

Control del Péndulo Invertido Simple - UV en Régimen de Pequeña Señal

Alejandro Astudillo Vigoya

alejandro.astudillo@correounivalle.edu.co

Trabajo de grado dirigido a la
Universidad del Valle

como requisito parcial para obtener el título de

INGENIERO ELECTRÓNICO



Escuela de Ingeniería Eléctrica y Electrónica

Santiago de Cali

Valle del Cauca, COLOMBIA

27 de agosto de 2015

Control del Péndulo Invertido Simple - UV en Régimen de Pequeña Señal

Alejandro Astudillo Vigoya

alejandro.astudillo@correounivalle.edu.co

Trabajo de grado dirigido a la
Universidad del Valle

Director:

Edinson Franco Mejía, Ph.D.

Co-Director:

José Tomás Buitrago Molina, M.Sc.

Escuela de Ingeniería Eléctrica y Electrónica

Santiago de Cali

Valle del Cauca, COLOMBIA

27 de agosto de 2015

Reconocimientos

Quiero agradecer a mis padres por haberme brindado tan excelente formación, educándome día a día para poder cumplir con este logro, apoyándome para continuar con mis sueños e inculcándome la necesidad de retribuir a la sociedad los conocimientos adquiridos durante mi aprendizaje diario. Gracias por brindarme la posibilidad de crecer cada día a través de sus actos, cariño y amor.

A mi abuela por cuidarme, apoyarme, brindarme todo su cariño y querer siempre mi progreso desde el día en que nací.

A mi novia por brindarme ese apoyo incondicional, escucharme, soportarme, entenderme y por soñar conmigo con tantas cosas de las cuales este trabajo es un paso más. Gracias por escuchar mis problemas, por permitirme desahogarme cuando mi mente no podía dejar de pensar en los problemas a solucionar en este proyecto y por brindarme tu ayuda y tu amor cuando más lo necesité.

Quiero también agradecer al Profesor Edinson Franco, quien ha sido mi mentor dentro del Grupo de Investigación, por su guía y apoyo en este proyecto, y por brindarme consejos para mi futuro profesional y para el desarrollo de este proyecto.

Al profesor José Tomás Buitrago y a Eisenhower Cabal por los consejos y la ayuda que me brindaron durante el desarrollo de este proyecto que me fueron muy útiles para llegar al buen término del mismo.

Por último quiero agradecerle a mis amigos, quienes han estado conmigo desde que inició este enorme reto de estudiar Ingeniería Electrónica en la Universidad del Valle, porque sin ustedes no estaría aquí. Tantos años compartidos y tanto apoyo brindado hacen que este y todos los trabajos de grado nuestros sean logros del grupo.

Índice general

Resumen	I
Abstract	II
1. Introducción	1
1.1. Introducción	1
2. Marco Teórico	3
2.1. Descripción del Péndulo Invertido	3
2.2. Modelado Matemático del Péndulo Invertido Simple Lineal	6
2.2.1. Ecuaciones Dinámicas del Sistema	6
2.2.2. Funciones de Transferencia del Sistema	9
2.2.3. Representación en Espacio de Estados	10
3. Caracterización Eléctrica y Mecánica de la Planta del Péndulo Invertido	13
3.1. Modelado del Motor Brushless de DC	14
3.1.1. Modelado Matemático del Motor	14
3.1.2. Función de Transferencia del Motor	15
3.2. Parametrización del Motor de DC	16
3.2.1. Determinación de la Zona Muerta y la Zona Lineal del Motor	16
3.2.2. Resistencia de Armadura R_m	18
3.2.3. Constante Eléctrica del Motor K_e	20
3.2.4. Constante de Torque K_m del Motor	22
3.2.5. Coeficiente de Fricción del Motor B_m	23
3.2.6. Inductancia L_m y constante de tiempo eléctrica del motor τ_e .	24
3.2.7. Momento de Inercia del Motor J_m	27
3.3. Modelado Mecánico de la Planta	28
3.4. Validación del Modelo del Motor	32
4. Integración Software-Hardware	35
4.1. Sensórica	36
4.1.1. Sensor de Posición Angular del Péndulo	36
4.1.2. Sensor de Posición Lineal del Carro	37
4.2. Alimentación de los Módulos	38
4.2.1. Alimentación de los Motores	38

4.2.2.	Alimentación de los Circuitos de Sensórica y Auxiliares	39
4.3.	Controladores de los Motores	39
4.4.	Tarjeta de Adquisición de Datos NI DAQ USB-6002	40
4.4.1.	Descripción Tarjeta NI DAQ USB-6002	40
4.4.2.	Software Necesario para Interacción Scilab-DAQ	41
4.4.3.	Problemas encontrados en la interacción Scilab-DAQ	41
4.5.	Arduino Mega 2560 como Tarjeta de Adquisición de Datos	41
4.5.1.	Descripción Arduino Mega 2560	41
4.5.2.	Software Necesario para Interacción Scilab-Arduino/XCOS	42
4.6.	Pruebas de Funcionamiento de Integración Hardware-Software	43
4.6.1.	Pruebas de los Sensores en XCOS	43
4.6.2.	Pruebas de Control del Motor desde XCOS	44
5.	Diseño y Simulación de Estrategia de Control	49
5.1.	Validación del Modelo del PIS-UV	49
5.2.	Control por Realimentación de Estados	53
5.2.1.	Controlabilidad del Sistema	54
5.2.2.	Regulador Lineal Óptimo Cuadrático LQR	55
5.2.3.	Ganancias de Realimentación a partir de la Forma Canónica Controlable	56
5.3.	Estimación de Estados	58
5.3.1.	Observabilidad del Sistema	58
5.3.2.	Estimación Mediante Filtro de Kalman	59
5.3.3.	Estimación Mediante Derivación Discreta	60
5.4.	Simulaciones del Sistema de Control	61
5.4.1.	Simulación del Sistema en Condiciones Ideales	61
5.4.2.	Simulación del Sistema con Ruidos y Error por Cuantización	64
6.	Validación del Sistema de Control en la Planta	68
6.1.	Validación de la ley de control en la planta PIS-UV	68
6.2.	Restricciones en la planta PIS-UV	71
7.	Conclusiones y Trabajo Futuro	76
7.1.	Conclusiones	76
7.2.	Trabajo Futuro	78
	Bibliografía	79
A.	Códigos utilizados en el desarrollo	82
A.1.	Modificaciones al Programa toolbox arduino v3.ino	82
A.2.	Codigo de Implementación de Algoritmo LQR en Scilab	84
A.3.	Codigo Diseño de Ganancias de Realimentación a partir de la Forma Canónica Controlable en Scilab	84
A.4.	Codigo de Diseño de Estimador Kalman en Scilab	86
A.5.	Codigo de Diseño de Filtro Digital en Scilab	86

Índice de cuadros

2.1. Símbolos y parámetros correspondientes en el DCL de la planta . . .	7
3.1. Resistencia de armadura medida utilizando un multímetro	19
3.2. Datos de V_m , I_m y R_m con rotor detenido	20
3.3. Inductancia del motor medida con Inductómetro	25
3.4. Constante de Tiempo medida mediante caracterización dinámica . . .	26
3.5. Datos correspondientes al experimento para encontrar μ_d del carro . .	31
3.6. Parámetros Mecánicos de la Planta	32
3.7. Parámetros del Motor	33

Índice de figuras

2.1. Péndulo Invertido Simple Lineal	4
2.2. Péndulo Invertido Doble sobre Carro	4
2.3. Péndulo de Furuta	5
2.4. Péndulo de Rueda Inercial	5
2.5. Diagrama de Cuerpo Libre Péndulo Invertido Simple	6
3.1. Circuito equivalente de Motor DC	14
3.2. Conexión para determinación de zona muerta y lineal del motor	16
3.3. Fuente Programable BK PRECISION XLN6024	16
3.4. Tacómetro Digital DT-2234C+	17
3.5. Caracterización Estática del Motor en Estudio	17
3.6. Zona Muerta del Motor en Estudio	18
3.7. Constante eléctrica del motor para cada valor de V_m	21
3.8. Conexión Mecánica entre los rotores para medición de K_e	21
3.9. Fuerza contra-electromotriz Vs. Velocidad angular del motor	22
3.10. Coeficiente de Fricción del Motor Vs. Voltaje de Excitación	24
3.11. Constante de Tiempo Mecánica τ_m para cada Voltaje V_m	28
3.12. Plano frontal de la planta del Péndulo Invertido Simple - UV	29
3.13. Plano del Cilindro Móvil que aporta masa al Péndulo	29
3.14. Diagrama de Cuerpo Libre del Carro sobre el Riel	30
3.15. Comparación respuesta dinámica del motor real y modelo	34
4.1. Modulo GY521 basado en MPU6050	36
4.2. Diagrama de Bloques del Procesamiento de la Señal Ángulo	37
4.3. Sensor infrarrojo de distancia SICK DT20-P130B1080	38
4.4. Diagrama de Bloques del Procesamiento de la Señal Posición	38
4.5. Fuente de 24 V / 500 W para alimentación de motores	39
4.6. Modulo controlador de motor BTS7960 43A	40
4.7. Tarjeta de Adquisición de Datos NI DAQ USB-6002	40
4.8. Arduino Mega 2560	42
4.9. Caracterización Bloque AnalogWrite	42
4.10. Subsistema conversor de voltaje deseado en numero de 8 bits	43
4.11. Diagrama para lectura de sensores en XCOS	43
4.12. Bloques de conversión de señales de sensores en datos reales de salidas	44
4.13. Prueba de los elementos de sensado en la Planta PIS-UV	44

4.14. Respuesta del Carro ante entrada Rampa Unitaria Positiva y Negativa en el Motor	45
4.15. Respuesta del Carro ante entrada Rampa Unitaria Positiva y Negativa tras compensación	46
4.16. Subsistema de Compensación de Zona Muerta del Motor	46
4.17. Identificación zona muerta motor con entrada rampa unitaria	47
4.18. Respuesta de la Posición del Carro tras Compensación de Zona Muerta en el Motor	48
5.1. Diagrama de Polos y Ceros de Sistema Continuo PIS-UV	50
5.2. Diagrama de Polos y Ceros de Sistema Continuo PIS-UV	51
5.3. Validación del modelo ante una entrada escalón unitario positivo . . .	52
5.4. Validación del modelo ante una entrada escalón unitario negativo . . .	53
5.5. Diagrama de Bloques de Realimentación de Estados	54
5.6. Implementación de los dos tipos de Estimadores de Estados en XCOS	61
5.7. Diagrama en XCOS para la Simulación del Sistema de Control en Condiciones Ideales	62
5.8. Simulación del Control en Condiciones Ideales utilizando Filtro Kalman como Estimador de Estados	62
5.9. Simulación del Control en Condiciones Ideales utilizando Filtro Kalman como Estimador de Estados	63
5.10. Simulación del Control en Condiciones Ideales utilizando Estimador de Estados basado en Derivación Discreta	63
5.11. Simulación del Control en Condiciones Ideales utilizando Estimador de Estados basado en Derivación Discreta	64
5.12. Diagrama en XCOS para la Simulación del Sistema de Control con Ruido y Cuantización	65
5.13. Simulación del Control en Condiciones Reales utilizando Filtro Kalman como Estimador de Estados	65
5.14. Simulación del Control en Condiciones Reales utilizando Filtro Kalman como Estimador de Estados	66
5.15. Simulación del Control en Condiciones Reales utilizando Estimador de Estados basado en Derivación Discreta	66
5.16. Simulación del Control en Condiciones Reales utilizando Estimador de Estados basado en Derivación Discreta	67
6.1. Diagrama en XCOS para la Validación del Sistema de Control en la Planta Real	69
6.2. Respuesta Real del Sistema de Control utilizando Filtro Kalman como Estimador de Estados	69
6.3. Respuesta Real del Sistema de Control utilizando Filtro Kalman como Estimador de Estados	70
6.4. Respuesta Real del Sistema de Control utilizando Estimador de Estados basado en Derivación Discreta	70
6.5. Respuesta Real del Sistema de Control utilizando Estimador de Estados basado en Derivación Discreta	71

6.6. Presencia de Pulso en la medición de la señal de ángulo utilizando Estimador de Estados basado en Derivación Discreta	72
6.7. Presencia de Pulso en la medición de la señal de ángulo utilizando Estimador de Estados basado en Derivación Discreta	72
6.8. Respuesta del sistema de control con estimador Filtro Kalman ante la presencia de retardos en medición	74
6.9. Respuesta del sistema de control con estimador Filtro Kalman ante la presencia de retardos en medición	74
6.10. Respuesta del sistema de control con estimador basado en derivación discreta ante la presencia de retardos en medición	75
6.11. Respuesta del sistema de control con estimador basado en derivación discreta ante la presencia de retardos en medición	75

Resumen

Este documento describe el desarrollo del proyecto de puesta en funcionamiento y control de la planta del Péndulo Invertido Simple (PIS-UV) perteneciente a la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle.

La planta PIS-UV poseía únicamente su estructura mecánica y un motor como actuador, del cual se desconocía todas sus características. A lo largo de este documento se presenta el desarrollo de la caracterización eléctrica y mecánica de los elementos existentes inicialmente en la planta, el diseño e implementación de los elementos de sensado y actuación, además del modelado de la planta y el diseño de un controlador lineal por realimentación de estados en conjunto con un observador de estados. El sistema de control diseñado deberá mantener el vástago del péndulo en su posición de equilibrio inestable, esto es, sostenido en la zona superior del eje vertical.

En el diseño de la estrategia de control se utiliza el método del Regulador Cuadrático Lineal (LQR) y el diseño a partir de la Forma Canónica Controlable. Por otro lado, para el diseño del observador o estimador de estados se utiliza un Observador basado en Filtro Kalman (LQE) y un estimador basado en Derivación Discreta. En este documento se compara el desempeño de la estrategia de control empleando los dos tipos de estimadores de estados mediante simulaciones y realizando la implementación en la planta real.

Los diseños y simulación de la estrategia de control y de los estimadores de estados se realizaron utilizando el software matemático Scilab. Esta herramienta gratuita permitió realizar además la implementación de este sistema de control en la planta real PIS-UV utilizando como tarjeta de adquisición de datos una tarjeta Arduino Mega 2560.

Se logró poner en funcionamiento la planta del Péndulo Invertido Simple de manera que es posible medir el ángulo del péndulo y la posición del carro y controlar el sentido y velocidad de giro del motor actuador. Por último, se identificaron algunas restricciones que condicionaron el éxito de la estrategia de control implementada sobre la planta real.

Palabras Clave: Péndulo Invertido, Control, Caracterización, Modelado, Realimentación de Estados, Estimador de Estados de Filtro Kalman, Estimador de Estados por Derivación Discreta, Scilab.

Abstract

This document describes the development of the Simple Inverted Pendulum (PIS-UV) commissioning and control. This plant belongs to the Electric and Electronics Engineering School of the Universidad del Valle.

The PIS-UV plant had only its mechanical structure and a DC motor as an actuator with unknown characteristics. Throughout this document it is introduced the development of the electric and mechanic characterization of the elements that at the beginning were in the plant, the design and implementation of the sensors and actuator electronic elements, besides the modelling of the plant and the linear state feedback controller design with a state observer. The designed control system must keep the pendulum stick in its unstable equilibrium position, this is, to keep the angular position of the pendulum in the top of the vertical axis.

In the design of the control strategy it is used the Linear-Quadratic Regulator (LQR) method and the design from the controllable canonical form. On the other hand, for the state observer design it is used a Kalman Filter observer (LQE) and a discrete derivative-based state estimator. In this document it is compared the control strategy performance using the two types of state estimators by performing simulations and implementing the strategy in the real plant.

The control strategy and state estimators designs and simulations were performed using the mathematic software Scilab. This free software tool allowed for the implementation of the control system in the real PIS-UV plant using an Arduino Mega 2560 as a data acquisition card.

It was possible to put the PIS-UV plant into operation so that it is possible to measure the angular position of the pendulum and the linear position of the cart, and control the direction and velocity of spinning of the actuator motor. Finally, some restrictions that limit the success of the control strategy implemented over the real plant were identified.

Keywords: Inverted Pendulum, Control, Characterization, Modelling, State Feedback, Kalman Filter State Estimator, Discrete Derivation State Estimator, Scilab.

Capítulo 1

Introducción

1.1. Introducción

La ingeniería de control busca aumentar la confiabilidad, efectividad, competitividad y calidad de los procesos con aplicación dentro de procesos de manufactura, robótica, transporte, generación de electricidad, salud, entre otros. En su aplicación considera técnicas de medición y regulación con el objetivo de mantener las diversas variables involucradas en un estado deseado, teniendo en cuenta a cada momento la diferencia entre el valor actual de las variables respecto al valor deseado de las mismas, conformando así un sistema de control automático realimentado.

La educación en ingeniería de control tiene gran importancia dentro de las instituciones universitarias de todo nivel debido a la gran variedad de aplicaciones que posee y la importancia que los sistemas de control tienen dentro del mundo actual. Las simulaciones por computador no son suficientes para el adecuado aprendizaje de los sistemas de control y se hace necesario la implementación de dispositivos operativos reales que faciliten el aprendizaje de los objetivos, requerimientos, métodos y teoría de las acciones de control.

Los péndulos invertidos son sistemas ampliamente utilizados para la enseñanza de sistemas de control. Debido a sus características de no-linealidad, inestabilidad y fase no mínima, representan un sistema de gran importancia para el estudio de sistemas de control y pruebas de estrategias de control, por lo que son herramientas esenciales que debe poseer toda institución que dicte cátedra en teoría de control y automatización.

En el Laboratorio de Automática de la Escuela de Ingeniería Eléctrica y Electrónica en la Universidad del Valle, existe una estructura mecánica de un péndulo invertido la cual carece de electrónica y control. La planta de péndulo invertido fue concebida en el marco del proyecto PERI que fue el resultado del proyecto “Plataforma de procesamiento distribuido, con acceso remoto multiusuario y emulación de sistemas dinámicos para investigación y emulación en ingeniería”, que ha sido desarrollado por el Grupo

de Investigación en Control Industrial (GICI) en asocio con el Grupo de Percepción y Sistemas Inteligentes (PSI) de la Universidad del Valle, y ha sido cofinanciado por Colciencias en el 2007 [1]. El sistema existente se encuentra completamente diseñado en su estructura mecánica y se tiene un par de motores de DC que servirán como parte de los actuadores, pero no cuentan con la electrónica de actuación ni sus controladores; tampoco existe documentación de los motores de DC, planos del sistema, ni estrategia de control.

Dado lo anterior, el problema de ingeniería que se planteó fue “Poner en funcionamiento la planta de Péndulo Invertido para que sirva como herramienta para el estudio de estrategias de control de servomecanismos y de apoyo en los procesos de Educación en Ingeniería en la Universidad del Valle” vinculando la utilización de herramientas de software libre, de manera que contribuya a la reducción de costos de operación del Laboratorio de Automática de la Universidad del Valle. Este problema de ingeniería generó el objetivo general para el proyecto de “Controlar el péndulo invertido lineal del Laboratorio de Automática (Péndulo Invertido Simple – UV) en régimen de pequeña señal”.

Para la ejecución del objetivo planteado se propuso desarrollar los objetivos específicos:

- Caracterizar los elementos eléctricos y mecánicos de la plataforma existente.
- Diseñar e implementar la etapa de potencia de la planta del péndulo invertido para ponerla en funcionamiento.
- Integrar el software y hardware para la puesta en operación del péndulo invertido.
- Diseñar una estrategia de control para el péndulo invertido en el régimen de pequeña señal.
- Validar el correcto funcionamiento de la planta del péndulo invertido.

El informe se encuentra organizado de la siguiente manera. En el Capítulo 2 se presenta el marco teórico que incluye la descripción y modelado de un péndulo invertido simple sobre carro, también llamado lineal. En el Capítulo 3 se muestra la caracterización eléctrica y mecánica de la planta del Péndulo Invertido Simple en estudio. En el Capítulo 4 se describe el diseño e implementación de la electrónica apropiada para el control y utilización de los motores de la planta como actuadores y de los elementos de sensórica adecuados para el sensado de posición angular del péndulo y posición lineal del carro además de la integración entre el software de Scilab, la tarjeta de adquisición de datos, los componentes electrónicos relacionados con la sensórica y los actuadores. En el Capítulo 5 se muestra el diseño y simulación de una estrategia de control para la planta, cuya validación en la planta real se presenta en el Capítulo 6. En el Capítulo 7 se presentan las conclusiones obtenidas tras el desarrollo de este proyecto además de algunas propuestas sobre trabajos que pueden desarrollarse a partir de lo aquí obtenido.

Capítulo 2

Marco Teórico

En este capítulo se describe la planta de péndulo invertido, el modelado de un péndulo invertido lineal tanto como función de transferencia como en espacio de estados.

La Sección 2.1 presenta la descripción del péndulo invertido y algunos tipos de plantas de péndulo invertido existentes. En la Sección 2.2 se busca encontrar una expresión matemática que represente el comportamiento físico del sistema, realizando así el modelado de la planta y obteniendo como resultado una función de transferencia junto con las variables de estado de la planta.

2.1. Descripción del Péndulo Invertido

Un péndulo invertido es un sistema mecánico que consiste en un dispositivo móvil que sostiene una varilla conectada en un extremo a una masa y en el otro a un eje que puede pivotar bidimensionalmente, teniendo en cuenta que el centro de masa del sistema se encuentra por encima del punto de pivote.

Mientras que un péndulo normal es un sistema estable ya que cuelga hacia abajo, un péndulo invertido es un sistema inestable y debe ser constantemente balanceado para mantenerlo levantado. Un ejemplo típico de control de péndulo invertido es mantener en equilibrio una escoba hacia arriba, apoyada sobre la mano de una persona, cuyo control se hace mediante oscilaciones constantes en la posición horizontal de la mano, observando la parte superior de la escoba.

Como un medio experimental relativamente ideal para la enseñanza y la investigación, los sistemas de control de péndulo invertido se han usado ampliamente como patrón de comparación para algoritmos de control. Otra cualidad de este sistema es la similitud de su dinámica con la de un transporte aéreo, con la de un robot bípedo y con la de muchos mecanismos más complejos, en los cuales se puede adaptar fácilmente los algoritmos de control del péndulo invertido.

Los tipos de péndulo invertido incluyen el péndulo invertido simple lineal o sobre carro, el péndulo invertido doble y triple lineal, el péndulo de Furuta, el péndulo de rueda inercial, entre otros.

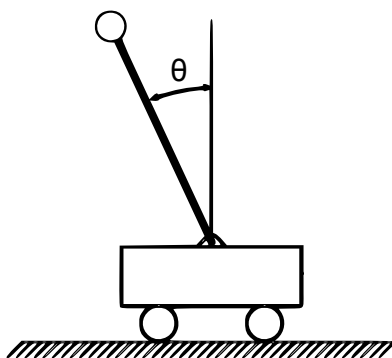


Figura 2.1: Péndulo Invertido Simple Lineal

En la figura 2.1 se muestra un Péndulo Invertido Simple Lineal. En este tipo de péndulo invertido el centro de masa se encuentra montado sobre un carro que puede moverse horizontalmente sobre un eje lineal.

El péndulo invertido doble (Figura 2.2) o triple, corresponde a un péndulo invertido lineal con una o dos articulaciones adicionales respectivamente. Esto genera un reto adicional para la estrategia de control a implementar.

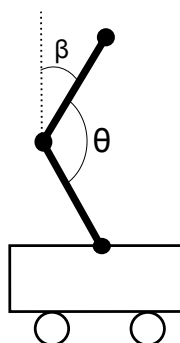


Figura 2.2: Péndulo Invertido Doble sobre Carro

El péndulo de Furuta o péndulo invertido rotacional consiste en un brazo controlado con capacidad de rotar en el plano horizontal y un péndulo unido a este brazo con capacidad libre de rotación en el plano vertical.

En el péndulo de Rueda Inercial, un motor sujeto al extremo libre del péndulo permite

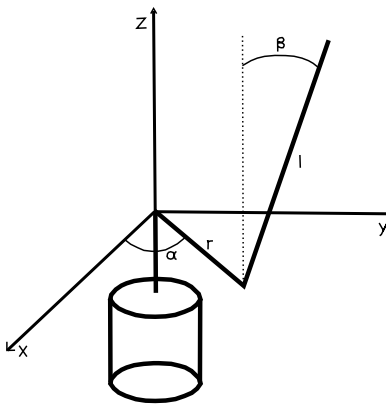


Figura 2.3: Péndulo de Furuta

controlar el giro de una rueda de inercia de manera que es con este que se controla la posición angular del péndulo.

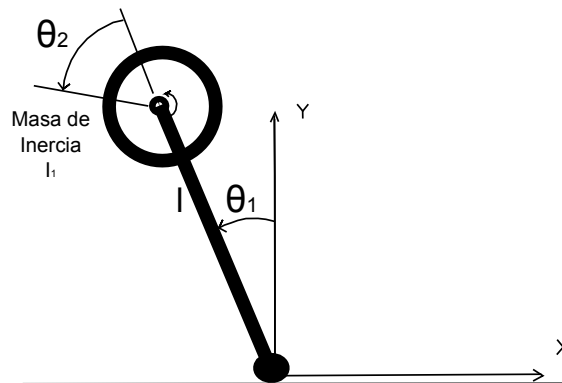


Figura 2.4: Péndulo de Rueda Inercial

La planta de péndulo invertido presente en el Laboratorio de Automática de la Universidad del Valle corresponde a un Péndulo Invertido Simple Lineal, con la adición de un riel inclinable mediante la acción de un segundo motor.

Donde:

Cuadro 2.1: Simbolos y parámetros correspondientes en el DCL de la planta

Símbolo	Parámetro
M	Masa del carro $[kg]$
m	Masa del péndulo $[kg]$
g	Aceleración gravitacional $[m/s^2]$
N_r	Fuerza normal del riel sobre el carro $[N]$
b	Coefficiente de fricción del carro $[N / m/s]$
l	Longitud del péndulo $[m]$
I	Momento de inercia de masa del péndulo $[kg * m^2]$
F	Fuerza aplicada al carro $[N]$
P	Fuerza de reacción vertical $[N]$
N	Fuerza de reacción horizontal $[N]$
x	Posición coordenada del carro $[m]$
v	Velocidad horizontal del carro $[m/s]$
a	Aceleración horizontal del carro $[m/s^2]$
θ	Ángulo del péndulo respecto a la vertical inferior $[rad]$
ω	Velocidad angular del péndulo $[rad/s]$
α	Aceleración angular del péndulo $[rad/s^2]$
ϕ	Ángulo del péndulo respecto a la vertical superior $[rad]$

Primero se analizan las fuerzas actuantes en el eje horizontal para el carro y el péndulo obteniendo:

$$\begin{aligned}
 \sum F_{x_{carro}} &= M\ddot{x} \\
 M\ddot{x} &= F - b\dot{x} - N \\
 N &= F - b\dot{x} - M\ddot{x}
 \end{aligned} \tag{2.1}$$

$$\begin{aligned}
 \sum F_{x_{pendulo}} &= m\ddot{x} \\
 m\ddot{x} &= N - ml\ddot{\theta} \cos \theta + ml\dot{\theta}^2 \sin \theta \\
 N &= m\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta
 \end{aligned} \tag{2.2}$$

Al igualar las ecuaciones 2.1 y 2.2 se obtiene:

$$\begin{aligned}
 F - b\dot{x} - M\ddot{x} &= m\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \\
 (M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta &= F
 \end{aligned} \tag{2.3}$$

La Ecuación 2.3 representa la primera ecuación dinámica del sistema. Para encontrar una ecuación dinámica adicional, se utilizan las fuerzas perpendiculares al péndulo además de las fuerzas de reacción presentes en la articulación que une los cuerpos (péndulo y carro). Dado lo anterior, se obtiene la siguiente ecuación.

$$P \sin \theta + N \cos \theta - mg \sin \theta = m\ddot{x} \cos \theta + ml\ddot{\theta} \quad (2.4)$$

Además, el torque debido a las diversas fuerzas sobre la barra del péndulo nos brinda otra ecuación que permita, junto a la Ecuación 2.4, obtener la segunda ecuación dinámica del sistema.

$$\begin{aligned} T &= \sum F_i r_i = I\ddot{\theta} \\ I\ddot{\theta} &= -Pl \sin \theta - Nl \cos \theta \end{aligned} \quad (2.5)$$

Combinando las ecuaciones 2.4 y 2.5 se obtiene:

$$\begin{aligned} m\ddot{x} \cos \theta + ml\ddot{\theta} + mg \sin \theta &= -I\ddot{\theta}/l \\ (ml^2 + I)\ddot{\theta} + mlg \sin \theta &= -ml\ddot{x} \cos \theta \end{aligned} \quad (2.6)$$

Las ecuaciones 2.3 y 2.6 no son ecuaciones lineales, por lo que es necesario linealizarlas y así poder aplicar estrategias de control de sistemas lineales.

Dado que el control del péndulo invertido se hace en el punto de equilibrio inestable o superior, las ecuaciones se deben linealizar utilizando el ángulo del péndulo respecto a la línea vertical superior es decir $\phi \rightarrow 0$.

Si θ es el ángulo del péndulo respecto a la vertical inferior, $\theta = \phi + \pi$ de manera que se obtienen las siguientes expresiones.

$$\begin{aligned} \cos \theta &= \cos(\pi + \phi) \approx -1 \\ \sin \theta &= \sin(\pi + \phi) \approx -\phi \\ \dot{\theta}^2 &= \dot{\phi}^2 \approx 0 \\ \dot{\theta} &= \dot{\phi} \\ \ddot{\theta} &= \ddot{\phi} \end{aligned}$$

Al sustituir estas expresiones en las ecuaciones 2.3 y 2.6, y considerar que la fuerza F corresponde a la señal de control u , se obtienen dos ecuaciones linealizadas de nuestro sistema con un rango de operación de $\phi \rightarrow 0$.

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad (2.7)$$

$$(ml^2 + I)\ddot{\phi} - mlg\phi = ml\ddot{x} \quad (2.8)$$

2.2.2. Funciones de Transferencia del Sistema

Para hallar la función de transferencia del sistema linealizado, se aplica la transformada de Laplace a las ecuaciones 2.7 y 2.8 y se obtienen las siguientes ecuaciones.

$$(M + m)s^2X(s) + bsX(s) - mls^2\phi(s) = U(s) \quad (2.9)$$

$$(ml^2 + I)s^2\phi(s) - mlg\phi(s) = mls^2X(s) \quad (2.10)$$

Una función de transferencia se define como la razón entre una señal de salida y una señal de entrada en el dominio de la frecuencia (s), por lo que es necesario encontrar dos funciones de transferencia en el sistema del péndulo invertido simple lineal. La función $H_\phi(s)$ representará el comportamiento del ángulo $\phi(s)$ tras la excitación de una entrada $U(s)$, y la función $H_X(s)$ describirá la dinámica de la posición lineal del carro $X(s)$ al excitar el sistema con la misma entrada $U(s)$.

Para encontrar $H_\phi(s)$ es necesario despejar $X(s)$ de la ecuación 2.10 y a continuación reemplazarla en la ecuación 2.9 con lo que se obtiene:

$$X(s) = \left[\frac{ml^2 + I}{ml} - \frac{g}{s^2} \right] \phi(s)$$

$$\left[(M + m) \left(\frac{ml^2 + I}{ml} - \frac{g}{s^2} \right) s^2 + b \left(\frac{ml^2 + I}{ml} - \frac{g}{s^2} \right) s - mls^2 \right] \phi(s) = U(s) \quad (2.11)$$

Así, la función de transferencia $H_\phi(s)$ se define como:

$$\begin{aligned} H_\phi(s) &= \phi(s)/U(s) \\ &= \frac{mls^2}{[(M + m)(ml^2 + I) - (ml)^2]s^4 + [b(ml^2 + I)]s^3 - [(M + m)mlg]s^2 - [bmlg]s} \\ &= \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(ml^2 + I)}{q}s^3 - \frac{(M + m)mlg}{q}s^2 - \frac{bmlg}{q}s} \end{aligned} \quad (2.12)$$

En donde $q = [(M + m)(ml^2 + I) - (ml)^2]$.

La ecuación 2.12 tiene tanto un cero como un polo en el origen ($s = 0$) por lo que estos pueden cancelarse mutuamente. De esta manera la función de transferencia $H_\phi(s)$ es:

$$H_\phi(s) = \frac{\frac{ml}{q}s}{s^3 + \frac{b(ml^2 + I)}{q}s^2 - \frac{(M + m)mlg}{q}s - \frac{bmlg}{q}} \quad [rad/N] \quad (2.13)$$

Para obtener la función de transferencia $H_X(s)$ se hace un procedimiento similar al anterior despejando $\phi(s)$ de la ecuación 2.10 y reemplazándolo en la ecuación 2.9 obteniendo:

$$\phi(s) = \left[\frac{mls^2}{(ml^2 + I)s^2 - mlg} \right] X(s)$$

$$\left[(M+m)s^2 + bs - \frac{(mls^2)^2}{(ml^2 + I)s^2 - mlg} \right] X(s) = U(s) \quad (2.14)$$

La función de transferencia $H_X(s)$ se define como:

$$\begin{aligned} H_X(s) &= X(s)/U(s) \\ &= \frac{(ml^2 + I)s^2 - mlg}{[(M+m)(ml^2 + I) - (ml)^2]s^4 + [b(ml^2 + I)]s^3 - [(M+m)mlg]s^2 - [bmlg]s} \end{aligned}$$

Teniendo en cuenta que:

$$q = I(M+m) + Mml^2 \quad (2.15)$$

$H_X(s)$ será:

$$H_X(s) = \frac{\frac{(ml^2 + I)s^2 - mlg}{q}}{s^4 + \frac{b(ml^2 + I)}{q}s^3 - \frac{(M+m)mlg}{q}s^2 - \frac{bmlg}{q}s} \quad (2.16)$$

2.2.3. Representación en Espacio de Estados

La representación de un sistema en espacio de estados corresponde a otra manera de modelar un sistema físico. Los estados, en conjunto con las funciones de entrada y las ecuaciones dinámicas del sistema, permiten conocer las salidas del sistema y una visión futura de los mismos estados. [3]

Las ecuaciones dinámicas se representan en forma matricial de manera que $\vec{x}$ es el Vector de estado de tamaño $(n \times 1)$, $\vec{u}$ es el Vector de entrada de tamaño $(p \times 1)$ y $\vec{y}$ es el Vector de salida de tamaño $(q \times 1)$, donde n es el numero de estados, p es el numero de entradas y q es el numero de salidas del sistema.

Así, en un sistema lineal invariante en el tiempo es posible representar las ecuaciones dinámicas de la siguiente manera.

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} \quad (2.17)$$

$$\vec{y} = C\vec{x} + D\vec{u} \quad (2.18)$$

En esta representación, A es una matriz de tamaño $(n \times n)$ llamada matriz del sistema, B es una matriz de tamaño $(n \times p)$, C es una matriz de tamaño $(q \times n)$ y D una matriz de tamaño $(q \times p)$.

En caso de realizarse una representación del sistema en tiempo discreto, la representación en espacio de estados se describe de según las ecuaciones 2.19 y 2.20.

$$\overrightarrow{x(k+1)} = G\overrightarrow{x(k)} + H\overrightarrow{u(k)} \quad (2.19)$$

$$\overrightarrow{y(k+1)} = C\overrightarrow{x(k)} + D\overrightarrow{u(k)} \quad (2.20)$$

Para el caso del sistema de péndulo invertido simple lineal, el vector de estado se compondrá de la posición y velocidad del carro, y de la posición angular y velocidad angular del péndulo. Las variables de salida sensables serán la posición del carro y posición angular del péndulo, y la única variable de entrada corresponderá al voltaje aplicado al motor como elemento actuador. Dado esto, los vectores de estado, salida y entrada del sistema serán:

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \quad (2.21)$$

$$\vec{y} = \begin{bmatrix} x \\ \phi \end{bmatrix} \quad (2.22)$$

$$\vec{u} = \begin{bmatrix} V \end{bmatrix} \quad (2.23)$$

Es necesario entonces utilizar una relación entre la fuerza aplicada al carro F , que es la variable de entrada utilizada para las funciones de transferencia y el modelado matemático, y el voltaje V aplicado al motor que mueve al carro.

$$u = F = \frac{K_m K_g}{R_m R_p} V - \frac{K_m^2 K_g^2}{R_m R_p^2} \dot{x} \quad (2.24)$$

Donde K_m es la constante de torque del motor, K_g es la relación de transmisión del motor y la polea que mueve al carro, R_m es la resistencia de armadura del motor y R_p es el radio de la polea movida por el motor [8].

Tras definir los vectores de estado, entrada y salida del sistema se deben encontrar los componentes de las matrices A , B , C y D . Esto es posible utilizando las dos ecuaciones linealizadas que describen la dinámica del sistema (ecuaciones 2.7 y 2.8) y la ecuación que relaciona el voltaje aplicado al motor con la fuerza aplicada por el motor sobre el carro (ecuación 2.24).

Para encontrar una expresión de $\ddot{x}$ en función de los cuatro componentes del vector de estados y del vector de entrada, es posible despejar $\ddot{\phi}$ de la ecuación 2.8, remplazar esta variable en la ecuación 2.7 y posteriormente remplazar la variable u por la ecuación 2.24.

$$\ddot{\phi} = \frac{ml\ddot{x} + mlg\phi}{ml^2 + I}$$

$$(M + m)\ddot{x} + b\dot{x} - ml \left[\frac{ml\ddot{x} + mlg\phi}{ml^2 + I} \right] = u$$

$$(M+m)\ddot{x} + b\dot{x} - ml \left[\frac{ml\ddot{x} + mlg\phi}{ml^2 + I} \right] = \frac{K_m K_g}{R_m R_p} V - \frac{K_m^2 K_g^2}{R_m R_p^2} \dot{x} \quad (2.25)$$

Al despejar $\ddot{x}$ de la ecuación 2.25 y remplazar q de la ecuación 2.15 se obtiene:

$$\ddot{x} = \left[\frac{m^2 l^2 g}{q} \right] \phi - \left[\frac{b(ml^2 + I)}{q} + \frac{K_m^2 K_g^2 (I + ml^2)}{q R_m R_p^2} \right] \dot{x} + \left[\frac{K_m K_g (ml^2 + I)}{q R_m R_p} \right] V \quad (2.26)$$

Un procedimiento similar se utiliza para encontrar una expresión de $\ddot{\phi}$ en función de los vectores de estado y entrada. En este caso se despeja la variable $\ddot{x}$ de la ecuación 2.8 para remplazarla en la ecuación 2.7.

$$\begin{aligned} \ddot{x} &= \frac{(ml^2 + I)\ddot{\phi} - mlg\phi}{ml} \\ (M+m) \left[\frac{(ml^2 + I)\ddot{\phi} - mlg\phi}{ml} \right] + b\dot{x} - ml\ddot{\phi} &= u \\ (M+m) \left[\frac{(ml^2 + I)\ddot{\phi} - mlg\phi}{ml} \right] + b\dot{x} - ml\ddot{\phi} &= \frac{K_m K_g}{R_m R_p} V - \frac{K_m^2 K_g^2}{R_m R_p^2} \dot{x} \end{aligned} \quad (2.27)$$

Al despejar $\ddot{\phi}$ de la ecuación 2.27 se obtiene:

$$\ddot{\phi} = \left[\frac{(M+m)mlg}{q} \right] \phi - \left[\frac{bml}{q} + \frac{K_m^2 K_g^2 ml}{q R_m R_p^2} \right] \dot{x} + \left[\frac{K_m K_g ml}{q R_m R_p} \right] V \quad (2.28)$$

Las expresiones contenidas en las ecuaciones 2.26 y 2.28, permiten conocer las matrices A y B de manera certera.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-b(ml^2 + I)}{q} - \frac{K_m^2 K_g^2 (I + ml^2)}{q R_m R_p^2} & \frac{m^2 l^2 g}{q} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-bml}{q} - \frac{K_m^2 K_g^2 ml}{q R_m R_p^2} & \frac{(M+m)mlg}{q} & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \frac{K_m K_g (ml^2 + I)}{q R_m R_p} \\ 0 \\ \frac{K_m K_g ml}{q R_m R_p} \end{bmatrix}$$

Donde tal y como se muestra en la ecuación 2.15: $q = I(M+m) + Mml^2$.

Dado que los componentes del vector de salida corresponden a dos componentes del vector de estados, sin incluir alguna componente del vector de entrada se obtiene como matrices C y D :

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

Capítulo 3

Caracterización Eléctrica y Mecánica de la Planta del Péndulo Invertido

En este capítulo se describe en detalle el procedimiento de caracterización eléctrica y mecánica de la planta del Péndulo Invertido Simple - UV (PIS-UV).

La Sección 3.1 presenta el trabajo matemático realizado para la identificación de las ecuaciones que describen la dinámica del motor de imán permanente de DC que sirve como actuador de la planta. Además se expone el desarrollo matemático para encontrar una función de transferencia general para este tipo de motores.

Los procedimientos realizados para obtener los parámetros propios del motor en estudio, se muestran en la Sección 3.2. Esta sección explica paso a paso la manera de encontrar los parámetros necesarios para conocer la función de transferencia propia del motor en estudio, tales como la constante eléctrica, la constante de torque, el coeficiente de fricción y el momento de inercia del motor, entre otros.

En la Sección 3.4 se presenta un resumen de los parámetros encontrados y la función de transferencia propia del motor en estudio obtenida utilizando dichos parámetros encontrados experimentalmente. En esta sección también se muestra una comparación entre la respuesta dinámica del motor real y la respuesta dinámica del motor simulado utilizando la función de transferencia encontrada.

La caracterización mecánica de la planta se lleva a cabo en la Sección 3.3. En esta sección se muestra el plano frontal de la planta creado utilizando un software de diseño asistido por computadora (CAD). Además se muestran los valores de variables mecánicas propias de la planta tales como masa del carro y péndulo, fricción del carro, momento de inercia del péndulo, radio de la polea que mueve al carro, longitud del péndulo, longitud del centro de gravedad del péndulo y radio del cilindro del péndulo.

3.1. Modelado del Motor Brushless de DC

3.1.1. Modelado Matemático del Motor

Un motor de corriente continua se puede modelar utilizando el circuito equivalente que se muestra en la Figura 3.1.

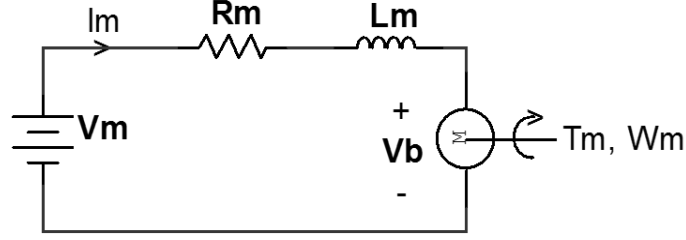


Figura 3.1: Circuito equivalente de Motor DC

Donde V_m es el voltaje aplicado al motor, I_m representa la corriente a través de la armadura del motor, R_m es la resistencia de la armadura del motor, L_m es la inductancia del motor, V_b es la fuerza contra-electromotriz generada por el motor, T_m es el torque del motor y ω_m es la velocidad angular del rotor.

Utilizando la ley de voltaje de Kirchhoff, en la malla única de la Figura 3.1, se obtiene la siguiente ecuación.

$$-V_m + I_m R_m + L_m \frac{dI_m}{dt} + V_b = 0 \quad (3.1)$$

La fuerza contra-electromotriz V_b es proporcional a la velocidad del motor y a la intensidad del campo magnético. Si el motor analizado tiene un rotor con imán permanente, la fuerza contra-electromotriz se puede expresar de la siguiente manera.

$$V_b = K_e \omega_m \quad (3.2)$$

Donde K_e es la constante eléctrica del motor que representa un valor en Volts por unidad de velocidad angular. Así, al remplazar la Ecuación (3.2) en la Ecuación (3.1) se obtiene la siguiente ecuación.

$$V_m = I_m R_m + L_m \frac{dI_m}{dt} + K_e \omega_m \quad (3.3)$$

El rotor puede realizar el movimiento giratorio debido al torque electromagnético o par motor T_m que es producido por el campo magnético proveniente de la armadura. Dado que el campo magnético generado en una bobina es directamente proporcional con la corriente que pasa por ella, el torque electromagnético también será proporcional a la corriente que circula por el embobinado de la armadura.

Al realizar el movimiento giratorio, el motor arrastra una carga generando un par-motor proporcional a la aceleración angular y al momento de inercia del rotor. Además, al girar el rotor se presenta una fricción proporcional a la velocidad angular del mismo causando un torque T_f que se opone al movimiento rotacional del rotor.

Dadas las anteriores consideraciones, es posible obtener la ecuación mecánica del sistema.

$$\begin{aligned} J_m \frac{d\omega_m}{dt} &= T_m - T_f \\ J_m \frac{d\omega_m}{dt} &= K_m I_m - B_m \omega_m \end{aligned} \quad (3.4)$$

Donde J_m es el momento de inercia del motor, K_m es la constante de torque electromagnético del motor y B_m es el coeficiente de fricción del motor.

Para simplificar los modelos eléctrico y mecánico encontrados en las ecuaciones (3.3) y (3.4), es posible asumir que los valores de L_m y B_m son despreciables obteniendo las siguientes ecuaciones simplificadas.

$$V_m = I_m R_m + K_e \omega_m \quad (3.5)$$

$$J_m \frac{d\omega_m}{dt} = K_m I_m \quad (3.6)$$

3.1.2. Función de Transferencia del Motor

Al despejar la corriente de armadura del motor I_m de la Ecuación 3.6 se obtiene:

$$I_m = \frac{J_m}{K_m} \frac{d\omega_m}{dt} \quad (3.7)$$

La ecuación que describe el comportamiento del motor se obtiene tras remplazar la Ecuación 3.7 en la Ecuación 3.5 y se muestra a continuación.

$$V_m = \frac{J_m}{K_m} \frac{d\omega_m}{dt} R_m + K_e \omega_m \quad (3.8)$$

Para obtener la función de transferencia del motor se debe primero aplicar la transformada de Laplace a la Ecuación 3.8.

$$\begin{aligned} V_m(s) &= \mathcal{L} \left\{ (K_e \omega_m(t) + \frac{R_m J_m}{K_m} \frac{d\omega_m(t)}{dt}) \right\} \\ &= \left(K_e + s \frac{R_m J_m}{K_m} \right) \omega_m(s) \\ &= \left(\frac{s R_m J_m + K_e K_m}{K_m} \right) \omega_m(s) \end{aligned} \quad (3.9)$$

Al despejar $\omega_m(s)/V_m(s)$ de la Ecuación 3.9 se obtiene la función de transferencia del motor.

$$\frac{\omega_m(s)}{V_m(s)} = \frac{K_m}{sR_mJ_m + K_eK_m} \quad (3.10)$$

Simplificando la Ecuación 3.10 se obtiene la función de transferencia simplificada del motor de DC.

$$\frac{\omega_m(s)}{V_m(s)} = \frac{K}{\tau s + 1} \quad \therefore K = \frac{1}{K_e}, \tau = \frac{R_mJ_m}{K_eK_m} \quad (3.11)$$

3.2. Parametrización del Motor de DC

3.2.1. Determinación de la Zona Muerta y la Zona Lineal del Motor

Para conocer las zonas de operación del motor de DC en estudio se debe utilizar una fuente variable de DC, un voltímetro, un amperímetro y un tacómetro, y realizar la conexión como se muestra en la Figura 3.2.

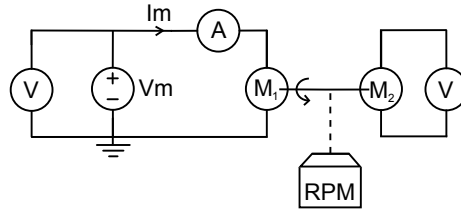


Figura 3.2: Conexión para determinación de zona muerta y lineal del motor

Se utilizó una fuente programable BK PRECISION XLN6024 mostrada en la Figura 3.3. Esta fuente puede entregar voltajes en el rango de 0 a 60 V y corrientes en el rango de 0 a 24 A. Además, mediante un Display LCD muestra el voltaje V_m y la corriente I_m entregada con una incertidumbre de $\pm 0,001$ V y $\pm 0,001$ A respectivamente, con lo que se omite la necesidad de un voltímetro y un amperímetro externos.



Figura 3.3: Fuente Programable BK PRECISION XLN6024

Para realizar las mediciones de velocidad angular se utilizó un Tacómetro Digital DT-2234C+ como el mostrado en la Figura 3.4. Este es un tacómetro láser que permite

realizar mediciones de entre 2,5 y 99999 revoluciones por minuto [RPM] y tiene una distancia de detección de hasta 1,8 metros. Las mediciones de velocidad angular de este instrumento tienen una incertidumbre de $\pm 0,1 \text{ RPM}$.

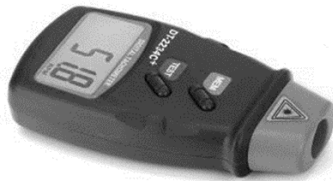


Figura 3.4: Tacómetro Digital DT-2234C+

Una vez realizada la conexión mostrada en la 3.2, se debe iniciar el procedimiento de toma de datos de corriente I_m , velocidad angular ω_m para múltiples valores del voltaje de excitación V_m y de V_b medido en los bornes del motor que funcionará como generador. Estas mediciones se deberán realizar tanto con un gradiente positivo de voltaje como con un gradiente negativo, de manera que se pueda identificar posible presencia de histéresis. Para identificar adecuadamente la zona muerta del motor es necesario realizar pequeñas variaciones en V_m hasta que el rotor empiece a girar. Los datos obtenidos en estas mediciones se muestran en las Figuras 3.5 y 3.6.

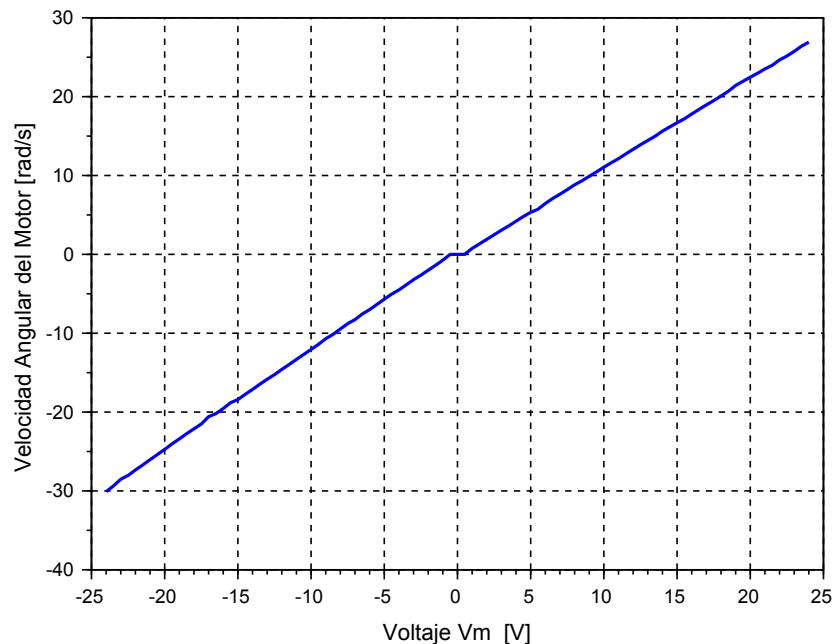


Figura 3.5: Caracterización Estática del Motor en Estudio

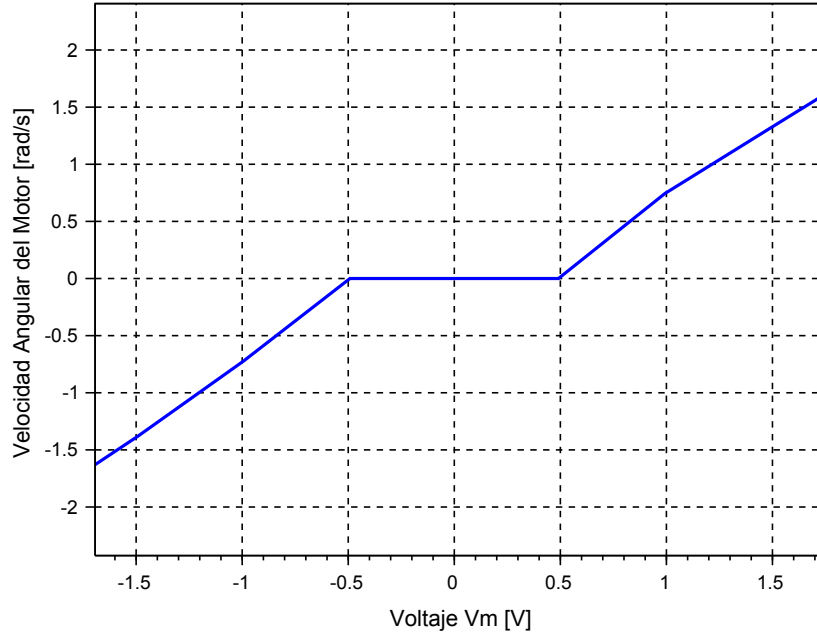


Figura 3.6: Zona Muerta del Motor en Estudio

En la Figura 3.6 es posible observar que la zona muerta del motor se encuentra en el rango entre $-0,5$ y $+0,5$ V mientras que en la Figura 3.5 se observa que la zona lineal comprende valores de voltajes absolutos mayores a $0,5$ V.

3.2.2. Resistencia de Armadura R_m

Este procedimiento puede realizarse utilizando un multímetro de manera que se mida la resistencia en los bornes del motor de DC cuando este esté totalmente desconectado de la excitación y de otros implementos. Para obtener un resultado de mayor confiabilidad, se deben tomar múltiples mediciones girando un poco el rotor del motor antes de tomar cada muestra y por último tomar como valor de R_m el promedio de las muestras obtenidas utilizando un multímetro. En el Cuadro 3.1 se muestra el resultado de las mediciones obtenidas.

De los 20 datos de R_m mostrados en el Cuadro 3.1 se obtiene un promedio de:

$$R_m = 1,24 \, \Omega$$

En otro método utilizado para la medición del parámetro R_m se considera la Ecuación (3.5). Al aplicar un Voltaje V_m al motor y bloquear el rotor de manera que este no pueda girar ($\omega_m = 0$), se podrá medir la resistencia de armadura R_m conociendo la corriente que circula por la armadura I_m y el voltaje V_m .

$$V_m = I_m R_m + K_e \omega_m|_{\omega_m=0}$$

Cuadro 3.1: Resistencia de armadura medida utilizando un multímetro

Muestra [#]	R_m [Ω]
1	1,2
2	1,4
3	1,3
4	1,2
5	1,2
6	1,2
7	1,3
8	1,3
9	1,4
10	1,5
11	1,2
12	1,2
13	1,1
14	1,1
15	1,3
16	1,2
17	1,2
18	1,2
19	1,1
20	1,2

$$R_m = \frac{V_m}{I_m} \quad si \quad \omega_m = 0 \quad (3.12)$$

Para esta prueba se utiliza un voltaje $V_m = 3 \text{ V}$, debido a que debe excitarse el motor con un voltaje que se encuentre dentro de su zona lineal pero que no vaya a representar algún peligro, como picos de corriente muy altos, en el momento que no se le permita al rotor girar libremente.

El rotor del motor en estudio posee un orificio pequeño, el cual es útil a la hora de inhibir su movimiento rotacional.

En este procedimiento se deben realizar diversas mediciones con voltajes V_m y $-V_m$ deteniendo el motor en diversas posiciones y tomando nota de los valores de corriente I_m en cada muestra.

El Cuadro 3.2 muestra los datos obtenidos tras realizar la prueba de medición de corriente con rotor detenido.

Cuadro 3.2: Datos de V_m , I_m y R_m con rotor detenido

V_m [V]	I_m [A]	R_m [Ω]
2,994	2,392	1,252
2,994	2,436	1,229
2,994	2,414	1,240
2,994	2,256	1,327
2,994	2,454	1,220
2,994	2,399	1,248
2,994	2,418	1,238
2,994	2,403	1,246
2,994	2,427	1,234
2,994	2,364	1,266
-2,994	-2,419	1,238
-2,994	-2,393	1,251
-2,994	-2,395	1,250
-2,994	-2,415	1,240
-2,994	-2,493	1,201
-2,994	-2,514	1,191
-2,994	-2,512	1,192
-2,994	-2,476	1,209
-2,994	-2,536	1,181
-2,994	-2,487	1,204

De los 20 datos de R_m mostrados en el Cuadro 3.2 se obtiene un promedio de:

$$R_m = 1,233 \, \Omega$$

Al promediar los valores de R_m obtenidos mediante los dos métodos experimentales, se obtiene un valor de Resistencia de Armadura:

$$R_m = 1,2365 \, \Omega$$

3.2.3. Constante Eléctrica del Motor K_e

Para encontrar la constante eléctrica del motor K_e se considera la ecuación (3.5). Dado que ya se conoce el valor de R_m , se puede determinar la constante eléctrica del motor fácilmente. Utilizando los datos de V_m , I_m y ω_m , obtenidos durante la caracterización dinámica del motor, y el valor de R_m obtenido experimentalmente es posible encontrar el valor de K_e por cada valor de voltaje de excitación V_m . El valor de K_e será

el promedio de todas las constantes eléctricas calculadas dentro de la zona lineal. Los diferentes valores de K_e calculados se muestran en la Figura 3.7.

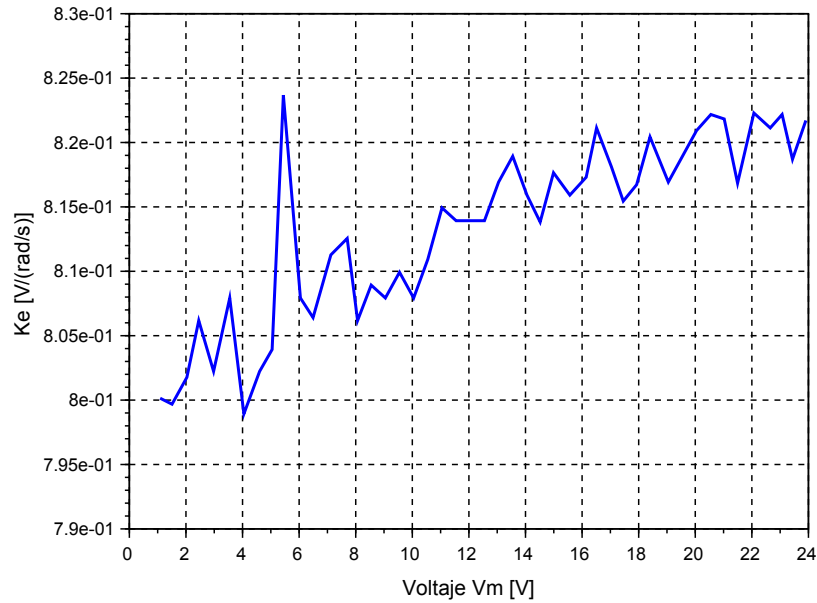


Figura 3.7: Constante eléctrica del motor para cada valor de V_m

De los datos de K_e mostrados en la Figura 3.7 se obtiene un promedio aritmético de:

$$K_e = 0,81802 \frac{V}{Rad/s}$$

Otro método para obtener el valor de la constante eléctrica del motor K_e es utilizando el motor en estudio como un generador. Para ello, como se muestra en la Figura 3.2 se ancla el rotor del motor en estudio al rotor de otro motor que se alimentara con V_m . La conexión mecánica de los motores se muestra en la Figura 3.8.

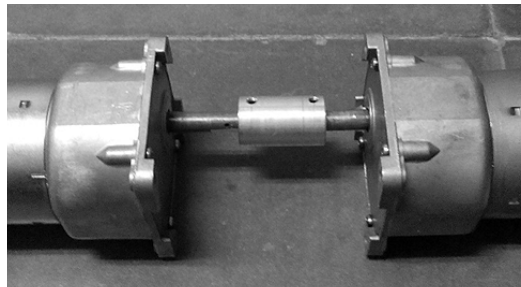


Figura 3.8: Conexión Mecánica entre los rotores para medición de K_e

El voltaje generado por el motor en estudio a partir del movimiento rotacional al

que es sometido por el motor auxiliar será la fuerza contra-electromotriz V_b . Midiendo V_b y ω_m es posible conocer K_e utilizando la Ecuación 3.2. Dada la Ecuación 3.2, encontrar la pendiente de la gráfica de la fuerza contra-electromotriz en función de la velocidad angular del rotor, corresponde a encontrar la constante eléctrica del motor K_e .

En la Figura 3.9 se muestra el Voltaje V_b en función de la velocidad angular del rotor ω_m .

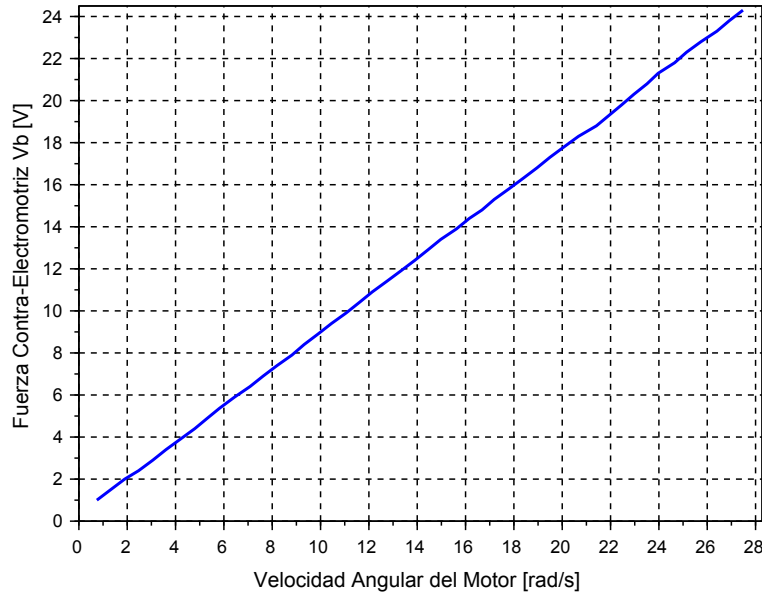


Figura 3.9: Fuerza contra-electromotriz Vs. Velocidad angular del motor

La pendiente de la línea de tendencia de la gráfica presente en la Figura 3.9 tiene un valor de 0,8135, por lo tanto se obtiene un valor de:

$$K_e = 0,8135 \frac{V}{Rad/s}$$

Promediando los valores de K_e obtenidos mediante los dos métodos experimentales, se obtiene un valor de la constante eléctrica del motor de:

$$K_e = 0,81576 \frac{V}{Rad/s}$$

3.2.4. Constante de Torque K_m del Motor

La constante de torque del motor es la constante de proporcionalidad existente entre el par motor T_m y la corriente de armadura del motor I_m , de manera que el par motor puede expresarse como $T_m = K_m I_m$.

La potencia generada por el par motor es directamente proporcional a la velocidad angular del rotor ω_m . Así, es posible expresar la potencia mecánica de salida del motor

P_{out} como:

$$P_{out} = \omega_m T_m = \omega_m K_m I_m \quad (3.13)$$

La potencia eléctrica de entrada P_{in} del motor es directamente proporcional al voltaje de entrada V_m y a la corriente de armadura I_m , de manera que es posible expresarse como:

$$P_{in} = V_m I_m \quad (3.14)$$

Al remplazar la Ecuación 3.5 en la Ecuación 3.14 se obtiene la siguiente expresión para la potencia eléctrica de entrada en el motor.

$$P_{in} = I_m^2 R_m + \omega_m I_m K_e \quad (3.15)$$

Las pérdidas de potencia de entrada en el motor se deben a la resistencia de armadura R_m por lo que la potencia perdida P_p se puede expresar como:

$$P_p = I_m^2 R_m \quad (3.16)$$

La ley de la conservación de la energía determina que la potencia de entrada es igual a la potencia de salida sumada a las pérdidas presentes en el sistema. Utilizando las ecuaciones 3.13, 3.15 y 3.16 y la ley de conservación de la energía se obtiene la ecuación 3.17.

$$\begin{aligned} P_{in} &= P_{out} + P_p \\ I_m^2 R_m + \omega_m I_m K_e &= \omega_m K_m I_m + I_m^2 R_m \end{aligned} \quad (3.17)$$

Despejando la Ecuación 3.17 se obtiene una igualdad entre la constante eléctrica y la constante de torque del motor, como se muestra en la Ecuación 3.18.

$$K_m [N * m/A] = K_e \left[\frac{V}{Rad/s} \right] \quad (3.18)$$

Con la Ecuación 3.18 se obtiene que la constante de torque del motor K_m tiene un valor de:

$$K_m = 0,81576 \quad N * \frac{m}{A}$$

3.2.5. Coeficiente de Fricción del Motor B_m

Utilizando la Ecuación (3.4) y asumiendo que el rotor gira a una velocidad angular constante se obtiene la Ecuación (3.19).

$$\begin{aligned} J_m \frac{d\omega_m}{dt} &= K_m I_m - B_m \omega_m \Big|_{\frac{d\omega_m}{dt}=0} \\ K_m I_m &= B_m \omega_m \\ B_m &= \frac{\omega_m}{K_m I_m} \end{aligned} \quad (3.19)$$

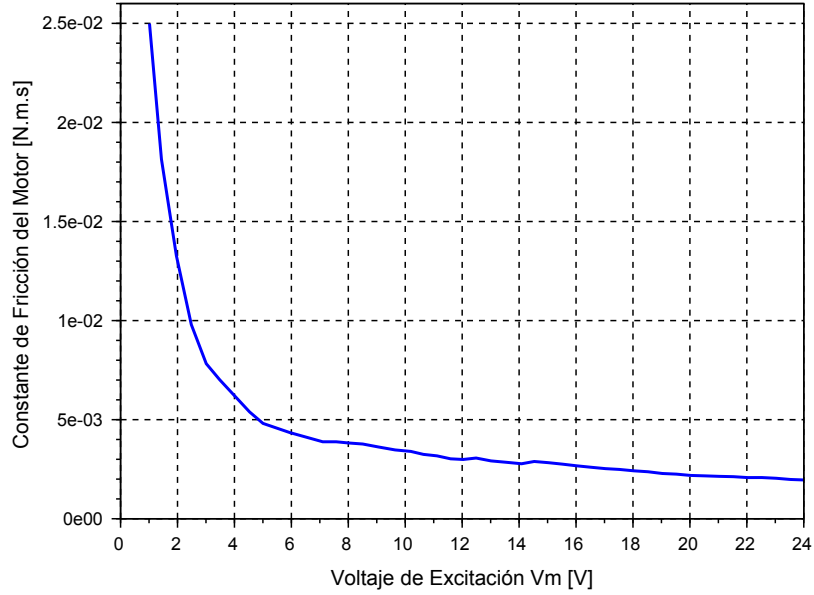


Figura 3.10: Coeficiente de Fricción del Motor Vs. Voltaje de Excitación

En la Figura 3.10 se muestran los datos de B_m encontrados para cada valor de V_m utilizando la Ecuación (3.19).

El valor promedio de B_m encontrado es el de:

$$B_m = 0,00365924 \text{ N} * \text{m} * \text{s}$$

Este será el valor del coeficiente de fricción del motor que se utilizará en el modelo del motor.

3.2.6. Inductancia L_m y constante de tiempo eléctrica del motor τ_e

Un método para conocer el valor de la inductancia L_m es midiendo con un inductómetro la inductancia en los bornes del motor en estudio. Se deben tomar múltiples mediciones con el rotor en diversas posiciones angulares para posteriormente promediar estas mediciones.

En el Cuadro 3.3 se muestran los datos de L_m obtenidos mediante este procedimiento. El promedio de los valores de L_m mostrados en el Cuadro 3.3 es:

$$L_m = 2,3905 \text{ mH}$$

Ya con el valor de L_m encontrado, es posible encontrar la constante de tiempo del motor τ_e teniendo en cuenta que el circuito equivalente del motor es un circuito RL . El valor de la constante de tiempo de un circuito RL se calcula utilizando la Ecuación (3.20)

$$\tau_e = \frac{L_m}{R_m} \quad (3.20)$$

Cuadro 3.3: Inductancia del motor medida con Inductómetro

Muestra [#]	L_m [mH]
1	2,32
2	2,34
3	2,38
4	2,52
5	2,42
6	2,41
7	2,38
8	2,37
9	2,38
10	2,39
11	2,39
12	2,37
13	2,35
14	2,39
15	2,4
16	2,41
17	2,4
18	2,39
19	2,39
20	2,41

Al aplicar en la (3.20) los valores de R_m y L_m encontrados con anterioridad se obtiene el valor de τ_e .

$$\tau_e = 1,93328 \text{ ms}$$

Otro método para encontrar τ_e y L_m es evaluando la respuesta dinámica de primer orden del motor utilizado como generador según la Figura 3.2, tras aplicar escalones en V_m . El tiempo que haya transcurrido desde que se aplica el escalón hasta que el rotor incremente su velocidad angular hasta el 63,2% del cambio de velocidad esperado será el valor de τ_e .

Es necesario aplicar múltiples escalones unitarios en V_m a lo largo de todo el rango de operación del motor para luego promediar los valores de τ_e obtenidos.

En el Cuadro 3.4 se muestran las constantes de tiempo encontradas en la respuesta dinámica del motor tras aplicar escalones unitarios en la excitación V_m cambiando el punto de operación a lo largo de la zona lineal hasta llegar a 24 V. De los 23 datos de

Cuadro 3.4: Constante de Tiempo medida mediante caracterización dinámica

Muestra [#]	τ [ms]
De 1 a 2 V	2,8
De 2 a 3 V	2,5
De 3 a 4 V	2,1
De 4 a 5 V	2,1
De 5 a 6 V	2,1
De 6 a 7 V	2,0
De 7 a 8 V	1,9
De 8 a 9 V	1,8
De 9 a 10 V	1,9
De 10 a 11 V	1,9
De 11 a 12 V	1,8
De 12 a 13 V	1,8
De 13 a 14 V	1,9
De 14 a 15 V	1,9
De 15 a 16 V	1,9
De 16 a 17 V	1,7
De 17 a 18 V	1,8
De 18 a 19 V	1,7
De 19 a 20 V	1,9
De 20 a 21 V	1,8
De 21 a 22 V	1,9
De 22 a 23 V	1,7
De 23 a 24 V	1,8

τ_e mostrados en el Cuadro 3.4 se obtiene un promedio de:

$$\tau_e = 1,94348 \text{ ms}$$

Utilizando la Ecuación (3.20) se puede hallar la Inductancia del motor L_m utilizando el τ_e encontrado mediante la respuesta dinámica.

$$L_m = \tau_e R_m$$

$$L_m = 2,40311 \text{ mH}$$

Al promediar los valores de L_m y τ_m obtenidos mediante los dos métodos descritos, se obtienen los siguientes valores de Inductancia y Constante de Tiempo eléctrica del

Motor.

$$L_m = 2,39681 \text{ mH}$$

$$\tau_m = 1,93838 \text{ ms}$$

3.2.7. Momento de Inercia del Motor J_m

Cuando la alimentación de la armadura es desconectada, la velocidad angular del rotor ω_m empieza a reducirse hasta llegar a ser nula. Para poder modelar esta dinámica, es necesario encontrar el modelo mecánico del motor de DC.

Dada la Ecuación 3.4 y la relación $T_m = K_m I_m$, se obtiene el modelo mecánico del motor de DC que se muestra en la Ecuación 3.21.

$$T_m = J_m \frac{d\omega_m}{dt} + B_m \omega_m \quad (3.21)$$

Para encontrar la función de transferencia mecánica del motor, se utiliza la transformada de Laplace sobre la Ecuación (3.21).

$$\begin{aligned} \mathcal{L}\{T_m(t)\} &= J_m \frac{d\omega_m(t)}{dt} + B_m \omega_m(t) \\ T_m(s) &= (sJ_m + B_m) \omega_m(s) \end{aligned} \quad (3.22)$$

Al despejar $\omega_m(s)/T_m(s)$ de la Ecuación (3.22) se obtiene la función de transferencia mecánica del motor.

$$\frac{\omega_m(s)}{T_m(s)} = \frac{1}{sJ_m + B_m} = \frac{1/B_m}{s \frac{J_m}{B_m} + 1} \quad (3.23)$$

Aplicando anti-transformada de Laplace a la Ecuación (3.23) se obtiene que si el rotor del motor está girando a velocidad angular ω_m constante y es desconectado del voltaje de excitación V_m , su dinámica se guiará por la Ecuación (3.24).

$$\omega_m(t) = \frac{T_m}{B_m} e^{-\frac{t}{\tau_m}} \quad \because \tau_m = \frac{J_m}{B_m} \quad (3.24)$$

Si el tiempo t se hace igual a τ_m , la velocidad angular del rotor será igual a e^{-1} veces el valor inicial de $\omega_m = T_m/B_m$. Esto es el 36,788 % del valor de ω_m justo antes de desconectar la excitación V_m .

Fue necesario entonces medir el tiempo que le toma al motor disminuir su velocidad angular hasta el 36,788 % del valor inicial. En la Figura 3.11 se muestran los valores de τ_m encontrados para cada valor de V_m antes de desconectar el motor de la excitación.

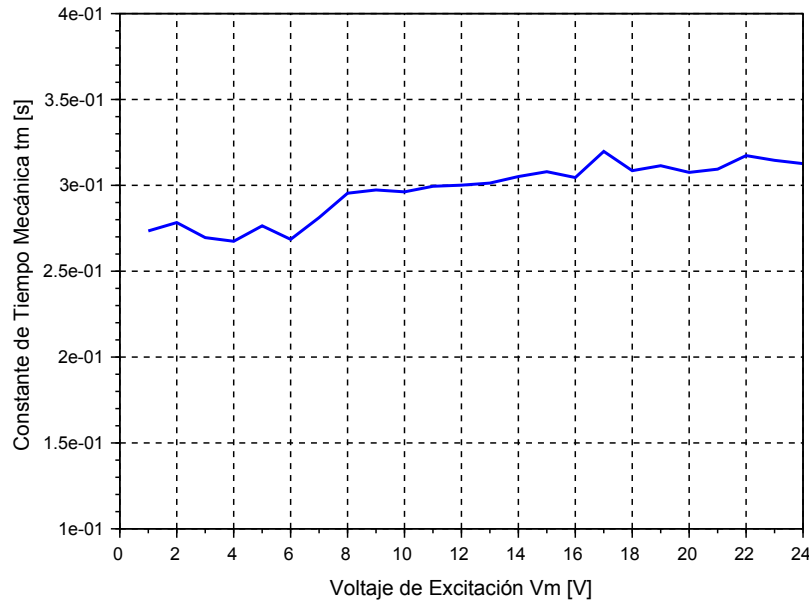


Figura 3.11: Constante de Tiempo Mecánica τ_m para cada Voltaje V_m

De los datos de τ_m mostrados en la Figura 3.11 se obtiene un promedio de:

$$\tau_m = 0,295525 \text{ s}$$

Dado que $\tau_m = \frac{J_m}{B_m}$, y conociendo los valores de τ_m y B_m el momento de inercia del motor sera:

$$J_m = 0,0010813969 \text{ kg} \cdot \text{m}^2$$

3.3. Modelado Mecánico de la Planta

Para el modelado mecánico de la planta PIS-UV se llevaron a cabo las mediciones correspondientes al peso y longitud de los elementos principales de la planta, además del coeficiente de fricción del carro en el riel. Para ello, se utilizó una báscula con una incertidumbre de $\pm 0,001 \text{ kg}$, un metro con una incertidumbre de $\pm 0,001 \text{ m}$ y un cronometro con una incertidumbre de $\pm 0,01 \text{ s}$.

Como primer paso para la caracterización mecánica de la planta, se tomaron las medidas correspondientes a longitudes de los diversos componentes de la planta tras lo cual se realizó el plano frontal en un software CAD. El plano frontal de la planta, con las principales medidas a tener en cuenta, se muestra en la Figura 3.12.

La longitud de la barra del péndulo es de 30 cm , el cilindro móvil en el péndulo tiene una longitud de $3,8 \text{ cm}$ y un radio de $1,9 \text{ cm}$, la longitud del riel es de 88 cm , el radio

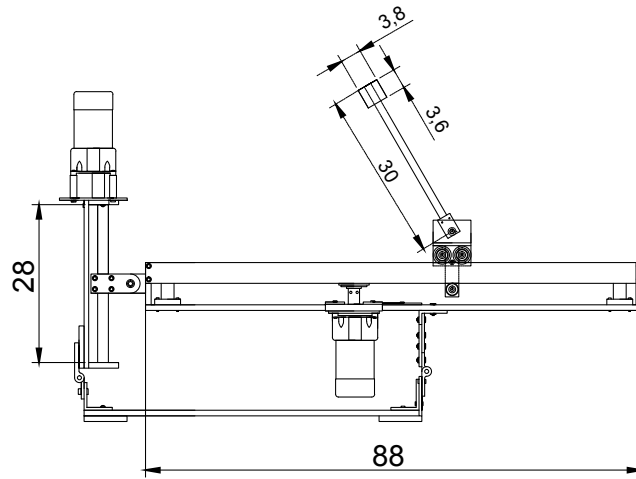


Figura 3.12: Plano frontal de la planta del Péndulo Invertido Simple - UV

de la polea que mueve al carro es de $0,049\text{ m}$, el radio externo del cilindro que forma el péndulo es de $0,635\text{ cm}$ y el radio interno del cilindro es de $0,58\text{ cm}$. Además, se encontró el valor de la masa del carro y del péndulo utilizando la báscula mencionada. Los valores obtenidos fueron de $1,275\text{ kg}$ y $0,080\text{ kg}$ respectivamente. Por su parte la masa del cilindro medida en la báscula fue de $0,287\text{ kg}$.

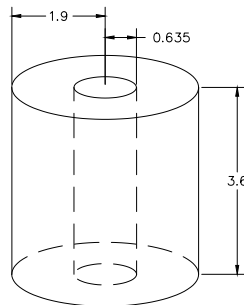


Figura 3.13: Plano del Cilindro Móvil que aporta masa al Péndulo

El momento de inercia del péndulo puede analizarse como el momento de inercia de dos cuerpos por separado.

El momento de inercia de la barra hueca respecto al punto de unión del péndulo al carro dado por la Ecuación (3.25).

$$I_b = M_b R_i^2 \quad (3.25)$$

Donde I_b es el momento de inercia de la barra, M_b es la masa de la barra, y R_i es el radio interno de la barra.

Por su parte el momento de inercia del cilindro móvil respecto al punto de unión del péndulo con el carro está dado por la Ecuación (3.26) [9].

$$I_c = \frac{1}{4}M_m r_c^2 + \frac{1}{3}M_m L_c^2 + M_m L^2 + M_m L_c L \quad (3.26)$$

Donde I_c es el momento de inercia del cilindro, M_m es la masa del cilindro, r_c es el radio del cilindro, L_c es la longitud del cilindro y L es la longitud que separa al cilindro del punto de unión del péndulo con el carro. Es importante que, dado que el cilindro móvil posee una zona hueca por donde se incrusta la barra del péndulo, la inercia de esa zona hueca debe ser restada de la inercia total. Por lo tanto la inercia del cilindro hueco queda representada por:

$$I_c = \frac{1}{4}M_m (r_c^2 - R_i^2) \quad (3.27)$$

Al remplazar los valores medidos en el cilindro y sumar las Ecuaciones (3.25) y (3.27), se obtiene el valor del momento de inercia del péndulo I_p .

$$I_p = 2,49 * 10^{-5} \text{ kg} * \text{m}^2 \quad (3.28)$$

La relación de transmisión del motor y la polea K_g corresponde a un valor de 1 debido a que la polea y el rotor del motor giran sobre el mismo eje y no presentan ningún tipo de piñonería adicional.

El coeficiente de fricción del carro sobre el riel, se encuentra mediante la Ley fundamental de la dinámica o segunda ley de Newton. El diagrama de cuerpo libre del carro sobre el riel se muestra en la Figura 3.14.

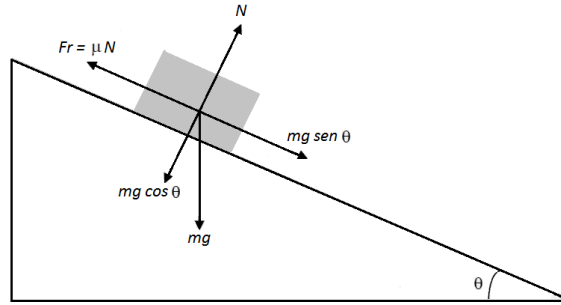


Figura 3.14: Diagrama de Cuerpo Libre del Carro sobre el Riel

Utilizando la segunda ley de Newton sobre el Diagrama de Cuerpo libre de la Figura 3.14 se obtiene la expresión para el coeficiente de fricción dinámica del carro μ_d .

$$\begin{aligned} \sum F_y &= N - mg \cos(\theta) \\ \sum F_x &= mg \sin(\theta) - \mu_d N = ma \end{aligned}$$

$$\mu_d = \tan(\theta) - \frac{a}{g \cos(\theta)} \quad (3.29)$$

A partir de las ecuaciones del movimiento rectilíneo acelerado uniforme, es posible encontrar la aceleración a de una masa que se mueve en línea recta, conociendo la posición inicial X_i , la posición final X_f , la velocidad inicial V_o y el tiempo t que le toma a la masa llegar de la posición inicial a la posición final.

$$X_f = \frac{1}{2}at^2 + V_o t + X_i \quad (3.30)$$

Para encontrar el coeficiente de fricción dinámico del carro, se utiliza una posición inicial $X_i = 0 \text{ m}$, una posición final $X_f = 0,5 \text{ m}$, una velocidad inicial $V_o = 0 \text{ m/s}^2$, y un ángulo de inclinación del riel $\theta = 14^\circ$. Utilizando la Ecuación (3.30) y teniendo en cuenta las consideraciones anteriores, se puede determinar que la aceleración del carro, será $a = \frac{1}{t^2}$.

En el Cuadro 3.5 se muestran los datos obtenidos durante el experimento para encontrar el coeficiente de fricción dinámico del carro sobre el riel.

Cuadro 3.5: Datos correspondientes al experimento para encontrar μ_d del carro

Tiempo [s]	$a [m/s^2]$	μ_d
1,11	0,812	0,166
1,14	0,769	0,170
1,03	0,943	0,152
1,1	0,826	0,164
1,02	0,961	0,150
1,11	0,812	0,166
1,06	0,890	0,157
1,09	0,842	0,163
1,11	0,812	0,166
1,13	0,783	0,169

El promedio de los valores de μ_d obtenidos en la tabla arroja un valor de:

$$\mu_d = 0,165$$

Los parámetros mecánicos de la planta se muestran en el Cuadro 3.6.

Cuadro 3.6: Parámetros Mecánicos de la Planta

Parámetro	Símbolo	Valor	Unidades
Masa del Carro	M	1,275	kg
Masa de la Barra del Péndulo	M_b	0,080	kg
Masa del Cilindro del Péndulo	M_m	0,287	kg
Masa Total del Péndulo	m	0,367	kg
Momento de Inercia del Péndulo	I_p	$2,49 * 10^{-5}$	$kg * m^2$
Relación de Transmisión Motor-Polea	K_g	1	—
Radio de la Polea	R_p	0,049	m
Longitud de la barra del Péndulo	l	0,30	m
Longitud del Cilindro del Péndulo	L_c	0,036	m
Radio Externo del Cilindro del Péndulo	R_e	0,00635	m
Radio Interno del Cilindro del Péndulo	R_i	0,0058	m
Longitud del Riel	L_{riel}	0,88	m
Coeficiente de Fricción Dinámico del Carro	μ_d	0,165	—

3.4. Validación del Modelo del Motor

La función de transferencia del motor, tal y como se encontró en la Ecuación 3.11 se define como:

$$\frac{\omega_m(s)}{V_m(s)} = \frac{1/K_e}{\frac{R_m J_m}{K_e K_m} s + 1} \quad (3.31)$$

En el Cuadro 3.7 se muestran los valores encontrados para cada uno de los parámetros del motor en estudio.

Contando ya con los valores de los parámetros necesarios, es posible definir la función de transferencia propia del motor en estudio.

Al remplazar los valores de los parámetros en la Ecuación 3.11 se obtiene:

$$\frac{\omega_m(s)}{V_m(s)} = \frac{1,2258507}{0,002009345s + 1} \quad (3.32)$$

Utilizando como herramienta de Simulación el paquete de software libre de código abierto Scilab se procedió a simular la respuesta del modelo del motor, presentado en la Ecuación (3.32), ante un escalón de 5 V y comparar esta respuesta frente a los datos obtenidos.

En la Figura 3.15 se puede observa que la respuesta simulada del modelo del motor ante un escalón de 5 V en su alimentación corresponde adecuadamente a la respuesta

Cuadro 3.7: Parámetros del Motor

Parámetro	Símbolo	Valor	Unidades
Resistencia de la Armadura	R_m	1,2365	Ω
Constante Eléctrica del Motor	K_e	0,81576	$\frac{V}{Rad/s}$
Constante de Torque del Motor	K_m	0,81576	$N * \frac{m}{A}$
Coefficiente de Fricción del Motor	B_m	0,00365924	$N * m * s$
Inductancia del Motor	L_m	2,39681	mH
Constante de Tiempo Eléctrica del Motor	τ_e	1,93838	ms
Momento de Inercia del Motor	J_m	0,0010813969	$kg * m^2$

real del motor ante el mismo escalón en V_m . Debido a esto es posible afirmar que el modelo encontrado describe efectivamente la dinámica presente en el motor de imán permanente que sirve como actuador en la planta PIS-UV.

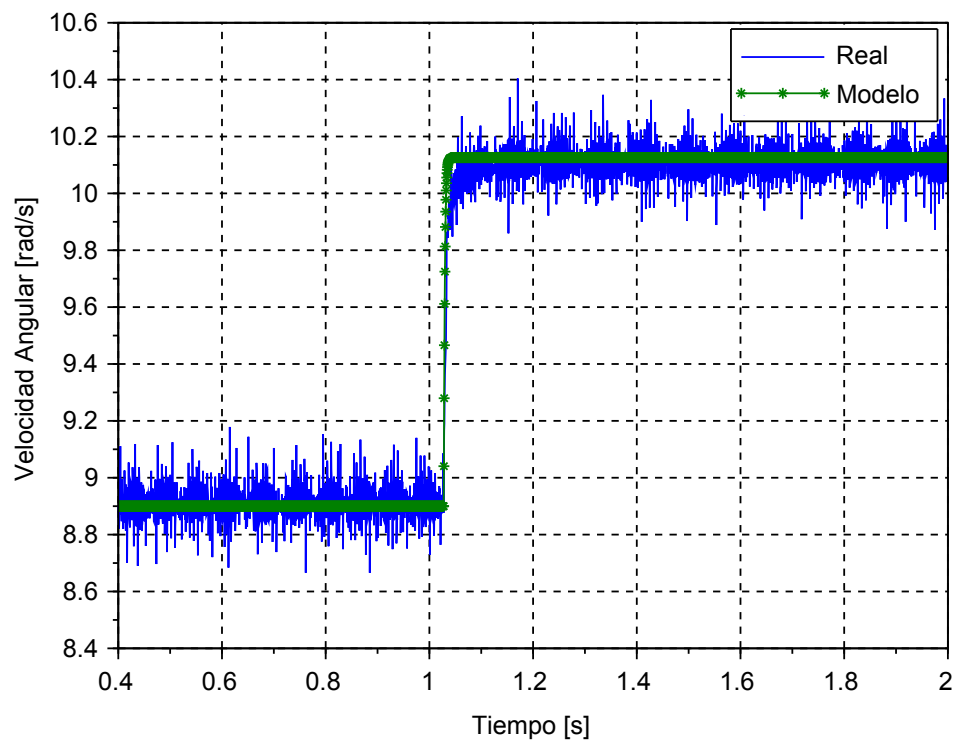


Figura 3.15: Comparación respuesta dinámica del motor real y modelo

Capítulo 4

Integración Software-Hardware

En este capítulo se describe el modo en que se realiza la integración entre los elementos de Hardware de la planta, tales como sensores y actuadores, y el software donde se ejecutará el controlador de la planta. Además se muestra el procedimiento de acondicionamiento de los sensores y actuadores para facultar su uso a través de la tarjeta de adquisición de datos (DAQ) utilizada.

En la Sección 4.1 se presentan los sensores utilizados para la medición de la posición angular del péndulo y la posición lineal del carro, además de los circuitos de acondicionamiento de señales para ser leídas mediante la tarjeta de adquisición de datos empleada.

La descripción de los circuitos de alimentación utilizados para proveer de potencia a los diversos circuitos auxiliares, de sensado y actuadores, se describe en la Sección 4.2.

En la Sección 4.3 se describen los circuitos que permiten transferir una señal de control desde la tarjeta de adquisición de datos hacia los actuadores.

La Sección 4.4 muestra las características de la tarjeta NI DAQ USB-6002 y el procedimiento para lograr que esta pueda interactuar con el software de Scilab. Además identifica el principal problema que esta interacción presenta para la integración dentro de un sistema de control.

Como sustituto de la tarjeta NI DAQ USB-6002 se utiliza un Arduino Mega 2560. Sus características y requerimientos para ser utilizado en el control del PIS-UV se muestran en la Sección 4.5.

En la Sección 4.6 se presentan las pruebas del adecuado funcionamiento de los diferentes elementos de Hardware y Software que se utilizarán en el sistema de control del PIS-UV además de exponerse los cambios y adecuaciones necesarias para que cada elemento de la planta funcione de la manera correcta.

4.1. Sensórica

4.1.1. Sensor de Posición Angular del Péndulo

Típicamente para la medición de la posición angular de una planta de péndulo invertido se utiliza un encoder óptico rotatorio que permite conocer cuanto ha girado el péndulo respecto a su posición inicial y dependiendo de la resolución del encoder [10]. Sin embargo la utilización de este elemento sensor conlleva a la alteración mecánica de la planta, además de incrementar los costos de la misma ya que los encoders son elementos de alto costo si se desea alta resolución en la medida. Un encoder de 1000 pulsos / revolución puede tener un costo hasta de 70 USD incluyendo los elementos de acondicionamiento y procesamiento de la señal.

Dado esto, se optó por utilizar el módulo GY-521 que utiliza el circuito integrado MPU6050 como elemento sensor. El MPU6050 contiene un acelerómetro de 3 ejes y un giróscopo de 3 ejes basados en sistemas microelectromecánicos (MEMS). El giróscopo permite realizar la medición de la aceleración angular de un cuerpo respecto a un eje, mientras que el acelerómetro permite medir la aceleración de un cuerpo a lo largo de una dirección [11]. El módulo GY-521, mostrado en la Figura 4.1, dispone del protocolo de comunicación estándar I^2C por lo que es fácil obtener su información desde un sistema microprocesado.



Figura 4.1: Modulo GY521 basado en MPU6050

Para obtener datos correctos de la posición angular de un objeto a partir de un acelerómetro y un giróscopo es necesaria la implementación de un algoritmo de fusión sensorial como un Filtro Kalman. Dada la pesada carga computacional que conlleva consigo un filtro Kalman y la necesidad de convertir la información de la posición angular del péndulo en un voltaje legible desde una tarjeta DAQ, se utiliza un Arduino Nano como elemento de procesamiento de la señal del ángulo del péndulo.

El diagrama de bloques del procesamiento de la señal de posición angular del péndulo se muestra en la Figura 4.2

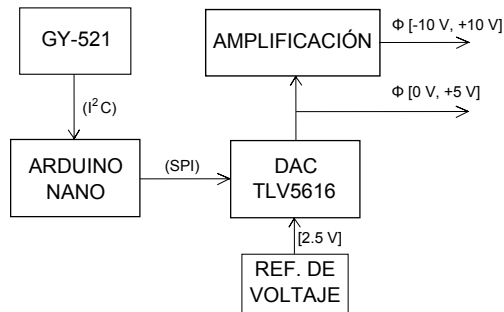


Figura 4.2: Diagrama de Bloques del Procesamiento de la Señal Ángulo

El algoritmo de Filtro Kalman implementado en el Arduino Nano fue desarrollado por Kristian Lauszus de TKJ Electronics en 2012 y liberado bajo Licencia GNU [12].

Una vez obtenido el valor de la posición angular del péndulo, es necesario convertir estos datos en una señal de voltaje legible desde una tarjeta DAQ. Para ello se utiliza un conversor digital-análogo (DAC) TLV5616 de 12 bits. Este DAC recibe 12 bits de datos por comunicación SPI, tiene un tiempo de establecimiento de entre $3 \mu s$ y $9 \mu s$ y permite salidas de voltaje de entre 0 y 5,5 V dependiendo del voltaje de referencia utilizado [13]. Para este caso, se utiliza una referencia de voltaje a 2,5 V de manera que el máximo voltaje de salida en el DAC sea de 5 V.

Por último, se añade una etapa de amplificación para que puedan utilizarse tarjetas DAQ cuyos voltajes de entrada permitidos se encuentran en el rango de 0 a 5 V así como tarjetas DAQ con rangos de entrada de -10 a 10 V.

4.1.2. Sensor de Posición Lineal del Carro

Para sensar la posición del carro sobre el riel, el motor que funciona como actuador cuenta con un encoder incremental. Sin embargo tras realizar pruebas de funcionamiento al encoder se evidencio que no emite señal alguna, por lo que se concluyó que se encuentra defectuoso. Debido a esto se decidió utilizar un sensor infrarrojo de distancia como el SICK DT20-P130B1080 mostrado en la Figura 4.3.

El SICK DT20-P130B1080 es un sensor de distancia por medio de luz infrarroja que tiene un rango de medida de entre 100 mm y 800 mm. Este sensor tiene una resolución de entre 1 mm y 2 mm, rata de actualización en la salida de 1 ms y una salida análoga de entre 4 mA y 20 mA [14].



Figura 4.3: Sensor infrarrojo de distancia SICK DT20-P130B1080

Para que la señal proveniente del sensor de distancia sea legible por una tarjeta DAQ es necesario convertir la señal de corriente en una señal de voltaje. Para ello se utiliza un Receptor de Bucle de Corriente RCV420. El circuito integrado RCV420 es un convertor de corriente en un rango de 4 - 20 mA a voltaje en un rango de 0 - 5 V . Posee una referencia de voltaje de alta precisión de 10 V y una precisión de conversión del 0,1 %.

Al igual que en el caso del sensor de posición angular, se añade una etapa de amplificación para permitir la utilización de tarjetas DAQ de entradas de entre 0 y 5 V y tarjetas DAQ con entradas de entre -10 y +10 V .

El diagrama de bloques del procesamiento de la señal de posición lineal del carro sobre el riel se muestra en la Figura 4.4

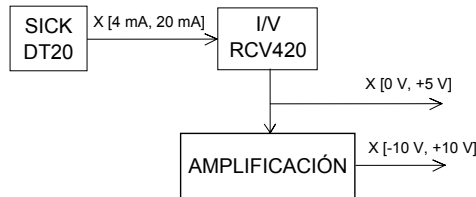


Figura 4.4: Diagrama de Bloques del Procesamiento de la Señal Posición

4.2. Alimentación de los Módulos

4.2.1. Alimentación de los Motores

Conociendo la caracterización dinámica mostrada en la Figura 3.5 se evidencia que se requiere alimentar el motor con un máximo de 24 V . Además dada la resistencia de armadura del motor $R_m = 1,2365 \Omega$ y la ecuación 3.5, es posible determinar la máxima corriente que consumirá el motor; esto sucede en caso de aplicar el voltaje máximo al motor (24 V) mientras el rotor posee una velocidad angular $\omega = 0 \text{ rad/s}$. La máxima corriente que consume el motor de la planta PIS-UV es de 19,41 A .

Los requerimientos máximos de alimentación del motor son $24\text{ V} / 466,83\text{ W}$ por lo tanto se optó por alimentar el motor utilizando una fuente de $24\text{ V} / 500\text{ W}$ como la que se muestra en la Figura 4.5.



Figura 4.5: Fuente de $24\text{ V} / 500\text{ W}$ para alimentación de motores

Esta fuente dispone de 3 salidas de entre $23,5$ y $24,5\text{ V}$ ajustables mediante perilla, extractor de calor siempre encendido y sistemas de protección de sobre-voltaje y corto-circuito.

4.2.2. Alimentación de los Circuitos de Sensórica y Auxiliares

Para la alimentación de los circuitos sensores, acondicionadores de señal y controladores de los motores se tiene en cuenta los diversos requerimientos de cada circuito.

El módulo GY-521 requiere de una alimentación de 5 V mientras que el sensor SICK DT20-P130B1080 puede alimentarse con entre 10 y 30 V . Los circuitos de acondicionamiento de la señal de posición angular del péndulo requieren una alimentación de 7 V y el conversor de corriente a voltaje utilizado en el procesamiento de la señal de posición lineal del carro requiere una alimentación de 12 V . Por otro lado, los circuitos de amplificación requieren alimentación dual de $\pm 12\text{ V}$ elevando los requerimientos de salida de la fuente alimentadora de estos circuitos a salidas de $+5$, $+7$, $+12$ y -12 V .

Para alimentar estos circuitos entonces, se utiliza una fuente de voltaje DC que cuenta con las 4 salidas especificadas y con una potencia máxima de salida de 50 W .

4.3. Controladores de los Motores

Para controlar la velocidad y sentido de giro del motor que sirve como actuador del PIS-UV es necesario utilizar un puente H controlado por PWM que soporte los requerimientos de voltaje y corriente del motor.

El controlador de motor BTS7960 43A, mostrado en la Figura 4.6, permite un voltaje máximo de alimentación de 24 V y una corriente continua de máximo 43 A además de permitir el control de velocidad y sentido de giro mediante dos señales PWM de hasta 25 kHz (uno por sentido de giro) y dos señales habilitadoras (uno por sentido de giro) por lo que es apto para controlar el motor actuador del PIS-UV [16].



Figura 4.6: Modulo controlador de motor BTS7960 43A

4.4. Tarjeta de Adquisición de Datos NI DAQ USB-6002

4.4.1. Descripción Tarjeta NI DAQ USB-6002

La tarjeta de adquisición de datos NI DAQ USB-6002 es un dispositivo DAQ de bajo costo adecuada para registros de datos simple, medidas portátiles y experimentos académicos de laboratorio, tal como la planta PIS-UV.

Esta tarjeta de adquisición de datos ofrece ocho canales de entrada analógica con un conversor análogo-digital (ADC) de 16 bits de resolución y frecuencia de muestreo de hasta 50 kHz . Además cuenta con 13 líneas de I/O digitales, un contador y dos salidas análogas con un conversor digital-análogo (DAC) de 16 bits, una rata de actualización de 5 kHz y una impedancia de salida de $0,2\ \Omega$. Tanto las entradas como las salidas análogas cuentan con un rango de voltaje de $\pm 10\text{ V}$ [17].

La NI DAQ USB-6002 se alimenta directamente vía USB desde el computador al cual se haya conectado.



Figura 4.7: Tarjeta de Adquisición de Datos NI DAQ USB-6002

4.4.2. Software Necesario para Interacción Scilab-DAQ

Para la utilización de cualquier tarjeta DAQ de National Instruments se requiere la previa instalación del controlador NI-DAQmx. Los archivos de instalación de este controlador se encuentran en el disco enviado junto a la tarjeta DAQ al momento de adquirirla y solo son validos para sistemas operativos Windows[®]. También es posible descargar este controlador desde internet en la pagina <http://www.ni.com/dataacquisition/nidaqmx>.

La tarjeta NI DAQ USB-6002 es aceptada solo por versiones del controlador NI-DAQmx superiores o iguales a la versión 9.9. Además de la instalación del controlador NI-DAQmx es necesaria la instalación de un módulo o toolbox para Scilab llamado NI-DAQ. Este módulo es una interfaz entre NI-DAQmx y Scilab. Para instalar este módulo es suficiente ejecutar en la consola de Scilab el comando `atomsInstall('NIDAQ')`. Para utilizar el módulo NIDAQ es necesario ejecutar Scilab teniendo en cuenta que la tarjeta NI DAQ-USB6002 debe haber sido conectada vía USB previamente y adicionalmente ejecutar el comando `atomsLoad('NIDAQ')`.

4.4.3. Problemas encontrados en la interacción Scilab-DAQ

El toolbox NIDAQ de Scilab permite leer entradas análogas y análogas y configurar valores a salidas análogas y digitales, sin embargo se encontró que solo es posible realizar una de estas acciones a la vez. Dado esto, cada vez que se desee realizar una acción diferente a la ejecutada anteriormente será necesario reconfigurar las tareas de la tarjeta y así cambiar de acción.

El proceso de reconfiguración de las tareas de la tarjeta DAQ mediante el toolbox NIDAQ fue medido utilizando la función `tictoc()` de Scilab obteniendo un valor promedio de 80 ms por reconfiguración. Para realizar el control del PIS-UV es necesario sensor dos entradas análogas y configurar el valor de una salida análoga lo cual hace que el tiempo de muestreo mínimo para esta necesidad especifica sea de 240 ms aproximadamente. Debido a esta restricción se descartó la posibilidad de utilizar la tarjeta NI DAQ-USB6002 mediante el toolbox NIDAQ de Scilab.

4.5. Arduino Mega 2560 como Tarjeta de Adquisición de Datos

4.5.1. Descripción Arduino Mega 2560

La tarjeta de desarrollo Arduino Mega 2560 es una plataforma de hardware libre basada en un microcontrolador ATmega 2560 que dispone de 54 entradas/salidas digitales de las cuales 15 pueden ser usadas como salidas PWM, 16 entradas análogas de entre 0 y 5 V, y alimentación via USB o externa [18].

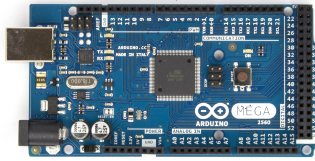


Figura 4.8: Arduino Mega 2560

4.5.2. Software Necesario para Interacción Scilab-Arduino/XCOS

Para utilizar cualquier tarjeta Arduino como tarjeta DAQ en XCOS (el entorno de diagramas de bloques de Scilab) es necesaria la instalación de un módulo para Scilab llamado Arduino. Este módulo crea diversos bloques en la paleta de XCOS para poder interactuar con el Arduino conectado además de permitir un tiempo de muestreo de mínimo 5 ms. Para instalar este módulo es suficiente ejecutar en la consola de Scilab el comando `atomsInstall('arduino')`.

Además, es necesario cargar en el Arduino Mega 2560 que se utilizará como tarjeta DAQ, un programa llamado `toolbox_arduino_V3.ino` que se descarga junto al módulo Arduino de Scilab.

El módulo de Arduino para XCOS posee bloques para controlar motores pero ninguno de las características del controlador BTS7960 que se utiliza en el PIS-UV. Por esta razón se modificó el programa `toolbox_arduino_V3.ino` para que en lugar de configurar una salida PWM mediante el bloque de `AnalogWrite` se controlara la velocidad y sentido de giro del motor mediante las dos señales PWM y las dos señales habilitadoras necesarias. Los cambios realizados al programa `toolbox_arduino_V3.ino` se muestran en el Apéndice A.1. La nueva forma de funcionamiento del bloque `AnalogWrite` se describe como muestra la Figura 4.9.

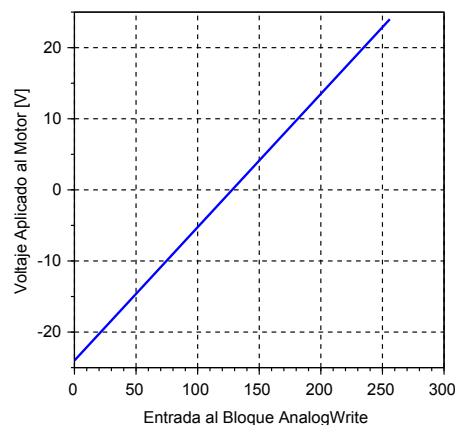


Figura 4.9: Caracterización Bloque AnalogWrite

Debido a la caracterización del bloque AnalogWrite mostrada en la Figura 4.9, es necesario convertir el voltaje deseado en la entrada del motor en un número entre 0 y 255. Para ello se utiliza el subsistema de XCOS mostrado en la Figura 4.10.

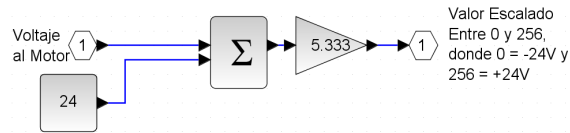


Figura 4.10: Subsistema conversor de voltaje deseado en numero de 8 bits

4.6. Pruebas de Funcionamiento de Integración Hardware-Software

4.6.1. Pruebas de los Sensores en XCOS

Para lograr obtener datos correctos y fiables a partir de los sensores y el acondicionamiento descritos anteriormente, es necesario realizar la caracterización de los sensores.

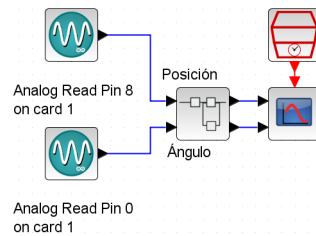


Figura 4.11: Diagrama para lectura de sensores en XCOS

Utilizando el diagrama mostrado en la Figura 4.11 se adquieren la señales provenientes del acondicionamiento de la señal de los sensores a través del Arduino Mega 2560 utilizado como tarjeta DAQ. En el Subsistema presente en el diagrama, se convierte la representación de 10 bits proveniente de los bloques AnalogRead en los datos reales de posición del carro (expresada en metros) y de ángulo del péndulo (expresado en radianes). Los valores de los bloques de conversión de los datos presentes en el Subsistema se obtienen de manera heurística y son mostrados en la Figura 4.12.

Una vez hecho posible la adquisición de los datos reales de posición y ángulo del péndulo se realizan pruebas para verificar que los datos adquiridos correspondan completamente al comportamiento real del PIS-UV. Para ello, se mueve manualmente y con el motor desconectado el péndulo de manera que vaya desde una posición angular de $-0,5 \text{ rad}$ hasta $+0,5 \text{ rad}$ y se varía la posición del carro desde $-0,1 \text{ m}$ hasta $+0,1 \text{ m}$.

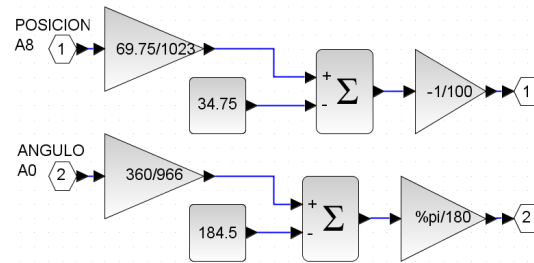


Figura 4.12: Bloques de conversión de señales de sensores en datos reales de salidas

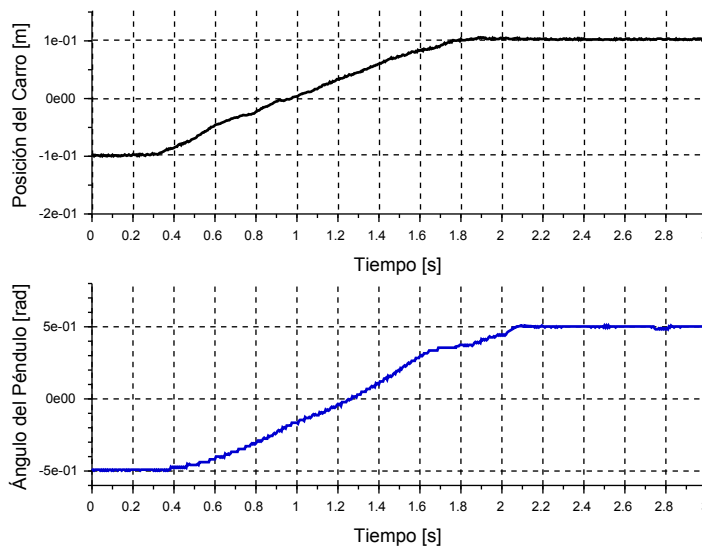


Figura 4.13: Prueba de los elementos de sensado en la Planta PIS-UV

Los datos sensados y mostrados en la Figura 4.13 corresponden a los valores reales de las variables de la planta PIS-UV por lo que se determina que los elementos de sensado funcionan adecuadamente y han sido caracterizados de la manera correcta.

4.6.2. Pruebas de Control del Motor desde XCOS

Como primera medida para probar el adecuado funcionamiento del motor controlado mediante el Puente H BTS7960 y un Arduino Mega 2560 como tarjeta DAQ, además de contar con la carga que representa el carro y el péndulo, es necesario identificar la zona muerta del actuador dentro del sistema. Para ello se aplica una rampa unitaria tanto con pendiente positiva como negativa a partir del primer segundo de ejecución.

En la Figura 4.14 es posible notar que las respuestas del motor ante excitación rampa

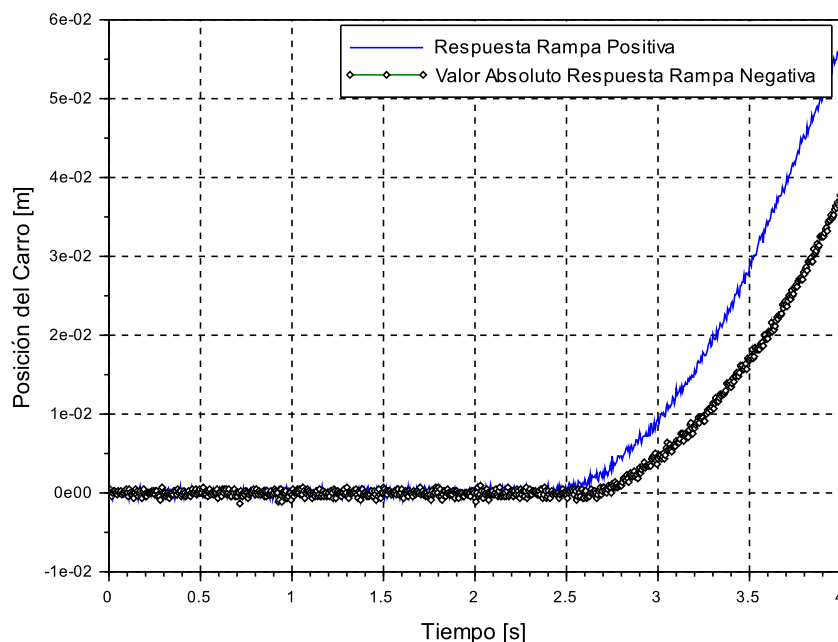


Figura 4.14: Respuesta del Carro ante entrada Rampa Unitaria Positiva y Negativa en el Motor

unitaria positiva y negativa son diferentes entre sí, por lo tanto es necesario compensar la respuesta a la rampa negativa de manera que esta se asemeje a la respuesta ante una rampa unitaria positiva. Para ello se modifica nuevamente el archivo `toolbox_arduino_V3.ino` de manera que cuando el motor se excite con un voltaje negativo la entrada sea amplificada por una ganancia tal que compense la respuesta del motor ante esta entrada. Las modificaciones al archivo `toolbox_arduino_V3.ino` se muestran en el Anexo A.1.

Para calcular la ganancia adecuada, se observa el voltaje de rompimiento de ambas respuestas. Conociendo que lo aplicado es una rampa unitaria que inicia tras el primer segundo de ejecución, es posible observar en la Figura 4.14 que el voltaje de rompimiento de la respuesta positiva es igual a 1,42 V mientras que para la respuesta negativa el voltaje de rompimiento es de 1,53 V. Con esto, es posible determinar que la ganancia que se debe aplicar a la señal de control cuando esta es negativa debe ser igual a $\frac{1,53}{1,42} = 1,08$.

Una vez modificado el archivo `toolbox_arduino_V3.ino` se analiza nuevamente la respuesta de la posición del carro ante una excitación tipo rampa unitaria positiva y negativa en el motor.

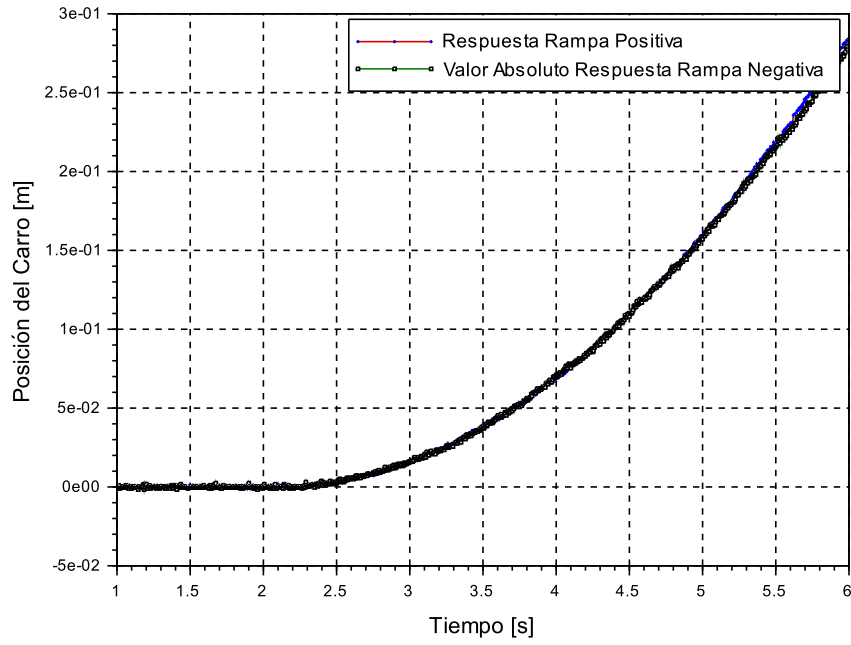


Figura 4.15: Respuesta del Carro ante entrada Rampa Unitaria Positiva y Negativa tras compensación

La Figura 4.15 muestra que tras aplicar la ganancia de 1,08 a la excitación rampa unitaria negativa del motor, el valor absoluto de la respuesta a esta excitación coincide plenamente con la respuesta ante una excitación rampa unitaria positiva.

Para compensar los efectos de la zona muerta del motor en el sistema se utiliza una señal de control con compensación de zona muerta como se muestra en la Figura 4.16 [19].

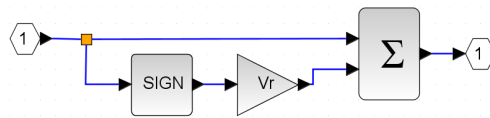


Figura 4.16: Subsistema de Compensación de Zona Muerta del Motor

Para poder implementar el subsistema de compensación de Zona Muerta del Motor y tras obtener una respuesta simétrica en el motor se procede a buscar el voltaje de rompimiento V_r del motor, este es el voltaje umbral en el que la operación del motor pasa

de la zona muerta a la zona lineal.

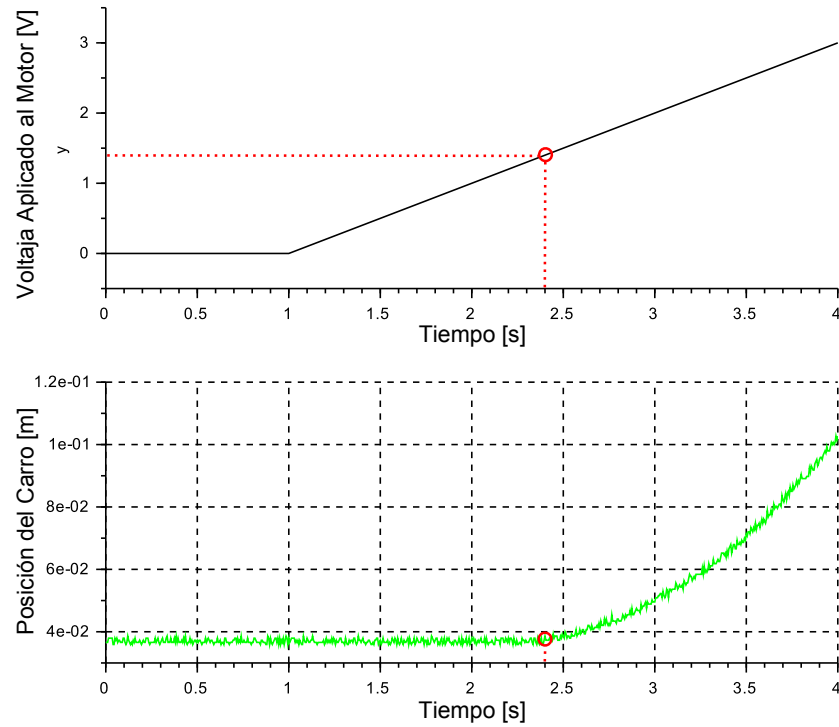


Figura 4.17: Identificación zona muerta motor con entrada rampa unitaria

La Figura 4.17 muestra las gráficas de voltaje aplicado al motor vs. tiempo y posición del carro vs. tiempo. En estas gráficas es posible observar que el motor solo empieza a girar una vez el voltaje aplicado supera el voltaje de rompimiento $V_r = 1,42 \text{ V}$.

Una vez aplicado este subsistema de compensación de zona muerta, la respuesta de la posición del carro ante una entrada rampa unitaria en el motor es medida nuevamente y se muestra en la Figura 4.18. La respuesta inmediata del motor es evidenciada en la Figura 4.18, comprobando el buen funcionamiento del subsistema de compensación de zona muerta del motor.

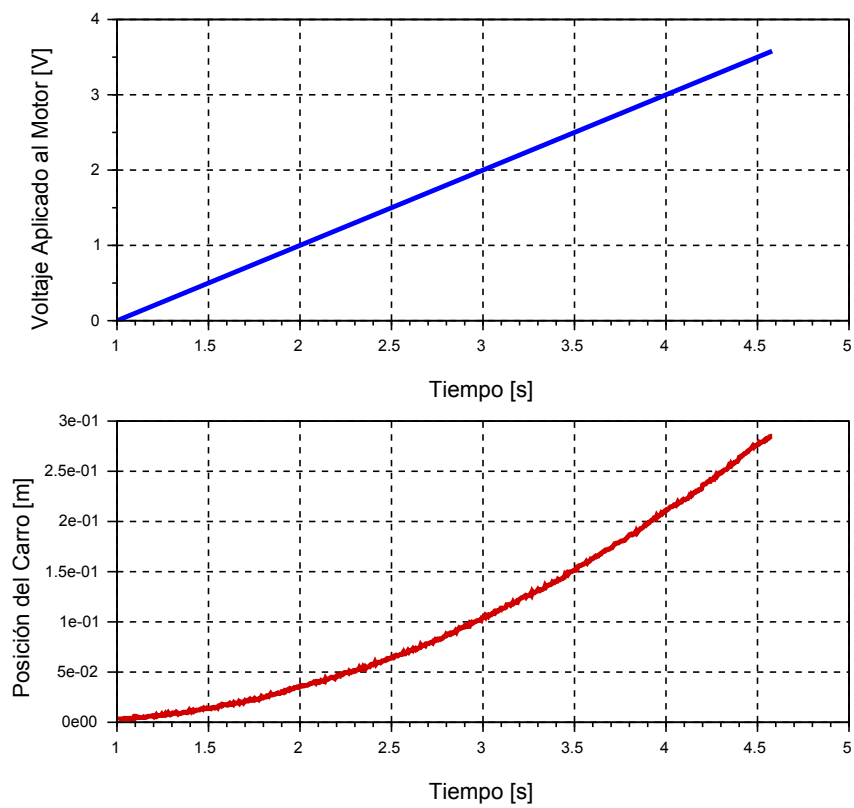


Figura 4.18: Respuesta de la Posición del Carro tras Compensación de Zona Muerta en el Motor

Capítulo 5

Diseño y Simulación de Estrategia de Control

En este capítulo se realiza el diseño de la estrategia de control para la planta del PIS-UV para sostener el péndulo en su posición de equilibrio inestable ($\phi = 0$). Se muestra además los diseños adicionales requeridos para lograr implementar la estrategia de control en la planta y las simulaciones realizadas para comprobar el adecuado funcionamiento del sistema de control diseñado.

En la Sección 5.1 se realiza la validación del modelo en representación de espacio de estados encontrado para la planta de péndulo invertido sobre carro utilizando los parámetros encontrados anteriormente y datos experimentales obtenidos de la planta real. La Sección 5.2 muestra la manera en que se diseñó el controlador que se aplicará en la planta PIS-UV teniendo en cuenta las características de inestabilidad de la planta y la posibilidad de realimentar cada uno de los elementos del vector de estados. El diseño del estimador de estados se realiza en la Sección 5.3 donde se muestran dos maneras diferentes en las que es posible obtener una estimación de los cuatro estados necesaria para la realimentación de los mismos. En la Sección 5.4 se muestran las diversas simulaciones realizadas para comprobar el adecuado funcionamiento del controlador en conjunto a los dos tipos de estimadores de estados diseñados.

5.1. Validación del Modelo del PIS-UV

Teniendo en cuenta el modelo en representación de espacio de estados encontrado en la Sección 2.2 y una vez encontradas todas las variables de la planta PIS-UV, como se muestra en el Capítulo 3, es posible encontrar el modelo de la planta. Dado que $\vec{x}$ es el vector de estados, $\vec{y}$ es el vector de salidas y $\vec{u}$ es el vector de entradas, la representación

en espacio de estados continuo del PIS-UV es:

$$\vec{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -175,88 & 2,82 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -665,58 & 47,78 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 10,56 \\ 0 \\ 39,95 \end{bmatrix} \vec{u}$$

$$\vec{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \vec{u}$$

Los valores propios de la matriz A obtenidos mediante el comando 'spec(A)' corresponden a:

$$\begin{bmatrix} \vec{\lambda}_A \end{bmatrix} = \begin{bmatrix} 0 & -175,95 & -6,06 & 6,12 \end{bmatrix}$$

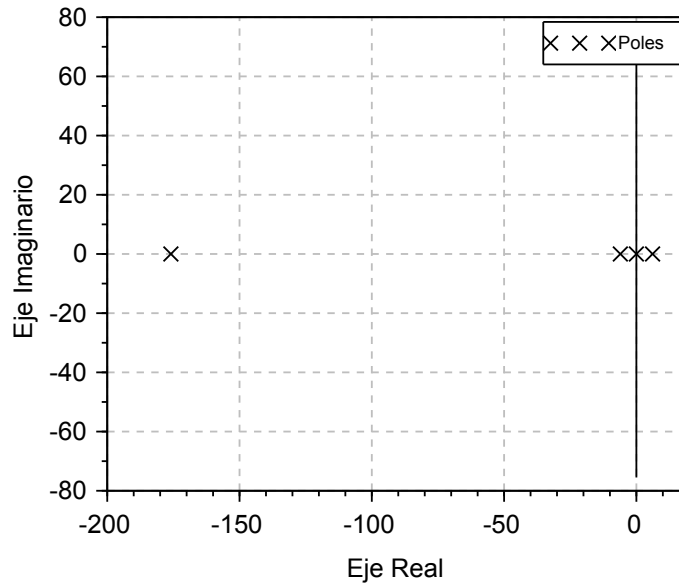


Figura 5.1: Diagrama de Polos y Ceros de Sistema Continuo PIS-UV

Dado que la Matriz A posee un valor propio con parte real positiva, se define el sistema representado por esta matriz como Inestable.

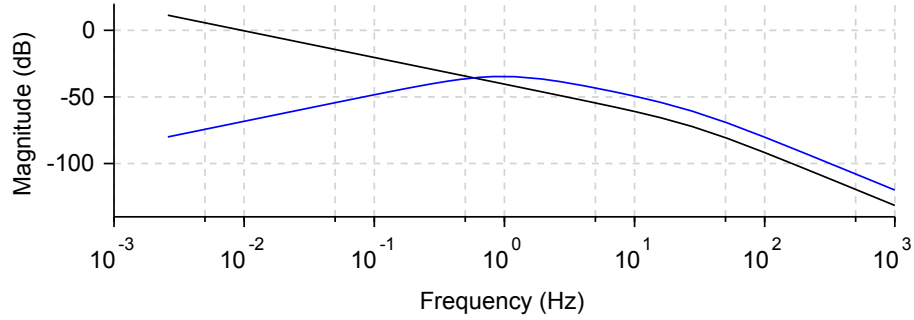


Figura 5.2: Diagrama de Polos y Ceros de Sistema Continuo PIS-UV

Para establecer el tiempo de muestreo adecuado para el sistema, se identifica la frecuencia de corte más alta en la respuesta en frecuencia de la planta. En la figura 5.2 se evidencia que la frecuencia de corte más alta en el sistema corresponde a $2,3 \text{ Hz}$. Según el teorema de muestreo de Nyquist-Shannon es necesario que la frecuencia de muestreo sea mayor a dos veces la máxima frecuencia presente en la señal a muestrear para poder replicar con exactitud la forma de la señal, por lo tanto se establece que la frecuencia de muestreo del sistema debe ser mayor a $4,6 \text{ Hz}$, lo que conlleva a un tiempo de muestreo menor a 216 ms . Conociendo que el módulo de Arduino para XCOS permite un tiempo de muestreo mínimo de 5 ms , se define este mínimo como el tiempo de muestreo que se utilizará en el control del PIS-UV.

Utilizando la función 'dscr' de Scilab se discretiza la representación en espacio de estados continuo de la planta con un tiempo de muestreo de $0,005 \text{ s}$ obteniendo la representación en espacio de estados discreto del PIS-UV que se muestra a continuación.

$$\overrightarrow{x(k+1)} = \begin{bmatrix} 1 & 3,32e-3 & 2,68e-5 & 4,78e-8 \\ 0 & 4,15e-1 & 9,38e-3 & 2,68e-5 \\ 0 & -6,34e-3 & 1,001 & 5e-3 \\ 0 & -2,21 & 2,21e-1 & 1,001 \end{bmatrix} \overrightarrow{x(k)} + \begin{bmatrix} 1e-4 \\ 3,51e-2 \\ 3,80e-4 \\ 1,33e-1 \end{bmatrix} \overrightarrow{u(k)}$$

$$\overrightarrow{y(k)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \overrightarrow{x(k)} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \overrightarrow{u(k)}$$

Para la representación discreta de la planta PIS-UV se hace el análisis de estabilidad para comprobar que el sistema sea inestable también en su modelo discreto. Los valores

propios de la matriz G obtenidos mediante el comando 'spec(G)' corresponden a:

$$\begin{bmatrix} \vec{\lambda}_A \end{bmatrix} = \begin{bmatrix} 1 & 0,42 & 1,03 & 0,97 \end{bmatrix}$$

Un sistema discreto es inestable si la magnitud de uno de los polos es mayor a 1. Dado que uno de los polos del sistema discretizado tiene como magnitud 1,03, se confirma que tras la discretización, el sistema sigue siendo inestable.

Una vez definido el modelo del PIS-UV es necesario realizar la validación de los valores teóricos respecto a datos reales en la planta, mediante el análisis de las respuestas ante un escalón unitario en la señal de entrada.

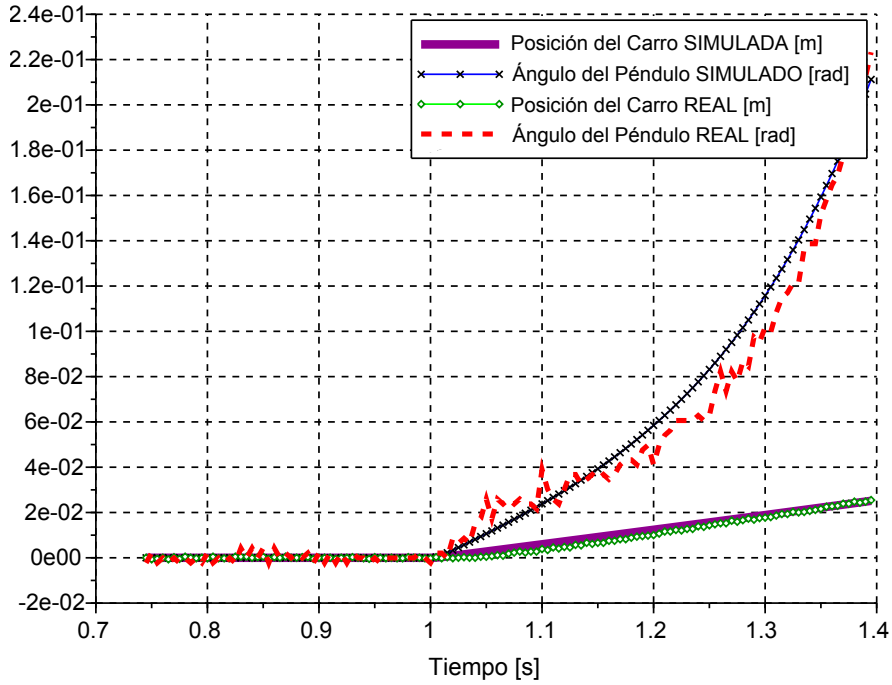


Figura 5.3: Validación del modelo ante una entrada escalón unitario positivo

Posteriormente se hace la validación del modelo encontrado ante una entrada negativa para corroborar que el modelo en representación de espacio de estados representa fielmente al PIS-UV.

Las Figuras 5.3 y 5.4 evidencian que el modelo encontrado, en representación de espacio de estados, para la planta del PIS-UV representa adecuadamente la dinámica real de la planta ante una entrada positiva y negativa.

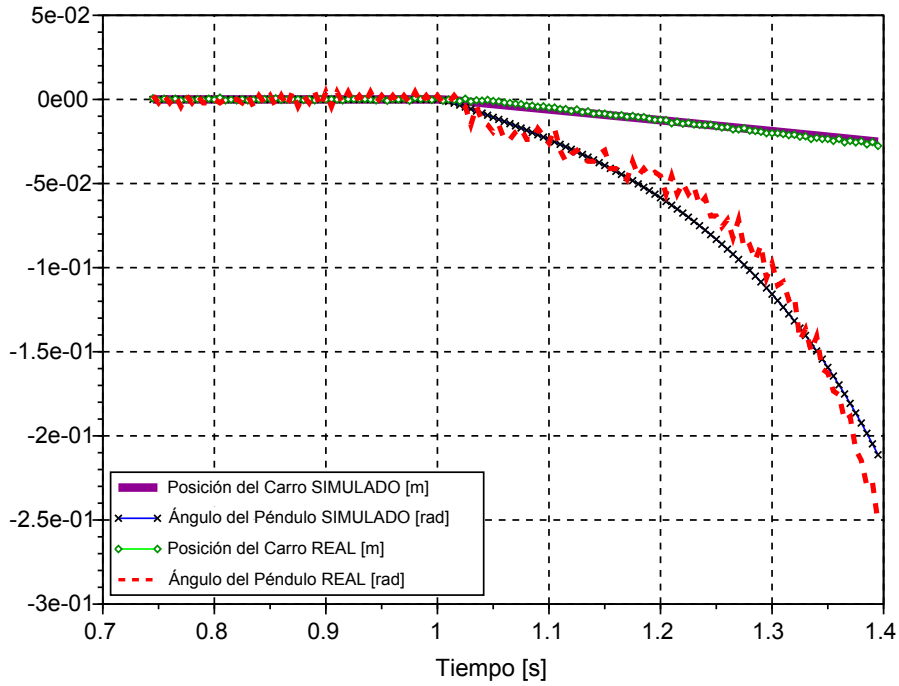


Figura 5.4: Validación del modelo ante una entrada escalón unitario negativo

5.2. Control por Realimentación de Estados

Para realizar el diseño de un controlador bajo la representación en espacio de estado se debe diseñar la ley de control asumiendo la disponibilidad de la totalidad de los estados para realimentación [2], a pesar de que es inusual contar con las medidas de todos los estados (para el caso del PIS-UV solo se cuenta con dos de los cuatro estados medibles). Una vez diseñada la ley de control, se diseña un observador que calcule el vector de estados a partir de los estados medibles y las entradas al proceso. El diseño del observador de estados se lleva a cabo en la Sección 5.3.

La ley de control por realimentación de estados consiste en realimentar la planta con una combinación lineal de los estados de la misma de manera que:

$$\overrightarrow{u(k)} = -\overrightarrow{K} \overrightarrow{x(k)} \quad (5.1)$$

Para el PIS-UV, se busca diseñar un sistema de control tal que la señal de control V sea

$$V = -k_1 x - k_2 \dot{x} - k_3 \phi - k_4 \dot{\phi}$$

Donde $k_{1,2,3,4}$ son las ganancias de realimentación para cada estado.

La representación del modelo realimentado se obtiene al remplazar la señal de control

por la Ecuación 5.1. De esta manera se obtiene:

$$\begin{aligned}\overrightarrow{x(k+1)} &= G\overrightarrow{x(k)} - HK\overrightarrow{x(k)} \\ \overrightarrow{x(k+1)} &= [G - HK]\overrightarrow{x(k)}\end{aligned}$$

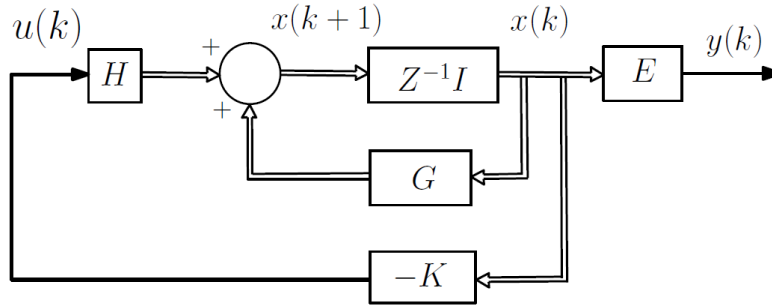


Figura 5.5: Diagrama de Bloques de Realimentación de Estados

Anteriormente se verificó la estabilidad del sistema mediante los valores propios de la matriz G , los cuales describen la dinámica de lazo abierto de la planta. Para diseñar la ley de control por realimentación de estados se debe verificar la estabilidad del sistema mediante los valores propios de $[G - HK]$ quienes representan la dinámica de lazo cerrado del sistema. Dado que G y H son matrices constantes, escoger adecuadamente los valores de las ganancias de realimentación $k_{1,2,3,4}$ que componen el vector K hace posible lograr una dinámica deseada para el sistema en lazo cerrado.

5.2.1. Controlabilidad del Sistema

Un sistema con estados x_i es controlable si se puede determinar una entrada tal que lleve los valores iniciales de los estados $x_i(t_0)$ a un estado deseado en un tiempo $t < \infty$, para cualquier valor de i y de t_0 .

Si se tiene:

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = x_0,$$

Reemplazando $k = 0, 1, 2, 3, \dots, n$ se obtiene que:

$$\begin{aligned}x(1) &= Ax(0) + Bu(0) \\ x(2) &= A^2x(0) + ABu(0) + Bu(1) \\ &\vdots \\ x(n) &= A^n x(0) + A^{n-1}Bu(0) + \dots + Bu(n-1) \\ x(n) - A^n x(0) &= \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \begin{bmatrix} u(n-1) & u(n-2) & \dots & u(1) & u(0) \end{bmatrix}^T\end{aligned}$$

La matriz de controlabilidad se define como:

$$CO = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$$

Para un sistema lineal invariante en el tiempo LTI es posible determinar su controlabilidad si y solo si su matriz de controlabilidad tiene rango completo, es decir si el rango de la matriz de controlabilidad es igual al numero de estados del sistema n , lo cual la hace una matriz invertible.

Si la matriz CO es invertible, es posible obtener que:

$$\begin{bmatrix} u(n-1) & u(n-2) & \dots & u(1) & u(0) \end{bmatrix}^T = CO^{-1} * [x(n) - A^n x(0)]$$

De esta manera, para cualquier valor de $x(n) = x_{final}$ es posible determinar la secuencia en la señal de entrada tal que se lleve el estado $x(0)$ a un valor x_{final} en un tiempo finito n [20].

La matriz de controlabilidad se obtiene en el software Scilab mediante el comando '`[CO]=cont_mat(A,B)`', y el rango de esta matriz se obtiene utilizando el comando '`rank(CO)`'. Para el caso del PIS-UV la matriz de controlabilidad posee un rango igual a 4, que coincide con la cantidad de estados del sistema por lo que se dice que el sistema es Completamente Controlable.

5.2.2. Regulador Lineal Óptimo Cuadrático LQR

En el control óptimo se busca llevar la operación de un sistema a realizarse a un costo mínimo. El control LQR permite conseguir un control óptimo en función de una función de costo que puede ser ajustada según el sistema a controlar. La función de costo se describe en la Ecuación 5.2 [21].

$$J = \int_0^{\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \quad (5.2)$$

La función de costo se define usualmente como la suma del error de mediciones específicas respecto a sus valores deseados. Esta función tiene en cuenta factores de peso o ponderación para cada uno de los estados del sistema y las entradas del mismo.

La dificultad de la utilización del algoritmo LQR para encontrar las ganancias de realimentación adecuadas para el control por realimentación de estados recae en la utilización de valores de ponderación adecuados para los estados y las entradas en las matrices Q y R respectivamente ya que no existe un método sistemático para encontrar dichos valores de peso. Sin embargo, es necesario tener en cuenta que una matriz Q con valores grandes requiere que los estados tomen valores pequeños para mantener mínima la función de costo J , mientras que una matriz R grande significa un menor esfuerzo de control y la introducción de polos lentos al sistema, lo cual provoca valores grandes en los estados para lograr minimizar la función de costo [22].

Se busca entonces que dada la Ecuación 5.1, se encuentre el vector $\vec{K}$ tal que se minimice la función de costo J sujeto a las restricciones del sistema que se desea controlar a partir de dos matrices que representan valores de ponderación para los estados y entradas.

Scilab cuenta con la función 'lqr' que genera las ganancias de realimentación para sistemas continuos más no para sistemas discretos. Para lograr obtener las ganancias de realimentación adecuadas para estabilizar un sistema discreto se utiliza la función contenida en el Anexo A.2.

Utilizando como matrices Q y R :

$$Q = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \quad R = \begin{bmatrix} 1 \end{bmatrix}$$

Se obtiene como vector de ganancias de realimentación el vector $\vec{K}$:

$$\vec{K} = \begin{bmatrix} -8,17 & -30,11 & 54,26 & 10,15 \end{bmatrix}$$

Los polos del sistema discreto realimentado con el vector K encontrado mediante la función 'dlqr' del Apéndice A.2 son:

$$\lambda = \begin{bmatrix} 0,144 & 0,983 + 0,008i & 0,983 - 0,008i & 0,996 \end{bmatrix} \quad (5.3)$$

La totalidad de estos polos esta contenida dentro del circulo de radio unidad en el plano Z por lo que el sistema realimentado con el vector K encontrado corresponde a un sistema estable.

5.2.3. Ganancias de Realimentación a partir de la Forma Canónica Controlable

Es muy simple calcular el vector $\vec{K}$ si el sistema se encuentra en la forma canónica controlable [2]. En la forma canónica controlable las matrices G y H del PIS-UV tienen

la forma:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_4 & -a_3 & -a_2 & -a_1 \end{bmatrix} \quad H = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Donde $a_{1,2,3,4}$ son los coeficientes de la ecuación característica 5.4.

$$|zI - G| = z^n + z^{n-1}a_1 + \dots + za_{n-1} + a_n = 0 \quad (5.4)$$

La matriz del sistema realimentado $G - HK$ es:

$$G - HK = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_4 - k_1 & -a_3 - k_2 & -a_2 - k_3 & -a_1 - k_4 \end{bmatrix}$$

La ecuación característica del sistema en red cerrada se muestra en la Ecuación 5.5 mientras que la ecuación característica deseada se muestra en la Ecuación 5.6.

$$z^4 + (a_1 + k_4)z^3 + (a_2 + k_3)z^2 + (a_3 + k_2)z + a_4 + k_1 = 0 \quad (5.5)$$

$$z^4 + \alpha_1 z^3 + \alpha_2 z^2 + \alpha_3 z + \alpha_4 = 0 \quad (5.6)$$

Igualando los coeficientes de las Ecuaciones 5.5 y 5.6, y despejando las variables $k_{1,2,3,4}$ se obtiene que:

$$k_4 = \alpha_1 - a_1$$

$$k_3 = \alpha_2 - a_2$$

$$k_2 = \alpha_3 - a_3$$

$$k_1 = \alpha_4 - a_4$$

Utilizando el código contenido en el Anexo A.3 en Scilab y utilizando como polos deseados aquellos obtenidos mediante el Algoritmo LQR (Ecuación 5.3) se obtiene la matriz G en su forma canónica controlable:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0,415 & 2,246 & -4,247 & 3,416 \end{bmatrix}$$

Y como vector de ganancias de realimentación el vector:

$$\vec{K} = \begin{bmatrix} -8,17 & -30,11 & 54,26 & 10,15 \end{bmatrix}$$

Así, mediante dos métodos diferentes se obtuvieron las mismas ganancias de realimentación que estabilizan la planta PIS-UV.

5.3. Estimación de Estados

Cuando la totalidad de las variables de estado de un sistema no pueden ser medidas, es necesario estimar los valores de las variables de estado desconocidas utilizando solo las variables de estado medibles y las señales de entrada del sistema.

Dado que solo es posible sensar dos de los estados del PIS-UV (X y ϕ), es necesario estimar los estados restantes de alguna manera. Para ello se utilizan dos métodos de estimación de estados. El primero corresponde a un observador basado en un estimador cuadrático lineal correspondiente a un Filtro de Kalman, mientras otro método posible es utilizar derivación discreta en las dos señales de salida y así obtener los dos estados restantes.

5.3.1. Observabilidad del Sistema

Un sistema es observable si el estado inicial x_0 puede ser determinado a partir de las salidas del sistema $\vec{y}$ en un tiempo finito.

Si se tiene:

$$\begin{aligned} x(k+1) &= Ax(k), \quad x(0) = x_0 = \text{desconocido}, \\ y(k) &= Cx(k) \end{aligned}$$

Reemplazando $k = 0, 1, 2, 3, \dots, n$ se obtiene que:

$$\begin{aligned} y(0) &= Cx(0) \\ y(1) &= CAx(0) \\ y(2) &= CA^2x(0) \\ &\vdots \\ y(n-1) &= CA^{n-1}x(0) \end{aligned}$$

$$\begin{bmatrix} y(0) & y(1) & y(2) & \dots & y(n-1) \end{bmatrix}^T = \begin{bmatrix} C & CA & CA^2 & \dots & CA^{n-1} \end{bmatrix}^T x(0)$$

La matriz de observabilidad se define como:

$$OB = \begin{bmatrix} C & CA & CA^2 & \dots & CA^{n-1} \end{bmatrix}^T$$

En un sistema LTI, el sistema es observable si y solo si su matriz de observabilidad tiene rango completo, ya que de esta manera es posible encontrar el valor de $x(0)$ a

partir del sistema de ecuaciones definido entre el historial de las salidas, la matriz de observabilidad y el estado inicial $x(0)$ [20].

La matriz de observabilidad se obtiene en el software Scilab mediante el comando `'[OB]=obsv_mat(A,C)'`, y el rango de esta matriz se obtiene utilizando el comando `'rank(OB)'`.

En el sistema en estudio la matriz de observabilidad posee un rango igual al número de estados del sistema, por lo que además de Completamente Controlable, este es un sistema Completamente Observable.

5.3.2. Estimación Mediante Filtro de Kalman

El filtro de Kalman se basa en un modelo de espacio de estados de un sistema para estimar optimamente el estado futuro teniendo en cuenta la aleatoriedad de las señales medibles y del ruido presente en ellas.

Este filtro solo depende de una medición anterior y de una medición actual disminuyendo considerablemente el uso de memoria requerido para su implementación [23].

El modelo del proceso que describe la transformación del vector de estados se representa mediante la Ecuación 5.7.

$$\vec{x}_k = G\vec{x}_{k-1} + H\vec{u}_{k-1} + W_{k-1} \quad (5.7)$$

Donde W_{k-1} representa ruido del proceso no correlacionado con los estados del sistema con densidad espectral de potencia W_p .

Existe además un modelo que representa la relación entre los estados y las medidas mostrado en la Ecuación 5.8.

$$\vec{Z}_k = H\vec{x}_k + V_k \quad (5.8)$$

Donde V_k representa ruido de la medición no correlacionado con los estados del sistema con densidad espectral de potencia V_p .

El estimador de estados basado en Filtro de Kalman posee la misma estructura básica de un observador [24]:

$$\hat{x} = H\hat{x} + Gu + L_k(y - C\hat{x})$$

Donde $\hat{x}$ es el vector de estados estimado y L_k es la matriz de diseño del estimador. El valor de esta matriz L_k se determina obtiene a partir de la siguiente expresión.

$$L_k = P_o C^T W_p^{-1}$$

Siendo P_o la matriz de covarianza óptima del error de estimación que es posible obtenerse mediante la resolución de la ecuación de Riccati, la cual es mostrada en la Ecuación 5.9 [24].

$$AP_o + P_o A^T - P_o C^T W_p^{-1} C P_o + V_p = 0 \quad (5.9)$$

El filtro de Kalman estima los estados de un sistema de manera similar a un controlador, en donde se estima el estado en una muestra y a la muestra siguiente se obtiene realimentación ruidosa de los valores reales de dicha estimación. Para diseñar un Estimador de Estados basado en Filtro de Kalman, Scilab cuenta con la función 'lqe' que permite obtener la matriz de ganancia de Kalman tal que minimice el error entre el estado estimado y el estado medido.

El observador discreto de Kalman obtenido tras la ejecución del código contenido en el Anexo A.4 se muestra en representación de espacio de estados discreto a continuación.

$$\begin{aligned}\overrightarrow{\hat{x}(k+1)} &= \begin{bmatrix} ObsA \end{bmatrix} \overrightarrow{\hat{x}(k)} + \begin{bmatrix} ObsB \end{bmatrix} \overrightarrow{u(k)} \\ \overrightarrow{y(k)} &= I \overrightarrow{\hat{x}(k)}\end{aligned}$$

Donde:

$$\begin{aligned}ObsA &= \begin{bmatrix} 1 & 3,33e-3 & -1,26e-4 & -3e-7 \\ -9,14e-4 & 4,15e-1 & 8,45e-3 & 2,45e-5 \\ -2,27e-3 & -6,20e-3 & 9,40e-1 & 4,85e-3 \\ -1,72e-2 & -2,21 & -1,47e-1 & 9,99e-1 \end{bmatrix} \\ ObsB &= \begin{bmatrix} 1e-4 & 8,20e-6 & 1,52e-4 \\ 3,51e-2 & 9,14e-4 & 9,33e-4 \\ 3,72e-4 & 2,27e-3 & 6,03e-2 \\ 1,32e-1 & 1,72e-2 & 3,69e-1 \end{bmatrix} \\ \overrightarrow{u(k)} &= \begin{bmatrix} V(k) \\ x(k) \\ \phi(k) \end{bmatrix}\end{aligned}$$

5.3.3. Estimación Mediante Derivación Discreta

Otro método para obtener una medición aproximada de los cuatro estados del sistema es mediante la derivación discreta de las señales medibles. Dado que los estados no medibles corresponden a las derivadas de primer orden de los dos estados medidos,

es posible implementar una derivada discreta a cada una de las mediciones y obtener los cuatro estados posteriormente.

La derivada discreta en Z se define como:

$$f'(z) = \frac{z-1}{zT_s} f(z) \quad (5.10)$$

Donde T_s es el tiempo de muestreo en segundos.

Dado que la derivación discreta de una señal con ruido entrega una señal aún más ruidosa, es necesaria la implementación de un filtro digital. El filtro digital es diseñado utilizando Scilab mediante el código contenido en el Apéndice A.5.

La implementación de los dos tipos de estimadores de estados en XCOS se muestra en la Figura 5.6.

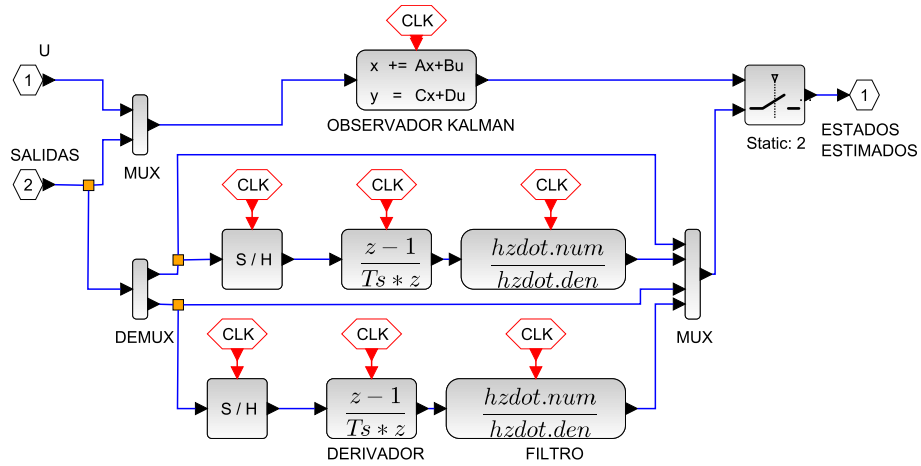


Figura 5.6: Implementación de los dos tipos de Estimadores de Estados en XCOS

5.4. Simulaciones del Sistema de Control

Para comprobar el adecuado funcionamiento de la ley de control diseñada en la Sección 5.2 y de los estimadores de estados diseñados en la Sección 5.3 utilizando el modelo en representación de espacio de estados discreto del PIS-UV, se realiza la simulación de cada caso en XCOS.

5.4.1. Simulación del Sistema en Condiciones Ideales

Inicialmente se realiza la simulación del sistema de control en el PIS-UV teniendo en cuenta que las mediciones de las salidas no poseen ruido electromagnético. Para esta

simulación se utiliza un tiempo de muestreo $T_s = 0,005 \text{ s}$ y un vector de estados inicial de:

$$\vec{x(0)} = \begin{bmatrix} 0,022 & 0 & 0,047 & 0 \end{bmatrix}^T$$

En la Figura 5.7 se muestra el diagrama creado en XCOS para realizar la simulación del sistema ideal teniendo en cuenta la zona muerta del actuador.

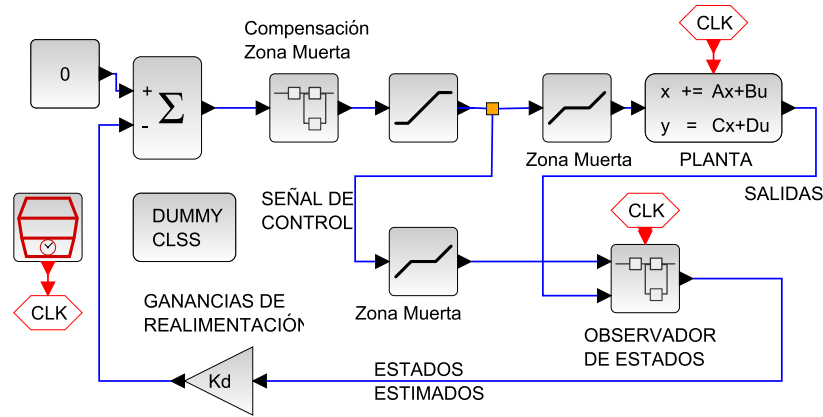


Figura 5.7: Diagrama en XCOS para la Simulación del Sistema de Control en Condiciones Ideales

La simulación en condiciones ideales realizada utilizando el estimador basado en Filtro de Kalman se muestra en las figuras 5.8 y 5.9.

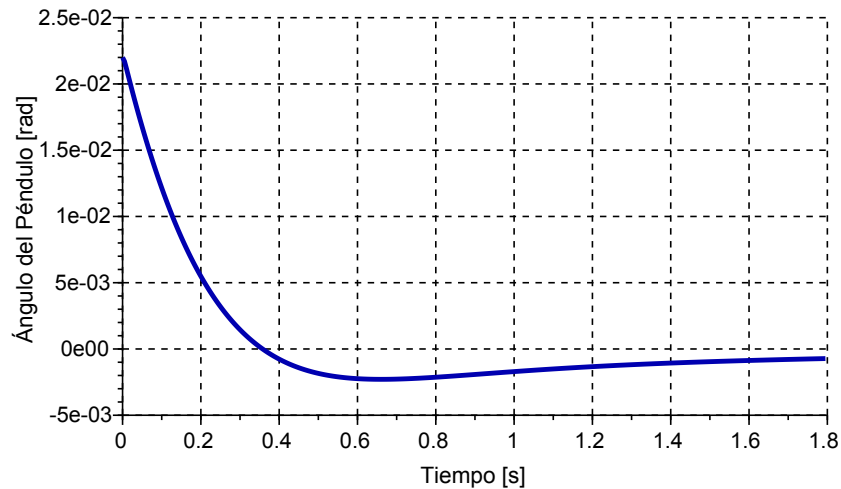


Figura 5.8: Simulación del Control en Condiciones Ideales utilizando Filtro Kalman como Estimador de Estados

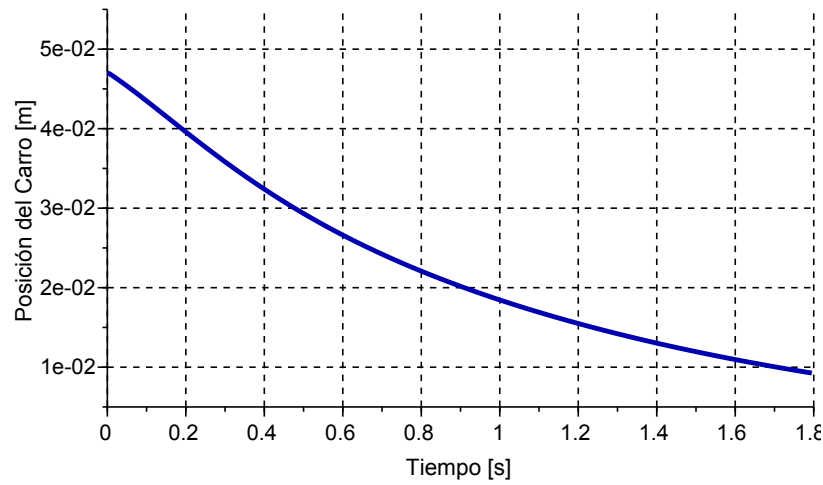


Figura 5.9: Simulación del Control en Condiciones Ideales utilizando Filtro Kalman como Estimador de Estados

En las figuras 5.10 y 5.11 se muestran los resultados de la simulación realizada utilizando el estimador de estados basado en derivador discreto.

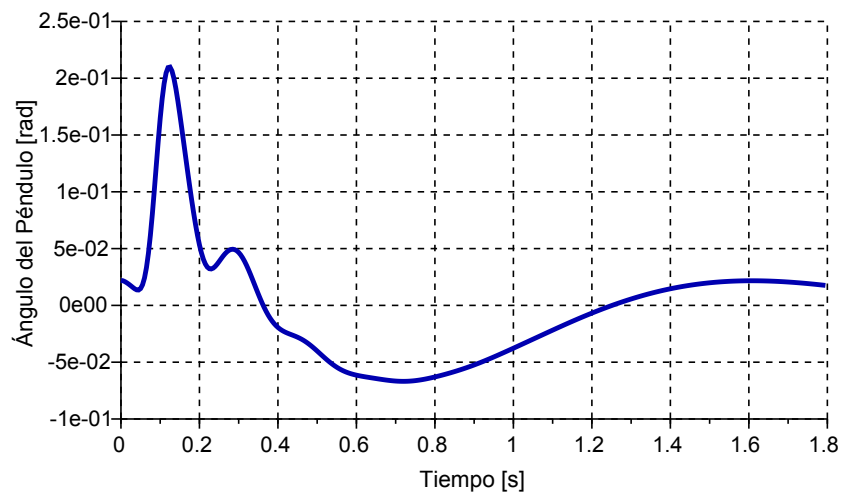


Figura 5.10: Simulación del Control en Condiciones Ideales utilizando Estimador de Estados basado en Derivación Discreta

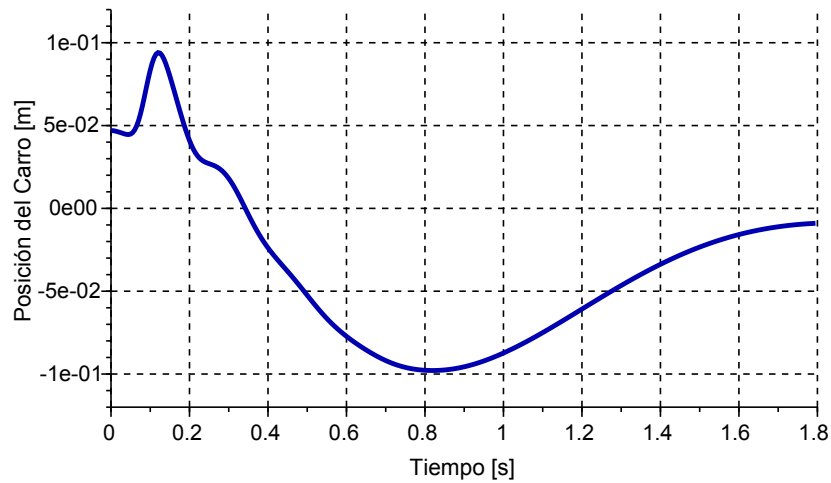


Figura 5.11: Simulación del Control en Condiciones Ideales utilizando Estimador de Estados basado en Derivación Discreta

En las figuras 5.8 y 5.9 se observa que utilizando el Filtro Kalman como estimador de estados se tiene una respuesta en el ángulo del péndulo con un sobrepaso del 11,36 % y un tiempo de estabilización de 1,4 s aproximadamente, mientras que la señal de posición del carro decrece suavemente sin estabilizarse tras 1,8 s de simulación.

Por otro lado, utilizando el estimador de estados basado en Derivación Discreta ambas señales logran la estabilización pero poseen sobrepasos de 318 % para el ángulo del péndulo y del 212 % para la posición el carro. Esto se debe a que los filtros digitales introducidos para eliminar ruido aleatorio en la derivada de las señales introducen un retardo en la señal.

Es notable el beneficio del observador de estados basado en Filtro de Kalman en el sistema de control al compararse con el método de la derivación discreta dado que no presenta sobrepasos de gran magnitud para la salida del ángulo del péndulo y su variación en posición es pequeña.

5.4.2. Simulación del Sistema con Ruidos y Error por Cuantización

Para obtener resultados más ajustados a la realidad se diseña una simulación de manera que se tenga en cuenta la zona muerta del actuador, el ruido electromagnético presente tanto en las mediciones como en la señal de control y los errores introducidos por cuantización en los convertidores análogo-digital y digital-análogo.

En la Figura 5.12 se muestra el diagrama creado en XCOS para realizar la simulación del sistema en condiciones reales.

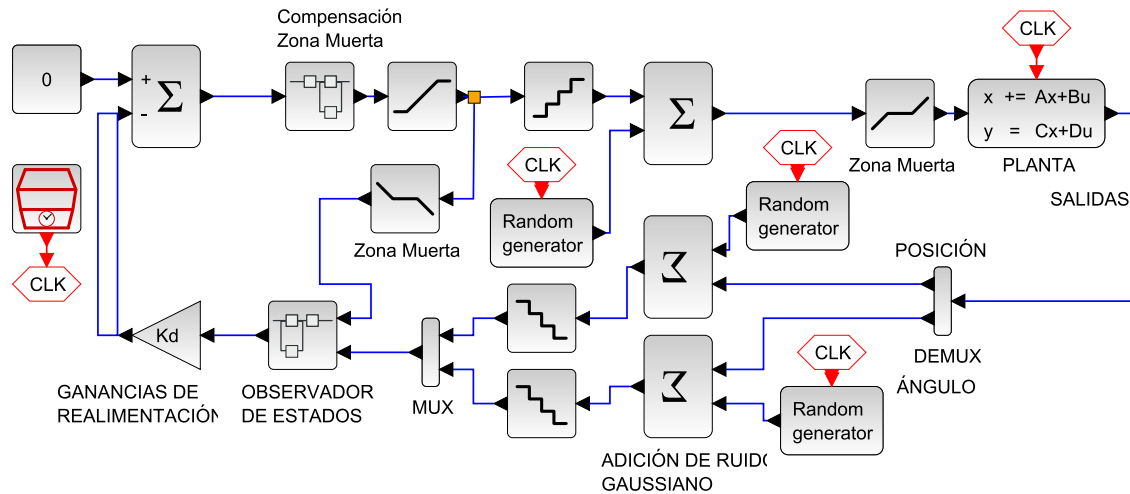


Figura 5.12: Diagrama en XCOS para la Simulación del Sistema de Control con Ruido y Cuantización

En las Figuras 5.13-5.14 y 5.15-5.16 se muestran los resultados de la simulación en condiciones reales utilizando como estimador de estados el Filtro Kalman y la Derivación Discreta respectivamente.

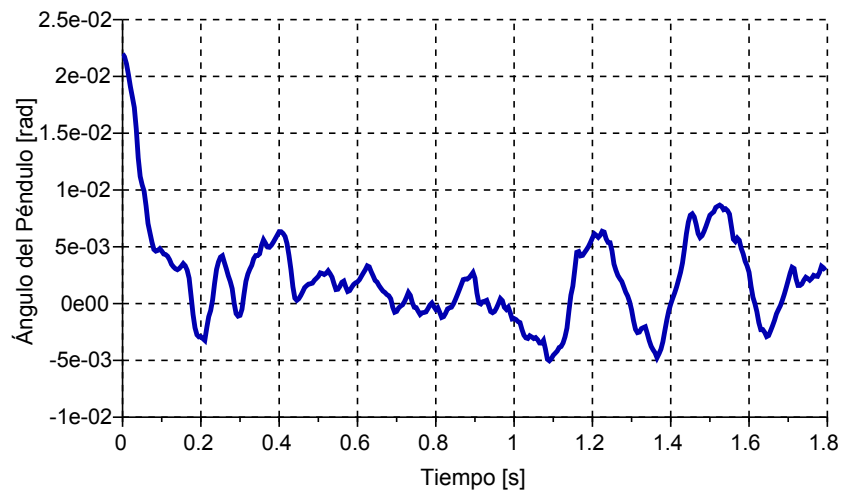


Figura 5.13: Simulación del Control en Condiciones Reales utilizando Filtro Kalman como Estimador de Estados

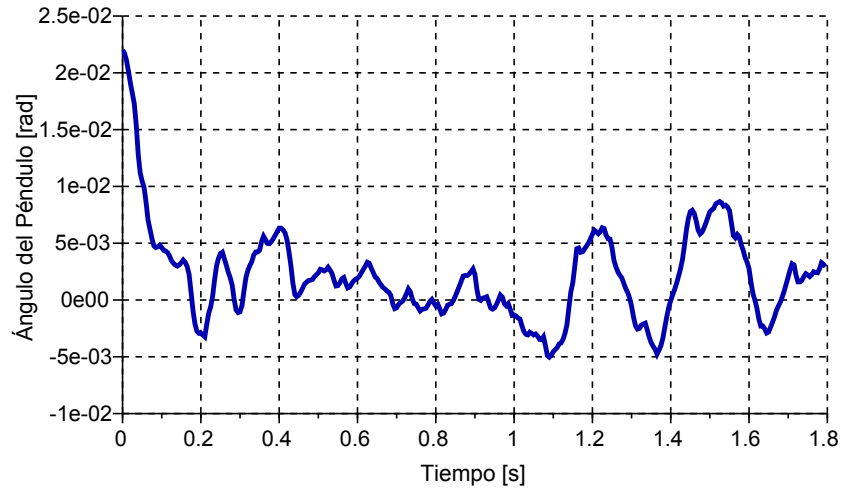


Figura 5.14: Simulación del Control en Condiciones Reales utilizando Filtro Kalman como Estimador de Estados

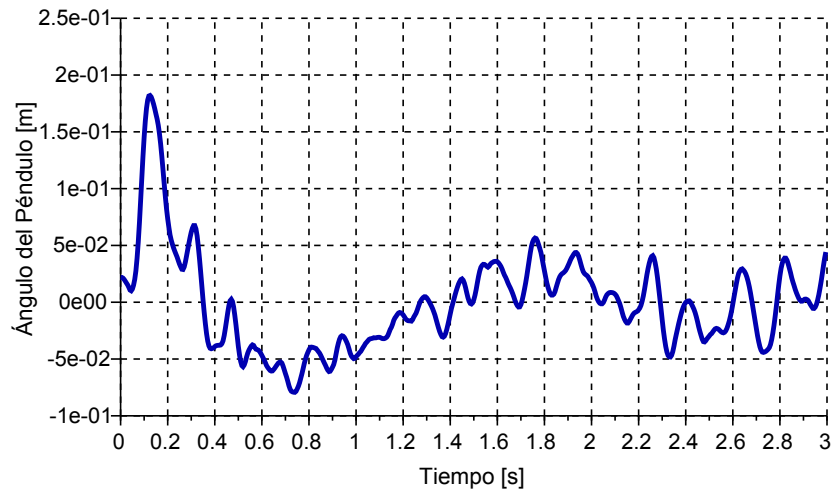


Figura 5.15: Simulación del Control en Condiciones Reales utilizando Estimador de Estados basado en Derivación Discreta

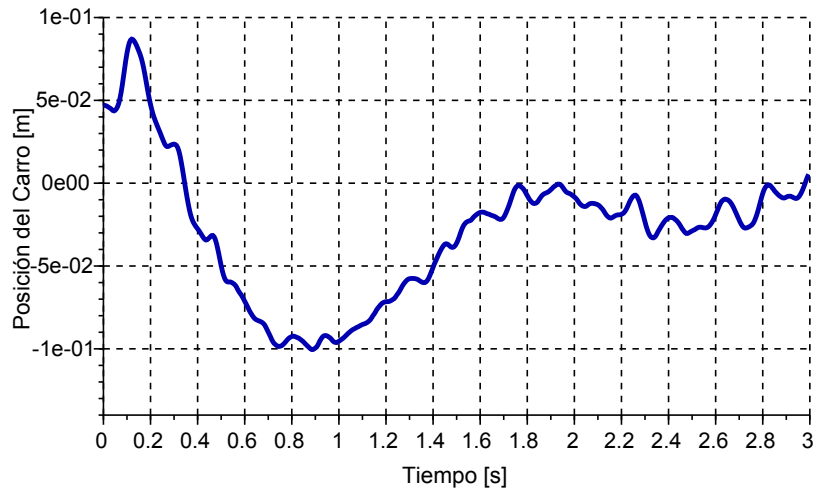


Figura 5.16: Simulación del Control en Condiciones Reales utilizando Estimador de Estados basado en Derivación Discreta

La Figura 5.13 evidencia un comportamiento similar al de la Figura 5.8 con los errores por cuantización y ruido presentes en la medición que hace que la señal sea más rizada. Lo mismo sucede con la Figura 5.15, donde al compararse con la Figura 5.10 se nota que describen la misma curva exceptuando el rizado y las constantes correcciones que realiza el controlador debido a las fuentes de error presentes en el sistema bajo condiciones reales.

Capítulo 6

Validación del Sistema de Control en la Planta

En este capítulo se realiza la validación del correcto funcionamiento de la planta PIS-UV con la estrategia de control diseñada en el Capítulo 5 utilizando como estimador de estados un Filtro de Kalman y la Derivación Discreta. Además se detallan las restricciones encontradas que no permiten el adecuado funcionamiento de la estrategia de control en la planta real.

En la Sección 6.1 se prueba en la planta PIS-UV el controlador diseñado anteriormente comparando además la respuesta según el estimador de estados utilizado y tomando como condiciones iniciales valores muy pequeños en los 4 estados del sistema.

Las restricciones encontradas en el sistema de control aplicado a la planta real se muestran en la Sección 6.2.

6.1. Validación de la ley de control en la planta PIS-UV

Para comprobar el desempeño de la ley de control diseñada para la planta PIS-UV, se utiliza Scilab/XCOS y el Módulo Arduino para Scilab.

En la Figura 6.1 se muestra el diagrama en XCOS que permite la ejecución del controlador y del estimador de estados en la planta PIS-UV real. Este diagrama incluye el subsistema de compensación de zona muerta, el bloque de conversión de señales de sensores en datos reales de salidas y el subsistema conversor de voltaje deseado en número de 8 bits.

El tiempo de muestreo corresponde a 5 ms y el vector de ganancias de realimentación es el mismo encontrado en la Sección 5.2.

$$\vec{K} = \begin{bmatrix} -8,17 & -30,11 & 54,26 & 10,15 \end{bmatrix}$$

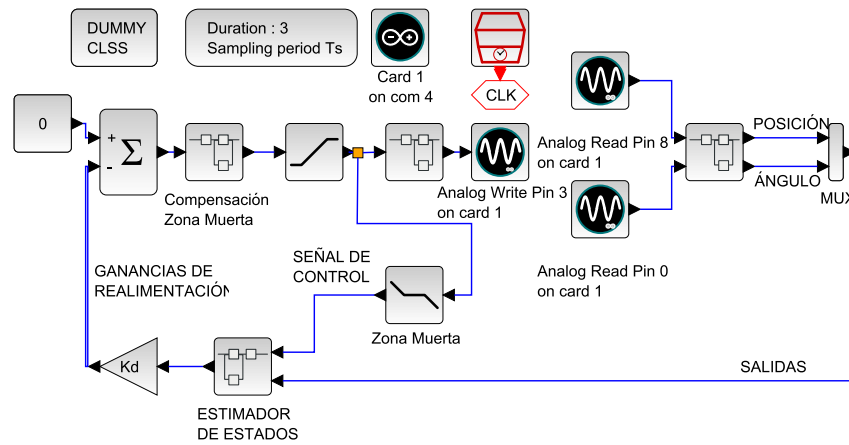


Figura 6.1: Diagrama en XCOS para la Validación del Sistema de Control en la Planta Real

Tras ubicar el péndulo con una inclinación de $0,022 \text{ rad}$ y el carro en una posición de $0,047 \text{ m}$ se ejecuta el sistema de control. La respuesta en Ángulo del Péndulo y Posición del Carro obtenida tras la ejecución del mismo se muestra en las figuras 6.2 y 6.3.

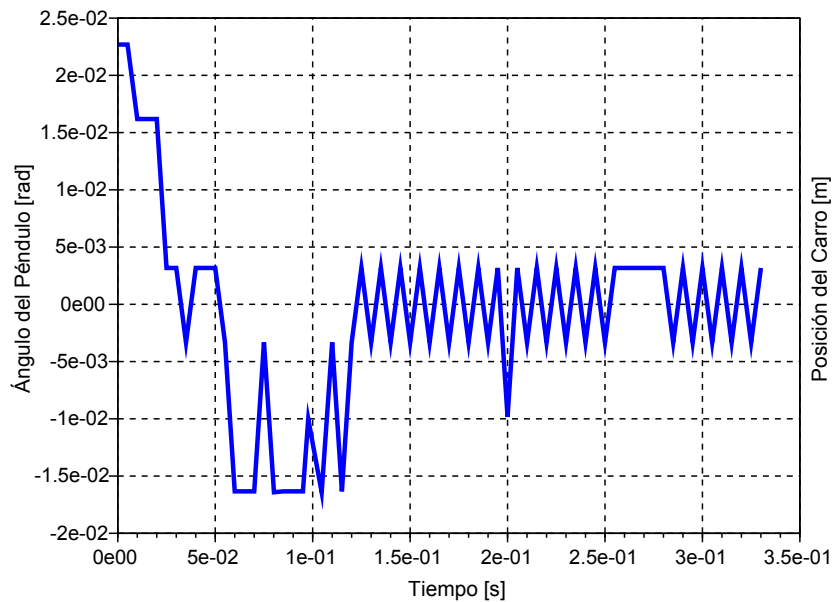


Figura 6.2: Respuesta Real del Sistema de Control utilizando Filtro Kalman como Estimador de Estados

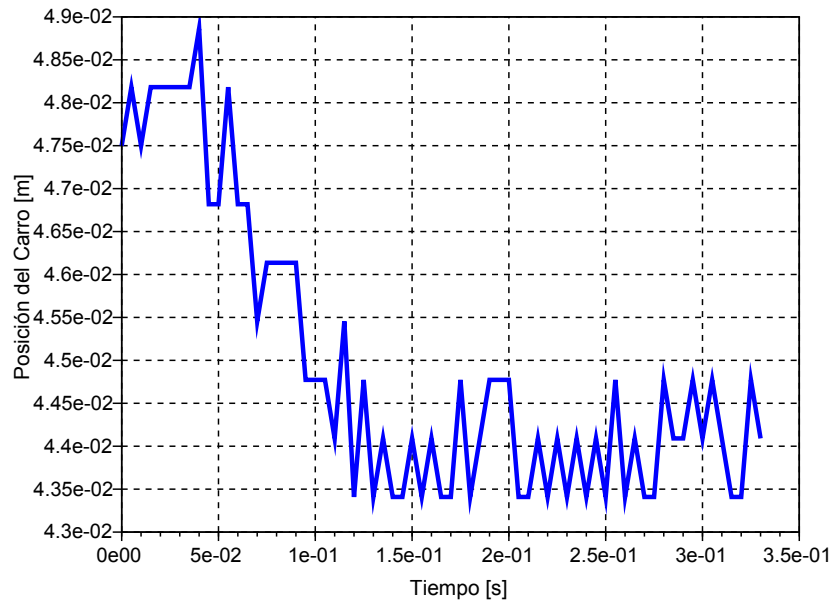


Figura 6.3: Respuesta Real del Sistema de Control utilizando Filtro Kalman como Estimador de Estados

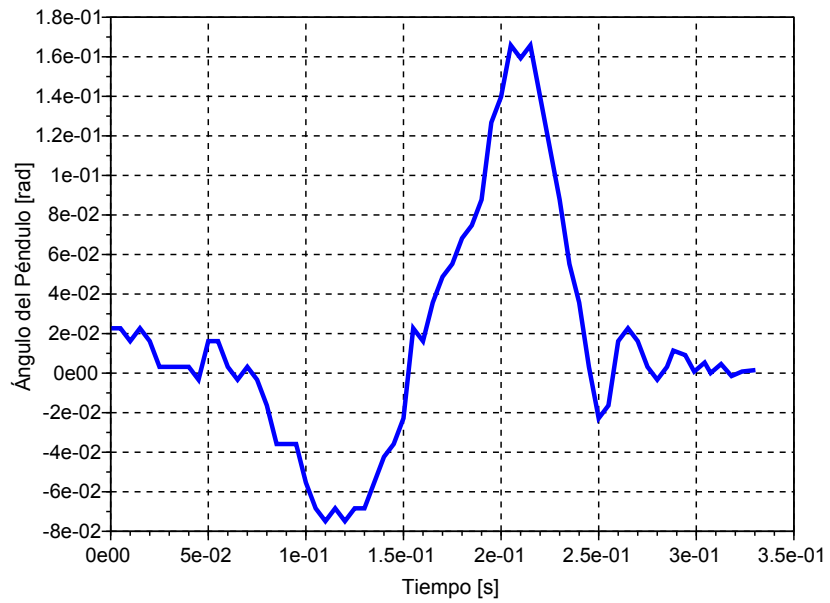


Figura 6.4: Respuesta Real del Sistema de Control utilizando Estimador de Estados basado en Derivación Discreta

Posteriormente se realiza la validación del sistema de control con el estimador de estados basado en Derivación Discreta cuyas salidas se muestran en las figuras 6.4 y 6.5.

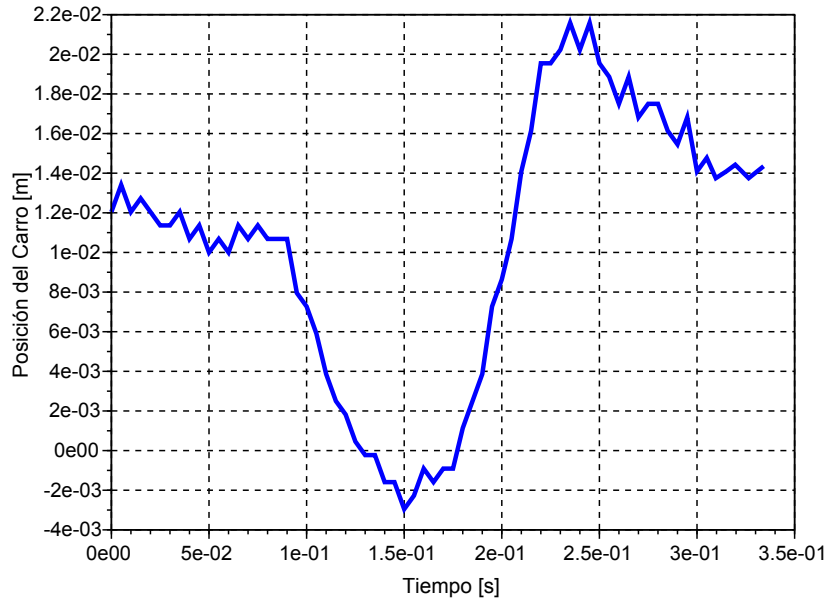


Figura 6.5: Respuesta Real del Sistema de Control utilizando Estimador de Estados basado en Derivación Discreta

Utilizando el estimador de estados basado en Filtro Kalman se estabiliza el ángulo en cero tras 0,13 s de ejecución, teniendo un sobrepaso del 72,7%. Por otro lado la posición del carro tiene una pequeña variación estabilizándose en un valor diferente a cero tras aproximadamente 0,12 s.

En las figuras 6.4 y 6.5 se evidencia que utilizando el estimador de estados basado en derivación discreta, el ángulo del péndulo logra estabilizarse tras 0,31 s con un sobrepaso del 341 % mientras que la posición del carro lo hace tras 0,3 s con un sobrepaso del 400 %

6.2. Restricciones en la planta PIS-UV

Tras analizar la respuesta real de la planta PIS-UV dentro del sistema de control diseñado, se procede a estudiar las restricciones que presentó la misma para su adecuado funcionamiento.

Para el caso del sistema de control con estimador de estados basado en Derivación Discreta se obtuvo una señal controlada tras 0,3 s de ejecución, sin embargo tras aproximadamente 0,035 s se presenta un pulso en la medición de la señal del ángulo del péndulo no correspondiente a la posición angular real como se muestra en la Figura 6.6.

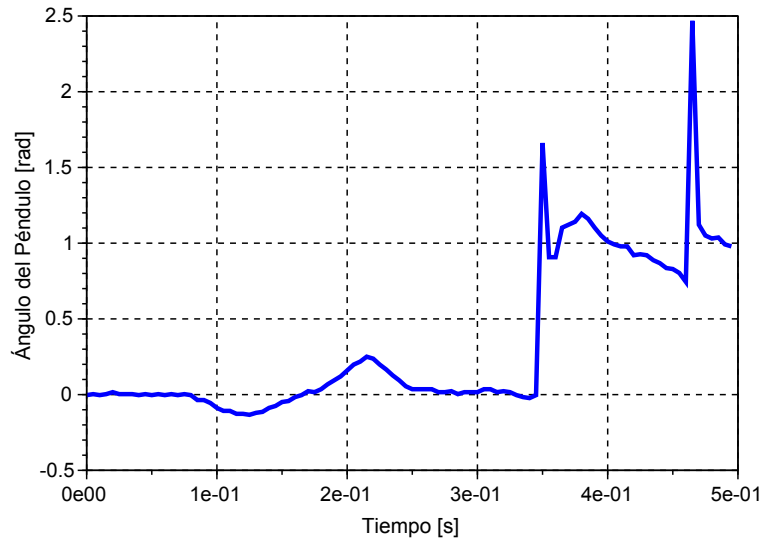


Figura 6.6: Presencia de Pulso en la medición de la señal de ángulo utilizando Estima-
dor de Estados basado en Derivación Discreta

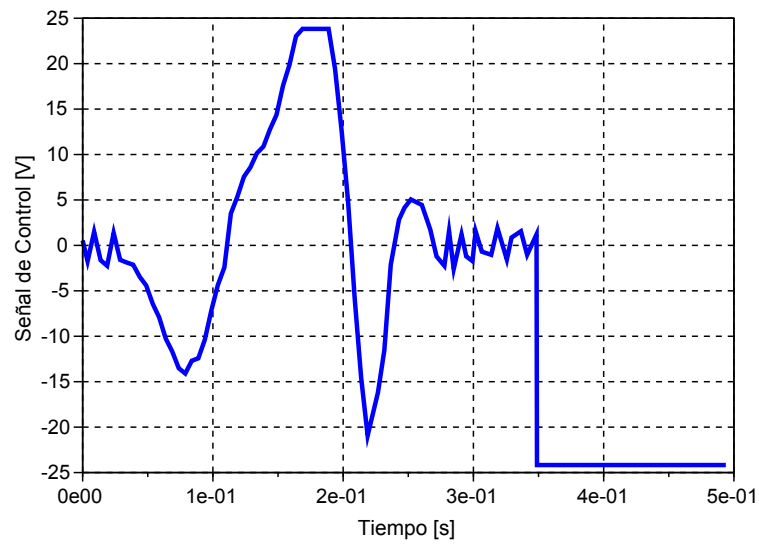


Figura 6.7: Presencia de Pulso en la medición de la señal de ángulo utilizando Estima-
dor de Estados basado en Derivación Discreta

Este pulso puede deberse a un error momentáneo en el segundo bit más significativo de la representación de la señal del ángulo proveniente del Arduino Nano hacia el Conversor Digital-Análogo TLV5616. Debido a este pulso tan grande (del orden de los $1,57 \text{ rad}$) la señal de control toma valores que saturan el actuador y por ende el sistema no es controlado. Esto no sucede al utilizar el Filtro Kalman como estimador de estados ya que este rechaza cambios bruscos en las señales de los 4 estados manteniendo una señal suave que es multiplicada por el vector $\tilde{K}$.

Por otro lado, el sistema de control no funcionó apropiadamente con valores iniciales de ángulo del péndulo mayor a $0,03 \text{ rad}$ ya que en estos casos el sistema de control no logró sostener el péndulo en su punto de equilibrio inestable. Esto puede deberse a características no lineales muy fuertes en la planta PIS-UV no tenidas en cuenta durante el modelado de la planta, diseño de la estrategia de control y diseño del observador de estados.

Utilizando la simulación en condiciones reales mostrada en la Sección 5.4, se estudió el efecto de los retardos de medición en la acción de control implementada. Se añadieron bloques de retardo en la señal de cada una de las salidas del diagrama de bloques mostrado en la Figura 5.12 para determinar heurísticamente el máximo retardo permitido en la medición de la señal del ángulo del péndulo para que el sistema de control siga siendo efectivo. Se escogió la señal del ángulo como fuente de retardo debido a que es esta señal la que puede presentar un retardo significativo para el sistema, dado el procesamiento de la señal proveniente del módulo GY-521 en el Arduino Nano utilizado y su posterior paso por el conversor digital-análogo.

Para estas simulaciones se configuró un retardo fijo en la señal de posición de $0,001 \text{ s}$ mientras se modificaba el retardo de la señal de ángulo paulatinamente.

Para el caso del sistema de control con un Filtro Kalman como estimador de estados, se obtuvo que el máximo retardo permisible para que el sistema de control actúe efectivamente en la planta es de $0,0042 \text{ s}$. La respuesta del sistema utilizando un retraso de $0,0043 \text{ s}$ en la señal de ángulo y de $0,001 \text{ s}$ en la señal de posición, se muestra en la Figura 6.8.

El sistema de control con estimador basado en Derivación Discreta presentó un límite máximo en el retardo de la señal de ángulo igual a $0,0041 \text{ s}$. Al aplicar un retardo de $0,0042 \text{ s}$ en la señal de ángulo y $0,001 \text{ s}$ en la señal de posición se obtiene la respuesta mostrada en las figuras 6.10 y 6.11.

Los retardos en la medición pueden ser otro factor condicionante que ocasionó el no funcionamiento del sistema de control implementado en la planta real para ángulos iniciales mayores a $0,03 \text{ rad}$.

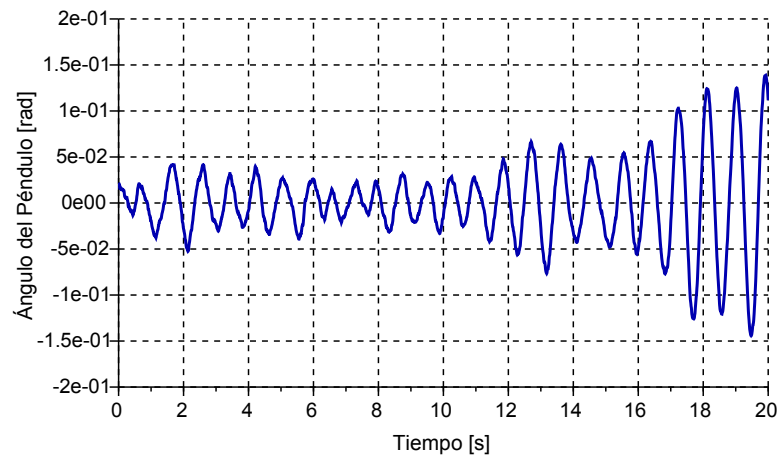


Figura 6.8: Respuesta del sistema de control con estimador Filtro Kalman ante la presencia de retardos en medición

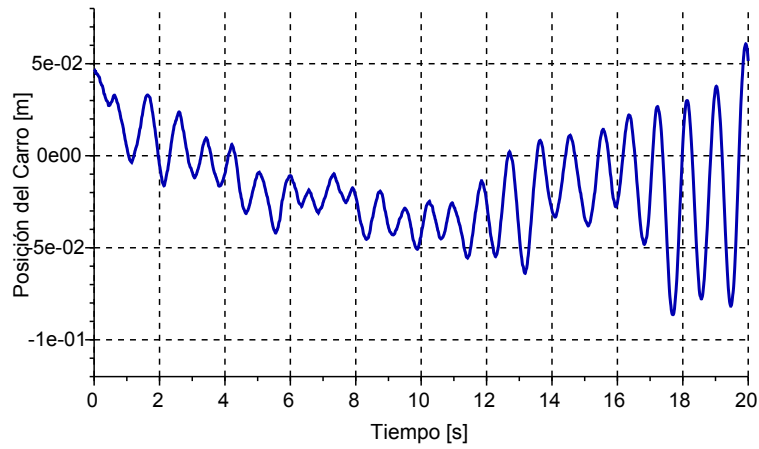


Figura 6.9: Respuesta del sistema de control con estimador Filtro Kalman ante la presencia de retardos en medición

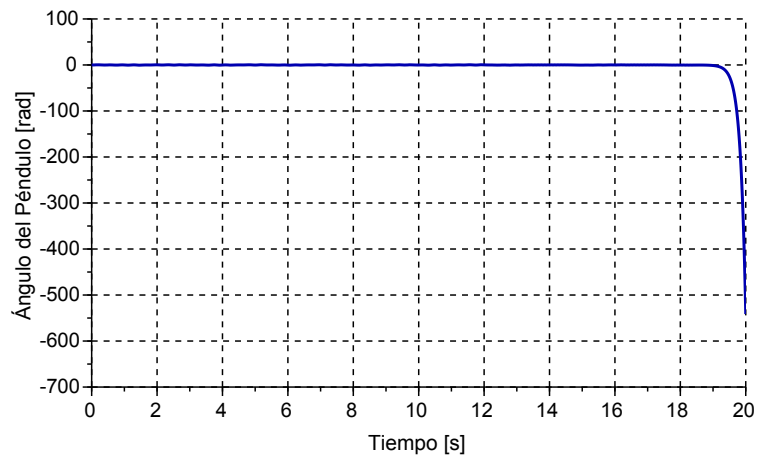


Figura 6.10: Respuesta del sistema de control con estimador basado en derivación discreta ante la presencia de retardos en medición

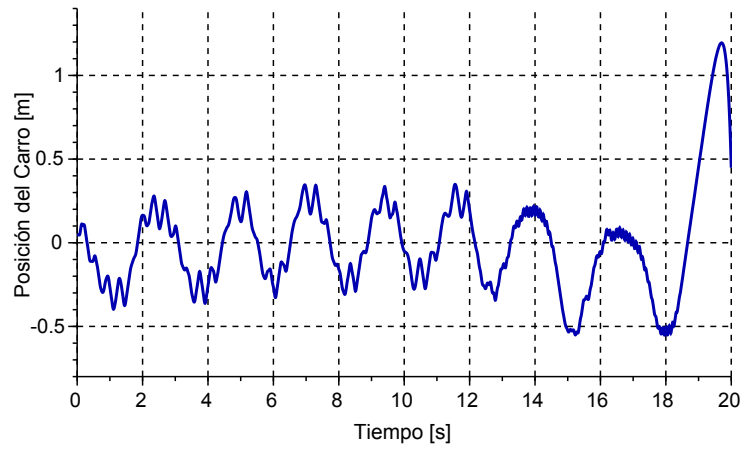


Figura 6.11: Respuesta del sistema de control con estimador basado en derivación discreta ante la presencia de retardos en medición

Capítulo 7

Conclusiones y Trabajo Futuro

Los resultados obtenidos en el desarrollo de este proyecto de ingeniería permiten definir ciertas conclusiones sobre el trabajo con el Arduino como tarjeta DAQ, los sensores utilizados, el software Scilab y la planta PIS-UV. Estas conclusiones se muestran en la Sección 7.1.

Es posible además conjeturar sobre diversas líneas de avance futuras para continuar mejorando la planta del Péndulo Invertido Simple - UV y aumentando los beneficios que esta le presta a la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle como se muestra en la Sección 7.2.

7.1. Conclusiones

Siguiendo la motivación que originó este proyecto de ingeniería, se puso en funcionamiento la planta de Péndulo Invertido Simple (PIS-UV) que no contaba con elementos de sensórica, potencia o control, utilizando como herramienta de cómputo el software libre Scilab que se comunica con la planta mediante una tarjeta Arduino Mega 2560 que funciona como DAQ. Como sensores se utilizó un módulo GY-521 de 6 grados de libertad que posee un giroscopio y un acelerómetro para medir el ángulo del péndulo tras la utilización de un Filtro Kalman, y un sensor infrarrojo SICK DT20-P130B1080 para medir la posición del carro.

Se realizó un estudio de la planta PIS-UV presente en el Laboratorio de Automática de la Universidad del Valle obteniendo los parámetros de la misma y de su actuador para obtener su modelo adecuadamente mediante representación en espacio de estados. Durante la caracterización del motor actuador de la planta, se encontró cual era el voltaje de ruptura de la zona muerta del mismo sin carga que correspondía a $V_r = 0,5 \text{ V}$, sin embargo una vez probado el funcionamiento del motor dentro de la planta PIS-UV se notó un aumento en la zona muerta del motor ($V_r = 1,42 \text{ V}$) y una desigualdad entre el comportamiento del motor al ser excitado positivamente respecto al comportamiento con excitación negativa. Dado esto fue necesario compensar esta desigualdad alterando

la señal dirigida al controlador del motor desde Scilab y desde el módulo de Arduino para Scilab.

Inicialmente se pretendía utilizar una tarjeta DAQ de National Instruments para la interacción entre Scilab y el PIS-UV, sin embargo tras analizar el funcionamiento del módulo NIDAQ para Scilab se determinó que este está dirigido principalmente a realizar el procesamiento de las señales una vez se ha finalizado su plena adquisición, no permitiendo interacción sobre la señal adquirida en tiempo real. Por ello se prefirió utilizar una tarjeta Arduino Mega 2560 que en conjunto con el módulo de Arduino para Scilab si permite la interacción de Scilab/XCOS con la planta en tiempo real bajo la restricción de un periodo de muestreo mayor o igual a 5 ms.

El modelo linealizado en representación de espacio de estados encontrado para la planta PIS-UV fue validado con datos reales medidos en la planta, evidenciando que este modelo describe aceptablemente la dinámica del sistema presentando pequeños errores debidos a la no linealidad y parámetros físicos presentes en la planta real. Para obtener un modelo de mayor fidelidad con la realidad del PIS-UV es necesario que este sea un modelo no lineal y que se tenga en cuenta una mayor cantidad de variables físicas de la planta incluidas las fricciones dinámicas y estáticas del carro con el riel y del eje de giro del péndulo con el carro.

Para el PIS-UV se decidió utilizar un controlador por realimentación de estados. En el cálculo de las ganancias de realimentación se utilizó el control LQR y el diseño a partir de la forma canónica controlable. Utilizando ambos métodos sobre la representación discreta de la planta se obtuvo como resultado el mismo vector de ganancias de realimentación ($\bar{K}$) corroborando la adecuada implementación de los algoritmos de diseño mostrados en los Anexos A.2 y A.3.

Utilizando las ganancias de realimentación encontradas y el modelo discreto de la planta, se diseñaron dos tipos de estimadores de estados, un Filtro Kalman y un estimador basado en Derivación Discreta. El desempeño de estos dos tipos de estimadores de estados se comparo mediante simulaciones cuya única diferencia es el estimador utilizado. En estas simulaciones se evidenció que el Filtro Kalman entrega unas señales de estados estimados con poco ruido y permiten que la acción de control actúe eficientemente sobre la planta. Por otro lado el estimador basado en Derivación Discreta introduce retardos en los estados no medibles debido a la necesidad de aplicar un filtro digital tras realizar la derivación, lo cual provoca que la acción de control tenga un mayor trabajo para estabilizar el sistema.

Al realizar la validación de la estrategia de control diseñada sobre la planta PIS-UV real, se encuentra que al utilizar el estimador basado en Derivación Discreta las señales de ángulo y posición se estabilizan aproximadamente 2,7 veces más tarde comparada con la respuesta al utilizar el Filtro Kalman como estimador; además de poseer sobrepasos del orden de 341 % para el ángulo comparado con un sobrepaso del 72,7 % utilizando el Filtro Kalman como estimador. No fue posible controlar la planta PIS-UV para valores iniciales de ángulo mayores a $\pm 0,03$ rad debido posiblemente a variables

físicas y no linealidades que no se tuvieron en cuenta durante ninguno de los diseños presentes en este trabajo además de posibles retardos en la adquisición y procesamiento de la señal proveniente de la sensórica del ángulo del péndulo.

7.2. Trabajo Futuro

Teniendo en cuenta el desarrollo logrado tras la finalización de este proyecto de ingeniería, se plantean una serie de avances que podrían realizarse a partir del trabajo aquí realizado.

Conociendo las limitaciones del toolbox NIDAQ para Scilab en el control en tiempo real y aprovechando las características de código libre de este, es posible editar el toolbox o crear uno nuevo basado en el actual de manera que se incluyan características funcionales adecuadas para realizar control en tiempo real mediante las tarjetas de adquisición de datos de National Instruments y Scilab sobre sistemas operativos de Microsoft.

Utilizando el hardware disponible tras la realización de este proyecto es posible desarrollar estrategias de control de mayor complejidad como el control predictivo basado en modelos con restricciones y estrategias de control no lineal para obtener mejores resultados en la implementación del sistema de control en la planta real.

Además se propone crear un laboratorio virtual del Péndulo Invertido en donde los estudiantes puedan probar los sistemas de control diseñados por ellos en un ambiente seguro que represente fielmente las características del PIS-UV presente en la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle.

Tras obtener resultados óptimos en la implementación del controlador en la planta PIS-UV y gracias a la posibilidad de sensar absolutamente la posición angular del péndulo se puede diseñar un controlador que realice el levantamiento *swing up* desde el punto de equilibrio estable ($\phi = \pi \text{ rad}$) hasta una zona donde empiece a actuar el controlador diseñado para mantener el péndulo en su zona de equilibrio inestable ($\phi = \pm 0,25 \text{ rad}$)

Mediante el uso de una plataforma de hardware libre o de un ordenador de placa reducida se abre la posibilidad de diseñar y construir un sistema embebido para el control de la planta PIS-UV utilizando también conectividad a Internet para obtener un laboratorio remoto del PIS-UV.

Para disminuir los errores no contemplados en el modelado de la planta es posible sustituir el motor actuador de la planta y la manera en que este aplica la fuerza de empuje que mueve el carro del PIS-UV.

Bibliografía

- [1] Universidad del Valle, Grupo de Investigación en Control Industrial. Infraestructura [En línea] [Consultado 19 Dic. 2014]. Disponible en: <http://gici.univalle.edu.co/PlataformaPERI.html>
- [2] RAMIREZ, J.M., ROSERO, E.E., *SISTEMAS DE CONTROL II*, Grupo de Investigación en Control Industrial-GICI, Universidad del Valle, Santiago de Cali, Colombia, 2008.
- [3] RAMIREZ, J.M., ROSERO, E.E., *Fundamentos de Control de Sistemas Lineales*, Universidad del Valle, Santiago de Cali, Colombia, 2008.
- [4] TRIVERIO, M., *Progetto e Implementazione del Sistema di Controllo per un Pendolo Inverso*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, 2008, [En línea] [Consultado 6 Oct. 2014]. Disponible en: http://www.cs.uic.edu/~mtriveri/theses/Triverio_Pendulum_Poli2008.pdf
- [5] PIN. A., *Progetto di un Sistema di Controllo di un Pendolo Inverso in ambiente MATLAB*, Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova, 2012, [En línea] [Consultado 8 Oct. 2014]. Disponible en: http://tesi.cab.unipd.it/40863/1/Alessandro_Pin.pdf
- [6] BINTI, N., *Linear Quadratic Gaussian (LQG) Controller for Inverted Pendulum*, Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn Malaysia, 2013, [En línea] [Consultado 16 Dic. 2014]. Disponible en: http://eprints.uthm.edu.my/4728/1/NORHIDAYAH_BINTI_AHMAD_24pg.pdf
- [7] LENKA, N., *Modeling and Controller Design for an Inverted Pendulum System*, Department of Electrical Engineering, National Institute of Technology, India, 2011, [En línea] [Consultado 3 Ene. 2015]. Disponible en: http://ethesis.nitrkl.ac.in/2213/1/Inverted_Pendulum.pdf
- [8] QUINTERO, S.A.P., *Controlling the Inverted Pendulum*, Department of Electrical and Computer Engineering, University of California, 2014, [En línea] [Consultado 16 Dic. 2014]. Disponible en: http://control.ee.ethz.ch/~ifaatic/2014/student_projects/Quintero_238_report.pdf

- [9] HERRERA, L.V., MELO, U.A., *Control en Espacio de Estados para un Prototipo Real de Péndulo Invertido*, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Universidad Industrial de Santander, 2013, [En línea] [Consultado 13 Nov. 2014]. Disponible en: <http://tangara.uis.edu.co/biblioweb/tesis/2013/150451.pdf>
- [10] RODRÍGUEZ, O.O., CELY, H.E., RIAÑO, J.A., *Diseño e Implementación de un Péndulo Invertido Sobre un Carro Aplicando Estrategias de Control Basado en LMI*, Revista Colombiana de Tecnologías de Avanzada vol. 1, Número 19, 2013.
- [11] CACCAVALE, G., *MPU-6050 (GY-521) Arduino Tutorial*, 2014, [En línea] [Consultado 10 Nov. 2014]. Disponible en: <http://www.giusepecaccavale.it/arduino/mpu-6050-gy-521-arduino-tutorial/>
- [12] JESPERSEN, T., LAUSZUS, K., *Kalman Filter / Arduino*, 2012, [En línea] [Consultado 12 Nov. 2014]. Disponible en: <https://github.com/TKJElectronics/KalmanFilter>
- [13] TEXAS INSTRUMENTS, *TLV5616 2.7-V to 5.5-V LOW POWER 12-BIT DAC*, 1997, [En línea] [Consultado 15 Nov. 2014]. Disponible en: <http://www.ti.com.cn/cn/lit/ds/symlink/tlv5616.pdf>
- [14] SICK SENSOR INTELLIGENCE, *Short range distance sensors (displacement) DT20-P130B1080 Online Datasheet*, [En línea] [Consultado 02 Feb. 2015]. Disponible en: https://www.sick.com/media/pdf/9/69/369/dataSheet_DT20-P130B1080_1028720_en.pdf
- [15] BURR-BROWN, *RCV420 Precision 4mA to 20mA CURRENT LOOP RECEIVER Online Datasheet*, [En línea] [Consultado 03 Feb. 2015]. Disponible en: <http://www.uni-kl.de/elektronik-lager/416556>
- [16] RAWASHDEH, M., *Instructables - Motor Driver BTS7960 43A*, [En línea] [Consultado 16 Ene. 2015]. Disponible en: <http://www.instructables.com/id/Motor-Driver-BTS7960-43A/>
- [17] NATIONAL INSTRUMENTS, *User Guide NI USB-6001/6002/6003 Low-Cost DAQ USB Device*, [En línea] [Consultado 14 Ene. 2015]. Disponible en: <http://www.ni.com/pdf/manuals/374259a.pdf>
- [18] ARDUINO, *Arduino Mega 2560*, [En línea] [Consultado 15 Nov. 2014]. Disponible en: <http://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [19] PICHARDO, J., *Compensación de Zona Muerta y Variación de Carga en el Control de Velocidad de Motores de Corriente Directa*, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Instituto Politécnico Nacional, México, 2008, [En línea] [Consultado 23 Feb. 2015]. Disponible en: <http://itzamna.bnct.ipn.mx/dspace/handle/123456789/2384>

- [20] GAJIC, Z. , *Controllability and Observability*, Department of Electrical and Computer Engineering, Rutgers School of Engineering, Estados Unidos, 2014, [En línea] [Consultado 10 May. 2015]. Disponible en: www.ece.rutgers.edu/~gajic/psfiles/chap5traC0.pdf
- [21] CHOI, J. , *Linear Control Systems, Lecture 14, Linear Quadratic Regulator (LQR)*, College of Engineering, Michigan State University, Estados Unidos, 2010, [En línea] [Consultado 3 Abr. 2015]. Disponible en: http://www.egr.msu.edu/classes/me851/jchoi/lecture/Lect_14.pdf
- [22] LEWIS, F.L. , *Linear Quadratic Regulator (LQR) State Feedback Design*, Advanced Controls and Sensors Group, University of Texas Arlington, Estados Unidos, 1998, [En línea] [Consultado 5 Abr. 2015]. Disponible en: <http://www.uta.edu/utari/acs/Lectures/lqr.pdf>
- [23] CASTAÑEDA, J.A., NIETO M.A. ORTIZ, V.A. , *Análisis y Aplicación del Filtro Kalman a una Señal con Ruido Aleatorio*, Universidad Tecnológica de Pereira, Colombia, 2013, [En línea] [Consultado 29 Mar. 2015]. Disponible en: dialnet.unirioja.es/descarga/articulo/4320424.pdf
- [24] CÁRDENAS, G., *Diseño de Regulador LQR y Observador por Filtro de Kalman: Sistema Ball and Beam*, Universidad de la Frontera, Chile, 2010, [En línea] [Consultado 29 Mar. 2015]. Disponible en: <https://www.scribd.com/doc/33944259/Diseño-de-regulador-LQR-y-Observador-por-%EF%AC%81ltro-de-Kalman-Sistema-Ball-and-Beam>

Anexo A

Códigos utilizados en el desarrollo

A.1. Modificaciones al Programa toolbox arduino v3.ino

```
int RPWM=3;
int LPWM=5;
int L_EN=7;
int R_EN=8;
toolbox_arduino_V3.ino
void setPWMfrequency(int freq){
    TCCR1B = TCCR2B & 0b11111000 | freq ;
    TCCR3B = TCCR2B & 0b11111000 | freq ;
}
void MotorActiveStatus(char Side,boolean s){
    boolean state=s;
    if(Side=='R'){
        digitalWrite(R_EN,s);
    }
    if(Side=='L'){
        digitalWrite(L_EN,s);
    }
}
void setMotor(char side,byte pwm){
    if(side=='R'){
        analogWrite(RPWM,pwm);
    }
    if(side=='L'){
        analogWrite(LPWM,pwm);
    }
}
```

```

void closeMotor(char side){
    if ( side=='R'){
        digitalWrite (RPWM,LOW);
    }
    if ( side=='L'){
        digitalWrite (LPWM,LOW);
    }
}

void setup() {
    Serial.begin(115200);
    setPWMfrequency(0x02); // timer 2 , 4 KHz
    for (int i=5; i<9; i++){
        pinMode(i,OUTPUT);
    }
    for (int i=5; i<9; i++){
        digitalWrite(i,LOW);
    }
    delay(1000);
    MotorActiveStatus('R',true);
    MotorActiveStatus('L',true);
}

//.....
//EDICION DEL BLOQUE ANALOGWRITE DEL ARDUINO TOOLBOX PARA SCILAB
// case A -> Analog
if (val==65){ //A -> Analog read
    while (Serial.available()==0) {};
    val=Serial.read();
    if (val>47 && val<67) {
        pin=val-48;
        agv=analogRead(pin);

        Serial.write((uint8_t*)&agv,2);
    }
    val=-1;
}
else if (val==87){ //W -> Analog write
    while (Serial.available()==0) {}; // Waiting char
    val=Serial.read();
    if (val>47 && val<67) { //from pin 0 to pin 19
        pin=val-48; //number of the pin
        while (Serial.available()==0) {}; // Waiting char
        val=Serial.read();

        if (val>=128){
            val = 2*(val - 128);
        }
    }
}

```

```

        val = val*1.08;

        if (val>255){ val = 255;}
        setMotor('R',val);
    }
    else{
        val = 2*(128 - val);
        val = val*1.17;
        setMotor('L',val);
    }
}
val=-1;
}

```

A.2. Código de Implementación de Algoritmo LQR en Scilab

```

function K=dlqr(Ad,Bd,Q,R)

    [n1,d1]=size(Ad);

    as=sysdiag(Q,R);
    [w,wp]=fullrf(as,1e-20);
    C1=wp(:,1:n1);
    D12=wp(:,n1+1:$);

    P=syslin('d',Ad,Bd,C1,D12);

    [K,X]=lqr(P);
    K=-K;

endfunction

```

A.3. Código Diseño de Ganancias de Realimentación a partir de la Forma Canónica Controlable en Scilab

```

G = Ad;
H = Bd;
E = Cd;
D = Dd;

[M]=cont_mat(G,H);

```



```

polesd = spec(G);
z1 = polesd(1);
z2 = polesd(2);
z3 = polesd(3);
z4 = polesd(4);
polid = (z-z1)*(z-z2)*(z-z3)*(z-z4);
cofd = coeff(polid);
a4 = cofd(1);
a3 = cofd(2);
a2 = cofd(3);
a1 = cofd(4);

[W] = [a3 a2 a1 1;
        a2 a1 1 0;
        a1 1 0 0;
        1 0 0 0]

T = M*W;

Gc = inv(T)*G*T;
Hc = inv(T)*H;
Ed = E*T;
Dd = D;

polesdes1 = [0.144; complex(0.983, -0.00765)];
polesdes2 = [0.996; complex(0.983, 0.00765)];
polesdes = [polesdes1; polesdes2];

zdes = (z-polesdes(1))*(z-polesdes(2));
zdes = zdes*(z-polesdes(3))*(z-polesdes(4));

zdes = real(zdes);
cofzdes = coeff(zdes);
alfa4 = cofzdes(1);
alfa3 = cofzdes(2);
alfa2 = cofzdes(3);
alfa1 = cofzdes(4);

kline1 = alfa4 - a4;
kline2 = alfa3 - a3;
kline3 = alfa2 - a2;
kline4 = alfa1 - a1;
Kline = [kline1 kline2 kline3 kline4];

Kr = Kline*inv(T);

```

A.4. Código de Diseño de Estimador Kalman en Scilab

```
G=2*B; H=-3; QN=2;
RN=5; NN=0;
BigR = [G*QN*G'      G*QN*H'+G*NN;
        H*QN*G'+NN*G'  H*QN*H'+RN];
[W,Wt]=fullrf(BigR);
B1=W(1:size(G,1),:);D21=W(($+1-size(C,1)):$,:);
C2=C;
P21=syslin('c',A,B1,C2,D21);
L2 = lqe(P21);
Plant = syslin('c',A,B,C,D);
ObsCont = observer(Plant,L2);
Ts = 0.005; // Periodo de Muestreo
ObservadorDiscreto = dscr(ObsCont,Ts);
```

A.5. Código de Diseño de Filtro Digital en Scilab

```
[hdot,hm,fr]=wfir('lp',33,[.05 0],'hm',[0 0]);
z=poly(0,'z');
hzdot=syslin('d',poly(hdot,'z','c')./z**33);
```