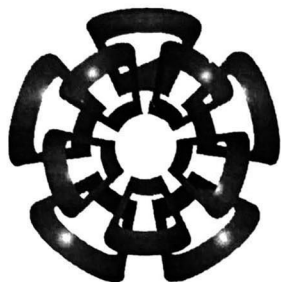




xx(29875,1)





# CINVESTAV - IPN

Centro de Investigación y de Estudios Avanzados del IPN  
Unidad Guadalajara

---



## **DISEÑO DE UN BANCO DE CONTROLADORES LINEALES PARA EL PENDUBOT**

**TESIS QUE PRESENTA  
BERNARDO HARO MARTÍNEZ**

**PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS**

**EN LA ESPECIALIDAD DE  
INGENIERÍA ELÉCTRICA**

Guadalajara, Jal. Agosto de 1999

CLASIF.	
ADQUIS.	TESIS-1999
FECHA:	24-XI-99
PROCED.	Depto. Seva

661

DIRECTOR DE TESIS  
**DRA. OFELIA BEGOVICH MENDOZA**

El hombre es fuego que conviene alimentar y que se apaga si no se aumenta.

*Voltaire*



Dedicado a:  
Fernanda Monserrat Haro Torres  
y  
Norma Griselda Torres Toscano

Y un especial agradecimiento a mis padres:

Alma Martínez de Haro

Y

Bernardo Haro Estrada.

Con cariño mis hermanos:

David Haro Martínez (Q.E.D.)

Irma Haro Martínez

Gerardo Haro Martínez

Humberto Haro Martínez

Alma Haro Martínez

Jaime Haro Martínez

Julio Haro Martínez

Blanca Haro Martínez

# Índice General

<b>Agradecimientos</b>	<b>5</b>
<b>NOTACION</b>	<b>7</b>
<b>1 Introducción</b>	<b>9</b>
<b>2 Modelado matemático.</b>	<b>11</b>
2.1 Introducción	12
2.2 Sistemas subactuados	12
2.3 Modelo del Pendubot	13
<b>3 Parámetros</b>	<b>25</b>
3.1 Introducción	26
3.2 Obtención de parámetros	26
3.2.1 Momentos de inercia	26
3.2.2 Coeficientes de fricción	30
3.2.3 Ganancia del par del motor	31
<b>4 Modelización lineal</b>	<b>35</b>
4.1 Introducción	36
4.2 Puntos de equilibrio del Pendubot	36
4.3 Linealización	37
4.4 Discretización	41
4.4.1 Método computacional	42
<b>5 Síntesis del Control</b>	<b>45</b>
5.1 LQ discreto	46
5.2 Especificaciones	46
5.3 Ganancias preprogramadas	47
5.4 Algoritmo de control	48
5.4.1 Posición angular	49
5.4.2 Aproximación de la velocidad angular	49
5.4.3 Par nominal $\tau_k^0$	50
5.4.4 Señal de control del motor	50

<b>6 Resultados</b>	<b>55</b>
6.1 Resultados	56
<b>7 Conclusiones</b>	<b>65</b>
7.0.1 Investigaciones futuras:	65
<b>Sección de apéndices</b>	<b>71</b>
<b>A Instrucciones de Matlab y Simnon</b>	<b>73</b>
<b>B Programa en Matlab para linealizar</b>	<b>75</b>
<b>C Programa en C</b>	<b>79</b>
<b>D Simulación con Simnon.</b>	<b>89</b>



# Índice de Figuras

2.1	Diagrama esquemático del Pendubot, vista frontal y lateral.	12
2.2	Diagrama del Pendubot	14
2.3	Detalle del eslabón 2 del Pendubot	16
3.1	Determinación del centroide del eslabón 1	28
3.2	Determinación del centroide del eslabón 2	28
3.3	Comparación de los resultados experimentales con los resultados de simulación del comportamiento pendular del eslabón 1 para la obtención de su fricción.	30
3.4	Comparación de los resultados experimentales con los resultados de simulación del comportamiento pendular del eslabón 2 para la obtención de su fricción.	31
3.5	Relación de la corriente de armadura y la fuerza del motor del Pendubot a una distancia $l_1$ .	32
4.1	Puntos de equilibrio del Pendubot.	36
5.1	Diagrama a bloques de un banco de controladores para el Pendubot	48
5.2	Dibujo esquemático de conexión del Pendubot.	51
5.3	Gráfica que relaciona la Corriente de Armadura de motor del Pendubot vs. la Salida Digital de la computadora personal	53
6.1	Respuesta de la posición angular $q_1$ en simulación y en tiempo real.	58
6.2	Respuesta de la posición angular $q_2$ en simulación y en tiempo real.	59
6.3	Error de seguimiento para $q_1$ en simulación y en tiempo real.	60
6.4	Resultados del error obtenidos de la simulación del modelo matemático y del modelo real con $0^\circ \leq q_2 \leq 70^\circ$	61
6.5	Respuesta de la señal de control en simulación.	62
6.6	Respuesta de la señal de control en tiempo real.	63
A.1	Menú de Simnon	74



# Agradecimientos

La preparación de esta tesis requirió de gran apoyo del grupo de trabajadores e investigadores del Cinvestav de la unidad Guadalajara, a quienes quiero agradecer su valiosa ayuda que me permitió concluir exitosamente este trabajo.

De manera muy especial quiero agradecer el apoyo incondicional y compañerismo recibido por el Dr. Antonio Ramírez Treviño, a la Dra. Ofelia Begovich con respeto y admiración por su invaluable ayuda y su gran tino en la corrección y contenido de esta tesis, al Dr. Bernardino Castillo Toledo por proporcionarme la idea de controlar el Pendubot con incrementos entre puntos de equilibrio, al Dr. Edgar Sánchez por su gran paciencia y por compartir su sabiduría conmigo al mostrarme algunos conceptos de control.

Quiero también expresar mi agradecimiento al M.C. Luis Enrique Ramos Velasco y al M.C. Carlos Mex Perera por su sugerencias en el control del Pendubot en tiempo real.

Dirijo también un especial agradecimiento al Dr. Jorge Aguilera Casillas quien es catedrático de la Universidad Autónoma de Guadalajara, al Dr. Rafael Jaime Alejo, Coordinador de Ingenierías de la Universidad Autónoma de Guadalajara que gracias a su constante apoyo permitieron el progreso de esta tesis.

Por último quiero agradecer calurosamente a todos aquellos que he omitido sin intención y que contribuyeron directa e indirectamente a la realización de esta tesis.





# NOTACION

$A_{ij}$	Matriz $A$ de dimensión $i \times j$
$a_{ij}$	Elemento de la $i$ -ésima fila y la $j$ -ésima columna de la matriz $A$
$A^T$	Matriz transpuesta de $A$
$A^{-1}$	Matriz inversa de $A$
$A > 0$	Matriz definida positiva
$A \geq 0$	Matriz semi-definida positiva
$\triangleq$	Definición de igualdad
$\mathcal{D}$	Operador derivativo
$x(t) _{t=T_s}$	$x(t)$ evaluada cuando $t = T_s$
$\approx$	Aproximadamente igual



# RESUMEN

---

Esta tesis presenta el diseño y aplicación en tiempo real de un banco de controladores para el sistema subactuado denominado “Pendubot”. Cada controlador se diseña para un punto de equilibrio específico mediante la técnica de Control Lineal Cuadrático ( $LQ$ ). Las ganancias de los controladores y pares nominales calculados son conmutados secuencialmente en función de una señal de referencia especificada. El banco de controladores obtenidos permite mover los eslabones del Pendubot de un punto de equilibrio a otro, en un rango determinado de posiciones angulares. Los experimentos realizados muestran un desempeño del sistema en lazo cerrado satisfactorio.

---

# Capítulo 1

## Introducción

El avance de la tecnología en las computadoras ha permitido que el hombre resuelva problemas de mayor complejidad en un menor tiempo y con mayor exactitud. Un factor importante se ha debido a la miniaturización de los dispositivos electrónicos que ha favorecido la reducción de costos de computadoras; de esta manera, un gran número de personas actualmente tienen acceso a estos dispositivos. Aunado a esto, se ha tenido un gran avance en el desarrollo de “software” para computadoras que ha puesto a disposición del usuario herramientas de trabajo muy sofisticadas para la solución de problemas. Por otro lado, el área de la robótica se ha beneficiado grandemente con estos avances, fabricándose múltiples modelos de robots con tecnología de vanguardia. El robot empleado en esta tesis es el Pendubot. Este robot es un planar de dos grados de libertad con un actuador y un codificador en la primera unión y un codificador en la segunda unión. Los codificadores determinan la posición angular de los eslabones.

El objetivo de esta tesis es diseñar un controlador a base de ganancias preprogramadas para controlar el Pendubot en distintos puntos de equilibrio en tiempo real ante unas especificaciones de desempeño dadas. Básicamente en la presente tesis se determinan los valores de los parámetros del Pendubot; se obtiene su modelo matemático; se diseña un conjunto de controladores que estabilicen al Pendubot en varias posiciones de equilibrio y finalmente, se presentan los resultados en simulación y en tiempo real obtenidos al aplicar el banco de controladores diseñados en el Pendubot.

El Pendubot es un prototipo usado en educación e investigación. Con este sistema se pueden realizar experimentos sobre identificación, control lineal, control no lineal, control difuso, control neuronal, control de conmutación e híbrido y otras metodologías [2], [3], [4], [5], [11]. Sin embargo debido a la dinámica no lineal del Pendubot, éste posee algunas características especiales y retos para la investigación que no se encuentran en otros dispositivos tales como el Péndulo invertido rotacional y el Péndulo invertido traslacional.

En esta tesis se obtiene el modelo matemático del Pendubot. Dicho modelo



se deduce empleando conceptos de física y se ha tomado especial cuidado en presentar su desarrollo, de tal forma, que el lector pueda seguir fácilmente su deducción. Los valores de los parámetros se validaron mediante experimentos realizados en tiempo real sobre el Pendubot. El procedimiento usado en la presente tesis para determinar los valores de los parámetros, puede considerarse como una aportación al conjunto de trabajos existentes en este campo [2], [3], [4], [5], [7], [11], [12]. Cabe mencionar, que con los valores de los parámetros reportados en la literatura previa (inercia, fricción, etc.), entre los resultados teóricos y prácticos existía una discrepancia considerable, mientras que con los valores aquí reportados, esta discrepancia no es significativa.

Para auxiliar a la obtención del modelo matemático del Pendubot se utilizaron herramientas computacionales. Estas herramientas fueron: Matlab, Simnon, Maple, Scientific WorkPlace, Matemática y Lenguaje C.

El control del Pendubot se realiza mediante el empleo de una computadora personal donde se implementa el programa de control en Lenguaje C. La computadora obtiene información de los codificadores de cada eslabón y envía una acción de control al actuador del Pendubot.

Mas en detalle, la estructura de esta tesis es la siguiente: En el capítulo 2, se presenta el modelado matemático del Pendubot. Partiendo de este modelo, en el capítulo 3 se determinan los valores de los diferentes parámetros del Pendubot tales como fricción, inercias, etc. En el capítulo 4, se obtienen los modelos lineales para varios puntos de equilibrio. En el capítulo 5 se diseña un conjunto de controladores  $LQ$  para los modelo linealizados y se establece el algoritmo para la conmutación de los controladores ("schedule gain"). El capítulo 7 presenta los resultados de simulación y en tiempo real obtenidos con el banco de controladores. Finalmente en el capítulo 8 se presentan las conclusiones.

## Capítulo 2

### Modelado matemático.

---

**Resumen:** En este capítulo se obtiene el modelo no lineal del Pendubot usando el concepto del Lagrangiano y principios de física.

---

## 2.1 Introducción

En este capítulo se obtiene el modelo matemático del Pendubot. Este modelo considera la fricción que existe en cada una de las uniones, obteniéndose así un modelo más completo que los encontrados en la literatura existente donde generalmente desprecian este parámetro [14], [15], [16]. La obtención del modelo se efectúa usando el concepto del Lagrangiano y principios de física.

La figura 2.1 muestra el dibujo del Pendubot. El nombre del Pendubot es un acrónimo de las palabras en inglés “PENDULum” y “roBOT”.

El Pendubot consiste de dos eslabones; el primer eslabón está directamente acoplado al eje de un motor fijado firmemente. El motor tiene acoplado un codificador óptico que determina la posición del primer eslabón. El eslabón 2 está acoplado al extremo del eslabón 1.

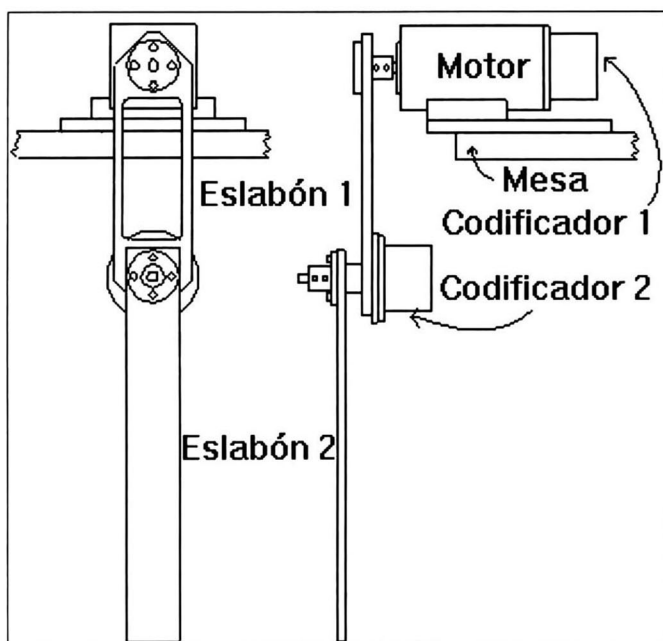


Figura 2.1: Diagrama esquemático del Pendubot, vista frontal y lateral.

## 2.2 Sistemas subactuados

Un sistema subactuado de  $n$  grados de libertad con coordenadas generalizadas  $q^1, \dots, q^n$  tendrá  $m < n$  actuadores, donde cada actuador tendrá efecto directamente sobre un solo grado de libertad. Los  $n$  grados de libertad actuados también son llamados uniones activas y los  $l = n - m$  grados de libertad subactuados, uniones pasivas. Para nuestro análisis dividiremos el vector  $q \in \mathbb{R}^n$

de coordenadas generalizadas en  $q_1 \in \mathbb{R}^m$  y  $q_2 \in \mathbb{R}^l$ , donde  $q_1 \in \mathbb{R}^m$  representa las uniones actuadas o activas y  $q_2 \in \mathbb{R}^l$  representa las uniones subactuadas o pasivas. Las ecuaciones del sistemas son:

$$D_{11}(q) \ddot{q}_1 + D_{12}(q) \ddot{q}_2 + C_1(q, \dot{q}) + G_1(q) + F_1(\dot{q}) = \tau$$

$$D_{12}(q) \ddot{q}_1 + D_{22}(q) \ddot{q}_2 + C_2(q, \dot{q}) + G_2(q) + F_2(\dot{q}) = 0$$

donde

$$D(q) = \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} > 0$$

es la matriz de inercias [13], las funciones vectoriales  $C_1(q, \dot{q}) \in \mathbb{R}^m$  y  $C_2(q, \dot{q}) \in \mathbb{R}^l$  contienen las fuerzas de Coriolis y los términos centrífugos. Las funciones vectoriales  $G_1(q) \in \mathbb{R}^m$  y  $G_2(q) \in \mathbb{R}^l$  contiene los términos gravitacionales; las funciones vectoriales  $F_1(\dot{q}) \in \mathbb{R}^m$  y  $F_2(\dot{q}) \in \mathbb{R}^l$  contienen los términos relativos a la fricción y  $\tau \in \mathbb{R}^m$  representan las fuerzas de entrada generalizadas, producidas por los  $m$  actuadores en las uniones activas [1].

Existen otros sistemas subactuados tales como el robot “brachiation” (cuyo nombre recuerda la características de moverse colgado entre las ramas como los monos) de Fukuda, el péndulo invertido lineal, el péndulo invertido rotacional, el Acrobot, el gimnástico, etc. [1].

El Pendubot posee sus dos ejes paralelos entre sí, con lo que se tiene un acoplamiento mecánico, que no aparece ni en el Péndulo invertido lineal, ni en el péndulo invertido rotacional, haciendo al Pendubot un modelo más interesante y con mayores retos para su control.

## 2.3 Modelo del Pendubot

La figura 2.2 ilustra las diferentes variables y parámetros que intervienen en desarrollo matemático del modelo.

El Pendubot considerado como un sistema no conservativo es representado por

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} + \frac{\partial R}{\partial \dot{q}} = \tau \quad (2.1)$$

donde  $L$  es el langrangiano definido como

$$L = K - V \quad (2.2)$$

$K$  es la energía cinética [6],  $V$  la energía potencial del modelo. Los vectores  $\tau$  y  $q$  representan el par y las posiciones angulares de las uniones de los eslabones en

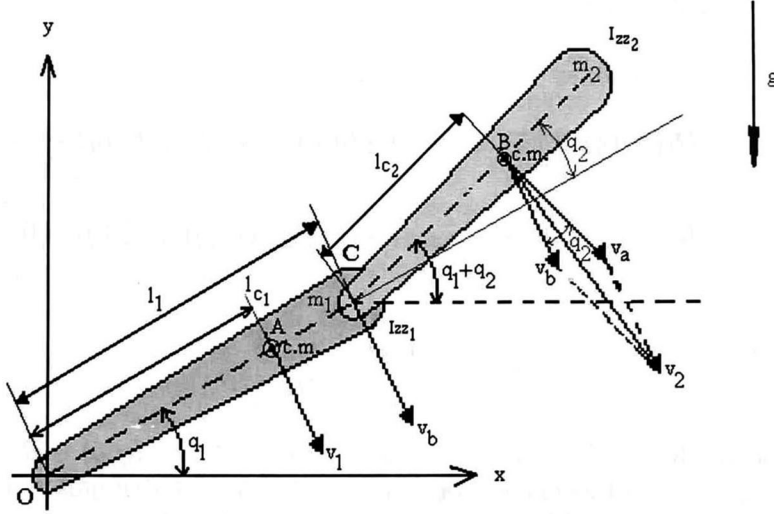


Figura 2.2: Diagrama del Pendubot

forma generalizada respectivamente,  $R$  es la función de disipación de Rayleigh y  $t$  es el tiempo.

A partir de la ecuación (2.1) se pueden obtener las ecuaciones correspondientes a cada eslabón

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_1} \right) - \frac{\partial L}{\partial q_1} + \frac{\partial R}{\partial \dot{q}_1} = \tau_1 \quad (2.3)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_2} \right) - \frac{\partial L}{\partial q_2} + \frac{\partial R}{\partial \dot{q}_2} = \tau_2 = 0 \quad (2.4)$$

donde  $\tau_1$  es el par del motor del Pendubot en la unión 1, y dado que no existe actuador en la unión 2, el par  $\tau_2$  es nulo.

Para la obtención del Lagrangiano de la ecuación (2.2) se requiere conocer el valor de la energía cinética  $K$ , la cual para este caso está dada por [6]:

$$K = \frac{1}{2} (m_1 v_1^2 + I_{zz1} w_1^2 + m_2 v_2^2 + I_{zz2} w_2^2) \quad (2.5)$$

donde  $I_{zz1}$  e  $I_{zz2}$  son los momentos de inercias de los eslabones 1 y 2 referidas a sus centros de masa;  $m_1$  y  $m_2$  son las masas de los eslabones 1 y 2;  $v_1, v_2, w_1, w_2$  son las velocidades lineales y angulares de los eslabones 1 y 2 respectivamente.

Los puntos  $A = (x_1, y_1)$  y  $B = (x_2, y_2)$  de la figura 2.2 representan los centros de masa de los eslabones 1 y 2. Los valores de  $x_1, x_2, y_1, y_2$



son

$$x_1 = l_{c_1} \cos q_1 \quad (2.6)$$

$$x_2 = l_1 \cos q_1 + l_{c_2} \cos (q_1 + q_2) \quad (2.7)$$

$$y_1 = l_{c_1} \sin q_1 \quad (2.8)$$

$$y_2 = l_1 \sin q_1 + l_{c_2} \sin (q_1 + q_2) \quad (2.9)$$

Las velocidades angulares de los eslabones son:

$$w_1 = \dot{q}_1$$

$$w_2 = (\dot{q}_1 + \dot{q}_2)$$

Elevando  $w_1$  y  $w_2$  al cuadrado se obtiene:

$$w_1^2 = \dot{q}_1^2 \quad (2.10)$$

$$w_2^2 = (\dot{q}_1 + \dot{q}_2)^2 \quad (2.11)$$

La velocidad tangencial  $v_1$  del punto A es perpendicular al eje longitudinal del eslabón 1 y está dada por:

$$v_1 = l_{c_1} \dot{q}_1$$

Por conveniencia la elevaremos al cuadrado teniéndose:

$$v_1^2 = (l_{c_1} \dot{q}_1)^2 \quad (2.12)$$

La velocidad tangencial  $v_a$  del punto B, es perpendicular al eje longitudinal del eslabón 2 y está dada por:

$$v_a = l_{c_2} (\dot{q}_1 + \dot{q}_2)$$

Elevándose al cuadrado se obtiene:

$$v_a^2 = (l_{c_2} (\dot{q}_1 + \dot{q}_2))^2$$

La velocidad tangencial de la unión 2, indicada por el punto C, es perpendicular al eje longitudinal del eslabón 1 y está dada por:

$$v_b = l_1 \dot{q}_1$$

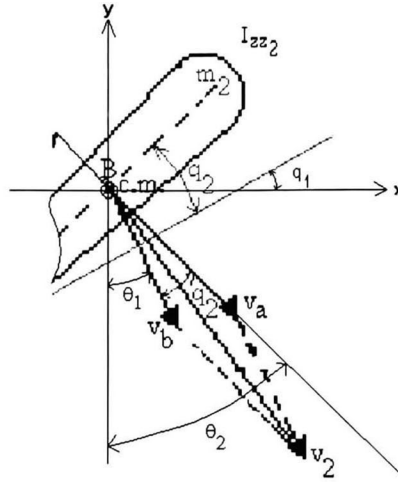


Figura 2.3: Detalle del eslabón 2 del Pendubot

Elevando esta expresión al cuadrado tenemos:

$$v_b^2 = (l_1 \dot{q}_1)^2$$

La velocidad lineal  $v_2$  resultante del eslabón 2 se obtiene de la suma vectorial de las velocidades  $v_a$  y  $v_b$  (ver figura 2.3)

La velocidad resultante  $v_2$  descompuesta en sus componentes  $v_{2x}$  y  $v_{2y}$  es

$$v_2 = \sqrt{v_{2x}^2 + v_{2y}^2}$$

$$v_2^2 = v_{2x}^2 + v_{2y}^2 \quad (2.13)$$

La suma de las velocidades en el eje  $x$  para  $v_{2x} = v_{ax} + v_{bx}$ , y la suma de las componentes de las velocidades en el eje  $y$  es  $v_{2y} = v_{ay} + v_{by}$ , donde

$$v_{ax} = v_a \sin(\theta_2)$$

$$v_{bx} = v_b \sin(\theta_1)$$

$$v_{ay} = v_a \cos(\theta_2)$$

$$v_{by} = v_b \cos(\theta_1)$$

sustituyendo estas ecuaciones en (2.13) se obtiene:

$$v_2^2 = (v_a \sin(\theta_2) + v_b \sin(\theta_1))^2 + (v_a \cos(\theta_2) + v_b \cos(\theta_1))^2$$

desarrollando se llega a

$$\begin{aligned} v_2^2 = & v_a^2 \sin^2(\theta_2) + v_a^2 \cos^2(\theta_2) + v_b^2 \sin^2(\theta_1) + v_b^2 \cos^2(\theta_1) \\ & + 2v_a v_b \sin(\theta_1) \sin(\theta_2) + 2v_a v_b \cos(\theta_1) \cos(\theta_2) \end{aligned}$$

factorizando se tiene

$$\begin{aligned} v_2^2 = & v_a^2 (\sin^2(\theta_2) + \cos^2(\theta_2)) + v_b^2 (\sin^2(\theta_1) + \cos^2(\theta_1)) \\ & + 2v_a v_b (\sin(\theta_1) \sin(\theta_2) + \cos(\theta_1) \cos(\theta_2)) \end{aligned}$$

simplificando

$$v_2^2 = v_a^2 + v_b^2 + 2v_a v_b (\cos(\theta_2) \cos(\theta_1) + \sin(\theta_2) \sin(\theta_1))$$

simplificando nuevamente se obtiene:

$$v_2^2 = v_a^2 + v_b^2 + 2v_a v_b \cos(q_2) \quad (2.14)$$

Para la obtención de la energía cinética sustituimos las ecuaciones (2.10), (2.11), (2.12) y (2.14) en (2.5):

$$\begin{aligned} K = & \frac{1}{2} [m_1 l_{c1}^2 \dot{q}_1^2 + I_{zz1} \dot{q}_1^2 \\ & + m_2 (l_{c2}^2 (\dot{q}_1 + \dot{q}_2)^2 + l_1^2 \dot{q}_1^2 + 2l_1 l_{c2} \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \cos(q_2)) \\ & + I_{zz2} (\dot{q}_1 + \dot{q}_2)^2] \end{aligned}$$

expandiendo la ecuación anterior tenemos:

$$\begin{aligned} K = & \frac{1}{2} [m_1 l_{c1}^2 \dot{q}_1^2 + I_{zz1} \dot{q}_1^2 + m_2 l_{c2}^2 \dot{q}_1^2 + 2m_2 l_{c2}^2 \dot{q}_1 \dot{q}_2 + m_2 l_{c2}^2 \dot{q}_2^2 + m_2 l_1^2 \dot{q}_1^2 \\ & + 2m_2 l_1 l_{c2} \dot{q}_1^2 \cos(q_2) + 2m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \cos(q_2) + I_{zz2} \dot{q}_1^2 \\ & + 2I_{zz2} \dot{q}_1 \dot{q}_2 + I_{zz2} \dot{q}_2^2] \end{aligned}$$

y agrupando convenientemente se obtiene:

$$\begin{aligned} K = & \frac{1}{2} [(m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2}) \dot{q}_1^2 \\ & + 2(m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \dot{q}_1 \dot{q}_2 + (m_2 l_{c2}^2 + I_{zz2}) \dot{q}_2^2] \end{aligned}$$

o bien:

$$K = \frac{1}{2} [D_{11} \dot{q}_1^2 + 2D_{12} \dot{q}_1 \dot{q}_2 + D_{22} \dot{q}_2^2] \quad (2.15)$$



donde:

$$D_{11} = m_1 l_{c_1}^2 + m_2 (l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2} \cos(q_2)) + I_{zz_1} + I_{zz_2}$$

$$D_{12} = m_2 (l_{c_2}^2 + l_1 l_{c_2} \cos(q_2)) + I_{zz_2}$$

$$D_{22} = m_2 l_{c_2}^2 + I_{zz_2}$$

La energía potencial es:

$$V = V_1 + V_2$$

donde:

$$V_1 = m_1 g y_1$$

$$V_2 = m_2 g y_2$$

así:

$$V = m_1 g y_1 + m_2 g y_2 \quad (2.16)$$

sustituyendo las ecuaciones (2.6), (2.7), (2.8), (2.9) en (2.16) la energía potencial del Pendubot es:

$$V = m_1 g l_{c_1} \sin(q_1) + m_2 g (l_1 \sin(q_1) + l_{c_2} \sin(q_1 + q_2)) \quad (2.17)$$

Sustituyendo las ecuaciones de la energía cinética (2.15) y la ecuación de la energía potencial (2.17) en (2.2) se obtiene

$$L = \frac{1}{2} (D_{11} \dot{q}_1^2 + 2D_{12} \dot{q}_1 \dot{q}_2 + D_{22} \dot{q}_2^2) - m_1 g l_{c_1} \sin(q_1) - m_2 g (l_1 \sin(q_1) + l_{c_2} \sin(q_1 + q_2)) \quad (2.18)$$

Por otro lado la función de disipación de Rayleigh es:

$$R(\dot{q}) = \frac{1}{2} (\mu_1 \dot{q}_1^2 + \mu_2 \dot{q}_2^2) \quad (2.19)$$

donde  $\mu_1$  y  $\mu_2$  son los coeficientes de fricción viscosa de las uniones del Pendubot.

La ecuación (2.18) se sustituye en (2.3). Para esto primeramente se obtiene

$$\frac{\partial L}{\partial \dot{q}_1} = D_{11} \dot{q}_1 + D_{12} \dot{q}_2$$

y derivando esta última ecuación con respecto a  $t$  obtenemos el primer miembro de (2.3)

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_1} \right) = D_{11}\ddot{q}_1 + D_{12}\ddot{q}_2 - 2m_2l_1l_{c_2}\dot{q}_1\dot{q}_2 \sin(q_2) - m_2l_1l_{c_2}\dot{q}_2^2 \sin(q_2) \quad (2.20)$$

De la ecuación (2.3) se observa que el segundo miembro es

$$-\frac{\partial L}{\partial q_1} = m_1gl_{c_1} \cos(q_1) + m_2g(l_1 \cos(q_1) + l_{c_2} \cos(q_1 + q_2)) \quad (2.21)$$

Finalmente también se observa que el tercer miembro ecuación (2.3) es

$$\frac{\partial R(\dot{q})}{\partial \dot{q}_1} = \mu_1\dot{q}_1 \quad (2.22)$$

y sustituyendo las ecuaciones (2.20), (2.21) y (2.22) en la ecuación (2.3) obtenemos

$$\tau_1 = D_{11}\ddot{q}_1 + D_{12}\ddot{q}_2 + C_{11}\dot{q}_1 + C_{12}\dot{q}_2 + G_1 + F_1 \quad (2.23)$$

donde

$$C_{11} = -2m_2l_1l_{c_2}\dot{q}_2 \sin(q_2)$$

$$C_{12} = -m_2l_1l_{c_2}\dot{q}_2 \sin(q_2)$$

$$G_1 = m_1gl_{c_1} \cos(q_1) + m_2g(l_1 \cos(q_1) + l_{c_2} \cos(q_1 + q_2))$$

$$F_1 = \mu_1\dot{q}_1$$

De manera similar el Lagrangiano de la ecuación (2.18) se sustituye en la ecuación (2.4). Primeramente se observa que

$$\frac{\partial L}{\partial \dot{q}_2} = D_{12}\dot{q}_1 + D_{22}\dot{q}_2$$

Derivando esta última ecuación con respecto a  $t$  se obtiene el primer miembro de la ecuación (2.4)

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_2} \right) = D_{12}\ddot{q}_1 + D_{22}\ddot{q}_2 - m_2l_1l_{c_2}\dot{q}_1\dot{q}_2 \sin(q_2) \quad (2.24)$$

El segundo miembro de la ecuación (2.4) es

$$-\frac{\partial L}{\partial q_2} = m_2 l_1 l_{c_2} \dot{q}_1^2 \sin(q_2) + m_2 l_1 l_{c_2} \dot{q}_1 \dot{q}_2 \sin(q_2) + m_2 g l_{c_2} \cos(q_1 + q_2) \quad (2.25)$$

Finalmente el tercer miembro de la ecuación (2.4) es

$$\frac{\partial R(\dot{q})}{\partial \dot{q}_2} = \mu_2 \dot{q}_2 \quad (2.26)$$

Sustituyendo las ecuaciones (2.24), (2.25) y (2.26) en la ecuación (2.4) obtenemos

$$0 = D_{12}\ddot{q}_1 + D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + G_2 + F_2 \quad (2.27)$$

donde

$$C_{21} = m_2 l_1 l_{c_2} \dot{q}_1 \sin(q_2)$$

$$G_2 = m_2 g l_{c_2} \cos(q_1 + q_2)$$

$$F_2 = \mu_2 \dot{q}_2$$

Las ecuaciones (2.23) y (2.27)

$$\begin{aligned} D_{11}\ddot{q}_1 + D_{12}\ddot{q}_2 + C_{11}\dot{q}_1 + C_{12}\dot{q}_2 + G_1 + F_1 &= \tau_1 \\ D_{12}\ddot{q}_1 + D_{22}\ddot{q}_2 + C_{21}\dot{q}_1 + C_{22}\dot{q}_2 + G_2 + F_2 &= 0 \end{aligned}$$

pueden escribirse como

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

En forma generalizada se escriben como [7]

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \quad (2.28)$$

donde

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}; \quad \dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}; \quad \ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix},$$

$$D(q) = \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix}, \quad (2.29)$$

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} -2m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) & -m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) \\ m_2 l_1 l_{c2} \dot{q}_1 \sin(q_2) & 0 \end{bmatrix}, \quad (2.30)$$

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} m_1 g l_{c1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2)) \\ m_2 g l_{c2} \cos(q_1 + q_2) \end{bmatrix}, \quad (2.31)$$

$$F = \begin{bmatrix} F_1(\dot{q}) \\ F_2(\dot{q}) \end{bmatrix} = \begin{bmatrix} \mu_1 \dot{q}_1 \\ \mu_2 \dot{q}_2 \end{bmatrix}, \quad (2.32)$$

y

$$\tau = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} \quad (2.33)$$

La ecuación (2.28) describe el modelo dinámico del Pendubot, donde  $D(q)$  es la matriz de inercias (obsérvese que es simétrica),  $C(q, \dot{q})$  es la matriz Coriolis-Centrípeta,  $G(q)$  es el vector gravitacional y  $F(\dot{q})$  representa la acción de la fricción viscosa del sistema. (obsérvese que  $C_{22}=0$ ).

Por otro lado, de la ecuación generalizada (2.28) obtenemos  $\ddot{q}$ :

$$\ddot{q} = D(q)^{-1} [\tau - C(q, \dot{q})\dot{q} - G(q) - F(\dot{q})] \quad (2.34)$$

desarrollando tenemos:

$$\ddot{q} = D(q)^{-1} \left[ \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} - \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} - \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} - \begin{bmatrix} F_1(\dot{q}) \\ F_2(\dot{q}) \end{bmatrix} \right]$$

$$\ddot{q} = D(q)^{-1} \left[ \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} - \begin{bmatrix} C_{11}\dot{q}_1 + C_{12}\dot{q}_2 \\ C_{21}\dot{q}_1 \end{bmatrix} - \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} - \begin{bmatrix} F_1(\dot{q}) \\ F_2(\dot{q}) \end{bmatrix} \right]$$

$$\ddot{q} = D(q)^{-1} \left[ \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} - \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} - \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} - \begin{bmatrix} F_1(\dot{q}) \\ F_2(\dot{q}) \end{bmatrix} \right]$$

con:

$$C_1 = C_{11}\dot{q}_1 + C_{12}\dot{q}_2$$

$$C_2 = C_{21}\dot{q}_1$$

ahora como:

$$D(q)^{-1} = \frac{\begin{bmatrix} D_{22} & -D_{12} \\ -D_{12} & D_{11} \end{bmatrix}}{[D_{11}D_{22} - D_{12}^2]}$$

se tiene

$$\begin{aligned}
 \ddot{q} &= \frac{1}{[D_{11}D_{22} - D_{12}^2]} \left( \begin{bmatrix} D_{22}\tau_1 \\ -D_{12}\tau_1 \end{bmatrix} - \begin{bmatrix} D_{22}C_1 - D_{12}C_2 \\ D_{11}C_2 - D_{12}C_1 \end{bmatrix} \right. \\
 &\quad \left. - \begin{bmatrix} D_{22}G_1 - D_{12}G_2 \\ D_{11}G_2 - D_{12}G_1 \end{bmatrix} - \begin{bmatrix} D_{22}F_1(\dot{q}) - D_{12}F_2(\dot{q}) \\ D_{11}F_2(\dot{q}) - D_{12}F_1(\dot{q}) \end{bmatrix} \right) \\
 \ddot{q} &= \frac{1}{[D_{11}D_{22} - D_{12}^2]} \\
 &\quad \begin{bmatrix} D_{12}C_2 - D_{22}C_1 + D_{12}G_2 - D_{22}G_1 + D_{12}F_2(\dot{q}) - D_{22}F_1(\dot{q}) \\ D_{12}C_1 - D_{11}C_2 + D_{12}G_1 - D_{11}G_2 + D_{12}F_1(\dot{q}) - D_{11}F_2(\dot{q}) \end{bmatrix} \\
 &\quad + \begin{bmatrix} \frac{D_{22}}{[D_{11}D_{12} - D_{12}^2]} \\ \frac{-D_{12}}{[D_{11}D_{12} - D_{12}^2]} \end{bmatrix} \tau_1 \\
 \ddot{q} &= \begin{bmatrix} \frac{D_{22}}{[D_{11}D_{22} - D_{12}^2]} \left( \frac{D_{12}C_2}{D_{22}} + \frac{D_{12}G_2}{D_{22}} + \frac{D_{12}F_2(\dot{q})}{D_{22}} - C_1 - G_1 - F_1(\dot{q}) \right) \\ \frac{-D_{12}}{[D_{11}D_{22} - D_{12}^2]} \left( \frac{D_{11}C_2}{D_{12}} + \frac{D_{11}G_2}{D_{12}} + \frac{D_{11}F_2(\dot{q})}{D_{12}} - C_1 - G_1 - F_1(\dot{q}) \right) \end{bmatrix} \\
 &\quad + \begin{bmatrix} \frac{D_{22}}{[D_{11}D_{12} - D_{12}^2]} \\ \frac{-D_{12}}{[D_{11}D_{12} - D_{12}^2]} \end{bmatrix} \tau_1 \tag{2.35}
 \end{aligned}$$

De la ecuación (2.35) se obtiene el modelo no lineal representado en variables de estado con:

$$x_1 = q_1, \quad x_2 = q_2, \quad x_3 = \dot{q}_1, \quad x_4 = \dot{q}_2, \quad u = \tau_1, \quad y = x_2$$

el sistema no lineal queda representado por:

$$\dot{x} = f(x) + g(x)u(t) = F(x(t), u(t), t)$$

$$y(t) = h(x)$$

donde:

$$f(x) = \begin{bmatrix} x_3 \\ x_4 \\ \frac{D_{22}}{D_{11}D_{22} - D_{12}^2} \left\{ \frac{D_{12}C_2}{D_{22}} + \frac{D_{12}G_2}{D_{22}} + \frac{D_{12}F_2}{D_{22}} - C_1 - G_1 - F_1 \right\} \\ \frac{-D_{12}}{D_{11}D_{22} - D_{12}^2} \left\{ \frac{D_{11}C_2}{D_{12}} + \frac{D_{11}G_2}{D_{12}} + \frac{D_{11}F_2}{D_{12}} - C_1 - G_1 - F_1 \right\} \end{bmatrix} \tag{2.36}$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{D_{22}}{D_{11}D_{22}-D_{12}^2} \\ \frac{-D_{12}}{D_{11}D_{22}-D_{12}^2} \end{bmatrix} \quad (2.37)$$

$$h(x) = x_2 \quad (2.38)$$



# Capítulo 3

## Parámetros

---

**Resumen:** En este capítulo se obtienen los valores de las inercias de los eslabones del Pendubot, los coeficientes de fricción y la constante de par del motor.

---



### 3.1 Introducción

Con el fin de obtener simulaciones con el modelo matemático acordes al desempeño del sistema real, es necesario contar con valores satisfactorios de los **parámetros** del Pendubot. Tales parámetros son: masa, inercia, longitud de los eslabones, fricción de las uniones, distancias de los centros de gravedad de los eslabones a los ejes de giro (uniones) respectivos. Para la obtención de los valores de los parámetros del Pendubot, se requiere analizar el comportamiento de cada eslabón del Pendubot actuando libremente.

Al inicio de esta investigación se emplearon los valores de los parámetros citados de la tesis de Ramos [7], en particular el valor de los coeficientes de fricción viscosa de las uniones 1 y 2 del Pendubot no estaban disponibles, así como la constante de par del motor [13], la cual es muy importante para el desarrollo del programa en la PC que controla el Pendubot. Al simular los eslabones del Pendubot como péndulos físicos empleando los valores de los parámetros de la tesis de Ramos [7] se observó que el período del primer eslabón tenía un error considerable con respecto a la realidad y que las ganancias  $LQ$  citadas requerían de ser ajustadas en el programa de control de la computadora personal para controlar al prototipo (por ejemplo  $K_1$  era de 20 y finalmente quedaba en 25). Lo anterior llevó al autor a proponer en este capítulo algunos procedimientos para determinar dichos valores.

## 3.2 Obtención de parámetros

### 3.2.1 Momentos de inercia

#### Mediante el período pendular

La ecuación del péndulo físico simple para ángulos pequeños [9]

$$T_i = 2\pi \sqrt{\frac{J_{zz_i}}{m_i g l_{c_i}}}; \quad i = 1, 2 \quad (3.1)$$

permite determinar el período de oscilación que tiene un péndulo físico (o eslabón de un robot por ejemplo) si se conoce la masa  $m_i$ , la distancia del centro de gravedad al eje de giro (unión) del péndulo  $l_{c_i}$ , el momento de inercia referido a eje de giro del péndulo  $J_{zz_i}$  y la constante de aceleración de la gravedad  $g$ .

Si lo que se desea obtener es el momento de inercia del péndulo  $J_{zz_i}$ , éste se obtiene de la ecuación (3.1) y es

$$J_{zz_i} = \frac{1}{4} \frac{T_i^2}{\pi^2} m_i g l_{c_i} \quad (3.2)$$

La inercia referida a su centro de gravedad  $I_{zz_i}$  se obtiene aplicando el teorema de los ejes paralelos [9]

$$J_{zz_i} = I_{zz_i} + m_i l_{c_i}^2 \quad (3.3)$$

igualando (3.2) y (3.3) se obtiene:

$$\frac{1}{4} \frac{T_i^2}{\pi^2} m_i g l_{c_i} = I_{zz_i} + m_i l_{c_i}^2 \quad (3.4)$$

despejando  $I_{zz_i}$  se obtiene

$$I_{zz_i} = \left( \frac{g T_i^2}{4 \pi^2} - l_{c_i} \right) m_i l_{c_i} \quad (3.5)$$

La ecuación (3.5) permite determinar los valores de los momentos de inercia de los eslabones del Pendubot.

Para obtener el período pendular  $T_2$ , el eslabón 1 se fija firmemente y se deja oscilar libremente el eslabón 2. Para obtener el período  $T_1$  del eslabón 1, éste se deja oscilar libremente (quitando previamente el eslabón 2). El período de oscilación puede medirse con un cronómetro o aparato de medición de períodos pendulares. Los valores de los períodos  $T_1$  y  $T_2$  obtenidos experimentalmente se sustituyen en la ecuación (3.5) para calcular los momentos de inercia de los eslabones 1 y 2 respectivamente.

Para determinar los centros de masa de cada eslabón  $l_{c_i}$  (previamente desensamblados del Pendubot), se equilibran los eslabones sobre un punto de apoyo como se indica en la figura 3.1 y 3.2, luego se toma la distancia de este punto a la unión correspondiente. Para obtener el peso de los eslabones se desensambla el Pendubot y se pesa separadamente cada eslabón, el eslabón 1 incluye la masa del codificador 2.

Los valores que se obtuvieron de  $l_{c_1}$  y  $l_{c_2}$  fueron

$$l_1 = 0.2032 \text{ m.}$$

$$l_{c_1} = 0.1551 \text{ m.}$$

$$l_{c_2} = 0.1635 \text{ m.}$$

Las masas de los respectivos eslabones del Pendubot fueron

$$m_1 = 0.8293 \text{ kg.}$$

$$m_2 = 0.3402 \text{ kg.}$$

Los períodos de estos eslabones fueron

$$T_1 = 0.8929 \text{ seg.}$$

$$T_2 = 0.9773 \text{ seg.}$$

Con  $g = 9.81 \frac{\text{m}}{\text{seg}^2}$ , de la ecuación (3.4) se obtiene

$$J_{zz_1} = \frac{g T_1^2}{4 \pi^2} m_1 l_{c_1} = 2.5473 \times 10^{-2} \text{ kg} - \text{m}^2$$

$$J_{zz_2} = \frac{g T_2^2}{4 \pi^2} m_2 l_{c_2} = 1.3201 \times 10^{-2} \text{ kg} - \text{m}^2$$

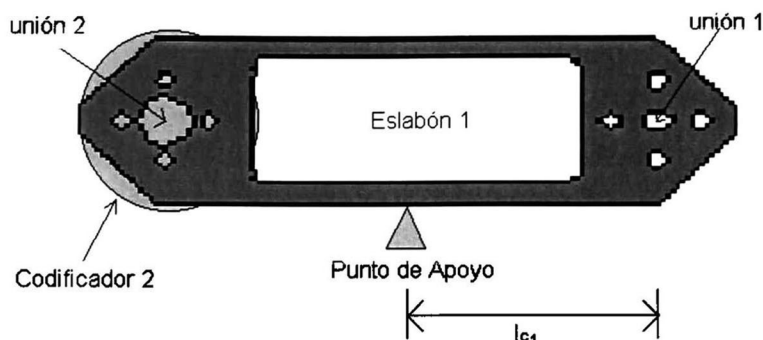


Figura 3.1: Determinación del centroide del eslabón 1

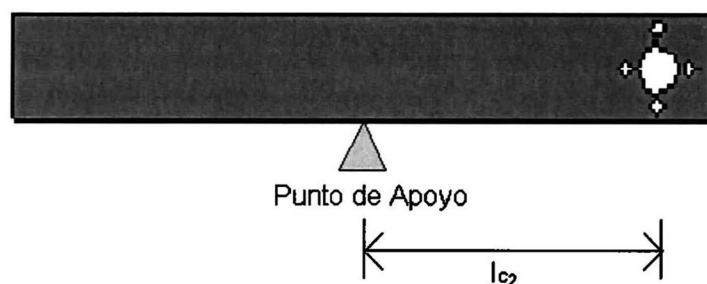


Figura 3.2: Determinación del centroide del eslabón 2

o bien de la ecuación (3.5) se obtienen los valores  $I_{zz1}$  e  $I_{zz2}$ :

$$I_{zz1} = \left( \frac{gT_1^2}{4\pi^2} - l_{c1} \right) m_1 l_{c1} = 5.5326 \times 10^{-3} \text{ kg} - \text{m}^2$$

$$I_{zz2} = \left( \frac{gT_2^2}{4\pi^2} - l_{c2} \right) m_2 l_{c2} = 4.1066 \times 10^{-3} \text{ kg} - \text{m}^2$$

### Determinación de las inercias analíticamente

Otra manera de calcular las inercias  $J_{zz1}$  y  $J_{zz2}$  se realizó teóricamente descomponiendo cada eslabón del Pendubot en figuras geométricas básicas y calculando las inercias de cada área individualmente según Beer [8].

Los resultados obtenidos fueron:

$$J_{zz_1} = 0.0259 \text{ kg} - m^2$$

$$J_{zz_2} = 0.0134 \text{ kg} - m^2$$

o bien

$$I_{zz_1} = 0.00522153318883 \text{ kg} - m^2$$

$$I_{zz_2} = 0.00430568855000 \text{ kg} - m^2$$

El procedimiento para la obtención de las inercias de los eslabones por este método es de fácil comprensión, pero resulta ser laborioso, especialmente cuando se obtiene un gran número de áreas, como en el caso del eslabón 1. La manera de obtener la inercia de los eslabones no es única, ya que se puede calcular la inercia de un eslabón, considerando que el área total de éste, sea todo lo que abarca su contorno externo y restarle a este cálculo, los momentos de inercia de las áreas correspondientes a las partes huecas del eslabón.

### Momentos de inercias por computadora

En esta sección los momentos de inercia se obtuvieron empleando un paquete de "software": AutoCad [17]. Para la obtención de los momentos de inercia se dibujaron los eslabones 1 y 2 del Pendubot en AutoCad y se obtuvieron los valores de sus masas, sus distancias a los centros de gravedad de los ejes de giro y sus momentos de inercia. Los valores obtenidos fueron:

$$m_2 = 0.3345967 \text{ kg.}$$

$$l_{c2} = 0.0164126 \text{ m}$$

$$I_{zz_2} = 0.0041693 \text{ kg} - m^2$$

$$J_{zz_2} = 0.01318 \text{ kg} - m^2$$

La inercia del eslabón 1 sin considerar el codificador arrojó los siguientes valores:

$$m_1 = 255.618 \text{ g.}$$

$$l_{c1} = 0.103958 \text{ m}$$

$$I_{zz_1} = 0.0020138 \text{ kg} - m^2$$

$$J_{zz_1} = 0.0047763 \text{ kg} - m^2$$

A continuación se muestra una tabla que resume los valores obtenidos para las inercias de las uniones del Pendubot. La primera columna indica el método utilizado, la segunda y tercera columna indican los momentos de inercias de los eslabones 1 y 2 referidos al centro de gravedad, la cuarta y quinta columna indican los momentos de inercia de los eslabones 1 y 2 referidos al eje de giro (unión).



Tabla 3.1.- Valores de los parámetros del Pendubot por varios métodos

Método	$I_{zz1}$	$I_{zz2}$	$J_{zz1}$	$J_{zz2}$
Experimental	0.005532	0.004106	0.02558	0.01320
Analítico	0.005221	0.004305	0.02590	0.01340
Por computadora	—	0.004169	—	0.01318

### 3.2.2 Coeficientes de fricción

Para la obtención de las fricciones de las uniones de los eslabones 1 y 2, fue necesario contar con los valores de tiempo y amplitud máxima del movimiento pendular de los eslabones 1 y 2 del sistema real, para lo cual se empleó un programa en una computadora personal que capturó los valores de  $q_1$ ,  $q_2$  con sus respectivos valores de tiempo, véase las gráficas de las figuras 3.3 y 3.4.

El resultado de estas gráficas (valores en tiempo real) se comparó con las gráficas resultantes de las simulaciones hechas en Simnon [20], empleando el modelo deducido en el capítulo anterior con los parámetros obtenidos experimentalmente en esta sección (inicialmente los valores de las fricciones fueron cero). Con estas simulaciones se ajustaron los valores de las fricciones de los eslabones en el modelo matemático empleado en Simnon, hasta obtener la máxima aproximación de las gráficas en tiempo real y las simuladas, ver figura 3.3 y 3.4.

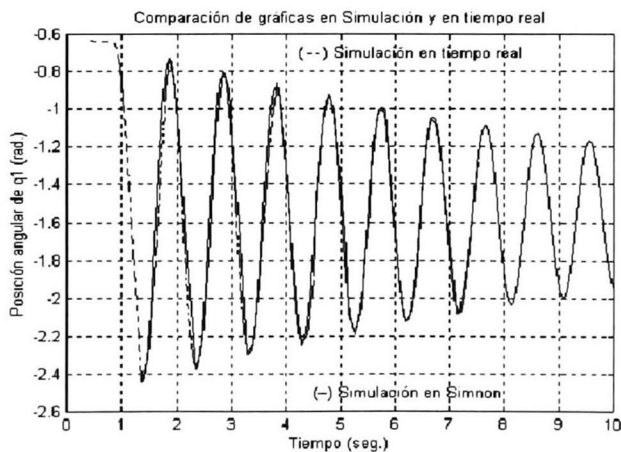


Figura 3.3: Comparación de los resultados experimentales con los resultados de simulación del comportamiento pendular del eslabón 1 para la obtención de su fricción.

Los valores de las fricciones  $\mu_1$  y  $\mu_2$  de los eslabones 1 y 2 respectivamente

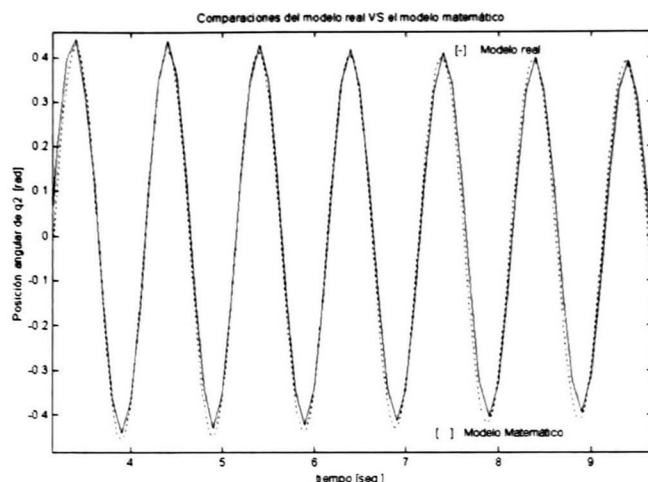


Figura 3.4: Comparación de los resultados experimentales con los resultados de simulación del comportamiento pendular del eslabón 2 para la obtención de su fricción.

fueron:

$$\mu_1 = 0.00545 \frac{kg}{seg}$$

$$\mu_2 = 0.00047 \frac{kg}{seg}$$

### 3.2.3 Ganancia del par del motor

La constante del par del motor es de fundamental importancia especialmente cuando se desea controlar el modelo real, ya que es necesario conocer la relación existente entre el par calculado y el valor digital que debe aplicarse a la tarjeta convertidora digital/análogo. Para la obtención del par del motor se removió el eslabón 2 y el codificador 2 del Pendubot, posteriormente se aplicó un valor digital a la tarjeta convertidora de analógico a digital (colocada dentro de la computadora personal), de manera que produjera una variación de la corriente del motor desde  $-8$  a  $8$  amperes, simultáneamente, mediante el empleo de un dinamómetro se tomaron lecturas del par producido en el extremo del eslabón 1 del Pendubot (unión 2). Con la medición de esta fuerza, se obtuvo la gráfica de la figura 3.5, que relaciona la corriente aplicada al motor y el par resultante de éste. Para la obtención del par del motor se consideró que  $\tau = F * l_1$ , donde  $F$  es la fuerza medida por el dinamómetro a una distancia  $l_1$ . De la gráfica

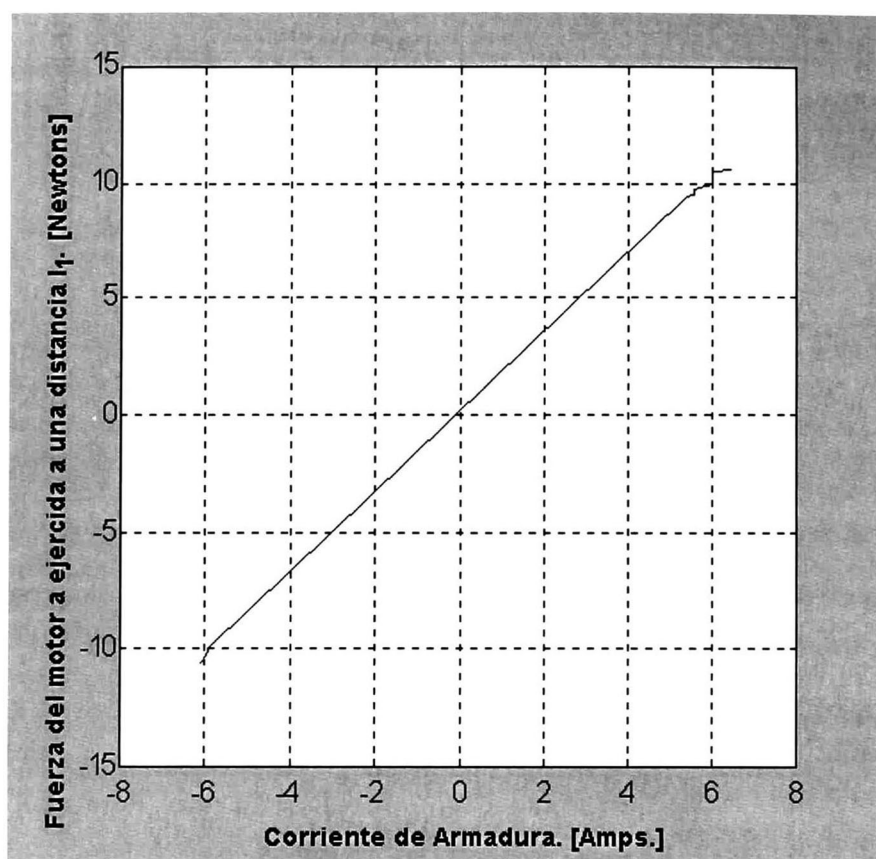


Figura 3.5: Relación de la corriente de armadura y la fuerza del motor del Pendubot a una distancia  $l_1$ .

3.5, se obtiene que la fuerza del motor a una distancia  $l_1$  ( $l_1 = 0.2032 \text{ m.}$ ) es

$$F = 1.6934 * i_a$$

pero

$$\tau = F * l_1 = (1.6934 * i_a) * l_1$$

$$\tau = 0.3441 * i_a$$

donde  $\tau$  es el par del motor e  $i_a$  es la corriente de armadura del motor en amperes. Ya que el par del motor se define como [13]

$$\tau = K_m * i_a$$

El valor de la ganancia del par del motor  $K_m$  será

$$K_m = 0.3441 \text{ N-m/A} \quad (3.6)$$

que es el valor buscado.

Otro procedimiento para determinar la constante de par del motor se cita en [19].





# Capítulo 4

## Modelización lineal

---

**Resumen:** En el presente capítulo se determinan los modelos lineales que describen al Pendubot en distintos puntos de equilibrio. Estos modelos serán utilizados para el diseño de los controladores lineales mas adelante.

---

## 4.1 Introducción

En los capítulos anteriores se determinó el modelo matemático no lineal del Pendubot y los valores de los parámetros requeridos para dicho modelo, contándose así con una representación matemática bastante buena del comportamiento del sistema físico. Con este modelo matemático no lineal, se obtienen los modelos linealizados en distintos puntos (ver figura 4.1) de equilibrio mediante linealización por series de Taylor. Cada modelo linealizado obtenido es en tiempo continuo. Debido a que la implementación de los controladores del Pendubot se realiza en tiempo real, es necesario discretizar cada modelo lineal, estos modelos lineales discretizados permiten determinar las ganancias del *LQR* (regulador lineal cuadrático) en tiempo discreto que estabilizan el Pendubot en distintos puntos de operación.

## 4.2 Puntos de equilibrio del Pendubot

Las posiciones de equilibrio  $q_1^0$  y  $q_2^0$  del Pendubot, deben cumplir con la condición  $q_1^0 + q_2^0 = n\frac{\pi}{2} = 90^\circ n$ ;  $\forall n = \pm 1, \pm 3, \pm 5, \pm 7 \dots \pm (2m - 1)$ ;  $m \in \mathbb{N}$  [7], o bien  $q_2^0 = n\frac{\pi}{2} - q_1^0$ . El análisis del comportamiento del Pendubot será con balanceo hacia arriba (ver figura 4.1), considerando por conveniencia  $n = 1$ . De

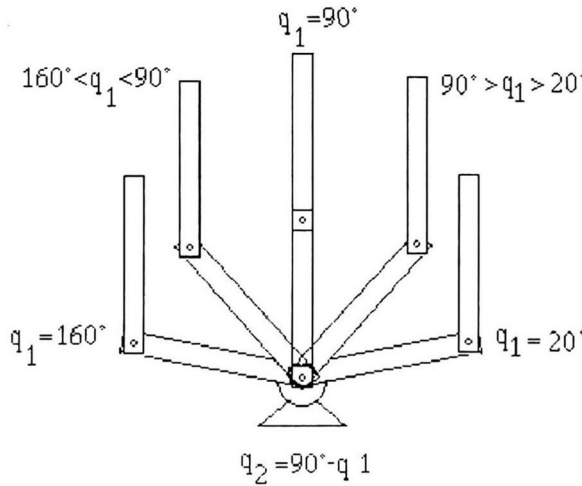


Figura 4.1: Puntos de equilibrio del Pendubot.

la figura 4.1 se observa que la  $k$ -ésima posición de equilibrio está en el rango  $160^\circ \geq k \geq 20^\circ$ . Puesto que en cualquier punto de equilibrio  $\dot{x}(t) = 0$ , de (2.34) tenemos que

$$\ddot{q} = D(q)^{-1} [\tau_k^0 - C(q, \dot{q})\dot{q} - G(q) - F(\dot{q})] = 0$$

donde  $\tau_k^0$  el par nominal en de  $k$ -ésima posición tal que  $\ddot{q} = 0$ , esto implica que

$$\tau_k^0 = G(q)$$

como

$$\tau_k^0 = \begin{bmatrix} \tau_1^0 \\ 0 \end{bmatrix}$$

y

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} m_1 g l_{c_1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c_2} \cos(q_1 + q_2)) \\ m_2 g l_{c_2} \cos(q_1 + q_2) \end{bmatrix}$$

entonces

$$\begin{bmatrix} \tau_1^0 \\ 0 \end{bmatrix} = \begin{bmatrix} m_1 g l_{c_1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c_2} \cos(q_1 + q_2)) \\ m_2 g l_{c_2} \cos(q_1 + q_2) \end{bmatrix}$$

dado que  $q_1 + q_2 = \frac{\pi}{2}$

$$\cos(q_1 + q_2) = 0$$

despejando para  $\tau_1^0$  se obtiene

$$\tau_1^0 = (m_1 l_{c_1} + m_2 l_1) g \cos(q_1) \quad (4.1)$$

donde  $\tau_k^0 = \mathfrak{S}_k^0 = \begin{bmatrix} \tau_1^0 \\ 0 \end{bmatrix}$  es el par de equilibrio nominal del Pendubot en el  $k$ -ésimo punto de equilibrio, este par ( $\mathfrak{S}_k^0$ ) es muy importante cuando se desea controlar el pendubot en las posiciones de equilibrio tal que  $\cos(q_1^0) \neq 0$ .

### 4.3 Linealización

En esta sección se determinan los diferentes modelos lineales que aproximan el comportamiento del Pendubot en los diferentes puntos de operación elegidos. Para la obtención de estos modelos se realiza la linealización mediante una aproximación por series de Taylor de primer orden.

El modelo no lineal del Pendubot es de la forma:

$$\dot{x}(t) = f(x) + g(x)u(t) = F(x(t), u(t), t)$$

suponiendo que

$$\begin{aligned} x(t) &= x^0(t) + \delta x(t) \\ u(t) &= u^0(t) + \delta u(t) \end{aligned} \quad (4.2)$$

la expansión en series de Taylor es

$$F(x(t), u(t), t) = F(x_0(t), u_0(t), t) + F_x(t) \delta x(t) + F_u(t) \delta u(t) + o(\delta x, \delta u) \quad (4.3)$$

donde

$$F_x(t) = \frac{\partial F(x(t), u(t), t)}{\partial x} \Big|_{x_0, u_0}, \quad F_u(t) = \frac{\partial F(x(t), u(t), t)}{\partial u} \Big|_{x_0, u_0} \quad (4.4)$$

y  $o(\delta x, \delta u)$  representa los términos de orden superior en  $x - x_0$  [10]. Si se desprecian los términos de orden superior y

$$\delta \dot{x} = F(x(t), u(t), t) - F(x_0(t), u_0(t), t)$$

se tiene

$$\delta \dot{x} = F_x \delta x + F_u \delta u$$

definiendo  $A(t) \triangleq F_x(t)$ ;  $B(t) \triangleq F_u(t)$  y redefiniendo por simplicidad a  $x \triangleq \delta x$ ,  $u \triangleq \delta u$

$$\dot{x}(t) = Ax(t) + Bu(t)$$

Evaluando

$$A = F_x(t) = \frac{\partial F}{\partial x} \Big|_{x_0, u_0} = \frac{\partial (f(x) + g(x)u(t))}{\partial x} \Big|_{x_0, u_0}$$

$$B = F_u(t) = \frac{\partial F}{\partial u} \Big|_{x_0, u_0} = \frac{\partial (f(x) + g(x)u(t))}{\partial u} \Big|_{x_0, u_0}$$

recordando de las ecuaciones (2.36), (2.37), y (2.38) que

$$f(x) = \begin{bmatrix} x_3 \\ x_4 \\ \frac{D_{22}}{D_{11}D_{22}-D_{12}^2} \left\{ \frac{D_{12}C_2}{D_{22}} + \frac{D_{12}G_2}{D_{22}} + \frac{D_{12}F_2}{D_{22}} - C_1 - G_1 - F_1 \right\} \\ \frac{-D_{12}}{D_{11}D_{22}-D_{12}^2} \left\{ \frac{D_{11}C_2}{D_{12}} + \frac{D_{11}G_2}{D_{12}} + \frac{D_{11}F_2}{D_{12}} - C_1 - G_1 - F_1 \right\} \end{bmatrix} \triangleq \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{D_{22}}{D_{11}D_{22}-D_{12}^2} \\ \frac{-D_{12}}{D_{11}D_{22}-D_{12}^2} \end{bmatrix} := \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}$$

entonces

$$A = \begin{bmatrix} 0 & 0 & \frac{\partial(x_3)}{\partial x_3} = 1 & 0 \\ 0 & 0 & 0 & \frac{\partial(x_4)}{\partial x_4} = 1 \\ \frac{\partial(f_3+g_3)}{\partial x_1} & \frac{\partial(f_3+g_3)}{\partial x_2} & \frac{\partial(f_3+g_3)}{\partial x_3} & \frac{\partial(f_3+g_3)}{\partial x_4} \\ \frac{\partial(f_4+g_4)}{\partial x_1} & \frac{\partial(f_4+g_4)}{\partial x_2} & \frac{\partial(f_4+g_4)}{\partial x_3} & \frac{\partial(f_4+g_4)}{\partial x_4} \end{bmatrix} \quad (4.5)$$

y

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial(f_3+g_3)}{\partial u} \\ \frac{\partial(f_4+g_4)}{\partial u} \end{bmatrix} \quad (4.6)$$

puesto que la salida del sistema no lineal es

$$h(x) = x_2$$

la salida del sistema lineal es

$$y = Cx = x_2$$

donde

$$C = [0 \ 1 \ 0 \ 0]$$

con esto finalmente tenemos definido el sistema lineal como:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

Mediante el empleo de un programa desarrollado en Matlab [22] (ver apéndice (B)) se obtuvieron los valores de los elementos  $a_{i,j}$ ,  $b_i$  de  $A_k$ ,  $B_k$ , siendo  $A_k$ ,  $B_k$  las matrices del modelo lineal del  $k$ -ésimo punto de operación, donde  $k = 20^\circ, 30^\circ, 40^\circ, \dots, 160^\circ$ . Estos valores se muestran en las tablas 4.1 y 4.2. Puesto que los modelos linealizados para  $20^\circ \leq k \leq 90^\circ$  de  $q_1$  son idénticos a los evaluados en  $160^\circ \geq k \geq 90^\circ$ , solamente se determinarán los valores de los elementos de  $A_k$  y  $B_k$  para  $20^\circ \leq k \leq 90^\circ$ .

Tabla 4.1.- Elementos de  $A_k$  y  $B_k$ 

$k$	$20^\circ(160^\circ)$	$30^\circ(150^\circ)$	$40^\circ(140^\circ)$	$50^\circ(130^\circ)$
$a(1,1)$	0	0	0	0
$a(1,2)$	0	0	0	0
$a(1,3)$	1	1	1	1
$a(1,4)$	0	0	0	0
$a(2,1)$	0	0	0	0
$a(2,2)$	0	0	0	0
$a(2,3)$	0	0	0	0
$a(2,4)$	1	1	1	1
$a(3,1)$	13.0323	19.6953	26.4139	32.9960
$a(3,2)$	-4.0544	-6.1272	-8.2174	-10.2651
$a(3,3)$	-0.1403	-0.1451	-0.1514	-0.1587
$a(3,4)$	0.0156	0.0178	0.0201	0.0225
$a(4,1)$	23.9317	12.7217	-0.0119	-13.5937
$a(4,2)$	45.9480	49.4354	53.3969	57.6222
$a(4,3)$	0.1808	0.2063	0.2334	0.2612
$a(4,4)$	-0.0552	-0.0604	-0.0661	-0.0722
$b(1)$	0	0	0	0
$b(2)$	0	0	0	0
$b(3)$	25.7522	26.6218	27.7721	29.1107
$b(4)$	-33.1819	-37.8500	-42.8305	-47.9216



Tabla 4.2.- Elementos de  $A_k$  y  $B_k$ 

$k$	$60^\circ(120^\circ)$	$70^\circ(110^\circ)$	$80^\circ(100^\circ)$	$90^\circ$
$a(1, 1)$	0	0	0	0
$a(1, 2)$	0	0	0	0
$a(1, 3)$	1	1	1	1
$a(1, 4)$	0	0	0	0
$a(2, 1)$	0	0	0	0
$a(2, 2)$	0	0	0	0
$a(2, 3)$	0	0	0	0
$a(2, 4)$	1	1	1	1
$a(3, 1)$	39.0722	44.1014	47.4651	48.6530
$a(3, 2)$	-12.1554	-13.7200	-14.7665	-15.1360
$a(3, 3)$	-0.1662	-0.1729	-0.1775	-0.1792
$a(3, 4)$	0.0248	0.0267	0.0280	0.0285
$a(4, 1)$	-26.8915	-38.3352	-46.1715	-48.9696
$a(4, 2)$	61.7592	65.3193	67.7572	68.6277
$a(4, 3)$	0.2876	0.3099	0.3250	0.3304
$a(4, 4)$	-0.0780	-0.0830	-0.0864	-0.0876
$b(1)$	0	0	0	0
$b(2)$	0	0	0	0
$b(3)$	30.4917	31.7184	32.5737	32.8816
$b(4)$	-52.7666	-56.8604	-59.6334	-60.6185

## 4.4 Discretización

Para implementar en tiempo real el algoritmo de control del Pendubot, es necesario determinar las ecuaciones de estado discretas partiendo de las ecuaciones de estado continuas.

Con los modelos lineales obtenidos en la sección anterior se obtiene la ecuación de estados para tiempo discreto [24]

$$\begin{aligned}x((k+1)T_s) &= G(T) x(kT_s) + Hu(kT_s) \\ y(k) &= Cx(kT_s)\end{aligned}$$

donde

$$\begin{aligned}G(T_s) &= e^{AT_s} \\ H(T_s) &= \left( \int_0^{T_s} e^{A\lambda} d\lambda \right) B; \quad \text{con } \lambda = T_s - t \\ \text{y } T_s &\text{ es el período de muestreo}\end{aligned} \tag{4.7}$$

Si la matriz  $A$  es no singular, entonces  $H(T_s)$  dada por la ecuación (4.7) se puede simplificar a

$$H(T_s) = (e^{AT_s} - I)A^{-1}B$$



### 4.4.1 Método computacional

Para la obtención de los valores discretos de  $G$  y  $H$  se utilizó el comando *c2d* de Matlab, los valores de los elementos  $g_{(i,j)}$  y  $h_{(i,j)}$  de  $G$  y  $H$  obtenidos, se muestran en la tabla 4.3 y 4.4. El período de muestreo es  $T_s = 5$  ms.

Tabla 4.3.- Elementos de  $G_k$  y  $H_k$

$k$	$20^\circ(160^\circ)$	$30^\circ(150^\circ)$	$40^\circ(140^\circ)$	$50^\circ(130^\circ)$
$g(1,1)$	1.00020,	1.0002	1.0003	1.0004
$g(1,2)$	-0.0001	-0.0001	-0.0001	-0.0001
$g(1,3)$	0.0050	0.0050	0.0050	0.0050
$g(1,4)$	0.0000	0.0000	0.0000	0.0000
$g(2,1)$	0.0003	0.0002	-0.0000	-0.0002
$g(2,2)$	1.0006	1.0006	1.0007	1.0007
$g(2,3)$	0.0000	0.0000	0.0000	0.0000
$g(2,4)$	0.005	0.0050	0.0050	0.0050
$g(3,1)$	0.0651	0.0985	0.1320	0.1649
$g(3,2)$	-0.0203	-0.0306	-0.0411	-0.0513
$g(3,3)$	0.9995	0.9995	0.9996	0.9996
$g(3,4)$	0.0000	0.0000	-0.0000	-0.0000
$g(4,1)$	0.1197	0.0637	0.0000	-0.0679
$g(4,2)$	0.2297	0.2472	0.2670	0.2881
$g(4,3)$	0.0012	0.0012	0.0012	0.0011
$g(4,4)$	1.0003	1.0003	1.0003	1.0004
$h(1)$	0	0	0	0
$h(2)$	0	0	0	0
$h(3)$	0.1287	0.1331	0.1388	0.1455
$h(4)$	-0.1658	-0.1892	-0.2141	-0.2395

Tabla 4.4.- Elementos de  $G_k$  y  $H_k$ 

$k$	$60^\circ(120^\circ)$	$70^\circ(110^\circ)$	$80^\circ(100^\circ)$	$90^\circ$
$g(1,1)$	1.0005	1.0006	1.0006	1.0006
$g(1,2)$	-0.0002	-0.0002	-0.0002	-0.0002
$g(1,3)$	0.0050	0.0050	0.0050	0.0050
$g(1,4)$	0.0000	0.0000	0.0000	0.0000
$g(2,1)$	-0.0003	-0.0005	-0.0006	-0.0006
$g(2,2)$	1.0008	1.0008	1.0008	1.0009
$g(2,3)$	0.0000	0.0000	0.0000	0.0000
$g(2,4)$	0.0050	0.0050	0.0050	0.0050
$g(3,1)$	0.1953	0.2205	0.2373	0.2432
$g(3,2)$	-0.0608	-0.0686	-0.0738	-0.0757
$g(3,3)$	0.9997	0.9997	0.9997	0.9997
$g(3,4)$	-0.0000	-0.0000	-0.0000	-0.0000
$g(4,1)$	-0.1343	-0.1916	-0.2307	-0.2447
$g(4,2)$	0.3088	0.3266	0.3388	0.3431
$g(4,3)$	0.0011	0.0011	0.0010	0.0010
$g(4,4)$	1.0004	1.0004	1.0004	1.0004
$h(1)$	0.0004	0.0004	0.0004	0.0004
$h(2)$	-0.0007	-0.0007	-0.0007	-0.0008
$h(3)$	0.1524	0.1585	0.1628	0.1644
$h(4)$	-0.2638	-0.2842	-0.2981	-0.3030



# Capítulo 5

## Síntesis del Control

---

**Resumen:** En este capítulo se obtienen las ganancias  $LQ$  discretas pre-programadas que satisfacen las especificaciones de control requeridas en el Pendubot y se describe el algoritmo empleado para la conmutación de los controladores.

---

## 5.1 LQ discreto

Sea un sistema lineal discreto de la forma [23] [25]:

$$\begin{aligned}x(k+1) &= Gx(k) + Hu(k) \\ x_0 &= x(t_0) = \dot{x}\end{aligned}$$

y el criterio de desempeño

$$J = \lim_{k \rightarrow \infty} \sum_{t=t_0}^k [x^T(k) Q x(k) + u^T(k) R u(k)]$$

bajo la hipótesis de que

$$\begin{aligned}[G, H] &\text{ es estabilizable} \\ [G, Q] &\text{ es detectable} \\ Q &= Q^T \geq 0 \\ R &= R^T \geq 0\end{aligned}$$

entonces la ley de control que minimiza  $J$  es:

$$u^*(k) = -Kx(k)$$

con  $K$  dada por

$$K = [H^T S H + R]^{-1} H^T S G \quad (5.1)$$

donde  $S$  es la solución de la ecuación de Ricatti

$$G^T S G - S - G^T S H [R + H^T S H]^{-1} H^T S G + Q = 0 \quad (5.2)$$

## 5.2 Especificaciones

Al observar el comportamiento en simulación del modelo matemático del Pendubot se proponen los siguientes valores:

- El tiempo de estabilización máximo es de 3 segundos con un error máximo de  $\pm 4^\circ$  en el rango de  $20^\circ$  hasta  $160^\circ$ , para  $q_1$  y  $q_2 = q_1 - 90^\circ$
- El par instantáneo aplicado al Pendubot no deberá exceder el par máximo  $\tau_{m\acute{a}x} = 2.75$  N-m del motor

## 5.3 Ganancias preprogramadas

Para la síntesis del controlador, se emplearon los modelos discretos de las tablas 4.3 y 4.4 del capítulo anterior. Después de varias pruebas realizadas en simulación se observó que los siguientes parámetros de síntesis

$$Q = C^T C$$

$$R = 1$$

satisfican las especificaciones requeridas, además de brindar adecuada robustez para cada controlador  $LQ$  calculado. Gracias a esta robustez se pudo apreciar, en estudios de simulación, que al conmutar los controladores no existe pérdida de estabilidad.

Los parámetros de síntesis obtenidos ponderan el error y la energía aplicada al Pendubot con el mismo peso.

Mediante el empleo de las ecuaciones (5.1) y (5.2) y los valores de  $G$  y  $H$  de cada sistema discreto contenidos en las tablas 4.3 y 4.4 se diseñan los controladores discretos  $LQ$  cuyas ganancias se muestran en las tablas 5.1 y 5.2.

Tabla 5.1.- Ganancias  $LQ$  discretas

$k$	$20^\circ(160^\circ)$	$30^\circ(150^\circ)$	$40^\circ(140^\circ)$	$50^\circ(130^\circ)$
$K_{d_1}$	-36.6693	-27.0395	-22.3794	-19.6856
$K_{d_2}$	-36.1987	-26.6980	-22.1273	-19.4977
$K_{d_3}$	-6.4003	-4.9953	-4.3605	-4.0193
$K_{d_4}$	-5.6300	-4.1333	-3.4082	-2.9879

Tabla 5.2.- Ganancias  $LQ$  discretas

$k$	$60^\circ(120^\circ)$	$70^\circ(110^\circ)$	$80^\circ(100^\circ)$	$90^\circ$
$K_{d_1}$	-18.0059	-16.9603	-16.3810	-16.1947
$K_{d_2}$	-17.8643	-16.8507	-16.2903	-16.1103
$K_{d_3}$	-3.8197	-3.7013	-3.6376	-3.6175
$K_{d_4}$	-2.7250	-2.5609	-2.4699	-2.4406

Con las ganancias discretas de las tablas 5.1 y 5.2 es posible controlar el Pendubot en los  $k$ -ésimos puntos de equilibrio para  $k = 20^\circ, 30^\circ, \dots, 160^\circ$ . El diagrama a bloques de la figura 5.1 muestra la acción de control en los puntos de equilibrio para una trayectoria de referencia  $x_k^0$  (ver figura 4.1). El control del Pendubot se realiza mediante una computadora personal que establece una trayectoria  $x_k^0$  de referencia a seguirse, así mismo emplea la ganancia preprogramada  $K_k = [K_{d_1} \ K_{d_2} \ K_{d_3} \ K_{d_4}]$  (ver tablas 5.1 y 5.2) y el par nominal calculado  $\tau_k^0$  mediante instrucciones *IF/THEN* que se calculan cada 5 segundos, estas instrucciones se explican con más detalle mas adelante. La conmutación entre cada controlador se realiza simultáneamente



por los interruptores SW1 y SW2, (ver figura 5.1). La señal de control del Pendubot es  $\tau_k^0 = K_{d_k}(x - x^0) + \mathfrak{I}_k^0$ .

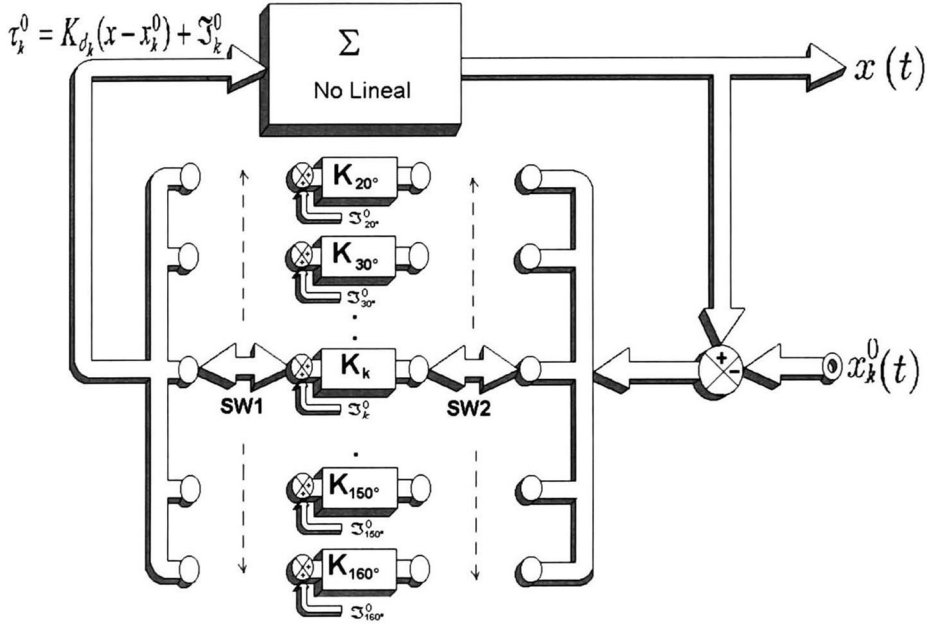


Figura 5.1: Diagrama a bloques de un banco de controladores para el Pendubot

## 5.4 Algoritmo de control

El control del modelo prototipo se realiza con una computadora personal [26] que contiene un paquete de herramientas de software para alarmas [27] y temporización. Este paquete permite programar una señal de alarma cada 5 ms. que se emplea para fijar el período de muestreo  $T_s$  requerido para el control en tiempo real del Pendubot. A continuación se muestra el algoritmo de control empleado:

*Inicializaciones*

*Iniciar alarma de 5 ms*

*MIENTRAS(Lazo\_de\_Control==VERDADERO)*

{

*Lectura de Codificadores*

*Calcular las velocidades angulares*

*Calcular la señal  $\tau_k^0$  para cada banco de controladores*

*Proporcionar la señal de control para el motor conformada*

```

MIENTRAS(Alarma_ expirada==FALSO)
{
  Continuar hasta que la alarma expire
} //Fin del segundo ciclo
} //Fin del primer ciclo

```

Este algoritmo permite el manejo de rutinas para graficación y almacenamiento de datos mientras se realiza el control del Pendubot. La programación de este algoritmo se efectúa en lenguaje *C* y se muestra en el apéndice (C).

### 5.4.1 Posición angular

Las lecturas de las posiciones angulares de los codificadores del Pendubot se realizan mediante una tarjeta interfaz especial para codificadores. Estas posiciones se emplean para calcular la señal de control del Pendubot en los puntos de equilibrio.

### 5.4.2 Aproximación de la velocidad angular

Debido a que el Pendubot no cuenta con sensores para medir la velocidad angular de cada eslabón del Pendubot, es necesario determinar estas velocidades empleando la discretización de la diferenciación [28] definida como

$$\dot{x}(t) \triangleq \mathcal{D}x(t) \triangleq \frac{dx(t)}{dt} \triangleq \lim_{T_s \rightarrow 0} \frac{x(kT_s) - x[(k-1)T_s]}{T_s} \quad (5.3)$$

con  $k = 0, 1, 2, 3, \dots$

donde  $x(t)$  tiene la característica de continuidad apropiada y  $\mathcal{D}$  es el operador derivativo. Así la ecuación (5.3) puede aproximarse por la ecuación de diferencias

$$\mathcal{D}x(kT_s) = \left. \frac{dx(t)}{dt} \right|_{t=kT_s} \approx \frac{x(kT_s) - x[(k-1)T_s]}{T_s}$$

Con el fin de filtrar el ruido numérico, las 3 últimas lecturas de posición angular se promedian mediante la siguiente instrucción

$$x(kT_s) = \frac{x(kT_s) + x[(k-1)T_s] + x[(k-2)T_s]}{3}$$



### 5.4.3 Par nominal $\tau_k^0$

Cada posición de equilibrio del Pendubot requiere de un par nominal que se calcula según la ecuación [21]:

$$\begin{aligned}\tau_k^0 &= K_k x_k^0 + \mathfrak{S}_k^0 \\ \text{con } \mathfrak{S}_k^0 &= (m_1 l_{c1} + m_2 l_1) g \cos(q_1) \\ \text{donde } \tau_k^0 &= K_k x_k^0 + (m_1 l_{c1} + m_2 l_1) g \cos(q_1)\end{aligned}$$

empleado para el control del Pendubot. Nótese que  $\tau_k^0$  es cero cuando  $q_1 = \pm 90^\circ$ . Los valores en las tablas 5.1 y 5.2 son las ganancias discretas preprogramadas que se emplean en un programa en lenguaje *C* para estabilizar el Pendubot. Al cambiar el Pendubot de una posición de equilibrio a otra es necesario conmutar a otro banco de control; esta conmutación se realiza con instrucciones *IF/THEN* dentro del programa en lenguaje *C*. La trayectoria de equilibrio comienza con las ganancias  $K = [K_{d1}, K_{d2}, K_{d3}, K_{d4}]$  para el control de Pendubot para  $q_1 = 90^\circ$ , después se emplean las ganancias para  $q_1 = 80^\circ$ ,  $q_1 = 70^\circ$ ,  $q_1 = 60^\circ$ ,  $q_1 = 50^\circ$ ,  $q_1 = 40^\circ$ ,  $q_1 = 30^\circ$ ,  $q_1 = 20^\circ$ ,  $q_1 = 30^\circ$ , ...,  $q_1 = 90^\circ$  con incrementos de  $10^\circ$ . El tiempo de conmutación entre cada banco de controladores es de 5 segundos.

### 5.4.4 Señal de control del motor

El control del motor del Pendubot se realiza por medio de una PC con un programa en lenguaje *C* que toma las lecturas de posición de los codificadores del Pendubot. Con estas lecturas el programa calcula las velocidades angulares de los eslabones y determina la señal de control mediante la fórmula  $\tau_k^0 = K_{dk} x_k^0 + \mathfrak{S}_k^0$ .

Con el valor del par calculado  $\tau_k^0$ , la computadora personal emplea una variable llamada *digout* [18] que se aplica a la tarjeta DAC, conectada a su vez al servo amplificador (ver figura 5.2), para que el motor ejerza el par real  $\tau_k^0$ ; esta variable esta dada por

$$digout = 2048 + K_a \tau_k^0 \quad (5.4)$$

$K_a$  es una constante de ajuste del programa, que permite que se produzca exactamente el par  $\tau_k^0$  en el motor del prototipo, cuando se aplica el valor digital *digout* a la tarjeta DAC.

Por otro lado, tenemos que [13]

$$\tau_k^0 = K_m i_a = \text{par producido por el motor}$$

o bien

$$i_a = \frac{\tau_k^0}{K_m} \quad (5.5)$$

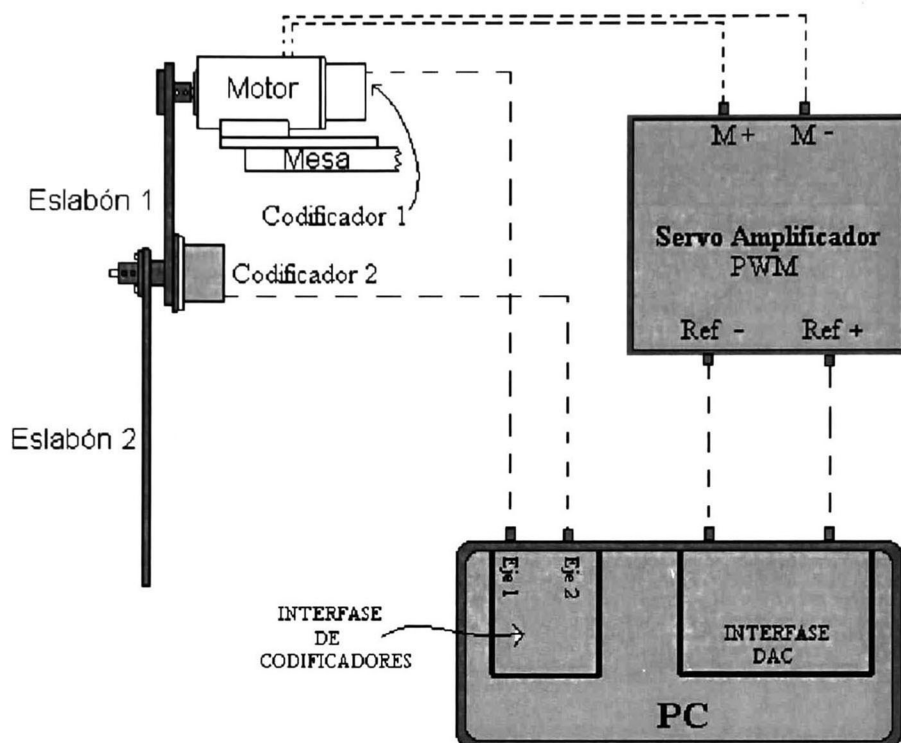


Figura 5.2: Dibujo esquemático de conexión del Pendubot.

y de la curva de la figura 5.3 obtenida experimentalmente, se tiene la ecuación de una recta que aproxima la curva de la figura 5.3

$$i_a = (0.0061) \text{ digout} - 12.4928 \quad (5.6)$$

así, igualando (5.5) y (5.6) se obtiene:

$$\frac{\tau_k^0}{K_m} = (0.0061) \text{ digout} - 12.4928$$

recordando de la ecuación (3.6), que el valor de la constante del par del motor obtenida experimentalmente es:[19]

$$K_m = 0.3441 \text{ N-m/A}$$

así

$$\text{digout} = 2048 + 476.42 \tau_k^0 \quad (5.7)$$

De (5.4) y (5.7) se obtiene  $K_a = 476.42$ .

La tarjeta DAC quedó configurada en modo bipolar [18], proporcionando valores de salida analógicos desde -10 volts hasta +10 volts y el servo amplificador quedó configurado en modo “corriente”

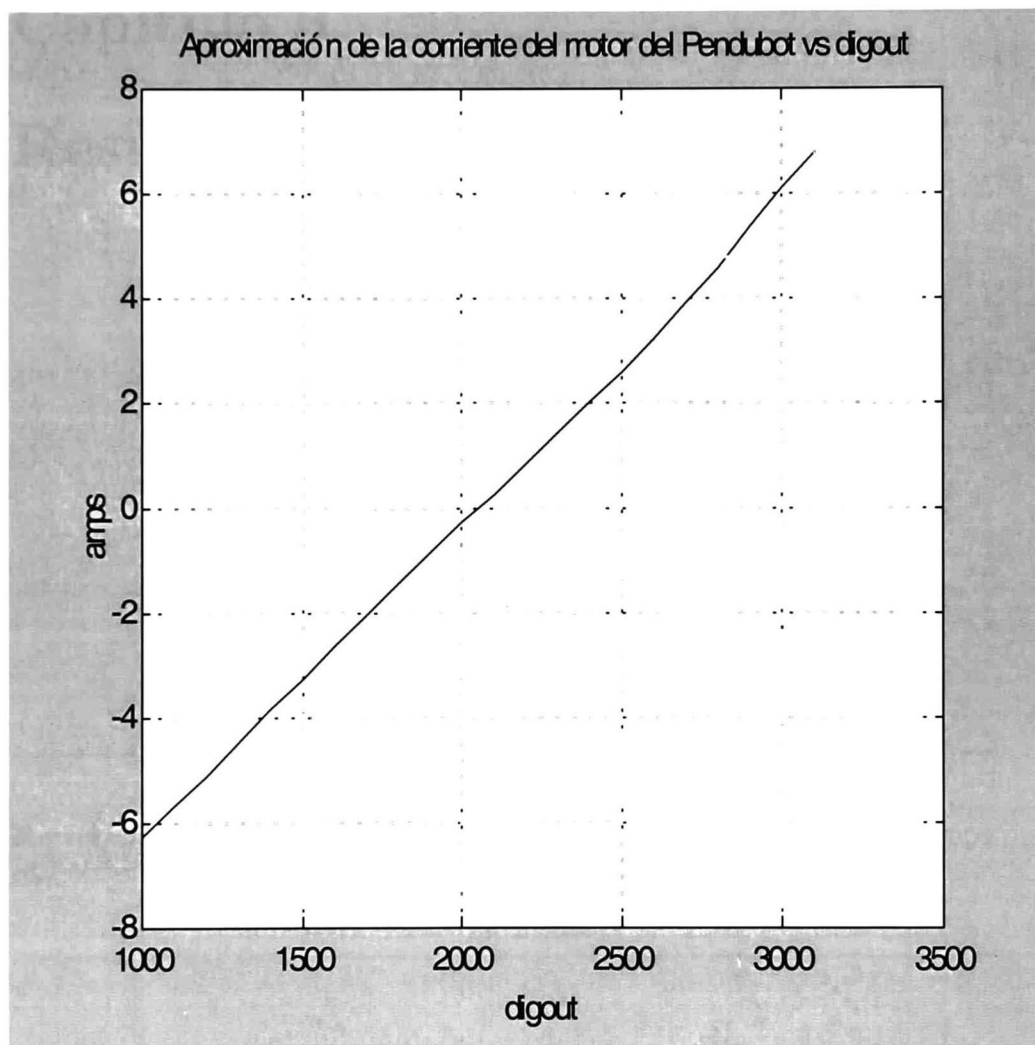


Figura 5.3: Gráfica que relaciona la Corriente de Armadura de motor del Pendubot vs. la Salida Digital de la computadora personal



# Capítulo 6

## Resultados

---

**Resumen:** Este capítulo presenta los resultados en simulación y en tiempo real obtenidos al aplicar las ganancias preprogramadas al Pendubot.

---

Este capítulo presenta los resultados en simulación y en tiempo real obtenidos al aplicar las ganancias pre-programadas al Pendubot. Las simulaciones del modelo matemático del Pendubot se efectúan en Simnon [20]. La trayectoria de referencia propuesta para la posición angular  $q_1$  en el eslabón 1 permitirá que el sistema cambie su punto de operación de  $20^\circ$  ( $0.34906$  rad)  $\leq q_1 \leq 160^\circ$  ( $2.7925$  rad); con incrementos de  $10^\circ$  ( $0.17453$  rad) cada 5 segundos, ver figura 6.1.

Los experimentos en tiempo real son hechos por una computadora personal que utiliza un programa en lenguaje C, que tiene incorporada la misma trayectoria de referencia que la empleada en la simulación del modelo matemático. Tanto los resultados experimentales como los obtenidos en simulación se almacenan en un archivo con formato tipo texto y se transportan a Matlab para obtener las gráficas correspondientes.

## 6.1 Resultados

El experimento de la figura 6.1 muestra la respuesta de  $q_1$  en simulación y en tiempo real. La trayectoria de referencia se fija para  $q_1 = 90^\circ$  ( $1.5707$  rad) durante 5 segundos, después se cambia para  $q_1 = 80^\circ$  ( $1.39626$  rad) durante 5 segundos. Este proceso se repite continuamente con incrementos de  $10^\circ$  hasta que  $q_1 = 20^\circ$  durante 5 segundos. Después se realiza el proceso inverso cambiando  $q_1$  a  $30^\circ$  ( $0.52359$  rad) permaneciendo en esta posición durante 5 segundos, este proceso continúa con incrementos de  $10^\circ$  hasta obtener  $q_1 = 90^\circ$ . En resumen el ángulo  $q_1$  del eslabón 1 del Pendubot varía desde  $90^\circ$  hasta  $20^\circ$  y de  $20^\circ$  a  $90^\circ$  con incrementos de  $10^\circ$  cada 5 segundos. Como puede verse de la figura los resultados en simulación y en tiempo real son muy parecidos, con lo que se confirma la validez del modelo obtenido. Por otra parte también puede observarse un buen seguimiento de la referencia propuesta y satisfaciéndose la especificación sobre el tiempo de estabilización y el error máximo permitido. En la figura 6.2 se observa la respuesta de la posición angular  $q_2$  del eslabón 2 en simulación y en tiempo real. Esta trayectoria varía iniciando con  $q_2 = 0^\circ$  hasta  $q_2 = 70^\circ$  ( $1.2217$  rad) y de  $q_2 = 70^\circ$  hasta  $q_2 = 0^\circ$  con incrementos de  $10^\circ$  cada 5 segundos. Como puede verse de la figura los resultados en simulación y en tiempo real son muy parecidos, con lo que se confirma la validez del modelo obtenido. Por otra parte también puede observarse un buen seguimiento de la referencia propuesta y satisfaciéndose la especificación sobre el tiempo de estabilización y el error máximo permitido.

El error de seguimiento de la posición angular  $q_1$  en simulación y en tiempo real se ilustra en la figura 6.3. Como puede verse el error en estado estacionario no excede  $\pm 4^\circ$  ( $\pm 0.1$  rad. aprox.) en ningún momento, lo cual cumple con las especificaciones requeridas.

El error de seguimiento de la posición angular  $q_2$  en simulación y en tiempo



real se ilustra en la figura 6.4. Como puede verse el error en estado estacionario no excede los  $\pm 2.9^\circ$  ( $\pm 0.05$  rad. aprox.) en ningún momento, lo cual cumple con las especificaciones requeridas.

En la figura 6.5 y 6.6 se muestra la respuesta de la señal de control en simulación y en tiempo real respectivamente. El valor de la señal de control que actúa sobre el motor se define como  $\tau_k^0 = K_k x_k^0 + (m_1 l_{c1} + m_2 l_1) g \cos(q_1)$ . En todo momento se observa que la señal de control no excede el límite de par máximo ( $\tau_{m\acute{a}x} = \pm 2.75$  N-m) requerido. Nótese que la figura 6.6 tiene sobrepuesta la señal de control de la gráfica de la figura 6.5 para fines de comparación.

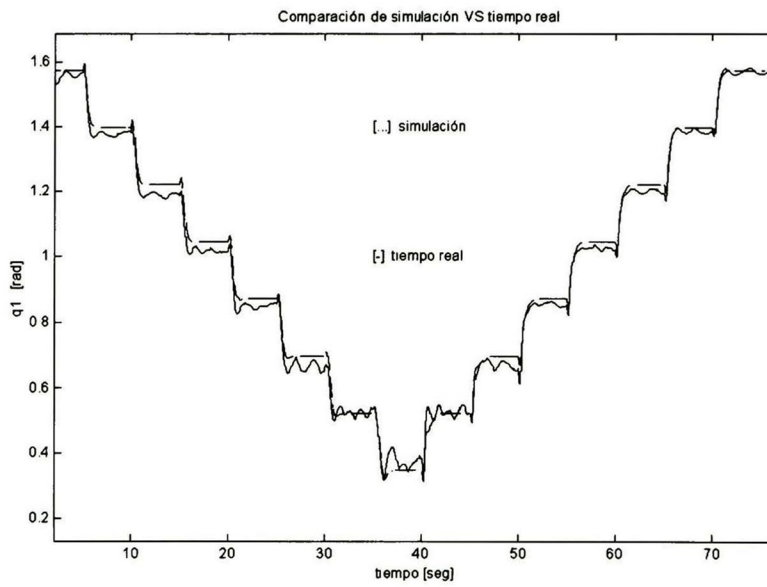


Figura 6.1: Respuesta de la posición angular  $q_1$  en simulación y en tiempo real.

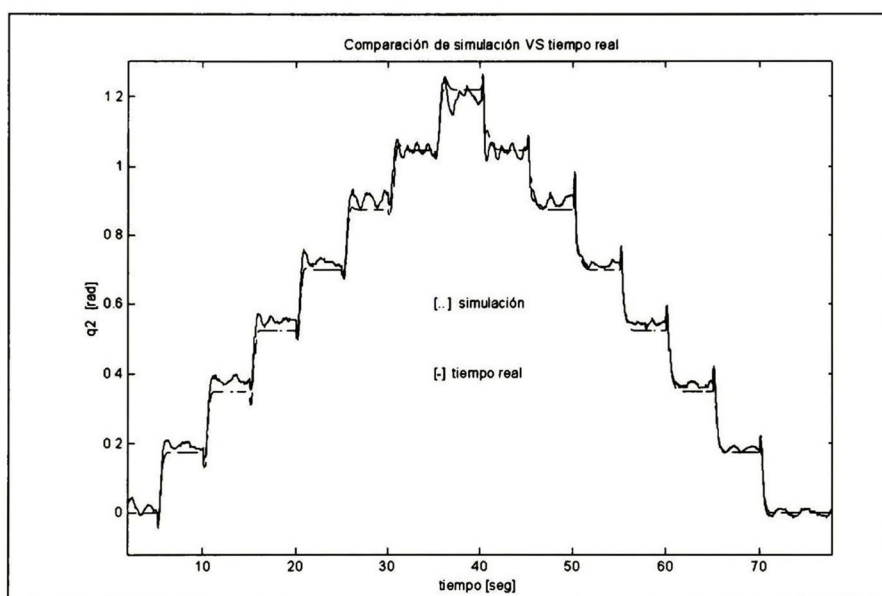


Figura 6.2: Respuesta de la posición angular  $q_2$  en simulación y en tiempo real.

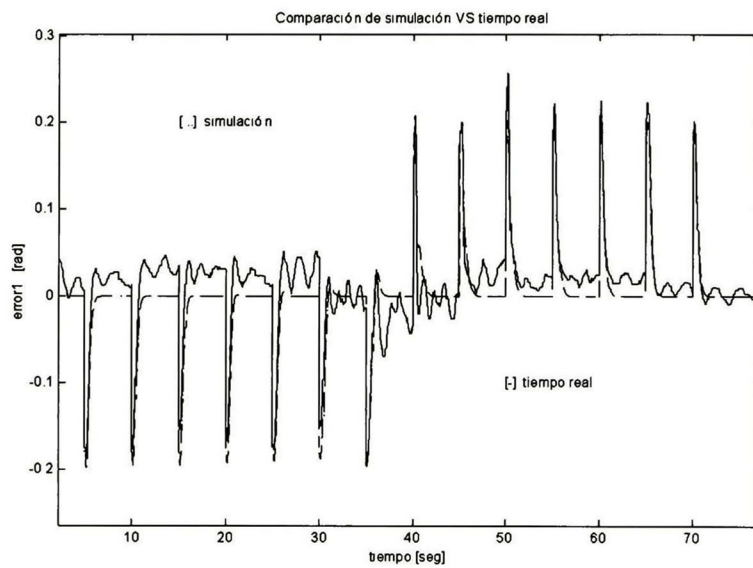


Figura 6.3: Error de seguimiento para  $q_1$  en simulación y en tiempo real.

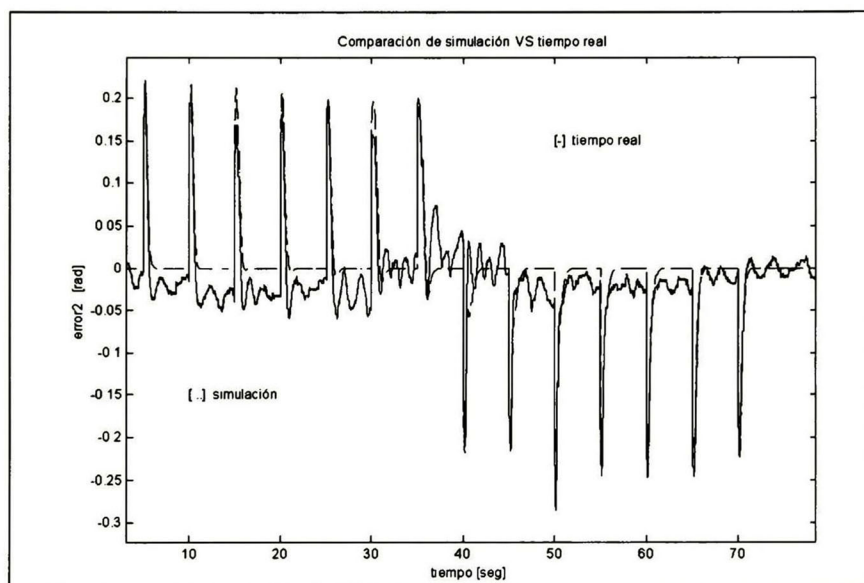


Figura 6.4: Resultados del error obtenidos de la simulación del modelo matemático y del modelo real con  $0^\circ \leq q_2 \leq 70^\circ$

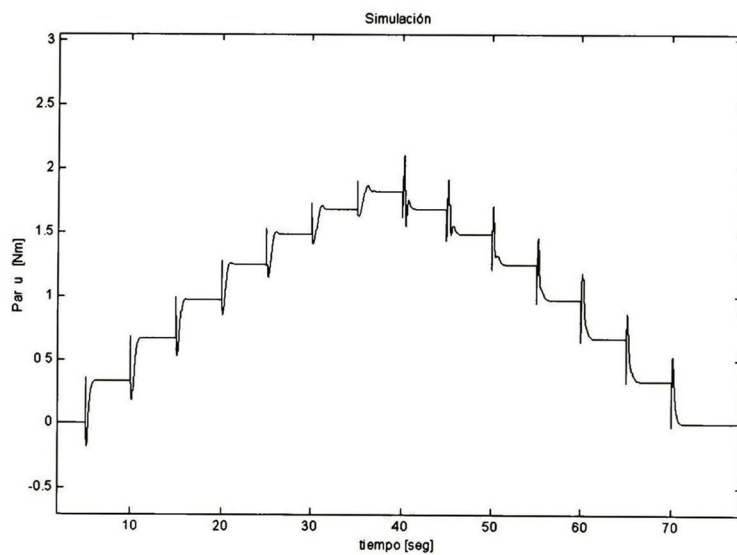


Figura 6.5: Respuesta de la señal de control en simulación.

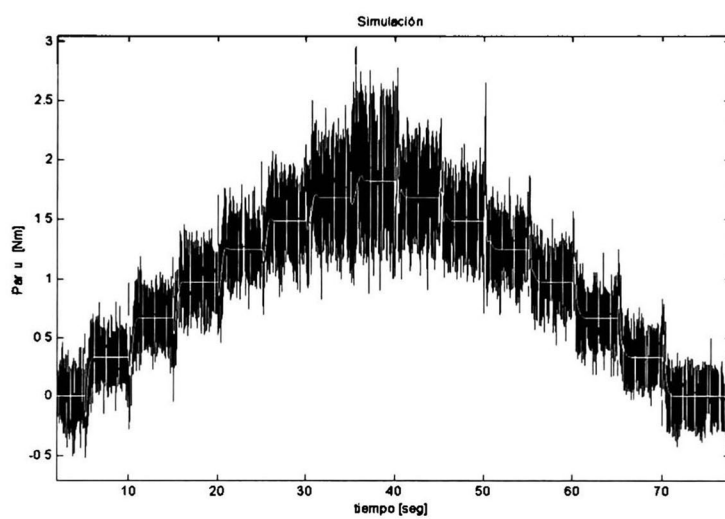


Figura 6.6: Respuesta de la señal de control en tiempo real.





# Capítulo 7

## Conclusiones

En esta tesis se obtiene el modelo matemático del Pendubot incluyendo los efectos de *fricción viscosa de las uniones*. La obtención de los *valores de los parámetros* representa un avance adicional, con respecto a los trabajos previos sobre este tópico, ya que con los valores con que se contaba inicialmente no era posible obtener resultados satisfactorios al comparar los resultados de las simulaciones con los de tiempo real. Las ganancias discretas del controlador que se obtienen a partir del *modelo matemático con fricción y sin fricción* son muy parecidas ya que ambas pueden controlar el Pendubot sin obtenerse cambios significativos.

Los valores de las *inercias de los eslabones* del Pendubot son bastante aproximados como lo muestran las comparaciones entre los datos experimentales y los obtenidos en simulación.

A partir de los experimentos en tiempo real se observó en todo momento que el valor del *error* fue menor que  $0.1$  ( $5^\circ$ ) radianes tanto para  $q_1$  como  $q_2$  cuando el sistema se estabiliza y por otro lado el valor absoluto del par del motor siempre se mantiene por debajo del par máximo del motor especificado ( $2.75$  N-m.), con lo cual los resultados experimentales satisfacen las especificaciones.

Experimentalmente se muestra que es posible implementar controladores por ganancias preprogramadas en este sistema subactuado.

### 7.0.1 Investigaciones futuras:

A continuación se citan otros trabajos que pudieran desarrollarse con el Pendubot:

- Analizar las propiedades de *estabilidad del esquema de conmutación* entre puntos de equilibrio.
- Obtener un observador que permita controlar el Pendubot con un solo codificador en la unión 2.

- Proponer mejores filtros que reduzcan el ruido numérico de las velocidades angulares estimadas a partir de las posiciones angulares de los codificadores del Pendubot.
- Emplear técnicas de visión para el sensado de las posiciones angulares de los eslabones del Pendubot
- Operar el Pendubot con  $q_1$  a posiciones (de equilibrio) angulares menores a  $20^\circ$
- Operar el Pendubot para una trayectoria de  $q_1$  de  $0^\circ$  a  $360^\circ$
- Controlar el Pendubot con microcontroladores.

# Bibliografía

- [1] Spong M.W., and Praly L., "Control of Underactuated Mechanical Systems Using Switching and Saturation," *Proc. of the Block Island Workshop on Control Using Logic Based Switching*, Springer-Verlag, 1996.
- [2] L.E. Ramos, Castillo-Toledo B., "Robust regulation via sliding mode for an underactuated system", *proc. of the IASTED International Conference on Robotics and Manufacturing*, Cancún, Mex., Mayo 1997.
- [3] L.E. Ramos, Castillo-Toledo B., Álvarez J., "Nonlinear regulation of an underactuated system", *proc. of the International Conference on Robotics and Automation*, Albuquerque, NM, USA, Abril 1997.
- [4] Sánchez Edgar, Nuño Luis A., Ya-Chen Hsu and Guanrong Chen, "Fuzzy PD Scheme for underactuated Robot Swing-up Control", *in proceedings of the World Congress on Computational Intelligence 1998*, pp. 145-149, June 1998.
- [5] Sánchez Edgar, Nuño Luis A., Ya-Chen Hsu and Guanrong Chen, "Real time Fuzzy Swing-up Control for underactuated Robot", *in proceedings of the JCIS '98 vol 1*, pp. 152-155, N.C. USA, October 1998.
- [6] Wells Dare A., *Dinámica de lagrange*, Colombia: Mc Graw Hill, 1972.-
- [7] Ramos Luis Enrique, *Control de un sistema Electromecánico Subactuado (Pendubot)*, Tesis de maestría, dept. de Ingeniería Eléctrica, Sección Control Aut., Cinvestav, Guadalajara Méx., 1996.
- [8] Beer P. Ferdinand y Jonhson Jr., *Mecánica de Materiales 2a. edición*, Colombia: Mc Graw Hill, 1993.
- [9] Serway Raymond A., *Física, Tomo I, 3a. edición*, México: Mc Graw Hill, 1972.
- [10] Kailath Thomas, *Linear Systems*, Englewood Cliffs, N. J.: Prentice Hall, 1980.

- [11] Alleyne A., et.al., "A Collegewide Laboratory-Based Program in Control Systems Technology at The University of Illinois at Urbana-Champaign", *1996 Conference on Decision and Control, Kobe, Japan*, Dec. 1996.
- [12] Spong M.W., and Block, D.J., "The Pendubot: A Mechatronic System for Control Research and Education," *34th IEEE Conf. on Decision and Control*, pp. 555-556, New Orleans, Dec., 1995.
- [13] Ogata Katsuhiko, *Ingeniería de Control Moderna*, México: Prentice-Hall, 1998.
- [14] Spong M. W., and Vidyasagar, M., *Robot Dynamics and Control*, New York: John Wiley and Sons, Inc., 1989.
- [15] Craig, John J., *Introduction to Robotics mechanics and control*, second edition, Canada: Addison-Wesley, 1989.
- [16] Lee C. S. G., González R. C., Fu, K. S., *Tutorial on Robotics*, USA, IEE Computer Society Press, 1983.
- [17] Karaïskos & Fulton, *AUTOCAD FOR MECHANICAL ENGINEERS & DESIGNERS*, 1995, ISBN: 0-471-01779-5
- [18] *CIO-DAC02 USER'S MANUAL*, REVISION 2.0, FEBRUARY 1994
- [19] Haro B., "Procedimiento de laboratorio para el cálculo de las constantes de motores de C.D.", *Congreso de Investigación UAG '98, 12, 13 y 14 de noviembre, 2do Encuentro de investigación Educativa*, Guadalajara, Jalisco, 1998.
- [20] *Simnon for Windows*, version 2.0. SSPA Maritime Consulting AB, S-400-22, Göteborg, Sweden.
- [21] Haro B., Begovich O., "A set of controllers: A real time application for the Pendubot", *proceedings of International Symposium on Robotics and Automation "ISRA '98*, Saltillo, Coahuila, México, Diciembre 12-14, 1998, pp. 365-369.
- [22] *The Matlab Works, Control System Toolbox*, Inc. 386-Matlab, 1991.
- [23] Lewis L., Frank and Syrmos, Vassilis L., *Optimal Control*, USA: John Wiley & Sons, Inc., 1995.
- [24] Ogata Katsuhiko, *Discrete Time Control Systems*, USA: Prentice Hall, 1995.
- [25] Philips Charles & Nagle H. Troy, *Digital Control System Analysis and Design, Third Edition*, USA: Prentice Hall, 1995



- [26] Block Daniel Jerome, *Mechanical design and control of the Pendubot*, Urbana, Illinois, USA, THESIS, B.S. University of Illinois, 1991.
- [27] Ryle Design, *PC Timer Tools alarm functions in a MSDOS C enviroment*, 1988, 1992.
- [28] Houpis Constantine, Gary B. Lamont, *Digital Control Systems*, Singapore: Mc Graw Hill, 1985.





## **Sección de apéndices**

# Apéndice A

## Instrucciones de Matlab y Simnon

El siguiente conjunto de instrucciones indica como transferir información de Simnon a Matlab, evitando así a los interesados la pérdida de tiempo para conocer la manera de realizar estas operaciones. Ayuda además en Matlab a obtener cada variable del archivo separadamente y poder así graficarlas.

La siguiente instrucción permite capturar los datos almacenados temporalmente en un archivo creado por Simnon llamado “store” a un archivo que se le denominará en este caso *angulo.t*:

$$\textit{export angulo} < \textit{store}$$

El nombre “store” es el nombre que simnon da por default al archivo donde se almacenan los resultados de una simulación. Este nombre puede ser modificado al seleccionar el menú de simulación de Simnon (ver figura A.1).

Una vez hecho lo anterior en el ambiente Matlab se cargan los datos del archivo “*angulo.t*” en código ASCII, mediante la siguiente instrucción

$$\textit{load angulo.t} - \textit{ascii}$$

esta instrucción carga el contenido del archivo *angulo.t* en una matriz de  $n \times m$  en Matlab es el número de variables almacenadas con el nombre de *angulo*, donde  $n$  es el el número de datos a graficar en el tiempo y  $m$ . Para el caso particular de  $m = 2$ , la siguiente instrucción selecciona únicamente la primera columna de la matriz llamada *angulo* y la almacena en la variable *tiempo*.

$$\textit{tiempo} = \textit{angulo} * [1; 0]$$

La instrucción que se muestra enseguida selecciona la segunda columna de la matriz llamada *angulo* almacenando su contenido en la variable *q1*.

$$\textit{q1} = \textit{angulo} * [0; 1]$$

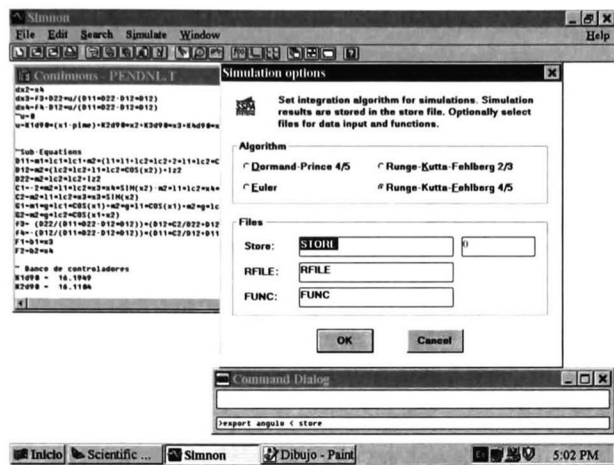


Figura A.1: Menú de Simnon

Finalmente Matlab ya tiene el valor de  $q1$  y tiempo, los cuales pueden graficarse con:

*plot(tiempo, q1)*

# Apéndice B

## Programa en Matlab para linealizar

Este apéndice contiene el listado del programa en Matlab que linealiza el modelo no lineal (2.36), (2.37), (2.38) en diferentes puntos de operación. Cada modelo linealizado es de la forma

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

El nombre de este programa es “noli2li.m” y se ejecuta en Matlab de la siguiente manera

```
> [A,B,C,D]=noli2li(q1,q2)
```

El programa “noli2li.m” llama a un subprograma llamdo “macro.m” el cual ejecuta un cálculo repetitivo dentro del programa. El contenido del programa se muestra a continuación (para obtener mas información sobre las instrucciones usadas aquí de Matlab emplee el comando “help”):

```
%Este programa linealiza el modelo no lineal del Pendubot
%para cualquier posición de equilibrio tal que q1+q2=±90°
%La notación de las variables aquí empleadas corresponde
%generalmente a la utilizada en el texto de la tesis de Bernardo Haro
%Se auxilia de la función MACRO.M
%Valores de Bernardo Haro*****
%16/Agosto/98
%Las condiciones Iniciales q1, q2 son en grados
%Para ejecutarse teclee:
%noli2li(q1,q2), obtendrá los valores de los elementos de la matriz A
%o bien:
 %[A,B,C,D]=noli2li(q1,q2)
```

```

%*****
%Datos de entrada: posiciones angulares del eslabón 1(q1) y eslabón 2 (q2)
% para el punto de equilibrio donde se desea linealizar.
%Datos de salida: Matrices A, B, C, D del modelo linealizado en q1 y q2
%Inicio del Programa
function[A,B,C,D]=noli2li(q1,q2)
%Inicialización de datos de salida
A=[0 0 1 0;0 0 0 1;0 0 0 0;0 0 0 0];
B=[0;0;0;0];
C=[0 1 0 0];
D=[0];
%Parámetros del Pendubot
J1=0.0259;
J2=0.01340;
m1=0.8293;
m2=0.3402;
l1=0.2032;
lc1=0.1551;
lc2=0.1635125;
fric1=0.00545;
fric2=0.00047;
g=9.81;
%x1=q1; x2=q2; x3=d(q1)/dt; x3=d(q2)/dt
syms x1 x2 x3 x4 u;
%Modelo no lineal
%fricc1=fricción del eslabón 1; fricc2=fricción del eslabón 2
F1=fric1*x3;
F2=fric2*x4;
%D11=m1*lc1*lc1+m2*(l1*l1+lc2*lc2+2*l1*lc2*cos(x2))+Izz1+Izz2;
%D12=m2*(lc2*lc2+l1*lc2*cos(x2))+Izz2;
%D22=m2*lc2*lc2+Izz2;
%J1=m1*lc1*lc1+Izz1
%J2=m2*lc2*lc2+Izz2
D11=J1+J2+m2*(l1*l1+2*l1*lc2*cos(x2));
D12=m2*(l1*lc2*cos(x2))+J2;
D22=J2;
C1=-2*m2*l1*lc2*x3*x4*sin(x2)-m2*l1*lc2*x4*x4*sin(x2);
C2=m2*l1*lc2*x3*x3*sin(x2);
G1=m1*g*lc1*cos(x1)+m2*g*l1*cos(x1)+m2*g*lc2*cos(x1+x2);
G2=m2*g*lc2*cos(x1+x2);
%Cálculo de dx/dt=[e1 e2 e3 e4]'
e1=x3;
e2=x4;

```

```

e3=(D22/(D11*D22-D12*D12))*(D12*C2/D22+D12*G2/D22+D12*F2/D22-C1-
G1-F1+u);
e4=(-D12/(D11*D22-D12*D12))*(D11*C2/D12+D11*G2/D12+D11*F2/D12-C1-
G1-F1+u);
%e4=-C2/D22-G2/D22+((D12*C2/D22)+(D12*G2/D22)-C1-G1)*(-D12/(D11*D22-
D12*D12));
%Linealización
F=diff(e3,x1);% d(e3)/dx1
%Cálculo de los elementos de A
A(3,1)=macro(F,q1,q2);
F=diff(e3,x2);
A(3,2)=macro(F,q1,q2);
F=diff(e3,x3);
A(3,3)=macro(F,q1,q2);
F=diff(e3,x4);
A(3,4)=macro(F,q1,q2);
F=diff(e4,x1);
A(4,1)=macro(F,q1,q2);
F=diff(e4,x2);
A(4,2)=macro(F,q1,q2);
F=diff(e4,x3);
A(4,3)=macro(F,q1,q2);
F=diff(e4,x4);
A(4,4)=macro(F,q1,q2);
%Inicio del cálculo de los elementos de B
F=diff(e3,u);
B(3,1)=macro(F,q1,q2);
F=diff(e4,u);
B(4,1)=macro(F,q1,q2);

```

La función auxiliar para la linealización es:

%Esta función es auxiliar de NOLI2LI

%By: B.HARO 16,Agosto,98

function [R]=macro(F,q1,q2)

R=simplify(F);

%Parámetros del Pendubot

m1=0.8293;

m2=0.3402;

l1=0.2032;

lc1=0.1551;

g=9.81;

x1=q1\*3.1415926535/180;

x2=q2\*3.1415926535/180;

```
x3=0.0;  
x4=0.0;  
%Cálculo del Par extra requerido  
u=(m1*g*lc1+m2*g*ll)*cos(x1);  
%Evaluación numérica de R  
R=eval(R);  
R=vpa(R,10);
```



# Apéndice C

## Programa en C

Este apéndice contiene el listado del programa en lenguaje C, que controla el Pendubot en tiempo real.

```
//Include Necesarios
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include<graphics.h>
#include<dos.h>
#include<stdlib.h>
#include "pchrt.h"
#include "counter.h"
//Inicializaciones
#define ADDRESS 0x300
#define CHANNEL 0
#define ENCADDRESS1 0x210
#define ENCADDRESS2 0x220
#define ZR_SETTING 0
#define QUAD_COUNT 4
#define NUM_AXIS 2
#define ENCINIT 30000L
/*Tiempo de muestreo en microsegundos*/
#define ALARMTIME 5000L
#define SAMPLE 0.005000
//Define necesarios
#define PI 3.1415926
#define HALFPI 1.5707963
//Datos del codificador
#define ENCDIV 4096
#define G 386
//pend1 y cad son variable auxiliares para crear un
```

```
//archivo tipo texto donde se almacenan las datos de las simulaciones en tiempo  
real
```

```
FILE *pend1;  
char cad[500];  
//Diseño de un ambiente grafico para datos de q1 y q2  
void Grafico(void);  
void Pintar(void);  
//Declaración de variables externas  
extern int scan_for_axis_location( int *axis );  
extern long read_counter( int axis );  
extern void reset_counter( int axis, int quad_count );  
extern void preset_counter( int axis, long value );  
extern void zr_settings( int axis, int zr_setting );  
extern unsigned long tiempo;  
//Inicio  
void main()  
{  
//declaración de variables  
unsigned int outputHi,outputLow;  
int niv=0;  
int firstime;  
float Ajuste1,Ajuste2;  
int a1;  
unsigned long istat;  
int digout=2048;  
float error1=0;  
float error2=0;  
float diff=0.0;  
float Pendiente=1.0/0.0021;  
float Ubarra=0;  
float m1=.8293;  
float m2=.3402;  
float g=9.81;  
float lc1=.1551;  
float ll=.2032;  
float f_digout=0.0;  
float u=0.0;  
float t=0.0;  
float Amp=0.0;  
float Kmot=.3465;  
long lCntA, lCntB;  
// BANCO DE CONTROLADORES
```

```

float K1[10] = {-16.1949,-16.3814,-16.9603,-18.0058,-19.6858,-22.3792,-27.0395,-
36.6704};
float K2[10] = {-16.1104,-16.2907,-16.8507,-17.8642,-19.4980,-22.1271,-26.6981,-
36.1998};
float K3[10] = {-3.6175,-3.6378,-3.7013,-3.8197,-4.0193,-4.3605,-4.9953,-6.4005};
float K4[10] = {-2.4406,-2.4700,-2.5609,-2.7250,-2.9879,-3.4081,-4.1333,-5.6301};
//declaración de variables
float x1k=0.0;
float x2k=0.0;
float t2=0.0;
float t1=0.0;
float x1old = 0.0;
float x2old = 0.0;
float x3k=0.0,x4k=0.0;
float x3old[2] = {0.0,0.0};
float x4old[2] = {0.0,0.0};
//instrucción de apertura de archivo
if ((pend1=fopen("c:\\pendu1.txt","wt"))==NULL) { free(pend1);printf("\nError
en la apertura del archivo\n");}
//Desactivación del motor
digout = 2048; // Envía cero Amperes al motor
outputHi = digout/16; outputLow = 16*(digout - outputHi*16);
outp(ADDRESS,outputLow); outp(ADDRESS+1,outputHi);
/* Inicialización de la tarjeta lectora de los codificadores*/
zr_settings(ENCADDRESS1,ZR_SETTING);
zr_settings(ENCADDRESS2,ZR_SETTING);
reset_counter(ENCADDRESS1,QUAD_COUNT);
reset_counter(ENCADDRESS2,QUAD_COUNT);
preset_counter(ENCADDRESS1,ENCINIT);
preset_counter(ENCADDRESS2,ENCINIT);
/* Inicialización del temporizador */
istat = t_start();
if (! istat)
{
printf("Imposible inicializar temporizadores ... existe\n");
return;
}
/* inicialización de alarmas */
istat = t_alarm_start();
if (istat)
printf("%d bytes asignados para alarmas\n\n",istat);
else
{

```

```

printf("Imposible inicializar alarmas ... existe\n");
t_stop();
return;
}
int controlador = DETECT, modo, coderror;
modo = VGAHI;
initgraph(&controlador, &modo, " ");
coderror = graphresult();
if (coderror != grOk)
{
printf ("Error en la función que grafica: %s\n", grapherrormsg(coderror));
getch();
exit(1);
}
setbkcolor(BLACK);
Grafico();
outtextxy(50,70,"Tomar el boton de seguridad y presionar");
/* Declaración de alarma con resolución de microsegundos */
a1 = t_alarm_alloc(T_USEC);
do {
} while (!kbhit() ); /* enddo */
if (!getch())
(void)getch();
printf("\n");
/*Señal de referencia de inicio con q1=90°*/
Ajuste1=90.0;// q1=90°
Ajuste2=90.0-Ajuste1; //q2=90-q1
//Conversión de q1 y q2 de grados a radianes
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
//Cálculo del par extra requerido para un valor de q1 dado  $\mathfrak{S}_k^0$ 
Ubarra=(m1*g*lc1+m2*g*ll1)*cos(Ajuste1);
getch();
/*Condiciones iniciales*/
x1old = Ajuste1;
x2old = Ajuste2;
x3old[0] = 0.0;
x4old[0] = 0.0;
x3old[1] = 0.0;
x4old[1] = 0.0;
firstime = 1;
do
{

```

```

//Visualización gráfica del Pendubot
Pintar();
if (firsttime)
{
firsttime = 0;
/* activa alarma para 5000 microsegundos (5 ms)*/
t_alarm_set(a1,ALARMTIME,T_RESTART);
}
/*Lectura de Codificadores*/
lCntA = read_counter(ENCADDRESS1); /* Lee la cuenta del codificador del
eslabón 1*/
lCntB = read_counter(ENCADDRESS2); /* Lee la cuenta del codificador del
eslabón 2*/
x1k = (lCntA-ENCINIT)*((2*PI)/ENCDIV)-HALFPI; /* convierte la cuenta
del codificador del eslabón 2 a radianes */
x2k = (lCntB-ENCINIT)*((2*PI)/ENCDIV); /* convierte la cuenta del codifi-
cador del eslabón 1 a radianes */
x3k = (x1k-x1old)/SAMPLE; /* calcula la velocidad del eslabón 1*/
x4k = (x2k-x2old)/SAMPLE; /* calcula la velocidad del eslabón 2*/
x3k = (x3k+x3old[0]+x3old[1])/3.0; /* estima la velocidad del eslabón 1 */
x4k = (x4k+x4old[0]+x4old[1])/3.0; /* estima la velocidad del eslabón 2 */
/*Almacenamiento de datos en disco duro*/
error1=Ajuste1-x1k;
error2=Ajuste2-x2k;
sprintf(cad,"%f\t%f\t%f\t%f\t%f\t%f\t\n",t,x1k,x2k,error1,error2,u);
fprintf(pend1,cad);
//Visualización grafica
if(t1>=10)
{
t1=0.0;
clearviewport();
Pintar();
}
putpixel(t1*38+20,150+x2k*50, GREEN);
putpixel(t1*38+20,150+x1k*50, RED);
t1=t1+SAMPLE;
//Commutacion de controladores
//Controlador de 80°
if(t2>5.0 && t2<10.0){
Ajuste1=80.0;
Ajuste2=90.0-Ajuste1;
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
}

```



```

niv=1;//variable que selecciona el Banco de control de 80°
Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1);}
//Controlador de 70°
if(t2>10.0 && t2<15.0){
Ajuste1=70.0;
Ajuste2=90.0-Ajuste1;
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
niv=2;
Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1);}
//Controlador de 60°
if(t2>15.0 && t2<20.0){
Ajuste1=60.0;
Ajuste2=90.0-Ajuste1;
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
niv=3;
Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1);}
//Controlador de 50°
t2=t2+SAMPLE;
if(t2>20.0 && t2 < 25.0){
Ajuste1=50.0;
Ajuste2=90.0-Ajuste1;
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
niv=4;
Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1); }
//Controlador de 40°
if(t2 > 25.0 && t2 < 30.0){
Ajuste1=40.0;
Ajuste2=90.0-Ajuste1;
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
niv=5;
Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1); }
//Controlador de 30°
if(t2 > 30.0 && t2 < 35.5 ){
Ajuste1=30.0;
Ajuste2=90.0-Ajuste1;
Ajuste1=Ajuste1*(PI/180.0);
Ajuste2=Ajuste2*(PI/180.0);
niv=6;
Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1);}

```

```

//Controlador de 20°
if(t2<40.0 && t2>35.0){
    Ajuste1=20.0;
    Ajuste2=90.0-Ajuste1;
    Ajuste1=Ajuste1*(PI/180.0);
    Ajuste2=Ajuste2*(PI/180.0);
    niv=7;
    Ubarra=(m1*g*lc1+m2*g*l1)*cos(Ajuste1);}
//Controlador de 30°
if(t2<45.0 && t2>40.0){
    Ajuste1=30.0;
    Ajuste2=90.0-Ajuste1;
    Ajuste1=Ajuste1*(PI/180.0);
    Ajuste2=Ajuste2*(PI/180.0);
    niv=6;
    Ubarra=(m1*g*lc1+m2*g*l1)*cos(Ajuste1);}
//Controlador de 40°
if(t2<50.0 && t2>45.0){
    Ajuste1=40.0;
    Ajuste2=90.0-Ajuste1;
    Ajuste1=Ajuste1*(PI/180.0);
    Ajuste2=Ajuste2*(PI/180.0);
    niv=5;
    Ubarra=(m1*g*lc1+m2*g*l1)*cos(Ajuste1);}
//Controlador de 50°
if(t2<55.0 && t2>50.0){
    Ajuste1=50.0;
    Ajuste2=90.0-Ajuste1;
    Ajuste1=Ajuste1*(PI/180.0);
    Ajuste2=Ajuste2*(PI/180.0);
    niv=4;
    Ubarra=(m1*g*lc1+m2*g*l1)*cos(Ajuste1);}
//Controlador de 60°
if(t2<60.0 && t2>55.0){
    Ajuste1=60.0;
    Ajuste2=90.0-Ajuste1;
    Ajuste1=Ajuste1*(PI/180.0);
    Ajuste2=Ajuste2*(PI/180.0);
    niv=3;
    Ubarra=(m1*g*lc1+m2*g*l1)*cos(Ajuste1);}
//Controlador de 70°
if(t2<65.0 && t2>60.0){
    Ajuste1=70.0;

```



```

    Ajuste2=90.0-Ajuste1;
    Ajuste1=Ajuste1*(PI/180.0);
    Ajuste2=Ajuste2*(PI/180.0);
    niv=2;
    Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1);}
//Controlador de 80°
    if(t2<70.0 && t2 > 65.0){
        Ajuste1=80.0;
        Ajuste2=90.0-Ajuste1;
        Ajuste1=Ajuste1*(PI/180.0);
        Ajuste2=Ajuste2*(PI/180.0);
        niv=1;
        Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1); }
//Controlador de 90°
    if( t2 > 70.0){
        Ajuste1=90.0;
        Ajuste2=90.0-Ajuste1;
        Ajuste1=Ajuste1*(PI/180.0);
        Ajuste2=Ajuste2*(PI/180.0);
        niv=0;
        Ubarra=(m1*g*lc1+m2*g*ll)*cos(Ajuste1); }
// Ley de control de balanceo
    u = -K1[niv]*(x1k-Ajuste1) - K2[niv]*(x2k-Ajuste2) - K3[niv]*x3k
        K4[niv]*x4k+Ubarra;
//Protección del motor
    if (fabs(u) >2.7528.0) { //Equivalente a 8 amperes en la Amadura del motor del
Pendubot.
        u = 0.0;
    } /* endif */
    f_digout = 2048.0 + Pendiente*u; //Señal conformada para el motor
// del Pendubot
    digout = (int)f_digout;
    diff = f_digout - (float)digout;
    if (diff > .5)
        digout++;

/* limitar la salida */
    if (digout > 3686)
        digout = 3686;
    if (digout < 410)
        digout = 410;
/* salida del control */
    outputHi = digout/16; outputLow = 16*(digout - outputHi*16);

```

```

outp(ADDRESS,outputLow); outp(ADDRESS+1,outputHi);
x3old[1] = x3old[0]; /* salva los posiciones y velocidades anteriores */
x4old[1] = x4old[0];
x1old = x1k;
x2old = x2k;
x3old[0] = x3k;
x4old[0] = x4k;
//Incrementa el tiempo transcurrido
t = t+SAMPLE;
istat = t_alarm_check(a1); /* checa alarmas*/
if (istat != 0)
{
setcolor(WHITE);
outtextxy(10,10,"se requiere cambiar el periodo de muestreo");
setcolor(BLACK);
outtextxy(10,10,"se requiere cambiar el periodo de muestreo");
setcolor(WHITE);
}
//Verificación de que la alarma del temporizador no exceda los 5 ms.
do {
istat = t_alarm_check(a1);
} while (istat == 0);
}while (!kbhit() );//Fin del do
//Configuración de finalización del programa
digout = 2048; /* coloca la salida CIO-DAC-02 a cero */
outputHi = digout/16; outputLow = 16*(digout - outputHi*16);
outp(ADDRESS,outputLow); outp(ADDRESS+1,outputHi);
t_alarm_stop();
t_stop();
fclose(pend1);
} /* Fin del main */
//Sub-programas
void Grafico()
{
int maxx, maxy;
cleardevice();
setcolor(WHITE);
maxx = getmaxx();
maxy = getmaxy();
line(5, 5, maxx-5, 5);
line(5, 5, 5, maxy-5);
line(5, maxy-5, maxx-5, maxy-5);
line(maxx-5, 5, maxx-5, maxy-5);

```

```

line(7, 7, maxx-7, 7);
line(7, 7, 7, maxy-7);
line(7, maxy-7, maxx-7, maxy-7);
line(maxx-7, 7, maxx-7, maxy-7);
setcolor (WHITE);
moveto (210, 30);
outtext ("GRAFICA DE q1 VS TIEMPO");
// moveto (180,7);
// lineto (180,maxy-7);
}
void Pintar(void)
{
    setviewport(100, 100, 500, 400, 0);
    line(20, 20, 20, 280);
    line(20, 150, 400, 150);
}

```

# Apéndice D

## Simulación con Simnon.

El programa de simulación del modelo matemático en Simnon está integrado por tres partes que son ejecutadas por un archivo tipo “macro” llamado go. Estas partes principales son:

Control.- Este programa tiene la opción de tipo de archivo “discrete” y calcula la señal de control  $\tau_k^0 = K_{d_k} x^0 + \mathfrak{I}_k^0$  que se aplica al modelo no lineal cada 5 ms, empleando las ganancias discretas  $K_{d_k}$  y el par extra requerido  $\mathfrak{I}_k^0$  obtenidas en el desarrollo de la tesis.

Pendubot.- Este programa tiene la opción de tipo de archivo “continuous”. Aquí se introduce el modelo no lineal obtenido en el capítulo 2.

Nexo.- Este programa tiene la opción de tipo de archivo “connecting” y permite “conectar” el programa Control con el programa Pendubot.

Este apéndice contiene el programa para simular el comportamiento en tiempo discreto del pendubot.

### MACRO GO

```
" Version: 1.0
" Abstract:
" Description:
" Revision: 1.0
" Created: 21/05/98
" Enter commands here:
syst control pendubot nexo
store sx1[pendubot] sx2[pendubot] error1[control] error2[control]
" sx3[pendubot] sx4[pendubot] su[pendubot]
simu 0 80
split 2 2
area 1 1
ashow sx1[pendubot]
area 1 2
ashow sx2[pendubot]
area 2 1
```

```

ashow error1[control]
area 2 2
ashow error2[control]
"ashow sx1[pendubot] sx2[pendubot]" su[pendubot]
END

```

## CONNECTING SYSTEM NEXO

```

" Version: 1.0
" Abstract:
" Description:
" Revision: 1.0
" Author:
" Created: 21/05/98
" Time, if needed:
TIME t
" Connections:
sx1[control]=sx1[pendubot]
sx2[control]=sx2[pendubot]
sx3[control]=sx3[pendubot]
sx4[control]=sx4[pendubot]
su[pendubot]=su[control]
END

```

## DISCRETE SYSTEM CONTROL

```

" Version: 1.0
" Abstract:
" Description:
" Revision: 1.0
" Author:
" Created: 21/05/98
" Inputs and outputs:
INPUT sx1 sx2 sx3 sx4
OUTPUT su
" States and time variables:
"STATE
"NEW
TIME t
TSAMP ts
" Equations:
"Par extra requerido
par90=(m1*lc1+m2*11)*g*COS(90*pime/90.0)
par80=(m1*lc1+m2*11)*g*COS(80*pime/90.0)
par70=(m1*lc1+m2*11)*g*COS(70*pime/90.0)

```



```

par60=(m1*lc1+m2*l1)*g*COS(60*pime/90.0)
par50=(m1*lc1+m2*l1)*g*COS(50*pime/90.0)
par40=(m1*lc1+m2*l1)*g*COS(40*pime/90.0)
par30=(m1*lc1+m2*l1)*g*COS(30*pime/90.0)
par20=(m1*lc1+m2*l1)*g*COS(20*pime/90.0)

```

” Conversión de q1 de grados a radianes

```

Aju1st90=90*pime/90.0
Aju1st80=80*pime/90.0
Aju1st70=70*pime/90.0
Aju1st60=60*pime/90.0
Aju1st50=50*pime/90.0
Aju1st40=40*pime/90.0
Aju1st30=30*pime/90.0
Aju1st20=20*pime/90.0

```

” Conversión de q2 de grados a radianes

```

Aju2st90=(90-90)*pime/90.0
Aju2st80=(90-80)*pime/90.0
Aju2st70=(90-70)*pime/90.0
Aju2st60=(90-60)*pime/90.0
Aju2st50=(90-50)*pime/90.0
Aju2st40=(90-40)*pime/90.0
Aju2st30=(90-30)*pime/90.0
Aju2st20=(90-20)*pime/90.0

```

” Ley de Control

```

u90=K1d90*(sx1-Aju1st90)+K2d90*(sx2-Aju2st90)+K3d90*sx3+K4d90*sx4
+par90
u80=K1d80*(sx1-Aju1st80)+K2d80*(sx2-Aju2st80)+K3d80*sx3+K4d80*sx4
+par80
u70=K1d70*(sx1-Aju1st70)+K2d70*(sx2-Aju2st70)+K3d70*sx3+K4d70*sx4
+par70
u60=K1d60*(sx1-Aju1st60)+K2d60*(sx2-Aju2st60)+K3d60*sx3+K4d60*sx4
+par60
u50=K1d50*(sx1-Aju1st50)+K2d50*(sx2-Aju2st50)+K3d50*sx3+K4d50*sx4
+par50
u40=K1d40*(sx1-Aju1st40)+K2d40*(sx2-Aju2st40)+K3d40*sx3+K4d40*sx4
+par40
u30=K1d30*(sx1-Aju1st30)+K2d30*(sx2-Aju2st30)+K3d30*sx3+K4d30*sx4
+par30
u20=K1d20*(sx1-Aju1st20)+K2d20*(sx2-Aju2st20)+K3d20*sx3+K4d20*sx4
+par20

```

” Aplicación del banco de controladores dependiendo del tiempo transcurrido

```

u2 = IF (t > 5 AND t < 10) THEN u80 ELSE u3

```

```

u3 = IF (t > 10 AND t < 15) THEN u70 ELSE u4
u4 = IF (t > 15 AND t < 20) THEN u60 ELSE u5
u5 = IF (t > 20 AND t < 25) THEN u50 ELSE u6
u6 = IF (t > 25 AND t < 30) THEN u40 ELSE u7
u7 = IF (t > 30 AND t < 35) THEN u30 ELSE u8
u8 = IF (t > 35 AND t < 40) THEN u20 ELSE u9
u9 = IF (t > 40 AND t < 45) THEN u30 ELSE u10
u10 = IF (t > 45 AND t < 50) THEN u40 ELSE u11
u11 = IF (t > 50 AND t < 55) THEN u50 ELSE u12
u12 = IF (t > 55 AND t < 60) THEN u60 ELSE u13
u13 = IF (t > 60 AND t < 65) THEN u70 ELSE u14
u14 = IF (t > 65 AND t < 70) THEN u80 ELSE u90
u = IF (t > 0 AND t < 5) THEN u90 ELSE u2

```

"Determinación del error del eslabón 2

```

e2rror2 = IF (t > 5 AND t < 10) THEN Aju2st80 ELSE e2rror3
e2rror3 = IF (t > 10 AND t < 15) THEN Aju2st70 ELSE e2rror4
e2rror4 = IF (t > 15 AND t < 20) THEN Aju2st60 ELSE e2rror5
e2rror5 = IF (t > 20 AND t < 25) THEN Aju2st50 ELSE e2rror6
e2rror6 = IF (t > 25 AND t < 30) THEN Aju2st40 ELSE e2rror7
e2rror7 = IF (t > 30 AND t < 35) THEN Aju2st30 ELSE e2rror8
e2rror8 = IF (t > 35 AND t < 40) THEN Aju2st20 ELSE e2rror9
e2rror9 = IF (t > 40 AND t < 45) THEN Aju2st30 ELSE e2rror10
e2rror10 = IF (t > 45 AND t < 50) THEN Aju2st40 ELSE e2rror11
e2rror11 = IF (t > 50 AND t < 55) THEN Aju2st50 ELSE e2rror12
e2rror12 = IF (t > 55 AND t < 60) THEN Aju2st60 ELSE e2rror13
e2rror13 = IF (t > 60 AND t < 65) THEN Aju2st70 ELSE e2rror14
e2rror14 = IF (t > 65 AND t < 70) THEN Aju2st80 ELSE (Aju2st90)
error2 = IF (t > 0 AND t < 5) THEN (Aju2st90-sx2) ELSE (e2rror2-sx2)

```

"Determinación del error del eslabón 1

```

elrror2 = IF (t > 5 AND t < 10) THEN Ajulst80 ELSE elrror3
elrror3 = IF (t > 10 AND t < 15) THEN Ajulst70 ELSE elrror4
elrror4 = IF (t > 15 AND t < 20) THEN Ajulst60 ELSE elrror5
elrror5 = IF (t > 20 AND t < 25) THEN Ajulst50 ELSE elrror6
elrror6 = IF (t > 25 AND t < 30) THEN Ajulst40 ELSE elrror7
elrror7 = IF (t > 30 AND t < 35) THEN Ajulst30 ELSE elrror8
elrror8 = IF (t > 35 AND t < 40) THEN Ajulst20 ELSE elrror9
elrror9 = IF (t > 40 AND t < 45) THEN Ajulst30 ELSE elrror10
elrror10 = IF (t > 45 AND t < 50) THEN Ajulst40 ELSE elrror11
elrror11 = IF (t > 50 AND t < 55) THEN Ajulst50 ELSE elrror12
elrror12 = IF (t > 55 AND t < 60) THEN Ajulst60 ELSE elrror13
elrror13 = IF (t > 60 AND t < 65) THEN Ajulst70 ELSE elrror14
elrror14 = IF (t > 65 AND t < 70) THEN Ajulst80 ELSE (Ajulst90)
error1 = IF (t > 0 AND t < 5) THEN (Ajulst90-sx1) ELSE (elrror2-sx1)

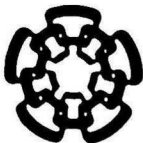
```



```

su=u
ts = t+sample
" Parameter values:
sample:0.005
"Banco de controladores
K1d90 = 16.1949
K2d90 = 16.1104
K3d90 = 3.6175
K4d90 = 2.4406
K1d80 = 16.3814
K2d80 = 16.2907
K3d80 = 3.6378
K4d80 = 2.4700
K1d70 = 16.9603
K2d70 = 16.8507
K3d70 = 3.7013
K4d70 = 2.5609
K1d60 = 18.0058
K2d60 = 17.8642
K3d60 = 3.8197
K4d60 = 2.7250
K1d50 = 19.6858
K2d50 = 19.4980
K3d50 = 4.0193
K4d50 = 2.9879
K1d40 = 22.3792
K2d40 = 22.1271
K3d40 = 4.3605
K4d40 = 3.4081
K1d30 = 27.0395
K2d30 = 26.6981
K3d30 = 4.9953
K4d30 = 4.1333
K1d20 = 36.6704
K2d20 = 36.1998
K3d20 = 6.4005
K4d20 = 5.6301
"parametros parexta=(m1*lc1+m2*l1)*g*COS(Ajuste1)
lc1=0.1551
l1=0.2032
m1=0.8293
m2=0.3402
g=9.81

```



**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN  
UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis: **“Diseño de un banco de controladores lineales para el Pendubot”** del Sr. **Bernardo Haro Martínez** el día 23 de agosto de 1999.

Dr. Edgar Nelson Sanchez Cameros  
Investigador Cinvestav 3B  
CINVESTAV DEL IPN  
Guadalajara

Dr. Bernardino Castillo Toledo  
Investigador Cinvestav 3A  
CINVESTAV DEL IPN  
Guadalajara

Dra. Ofelia Begovich Mendoza  
Investigador Cinvestav 3A  
CINVESTAV DEL IPN  
Guadalajara

Dr. Antonio Ramirez Treviño  
Investigador Cinvestav 2A  
CINVESTAV DEL IPN  
Guadalajara



CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000003837