

INSTITUTO POLITECNICO NACIONAL

**CENTRO DE INVESTIGACIÓN EN CIENCIA
APLICADA Y TECNOLOGIA AVANZADA**

**ANÁLISIS E IMPLEMENTACIÓN DE
LA SIMULACIÓN EN TIEMPO REAL
DEL SISTEMA SUBACTUADO
PENDUBOT**

T E S I S

Que para obtener el grado de
Maestro en Tecnología Avanzada

PRESENTA:

ING. ADOLFO ROJAS PACHECO

DIRECTOR:

DR. JOSÉ DE JESÚS MEDEL JUÁREZ

CO-DIRECTOR:

DR. PEDRO GUEVARA LÓPEZ



MÉXICO D. F. JUNIO DEL 2007



INSTITUTO POLITECNICO NACIONAL
SECRETARÍA DE INVESTIGACION Y POSTRADO

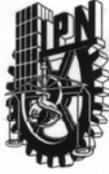
CARTA CESION DE DERECHOS

En la Ciudad de México, D. F. el día 01 del mes junio del año 2007, el que suscribe **Ing. ADOLFO ROJAS PACHECO** alumno del Programa de **MAESTRÍA EN TECNOLOGÍA AVANZADA** con número de registro **A060485**, adscrito a **CICATA LEGARIA**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del **Dr. José de Jesús Medel Juárez** y cede los derechos del trabajo intitulado **ANÁLISIS E IMPLEMENTACIÓN DE LA SIMULACIÓN EN TIEMPO REAL DEL SISTEMA SUBACTUADO PENDUBOT**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección **arrojas75@hotmail.com**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Ing. Adolfo Rojas Pacheco

Nombre y firma



**INSTITUTO POLITECNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

ACTA DE REVISION DE TESIS

En la Ciudad de MÉXICO, D. F. siendo las 11:00 horas del día 22 del mes de mayo del 2007 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CICATA-IPN para examinar la tesis de grado titulada:
ANÁLISIS E IMPLEMENTACIÓN DE LA SIMULACIÓN EN TIEMPO REAL DEL SISTEMA SUBACTUADO PENDUBOT.

Presentada por el alumno:

ROJAS

Apellido paterno

PACHECO

materno

ADOLFO

nombre(s)

Con registro:

A	0	6	0	4	8	5
---	---	---	---	---	---	---

aspirante al grado de:

MAESTRO EN TECNOLOGÍA AVANZADA

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISION REVISORA

Director de tesis

DR. JOSÉ DE JESÚS MEDEL JUÁREZ

DR. JOSE ANTONIO CALDERON ARENAS

DR. ERNESTO MARÍN MOARES

DR. JOSÉ LUIS FERNÁNDEZ MUÑOZ

DR. PEDRO GUEVARA LÓPEZ

EL PRESIDENTE DEL COLEGIO

DR. JOSÉ ANTONIO IRÁN DÍAZ GÓNGORA

Dedicatoria

*A Blanca Rosa, mi amada esposa. Gracias por todo
tu amor, apoyo y comprensión*

A Blanca Olivia, mi querida hija.

*A mis hermanas: María Guadalupe, Olivia Ofelia,
Adriana, Lucía, María de Jesús, Angélica y a todos
mis sobrinos.*

Agradecimientos

Al Dr. José de Jesús Medel Juárez por su amistad y por darme la oportunidad de trabajar con él.

Al Dr. Pedro Guevara López, por su amistad y por compartir sus conocimientos conmigo.

Al Dr. José Antonio Díaz Góngora, Director del CICATA-IPN (Unidad Legaria).

Al Dr. José Antonio Calderón Arenas, Ana Leticia Cardona Gómez, Laura Yadira Vega Haro, Andrés Bautista Huerta, todos de CICATA-IPN (Unidad Legaria), por todo el apoyo prestado en las cuestiones administrativas.

A todos mis compañeros de trabajo en la UPIITA-IPN, por los comentarios y apoyo en la realización de este proyecto, en particular a mis compañeros: Marisela Serrano Fragoso e Isaac Guzmán Domínguez.

Al Instituto Politécnico Nacional, por haberme otorgado el disfrute del año sabático para la realización de este proyecto.

Al CONACYT (No de Registro 67143), por el valioso apoyo económico que se me otorgó.

Índice de Figuras

Figura 1.1 Pendubot	3
Figura 1.2 Definición de coordenadas del sistema pendubot	3
Figura 2.1 Diagrama esquemático del pendubot	10
Figura 3.1 Pendubot	18
Figura 3.2 Respuesta en lazo abierto (inferior)	19
Figura 3.3 Respuesta en lazo abierto (superior)	20
Figura 3.4 Diagrama a bloques del control por linealización Parcial	21
Figura 3.5 Simulación de la ley de par calculado	22
Figura 3.6 Primeros 10 seg de la respuesta del sistema	22
Figura 3.7 Primeros 20 seg de la respuesta del sistema	23
Figura 3.8 Posiciones q1 y q2	24
Figura 3.9 150 seg de simulación, q1 y q2	25
Figura 3.10 Respuesta con el control lineal	26
Recuadro 4.1 pendubotdiscreto.m	31
Recuadro 4.2 pendsys2.m	32
Recuadro 4.3 pendsys.m	33
Figura 4.1 Tiempo de muestreo 10 ms	34
Figura 4.2 Tiempo de muestreo de 50 ms	34
Figura 4.3 Tiempo de muestreo de 100 ms	35
Figura 4.4 Tiempo de muestreo de 500 ms	35
Recuadro 4.4 Cálculo de los momentos de probabilidad	36
Figura 4.5 Tiempo de simulación de 10 seg	37
Figura 4.6 Tiempo de simulación de 20 seg	38
Figura 4.7 Nuevos tiempo de muestreo	38
Figura 4.8 Acercamiento de la fig. 4.7	39
Figura 4.9 Región de muestreos aceptables	39
Figura 4.10 Diagramas de Bode del sistema linealizado	41
Figura 4.11 Diagramas de Bode con etiquetas	41
Figura 4.12 Diagramas de Bode del sistema linealizado	42
Figura 4.13 Diagramas de Bode con etiquetas	42
Figura 4.14 Tiempos de ejecución	43
Figura 4.15 Comparación de tiempos	44
Figura 4.16 Simulación en tiempo real 1	46
Figura 4.17 Simulación en tiempo real 2	46
Figura 4.18 Simulación en tiempo real 3	47
Figura 4.19 Simulación en tiempo real 4	47
Figura 4.20 Simulación en tiempo real 5	48

Índice General

RESUMEN	ix
ABSTRACT	x
1. Introducción.	
1.0 Introducción	1
1.1 Objetivo (Objetivos específicos)	1
1.2 Antecedentes	2
1.3 Planteamiento del problema	5
1.4 Justificación	6
1.5 Contribución del trabajo	6
1.6 Metodología de solución	7
1.7 Hipótesis	7
1.8 Panorama general de la memoria	7
1.9 Conclusiones	8
2. Modelado y análisis del sistema pendubot.	
2.0 Introducción	9
2.1 Modelo dinámico del pendubot	10
2.2 Análisis del modelo matemático	14
2.2.1 Representación en variables de estado	14
2.2.2 Puntos de equilibrio	14
2.3 Aproximación lineal del modelo matemático	16
3. Control del sistema pendubot.	
3.0 Introducción	18
3.1 Simulaciones en lazo abierto	19
3.2 Leyes de control para el balanceo	20
3.2.1 Esquema de prealimentación y PD	20
3.2.2 Control basado en energía	23
3.3 Ley de control para la estabilización	25
4. Simulación en tiempo real del sistema pendubot.	
4.0 Introducción	27
4.1 Conceptos de sistemas en tiempo real	27
4.2 Modelo en tiempo discreto del sistema pendubot	29
4.3 Desarrollo del análisis para obtener el tiempo de muestreo	32
4.3.1 Mediante el segundo momento de probabilidad	32
4.3.2 Diagramas de Bode	40
4.4 Obtención de los tiempos de ejecución para el código del sistema pendubot	43
4.5 Comparación de los tiempos de ejecución y de muestreo	44
4.6 Implementación de la simulación en tiempo real del sistema pendubot	45

5. Conclusiones y perspectivas.	49
REFERENCIAS	52
Apéndice A. Programas en QNX.	55
Apéndice B. Valores numéricos de los parámetros del sistema pendubot.	59

RESUMEN

El control de un sistema no lineal (usando cualquier técnica) a través de un sistema digital en tiempo real, requiere de la selección del mejor tiempo de muestreo y del mejor resultado en calidad de respuesta respecto a una función de convergencia previamente establecida para asegurar la estabilidad del sistema. Si se tiene interés en emplear un sistema no lineal para probar nuevas técnicas de control a nivel de simulación en un sistema digital de tiempo real, se deben tomar en cuenta los tiempos de muestreo y de ejecución tanto del propio sistema como de la ley de control.

En este trabajo se desarrolla y prueba una metodología para la determinación del rango aceptable en los tiempos de muestreo y de ejecución para la implementación de la simulación en tiempo real de un sistema no lineal conocido como pendubot, considerando tanto la respuesta del sistema en lazo abierto como la respuesta con algunas leyes de control.

ABSTRACT

The control of a nonlinear system (by any technique) through a real-time digital system, needs the better choice of the sampling time and the better quality in the response respect to a convergence function previously established by assure the system stability. If we are interested in the use of a nonlinear system to probe new techniques of control in simulation in a real-time digital system, we most to consider the samplings time and the executions time as the own system as the control law.

In this thesis we develop and probe a methodology to calculate the acceptable range in the samplings time and in the execution time to the real-time simulation of the nonlinear system called pendubot, considering the time response as in open loop as with some control laws.

CAPÍTULO 1

INTRODUCCIÓN

1.0 INTRODUCCIÓN

En este capítulo se explica el por qué del desarrollo de este proyecto, partiendo del planteamiento de los objetivos, de los antecedentes y el estado del arte, la justificación del proyecto y temas que permitan ubicar y dimensionar el tema de trabajo, así como los resultados obtenidos del mismo.

1.1 OBJETIVO

Desarrollar una sistema de simulación en tiempo real considerando al sistema subactuado pendubot.

OBJETIVOS ESPECÍFICOS

- ▲ Desarrollar el análisis de las características de tiempo real del pendubot.
- ▲ Desarrollar la simulación en tiempo real del pendubot considerando las características determinadas.
- ▲ Aplicar en simulación un método de control basado en pasividad para el balanceo del pendubot y del método del Lagrangianos Controlados a su estabilización.
- ▲ Implementar en tiempo real las leyes de control obtenidas.

1.2 ANTECEDENTES

Sistemas subactuados y el pendubot.

Los sistemas subactuados son sistemas mecánicos que tienen menor cantidad de actuadores que grados de libertad y surgen de varias formas:

- Por diseño. Los sistemas subactuados por diseño más comunes son prototipos de laboratorio, como por ejemplo el robot “brachiation” de Fukuda [20], el Acrobot [4], el péndulo invertido sobre un carro [5] y el péndulo invertido sobre un brazo rotatorio [3], el pendubot [7] y más recientemente el péndulo con volante de inercia [8 y 6]. Estos sistemas son robots multieslabones que tienen motores (actuadores) sólo en algunas de las uniones.
- Sistemas móviles.- Existen una gran variedad de sistemas subactuados móviles, como por ejemplo: carros, satélites, submarinos, barcos. Sin embargo, se tienen sistemas subactuados móviles más complejos al colocar un brazo manipulador montado sobre una plataforma móvil, por ejemplo una plataforma espacial o un vehículo submarino [14]. Aquí las fuerzas de reacción debido al movimiento del brazo están acopladas a la plataforma que no puede ser considerada como un punto de referencia inercial. Para este tipo de sistemas se usan controles adicionales para mantener el control de posición de la base. Sin embargo, si estos actuadores fallan o son apagados para ahorrar energía, entonces el sistema completo es subactuado.
- Por modelo.- Una tercera forma en que surgen los sistemas subactuados es debida al modelo matemático que se esté usando para el diseño del controlador; por ejemplo, cuando la flexibilidad de la unión se incluye en el modelo [21].

Un sistema subactuado ampliamente usado es el Pendubot, debido a que permite ilustrar la aplicación y desempeño de diferentes técnicas de control (PID's, ley de control lineal con ganancias conmutadas, leyes de control lineales y no lineales, leyes de control difuso y por redes neuronales, conmutación de leyes, etc.) y representa un banco de pruebas adecuado para probar la validez y desempeño de nuevas leyes de control.

El pendubot, nombre formado por la contracción de las palabras PENDULum roBOT, es un sistema electromecánico que consiste de dos eslabones rígidos interconectados por uniones de revolución. La primera unión está actuada por un motor de CD y la segunda se encuentra sin actuador (figura 1). De esta forma, el segundo eslabón se puede ver como un péndulo simple cuyo movimiento puede ser controlado por la actuación del primero. El pendubot es similar en espíritu al péndulo invertido clásico sobre un carro o al más reciente péndulo invertido rotacional (de Furuta). Sin embargo, debido a la dinámica no lineal de acoplamiento entre los dos eslabones, el pendubot posee algunas características únicas para la investigación en control, que no se encuentran en otros dispositivos [7].

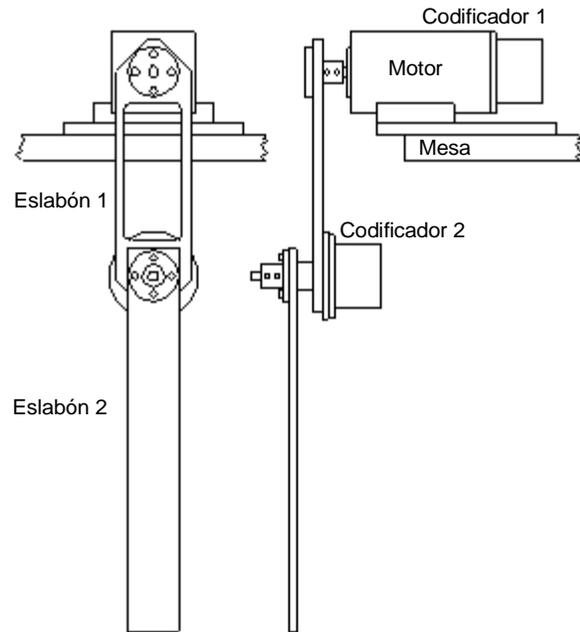


Figura 1.1 Pendubot

Con este sistema se pueden ilustrar un gran número de conceptos fundamentales en dinámica no lineal y teoría del control [9]. Por ejemplo, para ilustrar cómo la linealización con series de Taylor depende del punto de operación. También, para entender en forma física que el sistema linealizado es no controlable en algunos de sus puntos de equilibrio.

De aquí en adelante, se identificarán con (q_1, q_2) las posiciones angulares de los eslabones 1 y 2, respectivamente. Los ejes de referencia de cada eslabón se toman de acuerdo a como se ilustra en la figura 2. De acuerdo al modelo dinámico, el sistema cuenta con cuatro puntos de equilibrio principales (con entrada cero), uno estable: $(-90^\circ, 0^\circ)$ y tres inestables: $(90^\circ, 180^\circ)$, $(-90^\circ, 180^\circ)$ y $(90^\circ, 0^\circ)$.

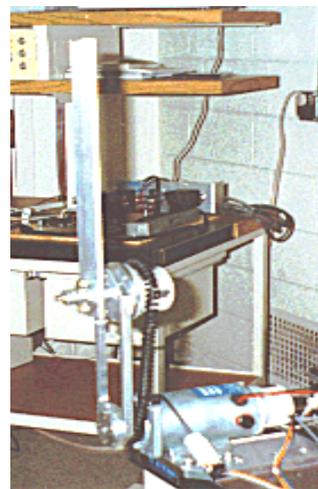
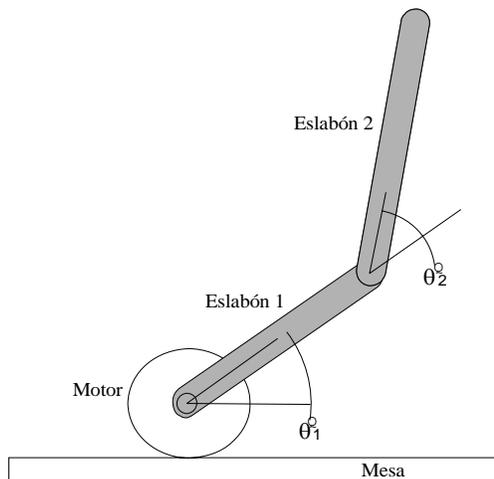


Figura 1.2 Definición de coordenadas. Pendubot en la posición superior.

En este sistema los objetivos básicos de control son:

- Balanceo, el cual consiste en llevar los eslabones desde su punto de equilibrio estable $(-90^{\circ}, 0^{\circ})$ a su punto de equilibrio inestable $(90^{\circ}, 0^{\circ})$, a través de los movimientos del primer eslabón.
- Estabilización, que consiste en hacer estable el punto de equilibrio $(90^{\circ}, 0^{\circ})$.
- Seguimiento, en el cual se trata de que el sistema una vez estabilizado en el punto $(90^{\circ}, 0^{\circ})$, siga una referencia de posición variante en el tiempo.

Uno de los problemas de control que presenta el sistema pendubot, es que su modelo matemático es altamente no lineal y por lo tanto es difícil encontrar una ley de control única que resuelva los tres objetivos antes mencionados.

Técnicas de control empleadas en el pendubot

El sistema ha sido estudiado y se han probado algunas leyes de control con las cuales se han obtenido resultados limitados. En la mayoría de los casos se han empleado leyes de control formadas de dos partes separadas, el control de balanceo y el control de estabilización. Para el diseño de cada controlador se han utilizado diferentes técnicas desde el diseño de controladores no lineal, hasta el uso de técnicas de control inteligente. En [8] se emplea el método de linealización por retroalimentación parcial para diseñar el controlador que resuelve el problema del balanceo. Para la segunda parte, el control que resuelve el problema de la estabilización, las ecuaciones de movimiento no lineales del pendubot se linealizan por medio de la expansión en series de Taylor y de ahí se obtiene un controlador con retroalimentación completa del estado.

En [11] el pendubot se usa como un ejemplo para ilustrar el empleo de una técnica para la reducción de sistemas mecánicos subactuados de alto orden. Esta reducción se realiza con el objetivo de permitir la aplicación de métodos de diseño no lineal existentes, como “saturaciones anidadas y prealimentación”, para el sistema subactuado.

En [13] se usa control óptimo (LQR) y un regulador lineal con retroalimentación del estado y un regulador no lineal con retroalimentación del estado para resolver la estabilización del sistema en su punto de equilibrio inestable. Para el problema del balanceo usan las técnicas de [15].

En [12] se usa “backsteeping” para atacar el problema de la estabilización. En [16] y [17] usan técnicas de control inteligente para la estabilización y el balanceo del pendubot.

Por el lado de las técnicas de control basadas en energía se tiene [22] en donde el controlador que proponen para el balanceo del pendubot no está basado en las técnicas estándar de linealización por retroalimentación, sino que consideran las propiedades de pasividad del pendubot y usa un enfoque basado en energía para establecer la ley de control propuesta. Según se demuestra en [22], con esta técnica la amplitud de la entrada de control no necesita ser muy grande ya que en cada ciclo (del segundo eslabón) sólo se requiere incrementar la energía ligeramente. En otras palabras no se requieren controladores con altas ganancias.

Como se puede ver, al pendubot se le han aplicado principalmente técnicas de control no lineal relacionadas con la linealización por retroalimentación de estados para el balanceo y la estabilización. Y en menor medida, se han utilizado algunas aplicaciones con pasividad, lógica difusa y redes neuronales.

Lagrangianos Controlados.

Una metodología para el diseño de leyes de control estabilizantes, basada en la manipulación de la energía del sistema, es el método de Lagrangianos Controlados (LC), el cual ha sido aplicado a sistemas subactuados como el péndulo invertido esférico y el péndulo invertido sobre un carro, que se mueven en un plano horizontal o inclinado, y el péndulo invertido rotacional [23], [24]. El método LC involucra varios pasos a seguir. En general, primero se hacen modificaciones adecuadas [24] al Lagrangiano del sistema sin controlar, construyendo lo que los autores han llamado el “Lagrangiano Controlado”. Las ecuaciones de Euler-Lagrange derivadas del Lagrangiano controlado describen el sistema de lazo cerrado donde los nuevos términos son identificados con las fuerzas de control.

En la actualidad se encuentra bastante desarrollado el método y se ha aplicado a nivel de simulación en algunos sistemas mecánicos sencillos [25] y sólo se ha encontrado una aplicación a un prototipo de laboratorio del péndulo de Furuta [26].

1.3 PLANTEAMIENTO DEL PROBLEMA

El control de un sistema no lineal (usando cualquier técnica) a través de un sistema digital en tiempo real, requiere de la selección del mejor tiempo de muestreo y del mejor resultado en calidad de respuesta respecto a una función de convergencia previamente establecida, para asegurar la estabilidad del sistema. Si se tiene interés en emplear un sistema no lineal para probar nuevas técnicas de control a nivel de simulación en un sistema digital de tiempo real, se deben tomar en cuenta los tiempos de muestreo y de ejecución tanto del propio sistema como de la ley de control, esto nos lleva a la siguiente pregunta, *¿Cómo simular un sistema no lineal, en particular un sistema pendubot y su ley de control, en un sistema digital de tiempo real?* Dicho de otra forma: *¿Es posible controlar en tiempo real al sistema pendubot considerando su balanceo y su estabilización?*

Por lo tanto y para responder lo anterior se plantea lo siguiente:

Diseñar y probar una simulación en tiempo real, de un conjunto de leyes de control que estabilice el sistema subactuado pendubot alrededor de sus puntos de equilibrio, estable $q_1 = -90^\circ$, $q_2 = 0^\circ$, e inestable $q_1 = 90^\circ$, $q_2 = 0^\circ$ y lo mantenga en equilibrio (lo estabilice) a pesar de las pequeñas perturbaciones que en él se generen. Se deben tomar en cuenta a las diferentes partes que lo forman para la estabilización del equilibrio inestable:

- Balanceo.- Probar una ley de control basada en pasividad [22], que permita desplazar el sistema desde el punto de equilibrio $q_1 = -90^\circ$, $q_2 = 0^\circ$, hasta una vecindad del punto de equilibrio inestable $q_1 = 90^\circ$, $q_2 = 0^\circ$.
- Estabilización.- Probar una ley de control lineal que vuelva asintóticamente estable el punto de equilibrio inestable. La ley de control lineal estabilizante puede usar técnicas conocidas.
- Desarrollar el análisis de las características de tiempo real del sistema para determinar el tiempo de muestreo adecuado.
- Evaluación experimental a nivel de simulación en tiempo real del esquema de control resultante.

1.4 JUSTIFICACIÓN

Los sistemas físicos para realizar experimentos de control son caros y complejos dado que en ellos se consideran diferentes situaciones que nada tienen que ver con la prueba de la técnica de control bajo análisis.

Por otro lado, en el caso de una simulación en tiempo real, resulta ser más económica, centrada en el análisis de las técnicas de control, permite observar el comportamiento del sistema de manera precisa y seleccionar las mejores condiciones de control y tiempos de muestreo considerando para ello índices de desempeño.

1.5 CONTRIBUCIÓN DEL TRABAJO

- La simulación en tiempo real del sistema pendubot y sus leyes de control, para la estabilización.
- Determinación de los tiempos de muestreo para un sistema no lineal.
- Evaluación de la efectividad de estas técnicas de control, mediante la aplicación en simulaciones de tiempo real.

1.6 METODOLOGÍA DE SOLUCIÓN

Para resolver el problema de estabilización se empleó el método de Lagrangianos Controlados (LC), ya que este método ha probado proporcionar estabilización del

espacio de estado completo con grandes regiones de atracción para sistemas subactuados tales como el péndulo invertido sobre un carro [24].

Para el problema del balanceo se empleó la técnica de control basado en pasividad [22], la cual está relacionada con el método de lagrangianos controlados, en el sentido de ser una técnica basada en el moldeo de la energía.

La evaluación del desempeño de estos esquemas de control se realizó mediante la implementación de la simulación en tiempo real del sistema bajo un sistema operativo capaz de soportar tales simulaciones, como lo es QNX. Se desarrollaron las ecuaciones del sistema pendubot en variables de estado de tiempo discreto, se midieron los tiempos de ejecución y se determinaron los tiempos de muestreo adecuados.

1.7 HIPOTESIS

Es posible desarrollar una simulación en tiempo real para el sistema subactuado no lineal, pendubot, considerando sus leyes de control obtenidas para el sistema en tiempo continuo.

1.8 PANORAMA GENERAL DE LA MEMORIA

En el capítulo 2 se realiza la obtención y el análisis del modelo matemático del sistema pendubot, el cual resulta ser altamente no lineal y con un número infinito de puntos de equilibrio. En el capítulo 3, se describen y diseñan las técnicas de control, en tiempo continuo, que se usarán en este trabajo para resolver el problema del balanceo y estabilización del pendubot. En el capítulo 4 se desarrolla la obtención y el análisis de las características de tiempo real del sistema, de tal forma que sea posible la implementación de la simulación en tiempo real del modelo matemático en tiempo discreto. Finalmente, en el capítulo 5, se dan las conclusiones a las que este trabajo ha permitido llegar en cuanto al objetivo planteado.

1.9 CONCLUSIONES

En el presente capítulo se hizo una descripción del sistema no lineal y subactuado conocido como pendubot, de sus objetivos de control y de las técnicas de control que se han empleado para resolverlos. Se realizó el planteamiento del problema que ocupa este trabajo, en el cual se cuestiona la posibilidad de implementar en tiempo real, la simulación del sistema y de sus leyes de control. Se realizó la descripción de la metodología para resolver el problema planteado, estableciendo que el objetivo principal de la tesis es demostrar la hipótesis planteada, la cual consiste en desarrollar la simulación y la determinación de las características de tiempo real del sistema pendubot.

CAPÍTULO 2

MODELADO Y ANÁLISIS DEL SISTEMA PENDUBOT

2.0 Introducción.

La obtención de un buen modelo matemático del sistema electromecánico subactuado (Pendubot) es importante para predecir su funcionamiento, ya que la validez de la predicción y el comportamiento de las estructuras de control diseñadas dependen, en gran medida, del modelo dinámico utilizado. El sistema electromecánico subactuado puede verse como un péndulo invertido, en el que el motor hace las veces del carro. Este sistema es, en general, difícil de controlar debido a que el segundo eslabón está libre y que su modelo es de fase no mínima cuando se selecciona como salida el desplazamiento angular de este eslabón.

La obtención del modelo dinámico del sistema subactuado se hace, en principio, sin considerar la fricción, bajo las siguientes suposiciones

- a) Cada eslabón es un cuerpo rígido.
- b) El sistema es del tipo conservativo sujeto a restricciones holonomas.

Posteriormente se obtiene un segundo modelo tomando en cuenta las fricciones. Para esto se hace uso del método desarrollado en [32] el cual permite derivar una cierta fuerza de amortiguamiento.

El objetivo de este capítulo es la obtención y análisis del modelo dinámico del sistema electromecánico subactuado. Este capítulo está dividido en tres secciones. En la sección 2.1 se presenta el modelo dinámico del sistema subactuado si considerar la fricción viscosa, dado que este modelo es el que

se utiliza en el resto del trabajo. En la sección 2.2 se hace el análisis del modelo dinámico, encontrando los puntos de equilibrio del sistema subactuado para un par u^* en el eje del motor. En la sección 2.3 se obtiene una aproximación lineal del modelo, así como también el cálculo de los polos y ceros del sistema linealizado.

2.1 Modelo dinámico del pendubot.

Debido a la naturaleza del sistema electromecánico subactuado se pueden encontrar ecuaciones diferenciales no lineales invariantes en el tiempo que determinan su comportamiento dinámico. Estas ecuaciones diferenciales se obtienen por el método de Euler-Lagrange.

Considérese el sistema mostrado en la figura 2.1, donde las coordenadas generalizadas necesarias para describir el movimiento del sistema son el desplazamiento angular del primer y segundo eslabón, $q_1(t)$ y $q_2(t)$, respectivamente. Además, l_1 longitud del eslabón uno, l_2 longitud del eslabón dos, l_{c1} longitud media del eslabón uno, l_{c2} longitud media del eslabón dos, m_1 masa del eslabón uno, m_2 masa del eslabón dos, I_1 momento de inercia del eslabón uno, I_2 momento de inercia del eslabón dos, h_1 distancia medida del centro de masa al eje x del eslabón uno, h_2 distancia medida del centro de masa al eje x del eslabón dos y g aceleración de la gravedad.

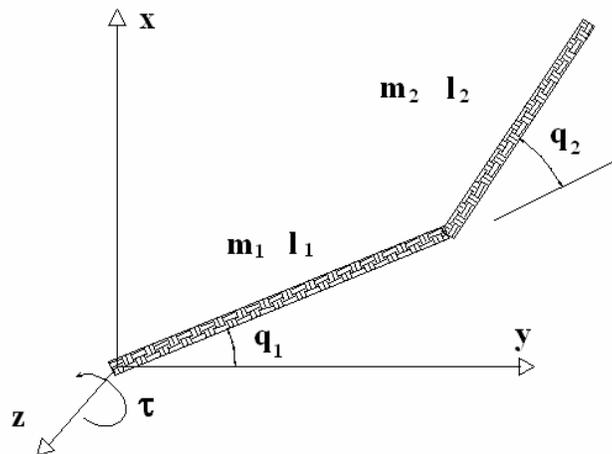


Figura 2.1. Diagrama esquemático del pendubot.

La ecuación de Euler-Lagrange para un sistema conservativo está dada por [34]

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

donde q y \dot{q} son vectores de dimensión n , τ es un vector de fuerzas generalizadas de dimensión n , y L el lagrangiano formado por la diferencia entre la energía cinética K y la energía potencial V , esto es

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

La energía cinética K para el sistema mostrado en la figura 2.1 está dada por

$$K = \frac{1}{2} \left(\sum_{i=1}^2 m_i v_{c_i}^T v_{c_i} + \omega_i^T I \omega_i \right) \quad (2.3)$$

donde v_{c_i} y ω_i son los vectores de velocidad lineal y de velocidad angular asociado al i -ésimo eslabón, cada uno de dimensión 2 y 3, respectivamente e I es el tensor de inercia de dimensión 3x3.

Para el cálculo del vector de velocidad lineal se considera un vector de posición del eje de referencia al centro de masa de cada eslabón, se supone que las masas están concentradas en la mitad de los eslabones. Para el primer eslabón, se tiene entonces que el vector de posición del origen al centro de masa es

$$r_{c1}(q) = \begin{bmatrix} r_1(q) \\ r_2(q) \end{bmatrix} = \begin{bmatrix} l_{c1} \cos(q_1) \\ l_{c1} \sin(q_1) \end{bmatrix} \quad (2.4)$$

donde $r_1(q)$ y $r_2(q)$ es la proyección del centro de masa del eslabón uno sobre el eje x y y , respectivamente. Derivando (2.4) con respecto al tiempo se tiene que

$$\dot{r}_{c1}(q) = v_{c1} = \begin{bmatrix} -l_{c1} \dot{q}_1 \sin(q_1) \\ l_{c1} \dot{q}_1 \cos(q_1) \end{bmatrix} \quad (2.5)$$

Para el segundo eslabón, el vector de posición del origen al centro de masa está dado por

$$r_{c2}(q) = \begin{bmatrix} r_4(q) \\ r_5(q) \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2) \end{bmatrix} \quad (2.6)$$

donde $r_4(q)$ y $r_5(q)$ es la proyección del centro de masa del eslabón dos sobre el eje x y y respectivamente. De igual manera derivando (2.6) con respecto al tiempo, se tiene que

$$\dot{r}_{c2}(q) = \begin{bmatrix} -l_1 \sin(q_1) - l_{c2} \sin(q_1 + q_2) & -l_{c2} \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2) & l_{c2} \cos(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (2.7)$$

Las velocidades angulares están dadas por

$$\omega_1 = [\omega_{1x} \quad \omega_{1y} \quad \omega_{1z}]^T = [0 \quad 0 \quad \dot{q}_1]^T, \quad (2.8)$$

$$\omega_2 = [\omega_{2x} \quad \omega_{2y} \quad \omega_{2z}]^T = [0 \quad 0 \quad \dot{q}_1 + \dot{q}_2]^T \quad (2.9)$$

El tensor de inercia se define como [13]

$$I_i = \begin{bmatrix} \frac{a_i l_i b_i \rho}{3} (a_i^2 + l_i^2) & -\frac{a_i l_i b_i \rho}{4} b_i l_i & -\frac{a_i l_i b_i \rho}{4} a_i b_i \\ -\frac{a_i l_i b_i \rho}{4} b_i l_i & \frac{a_i l_i b_i \rho}{3} (a_i^2 + b_i^2) & -\frac{a_i l_i b_i \rho}{4} a_i l_i \\ -\frac{a_i l_i b_i \rho}{4} a_i b_i & -\frac{a_i l_i b_i \rho}{4} a_i l_i & \frac{a_i l_i b_i \rho}{3} (b_i^2 + l_i^2) \end{bmatrix} \quad (2.10)$$

donde a_i , l_i y b_i son el espesor, longitud y ancho del i -ésimo eslabón respectivamente y ρ es la densidad de masa del material de los eslabones. Sustituyendo las ecuaciones (2.5), (2.7), (2.8), (2.9) y (2.10) en la ecuación (2.3), K está dada por

$$K = \frac{1}{2} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}^T \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (2.11)$$

o bien

$$K = \frac{1}{2} (D_{11} \dot{q}_1^2 + 2D_{12} \dot{q}_1 \dot{q}_2 + D_{22} \dot{q}_2^2) \quad (2.12)$$

donde

$$\begin{aligned} D_{11} &= m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2} \\ D_{12} &= m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2} \\ D_{22} &= m_2 l_{c2}^2 + I_{zz2} \end{aligned} \quad (2.13)$$

La energía potencial V para el sistema mostrado en la figura 2.1 es

$$V = \sum_{i=1}^2 m_i g_y h_i \quad (2.14)$$

donde m_i es la masa del i -ésimo eslabón, h_i es la distancia entre el centro de masa y el eje x del i -ésimo eslabón y g_y es la aceleración debido a la gravedad en dirección del eje y . De la figura 2.1 se observa que

$$\begin{aligned} h_1 &= l_{c1} \sin(q_1) \\ h_2 &= l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2) \end{aligned} \quad (2.15)$$

por lo que

$$V = m_1 g l_{c1} \sin(q_1) + m_2 g (l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2)) \quad (2.16)$$

El Lagrangiano del sistema electromecánico subactuado está entonces dado por

$$\begin{aligned} L &= \frac{1}{2} (D_{11} \dot{q}_1^2 + 2D_{12} \dot{q}_1 \dot{q}_2 + D_{22} \dot{q}_2^2) - m_1 g l_{c1} \sin(q_1) \\ &\quad - m_2 g (l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2)) \end{aligned} \quad (2.17)$$

se obtienen los correspondientes términos de la ecuación (2.1), y la dinámica del pendubot queda descrita por

$$\begin{aligned} \tau_1 &= (m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2}) \ddot{q}_1 \\ &\quad + (m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \ddot{q}_2 \\ &\quad - 2m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \sin(q_2) - m_2 l_1 l_{c2} \dot{q}_2^2 \sin(q_2) \\ &\quad + m_1 g l_{c1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2)), \end{aligned} \quad (2.18)$$

$$\begin{aligned} 0 &= (m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \ddot{q}_1 + (m_2 l_{c2}^2 + I_{zz2}) \ddot{q}_2 \\ &\quad + m_2 l_1 l_{c2} \dot{q}_1^2 \sin(q_2) + m_2 g l_{c2} \cos(q_1 + q_2). \end{aligned} \quad (2.19)$$

Estas ecuaciones pueden escribirse en forma más compacta como

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau \quad (2.20)$$

donde $q = [q_1 \quad q_2]^T$, $\dot{q} = [\dot{q}_1 \quad \dot{q}_2]^T$, $\ddot{q} = [\ddot{q}_1 \quad \ddot{q}_2]^T$, y

$$\begin{aligned} D(q) &= \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix}, \\ C(q, \dot{q}) &= \begin{bmatrix} -2m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) & -m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) \\ m_2 l_1 l_{c2} \dot{q}_1 \sin(q_2) & 0 \end{bmatrix}, \\ G(q) &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} m_1 g l_{c1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2)) \\ m_2 g l_{c2} \cos(q_1 + q_2) \end{bmatrix}, \\ \tau &= \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} \end{aligned} \quad (2.21)$$

$D(q)$ es la matriz simétrica y definida positiva de inercias, $C(q, \dot{q})$ es la matriz Coriolis-centrípeta, (los términos que tienen un producto del tipo \dot{q}_i^2 son

llamados fuerza centrípeta, cuando éste tiene un producto del tipo $\dot{q}_i \dot{q}_j$ $i \neq j$ son llamados términos de Coriolis) y $G(q)$ es el vector que contiene el término debido al efecto de la gravedad.

2.2 Análisis del modelo matemático.

2.2.1 Representación en variables de estado

El modelo dinámico del pendubot sin fricción puede representarse en variables de estado [33] al hacer la asignación

$$x_1 = q_1, x_2 = q_2, x_3 = \dot{q}_1, x_4 = \dot{q}_2, u = \tau_1, y = x_2 \quad (2.22)$$

Con esta asignación la dinámica subactuada está dada por

$$\begin{aligned} \dot{x}(t) &= f(x) + g(x)u(t) \\ y(t) &= h(x) \end{aligned} \quad (2.23)$$

donde

$$\begin{aligned} f(x) &= \begin{bmatrix} x_3 \\ x_4 \\ \frac{D_{22}}{D_{11}D_{22} - D_{12}^2} \left\{ \frac{D_{12}C_2}{D_{22}} + \frac{D_{12}G_2}{D_{22}} - C_1 - G_1 \right\} \\ \frac{-D_{12}}{D_{11}D_{22} - D_{12}^2} \left\{ \frac{D_{12}C_2}{D_{22}} + \frac{D_{12}G_2}{D_{22}} - C_1 - G_1 \right\} - \frac{C_2}{D_{22}} - \frac{G_2}{D_{22}} \end{bmatrix} \\ g(x) &= \begin{bmatrix} 0 \\ 0 \\ \frac{D_{22}}{D_{11}D_{22} - D_{12}^2} \\ \frac{-D_{12}}{D_{11}D_{22} - D_{12}^2} \end{bmatrix}, \quad h(x) = x_2, \end{aligned} \quad (2.24)$$

donde

$$\begin{aligned} C_1 &= -2m_2l_1l_{c2}x_3x_4 \sin(x_2) - m_2l_1l_{c2}x_4^2 \sin(x_2), \\ C_2 &= m_2l_1l_{c2}x_3^2 \sin(x_2) \end{aligned} \quad (2.25)$$

y D_{ij} , G_k están dadas por las ecuaciones (2.13) y (2.21), respectivamente.

Es importante hacer notar que la selección de x_2 como la variable de salida del sistema se debe a la importancia de la posición del segundo eslabón, ya que éste se encuentra subactuado, además, de que si se define una salida

diferente debe tomarse en cuenta la restricción física del sistema para seguimiento de trayectorias.

Al sustituir los valores numéricos de los parámetros del sistema electromecánico que se encuentra en el apéndice B, en la ecuación (2.24) se tiene que

$$f(x) = \begin{bmatrix} x_3 \\ x_4 \\ 0.029094[\alpha] \\ -(0.029094 + 0.0173445 \cos(x_2))[\alpha] - 21.6708 \cos(x_1 + x_2) - 0.596156 x_3^2 \sin(x_2) \end{bmatrix}$$

donde

$$\alpha = \frac{-1.39803 \cos(x_1) + 0.187935 \cos(x_1 + 2x_2) + 0.0173445 x_3^2 \sin(x_2) + 0.0346891 x_3 x_4 \sin(x_2) + 0.0173445 x_4^2 \sin(x_2) + 0.0051 x_3^2 \sin(2x_2)}{0.00124209 - 0.000150416 \cos(2x_2)}$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ 0.029094 \\ \frac{0.00124209 - 0.000150416 \cos(2x_2)}{-(0.029094 + 0.0173445 \cos(x_2))} \\ \frac{0.00124209 - 0.000150416 \cos(2x_2)}{0.00124209 - 0.000150416 \cos(2x_2)} \end{bmatrix}$$

2.2.2 Puntos de equilibrio.

Para propósitos del análisis del sistema y determinación de sus características más importantes es necesario encontrar los puntos de equilibrio del sistema, esto es, encontrar $(x, y) = (x^0, y^0)$, tales que $f(x^0, y^0) \equiv 0$. A partir de la condición $\dot{x}_1 = 0$, y $\dot{x}_2 = 0$ se tiene que

$$x_3 = x_3^* = 0 \text{ y } x_4 = x_4^* = 0 \quad (2.26)$$

A partir de estos valores y de las ecuaciones (2.23)-(2.24), se tiene que

$$\dot{x}_3 = p \cdot q = 0 \quad (2.27)$$

donde $p = \frac{D_{22}}{D_{11}D_{22} - D_{12}^2}$, $q = \frac{D_{12}G_2}{D_{22}} - G_1 + u^*$. De la ecuación (2.27) se tiene que $p = 0$ y $q \neq 0$ ó $p \neq 0$ y $q = 0$ ó $p = q = 0$. Como se puede observar el término p nunca es cero, lo que implica que $q = 0$, esto es,

$$\frac{D_{12}G_2}{D_{22}} - G_1 + u^* = 0 \quad (2.28)$$

Como $D_{22} \neq 0$, (2.28) también se puede escribir como

$$D_{12}G_2 - (G_1 - u^*)D_{22} = 0 \quad (2.29)$$

Por otro lado, al sustituir la ecuación (2.26) en la expresión para x_4 de (2.23)-(2.24) se tiene que

$$\dot{x}_4 = -\frac{D_{12}}{D_{11}D_{22} - D_{12}^2} \left\{ \frac{D_{12}G_2}{D_{22}} - G_1 + u^* \right\} - \frac{G_2}{D_{22}} = 0 \quad (2.30)$$

A partir de esta última ecuación y de (2.28) se obtiene que

$$G_2(x^*) = m_2 g l_{c2} \cos(x_1 + x_2) = 0 \quad (2.31)$$

es decir,

$$x_1^* + x_2^* = n \frac{\pi}{2} = 90^\circ n; \quad \forall n = \pm 1, \pm 3, \pm 5... \quad (2.32)$$

ya que $G_2(x^*) = 0$, la ecuación (2.29) se transforma en

$$(G_1 - u^*)D_{22} = 0 \quad (2.33)$$

lo que implica que

$$\begin{aligned} u^* &= G_1(x^*) = m_1 g l_{c1} \cos(x_1) + m_2 g (l_1 \cos x_1 + m_2 g l_{c2} \cos(x_1 + x_2)) \\ &= (m_1 g l_{c1} + m_2 g l_1) \cos x_1^* \end{aligned} \quad (2.34)$$

de donde, para un par u^* en el eje del motor

$$x_1^* = \arccos \frac{u^*}{(m_1 g l_{c1} + m_2 g l_1)} \quad (2.35)$$

Los puntos de equilibrio del sistema electromecánico subactuado están entonces determinados por las ecuaciones (2.26), (2.32) y (2.35) para un par u^* en el eje del motor dado.

2.3 Aproximación lineal del modelo matemático.

En esta sección se desarrolla la aproximación del modelo en variables de estado del sistema subactuado. La aproximación lineal se hace en los puntos de equilibrio ya que corresponden a aquellos en los cuales se quiere regular el sistema.

Si se definen las variables incrementales $\delta x = x - x^*$, $\delta u = u - u^*$, $\delta y = y - y^*$ donde el par (x^*, u^*) corresponde a un punto de equilibrio del sistema y

$y^* = h(x^*)$, la aproximación en serie de Taylor del sistema (2.23) está dado por:

$$\begin{aligned}\delta\dot{x}(t) &= A\delta x(t) + B\delta u(t) \\ \delta y(t) &= C\delta x(t)\end{aligned}\tag{2.36}$$

con $x^* = [90^\circ n \ 0 \ 0 \ 0]$ $\forall n = \pm 1, \pm 3, \pm 5, \dots, u^* = 0$ y

$$A = \left. \frac{\partial f(x^*)}{\partial x} \right|_{x=x^*} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \pm 32.2499 & \mp 10.0172 & 0 & 0 \\ \mp 29.8052 & \pm 37.6597 & 0 & 0 \end{bmatrix}$$

$$B = g(x^*) = \begin{bmatrix} 0 \\ 0 \\ 26.6507 \\ -42.5387 \end{bmatrix}$$

$$C = \left. \frac{\partial h}{\partial x} \right|_{x=x^*} = [0 \ 1 \ 0 \ 0]$$

donde se han sustituido los valores numéricos de los parámetros del sistema del apéndice B. Del correspondiente análisis de la aproximación lineal, puede verse que para el punto de equilibrio $x^* = [90^\circ m \ 0 \ 0 \ 0]$ con $m = 1$, los valores propios de la matriz A son $p_{1,2} = \pm 7.2418$ y $p_{3,4} = \pm 4.1792$ los cuales son inestables (puntos de equilibrio tipo silla [35]). Mientras que para el punto $x^* = [90^\circ z \ 0 \ 0 \ 0]$ con $z = -1$, los valores propios de la matriz A son $p_{1,2} = 0 \pm 7.2418i$ y $p_{3,4} = 0 \pm 4.1792i$ y los puntos de equilibrio son críticamente estables (puntos de equilibrio tipo centro [35]).

Ya que el sistema subactuado será regulado alrededor de los puntos de equilibrio $x^* = [90^\circ m \ 0 \ 0 \ 0]$, $m = 1$ la aproximación lineal (2.36) correspondiente es de especial interés. La función de transferencia de este sistema es

$$\frac{\delta Y(s)}{\delta U(s)} = \frac{-42.5387s^2 + 577.5349}{s^4 - 69.9096s^2 + 915.9569}\tag{2.37}$$

Cuyos polos están ubicados en $p_{1,2} = \pm 7.2418$, $p_{3,4} = \pm 4.1792$ y ceros en $c_{1,2} = \pm 3.6839$. Es decir, el sistema linealizado es inestable y de fase no mínima.

CAPÍTULO 3

CONTROL DEL SISTEMA PENDUBOT

3.0 Introducción.

Existen sistemas que son fáciles de controlar debido a que poseen un número adecuado de actuadores, tales como robots, automóviles, máquinas de producción, sin embargo existen algunos sistemas que son difíciles de controlar debido al número limitado de actuadores, es decir tienen menor cantidad de actuadores que grados de libertad y surgen de varias formas, por diseño, por naturaleza propia o debido a la aplicación; como son aviones, helicópteros o satélites. A estos últimos se les conoce como sistemas subactuados.

El pendubot, tal y como ya se explicó en el capítulo 2, es un sistema electromecánico subactuado que consiste de dos eslabones rígidos interconectados por uniones de revolución. La primera unión está actuada por un motor de CD y la segunda se encuentra sin actuador (figura 3.1).

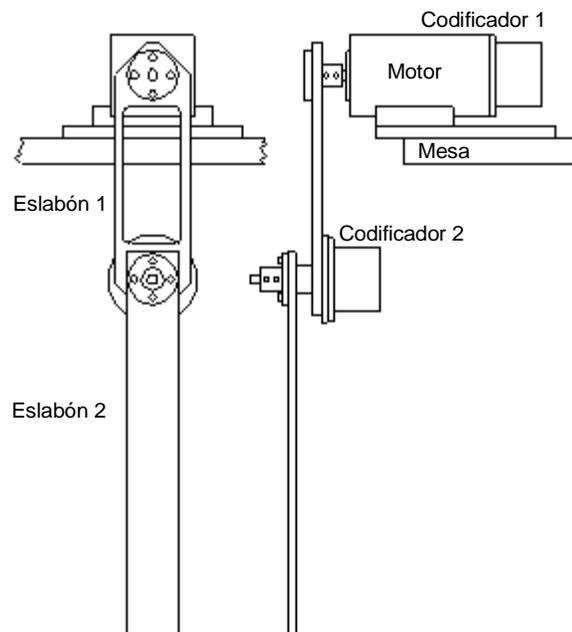


Figura 3.1 Pendubot

De esta forma, el segundo eslabón se puede ver como un péndulo simple cuyo movimiento puede ser controlado por la actuación del primero. Sin embargo, debido a la dinámica no lineal de acoplamiento entre los dos eslabones, el pendubot posee algunas características únicas para la investigación en control, que no se encuentran en otros dispositivos [15]. Es decir, permite ilustrar la aplicación y desempeño de diferentes técnicas de control y representa un banco de pruebas adecuado para probar la validez y desempeño de leyes de control nuevas, además permite también ilustrar un gran número de conceptos fundamentales en dinámica no lineal.

En este sistema los objetivos básicos de control son: **El Balanceo**, el cual consiste en llevar los eslabones desde su posición de equilibrio estable $(-90^\circ, 0^\circ)$, (ver capítulo 2 para una descripción de las posiciones de equilibrio) a la posición inestable $(90^\circ, 0^\circ)$, a través de los movimientos del primer eslabón y

La Estabilización, que consiste en volver estable la posición de equilibrio inestable $(90^\circ, 0^\circ)$.

En este capítulo se describen las técnicas de control que se utilizan en este trabajo para resolver los problemas del balanceo y la estabilización y que más tarde serán implementadas en tiempo real. El capítulo se divide en tres secciones. En la sección 3.1, se muestran algunas simulaciones del modelo matemático no lineal del pendubot con el fin de que se pueda tener una idea de la forma en que el sistema evoluciona. En la sección 3.2, se desarrolla la aplicación y simulación de las técnicas de control consideradas para resolver el balanceo. Finalmente, en la sección 3.3 se describe la técnica de control lineal empleada para resolver el problema de la estabilización.

3.1 Simulaciones en lazo abierto del sistema.

A continuación se muestran las gráficas de la respuesta en el tiempo del modelo del pendubot ante diferentes condiciones iniciales. Es necesario resaltar que en las simulaciones del sistema alrededor del punto de equilibrio estable (posición de reposo), dado que no se está considerando la fricción, el sistema muestra oscilaciones fijas no atenuadas.

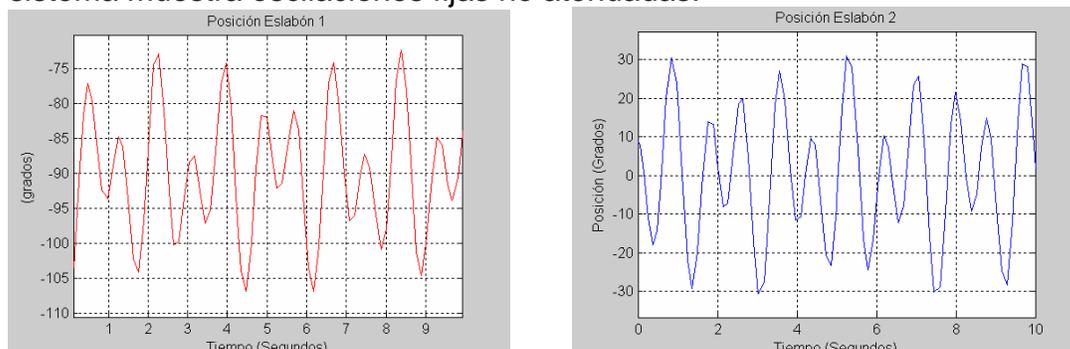


Figura 3.2 Condiciones iniciales $x = [-60.3^\circ \quad 9^\circ \quad 0 \quad 0]$
(cercano al punto inferior)

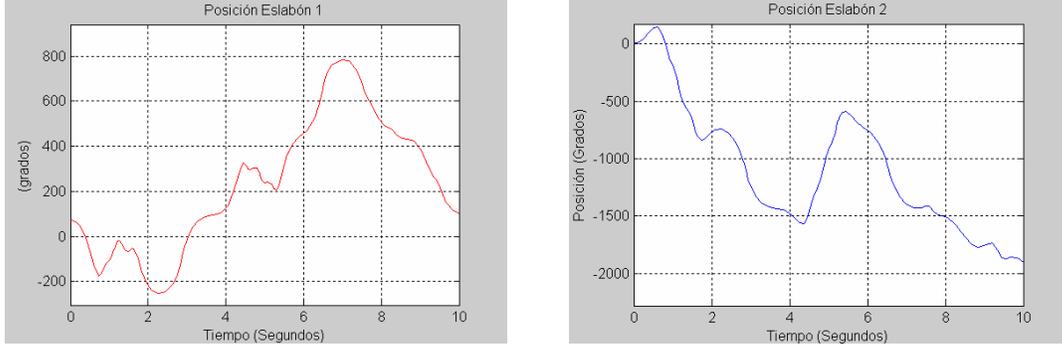


Figura 3.3 Condiciones iniciales $x = [69.305^\circ \ 9^\circ \ 0 \ 0]$
(cercano al punto superior)

En este segundo par de gráficas (figura 3.3) se puede observar que el sistema gira sin control en ambas direcciones una y otra vez.

3.2 Leyes de control para el balanceo.

3.2.1 Esquema de prealimentación y PD.

Primeramente se usa el método de linealización parcial por retroalimentación de estados, así se tomará en cuenta el efecto no lineal del segundo eslabón sobre el primero. Después se usa un compensador PD sobre el sistema parcialmente linealizado, para llevar al primer eslabón de su posición en -90° a su posición en $+90^\circ$, obviamente el segundo eslabón se moverá libremente, sin ningún control por el momento. Después, se emplea un esquema de prealimentación para asegurar que el segundo eslabón oscile de forma controlada (indirectamente, a través del primer eslabón), hasta acercarse al punto de equilibrio superior y eventualmente conmutar a una ley de control estabilizante.

Diseño.

Considérense las ecuaciones de movimiento del pendubot dadas por (2.18) y (2.19), que se repiten por claridad,

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + \phi_1 = \tau_1 \quad (3.1)$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + \phi_2 = 0 \quad (3.2)$$

donde los parámetros del sistema están dados por (2.20) y (2.21). El único cambio es $\phi(q) = g(q)$.

Dado que el segundo eslabón es no-actuado, esta parte de la dinámica no se puede linealizar. Sin embargo linealizar uno de los grados de libertad facilita el diseño de un lazo de control externo que seguirá una trayectoria dada para el grado de libertad linealizado.

La ecuación (3.2) se puede resolver para la aceleración angular del eslabón dos como sigue,

$$\ddot{q}_2 = \frac{-d_{12}\ddot{q}_1 - c_{21}\dot{q}_1 - \phi_2}{d_{22}} \quad (3.3)$$

sustituyendo (3.3) en (3.1) se obtiene

$$\bar{d}_{11}\ddot{q}_1 + \bar{c}_{11}\dot{q}_1 + \bar{c}_{12}\dot{q}_2 + \bar{\phi}_1 = \tau_1 \quad (3.4)$$

con

$$\begin{aligned} \bar{d}_{11} &= d_{11} - \frac{d_{12}d_{21}}{d_{22}} \\ \bar{c}_{11} &= c_{11} - \frac{d_{12}c_{21}}{d_{22}} \\ \bar{c}_{12} &= c_{12} \\ \bar{\phi}_1 &= \phi_1 - \frac{d_{12}\phi_2}{d_{22}} \end{aligned} \quad (3.5)$$

El lazo de control interno que linealiza la dinámica de q_1 se puede definir ahora como

$$\tau_1 = \bar{d}_{11}V_1 + \bar{c}_{11}\dot{q}_1 + \bar{c}_{12}\dot{q}_2 + \bar{\phi}_1 \quad (3.6)$$

Después de sustituir (3.6) en (3.4), se obtiene lo siguiente

$$\ddot{q}_1 = V_1 \quad (3.7)$$

$$d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + \phi_2 = -d_{21}V_1 \quad (3.8)$$

Ya que la ecuación (3.4) ha sido linealizada como se muestra en (3.7), el lazo de control externo se puede diseñar para seguir una trayectoria deseada para el eslabón uno, donde la respuesta del eslabón dos estará dada por la ecuación no lineal resultante dada en (3.8). Con base en la dinámica dada en (3.7) y la ecuación (3.8), se diseña el controlador PD con prealimentación de la aceleración, como sigue.

$$V_1 = \ddot{q}_1^d + k_d(\dot{q}_1^d - \dot{q}_1) + k_p(q_1^d - q_1) \quad (3.9)$$

Donde q_1^d , \dot{q}_1^d y \ddot{q}_1^d son la posición deseada, la velocidad deseada y la aceleración deseada, respectivamente. El ajuste de las ganancias se ha realizado a prueba y error, quedando en $k_d = 23$ y $k_p = 144$. En la figura 3.4, se muestra el diagrama resultante para este esquema de control.

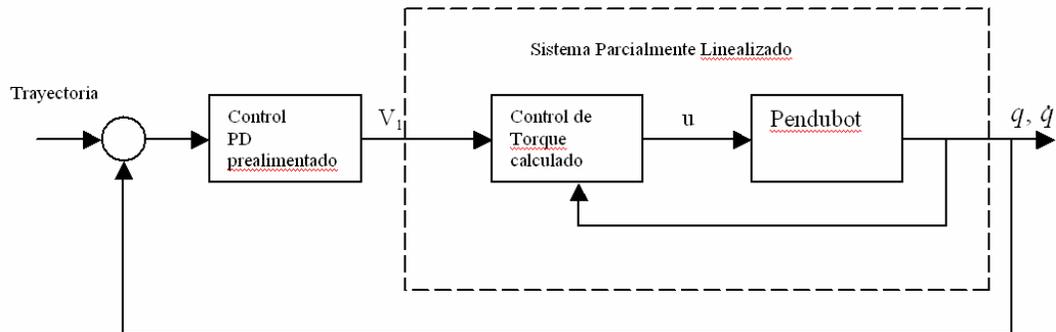


Figura 3.4. Diagrama a bloques del control por linealización parcial.

Simulaciones.

A continuación se muestran las gráficas de las respuestas para ambos eslabones partiendo desde la posición de equilibrio inferior y desde ahí el sistema evoluciona hasta oscilar alrededor del punto de equilibrio inestable.

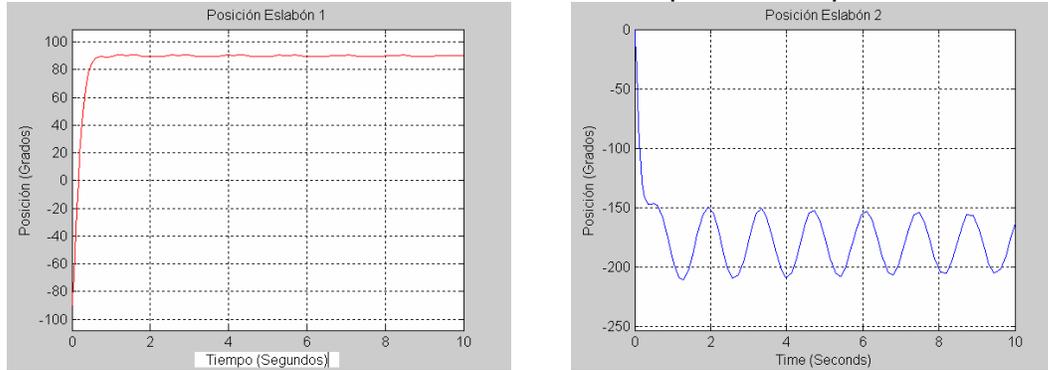


Figura 3.5. Simulaciones con la ley de par calculado.

Esto es lo que se logra con el control de par calculado (linealización parcial) y el PD (sin prealimentación). La referencia de posición es un escalón para el eslabón linealizado de valor, $x_1 = 90^\circ$. Como se puede observar el segundo eslabón oscila libremente siguiendo su propia dinámica. Las ganancias del control PD, se obtuvieron a prueba y error quedando finalmente como $k_p = 144, k_d = 23$.

En seguida se muestra la respuesta del pendubot al aplicarle la ley de control (3.6) y (3.9) que resuelve el balanceo y después de cierto tiempo, cuando el sistema está cerca del punto de equilibrio superior se conmuta a una ley de control lineal, la cual será obtenida en la próxima sección. Esta ley de control resuelve el problema de la estabilización. En ambas gráficas el sistema inició desde la posición inferior (la posición estable).

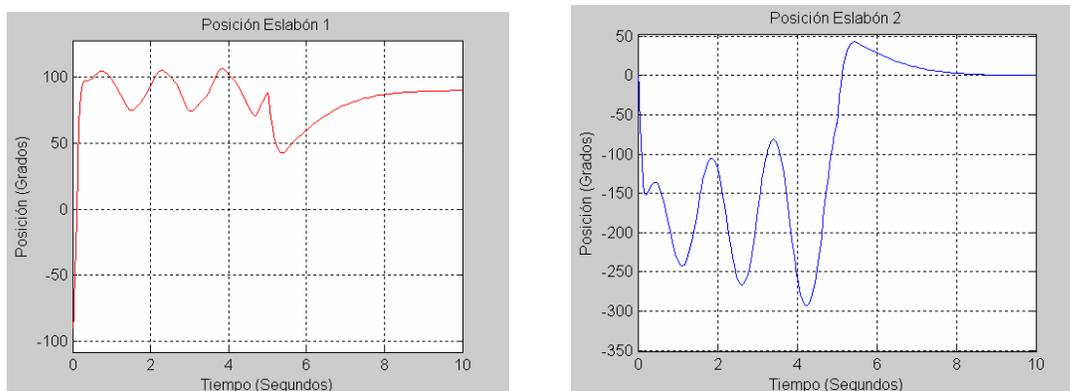


Figura 3.6. Primeros 10 segundos de la respuesta del sistema.

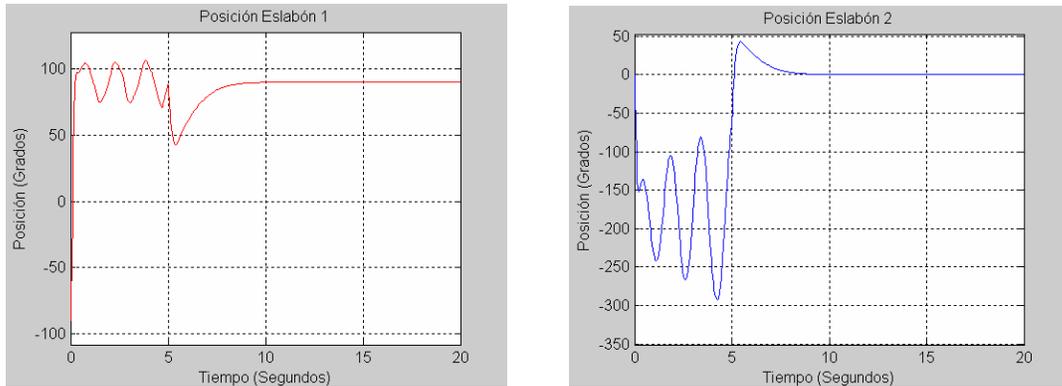


Figura 3.7. Primeros 20 segundos de la respuesta del sistema.

3.2.2 Control basado en energía.

En este caso el problema de control es el mismo que en el punto anterior. Se pretende llevar al sistema desde el punto de equilibrio estable a una vecindad del punto de equilibrio inestable, es decir se trata del problema de Balanceo. En este caso se emplea una técnica basada en el control de la energía y el concepto de pasividad.

Diseño.

Para este procedimiento se utilizará el modelo no lineal del pendubot, según se indica en (2.18)-(2.21). En [22] se demuestra que el sistema es pasivo teniendo τ como entrada y \dot{q}_2 como salida. Para $\tau = 0$, (2.20) tiene cuatro puntos de equilibrio $(q_1, \dot{q}_1, q_2, \dot{q}_2)$. Los puntos de equilibrio $(\frac{\pi}{2}, 0, 0, 0)$ y $(-\frac{\pi}{2}, 0, \pi, 0)$ son inestables. El objetivo es alcanzar la posición de superior, es decir $(\frac{\pi}{2}, 0, 0, 0)$. Los puntos $(\frac{\pi}{2}, 0, \pi, 0)$ y $(-\frac{\pi}{2}, 0, 0, 0)$ son posiciones de equilibrio (inestable y estable) que se desean evitar, lo cual se logra tomando en cuenta la energía total $E(q, \dot{q})$ que es diferente para cada uno de los cuatro puntos de equilibrio. Este hecho es el que se utiliza para obtener la ley de control. En [22], se indican los valores correspondientes de energía para cada posición. La energía para la posición que se desea alcanzar es $(\theta_4 + \theta_5)g$.

La energía total del pendubot es,

$$E = \frac{1}{2} \dot{q}^T D(q) \dot{q} + P(q) = \frac{1}{2} \dot{q}^T D(q) \dot{q} + \theta_4 g \sin q_1 + \theta_5 g \sin(q_1 + q_2) \quad (3.10)$$

Por lo tanto, considerando lo anterior, se plantean las siguientes condiciones.
Si

$$\begin{aligned} c_1) \quad \dot{q}_1 &= 0; \\ c_2) \quad E(q, \dot{q}) &= (\theta_4 + \theta_5)g; \\ c_3) \quad q_1 &= \frac{\pi}{2} \end{aligned} \quad (3.11)$$

entonces de (3.10), (2.18), (2.19), (2.20) y (2.21), se obtiene

$$\frac{1}{2} \theta_2 \dot{q}_2^2 = \theta_3 g (1 - \cos q_2) \quad (3.12)$$

Lo cual define una trayectoria que corresponde a una órbita homoclínica. Esto significa que la posición angular del eslabón 2 se mueve de uno a otro lado hasta alcanzar el punto de equilibrio $(q_2, \dot{q}_2) = (0, 0)$. Entonces el objetivo (resolver el problema del balanceo) se puede lograr si el sistema se lleva a la órbita (3.12) para $\dot{q}_1 = 0$ y $q_1 = \pi/2$.

Ahora siguiendo el procedimiento marcado en [22], se obtiene la siguiente ley de control,

$$\tau_1 = \frac{-k_D F(q, \dot{q}) - (\theta_1 \theta_2 - \theta_3^2 \cos^2 q_2)(\dot{q}_1 + k_P \tilde{q}_1)}{(\theta_1 \theta_2 - \theta_3^2 \cos^2 q_2) k_E \tilde{E} + k_D \theta_2} \quad (3.13)$$

donde $\tilde{q}_1 = (q_1 - \frac{\pi}{2})$ y $\tilde{E} = (E - E_{top})$.

Simulaciones.

El valor de los parámetros del sistema se toman del apéndice B. Ahora se muestran los resultados para $k_E = 1.5$, $k_P = 1$, $k_D = 1$ y una posición inicial de $(q_1, \dot{q}_1, q_2, \dot{q}_2) = (0, 0, 0, 0)$.

En la figura 3.8, se muestran los primeros 30 segundos de las posiciones angulares de q_1 y q_2 . La condición inicial fue la posición estable; es decir, la posición de reposo $(0, 0, 0, 0)$. En esta gráfica se puede ver cómo el primer eslabón se mueve hasta alcanzar la posición vertical, $\pi/2$ y el segundo eslabón comienza a oscilar de acuerdo a lo comentado en la teoría.

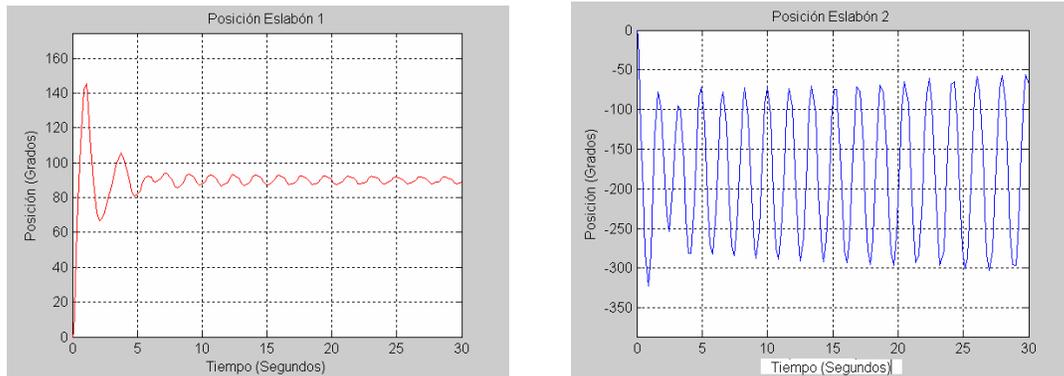


Figura 3.8. Posiciones q_1 y q_2 .

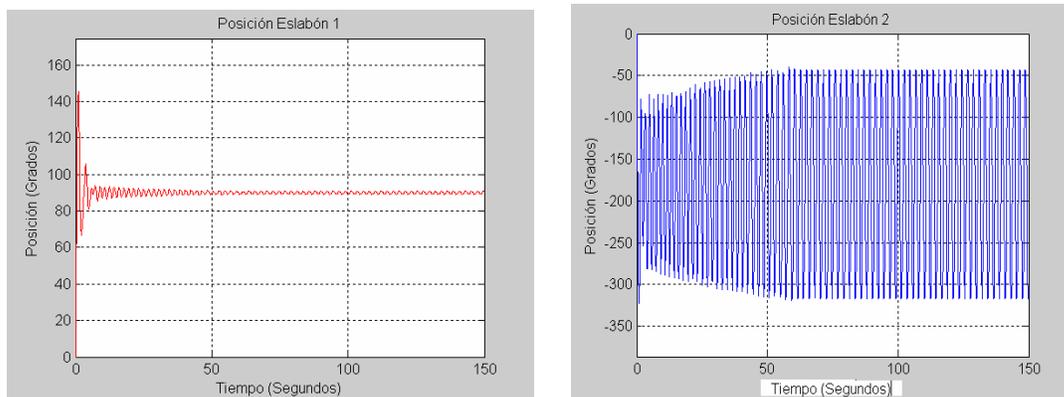


Figura 3.9. 150 segundos de simulación, q_1 y q_2 .

La figura 3.9, muestra la misma simulación pero hasta 150 segundos. Se puede ver como el segundo eslabón crece en su oscilación tendiendo a alcanzar los valores de 0 y 2π .

Como se puede observar el balanceo del modelo del pendubot es resuelto en forma satisfactoria en ambos casos. La diferencia principal es que para el Control Basado en Energía de la sección 3.2.2, el valor máximo que se alcanza en el balanceo del segundo eslabón es de aproximadamente -45° y se alcanza después de 50 segundos, mientras que para el esquema de control de la sección 3.2.1, el valor máximo que se alcanza es de casi 0° y se alcanza en menos de 10 segundos al costo de tener un mayor gasto de energía.

Durante los experimentos de simulación del Control Basado en Energía se observó que la ley de control obtenida no es robusta ante variaciones en los parámetros, de tal forma que al momento de implementarla en tiempo real se tendría que realizar un ajuste fino en los parámetros, o tal vez identificar los parámetros reales, para lograr un funcionamiento aceptable.

3.3 Ley de control para la estabilización (Control Lineal).

Para diseñar el control estabilizante se utilizó el modelo lineal que se obtuvo en las ecuaciones (2.36) y la asignación de variables de estado,

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

El punto de equilibrio para la linealización es el punto superior, es decir, $x^* = [90^\circ \ z \ 0 \ 0 \ 0]$ con $z = -1$.

Una vez obtenido el modelo lineal, se pueden usar técnicas como “ubicación de polos” y “LQR (Regulador Cuadrático Lineal)” para diseñar el control por retroalimentación de estados, $u = -Kx$, que estabilice al sistema. En este caso se usó la instrucción de MatLab “**place**” para obtener el vector de ganancias del controlador. El vector obtenido es,

$$K = [-16.4615 \quad -3.1287 \quad -16.2422 \quad -2.0658] \quad (3.14)$$

En la figura 3.10, se muestra la respuesta del sistema cuando la condición inicial es muy cercana al equilibrio superior $x = [69.305^\circ \ 9^\circ \ 0 \ 0]$, cuyo valor es muy próximo al máximo que se puede tener para este controlador. Si la posición inicial aumenta aproximadamente un 10% del valor actual, el sistema se inestabiliza.

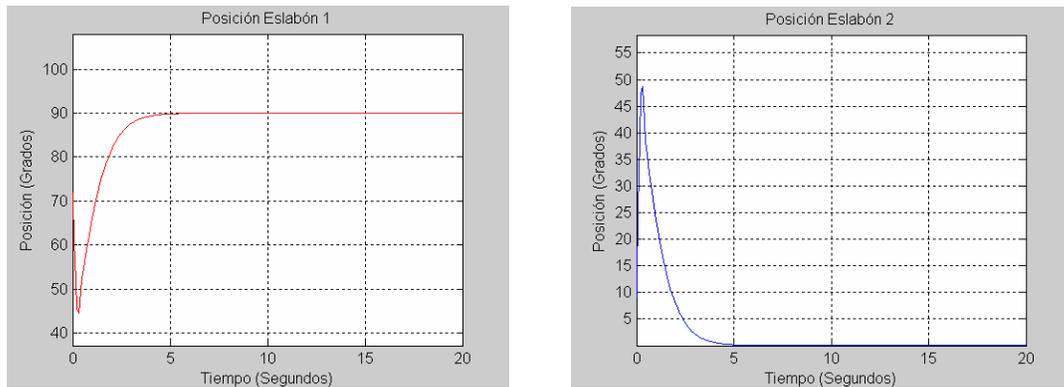


Figura 3.10. Condición inicial en $x = [69.305^\circ \ 9^\circ \ 0 \ 0]$
(cercano al punto superior)

CAPÍTULO 4

SIMULACION EN TIEMPO REAL DEL SISTEMA PENDUBOT

4.0 Introducción.

En este capítulo se muestran los procedimientos y resultados obtenidos en relación al objetivo de este trabajo; es decir, la implementación de la simulación en tiempo real del control estabilizante del pendubot. En la sección 4.1, se da un resumen de conceptos de sistemas en tiempo real, con el fin de enmarcar los resultados obtenidos. En la sección 4.2, se presenta la obtención del modelo en tiempo discreto no lineal, dado que con esto se trabajaría en la implementación. En la sección 4.3, se desarrollan los métodos para obtener los tiempos de muestreo permitidos para el pendubot. En la sección 4.4, se obtienen los tiempos de ejecución del código del pendubot. En la sección 4.5, se comparan los tiempos obtenidos y finalmente, en la sección 4.6 se muestran los resultados obtenidos de la implementación en tiempo real del sistema pendubot y las leyes de control consideradas.

4.1 Conceptos de sistemas en tiempo real.

Un sistema en tiempo real (STR) es aquel en el que la corrección de la computación no solo depende de la corrección de la lógica de la computación, sino también del tiempo en el cuál el resultado es producido. Si los límites temporales del sistema no son alcanzados, se puede decir que una falla del sistema ha ocurrido. [Sistemas en tiempo real]

Es decir en un sistema de tiempo no solamente es importante que las tareas se realicen, sino que también deben cumplirse en los plazos de tiempo previstos. Normalmente un STR interactúa con su entorno (objeto o medio que controla), recibe estímulos de este y actúa en consecuencia para producir cambios en el entorno. En estos casos se los denomina sistemas reactivos, ya que reaccionan ante estímulos del medio ambiente.

Si bien el tiempo es el factor determinante para definir el correcto funcionamiento de un STR, erróneamente se tiende a afirmar que cuanto más rápido se realiza una tarea, mejor funciona el STR. Para evaluar el correcto funcionamiento de un STR hay que tener en cuenta los límites según una línea de tiempo, que pueden estar expresados en segundos o milisegundos según corresponda.

Los sistemas en tiempo real se pueden clasificar de diferentes formas atendiendo a las características que de ellos se consideren. Aunque no todos los sistemas de tiempo real poseerán todas las características.

1) En base a la importancia que tienen los límites temporales de un STR pueden llegar a clasificarse en tres subgrupos:

- a) **Sistemas de tiempo real estricto (*hard real-time*)**. Son aquellos en los cuales no cumplir con alguno de los requerimientos temporales genera una falla general del sistema. Entendiendo como general a una falla completa sin posibilidad de continuar con el funcionamiento. Por lo tanto los límites de tiempo son estrictos y hay veces que es preferible un trabajo imperfecto pero terminado a tiempo. Ejemplo: Control de un reactor nuclear.
- b) **Sistemas de tiempo real flexible (*soft real-time*)**. A diferencia de los anteriores, se pueden llegar a no alcanzar algunos de los límites especificados pero igualmente el STR continúa su ejecución y la única consecuencia se refleja en una degradación de la calidad de las respuestas. Para estos STR se pueden definir niveles de tolerancia en el momento del diseño de los mismos. Por lo tanto son sistemas en los que los límites de tiempo son flexibles. Ejemplo: Sistemas de reserva de pasajes.
- c) **Sistemas de tiempo real firme (*firm real-time*)**. Realmente esto no es un subsistema en sí, sino una variación de los sistemas de tiempo real duros, ya que estos son sistemas de tiempo real duro que pueden tolerar pérdidas, si la probabilidad de ocurrencia de la misma es baja. Ejemplo: Pérdida de una trama de audio o video.

2) Clasificación según las escalas de tiempo:

- a) **Basados en reloj**. Se invoca y ejecuta repetidamente a intervalos constantes a partir de una invocación inicial. Por ejemplo: Tareas periódicas.

- b) **Basados en eventos.** Las acciones se realizan a partir de un evento ocurrido en el proceso que controla. Por ejemplo: Las acciones comienzan a partir de que se lee la señal de un sensor.
 - c) **Interactivos.** Se invoca y ejecuta a intervalos irregulares; es decir, sus invocaciones ocurren en instantes arbitrarios. Por ejemplo: Un operario metiendo datos.
- 3) Según la integración con el sistema físico (o con el medio):
- a) **Embebidos (o empotrados).** Es un sistema de tiempo real que forma parte de un sistema más complejo. Por ejemplo: el sistema de control de inyección de combustible de un automóvil.
 - b) **No embebidos.** Orgánicos, los que son independientes del *hardware* en que corren. Débilmente acoplados, los que pueden correr en otro *hardware* reescribiendo ciertos módulos.
- 4) Según la forma de procesamiento:
- a) **Sistemas centralizados.** Son aquellos sistemas en los que un único nodo (mono o multiprocesador) es el encargado de atender a todas las tareas, estas se comunican a través de memoria compartida (el gasto de tiempo en comunicación es insignificante).
 - b) **Sistemas distribuidos.** Son aquellos sistemas en los que hay varios nodos, unidos a través de una red entre los cuales se reparten la atención de los distintos procesos. Las tareas se comunican a través de la red, no hay memoria compartida (el gasto de tiempo en comunicación es importante).
- 5) Según la estrategia de planificación de las tareas:
- a) **Sistemas estáticos.** En estos sistemas todas las tareas, su naturaleza y características son conocidas de antemano y al momento del diseño se planifica la ejecución de las mismas. El sistema no admite la aparición de una nueva tarea sobre la marcha y tiene bajo costo de ejecución.
 - b) **Sistemas dinámicos (o adaptivos).** En estos sistemas puede haber un conjunto de tareas conocido de antemano, pero ante la aparición de una nueva tarea, el sistema analiza si la puede garantizar sin afectar a las tareas que ya maneja y en ese caso la agrega a la lista de tareas. Aquí hay un mayor costo de ejecución, porque el planificador tiene un trabajo de análisis adicional.

4.2 Modelo en tiempo discreto del sistema pendubot.

Las ecuaciones en tiempo discreto para el pendubot se obtienen partiendo de las ecuaciones en variables de estado en su forma general (2.23),

$$\begin{aligned} \dot{x}_1(t) &= f_1(x) \\ \dot{x}_2(t) &= f_2(x) \\ \dot{x}_3(t) &= f_3(x) + g_3(x)u(t) \\ \dot{x}_4(t) &= f_4(x) + g_4(x)u(t) \end{aligned} \quad (4.1)$$

Aproximando la derivada mediante una diferencia finita

$$\begin{aligned} \lim_{\Delta t \rightarrow k} \frac{x(t + \Delta t) - x(t)}{\Delta t} &\approx \dot{x}(t) \\ \frac{x(t + k) - x(t)}{k} &\approx \dot{x}(t) \end{aligned} \quad (4.2)$$

Y sustituyendo en las ecuaciones de estado se tiene que,

$$\begin{aligned} x_1(k + 1) &= f_1(x(k))\Delta t + x_1(k) \\ x_2(k + 1) &= f_2(x(k))\Delta t + x_2(k) \\ x_3(k + 1) &= [f_3(x(k)) + g_3(x(k))u(k)]\Delta t + x_3(k) \\ x_4(k + 1) &= [f_4(x(k)) + g_4(x(k))u(k)]\Delta t + x_4(k) \end{aligned} \quad (4.3)$$

Estas últimas son las ecuaciones en variables de estado en su forma de diferencias finitas.

Para simular estas ecuaciones en tiempo discreto (es decir, para obtener la respuesta del sistema en tiempo discreto) se desarrolló en MatLab el siguiente programa, "pendubotdiscreto.m".

Las gráficas que se obtienen de la ejecución de este programa se muestra en las figuras 4.1, 4.2, 4.3 y 4.4.

Recuadro 4.1. Programa “pendubotdiscreto.m”, para obtener, la respuesta simulada en tiempo discreto del pendubot.

```

clear;

l1=0.26987;
lc1=0.13494;
l2=0.38417;
lc2=0.19208;
lz1=1.3863e-2;
lz2=1.6749e-2;
m1=0.5289;
m2=0.3346;
g=9.81;

u=0;

D=0.5/1000;           %Este es el periodo de muestreo en milisegundos
N=10*(1/D);           % Se calculan la cantidad de muestras para
                      % una simulación de 10 segundos
N=round(N);           % Se redondea al entero mas próximo

x1(1)=-pi/2;          %Posiciones angulares en radianes
x2(1)=0.9;
x3(1)=0.0;
x4(1)=0.0;

A=m1*lc1*lc1+m2*(l1*l1+lc2*lc2)+lz1+lz2;
B=m2*l1*lc2;
C=m2*lc2*lc2+lz2;
E=m1*g*lc1+m2*g*l1;
F=m2*g*lc2;
G=m2*l1*lc2;

for i=1:N,             %N es la cantidad de muestras para cubrir 10 seg
                      % con el periodo de muestreo dado
    D11=A+2*B*cos(x2(i));
    D12=C+B*cos(x2(i));
    D22=C;

    G1=E*cos(x1(i))+F*cos(x1(i)+x2(i));
    G2=F*cos(x1(i)+x2(i));

    C1=-G*(2*x3(i)+x4(i))*x4(i)*sin(x2(i));
    C2=G*x3(i)*x3(i)*sin(x2(i));

    aux1=D11*D22-D12*D12;

    f1=x3(i);
    f2=x4(i);
    f3=(D22*((D12*(C2/D22))+(D12*(G2/D22))-C1-G1))/aux1;
    f4=(-D12*((D12*(C2/D22))+(D12*(G2/D22))-C1-G1))/aux1-(C2/D22)-(G2/D22);

    g3=D22/aux1;
    g4=-D12/aux1;

    x1(i+1)= f1*D + x1(i);           %Se divide entre 1000 para tenerlo en milisegundos*/
    x2(i+1)= f2*D + x2(i);
    x3(i+1)= (f3+g3*u)*D + x3(i);
    x4(i+1)= (f4+g4*u)*D + x4(i);

end

subplot(2,1,2), plot(x1);

```

4.3 Desarrollo del análisis para obtener el tiempo de muestreo del sistema pendubot.

4.3.1 Mediante el segundo momento de probabilidad.

Como se puede ver en la ecuación (4.3), la correcta implementación en tiempo real del sistema, depende del valor que se elija para el tiempo de muestreo (TM). Por tal razón es necesario identificar el rango de TM admitidos.

En primer lugar se intentó identificar la frecuencia de muestreo adecuada para el sistema pendubot mediante la comparación de la respuesta del sistema discreto (4.3) para diferentes TM, contra la respuesta del sistema continuo. Para tal efecto se desarrollaron los programas “pendsys.m” (recuadro 4.3) y “pendsys2.m” (4.2), con los cuales se obtiene la simulación del sistema en tiempo continuo.

Las simulaciones en tiempo continuo del pendubot se desarrollan usando la función ode45 de MatLab. La condición inicial en todos los casos es $[-\pi/2 \ 0.9 \ 0 \ 0]$. El valor de las constantes está definido dentro de los programas, ver recuadros de programas.

Recuadro 4.2. Programa “pendsys2.m”, segunda parte para la simulación en tiempo continuo del pendubot.

```
% Programa pendsys2.m, en el cual se implementa la función de MATLAB
% ode45, con la cual se integra la ecuación diferencial de los estados
% del sistema pendubot implementada en el programa "pendsys.m"
% Este programa llama a ejecución a pendsys.m a través de ode45.

% Creado por Adolfo Rojas Pacheco.
% agosto de 2006.

clear;
opciones = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-4 1e-4]);

u = 0; % Señal de control del sistema, por ahora no se usa
%par = [0.3 0.5]; % Estos son los parámetros que se le envían desde aquí.
intervalo = [0 10]; % Intervalo del tiempo de integración.
cond_inic = [-pi/2 0.9 0 0]; % Condiciones iniciales.
[t,x] = ode45(@pendsys, intervalo, cond_inic, opciones,u);
%El nombre de la función debe ser el mismo
%que el del archivo y q el de la
%función

%plot(t,x(:,1),'-.',t,x(:,2),'-');
subplot(2,1,1), plot(t,x(:,1));
```

Recuadro 4.3. Programa "pendsys.m", primera parte en la implementación de la simulación en tiempo continuo del pendubot.

```

% Se implementa la función "pendsys" en donde se calculan las derivadas del
% sistema pendubot que la función de Matlab "ode45" integrará.
% Este programa es parte de la simulación en tiempo continuo del pendubot.
% Creado por Adolfo Rojas Pacheco.
% agosto de 2006.

% La función "ode45" tiene el siguiente formato:
% [T,Y] = ode45(odefun,tspan,y0)

% t, es el tiempo de simulación definido en ode45
% x, son los estados
% u, es la entrada al sistema

function dx = pendsys(t,x,u)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I1=0.26987;
Ic1=0.13494;
I2=0.38417;
Ic2=0.19208;
Iz1=1.3863e-2;
Iz2=1.6749e-2;
m1=0.5289;
m2=0.3346;
g=9.81;

%u=0;

A=m1*Ic1*Ic1+m2*(I1*I1+Ic2*Ic2)+Iz1+Iz2;
B=m2*I1*Ic2;
C=m2*Ic2*Ic2+Iz2;
E=m1*g*Ic1+m2*g*I1;
F=m2*g*Ic2;
G=m2*I1*Ic2;

D11=A+2*B*cos(x(2));
D12=C+B*cos(x(2));
D22=C;

G1=E*cos(x(1))+F*cos(x(1)+x(2));
G2=F*cos(x(1)+x(2));

C1=-G*(2*x(3)+x(4))*x(4)*sin(x(2));
C2=G*x(3)*x(3)*sin(x(2));

aux1=D11*D22-D12*D12;

f1=x(3);
f2=x(4);
f3=(D22*((D12*(C2/D22))+(D12*(G2/D22))-C1-G1))/aux1;
f4=(-D12*((D12*(C2/D22))+(D12*(G2/D22))-C1-G1))/aux1-(C2/D22)-(G2/D22);

g3=D22/aux1;
g4=-D12/aux1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dx = zeros(4,1); %se crea un vector columna
dx(1) = f1;
dx(2) = f2;
dx(3) = f3+g3*u;
dx(4) = f4+g4*u;

```

Para obtener las gráficas de las respuestas que a continuación se muestran, los programas se ejecutan con la siguiente secuencia, primero “pendsys2”, para obtener la respuesta en tiempo continuo; luego, “pendubotdiscreto”, para obtener la respuesta en tiempo discreto con el tiempo de muestreo de interés, y se obtiene una figura con las dos gráficas para su comparación. En todos los casos se utilizó la misma condición inicial $[-\pi/2 \ 0.9 \ 0 \ 0]$.

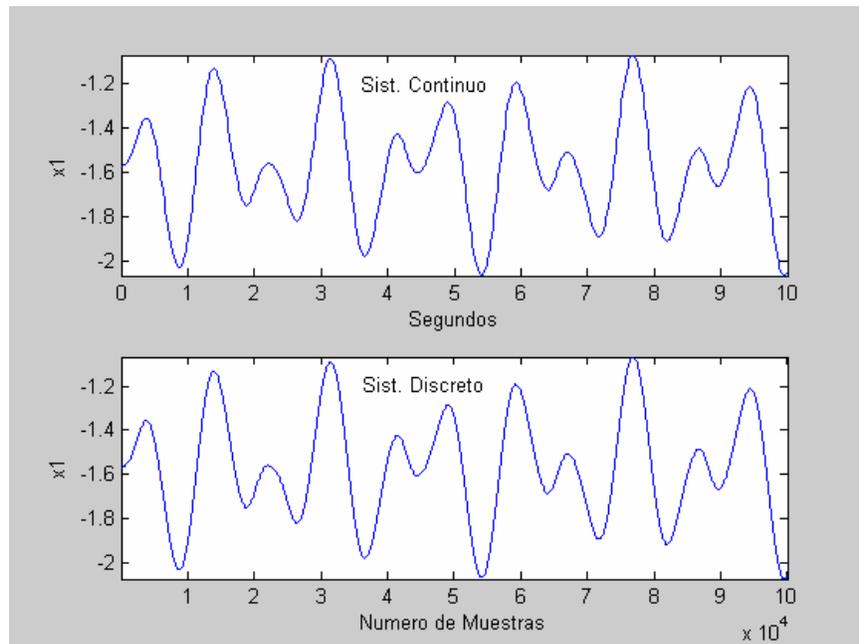


Figura 4.1 Tiempo de muestreo de 10 ms

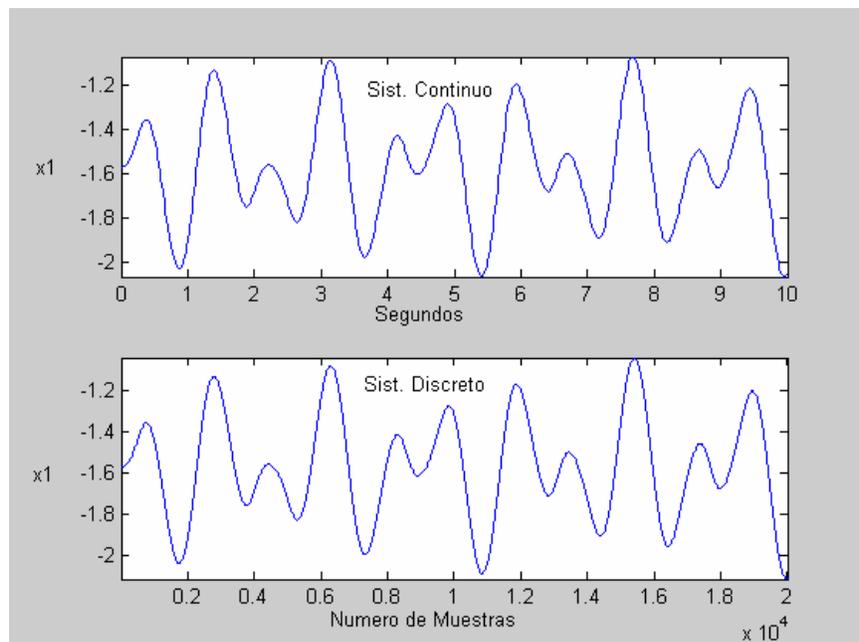


Figura 4.2 Tiempo de muestreo de 50 ms

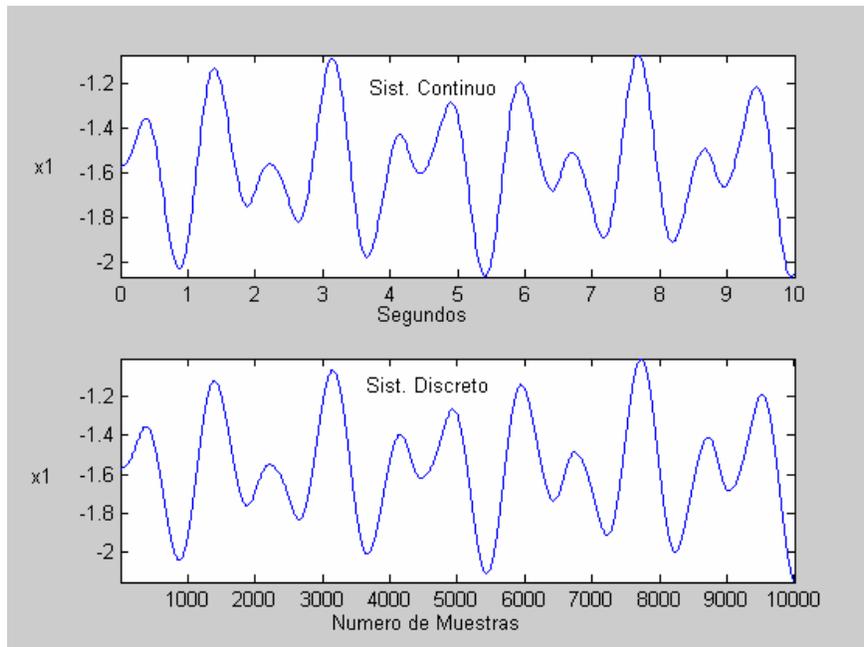


Figura 4.3 Tiempo de muestreo de 100 ms

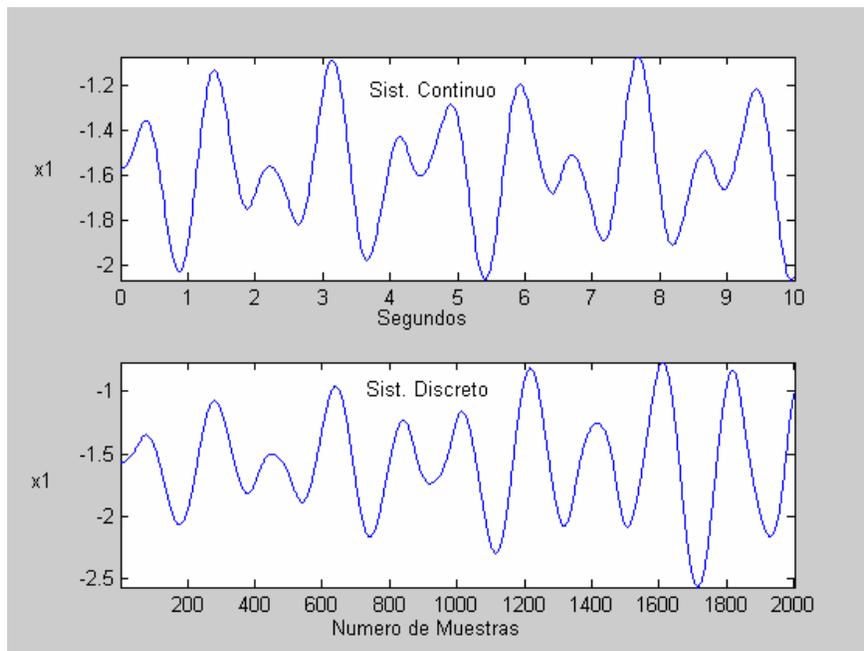


Figura 4.4 Tiempo de muestreo de 500 ms

Realizando una inspección detallada de las gráficas anteriores se pueden ver ciertas diferencias entre la respuesta del sistema continuo (referencia) y las respuestas del sistema en tiempo discreto para diferentes tiempos de muestreo; sin embargo, esta forma de comparación es muy imprecisa, por lo cual se decidió utilizar una comparación que dependiera menos de la observación. Para tal efecto se desarrollaron los cálculos de la desviación (segundo momento de probabilidad) de las respuestas del sistema discreto con respecto a la respuesta del sistema continuo. El programa implementado se muestra a continuación.

Recuadro 4.4 Programa para el cálculo de los momentos de probabilidad del sistema.

```

clear all;
close all;
pendsys2;      % Se ejecuta el sistema continuo para obtener la
% referencia.

x1sc=x(:,1);   % De la respuesta continua, se separa únicamente la columna de x1, del
% resto de los estados
tm=[0.5 0.25 0.1 0.05 0.025]; %Este es el tiempo de muestreo en miliseg que
% irá tomando diferentes valores, para los cuales
% se calcularán las desviaciones respecto de la
% referencia, es decir la respuesta del sistema continuo.
for j=1:5      % Ciclo mediante el cual se ejecuta el sistema discreto para los
% diferentes "tm".

pendubotdiscreto; % Se ejecuta el sistema discreto.

d(1,j)=.1; % Valor inicial para la desviación.

for s=2:length(x1sc-1)
% Se obtiene la desviacion de la reapesta del sistema discreto
% respecto del sistema en continuo, con base en el segundo
% momento de probabilidad.
d(s,j)=(1/s)*(d(s-1,j)*(s-1)+(x1sc(s)-x1r(s))^2);
end
end

% Se calcula la suma por columna y luego el minimo de tales sumas respecto
% de la desviacion
'la suma de las desviaciones son:'
sum(d)
'la menor suma es:'
min(sum(d))

'los tiempo de muestreo son:'
tm

figure;
plot(d);
grid;
title('Desviaciones respecto a cada tiempo de muestreo');
xlabel('Muestras');
ylabel('Desviacion');
'los minimos de las desviaciones son:'
[Y,I] = min(d)
min(Y)

```

Se obtuvieron varias curvas correspondientes a las diferentes desviaciones de la comparación entre las respuestas, ahora se debe elegir de todas ellas “las más adecuadas”. Para realizar esta discriminación entre las diferentes desviaciones que se obtuvieron, se calcularon además las sumas de los elementos de cada desviación, considerándose que la desviación con menor suma correspondería a una respuesta del sistema discreto más cercana a la respuesta del sistema continuo y a su vez al mejor tiempo de muestreo.

A continuación se muestran las gráficas obtenidas para diferentes tiempos de muestreo y para dos tiempos de simulación diferentes.

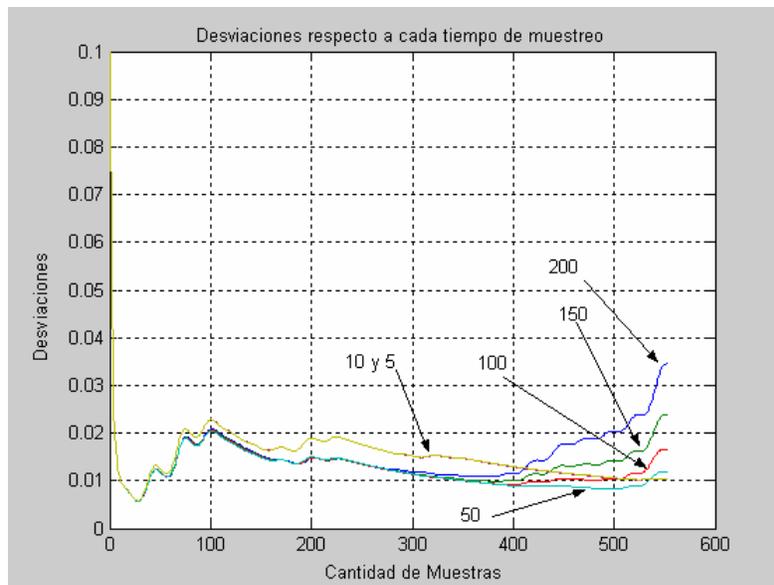


Figura 4.5 Tiempo de simulación de 10 segundos

En la figura 4.5, se muestran las desviaciones obtenidas para los tiempo de muestreo (TM en milisegundos) indicados en la gráfica. Las desviaciones acumuladas (suma de los datos de cada serie de desviación) resultan de la siguiente manera,

TM	200	150	100	50	10	5
Desviaciones	8.5578	7.6323	7.0654	6.7757	8.2646	8.2790

Si se observan las curvas y las sumas obtenidas, la menor suma y por lo tanto la menor desviación corresponde al tiempo de muestreo de 50 milisegundos. Sin embargo, debido a cierta tendencia de algunas desviaciones a incrementar su valor se decide extender el tiempo de simulación a 20 segundos y ver si esa tendencia tiene algún efecto en el resultado. Se obtiene la siguiente gráfica.

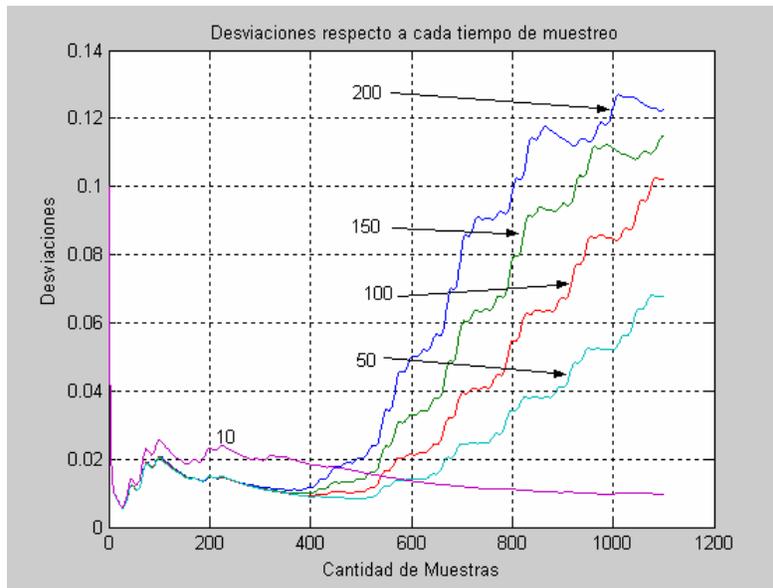


Figura 4.6 Tiempo de simulación de 20 segundos

Las desviaciones acumuladas son,

TM	200	150	100	50	10	
Desviaciones	60.8184	50.2777	38.6216	26.8313	16.6265	

Ahora se puede ver claramente la tendencia de las desviaciones a disminuir su valor para tiempos de muestreo alrededor de 10 milisegundos, lo cual sugiere la búsqueda de nuevos tiempos de muestreo, dentro de este rango.

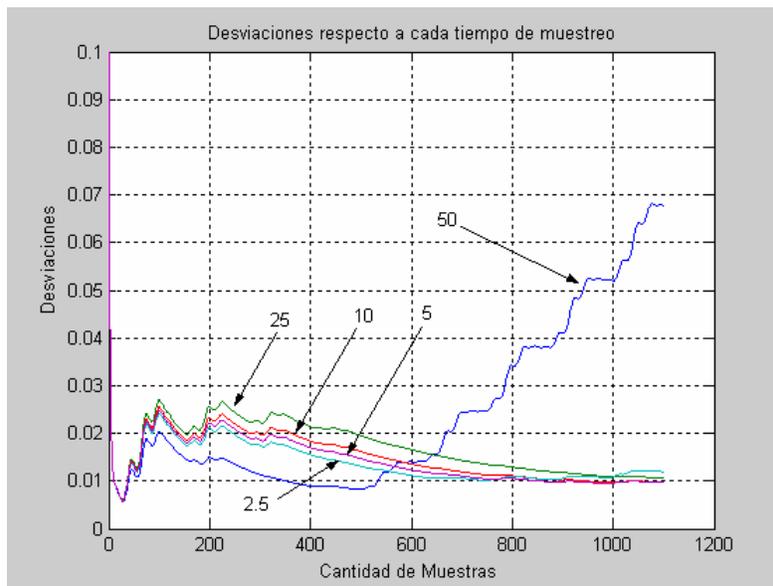


Figura 4.7 Tiempo de simulación de 20 segundos y nuevos tiempos de muestreo, alrededor de 10 milisegundos.

Las desviaciones acumuladas y los respectivos tiempos de muestreo fueron,

TM	50	25	10	5	2.5	
Desviaciones	26.8313	18.9145	16.6265	15.3338	15.7202	

En la figura 4.8 se muestra un acercamiento a la gráfica de la figura 4.7 para aclarar las imágenes. Se puede ver que la desviación que se mantienen más pequeña es la correspondiente al tiempo de muestreo de 5 milisegundos, mientras que la desviación del tiempo de muestreo más pequeño (2.5 milisegundos) empieza a crecer, esto mismo se ve reflejado en las sumas respectivas, 15.3338 y 15.7202.

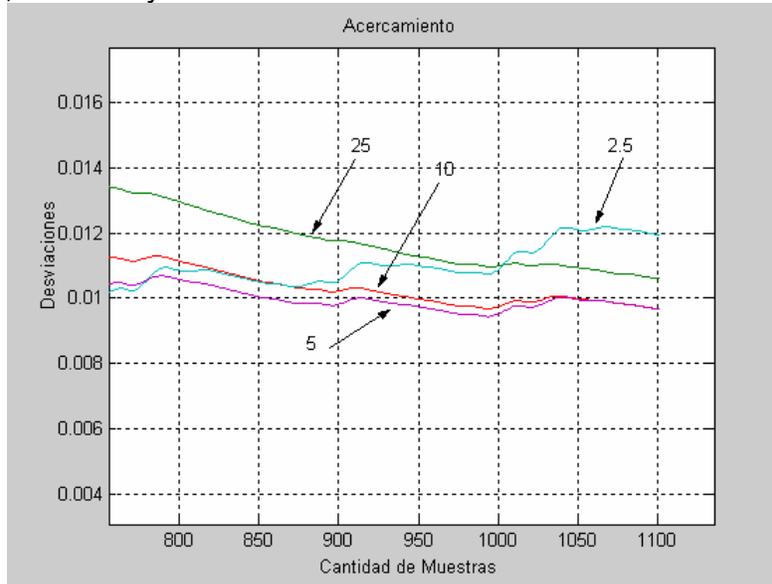


Figura 4.8 Acercamiento de la parte final de la gráfica 4.7

En la figura 4.9 se muestra la región de muestreos aceptables, la cual está formada por la combinación de las desviaciones acumuladas y los tiempo de muestreo respectivos obtenidos en la última simulación.

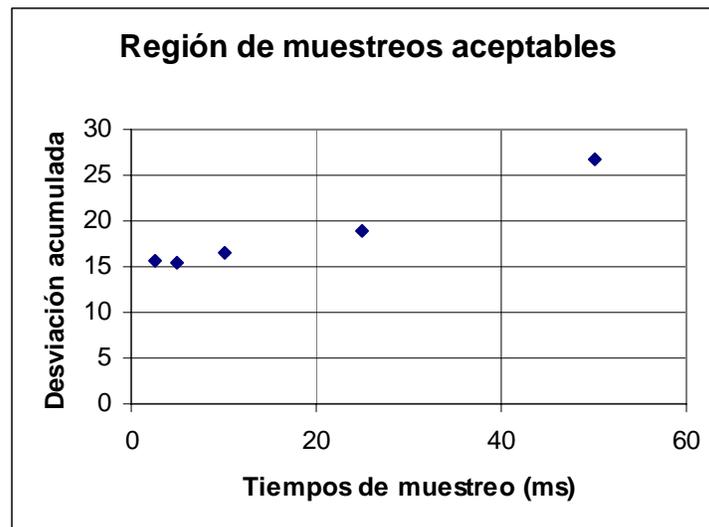


Figura 4.9.Región de muestreos aceptables

Ahora la selección del rango aceptable de los tiempos de muestreo se realiza con base en las siguientes expresiones

$$\tau_{mín}^* = \text{Pr oy}(\text{Inf}\{\sigma \times \tau_m\})$$

$$\tau_{máx}^* = \text{Pr oy}(\text{Sup}\{\sigma \times \tau_m\})$$

donde

$\tau_{mín}^*$ y $\tau_{máx}^*$, son los tiempo de muestreo mínimo y máximo permitidos, $\text{Pr oy}(\cdot)$, es la proyección.

$\text{Inf}\{\cdot\}$ y $\text{Sup}\{\cdot\}$, son el ínfimo y el supremo del conjunto entre llaves.

σ , es la desviación acumulada.

τ_m , es el tiempo de muestreo.

Por lo tanto, según la figura 4.9 los tiempos de muestreo aceptables para el sistema no lineal pendubot se ubican entre

$$\tau_{mín}^* = 5 \text{ ms}$$

$$\tau_{máx}^* = 25 \text{ ms}$$

4.3.2 Mediante diagramas de Bode.

Con el fin de reforzar el resultado anterior se muestran a continuación los diagramas de Bode (gráficas de respuesta en frecuencia) [29], [30] de la linealización del sistema pendubot para dos puntos de equilibrio, el superior ($x_1 = \frac{\pi}{2}$) y el inferior ($x_1 = -\frac{\pi}{2}$), correspondientes a las ecuaciones (2.36) y (2.37). De estos diagramas de bode se obtiene el ancho de banda, es decir, el contenido de frecuencias de la respuesta y por lo tanto la frecuencia de Nyquist, la cual es la menor frecuencia a la cual se puede muestrear el sistema linealizado tal que su respuesta se aproxime a la respuesta del sistema físico real. Por supuesto que esto es sólo una aproximación pues se trata de un método aplicado a sistemas lineales y el sistema de interés en este trabajo es no lineal.

En la figura 4.10 se muestra el diagrama de bode para la linealización en el punto $x_1 = -\frac{\pi}{2}$ y en la figura 4.11 se tiene lo mismo pero incluyendo las indicaciones de las ganancias y frecuencias del ancho de banda.

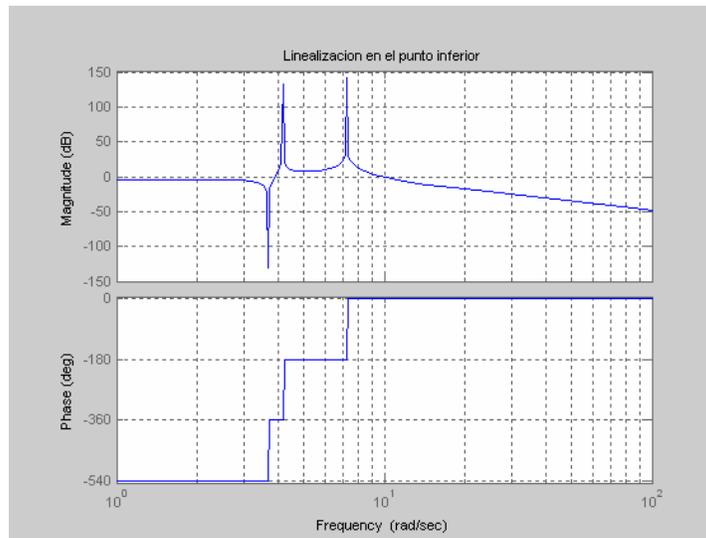


Figura 4.10 Diagramas de Bode del sistema linealizado

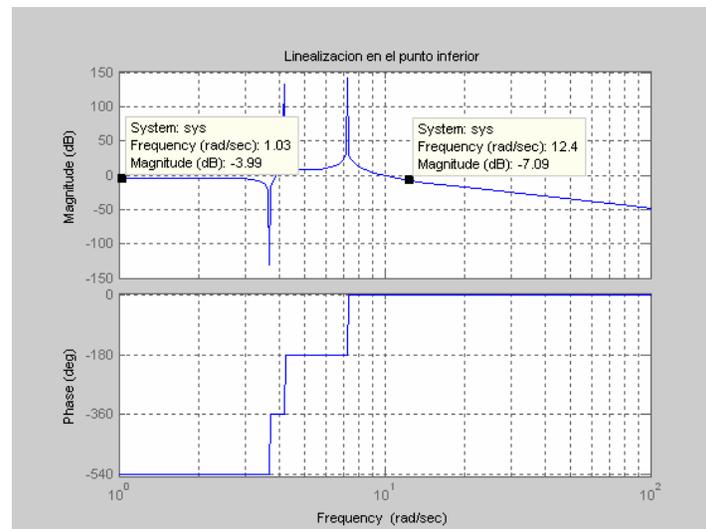


Figura 4.11 Diagramas de Bode con etiquetas en la ganancia

En la figura 4.11, se tiene un ancho de banda $B = 12.4 \frac{rad}{seg}$, lo cual representa la máxima frecuencia con ganancia cercana a 0 dB que el sistema linealizado es capaz de proporcionar. La relación de esta cantidad (B) con la frecuencia de Nyquist (F_N), y la frecuencia de muestreo (F_m), esta dada por $F_m > F_N = 2B$.

Por lo tanto, para este caso la $F_N = 24.8 \frac{rad}{seg}$, o también de 3.95 Hz, que correspondería a 0.25 segundos de tiempo de muestreo máximo.

En las figuras 4.12 y 4.13 se considera al sistema linealizado en el punto $x_1 = \frac{\pi}{2}$. Se puede ver que $B = 5.96 \frac{rad}{seg}$ y entonces $F_N = 11.92 \frac{rad}{seg}$, o también 1.9 Hz y el tiempo de muestreo máximo sería de 0.53 segundos.

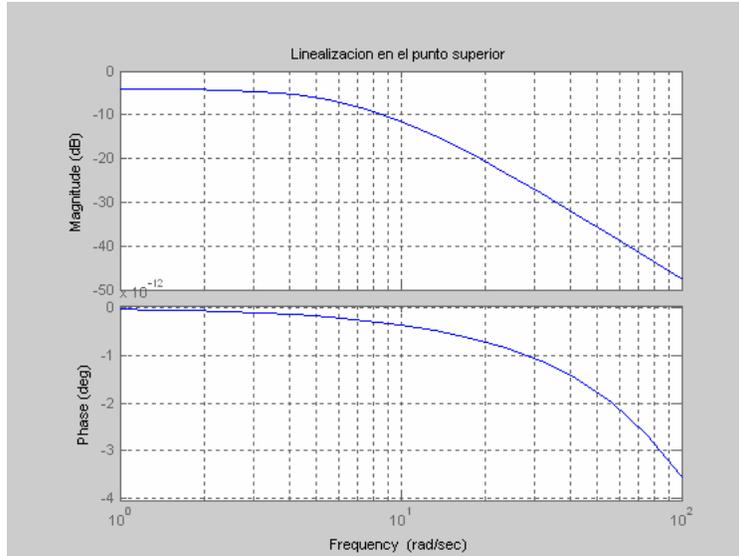


Figura 4.12 Diagramas de Bode del sistema linealizado

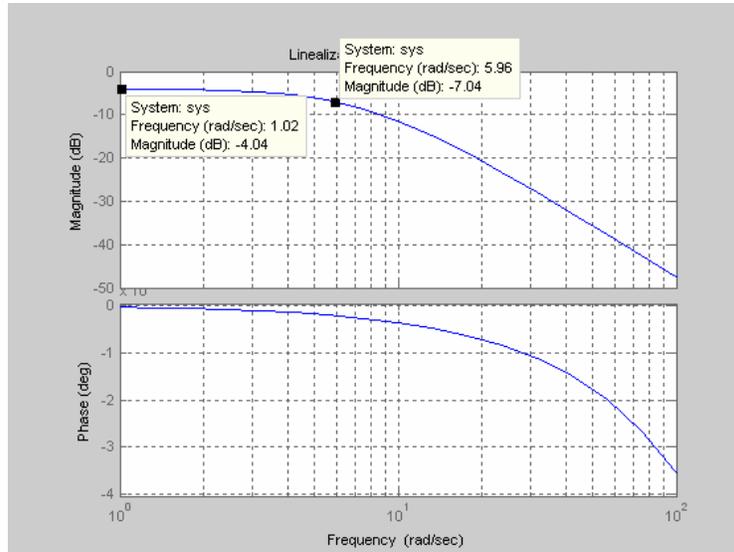


Figura 4.13 Diagramas de Bode con etiquetas

De manera resumida se tiene que,

Para el punto inferior:

$F_N = 24.8 \frac{rad}{seg}$, o también 3.95 Hz y 0.25 segundos de tiempo de muestreo máximo.

Para el punto superior:

$F_N = 11.92 \frac{rad}{seg}$, o también 1.9 Hz y 0.53 segundos de tiempo de muestreo máximo.

En este desarrollo se puede observar que las frecuencias de muestreo obtenidas, están perfectamente dentro del rango de las obtenidas en la sección 4.3.1, tal y como se esperaba, de tal forma que si se eligiera la frecuencia de muestreo 40 Hz, se estarían cubriendo todas las frecuencias de Nyquist y utilizaríamos un solo tiempo de muestreo.

4.4 Obtención de los tiempos de ejecución para el código del sistema pendubot.

Para lograr este objetivo se implementó el modelo no lineal del sistema pendubot en el sistema operativo de tiempo real QNX y se le añadió lo necesario para desplegar los tiempos de ejecución de cada corrida.

En la figura 4.14 se muestra el resultado de la medición. La mayor parte del código se ejecutó mostrando resultados en pantalla lo cual consumía parte del tiempo de ejecución. La región de la curva que cae en el rango de los 14 milisegundos, corresponde a la ejecución del código sin presentación de resultados en pantalla.

En cuanto al tiempo de simulación del código del pendubot, se puede apreciar que oscila entre 12 milisegundos (sin presentación de resultados en pantalla) y 22 miliseg (con presentación en pantalla). De acuerdo al tiempo de muestro máximo de 25 miliseg que se obtuvo en la sección 4.3, se puede ver que el sistema pendubot es de tipo “hard”, es decir que puede ser implementado en tiempo real pues cumple con la restricción del tiempo de muestreo hasta en el peor de los casos, es decir, con presentación de resultados en pantalla.

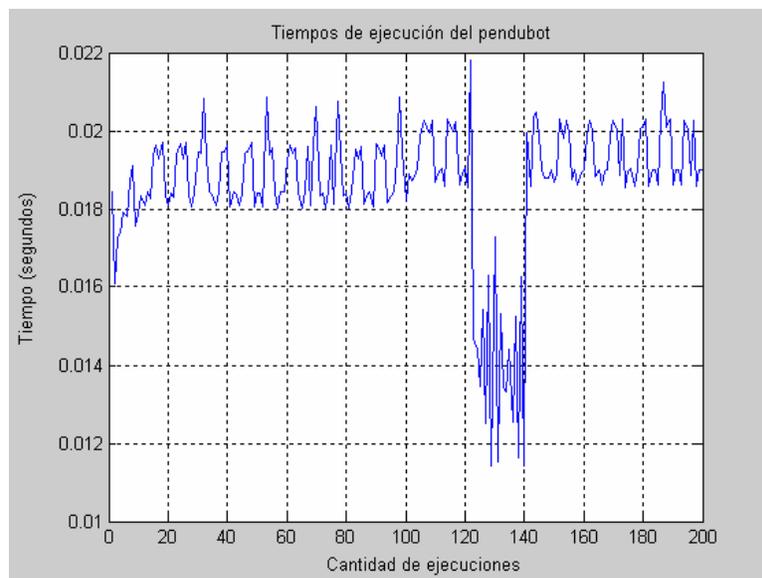


Figura 4.14 Tiempos de ejecución del código pendubot

El código desarrollado para medir estos tiempos de ejecución se muestra en el apéndice A.

4.5 Comparación de los tiempos de ejecución y de muestreo.

La comparación de los tiempos de ejecución y del tiempo de muestreo se puede apreciar en la figura 4.15. Los bloques J1, J2 y J3 representan la ejecución del código del sistema pendubot y los intervalos k1, k2 y k3 representan los intervalos en los que se realiza el muestreo. Se puede apreciar claramente que el tiempo de ejecución está completamente contenido en los intervalos de muestreo.

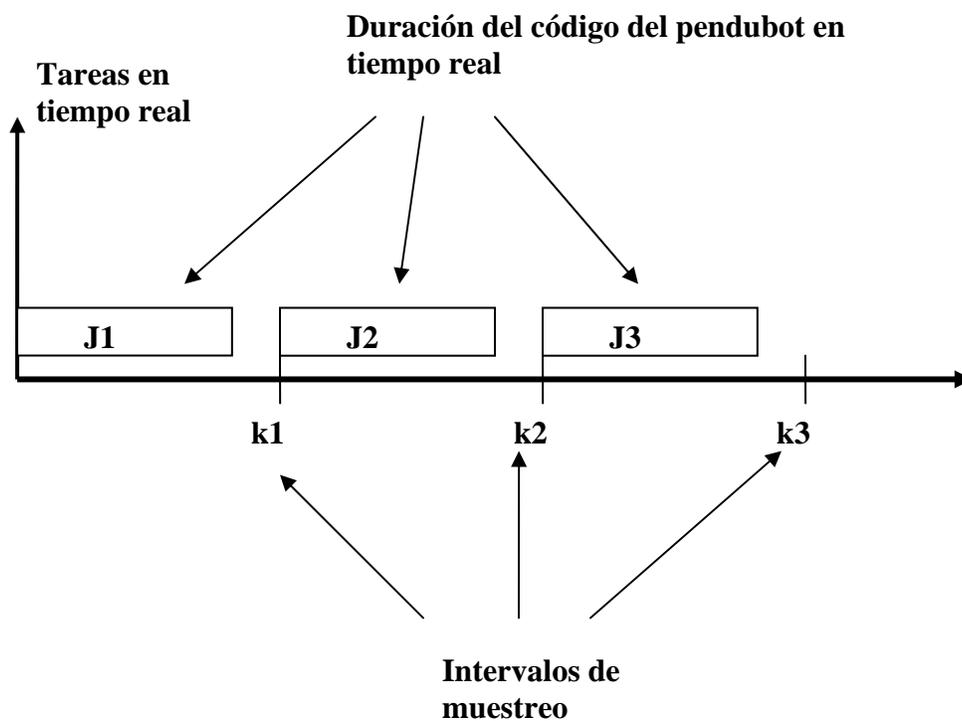


Figura 4.15 Comparación de los tiempos de ejecución y los tiempos de muestreo.

Con el fin de implementar una simulación en tiempo real del modelo no lineal del sistema pendubot, se desarrolló un análisis de la forma en que se tendría que elegir el tiempo de muestreo para este sistema. En primer lugar se realizó la comparación de las respuestas del sistema discreto a diferentes tiempos de muestreo contra la respuesta del sistema continuo obtenida de la simulación en MatLab. En segundo lugar se obtuvieron los diagramas de bode para la linealización del sistema con el fin de comparar su ancho de banda con los tiempos de muestreo obtenidos en la primera parte.

En el primer caso, para la comparación de las respuestas entre el sistema continuo y el sistema discreto, se realizó el cálculo de la desviación de la segunda señal respecto a la primera y a su vez, se obtuvieron las sumas de los elementos de cada desviación (desviación acumulada), como una forma de compararlas entre ellas y determinar cuál era la menor de las desviaciones.

Los resultados indicaron un rango de valores apropiados para realizar el muestreo del sistema. Dicho rango está ubicado entre los 25 milisegundos y los 5 milisegundos (40 Hz y 200 Hz de frecuencia de muestreo, respectivamente). Para valores por debajo del límite inferior las desviaciones tienden a disminuir, tendiendo a ser cero en el infinito, por tal razón se tomó el ínfimo y no el mínimo, es decir se tomó el mayor de los menores dado que el mínimo llegaría a ser cero. Por una razón similar, en el extremo superior, se tomó el supremo y no el máximo, es decir, se tomó el menor de los mayores.

Respecto a las frecuencias de Nyquist obtenidas en los diagramas de Bode, se puede observar que las obtenidas en el primer caso están perfectamente dentro del rango, tal y como se esperaba, de tal forma que si se eligiera la frecuencia de muestreo 40 Hz, se estarían cubriendo todas las frecuencias de Nyquist y utilizaríamos un solo tiempo de muestreo.

Finalmente, se midió el tiempo de ejecución del código de simulación del pendubot en el sistema operativo QNX, resultando que el sistema es de tipo "hard", es decir que el tiempo de ejecución es menor que los tiempos de muestreo necesario para el sistema.

4.6 Implementación de la simulación en tiempo real del sistema pendubot.

En esta sección se muestran las respuestas que se obtuvieron en la implementación en tiempo real del sistema pendubot y sus leyes de control. Para tal implementación se empleó, en todos los casos, una frecuencia de muestreo de 40 Hz correspondiente a 25 ms de tiempo de muestreo. En todas las gráficas la escala del tiempo está dada en cantidad de muestras, de tal forma que si se desea conocer el tiempo de simulación, se requiere de multiplicar la cantidad de muestras por el tiempo de muestreo, o bien dividir por la frecuencia de muestreo.

La figura 4.16 muestra la simulación en tiempo real de la ley de control estabilizante lineal con ganancias dadas por (3.14). Las condiciones iniciales son las mismas que las usadas en el capítulo 3.

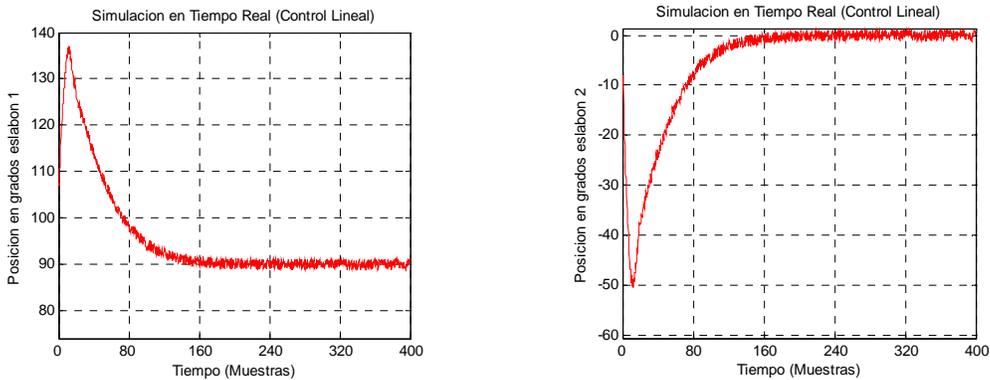


Figura 4.16 Simulación en Tiempo Real 1.

En la figura 4.17 se muestra la simulación en tiempo real de la ley de control basada en energía (3.13). El objetivo de esta ley de control es el de resolver el problema del balanceo, con lo cual se acerca al sistema pendubot al punto de equilibrio inestable (superior) partiendo del punto de equilibrio estable (inferior).

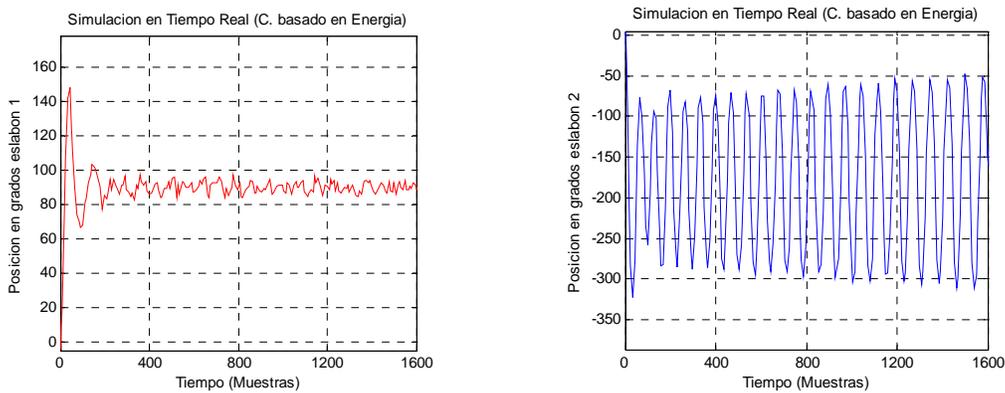


Figura 4.17 Simulación en Tiempo Real 2.

En la figura 4.18, se muestra la simulación en tiempo real del sistema pendubot y la ley de control con prealimentación (3.9). En la figura 3.4, de capítulo 3, se puede ver el diagrama general para este esquema de control. La simulación inicia en el punto de equilibrio inferior y la ley de control hace evolucionar al sistema hacia el punto de equilibrio superior.

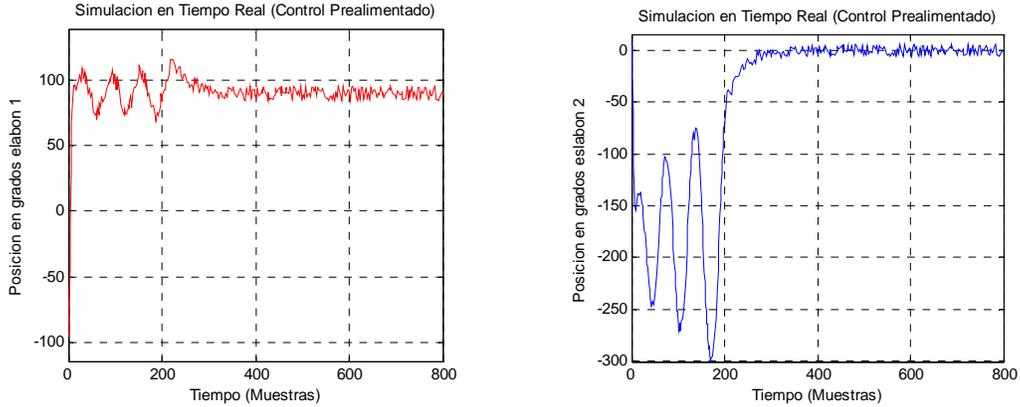


Figura 4.18 Simulación en Tiempo Real 3.

En las figuras 4.19 y 4.20, se muestran las simulaciones de tiempo real de otras leyes de control para el pendubot. En el primer caso se trata de estabilizar el punto de equilibrio estable del sistema; es decir, debido a que el modelo matemático (4.3) empleado no considera la fricción, el sistema nunca dejaría de moverse por si solo, si se deja en una condición inicial diferente del punto estable, por lo tanto se utiliza una ley de control no lineal [15] para lograr este objetivo.

En el segundo caso se trata de una técnica de control conocida como Lagrangianos Controlados [23], [24], [25], [26], la cual modifica la energía del sistema tal que se logre el objetivo deseado, por ejemplo estabilizar el segundo eslabón del pendubot en el equilibrio superior, a pesar de la estabilidad del primer eslabón.

Se muestran estas simulaciones para apoyar la idea de que una de las utilidades de la simulación desarrollada es la de probar diferentes técnicas de control en tiempo real y tomar en cuenta las características del sistema, por ejemplo el efecto del tiempo de muestreo y el tiempo de ejecución.

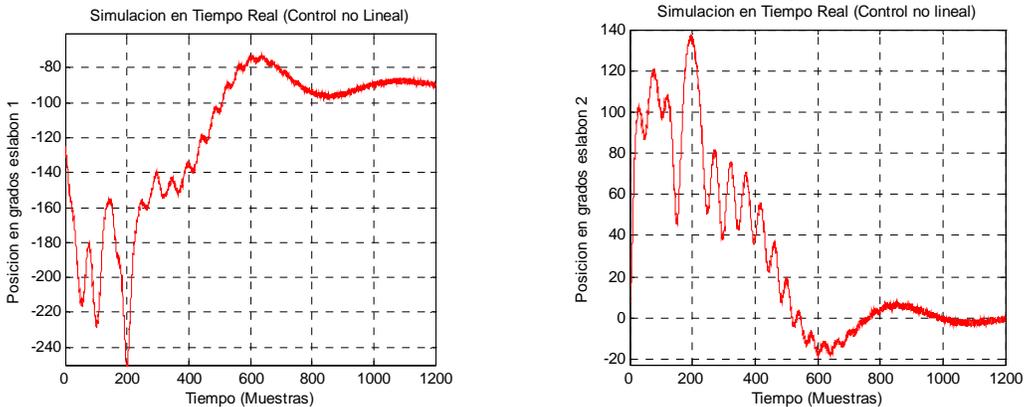


Figura 4.19 Simulación en Tiempo Real 4. Control no Lineal.

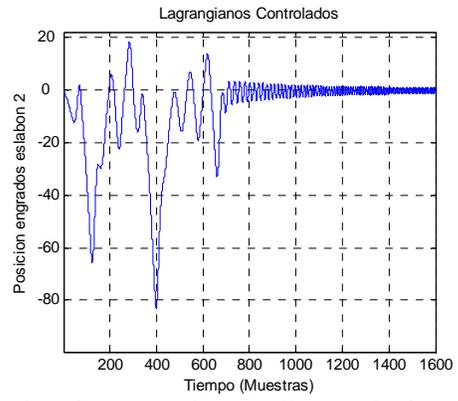
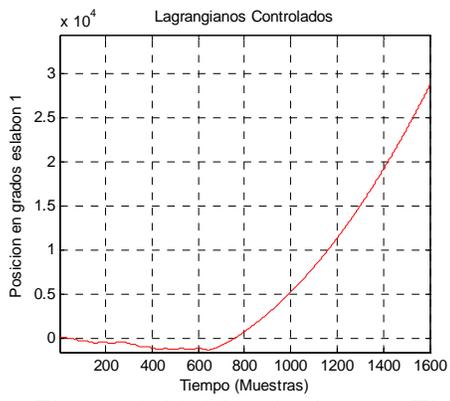


Figura 4.20 Simulación en Tiempo Real 5. Lagrangianos Controlados

CAPÍTULO 5

CONCLUSIONES Y PERSPECTIVAS

5.0 Conclusiones.

En el capítulo uno se hizo una descripción del sistema subactuado, no lineal conocido como pendubot; se describieron sus objetivos de control y las técnicas de control que se han empleado para resolverlos. Se realizó el planteamiento del problema que ocupa este trabajo, en el cual se cuestiona la posibilidad de implementar en tiempo real, la simulación del sistema y de sus leyes de control. Se realizó la descripción de la metodología para resolver el problema planteado, estableciendo que el objetivo principal de la tesis es demostrar la hipótesis descrita en la sección 1.7, la cual consiste en la posibilidad de desarrollar la simulación del pendubot.

Para comenzar con el desarrollo de la simulación arriba mencionada, se partió de la determinación y el análisis del modelo matemático del pendubot. Dicho análisis se describe con precisión en el capítulo dos. Se determinaron los puntos de equilibrio y la estabilidad de estos usando el modelo linealizado del sistema.

En el capítulo tres se explicaron los objetivos de control del pendubot con los que se trabajaría en esta tesis. Se habló del problema del balanceo y de la estabilización. Se describieron las técnicas de control y se diseñaron las correspondientes leyes, que para resolver cada uno de los objetivos de control del sistema se utilizaron en este trabajo; finalmente se mostraron las simulaciones de tiempo continuo del sistema controlado y en lazo abierto.

El siguiente paso consistió en obtener el modelo no lineal de tiempo discreto del sistema para realizar su simulación de tiempo real. Para completar de manera adecuada este desarrollo, era necesario determinar las características de tiempo real del sistema, ya que como se puede ver en (4.3) las ecuaciones en tiempo discreto dependen del tiempo de muestreo y la simulación depende de los tiempos de ejecución del código del sistema. En el capítulo cuatro se muestra el desarrollo del análisis para la elección del tiempo de muestreo del sistema. En primer lugar se realizó la comparación de las respuestas del sistema discreto a diferentes tiempos de muestreo contra la respuesta del sistema continuo obtenida de la simulación en MatLab. En

segundo lugar se obtuvieron los diagramas de bode para la linealización del sistema con el fin de comparar su ancho de banda con los tiempos de muestreo obtenidos en la primera parte.

En el primer caso, para la comparación de las respuestas entre el sistema continuo y el sistema discreto, se realizó el cálculo de la desviación de la segunda señal respecto a la primera y a su vez, se obtuvieron las sumas de los elementos de cada desviación (desviación acumulada), como una forma de compararlas entre ellas y determinar cuál era la menor de las desviaciones.

Los resultados indicaron un rango de valores apropiados para realizar el muestreo del sistema. Dicho rango se ubica entre los 25 milisegundos y los 5 milisegundos (40 Hz y 200 Hz de frecuencia de muestreo, respectivamente).

Para valores por debajo del límite inferior las desviaciones tienden a disminuir, tendiendo a ser cero en el infinito, por tal razón se tomó el ínfimo y no el mínimo, es decir se tomó el mayor de los menores dado que el mínimo llegaría a ser cero, pero tener un tiempo de muestreo de cero es prácticamente imposible. Por una razón similar, en el extremo superior, se tomó el supremo y no el máximo, es decir, se tomó el menor de los mayores.

Respecto a las frecuencias de Nyquist obtenidas en los diagramas de Bode, se puede observar que las obtenidas en el primer caso están perfectamente dentro del rango, tal y como se esperaba, de tal forma que si se eligiera la frecuencia de muestreo 40 Hz, se estarían cubriendo todas las frecuencias de Nyquist y utilizaríamos un solo tiempo de muestreo.

Se midió el tiempo de ejecución del código de simulación del pendubot en el sistema operativo QNX, resultando, con esto y lo anterior, que el sistema se puede clasificar de la siguiente manera: De acuerdo a sus límites temporales, es un sistema en tiempo real estricto (*hard real-time*), dado que el tiempo de ejecución es menor que los tiempos de muestreo necesarios para el sistema. De acuerdo a sus escalas de tiempo, se trata de un sistema basado en reloj, ya que como se puede ver en el apéndice A, dentro de los programas hay una subrutina para programar el reloj del sistema. Y debido a su interacción con el medio, se trata de un sistema no embebido, pues es independiente de que se implemente en QNX o en otro sistema operativo mientras que se respeten sus tiempos de muestreo.

Finalmente en el mismo capítulo cuatro, se muestran los resultados de las simulaciones en tiempo real que se realizaron del pendubot y sus leyes de control, encontrando que las respuestas corresponden muy bien al caso de las simulaciones en tiempo continuo mostradas en el capítulo tres. Lo cual no lleva a concluir que la hipótesis planteada en el capítulo uno es verdadera y queda demostrado con el análisis realizado.

5.2 Perspectivas (Trabajo a futuro).

Como una extensión del presente trabajo se sugiere que el método para obtener los tiempos de muestreo del pendubot, se aplique a sistemas no lineales en general y se busque el desarrollo de una forma generalizada de simulaciones en tiempo real para estos sistemas.

En cuanto a la implementación de estas simulaciones, se sugiere el desarrollo de simuladores dedicados, implementados en microcontroladores o DSP (procesador digital de señales), con los periféricos necesarios para la entrada y salida de información, así como para la programación de las ecuaciones no lineales. De esta forma se llevaría un paso más allá el concepto de simulación de tiempo real. Este tipo de sistemas sería más barato que contar con el sistema no lineal original completo; además de que con el mismo hardware se podrían implementar diferentes sistemas tanto lineales como no lineales y se emplearían para probar técnicas de control y para la enseñanza de la ingeniería en áreas como el modelado, control, robótica, mecatrónica, etc.

REFERENCIAS

- [1] Spong, M.W., "Energy Based Control of a Class of Underactuated Mechanical Systems," 1996 IFAC World Congress, San Francisco, CA, July, 1996.
- [2] Zhong, W. and Röck, H., "Energy and Passivity Based Control of the Double Inverted Pendulum on a Cart," in IEEE Conf. On Control Applications, pp 1-6, enero 2001.
- [3] Astrom, K.J. and Furuta, K., "Swinging up a pendulum by energy control," *Automatica*, 36(2):287-295, February 2000.
- [4] Bortoff, S. and Spong, M.W., "Pseudolinearization of the Acrobot Using Spline Functions," in IEEE Conf. On Decision and Control, Tucson, AZ, pp 593-598, Dec. 1992.
- [5] Furuta, K. and Yamakita, M., "Swing up Control of Inverted Pendulum," in IECON'91, pp 2193-2198, 1991.
- [6] Olfati-Saber, R. "Global Stabilization of a Flat Underactuated System: The Inertia wheel Pendulum"; Proc. Of the 40th Conf. On Decision and Control, Orlando, FL, December 2001.
- [7] Block, D. J. and Spong, M.W., "Mechanical Design and Control of the Pendubot"; SAE Earthmoving Industry Conference, Peoria, IL, April 4-5, 1995.
- [8] Spong, W.M., Corke, P and Lozano, R., "Nonlinear Control of the Inertia Wheel Pendulum," *Automatica*, vol 37, pp 1845-1851, 2001.
- [9] Spong, M.W., Block, D.J. and Astrom, K.J.; "The Mechatronics Control Kit for Education and Research"; IEEE Conference on Control Applications, México City, pp105-110, Sept. 5-7, 2001.
- [10] Ramos, L. E, Castillo B., Alvarez J. "Nonlinear Regulation of an underactuated systems", International Conference on Robotics and Automation, Albuquerque, NM, USA, Abril 1997.
- [11] Olfati-Saber, R. "Nonlinear Control and Reduction of Underactuated Systems with Symmetry II: Unactuated Shape Variables Case"; Proc. Of the 40th Conference on Decision and Control, Orlando FL, Dec 2001.

- [12] Rosas, A, Álvarez Gallegos, J. Castro Linares, ; “Stabilization of a Class of Underactuated Systems”; in CDC2000, Vol 1, pag 2168, paper num 60.
- [13] Ramos, V.L. “Control de un Sistema Mecatrónico Subactuado: Pendubot” Tesis de Maestría, CINVESTAV 1996.
- [14] Jain, A., and Rodríguez, G., “An Analysis of the Kinematics and Dynamics of Underactuated Manipulators,” IEEE Trans. On Robotics and Automation, Vol 9, No. 4, pp. 411-422, Aug. 1993.
- [15] Spong, M.W., “The Control of Underactuated Mechanical Systems”; I Conferencia Internacional de Mecatrónica, México, D.F. Enero 1994.
- [16] Haro, B. and Begovich, O. “A Set of Controllers: A real time applications for the pendubot,” Proc. Of International Symposium on Robotics and Automation, ISRA’98. Saltillo, Coah, México (1998) p. 145.
- [17] Sánchez, E.N., Leno, L.A., and Hsu, Y.C., “Real-Time Fuzzy Swing up control for an Underactuated Robot”; Proc of 2do Intell. Control Workshop, JCIS’98, Durham, NC, oct 24-28, 1998, pp 167-170.
- [18] Shiriaev, A. S., Kolesnichenko, O. “On Passivity Based Control for Partial Stabilization of Underactuated Systems”, Proc. of CDC2000, Vol 1, pag 2174, paper number 1932.
- [19] Ortega, R, Spong, M.W., “Stabilization of Underactuated Mechanical Systems via Interconnection and Damping Assignment”, March 16-18 2000.
- [20] Saito, F., Fukuda, T., and Arai, F., “Swing and Locomotion Control for Two-Link Brachiation Robot,” in Proc. 1993 IEEE Int. Conf. On Robotics and Automation, pp 719-724, Atlanta, GA, 1993.
- [21] Spong, M.W. “Modeling and Control of Elastic Joint Robots,” ASME J. Dyn. Systems, Meas. And Control, Vol. 109, pp. 310-319, Dec. 1987.
- [22] Fantoni, Lozano, Spong “Energy Based Control of the Pendubot”, IEEE Trans. On Automatic Control, Vol. 45, No. 4, April 2000.
- [23] Bloch, A. Leonhard, N and Marsden, J. “Controlled Lagrangians and the Stabilization of Mechanical Systems”; Proc. IEEE Conf. Decision and Control, Tampa, Fl, USA, Dec 1998.
- [24] Bloch, A, Chang, D.E., Leonhard, N and Marsden, J, “Controlled Lagrangians II: Potencial Shaping,” in Trans. Automat. Control, vol. 46, No. 10, pp. 1556-1571, oct 2001.

- [25] Bloch, A. Leonhard, N and Marsden, J, “Controlled Lagrangians and the Stabilization of mechanical systems I: The first matching theorem,” IEEE Trans. Automat. Control , vol. 45, pp.2253-2270, 2000.
- [26] Bloch, A. Leonhard, N and Marsden, J, “Stabilization of the Pendulum on a rotor arm by the methods of controlled Lagrangians,” in Proc. IEEE Int. Conf Robotics Automat., Detroit, MI, 1999, pp. 500-505.
- [27] Ramos, L. E.; Castillo B, “Robust Regulation via sliding mode for an underactuated system”, IASTED International Conference on Robotics and Manufacturing, Cancún, Mex, Mayo 1997.
- [28] González-Hernandez, H. G., Álvarez-Gallegos, J. Álvarez, J. “Experimental Analysis and Control of a Chaotic Pendubot” in The Internacional Journal of Robotics Research, Volume 23 Issue 9- Publication Date: 1 September 2004
- [29] Proakis, J. G.; Manolakis, D. G. “Tratamiento Digital de Señales: Principios, Algoritmos y Aplicaciones” 3ª edición, Prentice Hall, Madrid, 1998.
- [30] Nise, N. S.; “Sistemas de Control para Ingeniería, 1ª ed” Compañía Editorial Continental, México, 2002.
- [31] Ogata, K. “Ingeniería de Control Moderna” Prentice Hall, México, 1998.
- [32] Ogata, K “Dinámica de Sistemas” Prentice Hall, México, 1998.
- [33] Brogan, W. L. “Moder Control Theory, 3a ed”, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
- [34] Arnold, V. I. “Mathematical Methods of Classical Mechanics” Springer-Verlag, New York, 1978.
- [35] Khalil, H. K. “Non Linear Systems”, Prentice Hall, 3ra edición, 2002.
- [36] Kailath, T. “Linear Systems”, Prentice Hall, 1980.
- [36] Burns A. and Wellings A. J.. “*Real-time systems and their programming languages*”. Addison-Wesley, 3rd Edition, 2002.
- [37] Kopetz H. “*Real-Time Systems*”. Kluwer Academic Publishers, 1997.
- [38] Butazzo G, “*Hard Real-Time Computer Systems*”, Kluwer Academic Publishers, 2000.

APÉNDICE A

Programa con el que se midió el tiempo de ejecución en QNX.

El programa empleado para medir el tiempo de ejecución en QNX se muestra enseguida.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <sys/netmgr.h>
#include <sys/neutrino.h>
#include <signal.h>
#include <unistd.h>

#define BILLION 1000000000L;

/* Definición de constantes */

float pi=3.14159;
float l1=0.26987;
float lc1=0.13494;
float l2=0.38417;
float lc2=0.19208;
float lz1=1.3863e-2;
float lz2=1.6749e-2;
float m1=0.5289;
float m2=0.3346;
float g=9.81;
unsigned long Delta=27; /* Este es el muestreo mas grande para el que se tuvo un
comportamiento*/

/* Estable */

float A=0.0001;
float B=0.00001;
float C=0.0001;
float D=0.0001;
float E=0.0001;
float F=0.0001;
float G=0.001;

float D11=0.001;
float D12=0.001;
float D22=0.001;
float G1=0.001;
float G2=0.001;
float C1=0.001;
float C2=0.001;
```

```

float f1=0.001;
float f2=0.001;
float f3=0.001;
float f4=0.001;

float g3=0.001;
float g4=0.001;

float aux1=0.001;

float x1=M_PI/2-M_PI/10; /* Posiciones angulares en radianes */
float x2=M_PI/20;
float x3=0.00001;
float x4=0.00001;

float u=0.0001;
void pend(int); /* Declaracion de la funcion pend */

struct timespec start, stop; /* Las siguientes dos definiciones son de la prueba 2*/
unsigned long c;
double accum;
int i, j, k; /**/

float *tiempo; /* Apuntador que guarda los tiempo de ejecucion */
FILE *fp; /* Apuntador para guardar en el archivo */
long int fi[200]; /* Arreglo para guardar los tiempo de ejecucion enteros */
/* y es del tamaño de repeticiones */
char fc[200][33]; /* Arreglo para guardar los tiempos de ejecucion en char*/
char espacio = ' '; /* Espacio que se intercala para matlab */

main()
{
/*Asignacion dinamica del apuntador*/
tiempo = malloc(200*sizeof(float));
if(!tiempo){printf("fallo de asignacion de memoria\n");exit(0);}

/* Aquí se calculan las partes que son constantes en la ecuaciones */

A=m1*lc1*lc1+m2*(l1*l1+lc2*lc2)+Iz1+Iz2;
B=m2*l1*lc2;
C=m2*lc2*lc2+Iz2;
E=m1*g*lc1+m2*g*l1;
F=m2*g*lc2;
G=m2*l1*lc2;

//u=0.00001-0.00001;

/* Aquí termina */
/*****
*/
/* Aquí comienza el ciclo para medir el tiempo de ejecución del código para el pend */

```

```

/*****
*/

while(1)
{
for(k=0; k<=(200-1); k++)
{
/* Prueba 2,
*/
clock_gettime(CLOCK_REALTIME,&start);
for(i=1;i<=100;i++){pend(0);} /* Lo hace 100 veces para poder medir el
tiempo*/

clock_gettime(CLOCK_REALTIME,&stop);

//start = inicio.tv_sec + inicio.tv_nsec/BILLION;
//end = fin.tv_sec + fin.tv_nsec/BILLION;
//c = end - start;
nanosegundos */

//printf("duracion%.11f %lf mseg\n", c,c);

accum = ( stop.tv_sec - start.tv_sec ) + (double)( stop.tv_nsec - start.tv_nsec
)/ (double)BILLION;

//accum = (double)( stop.tv_nsec - start.tv_nsec )/ (double)BILLION;
printf( "accum = %lf\n", accum );
*(tiempo+k)=accum/100;
}

for(i=0; i<=(200-1); i++) /* Se preparan los datos para guardarlos
*/
{
printf("%f\n",*(tiempo+i)); fi[i]=100000000.0* *(tiempo+i); /* Se recorre el punto decimal */
ltoa(fi[i], fc[i], 10); /* Convierte de int a char base
10 */

printf("%i\n",fi[i]); /* presenta la secuencia en
enteros*/

}

/* Abrir para escritura */
if((fp=fopen("datos","w"))==NULL){printf("no se puede abrir el archivo\n");}

/* Es importante hacer que el tamaño de todos los datos sea igual, */

/* lo cual se logra al recorrer el punto decimal, tomando el factor adecuado */
/* Es importante dado que el proximo ciclo "for" para "j" nos da el numero */
/* de digitos en la cifra a guardar y si llega a ser diferente, se guardara */
/* basura.*/
/* El ciclo "for" para "i", nos da la cantidad de datos a guardar, es decir el */
/* el numero de repeticiones */

for (i=0; i<=(200-1); i++)
{
/* Se escriben los datos
en el archivo*/

for(j=0; j<=6; j++) fwrite(&fc[i][j], sizeof(char), 1, fp); /*los datos tienen 7 digitos*/
fwrite(&espacio,sizeof(char),1,fp); /* Se intercala el
espacio */
}
}

```

```

        }

fclose(fp); /* Se cierra el archivo */

free(tiempo);
exit(0);

} /*Aqui termina el ciclo infinito del while*/

} /*Aqui termina la funcion principal*/

void pend(int senal)
{

/* Aquí empieza la programación del modelo del pendubot */

D11=A+2*B*cos(x2);
D12=C+B*cos(x2);
D22=C;

G1=E*cos(x1)+F*cos(x1+x2);
G2=F*cos(x1+x2);

C1=-G*(2*x3+x4)*x4*sin(x2);
C2=G*x3*x3*sin(x2);

aux1=D11*D22-D12*D12;

f1=x3;
f2=x4;
f3=(D22*((D12*(C2/D22))+D12*(C2/D22))-C1-G1)/aux1;
f4=(-D12*((D12*(C2/D22))+D12*(C2/D22))-C1-G1)/aux1-(C2/D22)-(G2/D22);

g3=D22/aux1;
g4=-D12/aux1;

/****** Se calcula ley de control *****/
u=36.3905*(x1-M_PI/2)+36.1282*x2+10.1064*x3+7.7465*x4;

x1= f1*(Delta/1000.0) + x1; /*Se divide entre 1000 para tenerlo en milisegundos*/
x2= f2*(Delta/1000.0) + x2;
x3= (f3+g3*u)*(Delta/1000.0) + x3;
x4= (f4+g4*u)*(Delta/1000.0) + x4;

/* Aquí termina el modelo del pendubot */
j=j+1;
printf("x1= %f, x2= %f, j=%i\n", x1, x2, j);
//printf("x1= %f, x2= %f, j=%i, accum=%lf\n", x1, x2, j,accum);
}

```

APÉNDICE B

CONSTANTES DEL PENDUBOT

CONSTANTE	VALOR
Longitud del eslabón 1 (l_1)	0.26987 m
Longitud media del eslabón 1 (l_{c1})	0.13494 m
Longitud del eslabón 2 (l_2)	0.38417 m
Longitud media del eslabón 2 (l_{c2})	0.19208 m
Momento de Inercia 1 (I_1)	$1.3863 \times 10^{-2} \text{ Kg-m}^2$
Momento de Inercia 2 (I_2)	$1.6749 \times 10^{-2} \text{ Kg-m}^2$
Constante de la gravedad (g)	9.81 m/seg ²