help/ayuda. Estos menús son sensibles al contexto, lo que significa que estarán disponibles sólo los elementos relevantes para la tarea que esté realizando en ese momento.

8.5.2 Funciones básicas

Funciones digitales

pinMode(): Configura el pin especificado para comportarse como una entrada (INPUT) o una salida (OUTPUT).

digitalWrite(): Escibe un valor de HIGH o LOW hacia un pin digital. Si el pin ha sido configurado como salida (OUTPUT) con la función pinMode(), su voltaje será establecido a un valor de 5 voltios para HIGH y 0 voltios para LOW o tierra.

Si el pin es configurado como entrada, al escribir un valor de HIGH con digitalWrite() habilitará una resistencia interna de 20KOhms conectada en pullup. Escribir un valor de LOW inhabilitará la resistencia.

digitalRead(): Lee el valor de un pin digital especificado que sea configurado como entrada, leerá un valor de HIGH o LOW.

Funciones analógicas

analogRead(): Lee el valor de un pin analógico especificado. La tarjeta Mega tiene 16 pines de lectura analógica, con una resolución de convertidor análogo a digital de 10-bits. Esto significa que mapeará voltajes de entrada de 0 a 5 voltios en valores enteros entre 0 y 1023. Esto conlleva a una resolución de

lectura de: 5 voltios/1024 unidades o, 0.0049 voltios (4.9 mV) por unidad. El rango de entrada y la resolución puede ser cambiada usando la función `analogReference()`. Leer una entrada analógica toma alrededor de 100 microsegundos (0.0001s).

`analogWrite()`: Escribe un valor analógico (onda PWM) a un pin especificado. Puede ser usado para encender un LED variando su brillo, o para controlar un motor a varias velocidades. Luego de llamar la función `analogWrite()`, en ese pin se generará una onda cuadrada estable con un ciclo de carga (duty cycle) especificado hasta la siguiente llamada a la función `analogWrite()` (o a las funciones `digitalRead()` o `digitalWrite()` en el mismo pin). La frecuencia de la señal PWM es aproximadamente 490 Hz.

8.5.3 Estructura del programa

Función `void setup()`: Esta función es llamada cuando inicia el programa. Se utiliza para inicializar variables, declarar el estado de los pines, inicializar librerías, etc. La función `setup` correrá solo una vez, después de cada encendido o reinicio de la tarjeta.

Función `void loop()`: Después de crear la función `setup()`, la cual inicializa y establece los valores iniciales, la función `loop` hace precisamente lo que su nombre sugiere, corre una y otra vez sucesivamente, permitiéndole al programa cambiar y responder. Se usa activamente para controlar la tarjeta.

8.5.4 Registros de control

Los registros permiten la rápida manipulación de los pines de entrada y salida del microcontrolador en la tarjeta Arduino. El microcontrolador ATmega2560 posee once puertos desde la A a la L a excepción de la I.

Cada puerto es controlado por tres registros, los cuales también son variables definidas en el lenguaje de Arduino. El registro DDR determina si el pin es una entrada (INPUT) o salida (OUTPUT). El registro PORT controla si el pin está en un estado lógico uno (HIGH) o cero (LOW), y el registro PIN lee el estado de los pines que fueron configurados como entradas con la función `pinMode()`.

Los registros DDR y PORT pueden ser ambos de lectura y escritura, mientras que el registro PIN es sólo de lectura.

Por ejemplo para referirse al puerto C que corresponde a los pines 53 a 60 del chip ATmega 2560 (ver Anexo I) y a los pines 30 a 37 de la tarjeta Arduino, el registro DDRC configura los pines del puerto C ya sea como entrada o salida. DDR significa dirección del registro de datos (Data Direction Register por sus siglas en inglés), el registro PORTC controlará que pines tendrán un estado lógico de 0 o 1, y el registro PINC leerá la actividad en los pines del puerto C que correspondan a esos 8-bits.

8.6 DISEÑO DEL ALGORITMO DE CONTROL

En esta etapa se examina y analiza los componentes del programa de control que permitirá convertir un sistema inestable como lo es el péndulo invertido rotativo, en uno estable. El primer paso será diseñar e implementar un programa que permita leer la información del encoder. Posteriormente diseñar un programa que permita controlar la dirección y el PWM que se envía al motor.

8.6.1 Lectura del encoder

El programa de lectura de encoder deberá cumplir dos requisitos básicos, que sea capaz de leer los pulsos enviados por el encoder, almacenarlos de una forma que pueda indicar si la posición está en incremento o decremento, y convertir esos pulsos almacenados a un valor en ángulo.

Función attachInterrupt()

Especifica una función a llamar cuando ocurre una interrupción externa. Reemplaza cualquier función previa que fue adjuntada a la interrupción. La tarjeta Arduino mega posee diez pines para interrupción externa, de los cuales se usan cuatro para los encoder del péndulo invertido rotativo. La función tiene tres parámetros:

Interrupción: El número de la interrupción a utilizar. Para el encoder del brazo se utilizaron las interrupciones 3 y 2 que se encuentran en los pines 20 y 21 de la tarjeta mega. Para el encoder del péndulo se utilizaron las interrupciones 5 y 4 que corresponden a los pines 18 y 19 de la tarjeta.

Función: Este parámetro hace referencia al tipo de función que será llamada cuando se active la interrupción. La descripción de la función utilizada se hace más adelante.

Modo: Este parámetro define cuando debería ser ejecutada o activada la función. Existen cuatro opciones o modos de activación.

LOW: Activa la interrupción cuando sea que el pin este en un estado LOW.

RISING: Activa la interrupción cuando sea que en el pin pasa de un estado LOW a HIGH, es decir que se active en el flanco ascendente.

FALLING: Activa la interrupción cuando sea que en el pin pasa de un estado de HIGH a LOW, es decir que se active en el flanco descendente.

CHANGE: Activa la interrupción cuando sea que el pin cambia de valor.

Este último modo de activación es el que se utiliza en el programa para el control del péndulo invertido, ya que permite aumentar la resolución del encoder a cuatro veces el valor nominal. Es decir que para una resolución del encoder del brazo de 400 ppr (pulsos por revolución) con el software se obtendrá una resolución de 1600 ppr, debido a que la una interrupción se ejecuta dos veces por canal de el encoder. Para el encoder del péndulo con una resolución de 2500 ppr, se tiene que con el programa se aumenta a una resolución de 10000 ppr.

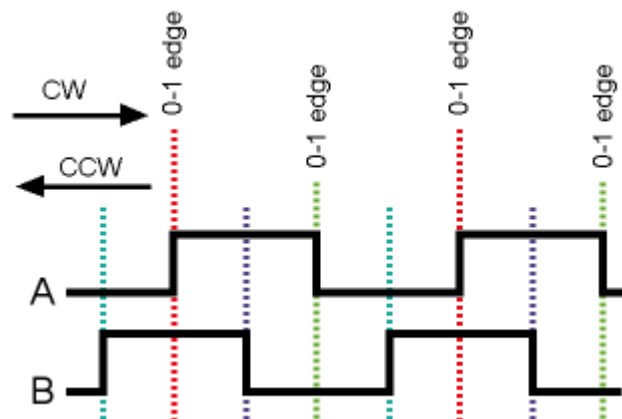
Funciones doA () y doB ()

Las funciones doA () y doB() son las encargadas de leer los pulsos provenientes de cada canal del encoder, los cuales han sido denominados como canal A y canal B, pero como son dos encoders, se especifica que para el encoder del brazo se tendrán las funciones doA1() y doB1() que

corresponden al canal A del encoder 1 y el canal B del encoder 1 respectivamente. Lo mismo aplica para el encoder del péndulo por lo tanto las funciones encargadas del dicho encoder son: doA2() y doB2() respectivamente.

Éstas funciones comparan cual canal cambio de estado lógico primero con respecto al otro y, si la secuencia del cambio corresponde a un sentido horario se incrementará el valor de una variable que se encarga de llevar la cuenta de los pulsos, si la secuencia corresponde a un sentido antihorario, dicha variable disminuirá su valor. La figura 58 se muestra un diagrama explicativo de cómo funciona el programa para leer el encoder.

Figura 58. Diagrama explicativo para leer encoder.



Fuente: <http://www.arduino.cc/playground/Main/RotaryEncoders>

Cuando la función encuentra una transición de bajo (LOW) a alto (HIGH) en el canal A, la función pasa a chequear si el canal B está en un estado HIGH o LOW luego incrementa o disminuye la variable según la secuencia del cambio que indicará el sentido de rotación.

Las variables donde se almacenan los pulsos provenientes de los encoders son: count_1 y count_2, para el encoder 1 (brazo) y 2 (péndulo) respectivamente, las cuales serán utilizadas para convertir el valor a ángulo y determinar la posición del brazo y el péndulo.

8.6.2 Control del puente H LMD18200

Como se explicó con anterioridad el chip de puente H cuenta con tres pines que permiten manipular la dirección, carga del PWM y el freno instantáneo del motor respectivamente. La forma en que a continuación se explica la configuración, si bien es un tanto más compleja que el Arduino, es más confiable, aunque requiere un estudio más profundo de la programación del chip ATmega2560.

Estos pines se configuran asignándoles estados lógicos de 0 (LOW) o 1 (HIGH). Para configurar dichos pines se recurre al uso de los registros de control previamente mencionados (DDR y PORT), ya que la función digitalWrite del arduino es de respuesta lenta.

Los pines de control del motor se encuentran conectados al arduino en los pines digitales 8, 7 y 6, que corresponden a DIR, PWM y BRAKE respectivamente. Los pines 6, 7 y 8 del arduino corresponden a los pines 15, 16 y 17 del chip ATmega (ver anexo I), a su vez estos pines corresponden a los bits PH3, PH4 y PH5 del puerto H.

Activación del registro PWM

Aquí se identifica el timer a utilizar para activar la salida pwm en el bit PH5, según el esquemático del anexo (), allí se ve que corresponde al Timer 4 y el registro de salida del pwm es el OCR4A (Output Compare Register del timer 4)

Para activar el Timer se necesita configurar dos registros el TCCR4A (Timer Counter Control Register A) y el TCCR4B (Timer Counter Control register).

Para el TCCR4A se tiene:

$$\text{TCCR4A} |= (1 \ll \text{COM4A1}) | (1 \ll \text{WGM40})$$

En la línea anterior se le asigna un uno al bit que corresponde al COM4A1 (Compare Output Mode) y al bit WGM40 (Waveform Generation Mode).

$$\text{TCCR4B} |= (1 \ll \text{CS41}) | (1 \ll \text{CS40})$$

En la línea anterior se le asigna un uno al bit que corresponde al CS41 (Clock Source bit 1) y al bit CS40 (Clock Source bit 0).

Para una descripción detallada de cómo se configura el Timer4 remítase al anexo F.

Configuración de los pines de control

El primer paso es establecer el modo en que trabajarán los pines de control, tomando en cuenta que las señales de control del motor serán desde el arduino hacia el puente H, es lógico que los pines utilizados del puerto H serán salidas de la tarjeta, por lo tanto para configurarlos como salidas se recurre al registro de dirección de datos del puerto H (DDRH), configurando los bits PH3, PH4, PH5 como salidas. Se usa el sistema hexadecimal para la configuración por ser más corto y sencillo y claro de escribir.

```
DDRH=0x38
```

Función motor_CW ()

Esta función es de tipo de dato void (vacío), es decir que no toma ni regresa ningún valor, tan solo se encarga de configurar los pines de dirección y pwm para que el motor gire en sentido horario a un ciclo de carga de pwm determinado, y lo hará al momento en que la función sea llamada en el programa.

Se hace uso del registro PORTH que permitirá configurar los pines DIR, PWM y BRAKE conectados al Arduino.

```
Void motor_CW ()
```

```
{ PORTH=0x30; }
```

Teniendo en cuenta que los optoaisladores se comportan como inversores es decir que el valor de salida que entregan es inverso al valor de entrada, es de vital importancia configurar el valor de entrada al optoaislador de una manera en que el valor de salida corresponda a un giro horario a un pwm dado.

Función motor_CCW ()

Esta función es de tipo de dato void (vacío), es decir que no toma ni regresa ningún valor, tan solo se encarga de configurar los pines de dirección y pwm para que el motor gire en sentido anti-horario a un ciclo de carga de pwm determinado, y lo hará al momento en que la función sea llamada en el programa.

Se hace uso del registro PORTH que permitirá configurar los pines DIR, PWM y BRAKE conectados al Arduino.

```
Void motor_CCW ()
```

```
{ PORTH=0x10; }
```

Teniendo en cuenta que los optoaisladores se comportan como inversores es decir que el valor de salida que entregan es inverso al valor de entrada, es de vital importancia configurar el valor de entrada al optoaislador de una manera en que el valor de salida corresponda a un giro anti-horario a un pwm dado.

Función motorStop ()

Esta función es de tipo de dato void (vacío), es decir que no toma ni regresa ningún valor, tan solo se encarga de configurar el pin de freno (BRAKE) , para que se detenga bruscamente en el momento requerido, y lo hará al momento en que la función sea llamada en el programa.

Se hace uso del registro PORTH que permitirá configurar el pin BRAKE conectado al Arduino.

```
Void motorStop ()
```

```
{ PORTH=0x28; }
```

Teniendo en cuenta que los optoaisladores se comportan como inversores es decir que el valor de salida que entregan es inverso al valor de entrada, es de vital importancia configurar el valor de entrada al optoaislador de una manera en que el valor de salida corresponda al de un freno instantáneo.

8.7 PROGRAMACIÓN DEL CONTROLADOR PID BASADO EN ISR ()

Aquí se explica la estructura del controlador PID, el corazón del programa, el cual se optó por realizarlo basado en el uso de interrupciones o ISR en vez del uso de la librería PID del Arduino.

El servicio de rutina por Interrupción o ISR (Interrupt Service Routine), es una señal especial enviada a la CPU que básicamente lo que hace es interrumpir el flujo del programa actual, y saltar hacia una dirección diferente temporalmente antes de regresar a lo que estaba haciendo previamente. Se recomienda fuertemente que el lector realice una inspección detallada de los anexo F para entender la configuración de la ISR().

8.7.1 Configuración de la ISR ()

Las interrupciones se realizan al configurar un Timer. Para realizar la interrupción primero se debe escoger el vector de interrupción, ya que existen varios tipos de interrupciones. El vector de interrupción que se utilizó fue de comparación COMPA (Compare Match en el bit A), es decir compara la cuenta del Timer con el valor del OCR (Output Compare Register). Para tal fin se recurre a la programación del Timer 3, este posee veinte fuentes independientes de interrupción, dentro de las cuales se encuentra la interrupción por comparación. Las líneas de configuración se muestran a continuación:

```
TCCR3A=0
```

```
TCCR3B |= (1<<WGM32) | (1<<CS31)
```

```
TIMSK3=(1<<OCIE3A)
```

```
OCR3A=32000*dt
```

Lo primero es determinar el modo de operación del Timer. El modo de operación del Timer se encuentra alojado en los bits -3 a 0 del WGM (Waveform Generation Mode). El modo seleccionado fue el CTC (Clear Timer on Compare Match) (ver pág. 148 anexo F tabla 17.2) el cual resetea o reinicia

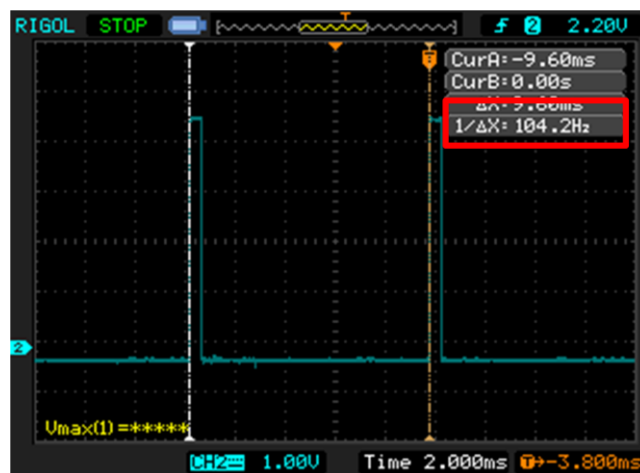
la cuenta del timer cada vez que coincide con el valor del comparador OCR3A (Output Compare Register del timer 3 bit A). Según dicha tabla el modo CTC es el cuarto modo de operación del Timer 3.

Los bits 3 a 0 del WGM se encuentran repartidos entre el registro TCCR3A y el TCCR3B (pag. 158 y 160 anexo F). Con el módulo de prescaler seleccionamos el reloj a usar en el timer, mediante pruebas realizadas en el osciloscopio, se encontró que el prescaler con el que se obtiene un mejor comportamiento es el del bit CS31 (Clock Select bit 1) el cual divide la frecuencia del chip a una octava parte es decir de 16 MHz a 2 MHz.

La máscara de interrupción activa la interrupción por comparación, que se encuentra alojado en el registro TIMSK3 (Timer Interrupt Mask Register). Activando el bit OCIE3A (Output Compare Interrupt Enable) el cual activa el vector de la interrupción.

Con esto se completa la configuración de la interrupción, la cual da como resultado que la ISR se ejecutará aproximadamente cada 104 Hz como se observa en la figura 59.

Figura 59. Frecuencia de la ISR



Fuente: Osciloscopio DICBOT

8.8 MODO SWING UP

El swing up o elevamiento por oscilación consiste en sacar al péndulo de su estado de reposo y llevarlo a uno que corresponda a la posición invertida.

Estudios como [9] concluyen que la mejor forma de elevar un péndulo a su posición invertida, es controlando su energía. Debido a la complejidad de la estrategia de control por energía que implica control robusto, las cuales están por fuera de los alcances y objetivos de este proyecto, se opta por realizar la elevación del péndulo mediante la oscilación del mismo por un pequeño control proporcional.

Inicialmente el péndulo se perturba para sacarlo de su estado de reposo, si este pequeño impulso lleva al péndulo a una posición no mayor a los -35 grados, entonces el péndulo se coloca en oscilación hasta que alcanza una posición mayor a los 160 grados, posición en la cual entrará al modo de estabilización donde la estrategia utilizada fue un control PID.

8.9 MODO DE ESTABILIZACIÓN

8.9.1 Controlador PID

Aquí se resume el algoritmo de control PID que se encarga de posicionar el brazo y el péndulo. Primero se calcula el valor en ángulo del brazo (α) y el péndulo (θ) de la siguiente forma:

```
alpha=count_1*(360.0/(1600.0));
```

```
theta=count_2*(360.0/10000.0);
```

Se toma la variable donde se almacenan los pulsos del encoder y se multiplica por el factor de conversión, el cual es simplemente la relación de los pulsos por una revolución del encoder. En el caso del brazo, el disco ranurado del encoder posee 400 ranuras por revolución pero mediante software se le aplica una cuadratura de 4x para un total de 1600 pulsos por revolución.

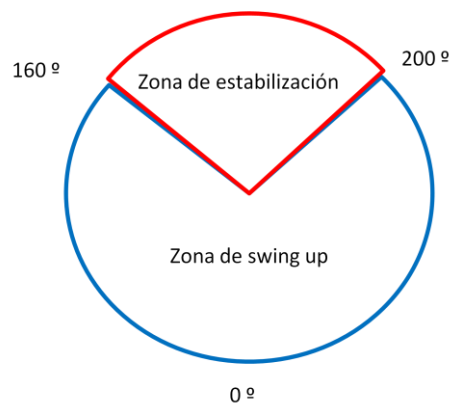
Para el péndulo, el encoder posee una resolución de 2500 pulsos por revolución pero mediante software se le aplica una cuadratura de 4x para un total de 10000 pulsos por revolución.

Una vez calculado el ángulo se procede al cálculo del error que viene siendo la diferencia entre el setpoint y el dato del ángulo enviado por el encoder.

El control tiene dos modos de actuación, uno para el swing up y el otro para la estabilización del péndulo. El modo swing up a su vez se compone en dos, lo mismo sucede para la zona de estabilización

Se definieron entonces dos zonas de trabajo, como se ve en la figura 60.

Figura 60. Zonas de trabajo.



Fuente: Autor

La función loop del arduino se encargará de enviar solamente el dato de los ángulos por el serial hacia simulink para ser graficados en tiempo real, además también se encarga del muestreo en la pantalla LCD como se ve en la figura 64.

Para la descripción que sigue a continuación, observar figura 62, donde se aprecia el contenido de la ISR.

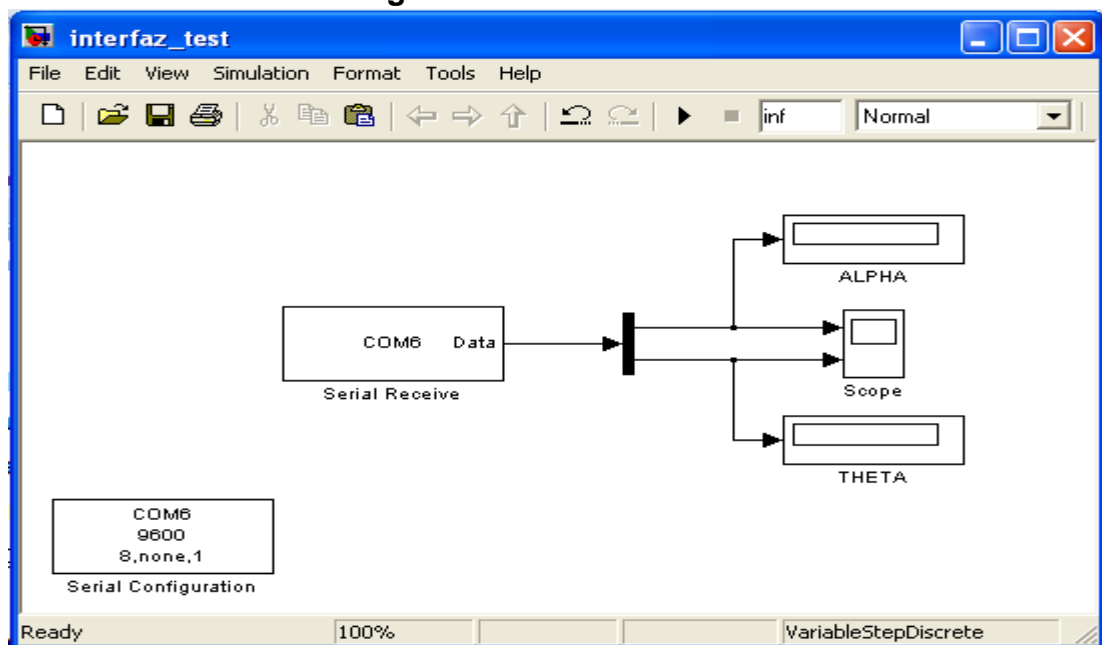
El péndulo marca cero grados en la posición vertical hacia abajo y 180 grados en la posición vertical hacia arriba. Primero se envía una pequeña perturbación en sentido horario al péndulo de manera que no sobrepase los 35 grados, luego, luego se hace oscilar el péndulo hasta que alcance más de 160 grados,

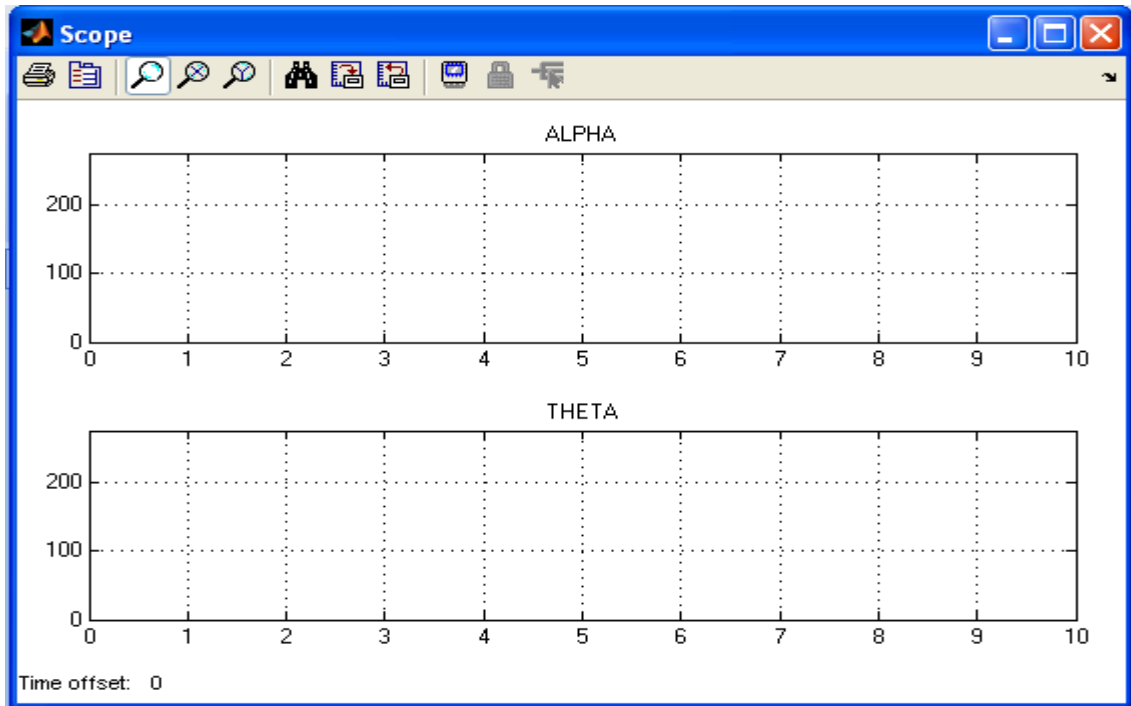
una vez allí, entra un lazo de control PID sobre el péndulo que se encargará de estabilizar la velocidad del péndulo cerca a cero, una vez que esto ocurre y que el brazo está en un rango entre los -10 y 10 grados, se activa un doble lazo de control, ver figura 63, uno para el brazo cuya salida se convierte en la entrada del control sobre el péndulo. Para una descripción más detallada revisar el código en el anexo G.

8.9.2 Interfaz con Simulink

En esta etapa, la información de la posición del brazo y el péndulo es enviada por el puerto serial del arduino en formato de número entero, una vez los recibe simulink, desglosa la información del serial y la muestra en el scope de simulink, una señal corresponde a la posición del brazo (alpha) y la otra señal corresponde a la posición del péndulo.

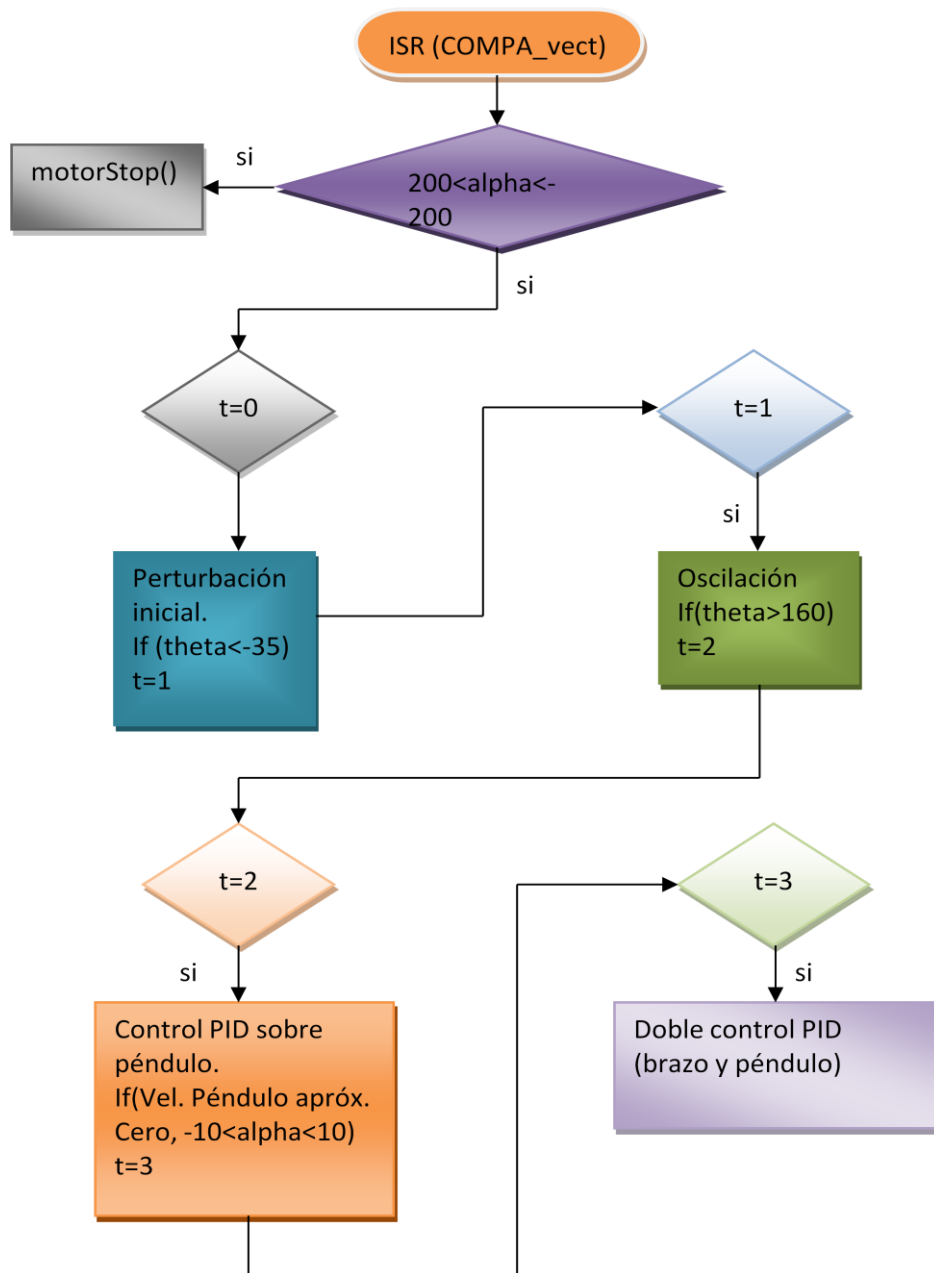
Figura 61. Interfaz SIMULINK





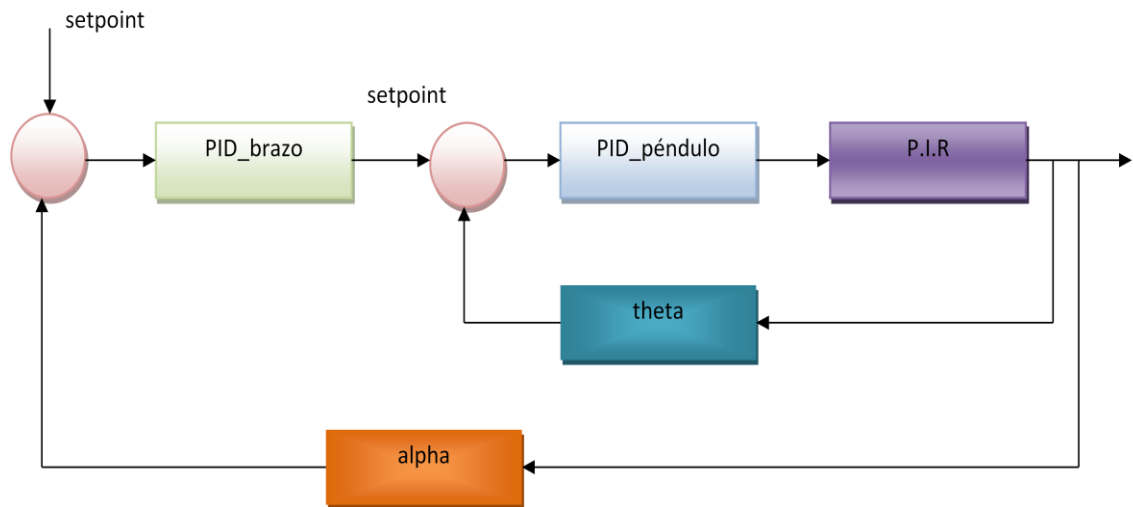
Fuente: Matlab-Simulink R2011b

Figura 62. Contenido de la ISR



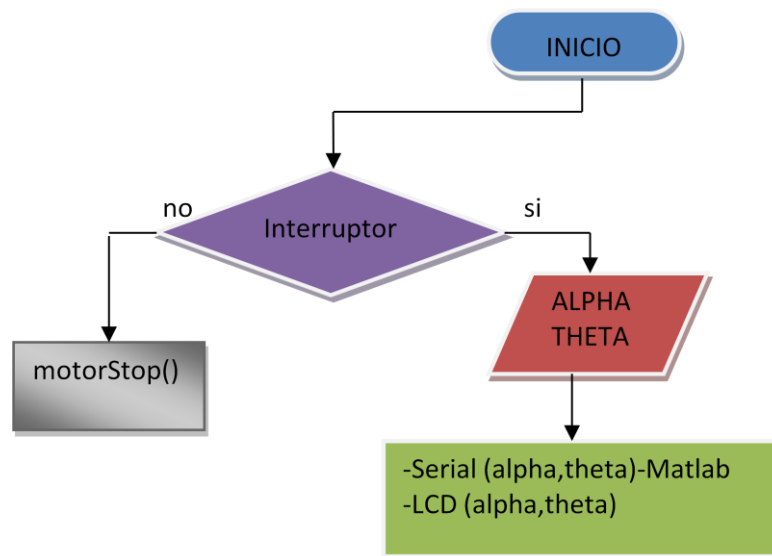
Fuente: Autor

Figura 63. Doble lazo de control



Fuente: Autor

Figura 64. Tareas en el loop del arduino

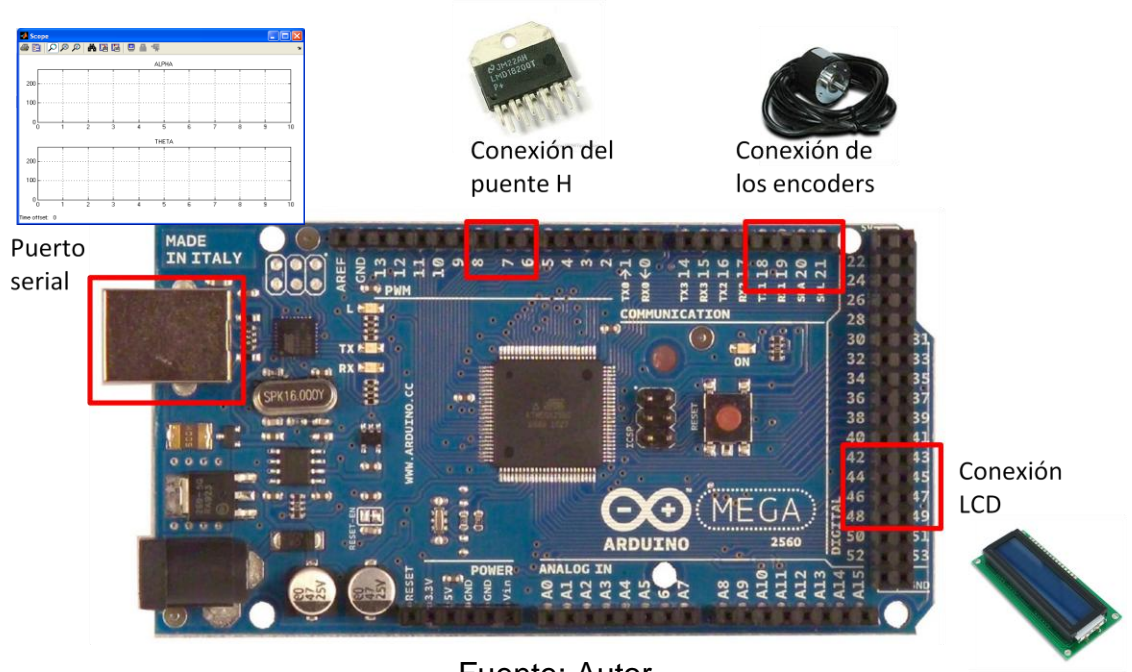


Fuente: Autor

Distribución de hardware

En la figura 65 se aprecia la conexión de los componentes a la tarjeta Arduino mega 2560.

Figura 65. Conexión a la tarjeta.

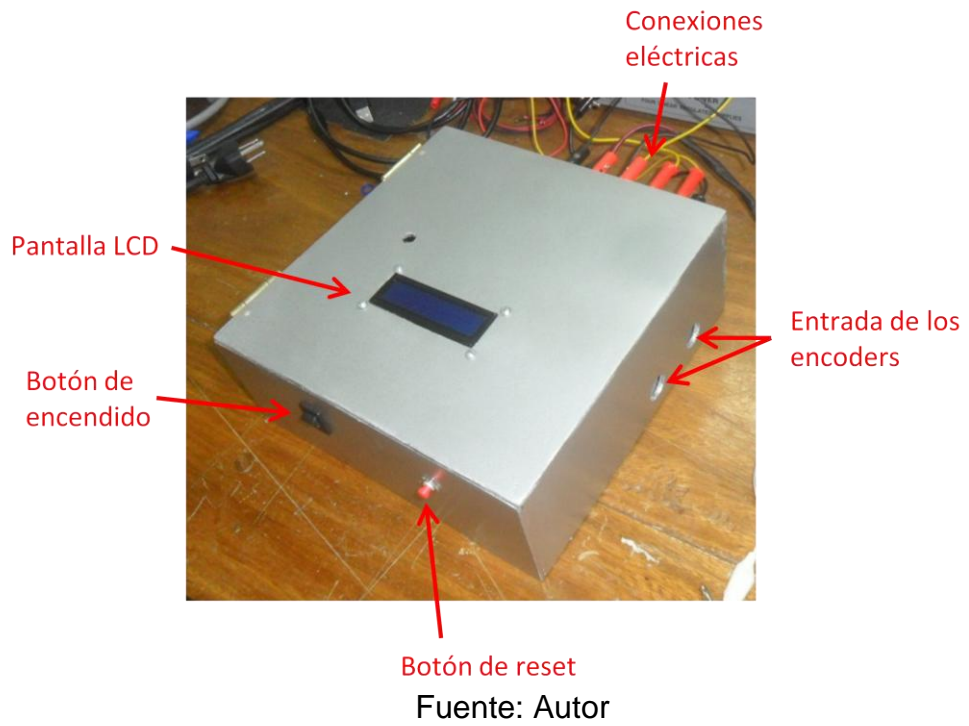


8.10 PANEL DE CONTROL

La caja de control permite el montaje de los elementos eléctricos y electrónicos necesarios para el control del banco experimental. En la parte superior se encuentra la pantalla LCD que permite visualizar el mensaje de bienvenida y el valor de los dos ángulos. En la parte frontal se encuentra el interruptor de encendido y seguridad, así como el botón reset. En la parte posterior se encuentran las conexiones eléctricas que corresponden a:

- Alimentación de puente H.
- Alimentación de encoders.
- Alimentación optoaisladores.
- Conexión del motor y ventilador (cooler).

Figura 66. Caja de control



9. PRUEBAS DEL CONTROLADOR PID

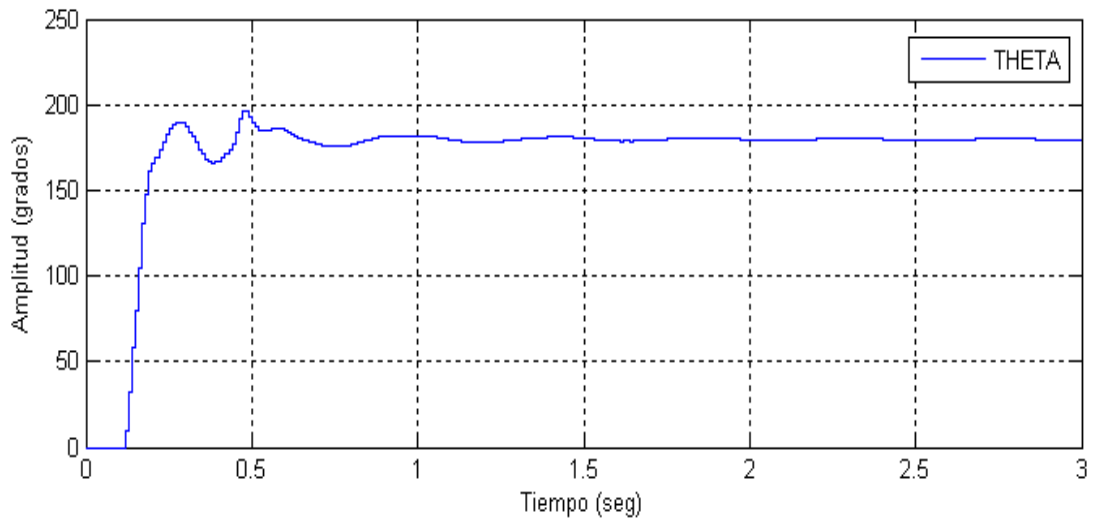
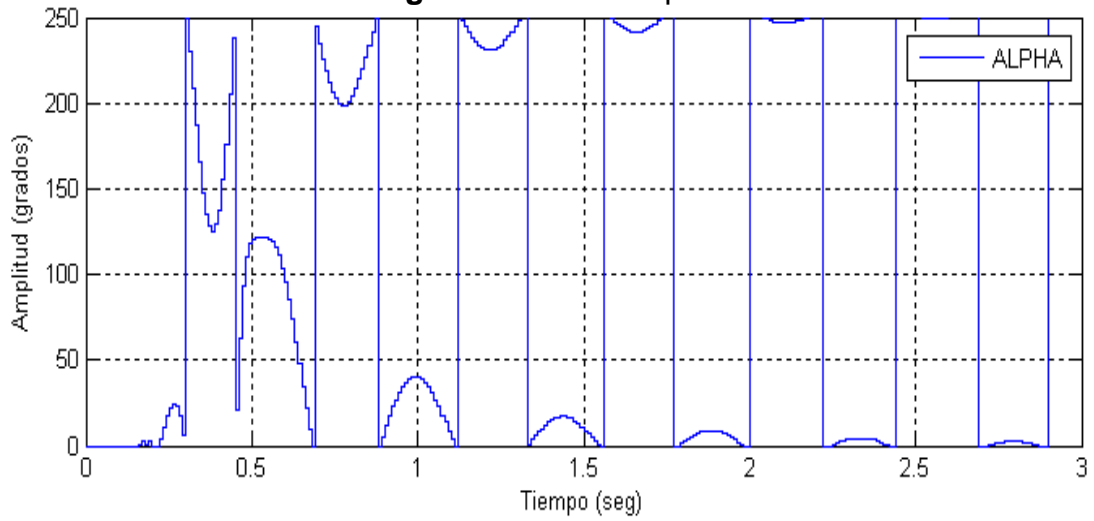
En este capítulo se expone el comportamiento del controlador PID implementado en el péndulo invertido rotativo. Con el fin de evidenciar el efecto del controlador, se realizaron varias pruebas que consistieron en la evaluación de la respuesta en el tiempo del controlador mediante la variación de las respectivas ganancias del controlador k_p , k_i y k_d , y respuesta a perturbaciones. Las pruebas se realizaron para una duración de 30 seg el eje del tiempo posee una escala de (1/10 seg). Se miden los parámetros que caracterizan la respuesta de un sistema de segundo orden tales como:

- Tiempo de retardo (t_d).
- Tiempo de elevamiento (t_r).
- Tiempo pico (t_p).
- Sobrepasso máximo (M_p).
- Tiempo de asentamiento (t_s).

Control PID

PRUEBA No. 1		RESPUESTA EN EL TIEMPO (seg)				
$K_p_1=5$	$K_p_2=20$	t_d	t_r	t_p	M_p	t_s
$K_i_1=0$	$K_i_2=2$	0.4	0.7	1.5	20%	3.1
$K_d_1=0.8$	$K_d_2=2$					
Esta prueba consiste en la variación de la ganancia proporcional del péndulo (k_p_2)						

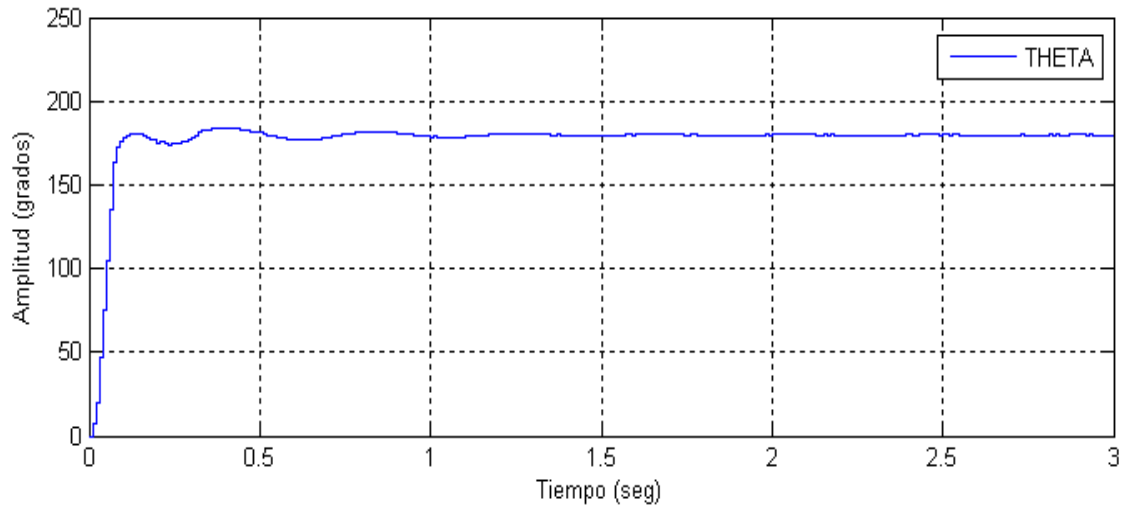
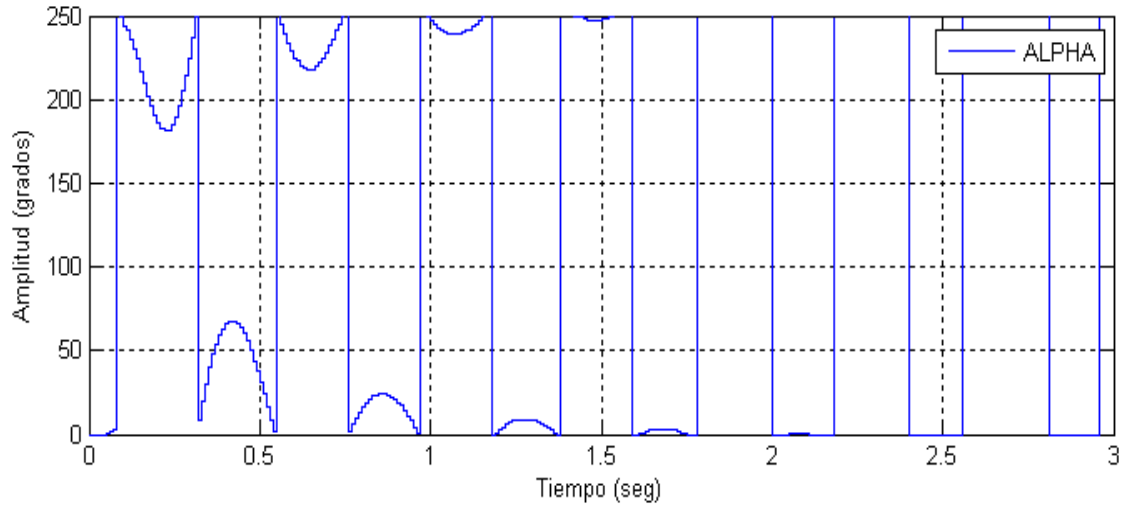
Figura 67. Gráficas prueba No.1



Fuente: Simulink R2011b

PRUEBA No. 2		RESPUESTA EN EL TIEMPO (seg)				
$K_p_1=5$	$K_p_2=50$	t_d	t_r	t_p	M_p	t_s
$K_i_1=0$	$K_i_2=2$	0.37	0.5	1.15	17.6%	2.8
$K_d_1=0.8$	$K_d_2=2$					
Esta prueba consiste en la variación de la ganancia proporcional del péndulo (k_p_2)						

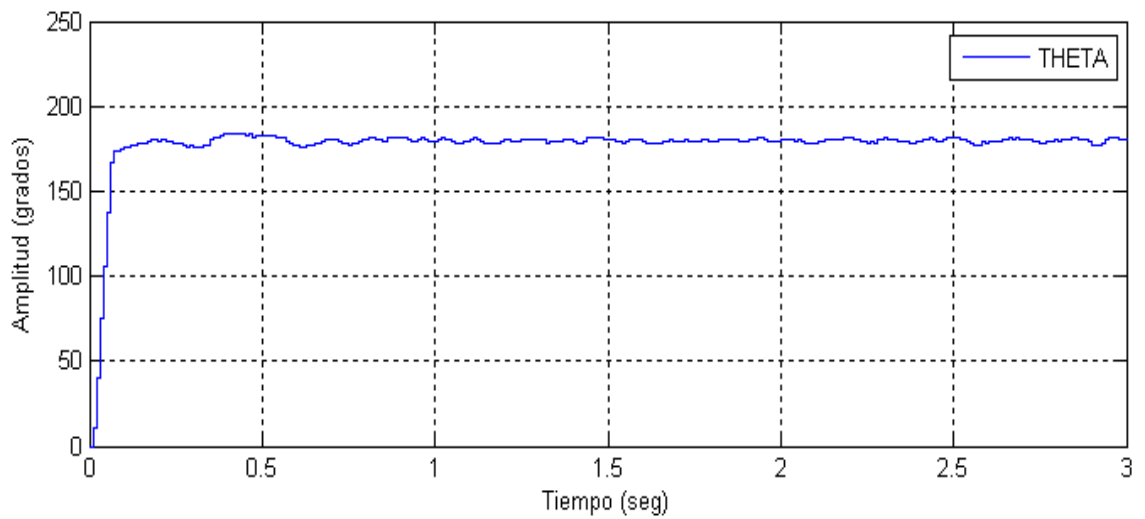
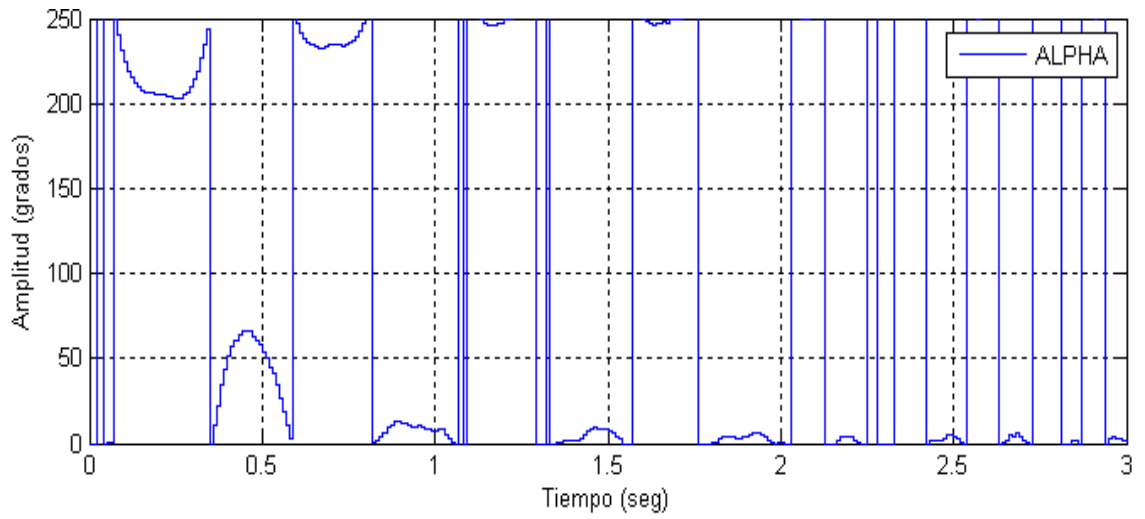
Figura 68. Prueba No. 2



Fuente: Simulink R2011b

PRUEBA No. 3		RESPUESTA EN EL TIEMPO (seg)				
Kp_1=5	Kp_2=40	td	tr	tp	Mp	ts
Ki_1=0	Ki_2=2	0.37	0.46	0.9	17%	2.75
Kd_1=0.8	Kd_2=5					
Esta prueba consiste en la variación de la ganancia derivativa del péndulo (kd_2)						

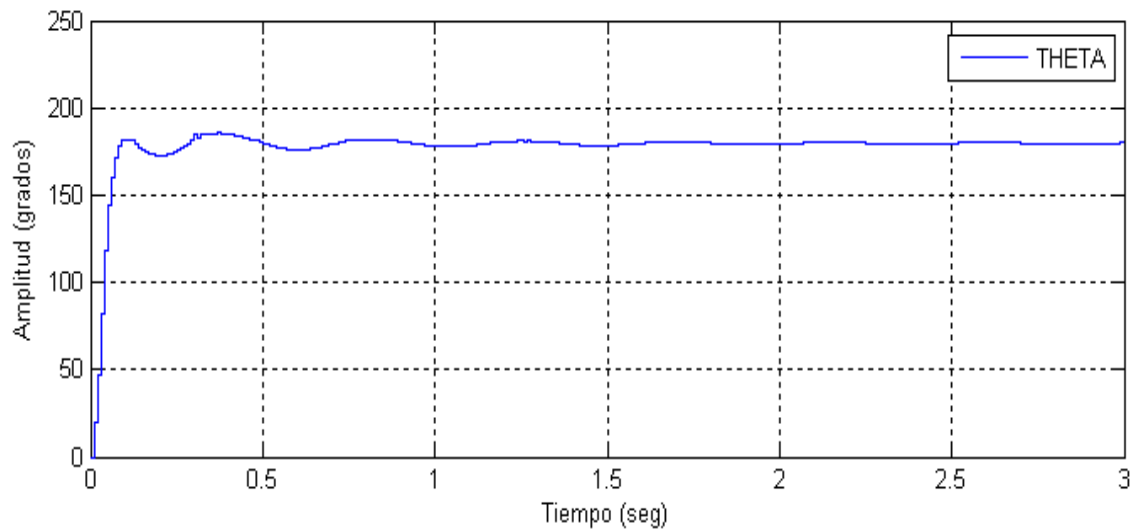
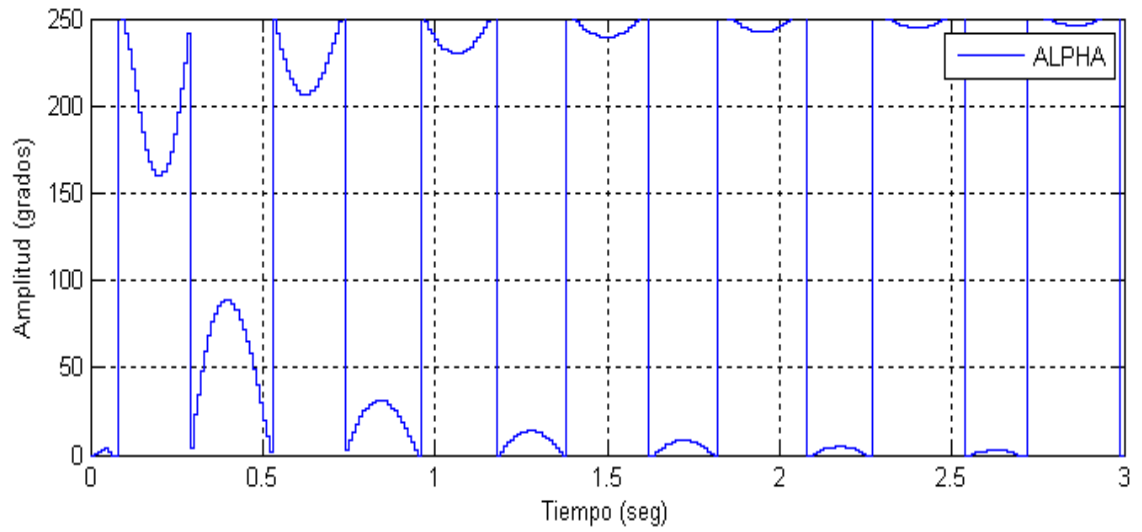
Figura 69. Prueba No. 3



Fuente: Simulink R2011b

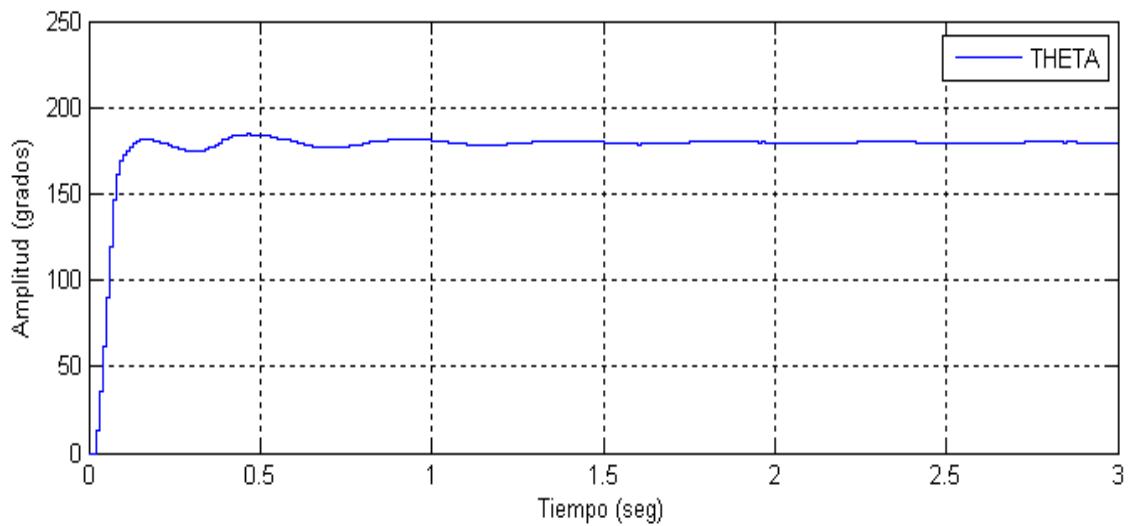
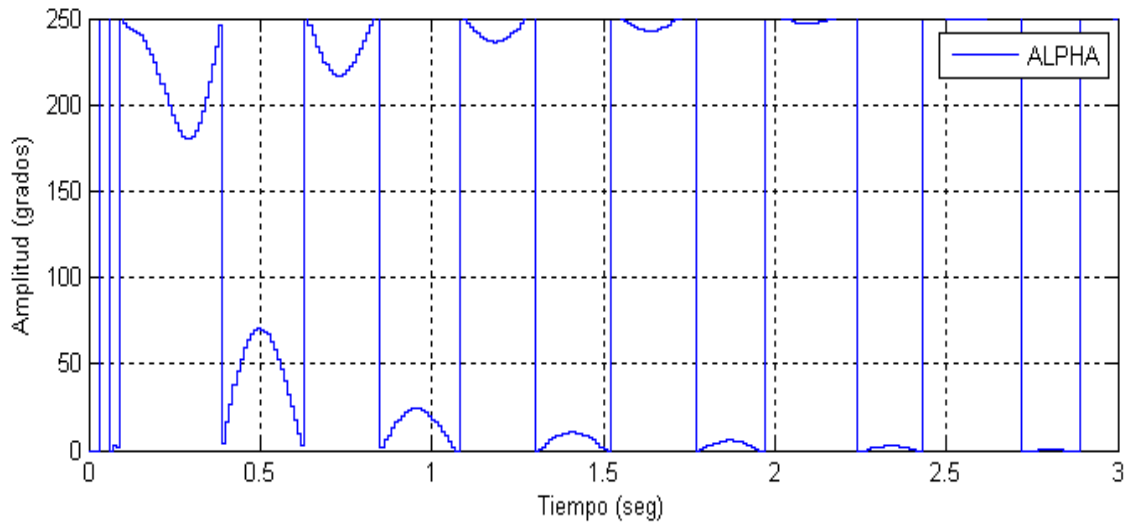
PRUEBA No. 4		RESPUESTA EN EL TIEMPO (seg)				
$Kp_1=5$	$Kp_2=40$	t_d	t_r	t_p	M_p	t_s
$Ki_1=0$	$Ki_2=2$	0.3	0.4	0.7	15%	2.5
$Kd_1=0.8$	$Kd_2=1$					
Esta prueba consiste en la variación de la ganancia derivativa del péndulo (k_d_2)						

Figura 70. Prueba No. 4



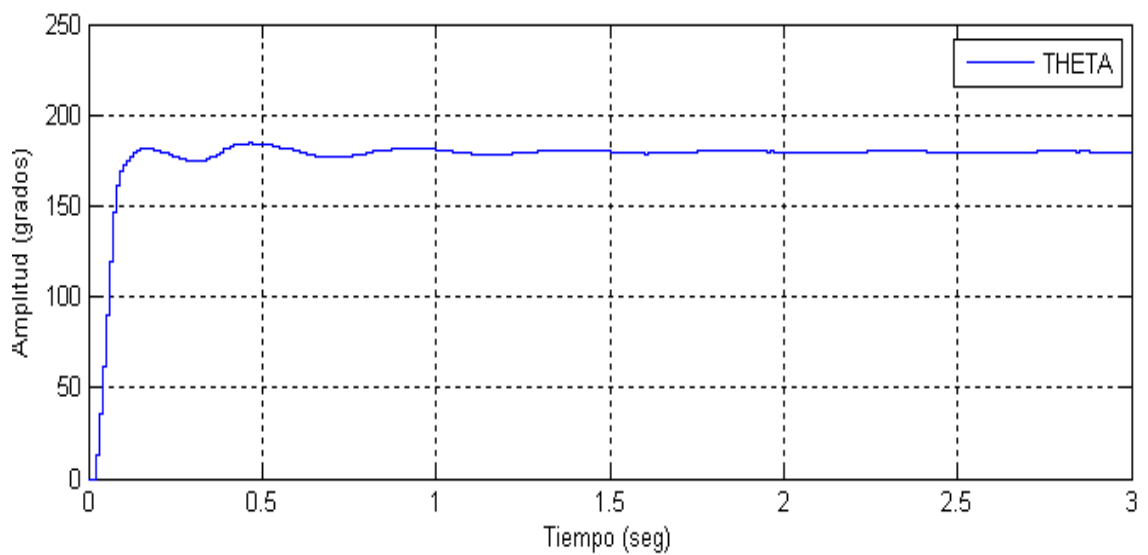
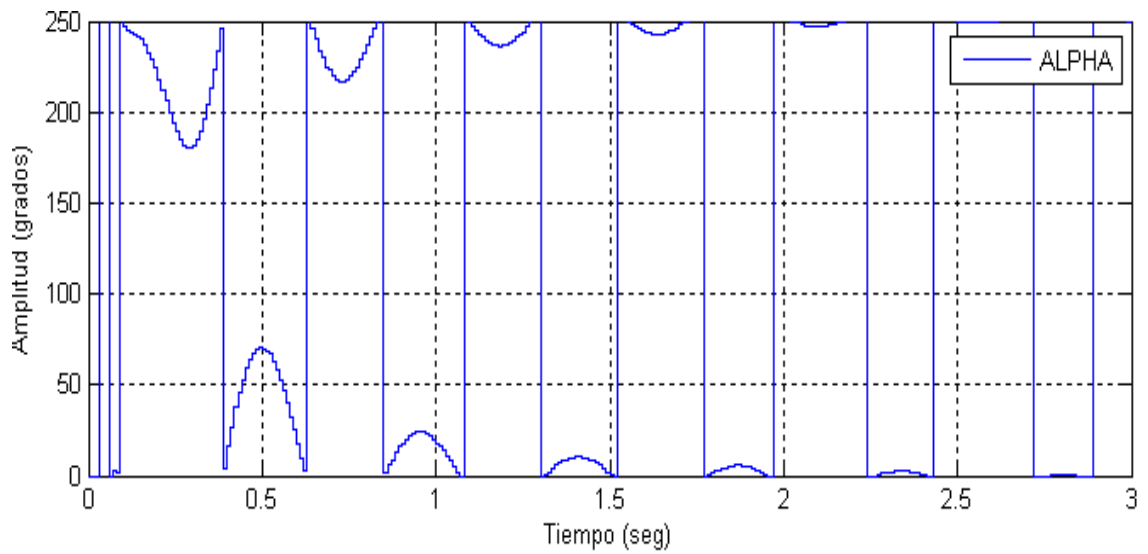
PRUEBA No. 5		RESPUESTA EN EL TIEMPO (seg)				
Kp_1=5	Kp_2=40	td	tr	tp	Mp	ts
Ki_1=0	Ki_2=1	0.27	0.3	0.34	19%	2.8
Kd_1=0.8	Kd_2=2					
Esta prueba consiste en la variación de la ganancia integral del péndulo (ki_2)						

Figura 71. Prueba No. 5



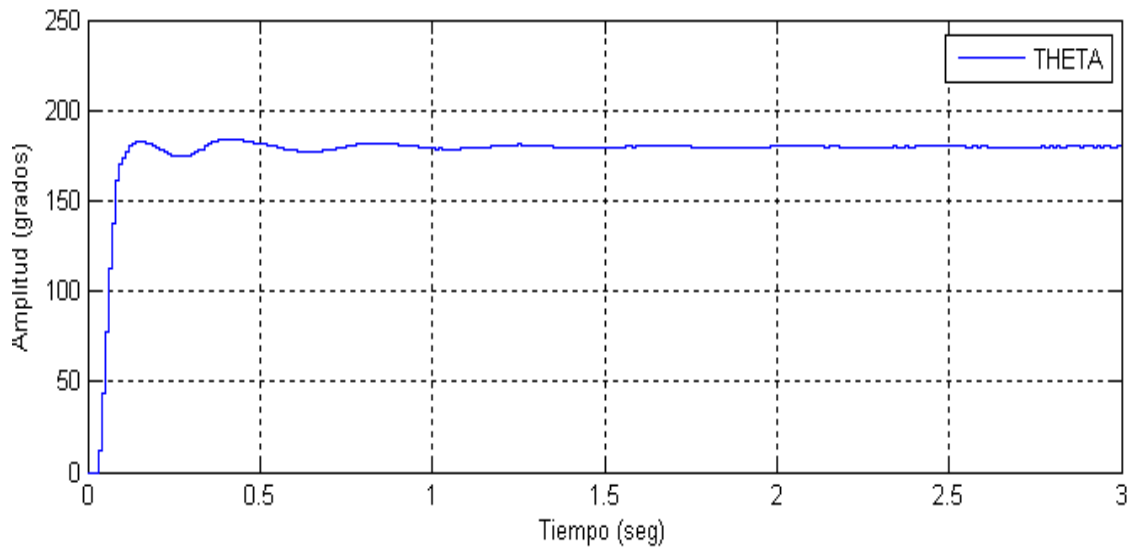
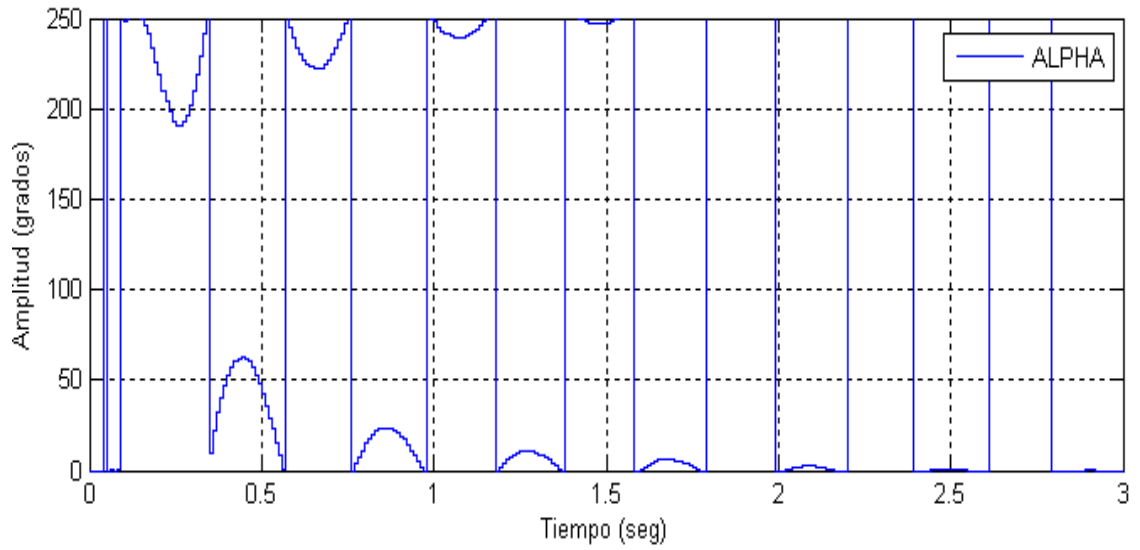
PRUEBA No. 6		RESPUESTA EN EL TIEMPO (seg)				
$Kp_1=5$	$Kp_2=40$	t_d	t_r	t_p	M_p	t_s
$Ki_1=0$	$Ki_2=5$	0.25	0.28	0.31	11%	2.3
$Kd_1=0.8$	$Kd_2=2$					
Esta prueba consiste en la variación de la ganancia integral del péndulo (k_{p_2})						

Figura 72. Prueba No. 6



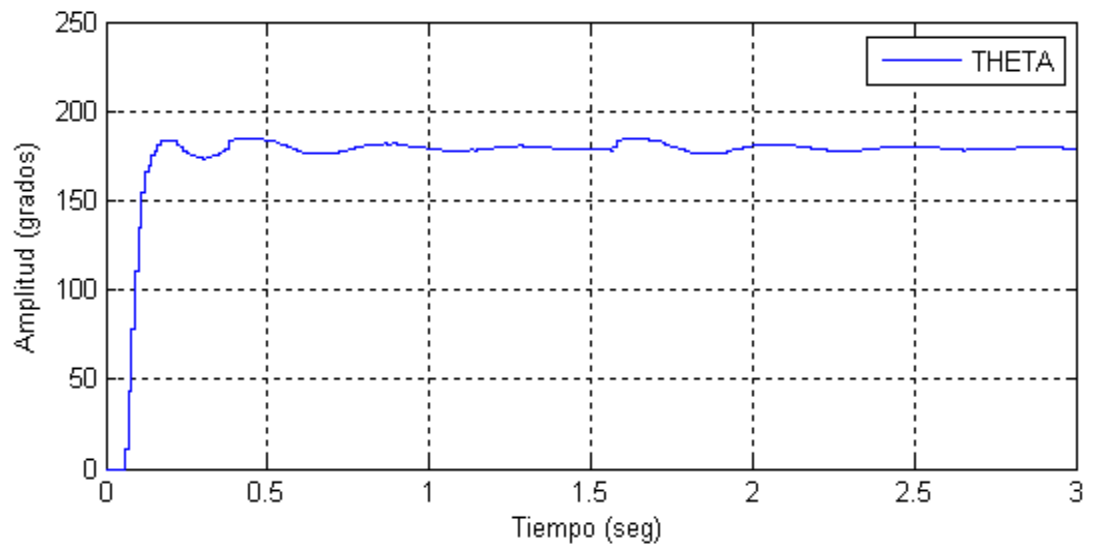
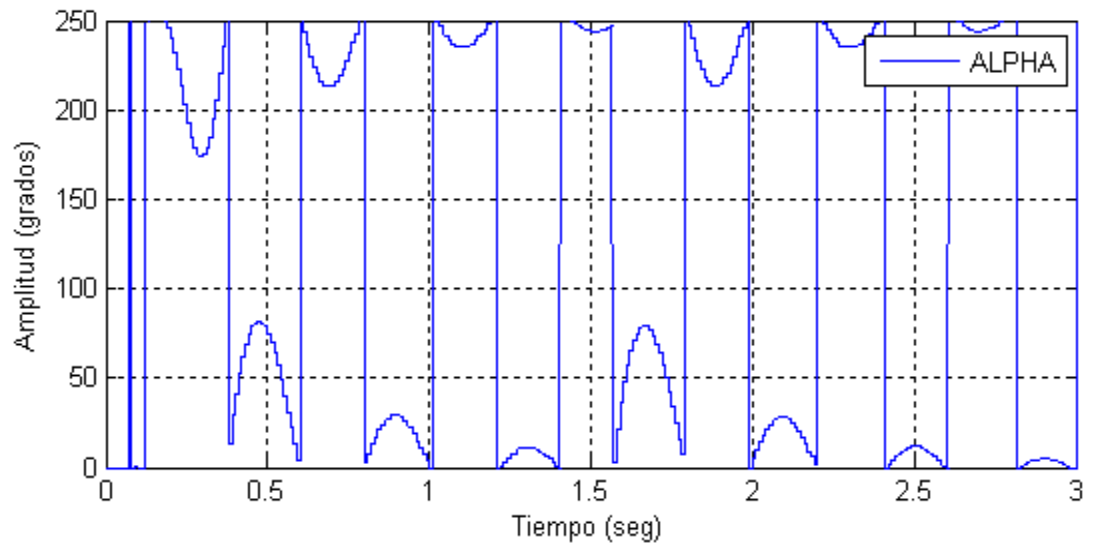
PRUEBA No. 7		RESPUESTA EN EL TIEMPO (seg)				
$K_p_1=5$	$K_p_2=40$	t_d	t_r	t_p	M_p	t_s
$K_i_1=0$	$K_i_2=2$	0.2	0.23	0.25	7.5%	2.2
$K_d_1=0.8$	$K_d_2=2$					
Esta prueba consiste en la variación de las ganancias del PID del péndulo						

Figura 73. Prueba No. 7



PRUEBA No. 8		RESPUESTA EN EL TIEMPO (seg)				
$Kp_1=5$	$Kp_2=40$	t_d	t_r	t_p	M_p	t_s
$Ki_1=0$	$Ki_2=2$	0.22	0.25	0.28	8%	1.5
$Kd_1=0.8$	$Kd_2=1$					
Esta prueba consiste en inducir perturbación al sistema						

Figura 74. Prueba No. 8



Fuente: Matlab-Simulink R2011b

10. MANUAL DE OPERACIÓN

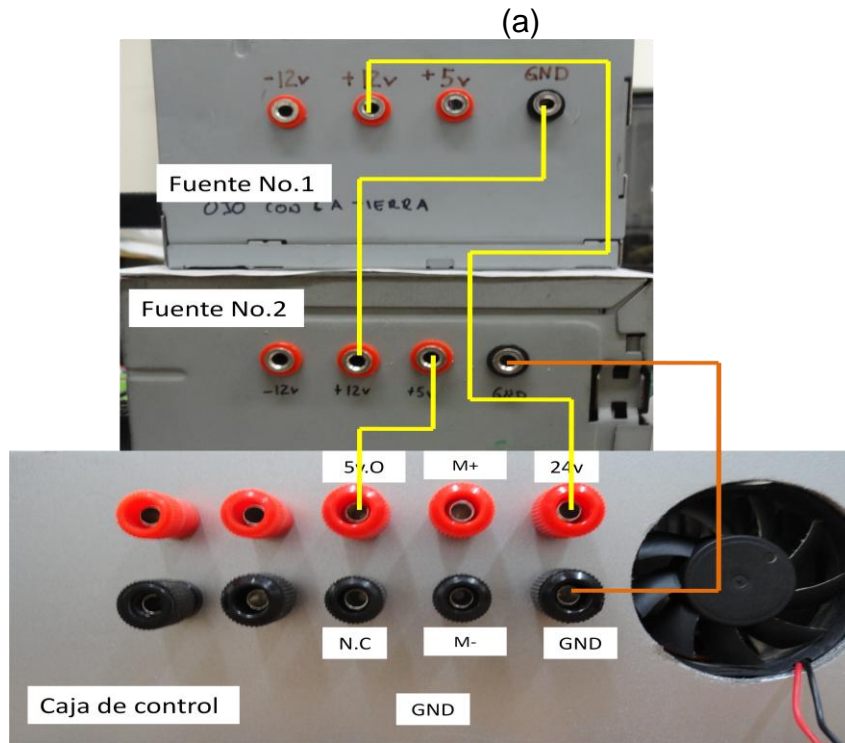
En este capítulo se describe la forma correcta de operar la planta dinámica del péndulo invertido rotativo, que consiste en como conectar y manipular eléctricamente los dispositivos que se encuentran alojados en la caja de control, y como cargar el código y la interfaz de simulink.

Conexión eléctrica

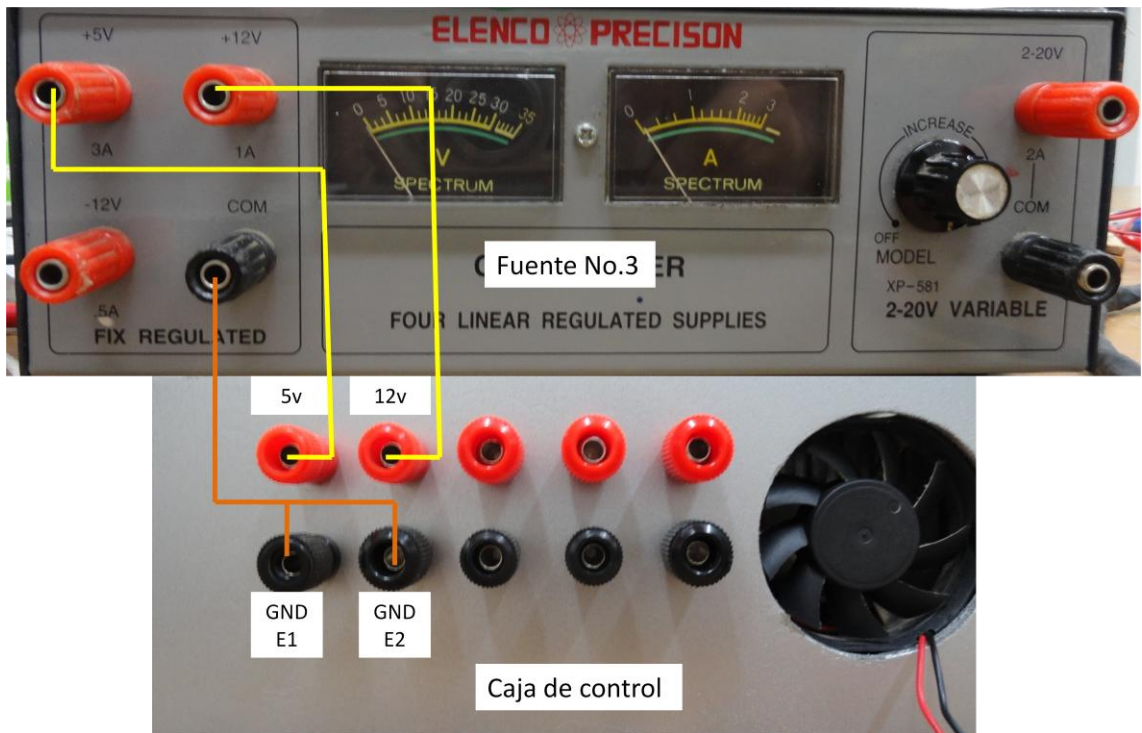
El motor y el puente H (LMD18200) se alimentan a 24 volts que provienen de dos fuentes de computador conectadas en serie. Los encoders se alimentan de una fuente regulable.

1. Verifique que las fuentes se encuentran apagadas y desconectadas.
2. Conecte la salida de 12v de la fuente No. 1 a la entrada de 24v de la caja de control. Luego conecte el GND de la fuente No.1 a la entrada de 12v de la fuente No. 2. Conecte el GND de la fuente No. 2 al GND de la caja de control (ver figura 75a).
3. Conecte la salida de 12v de la fuente No. 3 a la entrada de 12v de la caja de control. Conecte la salida de 5v de la fuente No. 3 a la entrada de 5v de la caja de control. Conecte el COM de la fuente No. 3 al GND E1 y el GND E2 de la caja de control (ver figura 75b).
4. Finalmente conecte los terminales del motor a las entradas M+ (terminal positivo del motor) y M- (terminal negativo del motor) de la caja de control como se observa en la figura 75a.

Figura 75. Conexión eléctrica



(b)



Fuente: Laboratorio DICBOT

Conexión del Arduino

Conecte el cable USB a la tarjeta arduino mega en el puerto de conexión como se observa en la figura 76, y luego conecte al puerto USB del computador en el que se encuentre instalado el software de arduino.

Figura 76. Conexión del arduino



Cargar programas

Asegúrese de que la tarjeta arduino está conectada en el puerto 'COM' correcto. Abra el software arduino . 

Cargue el archivo denominado "Furuta_control" como se aprecia en las figuras 77 y 78.

Figura 77. Apertura del programa de control PID

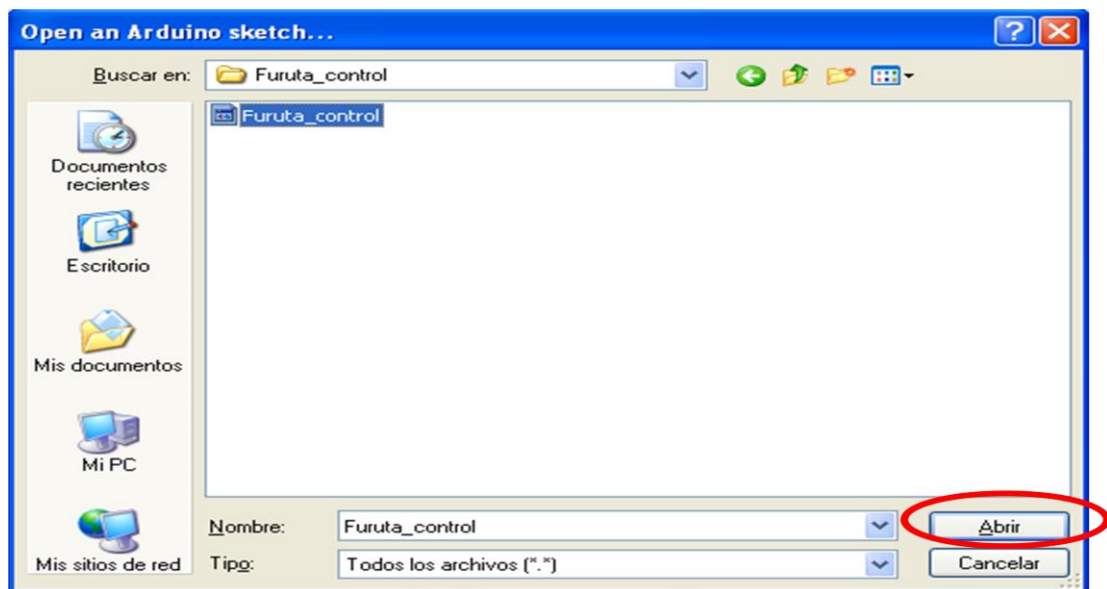
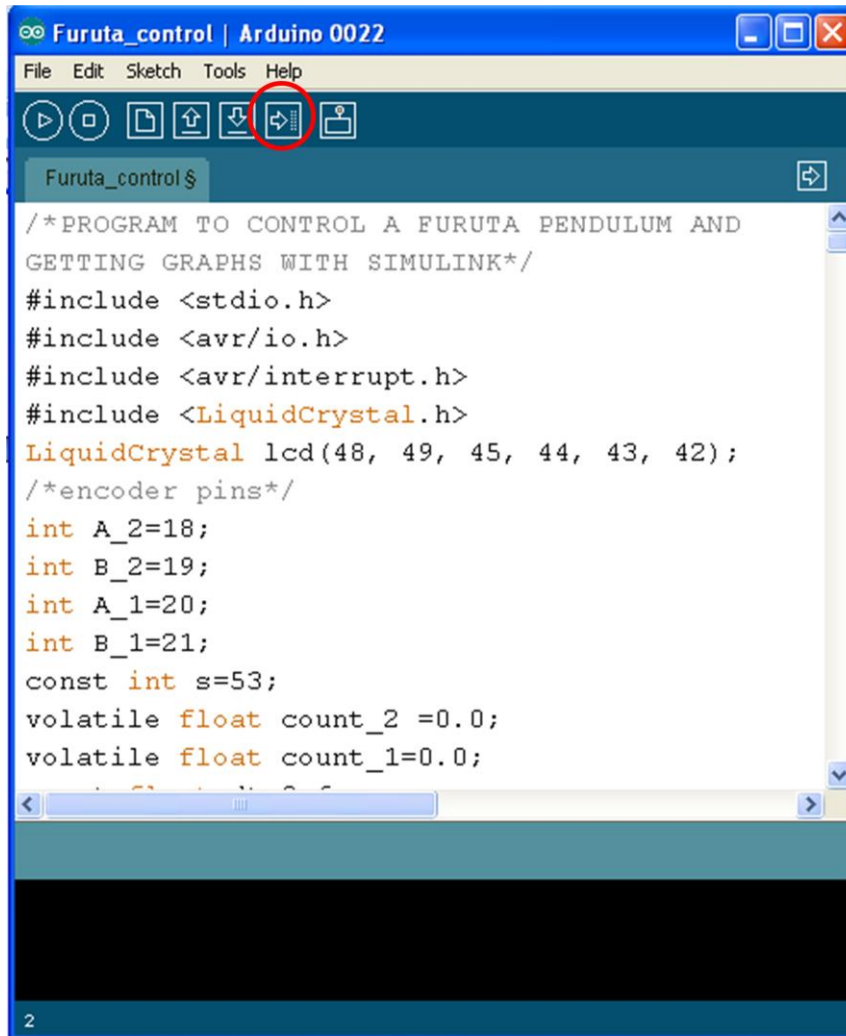


Figura 78. Cargado del programa a la tarjeta



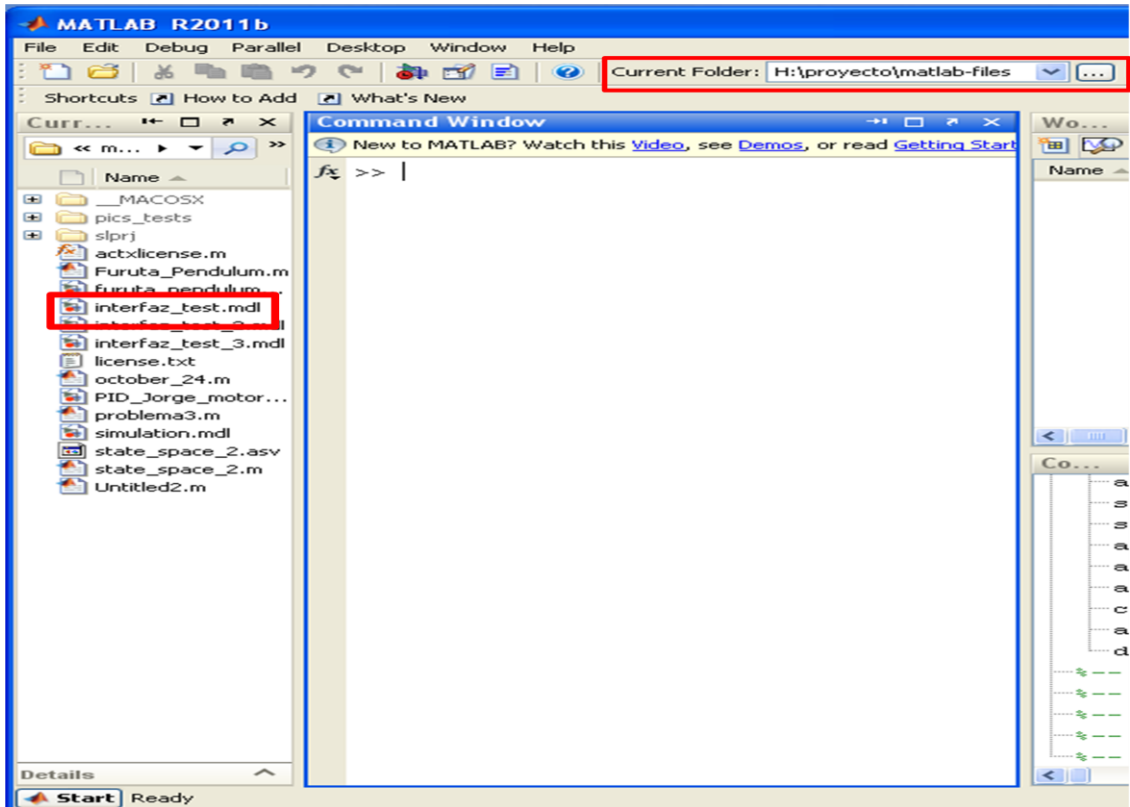
Apertura interfaz con SIMULINK

Abra el software MATLAB R2011b.



En el directorio actual de matlab abra el archivo de similink denominado "interfaz_test" como se muestra en la figura 79. En las propiedades de los bloques "Serial Configuration" y "Serial Receive" seleccione el puerto 'COM' donde esté conectado el arduino (ver figura 80). Abra el bloque "Scope" para visualizar la zona de gráficos y haga click en el icono comenzar simulación (ver figura 80). Finalmente encienda las fuentes de alimentación y el interruptor ubicado en el panel frontal de la caja de control (ver figura 81).

Figura 79. Apertura de la interfaz Arduino-Simulink



Fuente: MATLAB R2001b

Figura 80. Configuración de la interfaz

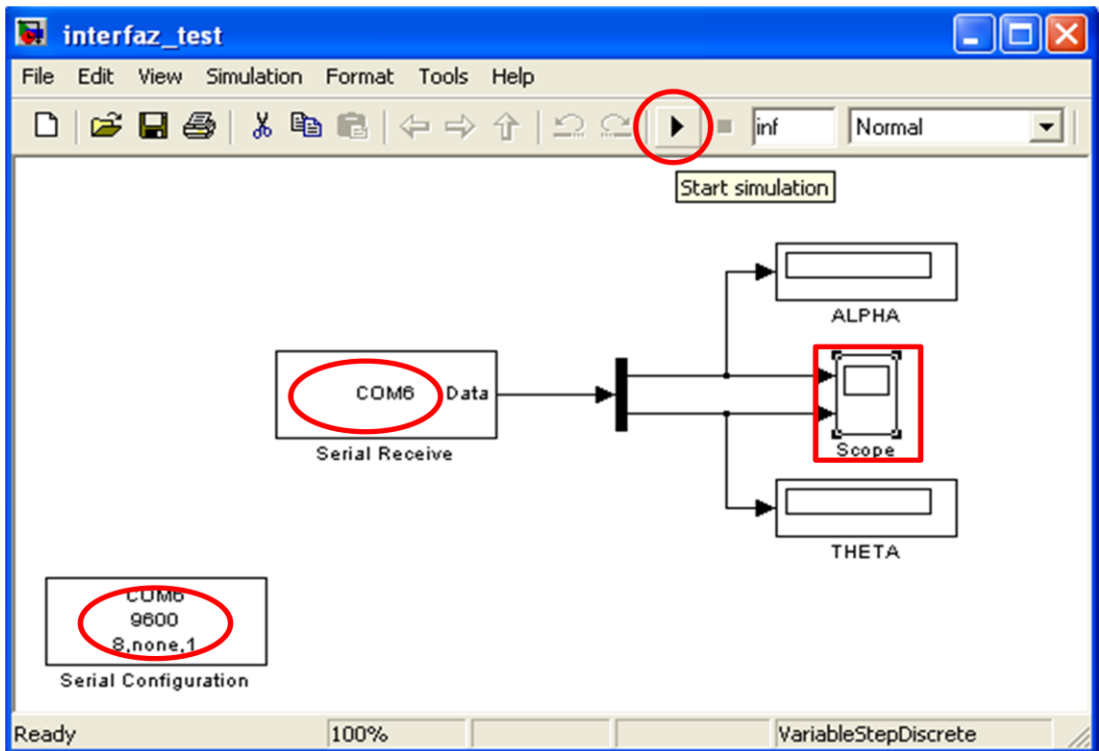
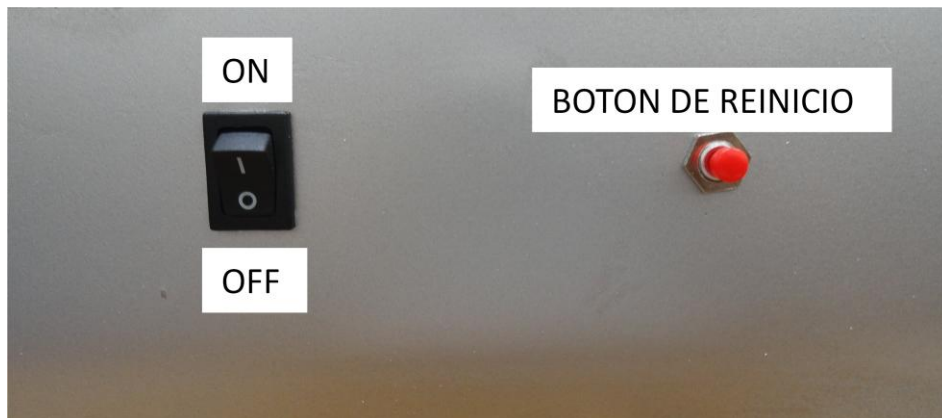


Figura 81. Encendido de la caja de control



Si se desea correr más de una prueba, se recomienda apagar el interruptor de la caja de control, llevar el péndulo a la posición no invertida, pulsar el botón de reinicio o reset, y encender de nuevo el interruptor.

11. CONCLUSIONES

- Se diseñó y construyó la estructura del péndulo invertido rotativo y una mesa para soportar la estructura principal junto con los componentes necesarios para su correcto funcionamiento.
- Se seleccionaron los componentes tales como sensores, circuitos integrados basándose en parámetros como requerimiento de voltaje, corriente, velocidad de respuesta, etc.
- Se desarrollaron dos métodos de lectura de encoder que requirieron tanto diseño de hardware como de software, se determinó que la forma más sencilla de leer los datos del encoder es la que corresponde a la expuesta en este proyecto.
- En el desarrollo de la etapa de implementación del controlador, se observó que el motor inducía ruido por los picos de corriente a través de la fuente de alimentación, dichos picos se alojaban en la tarjeta de arduino y causaban lecturas erróneas del encoder, mediante de pruebas realizadas en el osciloscopio se detectó la fuente del ruido, para solucionar este problema se implementó el uso de optoaisladores, los cuales permitieron aislar la tierra de puente H del arduino así como de la tarjeta de encoder. Se utilizaron tres optoaisladores, uno por cada pin de control del puente H.
- Para el pin de PWM se utilizó un optoaislador con un valor mayor de frecuencia de respuesta (6N137), ya que era imprescindible que la señal recibida por este opto fuera transmitida hacia al motor con el mínimo retardo posible, para los pines de DIR y BRAKE se incorporaron optos más comunes (4N25).
- Se desarrollaron algoritmos por separado tanto para la etapa de control de potencia como de adquisición datos, con el fin de comprobar la correcta funcionalidad de los dispositivos, y que sirvan de apoyo para la resolución de problemas.

- Se diseñaron varios controladores PID para el control del péndulo invertido rotativo, que se pueden agrupar en dos métodos: Usando la librería PID del arduino la cual presentaba conflictos con el uso de la función delay() y no permitía el correcto funcionamiento de la pantalla LCD y del algoritmo de control. El otro método fue construir el algoritmo PID mediante el uso de interrupciones, lo cual requirió la programación del chip del arduino el ATmega2560, y no presentó interferencia con la operación de la pantalla LCD.
- Se implementó una pantalla LCD con el fin de visualizar la posición de los parámetros a controlar (posición del brazo y posición del péndulo), comprobar la eficacia del control PID en mantener el sistema en los respectivos valores de referencia o setpoints, y proporcionar un ambiente más amigable y pedagógico en la realización del experimento.
- Se implementaron mecanismos de seguridad tanto por software y hardware que permiten evitar lesiones personales durante la operación de la planta. En el algoritmo de control se especificó un rango de trabajo y operación del motor, que permite frenar el motor en el caso que supere dicho límite de trabajo. La caja de control posee un interruptor el cual permite realizar el experimento solo si este se encuentra encendido. También posee un botón de reset que permite reiniciar la tarjeta desde una ubicación externa a la misma.
- Se realizó una interfaz con Simulink con el fin de visualizar los parámetros de control en tiempo real, ya que arduino no cuenta con un módulo de gráficos, esta implementación además aporta a un mejor entendimiento sobre la respuesta del sistema de control del péndulo invertido rotativo.

12. RECOMENDACIONES

- Leer el manual de operación antes de realizar el experimento.
- Implementar otras técnicas de control diferentes a las clásicas, de control robusto por ejemplo.
- Avanzar en la dificultad del proyecto, como realizar un doble péndulo invertido, para lo cual se necesita incluir un encoder y una barra más, sin más cambios a nivel de hardware, pero si realizar cambios a nivel del programa de control.
- Los estudiantes deben poseer una muy buena fundamentación en electrónica y programación para poder asimilar con facilidad el componente electrónico de esta clase de proyectos que se encuentran enfocados a la robótica y que integran las disciplinas de la Ingeniería Mecánica y la Ingeniería Electrónica.
- Hacer uso de otras tarjetas de desarrollo, basadas en otros microcontroladores como: National Instrument, microcontrolador PIC, MOTOROLA 68HC12 u otros.

BIBLIOGRAFIA

- [1] VALERA A, Desarrollo y control de un péndulo de Furuta, Dpto. de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia.
- [2] OSORIO ZUÑIGA Carlos Andrés, Diseño, Construcción y control de un péndulo invertido rotacional utilizando técnicas lineales y no lineales, Universidad Nacional de Colombia, 2009.
- [3] BOLTON W, Mecatrónica, Sistemas de Control Electrónico en Ingeniería Mecánica y Eléctrica, Segunda Edición, Alfaomega.
- [4] MALVINO Albert Paul, Principios de electrónica, Cuarta edición, MacGrawHill 1991.
- [5] MARRIOTT Jonathon, Construction and Control of a Furuta Pendulum, The University of Newcastle.
- [6] CRAIG K., AWTAR S. Inverted Pendulum Systems: Rotary and arm-driven a mechatronic system design case study, Department of Mechanical Engineering, Aeronautical Engineering and Mechanics Rensselaer Polytechnic Institute.
- [7]http://www.quanser.com/english/html/products/fs_product_challenge.asp?lang_code=english&pcat_code=exp-rot&prod_code=R8-seflerct&tmpl=1
- [8] CAIPA John, RODRÍGUEZ José, Modelado, Diseño y Control de Swing up y Balanceo del Péndulo Invertido Rotacional, Universidad Pedagógica y Tecnológica de Colombia.
- [9] ÅSTROM K, FURUTA K, Swinging a Pendulum by Energy Control.
- [10] http://www.quanser.com/english/html/home/fs_homepage.html.
- [11] www.todorobot.com.ar.

- [12] PEÑA Sebastián, CASTRO Diego, Diseño y Construcción de un Prototipo Experimental de un Vehículo Eléctrico Autobalanceado Unipersonal VEAU, Universidad Industrial de Santander, Mayo 2011.
- [13] OGATA Katsuhiko, Dinámica de Sistemas, Primera edición, Prentice Hall.
- [14] BEER Ferdinand, Vector Mechanics for Engineers, Novena edición, Mc Graw Hill.
- [15] OGATA Katsuhiko. Ingeniería de control moderna.
- [16] GALEANO, Gustavo. Programación de sistemas embebidos en C. Colombia. 2009
- [17] www.processing.org
- [18] www.atmel.com

ANEXOS

ANEXO A. Resumen hoja de datos encoder incremental

Light Duty Incremental Encoders

Features

A light-duty encoder is a cost-effective encoder for small applications and has the following features:

- Small body with 38 mm diameter and 30 mm depth
- Dust proof (IP40 rating)
- 6 mm standard shaft or 8 mm hollow shaft
- Resolution available from 100 pulses per revolution to 2500 pulses per revolution
- Open collector or line driver output
- Up to 200 kHz response frequency
- Two-meter cable, tinned ends



Standard shaft (TRD-S) model



Hollow shaft (TRD-SH) model

Note: Yellow shaded part numbers are non-stock. Availability may range from four to six weeks.

Light Duty Standard Shaft Incremental Encoders (NPN Open Collector and Line Driver models)						Light Duty Hollow Shaft Incremental Encoders (NPN Open Collector and Line Driver models)					
Part Number	Price	Pulses per Revolution	Input Voltage	Output	Body Diameter	Part Number	Price	Pulses per Revolution	Input Voltage	Output	Body Diameter
TRD-S100-BD	↔	100	12-24 VDC	NPN open collector	38mm	TRD-SH100-BD	↔	100	12-24 VDC	NPN open collector	38mm
TRD-S200-BD	↔	200				TRD-SH200-BD	↔	200			
TRD-S250-BD	↔	250				TRD-SH250-BD	↔	250			
TRD-S300-BD	↔	300				TRD-SH300-BD	↔	300			
TRD-S360-BD	↔	360				TRD-SH360-BD	↔	360			
TRD-S400-BD	↔	400				TRD-SH400-BD	↔	400			
TRD-S500-BD	↔	500				TRD-SH500-BD	↔	500			
TRD-S600-BD	↔	600				TRD-SH600-BD	↔	600			
TRD-S800-BD	↔	800				TRD-SH800-BD	↔	800			
TRD-S1000-BD	↔	1000				TRD-SH1000-BD	↔	1000			
TRD-S1024-BD	↔	1024				TRD-SH1024-BD	↔	1024			
TRD-S1200-BD	↔	1200				TRD-SH1200-BD	↔	1200			
TRD-S2000-BD	↔	2000				TRD-SH2000-BD	↔	2000			
TRD-S2500-BD	↔	2500				TRD-SH2500-BD	↔	2500			
TRD-S100-VD	↔	100				5VDC	Line driver (differential)	38mm			
TRD-S200-VD	↔	200	TRD-SH200-VD	↔	200						
TRD-S250-VD	↔	250	TRD-SH250-VD	↔	250						
TRD-S300-VD	↔	300	TRD-SH300-VD	↔	300						
TRD-S360-VD	↔	360	TRD-SH360-VD	↔	360						
TRD-S400-VD	↔	400	TRD-SH400-VD	↔	400						
TRD-S500-VD	↔	500	TRD-SH500-VD	↔	500						
TRD-S600-VD	↔	600	TRD-SH600-VD	↔	600						
TRD-S800-VD	↔	800	TRD-SH800-VD	↔	800						
TRD-S1000-VD	↔	1000	TRD-SH1000-VD	↔	1000						
TRD-S1024-VD	↔	1024	TRD-SH1024-VD	↔	1024						
TRD-S1200-VD	↔	1200	TRD-SH1200-VD	↔	1200						
TRD-S2000-VD	↔	2000	TRD-SH2000-VD	↔	2000						
TRD-S2500-VD	↔	2500	TRD-SH2500-VD	↔	2500						

- Company Information
- System Overview
- Programmable Controllers
- PLC I/O
- Software
- Control Hardware
- Drives
- Soft Starters
- Motor & Controls
- Clipped Circuits
- Motor Controls
- Proximity Sensors
- Photo Sensors
- Limit Switches
- Encoders**
- Current Sensors
- Pressure Sensors
- Temperature Sensors
- Pushbutton Lights
- Process
- Relay/Timers
- Comms
- Terminal Blocks & Wiring
- Power
- Circuit Protection
- Enclosures
- Tools
- Pneumatics
- Apparel
- Product Index
- Part # Index

Light Duty Incremental Encoders

Specifications

Electrical Specifications		
Model		TRD-S.###x-BD TRD-S.H.###x-BD (open collector)
		TRD-S.###x-VD TRD-S.H.###x-VD (line driver)
Power Supply	Operating Voltage	10.8 - 26.4VDC*
	Allowable Ripple	3% max.
	Current Consumption	50 mA max.
Signal Waveform		Two-phase + home position
Max. Response Frequency		2000/r
Duty Ratio		50 + 25%
Phase Difference Width		25 + 12.5%
Signal Width at Home Position		100 + 50%
Output	Rise/Fall Time	1µs max. (when cable length is 1m)
	Output Type	NPN open collector output, sinking
	Output Logic	Negative logic (active low)
	Output Current	H
	Output Voltage	L
	Influx Current	30mA max.
	Load Power Voltage	35 VDC max.
	Short-Circuit Protection	Between output and power supply
* To be supplied by Class II source		
Mechanical Specifications		
Starting Torque	Max. 0.001 Nm (0.0074 lb./in)	
Max. Allowable Shaft Load	Radial: 20N (4.5 lbs) Axial: 10N (2.25 lbs)	
Max. Allowable Speed	6000 rpm (highest speed that can support the mechanical integrity of encoder)	
Wire Size	AWG26	
Weight	Approx. 150g (5.3 oz) with 2m cable	
Environmental Specifications		
Ambient Temperature	10 to 70°C; 14 to 158°F	
Storage Temperature	-25 to 85°C; -13 to 185°F	
Operating Humidity	35-85% RH	
Voltage Withstand	500VAC (50/60Hz) for one minute	
Insulation Resistance	50MΩ min.	
Vibration Resistance	Durable for one hour along three axes at 10 to 55 Hz with 0.75 amplitude	
Shock Resistance	11 ms with 400 m/s ² applied three times along three axes	
Protection	IP40; dust proof	

Accessories

Couplings

If you selected an encoder with a solid shaft, please select a coupling that fits your encoder. All couplings are in stock, ready to ship.

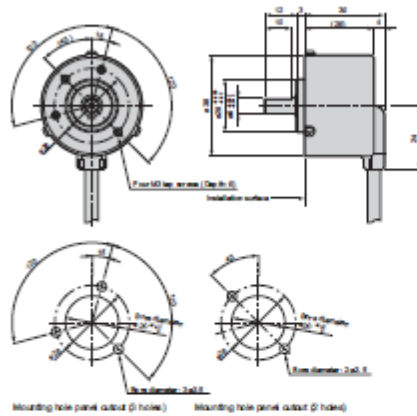
See page 21-16 for more information on couplings.

Mounting brackets are not available for light-duty encoders.

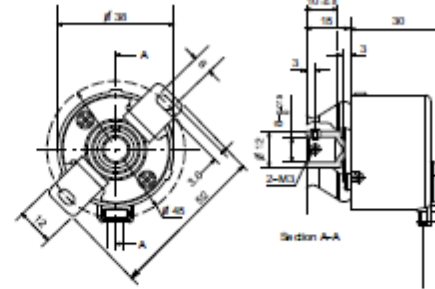
Light Duty Incremental Encoders

Dimensions

Standard shaft models



Hollow shaft models

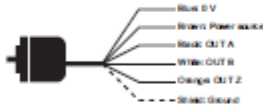


All dimensions in mm
Tolerance - 0.00075

Wiring diagrams

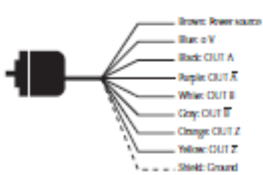
Open collector connections

Cable shield is not connected to the encoder body; enclosure is grounded through the DV wire.



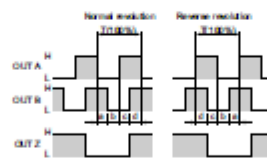
Line driver connections

Cable shield is not connected to the encoder body; enclosure is grounded through the DV wire.

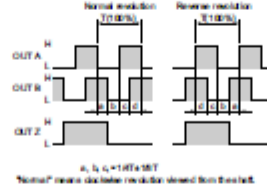


Channel timing charts

Open Collector Models



Line Driver Models



How to read the timing charts

Open Collector Models

Out A and Out B are 90 degrees out of phase. Like any quadrature encoder, four unique logic states are created internally to the encoder. This is based on the rising edge to rising edge (one cycle) on channel A or B that indicates one set of bars on the internal encoder disk has passed by the optical sensor. OUT Z is the absolute reference added to an incremental encoder and is also known as home position. It signifies a full rotation of the encoder disk.

Line Driver Models

Channel A (OUT A and A-not) and Channel B (OUT B and B-not) are also 90 degrees out of phase on line driver encoders. OUT Z is the same as an open collector models, and is the absolute reference (home position). It signifies one full rotation of the encoder.

ANEXO B. Resumen hoja de datos Puente H LMD18200



LMD18200

February 7, 2011

3A, 55V H-Bridge

General Description

The LMD18200 is a 3A H-Bridge designed for motion control applications. The device is built using a multi-technology process which combines bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure. Ideal for driving DC and stepper motors; the LMD18200 accommodates peak output currents up to 6A. An innovative circuit which facilitates low-loss sensing of the output current has been implemented.

Features

- Delivers up to 3A continuous output
- Operates at supply voltages up to 55V
- Low $R_{DS(ON)}$ typically 0.33 Ω per switch at 3A
- TTL and CMOS compatible inputs

- No "shoot-through" current
- Thermal warning flag output at 145°C
- Thermal shutdown (outputs off) at 170°C
- Internal clamp diodes
- Shorted load protection
- Internal charge pump with external bootstrap capability

Applications

- DC and stepper motor drives
- Position and velocity servomechanisms
- Factory automation robots
- Numerically controlled machinery
- Computer printers and plotters

Functional Diagram

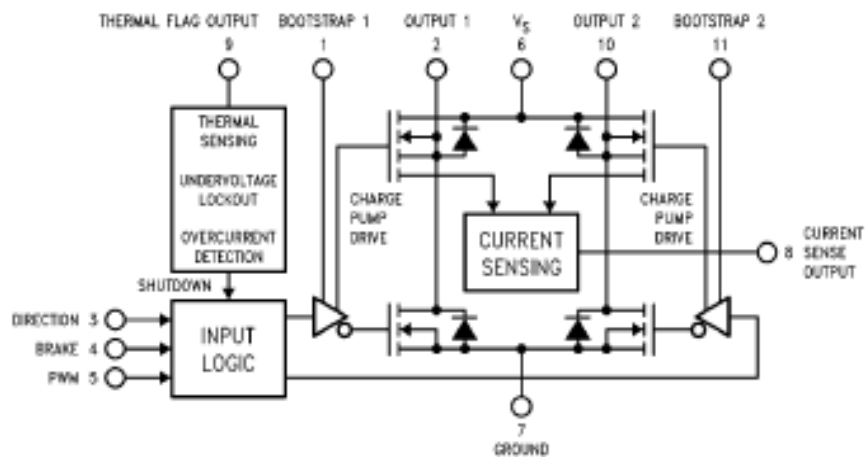
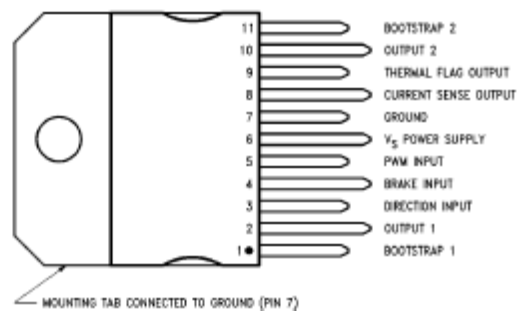


FIGURE 1. Functional Block Diagram of LMD18200
Connection Diagram and Ordering Information



11-Lead TO-220 Package
Top View
Order Number LMD18200T
See NS Package TA11B

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Total Supply Voltage (V_S , Pin 6)	60V
Voltage at Pins 3, 4, 5, 8 and 9	12V
Voltage at Bootstrap Pins (Pins 1 and 11)	$V_{OUT} + 16V$
Peak Output Current (200 ns)	6A
Continuous Output Current (Note 2)	3A

Power Dissipation (Note 3)	25W
Power Dissipation ($T_A = 25^\circ\text{C}$, Free Air)	3W
Junction Temperature, $T_{J(\text{max})}$	150°C
ESD Susceptibility (Note 4)	1500V
Storage Temperature, T_{STG}	-40°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Operating Ratings (Note 1)

Junction Temperature, T_J	-40°C to +125°C
V_S Supply Voltage	+12V to +55V

Electrical Characteristics (Note 5)

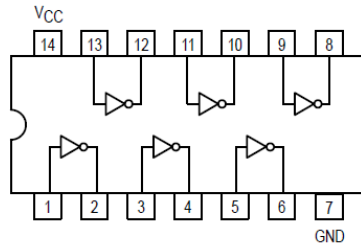
The following specifications apply for $V_S = 42V$, unless otherwise specified. **Boldface** limits apply over the entire operating temperature range, $-40^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$, all other limits are for $T_A = T_J = 25^\circ\text{C}$.

Symbol	Parameter	Conditions	Typ	Limit	Units
$R_{DS(ON)}$	Switch ON Resistance	Output Current = 3A (Note 6)	0.33	0.40/0.6	Ω (max)
$R_{DS(ON)}$	Switch ON Resistance	Output Current = 6A (Note 6)	0.38	0.45/0.6	Ω (max)
V_{CLAMP}	Clamp Diode Forward Drop	Clamp Current = 3A (Note 6)	1.2	1.5	V (max)
V_L	Logic Low Input Voltage	Pins 3, 4, 5		-0.1	V (min)
				0.8	V (max)
I_L	Logic Low Input Current	$V_{IN} = -0.1V$, Pins = 3, 4, 5		-10	μA (max)
V_{HI}	Logic High Input Voltage	Pins 3, 4, 5		2	V (min)
				12	V (max)
I_{HI}	Logic High Input Current	$V_{IN} = 12V$, Pins = 3, 4, 5		10	μA (max)
	Current Sense Output	$I_{OUT} = 1A$ (Note 8)	377	325/300 425/450	μA (min) μA (max)
	Current Sense Linearity	$1A \leq I_{OUT} \leq 3A$ (Note 7)	± 6	± 9	%
	Undervoltage Lockout	Outputs turn OFF		9	V (min)
				11	V (max)
T_{JW}	Warning Flag Temperature	Pin 9 $\leq 0.8V$, $I_L = 2\text{ mA}$	145		$^\circ\text{C}$
$V_F(ON)$	Flag Output Saturation Voltage	$T_J = T_{JW}$, $I_L = 2\text{ mA}$	0.15		V
$I_F(OFF)$	Flag Output Leakage	$V_F = 12V$	0.2	10	μA (max)
T_{SD}	Shutdown Temperature	Outputs Turn OFF	170		$^\circ\text{C}$
I_S	Quiescent Supply Current	All Logic Inputs Low	13	25	mA (max)
t_{ON}	Output Turn-On Delay Time	Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$	300 300		ns
t_{ON}	Output Turn-On Switching Time	Bootstrap Capacitor = 10 nF Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$	100 80		ns
t_{OFF}	Output Turn-Off Delay Times	Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$	200 200		ns
t_{OFF}	Output Turn-Off Switching Times	Bootstrap Capacitor = 10 nF Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$	75 70		ns
t_{PW}	Minimum Input Pulse Width	Pins 3, 4 and 5	1		μs
t_{CP}	Charge Pump Rise Time	No Bootstrap Capacitor	20		μs

ANEXO C. Hoja de Datos inversor 74LS04

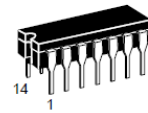


HEX INVERTER

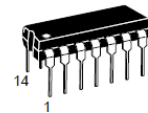


SN54/74LS04

**HEX INVERTER
LOW POWER SCHOTTKY**



**J SUFFIX
CERAMIC
CASE 632-08**



**N SUFFIX
PLASTIC
CASE 646-06**



**D SUFFIX
SOIC
CASE 751A-02**

ORDERING INFORMATION

SN54LSXXJ Ceramic
SN74LSXXN Plastic
SN74LSXXD SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High	54, 74			-0.4	mA
I _{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.5	V	
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	I _{OL} = 4.0 mA, V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table
		74	0.35	0.5	V	I _{OL} = 8.0 mA
I _{IH}	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current (Note 1)	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total, Output HIGH Total, Output LOW			2.4	mA	V _{CC} = MAX
				6.6		

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t _{PLH}	Turn-Off Delay, Input to Output		9.0	15	ns	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Turn-On Delay, Input to Output		10	15	ns	

ANEXO D. Optoaislador 4N25



4N25/ 4N26/ 4N27/ 4N28

Vishay Semiconductors

Optocoupler, Phototransistor Output, With Base Connection

Features

- Isolation Test Voltage 5300 V_{RMS}
- Interfaces with Common Logic Families
- Input-output Coupling Capacitance < 0.5 pF
- Industry Standard Dual-In-Line 6-pin Package

Agency Approvals

- UL File #E52744 System Code H or J
- DIN EN 60747-5-2(VDE0884)
- DIN EN 60747-5-5 pending
- Available with Option 1

Applications

AC Mains Detection
 Reed relay driving
 Switch Mode Power Supply Feedback
 Telephone Ring Detection
 Logic Ground Isolation
 Logic Coupling with High Frequency Noise Rejection

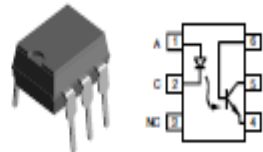
Description

The 4N25 family is an Industry Standard Single Channel Phototransistor Coupler. This family includes the 4N25/ 4N26/ 4N27/ 4N28. Each optocoupler consists of gallium arsenide infrared LED and a silicon NPN phototransistor.

These couplers are Underwriters Laboratories (UL) listed to comply with a 5300 V_{RMS} isolation test voltage. This isolation performance is accomplished through special Vishay manufacturing process.

Compliance to DIN EN 60747-5-2(VDE0884)/ DIN EN 60747-5-5 pending partial discharge isolation specification is available by ordering option 1.

These isolation processes and the Vishay ISO9001 quality program results in the highest isolation performance available for a commercial plastic phototransistor optocoupler.



The devices are also available in lead formed configuration suitable for surface mounting and are available either on tape and reel, or in standard tube shipping containers.

Note:

For additional design information see Application Note 45 Normalized Curves

Order Information

Part	Remarks
4N25	CTR > 20 %, DIP-6
4N26	CTR > 20 %, DIP-6
4N27	CTR > 10 %, DIP-6
4N28	CTR > 10 %, DIP-6
4N25-X006	CTR > 20 %, DIP-6 400 mil (option 6)
4N25-X007	CTR > 20 %, SMD-6 (option 7)
4N25-X009	CTR > 20 %, SMD-6 (option 9)
4N26-X006	CTR > 20 %, DIP-6 400 mil (option 6)
4N26-X007	CTR > 20 %, SMD-6 (option 7)
4N26-X009	CTR > 20 %, SMD-6 (option 9)
4N27-X007	CTR > 10 %, SMD-6 (option 7)
4N27-X009	CTR > 10 %, SMD-6 (option 9)
4N28-X009	CTR > 10 %, SMD-6 (option 9)

For additional information on the available options refer to Option Information.

4N25/ 4N26/ 4N27/ 4N28



Vishay Semiconductors

Absolute Maximum Ratings

$T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified

Stresses in excess of the absolute Maximum Ratings can cause permanent damage to the device. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of this document. Exposure to absolute Maximum Rating for extended periods of the time can adversely affect reliability.

Input

Parameter	Test condition	Symbol	Value	Unit
Reverse voltage		V_R	6.0	V
Forward current		I_F	60	mA
Surge current	$t < 10\text{ }\mu\text{s}$	I_{FSM}	2.5	A
Power dissipation		P_{diss}	100	mW

Output

Parameter	Test condition	Symbol	Value	Unit
Collector-emitter breakdown voltage		V_{CEO}	70	V
Emitter-base breakdown voltage		V_{EB0}	7.0	V
Collector current		I_C	50	mA
Collector current	$t < 1.0\text{ ms}$	I_C	100	mA
Power dissipation		P_{diss}	150	mW

Coupler

Parameter	Test condition	Symbol	Value	Unit
Isolation test voltage		V_{ISO}	5300	V_{RMS}
Creepage			≥ 7.0	mm
Clearance			≥ 7.0	mm
Isolation thickness between emitter and detector			≥ 0.4	mm
Comparative tracking index	DIN IEC 112/VDE0909, part 1		175	
Isolation resistance	$V_{IO} = 500\text{ V}$, $T_{amb} = 25\text{ }^{\circ}\text{C}$	R_{IO}	10^{12}	Ω
	$V_{IO} = 500\text{ V}$, $T_{amb} = 100\text{ }^{\circ}\text{C}$	R_{IO}	10^{11}	Ω
Storage temperature		T_{stg}	-55 to +150	$^{\circ}\text{C}$
Operating temperature		T_{amb}	-55 to +100	$^{\circ}\text{C}$
Junction temperature		T_J	100	$^{\circ}\text{C}$
Soldering temperature	max.10 s, dip soldering; distance to seating plane $\geq 1.5\text{ mm}$	T_{sld}	260	$^{\circ}\text{C}$



Electrical Characteristics

$T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified

Minimum and maximum values are testing requirements. Typical values are characteristics of the device and are the result of engineering evaluation. Typical values are for information only and are not part of the testing requirements.

Input

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Forward voltage ¹⁾	$I_F = 50\text{ mA}$	V_F		1.3	1.5	V
Reverse current ¹⁾	$V_R = 3.0\text{ V}$	I_R		0.1	100	μA
Capacitance	$V_R = 0\text{ V}$	C_D		25		pF

¹⁾ Indicates JEDEC registered values

Output

Parameter	Test condition	Part	Symbol	Min	Typ.	Max	Unit
Collector-base breakdown voltage ¹⁾	$I_C = 100\text{ }\mu\text{A}$		BV_{CBO}	70			V
Collector-emitter breakdown voltage ¹⁾	$I_C = 1.0\text{ mA}$		BV_{CEO}	30			V
Emitter-collector breakdown voltage ¹⁾	$I_E = 100\text{ }\mu\text{A}$		BV_{ECO}	7.0			V
$I_{CBO}(\text{dark})$ ¹⁾	$V_{CB} = 10\text{ V}$, (base open)	4N25			5.0	50	nA
		4N26			5.0	50	nA
		4N27			5.0	50	nA
		4N28			10	100	nA
$I_{CEO}(\text{dark})$ ¹⁾	$V_{CE} = 10\text{ V}$, (emitter open)				2.0	20	nA
Collector-emitter capacitance	$V_{CE} = 0$		C_{CE}		6.0		pF

¹⁾ Indicates JEDEC registered values

Coupler

Parameter	Test condition	Part	Symbol	Min	Typ.	Max	Unit
Isolation voltage ¹⁾	Peak, 60 Hz	4N25	V_{IO}	2500			V
		4N26	V_{IO}	1500			V
		4N27	V_{IO}	1500			V
		4N28	V_{IO}	500			V
Saturation voltage, collector-emitter	$I_{CE} = 2.0\text{ mA}$, $I_F = 50\text{ mA}$		$V_{CE(sat)}$			0.5	V
Resistance, input-output ¹⁾	$V_{IO} = 500\text{ V}$		R_{IO}	100			G Ω
Capacitance (input-output)	$f = 1.0\text{ MHz}$		C_{IO}		0.5		pF

¹⁾ Indicates JEDEC registered values

Current Transfer Ratio

Parameter	Test condition	Part	Symbol	Min	Typ.	Max	Unit
DC Current Transfer Ratio ¹⁾	$V_{CE} = 10\text{ V}$, $I_F = 10\text{ mA}$	4N25	CTR_{DC}	20	50		%
		4N26	CTR_{DC}	20	50		%
		4N27	CTR_{DC}	10	30		%
		4N28	CTR_{DC}	10	30		%

¹⁾ Indicates JEDEC registered value

4N25/ 4N26/ 4N27/ 4N28

Vishay Semiconductors

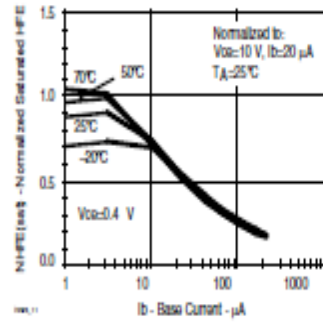


Fig. 11 Normalized HFE vs. Base Current and Temp.

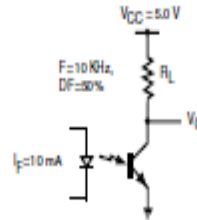


Fig. 14 Switching Schematic

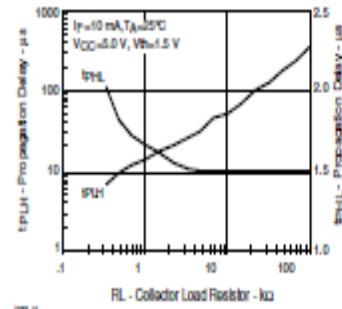


Fig. 12 Propagation Delay vs. Collector Load Resistor

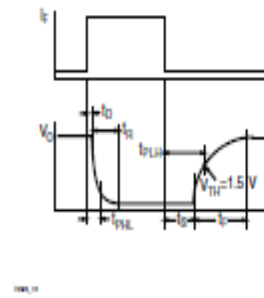




Fig. 13 Switching Timing

ANEXO E. Optoaislador 6N137



January 2011



Single-Channel: 6N137, HCPL2601, HCPL2611 Dual-Channel: HCPL2630, HCPL2631 High Speed 10MBit/s Logic Gate Optocouplers

Features

- Very high speed – 10 MBit/s
- Superior CMR – 10 kV/μs
- Double working voltage-480V
- Fan-out of 8 over -40°C to +85°C
- Logic gate output
- Strobable output
- Wired OR-open collector
- U.L. recognized (File # E90700)

Applications

- Ground loop elimination
- LSTTL to TTL, LSTTL or 5-volt CMOS
- Line receiver, data transmission
- Data multiplexing
- Switching power supplies
- Pulse transformer replacement
- Computer-peripheral interface

Description

The 6N137, HCPL2601, HCPL2611 single-channel and HCPL2630, HCPL2631 dual-channel optocouplers consist of a 850 nm AlGaAs LED, optically coupled to a very high speed integrated photo-detector logic gate with a strobable output. This output features an open collector, thereby permitting wired OR outputs. The coupled parameters are guaranteed over the temperature range of -40°C to +85°C. A maximum input signal of 5mA will provide a minimum output sink current of 13mA (fan out of 8).

An internal noise shield provides superior common mode rejection of typically 10kV/μs. The HCPL2601 and HCPL2631 has a minimum CMR of 5kV/μs. The HCPL2611 has a minimum CMR of 10kV/μs.

Schematics

6N137
HCPL2601
HCPL2611

HCPL2630
HCPL2631

A 0.1μF bypass capacitor must be connected between pins 8 and 5⁽¹⁾.

Package Outlines

Truth Table (Positive Logic)

Input	Enable	Output
H	H	L
L	H	H
H	L	H
L	L	H
H	NC	L
L	NC	H

Single-Channel : 6N137, HCPL2601, HCPL2611 Dual-Channel : HCPL2630, HCPL2631 — High Speed 10MBit/s Logic Gate Optocouplers

Absolute Maximum Ratings ($T_A = 25^\circ\text{C}$ unless otherwise specified)

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions and stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only.

Symbol	Parameter		Value	Units
T_{STG}	Storage Temperature		-55 to +125	$^\circ\text{C}$
T_{OPR}	Operating Temperature		-40 to +85	$^\circ\text{C}$
T_{SOL}	Lead Solder Temperature (for wave soldering only)*		260 for 10 sec	$^\circ\text{C}$
EMITTER				
I_F	DC/Average Forward	Single Channel	50	mA
	Input Current	Dual Channel (Each Channel)	30	
V_E	Enable Input Voltage Not to Exceed V_{CC} by more than 500mV	Single Channel	5.5	V
V_R	Reverse Input Voltage	Each Channel	5.0	V
P_I	Power Dissipation	Single Channel	100	mW
		Dual Channel (Each Channel)	45	
DETECTOR				
V_{CC} (1 minute max)	Supply Voltage		7.0	V
I_O	Output Current	Single Channel	50	mA
		Dual Channel (Each Channel)	50	
V_O	Output Voltage	Each Channel	7.0	V
P_O	Collector Output	Single Channel	85	mW
	Power Dissipation	Dual Channel (Each Channel)	60	

*For peak soldering reflow, please refer to the Reflow Profile on page 11.

Recommended Operating Conditions

The Recommended Operating Conditions table defines the conditions for actual device operation. Recommended operating conditions are specified to ensure optimal performance to the datasheet specifications. Fairchild does not recommend exceeding them or designing to absolute maximum ratings.

Symbol	Parameter	Min.	Max.	Units
I_{FL}	Input Current, Low Level	0	250	μA
I_{FH}	Input Current, High Level	*6.3	15	mA
V_{CC}	Supply Voltage, Output	4.5	5.5	V
V_{EL}	Enable Voltage, Low Level	0	0.8	V
V_{EH}	Enable Voltage, High Level	2.0	V_{CC}	V
T_A	Low Level Supply Current	-40	+85	$^\circ\text{C}$
N	Fan Out (TTL load)		8	

*6.3mA is a guard banded value which allows for at least 20% CTR degradation. Initial input current threshold value is 5.0mA or less.

Electrical Characteristics ($T_A = 0$ to 70°C unless otherwise specified)**Individual Component Characteristics**

Symbol	Parameter	Test Conditions	Min.	Typ.*	Max.	Unit	
EMITTER							
V_F	Input Forward Voltage	$I_F = 10\text{mA}$ $T_A = 25^\circ\text{C}$			1.8	V	
				1.4	1.75		
B_{VR}	Input Reverse Breakdown Voltage	$I_R = 10\mu\text{A}$	5.0			V	
C_{IN}	Input Capacitance	$V_F = 0, f = 1\text{MHz}$		60		pF	
$\Delta V_F / \Delta T_A$	Input Diode Temperature Coefficient	$I_F = 10\text{mA}$		-1.4		mV/°C	
DETECTOR							
I_{OCH}	High Level Supply Current	$V_{CC} = 5.5\text{V}, I_F = 0\text{mA}, V_E = 0.5\text{V}$	Single Channel		7	10	mA
			Dual Channel		10	15	
I_{OCL}	Low Level Supply Current	$V_{CC} = 5.5\text{V}, I_F = 10\text{mA}, V_E = 0.5\text{V}$	Single Channel		9	13	mA
			Dual Channel		14	21	
I_{EL}	Low Level Enable Current	$V_{CC} = 5.5\text{V}, V_E = 0.5\text{V}$		-0.8	-1.6	mA	
I_{EH}	High Level Enable Current	$V_{CC} = 5.5\text{V}, V_E = 2.0\text{V}$		-0.6	-1.6	mA	
V_{EH}	High Level Enable Voltage	$V_{CC} = 5.5\text{V}, I_F = 10\text{mA}$	2.0			V	
V_{EL}	Low Level Enable Voltage	$V_{CC} = 5.5\text{V}, I_F = 10\text{mA}^{(2)}$			0.8	V	

Switching Characteristics ($T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$, $V_{CC} = 5\text{V}$, $I_F = 7.5\text{mA}$ unless otherwise specified)

Symbol	AC Characteristics	Test Conditions	Min.	Typ.*	Max.	Unit	
T_{PLH}	Propagation Delay Time to Output HIGH Level	$R_L = 350\Omega, C_L = 15\text{pF}^{(4)}$ (Fig. 12) $T_A = 25^\circ\text{C}$		20	45	75	ns
						100	
T_{PHL}	Propagation Delay Time to Output LOW Level	$T_A = 25^\circ\text{C}^{(5)}$ $R_L = 350\Omega, C_L = 15\text{pF}$ (Fig. 12)		25	45	75	ns
						100	
$ T_{PHL} - T_{PLH} $	Pulse Width Distortion	$(R_L = 350\Omega, C_L = 15\text{pF})$ (Fig. 12)		3	35	ns	
t_r	Output Rise Time (10-90%)	$R_L = 350\Omega, C_L = 15\text{pF}^{(6)}$ (Fig. 12)		50		ns	
t_f	Output Rise Time (90-10%)	$R_L = 350\Omega, C_L = 15\text{pF}^{(7)}$ (Fig. 12)		12		ns	
t_{ELH}	Enable Propagation Delay Time to Output HIGH Level	$I_F = 7.5\text{mA}, V_{EH} = 3.5\text{V}, R_L = 350\Omega, C_L = 15\text{pF}^{(8)}$ (Fig. 13)		20		ns	
t_{ELL}	Enable Propagation Delay Time to Output LOW Level	$I_F = 7.5\text{mA}, V_{EH} = 3.5\text{V}, R_L = 350\Omega, C_L = 15\text{pF}^{(9)}$ (Fig. 13)		20		ns	
ICM_{IH}	Common Mode Transient Immunity (at Output HIGH Level)	$T_A = 25^\circ\text{C}, V_{CM} = 50\text{V}$ (Peak), $I_F = 0\text{mA}, V_{OH}(\text{Min.}) = 2.0\text{V}, R_L = 350\Omega^{(10)}$ (Fig. 14)	6N137, HCPL2630		10,000	V/μs	
			HCPL2601, HCPL2631	5000	10,000		
ICM_{IL}	Common Mode Transient Immunity (at Output LOW Level)	$R_L = 350\Omega, I_F = 7.5\text{mA}, V_{OL}(\text{Max.}) = 0.8\text{V}, T_A = 25^\circ\text{C}^{(11)}$ (Fig. 14)	6N137, HCPL2630		10,000	V/μs	
			HCPL2601, HCPL2631	5000	10,000		
		$ V_{CM} = 400\text{V}$	HCPL2611	10,000	15,000		

ANEXO F. Configuración Timer 3 y 4 (hoja de datos)

ATmega640/1280/1281/2560/2561

17. 16-bit Timer/Counter (Timer/Counter 1, 3, 4, and 5)

17.1 Features

- True 16-bit Design (that is, allows 16-bit PWM)
- Three Independent Output Compare Units
- Double Buffered Output Compare Registers
- One Input Capture Unit
- Input Capture Noise Canceler
- Clear Timer on Compare Match (Auto Reload)
- Glitch-free, Phase Correct Pulse Width Modulator (PWM)
- Variable PWM Period
- Frequency Generator
- External Event Counter
- Twenty Independent Interrupt sources (TOV1, OCF1A, OCF1B, OCF1C, ICF1, TOV3, OCF3A, OCF3B, OCF3C, ICF3, TOV4, OCF4A, OCF4B, OCF4C, ICF4, TOV5, OCF5A, OCF5B, OCF5C and ICF5)

17.2 Overview

The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement.

Most register and bit references in this section are written in general form. A lower case "n" replaces the Timer/Counter number, and a lower case "x" replaces the Output Compare unit channel. However, when using the register or bit defines in a program, the precise form must be used, that is, TCNT1 for accessing Timer/Counter1 counter value and so on.

A simplified block diagram of the 16-bit Timer/Counter is shown in [Figure 17-1 on page 137](#). For the actual placement of I/O pins, see ["TQFP-pinout ATmega640/1280/2560" on page 2](#) and ["Pinout ATmega1281/2561" on page 4](#). CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the ["Register Description" on page 158](#).

The Power Reduction Timer/Counter1 bit, PRTIM1, in ["PRR0 – Power Reduction Register 0" on page 56](#) must be written to zero to enable Timer/Counter1 module.

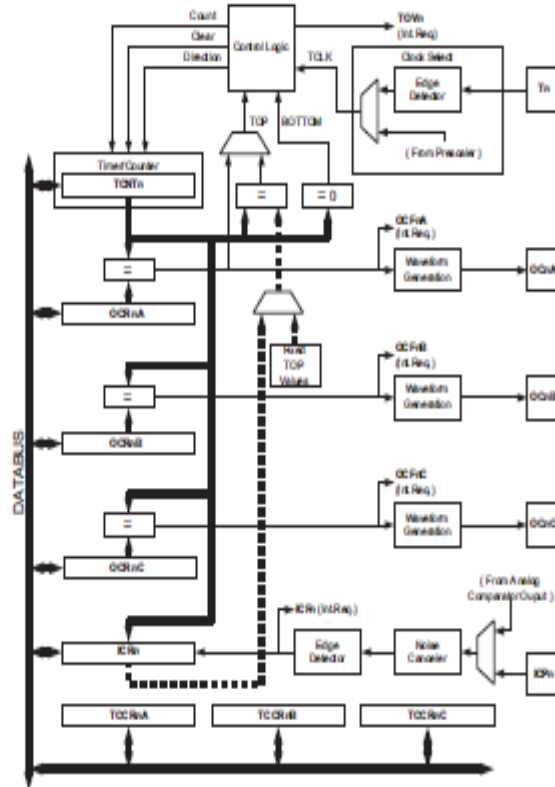
The Power Reduction Timer/Counter3 bit, PRTIM3, in ["PRR1 – Power Reduction Register 1" on page 57](#) must be written to zero to enable Timer/Counter3 module.

The Power Reduction Timer/Counter4 bit, PRTIM4, in ["PRR1 – Power Reduction Register 1" on page 57](#) must be written to zero to enable Timer/Counter4 module.

The Power Reduction Timer/Counter5 bit, PRTIM5, in ["PRR1 – Power Reduction Register 1" on page 57](#) must be written to zero to enable Timer/Counter5 module.

Timer/Counter4 and Timer/Counter5 only have full functionality in the ATmega640/1280/2560. Input capture and output compare are not available in the ATmega1281/2561.

Figure 17-1. 16-bit Timer/Counter Block Diagram⁽¹⁾



Note: 1. Refer to Figure 1-1 on page 2, Table 13-5 on page 79, and Table 13-11 on page 83 for Timer/Counter1 and 3 and 3 pin placement and description.

17.2.1 Registers

The Timer/Counter (TCNTn), Output Compare Registers (OCRnA/B/C), and Input Capture Register (ICRn) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in the section “Accessing 16-bit Registers” on page 138. The Timer/Counter Control Registers (TCCRnA/B/C) are 8-bit registers and have no CPU access restrictions. Interrupt requests (shortened as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFRn). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSKn). TIFRn and TIMSKn are not shown in the figure since these registers are shared by other timer units.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the Tn pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (clk_{Tn}).

The double buffered Output Compare Registers (OCRnA/B/C) are compared with the Timer/Counter value at all time. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OCnA/B/C).

non-PWM modes refer to [Table 17-3 on page 159](#). For fast PWM mode refer to [Table 17-4 on page 159](#), and for phase correct and phase and frequency correct PWM refer to [Table 17-5 on page 160](#).

A change of the COMnx1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOCnx strobe bits.

17.9 Modes of Operation

The mode of operation, that is, the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the *Waveform Generation mode* (WGMn3:0) and *Compare Output mode* (COMnx1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COMnx1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COMnx1:0 bits control whether the output should be set, cleared or toggle at a compare match. See [“Compare Match Output Unit” on page 147](#).

Table 17-2. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to [“Timer/Counter Timing Diagrams” on page 156](#).

17.11 Register Description

17.11.1 TCCR1A – Timer/Counter 1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.2 TCCR3A – Timer/Counter 3 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x00)	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.3 TCCR4A – Timer/Counter 4 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x00)	COM4A1	COM4A0	COM4B1	COM4B0	COM4C1	COM4C0	WGM41	WGM40	TCCR4A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.4 TCCR5A – Timer/Counter 5 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x120)	COM5A1	COM5A0	COM5B1	COM5B0	COM5C1	COM5C0	WGM51	WGM50	TCCR5A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7:6 – COMnA1:0: Compare Output Mode for Channel A
- Bit 5:4 – COMnB1:0: Compare Output Mode for Channel B
- Bit 3:2 – COMnC1:0: Compare Output Mode for Channel C

The COMnA1:0, COMnB1:0, and COMnC1:0 control the output compare pins (OCnA, OCnB, and OCnC respectively) behavior. If one or both of the COMnA1:0 bits are written to one, the OCnA output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COMnB1:0 bits are written to one, the OCnB output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COMnC1:0 bits are written to one, the OCnC output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OCnA, OCnB or OCnC pin must be set in order to enable the output driver.

When the OCnA, OCnB or OCnC is connected to the pin, the function of the COMnx1:0 bits is dependent of the WGMn3:0 bits setting. Table 17-3 on page 159 shows the COMnx1:0 bit functionality when the WGMn3:0 bits are set to a normal or a CTC mode (non-PWM).

• **Bit 1:0 – WGMn1:0: Waveform Generation Mode**

Combined with the WGMn3:2 bits found in the TCCRnB Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see [Table 17-2 on page 148](#). Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare match (CTC) mode, and three types of Pulse Width Modulation (PWM) modes. For more information on the different modes, see [“Modes of Operation” on page 148](#).

Table 17-3. Compare Output Mode, non-PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	Toggle OCnA/OCnB/OCnC on compare match
1	0	Clear OCnA/OCnB/OCnC on compare match (set output to low level)
1	1	Set OCnA/OCnB/OCnC on compare match (set output to high level)

[Table 17-4](#) shows the COMnx1:0 bit functionality when the WGMn3:0 bits are set to the fast PWM mode.

Table 17-4. Compare Output Mode, Fast PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	WGM13:0 – 14 or 15: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected
1	0	Clear OCnA/OCnB/OCnC on compare match, set OCnA/OCnB/OCnC at BOTTOM (non-inverting mode)
1	1	Set OCnA/OCnB/OCnC on compare match, clear OCnA/OCnB/OCnC at BOTTOM (inverting mode)

Note: A special case occurs when OCRnA/OCRnB/OCRnC equals TOP and COMnA1/COMnB1/COMnC1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See [“Fast PWM Mode” on page 150](#) for more details.

Table 17-5 shows the COMnx1:0 bit functionality when the WGMn3:0 bits are set to the phase correct and frequency correct PWM mode.

Table 17-5. Compare Output Mode, Phase Correct and Phase and Frequency Correct PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	WGM13:0 =9 or 11: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected
1	0	Clear OCnA/OCnB/OCnC on compare match when up-counting Set OCnA/OCnB/OCnC on compare match when downcounting
1	1	Set OCnA/OCnB/OCnC on compare match when up-counting Clear OCnA/OCnB/OCnC on compare match when downcounting

Note: A special case occurs when OCRnA/OCRnB/OCRnC equals TOP and COMnA1/COMnB1/COMnC1 is set. See "Phase Correct PWM Mode" on page 152, for more details.

17.11.5 TCCR1B – Timer/Counter 1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x01)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.6 TCCR3B – Timer/Counter 3 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x01)	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.7 TCCR4B – Timer/Counter 4 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0xA1)	ICNC4	ICES4	–	WGM43	WGM42	CS42	CS41	CS40	TCCR4B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.8 TCCR5B – Timer/Counter 5 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x121)	ICNC5	ICES5	–	WGM53	WGM52	CS52	CS51	CS50	TCCR5B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNCn: Input Capture Noise Canceler**

Setting this bit (to one) activates the Input Capture Noise Canceler. When the Noise Canceler is activated, the input from the Input Capture Pin (ICPn) is filtered. The filter function requires four successive equal valued samples of the ICPn pin for changing its output. The input capture is therefore delayed by four Oscillator cycles when the noise canceler is enabled.

• **Bit 6 – ICESn: Input Capture Edge Select**

This bit selects which edge on the Input Capture Pin (ICPn) that is used to trigger a capture event. When the ICESn bit is written to zero, a falling (negative) edge is used as trigger, and when the ICESn bit is written to one, a rising (positive) edge will trigger the capture.

When a capture is triggered according to the ICESn setting, the counter value is copied into the Input Capture Register (ICRn). The event will also set the Input Capture Flag (ICFn), and this can be used to cause an Input Capture Interrupt, if this interrupt is enabled.

When the ICRn is used as TOP value (see description of the WGMn3:0 bits located in the TCCRnA and the TCCRnB Register), the ICPn is disconnected and consequently the input capture function is disabled.

• **Bit 5 – Reserved Bit**

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCRnB is written.

• **Bit 4:3 – WGMn3:2: Waveform Generation Mode**

See TCCRnA Register description.

• **Bit 2:0 – CSn2:0: Clock Select**

The three clock select bits select the clock source to be used by the Timer/Counter, see [Figure 17-10 on page 156](#) and [Figure 17-11 on page 156](#).

Table 17-6. Clock Select Bit Description

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{IC}/1$ (No prescaling)
0	1	0	$clk_{IC}/8$ (From prescaler)
0	1	1	$clk_{IC}/64$ (From prescaler)
1	0	0	$clk_{IC}/256$ (From prescaler)
1	0	1	$clk_{IC}/1024$ (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

If external pin modes are used for the Timer/Counter, transitions on the Tn pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

17.11.9 TCCR1C – Timer/Counter 1 Control Register C

Bit	7	6	5	4	3	2	1	0	
(0x82)	FOC1A	FOC1B	FOC1C	–	–	–	–	–	TCCR1C
Read/Write	W	W	W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The Input Capture is updated with the counter (TCNTn) value each time an event occurs on the ICPn pin (or optionally on the Analog Comparator output for Timer/Counter1). The Input Capture can be used for defining the counter TOP value.

The Input Capture Register is 16-bit in size. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See "Accessing 16-bit Registers" on page 138.

17.11.33 TIMSK1 – Timer/Counter 1 Interrupt Mask Register

BIT	7	6	5	4	3	2	1	0	
(0x0F)	–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.34 TIMSK3 – Timer/Counter 3 Interrupt Mask Register

BIT	7	6	5	4	3	2	1	0	
(0x71)	–	–	ICIE3	–	OCIE3C	OCIE3B	OCIE3A	TOIE3	TIMSK3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.35 TIMSK4 – Timer/Counter 4 Interrupt Mask Register

BIT	7	6	5	4	3	2	1	0	
(0x72)	–	–	ICIE4	–	OCIE4C	OCIE4B	OCIE4A	TOIE4	TIMSK4
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.36 TIMSK5 – Timer/Counter 5 Interrupt Mask Register

BIT	7	6	5	4	3	2	1	0	
(0x73)	–	–	ICIE5	–	OCIE5C	OCIE5B	OCIE5A	TOIE5	TIMSK5
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5 – ICIE_n: Timer/Counter_n, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter_n Input Capture interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 105) is executed when the ICF_n Flag, located in TIFR_n, is set.

- **Bit 3 – OCIE_nC: Timer/Counter_n, Output Compare C Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter_n Output Compare C Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 105) is executed when the OCF_nC Flag, located in TIFR_n, is set.

- **Bit 2 – OCIE_nB: Timer/Counter_n, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter_n Output Compare B Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 105) is executed when the OCF_nB Flag, located in TIFR_n, is set.

ANEXO G. Código Fuente en Arduino.

```
/*PROGRAM TO CONTROL A FURUTA PENDULUM AND  
GETTING GRAPHS WITH MATLAB*/
```

```
#include <stdio.h>
```

```
#include <avr/io.h>
```

```
#include <avr/interrupt.h>
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(48, 49, 45, 44, 43, 42);
```

```
/*encoder pins*/
```

```
int A_2=18;
```

```
int B_2=19;
```

```
int A_1=20;
```

```
int B_1=21;
```

```
const int s=53;
```

```
volatile float count_2 =0.0;
```

```
volatile float count_1=0.0;
```

```
const float dt=0.6;
```

```
/*PID_1 gains*/
```

```
const float kp_1=5.0;
```

```
const float ki_1=0.0;//0.0*n;
```

```
const float kd_1=0.8;//0.5/n;
```

```

/*PID_2 gains*/

const float kp_2=40;//60.0;

const float ki_2=2;//4.0*n;

const float kd_2=2.0;//2.0/n;

long i=0;

int t=0;

float setpoint_theta=180;

float setpoint_alpha=0;

/*PID variables*/

volatile float alpha=0,error1=0,lastE1=0,Area_1=0,Int_E1=0,Der_E1=0,out1=0;

volatile float theta=0,error2=0,lastE2=0,Area_2=0,Int_E2=0,Der_E2=0,out2=0;

/*Interrupt Service Routine*/

ISR(TIMER3_COMPA_vect)

{ sei();

digitalWrite(22,HIGH);

    theta=count_2*(360.0/10000.0);

    alpha=count_1*(360.0/(1600));

    if (digitalRead(s)==HIGH)

    {

        if(alpha>200 || alpha<-200){

            // Protección, impide que el motor gire mas alla de lo que permite el cable

            motorStop();

```

```
t=10;
```

```
}
```

```
/*
```

Swing up y control PID

t=0 es una perturbación inicial, luego, t=1, hace que el “brazo” del péndulo

vaya hacia la posición inicial y trata de evitar que salga de allí

(mini control proporcional), una vez el péndulo alcanza los 160 grados,

comienza t=2 que es un control pid, sin realimentación, sobre el ángulo

del péndulo, y que se encarga de estabilizar la velocidad del péndulo,

luego cuando la velocidad del péndulo es cercana a cero y el ángulo del

brazo es menor a 10 grados, inicia t=3 que es el control pid sobre el

ángulo del péndulo y retroalimentado con la señal de un pid sobre el

ángulo del brazo.

```
*/
```

```
if(t==0){
```

```
    motor_CW();
```

```
    OCR4A=255.0-90;
```

```
    if(theta<-35){
```

```
        t=1;
```

```
    }
```

```
}
```

```

if(t==1){

    error1=alpha;

    out1=6*error1;

    if (out1<0)

    { out1=out1*(-1);

        if(out1>255){out1=255;}

        motor_CW();

        OCR4A=255.0-out1;

        out1=out1*(-1);

    }

    if (out1>0)

    { if(out1>255){out1=255;}

        motor_CCW();

        OCR4A=255.0-out1;

    }

    if (out1==0){

        motorStop();

    }

    if(theta>160){

        t=2;

    }

}

```

```

if(t==2){

    error1=alpha;

    Area_1=(100*lastE1+100*error1)*0.01*dt/2;

    Int_E1=(Area_1+Int_E1);

    Der_E1=(100*error1-100*lastE1)/dt;

    out1=5*error1;

    theta=theta-out1/80;

    error2=theta-setpoint_theta;

    Area_2=(100*lastE2+100*error2)*0.01*dt/2;

    Int_E2=(Area_2+Int_E2);

    Der_E2=(100*error2-100*lastE2)/dt;

    out2=kp_2*error2+ki_2*Int_E2+3*kd_2*Der_E2;

    if(Der_E2>-1 && Der_E2<1){

        i=1;

    }

    if(alpha>-10 && alpha<10 && i==1){

        t=3;

    }

    if (theta>150 && theta <210){

        if (out2<0)

```

```

{ out2=out2*(-1);

  if(out2>255){out2=255;}

  motor_CW();

  OCR4A=255.0-out2;

  out2=out2*(-1);

}

if (out2>0)

{ if(out2>255){out2=255;}

  motor_CCW();

  OCR4A=255.0-out2;

}

if (out2==0){

  motorStop();

}

}else{

  motorStop();

}

lastE1=error1;

lastE2=error2;

}

if(t==3){

```

```

error1=alpha-setpoint_alpha;

Area_1=(100*lastE1+100*error1)*0.01*dt/2;

Int_E1=(Area_1+Int_E1);

Der_E1=(100*error1-100*lastE1)/dt;

out1=kp_1*error1 + ki_1*Int_E1 + kd_1*Der_E1;

theta=theta-out1/80;

error2=theta-setpoint_theta;

Area_2=(100*lastE2+100*error2)*0.01*dt/2;

Int_E2=(Area_2+Int_E2);

Der_E2=(100*error2-100*lastE2)/dt;

out2=kp_2*error2+ki_2*Int_E2+kd_2*Der_E2;

if (theta>155 && theta <205){

    if (out2<0)

    { out2=out2*(-1);

        if(out2>255){out2=255;}

        motor_CW();

        OCR4A=255.0-out2;

        out2=out2*(-1);

    }

    if (out2>0)

    { if(out2>255){out2=255;}

        motor_CCW();

```

```

        OCR4A=255.0-out2;

    }

    if (out2==0){

        motorStop();

    }

    }else{

        motorStop();

        //t=1;

    }

    lastE1=error1;

    lastE2=error2;

}

}

}else{

    motorStop();

    Int_E2=0;

    Int_E1=0;}

digitalWrite(22,LOW);

}

void setup()

{/*open serial communication*/

    Serial.begin(9600);

    /*switch*/

```

```

pinMode(s,INPUT);

pinMode(22,OUTPUT);

/*channels A and B as inputs*/

pinMode(A_2, INPUT);

pinMode(B_2, INPUT);

pinMode(A_1, INPUT);

pinMode(B_1, INPUT);

/*to active encoder interruptions*/

attachInterrupt(5, doA2, CHANGE);

attachInterrupt(4, doB2, CHANGE);

attachInterrupt(3, doA1, CHANGE);

attachInterrupt(2, doB1, CHANGE);

/*Enable LCD interface*/

lcd.begin(16,2);

/*pins to H-bridge as outputs*/

DDRH=DDRH|B00111000;//PH5=D PH4=B PH3=P

TCCR4A|=(1<<COM4A1)|(1<<WGM40);

TCCR4B=0x02;//TCCR4B|=(1<<CS41)|(1<<CS40);

/*to set ISR*/

TCCR3A=0;

TCCR3B=0;

OCR3A=32000*dt;

```

```

TCCR3B |= (1<<WGM32);

TCCR3B |= (0<<CS30);

TCCR3B |= (1<<CS31);

TIMSK3=(1<<OCIE3A);

sei();

// message();

}

void loop()

{Serial.write(int(count_1*360.0/1600.0));

delay(10);

Serial.write(int(count_2*360.0/10000.0));

delay(10);

  lcd.clear();

  lcd_prints();

}

/*Functions to read encoders*/

void doA1(){

  // look for a low-to-high on channel A

  if (digitalRead(A_1) == HIGH) {

    // check channel B to see which way encoder is turning

    if (digitalRead(B_1) == LOW) {

      count_1 = count_1 + 1;    // CW

```

```

    }

    else {

        count_1 = count_1 - 1;    // CCW

    }

}

else // must be a high-to-low edge on channel A

{

    // check channel B to see which way encoder is turning

    if (digitalRead(B_1) == HIGH) {

        count_1 = count_1 + 1;    // CW

    }

    else {

        count_1 = count_1 - 1;    // CCW

    }

}

}

void doB1(){

    // look for a low-to-high on channel B

    if (digitalRead(B_1) == HIGH) {

        // check channel A to see which way encoder is turning

        if (digitalRead(A_1) == HIGH) {

            count_1 = count_1 + 1;    // CW


```

```

    }

    else {

        count_1 = count_1 - 1;    // CCW

    }

}

// Look for a high-to-low on channel B

else {

    // check channel B to see which way encoder is turning

    if (digitalRead(A_1) == LOW) {

        count_1 = count_1 + 1;    // CW

    }

    else {

        count_1 = count_1 - 1;    // CCW

    }

}

}

void doA2(){

    // digitalWrite(28,HIGH);

    // look for a low-to-high on channel A

    if (digitalRead(A_2) == HIGH) {

        // check channel B to see which way encoder is turning

        if (digitalRead(B_2) == LOW) {

```

```

        count_2 = count_2 + 1;    // CW
    }

    else {

        count_2 = count_2 - 1;    // CCW
    }

}

else // must be a high-to-low edge on channel A
{

    // check channel B to see which way encoder is turning
    if (digitalRead(B_2) == HIGH) {

        count_2 = count_2 + 1;    // CW
    }

    else {

        count_2 = count_2 - 1;    // CCW
    }

}

// digitalWrite(28,LOW);

}

void doB2(){

    // look for a low-to-high on channel B

    if (digitalRead(B_2) == HIGH) {

        // check channel A to see which way encoder is turning

```

```

if (digitalRead(A_2) == HIGH) {
    count_2 = count_2 + 1;    // CW
}

else {
    count_2 = count_2 - 1;    // CCW
}

}

// Look for a high-to-low on channel B

else {

    // check channel B to see which way encoder is turning

    if (digitalRead(A_2) == LOW) {
        count_2 = count_2 + 1;    // CW
    }

    else {
        count_2 = count_2 - 1;    // CCW
    }

}

}

/*Functions to H-bridge control*/

void motor_CCW()

{

    PORTH=0x10;

```

```

}

void motor_CW()

{
    PORTH=0x30;
}

void motorStop()

{
    PORTH=0x28;
}

/*Functions to display on LCD*/

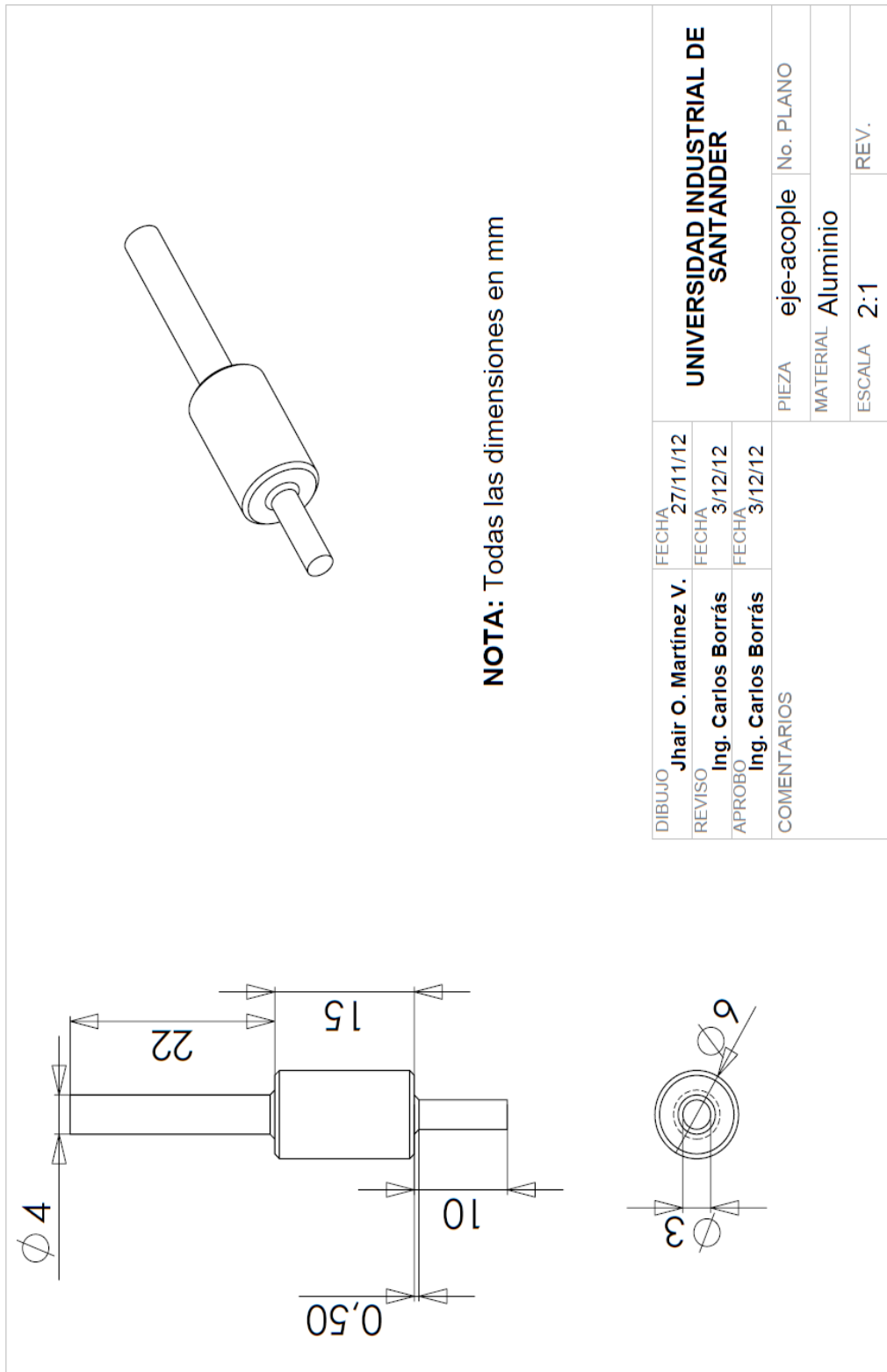
void lcd_prints()

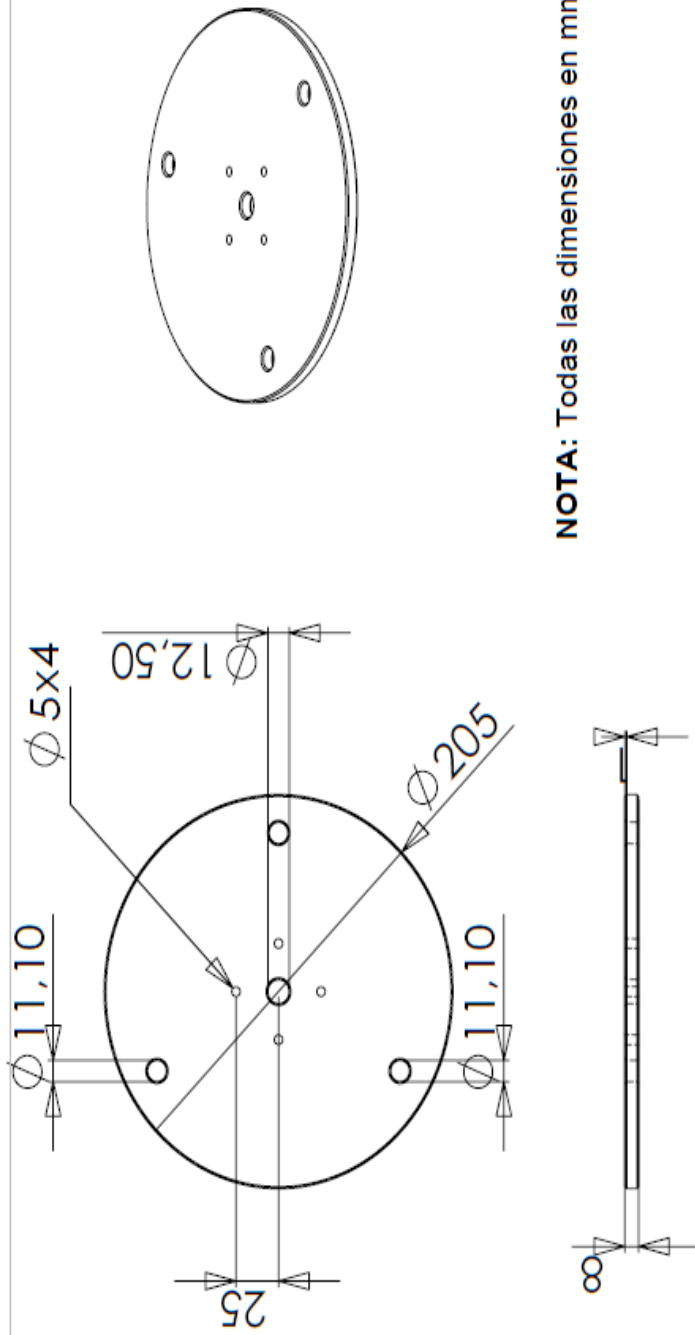
{
    lcd.setCursor(0,0);
    lcd.print("alpha= ");
    lcd.setCursor(6, 0);
    lcd.print(int(alpha));
    lcd.setCursor(0,1);
    lcd.print("theta=");
    lcd.print(int(theta));
    delay(50);
}

```

```
void message()
{
  lcd.setCursor(2,0);
  lcd.print("WELLCOME TO");
  delay(1500);
  lcd.clear();
  lcd.setCursor(1,1);
  lcd.print("CONTROL LAB OF");
  delay(1500);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("FURUTA PENDULUM");
  delay(1500);
}
```

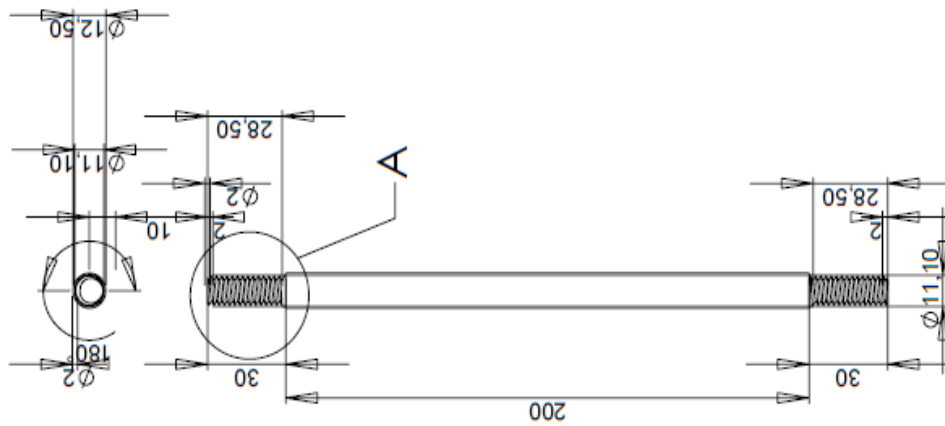
ANEXO H. Planos mecánicos



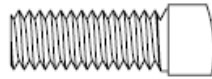


NOTA: Todas las dimensiones en mm

DIBUJO	Jhair O. Martínez V.	FECHA	27/11/12
REVISO	Ing. Carlos Borrás	FECHA	3/12/12
APROBO	Ing. Carlos Borrás	FECHA	3/12/12
COMENTARIOS			
PIEZA	Disco	No. PLANO	
MATERIAL	Acero inoxidable		
ESCALA	1:3	REV.	



NOTA: Rosca UNC, 14 hilos por pulgada, paso de la rosca 1,8143 mm



DETALLE A

NOTA: Todas las dimensiones en mm

DIBUJO	Jhair O. Martínez V.	FECHA	27/11/12	UNIVERSIDAD INDUSTRIAL DE SANTANDER	
REVISO	Ing. Carlos Borrás	FECHA	3/12/12		
APROBO	Ing. Carlos Borrás	FECHA	3/12/12		
COMENTARIOS					
PIEZA	Espigo	No. PLANO		MATERIAL	Acero Inoxidable
ESCALA	1:2	REV.			

ANEXO I. Esquemático del microcontrolador ATmega2560 de Atmel.

