

**Instituto Tecnológico de Costa Rica**

Escuela de Ingeniería en Electrónica



Escuela de Ingeniería en Electrónica

“Creación de un sistema didáctico e interactivo de control  
automático basado en el péndulo de Furuta”

Informe de Proyecto de Graduación para optar por el título de  
Ingeniero en Electrónica con el grado académico de Licenciatura

Víctor Edgar Ramírez Rojas

Cartago, agosto de 2016

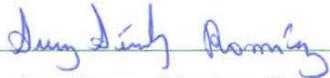
**Acta de Aprobación**  
Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería en Electrónica  
Proyecto de Graduación

**Acta de Aprobación**

Defensa del Proyecto de Graduación  
Requisito para optar por el título de Ingeniero en Electrónica  
Grado Académico de Licenciatura  
Instituto Tecnológico De Costa Rica

El Tribunal Evaluador aprueba la defensa del Proyecto de Graduación denominado “Creación de un sistema didáctico e interactivo de control automático basado en el péndulo de Furuta” realizado por el señor Víctor Edgar Ramírez Rojas y, hace constar que cumple con las normas establecidas por la Escuela de Ingeniería en Electrónica del Instituto Tecnológico de Costa Rica

Miembros del Tribunal Evaluador



Ing. Sussy Sánchez Ramírez

Profesora Lectora



Ing. Guillermo Castro Badilla

Profesor Lector



Ing. Eduardo Interiano Salguero

Profesor Asesor

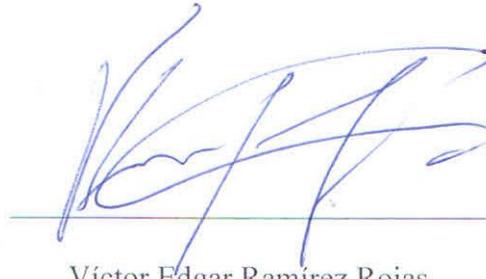
Cartago, 22 de agosto del 2016

## Declaratoria de Autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado, en su totalidad, por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Víctor Edgar Ramirez Rojas

Cédula: 2-0565-0707

## Resumen



Actualmente, en el curso Laboratorio de Control Automático de la Escuela Ingeniería en Electrónica, el estudiante debe realizar dos proyectos cortos en donde se les enseña a caracterizar la planta en estudio y a diseñar un control adecuado para dicha planta. Lo idóneo sería que el estudiante cuente con las herramientas para que pueda aprender y experimentar precisamente con los diferentes métodos y estrategias de control, que es el objetivo principal del curso, en lugar del diseño de un solo método de control.

Por lo tanto, se creó un sistema donde se le facilite al estudiante la realización de los proyectos cortos del Laboratorio de Control Automático, diseñado de manera tal que carezca de problemas mecánicos y posea los sensores adecuados para la interpretación de las variables de control.

Además, el sistema cuenta con la opción de graficar sus respuestas y definir los valores de las variables de control necesarias, y así poder visualizar los resultados de su control y comportamiento de la planta, con el fin de poder comparar estos valores experimentales con el de sus cálculos teóricos.

Palabras claves: Laboratorio de Control Automático, Péndulo Invertido de Furuta, Caracterización de una Planta, Control PID, Control LQR.

## Abstract

Currently, in the Automatic Control Laboratory course at the Electronics Engineering School, the student must perform two short projects where they are taught to characterize the plant and to adequately control the plant. The ideal would be that the student has the tools so he can learn and experience precisely the different methods and control strategies, which is the main objective of the course, rather than the design of a single method of control.

Therefore, a system was created which facilitates the student the realization of specific Automatic Control Laboratory projects in an easy way. The system was designed to lack mechanical problems and to facilitate proper interpretation of the control data acquired by the sensors.

The system also has the option to plot their answers and define the values of the necessary control variables, so the student can view the results of their control and plant behavior, in order to compare these experimental values with theoretical calculations.

Keywords: Automatic Control Laboratory, Inverted Pendulum Furuta, Characterization of a Plant, PID Control, LQR.

## Dedicatoria

Al Padre Celestial por guiar mis pasos y haberme permitido culminar, con mucho esfuerzo, este Proyecto de Graduación.

A mis queridos padres, Gerardo y Vilma, por enseñarme a valorar todo cuanto me rodea y poseo, por estar siempre presentes y por motivarme cada día para lograr esta nueva meta. Mis triunfos son también de ustedes.

A mis hermanos, Adriana y Diego Alonso, por el apoyo moral y motivación para seguir adelante.

A mi esposa, Sandra Johanna, por su apoyo y paciencia en todo momento.

A mi hija, Celina Valeska, para que mi esfuerzo sea ejemplo de superación, entrega y constancia en el transcurso de nuestra existencia.

Víctor Edgar Ramírez Rojas.

## Agradecimientos

Agradezco profundamente a Dios, Todopoderoso, por regalarme el don de la vida y permitirme alcanzar esta meta.

Mi sincero agradecimiento para todas las personas que de una u otra forma contribuyeron desinteresadamente a la realización de este estudio.

Al Máster, Eduardo Interiano Salguero, Director de Tesis, por su desempeño y esmero como guía de la presente investigación.

Gracias

# Índice General

Acta de Aprobación.....	II
Declaratoria de Autenticidad .....	III
Resumen.....	III
Abstract .....	IV
Dedicatoria .....	V
Agradecimientos .....	VI
Índice General .....	VII
Índice de Figuras .....	IX
Índice de Tablas.....	XI
1. Introducción .....	1
1.1. Solución propuesta .....	2
1.2. Objetivos .....	3
1.2.1. Objetivo General .....	3
1.2.2. Objetivos Específicos.....	4
1.3. Estructura de la Tesis .....	4
2. Marco teórico.....	6
2.1. Control Automático.....	6
2.2. Funcionamiento del Péndulo de Furuta.....	7
2.3. Los sensores .....	9
2.4. El Actuador .....	11
2.4.1. No linealidad del motor .....	12
2.4.2. No simetría del motor .....	13
2.5. Función de transferencia y variables de estado .....	15
3. Diseño y desarrollo de la solución .....	18
3.1. Estructura de la planta .....	18
3.2. Innovación en el diseño de la planta.....	20
3.3. El programa para la interfaz de usuario.....	23
3.4. El microprocesador .....	23
3.5. Los sensores .....	24

3.6.	La aplicación SiDiCo.....	26
4.	Validación y Análisis de Resultados .....	40
5.	Conclusiones y Recomendaciones .....	62
5.1.	Conclusiones .....	62
5.2.	Recomendaciones .....	63
6.	Apéndices.....	65
6.1.	Hoja de Datos de la Planta .....	65
6.1.1.	Chasis del Péndulo de Furuta.....	65
6.1.2.	Constantes del Motor .....	65
6.1.3.	Dinámica del Péndulo .....	66
6.1.4.	Función de Transferencia [10].....	66
6.1.5.	Variables de Estado [10].....	66
6.2.	Proyecto Corto 1 SiDiCo (Furuta SIMO) .....	68
6.3.	Proyecto Corto 1 SiDiCo (Motor SIMO).....	78
6.4.	Proyecto Corto 1 SiDiCo (Péndulo SISO) .....	89
6.5.	Proyecto Corto 2 SiDiCo (Control Furuta PID).....	99
6.6.	Diagrama de Conexión Esquemático de la Planta .....	108
6.7.	Diseño del PCB de la Planta.....	109
6.8.	Diagrama de Conexión Esquemático y Diseño del PCB del brazo.....	110
6.9.	Estructura Protectora de Acrílico .....	111
6.10.	Carta de Entendimiento .....	112
7.	Bibliografía .....	113

## Índice de Figuras

Imagen 1.1: Péndulo de Furuta.....	3
Imagen 2.1: Diagrama de bloques de un control de motor CD .....	7
Imagen 2.2: Péndulo de Furuta.....	8
Imagen 2.3: Diagrama de giro de un codificador incremental .....	9
Imagen 2.4: Diagrama de posición de un codificador absoluto.....	10
Imagen 2.5: Fricción en anillos rozantes.....	11
Imagen 2.6: Comportamiento de un motor real [4] .....	12
Imagen 2.7: Partes del motor DC.....	14
Imagen 2.8: Composición básica del controlador de giro bidireccional de un motor (puente H).....	15
Imagen 3.1: Ejemplo de pieza disponibles en ServoCity.....	19
Imagen 3.2: Péndulo de Furuta Diseñado.....	19
Imagen 3.3: Tarjeta Bluetooth - serie utilizado en SiDiCo (HC-05) .....	21
Imagen 3.4: Módulo inalámbrico de acople de energía .....	21
Imagen 3.5: Diagrama de comunicación de SiDiCo .....	22
Imagen 3.6: KitProg y Chip de desarrollo PSoC-059LP utilizado en SiDiCo.....	22
Imagen 3.7: Motor con codificador magnético incremental utilizado en SiDiCo .....	25
Imagen 3.8: Codificador E5 utilizado en SiDiCo .....	26
Imagen 3.9: Pantalla principal de SiDiCo .....	27
Imagen 3.10: Pruebas y Ajustes de SiDiCo.....	28
Imagen 3.11: Ventana de configuraciones de SiDiCo .....	29
Imagen 3.12: Modelado del motor de SiDiCo.....	30
Imagen 3.13: Modelado del Péndulo de SiDiCo.....	31
Imagen 3.14: Modelado completo de SiDiCo .....	32
Imagen 3.15: Modulo de Motor Real en SiDiCo.....	33
Imagen 3.16: Modulo de control del programa SiDiCo .....	36
Imagen 3.17: Diagrama de bloques del control PID implementado.....	37
Imagen 3.18: Ecuación del PID implementado .....	37
Imagen 3.19: Diagrama de bloques del control LQR implementado.....	39
Imagen 4.1: Tensión suministrada por el módulo inalámbrico (sin carga).....	40

Imagen 4.2: Tensión suministrada por el módulo inalámbrico (con carga) .....	41
Imagen 4.3: Diagrama de bloques implementado para controlar la periodicidad de transmisión .....	42
Imagen 4.4: Rutina de prueba del Bluetooth.....	42
Imagen 4.5: Verificación de tiempos de transmisión entre etapas .....	43
Imagen 4.6: Componente UART necesario para comunicase con el Bluetooth .....	45
Imagen 4.7: Módulo de implementación PWM utilizado .....	46
Imagen 4.8: Señal de PWM aplicada al motor .....	46
Imagen 4.9: Diagrama de conexión del decodificador de cuadratura.....	48
Imagen 4.10: Prueba de captura de ángulo del péndulo.....	48
Imagen 4.11: Comportamiento no lineal y asimétrico del motor DC .....	50
Imagen 4.12: Trayectoria teórica de un péndulo subamortiguado [10].....	51
Imagen 4.13: Respuesta del péndulo subamortiguado de SiDiCo .....	52
Imagen 4.14: Comparación de señal y modelo obtenido experimentalmente .....	54
Imagen 4.15: Representación gráfica del lugar de las raíces .....	55
Imagen 4.16: Diagrama de Simulink de la planta.....	56
Imagen 4.17: Parámetros del auto ajuste PID .....	57
Imagen 4.18: Calculo de la matriz K mediante la función LQR de Matlab.....	58
Imagen 6.1: Diagrama de Conexión Esquemático de la Planta .....	108
Imagen 6.2: Diseño del PCB de la Planta (tamaño a escala) .....	109
Imagen 6.3: Diagrama de Conexión Esquemático del distribuidor de energía del brazo.....	110
Imagen 6.4: Diseño del PCB distribuidor de energía del brazo (tamaño a escala) .....	110
Imagen 6.5: Estructura Protectora de Acrílico (tamaño a escala) .....	111

## Índice de Tablas

Tabla 1.1: Clasificaciones de una planta .....	2
Tabla 3.1: Listado de GPIO necesarios. ....	24
Tabla 3.2: Mediciones de fuerza en el motor CD, Valores en Newton .....	34
Tabla 3.3: Mediciones de tensión mínima que requiere el motor para generar movimiento .....	35
Tabla 4.1: Tabla de ajustes PID encontrados por diferentes métodos .....	57
Tabla 4.2: Tabla de ajustes LQR encontrados por diferentes métodos .....	60
Tabla 6.1: Parámetros del chasis creado .....	65
Tabla 6.2: Parámetros del Motor Utilizado.....	65
Tabla 6.3: Parámetros del Péndulo creado .....	66

# 1. Introducción

El Tecnológico de Costa Rica (TEC) es una universidad perteneciente al sistema estatal, fundada el 10 de junio de 1971, y se encuentra en la ciudad de Cartago, asentada en las faldas del Volcán Irazú. En más de 90 hectáreas de terreno y 105 000 metros cuadrados de construcción El Tecnológico de Costa Rica alberga las aulas, laboratorios, centros de investigación, instalaciones deportivas y culturales. [1]

En la actualidad, dicho centro educativo cuenta con 20 escuelas en la Sede Central (Cartago), entre ellas la Escuela de Ingeniería Electrónica, responsable de las carreras de Ingeniería en Electrónica, Ingeniería en Computadores e Ingeniería en Mecatrónica. La mayoría de programas del TEC cuentan con tres niveles de salida: Técnico, Licenciatura y Maestría. [1]

El curso Laboratorio de Control Automático está a cargo de la Escuela de Ingeniería en Electrónica, y dentro del currículo de varias carreras. En este curso el estudiante realiza dos proyectos cortos, en donde obtiene el modelo, de forma analítica y empírica, de una planta, según el problema asignado. Además, compara los resultados de la simulación en computadora, de ambos modelos, con la respuesta experimental de la planta; para corroborar la validez de sus modelos y escoger aquel que mejor se ajuste al modelo real.

A pesar de que la institución tiene plantas para fines académicos en las que se han venido ejecutando estos proyectos cortos, el problema es que, hasta el momento, no cuenta con sistemas académicos completos y especializados para el área de control automático, donde el estudiante pueda manipular y configurar de manera ágil los diferentes métodos de control, incluso poder graficar los resultados.

Es útil comentar que las plantas de control automático se pueden clasificar de acuerdo a la linealidad de la respuesta, así como la estabilidad, por lo que ambas clasificaciones juntas segmentan a las plantas en cuatro grupos tal y como se observa en la imagen #1

Tabla 1.1: Clasificaciones de una planta

		Estabilidad	
		NO	SI
Linealidad	NO	NO Lineal e Inestable	NO Lineal y Estable
	SI	Lineal e inestable	Lineal y Estable

### 1.1.Solución propuesta

Se decidió implementar una nueva planta, de manera que desde sus inicios de fabricación se cuiden los detalles, tanto aspectos mecánicos y físicos como el tamaño adecuado, bajo peso de la estructura y materiales de alta resistencia; también los aspectos eléctricos como que los dispositivos seleccionados sean de tensiones y potencias compatibles, muestren una precisión adecuada en sus valores e incluso la velocidad con que los muestre sea óptima.

De manera muy general, se requiere de una planta ampliamente modelable, que contenga embebida la cantidad suficiente de sensores y actuadores para poder cumplir su objetivo final: controlar la planta. Además, deberá tener un microprocesador capaz de recibir instrucciones desde el computador y que las mismas sean ejecutadas desde la planta. La planta que se ha seleccionado para dicho fin es el péndulo invertido rotacional, conocido popularmente como el péndulo de Furuta.

Una planta muy similar, es la que se observa en la siguiente imagen, denominado péndulo invertido de Furuta, la cual se clasifica como inestable y no lineal. En la actualidad, la Escuela de Ingeniería en Electrónica no cuenta con un sistema integrado de control para el Péndulo de Furuta.



*Imagen 1.1: Péndulo de Furuta*

Por lo tanto, éste proyecto viene a aportar a la Escuela de Ingeniería en Electrónica un sistema didáctico e interactivo de control automático basado en el Péndulo de Furuta, con un sistema no lineal e inestable para la implementación de los proyectos cortos del laboratorio.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Crear un sistema didáctico e interactivo de control automático, que sea inestable y no lineal, basado en el Péndulo de Furuta, con su respectiva interfaz de usuario, capaz de permitir definir las estrategias y parámetros de control.

### 1.2.2. Objetivos Específicos

- Diseñar una interfaz gráfica, que contenga todos los diferentes elementos que permitan una correcta manipulación de los algoritmos y parámetros de control
  - Implementar la interfaz gráfica.
  - Diseñar una planta de Péndulo de Furuta, que cumpla con los requisitos.
  - Construir la planta de Péndulo de Furuta.
  - Obtener los modelos teórico y experimental del Péndulo de Furuta con los mismos procedimientos que los estudiantes emplean en los proyectos cortos.
- Diseñar el sistema de control necesario para poder implementar varias estructuras de control sobre el Péndulo de Furuta.
  - Implementar el sistema de control para el Péndulo de Furuta.
  - Documentar el proyecto para su aplicación como proyecto corto en el laboratorio de Control Automático.

### 1.3. Estructura de la Tesis

Esta tesis está estructurada de la siguiente manera:

El primer capítulo posee una breve introducción, donde se describe tanto el lugar a desarrollar el proyecto, como el proyecto en sí, de esta manera facilita el entendimiento y comprensión de la necesidad y utilidad del mismo. Luego un breve resumen sobre cómo se abordará la solución y, finalmente, se procede a mostrarle al lector los objetivos u alcances del proyecto, los cuales han sido desarrollados a lo largo de la presente tesis.

Seguidamente, en el capítulo número dos, se le muestra al lector todos los conceptos teóricos necesarios, para poder desarrollar la solución.

En el capítulo número tres se describe de manera amplia la solución seleccionada y se fundamentarán las decisiones tomadas.

En el cuarto capítulo se hace una breve validación y análisis de los resultados obtenidos en la planta, así como una breve explicación de funciones extras con fines pedagógicos que se le instalaron a la planta.

Las conclusiones y recomendaciones han sido plasmadas en el capítulo quinto.

Finalmente, en la parte de apéndices se adjuntan los instructivos necesarios para guiar al estudiante a la hora de utilizar el sistema didáctico creado y que su experiencia de aprendizaje pueda ser de provecho.

## 2. Marco teórico

En este capítulo se explican los conceptos necesarios para la comprensión del presente trabajo.

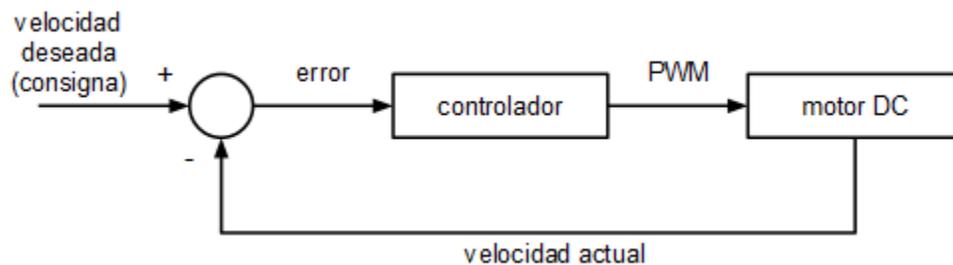
### 2.1. Control Automático

Es un conjunto de acciones destinadas a mantener o mejorar el funcionamiento del objeto administrado, para nuestros efectos “La Planta”, sin intervención humana directa, sino por medio de un control predeterminado. Esta ciencia es ampliamente utilizada en muchos sistemas mecánicos y eléctricos, que de otra manera al hombre no le es factible realizar, esto por la gran cantidad de información y el tiempo limitado. Normalmente el control automático es empleado para mejorar la productividad, la calidad y la precisión.

Es de suma importancia que el futuro profesional que prepara la Escuela de Ingeniería Electrónica tenga un vasto conocimiento en la materia de control automático, puesto que en la actualidad juegan un papel muy importante en la exploración de la aviación y el espacio, tales como el control del piloto automático, en aplicaciones militares que utilizan los sistemas de seguimiento de alta precisión, por ejemplo, el sistema angular de un radar de seguimiento, o bien, en la industria para automatizar diferentes tareas de la producción, desde llenadoras automáticas hasta la velocidad de una banda transportadora.

El objetivo de un control de esta índole consiste en medir constantemente el valor de la señal controlada, usualmente es denominado como “el valor de salida”. Con base a esta señal se genera una diferencia o “error” al comparar la salida con una señal de referencia o un “punto de referencia”, y ante el error el control genera una “acción de control”, la cual ha sido calculada con base a las características propias del control implementado.

Normalmente, dicho control también está diseñado para compensar el efecto de “las perturbaciones externas”, las cuales, con su presencia tratan de perturbar el comportamiento deseado de la variable de proceso. El conjunto entre el dispositivo de control y la planta conforma un “sistema de control automático”.



*Imagen 2.1: Diagrama de bloques de un control de motor CD*

El ejemplo más simple de control automático es un sistema de regulación directa de la velocidad o fuerza del motor (ver figura 1). El objetivo de la gestión es mantener una velocidad constante del rotor del motor. El diagrama de bloques de este ejemplo es típico de muchos sistemas de control automático, independientemente de su naturaleza física. Este sistema es un sistema continuo de un solo lazo cerrado de control automático.

## 2.2. Funcionamiento del Péndulo de Furuta

El péndulo de Furuta es como normalmente se le llama al péndulo invertido rotacional. Llamado así por su inventor Katsuhisa Furuta y sus colegas, en 1992, en el Instituto de Tecnología de Tokio [2]. Es un ejemplo de un oscilador no lineal, de ahí su amplio interés de estudio en la teoría de sistemas de control. El péndulo es sub-actuado y extremadamente no lineal debido a las fuerzas gravitacionales y el acoplamiento que surge de las fuerzas Coriolis y

centrípetas [2]. Por ello es que decenas, posiblemente cientos de artículos y tesis han utilizado el sistema para demostrar las leyes de control lineal y no lineal.

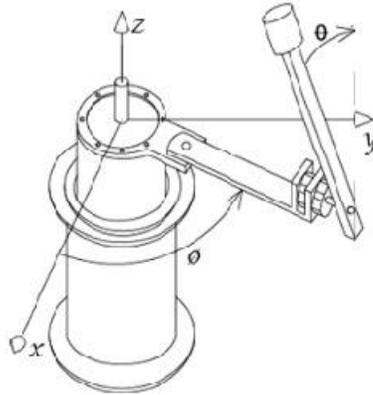


Imagen 2.2: Péndulo de Furuta

Dicha planta consiste en una barra vertical, que tiene en su extremo superior instalado su único actuador, un motor CD y éste a su vez posee una barra horizontal, de manera que el motor pueda girarla en un plano horizontal. Además, esta barra tiene instalado en su extremo un péndulo libre, y por la dinámica del sistema, cuando el motor gira y luego se detiene, insta a que el péndulo se columpie alrededor de su posición natural.

El péndulo cuenta con una posición natural, o estable, la cual se encuentra en dirección de la gravedad. Lo peculiar del sistema es que, así como tiene una posición estable cuenta también con una posición inestable, y ambas se oponen en 180 grados entre sí.

El propósito del péndulo de Furuta consiste en tratar de mantener el péndulo erguido en su posición inestable, o sea que el actuador impulse con pequeños movimientos el brazo, y éste a su vez el péndulo para que se mantenga arriba.

Nótese que para el control de dicha planta su punto de referencia estará ubicado en dirección contraria a la gravedad, no obstante, los errores en la posición que pueda tener el péndulo se dan en radianes y podrán ser tanto positivos como negativos.

### 2.3. Los sensores

Es de esperar que dicha planta debe contar con algún dispositivo o sensor que mida la posición del péndulo. Los hay de varios tipos, desde resistencias variables hasta codificadores ópticos rotacionales, los cuales pueden ser incrementales o absolutos. Las resistencias variables casi no son utilizadas por tratarse de un sensor de contacto, que por su naturaleza agrega fricciones al sistema, que eventualmente afectan sustancialmente la dinámica de la planta.

Codificadores incrementales son los que no tienen una posición de inicio o cero definidos, su cero es tomado a partir del punto de inicio o bien alguna otra posición por criterio del diseñador, su funcionamiento consiste en contar los pasos y dirección con que se desplaza, y con base en esto se determina su posición. Para saber su dirección de giro, este tipo de codificador utiliza dos pistas perforadas, entre ellas hay un desfase de 90 grados, por lo que para saber su giro o posición se utiliza la lógica que se muestra en la siguiente imagen.

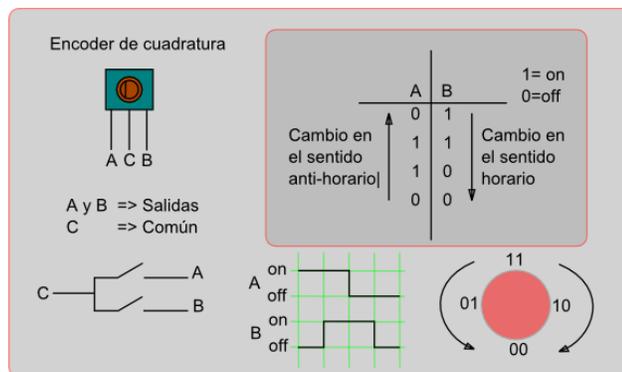


Imagen 2.3: Diagrama de giro de un codificador incremental

Los codificadores ópticos absolutos son los que tienen posiciones fijas, sin importar en qué posición inicien a funcionar siempre tendrá el punto cero en el mismo lugar. Logran saber su posición exacta mediante una lectura binaria óptica que tiene impresa en un disco. Entre mayor cantidad de bits tenga impreso el disco, mayor va a ser la exactitud de su ubicación.

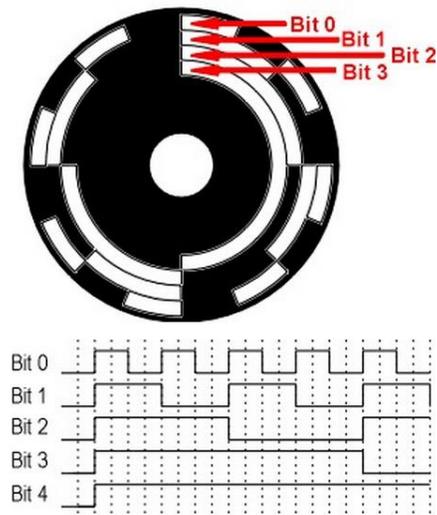


Imagen 2.4: Diagrama de posición de un codificador absoluto

Es muy común ver informes o estudios basados en el Péndulo de Furuta donde le han instalado dos sensores de posición, uno en el péndulo y otro en el giro del motor. Esto porque la lógica de control que ellos emplean lo requiere y, además, por limitaciones mismas del sistema, sus sistemas no pueden poner a actuar el motor más de 360 grados, puesto que a ellos les es indispensable tener conectado mediante cables el sensor que se encuentra en la parte del péndulo.

De lo anterior se deduce que a pesar de que el Péndulo de Furuta ha sido ampliamente estudiado, a lo largo del tiempo ha carecido de modificaciones o mejoras que permitan aplicar

su control con diferentes lógicas. Es de recordar que el presente proyecto no se enfoca solo en el control de la planta, sino en hacer de la misma una herramienta pedagógica versátil, por lo que si es una planta de un solo control sería un gran impedimento.

Recientemente el fabricante Quanser Inc diseñó una planta similar donde solucionaban el problema del cable, porque el motor podía girar 360 grados, utilizando anillos rozantes que permiten la comunicación con el sensor del péndulo, aunque el motor gire más de una revolución. Sin embargo, además de su alto costo, cuenta con una fricción en los anillos, que hasta cierto punto viene a solucionar un problema, pero agrega efectos indeseados en el movimiento de la planta. [3]



*Imagen 2.5: Fricción en anillos rozantes*

## 2.4.El Actuador

El único actuador con que cuenta la planta es el motor DC, de ahí lo importante de escoger un motor adecuado. Recordemos que el péndulo se mantiene en estado controlado gracias al impulso que el motor le ejerce. Por lo tanto, debe ejercer la fuerza suficiente para que logre mover el péndulo con holgura, sin embargo, a mayor fuerza menor velocidad de giro, y la velocidad o tiempo de reacción es muy importante; en resumen, el motor deberá tener buena fuerza sin castigar mucho su velocidad.

### 2.4.1. No linealidad del motor

A la hora de modelar una planta normalmente se llega a un modelo de posición angular versus fuerza, en este caso en particular, se refiere a la fuerza que se necesita que el motor aplique en ese instante en función de la dinámica y posición del péndulo; pero debemos recordar que un motor DC, a pesar que usualmente a la hora de su estudio o uso se parte del hecho que es un dispositivo lineal, consideraciones que son válidas en una zona determinada, sin embargo, éste no lo es para todo su rango de operación, por lo que se recomienda hacer un análisis mediante una serie de experimentos al motor que se va a emplear. [4]

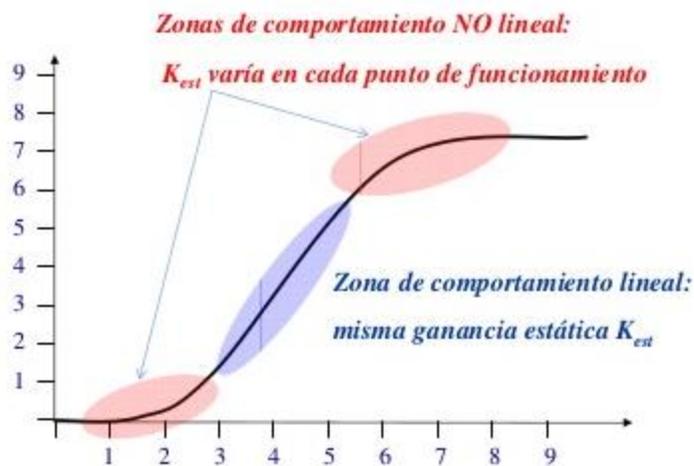


Imagen 2.6: Comportamiento de un motor real [4]

Si se considera un motor real se podrá demostrar que si bien es cierto posee una zona que se puede considerar como lineal, también cuenta con dos zonas claramente no lineales. La zona de arranque es la primera zona no lineal donde el motor perfectamente puede no estar ejerciendo fuerza a pesar que la solicitud de su controlador sí lo es. Esto se debe al fenómeno

de arranque del motor, principalmente a la fricción estática e inercia del dispositivo que éste tenga conectado.

En el caso del Péndulo de Furuta, esta zona debe ser considerada puesto que se trata de una planta que no necesariamente tiene su actuador en funcionamiento constante, es más, un control bien ajustado y una planta bien modelada normalmente cuenta con solicitudes de acción muy reducidas e interrumpidas por lapsos considerables.

La otra zona de conducta no lineal es su etapa superior de operación, al tratarse de un dispositivo real cuenta con límites de operación o saturación, si se le llegase a solicitar fuerzas superiores a las de sus límites para los cuales fue diseñado, su respuesta no va ser superior a la máxima posible, o bien, incluso, se pudiese dar un fenómeno destructivo.

#### 2.4.2. No simetría del motor

Otro aspecto muy importante a la hora de trabajar con motores como actuador de un diseño es la asimetría del mismo, o sea, la diferencia de fuerza generada por el motor si se le suministra una misma tensión en ambos sentidos de giro. Esto se debe a aspectos propiamente de la composición y fabricación del mismo, por ejemplo, la ubicación exacta de las bobinas inductoras, desgaste sufrido en el entrehierro, cantidad desigual de cobre en las bobinas del inducido o hasta la forma no heterogénea como fue enrollado.

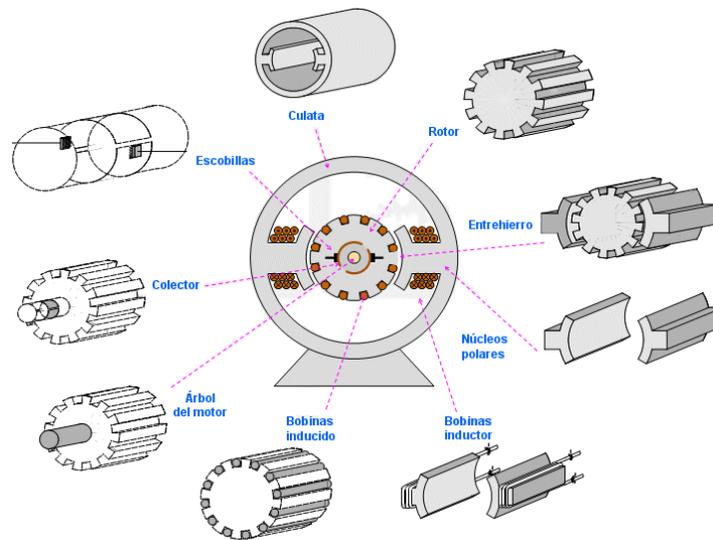


Imagen 2.7: Partes del motor DC

La asimetría en un motor en ocasiones también es generada por una asimetría en el controlador del motor, por diferencias en sus circuitos, produciendo que la potencia que se le está suministrando al motor sea diferente en un sentido o en otro. En un controlador, el camino que toma la corriente cuando el motor gira en un sentido es diferente si gira hacia el otro lado, no obstante, en estos segmentos puede haber diferencias implícitas.

A pesar de que se podría analizar la asimetría por segmentos, de manera que se determine la asimetría del controlador independiente de la asimetría del motor, para fines prácticos en aplicaciones como este proyecto, bien se puede consolidar el dato como uno solo, puesto que a la hora de ejecutar el control la señal es única y nada más tendría que ser compensado un valor. Sin embargo, sí es importante que el futuro ingeniero tenga claro las posibles fuentes de asimetría en el giro del motor, para que esté preparado a enfrentar diferentes diseños de plantas en su vida profesional.

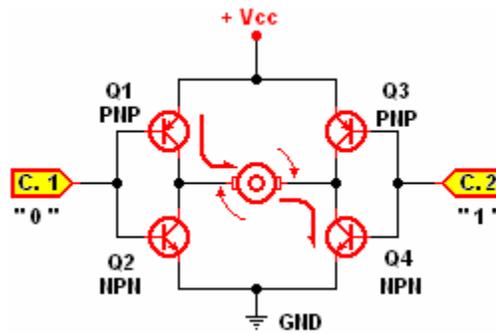


Imagen 2.8: Composición básica del controlador de giro bidireccional de un motor (puente H)

## 2.5. Función de transferencia y variables de estado

El objetivo deseado de la teoría de control consiste en definir un método de control adecuado para la planta en cuestión. Como resultado de ello resulta muy común analizar el componente o lógica de control en forma gráfica, mediante el diagrama de bloques del sistema de control. Pero es importante notar que para ello es necesario contar con un modelo válido de la planta, por lo que normalmente se utilizan dos métodos para resolver el problema de obtener un modelo de la planta, el primero sería mediante variables de estado y el segundo sería mediante función de transferencia.

De manera general, para un sistema lineal con varias entradas y salidas, su representación matricial sería de la siguiente forma:

$$x^*(t) = A(t) * x(t) + B(t) * u(t) \quad (2.1)$$

$$y(t) = C(t) * x(t) + D(t) * u(t) \quad (2.2)$$

En cuanto a las funciones de transferencia, normalmente se expresa así:

$$H(s) = \frac{Y(s)}{X(s)} \quad (2.3)$$

Es importante recalcar que ambos métodos se pueden emplear en sistemas que operen y sean controlados para tiempo continuo, o sea, sistemas que su salida o control puede ser distinto para cualquier tiempo. En el caso de la función de transferencia por simplicidad de cálculos se utiliza la transformada de Laplace [5], la cual de manera generalizada se resume en la siguiente ecuación:

$$H(s) = L\{h(t)\} = \int_0^{\infty} e^{-st} h(t) dt \quad (2.4)$$

Estos métodos también pueden ser expresados para sistemas discretos, o sea, sistemas donde por un intervalo de tiempo su valor de salida o control mantendrá un valor constante, denominado periodo de control. Las variables de estado en forma discreta se expresan de la siguiente manera:

$$x(k+1) = Ax(k) + Bu(k) \quad (2.5)$$

$$y(k) = Cx(k) + Du(k) \quad (2.6)$$

Donde “k” es el periodo de muestreo que se esté analizando y es de duración T segundos. En el caso de la función de transferencia, en esta oportunidad se emplea la transformada Z,

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.7)$$

Donde se es válido mediante la propiedad de transformada Z siguiente:

$$X(z) = Z\{x[n]\} = \sum_{n=0}^{\infty} x[n] z^{-n} \quad (2.8)$$

Ambos métodos de expresión pueden ser correlacionados o transformados entre sí, para lo cual existen varios métodos matemáticos de traslación, entre los que podemos enumerar el método de retenedor de orden cero “zoh”, el método de aproximación triangular “foh”, el método trapezoidal o de “Tustin”, el método de cancelación polo/cero “matched”, entre otros. En la actualidad, el estudiante cuenta con herramientas que facilitan el cálculo de los métodos antes numerados, como lo son las funciones c2d (*continuos to discrete*) y d2c (*discrete to continuos*) de Matlab [6].

A pesar de que en el modelado de una planta se desea la mayor precisión posible, en ocasiones no es tan conveniente porque a menudo conduciría a complicados cálculos y engorrosos modelos matemáticos, cuando bien se puede emplear alguna serie de consideraciones o aproximaciones analíticas, que, aunque no arrojen como resultado exactamente el mismo valor del modelado original, el costo-beneficio de emplear esta modalidad es superior. Por lo tanto, se debe ser cuidadoso a la hora de definir y respetar los parámetros válidos de dicha aproximación.

### 3. Diseño y desarrollo de la solución

#### 3.1. Estructura de la planta

Parte indispensable para el buen funcionamiento de un sistema de control es su ensamblaje del chasis, para una buena estructura son importantes los siguientes elementos: materiales, pesos, resistencia, fricciones, entre otros. Para el chasis se tomó en cuenta, principalmente, tres materiales que estaban a nuestra disposición: acrílico, hierro y aluminio.

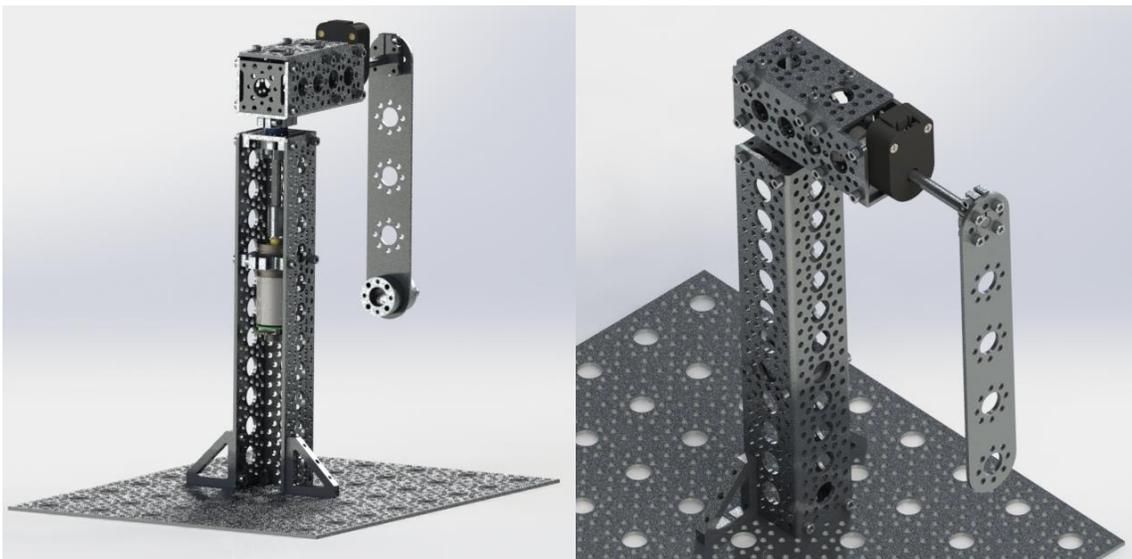
Al tratarse de un proyecto que va a tener mucha manipulación por parte de los estudiantes, el acrílico fue descartado por ser un material frágil que no es muy resistente al uso pesado. El hierro, a pesar de que es muy resistente, desde el punto de vista de la dinámica de la planta no era conveniente porque iba a aportar mucho peso a la estructura giratoria y al péndulo, resultando un sistema un poco difícil de controlar, lo cual no es la idea; además de que se oxida.

El aluminio resultó ser la mejor opción, porque no aportaba tanto peso y su resistencia es muy buena para dicha aplicación, además se contaba con un proveedor que cuenta con gran variedad de piezas compatibles entre ellas y listas para armar. Los precios del proveedor (ServoCity.com) eran razonables e incluso en la mayoría de los casos inferior que si se pensara en producir la pieza, y con la gran ventaja de que la pieza se encontraba disponible para despachar, lo que eliminaba tiempos de producción. Además, por tratarse de piezas de aluminio cuentan con la ventaja de no oxidarse a lo largo del tiempo, lo que ayudaría a la conservación del objeto diseñado.



*Imagen 3.1: Ejemplo de pieza disponibles en ServoCity*

Otra ventaja de las piezas de ServoCity.com es que su diseño con perforaciones circulares ayuda a que se reduzca el peso de las piezas en comparación si estuviera toda sólida, sin que se sobreentienda que esta perforación afectase la rigidez y resistencia de la planta. En cuanto a las piezas giratorias se tuvo el cuidado de que en todas las juntas se utilizara rodamientos de bajo coeficiente de fricción.



*Imagen 3.2: Péndulo de Furuta Diseñado*

### 3.2. Innovación en el diseño de la planta

En la investigación preliminar de diseños existentes del Péndulo de Furuta se encontró de manera generalizada el mismo diseño. A pesar de que los informes y estudios de esta planta abundan, todos contaban con las mismas limitaciones:

- El brazo no gira más de 360 grados.
- Utilizan solo LQR como lógica de control.
- Utilizan cable para comunicarse con el sensor, el cual se enreda en la estructura.

Puesto que se pretendía crear un sistema didáctico versátil y que careciese de limitaciones, se ideó en cómo resolver estos problemas y se llegó a visualizar la necesidad de que la nueva planta no debía tener cables que condujeran hacia la parte superior, que básicamente era lo que impedía su movimiento libre, lo que llevó a la interrogante de cómo llevar la alimentación al sensor de ángulo ubicado arriba y como transmitir sus datos al controlador ubicado en la base.

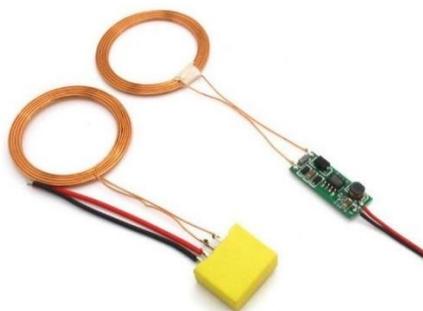
Se concluyó que este problema debía ser resuelto mediante tecnologías inalámbricas, en el caso de la transmisión de datos las opciones eran más variadas, sin embargo, se optó por utilizar dispositivos Bluetooth, por tratarse de una tecnología bastante documentada y por existir tarjetas de fácil uso y económicas en donde el protocolo inalámbrico no iba a generar mayor dificultad, porque esta comunicación se iba a tratar como un típico enlace serie. Se utilizó una tarjeta HC-05 que tiene la propiedad de desempeñar un rol de maestro en la comunicación y también una tarjeta HC-06 como esclavo.



*Imagen 3.3: Tarjeta Bluetooth - serie utilizado en SiDiCo (HC-05)*

En cuanto a la alimentación de los circuitos aislados se analizaron tres opciones, el uso de baterías, panel solar y transmitir la potencia a través de electromagnetismo. Las dos primeras opciones le iban a agregar al proyecto restricciones, entonces se decidió hacer las pruebas de funcionamiento al módulo de carga inalámbrico proporcionado por un distribuidor local. En sus especificaciones el fabricante define una tensión de entrada de 12 voltios y que puede suministrar 5 voltios de salida [7], exactamente lo necesario en el proyecto.

Su forma de diseño circular se prestaba para ser instalado en la base de giro, y se verificó que se mantengan “enganchados” aun cuando el brazo se encuentra en movimiento. La distancia máxima de transmisión que garantizaba el fabricante era de hasta 20mm, sin embargo, para evitar problemas, se logró instalar a una distancia de 5mm, en fin, su funcionamiento fue satisfactorio en todas las pruebas realizadas.



*Imagen 3.4: Módulo inalámbrico de acople de energía*

Ahora bien, mediante las soluciones anteriores se dio por resuelto la transmisión de datos y energía, sin embargo, esta independización generaba una nueva necesidad, instalar un componente que se encargara de procesar los datos del codificador incremental y transmitirlo en comunicación serie al HC-06 instalado en el brazo. Debía de consumir poca potencia, ser liviano y de reducido tamaño, ya que por dinámica del sistema no es muy conveniente que la masa del brazo supere la del péndulo.

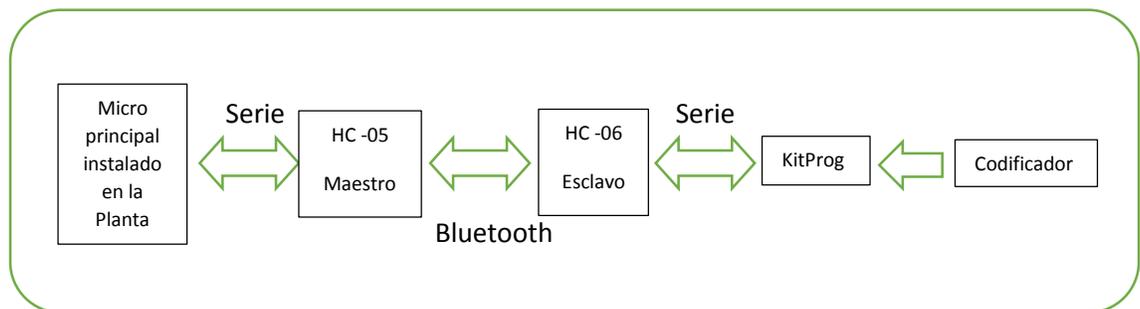


Imagen 3.5: Diagrama de comunicación de SiDiCo

Se ideó una solución muy ingeniosa, que era modificar la programación del programador del microcontrolador PSoC-059LP para que funcionara con una unidad de desarrollo. Es importante mencionar que, por un tema de secreto comercial, no existía información a la mano de los procedimientos para programar este dispositivo, por tanto, se tuvo que recurrir a investigar en fuentes diversas en la internet para poder solucionar el problema de la falta de información directa del fabricante.



Imagen 3.6: KitProg y Chip de desarrollo PSoC-059LP utilizado en SiDiCo

### 3.3.El programa para la interfaz de usuario

En cuanto a la plataforma de programación, se seleccionó el programa Matlab de Mathworks Inc. para desarrollar la interfaz de usuario, esto mediante las funciones del módulo “guide” por su versatilidad y alta compatibilidad con el resto de las funciones del programa principal, sean matemáticas, de comunicación o representación gráfica, tres de las características primordiales que debía tener la aplicación por utilizar.

### 3.4.El microprocesador

La planta debería poseer un microprocesador capaz de tener al menos un puerto de comunicación hacia el computador, y por versatilidad y tendencia del mercado en preferencia del tipo USB, puertos ADC, decodificadores de señales en cuadratura y una suficiente cantidad de puertos de uso general para poder abastecer las necesidades de sensores y actuadores. Otro requerimiento muy crítico es que el sistema debía de contar con un tiempo de procesamiento muy breve, inferior a 1 milisegundo por lo cual el procesador debía ser muy rápido.

Teniendo claro las exigencias de este dispositivo tan esencial para la planta se decidió seleccionar un PSoC de Cypress, más específicamente uno de la familia 5LP, principalmente por cumplir bastante con los requerimientos básicos citados, y sin dejar de lado que cuentan con el modelo CY8CKIT-059 que es de bajo costo [8].

Además, otras de las ventajas de este dispositivo es que el *software* para programarlo es gratuito, lo proporciona el mismo fabricante, y no requiere de *hardware* adicional para ser programado.

Para tomar la adecuada decisión se elaboró un listado con los diferentes puertos de uso general que iban a ser necesarios, los cuales se resumen en la siguiente tabla

*Tabla 3.1: Listado de GPIO necesarios.*

<b>Nombre del Puerto</b>	<b>Descripción de operación</b>
Tx Bluetooth	Serie
Rx Bluetooth	Serie
Tx PC	USB-UART
Rx PC	USB-UART
PinA encoder motor	decodificador de cuadratura
PinB encoder motor	decodificador de cuadratura
Generador Motor	PWM
PinA giro Motor	salida al IN 1 del driver
PinB giro Motor	salida al IN 2 del driver
I motor corriente	ADC
Motor enable	salida al En del driver

### 3.5.Los sensores

En la actualidad, una manera muy empleada para medir ángulos es mediante codificadores rotacionales, hay análogos o digitales, incrementales o absolutos, e incluso, en variedad de cuentas por revolución, tantas combinaciones como el diseñador, el presupuesto y los requisitos del proyecto lo definan.

Se ideó instalarle a la planta dos codificadores incrementales rotacionales, uno, magnético instalado en el eje del motor y el segundo, óptico, instalado en el eje del péndulo, de manera que se pueda medir la posición y velocidad de giro, tanto del péndulo como del actuador.

El mercado actual ofrece una serie de codificadores con gran variedad de características, incluso algunos integrados al motor, lo que permite una mayor simplicidad a la hora de su instalación, sin que signifique que se ha castigado la calidad de la medición. Por lo tanto, se ha decidido seleccionar uno de esta categoría que cuente con una resolución mayor a las dos mil cuentas por revolución. Resolución que se considera más que adecuada para medir la posición y velocidad del motor.



Imagen 3.7: Motor con codificador magnético incremental utilizado en SiDiCo

En cuanto al movimiento del péndulo se decidió medirlo con ayuda de un sensor de cuatro mil cuentas por revolución del fabricante USDigital, modelo E5, el cual cumple a cabalidad con las características delimitadas para el diseño e incluso cualidades de dimensiones y eléctricas [9]. A futuro éste mismo codificador permitiría aumentar su resolución a dieciséis mil cuentas por revolución mediante la configuración del decodificador en modo 4x, en caso de que se determine necesaria una mayor resolución.

A pesar de que dicho codificador cuenta con una salida de índice (*index*), se determinó que para este proyecto no se emplearía y se optó por una calibración del punto cero mediante *software* al inicio de la planta, en lugar de calibrar de manera manual que el *index* coincidiera

con el cero real, lo que sería una acción de suma dificultad puesto que debería tener una precisión de  $360 / 4000$ .

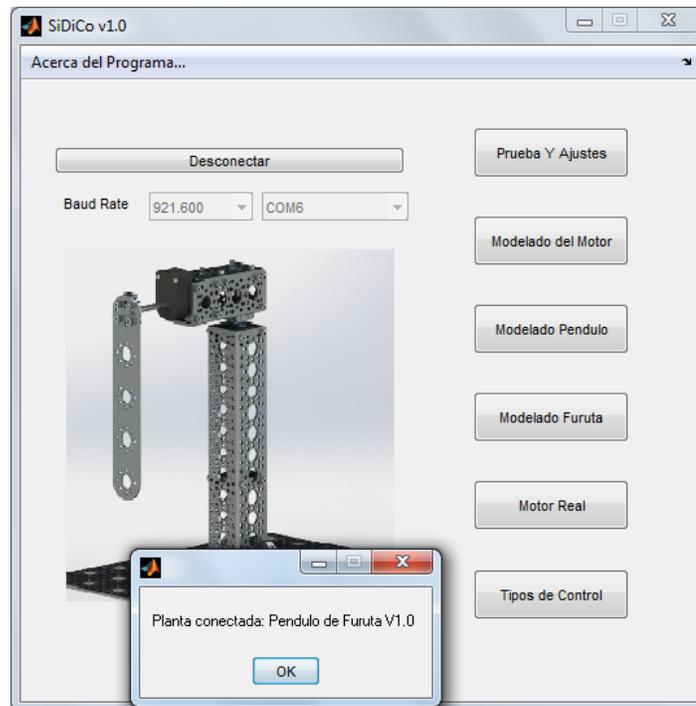


*Imagen 3.8: Codificador E5 utilizado en SiDiCo*

### 3.6.La aplicación SiDiCo

SiDiCo es la aplicación propia de esta Tesis, creada desde cero con las herramientas ya mencionadas. Al ejecutar SiDiCo en la línea de instrucciones de Matlab se abre el módulo principal de la aplicación creada, donde una vez que se ha verificado la conexión con la planta le permite al usuario ingresar a diferentes módulos didácticos, ordenados acorde al tema en estudio. Estos temas se pueden enunciar como Pruebas y Ajustes, Modelado del Motor, Modelado del Péndulo, Modelado de Furuta, Motor Real y Tipos de Control.

Es importante mencionar, que en la interfaz de conexión del programa se dejó prevista para que a futuro el mismo programa pueda ser utilizado con otras plantas diseñadas, siempre y cuando respeten la comunicación que se ha desarrollado.



*Imagen 3.9: Pantalla principal de SiDiCo*

En el apartado Pruebas y Ajustes, el estudiante podrá hacer pequeñas pruebas mediante el botón “Test” para verificar el adecuado funcionamiento de la planta, así mismo permite realizar algunas funciones donde se familiarice el usuario con la máquina. Estas funciones lo que prueban es el comportamiento adecuado del motor en ambos giros, hace una prueba de peso del péndulo para verificar si le es posible a la planta levantar el peso instalado.

En su extremo inferior izquierdo, el módulo de Pruebas cuenta con un par de funciones donde prueba el tiempo de comunicación entre la planta y el computador, además donde puede experimentar la transmisión en línea de la posición del péndulo. Mediante estas pruebas el estudiante podrá luego ajustar los parámetros propios de la comunicación y representación gráfica, parámetros que son inherentes a las exigencias del experimento, a la planta y al computador donde se esté ejecutando el programa.

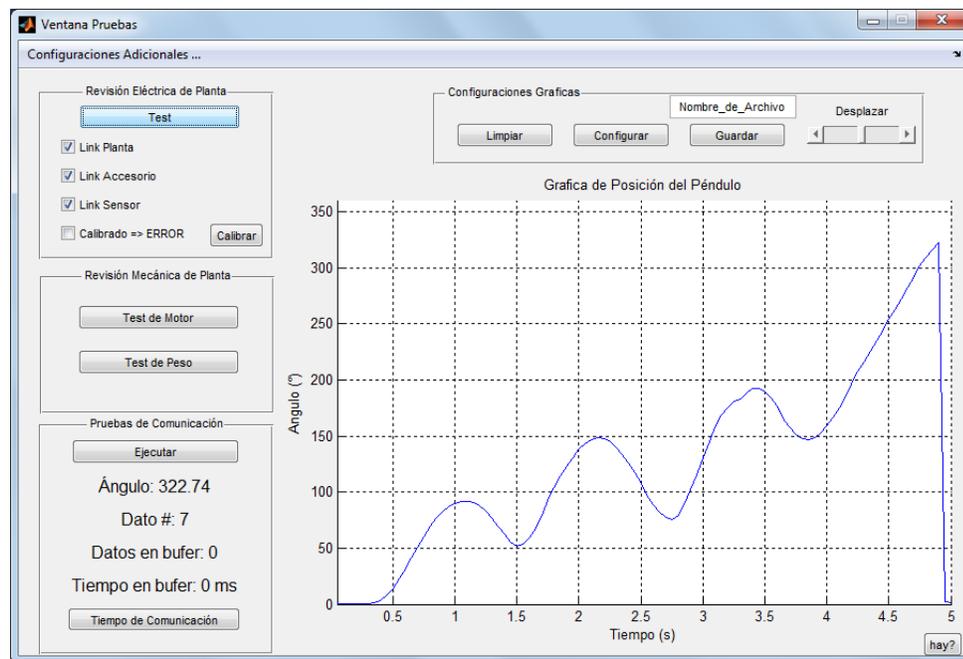


Imagen 3.10: Pruebas y Ajustes de SiDiCo

Estos ajustes se pueden hacer en la ventana de configuración, que se encuentra en la esquina superior izquierda, también está disponible en el resto de los módulos que requiera de comunicación y representación gráfica. Los parámetros que el sistema puede configurar son:

- Histéresis: es el rango de no operación del motor por aproximarse al punto de equilibrio.
- Periodo: es el periodo de muestreo con que el control va a operar.
- Ángulo de Op: es el ángulo con que el control va a estar operando, si llegara a ubicarse fuera de estos límites el control se apaga.
- Muestras en X: es la cantidad de datos que el gráfico va a contener en X.
- Actualiza: es el diezmado entre el control y la representación gráfica, sirve para mejorar la velocidad de trazo del gráfico.
- Máximo en Y: es el dato máximo mostrado en el eje Y, normalmente sirve reducirlo para mejorar la resolución del área en estudio
- Tensión Fricción: sirve para compensar la no linealidad del motor por fricción estática.

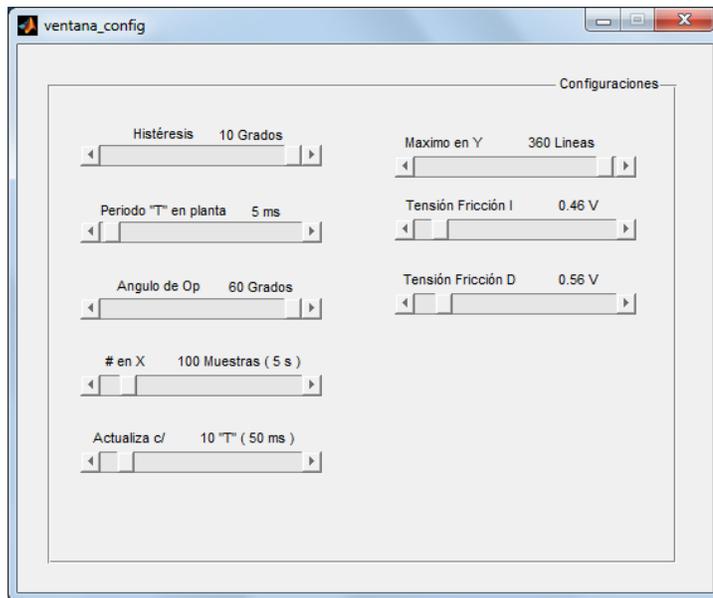


Imagen 3.11: Ventana de configuraciones de SiDiCo

En el segundo módulo del programa principal se encuentra “Modelado del Motor”, es una pantalla en la que el estudiante podrá realizar el proyecto corto número uno de laboratorio de control, proyecto que consiste en estimular el motor por cortos lapsos para ver su respuesta de velocidad y de corriente en función del estímulo de la entrada y del tiempo. Cabe recordar la importancia de comprender, por parte del estudiante, el funcionamiento del motor real, porque la mayoría de sistemas de control cuentan con motores como actuadores.

El módulo de modelado del motor, además de las opciones de graficar y guardar las imágenes del experimento, cuenta con funciones para guardar datos en forma de tablas, que serán necesarios para encontrar el modelado empírico de la planta mediante herramientas como **ident** de Matlab.

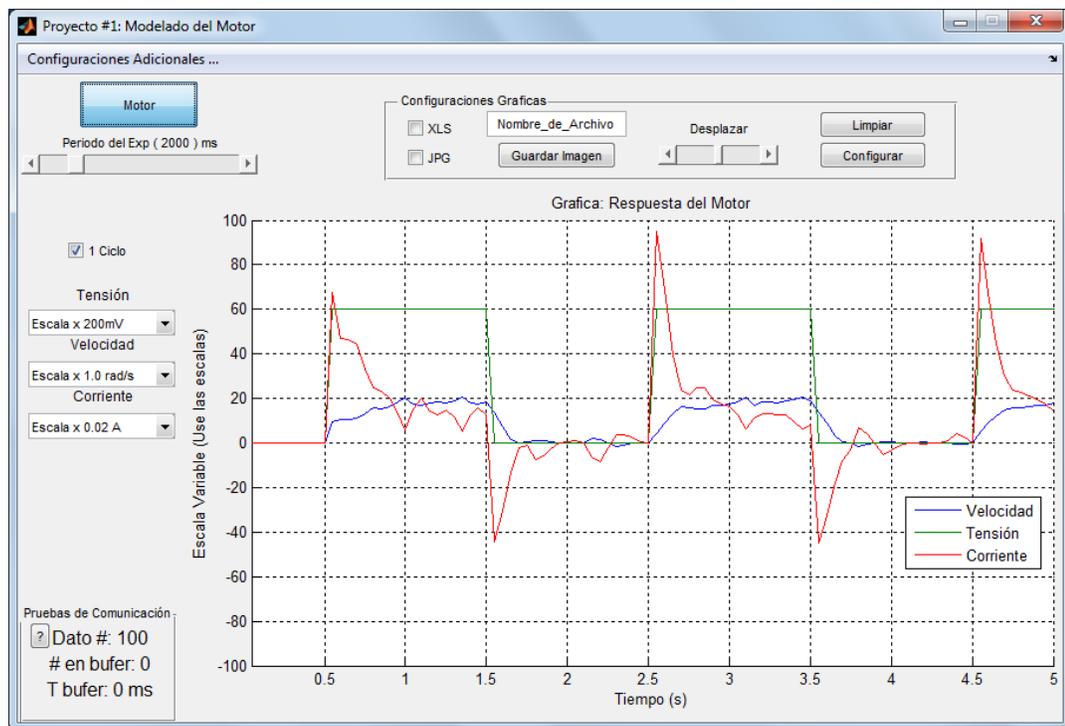


Imagen 3.12: Modelado del motor de SiDiCo

En el siguiente apartado de SiDiCo, permite modelar el péndulo y así el estudiante podrá observar en el área del gráfico el comportamiento de un péndulo sub-amortiguado. Como sistema didáctico es de bastante provecho para el estudiante el tener la oportunidad de modelar y ensayar con diferentes tipos de plantas, y por tratarse de un sistema integrado diseñado para tal fin, se puede destacar la ventaja de que cuente con todas ellas en el mismo sistema.

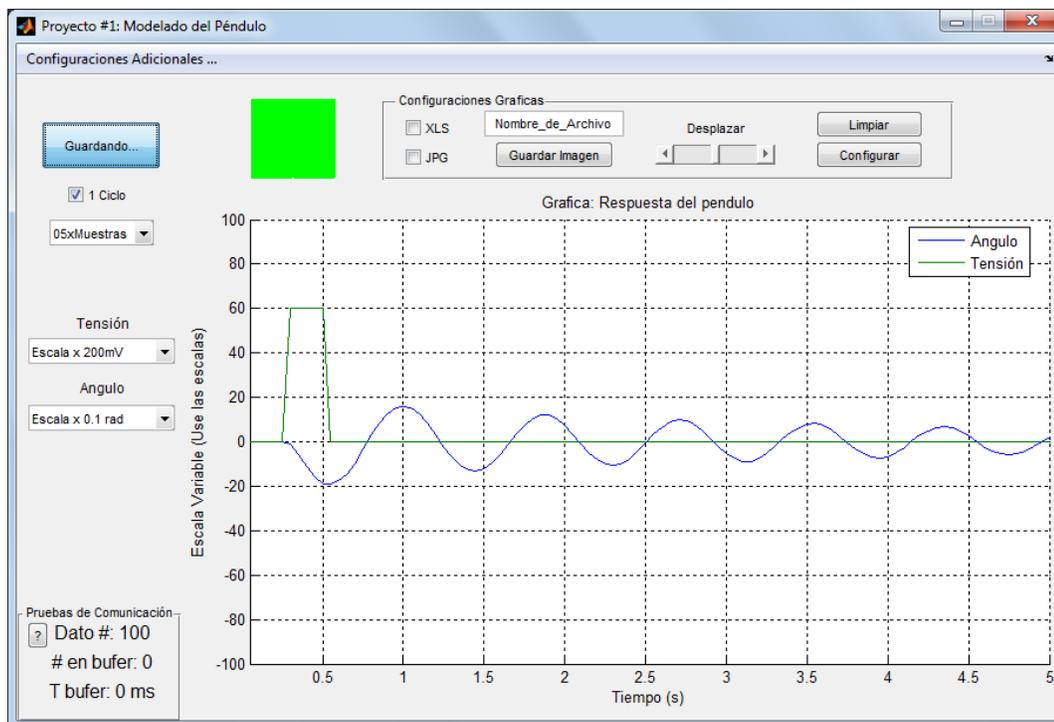


Imagen 3.13: Modelado del Péndulo de SiDiCo

Como tercer ejercicio de modelado SiDiCo está el módulo de “Modelado Furuta”, en el cual el estudiante podrá realizar el modelado completo de la planta tal y como si se comportara como un péndulo. En éste ejercicio el estudiante podrá observar la respuesta y acción del desplazamiento del motor y las oscilaciones del péndulo en el mismo plano.

Es de recordar que el Péndulo de Furuta es una planta totalmente inestable, por lo que diseñar un experimento que capture datos para que luego los mismos sean identificados en un modelo específico sería muy inexacto, porque para realizar este experimento se necesitaría de la intervención humana, por lo tanto, para una adecuada comprensión de esta planta se debe hacer mediante ejercicios auxiliares en plantas modificadas. El ejercicio más concluyente sería analizar el péndulo de Furuta, pero, en su posición natural, algo parecido a la grúa pórtica, pero con un desplazamiento circular.

Todos los módulos de modelado contienen la opción de seleccionar la extensión del ejercicio, mediante la opción “1 Ciclo”, con la cual, si está seleccionado, el ejercicio tarda el tiempo que se ha definido para una pantalla de muestreo; caso contrario, el estudiante cuenta con la posibilidad de hacer los experimentos más extensos y minuciosos, y su duración será indefinida, por lo que deberá detener el experimento cuando desee.

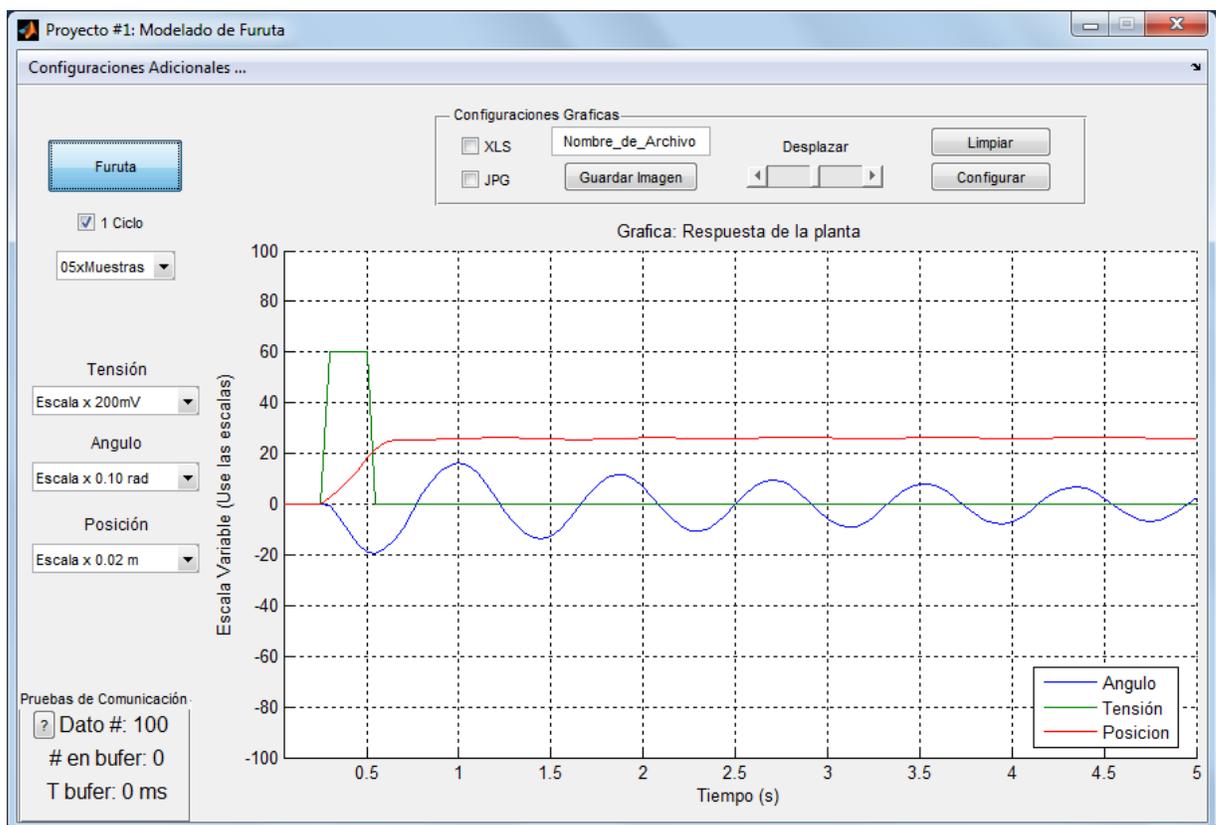
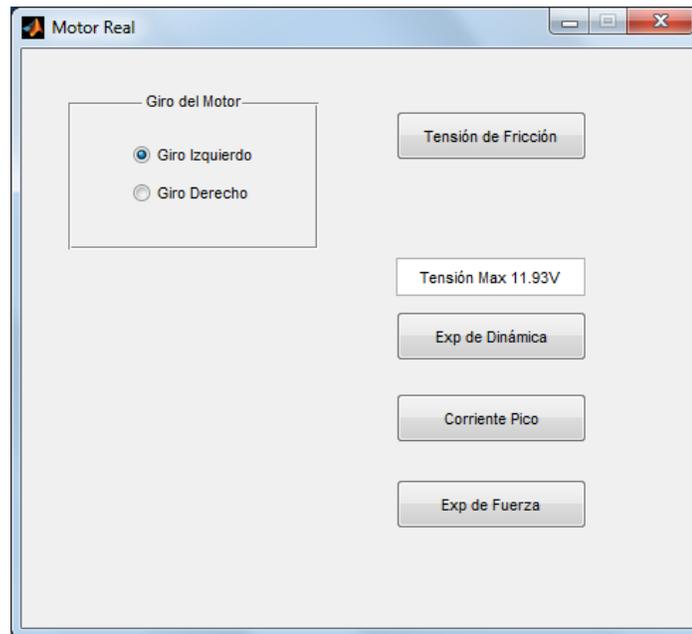


Imagen 3.14: Modelado completo de SiDiCo

El módulo principal también tiene espacio un apartado que se le denominó “Motor Real”. Sirve para brindarle al estudiante una serie de experimentos donde prueba el motor, desde una perspectiva de un dispositivo real, que le proporcionan al estudiante herramientas que le ayudan

a demostrar que el motor real tiene zonas de no linealidad e incluso en muchas ocasiones asimetría.



*Imagen 3.15: Modulo de Motor Real en SiDiCo*

En los objetivos iniciales del presente proyecto, éste módulo no había sido considerado, sin embargo, a lo largo del desarrollo del mismo se vio la necesidad de estudiar más detalladamente al motor, puesto que algunos algoritmos de control no operaban como se esperaba, y mediante estos ejercicios se determinó que era porque en todo momento se estaba considerando al actuador como un dispositivo lineal y simétrico, pero en realidad su zona de funcionamiento lineal no era todo su rango de operación.

Después de realizar los experimentos propuestos, se encontró una asimetría bastante considerable, y se demuestra que el motor en estudio posee una asimetría nada despreciable y que debería ser considerada a la hora de implementar el control, por ejemplo, el valor máximo

de fuerza que logra desarrollar el motor girando hacia la derecha es de 3.054 Newton y en el caso de que su giro sea hacia la izquierda es de solo 1.852 Newton, un 40% menos.

*Tabla 3.2: Mediciones de fuerza en el motor CD, Valores en Newton*

Medición	Izquierdo [N]	Derecho [N]
1	2.02	2.98
2	1.98	3.26
3	1.84	3.14
4	1.98	3.2
5	1.94	2.56
6	1.88	3.26
7	1.76	3.1
8	1.7	3.08
9	1.72	3.02
10	1.7	2.94
Promedio	1.852	3.054
Constante	4.21166307	2.5540275

Esta asimetría, si no es considerada a la hora de implementar el control, ocasionaría que la planta le devengue más esfuerzo controlar perturbaciones que provengan desde la derecha en comparación con los provenientes de la izquierda. Esfuerzo que en el análisis del control se interpretarían como vibraciones, más potencia, oscilaciones, sobre-impulsos y un mayor tiempo de estabilización.

En el caso de la no linealidad se le hizo un experimento de arranque al motor, donde se logra demostrar que es imposible que un motor real logre generar fuerza o movimiento ante cualquier tensión aplicada. En el experimento que se hizo se le empieza a suministrar tensiones al motor desde cero y en incremento, hasta el punto que el motor sí genere movimiento. En la siguiente tabla se logra evidenciar cómo la tensión es bastante superior a la tensión que requeriría si fuese lineal. Incluso se vuelve a evidenciar un tema de asimetría, puesto que la

tensión de arranque hacia lado izquierdo es de 3.65 voltios y hacia lado derecho es de 3.47 voltios

*Tabla 3.3: Mediciones de tensión mínima que requiere el motor para generar movimiento*

Medición	Izquierdo [V]	Derecho [V]
1	3.728	3.611
2	3.669	3.351
3	3.679	3.419
4	3.497	3.445
5	3.517	3.513
6	3.796	3.555
7	3.809	3.367
8	3.614	3.497
9	3.679	3.62
10	3.517	3.37
Promedio	3.6505	3.4748

Es de suma importancia que el diseñador tenga presente aspectos como estos a la hora de crear una planta, por ejemplo, si por criterio de diseño decidiera alimentar este motor con una tensión máxima de 5 voltios, su rango de operación disponible sería muy pequeño, lo que limitaría bastante la respuesta del control, lo cual se consideraría un criterio errado. He ahí la importancia de que el estudiante comprenda las limitaciones de los actuadores físicos, más uno tan empleado como lo es el motor DC.

El programa diseñado cuenta con un módulo de control, donde el usuario puede seleccionar el método de control deseado, y así mismo, los parámetros necesarios para la ejecución del mismo. Una vez en ejecución el control, el usuario podrá visualizar una serie de parámetros desplegados en pantalla en forma gráfica. Los datos visualizados son el ángulo del péndulo, el error en la posición y la acción del control.

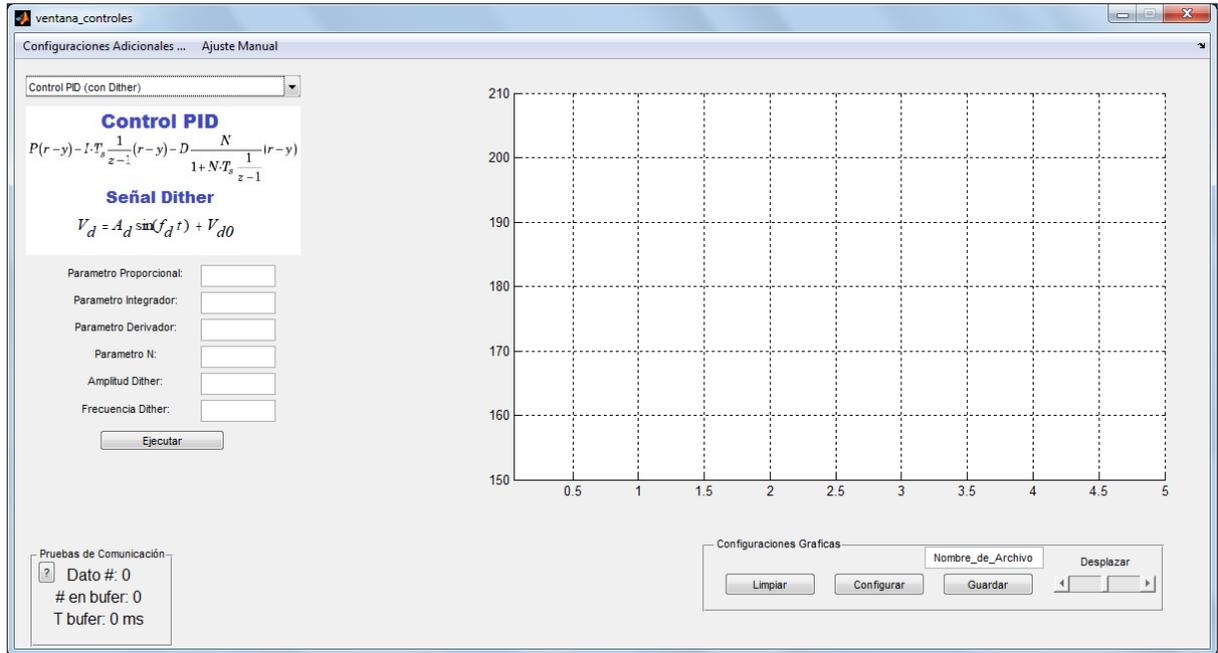


Imagen 3.16: Modulo de control del programa SiDiCo

Desde el punto de vista pedagógico, es de gran utilidad poderle mostrar al estudiante cómo ante un eventual error en la posición, el control ejecuta una señal correctiva y cómo esta puede variar acorde con la dinámica del péndulo. Además, con la finalidad de que el estudiante pueda documentar las imágenes adquiridas en su experimento, cuenta con funciones de captura de datos, las cuales probablemente aporte en su informe.

Actualmente SiDiCo tiene dos lógicas de control implementadas, PID y LQR. En el caso de PID, se desarrolló para demostrar que el péndulo de Furuta era posible estabilizarlo con la lectura de tan solo el sensor de posición angular del péndulo y mediante este algoritmo, puesto que la información preliminar que se encontró de dicha planta involucraba siempre la lógica LQR que utiliza dos sensores.

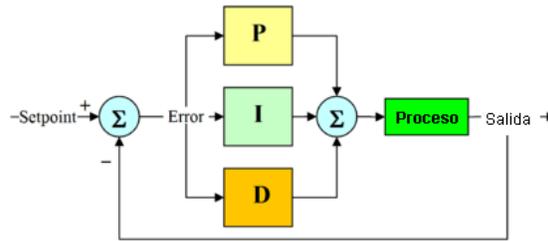


Imagen 3.17: Diagrama de bloques del control PID implementado

Como el control PID cuenta con varias configuraciones, se aclara que se utilizaron los tres bloques en paralelo, tanto el bloque integrador como el derivador utilizaron la aproximación *Forward Euler*. Para su ajuste en esta oportunidad se empleó Simulink de Matlab, principalmente por simplicidad de la herramienta de ajuste automático, sin embargo, no se descarta que el estudiante pueda utilizar otras herramientas de su preferencia, siempre y cuando respete las configuraciones de los bloques en su lógica de cálculo.

$$P(br - y) + IT_s \frac{1}{z-1} (r - y) + D \frac{N}{1 + NT_s \frac{1}{z-1}} (cr - y)$$

Imagen 3.18: Ecuación del PID implementado

En la ecuación anterior se puede observar la representación matemática general implementada en el control PID, nótese las dos constantes “b” y “c” afectando la referencia “r”, sin embargo, las mismas pueden obviarse, puesto que en nuestros cálculos e implementaciones en todo momento fueron 1, en el caso de la referencia siempre fue cero y el valor de la salida “y” estuvo dada en radianes.

$$\frac{D(z)}{E(z)} = \frac{K_d N}{1 + \frac{NT_s}{(z-1)}} = \frac{(z-1)K_d N}{(z-1) + NT_s} = \frac{(1-z^{-1})K_d N}{1 + z^{-1}(T_s N - 1)} \quad (3.1)$$

$$D(z) = \frac{(1-z^{-1})K_d N}{1+z^{-1}(T_s N-1)} E(z) = E(z)(1-z^{-1})K_d N - D(z)z^{-1}(T_s N-1) \quad (3.2)$$

$$d(k) = [e(k) - e(k-1)]K_d N - d(k-1)(T_s N-1) \quad (3.3)$$

La variable “N” cumple con el propósito de mitigar los cambios bruscos que intente tener la respuesta derivativa, actuando como un filtro al comparar cada cálculo contra el cálculo derivativo anterior. Además, funciona como filtro del cambio percibido de la señal de error, lo que ayuda a que los cambios de movimiento del péndulo no sean bruscos, e incluso, éste filtro ayuda a reducir el ruido existente en la señal de movimiento

En el caso del control integral su composición es más común, la única diferencia al control integral tradicional es que por haber implementado la aproximación *Forward Euler* los cálculos se hacen basándose en el error del periodo anterior. En la siguiente imagen se muestran las transformaciones necesarias para poder implementar la fórmula del control integral en el dispositivo discreto.

$$\frac{I(z)}{E(z)} = \frac{K_i T_s}{(z-1)} \quad (3.4)$$

$$I(z)(1-z^{-1}) = E(z)z^{-1}K_i T_s \quad (3.5)$$

$$I(z) = E(z)z^{-1}K_i T_s + I(z)z^{-1} \quad (3.6)$$

$$I(k) = e(k-1)K_i T_s + I(k-1) \quad (3.7)$$

El segundo método de control es el LQR, básicamente es un control que se implementa mediante el conocimiento previo de las variables de estado de la planta. Consiste en un doble control PD, uno aplicado al error del péndulo y otro al error de la posición del motor, mismo referenciado a partir de una posición inicial o bien una deseada. Por su complejidad matemática

nuevamente se hizo uso del programa Matlab, porque tiene una función que mediante la matriz de A de variables de estado se encarga de calcular las variables de la matriz K del control, esto en función de la intensidad deseada por el diseñador.

Para ello Matlab necesita los parámetros Q y R, estos valores cargan el peso con que se quiera que el control actúe. Desde el punto de vista teórico todos los valores de Q y R funcionan, sin embargo, depende mucho, también, de la veracidad del cálculo de las ecuaciones de estado y la calidad de las mediciones de sus parámetros. Además, es importante aclarar que eventualmente con algunos valores altos de Q y R se podría estar motivando a valores de esfuerzo inalcanzables, lo que desde el punto de vista de una planta real lo haría no controlable.

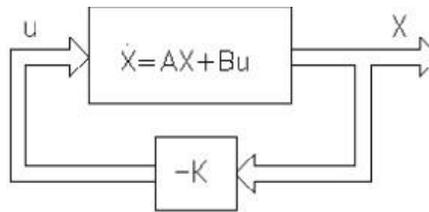


Imagen 3.19: Diagrama de bloques del control LQR implementado

Así mismo, por un mal criterio de selección estas matrices de esfuerzo podrían provocar que los parámetros obtenidos de la matriz K exijan el control a tal punto que la planta haga movimientos muy bruscos, rápidos e innecesarios, y a pesar de que estos valores sí controlan la planta, haría un sobreesfuerzo e incurriría eventualmente en un deterioro o desgaste prematuro.

## 4. Validación y Análisis de Resultados

En este capítulo se mostrarán los resultados obtenidos para las diferentes etapas del diseño y verificación del funcionamiento de la planta, así como su comparación con los resultados teóricos.

Se iniciará por analizar el funcionamiento de la circuitería instalada en la parte superior del brazo, la cual, tal y como se ha explicado, carece de conexión cableada con el resto de los componentes del sistema. A la hora de su instalación se probó primero la etapa de alimentación, ya que sin ella la posibilidad de tener componentes eléctricos independientes no sería factible. (Ver imagen 4.1)

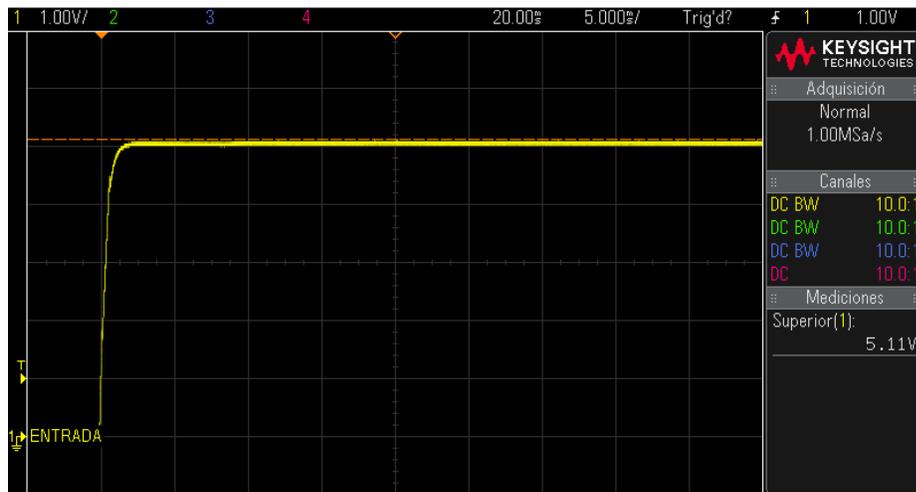


Imagen 4.1: Tensión suministrada por el módulo inalámbrico (sin carga)

Se hicieron mediciones en vacío de esta nueva fuente, y se obtuvo una tensión de 5.11 voltios, seguido de eso era necesario comprobar su capacidad de alimentar todos los componentes, por lo que se procedió a instalarlos y medir nuevamente la tensión que estaba suministrando, siendo éste nuevo valor de 4.99 voltios, por lo que es una tensión apta para que

operen todos los componentes en la parte superior. Así mismo, se hicieron pruebas de operación mediante el módulo de pruebas de SiDiCo y se estuvo monitoreando posibles picos o caídas de tensión las cuales no fueron perceptibles. (Ver imagen 4.2)

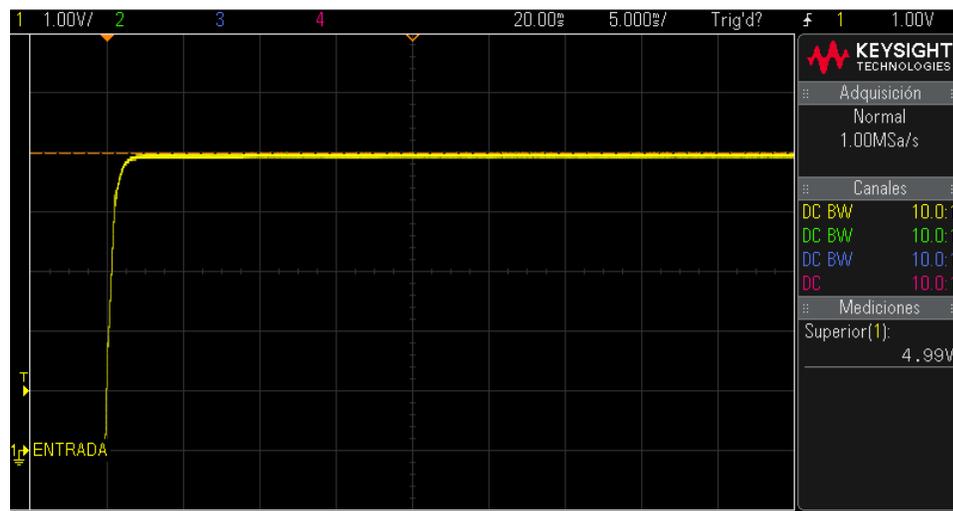


Imagen 4.2: Tensión suministrada por el módulo inalámbrico (con carga)

La transmisión vía Bluetooth debía ser probada desde dos perspectivas muy críticas, la veracidad de los datos y el tiempo de respuesta en transmitir desde el brazo. En cuanto a la veracidad de los datos se debía de hacer una exhaustiva prueba para corroborar que todos los datos transmitidos debían ser recibidos, e incluso igual al transmitido puesto que una inconsistencia en el dato recibido ocasionaría que el control calcule una respuesta errónea, basado en una diferencia de error fallida.

En la imagen 4.3 se puede observar el tipo de bloque contador que se encarga de temporizar la transmisión y su configuración utilizada.

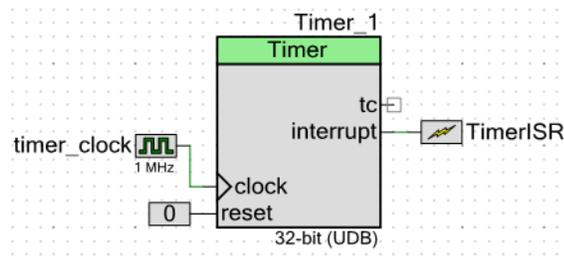


Imagen 4.3: Diagrama de bloques implementado para controlar la periodicidad de transmisión

Para la anterior verificación se hizo una rutina con el propósito de transmitir una serie de números secuenciales dentro de la rutina de una interrupción, la interrupción se hacía operar mediante la conexión de un contador ajustado a 1 milisegundo, tiempo mínimo de operación que se ha definido para la planta. En primera instancia la recepción de los datos se hizo mediante el receptor del Bluetooth de un Smartphone y diferentes aplicaciones como “Bluetooth spp pro” y “BlueTerm”, y en etapas más avanzadas del proyecto mediante la transmisión directa al programa de SiDiCo, y en todo momento se detectó una consistencia en los datos recibidos. (Ver imagen 4.4)

```

CY_ISR(InterruptHandler_Timer)
{
    Timer_1_STATUS;
    angulo ++;
    UART_1_PutChar(angulo); /* envía el dato */
}

```

Imagen 4.4: Rutina de prueba del Bluetooth

Para corroborar que la transmisión se está haciendo con la periodicidad ajustada y el tiempo que dura en transmitir es aceptable se hizo uso del osciloscopio, puesto que se habilitó una salida para que fuese la señal de monitoreo, de manera que cuando la base solicite un ángulo genere una señal en alto, y cuando el dato fuese recibido en el retorno, el pin se pondría en bajo.

En la imagen 4.5 se muestra el resultado obtenido a un periodo ajustado de un milisegundo, y se puede ver que el tiempo real es de 1.0033 milisegundos, por lo que se considera una calibración muy buena y el tiempo que tarda en transmitir y recibir la respuesta es de 87 microsegundos.

En el caso de un diseño de control, es normal y aceptable encontrar máquinas que procesen la lógica de control a un décimo del periodo de muestreo, por lo tanto, el tiempo que está tardando la señal transmitida está dentro del rango válido, e incluso queda un rango de tiempo aceptable para que un eventual control termine de procesar los datos hasta generar la acción de control. Para fines prácticos, se considera una regla que la señal de control sea generada en un tiempo máximo de un décimo del periodo de muestreo.

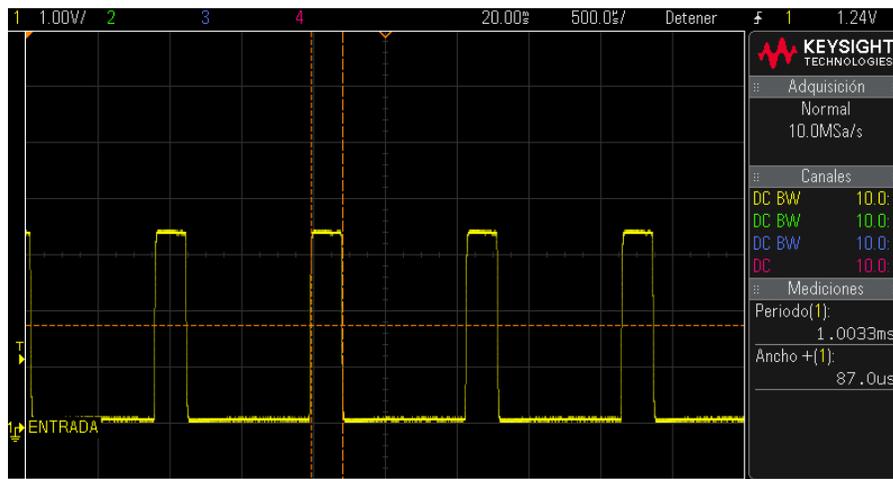


Imagen 4.5: Verificación de tiempos de transmisión entre etapas

Se debe reconocer que si a la máquina se le pide un muestreo de un milisegundo el tiempo resulta un poco ajustado a causa de la transmisión, sin embargo, es suficiente para cumplir con la regla de tiempo máximo, además hay que considerar que, si se le solicita un

muestreo mayor al mencionado, se estaría ante un mejor panorama y dejaría de ser crítico, puesto que el tiempo en generar la señal de control sería en estos casos menor a la regla, debido a que el tiempo de transmisión no está en función del periodo de muestreo, lo cual es bueno puesto que entre menos dure la acción de control en ser generada, más rápido se intentará corregir el error.

Eventualmente, si las especificaciones del sistema se tornaran más críticas o se quisiera reducir el tiempo de comunicación se podría optar por cambiar la lógica de solicitud de la información, de manera que continúe aprovechando las ventajas descritas del mismo *hardware* inalámbrico, pero utilizar otras de las propiedades de PSoC denominado DMA.

El Uso de éstos módulos permite que el microprocesador tenga dos rutinas trabajando en paralelo de manera que exista un ciclo muy veloz y continuo actualizando el dato en memoria de la posición del ángulo del péndulo en el PSoC principal, y que la segunda rutina se encargue de procesar la lógica de control de manera independiente.

La ventaja de utilizar este procedimiento es que cuando el microprocesador principal necesite el dato de posición no tiene que hacer la solicitud y esperar la respuesta, sino que tendrá en memoria un dato disponible y siempre actualizado de manera directa, lo que va a disminuir los tiempos en generar la acción de control.

Otro aspecto muy importante es que la frecuencia con que el microprocesador le transmite datos al dispositivo Bluetooth afecta a la hora de medir los tiempos de transmisión, el caso en particular de la planta elaborada, la velocidad máxima obtenida fue de 230400 baudios por segundo, debido que a frecuencias mayores se perdía la comunicación con el dispositivo inalámbrico.

En la imagen 4.6 se observa la conexión del módulo UART encargado de la comunicación mencionada.

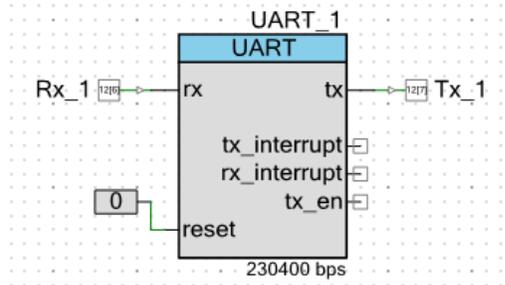


Imagen 4.6: Componente UART necesario para comunicarse con el Bluetooth

Para que la comunicación esté equilibrada en la base de la planta existe un módulo de comunicación igual al del transmisor y con las mismas configuraciones, operando como receptor, a pesar que se pudo optar por configuraciones diferentes, esta decisión ayudó a que la comunicación fuese más rápida.

Otro resultado obtenido que es valioso analizar es la forma como se controla la tensión que se le suministra al motor, lo anterior se realizó controlando el ancho del pulso de una señal PWM. Esta tarea fue realizada mediante un módulo de control de ancho de pulso de PSoC. Se ajustó de manera tal que su potencia fuese suministrada en fracciones de 2400 segmentos. (Ver imagen 4.7)

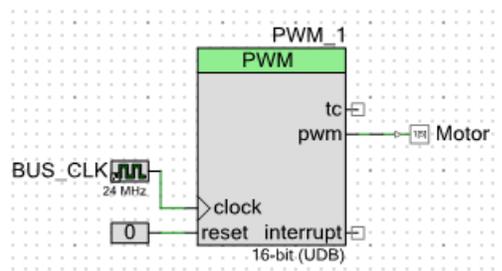


Imagen 4.7: Módulo de implementación PWM utilizado

Esta decisión definió que la resolución cuentas con que se controle el motor es tal que se pueden dar variaciones de hasta 0.005 voltios por lo que su ajuste es muy fino, y eso es de gran ventaja a la hora de aplicar las diferentes respuestas de control. En el módulo de ajustes y pruebas se pudo experimentar con estos cambios paulatinos desde cero hasta su tensión máxima y se observa como de manera gradual la tensión es incrementada. (Ver imagen 4.8)

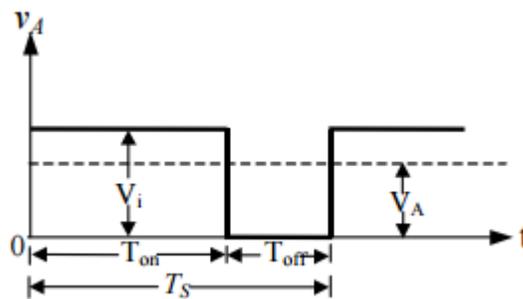


Imagen 4.8: Señal de PWM aplicada al motor

La forma matemática como se expresa la tensión aplicada se define por la siguiente ecuación:

$$V_A = \frac{T_{on}}{T_s} V_i \quad (4.1)$$

Donde  $T_{on}$  se puede representar como

$$T_{on} = \frac{\text{valor de cuentas utilizado}}{\text{m\u00e1ximo de cuentas configurado}} T_s \quad (4.2)$$

Por lo que se puede expresar a  $V_A$  de la siguiente manera

$$V_A = \frac{T_{on}}{T_s} V_i = \frac{\text{valor de cuentas utilizado}}{\text{m\u00e1ximo de cuentas configurado}} \frac{T_s}{T_s} V_i \quad (4.3)$$

$$V_A = \frac{\text{valor de cuentas utilizado}}{\text{m\u00e1ximo de cuentas configurado}} V_i \quad (4.4)$$

Si bien es cierto, el valor que se le aplic\u00f3 de tensi\u00f3n al motor no es dependiente de la frecuencia del PWM, tal y como se despej\u00f3 en la anterior ecuaci\u00f3n, se debe ser enf\u00e1tico que cuando se utiliz\u00f3 una baja frecuencia en el PWM, los resultados que se obtuvieron no eran los deseados en el control, porque se observ\u00f3 un comportamiento m\u00e1s brusco en la planta.

La justificaci\u00f3n de dicho comportamiento es que, a menor frecuencia, el tiempo que transcurre para ejecutar un cambio en la tensi\u00f3n es mayor que si la frecuencia fuese mayor, lo anterior se debe a que los cambios en la se\u00f1al que se generan en el m\u00f3dulo PWM los hace hasta que haya concluido un ciclo de trabajo, y es de conocimiento general que el tiempo que tarda un ciclo es inversamente proporcional a la frecuencia.

Tomando en cuenta que la planta pod\u00eda operar con un periodo de hasta un m\u00ednimo de un milisegundo en su tiempo de muestreo, se decidi\u00f3 que el ciclo de PWM fuese un d\u00e9cimo de dicho valor, o sea 100 microsegundos, por lo tanto, el PWM se ajust\u00f3 para una frecuencia operaci\u00f3n de 10 Khz.

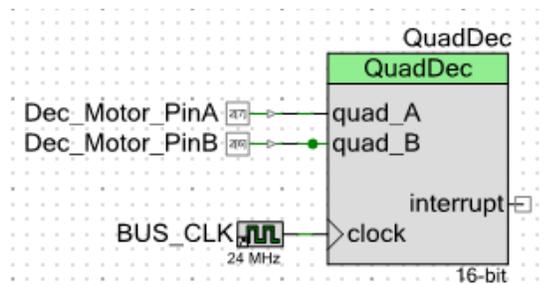


Imagen 4.9: Diagrama de conexión del decodificador de cuadratura

En cuanto a las posiciones angulares del péndulo y motor se detectaron mediante la implementación de módulos decodificadores de cuadratura, configurados como la imagen 4.9, teniendo presente la cantidad de cuentas por revolución de cada decodificador y su relación en grados o radianes según fuese el caso a utilizar. Una vez implementada estas funciones de lectura de posición angular, se procedió a verificar y validar los datos obtenidos mediante una prueba de posición y dibujo de la señal de trayectoria, el resultado de éste experimento se observa en la siguiente imagen y puede ser reproducido en el módulo de pruebas y ajustes. A continuación, se ilustran los movimientos aleatorios que se le hicieron de manera manual al péndulo en el transcurso de una rotación completa, con el objetivo de verificar de su correcto funcionamiento.

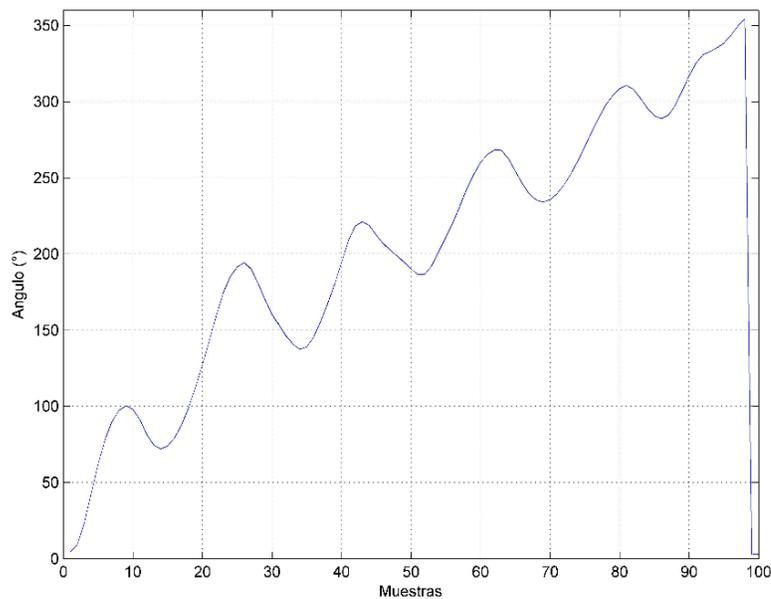


Imagen 4.10: Prueba de captura de ángulo del péndulo

Es importante analizar de manera general el diseño, selección de piezas y su resultado final, el cual fue bastante satisfactorio, puesto que es rígido, sólido, no se doblega ante movimientos del motor o cambios de inercia en el péndulo, la caja de acrílico protector permite al estudiante inspeccionar los elementos que lo componen sin que los componentes estén vulnerables a manipulaciones indebidas.

Como se observó que la fuerza con que se movía el motor, en ocasiones es tal que movía toda la estructura, se decidió solucionar este movimiento anclando el chasis a una base de madera sólida, con lo cual solucionó el inconveniente completamente.

Se recomienda posicionar la planta sobre una superficie sólida y totalmente a nivel para su operación, cuando se hace caso omiso a ésta recomendación el plano de movimiento del brazo no queda completamente horizontal y provoca que prefiera una zona para posicionarse, lo que en cierta medida afecta la acción de control.

La fuerza del motor seleccionado fue la correcta, puesto que mueve con facilidad el péndulo incluso con masas adicionales, su velocidad de reacción y acción son suficientes y en la mayoría de las oportunidades no es necesario aplicarle toda la potencia para que logre controlar el péndulo aún expuesto a perturbaciones. El único inconveniente que se encontró es en el tipo de caja reductora, puesto que entre sus engranajes internos cuenta con un juego, donde el brazo se puede mover libremente sin que sea excitado por el motor, o bien, sin que el decodificador detecte el movimiento.

Este “juego” o zona de libertad es de aproximadamente unos 5 grados, por lo que no es tanto como para que impida su uso en los diferentes módulos, sin embargo, sí es evidente que modifica en cierta medida los valores y resultados obtenidos.

En cuanto a la operación del motor, se detectó mediante los experimentos instalados en el módulo de motor real, que es completamente asimétrico en relación con a la dirección del giro, lo cual del todo no es un inconveniente porque estas pequeñas diferencias se compensaron en el programa de control desarrollado.

Además, desde el punto de vista académico, estas diferencias más bien pueden que sean valiosas para enseñarle al estudiante los inconvenientes de trabajar con dispositivos reales cuando en la teoría habitualmente han sido tratados desde el punto de vista de teórico como elementos ideales, e incluso discutir posibles soluciones que ayuden a mitigar estos efectos y logre reducir la brecha entre el dispositivo real con que se encuentra trabajando y el ideal que propone la teoría.

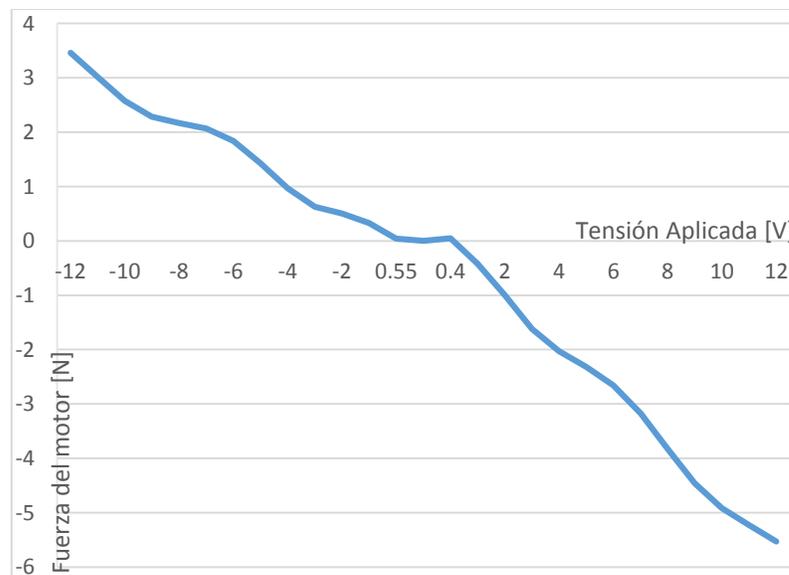


Imagen 4.11: Comportamiento no lineal y asimétrico del motor DC

En la imagen 4.11 se observan los dos problemas detectados de asimetría, el primero es que bajo una misma tensión aplicada al sentido contrario no genera la misma cantidad de fuerza; fue solucionado mediante una ganancia previamente calculada con el uso de un dinamómetro y el experimento de fuerza.

El segundo problema que se observó en los experimentos es la “zona muerta” del motor ante tensiones muy bajas, ocasionada por la fricción estática. La fricción estática en un motor real se puede solucionar añadiéndole el valor de la tensión que compense esta carga adicional del motor. Se logró corroborar que éste valor no es el mismo para ambos sentidos de giro.

Una vez que se han analizado los diferentes componentes, configuraciones y lógicas de forma independiente, se procedió a validar el comportamiento de la planta de manera completa, interactuando todos los componentes simultáneamente, para así poder validar que el comportamiento de la misma es tal y como la teoría lo describe.

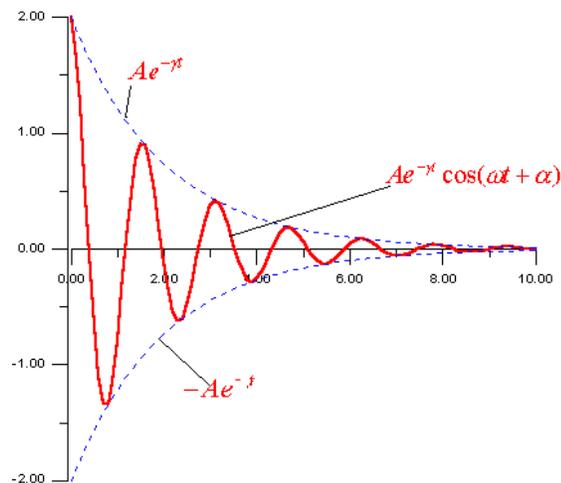


Imagen 4.12: Trayectoria teórica de un péndulo subamortiguado [10]

En cuanto al péndulo en modo estable, su comportamiento se puede clasificar del tipo subamortiguado, donde según la teoría, después de haber sufrido una perturbación o energía inicial, se desplaza tal y como se muestra en la imagen 4.12, de forma sinusoidal, pero sufriendo una paulatina atenuación, cambio sufrido por efecto de fuerzas de fricción en la estructura y contra el aire.

En la imagen 4.13, se observa el comportamiento del péndulo registrado por SiDiCo, como ante un impulso en la tensión de armadura del motor (señal verde) y trasladado al brazo, genera un desplazamiento en la posición del péndulo (señal azul), y seguidamente una serie de oscilaciones que se atenúan poco a poco en busca de su posición estable, o sea hacia la posición de los cero grados.

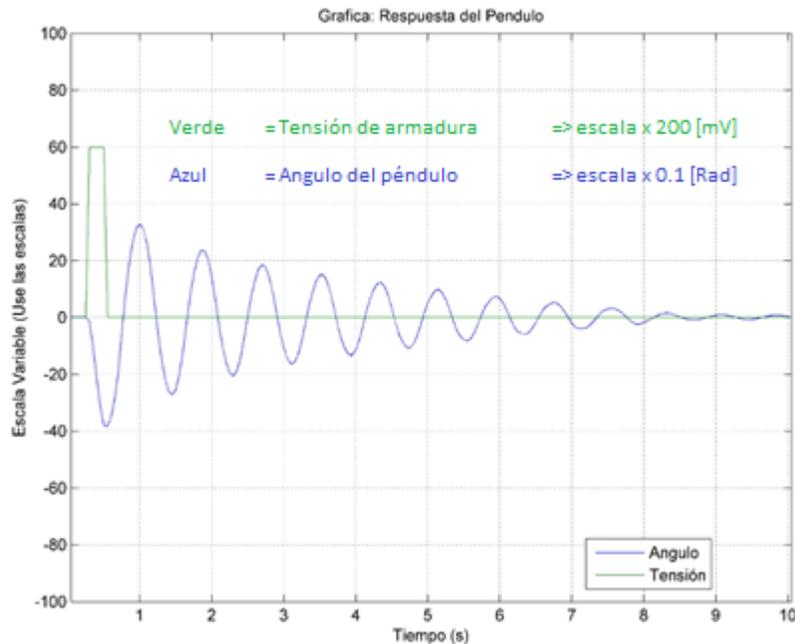


Imagen 4.13: Respuesta del péndulo subamortiguado de SiDiCo

Mediante diferentes métodos de análisis matemático el estudiante ya posee el conocimiento adecuado para deducir de manera analítica el modelo del sistema, pero en este caso, por tratarse SiDiCo de un sistema para laboratorio, brinda la posibilidad de registrar los datos del movimiento para que luego sean identificados mediante procedimientos que se detallan en el instructivo del proyecto corto denominado “Obtención del modelo empírico de un Péndulo (SISO)” (Apéndice 6.4).

Una vez que se han procesado los datos experimentales de la planta, se obtiene el siguiente modelo continuo y discreto, con un periodo de muestreo de cincuenta milisegundos.

$$\frac{-0.17953 (s + 35.75)}{(s^2 + 0.5832s + 60.02)} \quad (4.5)$$

$$\frac{-0.016474 (z - 0.05203)}{(z^2 - 1.825z + 0.9713)}, T = 50ms \quad (4.6)$$

E incluso la realización del mismo instructivo nos permite visualizar y comparar los resultados obtenidos mediante la imagen 4.14, lo que permite validar los resultados obtenidos al ser tangible su similitud entre la señal experimental y la calculada mediante la herramienta ident

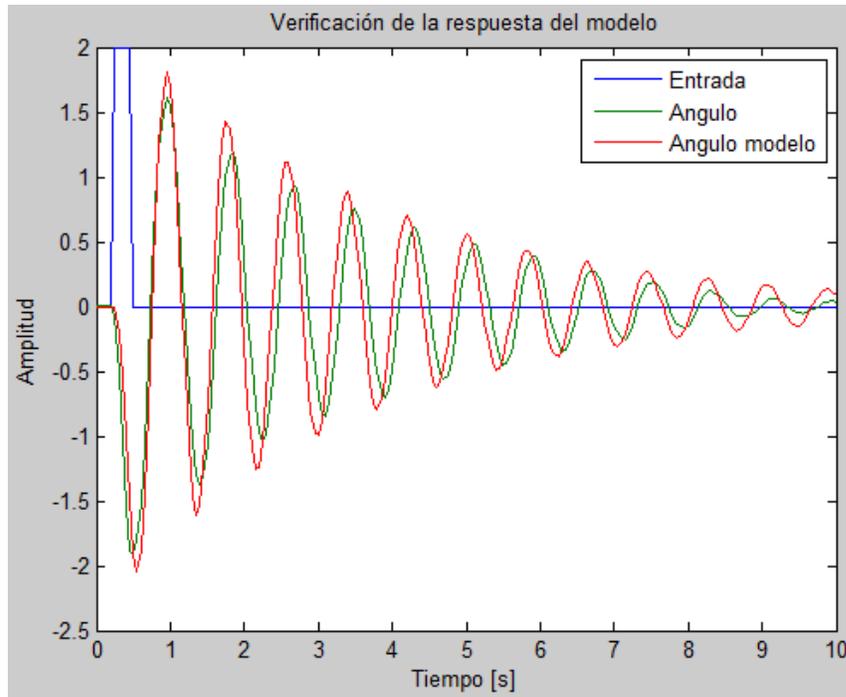


Imagen 4.14: Comparación de señal y modelo obtenido experimentalmente

En la imagen anterior se aprecian algunas diferencias pequeñas entre las señales, sin embargo, el estudiante puede mejorar el modelo dependiendo de una serie de parámetros ajustables desde la interfaz gráfica de SiDiCo, como lo son el periodo de muestreo, duración del experimento, diezmado, entre otros.

Por último, para verificar si la planta efectivamente es inestable, tanto de la manera simulada como práctica, se procederá a completar el modelo analítico encontrado de la planta

$$p\acute{e}ndulo = \frac{\frac{m * l}{q} * s}{s^3 + \frac{b * (I + m * l^2)}{q} * s^2 - \frac{m * g * l * (M + m)}{q} * s - \frac{b * m * g * l}{q}} \quad (4.7)$$

Donde:

$$q = (M + m) * (I + m * l^2) - (m * l)^2 \quad (4.8)$$

Los valores de las variables que lo componen se muestran en el apéndice 6.1, y luego de sustituirlos en la ecuación 4.7 se obtiene una función de transferencia con los siguientes valores

$$pendulo = \frac{53.843 * s}{(s + 7.766) (s - 7.76) (s + 0.01059)} \quad (4.9)$$

Se puede apreciar como la planta es inestable a causa de un polo en  $S=7.76$ , lo cual es consistente con el comportamiento práctico de la misma, donde no le es posible posicionarse en el ángulo cero (vertical superior) estando en lazo abierto. De manera gráfica los polos se pueden observar mediante la herramienta *rlocus de Matlab*. (Ver imagen 4.15)

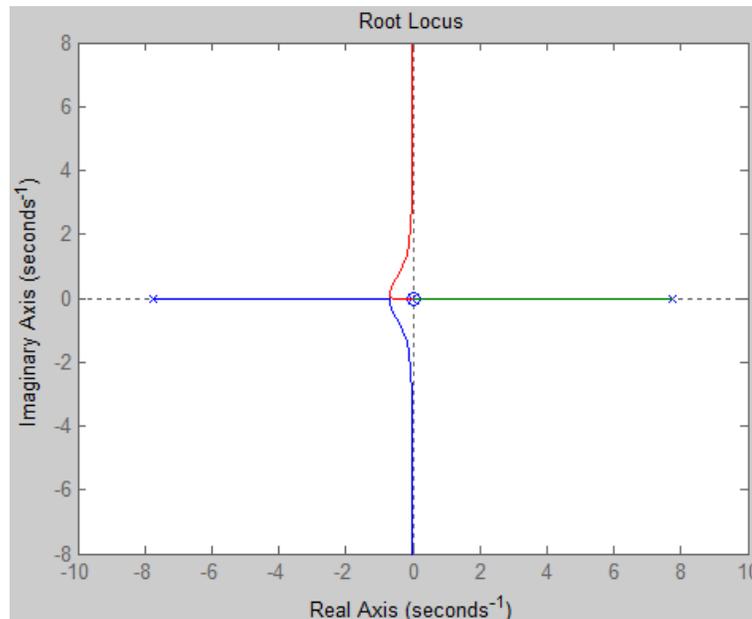


Imagen 4.15: Representación gráfica del lugar de las raíces

Como parte de los procedimientos para encontrar el control que se establecen en los instructivos que se encuentran en los apéndices, se sugiere utilizar herramientas de simulación, en esta oportunidad para analizar su respuesta se utilizó la herramienta simulink de Matlab.

Para simular el primer control se decidió generar uno del tipo PID, se diseñó en dicha aplicación el diagrama de bloques correspondiente para poder comparar las respuestas de la planta partiendo del modelo analítico anteriormente mencionado. A continuación, el diagrama realizado. (Ver imagen 4.16)

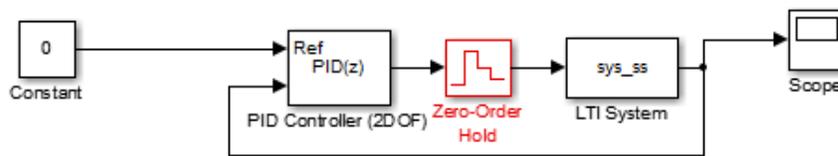


Imagen 4.16: Diagrama de Simulink de la planta

Mediante la función de *Auto Tune* del bloque PID se logró ajustar los valores de sus variables, el tiempo de estabilización de la simulación es de 0.04 segundos, tiempo adecuado para una planta de esta naturaleza.

En la imagen 4.17 se muestran los valores ajustados, las gráficas de simulación y los parámetros dinámicos obtenidos.

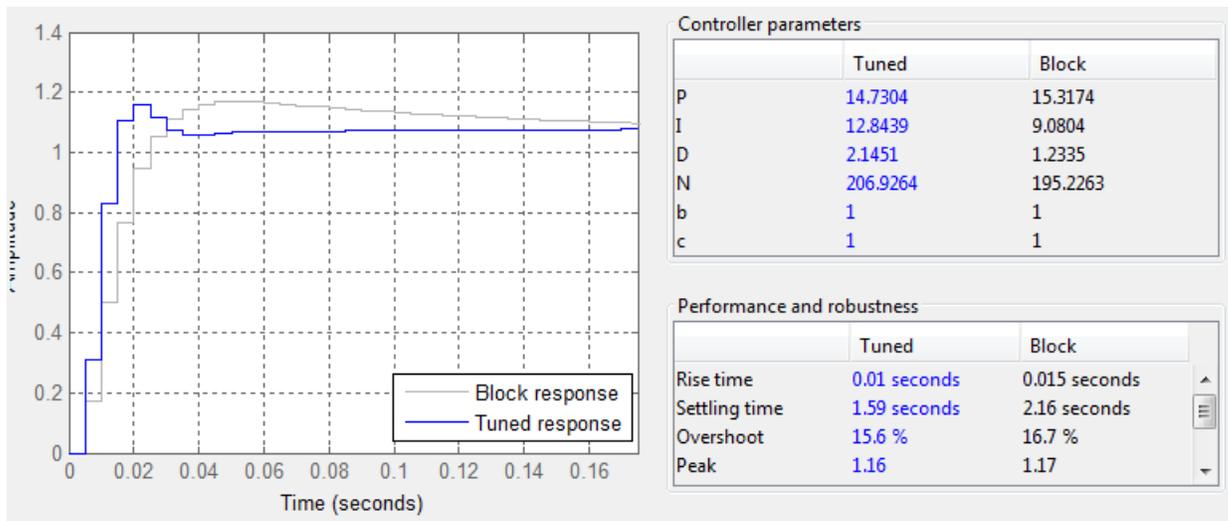


Imagen 4.17: Parámetros del auto ajuste PID

La validación de los datos teóricos en esta oportunidad se hace evaluando estos parámetros en el control instalado en la planta, a pesar de que su comportamiento es cercano a mantener el control del péndulo, se debe reconocer que esos intentos no se pueden interpretar como tal, por lo que se procedió a utilizar el método heurístico o ajuste con criterio y se obtuvieron los siguientes valores (ver tabla 4.1)

Tabla 4.1: Tabla de ajustes PID encontrados por diferentes métodos

	Simulación	Calibración con criterio
P	14.73	15
I	12.84	19
D	2.14	0.2
N	206.92	200

Como segundo método de verificación de la planta se seleccionó la lógica de control LQR, por ser la más utilizada en este tipo de planta y por contar con herramientas de cálculo

disponibles dentro de la plataforma de Matlab. En la imagen 4.18 se observa la función LQR utilizada, su modo de uso y resultado empleado.

```
>> K = lqr(A,B,Q,R)

K =

    -1.4142    -1.0014     4.5769     0.6489

fx >>
```

Imagen 4.18: Cálculo de la matriz K mediante la función LQR de Matlab

El control LQR se basa en encontrar una ley de control que minimice la suma de los esfuerzos de control y las desviaciones de la señal de salida de su valor deseado, basando su cálculo en la matriz “A” de estados de la planta.

Para el uso de éste método es necesario tener modelada la planta en forma de variables de estado, al utilizar las ecuaciones del apéndice 6.1.5 y evaluándolas con los parámetros de la planta del apéndice 6.1.1 se llega a los siguientes resultados:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0165 & 5.4713 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0653 & 60.2657 & 0 \end{bmatrix} \quad (4.10)$$

$$B = \begin{bmatrix} 0 \\ 13.6224 \\ 0 \\ 53.8434 \end{bmatrix} \quad (4.11)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.12)$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.13)$$

Conociendo las variables de estado, se definió mediante la entrada “Q” del cálculo (ver imagen 4.18) cuáles son las salidas del sistema: la posición del péndulo y del brazo. La matriz “Q” tiene sus celdas en cero, a excepción de las que corresponden a salidas del sistema.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \quad (4.14)$$

Éstos parámetros  $q_2$  y  $q_4$  se definieron mediante el criterio de pesos máximos, o sea es el cuadrado del valor máximo permitido para éstas salidas durante su funcionamiento. Para ambos casos se escogió que tuviese una oscilación máxima de  $\pi/6$  radianes ( $15^\circ$  cada lado) y luego elevándolo al cuadrado.

$$q_2 = q_4 = \left(\frac{\pi}{6}\right)^2 \quad (4.15)$$

El criterio descrito lo que busca es que cuando el error de la salida sea el máximo permitido, saber cuál es la razón de corrección que debe calcular. [11]

$$q_2 x_2^2 = 1 \gg \text{cuando } x_2 = \frac{\pi}{6} \quad (4.16)$$

En cuanto a la entrada R del cálculo, existen varios criterios de selección, sin embargo, se optó por definir arbitrariamente  $R = 1$  ya que el objetivo de ésta variable es sintonizar el resultado a obtener.

La validación de los datos teóricos del control LQR se hace evaluando estos parámetros en el control instalado en la planta, a pesar de que su comportamiento nuevamente es muy cercano a mantener el control del péndulo, se debe reconocer que no es logrado, por lo que se ajustaron los resultados mediante el método heurístico o ajuste con criterio y se obtuvieron los siguientes valores (ver tabla 4.2)

*Tabla 4.2: Tabla de ajustes LQR encontrados por diferentes métodos*

	LQR Matlab	Calibración con Criterio
$K_1$	-1.4142	-1.5
$K_2$	-1.0014	-1.8
$K_3$	4.5769	4.5
$K_4$	0.6489	0.98

Si bien es cierto, los valores teóricos encontrados por ambos métodos son diferentes al método heurístico correspondiente, hay que reconocer que su semejanza es muy alta, además, tomando en cuenta que el modelo teórico con que se realizaron las simulaciones o cálculos puede contener errores de medición de las variables, o incluso hubo comportamientos mecánicos como la zona muerta del motor que no se tomaron en cuenta en el modelo teórico, por lo tanto, se puede concluir que es una muy buena aproximación.

En éste caso en particular, se emplean los datos simulados y calculados para encontrar una zona aproximada de control y luego mediante ajuste llegar a sintonizar la planta. Después de este exhaustivo análisis de los resultados se toma como válido el método de simulación de

PID, el método de cálculo LQR y el comportamiento de la planta, pero se recomienda acompañarlo de un pequeño ajuste con criterio, que sería el procedimiento que viene a compensar los errores de medición, valores que se despreciaron en el proceso de cálculo y, además, pequeñas consideraciones en el modelo que no se contemplaron.

## 5. Conclusiones y Recomendaciones

### 5.1. Conclusiones

- Mediante el uso del programa Matlab se logró diseñar la interfaz gráfica que permite al usuario ingresar datos para ser procesados por la planta y obtener los resultados en las diferentes pruebas
- La interfaz gráfica implementada permite al usuario establecer comunicación con la planta y así comunicarle los datos y ejercicios para obtener los resultados del estudio.
- La rápida respuesta, compatibilidades eléctricas, la correcta selección de los materiales de la estructura, la precisión y exactitud en los datos son requisitos que se cumplen a plenitud en el diseño de la planta.
- Se logró construir una planta de péndulo de Furuta, que, aunque tiene pequeños problemas en el motor, tales como asimetría, zona muerta y juego, pudo ser controlada con reguladores PID y LQR
- Mediante herramientas utilizadas en los proyectos cortos de laboratorio de control, se puede llegar al modelo experimental del péndulo de Furuta y compararlo con el modelo teórico conocido para la planta.
- Mediante el sistema de control diseñado se logró implementar los controladores LQR y PID, los cuales son un buen algoritmo para lograr la respuesta deseada sobre el Péndulo de Furuta.

- Se implementó el sistema de control para el péndulo de Furuta que, ante las perturbaciones inducidas en el péndulo, genera una señal de control que responde de manera adecuada, pero se recomienda utilizar ya sea el método de ajuste de control por simulación o por cálculo para que defina una zona aproximada de los parámetros y, luego, mediante el ajuste con criterio, compensar los errores de cálculo y variables no considerados en el modelo.
- La versatilidad de la planta permitió elaborar una serie de instructivos para que los estudiantes puedan realizar sus proyectos cortos de Laboratorio de Control Automático con ella.

## 5.2.Recomendaciones

- Utilizar un motor con una menor zona muerta para que el péndulo oscile menos y se logre un mayor estado de equilibrio (cuando está en lazo cerrado).
- Reemplazar el motor por uno de caja reductora planetaria para poder minimizar el juego en el brazo.
- Instalarle a la planta un codificador de cuadratura directamente en el eje del brazo, para lograr registrar todos los movimientos hechos en el brazo.
- Implementarle otros algoritmos de control a SiDiCo, para que el estudiante pueda experimentar diferentes técnicas y comparar resultados obtenidos.
- Modificar la lógica de captura del ángulo (utilizar DMA) para que este se haga en paralelo y así minimizar los tiempos de captura de información.

- Instalarle a la planta algún sistema de ajuste de nivel, de manera que se le pueda calibrar el nivel para que el plano de movimiento del brazo sea completamente horizontal y no tenga zonas de preferencia.

## 6. Apéndices

### 6.1. Hoja de Datos de la Planta

A continuación, se muestran los parámetros de la planta Péndulo Invertido de Furuta empleado en SiDiCo.

#### 6.1.1. Chasis del Péndulo de Furuta

Tabla 6.1: Parámetros del chasis creado

Nombre	Variable	Valor	Unidad
Masa paleta	M_palet	$27.69 \cdot 10^{-3}$	Kg
Masa externa	M_ext	$54.48 \cdot 10^{-3}$	Kg
Distancia de la masa externa	D_ext	0.1524	M
Masa total del Péndulo	M	0.08217	Kg
Distancia del centro de masa del péndulo	Lp	0.1265	M
Longitud brazo horizontal	Radio	$139 \cdot 10^{-3}$	M
Masa brazo horizontal	M1	$222.33 \cdot 10^{-3}$	Kg
Distancia del centro de masa del brazo	l_brazo	$53 \cdot 10^{-3}$	M
Masa equivalente del brazo horizontal	M	0.03232363	Kg
Gravedad	G	9.77589	m / s <sup>2</sup>

#### 6.1.2. Constantes del Motor

Tabla 6.2: Parámetros del Motor Utilizado

Nombre	Variable	Valor	Unidad
Tensión aplicada en el Motor	Vs	7.8	V
Resistencia	Rm	10.745699	$\Omega$
Inductancia	La	0.0061504	H
Inercia	J	$6.62216 \cdot 10^{-4}$	Kg * m <sup>2</sup>
Constante de torque	Kt	0.424827286	N * m / A

Constante de fuerza electromotriz	Km	0.424827286	V * s / rad
-----------------------------------	----	-------------	-------------

### 6.1.3. Dinámica del Péndulo

Tabla 6.3: Parámetros del Péndulo creado

Nombre	Variable	Valor	Unidad
	Bp	1.2119*10^-3	
	lp	1.3149*10^-3	
Masa equivalente del brazo horizontal	M	0.03232363	Kg
Masa total del Péndulo	M	0.08217	Kg
Distancia del centro de masa del péndulo	L	0.1265	M

### 6.1.4. Función de Transferencia [10]

$$pendulo = \frac{\frac{m * l}{q} * s}{s^3 + \frac{b * (I + m * l^2)}{q} * s^2 - \frac{m * g * l * (M + m)}{q} * s - \frac{b * m * g * l}{q}}$$

Donde:

$$q = (M + m) * (I + m * l^2) - (m * l)^2$$

### 6.1.5. Variables de Estado [10]

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I + m * l^2) * b}{p} & \frac{(m^2 * g * l^2)}{p} & 0 \\ 0 & 0 & p & 0 \\ 0 & \frac{-(m * l * b)}{p} & \frac{m * g * l * (M + m)}{p} & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ (I + m * l^2) \\ p \\ 0 \\ \frac{m * l}{p} \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Donde:

$$p = I * (M + m) + M * m * l^2$$

Y el orden de las variables de estado es:

$$x = \begin{bmatrix} \textit{Posición del brazo} \\ \textit{Velocidad del Brazo} \\ \textit{Posición del péndulo} \\ \textit{Velocidad del péndulo} \end{bmatrix}$$

## 6.2. Proyecto Corto 1 SiDiCo (Furuta SIMO)



INSTITUTO TECNOLÓGICO DE COSTA RICA

**II SEMESTRE 2016**

ESCUELA DE INGENIERIA EN ELECTRÓNICA

CURSO: EL-5409 LABORATORIO DE CONTROL AUTOMÁTICO

MEDIO: Proyecto corto 1 (Furuta SiDiCo)

FECHA: 18 de julio de 2016

PROF: Ing. Eduardo Interiano

# PROYECTO CORTO 1

**Tema: Obtención del modelo empírico de un Péndulo de Furuta en modo estable (Grua).**

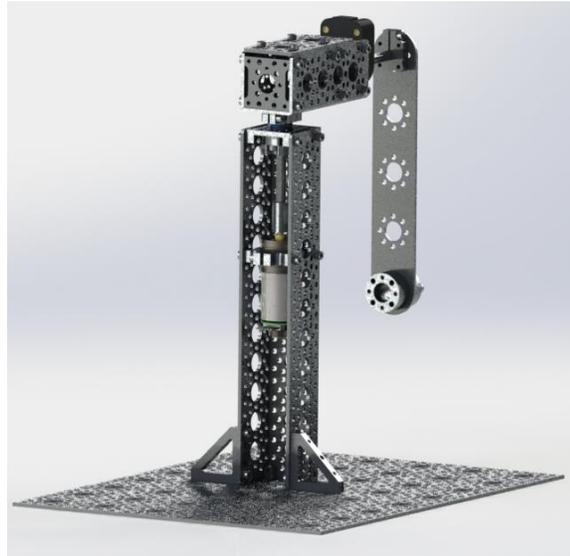


Figura 1: Péndulo de Furuta SiDiCo

**Recursos:** Sistema Didáctico de Control “SiDiCo” con el software V1.0 y planta Péndulo invertido de Furuta V1.0 (con motor CD de imán permanente de 12V con caja reductora 46.85:1; sensor de posición angular E5 de 4000cpr en el péndulo y un sensor de posición angular integrado en el giro del motor de 2248.86cpr). Computador con Matlab R2013b o superior e *ident* instalado.

### Descripción del trabajo:

Debe utilizar SiDiCo V1.0 para capturar los datos de un experimento para obtener las respuestas de posición angular del péndulo y posición del brazo de la planta Péndulo de Furuta ante un estímulo pseudoaleatorio de tensión de armadura del motor CD. Posteriormente debe utilizar la

herramienta *ident* de Matlab para identificar y estimar el modelo SIMO adecuado para la planta y finalmente debe verificar el modelo así obtenido.

## Parte I: El experimento

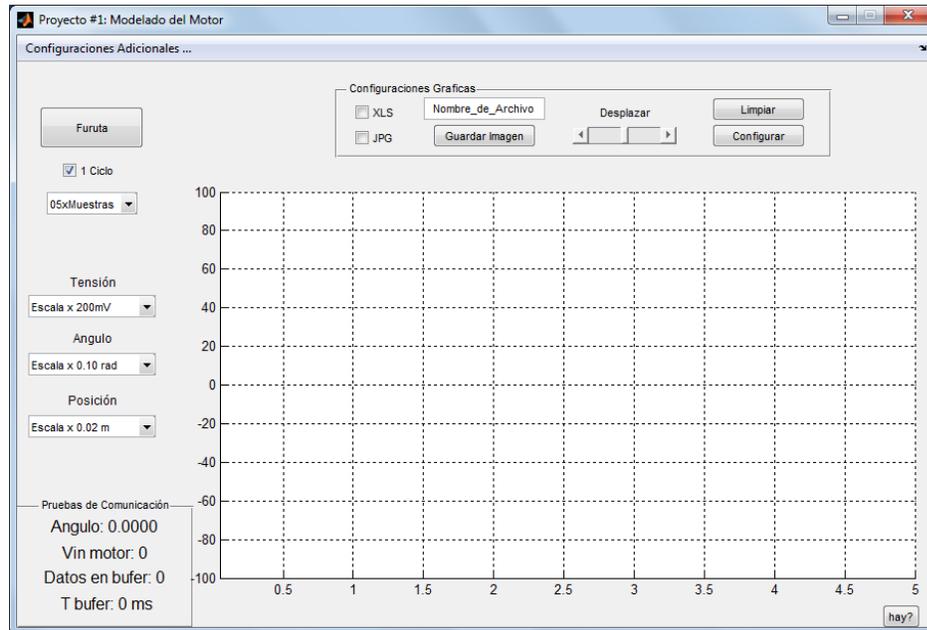


Figura 2: Modulo Modelado de Furuta (SiDiCo V1.0)

Paso 1) Conectar la Planta Péndulo Invertido de Furuta al software SiDiCo V1.0

Paso 2) Ingresar al Módulo “Modelado Furuta”

Paso 3) Antes de iniciar cada medición asegurarse que el área está despejada para el libre movimiento del brazo y péndulo.

Paso 4) Se sugiere ajustar las propiedades del programa (esquina superior izquierda) como se muestra en la Tabla 1. Usted puede modificar estos ajustes para adecuarlos a su experimento si es necesario. Para el primer experimento se sugiere conservar los valores por defecto.

Paso 5) Se sugiere ajustar el largo del pulso a aplicar al motor en “cantidad de muestras” y la extensión del experimento en “1 Ciclo” ubicadas bajo el botón principal de la pantalla como se muestra en la imagen 2. Usted puede modificar estos ajustes para adecuarlos a su experimento si es necesario. Para el primer experimento se sugiere conservar los valores por defecto.

Paso 6) Establezca un “Nombre\_de\_Archivo” y seleccione “XLS” y “JPG” para que el futuro experimento sea guardado.

Paso 7) Ejecute el experimento “Furuta” y espere hasta que la planta y programa se detengan completamente.

Paso 8) Si durante el experimento los “Datos en Buffer” crecen, se recomienda aumentar el valor de muestreo o periodo de la planta, esto debido a que la velocidad de procesamiento del computador en uso es menor al deseado. Caso contrario, si el valor no crece, puede disminuir estos valores para obtener más detalles en la gráfica. De ser necesario repita el experimento.

Tabla 1: Ajustes del Programa

Función	Propiedades	Comentario
Periodo T en la planta	10 ms	El periodo de operación de la planta puede variar entre 1 y 255ms, sin embargo, si se escogen valores pequeños satura el buffer o altos se pierden detalles de la gráfica
Angulo de Op	60 grados	Se recomienda ajustar el Angulo en el máximo permitido para que el experimento no se vea interrumpido
# en X	100 muestras	En 100 muestras el experimento despliega los detalles necesarios, si se desea ver más se puede ampliar hasta 550 muestras, o deshabilitar la opción “1 ciclo” de la pantalla principal
Actualiza c/	5 T	Regula el tiempo de diezmo, se puede reducir dependiendo de la velocidad de procesamiento del computador. El objetivo es no saturar el USB.
Máximo en Y	100 líneas	En 100 muestras el experimento despliega los detalles necesarios, si se desea ver más se puede ampliar hasta 365 muestras
Otros	No relevantes	No es necesario ajustar los demás parámetros, puesto que no se emplean en este experimento y, por lo tanto, no generan variante.

## Parte II: Estimación y verificación del modelo SIMO de la planta

**Objetivo:** Estimar, a partir de los resultados experimentales de entrada-salida de la planta, el modelo numérico del Péndulo de Furuta.

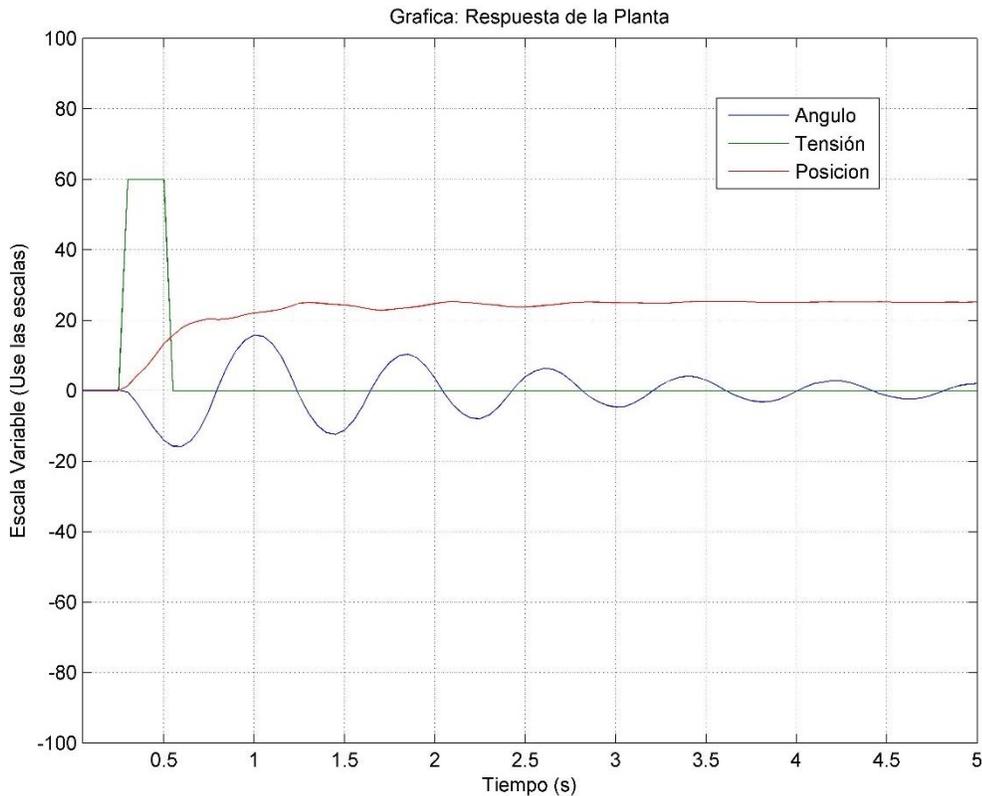


Figura 3: Captura del impulso de entrada [V], la posición angular [rad] del péndulo y el desplazamiento lineal del brazo [m].

**Procedimiento:**

**NOTA 1:** Antes de iniciar el procesamiento de los datos experimentales, puede ejecutar *iddemo* en Matlab, seleccione el tipo de demostración 1 y luego seleccione la demostración de la interfaz gráfica de usuario para estimación (opción 1) o consulte [3], [4].

Paso 1) Identifique y estime, con ayuda de *ident* de Matlab, varios modelos en variables de estado para la planta. Seleccione el modelo con el mejor ajuste a los datos experimentales y valide ese modelo seleccionado en simulación contra los datos experimentales.

**NOTA 2:** En el ejemplo aquí mostrado se utilizan otros valores que no necesariamente son iguales a los que usted deberá de obtener, solo es una guía que muestra los procedimientos, y no los valores a conseguir. **Usted debe utilizar los datos de su experimento que corresponden a un caso SIMO.**

**NOTA 3:** Antes de iniciar el procesamiento de los datos verifique la **consistencia del separador decimal** de sus datos con las herramientas a usar; esto es, si las herramientas están en inglés, el separador decimal debe ser punto (.); si están en español, el separador decimal debe ser coma (,). **De preferencia use un editor de texto simple** para esta labor, (*Notepad* por ejemplo).

Paso 2) Importe a Matlab los datos contenidos en el archivo .XLS guardado en la parte I; para ello, inicie Matlab y en vaya al menú *Home\Import Data...*, seleccione el archivo y luego oprima *Open*, en la siguiente ventana seleccione *Column vectors ...*, y en *Range* asegúrese que inicie en la columna B como se muestra en la figura 4 (el largo de la selección dependerá del largo de su experimento) y luego oprima *Import Selection*. Los datos están ahora disponibles como vectores en el *workspace* de Matlab.

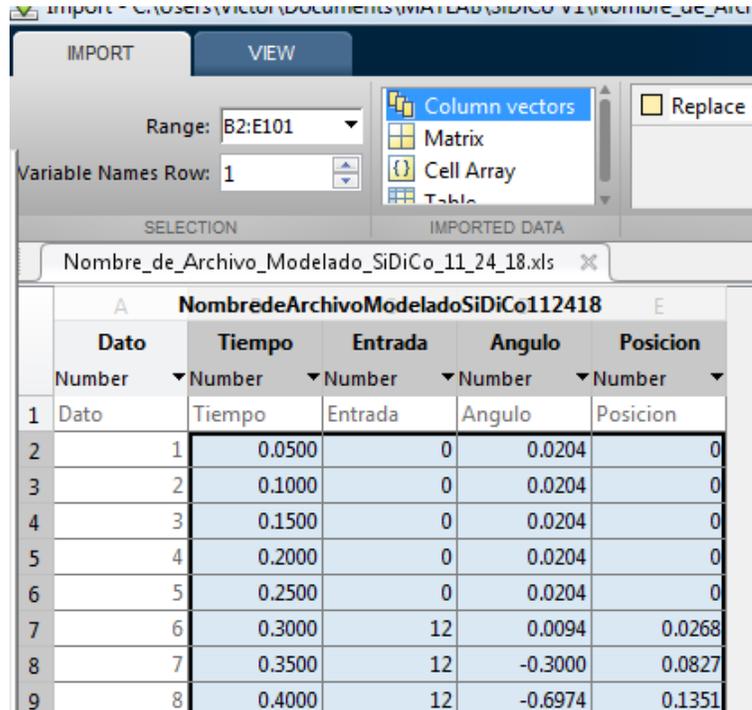


Figura 4: Importar datos en forma de vectores al *workspace* de Matlab.

Paso 3) Cree en Matlab un objeto de datos para identificación, en este caso llamado 'Furuta', usando las siguientes sentencias:

**NOTA 4:** Antes de crear el objeto de datos asegúrese de filtrar el ruido de los datos (un filtro de media móvil es muy práctico para esto) y de linealizar los datos (restar el valor inicial), de lo contrario, el ruido y la no linealidad de los datos evitarán que obtenga modelos apropiados y con la exactitud suficiente. Para este experimento SiDiCo ya guarda los datos filtrados y linealizados, sin embargo, no descarte la necesidad de usar estas herramientas si con criterio lo considera necesario.

**NOTA 5:** Asegúrese que el periodo de muestreo sea constante, revisando el vector de **Tiempo**; y en Matlab use la orden **format shortg** para poder ver los datos en formato sin redondeos.

```
deltaT = Tiempo(2)-Tiempo(1);
Furuta = iddata([Angulo, Posicion], [Entrada], deltaT);
Furuta.inputname = {'Entrada'};
Furuta.outputname = {'Angulo'; 'Posicion'};
```

Paso 4) Luego en el *workspace* de Matlab ejecute la orden *ident*; espere unos segundos y se abrirá la ventana del *ident*, en la cual debe seleccionar *Import data>Data object ...*, llene los campos de la ventana con los datos correspondientes **a su experimento**: nombre del objeto de datos, creado en el paso anterior, que contiene los datos experimentales como se muestra en la figura 5 y oprima *Import*. Una vez hecho lo anterior puede cerrar esa ventana.

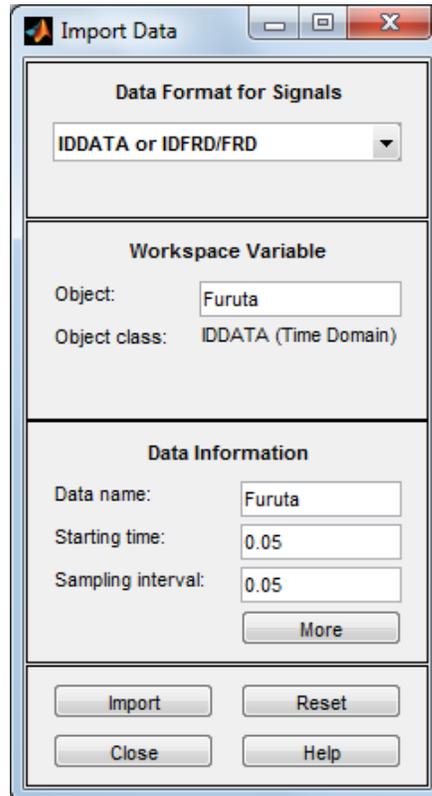


Figura 5: Ventana para importar datos al *ident*. Ponga los datos de **su experimento**.

En este ejemplo, en la ventana principal del *ident*, figura 8, debe tener ahora un objeto, llamado **Furuta**, debajo de *Import data*. Este objeto contiene todos los datos del experimento, los cuales pueden visualizarse en dos partes. Usaremos para ello la casilla *Time plot*, que abrirá una ventana en la cual podemos seleccionar el canal (*Channel*) que deseamos ver, tal como se muestra en la figura 6.

Paso 5) En la ventana principal del *ident* seleccionamos *Estimate* → *State Space models...* y en esta ventana, ya que nuestro modelo tiene una entrada y dos salidas (SIMO) escogeremos una estructura de variables de estado y oprimiendo la selección *Pick best value in the range* buscaremos modelos desde orden 1 a 10 como se muestra en la figura 7a. Una vez hecho lo anterior oprimimos el botón *Estimate*, el cual hará que abra la ventana de selección del orden del modelo, como se muestra en la figura 7b. Aquí seleccionaremos modelos que tengan valores singulares pequeños, sin aumentar excesivamente el orden del modelo.

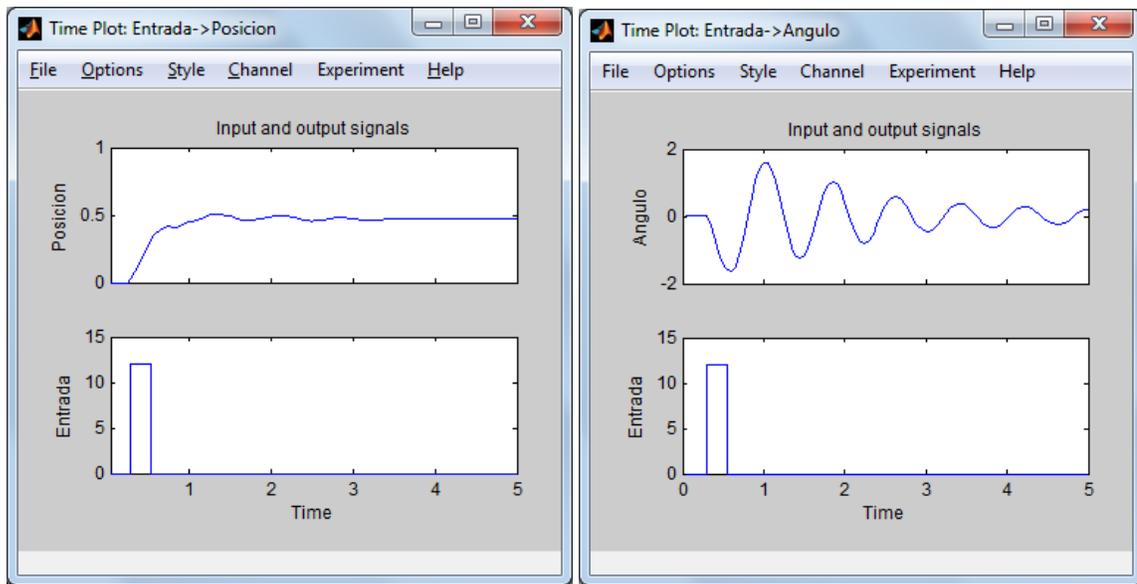


Figura 6: Visualización de los datos experimentales importados en *ident*.

Para validar nuestros modelos obtenidos, podemos comparar su salida ante el estímulo original del experimento seleccionando la opción *Model output* que nos presenta el porcentaje de aproximación entre nuestro modelo y la respuesta original como se muestra en la figura 9.

Paso 6) Como puede observarse en la figura 9, para la salida **Angulo**, el modelo *Primero* posee la mejor aproximación a los datos experimentales (74.72) y para la salida **Posición** el mejor modelo, con una aproximación del 91.32, es *Primero*; pero, realmente los tres modelos son casi iguales por lo que lo seleccionamos a *Primero* como nuestro modelo.

**NOTA 6:** En **su experimento**, para seleccionar su modelo final de entre los diferentes modelos obtenidos, debe utilizar criterios de aproximación y de complejidad, (la mejor aproximación en simulación a los resultados experimentales; con la menor complejidad del modelo y que este llene las expectativas de la dinámica del modelo teórico, si este existe).

Paso 7) Para exportar el modelo lo arrastramos hasta la posición *To workspace* en la ventana del *ident*. Luego en el *workspace* de Matlab, podemos simular para efectos de verificación o extraer las funciones de transferencia de entrada-salida. Para extraer la función de transferencia en versiones de Matlab anteriores a la v2008b, usamos la sentencia `modelo = zpk(Primero)`; donde "modelo" es el nombre que le asignamos al resultado; en versiones de Matlab más recientes, usamos una variable intermedia, por ejemplo **M**, así: `M = zpk(Primero)`; y luego extraemos por ejemplo la primera función de transferencia del modelo con `pendulo = M(1,1)`; y luego la segunda función de transferencia con `brazo = M(2,1)`;

**NOTA 7:** En este caso, extraer las funciones de transferencia de esta forma, **no se contempla** el hecho de que internamente las **variables estén acopladas entre sí** y debe ser evaluada su conveniencia como modelos SISO para representar al sistema.

**NOTA 8:** En otros casos, eventualmente, si el orden estimado para la planta es muy grande, será necesario reducirlo [4], cambiando polos y ceros no dominantes por sus ganancias estáticas u otra forma adecuada; por ejemplo, con las funciones *balreal* y *modred* de Matlab.

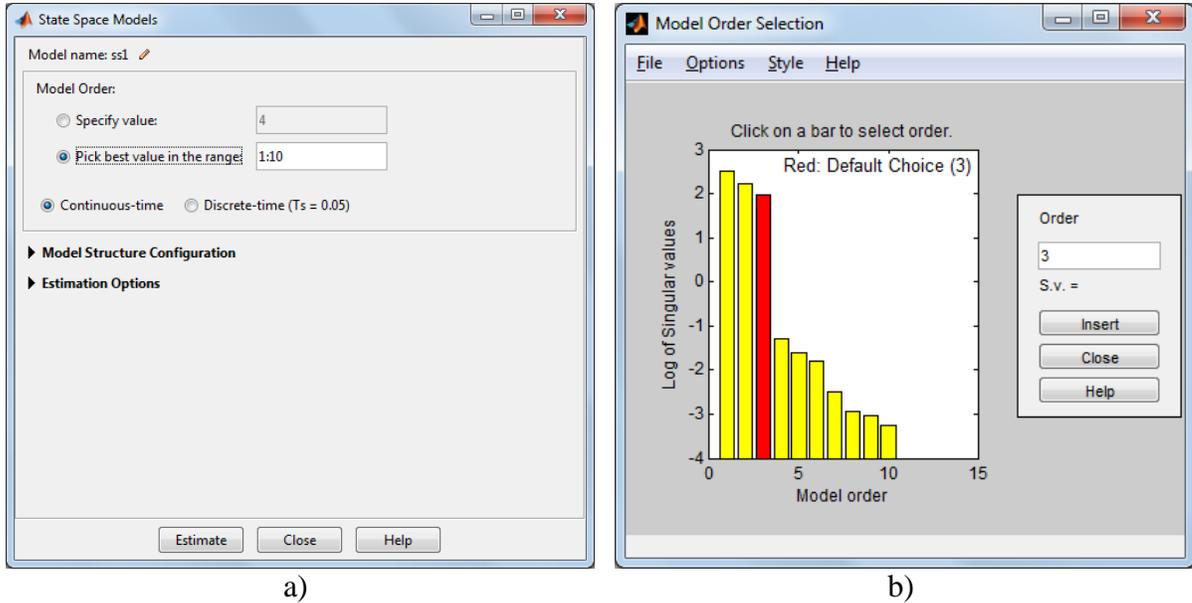


Figura 7: Ventanas del proceso de obtención de modelos en variables de estado del *ident* con: a) la selección de parámetros y b) los modelos obtenidos para el ejemplo.

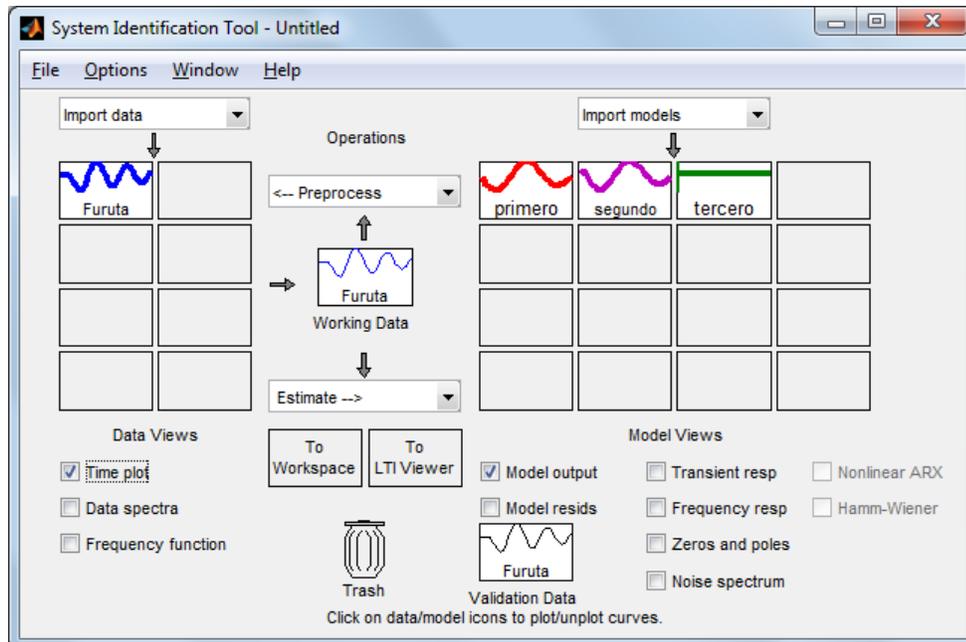


Figura 8: Ventana del *ident* con el objeto de experimento y los modelos obtenidos para el ejemplo. **¡Sus resultados pueden ser diferentes! Los nombres de los modelos obtenidos no tienen que ver con el orden del modelo; sino, con la secuencia en la que fueron obtenidos.**

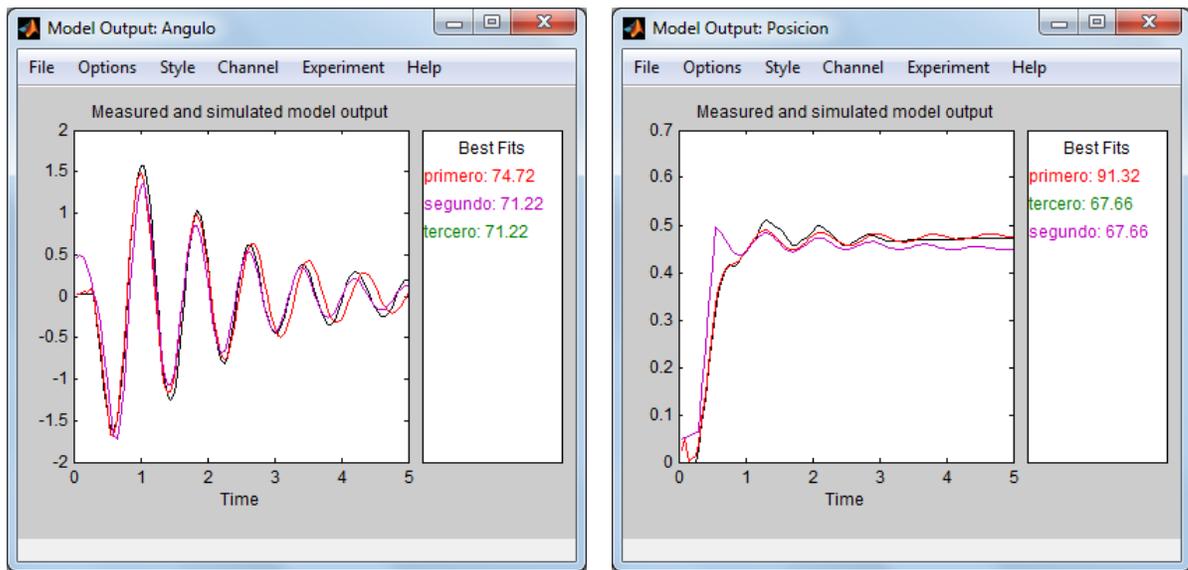


Figura 9: Comparación de modelos con *Model output* para el ejemplo.  
**¡Sus resultados serán diferentes!**

Una verificación del modelo obtenido se debe realizar también graficando, con la función *plot* de Matlab, los resultados experimentales junto a resultados de simulación producidos con la función *lsim*. La figura 10 muestra esta comparación realizada con las sentencias mostradas en el recuadro a continuación, que además sirven para documentar la gráfica:

```
M = ss(Primero);
cero = zeros(length(Entrada),1);
X0 = [0; 0; 0; 0]; % vector de estado inicial en cero (su longitud depende
de la cantidad de variables de estado del modelo seleccionado)
[Y,T,X] = lsim(M, [Entrada], Tiempo, X0);
Entrada2 = Entrada / 6; % ajusta el valor, solo para efectos visuales
plot(T, Entrada2, T, Posicion, T, Angulo, T, Y)
title('Verificación de la respuesta del modelo')
xlabel('Tiempo [s]')
ylabel('Amplitud')
legend('Entrada', 'Posición', 'Angulo', 'Angulo modelo', 'Posición modelo')
```

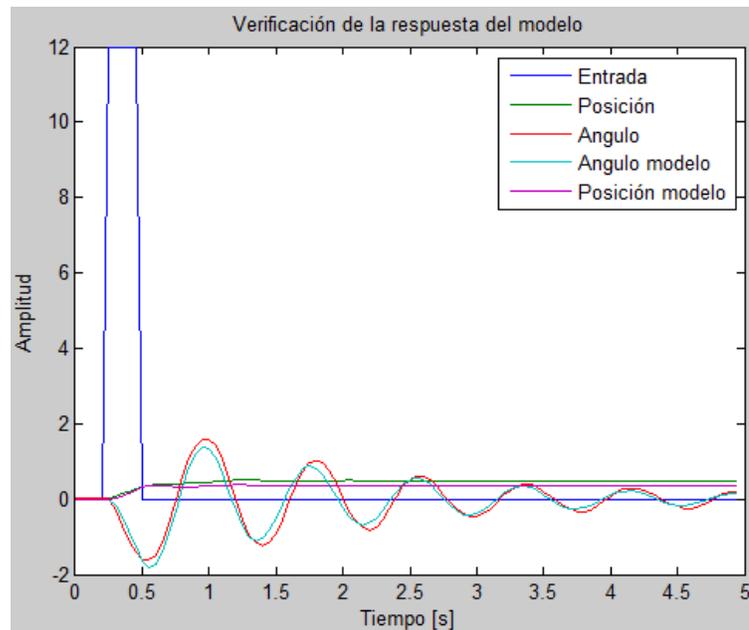


Figura 10: Verificación del modelo la grúa utilizado en este ejemplo.

Los resultados deben mostrarse al profesor durante la clase y la documentación debe enviarse **a más tardar** a la medianoche del día en que se cumple una semana después de realizado el experimento, el **18 o el 21 de agosto 2016**, al correo electrónico [einteriano@tec.ac.cr](mailto:einteriano@tec.ac.cr), con todos los documentos y archivos utilizados para resolver el problema en un ZIP con el nombre (letra) del equipo.

## Referencias

- [1] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.Models.pdf>
- [2] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.6ModeladoEstocastico.pdf>
- [3] <http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/3.7Identificacion%20de%20sistemas.PDF>
- [4] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.6.1EjemploModeladoEstocastico.pdf>
- [5] [http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1\\_Grua\\_II\\_2015.pdf](http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1_Grua_II_2015.pdf)

## Aclaración:

El presente instructivo es una adaptación autorizada de la versión original del Proyecto Corto 1 elaborada por el Msc. Eduardo Interiano Salguero, el cual ha puesto en práctica con sus alumnos en varias oportunidades. [11]

Víctor Ramírez Rojas

EIS/VRR/vrr  
2016

### 6.3. Proyecto Corto 1 SiDiCo (Motor SIMO)



INSTITUTO TECNOLÓGICO DE COSTA RICA

**II SEMESTRE 2016**

ESCUELA DE INGENIERIA ELECTRÓNICA

CURSO: EL-5409 LABORATORIO DE CONTROL AUTOMÁTICO

MEDIO: Proyecto corto 1 (Motor SiDiCo)

FECHA: 18 de julio de 2016

PROF: Ing. Eduardo Interiano

## PROYECTO CORTO 1

**Tema: Obtención del modelo empírico SIMO del sistema motor CD SiDiCo.**

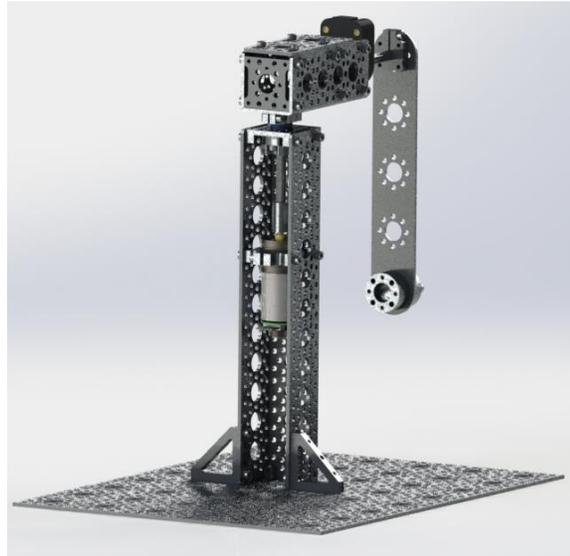


Figura 1: Péndulo de Furuta SiDiCo

**Recursos:** Sistema Didáctico de Control “SiDiCo” con el software V1.0 y planta Péndulo invertido de Furuta V1.0 (con motor CD de imán permanente de 12V con caja reductora 46.85:1; sensor de posición angular E5 de 4000cpr en el péndulo y un sensor de posición angular integrado en el giro del motor de 2248.86cpr). Computador con Matlab R2013b o superior e *ident* instalado.

#### **Descripción del trabajo:**

Debe utilizar SiDiCo V1.0 para capturar los datos de un experimento para obtener las respuestas de velocidad y de corriente de un motor de CD ante estímulos pseudoaleatorios de tensión de

armadura. Posteriormente debe utilizar la herramienta *ident* de Matlab para identificar y estimar modelos adecuados para el motor y finalmente debe verificar los modelos así obtenidos.

## Parte I: El experimento

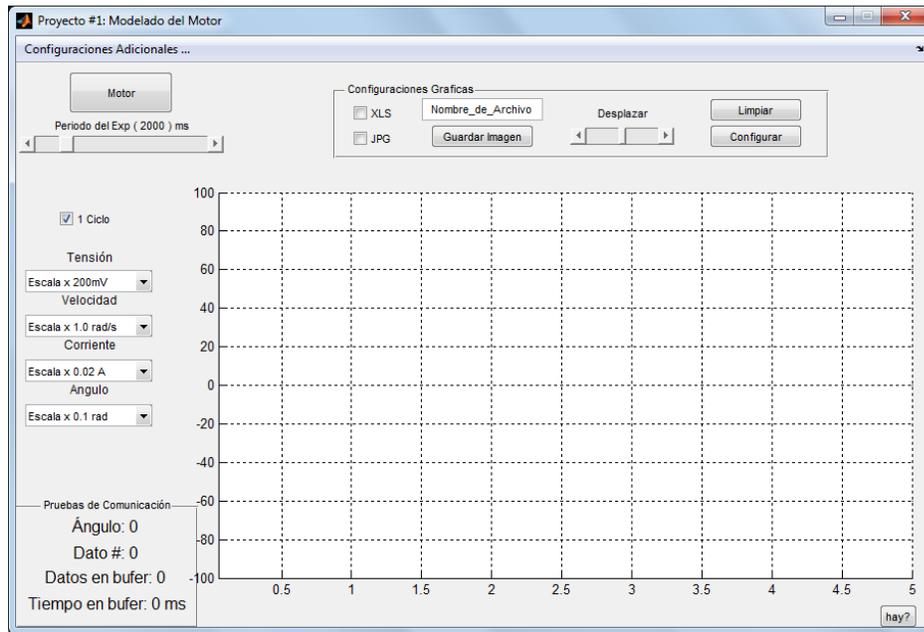


Figura 2: Modulo Modelado de Motor (SiDiCo V1.0)

Paso 1) Conectar la Planta Péndulo Invertido de Furuta al software SiDiCo V1.0

Paso 2) Ingresar al Módulo “Modelado del Motor”

Paso 3) Antes de iniciar cada medición asegúrese que el área está despejada para el libre movimiento del brazo y péndulo.

Paso 4) Se sugiere ajustar las propiedades del programa (esquina superior izquierda) como se muestra en la Tabla 1. Usted puede modificar estos ajustes para adecuarlos a su experimento si es necesario. Para el primer experimento se sugiere conservar los valores por defecto.

Paso 5) Se sugiere ajustar el largo del pulso a aplicar al motor en “Periodo del Exp” y la extensión del experimento en “1 Ciclo” ubicadas bajo el botón principal de la pantalla como se muestra en la imagen 2. Usted puede modificar estos ajustes para adecuarlos a su experimento si es necesario. Para el primer experimento se sugiere conservar los valores por defecto.

Paso 6) Establezca un “Nombre\_de\_Archivo” y seleccione “XLS” y “JPG” para que el futuro experimento sea guardado.

Paso 7) Ejecute el experimento “Motor” y espere hasta que la planta y programa se detengan completamente.

Paso 8) Si durante el experimento los “Datos en Buffer” crecen, se recomienda aumentar el valor de muestreo o periodo de la planta, esto debido a que la velocidad de procesamiento del computador en uso es menor al deseado. Caso contrario, si el valor no crece, puede disminuir estos valores para obtener más detalles en la gráfica. De ser necesario repita el experimento.

Tabla 1: Ajustes del Programa

Función	Propiedades	Comentario
Periodo T en la planta	10 ms	El periodo de operación de la planta puede variar entre 1 y 255ms, sin embargo, si se escogen valores pequeños satura el buffer o altos se pierden detalles de la gráfica
Angulo de Op	60 grados	Se recomienda ajustar el Angulo en el máximo permitido para que el experimento no se vea interrumpido
# en X	100 muestras	En 100 muestras el experimento despliega los detalles necesarios, si se desea ver más se puede ampliar hasta 550 muestras, o deshabilitar la opción “1 ciclo” de la pantalla principal
Actualiza c/	5 T	Regula el tiempo de diezmo, se puede reducir dependiendo de la velocidad de procesamiento del computador. El objetivo es no saturar el USB.
Máximo en Y	100 líneas	En 100 muestras el experimento despliega los detalles necesarios, si se desea ver más se puede ampliar hasta 365 muestras
Otros	No relevantes	No es necesario ajustar los demás parámetros, puesto que no se emplean en este experimento y, por lo tanto, no generan variante.

## Parte II: Estimación y verificación del modelo del motor

**Objetivo:** Estimar, a partir de los resultados experimentales de entrada-salida de la planta, los modelos numéricos de velocidad y corriente de un motor de CD.

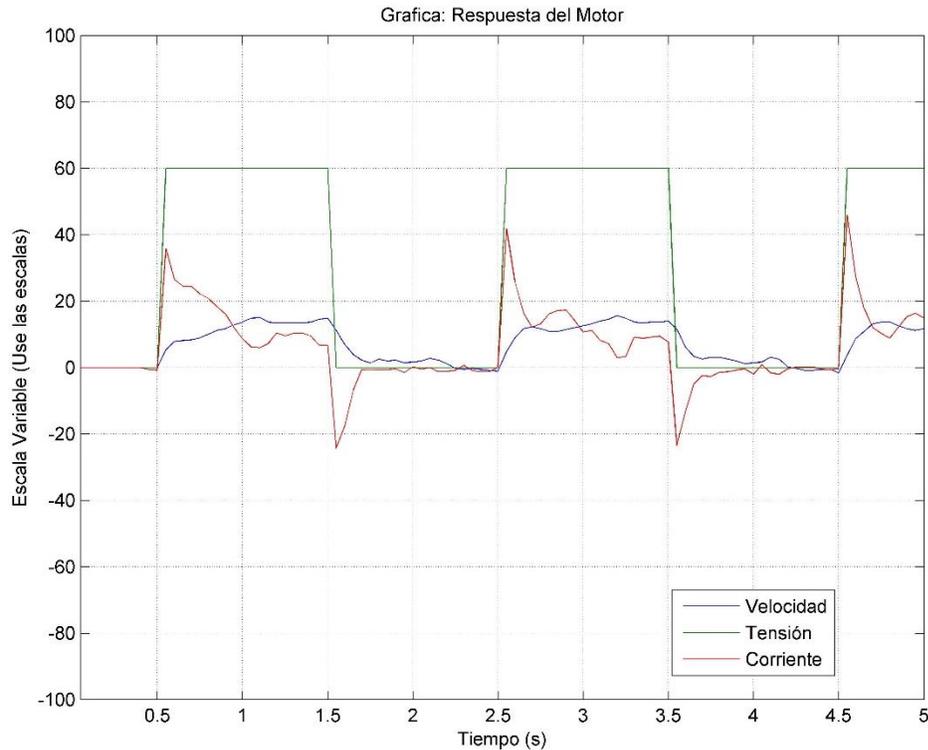


Figura 3: Captura de la entrada [V]; de la velocidad angular [rad/s]; y la corriente [A] del motor de SiDiCo mostrado en la figura 1.

**Procedimiento:**

**NOTA 1:** Antes de iniciar, opcionalmente puede ejecutar *iddemo* en Matlab, seleccione el tipo de demostración 1 y luego seleccione la demostración de la interfaz gráfica de usuario para estimación (opción 1) o consulte [3] y [4].

Paso 1) Identifique y estime, con ayuda de *ident* de Matlab, varios modelos de función de transferencia para la velocidad y corriente del motor CD (Motor(s)). Seleccione el modelo con el mejor ajuste a los datos y valide ese modelo seleccionado contra los datos experimentales.

**NOTA 2:** En el ejemplo aquí mostrado se utilizan otros valores que no necesariamente son iguales a los que usted deberá de obtener, solo es una guía que muestra los procedimientos, y no los valores a conseguir. **Usted debe utilizar los datos de su experimento que corresponden a un caso SIMO.**

**NOTA 3:** Antes de iniciar el procesamiento de los datos verifique la **consistencia del separador decimal** de sus datos con las herramientas a usar; esto es, si las herramientas están en inglés, el separador decimal debe ser punto (.); si están en español, el separador decimal debe ser coma (,). **De preferencia use un editor de texto simple** para esta labor, (*Notepad* por ejemplo).

Paso 2) Importe a Matlab los datos contenidos en el archivo .XLS guardado en la parte I; para

ello, inicie Matlab y en vaya al menú *Home\Import Data...*, seleccione el archivo y luego oprima *Open*, en la siguiente ventana seleccione *Column vectors ...*, y en *Range* asegúrese que inicie en la columna B como se muestra en la figura 4 (el largo de la selección dependerá del largo de su experimento) y luego oprima *Import Selection*. Los datos están ahora disponibles como vectores en el *workspace* de Matlab.

Nombre de Archivo Modelado Motor SiDiCo 152531					
	Dato	Tiempo	Entrada	Velocidad	Corriente
	Number	Number	Number	Number	Number
1	Dato	Tiempo	Entrada	Velocidad	Corriente
2	1	0.0500	0	0	0
3	2	0.1000	0	0	0
4	3	0.1500	0	0	0
5	4	0.2000	0	0	0
6	5	0.2500	0	0	0
7	6	0.3000	0	0	0
8	7	0.3500	0	0	0
9	8	0.4000	0	0	0
10	9	0.4500	0	0	-0.0035
11	10	0.5000	0	0	-6.5106e-05
12	11	0.5500	12	5.5880	0.8147
13	12	0.6000	12	8.3820	0.5506
14	13	0.6500	12	8.6610	0.5350
15	14	0.7000	12	8.6610	0.5387
16	15	0.7500	12	9.2200	0.4779
17	16	0.8000	12	10.8960	0.4161

Figura 4: Importar datos en forma de vectores al *workspace* de Matlab.

Paso 3) Cree en Matlab un objeto de datos para identificación, en este caso llamado 'Motor', usando las siguientes sentencias:

**NOTA 4:** Antes de crear el objeto de datos asegúrese de filtrar el ruido de los datos (un filtro de media móvil es muy práctico para esto) y de linealizar los datos si es necesario (restar e valor inicial), de lo contrario, el ruido y la no linealidad de los datos evitarán que obtenga modelos apropiados con la exactitud suficiente. Para este experimento SiDiCo ya guarda los datos filtrados y linealizados, sin embargo, no descarte la necesidad de usar estas herramientas si con criterio lo considera necesario.

**NOTA 5:** Asegúrese que el periodo de muestreo sea constante, revisando el vector de **Tiempo**; y en Matlab use la orden *format shortg* para poder ver los datos en formato sin redondeos.

```
deltaT = Tiempo(2)-Tiempo(1);  
Motor = iddata([Velocidad, Corriente], [Entrada], deltaT);  
Motor.inputname = {'Entrada'};  
Motor.outputname = {'Velocidad'; 'Corriente'};
```

Paso 4) Luego en el *workspace* de Matlab ejecute la orden *ident*; espere unos segundos y se abrirá la ventana del *ident*, en la cual debe seleccionar *Import data>Data object ...*, llene los campos de la ventana con los datos correspondientes **a su experimento**: nombre del objeto de datos, creado en el paso anterior, que contiene los datos experimentales como se muestra en la figura 5 y oprima *Import*. Una vez hecho lo anterior puede cerrar esa ventana.

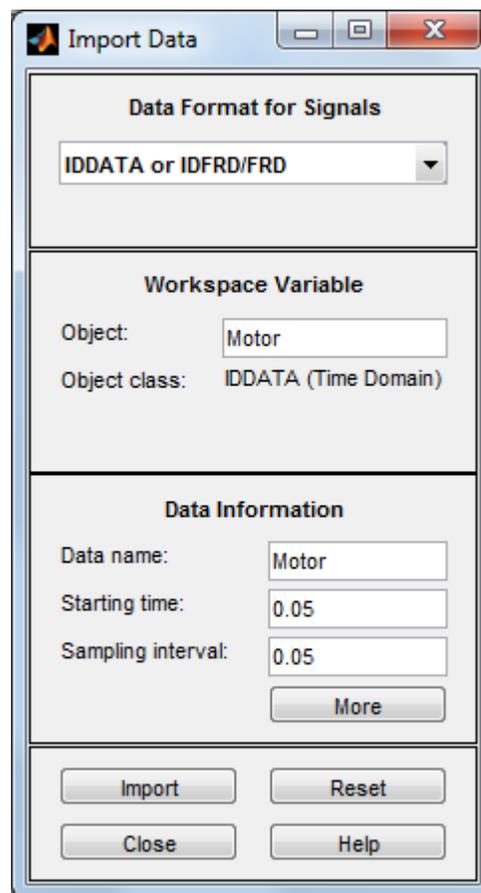


Figura 5: Ventana para importar datos al *ident*. Ponga los datos de **su experimento**.

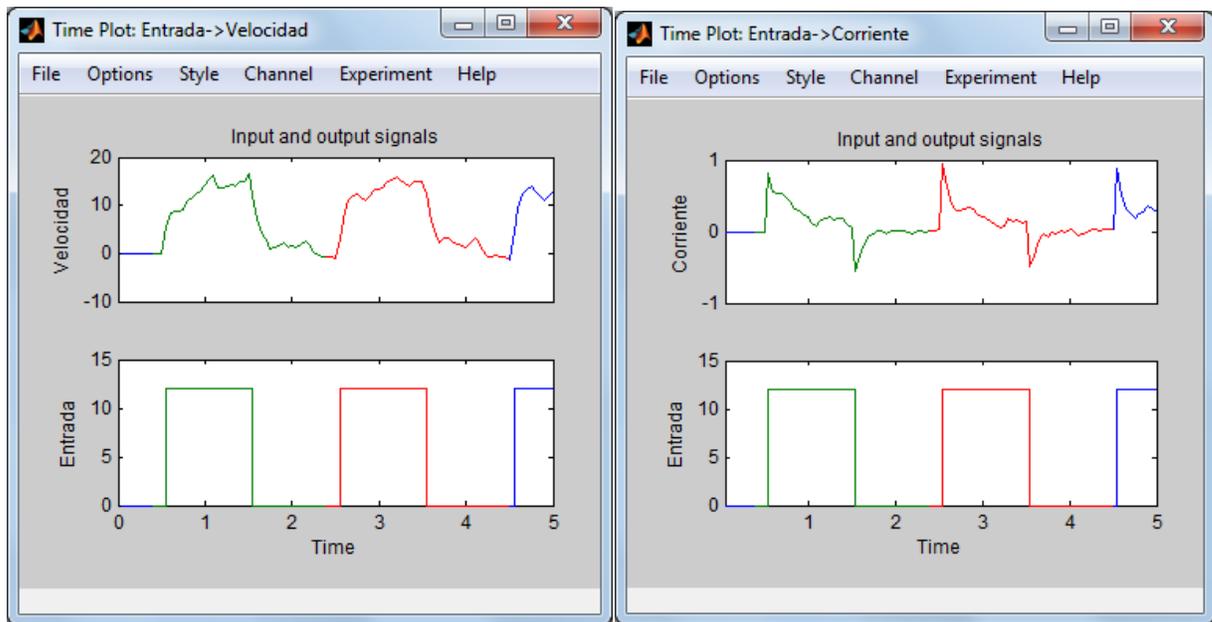


Figura 6: Visualización de los datos experimentales importados en *ident*.

En este ejemplo, en la ventana principal del *ident*, figura 8, debe tener ahora un objeto, llamado **Motor**, debajo de *Import data*. Este objeto contiene todos los datos del experimento, los cuales pueden visualizarse en dos partes. Usaremos para ello la casilla *Time plot*, que abrirá una ventana en la cual podemos seleccionar el canal (*Channel*) que deseamos ver, tal como se muestra en la figura 6.

Paso 5) En la ventana del *ident*, figura 8, debe tener ahora un objeto, llamado **Motor** en este ejemplo. Este objeto contiene todos los datos del experimento, el cual se dividirá en dos partes, una para estimar el modelo ("**Motore**") y otra para validar los datos ("**Motorv**"). La figura 6 muestra esta selección con los colores verde y rojo respectivamente. Se usa para la selección de los datos la operación *Preprocess|Select range*, que abrirá una ventana en la cual con el ratón se crea un rectángulo alrededor de los datos a seleccionar y luego se oprime *Insert*; repitiendo para cada rango a crear; al terminar de insertar los segmentos seleccionados podemos cerrar esta ventana. Al finalizar se debe tener tres objetos en la ventana del *ident*, el objeto original **Motor** y dos más que llamaremos **Motore** y **Motorv**. Figura 8.

Se arrastra y coloca el segundo objeto (**Motore**, color verde) en la posición llamada *Working Data* y el tercer objeto (**Motorv**, color rojo) en la posición llamada *Validation Data*.

Paso 6) En la ventana principal del *ident* seleccionamos *Estimate → State Space models...* y en esta ventana, ya que nuestro modelo tiene una entrada y dos salidas (SIMO) escogeremos una estructura de variables de estado y oprimiendo la selección *Pick best value in the range* buscaremos modelos desde orden 1 a 10 como se muestra en la figura 7a. Una vez hecho lo anterior se oprime el botón *Estimate*, el cual hará que se abra la ventana de selección del orden del modelo, como se muestra en la figura 7b. Aquí se seleccionan los modelos que tengan valores singulares pequeños, sin aumentar excesivamente el orden del modelo.

Paso 7) Para validar nuestros modelos obtenidos, podemos comparar su salida ante el estímulo original del experimento seleccionando la opción **Model output** que nos presenta el porcentaje de aproximación entre nuestro modelo y la respuesta original. Como puede observarse en la figura 8, para la salida **Velocidad**, el modelo **Segundo** posee la mejor aproximación a los datos experimentales (83.14) y para la salida **Corriente** el mejor modelo, con una aproximación del 45.85, también es **Segundo**; aunque ambos modelos difieren muy poco. Pruebe **varios métodos**; además pruebe también **filtrando** los datos para eliminar el ruido, un filtro de media móvil en **Excel** (promedio de la muestra actual con las 7 anteriores) puede ser el adecuado.

**NOTA 6:** En su experimento, para seleccionar su modelo final de entre los diferentes modelos obtenidos, debe utilizar criterios de aproximación y de complejidad, (la mejor aproximación en simulación a los resultados experimentales; con la menor complejidad del modelo y que este llene las expectativas de la dinámica del modelo teórico, si este existe). En este caso, como puede observarse en la figura 3, el motor se comporta como un sistema de primer orden, sin sobreimpulso; así que no es conveniente obtener modelos de órdenes altos, por ello después de evaluar las funciones de transferencia se escoge el modelo **Segundo** por tener un orden adecuado.

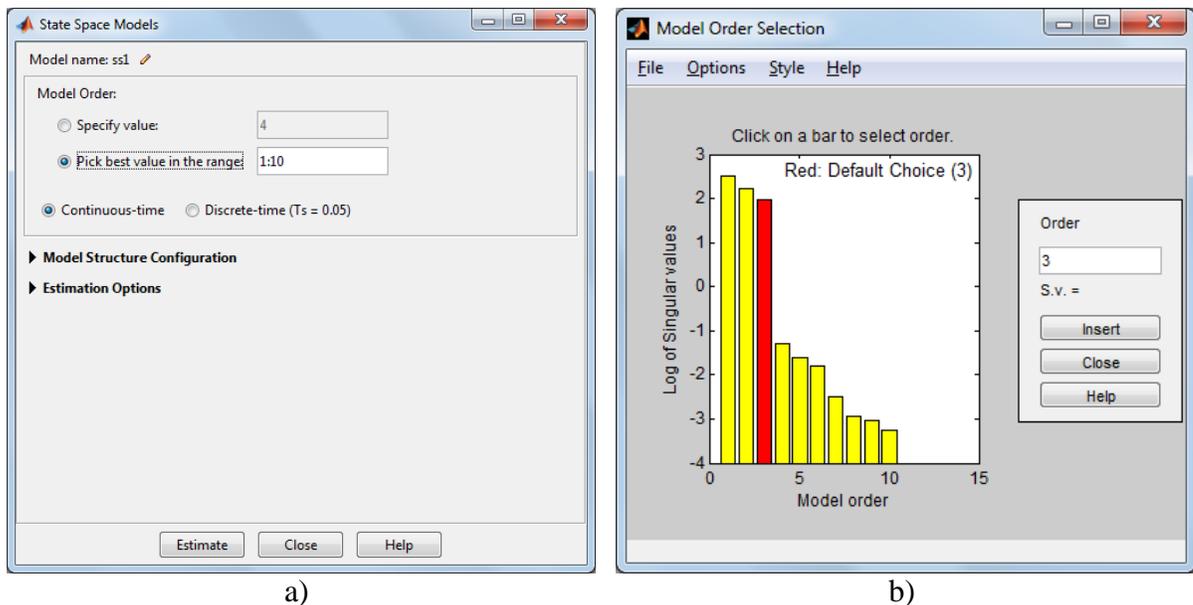


Figura 7: Ventanas del proceso de obtención de modelos en variables de estado del *ident* con: a) la selección de parámetros y b) los modelos obtenidos para el ejemplo.

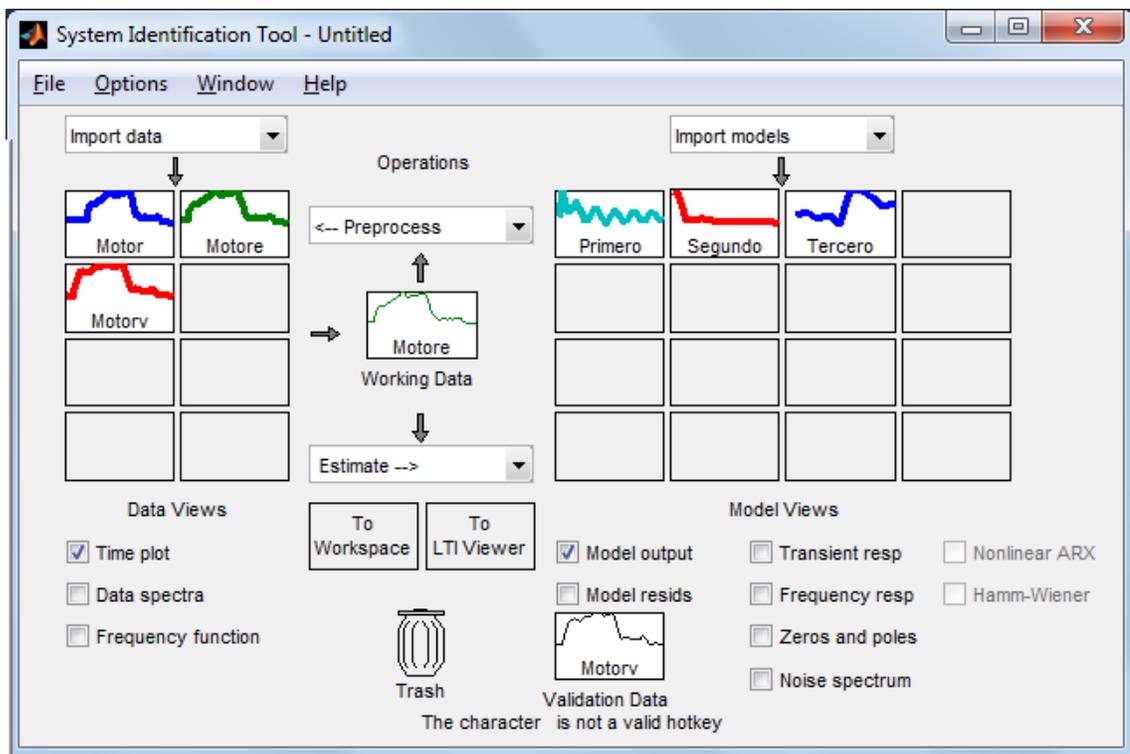


Figura 8: Ventana del *ident* con los objetos de experimento y los modelos obtenidos

Paso 8) Para exportar el modelo seleccionado se arrastra hasta la posición *To workspace* en la ventana del *ident*. Luego en el *workspace* de Matlab, se puede extraer la función de transferencia de entrada-salida. Para extraer la función de transferencia se usa una variable intermedia, por ejemplo **M**, así: **M = zpk(Segundo)**; y luego se extrae por ejemplo la primera función de transferencia del modelo con **modelovelocidad = M(1,1)**; y luego la segunda función de transferencia con **modelocorriente = M(2,1)**;

**NOTA 7:** En este caso, al extraer las funciones de transferencia de esta forma, **no se contempla** el hecho de que internamente las **variables estén acopladas entre sí** y debe ser evaluada su conveniencia como modelos SISO para representar al sistema. Puede ser preferible extraer el modelo SIMO en variables de estado con **M = ss(Segundo)**.

**NOTA 8:** En otros casos, eventualmente, si el orden estimado para la planta es muy grande, será necesario reducirlo [4], cambiando polos y ceros no dominantes por sus ganancias estáticas u otra forma adecuada; por ejemplo, con las funciones *balreal* y *modred* de Matlab.

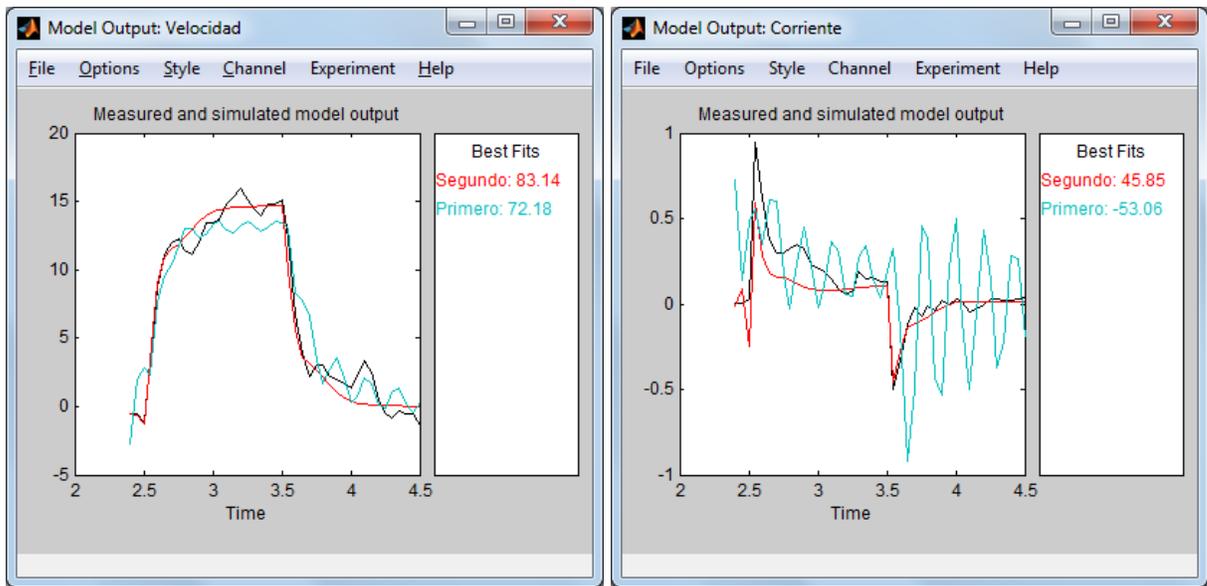


Figura 9: Comparación de modelos con *Model output*

Una verificación del modelo obtenido se puede realizar también graficando, con la función *plot* de Matlab, los resultados experimentales junto a resultados de simulación producidos con la función *lsim*. La figura 11 muestra esta comparación realizada con las sentencias mostradas en el recuadro a continuación, que además sirven para documentar la gráfica:

```

y = lsim(modelocorriente,Entrada,Tiempo);
Entrada2 = Entrada / 6; % ajusta el valor, solo para efectos visuales
plot(Tiempo, Entrada2, Tiempo, Corriente, Tiempo, y);
grid;
title('Verificación de la respuesta del modelo de corriente');
xlabel('Tiempo [s]');
ylabel('Corriente');
legend('Entrada', 'Corriente', 'Modelo');

```

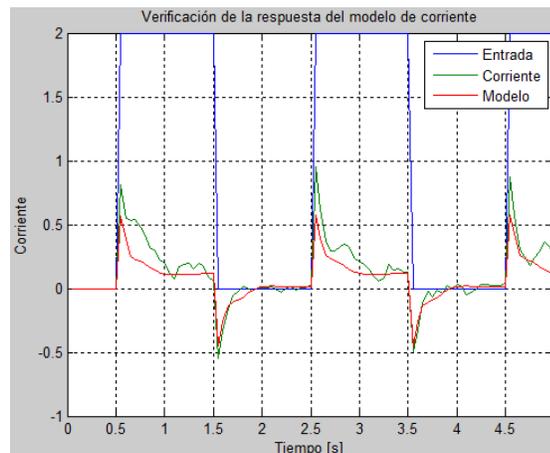


Figura 9: Verificación del modelo de corriente del motor

Los resultados deben mostrarse al profesor durante la clase y la documentación debe enviarse a más tardar a la medianoche del día en que se cumple una semana después de realizado el experimento, esto es el **11 o el 15 de marzo 2016**, al correo electrónico [einteriano@tec.ac.cr](mailto:einteriano@tec.ac.cr), con todos los documentos y archivos utilizados para resolver el problema en un ZIP con el nombre (letra) del equipo.

### Referencias

- [1] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.Models.pdf>
- [2] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.6ModeladoEstocastico.pdf>
- [3] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/Identificacion de sistemas.PDF>
- [4] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.6.1EjemploModeladoEstocastico.pdf>
- [5] [http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1\\_I\\_2016.pdf](http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1_I_2016.pdf)

### Aclaración:

El presente instructivo es una adaptación autorizada de la versión original del Proyecto Corto 1 elaborada por el Msc. Eduardo Interiano Salguero, el cual ha puesto en práctica con sus alumnos en varias oportunidades. [11]

Víctor Ramírez Rojas

EIS/VRR/vrr  
2016

## 6.4. Proyecto Corto 1 SiDiCo (Péndulo SISO)



INSTITUTO TECNOLÓGICO DE COSTA RICA

**II SEMESTRE 2016**

ESCUELA DE INGENIERIA EN ELECTRÓNICA

CURSO: EL-5409 LABORATORIO DE CONTROL AUTOMÁTICO

MEDIO: Proyecto corto 1 (Furuta SiDiCo)

FECHA: 18 de julio de 2016

PROF: Ing. Eduardo Interiano

# PROYECTO CORTO 1

**Tema: Obtención del modelo empírico de un Péndulo (SISO).**

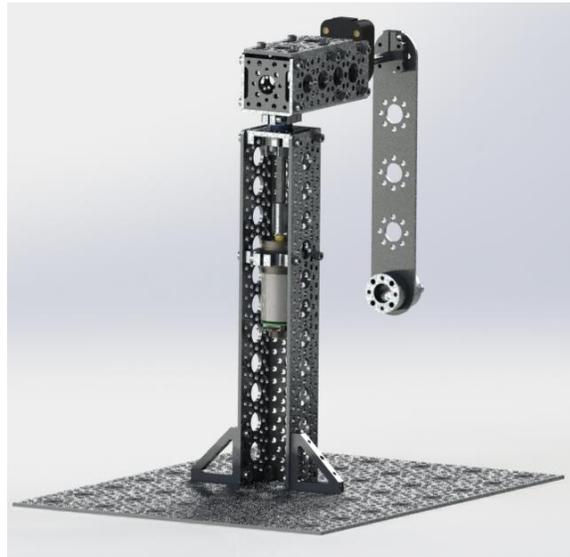


Figura 1: Péndulo de Furuta SiDiCo

**Recursos:** Sistema Didáctico de Control “SiDiCo” con el software V1.0 y planta Péndulo invertido de Furuta V1.0 (con motor CD de imán permanente de 12V con caja reductora 46.85:1; sensor de posición angular E5 de 4000cpr en el péndulo y un sensor de posición angular integrado en el giro del motor de 2248.86cpr). Computador con Matlab R2013b o superior e *ident* instalado.

### Descripción del trabajo:

Debe utilizar SiDiCo V1.0 para capturar los datos de un experimento para obtener la respuesta de posición angular del péndulo ante un estímulo pseudoaleatorio del brazo donde se encuentra instalado. Posteriormente debe utilizar la herramienta *ident* de Matlab para identificar y estimar

modelos adecuados para el péndulo y finalmente debe verificar los modelos así obtenidos.

## Parte I: El experimento

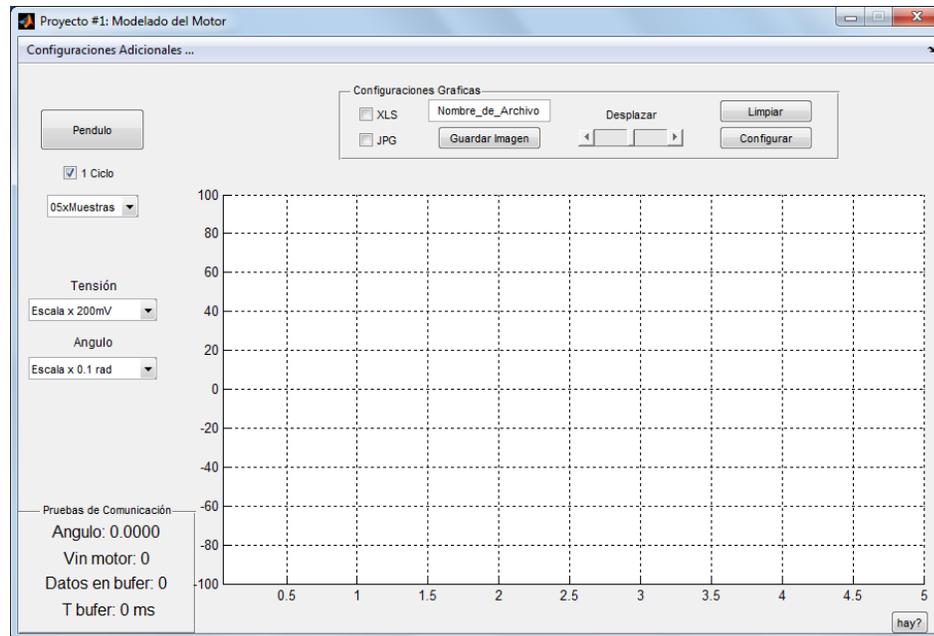


Figura 2: Módulo Modelado del Péndulo (SiDiCo V1.0)

Paso 1) Conectar la Planta Péndulo Invertido de Furuta al software SiDiCo V1.0

Paso 2) Ingresar al Módulo “Modelado Péndulo”

Paso 3) Antes de iniciar cada medición asegúrese que el área está despejada para el libre movimiento del brazo y péndulo.

Paso 4) Se sugiere ajustar las propiedades del programa (esquina superior izquierda) como se muestra en la Tabla 1. Usted puede modificar estos ajustes para adecuarlos a su experimento si es necesario. Para el primer experimento se sugiere conservar los valores por defecto.

Paso 5) Se sugiere ajustar el largo del pulso a aplicar al motor en “cantidad de muestras” y la extensión del experimento en “1 Ciclo” ubicadas bajo el botón principal de la pantalla como se muestra en la imagen 2. Usted puede modificar estos ajustes para adecuarlos a su experimento si es necesario. Para el primer experimento se sugiere conservar los valores por defecto.

Paso 6) Establezca un “Nombre\_de\_Archivo” y seleccione “XLS” y “JPG” para que el futuro experimento sea guardado.

Paso 7) Ejecute el experimento “Péndulo” y espere hasta que la planta y programa se detengan completamente.

Paso 8) Si durante el experimento los “Datos en Buffer” crecen, se recomienda aumentar el valor de muestreo o periodo de la planta, esto debido a que la velocidad de procesamiento del computador en uso es menor al deseado. Caso contrario, si el valor no crece, puede disminuir estos valores para obtener más detalles en la gráfica. De ser necesario repita el experimento.

Tabla 1: Ajustes del Programa

Función	Propiedades	Comentario
Periodo T en la planta	10 ms	El periodo de operación de la planta puede variar entre 1 y 255ms, sin embargo, si se escogen valores pequeños satura el buffer o altos se pierden detalles de la gráfica
Angulo de Op	60 grados	Se recomienda ajustar el Angulo en el máximo permitido para que el experimento no se vea interrumpido
# en X	100 muestras	En 100 muestras el experimento despliega los detalles necesarios, si se desea ver más se puede ampliar hasta 550 muestras, o deshabilitar la opción “1 ciclo” de la pantalla principal
Actualiza c/	5 T	Regula el tiempo de diezmo, se puede reducir dependiendo de la velocidad de procesamiento del computador. El objetivo es no saturar el USB.
Máximo en Y	100 líneas	En 100 muestras el experimento despliega los detalles necesarios, si se desea ver más se puede ampliar hasta 365 muestras
Otros	No relevantes	No es necesario ajustar los demás parámetros, puesto que no se emplean en este experimento y, por lo tanto, no generan variante.

## Parte II: Estimación y verificación del modelo SISO del péndulo

**Objetivo:** Estimar, a partir de los resultados experimentales de entrada-salida de la planta, el modelo numérico de la posición de un péndulo libre.

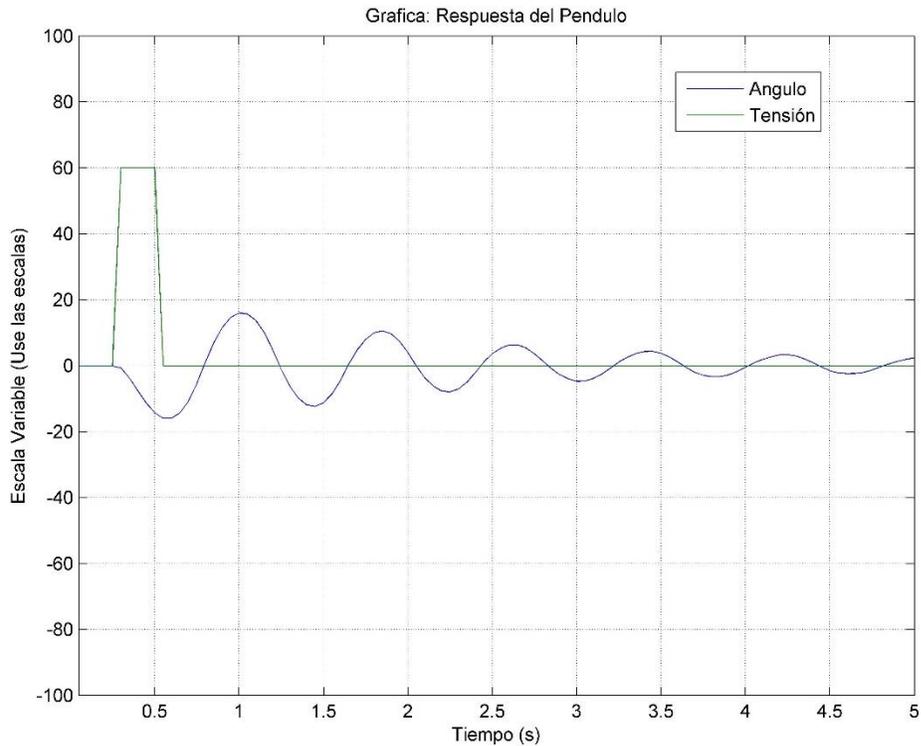


Figura 3: Captura de la entrada [V] y de la posición angular del péndulo [rad] de SiDiCo mostrado en la figura 1.

**Procedimiento:**

**NOTA 1:** Antes de iniciar el procesamiento de los datos experimentales, puede ejecutar *iddemo* en Matlab, seleccione el tipo de demostración 1 y luego seleccione la demostración de la interfaz gráfica de usuario para estimación (opción 1) o consulte [3], [4].

Paso 1) Identifique y estime, con ayuda de *ident* de Matlab, varios modelos en variables de estado para el péndulo. Seleccione el modelo con el mejor ajuste a los datos experimentales y valide ese modelo seleccionado en simulación contra los datos experimentales.

**NOTA 2:** En el ejemplo aquí mostrado se utilizan otros valores que no necesariamente son iguales a los que usted deberá de obtener, solo es una guía que muestra los procedimientos, y no los valores a conseguir. **Usted debe utilizar los datos de su experimento que corresponden a un caso SISO.**

**NOTA 3:** Antes de iniciar el procesamiento de los datos verifique la **consistencia del separador decimal** de sus datos con las herramientas a usar; esto es, si las herramientas están en inglés, el separador decimal debe ser punto (.); si están en español, el separador decimal debe ser coma (,). **De preferencia use un editor de texto simple** para esta labor, (*Notepad* por ejemplo).

Paso 2) Importe a Matlab los datos contenidos en el archivo .XLS guardado en la parte I; para ello, inicie Matlab y en vaya al menú *Home\Import Data...*, seleccione el archivo y luego

oprime **Open**, en la siguiente ventana seleccione **Column vectors ...**, y en **Range** asegúrese que inicie en la columna B como se muestra en la figura 4 (el largo de la selección dependerá del largo de su experimento) y luego oprime **Import Selection**. Los datos están ahora disponibles como vectores en el *workspace* de Matlab.

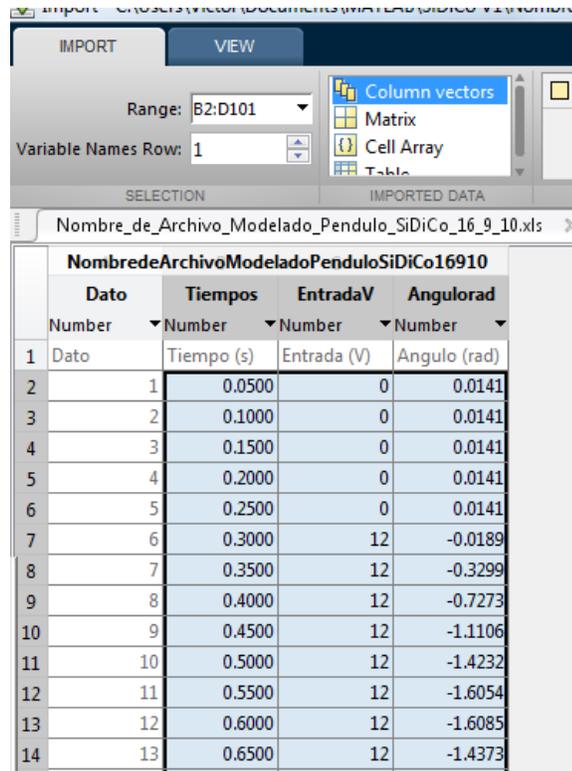


Figura 4: Importar datos en forma de vectores al *workspace* de Matlab.

Paso 3) Cree en Matlab un objeto de datos para identificación, en este caso llamado 'Pendulo', usando las siguientes sentencias:

**NOTA 4:** Antes de crear el objeto de datos asegúrese de filtrar el ruido de los datos (un filtro de media móvil es muy práctico para esto) y de linealizar los datos (restar el valor inicial), de lo contrario, el ruido y la no linealidad de los datos evitarán que obtenga modelos apropiados y con la exactitud suficiente. Para este experimento SiDiCo ya guarda los datos filtrados y linealizados, sin embargo, no descarte la necesidad de usar estas herramientas si con criterio lo considera necesario.

**NOTA 5:** Asegúrese que el periodo de muestreo sea constante, revisando el vector de **Tiempo**; y en Matlab use la orden **format shortg** para poder ver los datos en formato sin redondeos.

```
deltaT = Tiempo(2)-Tiempo(1);
Pendulo = iddata([Angulo], [Entrada], deltaT);
Pendulo.inputname = {'Entrada'};
Pendulo.outputname = {'Angulo'};
```

Paso 4) Luego en el *workspace* de Matlab ejecute la orden *ident*; espere unos segundos y se abrirá la ventana del *ident*, en la cual debe seleccionar *Import data>Data object ...*, llene los campos de la ventana con los datos correspondientes **a su experimento**: nombre del objeto de datos, creado en el paso anterior, que contiene los datos experimentales como se muestra en la figura 5 y oprima *Import*. Una vez hecho lo anterior puede cerrar esa ventana.

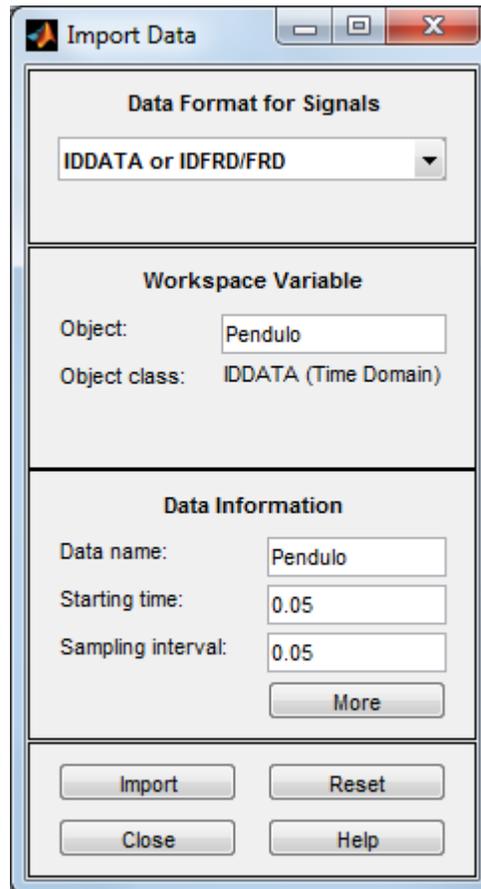


Figura 5: Ventana para importar datos al *ident*. Ponga los datos de **su experimento**.

En este ejemplo, en la ventana principal del *ident*, figura 8, debe tener ahora un objeto, llamado **Pendolo**, debajo de *Import data*. Este objeto contiene todos los datos del experimento, los cuales pueden visualizarse usando para ello la casilla *Time plot*, que abrirá una ventana, tal como se muestra en la figura 6.

Paso 5) En la ventana principal del *ident* seleccionamos *Estimate* → *State Space models...* y en esta ventana, ya que nuestro modelo tiene una entrada y dos salidas (SIMO) escogeremos una estructura de variables de estado y oprimiendo la selección *Pick best value in the range* buscaremos modelos desde orden 1 a 10 como se muestra en la figura 7a. Una vez hecho lo anterior oprimimos el botón *Estimate*, el cual hará que abra la ventana de selección del orden del modelo, como se muestra en la figura 7b. Aquí seleccionaremos modelos que tengan valores singulares pequeños, sin aumentar excesivamente el orden del modelo.

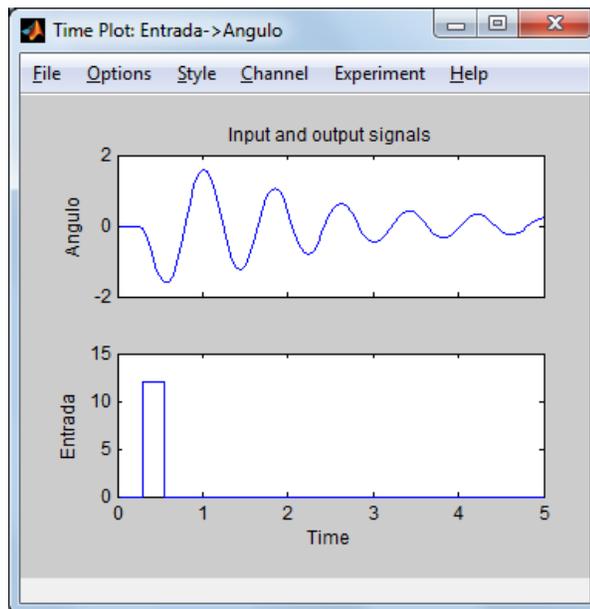


Figura 6: Visualización de los datos experimentales importados en *ident*.

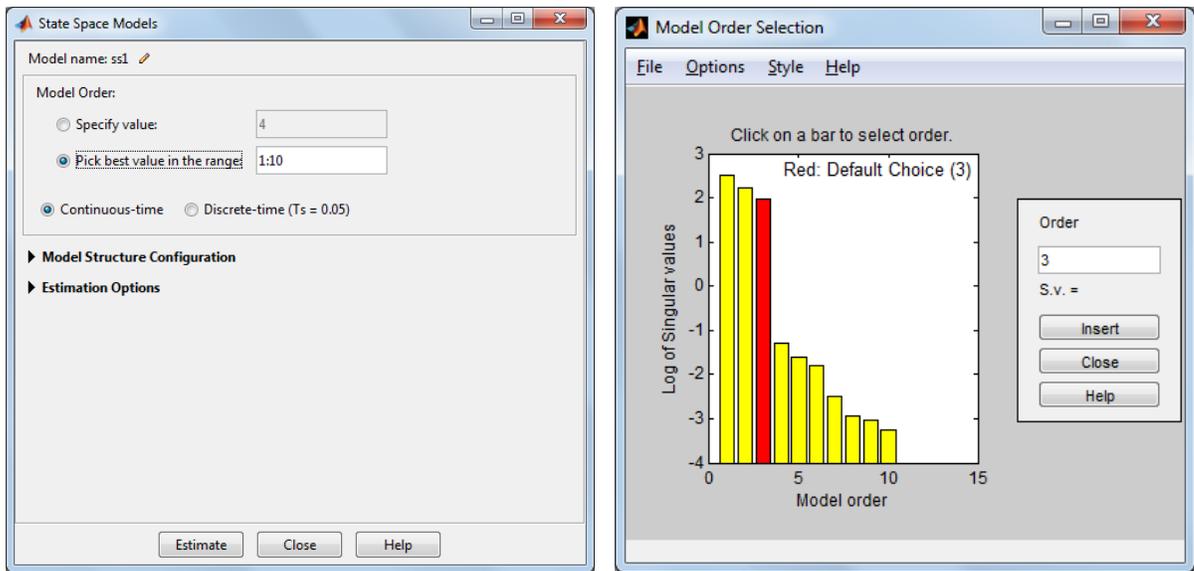
Para validar nuestros modelos obtenidos, podemos comparar su salida ante el estímulo original del experimento seleccionando la opción *Model output* que nos presenta el porcentaje de aproximación entre nuestro modelo y la respuesta original como se muestra en la figura 9.

Paso 6) Como puede observarse en la figura 9, para la salida **Angulo**, el modelo *Tercero* posee la mejor aproximación a los datos experimentales (74.47); pero, realmente los dos primeros modelos son casi iguales por lo que lo seleccionamos a **Primero** como nuestro modelo.

**NOTA 6:** En su **experimento**, para seleccionar su modelo final de entre los diferentes modelos obtenidos, debe utilizar criterios de aproximación y de complejidad, (la mejor aproximación en simulación a los resultados experimentales; con la menor complejidad del modelo y que este llene las expectativas de la dinámica del modelo teórico, si este existe).

Paso 7) Para exportar el modelo lo arrastramos hasta la posición *To workspace* en la ventana del *ident*. Luego en el *workspace* de Matlab, podemos simular para efectos de verificación o extraer las funciones de transferencia de entrada-salida. Para extraer la función de transferencia usamos la sentencia `modelo = zpk(Tercero)`; donde "modelo" es el nombre que le asignamos al resultado.

**NOTA 7:** En otros casos, eventualmente, si el orden estimado para la planta es muy grande, será necesario reducirlo [4], cambiando polos y ceros no dominantes por sus ganancias estáticas u otra forma adecuada; por ejemplo, con las funciones *balreal* y *modred* de Matlab.



a)

b)

Figura 7: Ventanas del proceso de obtención de modelos en variables de estado del *ident* con: a) la selección de parámetros y b) los modelos obtenidos para el ejemplo.

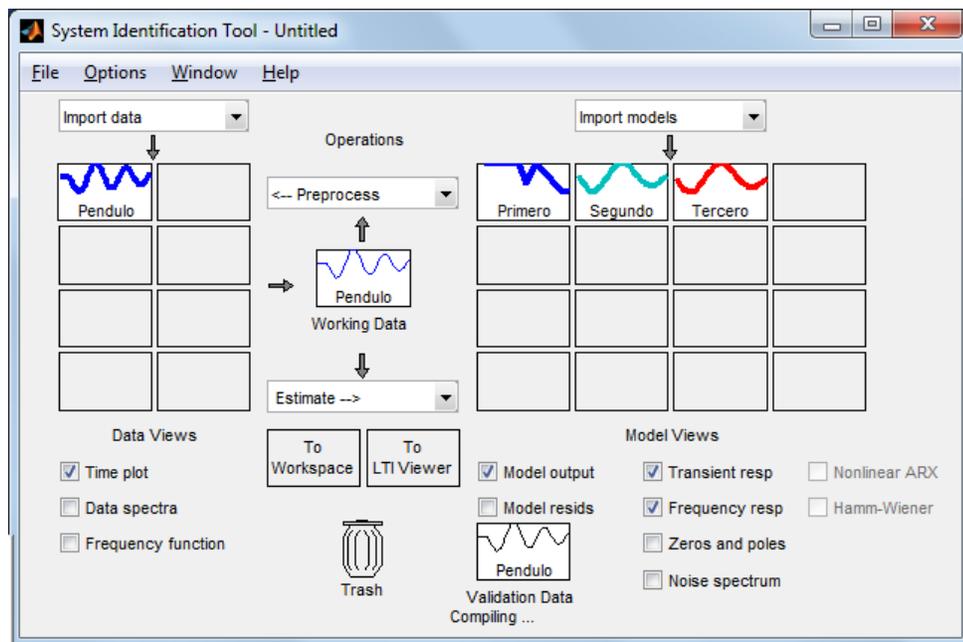


Figura 8: Ventana del *ident* con el objeto de experimento y los modelos obtenidos para el ejemplo. **Sus resultados pueden ser diferentes! Los nombres de los modelos obtenidos no tienen que ver con el orden del modelo; sino, con la secuencia en la que fueron obtenidos.**

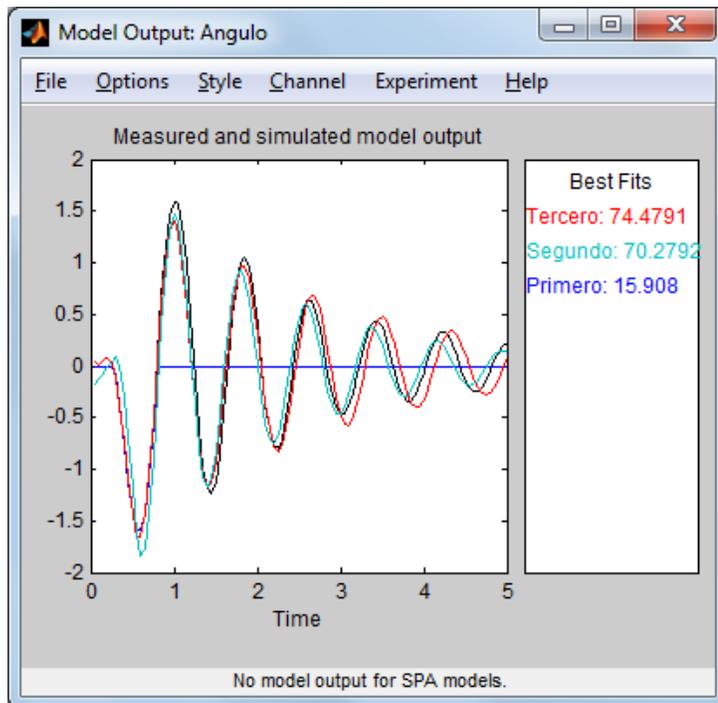


Figura 9: Comparación de modelos con *Model output* para el ejemplo.  
**Sus resultados serán diferentes!**

Una verificación del modelo obtenido se debe realizar también graficando, con la función *plot* de Matlab, los resultados experimentales junto a resultados de simulación producidos con la función *lsim*. La figura 10 muestra esta comparación realizada con las sentencias mostradas en el recuadro a continuación, que además sirven para documentar la gráfica:

```
M = ss(Tercero);
cero = zeros(length(Entrada),1);
X0 = [0; 0]; % vector de estado inicial en cero (su longitud depende de la
cantidad de variables de estado del modelo seleccionado)
[Y,T,X] = lsim(M, [Entrada], Tiempo, X0);
Entrada2 = Entrada / 6; % ajusta el valor, solo para efectos visuales
plot(T, Entrada2, T, Angulo, T, Y)
title('Verificación de la respuesta del modelo')
xlabel('Tiempo [s]')
ylabel('Amplitud')
legend('Entrada', 'Angulo', 'Angulo modelo')
```

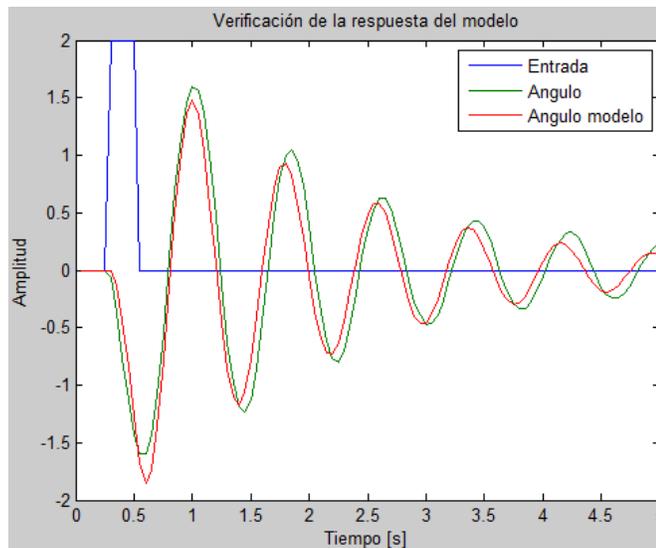


Figura 10: Verificación del modelo del péndulo utilizado en este ejemplo.

Los resultados deben mostrarse al profesor durante la clase y la documentación debe enviarse a **más tardar** a la medianoche del día en que se cumple una semana después de realizado el experimento, el **18 o el 21 de agosto 2016**, al correo electrónico [einteriano@tec.ac.cr](mailto:einteriano@tec.ac.cr), con todos los documentos y archivos utilizados para resolver el problema en un ZIP con el nombre (letra) del equipo.

## Referencias

- [1] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.Models.pdf>
- [2] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.6ModeladoEstocastico.pdf>
- [3] <http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/3.7Identificacion%20de%20sistemas.PDF>
- [4] <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.6.1EjemploModeladoEstocastico.pdf>
- [5] [http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1\\_Grua\\_II\\_2015.pdf](http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1_Grua_II_2015.pdf)

## Aclaración:

El presente instructivo es una adaptación autorizada de la versión original del Proyecto Corto 1 elaborada por el Msc. Eduardo Interiano Salguero, el cual ha puesto en práctica con sus alumnos en varias oportunidades. [11]

Víctor Ramírez Rojas

EIS/VRR/vrr  
2016

## 6.5. Proyecto Corto 2 SiDiCo (Control Furuta PID)



INSTITUTO TECNOLÓGICO DE COSTA RICA

**II SEMESTRE 2016**

ESCUELA DE INGENIERIA EN ELECTRÓNICA

CURSO: EL-5409 LABORATORIO DE CONTROL AUTOMÁTICO

MEDIO: Proyecto corto 2

FECHA: 18 de Julio de 2016

PROF: Ing. Eduardo Interiano

# PROYECTO CORTO 2

**Tema: Control de posición angular del Péndulo de Furuta (SiDiCo).**

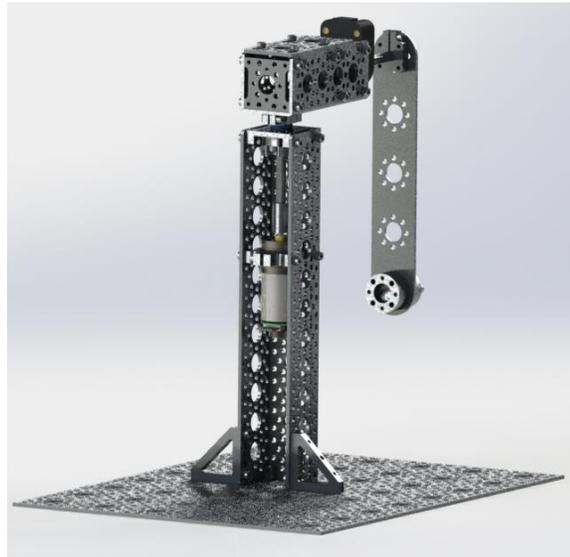


Figura 1: Imagen del Péndulo Invertido de Furuta SiDiCo

**Elementos necesarios:** Sistema Didáctico de Control “SiDiCo” con el software V1.0 y planta Péndulo invertido de Furuta V1.0 (con motor CD de imán permanente de 12V con caja reductora 46.85:1; sensor de posición angular E5 de 4000cpr en el péndulo y un sensor de posición angular integrado en el giro del motor de 2248.86cpr). Computador con Matlab R2013b o superior e *ident* instalado.

## Descripción de la planta

El péndulo de Furuta, es como normalmente se le llama al péndulo invertido rotacional. Llamado así por su inventor Katsuhisa Furuta y sus colegas en 1992 en el Instituto de Tecnología de Tokio. Es un ejemplo de un oscilador no lineal, de ahí su amplio interés de estudio en la teoría de sistemas de control. El péndulo es sub-actuado y extremadamente no lineal debido a las fuerzas gravitacionales y el acoplamiento que surge de las fuerzas Coriolis y centrípetas.

El propósito del péndulo de Furuta consiste en tratar de mantener el péndulo en su posición inestable, o sea es que el actuador impulse con pequeños movimientos el brazo, y este a su vez el péndulo, con el objetivo que este se mantenga arriba. Nótese, que para el control de dicha planta su punto de referencia estará ubicado en dirección contraria a la gravedad, no obstante, los errores en la posición que pueda tener el péndulo se dan en radianes, y podrán ser tanto positivos como negativos.

La planta cuenta con dos sensores digitales de ángulo, uno de 4000 cpr instalado en el péndulo y el segundo instalado en la base del motor, de 2248 CPR para poder saber la posición del brazo. Este motor DC cuenta con una caja reductora de 47:1, su velocidad máxima nominal es de 160 RPM y puede desarrollar una fuerza máxima de 85 oz-in a su tensión máxima de 12Voltios.

Por tratarse de un Sistema Didáctico, cuenta con un microcontrolador PSoC 5LP integrado, que se encarga de recopilar la información necesaria cada periodo y calcular la respuesta que estará ejecutando en el motor. Tanto la lógica de control como los parámetros y ajustes necesarios son definidos por el usuario desde la interfaz gráfica SiDiCo V1.0, más específicamente en el módulo “Tipos de Control”

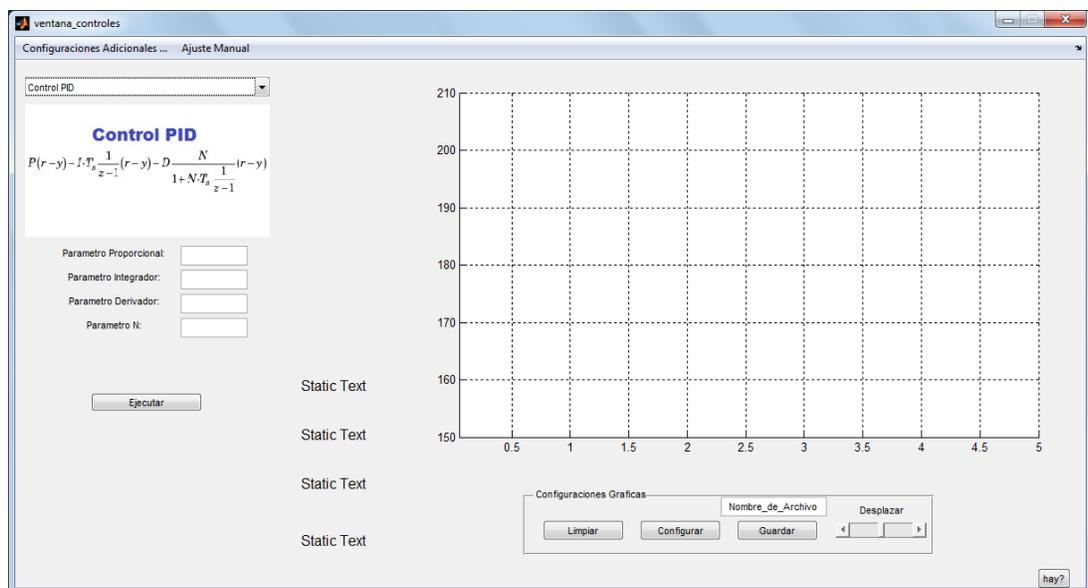


Figura 2: Modulo de control del programa SiDiCo

La **¡Error! No se encuentra el origen de la referencia.** muestra la pantalla principal del módulo de control, donde el usuario puede seleccionar lógicas de control como PID. Si seleccionara un PID está optando por la configuración en paralelo que se muestra a continuación:

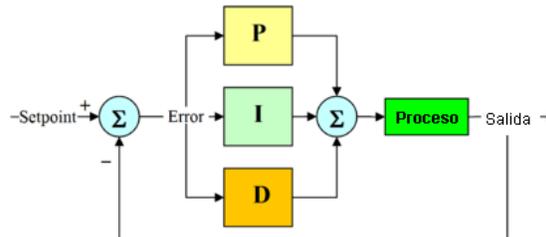


Figura 3: Diagrama de bloques del control PID implementado

### Descripción del trabajo:

El problema de diseño consiste en escoger adecuadamente los polos de lazo cerrado del sistema para cumplir con las especificaciones dadas para el comportamiento del sistema; para luego realizar el cálculo de las constantes P, I y D o los valores de la matriz K adecuadas para la implementación del control correspondiente.

Para este proyecto 2, sus tareas consisten en:

a) Utilizar la herramienta *sisotool* de Matlab para seleccionar la ubicación de los polos de lazo cerrado adecuados para lograr que el Péndulo de Furuta cumpla los requisitos dados en la siguiente tabla y tenga capacidad para eliminar las perturbaciones de entrada o salida a la planta (no forzar el eje).

Parámetro	Sobrepaso	Tiempo de estabilización	Error de estado estacionario
<b>Valor requerido</b>	< 1 %	menor o igual 0.2s	0%

b) Realizar el cálculo de las constantes del PID utilizando el método de auto Tuning de simulink.

c) Verificar el diseño usando *SiDiCo* ante perturbaciones.

d) Probar el control de posición angular y posición del motor, además verificar que el comportamiento deseado se cumple.

## Parte I: Descripción del entorno del regulador PID discreto

### Objetivo: Conocer el entorno del regulador PID discreto

$$P(b \cdot r - y) + I \cdot T_s \frac{1}{z-1} (r - y) + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}} (c \cdot r - y)$$

Ecuación 1: Estructura del regulador PID paralelo en tiempo discreto

La ecuación 1 que representa la estructura de un PID paralelo Euler Forward – Euler Forward. La configuración tradicional de PID no es adecuada cuando la entrada y consecuentemente la señal de error puede tener cambios grandes en forma de escalón ya que los términos proporcional y derivativo podrían llegar a proporcionar una salida demasiado grande, la cual podría saturar la salida del regulador.

En esta ecuación 1, los valores para las constantes P, I, D son los mismos en s (tiempo continuo) que en z (para tiempo discreto), la transformación se realiza con la constante  $T_s$ , el periodo de muestreo, que escala los valores adecuadamente entre dominios. La constante N es parte del filtro derivativo, que realiza la función de volver propio al término derivativo.

En la **¡Error! No se encuentra el origen de la referencia.** se muestra el diagrama de bloques del control PID. Esta variante tiene tres términos, los cuales se encuentran en el camino directo y en forma paralela, actuando sobre la señal de error.

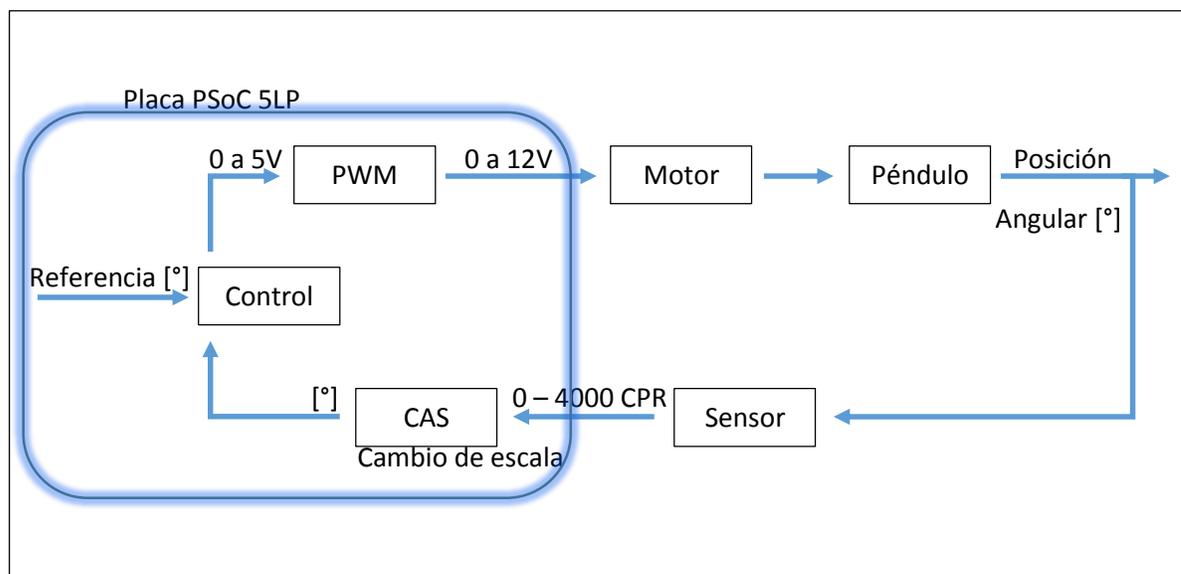


Figura 2: Diagrama del control para la planta Furuta (SiDiCo)

En la Figura 2 se muestra de forma simplificada el sistema de control realizado alrededor de un PSoC de la familia 5LP de la compañía Cypress. Una de las características importantes de esos dispositivos PSoC es que son sistemas híbridos y poseen tanto electrónica digital fija y programable como electrónica analógica, lo que los hace extremadamente adecuados para labores de control automático ya que requieren de pocos elementos analógicos externos.

### Implementación del PID Forward Euler – Forward Euler

Es importante para comprender la operación del sistema mostrar la forma como la ecuación 1, correspondiente al control implementado, es implementada mediante lógica discreta, por tratarse de un proceso un poquito extenso se comparte por partes. El módulo derivativo se demuestra así:

$$\frac{D(z)}{E(z)} = \frac{K_d N}{1 + \frac{N T_s}{(z-1)}} = \frac{(z-1)K_d N}{(z-1) + N T_s} = \frac{(1-z^{-1})K_d N}{1 + z^{-1}(T_s N - 1)}$$

$$D(z) = \frac{(1-z^{-1})K_d N}{1 + z^{-1}(T_s N - 1)} E(z) = E(z)(1-z^{-1})K_d N - D(z)z^{-1}(T_s N - 1)$$

$$d(k) = [e(k) - e(k-1)]K_d N - d(k-1)(T_s N - 1)$$

En cuanto a la variable “N” cumple con el propósito de mitigar los cambios bruscos que intente tener la respuesta derivativa, actuando como un filtro al comparar cada cálculo contra el cálculo derivativo anterior. Además, funciona como amplificador del cambio percibido de la señal de error lo que ayuda que los cambios de movimiento del péndulo no sean bruscos.

En el caso del control integral su composición es más común, la única diferencia al control integral tradicional es que por haber implementado la versión Forward Euler los cálculos se hacen basándose en el error del periodo anterior. En la siguiente imagen se muestran los procedimientos necesarios para poder implementar la fórmula del control integral en el dispositivo discreto.

$$\frac{I(z)}{E(z)} = \frac{K_i T_s}{(z-1)}$$

$$I(z)(1-z^{-1}) = E(z)z^{-1}K_i T_s$$

$$I(z) = E(z)z^{-1}K_i T_s + I(z)z^{-1}$$

$$I(k) = e(k-1)K_i T_s + I(k-1)$$

## Parte II: Diseño y simulación de un control PID para el Péndulo de Furuta

**Objetivo:** Calcular un regulador PID para el servomotor según las especificaciones

Este diseño puede ser realizado tanto en tiempo continuo como en tiempo discreto. En este caso se muestran los pasos para el diseño en tiempo continuo por el método llamado emulación analógica ya que la implementación será digital. Se sugiere al lector realizar por su cuenta el diseño en tiempo discreto y verificar que se obtienen resultados similares. Para el diseño directo en tiempo continuo se debe primero discretizar la planta.

### Procedimiento:

Paso 1) Digite las variables de la planta, para luego estén accesibles de una manera más fácil.

```
M = 0.032323635940169;  
m = 0.08217;  
b = 0.0012119478302;  
I = 0.0013149048825;  
l = 0.1265;
```

Paso 2) Luego escriba el modelo para la planta (como se muestra en el siguiente cuadro), simule el modelo y verifique que posee un comportamiento inestable con *step(sys\_ss)* y *zpk(sys\_ss)*.

```
p = I*(M+m)+M*m*l^2;  
A = [0      1      0      0;  
      0 -(I+m*l^2)*b/p (m^2*g*l^2)/p 0;  
      0      0      0      1;  
      0 -(m*l*b)/p    m*g*l*(M+m)/p 0];  
B = [ 0;  
      (I+m*l^2)/p;  
      0;  
      m*l/p];  
C = [0 0 1 0];  
D = [0];  
  
states = {'brazo' 'veloc_brazo' 'ang' 'veloc_ang'};  
inputs = {'u'};  
outputs = {'ang'};  
  
sys_ss =  
ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs)
```

Paso 2) Encuentre y anote los valores propios del modelo con la orden *eig(sys\_ss)* y utilizando la herramienta *sisotool(sys\_ss)* grafique el lugar de las raíces para el sistema realimentado unitariamente.

Paso 3) Exprese la función de transferencia del péndulo en forma canónica controlable, FCC, con la tercera variable de estado como salida y verifique que el sistema es controlable. (para que sea controlable, el rango de la matriz de controlabilidad debe ser menor o igual a la cantidad de variables de estado del sistema)

```
co = ctrb(sys_ss);
controlabilidad = rank(co)
```

Paso 4) en simulink diseñe un diagrama de bloques tal y como se muestra en la figura 8, procure dentro del módulo *LTI System* defina la variable *sys\_ss*, e incluso establezca las variables iniciales como  $[0 \ 0 \ 0.05 \ 0]$  de manera que le establece una pequeña perturbación inicial a la posición angular del péndulo. Este dato está dado en radianes.

Paso 5) Verifique que el periodo de muestreo del retenedor de orden zero y el controlador discreto PID de 2 grados de libertad es el mismo (se recomienda 10ms para el primer experimento).

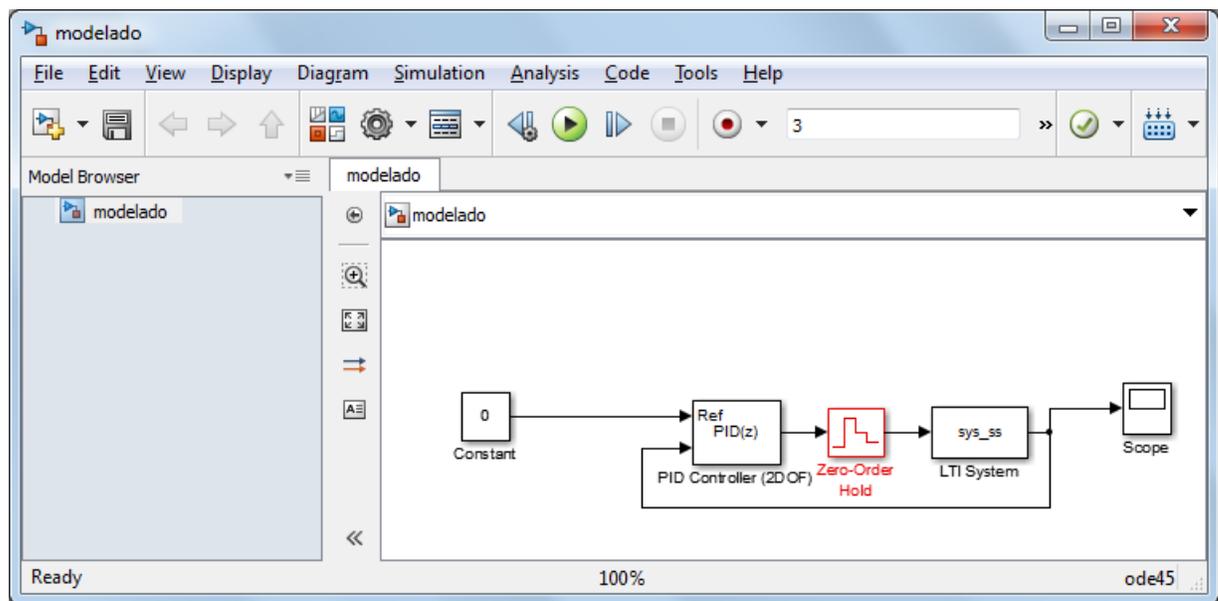


Figura 3: Diagrama para simular la planta en simulink

Paso 6) En el bloque *PID controller* asegúrese haber seleccionado los métodos Forward Euler tanto para el integrador como para el filtro y haber seleccionado un control discreto. Compare las configuraciones del control con los de la Figura 8.

Paso 7) Utilice la herramienta de *PID Tuner* para autoajustar los parámetros de control. Además de la herramienta puede que sea necesario ajustar de manera manual el tiempo de respuesta y agresividad del control. Para esta etapa se le sugiere al estudiante estar observando tras cada cambio las escalas de tiempo y amplitud, puesto que no es correcto diseñar un control que demore una exageración o bien necesite aplicar picos imposibles para un motor de esta categoría.

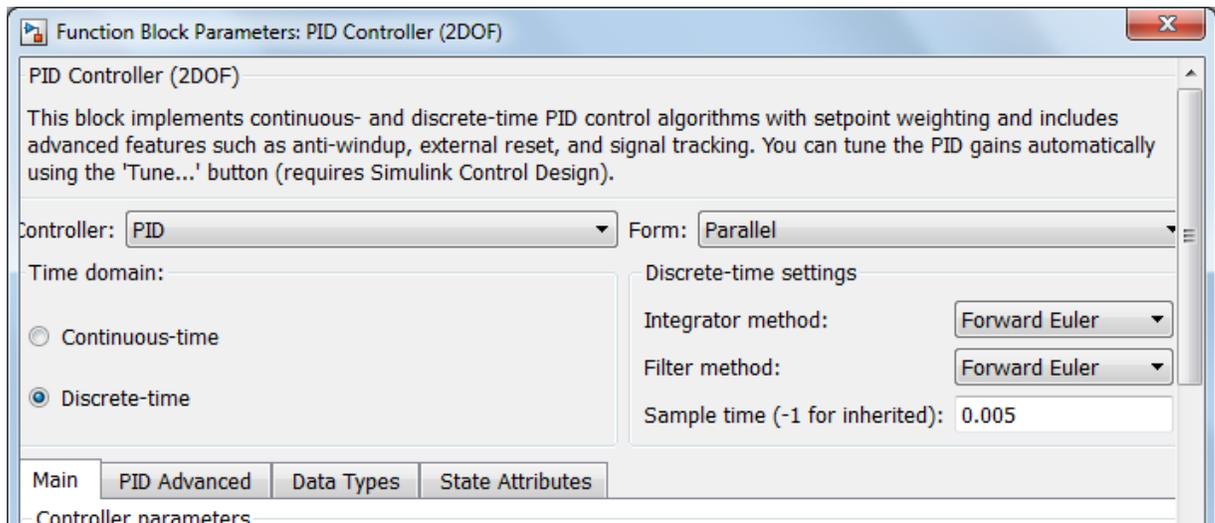


Figura 4: Sistema de control en simulink con PID de dos grados de libertad

Paso 8) No siempre los modelos de una planta se logran modelar en forma de variables de estados, en ocasiones la forma de función de transferencia es más conveniente, por lo que se le sugiere al estudiante repetir los pasos del 2 al 7 con la función de transferencia mostrada a continuación y observe las diferencias y similitudes del proceso

$$q = (M+m) * (I+m*l^2) - (m*l)^2;$$

$$P_{pend} = (m*l*s/q) / (s^3 + (b*(I + m*l^2)) * s^2/q - ((M + m) * m*g*l) * s/q - b*m*g*l/q)$$

### Parte III: Verificación del diseño en el sistema de desarrollo SiDiCo

Paso 1) Introduzca las constantes de su diseño utilizando los mecanismos dispuestos para ello en el sistema de desarrollo, el asistente le dará las indicaciones necesarias.

Paso 2) Oprima el botón de **Ejecutar** en el sistema de desarrollo para iniciar la ejecución del ejercicio. Analice su comportamiento en comparación con la simulación. Puede que sea necesario que tenga que hacer un pequeño ajuste con criterio en los parámetros, lo anterior se debe a que el modelo ha sido linealizado o aproximado para obtener un modelo reducido, sin embargo, en este proceso se tiende a perder pequeños detalles que con un pequeño ajuste se soluciona.

Paso 3) Guarde las gráficas obtenidas con los valores de error, posición y señal de control.

Los resultados deben enviarse a más tardar **ocho días después de haber realizado el experimento** al correo electrónico [einteriano@tec.ac.cr](mailto:einteriano@tec.ac.cr), con todos los documentos y archivos creados para resolver el problema.

## Referencias

[ 1 ]

[http://www.ie.itcr.ac.cr/einteriano/control/clase/Zomorodi\\_Shahrokhi\\_PID\\_Tunning\\_Comparison.pdf](http://www.ie.itcr.ac.cr/einteriano/control/clase/Zomorodi_Shahrokhi_PID_Tunning_Comparison.pdf)

[ 2 ] <http://www.ie.itcr.ac.cr/einteriano/control/clase/Clase17ElReguladorPID.pdf>

### **Aclaración:**

El presente instructivo es una adaptación autorizada de la versión original del Proyecto Corto 2 elaborada por el Msc. Eduardo Interiano Salguero, el cual ha puesto en práctica con sus alumnos en varias oportunidades. [11]

Víctor Ramírez Rojas

EIS/VRR/vrr  
2016

## 6.6. Diagrama de Conexión Esquemático de la Planta

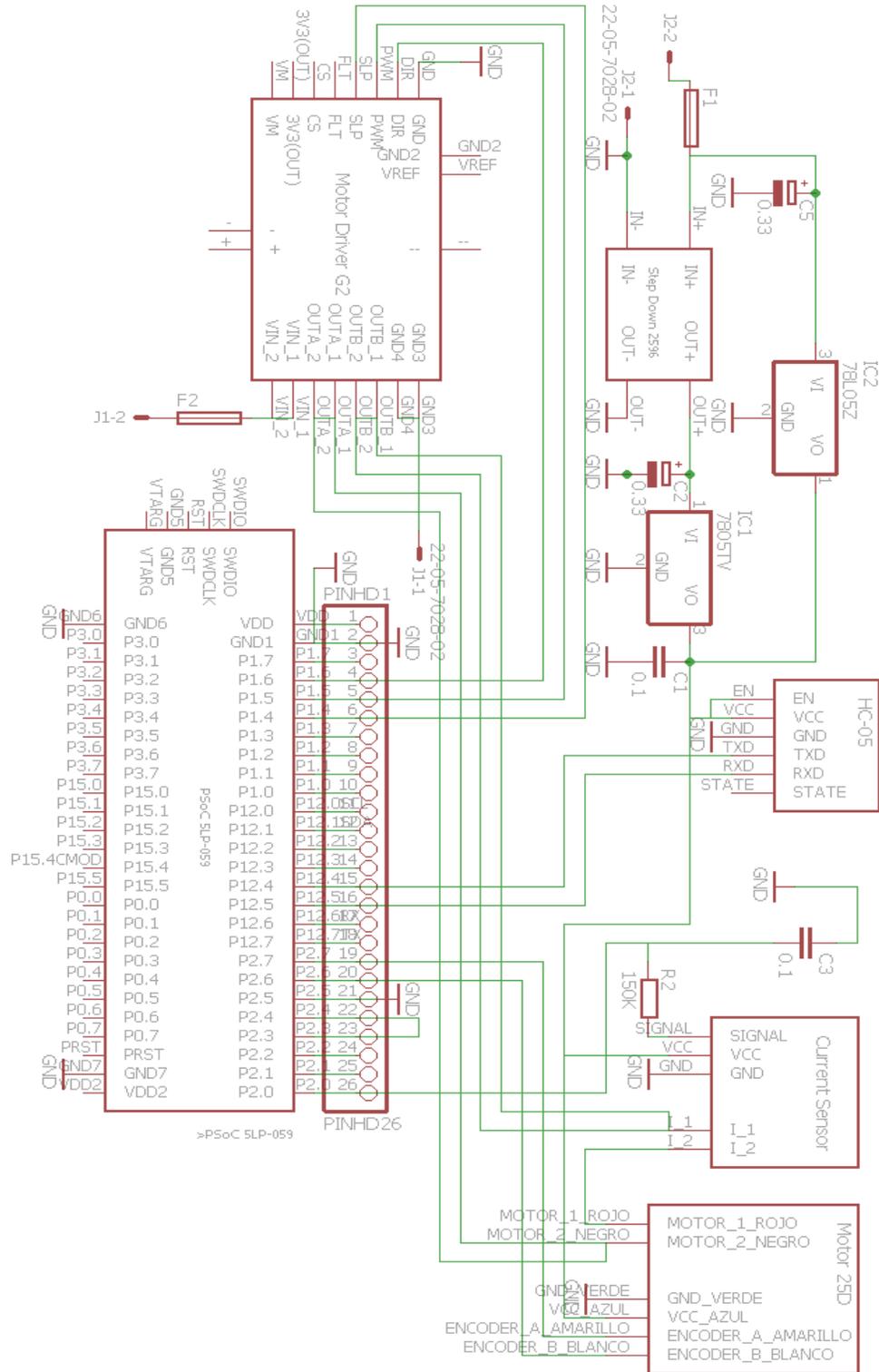


Imagen 6.1: Diagrama de Conexión Esquemático de la Planta

## 6.7. Diseño del PCB de la Planta

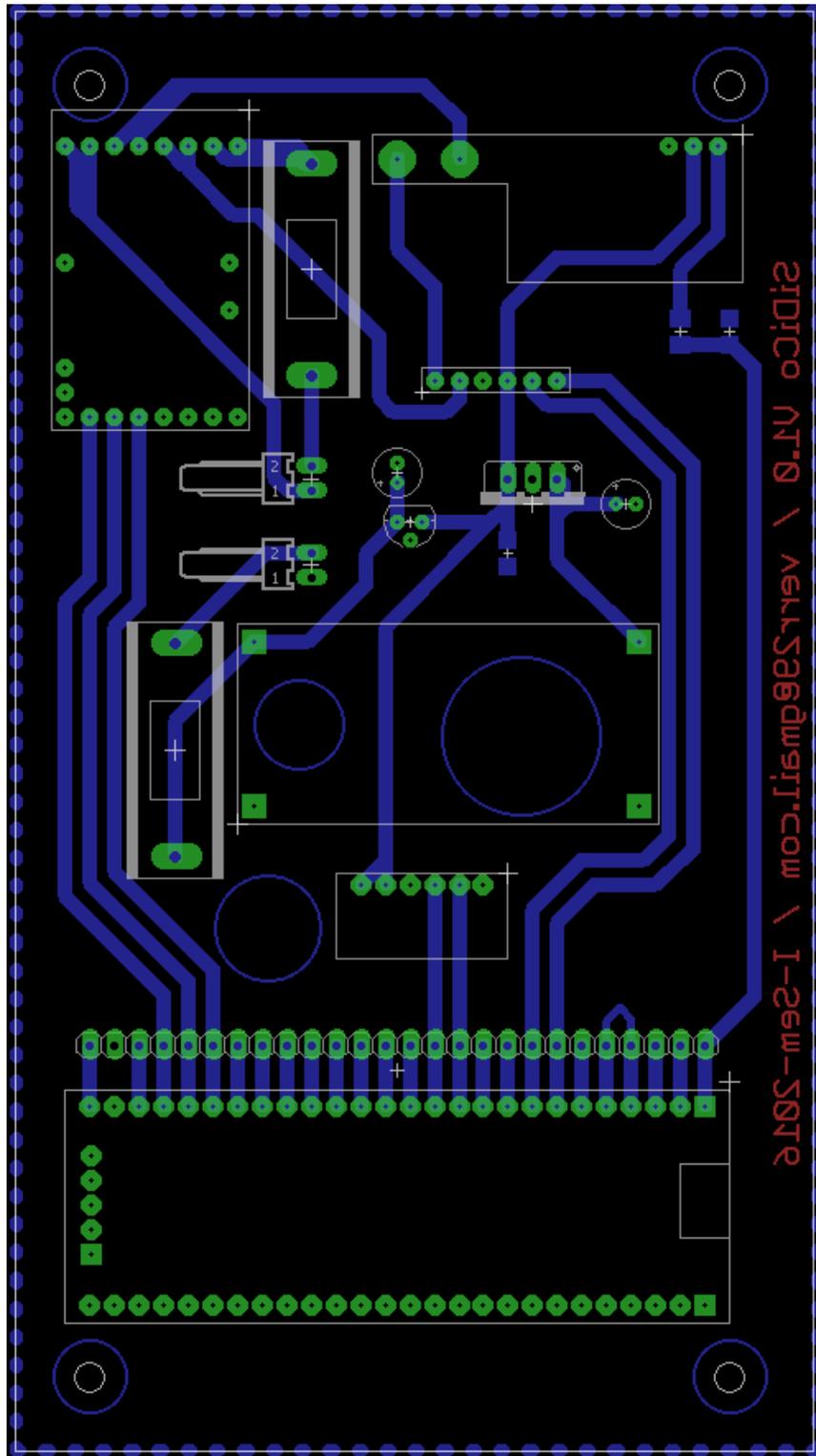


Imagen 6.2: Diseño del PCB de la Planta (tamaño a escala)

## 6.8. Diagrama de Conexión Esquemático y Diseño del PCB del brazo

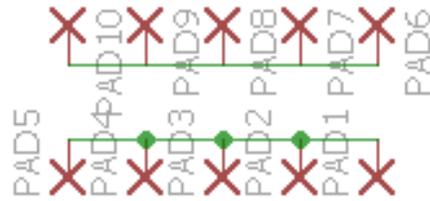


Imagen 6.3: Diagrama de Conexión Esquemático del distribuidor de energía del brazo

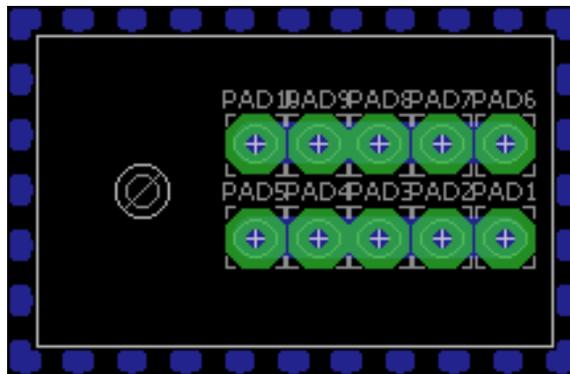


Imagen 6.4: Diseño del PCB distribuidor de energía del brazo (tamaño a escala)

## 6.9. Estructura Protectora de Acrílico

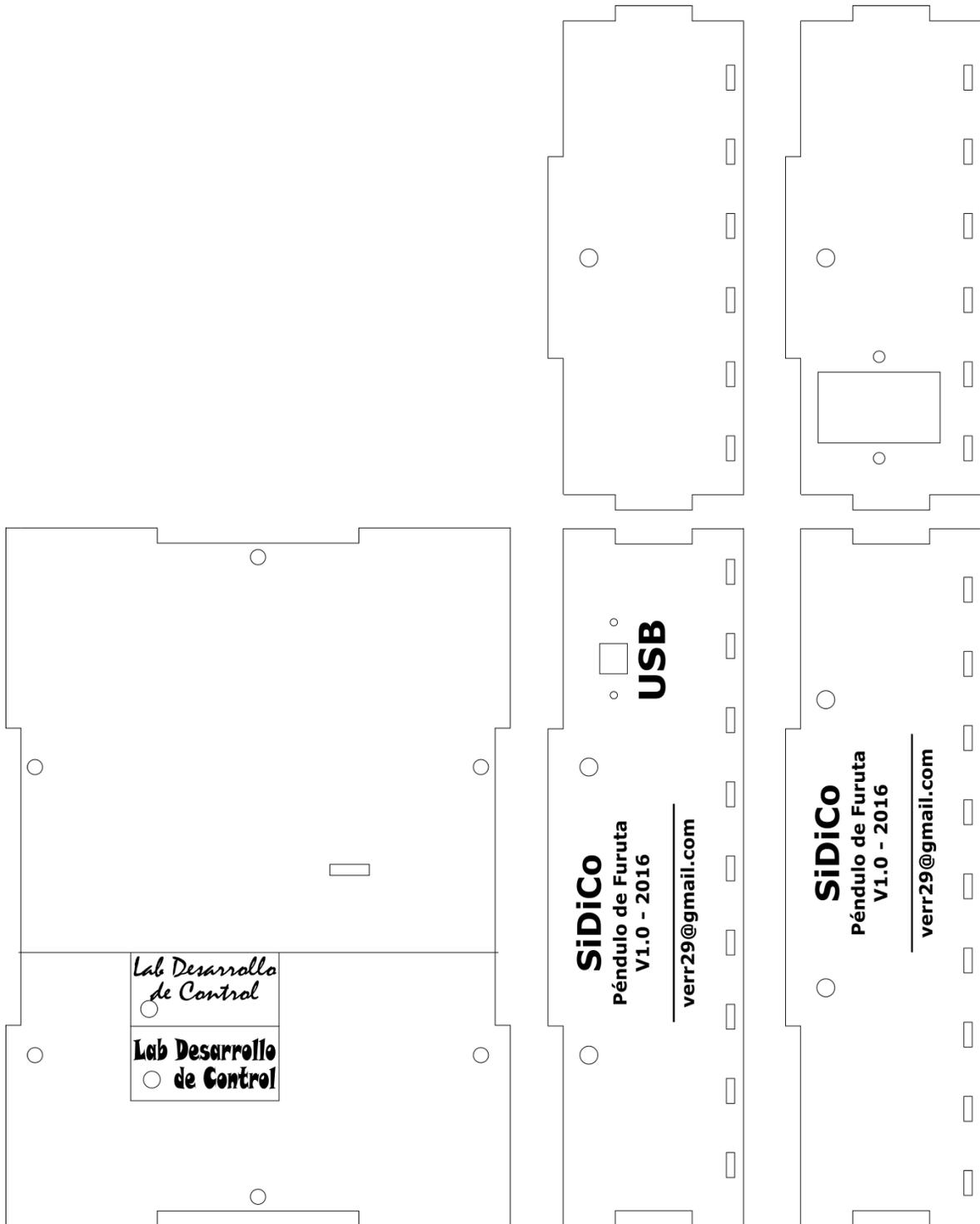


Imagen 6.5: Estructura Protectora de Acrílico (tamaño a escala)

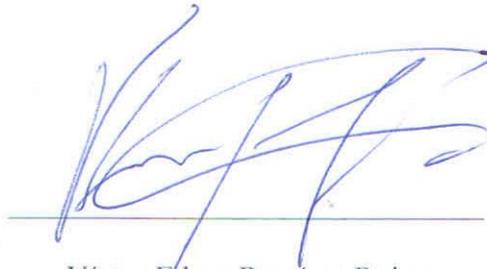
## 6.10. Carta de Entendimiento

Señores

Instituto Tecnológico de Costa Rica

Biblioteca José Figueres Ferrer

Yo Víctor Edgar Ramírez Rojas carné 9927178, autorizo a la Biblioteca José Figueres del Instituto Tecnológico de Costa Rica disponer del Trabajo Final realizado por mi persona, con el título “Creación de un sistema didáctico e interactivo de control automático basado en el péndulo de Furuta” para ser ubicado en la Biblioteca Digital y ser accesado a través de la red Internet.



Víctor Edgar Ramírez Rojas

Cédula: 2-0565-0707

## 7. Bibliografía

- [1] Instituto Tecnológico de Costa Rica, «Acerca del Tec,» 2015. [En línea]. Disponible en: <http://www.tec.ac.cr/eltec/Paginas/acercaDelTec.aspx>. [Último acceso: 15 01 2016].
- [2] WikipediA, «Furuta Pendulum,» 04 08 2017. [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Furuta\\_pendulum](https://en.wikipedia.org/wiki/Furuta_pendulum). [Último acceso: 25 01 2016].
- [3] Quanser Inc, «Rotary Motion Control Systems,» 2016. [En línea]. Disponible en: [http://www.quanser.com/Products/rotary\\_motion](http://www.quanser.com/Products/rotary_motion). [Último acceso: 25 01 2016].
- [4] Universidad de Sevilla, «Motor CC,» [En línea]. Disponible en: <http://es.slideshare.net/edzam/motor-cc-17217844>. [Último acceso: 26 08 2016].
- [5] J. Barragán Piña, «Función de transferencia de la ecuación de estado,» 30 05 2011. [En línea]. Disponible en: <http://uhu.es/antonio.barragan/content/23-funcion-transferencia-ecuacion-estado>. [Último acceso: 26 01 2016].
- [6] MathWorks Inc, «c2d,» [En línea]. Disponible en: <http://www.mathworks.com/help/control/ref/c2d.html>. [Último acceso: 27 01 2016].
- [7] CRCibernética, «Wireless Charging Module - 5V OUTPUT,» [En línea]. Disponible en: <http://www.crcibernetica.com/wireless-charging-module-5v-output/>. [Último acceso: 28 01 2016].
- [8] Cypress, «CY8CKIT-059 PSoC® 5LP Prototyping Kit With Onboard Programmer and Debugger,» 10 02 2016. [En línea]. Disponible en: <http://www.cypress.com/documentation/development-kitsboards/cy8ckit-059-psoc-5lp-prototyping-kit-onboard-programmer-and>. [Último acceso: 18 02 2016].
- [9] US Digital, «E5 Optical Kit Encoder,» [En línea]. Disponible en: <http://www.usdigital.com/products/e5>. [Último acceso: 12 02 2016].
- [10] D. Smooth, «Consultorio de Física,» 20 03 2014. [En línea]. Disponible en: <http://zonaforo.meristation.com/topic/2199504/>. [Último acceso: 01 08 2016].
- [11] R. M. Murray, «LQR Control,» California, 2016, p. 2 y 3.  
]

[12 University of Michigan, «Inverted Pendulum,» [En línea]. Disponible en:

] <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling>. [Último acceso: 21 05 2016].

[13 E. Interiano Salguero, «Proyecto Corto 1,» 2015. [En línea]. Disponible en:

] [http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1\\_Grua\\_II\\_2015.pdf](http://www.ie.itcr.ac.cr/einteriano/control/Laboratorio/ProyectosCortos/ProyectoCorto1_Grua_II_2015.pdf). [Último acceso: 12 05 2016].