



UNIVERSIDAD AUTÓNOMA DEL ESTADO  
DE HIDALGO

---

---

INSTITUTO DE CIENCIAS BÁSICAS E  
INGENIERÍA

Construcción y Control de un Sistema Pendubot  
Usando Un Microcontrolador Freescale  
DSP MC56F8367

T E S I S  
PARA OBTENER EL TÍTULO DE  
Lic. en Ing. en Electrónica y Telecomunicaciones  
P R E S E N T A :  
P. de I. Pedro Lagos Eulogio

DIRECTOR DE TESIS:  
Dr. Norberto Hernández Romero

Diciembre de 2009

# Resumen

El PENDUBOT (PENDUlum roBOT) es un sistema mecatrónico con modelo matemático caótico y no lineal que se usa académicamente en sistemas lineales y con fines de investigación en sistemas de control no lineal y robótica. El proyecto está enfocado a la construcción, instrumentación y control del PENDUBOT. El dispositivo para implementar el control computacional es un microcontrolador de Freescale DSP MC56F8367, una interfaz de potencia de 20 A. basada en MOSFETs y una interfaz gráfica soportada en LabVIEW. Para la construcción del sistema mecánico se usará un motor de c.d. de 100W, 2 encoders del tipo absoluto de 2000 pulsos por revolución (ppr), dos eslabones de aluminio y algunas bridas.

# Contenido

Lista de Figuras . . . . .	v
Lista de Tablas . . . . .	viii
<b>Introducción</b>	<b>1</b>
Objetivos . . . . .	1
Objetivo General . . . . .	1
Objetivos Específicos . . . . .	2
Justificación . . . . .	3
Estado del Arte . . . . .	4
Hipótesis . . . . .	4
<b>1 Modelado Matemático</b>	<b>6</b>
1.1 Estrategias de Modelado . . . . .	6
1.2 Derivación de los Parámetros . . . . .	7
1.2.1 Brazo Principal . . . . .	9
1.2.2 Brazo Secundario . . . . .	10
1.3 Derivación del Modelo No-lineal . . . . .	12
1.3.1 Ecuaciones Dinámicas . . . . .	12
1.3.2 Ecuaciones de Estado . . . . .	18
1.3.3 Validación del Modelo: comparado con el sistema real . . . . .	21
1.4 Linealización . . . . .	26
1.4.1 Aproximación Teórica . . . . .	26
1.4.2 Cálculos . . . . .	27
<b>2 Diseño del Control por Retroalimentación de Estados</b>	<b>35</b>
2.1 Control de Subida . . . . .	35
2.1.1 Linealización Parcial . . . . .	36
2.1.2 Control PD . . . . .	38

<i>CONTENIDO</i>	iv
2.2 Control de Equilibrio . . . . .	40
<b>3 Pruebas y Resultados</b>	<b>45</b>
3.1 Prueba del Control de Subida . . . . .	45
3.1.1 Resultados . . . . .	47
3.2 Prueba del Control de Equilibrio . . . . .	48
3.2.1 Perturbación en el Brazo Principal . . . . .	49
3.2.2 Perturbación en el Brazo Secundario . . . . .	50
3.2.3 Resultado . . . . .	51
<b>4 Conclusiones</b>	<b>53</b>
4.1 Conclusiones . . . . .	53
4.2 Trabajos Futuros . . . . .	56
<b>Apéndices</b>	<b>60</b>
<b>A Construcción de la Interfaz de Potencia</b>	<b>60</b>
<b>B Interfaz Gráfica con Labview</b>	<b>65</b>
B.1 Diseño de la Interfaz de Comunicación . . . . .	66
B.1.1 Interfaz Análoga . . . . .	66
B.1.2 Interfaz digital . . . . .	67
B.2 Diseño Gráfico de las Posiciones de los Brazos . . . . .	67
B.3 Diseño Gráfico de los Botones de Control . . . . .	68
B.4 Diseño Gráfico de los Brazos Primario y Secundario . . . . .	72
<b>C Interfaz de Posición Angular</b>	<b>76</b>
C.1 Encoders . . . . .	76
C.2 Lectura de las Posiciones . . . . .	80
C.3 Filtro Pasa Bajos . . . . .	80
<b>D Materiales y Costos</b>	<b>83</b>
<b>E Programa de Control en el Microcontrolador</b>	<b>85</b>

# Lista de Figuras

1.1	PENDUBOT . . . . .	8
1.2	Modelo en Matlab . . . . .	22
1.3	Theta 1 con Labview . . . . .	23
1.4	Simulación de Theta 1 en Matlab . . . . .	23
1.5	Theta 2 con Labview . . . . .	24
1.6	Simulación de Theta 2 en Matlab . . . . .	24
1.7	Theta 1 y Theta 2 con Labview . . . . .	25
1.8	Simulación de Theta 2 y Theta 2 en Matlab . . . . .	25
1.9	Punto de Equilibrio. . . . .	31
2.1	Control de Subida . . . . .	39
2.2	Retroalimentación de Estados . . . . .	41
2.3	Control de Retroalimentación. . . . .	42
2.4	Simulación de K en Matlab . . . . .	44
3.1	Posición de Equilibrio . . . . .	46
3.2	Primer Prueba del Control de Subida . . . . .	46

3.3	Segunda Prueba del Control de Subida . . . . .	48
3.4	Perturbación en el Brazo Primario . . . . .	50
3.5	Perturbación en el Brazo Secundario . . . . .	51
3.6	Perturbación Grande . . . . .	52
A.1	Pulse Width Modulation. . . . .	61
A.2	Salida de la Interfaz de Potencia. . . . .	63
A.3	Circuito Electrónico de la Interfaz de Potencia. . . . .	64
B.1	Bloque de la Interfaz Análoga. . . . .	66
B.2	Bloque de la Interfaz Digital. . . . .	67
B.3	Gráfica para las Posiciones Angulares. . . . .	68
B.4	Diagrama de Bloques de un Pin de Control. . . . .	69
B.5	Bloques de Control. . . . .	71
B.6	Diagrama a bloques de los gráficos. . . . .	72
B.7	Diagrama a Bloques de la Interfaz Gráfica. . . . .	73
B.8	Panel Frontal de la Interfaz Gráfica. . . . .	74
C.1	Encoder Rotativo Codificador en Cuadratura. . . . .	77
C.2	Primera Tipo de Decodificación. . . . .	78
C.3	Segundo Tipo de Decodificación. . . . .	79
C.4	Tercer Tipo de Decodificación. . . . .	79
C.5	Filtro Pasa Bajos de Primer Orden. . . . .	81
C.6	Espectro de Frecuencia para el Filtro Pasa Bajos. . . . .	81

*LISTA DE FIGURAS*

vii

E.1 Diagrama de bloques del Programa de Control . . . . .	86
---	----

# Lista de Tablas

1.1	Parámetros del sistema. . . . .	9
1.2	Parámetros del Brazo Principal. . . . .	10
1.3	Parámetros del Brazo Secundario . . . . .	11
1.4	Punto de linealización . . . . .	31
A.1	Tabla de Verdad de la Interfaz. . . . .	62
D.1	Presupuesto. . . . .	84



## **Introducción**

El nombre PENDUBOT, proviene de (PENDUulum robot), que es un sistema electromecánico, subactuado de enlaces rígidos, planar y rotacional de dos grados de libertad, cuenta con un actuador, localizado en el primera brazo, mientras que la segunda articulación, se mueve libremente a lo largo del movimiento de la primera, dado por el acoplamiento que existe entre ambas articulaciones[16].

## **Objetivo General**

Diseñar el prototipo de un PENDUBOT, que permita realizar pruebas de control por retroalimentación de estados y sirva de base para la validación de nuevas técnicas de control.

## Objetivos Específicos

1. Construir el sistema mecánico de dos eslabones con la conexión a un motor de c.d. y dos encoders para cada eslabón.
2. Desarrollar una interfaz de potencia para el motor de c.d. de 30V, 20 A.
3. Desarrollar una interfaz para los encoders absolutos con resolución de 2000 ppr, incluyendo el programa en C para poder contabilizar la posición en radianes.
4. Desarrollar una interfaz gráfica para controlar el inicio, magnitud de las ganancias del control, visualización virtual del PENDUBOT y graficación  $X - Y$  de las posiciones de cada eslabón.
5. Obtener el modelo matemático mediante *Euler-Lagrange*, linealizar el sistema en un punto de operación y diseñar las ganancias para la retroalimentación de estados.
6. Desarrollar el algoritmo de control en el microcontrolador DSP MC56F8367.
7. Realizar la integración del sistema mecánico, interfaz de potencia, interfaz de sensores, microcontrolador y programa en lenguaje C.
8. Realizar pruebas y reportar resultados.

## Justificación

El PENDUBOT, es un sistema electromecánico, que ha sido ampliamente estudiado para probar diferentes técnicas de control, que van desde controles por retroalimentación de estados hasta controles adaptables por medio de técnicas heurísticas [3], siendo un sistema sencillo en su construcción mecánica, simple en la instrumentación, pero complejo en su modelo matemático. Sin embargo, cuando el objetivo es probar alguna técnica de control es ideal para esta tarea. Por lo tanto, este trabajo consiste en desarrollar el prototipo didáctico de un PENDUBOT con la instrumentación necesaria y probar el algoritmo de control por retroalimentación de estados y que este sea utilizado para la investigación. El sistema desarrollado también cuenta con una interfaz gráfica desarrollada en LabVIEW con la finalidad de adquirir los datos y poderlos visualizar en una gráfica  $X - Y$ , con el objetivo de reportar resultados. La posibilidad de adaptar nuevos controles en el PENDUBOT, es de gran importancia en diferentes áreas, siendo este un sistema caótico no lineal, ayuda a la validación de nuevas técnicas de control, estas nuevas técnicas de control pueden ser adaptadas a procesos químicos, mecánicos, eléctricos, etc., con el fin de mejorar la eficiencia de estos.

## Estado del Arte

El PENDUBOT, se utiliza como material didactico para la investigación en el control de la dinámica no lineal, la robótica y el diseño de sistemas de control. Este dispositivo es de dos enlaces, con un actuador en el hombro (enlace 1) y sin actuador en el codo (enlace 2). El enlace 2 se mueve libremente alrededor del enlace 1 y el objetivo de control es el mecanismo para llevar a los puntos inestables a un punto de equilibrio.

Existen diversos sistemas mecánicos similares, entre los que se encuentran: el péndulo simple invertido, el doble péndulo invertido, el acrobot[4], el robot planar sobreactuado [1], etc. Diversas estrategias de control se han propuesto para el péndulo invertido en [2], [10], [11] y [15].

Block [3] propone una estrategia de control basada en dos algoritmos para controlar el PENDUBOT. Para el control de subida, Spong y Block [16] utilizaron técnicas de linealización parcial y para el control de equilibrio, utilizarón la linealización alrededor del punto de equilibrio deseado, además, ocuparon técnicas de control basadas en el regulador lineal cuadrático (LQR) y la colocación de polos, como resultado de esta implementación de controles, se observa que la posición vertical del enlace 2, se alcanzada rápidamente, sin embargo, no presentan un análisis de estabilidad para dichos enlaces.

Para nuestro conocimiento, sólo existe esta solución en la literatura para resolver el problema de subida del enlace 2 del PENDUBOT.

## **Hipótesis**

El PENDUBOT, es controlable en un punto de operación sujeto a pequeñas perturbaciones en tanto este linealizado y puede ser controlado por la técnica de retroalimentación de estados, este y otros tipos de controles pueden implementarse en un microcontrolador DSP MC56F8367.

# Capítulo 1

## Modelado Matemático

### 1.1 Estrategias de Modelado

En mecánica dos métodos de aproximación se pueden utilizar para modelar un sistema, estos métodos pueden ser:

- Aproximación de Newton
- Aproximación de Lagrange

El método de Newton se basa en el principio fundamental de la dinámica, dicho método consta en el calculo de las fuerzas que actúan sobre los cuerpos en movimiento, puesto que, ambos brazos están en constante movimiento es difícil conocer las fuerzas que interactúan en el sistema, por lo tanto, resulta complicado y laborioso modelarlo con base en esta aproximación. Por otro lado, por medio de la aproximación de Lagrange basada en una descripción energética del sistema [16], resulta fácil modelar

este sistema, ya que únicamente es necesario conocer la energía cinética y potencial del mismo. Con una asignación correcta de los parámetros se conoce como se comporta este sistema en cualquier posición, en este caso el conocimiento del ángulo del Brazo Principal y el Brazo Secundario son suficientes para describir completamente el sistema. La derivación del modelo físico es efectuada en 3 pasos:

1. Derivación de todos los parámetros físicos del PENDUBOT.
2. Derivación de las ecuaciones no-lineales usando la teoría de Lagrange.
3. Linealización del modelo no-lineal.

## **1.2 Derivación de los Parámetros**

La Tabla 1.1 muestra los parámetros asignados al sistema PENDUBOT, y la Figura 1.1 muestra gráficamente como fueron asignados dichos parámetros.

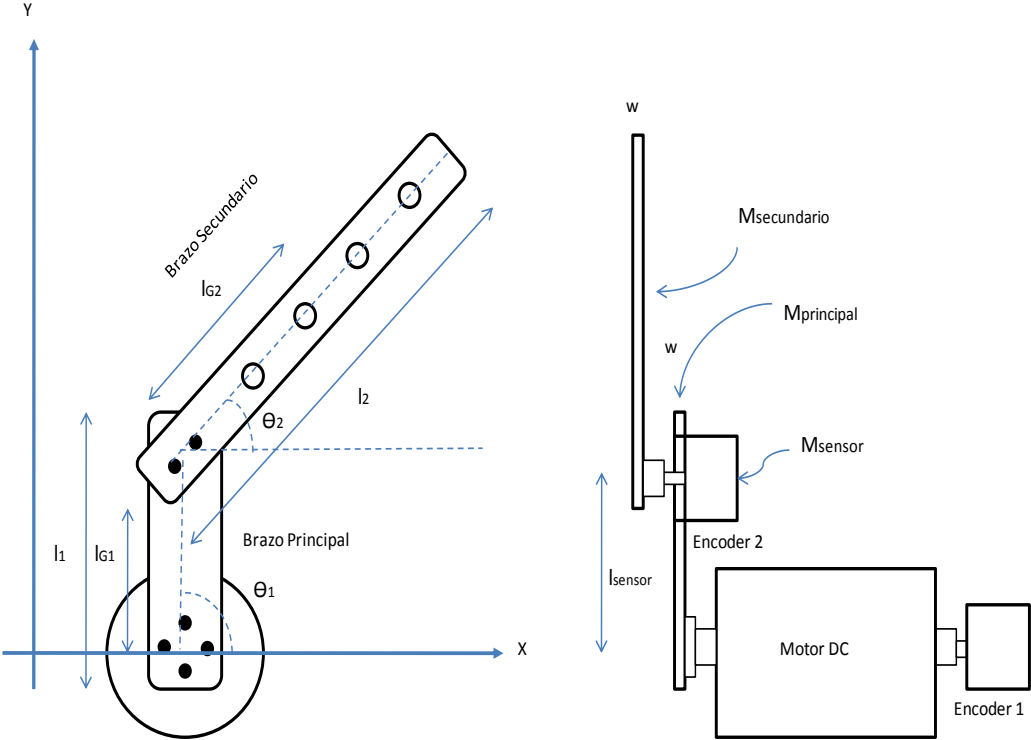


Figura 1.1: PENDUBOT



Parámetros	
Nomenclatura	Significado
$M_1$	Masa total del Brazo Principal
$M_2$	Masa total del Brazo Secundario
$l_1$	Longitud del Brazo Principal
$l_2$	Longitud del Brazo Secundario
$l_{G1}$	Distancia desde el eje de rotación al centro de gravedad del Brazo Principal
$l_{G2}$	Distancia desde el eje de rotación al centro de gravedad del Brazo Secundario
$\theta_1$	Ángulo del Brazo Principal respecto al eje $x$
$\theta_2$	Ángulo del Brazo Secundario respecto al eje $x$

Tabla 1.1: Parámetros del sistema.

### 1.2.1 Brazo Principal

Para modelar el sistema, es fundamental conocer los parámetros del mismo y calcular los momentos de inercia, los parámetros del Brazo Principal, se calculan por simple medición de longitud y peso, esto se realiza apartando cada una de las piezas del Brazo Principal, los resultados se muestran en la Tabla 1.2.

Una vez conociendo los parámetros mencionados es posible calcular el momento de inercia del Brazo Principal. El momento total del Brazo Principal es directamente proporcional a la suma de todos los momento que actúan en dicho brazo[9], esto es:

$$J_1 = J_{principal} + J_{sensor} \quad (1.1)$$

PARÁMETROS DEL BRAZO PRINCIPAL	
Nombre del Parámetro	Medida
Longitud total del Brazo Principal	$l_1 = 0.20 \text{ m}$
Distancia desde el eje de rotación al centro de gravedad	$l_{g1} = 0.08 \text{ m}$
Grosor del Brazo Principal	$w = 0.004 \text{ m}$
Longitud del sensor	$l_{sensor} = 0.145 \text{ m}$
Masa del Brazo Principal	$M_{principal} = 0.082 \text{ Kg}$
Masa del sensor	$M_{sensor} = 0.2815 \text{ Kg}$
Masa total del Brazo Principal	$M_1 = 0.3635 \text{ Kg}$

Tabla 1.2: Parámetros del Brazo Principal.

Donde:

- $J_{principal} = \frac{1}{12}M_{principal}(l_1^2 + w^2) = 0.2734 \times 10^{-3} \text{ Kg.m}^2$
- $J_{sensor} = M_{sensor}l_{sensor}^2 = 5.9185 \times 10^{-3} \text{ Kg.m}^2$

El momento de inercia total del Brazo Principal es:

$$J_1 = 6.1919 \times 10^{-3} \text{ Kg.m}^2$$

### 1.2.2 Brazo Secundario

Para modelar el sistema es fundamental conocer los parámetros del mismo y así poder calcular los momentos de inercia, los parámetros del Brazo Secundario se calculan por simple medición de longitud y peso, esto se realiza apartando cada una de las piezas del Brazo Secundario, los resultados se muestran en la Tabla 1.3.

PARÁMETROS DEL BRAZO SECUNDARIO	
Nombre del Parámetro	Medida
Longitud total del Brazo Secundario	$l_2 = 0.35 \text{ m}$
Distancia desde el eje de rotación al centro de gravedad	$l_{g2} = 0.158 \text{ m}$
Grosor del Brazo Secundario	$w = 0.005 \text{ m}$
Masa del Brazo Secundario	$M_{secundario} = 0.1113 \text{ Kg}$
Masa total del Brazo Secundario	$M_2 = 0.1113 \text{ Kg}$

Tabla 1.3: Parámetros del Brazo Secundario

Una vez conociendo los parámetros mencionados es posible calcular el momento de inercia del Brazo Secundario. El momento total del Brazo Secundario es directamente proporcional a la suma de todos los momento que actúan en dicho brazo[9], esto es:

$$J_2 = J_{secundario} \quad (1.2)$$

Donde:

- $J_{secundario} = \frac{1}{12} M_{secundario} (l_2^2 + w^2) = 1.1364 \times 10^{-3} \text{ Kg.m}^2$

El momento de inercia total del Brazo secundario es:

$$J_2 = 1.1364 \times 10^{-3} \text{ Kg.m}^2$$

## 1.3 Derivación del Modelo No-lineal

### 1.3.1 Ecuaciones Dinámicas

De acuerdo a la teoría de Lagrange, un sistema puede ser completamente descrito por medio de la asignación y el uso de parámetros cuidadosamente asignados

$q_i(t) = 1 \dots n$ , entonces, el sistema estudiado puede ser normalizado por medio de las siguientes ecuaciones de Lagrange[3].

$$L(q_i, \dot{q}_i) = E_k - E_p, i = 1 \dots n \quad (1.3)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum N \text{ torques conservados} \quad (1.4)$$

Donde  $L(q_i, \dot{q}_i)$  es llamado el Lagrangiano del sistema.

En este caso, un conveniente conjunto de parámetros es  $\{\theta_1(t); \theta_2(t)\}$  tal como se mostró con anterioridad, de hecho, no es necesario considerar algún otro parámetro para poder conocer la posición exacta del PENDUBOT en cualquier instante de tiempo.

Ahora es necesario calcular la energía potencial y cinética de los dos brazos, para ello, la notación  $\{\theta_1(t); \theta_2(t)\}$  será sustituida por la notación  $\{\theta_1; \theta_2\}$  [9].

### Energía Cinética del Brazo Principal

El Brazo Principal únicamente genera un movimiento de rotación alrededor de un eje, por lo tanto, se puede calcular la energía cinética de la siguiente manera:

$$E_{k1} = \frac{1}{2} J_1 \dot{\theta}_1^2 \quad (1.5)$$

Donde  $\frac{d}{dt}\theta_1 = \dot{\theta}_1$  es la velocidad angular del Brazo Principal [3].

### Energía Cinética del Brazo Secundario

El movimiento del Brazo Secundario está compuesto por un movimiento de tipo rotatorio y un movimiento de tipo traslacional. La Ecuación (1.6) muestra la relación de estos dos movimientos.

$$E_{k2} = \frac{1}{2} J_2 \dot{\theta}_2^2 + \frac{1}{2} M_2 \left| \vec{V}_{G2} \right|^2 \quad (1.6)$$

Donde  $\frac{d}{dt}\theta_2 = \dot{\theta}_2$  es la velocidad angular del Brazo Secundario y  $V_{G2}$  es la velocidad del centro de gravedad de este. Esto puede ser calculado usando las leyes clásicas de la Cinética y la geometría del sistema [3].

$$\vec{V}_{G2} = \frac{d\vec{OG}_2}{dt} \quad (1.7)$$

Donde el centro de gravedad del Brazo secundario está dado por:

$$\vec{OG}_2 = [l_1 \cos(\theta_1) + l_{G2} \cos(\theta_2)] \vec{x} + [l_1 \sin(\theta_1) + l_{G2} \sin(\theta_2)] \vec{y}$$

Esto nos guía al siguiente resultado:

$$\vec{V}_{G2} = [-\dot{\theta}_1 l_1 \sin(\theta_1) - \dot{\theta}_2 l_{G2} \sin(\theta_2)] \vec{x} + [\dot{\theta}_1 l_1 \cos(\theta_1) + \dot{\theta}_2 l_{G2} \cos(\theta_2)] \vec{y} \quad (1.8)$$

Así pues,

$$\left| \vec{V}_{G2} \right|^2 = [-\dot{\theta}_1 l_1 \sin(\theta_1) - \dot{\theta}_2 l_{G2} \sin(\theta_2)]^2 + [\dot{\theta}_1 l_1 \cos(\theta_1) + \dot{\theta}_2 l_{G2} \cos(\theta_2)]^2$$

Desarrollando los binomios cuadrados tenemos:

$$\begin{aligned} \left| \vec{V}_{G2} \right|^2 &= \dot{\theta}_1^2 l_1^2 \sin^2(\theta_1) + \dot{\theta}_2^2 l_{G2}^2 \sin^2(\theta_2) + 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_{G2} \sin(\theta_1) \sin(\theta_2) \\ &\quad + \dot{\theta}_1^2 l_1^2 \cos^2(\theta_1) + \dot{\theta}_2^2 l_{G2}^2 \cos^2(\theta_2) + 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_{G2} \cos(\theta_1) \cos(\theta_2) \\ \left| \vec{V}_{G2} \right|^2 &= 2\dot{\theta}_1^2 l_1^2 + 2\dot{\theta}_2^2 l_{G2}^2 + 4\dot{\theta}_1 \dot{\theta}_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) \\ \left| \vec{V}_{G2} \right|^2 &= \dot{\theta}_1^2 l_1^2 + \dot{\theta}_2^2 l_{G2}^2 + 2\dot{\theta}_1 \dot{\theta}_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) \end{aligned} \quad (1.9)$$

Finalmente, la energía cinética del Brazo Secundario es;

$$E_{k2} = \frac{1}{2} J_2 \dot{\theta}_2^2 + \frac{1}{2} M_2 \left( \dot{\theta}_1^2 l_1^2 + \dot{\theta}_2^2 l_{G2}^2 + 2l_1 l_{G2} \cos(\theta_1 - \theta_2) \dot{\theta}_1 \dot{\theta}_2 \right) \quad (1.10)$$

### Energía Cinética Total del Sistema

Ya que hemos calculado la energía cinética, tanto del Brazo Principal como del Brazo Secundario, se calcula ahora la energía cinética total del sistema, esto se lo-

gra sumando la energía cinética de cada brazo, tal como se muestra en la Ecuación (1.11)[16].

$$E_K = E_{k1} + E_{k2}$$

$$E_K = \frac{1}{2}J_1 \dot{\theta}_1^2 + \frac{1}{2}J_2 \dot{\theta}_2^2 + \frac{1}{2}M_2 \left( \dot{\theta}_1^2 l_1^2 + \dot{\theta}_2^2 l_{G2}^2 + 2l_1 l_{G2} \cos(\theta_1 - \theta_2) \dot{\theta}_1 \dot{\theta}_2 \right) \quad (1.11)$$

### Energía Potencial del Brazo Principal

La energía potencial del Brazo Principal únicamente proviene de la acción de la gravedad sobre este. Esto puede calcularse fácilmente usando la geometría del sistema:

$$E_{p1} = M_1 l_{G1} g \sin(\theta_1) \quad (1.12)$$

Donde  $g = 9.81 \text{ m/s}^2$  es la constante de gravedad.

### Energía Potencial del Brazo Secundario

La energía potencial del Brazo Secundario únicamente proviene de la acción de la gravedad sobre este, sin dejar de considerar la longitud del Brazo Principal, tal como se describe:

$$E_{p2} = M_2 g \left( l_1 \sin(\theta_1) + l_{G2} \sin(\theta_2) \right) \quad (1.13)$$

Donde  $g = 9.81 \text{ m/s}^2$  es la constante de gravedad.

### Energía Potencial Total del Sistema

La energía potencial total del sistema se puede calcular fácilmente, sumando la energía potencial del Brazo Principal y la energía potencial del Brazo secundario [16]:

$$E_P = E_{p1} + E_{p2}$$

$$E_P = M_1 l_{G1} g \sin(\theta_1) + M_2 g \left( l_1 \sin(\theta_1) + l_{G2} \sin(\theta_2) \right) \quad (1.14)$$

### Derivación de las Ecuaciones

Ya que se ha calculado tanto la energía potencial y la energía cinética del sistema, es posible hacer el calculo del Lagrangiano, tal como se muestra:

$$L(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = E_K - E_P \quad (1.15)$$

$$\begin{aligned} L(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) &= \dot{\theta}_1^2 \left[ \frac{1}{2} J_1 + \frac{1}{2} M_2 l_1^2 \right] \\ &+ \dot{\theta}_2^2 \left[ \frac{1}{2} J_2 + \frac{1}{2} M_2 l_{G2}^2 \right] \\ &+ \dot{\theta}_1 \dot{\theta}_2 M_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) \\ &- (M_1 l_{G1} + M_2 l_1) g \sin(\theta_1) - M_2 g l_{G2} \sin(\theta_2) \end{aligned}$$



A continuación todos los cálculos necesarios se hacen con estrictos pasos:

- Derivación parcial del Lagrangiano respecto a la velocidad angular.

$$\frac{\partial L}{\partial \dot{\theta}_1} = \dot{\theta}_1 [J_1 + M_2 l_1^2] + M_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) \dot{\theta}_2 = A(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \quad (1.16)$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = \dot{\theta}_2 [J_2 + l_{G2}^2 M_2] + M_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) \dot{\theta}_1 = B(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \quad (1.17)$$

- Derivación parcial de las ecuaciones previas respecto al tiempo.

$$\frac{d}{dt} A(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{\partial A}{\partial \theta_1} \dot{\theta}_1 + \frac{\partial A}{\partial \theta_2} \dot{\theta}_2 + \frac{\partial A}{\partial \dot{\theta}_1} \ddot{\theta}_1 + \frac{\partial A}{\partial \dot{\theta}_2} \ddot{\theta}_2$$

$$\begin{aligned} \frac{d}{dt} A(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) &= -M_2 l_1 l_{G2} \sin(\theta_1 - \theta_2) \dot{\theta}_1 \dot{\theta}_2 + M_2 l_1 l_{G2} \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ &\quad - M_2 l_1 l_{G2} \sin(\theta_1 - \theta_2) \dot{\theta}_1 \dot{\theta}_2 + M_2 l_1 l_{G2} \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \end{aligned} \quad (1.18)$$

$$\frac{d}{dt} B(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{\partial B}{\partial \theta_1} \dot{\theta}_1 + \frac{\partial B}{\partial \theta_2} \dot{\theta}_2 + \frac{\partial B}{\partial \dot{\theta}_1} \ddot{\theta}_1 + \frac{\partial B}{\partial \dot{\theta}_2} \ddot{\theta}_2$$

$$\begin{aligned} \frac{d}{dt} B(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) &= -\dot{\theta}_1^2 M_2 l_1 l_{G2} \sin(\theta_1 - \theta_2) + \dot{\theta}_1 \dot{\theta}_2 M_2 l_1 l_{G2} \sin(\theta_1 - \theta_2) \\ &\quad + \ddot{\theta}_1 M_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) + \ddot{\theta}_2 (J_2 + M_2 l_{G2}^2) \end{aligned} \quad (1.19)$$

- Derivación parcial del Lagrangiano respecto a los ángulos.

$$\frac{\partial L}{\partial \theta_1} = -M_2 l_1 l_{G2} \sin(\theta_1 - \theta_2) \dot{\theta}_1 \dot{\theta}_2 - [l_{G1} M_1 + M_2 l_1] g \cos(\theta_1) \quad (1.20)$$

$$\frac{\partial L}{\partial \theta_2} = M_2 l_1 l_{G2} \sin(\theta_1 - \theta_2) \dot{\theta}_1 \dot{\theta}_2 - M_2 l_{G2} g \cos(\theta_2) \quad (1.21)$$

Las relaciones finales del modelo pueden ser entonces obtenidas, donde  $\tau$  es la entrada de torque del sistema:

$$\tau = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} \quad (1.22)$$

$$0 = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} \quad (1.23)$$

Substituyendo tendremos las ecuaciones finales:

$$\begin{aligned} \tau = & \ddot{\theta}_1 (J_1 + M_2 l_1^2) + \ddot{\theta}_2 M_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) + M_2 l_1 l_{G2} \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \\ & + (M_1 l_{G1} + M_2 l_1) g \cos(\theta_1) \end{aligned} \quad (1.24)$$

$$\begin{aligned} 0 = & \ddot{\theta}_1 M_2 l_1 l_{G2} \cos(\theta_1 - \theta_2) + \ddot{\theta}_2 (J_2 + M_2 l_{G2}^2) - M_2 l_1 l_{G2} \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \\ & + M_2 g l_{G2} \cos(\theta_2) \end{aligned} \quad (1.25)$$

### 1.3.2 Ecuaciones de Estado

Las ecuaciones previas son las relaciones generales que describen el modelo. Para simplificarlas, nuevas constantes deben de ser introducidas, tal como se muestra a continuación:

$$\tau = A_1 \ddot{\theta}_1 + A_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + A_3 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + A_4 \cos(\theta_1) \quad (1.26)$$

$$0 = B_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + B_2 \ddot{\theta}_2 - B_3 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + B_4 \cos(\theta_2) \quad (1.27)$$

Donde las siguientes constantes han sido introducidas:

$$\left. \begin{array}{l} A_1 = J_1 + M_2 l_1^2 \\ A_2 = M_2 l_1 l_{G2} \\ A_3 = M_2 l_1 l_{G2} \\ A_4 = (M_1 l_{G1} + M_2 l_1) g \end{array} \right\} \text{Reducción de constantes} \left\{ \begin{array}{l} B_1 = M_2 l_1 l_{G2} \\ B_2 = J_2 + M_2 l_{G2}^2 \\ B_3 = M_2 l_1 l_{G2} \\ B_4 = M_2 g l_{G2} \end{array} \right.$$

Las dos variables  $\ddot{\theta}_1$  y  $\ddot{\theta}_2$  deben de ser desacopladas. Con base a la Ecuación (1.27),  $\ddot{\theta}_2$  puede ser expresada como una función de todas las otras variables:

$$\ddot{\theta}_2 = \frac{1}{B_2} \left[ -B_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + B_3 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - B_4 \cos(\theta_2) \right] \quad (1.28)$$

Entonces, esta relación puede ser puesta dentro de la Ecuación (1.26):

$$\begin{aligned} \tau = \ddot{\theta}_1 A_1 + \frac{A_2}{B_2} \cos(\theta_1 - \theta_2) \left[ -B_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + B_3 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - B_4 \cos(\theta_2) \right] \\ + A_3 \sin(\theta_1 - \theta_2) \dot{\theta}_2^2 + A_4 \cos(\theta_1) \end{aligned} \quad (1.29)$$

Una primer relación involucra unicamente a  $\ddot{\theta}_1$  y no a  $\ddot{\theta}_2$ :

$$\boxed{\ddot{\theta}_1 = \frac{1}{\frac{B_1 A_2}{B_2} \cos^2(\theta_1 - \theta_2) - A_1} \left[ \frac{A_2 B_3}{2 B_2} \sin(2(\theta_1 - \theta_2)) \dot{\theta}_1^2 + A_3 \sin(\theta_1 - \theta_2) \dot{\theta}_2^2 - A_2 \frac{B_4}{B_2} \cos(\theta_2) \cos(\theta_1 - \theta_2) + A_4 \cos(\theta_1) - \tau \right]} \quad (1.30)$$

Por un camino similar se puede obtener  $\ddot{\theta}_2$  que no involucre a  $\ddot{\theta}_1$ , partiendo de la Ecuación (1.26),  $\ddot{\theta}_1$  puede ser expresada como una función de todas las otras variables:

$$\ddot{\theta}_1 = \frac{1}{A_1} \left[ -A_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - A_3 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - A_4 \cos(\theta_1) + \tau \right] \quad (1.31)$$

Entonces, esta relación puede ser puesta dentro de la Ecuación (1.27):

$$0 = \frac{B_1}{A_1} \cos(\theta_1 - \theta_2) \left[ -\ddot{\theta}_2 A_2 \cos(\theta_1 - \theta_2) - A_3 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - A_4 \cos(\theta_1) + \tau \right] \\ B_2 \ddot{\theta}_2 - B_3 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + B_4 \cos(\theta_2) \quad (1.32)$$

Una segunda relación involucra unicamente a  $\ddot{\theta}_2$  y no a  $\ddot{\theta}_1$ :

$$\ddot{\theta}_2 = \frac{1}{\frac{B_1 A_2}{A_1} \cos^2(\theta_1 - \theta_2) - B_2} \left[ -B_3 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - B_1 \frac{A_3}{2A_1} \sin(2(\theta_1 - \theta_2)) \dot{\theta}_2^2 \right. \\ \left. - B_1 \frac{A_4}{A_1} \cos(\theta_1) \cos(\theta_1 - \theta_2) + B_4 \cos(\theta_2) + \frac{B_1}{A_1} \cos(\theta_1 - \theta_2) \tau \right] \quad (1.33)$$

Por lo tanto, el sistema ahora está en la forma de ecuaciones de estados. Ya que  $X = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T$  denota el vector de estados que contiene las cuatro variables de estado, la siguiente relación es obtenida:

$$\dot{X} = F(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) + G(\tau) \quad (1.34)$$

Donde:

$$F(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \\ f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \end{bmatrix} \quad (1.35)$$

$$G(\tau) = \begin{bmatrix} 0 \\ 0 \\ f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \\ f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \end{bmatrix} \quad (1.36)$$

$f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  denota a la Ecuación (1.30) y  $f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  denota a la Ecuación (1.33).

### 1.3.3 Validación del Modelo: comparado con el sistema real

Por medio de la función de adquisición de datos de LabVIEW, es posible observar el comportamiento real del sistema. Por medio de Matlab podremos observar el comportamiento teórico de nuestro sistema ya modelado ver Apéndice B, para esta validación se han agregados los parámetros de fricción de cada brazo, los cuales han sido puestos proporcionales a la velocidad de los ángulos.

La Figura 1.2 muestra el modelo que se realizó en Matlab con base a las Ecuaciones (1.30) y (1.33).

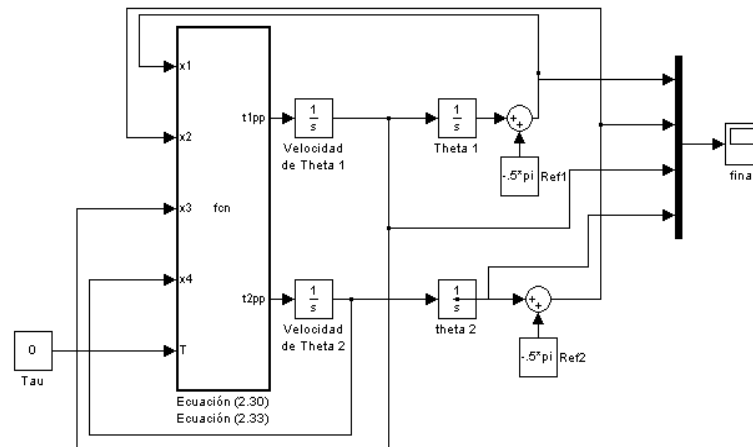


Figura 1.2: Modelo en Matlab

Para la validación se realizaron tres pruebas de manera independiente:

1. Caída libre del Brazo Principal sin el Brazo Secundario.
2. Caída libre del Brazo Secundario con el Brazo Principal fijo.
3. Caída libre del Brazo secundario con el Brazo Principal.

**Prueba 1:**

En la primer prueba se retira el Brazo Secundario y se deja caer el Brazo Principal desde la posición más alta, esto es  $\frac{\pi}{2}$ . El movimiento estudiado es la caída libre del Brazo Principal, tal como se ilustra en la Figura 1.3 y en la Figura 1.4 se ilustra el resultado de la simulación en Matlab.

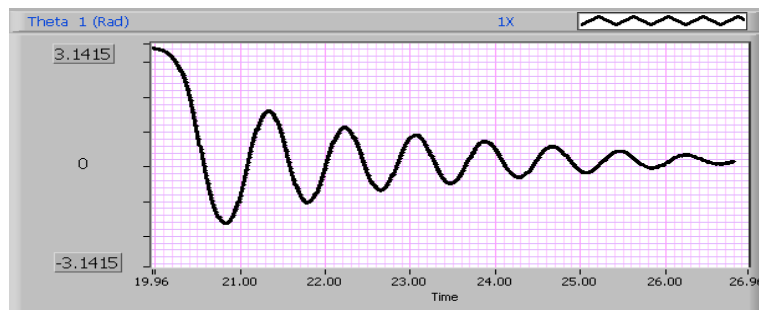


Figura 1.3: Theta 1 con Labview

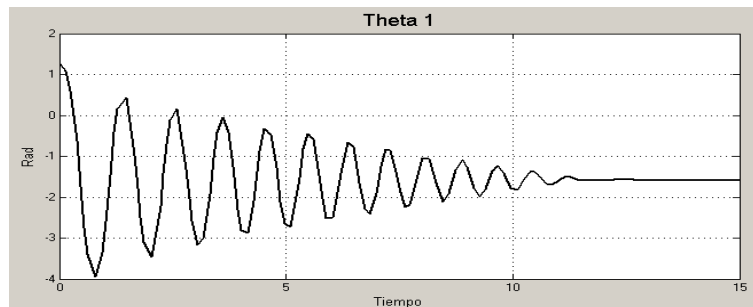


Figura 1.4: Simulación de Theta 1 en Matlab

**Prueba 2**

En la segunda prueba se deja fijo el Brazo Principal en la posición de  $-\frac{\pi}{2}$  y se deja en caída libre el Brazo Secundario desde la posición de  $\frac{\pi}{2}$ . El movimiento estudiado es la caída libre del Brazo Secundario, sin intervención del Brazo Principal, el resultado real de esta prueba, se observa en la Figura 1.5 y el resultado de la simulación se observa en la Figura 1.6.

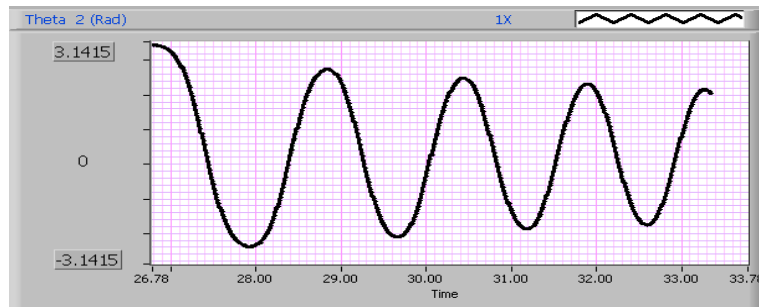


Figura 1.5: Theta 2 con Labview

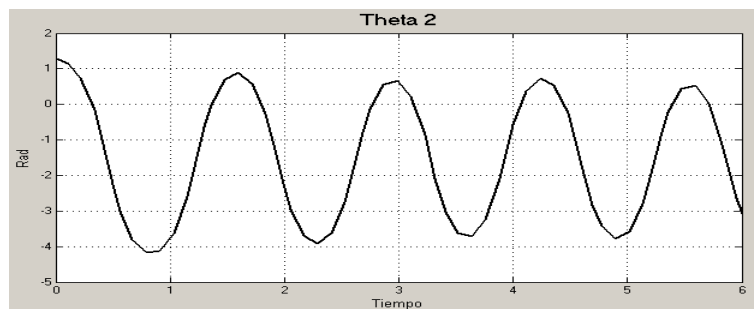


Figura 1.6: Simulación de Theta 2 en Matlab



**Prueba 3**

En la tercera prueba se deja el Brazo Principal en la posición de  $-\frac{\pi}{2}$  y se deja en caída libre el Brazo Secundario desde la posición de  $\frac{\pi}{2}$ . El movimiento estudiado es la relación del Brazo Principal cuando el Brazo Secundario está en caída libre, el resultado real de esta prueba se observa en la Figura 1.7 y el resultado de la simulación se observa en la Figura 1.8.

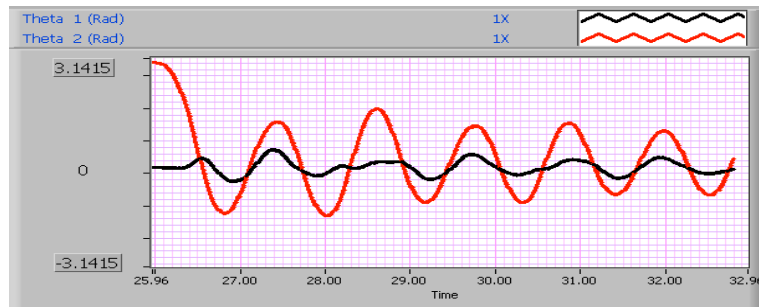


Figura 1.7: Theta 1 y Theta 2 con Labview

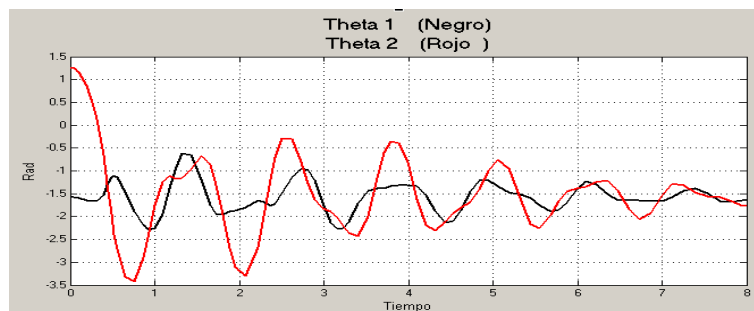


Figura 1.8: Simulación de Theta 2 y Theta 2 en Matlab

Con base a las tres pruebas realizadas se observa la similitud entre el sistema real y el modelo calculado, el sistema real oscila a 1.1 Hz, mientras que el modelo matemático oscila a 1.2 Hz con esta relación tenemos un error de 9.09%. Este porcentaje es suficiente para validar el modelo, ya que existen otros parámetros que no se incluyen, por ejemplo el cable que conecta al encoder del Brazo Secundario, también la fricción se consideró proporcional a la velocidad de los ángulos y no la real desarrollada por el sistema.

Con estos resultados se puede considerar aceptable el modelo matemático.

## 1.4 Linealización

### 1.4.1 Aproximación Teórica

El modelo previo no es directamente utilizable, para poder ocuparlo es necesario linealizarlo con el fin de controlar dicho sistema por medio de un método sencillo.

Sí  $[\theta_{1eq}, \theta_{2eq}, \dot{\theta}_{1eq}, \dot{\theta}_{2eq}]^T$  denota los valores de las variables de estado en un punto de equilibrio, la linealización alrededor de este punto se obtendrá usando la expansión de Taylor[17]:

$$\dot{X} = \dot{X}_{eq} + A.\Delta X + B.\Delta U \quad (1.37)$$

Donde  $A$ , es el jacobiano de la matriz  $F$ , definida en la Ecuación (1.35) y  $B$ , es el jacobiano de la matriz  $G$ , definida en la Ecuación (1.36).

$A$  y  $B$  pueden ser calculadas como se muestra a continuación:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \dot{\theta}_1} & \frac{\partial f_1}{\partial \dot{\theta}_2} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \frac{\partial f_2}{\partial \dot{\theta}_1} & \frac{\partial f_2}{\partial \dot{\theta}_2} \end{bmatrix} \begin{bmatrix} \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2 \end{bmatrix}^T = \begin{bmatrix} \theta_{1eq}, \theta_{2eq}, \dot{\theta}_{1eq}, \dot{\theta}_{2eq} \end{bmatrix}^T \quad (1.38)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial f_1}{\partial \tau} \\ \frac{\partial f_2}{\partial \tau} \end{bmatrix} \quad (1.39)$$

Finalmente, la relación general que se usara en este proyecto es:

$$\Delta \dot{X} = A \Delta X + B \Delta U \quad (1.40)$$

## 1.4.2 Cálculos

### Cálculo Matemático

De acuerdo a la expresión  $f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  que denota a la Ecuación (1.30), podemos obtener la siguiente ecuación:

$$\ddot{\theta}_1 = \frac{A_2 B_3 \sin(2(\theta_1 - \theta_2)) \dot{\theta}_1^2}{2B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} + \frac{A_3 B_2 \sin(\theta_1 - \theta_2) \dot{\theta}_2^2}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} - \frac{B_4 \cos \theta_2 \cos(\theta_1 - \theta_2)}{B_1 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} - \frac{A_4 B_2 \cos \theta_1}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} - \frac{B_2 \tau}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \quad (1.41)$$

Partiendo de la Ecuación (1.41), se calculan las derivadas parciales respecto a  $\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$  que corresponde a las variables de estado, de esta manera se puede obtener los términos de la matriz  $A$  y  $B$ . Los cálculos se hacen a continuación:

- Derivación parcial de  $f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  respecto a la velocidad angular  $\dot{\theta}_1$ .

$$\frac{\partial f_1}{\partial \dot{\theta}_1} = \frac{\partial}{\partial \dot{\theta}_1} \left( \frac{A_2 B_3 \sin(2(\theta_1 - \theta_2)) \dot{\theta}_1^2}{2B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \right) + 0 \quad (1.42)$$

$$\frac{\partial f_1}{\partial \dot{\theta}_1} = \frac{A_2 B_3 \sin(2(\theta_1 - \theta_2)) \dot{\theta}_1}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \quad (1.43)$$

- Derivación parcial de  $f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  respecto a la velocidad angular  $\dot{\theta}_2$ .

$$\frac{\partial f_1}{\partial \dot{\theta}_2} = \frac{\partial}{\partial \dot{\theta}_2} \left( \frac{A_3 B_2 \sin(\theta_1 - \theta_2) \dot{\theta}_2^2}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \right) + 0 \quad (1.44)$$

$$\frac{\partial f_1}{\partial \dot{\theta}_2} = \frac{2A_3 B_2 \sin(\theta_1 - \theta_2) \dot{\theta}_2}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \quad (1.45)$$

- Derivación parcial de  $f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  respecto a el torque de entrada  $\tau$ .

$$\frac{\partial f_1}{\partial \tau} = \frac{\partial}{\partial \tau} \left( \frac{B_2 \tau}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \right) + 0 \quad (1.46)$$

$$\frac{\partial f_1}{\partial \tau} = \frac{B_2}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - \frac{A_1 B_2}{A_2 B_1}} \quad (1.47)$$

De acuerdo a la expresión  $f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  que denota a la Ecuación (1.33), podemos obtener la siguiente ecuación:

$$\begin{aligned} \ddot{\theta}_2 = & -\frac{A_1 B_3 \sin(\theta_1 - \theta_2) \dot{\theta}_1^2}{A_2 B_1 \cos^2(\theta_1 - \theta_2) - A_1 B_2} - \frac{A_3 B_1 \sin(2(\theta_1 - \theta_2)) \dot{\theta}_2^2}{A_1 A_2 B_1 \cos^2(\theta_1 - \theta_2) - A_1^2 B_2} - \frac{A_4 B_1 \cos(\theta_1) \cos(\theta_1 - \theta_2)}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - A_1 B_2} \\ & + \frac{A_1 B_4 \cos(\theta_2)}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - A_1 B_2} + \frac{B_1 \cos(\theta_1 - \theta_2) \tau}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - A_1 B_2} \end{aligned} \quad (1.48)$$

Partiendo de la Ecuación (1.48), se calculan las derivadas parciales respecto a  $\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$  que corresponde a las variables de estado, de esta manera se puede obtener los términos de la matriz  $A$  y  $B$ . Los cálculos se hacen a continuación:

- Derivación parcial de  $f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  respecto a la velocidad angular  $\dot{\theta}_1$ .

$$\frac{\partial f_2}{\partial \dot{\theta}_1} = \frac{\partial}{\partial \dot{\theta}_1} \left( -\frac{A_1 B_3 \sin(\theta_1 - \theta_2) \dot{\theta}_1^2}{A_2 B_1 \cos^2(\theta_1 - \theta_2) - A_1 B_2} \right) + 0 \quad (1.49)$$

$$\frac{\partial f_2}{\partial \dot{\theta}_1} = -\frac{2\dot{\theta}_1 A_1 B_3 \sin(\theta_1 - \theta_2)}{A_2 B_1 \cos^2(\theta_1 - \theta_2) - A_1 B_2} \quad (1.50)$$

- Derivación parcial de  $f_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  respecto a la velocidad angular  $\dot{\theta}_2$ .

$$\frac{\partial f_2}{\partial \dot{\theta}_2} = \frac{\partial}{\partial \dot{\theta}_2} \left( -\frac{A_3 B_1 \sin(2(\theta_1 - \theta_2)) \dot{\theta}_2^2}{A_1 A_2 B_1 \cos^2(\theta_1 - \theta_2) - A_1^2 B_2} \right) + 0 \quad (1.51)$$

$$\frac{\partial f_2}{\partial \dot{\theta}_2} = -\frac{2\dot{\theta}_2 A_3 B_1 \sin(2(\theta_1 - \theta_2))}{A_1 A_2 B_1 \cos^2(\theta_1 - \theta_2) - A_1^2 B_2} \quad (1.52)$$

- Derivación parcial de  $f_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$  respecto a el torque de entrada  $\tau$ .

$$\frac{\partial f_2}{\partial \tau} = \frac{\partial}{\partial \tau} \left( \frac{B_1 \cos(\theta_1 - \theta_2) \tau}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - A_1 B_2} \right) + 0 \quad (1.53)$$

$$\frac{\partial f_2}{\partial \tau} = \frac{B_1 \cos(\theta_1 - \theta_2)}{B_1 A_2 \cos^2(\theta_1 - \theta_2) - A_1 B_2} \quad (1.54)$$

Ya que se han calculado las respectivas derivadas parciales, es necesario conocer tanto las constantes de las ecuaciones como el valor de los estados en el punto de equilibrio. En la Tabla 1.4 se muestran los datos que se sustituyen en las ecuaciones obtenidas anteriormente.

Para este proyecto se considera un punto de equilibrio, cuando el Brazo Principal está perpendicular al eje  $x$  en la posición  $\theta_1 = \frac{3}{2}\pi$  y el Brazo Secundario también está perpendicular al eje  $x$  en la posición  $\theta_2 = \frac{1}{2}\pi$ , como se considera en un punto de equilibrio, las velocidades angulares tanto  $\dot{\theta}_1$  como  $\dot{\theta}_2$  son iguales a cero, esto se ilustra en la Figura 1.9.

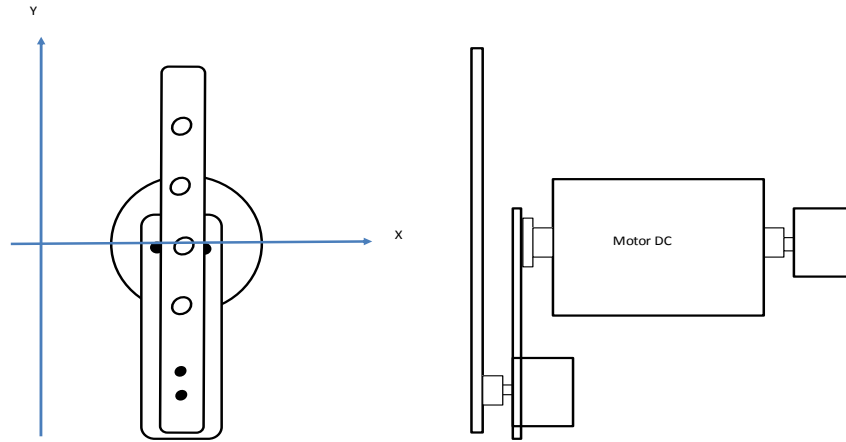


Figura 1.9: Punto de Equilibrio.

**Punto de Equilibrio**

Punto de Linealización		
Constantes del Sistema		Punto de Equilibrio
$A_1 = 0.01064 \text{ Kg.m}^2$	$B_1 = 0.00351 \text{ Kg.m}^2$	$\theta_1 = \frac{3}{2} \pi$
$A_2 = 0.00351 \text{ Kg.m}^2$	$B_2 = 0.00391 \text{ Kg.m}^2$	$\theta_2 = \frac{1}{2} \pi$
$A_3 = 0.00351 \text{ Kg.m}^2$	$B_3 = 0.00351 \text{ Kg.m}^2$	$\dot{\theta}_1 = 0 \pi/s$
$A_4 = 0.50364 \frac{\text{Kg.m}^2}{s^2}$	$B_4 = 0.17251 \frac{\text{Kg.m}^2}{s^2}$	$\dot{\theta}_2 = 0 \pi/s$

Tabla 1.4: Punto de linealización

Se calcula la matriz A y la matriz B, estas son únicas para esta posición, por medio de las Ecuaciones (1.43), (1.45), (1.50) y (1.52), podemos obtener dichos valores de la matriz A. Para la matriz B, solamente se consideran las Ecuaciones (1.47) y (1.54).

Al sustituir los valores de la Tabla 1.4 en las ecuaciones antes mencionadas, se obtiene lo siguiente:

$$\begin{array}{ccc} \frac{\partial f_1}{\partial \dot{\theta}_1} = 0 & \frac{\partial f_1}{\partial \dot{\theta}_2} = 0 & \frac{\partial f_1}{\partial \tau} = 133.6126 \\ \frac{\partial f_2}{\partial \dot{\theta}_1} = 0 & \frac{\partial f_2}{\partial \dot{\theta}_2} = 0 & \frac{\partial f_2}{\partial \tau} = 120.0350 \end{array}$$

Como las derivadas parciales respecto a  $\theta_1$  y  $\theta_2$ , generan ecuaciones más complejas y laboriosas, es necesario auxiliarnos de algún programa que nos facilite este procedimiento, para ello, MATLAB (MATrix LABoratory) es el programa que ocuparemos para desarrollar dichas ecuaciones[13].

Los términos que faltan de las matrices, serán computadas por medio de MATLAB (MATrix LABoratory), esto es por lo largo y complicado que son estas ecuaciones, en la siguiente sección se completan dichas matrices.

### Cálculo con MATLAB

En esta sección se presenta el algoritmo en MATLAB para obtener el jacobiano de la matriz F definida en la Ecuación (1.35) y el jacobiano de la matriz G definida en la Ecuación (1.36).

El algoritmo se define en 4 pasos:

1. Se introduce la Ecuación (1.30) como **f1** y la Ecuación (1.33) como **f2**.
2. Se derivan parcialmente **f1** y **f2** respecto a  $\theta_1$ ,  $\theta_2$ ,  $\dot{\theta}_1$ ,  $\dot{\theta}_2$  y  $\tau$ .



3. Se introducen las constantes del modelo.

4. se intruducen los valores de los estados en la posición de equilibrio y se obtienen las matrices A y B del modelo.

```
%***** PASO 1 *****
f1 = '(1/(((B1*A2)/B2)*cos(x1-x2)^2-A1))*(((A2*B3)/(2*B2))*sin(2*(x1-x2))*x3^2
      +A3*sin(x1-x2)*x4^2-A2*(B4/B2)*cos(x2)*cos(x1-x2)+A4*cos(x1)-T)';

f2 = '(1/((B1*A2/A1*(cos(x1-x2)^2)-B2))*[(-B3*x3^2*sin(x1-x2))
      -(((A3*B1)/(2*A1))*sin(2*(x1-x2))*x4^2)-(B1*(A4/A1)*cos(x1)*cos(x1-x2))
      +(B4*cos(x2))+(B1/A1)*cos(x1-x2)*T] )';

%***** PASO 2 *****
y1 = diff(f1,'x1');      yy1 = diff(f2,'x1');
y2 = diff(f1,'x2');      yy2 = diff(f2,'x2');
y3 = diff(f1,'x3');      yy3 = diff(f2,'x3');
y4 = diff(f1,'x4');      yy4 = diff(f2,'x4');
u1 = diff(f1,'T');      u2 = diff(f2,'T');

%***** PASO 3 *****
A1 = 0.01064400;      B1 = 0.00351708;
A2 = 0.00351708;      B2 = 0.00391491;
A3 = 0.00351708;      B3 = 0.00351708;
A4 = 0.50364500;      B4 = 0.17251300;

%***** PASO 4 *****
x1 = 3/2*pi;      x3 = 0;      T = 0;
x2 = pi/2;      x4 = 0;
```

A = [0 0 1 0; 0 0 0 1; subs(y1) subs(y2) subs(y3) subs(y4);

subs(yy1) subs(yy2) subs(yy3) subs(yy4)];

B = [0; 0; subs(u1); subs(u2)];

El programa genera las siguientes matrices:

Punto de Equilibrio.

$$A = \begin{bmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ -67.2933 & 20.7076 & 0 & 0 \\ -60.4550 & 62.6689 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 133.6126 \\ 120.0350 \end{bmatrix}$$

## Capítulo 2

# Diseño del Control por Retroalimentación de Estados

Para diseñar el control del PENDUBOT se consideran dos problemas principales, el primero es el referente a posicionar el Brazo Secundario en  $\pi/2$  desde la posición de equilibrio natural, esto es,  $-\pi/2$  tanto para el Brazo Principal como para el Brazo Secundario. El segundo problema y quizás el más importante es equilibrar el Brazo Secundario en la posición  $\pi/2$ .

### 2.1 Control de Subida

El control de Subida se ha diseñado en dos pasos:

- Una linealización parcial del modelo
- El diseño de un control Proporcional Derivativo

### 2.1.1 Linealización Parcial

El pendubot consiste en dos brazos enlazados por una unión libre. Uno de estos brazos puede controlar este sistema (el cual tiene dos grados de libertad) por medio de un motor de corriente directa aplicando un torque en el Brazo Principal.

La idea es alimentar el sistema con un torque modificado para cancelar las no-linealidades[5]. Esto se puede conseguir por medio de un bucle interno dentro del sistema y por una nueva entrada de control, llamandola  $v_1$ .

A continuación se muestran los pasos para conseguir una relación lineal entre la entrada de control (el torque  $\tau$ ) y  $\ddot{\theta}_1$ :

Las ecuaciones iniciales son:

$$\tau = \ddot{\theta}_1 A_1 + \ddot{\theta}_2 A_2 \cos(\theta_1 - \theta_2) + A_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + A_4 \cos(\theta_1) \quad (2.1)$$

$$0 = \ddot{\theta}_1 B_1 \cos(\theta_1 - \theta_2) + \ddot{\theta}_2 B_2 - A_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + B_4 \cos(\theta_2) \quad (2.2)$$

De la Ecuación (2.2) la expresión de  $\ddot{\theta}_2$  puede ser obtenida, como se muestra en el capitulo anterior:

$$\ddot{\theta}_2 = \frac{1}{B_2} \left[ \dot{\theta}_1^2 A_2 \sin(\theta_1 - \theta_2) - B_4 \cos(\theta_1 - \theta_2) - \ddot{\theta}_1 A_2 \cos(\theta_1 - \theta_2) \right] \quad (2.3)$$

Una nueva ecuación para el torque puede ser derivada reemplazando  $\ddot{\theta}_2$  en la Ecuación (2.1).

$$\begin{aligned} \tau = & \ddot{\theta}_1 A_1 + A_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + A_4 \cos(\theta_1) + \frac{A_2}{B_2} \cos(\theta_1 - \theta_2) [\theta_1^2 A_2 \sin(\theta_1 - \theta_2) \\ & - B_4 \cos(\theta_1 - \theta_2) - \ddot{\theta}_1 A_2 \cos(\theta_1 - \theta_2)] \end{aligned} \quad (2.4)$$

esta puede ser escrita como se muestra enseguida:

$$\begin{aligned} \tau = & \ddot{\theta}_1 \left[ A_1 - \frac{A_2^2}{B_2} \cos^2(\theta_1 - \theta_2) \right] + \dot{\theta}_2^2 A_2 \sin(\theta_1 - \theta_2) + \dot{\theta}_1^2 \frac{A_2^2}{B_2} \cos(\theta_1 - \theta_2) \sin(\theta_1 - \theta_2) \\ & + A_4 \cos(\theta_1) - B_4 \frac{A_2}{B_2} \cos(\theta_2) \cos(\theta_1 - \theta_2) \end{aligned} \quad (2.5)$$

Gracias a las ecuaciones deducidas anteriormente, una nueva entrada de control puede ser definida, llamada  $v_1$ , usando las dos notaciones siguientes  $\alpha(\theta_i)$  y  $\beta(\theta_i)$ :

$$\begin{aligned} \alpha(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = & \dot{\theta}_2^2 A_2 \sin(\theta_1 - \theta_2) + \dot{\theta}_1^2 \frac{A_2^2}{B_2} \cos(\theta_1 - \theta_2) \sin(\theta_1 - \theta_2) \\ & + A_4 \cos(\theta_1) - B_4 \frac{A_2}{B_2} \cos(\theta_2) \cos(\theta_1 - \theta_2) \quad (2.6) \\ \beta(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = & A_1 - \frac{A_2^2}{B_2} \cos^2(\theta_1 - \theta_2) \end{aligned}$$

$v_1$  es definida por la siguiente relación y está diseñada con el fin de suprimir los términos no-lineales:

$$\tau = v_1 \beta(\theta_i) + \alpha(\theta_i) \quad (2.7)$$

Usando esta retroalimentación para la linealización parcial nos guía a una nueva asignación de ecuaciones de estado:

$$\ddot{\theta}_1 = v_1 \quad (2.8)$$

$$\ddot{\theta}_2 = \frac{1}{B_2} \left[ \theta_1^2 A_2 \sin(\theta_1 - \theta_2) - B_4 \cos(\theta_1 - \theta_2) - v_1 A_2 \cos(\theta_1 - \theta_2) \right] \quad (2.9)$$

### 2.1.2 Control PD

El sistema es ahora lineal con respecto a  $\ddot{\theta}_1$ . Por otra parte, ya que es una relación de segundo orden, un controlador PD puede ser aplicado[8]. La ley de control que se utiliza aquí es:

$$v_1 = K_p(\theta_{1ref} - \theta_1) - K_d\dot{\theta}_1 \quad (2.10)$$

Por último, el sistema de corrección global será de doble bucle, como se muestra en la Figura 2.1. El bucle interior linealiza parcialmente el sistema y el lazo externo se utiliza para controlar este sistema modificado.

Para lograr posicionar el Brazo Secundario en  $\pi/2$ , tenemos que considerar una estrategia que nos permita lograr esto, considerando que una referencia estática no es suficiente desde el Brazo Principal, ya que este comienza en la posición de  $-\pi/2$  y debe de terminar en esta misma posición. El objetivo es subir el Brazo secundario de una manera inteligente que sea capaz de posicionarse en  $\pi/2$  y que al llegar a dicha posición se pueda hacer el cambio al control de equilibrio.

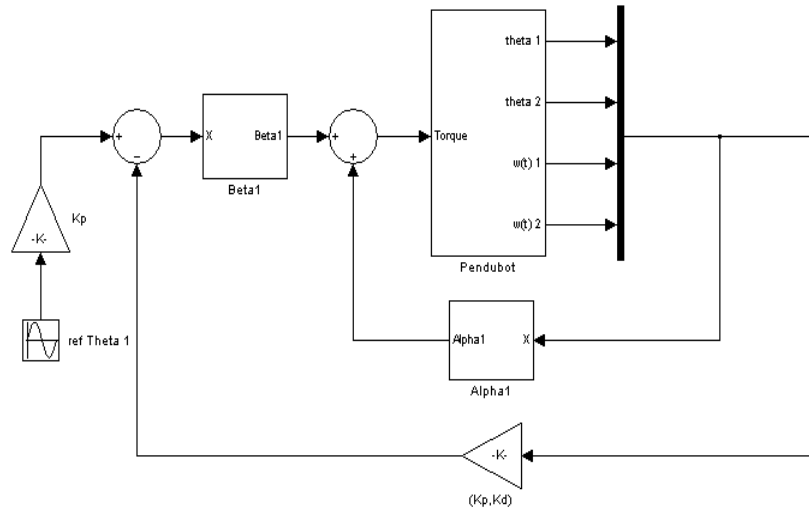


Figura 2.1: Control de Subida

Intuitivamente, se puede deducir fácilmente que el Brazo Principal tiene que almacenar la energía de regulación del Brazo Secundario con el fin de subirlo a la posición  $\pi/2$ . Para solucionar este problema de energías, se considera una señal de referencia para el Brazo Principal que sea capaz de almacenar la energía de regulación necesaria para controlar el Brazo Secundario, esto se puede lograr si consideramos una señal de referencia cosenoidal la cual se tiene que afinar tanto en amplitud como frecuencia.

$$\theta_{1ref} = A \cos(x(t)) \quad (2.11)$$

Donde  $A$  es la amplitud de la señal y  $x(t)$  es la frecuencia natural del sistema, en la sección anterior se encontró la frecuencia natural del sistema, que fue de 1.1 Hz.,

esta frecuencia es la que se considera para la señal de referencia y la magnitud de la amplitud será de 1, de esta manera se reducen variables y únicamente afinamos tanto la ganancia  $K_p$  como la  $K_d$ . Los valores que se implementaron en el proyecto son los que se muestran a continuación:

$$\begin{aligned}
 A &= 1 \\
 x(t) &= 1.1 \text{ Hz.} \\
 K_p &= 50 \\
 K_d &= 1.5
 \end{aligned}$$

La señal del torque del motor de c.d. que se ocupa para el control de subida es:

$$\tau = \left[ K_p \left( A \cos(x(t)) - \theta_1 \right) - K_d \dot{\theta}_1 \right] \beta(\theta_i) + \alpha(\theta_i) \tag{2.12}$$

## 2.2 Control de Equilibrio

Para estabilizar el Brazo Secundario en la posición  $\pi/2$  se considera la técnica de retroalimentación de estados y la técnica de asignación de polos, para obtener los valores de las ganancia de retroalimentación, todos estos cálculos se realizan con MATLAB.

Si consideramos las siguientes ecuaciones de estado de n-dimensiones:

$$\begin{aligned}
 \dot{x} &= Ax + bu \\
 y &= cx
 \end{aligned} \tag{2.13}$$



En esta relación consideramos que el vector  $d = [0]$ , en retroalimentación de estados, la entrada  $u$  está dada por:

$$u = r - Kx = r - [K_1 \quad K_2 \cdots K_n]x = r - \sum_{i=1}^n K_i x_i \quad (2.14)$$

En la Figura 2.2, cada ganancia de retroalimentación  $K_i$  es una constante real, la cual, también es llamada, ganancia constante de retroalimentación de estados negativa, o simplemente, retroalimentación de estados[5].

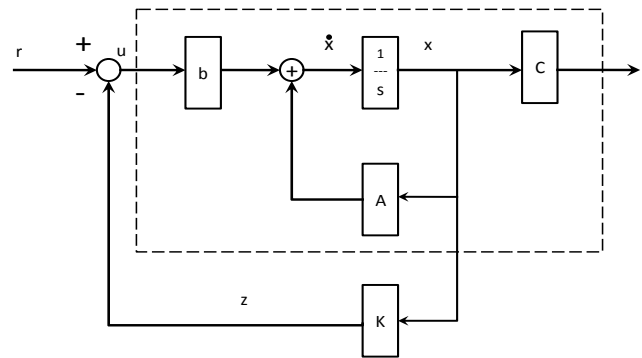


Figura 2.2: Retroalimentación de Estados

La representación en ecuaciones de estado del diagrama de la Figura 2.2 es:

$$\begin{aligned} \dot{x} &= (A - bK)x + br \\ y &= cx \end{aligned} \quad (2.15)$$

De lo anterior se deduce una función para la señal de control que se ocupa para el equilibrio del Brazo secundario en la posición  $\pi/2$ .

$$\tau = \tau_0 - K(x - x_0) \tag{2.16}$$

Donde  $K$ , es el vector  $1 \times 4$ , que contiene las ganancias de retroalimentación, y se calcula por la asignación de polos. El control se implementa de acuerdo a la Ecuación (2.16) y se muestra en la Figura 2.3.

MATLAB cuenta con un comando para poder determinar el vector de ganancias para

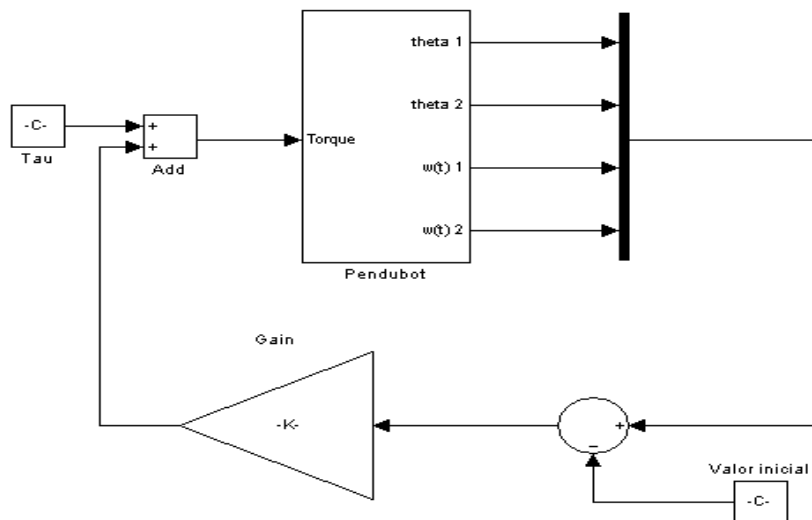


Figura 2.3: Control de Retroalimentación.

la retroalimentación de estados[13], para poder ejecutar esta instrucción es necesario contar con las matrices  $A$  y  $B$ , las cuales fueron obtenidas en la sección anterior, y también es necesaria la ubicación de los polos.

Para obtener el valor de los polos, primero se analizó el comportamiento del sistema, así se pudo dar una primera postulación, esta se fue modificando de acuerdo con las necesidades del control, dichas pruebas dieron como resultado la siguiente asignación final de los polos:

$$p = \begin{bmatrix} -5 + 0.1j & -5 - 0.1j & -5 + 0.1j & -5 - 0.1j \end{bmatrix}$$

El siguiente algoritmo en MATLAB muestra como se obtubieron las ganancias de retroalimentación [13].

```
K=acker(A,B,p);
```

El valor de la ganancia de retroalimentación de los estados es:

$$K = \begin{bmatrix} -0.6099 & 1.8901 & -0.0850 & 0.2612 \end{bmatrix}$$

La Figura 2.4 muestra la simulación del sistema con las ganancias de retroalimentación en MATLAB con un valor inicial de  $\theta_1 = 1 \text{ rad}$ .

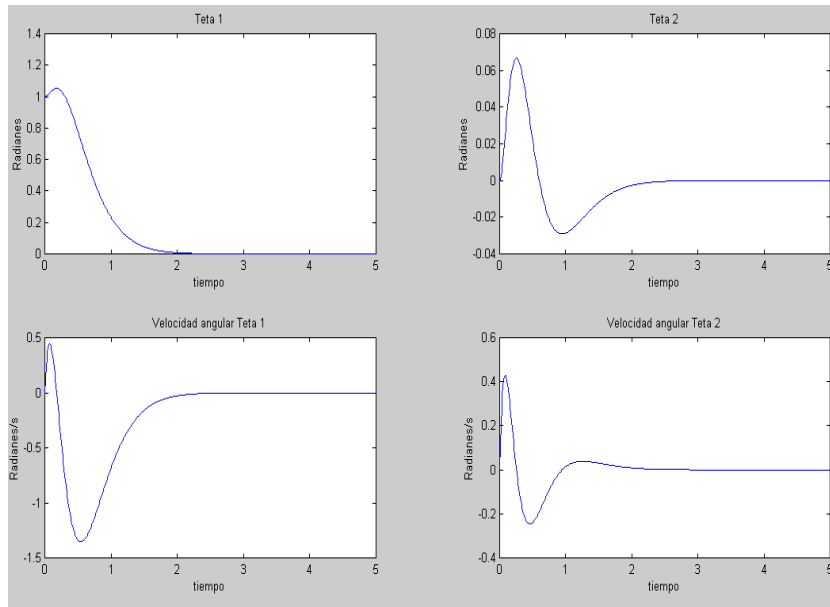


Figura 2.4: Simulación de K en Matlab

# Capítulo 3

## Pruebas y Resultados

### 3.1 Prueba del Control de Subida

Para equilibrar el Brazo Secundario en la posición de  $\pi/2$  rad, Figura 3.1, se diseñó el Control de Subida, este control es capaz de posicionar el Brazo Secundario en  $\pi/2$  y también es capaz de posicionar el Brazo Primario en la posición  $-\pi/2$ , como se describe en el Capítulo 2, Figura 2.1, además se programó en el microcontrolador DSP MC56F8367, Apéndice E. Esta prueba consiste en medir el tiempo que le tarda al motor en posicionar a los dos brazos, en sus posiciones correspondientes, el resultado práctico del control adaptado en el microcontrolador se observa en la Figura 3.2.

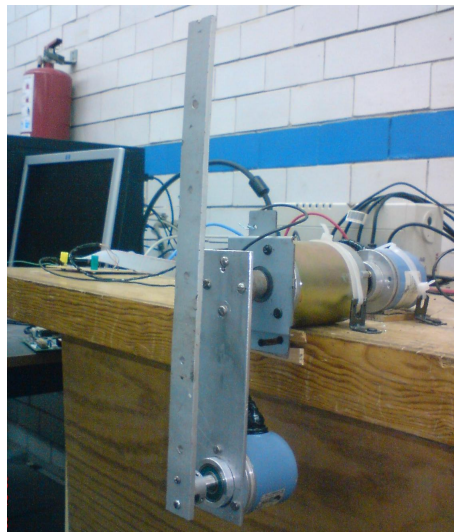


Figura 3.1: Posición de Equilibrio

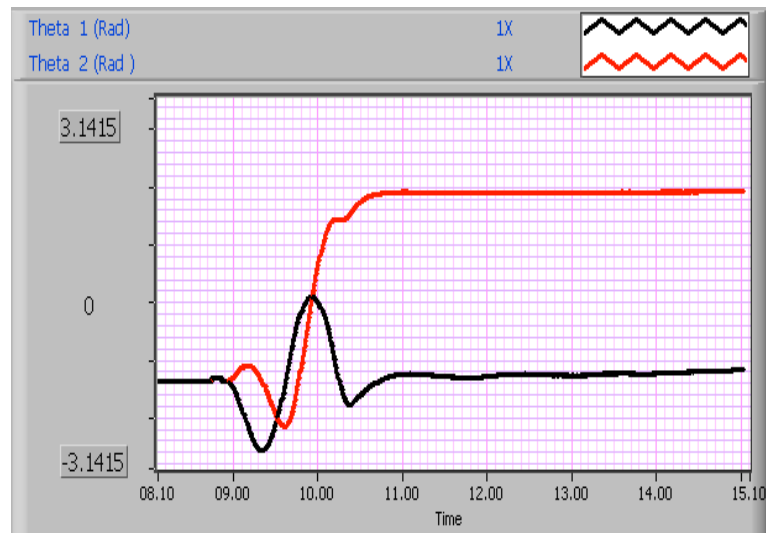


Figura 3.2: Primer Prueba del Control de Subida

Como se puede ver en la gráfica de posiciones de la Figura 3.2, Apéndice B, el posicionamiento de los Brazos se realiza en 2 segundos, en este caso la limitante del tiempo se debe al trabajo que desarrolla el motor. El motor que se ocupa para este proyecto desarrolla un trabajo de  $W = 0.5822 \frac{Kg \cdot m^2}{s^2}$ , esto significa que el Brazo Principal junto con el encoder, pueden ser desplazados un ángulo de  $0.196\pi$  rad, cuando se aplica todo el ciclo de trabajo del PWM. La limitante de potencia del motor genera que la velocidad del control sea menor, pero a pesar de esta limitante, el control se considera aceptable dentro de los márgenes de diseño.

En la Figura 3.3, se observa que los dos Brazos parten de la posición  $-\pi/2$ , la posición del Brazo Principal esta representada por la señal de color negro, la cual se mantiene en esa misma posición después de realizar un movimiento de tipo senoidal, con el objetivo de proporcionar la energía suficiente al Brazo Secundario para que este sea llevado a la posición de  $\pi/2$  (señal roja).

Esta misma prueba se realiza en un instante de tiempo diferente y proporciona el resulta de la Figura 3.3. Al igual que lo observado en la Figura 3.2, los brazos se posicionan en dos segundos.

### 3.1.1 Resultados

Se concluye que el Control de Subida funciona de acuerdo al modelo matemático, además se observó, que conforme el numero de pruebas aumenta el motor y la interfaz de potencia tienden a sufrir un calentamiento, esto se debe al trabajo que se

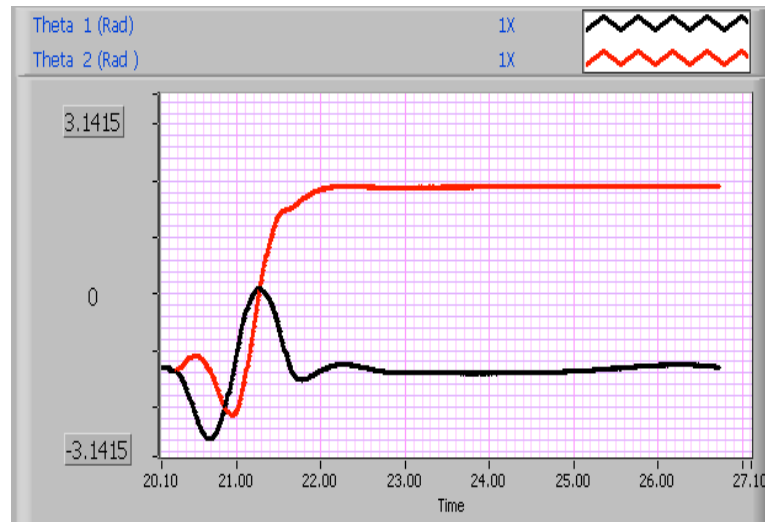


Figura 3.3: Segunda Prueba del Control de Subida

realiza por el cambio de dirección, es por eso que el Brazo Secundario tiende a no llegar a la posición de  $\pi/2$ , para solucionar esto, se modifica el valor de la ganancia  $K_d$ , del Control de Subida, dando como resultado un aumento de la inercia del Brazo Secundario, logrando con esto, solucionar el posicionamiento del Brazo Secundario debido al efecto del calentamiento[20].

## 3.2 Prueba del Control de Equilibrio

Una vez que el Control de Subida se ejecuta, el Brazo Secundario llega a la posición de  $\pi/2$  y el Brazo Principal a la posición  $-\pi/2$ , en ese instante el Control de Equilibrio mantiene a los dos brazos en sus posiciones correspondientes, esto se puede comprobar,



por medio de dos pruebas, la primera consiste en dar una perturbación con una magnitud de fuerza aleatoria a el Brazo Secundario y la segunda consiste en generar una perturbación controlada a el Brazo Principal, en ambas pruebas el Control de Equilibrio debe de compensar ambas perturbaciones.

### 3.2.1 Perturbación en el Brazo Principal

Esta prueba consiste en dar una perturbación controlada y comprobar que el control por retroalimentación de estados esta funcionando correctamente. Como se puede observar en la Figura 3.2 y 3.3, el Control de Subida hace correctamente su función en 2 seg., después de esto, el Control de Equilibrio mantiene a los dos brazos es sus posiciones.

La Figura 3.4 se puede observar la respuesta a una perturbación controlada, esto se debe a que la perturbación al brazo Principal se hace por medio del programa en el microcontrolador DSP MC56F8367, ver apéndice E, la perturbación se logra aplicando todo el voltaje al motor en un tiempo de 50 mseg. en una sola dirección, esto hace que se desplacen ambas posiciones, en un determinado tiempo, después de este desplazamiento el control por retroalimentación de estados vuelve a tomar el control del motor, retomando la posición del Brazo Secundario en poco menos de 2 seg., este tiempo de respuesta es el mismo que se obtuvo en las gráficas de MatLab (Figura 2.4).

Para esta prueba se considera aceptable el control por retroalimentación de estados, ya que no deja caer al brazo Secundario de su posición de  $\pi/2$  rad.

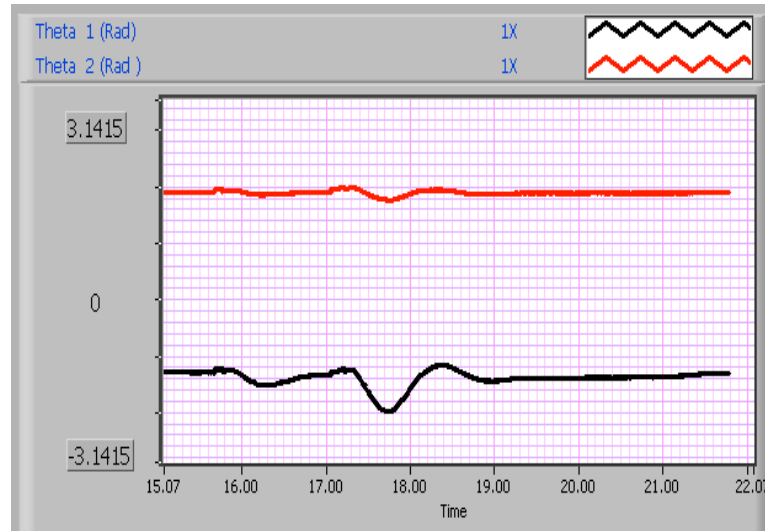


Figura 3.4: Perturbación en el Brazo Primario

### 3.2.2 Perturbación en el Brazo Secundario

Otro tipo de perturbación es puesta a prueba, esta es cuando una fuerza externa no controlada modifica la posición del Brazo Secundario, este tipo de perturbación es más agresiva, ya que el desplazamiento es mayor, y el motor debe de compensar este desplazamiento con un mayor par en un menor tiempo, el inconveniente de esta prueba es que no se puede repetir de la misma forma que la anterior, ya que no se tiene el control de esta perturbación.

Como se observa en la Figura 3.5. el desplazamiento del Brazo Secundario es mayor que en la perturbación anterior y de la misma manera el desplazamiento de  $\theta_1$  es mayor, esto se debe a que el motor debe de compensar con mayor ángulo de inclinación para evitar que el ángulo de  $\theta_2$  salga de la zona linealizada.

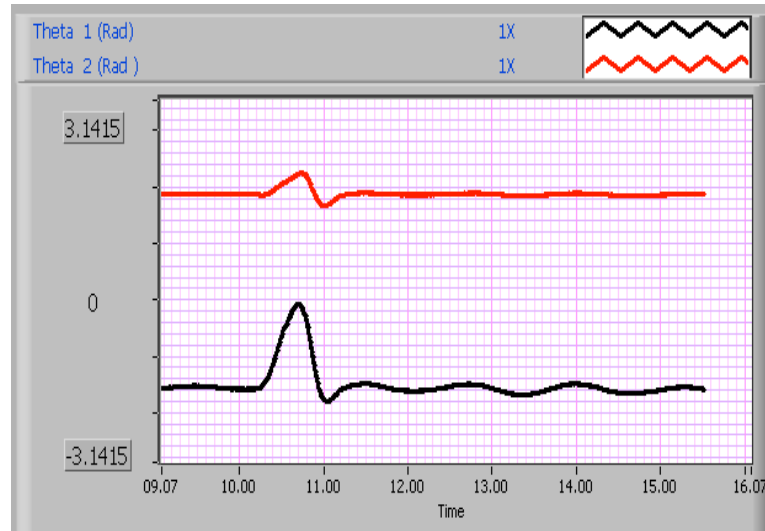


Figura 3.5: Perturbación en el Brazo Secundario

### 3.2.3 Resultado

Con estas dos perturbaciones se considerará aceptable el Control de Equilibrio por retroalimentación de estados, ya que en las dos pruebas realizadas, el resultado es similar a lo simulado en MATLAB, a pesar que la segunda perturbación es más agresiva, el control actúa eficientemente, tomándole solo 1 seg. corregir esta perturbación, pero no siempre se puede compensar a una perturbación, tal y como se muestra en la Figura 3.6, donde se observa que si aplicamos una perturbación controlada de 100 msec. al motor tanto  $\theta_1$  y  $\theta_2$  salen de la zona linealizada, dando como resultado la caída del Brazo Secundario.

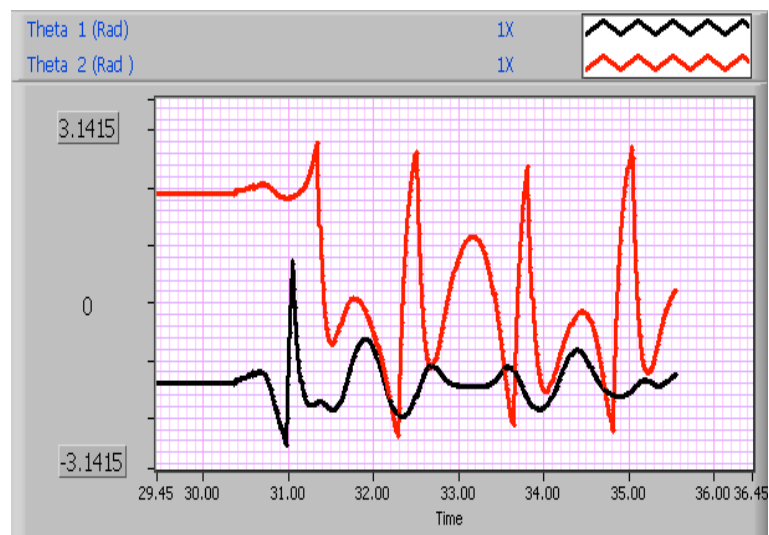


Figura 3.6: Perturbación Grande

# Capítulo 4

## Conclusiones

### 4.1 Conclusiones

De acuerdo con los objetivos planteados, se construyó un sistema PENDUBOT, con una frecuencia natural de 1.1Hz., masa de 0.3635 Kg. en el Brazo Principal y 0.1113 Kg. en el Brazo Secundario, momento de inercia total en el Brazo Principal de  $J_1 = 6.1919 \times 10^{-3} \text{ Kg.m}^2$ , momento de inercia total del Brazo Secundario de  $J_2 = 1.1364 \times 10^{-3} \text{ Kg.m}^2$ , motor de c.d. que genera un trabajo de  $W = 0.5822 \frac{\text{Kg.m}^2}{\text{s}^2}$  y eslabones de aluminio de 0.20 m y 0.35 m respectivamente para el Brazo Principal y el Brazo Secundario, todo el algoritmo de control se implementó en un Microcontrolador Freescale DSP MC56F8367. de 65 Mhz.

1. Se desarrolló la interfaz de potencia para el motor de c.d. a 10 Khz., con una corriente nominal de 20 A. y un voltaje nominal de 30 V. Esta interfaz sufre

- calentamiento, después de un tiempo prolongado de trabajo, debido al cambio de dirección del motor, además, ya que el motor es 20 V. y 5 A., se redujo a 20 V. la interfaz de potencia.
2. Se desarrolló la interfaz para los dos encoders absolutos, WDG 58b-2000-AB-G24-93, que son energizados con 5 V., además cuenta con dos filtros pasa bajos y el algoritmo para los cálculos tanto de las posiciones como las velocidades, en rad y rad/s. Las posiciones sufren un corrimiento, debido a la interferencia electromagnética que genera el motor sobre los encoders y por la vibración del mismo.
  3. La interfaz gráfica está soportada en LabVIEW, la programación cumple la tarea para la cual fue desarrollada. La velocidad de los gráficos, dependen de la tarjeta de video de la PC. donde sea instalado LabVIEW, también en gran parte al diseño del filtro pasa bajos, ya que si se diseña cerca de la frecuencia del PWM, genera vibraciones en los gráficos, pero si se diseña muy lejos de la frecuencia del PWM, genera retardo en estos.
  4. El modelo matemático se obtiene fácilmente por el Método de Euler Lagrange y para la linealización se ocupa la Expansión de Taylor. Los cálculos para las ganancias de retroalimentación de estados, se hacen por medio de MATLAB, de manera rápida y eficiente.
  5. La integración de las interfaces y del sistema mecánico se hace en un microcontrolador DSP MC56F8367, la programación se desarrolla en lenguaje C. El

microcontrolador soporta la programación del prototipo de forma eficiente, pero se puede sustituir con la familia de microcontroladores coldfire, ya que resulta ser más barato e igual de eficiente.

6. Con base a las pruebas realizadas al prototipo, se asegura, que el modelo matemático obtenido es el correcto, además, se comprueba que la plataforma para implementar diferentes tipos de controles es funcional.

Este prototipo está preparado para realizar pruebas de Control Convencional, Robusto, Heurísticos, Autómatas Celulares, etc. Para poder realizar pruebas de control se debe de considerar lo siguiente:

- La plataforma para programar debe de ser en C.
- Las entradas del sistema son los ángulos  $\theta_1$  y  $\theta_2$  en radianes, en el sistema son interpretadas por las variables ContARad y ContBRad respectivamente.
- La salida del sistema que es el voltaje para el Par del motor esta dado por la salida del microcontrolador PWM1 SetRatio16(pwm1) que es un PWM de 16 bits y la dirección del motor la proporciona la salida del Bit1, Bit1.SetVal() y Bit1.ClrVal() uno para cada sentido respectivamente.

Bajo estas condiciones cualquier clase de control puede ser probado en este prototipo, siempre y cuando se consideren las especificaciones mencionadas anteriormente, para poder diseñar el algoritmo de control.

## 4.2 Trabajos Futuros

Como trabajos futuros se puede implementar alguna otra técnica de control, dado que la técnica de control por retroalimentación de estados se implementó en este trabajo.

Se consideran los siguientes trabajos futuros:

1. Diseñar e implementar un control difuso para el sistema PENDUBOT.
2. Diseñar e implementar un control por medio de autómatas celulares para el sistema PENDUBOT.
3. Capacidades y limitaciones de este sistema PENDUBOT.



# Referencias

- [1] H. Arai and S. Tachi. *Position control of a manipulator with passive joints using dynamic coupling*. IEEE Trans., 1991.
- [2] K.J. Astrom and K Furuta. *Swinging Up a pendulum by energy control*. Automatica, 2000.
- [3] Daniel Jerome Block. *Word me-cha-ni-cal De-sign and Con-trol of the Pen-du-bot*. PhD thesis, University of Illinois, Illinois, CA 94305, June 1991. Submitted in partial fulfillment of the requirements for the degree of Master of Science in General Engineering.
- [4] S.A. Bortoff. *Pseudolinearization using Spline Functions with Application to the Acrobot*. University of Illinois, 1992.
- [5] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, New York, 1999.
- [6] James W. Bignell & Robert L. Donovan. *Electrónica Digital*. Compañía Editorial Continental, S.A. de C.V. & Grupo Patria Cultural S.A. de C.V., 2005.

- 
- [7] National Instruments. *LabVIEW 7.1 User Manual*. E.U.A, 2003.
- [8] Benjamin C. Kuo. *Sistemas de Control Automático*. University of Illinois, 1996.
- [9] Isabelle Fantoni & Rogelio Lozano. *Non-Linear Control for Underactuated Mechanical Systems*. Springer Verlag London, 2002.
- [10] R. Lozano, I. Fantoni, and D.J. Block. *Stabilization of the inverted pendulum around its homoclinic orbit*. Systems y Control Letters, 2000.
- [11] L.Praly. *Stabilisation du système pendule-chariot: Approche par assination dénergie*. Personal communication, 1995.
- [12] Alberto Paul Malvino. *Principios de Electrónica*. Mc.Graw Hill, 1992.
- [13] The math works inc. con tutorial de Duane Hanselman y Bruce Littlefield. *MATLAB educación de estudiante*. Prentice Hall, 1996.
- [14] Robert L. Boylestad Louis Nashelsky. *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*. Pearson Educación, 2006.
- [15] A.S. Shiriaev, O. Egeland, and H. Ludvigsen. *Global stabilization of unstable equilibrium point of pendulum*. Tampa, FL, USA, 1998.
- [16] Mark W. Spong. Energy Base Control of the Pendubot. *Transactions on Automatic Control: a publication of IEEE control systems society*, pages 725–730, 2000.

- 
- [17] Antonio Flores T. Linealización de Funciones No Lineales. *Internet*, pages 1–17, 2006.
- [18] MCU Application Design Team. D/A Converter by Using PWM . *Internet*, pages 5–1–5–5, 2001.
- [19] Ronald J. Tocci. *Sistemas Digitales, Principios y Aplicaciones*. Pearson Educación, 1996.
- [20] William H. Roadstrum & Dan H. Wolaver. *Ingeniería Electrica para todos los Ingenieros*. ALFAOMEGA, 1999.

# Apéndice A

## Construcción de la Interfaz de Potencia

Una técnica para controlar el par y la velocidad de los motores de corriente continua es por medio de una modulación por ancho de pulsos,(PWM, siglas del inglés *Pulse Width Modulation*), con esto, se puede controlar la velocidad del giro, manteniendo el par motor constante y no supone un desaprovechamiento de la energía eléctrica[20].

Para controlar el motor del sistema PENDUBOT ocuparemos la técnica del PWM, con una frecuencia de 10 KHz. y un voltaje de 3.3 V, con esto se pretende que el motor tenga el par suficiente para corregir las posiciones de referencia, la interfaz de potencia amplifica esta señal a 20 V, con esto se provee al motor con el voltaje suficiente para hacer la función de corrección para las posiciones.

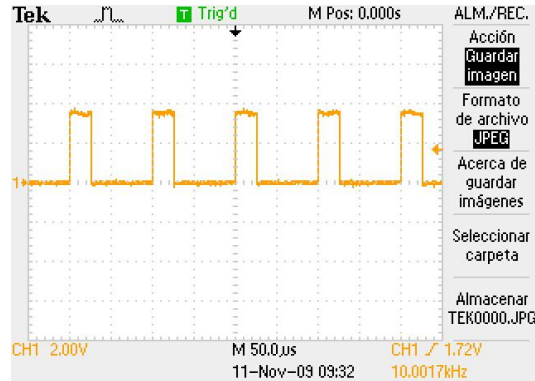


Figura A.1: Pulse Width Modulation.

La figura A.1 muestra la gráfica del osciloscopio del PWM que proporciona el microcontrolador DSP MC56F8367 para la interfaz de potencia, esta señal tiene una frecuencia de 10 KHz. y una amplitud de 3.3 V.

La interfaz de potencia se muestra en la Figura A.3, este circuito que está compuesto en su forma básica por un puente H, amplificadores de baja señal y un sistema de seguridad para evitar el uso de 2 PWM, esto es para saturar al doble Par de transistores con un solo PWM[12]. El funcionamiento de este circuito se describe a continuación:

Como datos de entrada a nuestra interfaz de potencia tenemos al PWM y al  $Key = A$ , ambas señales provienen del microcontrolador DSP MC56F8367, el PWM del puerto PWMA del pin 4 (PWM3) y el bit de dirección del puerto D pin 10 (Bit1-GPI0F2\_D9), estas dos señales entran al decoder 74LS156N ambas con una resistencia a tierra (R19 y R20) para evitar el estado true cuando estas señales no

estén conectadas a la interfaz[19], el PWM se conecta en el pin 1 del decodificador, este dato es el que saldrá de forma negada en los pines de salida, el bit de dirección se conecta al pin 13 (A), este junto con el pin 3 (B) dan la dirección del pin de salida este último esta puesto a tierra para trabajar solo con la dirección cero(pin 7) y uno(pin 6) estas son la salida del PWM para cada par de transistores de dirección, esta salida esta negada, es por eso que se colocan los circuitos 7404N para negar la salida[6], en la Tabla A.2 se observa la interacción de lo descrito anteriormente.

<b>TABLA DE VERDAD</b>			
<b>Bit de Dirección (A)</b>	<b>PWM (1C)</b>	<b>Giro Izquierdo (1Y0)</b>	<b>Giro Derecho (1Y1)</b>
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Tabla A.1: Tabla de Verdad de la Interfaz.

Para el giro izquierdo del motor, con el bit de dirección en cero, la salida del decodificador en 1Y0 polariza a los transistores Q2 y Q5, el primero es una etapa de preamplificación para Q10, el cual polariza al MOSFET Q8 y a su vez Q5 polariza al MOSFET Q1, siendo ambos MOSFET complementarios al igual que Q10 y Q5, de la misma manera ocurre con el giro derecho, con el bit de dirección en uno, la salida del PWM es en 1Y1, la cual polariza a Q9 y Q11 siendo el segundo la etapa de preamplificación para Q6 el cual polariza al MOSFET Q3, el MOSFET Q7 es polarizo por Q9, siendo los dos MOSFET complementarios, esto genera un voltaje de 20 V. para el motor, a una frecuencia de 10 KHz.[14] La Figura A.2 muestra la salida del

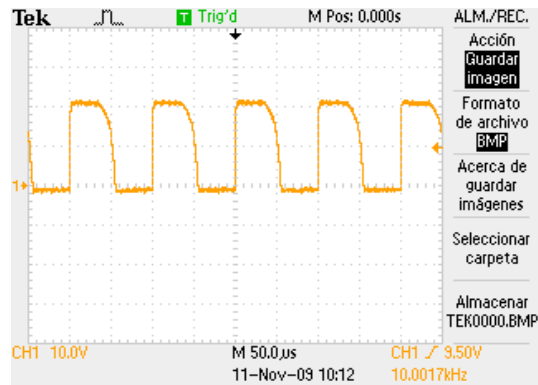


Figura A.2: Salida de la Interfaz de Potencia.

circuito de la Figura A.3, como se observa, la señal tiene la misma frecuencia que el PWM, pero la amplitud de la señal es de 20 V.





# Apéndice B

## Interfaz Gráfica con Labview

El diseño de la interfaz gráfica con LabVIEW esta dividida en cuatro bloques, estos bloques se mencionan a continuación:

- Diseño de la interfaz de comunicación.
- Diseño gráfico de las posiciones de los Brazos.
- Diseño gráfico de los botones de control.
- Diseño gráfico de los Brazos Primario y Secundario.

La interfaz gráfica esta diseñada para representar la posición de ambos brazos de forma gráfica, también por medio de esta podemos representar en una gráfica con respecto al tiempo dichas posiciones, la interfaz gráfica también proporciona un control de las diferentes actividades que realiza el sistema PENDUBOT, tales como inicio

del control de subida, el break de cualquier actividad que realice y otras más que se detallan más adelante:

## B.1 Diseño de la Interfaz de Comunicación

### B.1.1 Interfaz Análoga

Para poder diseñar la interfaz gráfica se ocupa el accesorio CB-68LP. Para graficar las posiciones de los brazos, es necesario leer el dato análogo que proporciona el microcontrolador DSP MC56F8367, es por eso que se ocupa el DAC Assistant[7], para leer un dato análogo ver Figura B.1, este bloque lee el dato que se encuentra en los pines 33 y 68 del accesorio CB-68LP, y entrega un valor numérico el cual será decodificado para representar las posiciones de los brazos, esto se detalla más adelante.



Figura B.1: Bloque de la Interfaz Análoga.

### B.1.2 Interfaz digital

Para el diseño de los botones de control es necesario una interfaz digital, la cuál se diseña con el bloque DAC Assistant[7], de pin de salida digital ver Figura B.2, este bloque genera un salida digital en los pines 52, 51, 17, 16 y 46 del accesorio CB-68LP, el microcontrolador DSP MC56F8367 lee los datos que salen de dichos pines, para poder decodificar la información que genera LabVIEW.



Figura B.2: Bloque de la Interfaz Digital.

## B.2 Diseño Gráfico de las Posiciones de los Brazos

La gráfica que proporciona la posición de los brazos se presenta en la Figura B.3, los datos que genera la interfaz análoga son gráficos de forma directa, ya que el microcontrolador DSP MC56F8367 genera un PWM de 16 bits para cada brazo, esto significa que la posición de los brazos de 0 rad. a  $2\pi$  rad. es proporcional al PWM de 16 bits en su valor entero de 0 a 65535, si el Brazo Primario se desplaza  $\pi$  rad., el PWM1 tendrá el valor de 32767.

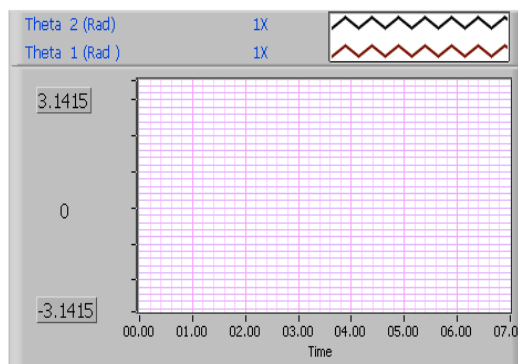


Figura B.3: Gráfica para las Posiciones Angulares.

Para poder leer el valor del PWM con LabVIEW es necesario convertirlo a un dato análogo, para lograr esto nos ayudaremos de un filtro pasa bajos sintonizado a una frecuencia de 1 KHz. ya que la frecuencia del PWM es de 10 KHz., con esto podemos recibir un dato de tipo análogo.

### B.3 Diseño Gráfico de los Botones de Control

Para poder controlar el sistema PENDUBOT, ocupamos el microcontrolador DSP MC56F8367, el programa de control esta seccionado en diferentes bloques, los cuales son seleccionados por medio de la interfaz gráfica de control, estos bloques son:

- Inicio
- Break
- Referencia

- Perturbación
- +Kd
- -Kd

La configuración de estos botones de control en LabVIEW se hacen por medio de la interfaz digital mencionada en la sección B.1, la Figura B.4 muestra como se realiza el diagrama a bloques de un pin de control.

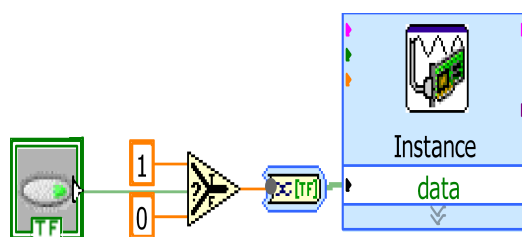


Figura B.4: Diagrama de Bloques de un Pin de Control.

La figura anterior cuenta con un push button (izquierda) de tipo digital interpretado por LabVIEW de forma booleana (False and True) el cual está conectado a un bloque selector que entrega el valor cero para false y uno para true, a su vez, este bloque se conecta a un convertidor de un dato de tipo dinámico a un dato de tipo array (arreglo o vector) el cual entra a la interfaz digital y da como salida cero y 5 V.

Cada uno de los 6 bloques de control se diseñaron de la misma manera que el bloque anterior, cada uno cuenta con un pin de salida diferente a excepción del bloque

de Inicio y Perturbación que comparten el mismo pin de salida. La función de cada uno de los bloques se menciona a continuación:

**Inicio**

Este bloque inicializa el programa de control del microcontrolador , primero ejecuta el control de subida y despues el de equilibrio, este pin solo funciona cuando los dos brazos estan en posición de equilibrio natural( $-\pi/2$ ). Este bloque ocupa el pin 17 del CB-68LP.

**Break**

Este bloque cuenta con dos funciones, la primera es abortar el programa en cualquier momento y la segunda es corregir la posición de los dos brazos, ya que estos tienen un desplazamiento por el ruido que genera el motor de DC. Este bloque ocupa el pin 52 del BC-68LP.

**Referencia**

Como parte adicional se genera una señal de referencia para la posición del Brazo Principal de tipo senoidal, este bloque inicializa dicha referencia esta señal fue diseñada para la capacidad del motor. Este bloque ocupa el pin 17 del CB-68LP.

**Perturbación**

Como su nombre lo indica, este bloque inicializa una pequeña perturbación en el Brazo Principal la cual consta de un voltaje constante en un tiempo constante. Este

bloque ocupa el pin 16 del CB-68LP.

### +Kd

Tanto es bloque como el siguiente influyen directamente en el control de subida, este bloque es un aumento a la ganancia derivativa, al aumentar esta ganancia el Brazo Secundario tiende a subir más despacio, esto genera que llegue con menor velocidad a la posición  $\pi/2$ . Este bloque ocupa el pin 51 del CB-68LP.

### -Kd

Este bloque disminuye el valor de la ganancia derivativa, esto hace que el Brazo secundario tienda a subir más rápidamente a la posición  $\pi/2$ . Estos últimos dos bloques se ocupan para calibrar el control de subida, ya que por el calentamiento del motor la relación del, PWM de control y el Par del motor cambien de forma drástica. La siguiente figura muestra los bloques anteriormente mencionados en el panel frontal.



Figura B.5: Bloques de Control.

## B.4 Diseño Gráfico de los Brazos Primario y Secundario

Por último se presenta el diseño del Brazo Principal y el Brazo Secundario, la Figura B.6 muestra el diagrama a bloques de esta sección, este diagrama dibuja ambos brazos en el panel frontal, por medio de la interfaz análoga se realiza la lectura de la señal que llega del filtro pasa bajos, con la finalidad de decodificarlo, para obtener la posición de los brazos en el plano frontal. La longitud del Brazo Principal y el Secundario y así como las coordenadas de estos, el grosor y la relación de los radios de los eslabones, se encuentran como constantes que se pueden alterar para modificar la presentación de los gráficos, también se añadió como extra, un control para el color de los mismos.

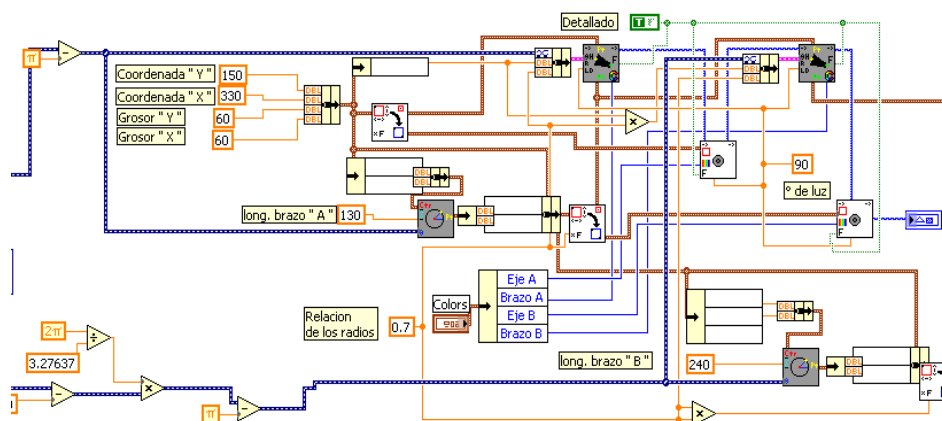


Figura B.6: Diagrama a bloques de los gráficos.

El diagrama a bloques completo de la interfaz gráfica se muestra en la Figura B.7,



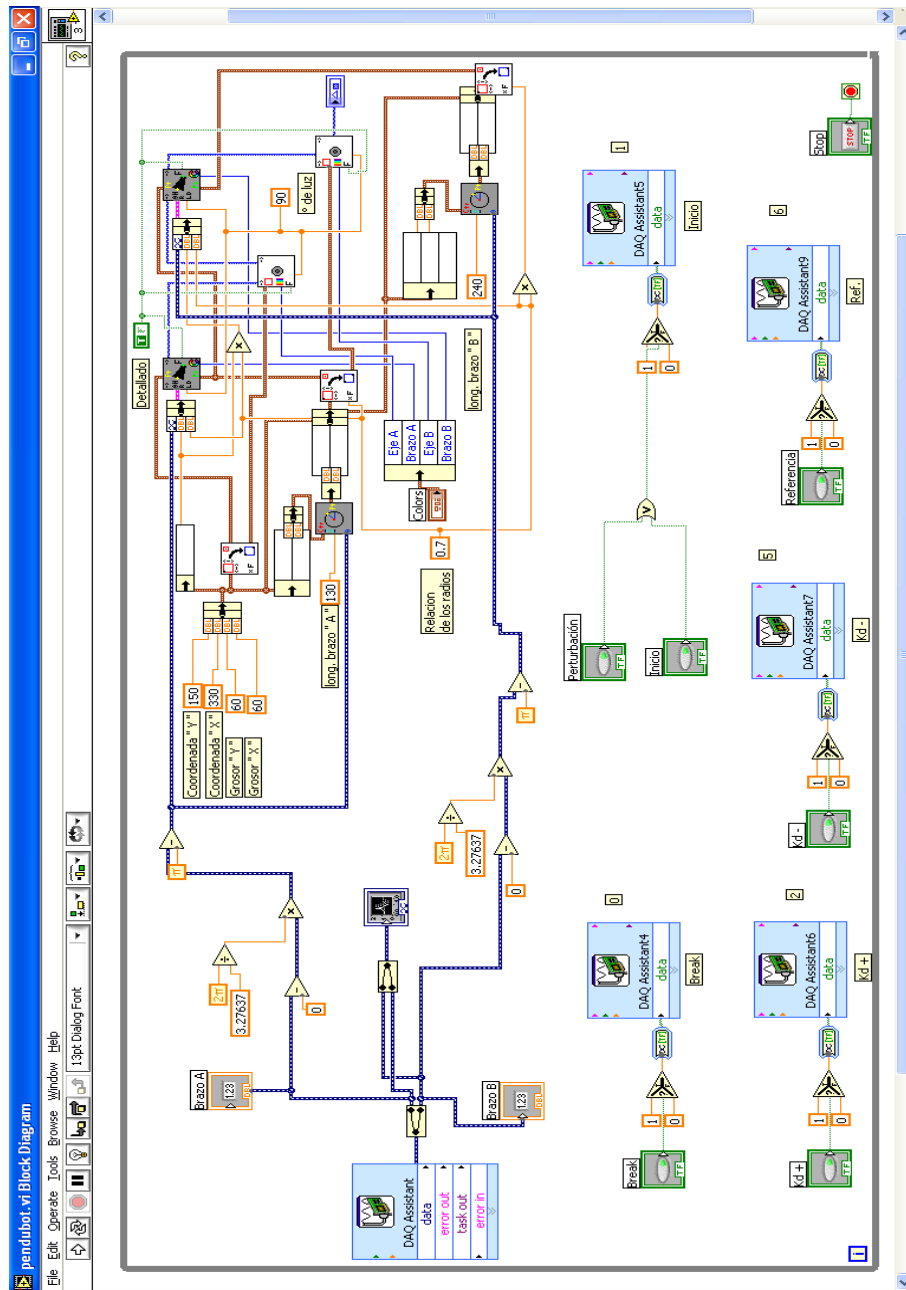


Figura B.7: Diagrama a Bloques de la Interfaz Gráfica.

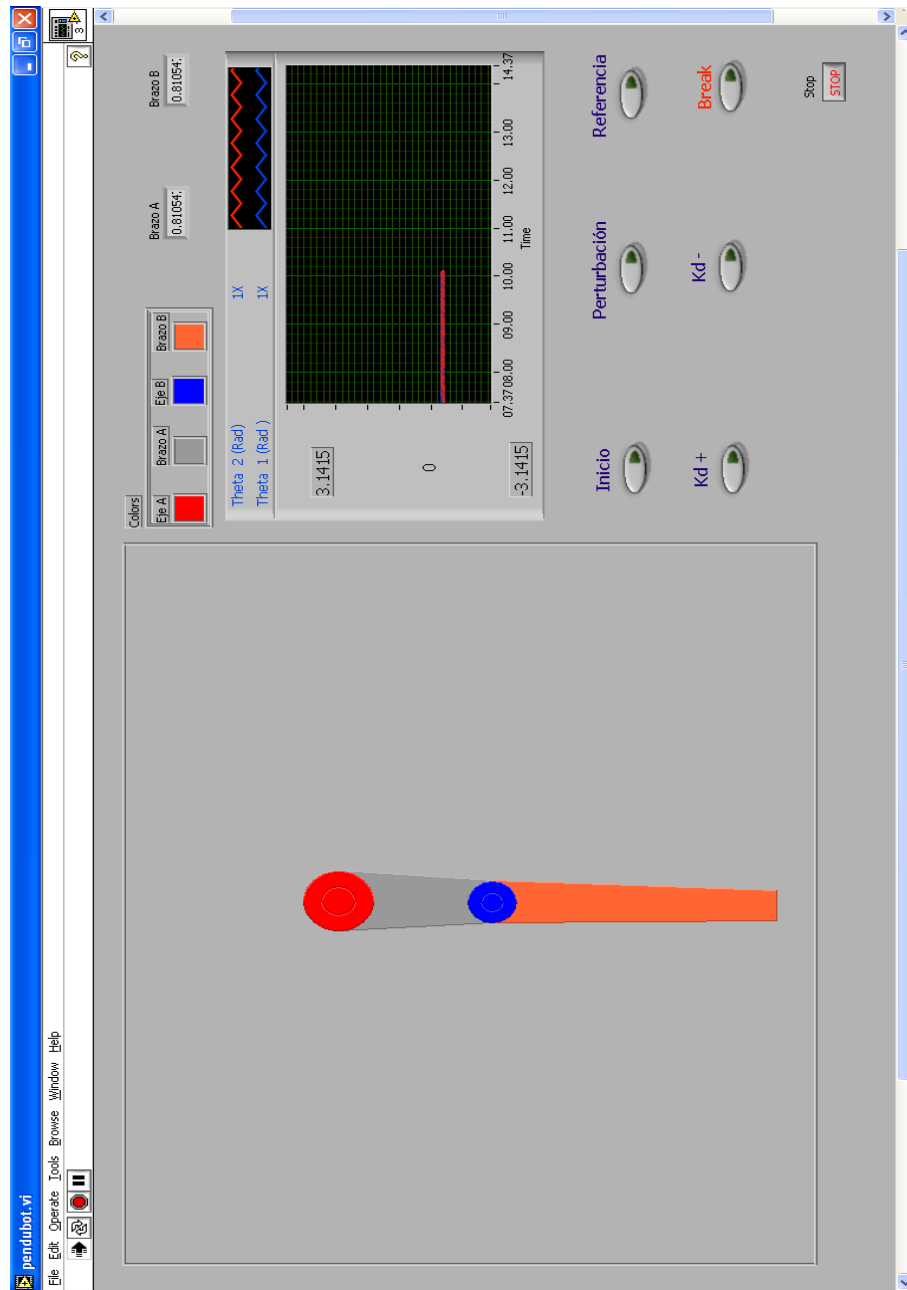


Figura B.8: Panel Frontal de la Interfaz Gráfica.

---

en esta figura se conjuntan todos los bloques mencionados anteriormente así como su interacción entre estos. El diagrama a bloques genera un panel frontal ver Figura B.8, en el cuál se presenta de forma gráfica los bloques descritos anteriormente.

# Apéndice C

## Interfaz de Posición Angular

### C.1 Encoders

Para poder calcular la posición angular de ambos brazos, es necesario contar con un instrumento que sea capaz de hacer esta tarea, para este proyecto el instrumento en cuestión es un encoder rotativo codificador en cuadratura, este encoder cuenta con cuatro pines, dos de estos son para el voltaje de alimentación que varía de 5 hasta 30 V. de corriente continua, en nuestro caso de 5 V., los otros dos pines corresponden a un tren de pulsos, defasada 90 grados una respecto a la otra, una la podemos llamar señal A y la otra señal B, la Figura C.1 muestra una de las laminas metálicas que generan los trenes de pulso, dentro del encoder hay dos laminas posicionadas de tal forma que las dos señales de salida están defasadas 90 grados, además la resolución de los dos encoders es de 2000 PPR(pulsos por revolución).

Este tipo de encoder se puede leer de tres formas, la primera es cuando la señal tiene

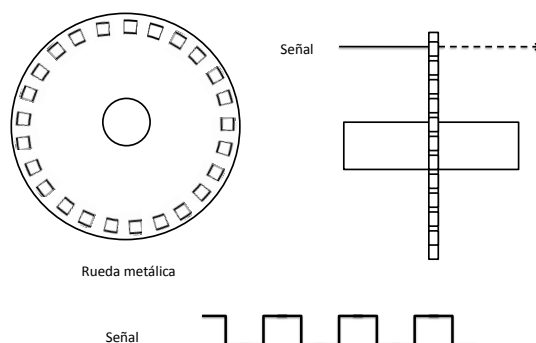


Figura C.1: Encoder Rotativo Codificador en Cuadratura.

un cambio de voltaje, generando una caída de voltaje del valor de 5 V. a 0 V. y sí en ese mismo instante de tiempo la señal B tiene un valor bajo, el contador del encoder es ascendente y la dirección de rotación es horaria, pero sí en ese instante de tiempo la señal B tiene un valor alto, el contador es descendente con dirección anti horaria, este tipo de codificación genera un aumento y decremento del contador por cada ciclo de la fase A, esto se puede observar en la Figura C.2.

El segundo tipo de decodificación es cuando la señal A tiene un cambio de voltaje de 5 V. a 0 V. y el valor de la señal B en ese mismo instante de tiempo es de 0 V. el contador aumenta, también cuando la señal A tiene un cambio de voltaje de 0 V. a 5 V. y el valor de la señal B en ese mismo instante de tiempo es de 5 V., en este caso al igual que el anterior, el contador es ascendente y la dirección de la rotación es horaria, pero si la señal A tiene un cambio de nivel, de alto a bajo y la señal B tiene el nivel alto ó si el cambio de nivel de la señal A es de bajo a alto y el nivel de la señal

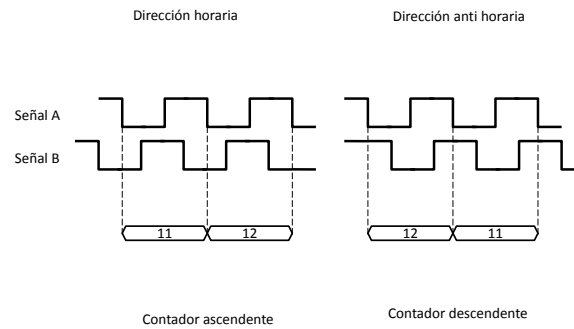


Figura C.2: Primera Tipo de Decodificación.

es bajo, el contador decrementa su valor y la dirección de giro es anti horario, para este tipo de codificación se genera un cambio en el contador por cada medio ciclo de la señal A, Figura C.3.

El tercer tipo de decodificación se observa en la Figura C.4, se genera un cambio en el contador por cada cambio de nivel de las señales A y B, dando como resultado, cuatro conteos positivos o negativos, por cada ciclo de la señal A, este tipo de decodificación es más exacto, pero también es más propenso a lecturas erróneas por ruido, es por eso que para la lectura de las posiciones del PENDUBOT se hace la decodificación de Tipo 1.

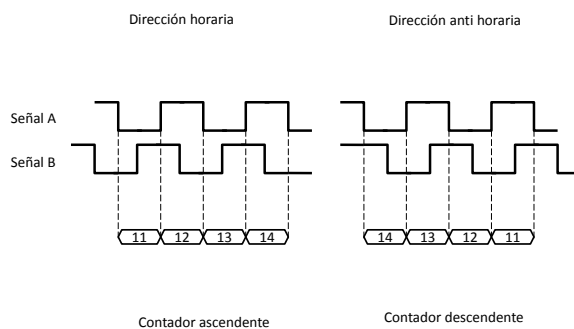


Figura C.3: Segundo Tipo de Decodificación.

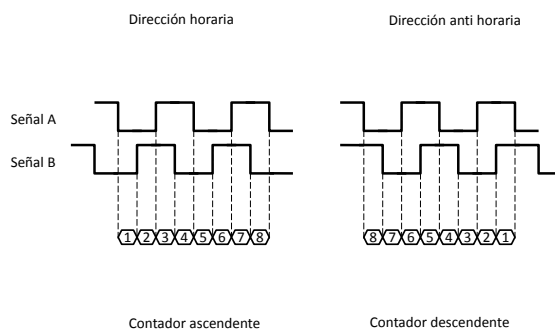


Figura C.4: Tercer Tipo de Decodificación.

## C.2 Lectura de las Posiciones

En el microcontrolador DSP MC56F8367 por medio del uso de interrupciones externas podemos saber cuando la señal A de los encoders han tenido un cambio de nivel de alto a bajo y en ese momento podemos saber que valor tiene la señal B, si la señal B tiene un valor de 0 vcc, entonces se incrementa el valor del contador, de lo contrario se le decrementa el valor. Esto se observa en la línea de programa de EInt1\_OnInterrupt (module Events) y en EInt2\_OnInterrupt (module Events), las cuales se muestran en el Apéndice E.

Ya que se se tiene el valor numérico de las posiciones se entregan a LabVIEW por medio de dos PWM, variando su ciclo de trabajo de forma proporcional al valor de la posición.

## C.3 Filtro Pasa Bajos

Para la interfaz de posición angular nos auxiliaremos de LabVIEW, como se vio en el Apéndice B, podemos configurar una entrada análoga, que sea capaz de leer una señal proveniente de un filtro pasa bajos, este filtro es necesario para convertir el PWM proveniente del microcontrolador a una señal análoga, la Figura C.5 muestra un filtro pasa bajos de primer orden.

Un análisis de Fourier de una señal PWM típicos, muestra que hay una señal muy fuerte a la frecuencia  $F_n = 1/T$ . Otros armónicos fuertes también existen en



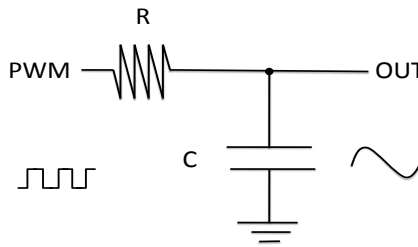


Figura C.5: Filtro Pasa Bajos de Primer Orden.

$F = K/T$ , donde  $K$  es un entero. Estos picos son el ruido no deseado que es un valor de CA. y deben ser eliminados para obtener un valor DC. Esto requiere que la señal PWM este muy por debajo de  $F_n$ , ver Figura C.6. La frecuencia de corte del filtro pasa bajos debe de ser mucho menor que la frecuencia del PWM [18] esto es:

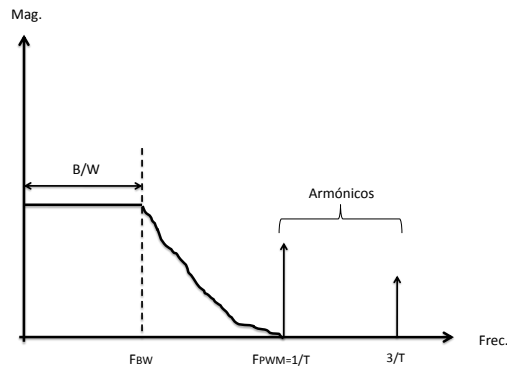


Figura C.6: Espectro de Frecuencia para el Filtro Pasa Bajos.

$$F_{BW} \ll F_{PWM}$$

$$F_{PWM} = F_{BW}K$$

$$K \gg 1$$

Donde  $F_{BW}$  es la frecuencia de corte para el filtro y  $K$  es la relación de las dos frecuencias.

La relación para calcular la frecuencia de corte con respecto a la resistencia y capacitancia del filtro es:

$$RC = \frac{1}{2\pi F_{BW}} \quad (C.1)$$

La Ecuación C.1 muestra como diseñar un filtro tomando en cuenta el valor de la Resistencia o el de la Capacitancia como constante y fijando la frecuencia de corte, de esta manera se diseño el filtro pasa bajos con :

$$R = 10Khz$$

$$C = 10\mu F$$

# Apéndice D

## Materiales y Costos

El PENDUBOT, esta integrado por los siguientes componentes:

- Encoders WDG 58b-2000-AB-G24-93, de la marca Wachendorff Electronik de 2000 ppr. de 5V. a 30V.
- Motor de c.d. de 5 Amp. ,de hasta 30 V
- Interfaz de potencia de 20 a 30 V, 20 A., basada en el principio del puente H.
- Fuente de c.d. de hasta 30 V., 5 A.
- Tarjeta de adquisición de datos, marca National Instruments NI USB-600X, con una resolución de 14 bits, 48KS/s , 12 líneas de entradas y salidas digitales con tecnología TTL/CMOS.
- Microcontrolador DSP MC56F8367
- Eslabón de aluminio

<b>Presupuesto</b>		
<b>Cantidad</b>	<b>Materiales</b>	<b>Costos</b>
2	Encoder WDG 58b-2000-AB-G24-93	\$14,000.00
1	Motor de c.d.	\$2,000.00
1	Interfaz de potencia	\$500.00
1	Fuente de c.d	\$2,500.00
1	Microcontrolador DSP MC56F8367	\$3,500.00
1	DAC NI USB-600X	\$2,500.00
2	Eslabón de aluminio	\$200.00
<b>Total</b>		<b>\$25,200.00</b>

Tabla D.1: Presupuesto.

# Apéndice E

## Programa de Control en el Microcontrolador

El programa de control esta dividido en dos secciones:

- Main (Programa Principal)
- Events (Eventos)

Main es el programa donde se manipulan todas las variables del sistema, ademas se diseño con el fin de proporcionar el algoritmo de control y las opciones que se generan desde la interfaz gráfica de LabVIEW. El programa está estructurado según la Figura E.1, que es la estructura general del programa de control.

Events es la sección que se encarga de hacer los cálculos que dependan del tiempo, tales como el cálculo de  $\theta_1$  y  $\theta_2$  en Radianes, ademas se encarga del cálculo de las velocidades de los brazos en Radianes/seg., también realiza los cálculos de las señales de

referencia para posicionar los brazos en las posiciones correspondientes para el control de balancín y la señal para probar este control, a su vez se encarga de interpretar las interrupción, como se observa en el algoritmo, las interrupciones las generan las fases A, de los encoders.

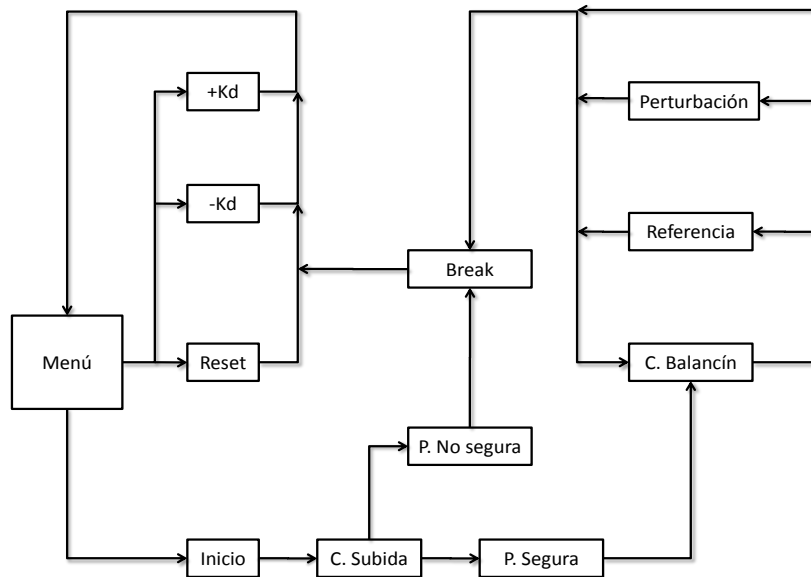


Figura E.1: Diagrama de bloques del Programa de Control

```

PROGRAMA PRINCIPAL
/** #####
/**  Filename : pendubot1.C
/**  Project  : pendubot1
/**  Processor : 56F8367
/**  Version  : Driver 01.13
/**  Compiler  : Metrowerks DSP C Compiler
/**  Date/Time : 23/10/2009, 18:28
/**  Abstract  :
    
```

```
/**      Main module.
/**      This module contains user's application code.
/**      Settings :
/**      Contents :
/**      No public methods
/**
/**      (c) Copyright UNIS, a.s. 1997-2008
/**      UNIS, a.s.
/**      Jundrovska 33
/**      624 00 Brno
/**      Czech Republic
/**      http : www.processorexpert.com
/**      mail : info@processorexpert.com
/** #####
/* MODULE pendubot */

/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "EInt1.h"
#include "InpB1.h"
#include "EInt2.h"
#include "InpB2.h"
#include "TI1.h"
#include "PWM1.h"
#include "PWM2.h"
#include "PWM3.h"
#include "InpB3.h"
#include "InpB4.h"
#include "InpB5.h"
#include "InpB6.h"
#include "InpB7.h"
#include "Bit1.h"
#include "Bit2.h"
```

```
/* Including shared modules, which are used for whole project */
#include "PE.Types.h"
#include "PE.Error.h"
#include "PE.Const.h"
#include "IO.Map.h"
#include "stdio.h"
#include "math.h"
#include "stdlib.h"

float alf=0;           //Control de Subida
float bta=0;
float v1=0;
float tau=0;

float A1=0.010644;
float A2=0.00351708;
float A3=0.00351708;
float A4=0.503645;
float B1=0.00351708;
float B2=0.00391491;
float B3=0.00351708;
float B4=0.172513;

word pwm1=65535;      //PWM del motor

extern float contARad; //Variables compartidas con eventos
extern float contBRad;
extern float w1;
extern float w2;
extern float reft1;
extern float x;
extern int contA;
extern int contB;
extern float pi;
```



```
extern int per=0;
extern float ref1=0;
extern int xt=0;

float a=0;           //Variables de control
float b=0;
int c=0;
int r=0;
int t=0;

float ku=0;          //Factores
float Fr=7;
float Frs=10;
float Kp=50;
float Kd=1.6;

float kb1=-0.6099;   //Posición Theta 1
float kb2= 1.8901;   //Posición Theta 2
float kb3= -0.0850;  //Velocidad Theta 1
float kb4= 0.2612;   //Velocidad Theta 2

void main(void)
{
/* Write your local variable definition here */

/** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! **/
PE_low_level_init();
/** End of Processor Expert internal initialization. **/

/* Write your code here */

PWM1_SetRatio16(pwm1);
for(;;) {
```

```
/*****          " INICIO "          *****/
/*****          " MENU "           *****/
PWM1_SetRatio16(65535);
xt=0;
while(InpB3.GetVal()!=0)
{
/*****          " ESPERA "          *****/
while(InpB5.GetVal()==0)
{
/*****          " RESET "          *****/
if(InpB3.GetVal()!=0)
{
while(InpB3.GetVal()!=0)
{}
contA=1500;
contB=0;
}
/*****          " KD + "           *****/
if(InpB7.GetVal()!=0)
{
while(InpB7.GetVal()!=0)
{}
Kd=Kd+0.05;
}
/*****          " KD - "           *****/
if(InpB4.GetVal()!=0)
{
while(InpB4.GetVal()!=0)
{}
Kd=Kd-0.05;
} }
/*****          " CONTROL DE SUBIDA" *****/
while(InpB5.GetVal()!=0)
{
```

```

x=0;
c=0;
while (InpB3.GetVal()==0&&c==0)
{
if(x<=(2*pi))
{
alf =
(w2*w2)*A2*sin(contARad-contBRad) +
(w1*w1)*((A2*A2)/B2)*cos(contARad-contBRad)*sin(contARad-contBRad) +
A4*cos(contARad)-B4*(A2/B2)*cos(contBRad)*cos(contARad-contBRad);
bta=
A1-((A2*A2)/B2)*cos(contARad-contBRad)*cos(contARad-contBRad);
v1=Kp*(reft1-contARad)-Kd*w1;
tau=v1*bta+alf;
tau=(tau*65535)/Frs;
if(tau>65535)
tau=65535;
if(tau<0)
{
b=65535+tau;
pwm1=(word)(b);
PWM1_SetRatio16(pwm1);
Bit1_SetVal();
}
else
{
b=65535-tau;
pwm1=(word)(b);
PWM1_SetRatio16(pwm1);
Bit1_ClrVal();
Bit1_ClrVal();
} }
/*****          " CONTROL DE BALANCIN "          *****/
if((contBRad-(pi*.5))<=0.3)

```

```
{
x=3*pi;
ku=kb1*(contARad-((pi*1.5)+ref1))+kb2*(contBRad-(pi*.5))+kb3*w1+kb4*w2;
ku=(ku*65535)/Fr;
if(ku<0)
{
b=65535+ku;
if(b<0)
b=0;
pwm1=(word)(b);
PWM1_SetRatio16(pwm1);
Bit1_ClrVal();
}
else
{
ku=ku*-1;
b=65535+ku;
if(b<0)
b=0;
pwm1=(word)(b);
PWM1_SetRatio16(pwm1);
Bit1_SetVal();
} }
/*****          " POSICIÓN SEGURA "          *****/
if(x>1.9*pi&&contBRad-(pi*.5)>0.5)
c=1;
/*****          " PERTURBACION "          *****/
if(InpB5.GetVal()!=0)
r=1;
if(r==1&&InpB5.GetVal()==0)
{
per=0;
PWM1_SetRatio16(0);
Bit1_SetVal();
```

```
while(per<50)
{
r=0;
}
/*****          " REFERENCIA"          *****/
if(InpB6.GetVal()!=0&&ref1==0)
{ xt=1;
} } }
/* END pendubot */
```

```
PROGRAMA PARA LA SECCIÓN DE EVENTOS:

/** #####
/**  Filename : Events.C
/**  Project  : pendubot1
/**  Processor : 56F8367
/**  Beantype  : Events
/**  Version  : Driver 01.03
/**  Compiler  : Metrowerks DSP C Compiler
/**  Date/Time : 23/10/2009, 18:28
/**  Abstract  :
/**  This is user's event module.
/**  Put your event handler code here.
/**  Settings  :
/**  Contents  :
/**  TI1_OnInterrupt - void TI1_OnInterrupt(void);
/**
/**  (c) Copyright UNIS, a.s. 1997-2008
/**  UNIS, a.s.
/**  Jundrovská 33
/**  624 00 Brno
/**  Czech Republic
/**  http : www.processorexpert.com
/**  mail : info@processorexpert.com
/** #####
/** MODULE Events */

#include "Cpu.h"
#include "Events.h"
#include "stdio.h"
#include "math.h"

extern int contA=1500;
int contA1=0;
float contA.1=0;
```

```
extern float contARad=4.71238898025;
extern float reft1=4.71238898025;
float x=0;

extern int contB=0;
float contB.1Rad=4.71238898025;
extern float contBRad=4.71238898025;

extern float w1=0;
float w1.1=0;
float w1.2=0;

extern float w2=0;
float w2.1=0;
float w2.2=0;

word pwm2=65535;
word pwm3=65535;

float mk1=0;
float mk=0;
float mk.1=0;
float ek=0;
float ek.1=0;
static float Rf=0;

static float referencia=180;
static float kp=.7;
static float ki=.002;
static float kd=.7;
float T=.001;
extern float pi=3.1415926535;
extern int per;
extern float ref1;
```

```

extern int xt;

float xy=0;

int i1=0;

int i2=0;

/** =====
/**  Event : TI1_OnInterrupt (module Events)
/**
/**  From bean : TI1 [TimerInt]
/**  Description :
/**      When a timer interrupt occurs this event is called (only
/**      when the bean is enabled - <Enable> and the events are
/**      enabled - <EnableEvent>). This event is enabled only if a
/**      <interrupt service/event> is enabled.
/**  Parameters : None
/**  Returns : Nothing
/** =====

#pragma interrupt called

/* Comment this line if the appropriate 'Interrupt preserve registers' property */
/* is set to 'yes' (#pragma interrupt saveall is generated before the ISR) */
void TI1_OnInterrupt(void)
{ /*****          " calculo de theta 1 Rad "          *****/
contARad=(contA*pi)/1000;
pwm2=(word)((contARad*65535)/(2*pi)+0);
PWM2_SetRatio16(pwm2);
/*****          " calculo de theta 2 Rad "          *****/
contBRad=((contB*pi)/1000)+contARad;
if(contBRad>=(2*pi))
contBRad=contBRad-(2*pi);
pwm3=(word)((contBRad*65535)/(2*pi)+0);
PWM3_SetRatio16(pwm3);
/*****          " calculo de la velocidad angular rad/s theta 1 "          *****/
w1=((contA1-contA_1)/T)*pi/1000;
contA_1=contA1;

```



```
w1=(w1+w1.1+w1.2)/3;
w1.2=w1.1;
w1.1=w1;
/*****          " calculo de la velocidad angular rad/s theta 2 "          *****/
w2=((contBRad-contB.1Rad)/T);
contB.1Rad=contBRad;
w2=(w2+w2.1+w2.2)/3;
w2.2=w2.1;
w2.1=w2;
/*****          "señal de referencia para theta 1"          *****/
if(x<=(2*pi))
{
ref1=cos(x)+(pi*1.5);
x=x+pi/1100;
}
else
ref1=pi*1.5;
/*****          "perturbacion"          *****/
if(per<(50))
per++;
/*****          "referencia"          *****/
if(xt==1)
{
if(xy<=(2*pi))
{
ref1=.14*sin(xy);
xy=xy+pi/3000;
}
else
{
ref1=0;
xt=0;
}}
else
```

```
{
xy=0;
ref1=0;
}}
/** =====
/**  Event :  EInt2_OnInterrupt (module Events)
/**
/**  From bean :  EInt2 [ExtInt]
/**  Description :
/**      This event is called when an active signal edge/level has
/**      occurred.
/**  Parameters :  None
/**  Returns :  Nothing
/** =====
#pragma interrupt called
/* Comment this line if the appropriate 'Interrupt preserve registers' property */
/* is set to 'yes' (#pragma interrupt saveall is generated before the ISR) */
void EInt2_OnInterrupt(void)
{
if(InpB2.GetVal()==0)
contB++;
else
contB--;
if(contB<0)
contB=1999;
if(contB>=2000)
contB=0;
}
```

```
/** =====
 * Event : EInt1_OnInterrupt (module Events)
 *
 * From bean : EInt1 [ExtInt]
 * Description :
 *     This event is called when an active signal edge/level has
 *     occurred.
 * Parameters : None
 * Returns : Nothing
 * =====
#pragma interrupt called
/* Comment this line if the appropriate 'Interrupt preserve registers' property */
/* is set to 'yes' (#pragma interrupt saveall is generated before the ISR) */
void EInt1_OnInterrupt(void)
{
if(InpB1.GetVal()==0)
{
contA--;
contA1--;
}
else
{
contA++;
contA1++;
}
if(contA<0)
contA=1999;
if(contA>=2000)
contA=0;
}
/* END Events */
```