

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS INDUSTRIALES

**Diseño y construcción de un vehículo autoestabilizable
de dos ruedas con reconstrucción paramétrica en línea.**

TRABAJO RECEPCIONAL

PARA OBTENER EL TÍTULO DE LICENCIADOS EN
INGENIERÍA EN SISTEMAS ELECTRÓNICOS INDUSTRIALES

PRESENTAN

**JUAN CARLOS CASTRO RODARTE
MAYRA BERENICE PÉREZ CASTILLO**

Director del trabajo recepcional

Dr. Marcos Ángel González Olvera

Codirector

Dr. Juan Mauricio Ángeles Cervantes

Ciudad de México, septiembre 2016

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

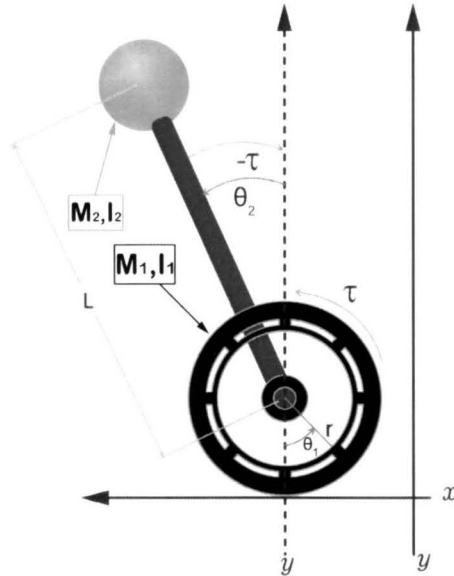
Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

UACMA TST 1050

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno



DISEÑO Y CONSTRUCCIÓN DE UN VEHÍCULO AUTOESTABILIZABLE
DE DOS RUEDAS CON RECONSTRUCCIÓN PARAMÉTRICA EN LÍNEA

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
CONTROL E INSTRUMENTACIÓN

Resumen

Este trabajo describe la construcción física de un vehículo de dos ruedas autoestabilizable. El sistema es similar a un péndulo invertido el cual está definido como un sistema fuera del equilibrio. Para lograr su estabilidad son necesarias las leyes de control que derivan del análisis del modelo físico, considerando las fuerzas de fricción y torques. El diseño de un control PD y un control LQR para el autobalance, así como la implantación de un observador que permita reconstruir ciertos parámetros en línea, son algunas de las características más importantes en este trabajo. Con ayuda de un filtro complementario se limitan las frecuencias no deseadas y con ello se evitan señales ruidosas que afectan al sistema. El uso de herramientas computacionales como MATLAB[®] permite un análisis más profundo al realizar las simulaciones correspondientes de cada etapa del sistema y así verificar su comportamiento. La implantación del algoritmo en el sistema se realiza con la plataforma ARDUINO[®].

0.1. Agradecimientos

Es de gran importancia destacar el incondicional apoyo recibido por el programa: SIMULACIÓN DE ESQUEMAS DE OBSERVACIÓN DE ESTADOS E IDENTIFICACIÓN PARAMÉTRICA DE SISTEMAS DINÁMICOS, bajo la dirección y colaboración de nuestro profesor M. en I. Fermi Vázquez Villanueva, otorgado por la Secretaría de Ciencia, Tecnología e Innovación (SECITI) con clave PI2013-25, y al Prof. Agustín Ortiz Castro por su apoyo mediante el proyecto UACM-SECITI PI2014-65; así como al programa de Proyectos Estudiantiles de Investigación ligados a la Comunidad de la Universidad Autónoma de la Ciudad de México (UACM-SLT), quienes nos brindaron los recursos necesarios para realizar el prototipo del vehículo desarrollado en este trabajo mediante el proyecto SOLAR-SEGWAY.

Al Dr. Marcos Ángel González Olvera, por la invitación a los proyectos mencionados y al estar siempre al pendiente de nosotros brindándonos su incondicional apoyo en el desarrollo de este trabajo.

Al M. en I. Luis René Sagredo Hernández, por la motivación, las ideas y el aporte hecho a este trabajo para una mejor culminación.

Al Dr. Juan Mauricio Ángeles Cervantes, quien aportó diversas ideas para el desarrollo del sistema.

A la Profesora Catalina Trevilla Román, quien siempre tuvo la total confianza en nosotros para la realización de este proyecto, al colaborar con su apoyo en el traslado de los componentes sin olvidar mencionar la maravillosa persona que es.

A mi mamá, que siempre has estado en cada meta, en cada logro y tropiezo, TE AMO, gracias por tu apoyo incondicional, tus palabras de aliento, de ánimos, porras y regaños. Eres el mejor ejemplo de vida que pueda tener. Gracias por tu dedicación y por ser la mejor mamá del mundo.

A mi papá, eres una parte fundamental en mi vida, gracias por el apoyo que me has dado y por creer en mí. Te amo.

A mis hermanos y sobrino que siempre han estado en todo momento apoyándome y acompañándome en cada paso que doy. Gracias por tolerarme y por ser mis cómplices de aventuras, los amo.

A mi gran amigo y compañero Juan Carlos que durante la carrera me brindó su apoyo incondicional, pero sobretodo sus conocimientos para llevar al éxito al equipo que formamos. Gracias por todo y por los momentos compartidos.

A la familia Machado Castillo, a ustedes quiero agradecerles por todo el apoyo brindado en cada paso que he dado, por su compañía en todo momento porque son parte fundamental de mi vida, los amo.

A mi familia, que cada uno ha contribuido con sus palabras de aliento y ánimos que siempre me han dado. A mis amigos, por sus palabras que incitaron a concluir esta etapa de mi vida.

MayBere

A mi madre Guadalupe Rodarte, ya que con su amor y cariño supo decirme las palabras mágicas para seguir estudiando, fue ella quien me impulsó en todo momento.

A mi padre Guillermo Castro, a quién admiro demasiado. Un hombre de carácter fuerte pero de corazón sabio y amoroso. Gracias por decirme cuando estaba mal, y aunque muchas veces no te supe escuchar, me has apoyado siempre hasta cuando cometía mis errores.

A mis hermanos y hermanas, que juntos aprendimos las leyes de la vida y ahora estamos más unidos que nunca.

A mis sobrinas y sobrinos, quienes siempre logran dibujar una sonrisa en mi rostro hasta en los peores días.

A la Familia Montiel Cerda quienes me abrieron las puertas de su corazón y de su hogar.

A mi compañera Mayra B. que a lo largo de la carrera siempre estuvo conmigo en todo momento, juntos aprendimos y crecimos como estudiantes pero sobre todo como grandes amigos.

A mi súper amiga Catalina Trevilla, a quien considero una grandiosa mujer. Gracias por todo tu apoyo Caty.

A los integrantes de la banda THEHEADPHONES, todos ellos son y siempre serán mis hermanos de Rock.

A mis compañeros y amigos de la UACM, con quienes viví diversas experiencias, ellos impulsaron con sus palabras mis ganas de seguir y concluir el presente trabajo.

A todos ustedes, gracias.

Juan Castro

Índice general

0.1. Agradecimientos	III
	VII
1. Introducción	2
1.1. Antecedentes	2
1.2. Descripción del problema	3
1.3. Objetivos	4
1.4. Justificación	4
1.5. Metodología	5
2. Antecedentes Teóricos	6
2.1. Modelado de sistemas	6
2.2. Sistemas de Control	7
2.3. Observación de Estados	17
2.4. Identificación paramétrica	19
2.4.1. Experimentación	20
2.4.2. Estructura del modelo	20
2.4.3. Estimación de parámetros	20
2.4.4. Validación	20
2.4.5. Lazo de identificación del sistema	20
2.5. Observadores adaptables	21
3. Planteamiento del problema	24
3.1. Descripción del sistema	24
3.2. Modelo del sistema	25
3.3. Análisis del modelo	32
4. Diseño y Construcción	37
4.1. Diseño mecánico	37
4.2. Diseño electrónico	39
4.3. Diseño del controlador	47
4.4. Diseño del observador adaptable	52
5. Resultados	54
5.1. Resultados en simulación	54
5.2. Resultados experimentales	62
5.2.1. Modelo a escala	65

5.2.2. Observador Adaptable	67
5.2.3. Prototipo final	69
6. Conclusiones	76
6.1. Conclusiones	76
6.2. Trabajo futuro	77
6.3. Resultados	77
7. Apendice	79
7.1. Archivo .m para la identificación del motor	79
7.2. Archivo .m para el modelo del sistema ISEIWAY	80
7.3. Código de la Figura 5.2: Diagrama de bloques del control PD y del observador	81
7.4. Código de la Figura 5.12: Diagrama de bloques del control LQR en Simulink	82
7.5. Código de la Figura 5.15: Diagrama de bloques del control LQR con el Observador Adaptable	83
7.6. Código del ISEIWAY a escala con observador	83
7.7. Código del ISEWAY tamaño real	98

Índice de figuras

1.1. Modelos Segway®	2
2.1. Diagrama de bloques de un sistema	7
2.2. Diagrama de bloques de un sistema de control	7
2.3. Diagrama de bloques de configuración de lazo abierto	8
2.4. Diagrama de bloques de configuración de lazo cerrado	9
2.5. Diagrama de bloques de un control P	10
2.6. Respuesta de un control P para un motor DC ante una entrada escalón	10
2.7. Diagrama de bloques de un control PI	11
2.8. Respuesta de un control PI para un motor DC ante una entrada escalón	11
2.9. Diagrama de bloques de un control PD	12
2.10. Respuesta de un control PD para un motor DC ante una entrada escalón	12
2.11. Diagrama de bloques de un control PID	13
2.12. Respuesta de un control PID para un motor DC ante una entrada escalón	13
2.13. Respuesta en tiempo continuo con parámetros de diseño	15
2.14. Diagrama de bloques de un sistema en lazo cerrado	15
2.15. Configuración de polos y ceros	17
2.16. Diagrama de bloques de un observador	19
2.17. Lazo de identificación del sistema	21
3.1. Prototipo ISEIWAY	24
3.2. Diagrama eléctrico de un motor DC	25
3.3. Suma de torques	26
3.4. Diagrama de fuerzas péndulo invertido	27
3.5. Diagrama de bloques para la adquisición de parámetros de un motor DC.	32
3.6. Respuesta transitoria de un motor DC	33
3.7. Diagrama de bloques del sistema ISEIWAY en SIMULINK®	34
3.8. Respuesta inestable de los estados del sistema ISEIWAY	35
4.1. Componentes principales del sistema ISEIWAY	37
4.2. Fabricación ISEIWAY	38
4.3. Ajuste de la rueda al motor	38
4.4. Placa Arduino Due	40
4.5. Estructura del filtro complementario	42

4.6. Señal del IMU con filtro complementario usando SerialChart	44
4.7. IMU MPU6050	45
4.8. MOTOR DC 24V 450 WATTS	45
4.9. DRIVER VNH5019	46
4.10. BATERÍA 12V 32Ah	47
4.11. Polos en lazo abierto en el dominio z	49
4.12. Polos y ceros del sistema representados en el plano z	49
4.13. Diseño del cero del control PD	50
4.14. Ubicación de polos y ceros del sistema ISEIWAY con control PD	51
4.15. Respuesta del control PD del sistema ISEIWAY en simulación	52
5.1. Posición del sistema	54
5.2. Diagrama de bloques del control PD y el observador	55
5.3. Desempeño del control PD para la posición del sistema	55
5.4. Desempeño del observador y del control PD para la velocidad de las ruedas	56
5.5. Desempeño del observador y del control PD para la posición del sistema	56
5.6. Desempeño del observador y del control PD para la velocidad del sistema	56
5.7. Desempeño del observador y del control PD para el parámetro β_1	57
5.8. Desempeño del observador y del control PD para el parámetro L_1	57
5.9. Desempeño del observador y del control PD para el parámetro L_2	57
5.10. Desempeño del observador y del control PD para el parámetro β_2	58
5.11. Desempeño del observador y del control PD para el parámetro L_3	58
5.12. Diagrama de bloques del control LQR en Simulink	58
5.13. Salida de la posición con control LQR	59
5.14. Estados del sistema ISEIWAY con control LQR	59
5.15. Diagrama de bloques del control LQR con el Observador Adaptable	60
5.16. Desempeño del observador adaptable	60
5.17. Análisis entre el sistema y el observador para la velocidad de la rueda.	61
5.18. Análisis entre el sistema y el observador para la posición del vehículo.	61
5.19. Análisis entre el sistema y el observador para la velocidad del Vehículo.	61
5.20. Acoplamiento entre el motor DC del sistema ISEWAY y el motor DC para pruebas	62
5.21. Parametrización del motor de DC de 24V	63
5.22. Gráfica de constante de torque	64
5.23. Acoplamiento motor-rueda	65
5.24. Acoplamiento motor-base	66
5.25. Arduino due y driver POLOLU® en el ISEIWAY-MINI	66
5.26. Sensor IMU MPU6050 sobre la base del ISEIWAY-MINI	66
5.27. Batería de 12V	67
5.28. ISEIWAY mini, diseño final	67
5.29. Desempeño del observador adaptable para los parámetros del prototipo ISEIWAY mini	68

5.30. Desempeño del observador adaptable para la velocidad de las ruedas del prototipo ISEIWAY mini	68
5.31. Desempeño del observador adaptable para la posición del prototipo ISEIWAY mini	69
5.32. Comparación del desempeño entre el observador adaptable y la respuesta del sistema para la posición del prototipo ISEIWAY mini	69
5.33. Sistema ISEIWAY final	70
5.34. Acoplamiento motor-rueda	70
5.35. Acoplamiento sensor IMU	71
5.36. Acoplamiento de la rueda y motor a la base	71
5.37. Vista superior de componentes	72
5.38. Vista superior de driver y conexión a la tarjeta Arduino Due	72
5.39. Acoplamiento de ventilador	73
5.40. Integrantes del sistema ISEIWAY	73
5.41. Vista superior ISEIWAY	74
5.42. Funcionamiento del sistema ISEIWAY	74
5.43. Funcionamiento del sistema ISEIWAY	75

1

Introducción

1.1. Antecedentes

Hoy en día, las técnicas de identificación paramétrica de sistemas han logrado tomar un importante papel en muchas de las áreas de ingeniería, en donde es necesario un modelo preciso y bien definido para fines de análisis, simulación, control, diseño y predicción. Algunas técnicas de control actuales demandan modelos matemáticos cada vez más exactos para obtener un resultado óptimo y funcional. La identificación de sistemas dinámicos es de gran importancia ya que es la herramienta capaz de proporcionar los métodos necesarios para obtener de manera más sencilla los modelos matemáticos requeridos [5].

Existen diversos vehículos autoestabilizables como por ejemplo el sistema Segway[®], como se muestra en la Figura 1.1, fue el primer dispositivo fabricado para el transporte personal autoestabilizable. El motor es eléctrico, no contaminante y silencioso, alcanzando aproximadamente los 20 km/h [1].

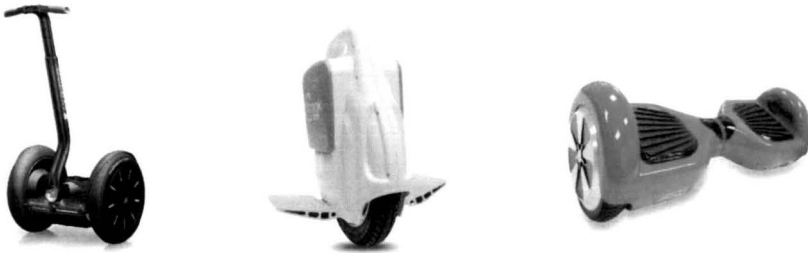


Figura 1.1: Modelos Segway[®]

En particular el sistema realizado en este trabajo, hace énfasis en el diseño y la obtención de un modelo matemático que permite que el sistema físico mantenga su equilibrio proporcionando a los usuarios el traslado de un punto a otro. Al ser el vehículo eléctrico, aporta una característica importante ya que no es dañino para el medio ambiente y por consiguiente para todos.

El propósito de crear un controlador digital para tal sistema es debido a que al usar un microcontrolador, se obtienen grandes ventajas como son; menor ruido,

bajos costos que van desde los \$200.00, un desempeño óptimo con bajo consumo de energía. La capacidad en la toma de decisiones y flexibilidad en los programas de control es otra ventaja más de los sistemas digitales. Existe una gran variedad de computadoras que permiten operar las diversas señales de interés, logrando sistemas más reducidos y eficaces.

El desarrollo de este proyecto describe el diseño y construcción de un vehículo eléctrico de transporte personal, ligero y giroscópico de dos ruedas. El equilibrio del mismo se llevará a cabo por medio de dispositivos electrónicos, etapas de programación, diversos elementos mecánicos y materiales como llantas, aluminio, tornillos y acrílico entre otros.

El vehículo **ISEIWAY** consta de dos motores situados en una base, que a su vez hacen girar mediante una cadena a dos ruedas paralelas, las cuales tienen la finalidad de mantener la base en forma horizontal. El usuario se posicionará sobre la misma y, tomando como referencia un manubrio, podrá inclinarse de tal forma que se moverá hacia la dirección deseada: atrás o adelante, adaptando dos interruptores (uno en cada extremo del manubrio) se permitirá el giro hacia la izquierda o derecha, logrando así que el usuario pueda moverse en cualquier dirección. Los motores actúan dependiendo de lo que el usuario elija y se controlarán en consecuencia. Será de fácil operación y se alimentará con baterías recargables.

Por otra parte, se diseña un observador adaptable lineal con la finalidad de conseguir estimar los estados y parámetros no medibles en el sistema original. El observador permitirá reconstruir tres estados: velocidad de las ruedas, posición del péndulo y velocidad del péndulo. Para este caso también se estima la masa de la persona que suba al vehículo, debido a que dicho parámetro está involucrado en el sistema, logrando así una mejor aproximación en el modelo matemático del mismo. Las diversas simulaciones del sistema permiten un mejor análisis de los estados estimados, al converger a un punto deseado siendo este el objetivo del observador adaptable. Se implantó dicho observador en el modelo a escala y se corroboró su desempeño.

En el Apéndice A, se muestra el desarrollo de un algoritmo en MATLAB[®], el cual permitió la identificación paramétrica de los motores DC usados en el sistema, logrando identificar parámetros como la inercia de los motores, la constante eléctrica y la constante de fricción viscosa.

1.2. Descripción del problema

Se desea crear un vehículo autoestabilizable de dos ruedas partiendo del análisis del sistema real hasta obtener un modelo matemático linealizado, para llevarlo a la implementación física y tomando como punto de partida que el sistema se encuentra fuera del equilibrio. Al ser inestable el punto de operación del sistema, se requiere la construcción de un controlador que permita estabilizarlo y sea capaz de mantenerlo

estable al ser maniobrado por diferentes personas y otras perturbaciones. Se diseñara un filtro que garantice sólo las señales de interés para su efectiva manipulación. Por otro lado se planea la fabricación de la estructura en las dimensiones necesarias que proporcionen la funcionalidad del sistema. La adaptación de los motores a las ruedas y la sujeción de las mismas a la estructura. Es importante implementar un prototipo a escala para realizar las pruebas iniciales antes de pasar al prototipo de tamaño real.

1.3. Objetivos

- a) Analizar el sistema para obtener el mejor modelo matemático y posteriormente linealizarlo respecto a un punto de equilibrio.
- b) Implementar en el sistema el control que permita el equilibrio del vehículo, aún cuando el usuario haya subido al **ISEIWAY**.
- c) Desarrollar el algoritmo que proporcione dicha estabilidad y simularlo en una computadora para determinar posibles errores y corregirlos.
- d) Armar la base con un diseño ergonómico tal que sea cómodo para el usuario.
- e) Acondicionar las conexiones para la recarga de baterías de forma sencilla.
- f) Realizar las pruebas reales en el vehículo para determinar su eficiencia.
- g) Al finalizar el proyecto, se logrará un vehículo ligero, rápido, seguro, de fácil manejo y que requiera un mantenimiento casi nulo.
- h) Reducir gastos en su producción para obtener un diseño económico y funcional.
- i) Apoyar no sólo al personal de la institución sino también a todos los elementos de seguridad o usuarios con capacidades especiales en la Ciudad de México.
- j) Al terminar el sistema **ISEIWAY**, se alentara a los estudiantes a desarrollar sus propios proyectos para que en conjunto con la Universidad contribuyan a la solución de problemas que existen tanto en la institución, como en la sociedad en general.

1.4. Justificación

El desarrollo del sistema **ISEIWAY** permite aplicar los diversos conocimientos obtenidos a lo largo de la carrera Ingeniería en Sistemas Electrónicos Industriales, que actualmente se imparte en esta casa de estudios (UACM-SLT), con la finalidad de lograr un prototipo funcional y con un mantenimiento mínimo. El proyecto fue pensado como un gran apoyo para los elementos de seguridad de la escuela, los policías podrán hacer rondines en forma más rápida y sin desgaste físico al trasladarse de un lugar a otro en poco tiempo y sin contaminar al medio ambiente. El sistema será silencioso, seguro, recargable, rápido y de fácil operación. La idea nace

con este propósito, pero se puede aplicar a diversas áreas tanto de la Universidad como de la Ciudad de México, esto va desde robots en los tamaños requeridos que permitan explorar diversas superficies no aptas para el ser humano, trasladar a personas adultas que no pueden caminar distancias prolongadas, permitiendo también transportar carga extra junto con el conductor. Cabe destacar que dicho proyecto se llevaría a cabo con un costo por debajo de los que se encuentran en el mercado (aproximadamente 22,000.00 pesos) y así se impulsaría el aspecto tecnológico que se desarrolla en la UACM.

La realización de los diversos controles presentan una gran variedad de soluciones para el autobalance del sistema, el control PD, el control PID y el control LQR, son métodos que contrastados en un mismo sistema, proporcionan una mejor selección del controlador para garantizar el desempeño óptimo del vehículo **ISEIWAY**.

La identificación paramétrica, así como la observación de estados, juega un papel importante en la realización de este trabajo, ya que es fundamental conocer los parámetros de los motores DC para obtener el modelo matemático que mejor describa la dinámica del sistema. Por parte de la observación paramétrica se desarrolló un observador adaptable lineal para obtener cierta información como la adquisición en línea de la inercia del sistema.

Este dispositivo da pauta hacia el desarrollo de vehículos con las características especiales para personas con alguna capacidad diferente, y así ser de gran apoyo en su forma de vida diaria.

1.5. Metodología

El desarrollo del vehículo **ISEIWAY** describe un análisis profundo del sistema similar al péndulo invertido, en donde el reto más grande es la implementación de un control que garantice el equilibrio del mismo, sin dejar de lado economizar en gastos de su fabricación y mantenimiento, sin sacrificar la robustez y fiabilidad del controlador. En el Capítulo 2 se abordan los temas más básicos de modelado de sistemas para la obtención de un modelo matemático eficiente. En el Capítulo 3 se presentan las características del sistema **ISEIWAY**, su modelado y análisis. En el Capítulo 4 se describe cómo se realizó la construcción del vehículo, el acondicionamiento electrónico y mecánico, así como el diseño de los controladores que garantizan la estabilidad del sistema. En el Capítulo 5 se hace énfasis en las diversas simulaciones del sistema que tuvieron que realizarse para lograr el objetivo principal, el autobalance del vehículo. En el Capítulo 6 se concluyen los procesos a los cuales se enfrentó, las mejoras, los resultados y la innovación a futuro.

2

Antecedentes Teóricos

2.1. Modelado de sistemas

Para llevar a cabo el análisis de un sistema, en principio se debe conocer la relación entre sí de sus variables. En general, se denomina modelo de un sistema a la relación de dichas variables a partir de la observación, es decir, los modelos se construyen a partir de los datos observados y pueden ser de diferente tipo [2]:

- Modelos mentales (no involucran matemáticas formales)
- Matemáticos
- Gráficos (tablas o gráficas)
- Por software (programación, simulación por computadora)

Los modelos mentales hacen referencia a la serie de procesos que se llevan a cabo sin un previo análisis y que resultan descriptivos. Ejemplo de ello puede ser el presionar las teclas de una computadora, en ello se define la fuerza y los movimientos para escribir la palabra deseada, realizados sin matemáticas formales. *Los modelos matemáticos* son aplicados generalmente en el área de las ciencias y la ingeniería, debido al grado de complejidad para analizar un sistema, y cuya complejidad depende de la aplicación requerida. Estos modelos son obtenidos a partir del conocimiento de las características físicas del sistema. *Los modelos gráficos* son representados por tablas numéricas y gráficos que describen el comportamiento de un sistema. *Los modelos por software* son una serie de programas que resuelven un sistema por medio de la simulación numérica por computadora.

Con respecto a *los modelos matemáticos* los cuales serán empleados en este trabajo, existen modelos lineales y no lineales, por lo general antes de comenzar el desarrollo de un modelo no lineal, en principio se debe considerar un modelo lineal. Si el desempeño como modelo lineal no es suficiente y satisfactorio, entonces esto nos dice que el comportamiento del sistema es no lineal. El análisis que se debe realizar para un modelo no lineal será complejo, ya que se requiere elegir una estructura del modelo con un cierto número de parámetros y el diseño de un algoritmo que estime dichos parámetros. Una vez definido lo anterior se realiza la simulación numérica que permitirá saber si el modelo propuesto, así como el algoritmo son suficientemente exactos tomando en cuenta las condiciones involucradas. La relación entre un sistema y un modelo está dada por las características de ciertos aspectos del sistema físico

con su descripción matemática, nunca son exactos uno con el otro, el modelo sólo es una representación matemática del sistema y es construido a partir de datos observados en el sistema. La Figura 2.1 describe la representación de un sistema por medio de un diagrama de bloques con entradas (u, w, v) y salidas (y).

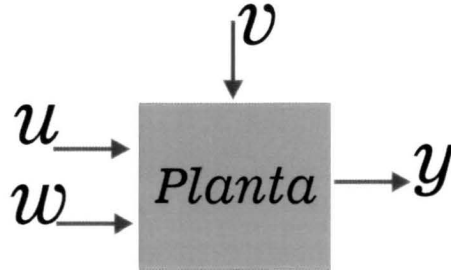


Figura 2.1: Diagrama de bloques de un sistema

2.2. Sistemas de Control

Lazo abierto/Lazo cerrado

Los sistemas de control forman parte de nuestra vida diaria y están presentes en cualquier entorno, sin embargo la mayoría de estos sistemas requieren de un esquema de control específico para comportarse de cierta manera hasta lograr su manipulación en el mejor de los casos [3].

Los sistemas de control están formados por etapas que interactúan entre sí para lograr un fin. Los principales objetivos de control son: estabilización, regulación, seguimiento, desempeño y robustez. Algunos ejemplos de ellos puede ser, el seguimiento una trayectoria deseada, colocar un objeto con mucha precisión en algún punto definido, mantener el equilibrio de un robot péndulo invertido entre otros. Estos son sistemas controlados que responden a estímulos externos y/o entradas al mismo. La Figura 2.2 representa un sistema de control con entrada (u) y salida(y).

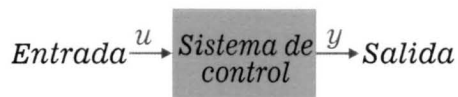


Figura 2.2: Diagrama de bloques de un sistema de control

En la Figura 2.2 se presenta el diagrama de bloques del proceso que realiza un sistema de control, en donde la entrada son todas las señales de sensores que contenga el sistema, las cuales indican el estado actual del mismo.

Antes de analizar los sistemas de control [3], deben definirse ciertos términos básicos.

Variable controlada y variable manipulada. La variable controlada es la cantidad o condición que se mide y controla. La variable manipulada es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada. Por lo común, la variable controlada es la salida (el resultado) del sistema. Controlar significa medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar una desviación del valor medido a partir de un valor deseado.

En el estudio de la ingeniería es importante definir términos adicionales que resultan necesarios para describir los sistemas de control.

Planta, es un conjunto de las partes de un sistema que funcionan juntas, el propósito de la misma es ejecutar una operación particular.

Hay dos configuraciones de sistemas de control; *lazo abierto* y *lazo cerrado*. El sistema de control en lazo abierto es aquel sistema en el que la salida no se compara con la entrada, una característica principal es que la entrada se emplea en el sistema para regular su comportamiento, otra característica de este sistema de control está vinculada con la no retroalimentación y debido a ello no puede compensar las perturbaciones que se presentan en el sistema. Ver Figura 2.3.



Figura 2.3: Diagrama de bloques de configuración de lazo abierto

Los sistemas de control en lazo cerrado Figura 2.4, son aquellos en los que se mide la señal de salida y ésta a su vez retroalimenta al sistema al compararse con la señal de entrada, haciendo que el control compense el error que existe entre ambas señales. En los sistemas reales se tienen perturbaciones o ruido, para tener un mejor modelado del sistema, estas señales exógenas se emulan al agregar un bloque (perturbaciones). Para esta configuración las perturbaciones son consideradas como una entrada extra al sistema por lo tanto el controlador añade una entrada más al mismo corrigiendo el desequilibrio que pueden causar las perturbaciones.

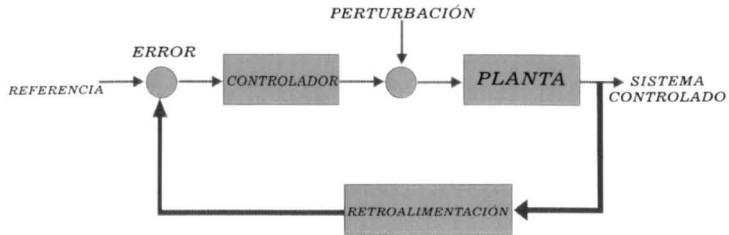


Figura 2.4: Diagrama de bloques de configuración de lazo cerrado

Clasificación de los controladores

Es común emplear las especificaciones de diseño para describir qué debe hacer el sistema y cómo hacerlo. Estas especificaciones son únicas para cada aplicación individual. La elección del controlador depende de la naturaleza del sistema y las condiciones de operación a las que se someterá tomando en cuenta costos, seguridad, disponibilidad, confiabilidad, precisión, peso y tamaño. Existe una combinación de controladores los cuales son:

1. **Control Proporcional (P)**
2. **Control Integral (PI)**
3. **Control Derivativo (PD)**
4. **Control Proporcional-Integral-Derivativo (PID)**

Las simulaciones para ejemplificar los desempeños de cada controlador de esta sección, fueron realizados con ayuda de MATLAB® [12] para determinar la respuesta de un motor DC frente a una entrada escalón.

1. Control Proporcional (P)

El control proporcional es el más sencillo de todos, ya que consta de una ganancia proporcional ajustada $u(t) = K_p e(t)$. Donde la entrada es proporcional al error existente entre la referencia y la salida. Su principal objetivo es aplicar esa ganancia en un tiempo determinado para la salida de un sistema hasta que éste llegue a un estado requerido. La mejor forma de ejemplificar y comprender un control proporcional es la velocidad de un automóvil, al aumentar o disminuir su velocidad se hace presente el control con un comportamiento suave para lograr la velocidad deseada, aquí se compara la velocidad real y la velocidad requerida, haciendo que el error cada vez sea menor y la potencia disminuya ligeramente en proporción al error. Su estructura y su respuesta se pueden observar en la Figura 2.5. y la Figura 2.6.

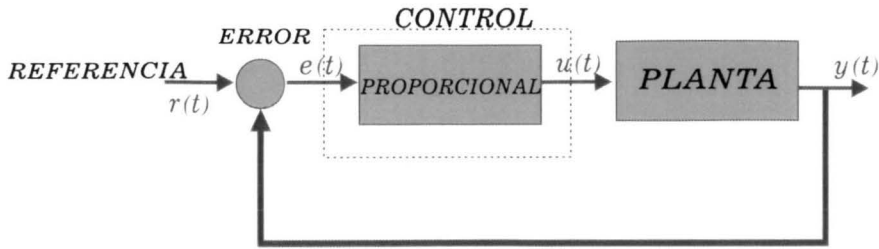


Figura 2.5: Diagrama de bloques de un control P

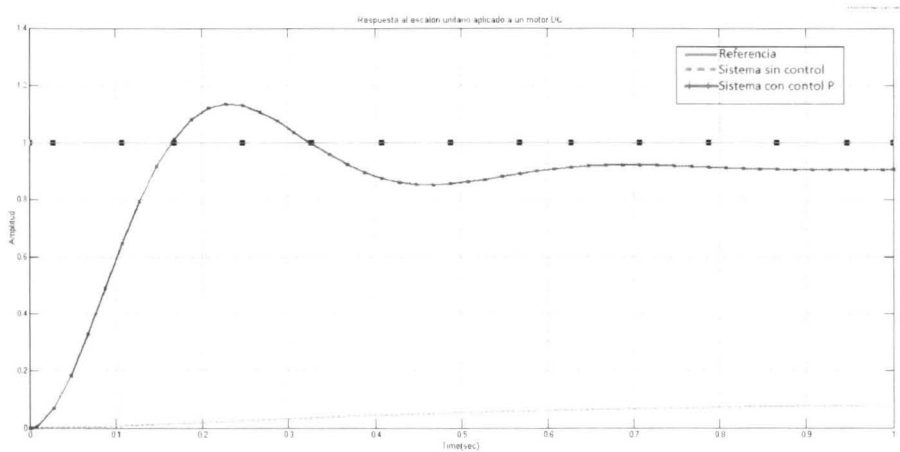


Figura 2.6: Respuesta de un control P para un motor DC ante una entrada escalón

2. Control Integral (PI)

El control Integral tiene como propósito disminuir y eliminar el error en estado estacionario (es una medida de la exactitud de un sistema de control para seguir una entrada dada) provocado por el modo proporcional. El control integral actúa cuando hay una desviación entre la variable y el punto de referencia, integrando esta desviación en el tiempo y sumándola a la acción proporcional. El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un período determinado; Luego es multiplicado por una constante K_i . Posteriormente, la respuesta integral es adicionada al modo Proporcional para formar el control P + I con el propósito de obtener una respuesta estable del sistema sin error estacionario. Su estructura y su comportamiento se pueden ver en la Figura 2.7 y la Figura 2.8. Su descripción matemática es:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt$$

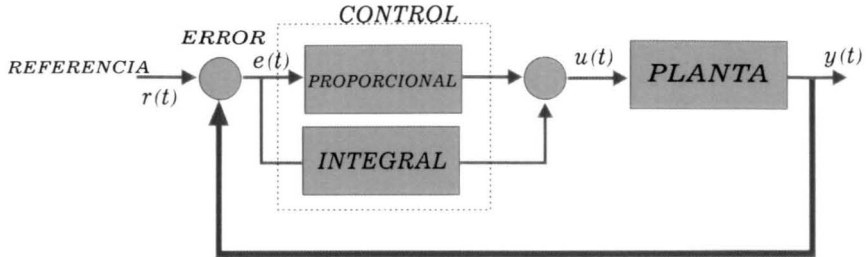


Figura 2.7: Diagrama de bloques de un control PI

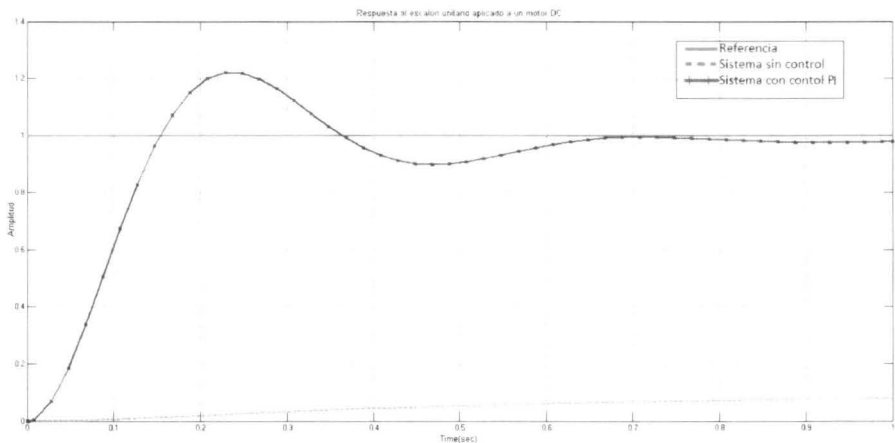


Figura 2.8: Respuesta de un control PI para un motor DC ante una entrada escalón

3. Control Derivativo (PD)

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error. El error es la desviación existente entre el punto de medida y el valor de referencia. La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente. Se deriva numéricamente con respecto al tiempo y se multiplica por una constante K_d , luego se suma a la señal anterior para formar el control (P+D). Se observa el diagrama de bloques y su comportamiento en la Figura 2.9 y la Figura 2.10. Su descripción matemática es la siguiente.

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

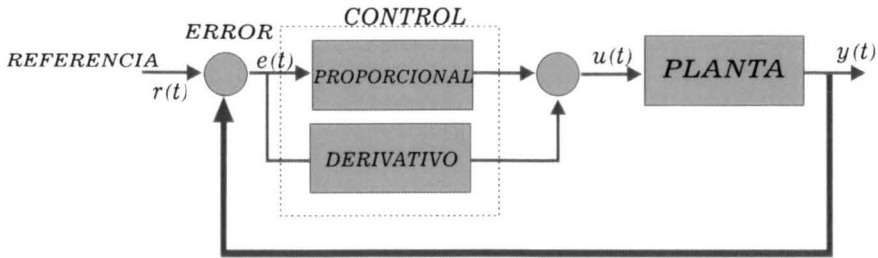


Figura 2.9: Diagrama de bloques de un control PD

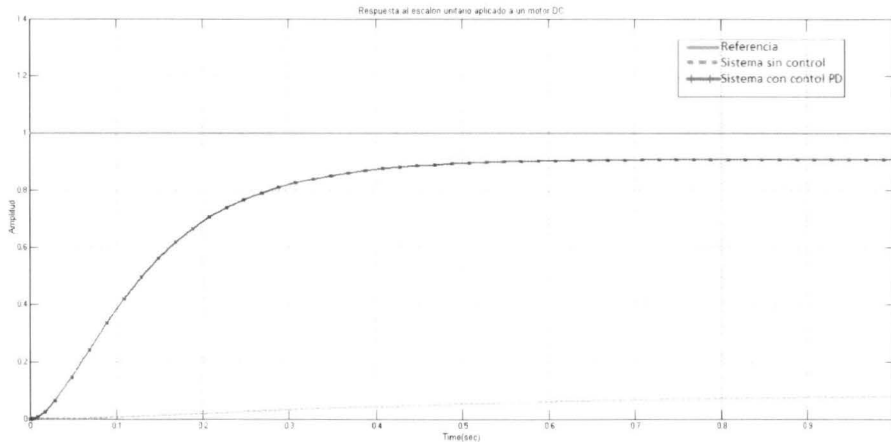


Figura 2.10: Respuesta de un control PD para un motor DC ante una entrada escalón

4. Control Proporcional-Integral-Derivativo (PID)

El control PID consiste de tres parámetros diferentes: el proporcional, el integral, y el derivativo. El valor Proporcional depende del error actual. El Integral depende de los errores pasados y el Derivativo es una predicción de los errores futuros. La suma de estas tres acciones es usada para ajustar al proceso por medio de un elemento de control como por ejemplo la posición de un brazo robótico o la potencia suministrada en un calentador.

Cuando no se tiene conocimiento del proceso, históricamente se ha considerado que el controlador PID es el controlador más adecuado. Ajustando estas tres variables en el algoritmo de control PID, el controlador puede proveer una acción de

control diseñado para los requerimientos del proceso en específico. De igual forma que los controladores anteriores, se puede observar su diagrama de bloques y su comportamiento en la Figura 2.10 y la Figura 2.11. Su descripción matemática es:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int \epsilon(t) dt + K_d T_d \frac{de(t)}{dt}$$

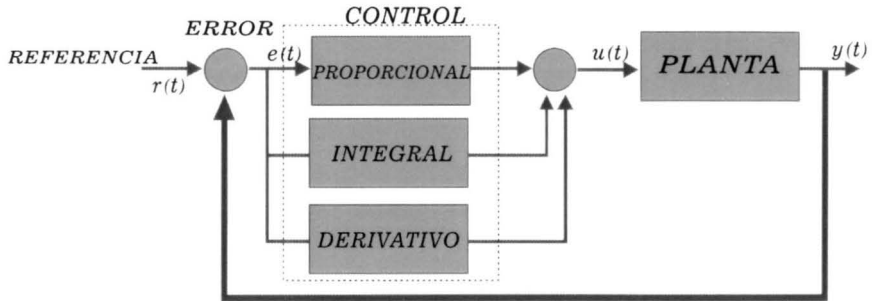


Figura 2.11: Diagrama de bloques de un control PID

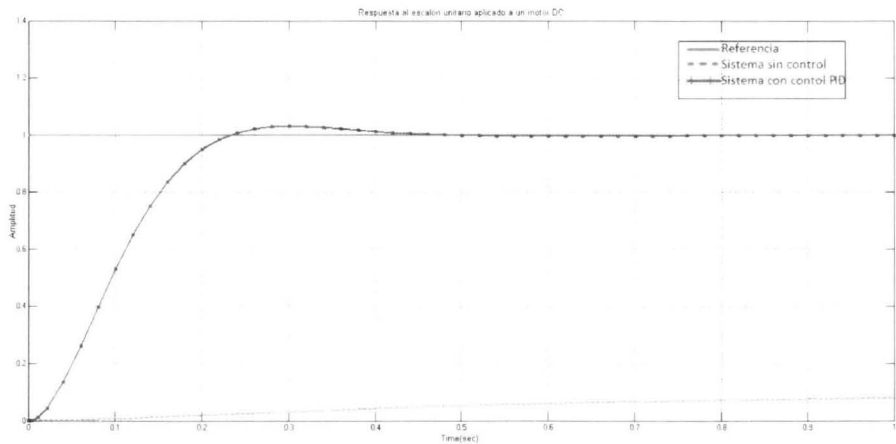


Figura 2.12: Respuesta de un control PID para un motor DC ante una entrada escalón

Lugar Geométrico de las Raíces (LGR)

Una función de transferencia está compuesta por la Transformada de Laplace de la EDO (Ecuación Diferencial Ordinaria) salida/entrada que describe el comportamiento del sistema y las propiedades intrínsecas del mismo, las raíces en el numerador son llamadas ceros, en la parte del denominador se encuentra el polinomio característico del sistema y es ahí donde las raíces responden al nombre de polos de la función de transferencia. Los ceros determinan la respuesta transitoria del sistema, por otro lado los polos nos describen la estabilidad del sistema y la rapidez de convergencia.

Cuando se trata de sistemas de control es sumamente importante conocer la ubicación de las raíces de la ecuación característica en lazo cerrado, lo cual puede conocerse utilizando un método sistemático y sencillo que muestra el movimiento de dichas raíces cuando se modifica un parámetro de la ecuación. Dicho método permite elaborar lo que se conoce como el lugar geométrico de las raíces (LGR), que nos es otra cosa que las soluciones de la ecuación característica en lazo cerrado cuando se varía un parámetro. Si el sistema tiene una ganancia de lazo variable, la ubicación de los polos en lazo cerrado depende del valor de la ganancia de lazo elegida.

La idea básica del método del LGR es que la función de transferencia en lazo cerrado ($1 + G(s)H(s) = 0$) debe satisfacer la ecuación característica del sistema. Desde el punto de vista del diseño, un simple ajuste de la ganancia en algunos sistemas mueve los polos en lazo cerrado a las posiciones deseadas, aquí el problema de diseño se centra en la selección de un valor de ganancia adecuada, si el ajuste de la ganancia no presenta un resultado conveniente, es necesario agregar un compensador.

El objetivo principal del método *LGR* [6], es ajustar la ubicación de polos y ceros en lazo cerrado del sistema; como se muestra en la Figura 2.15. El reto es determinar dicha ganancia para ubicar los polos en el lugar deseado. El diseño del LGR en tiempo discreto, se basa en el método LGR en tiempo continuo de la respuesta deseada obtenida por los parámetros tales como: tiempo de levantamiento (tr), sobre paso ($\%Os$), tiempo de pico (tp), entre otros. Ver Figura 2.13.

Donde: tr es el tiempo requerido para que la respuesta pase del 10 al 90%, del 5 al 95% o del 0 al 100% de su valor final, tp es el tiempo requerido para que la respuesta alcance el primer pico del sobrepaso, $\%Os$ es el valor pico máximo de la curva de respuesta, medido a partir de la unidad.

La razón por la cual el método del LGR puede extenderse a sistemas en tiempo discreto es por que su ecuación característica tiene la misma forma que la del sistema en tiempo continuo en el plano complejo. Otra consideración es que suele ser más sencillo realizar el análisis en tiempo discreto de un sistema [6].

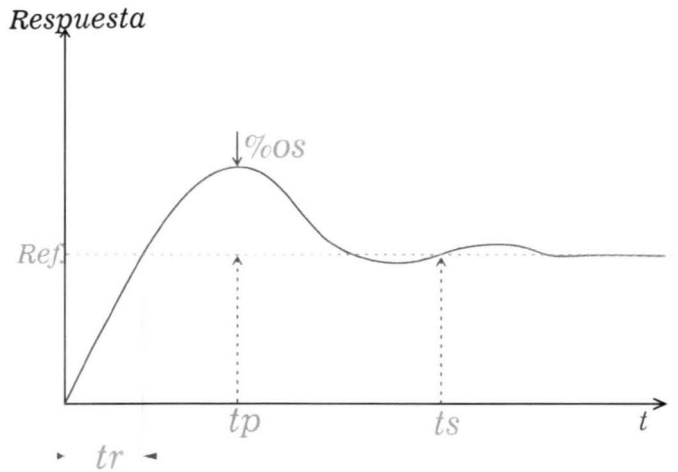


Figura 2.13: Respuesta en tiempo continuo con parámetros de diseño

Con el fin de comprender mejor el método del LGR se analiza el siguiente diagrama de bloques en lazo cerrado. Ver Figura 2.14.

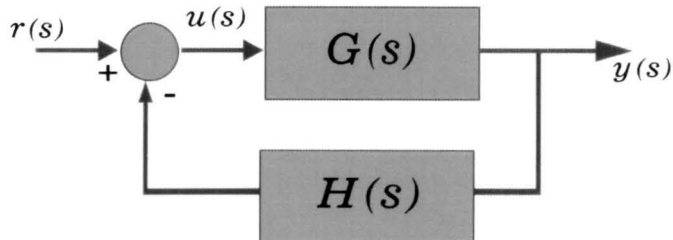


Figura 2.14: Diagrama de bloques de un sistema en lazo cerrado

La función de transferencia en lazo cerrado es:

$$\frac{y(s)}{r(s)} = \frac{G(s)}{1 + G(s)H(s)}. \quad (2.1)$$

Y la ecuación característica está dada por: $1 + G(s)H(s) = 0$

Analizando el movimiento de las raíces de $H(s)$ debido al efecto de los ceros y polos de $G(s)$ para lograr que los polos logren la respuesta deseada, donde los polos dominantes serán:

$$s_{1,2} = -\xi\omega_n \pm j\omega_d. \quad (2.2)$$

Donde ξ es el factor de amortiguamiento, ω_n la frecuencia natural y ω_d la frecuencia de amortiguamiento.

Los cuales corresponden con los polos en tiempo discreto con un tiempo de muestreo (T_s) de la forma:

$$z_{1,2} = e^{T_s(s_{1,2})} = e^{T_s(-\xi\omega_n)} e^{\pm T_s j\omega_d}. \quad (2.3)$$

Es importante mencionar que para lograr el diseño del método LGR se deben cumplir dos condiciones:

Los valores de s que cumplen tanto las condiciones de ángulo como las de magnitud son las raíces de la ecuación característica, o los polos en lazo cerrado.

El lugar geométrico de las raíces es una gráfica de los polos del plano complejo que sólo satisfacen la condición de ángulo. Si la función de transferencia de lazo cerrado tiene t polos más que ceros, entonces el LGR tiene t asíntotas equiespaciadas, formando entre ellas un ángulo de fase. Las raíces de la ecuación característica (los polos en lazo cerrado) que corresponden a un valor específico de la ganancia se determinan a partir de la condición de magnitud y esta se obtiene de:

Condición de magnitud

$$1 = |C(z)G(z)|. \quad (2.4)$$

Condición de ángulo o fase

$$\angle C(z)G(z) = \pm 180^\circ(2N + 1); N = 0, 1, \dots \quad (2.5)$$

El término de diseño adicional en tiempo discreto $\frac{\omega_s}{\omega_d}$ es el número de muestras por oscilación y se relaciona con la frecuencia f_s y el tiempo de muestreo T_s .

$$\frac{\omega_s}{\omega_d}, \omega_s = \frac{2\pi}{T_s} = 2\pi f_s \quad (2.6)$$

Donde ω_s es la frecuencia de oscilación, ω_d es la frecuencia natural de amortiguamiento, T_s es el periodo de muestreo y f_s es la frecuencia de muestreo.

Para lograr el comportamiento deseado se diseña a $C(z)$ con una ganancia K , un cero C_n y un polo P_n , es decir:

$$C(z) = K \frac{z - C_n}{z - P_n}. \quad (2.7)$$

La condición del LGR es básicamente un problema gráfico. La construcción gráfica del lugar geométrico de las raíces se basa en el conocimiento de los polos y ceros de la función de $G(z)C(z)$.

Una vez que el LGR se ha construido, los valores de K a lo largo del lugar geométrico se pueden determinar de la siguiente manera. La Figura 2.15 representa la construcción gráfica del LGR.

$$|K| = \frac{\prod_{j=1}^n |s + P_j|}{\prod_{k=1}^m |s + z_k|}. \quad (2.8)$$

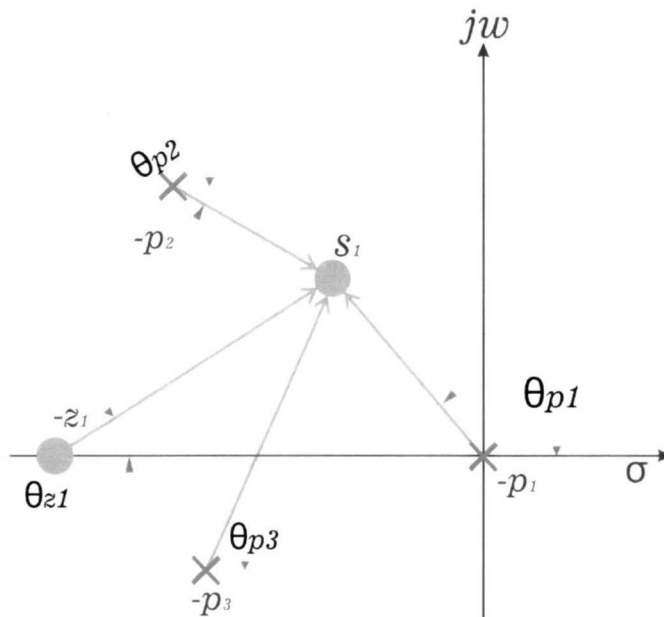


Figura 2.15: Configuración de polos y ceros

2.3. Observación de Estados

Un observador de estados busca reconstruir los valores ocultos de los estados mediante \hat{x} (estados estimados). Consideremos que se desean estimar los estados x de un sistema lineal [10].

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.9}$$

Siendo las matrices A, B y C conocidas, se propone la siguiente estructura para el observador

$$\dot{\hat{x}} = A_o \hat{x} + Ly + z \tag{2.10}$$

Las matrices A_o y L deben ser diseñadas para cumplir el objetivo de forzar la convergencia de los estados del observador a los del sistema. Por otra parte, z es una señal a determinar, si bien aún no es conocida es razonable pensar que dependa de la excitación u (vector de entrada) del sistema a observar.

La dinámica del error definido por la diferencia entre los estados del sistema y los estados del observador resulta de la diferencia entre la ecuación (2.11) y la ecuación (2.12).

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ax + Bu - A_o \hat{x} - Ly - z \tag{2.11}$$

$$\dot{e} = (A - LC)x - A_o \hat{x} + Bu - z \tag{2.12}$$

Para asegurar que el error $e = x - \hat{x}$ converja a cero, más allá de la excitación u del sistema, de su salida y y del valor inicial del error $e(0)$, la ecuación (2.12) puede reducirse a:

$$\dot{e} = \tilde{A}e \tag{2.13}$$

donde los valores de la matriz \tilde{A} deben pertenecer al semiplano izquierdo. Esta reducción es posible si

$$A_o = A - LC = \tilde{A}, \tag{2.14}$$

$$z = Bu \tag{2.15}$$

Por tanto el diseño del observador se reduce a encontrar una matriz L que garantice que los valores de dicha matriz cumplan lo siguiente:

- Para asegurar la estabilidad del observador, los polos deben ubicarse en el semiplano izquierdo .
- Los valores de la matriz L deben estar más hacia la izquierda que los polos del sistema, con ello se garantiza que la dinámica del error $\dot{e} = A_o e$ sea más rápida que la del sistema.

El diagrama de bloques para representar un observador esta constituido por el sistema a observar, esto se puede ver en la Figura 2.16.

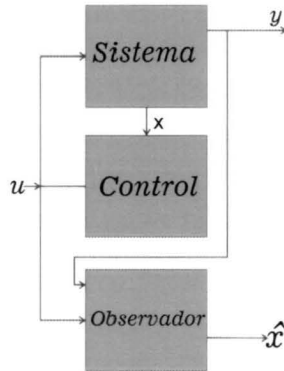


Figura 2.16: Diagrama de bloques de un observador

2.4. Identificación paramétrica

El proceso de identificación para la construcción del modelo involucra una serie de aspectos a tomar en cuenta como son [2]:

- Experimentación
- Estructura del modelo
- Estimación de parámetros
- Validación.

Una característica importante para realizar la construcción de modelos implica lo siguiente:

- Los datos
- Un conjunto de modelos candidatos
- Un algoritmo que permita que los modelos candidatos puedan ser parametrizados y evaluados con los datos proporcionados.

2.4.1. Experimentación

Los procesos experimentales deben ser llevados a cabo con las mismas consideraciones de principio a fin, con la finalidad de obtener datos lo más eficiente posible y realizando las mediciones con instrumentos fiables. Otra consideración importante es que al aplicar la entrada al sistema, se cuente con las características necesarias para generarla. Una vez aplicada la señal de entrada, las señales de salida arrojarán los datos de interés mezclados con ruido o perturbaciones, en donde se deben considerar las señales que se tomarán en cuenta y las que no. El objetivo de este experimento diseñado es escoger los datos que llegarán a ser más informativos [2].

2.4.2. Estructura del modelo

Éstos se obtienen generalmente del conocimiento previo que se tenga del sistema y de las perturbaciones. Se deben elegir un conjunto de modelos candidatos dentro de un grupo de modelos que parezcan acomodarse al sistema. Esto es sin duda lo más importante ya que existe cierta dificultad tanto para la elección del modelo como para el proceso de identificación, por ello debe realizarse la selección del modelo en forma cautelosa, tomando en cuenta las características y el conocimiento que se tiene del sistema.

2.4.3. Estimación de parámetros

La estimación de parámetros en sistemas no lineales se considera un problema difícil. La razón es que identificar un sistema no lineal conlleva dos grandes etapas: la selección de la estructura del modelo con un cierto número de parámetros y la selección de un algoritmo que estime dichos parámetros.

2.4.4. Validación

Una vez elegido el modelo, se debe determinar su comportamiento y lo bien que se ajuste al sistema, se evaluará la calidad del modelo intentando reproducir las medidas de los datos estimados; si el comportamiento es deficiente, debemos cambiar el modelo; por el contrario, si presenta un buen desempeño, entonces la aproximación habrá sido satisfactoria según las consideraciones tomadas en cuenta.

2.4.5. Lazo de identificación del sistema

El proceso de identificación del sistema, responde a un flujo de lógica natural. Como inicio se deben recolectar los datos de interés, posteriormente, se elige la estructura del modelo establecido y se procede a la selección del mejor modelo del conjunto de modelos candidatos, ver Figura 2.17.

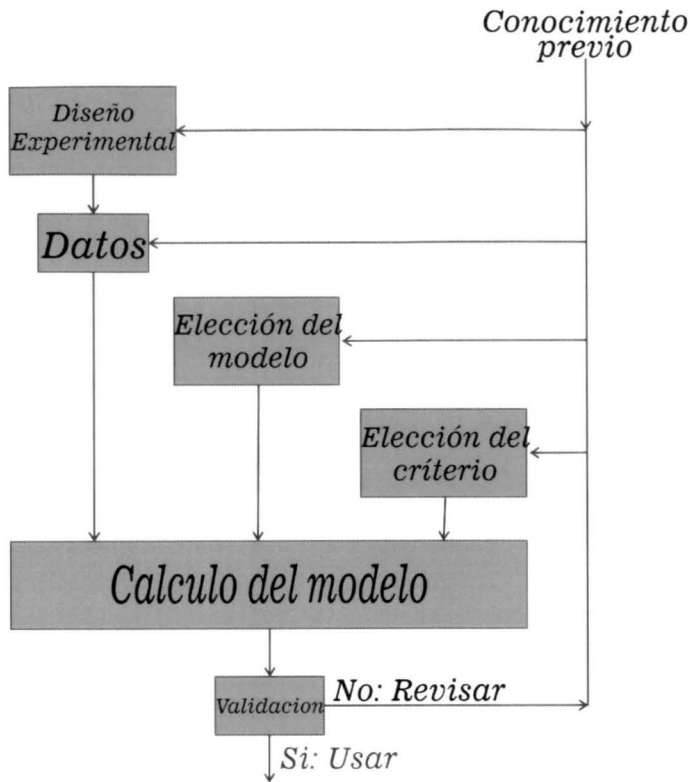


Figura 2.17: Lazo de identificación del sistema

2.5. Observadores adaptables

Para profundizar en el tema de identificación paramétrica, se analizan los artículos: “Adaptive Observer for Multiple-Input-Multiple-Output (MIMO) Linear Time-Varying Systems” y “Adaptive Observer for Discrete Time Linear Time Varying Systems” [9]. En ellos se plantean algoritmos numéricos que contribuyen al diseño de un observador adaptable lineal. La finalidad es la estimación de parámetros de diversos sistemas dinámicos lineales en tiempo continuo o discreto. El diseño se basa en proponer un conjunto de estados estimados y parámetros en tiempo discreto, con la condición de tener una entrada constante que excite al sistema y un análisis que garantice la convergencia. La adaptación es de tiempo continuo a tiempo discreto.

Se considera un sistema lineal variante en el tiempo MIMO (Múltiples Entradas, Múltiples Salidas) de la forma:

$$\begin{aligned}x(t) &= Ax(t) + Bu(t) + \Psi\theta(t) \\y(t) &= Cx(t)\end{aligned}\tag{2.16}$$

$A(t)$, $B(t)$, $C(t)$ son matrices conocidas variantes en el tiempo de dimensiones apropiadas, $\theta \in \mathbb{R}^p$ es un vector de parámetros desconocidos y se supone constante a menos que se indique lo contrario. $\Psi(t) \in \mathbb{R}^n \times \mathbb{R}^p$ es una matriz de señales conocidas. Las matrices $A(t)$, $B(t)$, $C(t)$ y $\Psi(t)$ son todas asumidas continuas y uniformemente acotadas en el tiempo.

El problema que se considera es la relación de la estimación de $x(t)$, $y(t)$ y $\Psi(t)$. En lugar de asumir la observación completa de la estabilidad de un sistema extendido, el método propuesto en este trabajo se basa en la estabilidad de las matrices $A(t)$, $C(t)$ y en algunas condiciones de excitación persistente.

Teorema 1. Se establece que para algún parámetro θ conocido el observador de estados puede diseñarse con una matriz de ganancia K . Suponiendo $\Gamma \in \mathbb{R}^p \times \mathbb{R}^p$ una matriz simétrica positiva definida.

$$\begin{aligned}\hat{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + \Psi\hat{\theta}(t) + [K + \Upsilon\Gamma\Upsilon^T C^T \Sigma][y(t) - C\hat{x}(t)] \\ \hat{\hat{\theta}}(t) &= \Gamma\Upsilon^T C^T \Sigma[y(t) - C\hat{x}(t)]\end{aligned}\tag{2.17}$$

Donde: La matriz de señales generadas por la ecuación diferencial ordinaria del sistema es

$$\dot{\Upsilon}(t) = [A - KC]\Upsilon(t) + \Psi\tag{2.18}$$

El observador adaptable con condiciones iniciales $x(t_0)$, $\hat{x}(t_0)$, $\hat{\theta}(t_0)$ y $\forall \theta \in \mathbb{R}^p$, los errores $\hat{x}(t) - x(t)$ y $\hat{\theta}(t) - \theta(t)$ convergen exponencialmente a cero cuando $t \rightarrow \infty$.

Una de las partes importante de este método es el diseño de la matriz de ganancia K . La simulación se lleva a cabo en *Simulink* para verificar que en realidad el observador converge al valor propuesto y así obtener el sistema estimado.

Observador adaptable en tiempo discreto

Considerando el siguiente sistema en tiempo discreto MIMO de la forma

$$\begin{aligned}
\theta_{k+1} &= \theta_k + e_k \\
x_{k+1} &= A_k x_k + B_k u_k + \Psi_k \theta_k + w_k \\
y_k &= C_k x_k + v_k
\end{aligned} \tag{2.19}$$

A partir del modelo en tiempo continuo de un observador, se realiza la adaptación correspondiente dando como resultado el siguiente sistema en tiempo discreto para un nuevo observador, para el caso libre de ruido.

$$\begin{aligned}
\Upsilon_{k+1} &= [A_k - K_k C_k] \Upsilon_k + \Psi_k \\
\hat{\theta}_{k+1} &= \hat{\theta}_k + \mu_k \Upsilon_k^T C_k^T [y(t) - C \hat{x}(t)] \\
\hat{x}_{k+1} &= A_k \hat{x}_k + B_k u_k + \Psi_k \hat{\theta}_k + K_k (y_k - C_k \hat{x}_k) + \Upsilon_{k+1} (\hat{\theta}_{k+1} - \hat{\theta}_k)
\end{aligned} \tag{2.20}$$

donde \hat{x}_k y $\hat{\theta}_k$ son los estados y los parámetros estimados, respectivamente, μ_k es una ganancia escalar, asumiendo $\mu_k > 0$, debe ser lo suficiente pequeña tal que: $\|\sqrt{\mu_k} C_k \Upsilon_k\| \leq 1$, para toda $k > 0$ y $\Upsilon_k \in \mathfrak{R}^n \times \mathfrak{R}^p$ es la matriz secuencial obtenida del filtro lineal.

Teorema 2. Las propiedades para el caso libre de ruido, se adapta al siguiente teorema: Si el ruido es ausente en el sistema (2.19), esto es que $e_k = 0$, $w_k = 0$, $v_k = 0$ para todo $k \geq 0$, es decir que la estimación de errores $\hat{x}_k - x_k$ y $\hat{\theta}_k - \theta_k$ tienden exponencialmente a cero cuando $k \rightarrow \infty$, el observador adaptable es global exponencial.

3

Planteamiento del problema

3.1. Descripción del sistema

En este trabajo se desarrolla el prototipo de un vehículo de transporte personal como se muestra en la Figura 3.1, la estructura es una base de metal a la cual se le acoplaron dos ruedas impulsadas por un motor cada una, un manubrio y un sistema electrónico de control. La funcionalidad del vehículo se enfoca en mantener el equilibrio por medio del autobalance vertical para no caer al piso. Con ayuda de las diferentes etapas para cada área, como la mecánica, los motores y la electrónica, se logra el objetivo de mantener equilibrado al sistema pese a cambios en su masa o perturbaciones externas.

El sensor utilizado es un IMU (Inertial Measurement Unit o Unidad de Medición Inercial) [22] ver Figura 4.7, que se encarga de medir la velocidad angular, orientación y fuerzas gravitacionales del sistema por medio de un acelerómetro y un giroscopio.

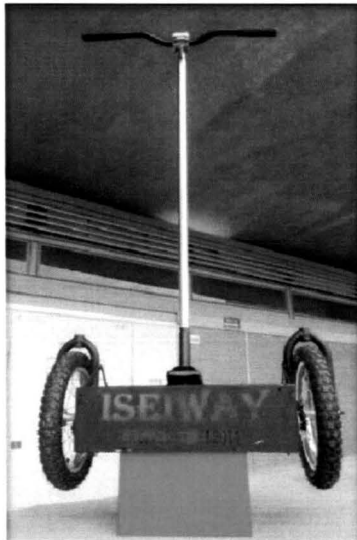


Figura 3.1: Prototipo ISEIWAY

3.2. Modelo del sistema

Para comenzar el análisis del sistema [14], se toma como referencia el diagrama eléctrico de un motor DC, logrando así, el modelo en variables de estado que rige su comportamiento, la dinámica del modelo está relacionada con el voltaje de entrada entre los motores y el torque necesario para lograr el equilibrio del vehículo [15].

La Figura 3.2 describe el diagrama eléctrico de un motor de corriente directa, en

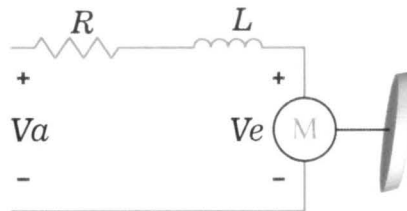


Figura 3.2: Diagrama eléctrico de un motor DC

el cual el voltaje se aplica en las terminales del motor y como consecuencia una corriente circula en la armadura del motor. El motor produce un torque (τ_m) que es proporcional a la corriente, la relación resultante es la siguiente:

$$\tau_m = K_m i(t) \quad (3.1)$$

donde K_m es la constante de torque electromagnético e $i(t)$ es la corriente que circula en la armadura.

Como la resistencia y el inductor están en serie con el voltaje V_a , el voltaje electromotriz (V_e) puede ser usado para modelar el circuito eléctrico del motor. El voltaje V_e se produce debido a que la bobina del motor se mueve a través de un campo electromagnético. El voltaje producido puede ser aproximado como una función lineal de la velocidad del eje. Esto puede ser descrito en la ecuación (3.2), donde K_e es la constante de fuerza contra-electromotriz y ω es la velocidad angular.

$$V_e = K_e \omega \quad (3.2)$$

Usando la ley de voltajes de Kirchoff que dice que la suma de todos los voltajes en un circuito es igual a cero. Para el motor DC se puede escribir lo siguiente.

$$V_e = V_a - Ri(t) - L \frac{di(t)}{dt} \quad (3.3)$$

Con V_a como el voltaje de entrada, R es la resistencia de armadura y L la inductancia del motor DC.

La ley de movimiento de Newton establece que la suma de todos los momentos de fuerza producidos en el eje, están linealmente relacionados con la aceleración del eje por la carga de inercia en la armadura, así como se muestra en la Figura 3.3.

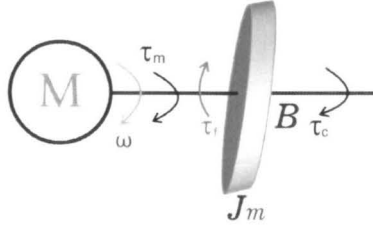


Figura 3.3: Suma de torques

$$\Sigma M = \tau_m = \tau_v + \tau_f + \frac{\tau}{r} \quad (3.4)$$

donde τ es el torque generado después del conjunto de engranes sobre el eje de la carga y r es la relación de reducción de engranes. Por otro lado:

$$\tau_v = J_m \ddot{\theta}_m \quad (3.5)$$

$$\tau_f = B \dot{\theta}_m \quad (3.6)$$

dado que J_m es el momento de inercia equivalente al eje del rotor, $\ddot{\theta}_m$ es la aceleración angular de la carga, B es el coeficiente de rozamiento viscoso y $\dot{\theta}_m$ es la velocidad angular de la carga.

Sustituyendo la ecuación (3.5) y la ecuación (3.6) en la ecuación (3.4) se obtiene:

$$K_m i(t) = J_m \ddot{\theta}_m + B \dot{\theta}_m + \frac{\tau}{r} \quad (3.7)$$

Para llevar a cabo el análisis completo del sistema, se relaciona la parte eléctrica con la mecánica, para ello se debe abordar el sistema péndulo invertido mostrado en la Figura 3.4.

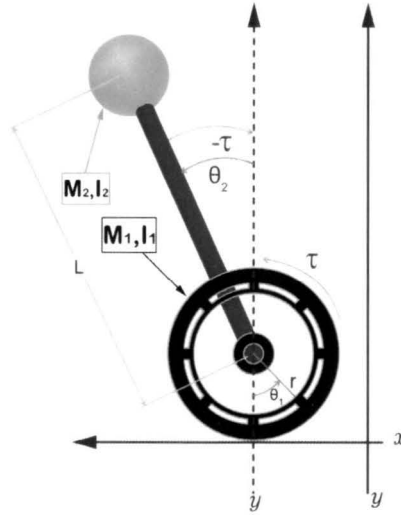


Figura 3.4: Diagrama de fuerzas péndulo invertido

El sistema se compone de un cuerpo largo con dos ruedas montadas a su extremo, debido a la complejidad de su análisis, las dos ruedas serán asumidas como una unidad, y se considera que el vehículo se desplaza sólo en una línea recta. Se analiza el sistema con la posición de las ruedas en las siguientes coordenadas.

$$(x_i, y_i) = (x_d, y_d) = (R\theta, R) \quad (3.8)$$

Donde x_i es la coordenada en x de la rueda izquierda, x_d es la coordenada en x de la rueda derecha, y_i es la coordenada en y de la rueda izquierda, y_d es la coordenada en y de la rueda derecha, R es el radio de las ruedas y θ es el ángulo del péndulo respecto a la posición vertical.

Las coordenadas del péndulo invertido (x_{m2}, y_{m2}) están dadas por las ecuaciones de la posición de las ruedas.

$$x_i = x_d = x_{m2}, y_i = y_d = y_{m2}$$

Por lo tanto y considerando que las dos ruedas tienen las mismas coordenadas, se realiza el análisis para una sola rueda.

$$(x_{m2}, y_{m2}) = (R\theta + l \sin \theta_2, R + l \cos \theta_2) \quad (3.9)$$

El modelo se analiza mediante el método Euler-Lagrange [16]. Para ello se necesita del cálculo de las energías como se muestra a continuación.

Energía cinética traslacional:

$$K = \frac{1}{2}m_1\dot{x}_i^2 + \frac{1}{2}m_1\dot{x}_d^2 + \frac{1}{2}m_2\dot{x}_{m2}^2 + \frac{1}{2}m_2\dot{y}_{m2}^2 \quad (3.10)$$

donde m_1 es la masa de la rueda y m_2 es la masa del péndulo.

Derivando la ecuación (3.9) para sustituirla en la ecuación (3.10).

$$K = \frac{1}{2}m_1R\dot{\theta}_i^2 + \frac{1}{2}m_1R\dot{\theta}_d^2 + \frac{1}{2}m_2\left(R\dot{\theta}_2 + l\dot{\theta}_2\cos\theta_2\right)^2 + \frac{1}{2}m_2\left(-l\dot{\theta}_2\sin\theta_2\right)^2 \quad (3.11)$$

Energía cinética rotacional

$$K = \frac{1}{2}J_{m1}\dot{\theta}_i^2 + \frac{1}{2}J_{m1}\dot{\theta}_d^2 + \frac{1}{2}J_{m2}\dot{\theta}_2^2 \quad (3.12)$$

La energía cinética total es la suma de la energía cinética traslacional y la energía cinética rotacional, considerando que el ángulo de rotación de las ruedas es el mismo [17]:

$$\dot{\theta}_1 = \dot{\theta}_i = \dot{\theta}_d$$

$$K = m_1(R\dot{\theta}_1)^2 + \frac{1}{2}m_2(R\dot{\theta}_1 + l\dot{\theta}_2\cos\theta_2)^2 + \frac{1}{2}m_2\left(-l\dot{\theta}_2\sin\theta_2\right)^2 + J_{m1}\dot{\theta}_1^2 + \frac{1}{2}J_{m2}\dot{\theta}_2^2 \quad (3.13)$$

Energía potencial

$$U = m_1gy_{mi} + m_1gy_{md} + m_2gy_{m2} \quad (3.14)$$

$$U = 2m_1gR + m_2g(R + l\cos\theta_2) \quad (3.15)$$

Aplicando el Lagrangiano

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

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

$$m_1 R^2 \ddot{\theta}_1 + m_2 R^2 \ddot{\theta}_1 + J_{m1} \ddot{\theta}_1 + m_2 R l \ddot{\theta}_2 \cos \theta_2 - m_2 R l \dot{\theta}_2^2 \sin \theta_2 = \tau \quad (3.16)$$

$$m_2 R l \ddot{\theta}_1 \cos \theta_2 + m_2 l^2 \ddot{\theta}_2 + J_{m1} \ddot{\theta}_2 + m_2 g l \sin \theta_2 = -\tau \quad (3.17)$$

Las dos ecuaciones anteriores pueden ser linealizadas considerando $\theta_2 = \pi + \phi$, donde ϕ representa un pequeño ángulo en dirección del eje y , obteniendo así un modelo lineal en espacio de estados:

$$\left(\frac{d\theta_2}{dt} \right)^2 = 0, \cos \theta_2 = -1, \sin \theta_2 = -\phi$$

El análisis anterior se llevó a cabo para una sola rueda y asumiendo que se tiene la misma masa e inercia en ambas ruedas, la ecuación de movimiento puede ser linealizada sin considerar $\cos \theta_2 = -1$ por cuestiones de simplicidad, donde posteriormente con ayuda de MATLAB[®] se hará dicha sustitución:

$$\left(2m_2 R^2 + m_2 R^2 + 2J_{m1} \right) \ddot{\theta}_1 + m_2 R l \ddot{\theta}_2 \cos \theta_2 = \tau \quad (3.18)$$

$$m_2 R l \ddot{\theta}_1 \cos \theta_2 + \left(m_2 l^2 + J_{m1} \right) \ddot{\theta}_2 - m_2 g l \theta_2 = -\tau \quad (3.19)$$

Usando la ecuación (3.3) y la ecuación (3.7) se relaciona la etapa eléctrica con la mecánica y considerando la inductancia igual a cero, esto debido a que su valor es del orden de los μH (microhenries), por lo tanto $L \frac{di(t)}{dt} = 0$.

$$K_e (\dot{\theta}_m - \dot{\theta}_2) = V_a - R i(t) \quad (3.20)$$

$$K_m i(t) = J_m \ddot{\theta}_m + B (\dot{\theta}_m - \dot{\theta}_2) + \frac{\tau}{r} \quad (3.21)$$

$$i(t) = \frac{J_m}{K_m} \ddot{\theta}_m + \frac{B}{K_m} (\dot{\theta}_m - \dot{\theta}_2) + \frac{\tau}{r K_m} \quad (3.22)$$

$$\ddot{\theta}_m = r \ddot{\theta}_1 \quad \dot{\theta}_m = r \dot{\theta}_1$$

$$\tau = \frac{V_r K_m}{R_a} - R_a J_m r^2 \ddot{\theta}_1 - r B (r \dot{\theta}_1 - \dot{\theta}_2) + \frac{K_e K_m r}{R_a} (r \dot{\theta}_1 - \dot{\theta}_2) \quad (3.23)$$

3. Planteamiento del problema

Sustituyendo la ecuación (3.23) en la ecuación (3.18) y la ecuación (3.19), para posteriormente despejar $\ddot{\theta}_1$ y $\ddot{\theta}_2$ se obtiene el siguiente modelo no lineal:

$$\ddot{\theta}_1 = \frac{I_2 K_m V r + B I_2 R_a \dot{\theta}_2 r + I_2 K_t K_c \dot{\theta}_2 r - B I_2 R_a \dot{\theta}_1 r^2 - I_2 K_m K_c \dot{\theta}_1 r^2 + K_m V l^2 m_2 r + B R_a \dot{\theta}_2 l^2 m_2 r + K_m K_c \dot{\theta}_2 l^2 m_2 r - B R_a \dot{\theta}_1 l^2 m_2 r^2 - K_m K_c \dot{\theta}_1 l^2 m_2 r^2 + K_m V l m_2 r^2 \cos(\theta_2) - R_a g l^2 m_2^2 r \theta_2 \cos(\theta_2) - B R_a \dot{\theta}_1 l m_2 r^3 \cos(\theta_2) + B R_a \dot{\theta}_2 l m_2 r^2 \cos(\theta_2) - K_m K_c \dot{\theta}_1 l m_2 r^3 \cos(\theta_2) + K_m K_c \dot{\theta}_2 l m_2 r^2 \cos(\theta_2)}{J_m R_a^2 l^2 m_2 r^2 + J_m R_a^2 l m_2 r^3 \cos(\theta_2) + I_2 J_m R_a^2 r^2 - R_a l^2 m_2^2 r^2 \cos(\theta_2)^2 + R_a l^2 m_2^2 r^2 + 2 m_1 R_a l^2 m_2 r^2 + 2 I_1 R_a l^2 m_2 + I_2 R_a m_2 r^2 + 2 I_2 m_1 R_a r^2 + 2 I_1 I_2 R_a} \quad (3.24)$$

$$\ddot{\theta}_2 = \frac{2 K_m V m_1 r^3 + K_m V m_2 r^3 + 2 I_1 K_m V r + 2 B I_1 R_a \dot{\theta}_2 r + 2 I_1 K_m K_c \dot{\theta}_2 r - 2 B I_1 R_a \dot{\theta}_1 r^2 - 2 I_1 K_m K_c \dot{\theta}_1 r^2 - 2 B R_a \dot{\theta}_1 m_1 r^4 + 2 B R_a \dot{\theta}_2 m_1 r^3 - B R_a \dot{\theta}_1 m_2 r^4 + B R_a \dot{\theta}_2 m_2 r^3 - 2 K_m K_c \dot{\theta}_1 m_1 r^4 + 2 K_m K_c \dot{\theta}_2 m_1 r^3 - K_m K_c \dot{\theta}_1 m_2 r^4 + K_m K_c \dot{\theta}_2 m_2 r^3 + K_m V l m_2 r^2 \cos(\theta_2) - R_a g l m_2^2 r^2 \theta_2 - 2 I_1 R_a g l m_2 \theta_2 - 2 R_a g l m_1 m_2 r^2 \theta_2 - B R_a \dot{\theta}_1 l m_2 r^3 \cos(\theta_2) + B R_a \dot{\theta}_2 l m_2 r^2 \cos(\theta_2) - K_m K_c \dot{\theta}_1 l m_2 r^3 \cos(\theta_2) + K_m K_c \dot{\theta}_2 l m_2 r^2 \cos(\theta_2) - J_m R_a^2 g l m_2 r^2 \theta_2}{J_m R_a^2 l^2 m_2 r^2 + J_m R_a^2 l m_2 r^3 \cos(\theta_2) + I_2 J_m R_a^2 r^2 - R_a l^2 m_2^2 r^2 \cos(\theta_2)^2 + R_a l^2 m_2^2 r^2 + 2 m_1 R_a l^2 m_2 r^2 + 2 I_1 R_a l^2 m_2 + I_2 R_a m_2 r^2 + 2 I_2 m_1 R_a r^2 + 2 I_1 I_2 R_a} \quad (3.25)$$

La representación en variables de estado del sistema es:

$$x = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} \dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \ddot{\theta}_1 \\ \dot{x}_2 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \text{Posición angular de la rueda} \\ \text{Velocidad angular de la rueda} \\ \text{Posición angular del péndulo} \\ \text{Velocidad angular del péndulo} \end{bmatrix}$$

Matriz en variables de estado

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

$$y = Cx(t)$$

$$\dot{x} = \begin{bmatrix} x_2 \\ a_1 x_2 + a_2 x_3 + a_3 x_4 \\ x_4 \\ a_4 x_2 + a_5 x_3 + a_6 x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ a_7 u \\ 0 \\ a_8 u \end{bmatrix}$$

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

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & 1 \\ 0 & a_4 & a_5 & a_6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ a_7 \\ 0 \\ a_8 \end{pmatrix} u$$

$$a_1 = - \frac{B I_2 R a r^2 + I_2 K t K w r^2 + K t K w l^2 m_2 r^2 + B R a l^2 m_2 r^2 + B R R a l m_2 r^2 + K t K w R l m_2 r^2}{2 I_1 I_2 R a + 2 I_2 R^2 R a m_1 + I_2 R^2 R a m_2 + 2 I_1 R a l^2 m_2 + I_2 J m R a^2 r^2 + 2 R^2 R a l^2 m_1 m_2 + J m R a^2 l^2 m_2 r^2 + J m R R a^2 l m_2 r^2}$$

$$a_2 = \frac{R R a g l^2 m_2^2}{2 I_1 I_2 R a + 2 I_2 R^2 R a m_1 + I_2 R^2 R a m_2 + 2 I_1 R a l^2 m_2 + I_2 J m R a^2 r^2 + 2 R^2 R a l^2 m_1 m_2 + J m R a^2 l^2 m_2 r^2 + J m R R a^2 l m_2 r^2}$$

$$a_3 = \frac{B I_2 R a r + I_2 K t K w r + B R a l^2 m_2 r + K t K w l^2 m_2 r + B R R a l m_2 r + K t K w R l m_2 r}{2 I_1 I_2 R a + 2 I_2 R^2 R a m_1 + I_2 R^2 R a m_2 + 2 I_1 R a l^2 m_2 + I_2 J m R a^2 r^2 + 2 R^2 R a l^2 m_1 m_2 + J m R a^2 l^2 m_2 r^2 + J m R R a^2 l m_2 r^2}$$

$$a_4 = \frac{2 B I_1 R a r^2 + 2 I_1 K t K w r^2 + 2 B R^2 R a m_1 r^2 + B R^2 R a m_2 r^2 + 2 K t K w R^2 m_1 r^2 + K t K w R^2 m_2 r^2 + B R R a l m_2 r^2 + K t K w R l m_2 r^2}{2 I_1 I_2 R a + 2 I_2 R^2 R a m_1 + I_2 R^2 R a m_2 + 2 I_1 R a l^2 m_2 + I_2 J m R a^2 r^2 + 2 R^2 R a l^2 m_1 m_2 + J m R a^2 l^2 m_2 r^2 + J m R R a^2 l m_2 r^2}$$

$$a_5 = - \frac{g l R^2 R a m_2^2 + 2 g l m_1 R^2 R a m_2 + J m g l R a^2 m_2 r^2 + 2 I_1 g l R a m_2}{2 I_1 I_2 R a + 2 I_2 R^2 R a m_1 + I_2 R^2 R a m_2 + 2 I_1 R a l^2 m_2 + I_2 J m R a^2 r^2 + 2 R^2 R a l^2 m_1 m_2 + J m R a^2 l^2 m_2 r^2 + J m R R a^2 l m_2 r^2}$$

$$a_6 = - \frac{2 B I_1 R a r + 2 I_1 K t K w r + 2 B R^2 R a m_1 r + B R^2 R a m_2 r + 2 K t K w R^2 m_1 r + K t K w R^2 m_2 r + B R R a l m_2 r + K t K w R l m_2 r}{2 I_1 I_2 R a + 2 I_2 R^2 R a m_1 + I_2 R^2 R a m_2 + 2 I_1 R a l^2 m_2 + I_2 J m R a^2 r^2 + 2 R^2 R a l^2 m_1 m_2 + J m R a^2 l^2 m_2 r^2 + J m R R a^2 l m_2 r^2}$$

$$a7 = \frac{Kt m2 r l^2 + Kt R m2 r l + I2 Kt r}{2 I1 I2 Ra + 2 I2 R^2 Ra m1 + I2 R^2 Ra m2 + 2 I1 Ra l^2 m2 + I2 Jm Ra^2 r^2 + 2 R^2 Ra l^2 m1 m2 + Jm Ra^2 l^2 m2 r^2 + Jm R Ra^2 l m2 r^2}$$

$$a8 = -\frac{2 I1 Kt r + 2 Kt R^2 m1 r + Kt R^2 m2 r + Kt R l m2 r}{2 I1 I2 Ra + 2 I2 R^2 Ra m1 + I2 R^2 Ra m2 + 2 I1 Ra l^2 m2 + I2 Jm Ra^2 r^2 + 2 R^2 Ra l^2 m1 m2 + Jm Ra^2 l^2 m2 r^2 + Jm R Ra^2 l m2 r^2}$$

3.3. Análisis del modelo

Una vez obtenido el modelo en variables de estado del sistema, se realiza el proceso de identificación paramétrica de los motores para posteriormente sustituir dichos valores estimados en el modelo y así verificar cómo se comporta el sistema por medio de la simulación en SIMUKINK® [12].

La matriz de estados incluye diversos parámetros desconocidos que se requieren obtener, algunos de los parámetros involucrados provienen del motor, en la Figura 3.5 se describe en diagrama de bloques el procedimiento para adquirir los parámetros del motor DC. Usando un convertidor voltaje-frecuencia LM331 [20], una tarjeta de adquisición de datos (NI-PCIE 6321) [19] y un encoder (AMT102-V) [21] acoplado a la flecha del motor.

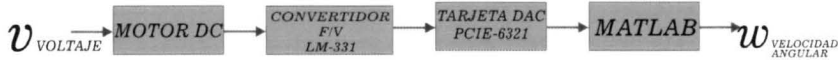


Figura 3.5: Diagrama de bloques para la adquisición de parámetros de un motor DC.

Para obtener la mejor aproximación de la función de transferencia se hace la identificación de parámetros del motor, se recurre a un algoritmo que proporcionará los parámetros desconocidos mostrado en el apéndice A. La forma de obtener dicha función es; acoplando el encoder con una resolución de 48 pulsos por revolución a la flecha del motor se genera un tren de pulsos a la salida del encoder cuando se aplica un voltaje al motor. La tarjeta de adquisición de datos proporciona la información necesaria que junto con MATLAB® y la herramienta *Ident*, realizan una aproximación de la función de transferencia que mejor se ajuste a la respuesta del motor. Para este caso, la mejor función de transferencia encontrada fue la siguiente:

$$G(s) = \frac{0.4139}{1.25 \times 10^{-6} s^2 + 0.00604 s + 0.5915} \quad (3.26)$$

Que responde a una entrada escalón, como se muestra en la Figura 3.6.

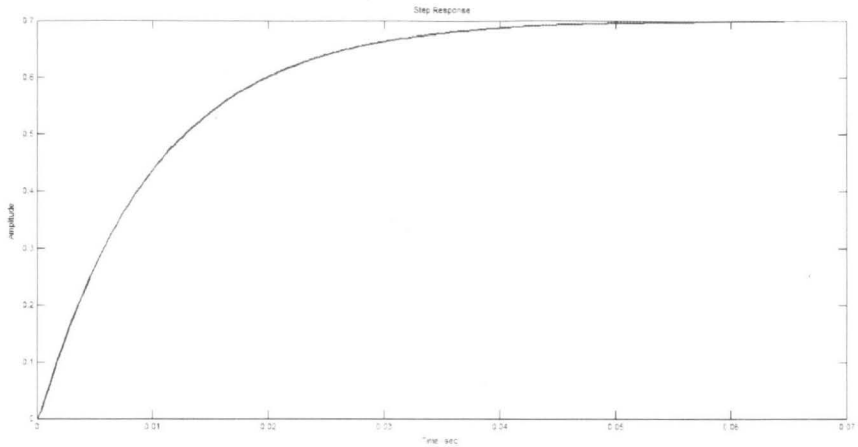


Figura 3.6: Respuesta transitoria de un motor DC

Para la obtención de los parámetros del motor se realizó un algoritmo en SIMULINK[®] que permitiera el cálculo de los mismos; a partir del voltaje de entrada, el voltaje final y la constante de tiempo τ . La resistencia y la inductancia del motor fueron medidas con un puente LCR (Bkprecision 878) La inercia del motor, la constante eléctrica y la fricción viscosa son los parámetros internos del motor DC.

En el apéndice A se puede verificar la construcción de dicho algoritmo, así como las variables encontradas mencionadas anteriormente.

Los parámetros obtenidos se muestran en el Cuadro 3.1.

Con los parámetros obtenidos del motor, se procede a sustituir los valores en la matriz de estados para analizar el modelo del sistema y verificar las simulaciones correspondientes.

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1.5592 & 58.5689 & 1.5592 \\ 0 & 0 & 0 & 1 \\ 0 & 0.2322 & -52.5154 & -0.2322 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0.4073 \\ 0 \\ -0.0607 \end{bmatrix} V_a \\ C &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (3.27)$$

3. Planteamiento del problema

Parámetro	Simbolo	Valor	Unidades
Gravedad	g	9.8	$\frac{m}{s^2}$
Radio de las ruedas	r	0.026	m
Masa de las ruedas	m_1	0.03	Kg
Masa del péndulo	m_2	0.781	Kg
Constante de fricción viscosa	B	0.00234	Nms
Inercia de las ruedas	I_1	0.00002028	$\frac{Kg}{m^2}$
Inercia del péndulo	I_2	0.00765458	$\frac{Kg}{m^2}$
Longitud del péndulo	l	0.099	m
Inercia del motor	J_m	0.0001234	$\frac{m}{s^2}$
Constante de torque del motor	K_t	0.006123	$\frac{Vs}{rad}$
Constante eléctrica del motor	K_e	0.006087	$\frac{Vs}{rad}$
Resistencia de armadura del motor	R_a	3	Ω

Cuadro 3.1: Tabla de parametros encontrados del motor DC.

Con el fin de analizar el comportamiento del sistema y validarlo, se realiza el diagrama de bloques en SIMULINK[®], como se muestra en la Figura 3.7.

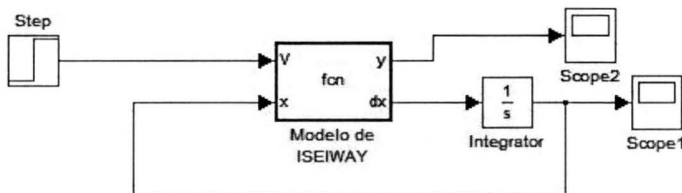


Figura 3.7: Diagrama de bloques del sistema ISEIWAY en SIMULINK[®]

La respuesta del sistema se representa en la Figura 3.8, en donde los estados describen un comportamiento totalmente inestable, es aquí donde comienza el análisis y el diseño de un controlador que permita conseguir la estabilidad del sistema para mantenerse equilibrado aún y cuando existan cambios de masa sobre él.

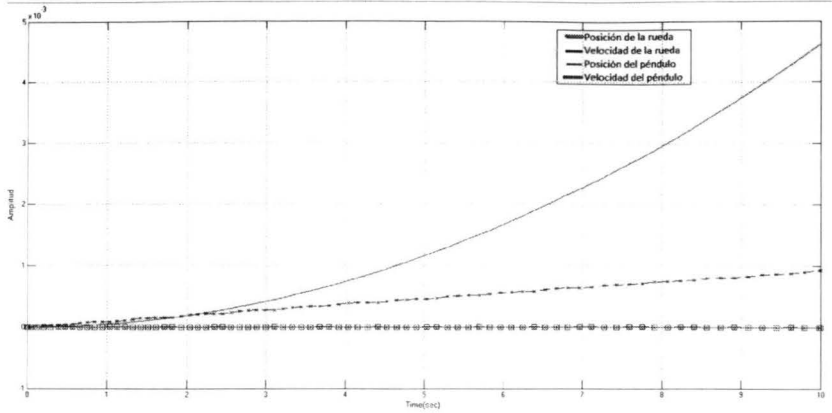


Figura 3.8: Respuesta inestable de los estados del sistema ISEIWAY

Haciendo un análisis al modelo en variables de estado se determina que se puede reducir a un modelo de tres estados ya que el estado θ_1 no queda involucrado en la dinámica del sistema.

El modelo resultante es el de la ecuación (3.28)

$$\begin{aligned} \dot{x} &= \bar{A}x + \bar{B}u, \\ y &= \bar{C}x, \end{aligned} \quad (3.28)$$

donde

$$x = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \text{velocidad angular de la rueda} \\ \text{posición angular del péndulo} \\ \text{velocidad angular del péndulo} \end{bmatrix}$$

Teniendo las matrices de estados siguientes:

$$\begin{aligned} \bar{A} &= \begin{bmatrix} -1.5592 & 58.5689 & 1.5592 \\ 0 & 0 & 1 \\ 0.2322 & -52.5154 & -0.2322 \end{bmatrix} \\ \bar{B} &= \begin{bmatrix} 0.4073 \\ 0 \\ -0.0607 \end{bmatrix} \\ \bar{C} &= [0 \ 1 \ 0] \end{aligned} \quad (3.29)$$

3. Planteamiento del problema

se obtiene la función de transferencia $G(s)$:

$$G(s) = \frac{-0.0607s - 6.838 \times 10^{-5}}{s^3 + 1.791s^2 + 52.52s + 68.28} \quad (3.30)$$

Para simplificar el análisis del sistema se pasa de tiempo continuo a tiempo discreto con un tiempo de muestreo de 0,01 *segundos* mediante la discretización aproximada.

$$\begin{aligned} A_d &= T_s \bar{A} + I, \\ B_d &= T_s \bar{B}, \\ C_d &= \bar{C}. \end{aligned} \quad (3.31)$$

Donde A_d , B_d y C_d son las matrices discretizadas, T_s es el tiempo de muestreo, \bar{A} , \bar{B} y \bar{C} son las matrices en tiempo continuo e I es la matriz identidad. Con la forma anterior se obtiene la siguiente función de transferencia en tiempo discreto:

$$G(z) = \frac{-3.016 \times 10^{-6} z^2 + 1.79 \times 10^{-8} z + 2.998 \times 10^{-6}}{z^3 - 2.977z^2 + 2.959z - 0.9822} \quad (3.32)$$

Las variables x , \dot{x} , ϕ y $\dot{\phi}$, son la posición y velocidad de las ruedas y la posición y velocidad del péndulo respectivamente. En el Capítulo 4 se aborda el diseño de un control LQR y un control PD para controlar el sistema.

4

Diseño y Construcción

Tomando como referencia la gran variedad de modelos existentes de sistemas autoestabilizables, se procede a implementar un modelo que permita ser funcional utilizando materiales de fácil maquilado, logrando así la fabricación del prototipo **ISEIWAY**.

4.1. Diseño mecánico

Para realizar un diseño eficiente se comienza por el análisis de los componentes elementales del sistema, así como se muestra en la Figura 4.1.

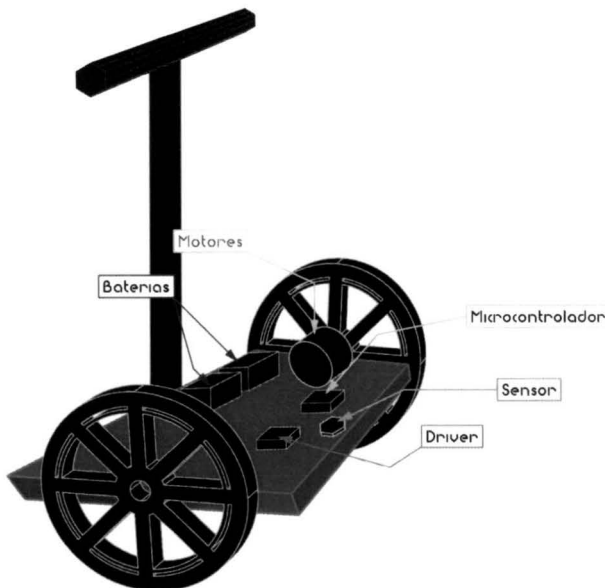


Figura 4.1: Componentes principales del sistema **ISEIWAY**

Debido a que la estructura del sistema debe ser resistente y rígida, se decide fabricar la base de ángulo de acero (ANGULO Fo. 3x19mm) en forma de una caja,

colocando una lámina en la parte inferior para reforzarla. Es ahí donde se acomodan los componentes tanto electrónicos como mecánicos. Se adaptaron dos horquillas delanteras de bicicleta para sujetar las ruedas y mantener el eje de giro centrado. La Figura 4.2 muestra el proceso de construcción.



Figura 4.2: Fabricación ISEIWAY

Uno de los problemas principales fue el acoplamiento del engrane del motor hacia la rueda, para ello se colocó una cadena que permitiera la interacción adecuada entre el motor y la rueda. Fue necesario soldar la catarina de la rueda al rin para permitir el giro de la rueda en ambos sentidos. Para ajustar los motores en la posición adecuada y permitir el giro correcto de las ruedas hacia los dos sentidos fue necesario realizar una placa que sujetara los motores a la base para que los mismos no se salieran de su posición original. Se ajustó el tamaño correspondiente de la cadena para lograr la tensión de la misma entre los motores y las ruedas de forma que se permitiera el giro sin que la cadena se saliera de la catarina. Esto se muestra en la Figura 4.3.



Figura 4.3: Ajuste de la rueda al motor

Utilizando una placa de acrílico se hicieron las tapas de la base para cubrir y proteger la parte interna del sistema. Se acopló una estructura para el manubrio sobre la superficie principal de tal forma que esto permita al usuario apoyarse sobre el sistema y definir hacia qué sentido dirigirse. Para ir hacia adelante o hacia atrás, se debe inclinar el manubrio en cualquiera de esas dos direcciones, en el caso de la vuelta, se colocaron dos micro switch (marca zippy) al manubrio, uno en cada extremo, cuando el usuario presione el switch derecho el torque generado en la rueda

izquierda aumenta mientras que el de la rueda derecha se reduce, de igual forma cuando se presione el switch izquierdo, la rueda derecha aumenta el torque mientras que el de la izquierda disminuye.

4.2. Diseño electrónico

Los microcontroladores son circuitos integrados que en su interior contienen una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Para este caso, se utilizó un microcontrolador con el fin de programar en él las órdenes específicas que logren mantener el autobalance del vehículo **ISEIWAY**.

La tecnología ARDUINO® es una plataforma de hardware y software que permite la interacción de los usuarios a un entorno de código abierto, ésta se basa en una placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está fundado en el lenguaje de programación *Processing*. El hardware consiste en una tarjeta de circuito impreso con un microcontrolador, usualmente Atmel AVR®, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a tarjetas de expansión (shields) que amplían las características de funcionamiento dicha placa. El *Arduino Due* [24], es una tarjeta basada en un microcontrolador Atmel SAM3X8E que permite implementar el código necesario para lograr el autobalance del sistema.

A continuación se contemplan los componentes electrónicos empleados en el vehículo.

- Arduino Due
- Sensor IMU MPU6050
- Driver Pololu® VNH5019
- Dos motores 24V 450Watts con un torque de 1.43Nm (14.58 Kgcm) modelo MOT-24450G
- Dos baterías de 12V a 32A/h.

Arduino Due

Es una placa de desarrollo de Arduino basada en ARM (Arquitectura RISC= Ordenador con Conjunto Reducido de Instrucciones). Esta placa esta basada en un microcontrolador ARM CortexM3 de 32 bits, que permite la adquisición de las señales a procesar. Dependiendo de su configuración cada uno de los puertos o líneas pueden ser definidos para emitir o recibir una o más señales. Este modelo proporciona dos puertos USB, uno para programación y comunicación, y otro que

IMU MPU6050 (Acelerómetro y Giroscopio)

El sensor IMU (Unidad de Medición Inercial) [27] como se muestra en la Figura 4.7, es un dispositivo capaz de adquirir valores referentes a un plano, proporcionando dichos valores en bruto de las coordenadas x, y, z . Tiene una resolución de 15mV por grado por segundo. Si por ejemplo, se sabe que la gravedad terrestre es de aproximadamente $9.8m/s^2$, es posible calcular el ángulo de inclinación de la IMU con la siguiente relación:

Para el ángulo y se tiene,

$$\theta_y = \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \quad (4.1)$$

Para el ángulo x se tiene,

$$\theta_x = \arctan\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \quad (4.2)$$

El sensor MPU6050 [22] es una IMU de 6 grados de libertad (6GDL). Esto significa que tiene un acelerómetro y un giroscopio, ambos de 3 ejes ($3+3 = 6GDL$).

Acelerómetro: Este sensor mide la aceleración debido al cambio de velocidad de un objeto, esto se mide en fuerza de gravedad. El acelerómetro es útil ya que con él se puede detectar las vibraciones y la orientación del sistema. Su comunicación es a través de un convertidor analógico, digital o interfaz de conexión modulada por ancho de pulso. La interfaz analógica proporciona un valor en voltajes en cada uno de sus ejes y por medio de funciones matemáticas es posible determinar dicho valor en línea para permitir que el sistema **ISEIWAY** logre autobalancearse cuando se presenten movimientos bruscos que lo perturben. Las señales del sensor serán enviadas a la tarjeta ARDUINO® y con el algoritmo de programación se realizará el procesamiento de dichas señales.

Giroscopio: Este sensor sirve para monitorear la orientación, dirección, movimiento angular y la rotación de un sistema, puede medir la rotación en los tres ejes $x, y, y z$. Es usado para medir la posición de equilibrio del sistema. Esta información será relacionada con la del acelerómetro y se enviarán a la placa ARDUINO® para que ahí se lleven a cabo las instrucciones adecuadas y el sistema tenga el comportamiento deseado (mantener el equilibrio y no caer).

La forma en que se relacionan los sensores es: si conocemos el ángulo inicial del acelerómetro, podemos sumarle el valor que proporciona el giroscopio para saber así el nuevo ángulo a cada momento; si por ejemplo la IMU tiene 0° y el giroscopio realiza una medida cada segundo y marca 5 en el eje x , obtendremos el ángulo con la siguiente fórmula:

$$\theta_y = \theta_{y\text{anterior}} + x\Delta t \quad (4.3)$$

donde θ_y es el ángulo calculado, $\theta_{y_{anterior}}$ es el valor que se tiene en ese instante de tiempo, x es el valor registrado en esa coordenada y Δt es el tiempo que transcurre cada vez que se calcula dicha fórmula.

Es muy común tener errores en las medidas, para ello se procede a la implantación de un filtro complementario que permita dejar pasar sólo los datos de interés.

Filtro Complementario

Una vez obtenidos los valores tanto del acelerómetro como del giroscopio, se puede aplicar un filtro para facilitar la adquisición de la señal necesaria que será procesada por el control digital elegido. El objetivo principal de este filtro permite usar la medición obtenida por el giroscopio y realizar la corrección de la deriva con la medición realizada por el acelerómetro, debido a que esta última medición tiende a ser la aceleración de la gravedad.

El filtro complementario resulta sencillo de tratar matemáticamente y en razón de su baja complejidad de implementación consume pocos recursos computacionales. La finalidad del filtro complementario es combinar la salida del acelerómetro y del giroscopio para obtener una buena estimación del ángulo de orientación de alguna plataforma, compensando la deriva del giroscopio.

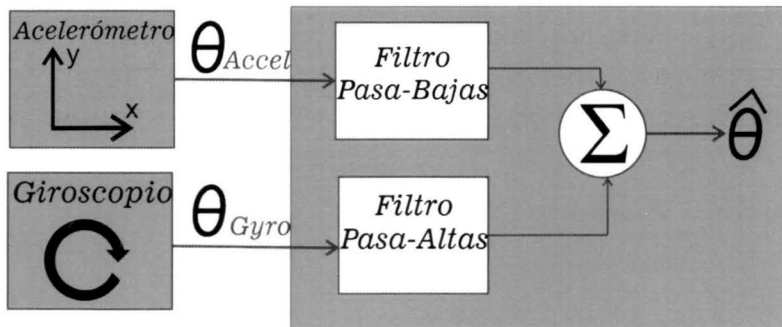


Figura 4.5: Estructura del filtro complementario

El filtro complementario propuesto se muestra en la Figura 4.5, donde θ_{Accel} es el ángulo medido por el acelerómetro cuya señal está afectada por ruidos de alta frecuencia provenientes de las vibraciones. θ_{Gyro} es el ángulo medido por el giroscopio, afectado por la deriva y $\hat{\theta}$ es el ángulo estimado. Las funciones de transferencia del filtro deben ser elegidas de acuerdo a la ecuación:

$$H_a(s)G(s) + H_g(s)(1 - G(s)) \quad (4.4)$$

donde $H_a(s)$ y $H_g(s)$ representan la función de transferencia del acelerómetro y el giróscopio respectivamente. Suponemos que las funciones de transferencia de los sensores son iguales a 1. Esto es $H_a(s) = H_g(s) = 1$. La función de transferencia elegida para $G(s)$ es un filtro pasa bajos de primer orden, lo cual hace que la estimación en baja frecuencia dependa de la medición del acelerómetro.

$$G(s) = \frac{\alpha}{s + \alpha} \quad (4.5)$$

En tanto que la función de transferencia elegida para $1 - G(s)$ es un filtro pasa altos de un polo:

$$1 - G(s) = \frac{\alpha}{s + \alpha} \quad (4.6)$$

este filtro nos permite hacer que las componentes de alta frecuencia estimadas estén dominadas por las mediciones provenientes del giróscopio. Como podemos ver en la Figura 4.5, si ambas mediciones son ideales, la función de transferencia total del filtro resulta:

$$\frac{\hat{\theta}(s)}{\theta(s)} = G(s) + (1 - G(s)) \quad (4.7)$$

Y esto hace que:

$$\hat{\theta}(s) = \theta(s)$$

Para fines de implementación en el microcontrolador, la ecuación (4.8) representa el filtro complementario en términos de programación, donde $A = G(s)$ y $(1 - A) = (1 - G(s))$.

$$\theta = A * (\theta_{ant} + \theta_{Gyro}) + (1 - A) * \theta_{Accel} \quad (4.8)$$

donde θ_{ant} es el último ángulo calculado, θ_{Accel} es el valor obtenido por el acelerómetro, θ_{Gyro} corresponde al valor obtenido por el giróscopio y A es el coeficiente definido por la frecuencia de muestreo (f_m) y la frecuencia de corte del filtro (f_c). Su estructura es:

$$A = \frac{\frac{1}{f_c}}{\frac{1}{f_c} + \frac{1}{f_m}} \quad (4.9)$$

La ecuación (4.10) describe el filtro complementario digital a implementar en el sistema.

$$\theta = 0.98 * (\theta_{\text{aut}} + \theta_{\text{Cgyro}}) + 0.02 * \theta_{\text{Accel}} \quad (4.10)$$

Las constantes que multiplican a las variables anteriores se pueden ir ajustando sin dejar de lado que el resultado en todo momento debe ser igual a 1 debido a la suposición planteada para las funciones de transferencia del filtro. Esta fórmula es la misma para el eje x y y .

La gráfica de la Figura 4.6 describe el comportamiento de los valores en bruto y los valores filtrados simultáneamente [28].

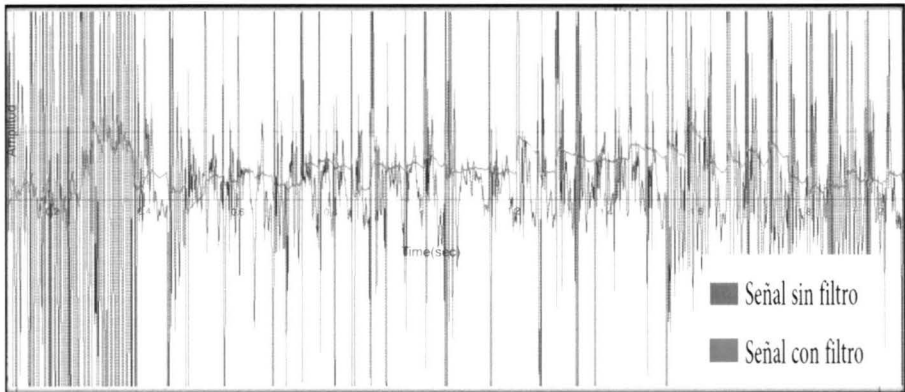


Figura 4.6: Señal del IMU con filtro complementario usando SerialChart

El IMU proporciona valores en bruto (raw values), debido a ello es necesario aplicar el filtro complementario descrito anteriormente para filtrar los datos con la finalidad de obtener sólo los valores de interés. Sus principales características son:

- Protocolo I2C [26], salida digital de datos de 6 o 9 ejes.
- Voltaje de entrada: 2.3 a 3.4V.
- Sensor de velocidad angular (giroscopio) de ± 250 , ± 500 , ± 1000 y ± 2000 dps.
- Acelerómetro con rangos de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.

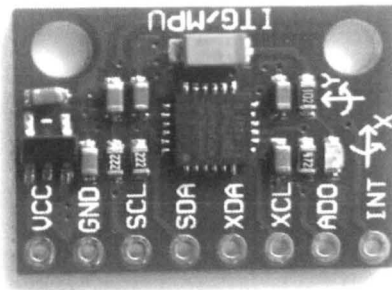


Figura 4.7: IMU MPU6050

Motor DC

En el sistema **ISEIWAY** se incluyeron dos motores de 24V con una transmisión de engranes de reducción y cepillo de imán permanente, sus características son las siguientes:

- Alimentación DC de 24V
- Potencia 450 Watts
- Rotación 420 rpm
- Torque 1.43 Nm
- Eficiencia $\geq 78\%$

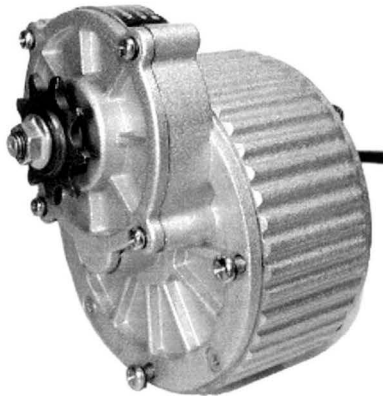


Figura 4.8: MOTOR DC 24V 450 WATTS

Driver POLOLU® VNH5019

Este controlador es de tipo shield (escudo) y permite controlar hasta dos motores de corriente continua de alta potencia de forma sencilla. Ver Figura 4.9. El driver cuenta con un par de controladores robustos (*VNH5019*) para motores DC [25]. Opera en un rango de 5.5 a 24 V y puede ofrecer una corriente continua de 12A (30A pico) por canal, o una corriente continua de 24A (60A pico) a un sólo motor conectado a ambos canales. Las especificaciones de este dispositivo son:

- Amplia gama de tensión de funcionamiento: 5.5 a 24 V.
- Alta corriente de salida: hasta 12A continua (30A máximo) por motor.
- Las salidas del motor se pueden combinar para ofrecer hasta 24A de corriente continua (60A como máximo) a un solo motor.
- El PWM opera hasta 20 kHz, al ser ultrasónico permite un funcionamiento más silencioso del motor.
- LEDs indicadores que permiten percatarse de la activación de los canales, incluso cuando no está conectado un motor.

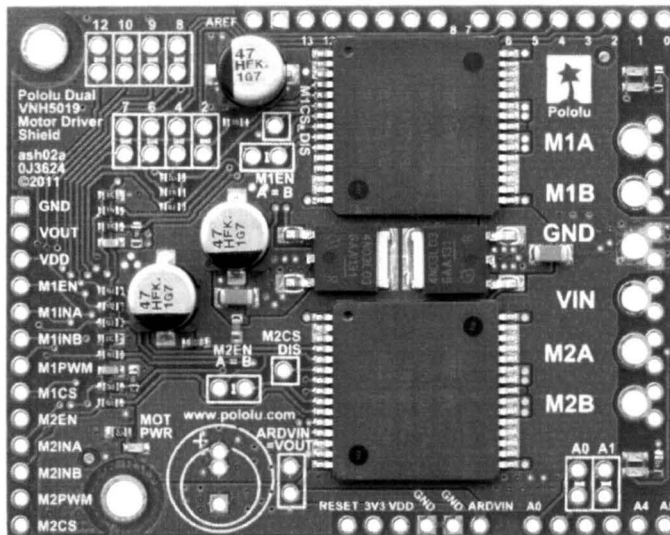


Figura 4.9: DRIVER VNH5019

Batería

El vehículo exige una corriente alta de pico al arranque, debido a esto se opta por adquirir dos baterías de 12V conectadas en serie para proporcionar la energía

necesaria y el sistema funcione correctamente con un mayor tiempo de operación. Las baterías son de gel a 12V 32Ah como se muestra en la Figura 4.10, cada una resistente a choques, vibraciones, productos químicos y al calor. Soporta altas corrientes de salida.



Figura 4.10: BATERÍA 12V 32Ah

4.3. Diseño del controlador

El diseño del control para el sistema se realiza por el método del lugar geométrico de las raíces, partiendo del modelo analizado anteriormente (visto el Capítulo 3).

$$\begin{aligned}\dot{x} &= \bar{A}x + \bar{B}u \\ y &= \bar{C}x\end{aligned}$$

Se obtiene la función de transferencia.

$$G(s) = \frac{-0.0607 - 6.838 \cdot 10^{-5}s}{s^3 + 1.791s^2 + 52.52s + 68.28s} \quad (4.11)$$

Para la cual se diseñará un control PD considerando los parámetros de la respuesta en tiempo continuo tal que el sobrepaso sea del 10%, el tiempo de asentamiento de 1.3 segundos y con un tiempo de muestreo de 0.01 segundos.

$$\omega_s = \frac{2\pi}{T_s} = \frac{2\pi}{0.01} = 628.3185 \quad (4.12)$$

$$\xi = \frac{1}{\sqrt{1 + \left(\frac{\pi}{\ln 0.10}\right)^2}} = 0.6 \quad (4.13)$$

$$\omega_n = \frac{4}{\xi t_s} = \frac{4}{0.6 * 1.3} = 5.0891 \frac{rad}{s} \quad (4.14)$$

$$\omega_d = \omega_n \sqrt{1 - \xi^2} = 5.0891 \sqrt{1 - (0.6)^2} = 4.0713 \frac{\text{rad}}{\text{s}} \quad (4.15)$$

Polos deseados:

$$\begin{aligned} s_{1,2} &= -\xi\omega_n \pm j\omega_d \\ s_{1,2} &= -3.0531 \pm j4.0713 \end{aligned} \quad (4.16)$$

Polos deseados en tiempo discreto:

$$\begin{aligned} z_{1,2} &= e^{-0.01 \cdot 3.0531} e^{\pm j(0.01 \cdot 4.0713)} \\ z_{1,2} &= 0.9699 \angle \pm 2.3327^\circ = -0.6695 \pm j0.7017 \end{aligned} \quad (4.17)$$

Para realizar el control mediante el método del LGR se pasa la función de transferencia obtenida de tiempo continuo a tiempo discreto con un tiempo de muestreo de 0.01 segundos y un retén de orden cero, obteniendo como resultado la siguiente función de transferencia en el dominio z .

$$G(z) = \frac{-3.016 \times 10^{-6} z^2 + 1.79 \times 10^{-8} z + 2.998 \times 10^{-6}}{z^3 - 2.977 z^2 + 2.959 z - 0.9822} \quad (4.18)$$

Se obtiene la ecuación característica del sistema para extraer sus polos y determinar si los polos en lazo abierto se encuentran dentro del círculo unitario en el plano z .

$$\lambda_1 = 0.9950 \pm 0.0718i, \quad y \lambda_2 = 0.9869 \quad (4.19)$$

Con ayuda de MATLAB[®] se observa en la Figura 4.11 cómo se comportan los polos y ceros del modelo del sistema.

Los polos se encuentran dentro del círculo unitario y se puede observar que son críticamente inestables. Para lograr la estabilidad del sistema se diseña un control PD mediante el LGR. Ver Figura 4.11. Dicho control tiene la siguiente estructura:

$$C(z) = K \frac{z - C_c}{z} \quad (4.20)$$

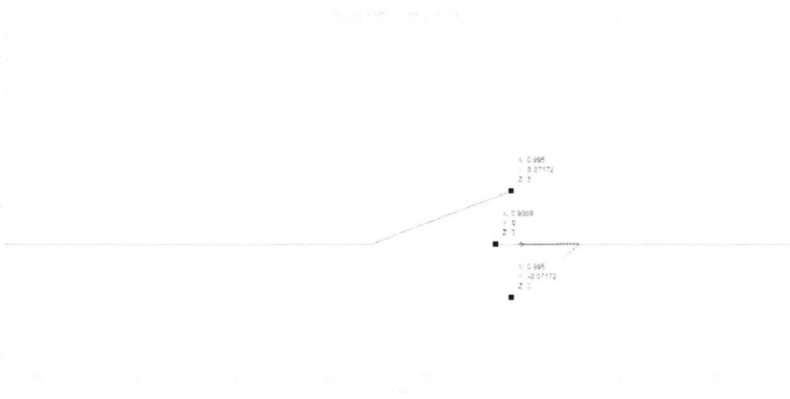


Figura 4.11: Polos en lazo abierto en el dominio z

donde:

$$K = k + k_D; C_c = \frac{k_D}{k + k_D}$$

Análisis gráfico en el plano complejo, Figura 4.12.

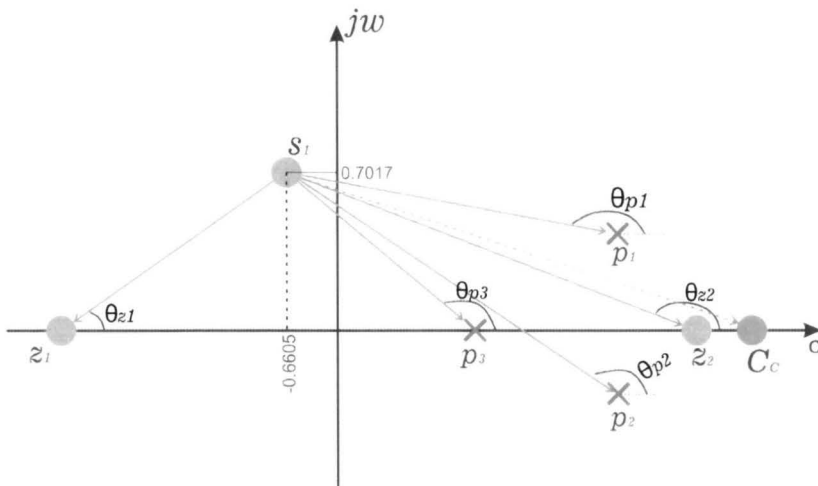


Figura 4.12: Polos y ceros del sistema representados en el plano z

Tomando en cuenta las condiciones para el diseño del LGR, se procede a sumar todos los ángulos de los ceros y todos los ángulos de los polos, para hacer el despeje correspondiente y determinar la posición del cero del control como se muestra en la Figura 4.13.

$$\begin{aligned}
 \theta_{p1} &= \tan^{-1} \left(\frac{0.7017 - 0.0718}{0.9950 + 0.6695} \right) = 0.3620 \\
 \theta_{p2} &= \tan^{-1} \left(\frac{0.7017 + 0.0718}{0.9950 + 0.6695} \right) = 0.4352 \\
 \theta_{p3} &= \tan^{-1} \left(\frac{0.7017}{0.9869 + 0.6695} \right) = 0.4007 \\
 \theta_{z1} &= \tan^{-1} \left(\frac{0.7017}{0.9941 - 0.6695} \right) = 1.1375 \\
 \theta_{z2} &= \tan^{-1} \left(\frac{0.7017}{1 + 0.6695} \right) = 0.3978
 \end{aligned} \tag{4.21}$$

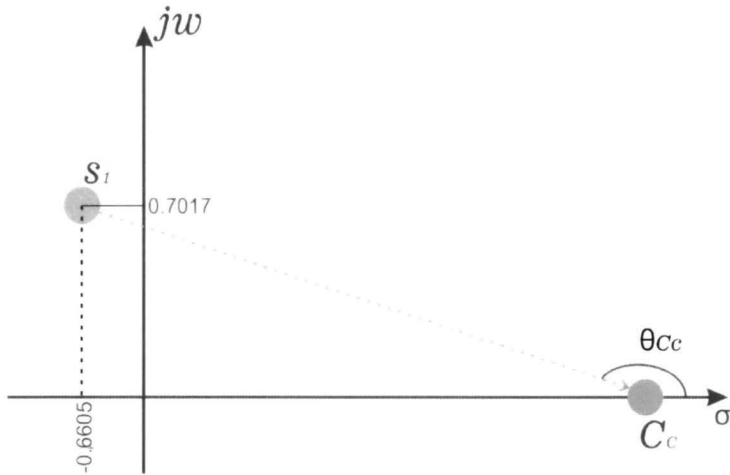


Figura 4.13: Diseño del cero del control PD

Ángulo complementario $180 - 179.6625 = 0.3375$

$$\begin{aligned}
 x &= \frac{0.7017}{\tan 0.3375} \\
 x &= C_c = 1.33
 \end{aligned} \tag{4.22}$$

Condición de magnitud para determinar el valor de K

$$\begin{aligned}
 1 &= \left| K \frac{(z - 1.33)(z - 1)(z + 0.9940)}{z(z - 0.9869)(z - 0.9950 - j0.0718)(z - 0.9950 + j0.0718)} \right| \\
 K &= 1.929
 \end{aligned} \tag{4.23}$$

con lo cual nos da un control:

$$C(z) = 1.929 \frac{z - 1.33}{z} \quad (4.24)$$

Por lo tanto, la ecuación en diferencias del controlador es:

$$u[k] = 1.929e[k] - 2.5548[k - 1] \quad (4.25)$$

La interfaz gráfica de MATLAB[®], permite corroborar la eficiencia de los cálculos obtenidos anteriormente. El control PD se lleva a la simulación con la herramienta *rltool*. Con dicha herramienta se puede manipular la ubicación de los polos y ceros así como también es posible agregar los mismos para obtener la mejor respuesta del sistema como se muestra en la Figura 4.14.

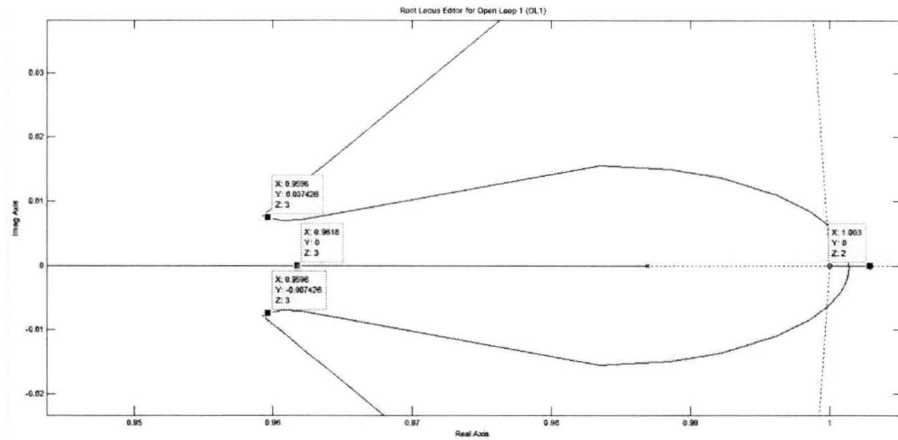


Figura 4.14: Ubicación de polos y ceros del sistema ISEIWAY con control PD

Para el sistema se agregó un cero fuera del círculo unitario ubicado en 1.003, de esta forma el cero agregado permite que los polos tengan una mejor dinámica. El análisis anterior permite obtener el siguiente control:

$$C(z) = -16887.5129(z - 1.003) \quad (4.26)$$

Teniendo como respuesta del sistema la gráfica que se muestra en la Figura 4.15.

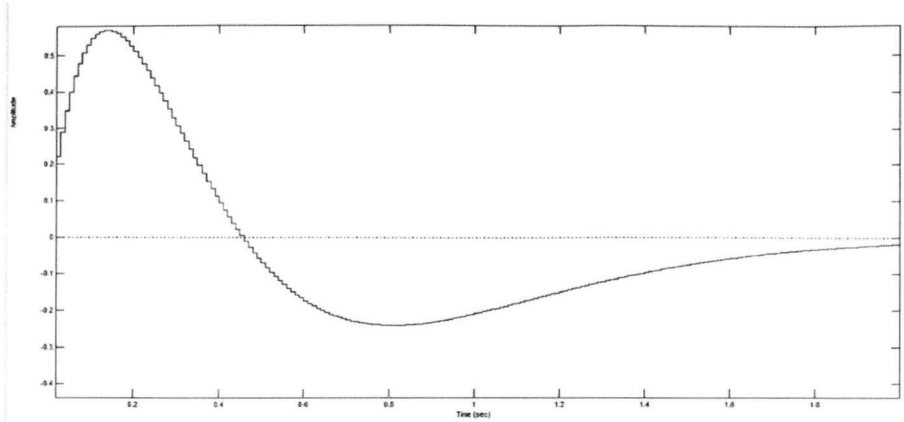


Figura 4.15: Respuesta del control PD del sistema ISEIWAY en simulación

4.4. Diseño del observador adaptable

Con base en el sistema controlado y discretizado, se diseña un observador adaptable, se acopla al sistema en paralelo para estimar algunos parámetros desconocidos. Para diseñar el observador adaptable lineal mostrado en el Capítulo 2, se reescribe el modelo de la siguiente forma, agregando la matriz de diseño $L_d \in \mathfrak{R}^{n \times p}$ que garantice la estabilidad del observador.

$$\begin{aligned} x[k+1] &= A_d x[k] + B_d u[k] + L_d y - L_d y \\ y &= C_d x[k] \end{aligned} \quad (4.27)$$

$$x[k+1] = (A_d - L_d C_d) x[k] + B_d u[k] + L_d y \quad (4.28)$$

Reescribiendo la Ec. (4.21)

$$\begin{aligned} x[k+1] &= A_k x[k] + \psi[u, y] \theta[k] \\ y &= C_d x[k] \end{aligned} \quad (4.29)$$

Donde L_d es una matriz de diseño que posiciona a los polos de $(A_d - L_d C_d)$ en algún lugar del diseño donde se asegure la rápida convergencia del error de estimación a cero.

Para realizar el diseño de L_d se usa la instrucción *place* del software de MATLAB®. Los polos agregados fueron obtenidos de tal forma que no salgan del círculo unitario y así el sistema no se inestabilice.

Como se menciona en el Capítulo 2.5, un observador permite estimar los parámetros desconocidos de un sistema [9]. Suponiendo que no se conocen todos los parámetros de:

$$\begin{aligned} B_k u[k] + L_d y[k] &= \psi \theta[k] \\ \begin{bmatrix} \beta_1 \\ 0 \\ \beta_2 \end{bmatrix} u + \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} y &= \begin{bmatrix} \beta_1 u + L_1 y \\ L_2 y \\ \beta_2 u + L_3 y \end{bmatrix} \end{aligned} \quad (4.30)$$

Donde $\theta \in \mathfrak{R}^p$ es un vector de parámetros desconocidos (regresor) y está dado por:

$$\theta = \begin{bmatrix} \beta_1 \\ L_1 \\ L_2 \\ \beta_2 \\ L_3 \end{bmatrix} \quad (4.31)$$

De manera que $\psi [u, y] \in \mathfrak{R}^n \times \mathfrak{R}^p$ es una matriz de señales conocidas de la forma:

$$\psi = \begin{bmatrix} u & y & 0 & 0 & 0 \\ 0 & 0 & y & 0 & 0 \\ 0 & 0 & 0 & u & y \end{bmatrix} \quad (4.32)$$

Por lo tanto el modelo correspondiente es:

$$\begin{aligned} x[k+1] &= A_k x[k] + \psi \theta[k] \\ y[k] &= C_k x[k] \end{aligned} \quad (4.33)$$

De esta manera el observador adaptable para el nuevo sistema está descrito de la siguiente forma:

$$\begin{aligned} \Upsilon_{k+1} &= (A_k - K_k C_k) \Upsilon_k + \Psi_k \\ \hat{\theta}_{k+1} &= \hat{\theta}_k + \mu_k \Upsilon_k^T C_k^T (y_k - C_k \hat{x}_k) \\ \hat{x}_{k+1} &= A_k \hat{x}_k + \Psi_k \hat{\theta}_k + K_k (y_k - C_k \hat{x}_k) + \Upsilon_{k+1} (\hat{\theta}_{k+1} - \hat{\theta}_k) \end{aligned} \quad (4.34)$$

5

Resultados

Una vez desarrollado el control PD para el sistema, se procede a verificar por medio de la simulación el desempeño del mismo, de igual forma se realiza la simulación para el observador adaptable lineal.

5.1. Resultados en simulación

Las siguientes simulaciones se llevaron a cabo mediante MATLAB® y SIMULINK® con la finalidad de comprobar el comportamiento del **ISEIWAY** y verificar la eficiencia del controlador y el observador. Tomando como referencia que el sistema se encuentra fuera del equilibrio respecto a un punto deseado, la simulación permite corroborar dicha inestabilidad al visualizar el comportamiento a la salida, en donde se tiene como entrada un escalón unitario con valor final de 0.001 segundos, por lo que se observa en la respuesta del sistema un comportamiento de inestabilidad para la posición del vehículo. Ver Figura 5.1

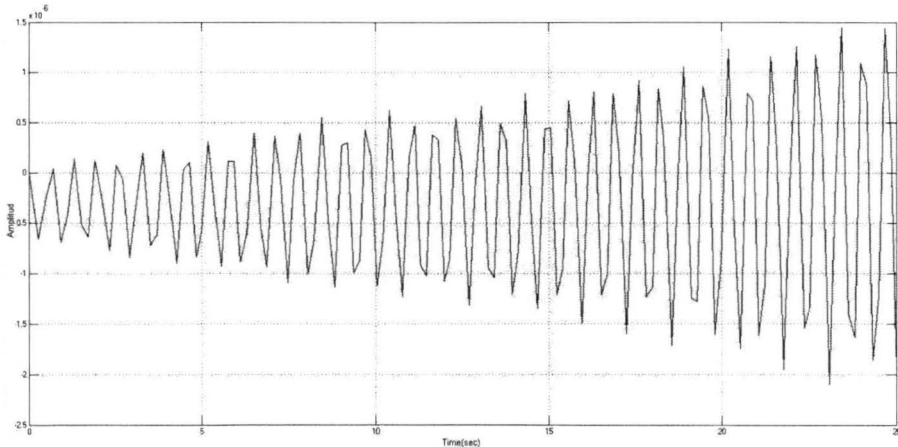


Figura 5.1: Posición del sistema

En el Apéndice A se presenta el código utilizado en cada bloque de la Figura 5.2. Para un análisis más profundo se procede a implementar un control PD para

verificar la eficiencia de este control sobre el sistema. El diagrama de la Figura 5.2 describe la estructura del control PD, así como la del observador adaptable lineal.

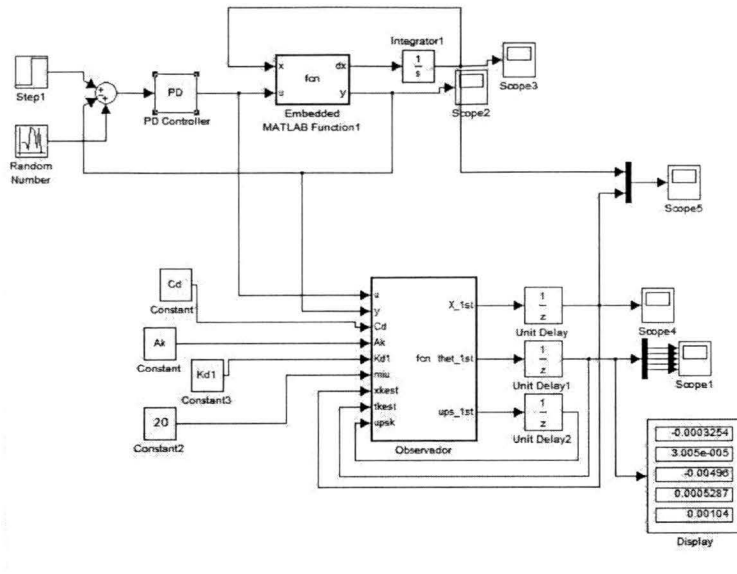


Figura 5.2: Diagrama de bloques del control PD y el observador

En la simulación de la Figura 5.3 se puede observar la convergencia de la posición del péndulo hacia un punto deseado (el cero).

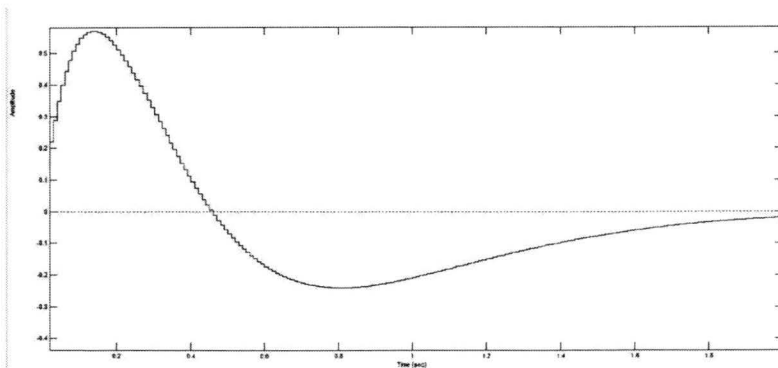


Figura 5.3: Desempeño del control PD para la posición del sistema

El desempeño de los estados afectados por el control PD y el observador adaptable se muestran en la Figura 5.4, la Figura 5.5 y la Figura 5.6.

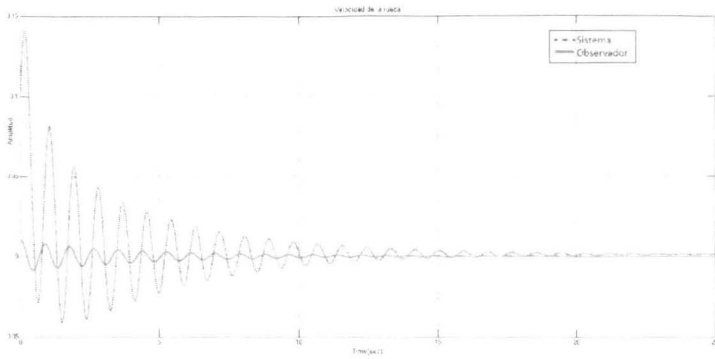


Figura 5.4: Desempenho do observador e do controle PD para a velocidade de as rodas

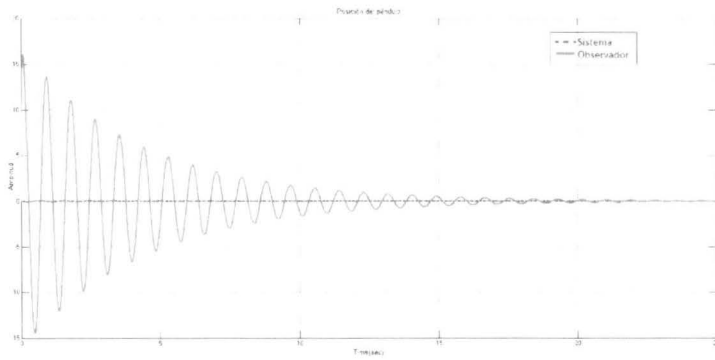


Figura 5.5: Desempenho do observador e do controle PD para a posição do sistema

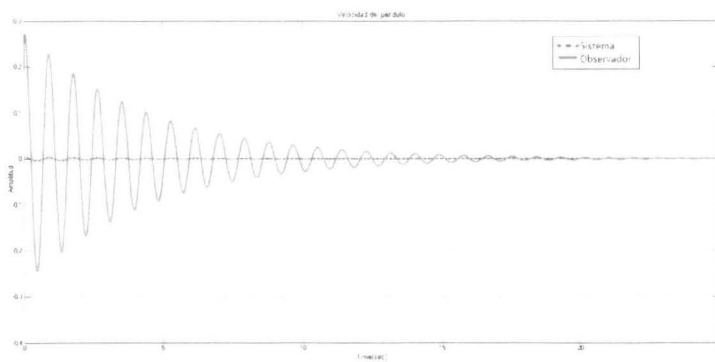


Figura 5.6: Desempenho do observador e do controle PD para a velocidade do sistema

Las Figura 5.7, la Figura 5.8, la Figura 5.9, la Figura 5.10 y la Figura 5.11 muestran la convergencia del vector de parámetros desconocidos (regresor).

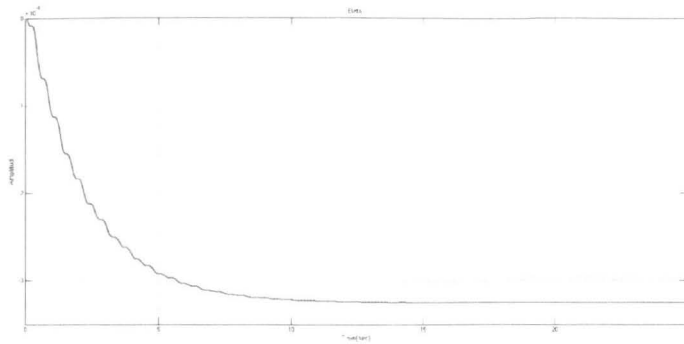


Figura 5.7: Desempeño del observador y del control PD para el parámetro β_1

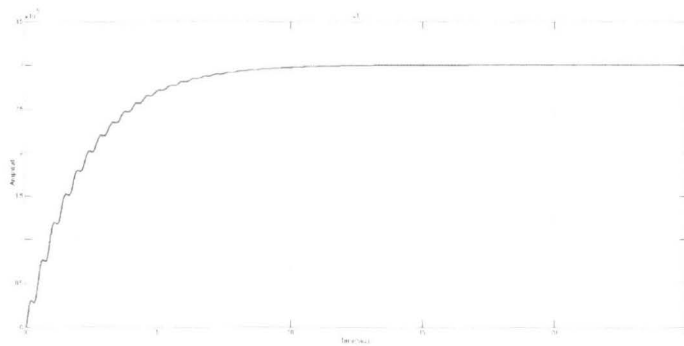


Figura 5.8: Desempeño del observador y del control PD para el parámetro $L1$

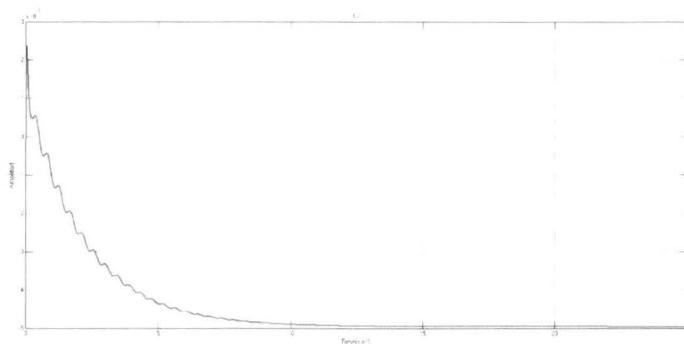


Figura 5.9: Desempeño del observador y del control PD para el parámetro $L2$

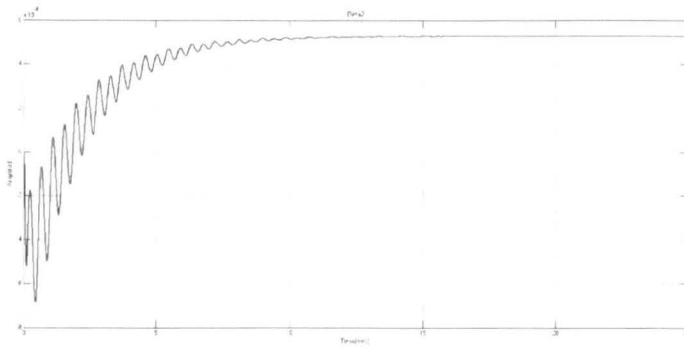


Figura 5.10: Desempenho do observador e do controle PD para o parâmetro β_2

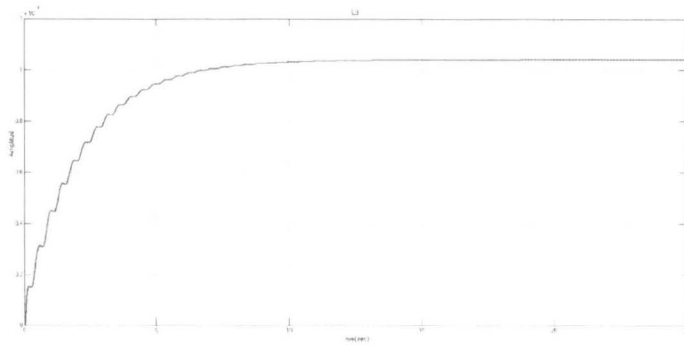


Figura 5.11: Desempenho do observador e do controle PD para o parâmetro L3

El Apéndice A proporciona el contenido del código utilizado en cada bloque de la Figura 5.12. Se realiza un control LQR con la finalidad de comparar el desempeño entre éste y el control PD sobre el mismo sistema. Usando SIMULINK® se diseña un diagrama de bloques del sistema añadiendo un control LQR para comparar la convergencia de los estados a un punto deseado. Ver Figura 5.12.

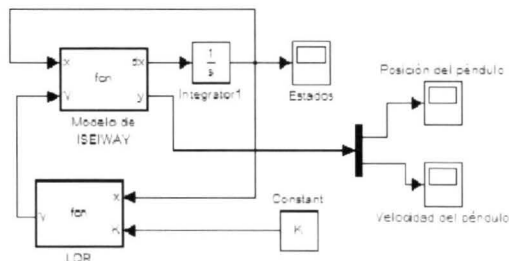


Figura 5.12: Diagrama de bloques do controle LQR em Simulink

La simulación de la Figura 5.13 representa la posición del sistema **ISEIWAY** con el desempeño de un control LQR, se observa que la posición del sistema converge a cero en un tiempo menor comparandolo con el control PD.

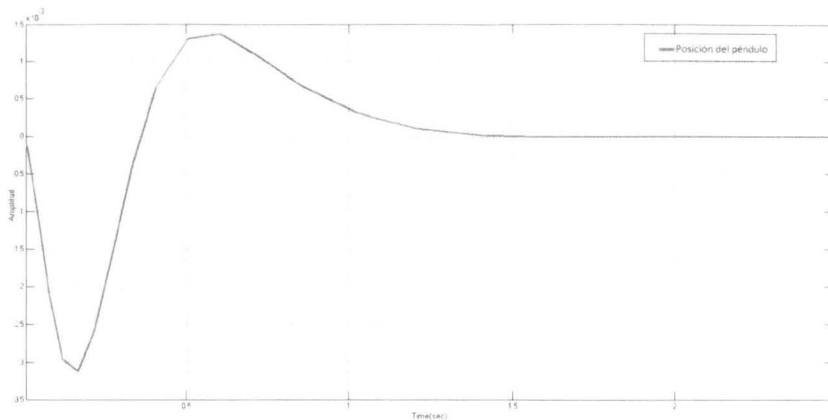


Figura 5.13: Salida de la posición con control LQR

El sistema controlado puede apreciarse en la gráfica de la Figura 5.14, donde los estados como la velocidad y posición del sistema convergen al punto deseado (a cero) en aproximadamente un segundo.

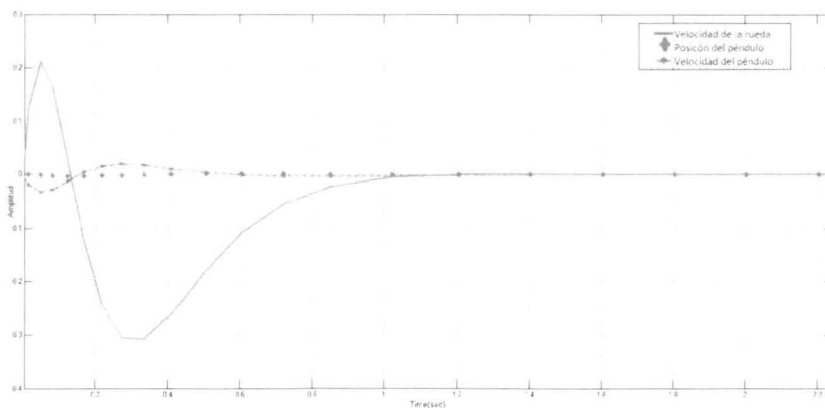


Figura 5.14: Estados del sistema ISEIWAY con control LQR

El diseño del observador adaptable lineal es añadido al diagrama de bloques del sistema controlado, esto con ayuda de SIMULINK® y con la finalidad de verificar que tan bien puede reconstruir los estados. El Apéndice A proporciona el contenido del código utilizado en cada bloque de la Figura 5.15.

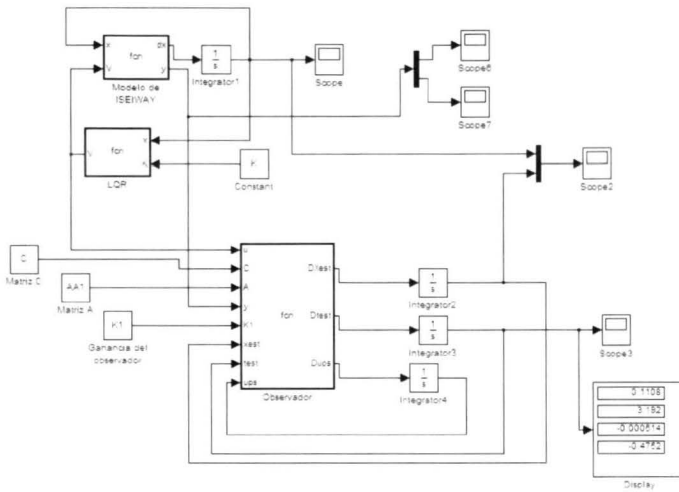


Figura 5.15: Diagrama de bloques del control LQR con el Observador Adaptable

El resultado de la reconstrucción de estados garantiza un buen diseño del control adaptable. Se puede verificar su desempeño en la gráfica de la Figura 5.16.

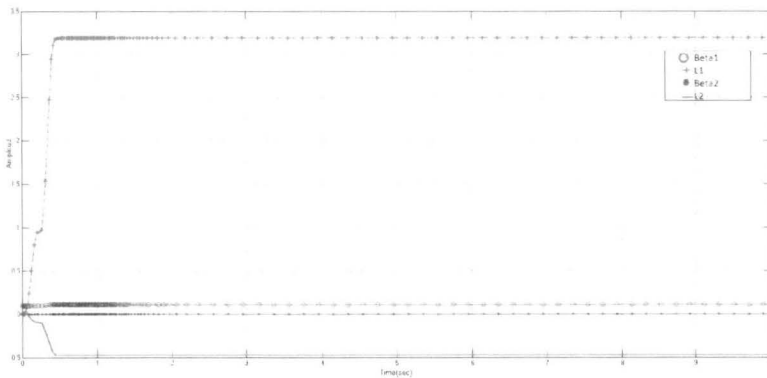


Figura 5.16: Desempeño del observador adaptable

En la Figura 5.17, la Figura 5.18 y la Figura 5.19, se hace la comparación entre los estados del sistema y los estados observados, de igual forma es posible apreciar que el desempeño del observador adaptable lineal es bastante eficiente.

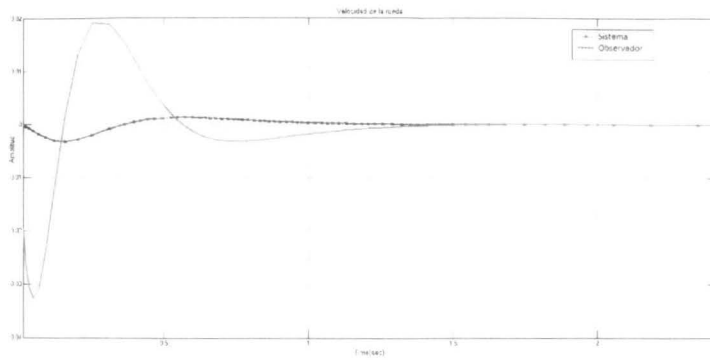


Figura 5.17: Análisis entre el sistema y el observador para la velocidad de la rueda.

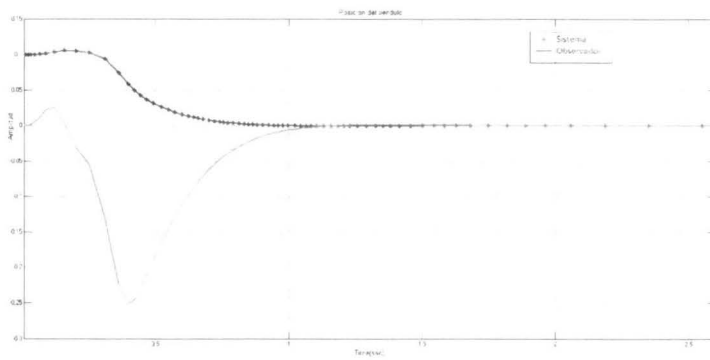


Figura 5.18: Análisis entre el sistema y el observador para la posición del vehículo.

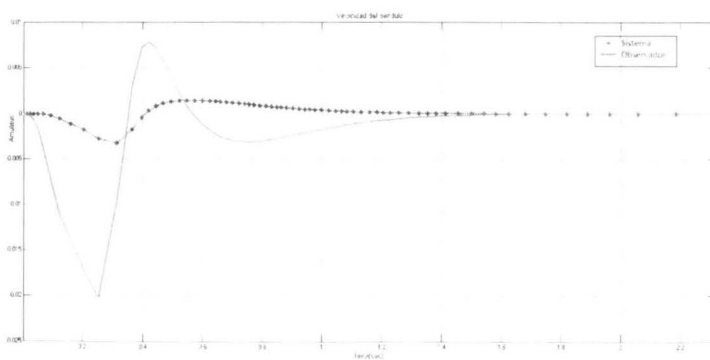


Figura 5.19: Análisis entre el sistema y el observador para la velocidad del Vehículo.

5.2. Resultados experimentales

La identificación de sistemas es una forma de conocer los parámetros de una planta, para este caso un motor. El análisis se lleva a cabo para determinar los parámetros de un motor de 24V, 450Watts y una velocidad de 420rpm (revoluciones por minuto). La necesidad de conocer el modelo matemático que describe el comportamiento del motor es fundamental para el manejo óptimo del mismo. Se describe la técnica de construir un modelo a partir de; las variables medidas del proceso: entradas o variables de control, salidas o variables controladas y perturbaciones. En principio el motor es visto como una caja negra, esto debido a que los parámetros del modelo no tienen una interpretación física. Un motor de DC de imanes permanentes es una topología comúnmente usado para proporcionar movimiento a sistemas mecánicos. Para ello se realizan las pruebas para la identificación de parámetros.

Procedimiento:

Se realiza el acoplamiento del motor DC con ayuda del eje de rotación, se alinea la flecha del motor de interés a la flecha de un segundo motor que servirá de apoyo para realizar las pruebas necesarias, una vez que se tienen estos dos motores acoplados, se procede a realizar una serie de experimentos que permitirán la identificación de parámetros del motor DC para el sistema. Esto se muestra en la Figura 5.20.

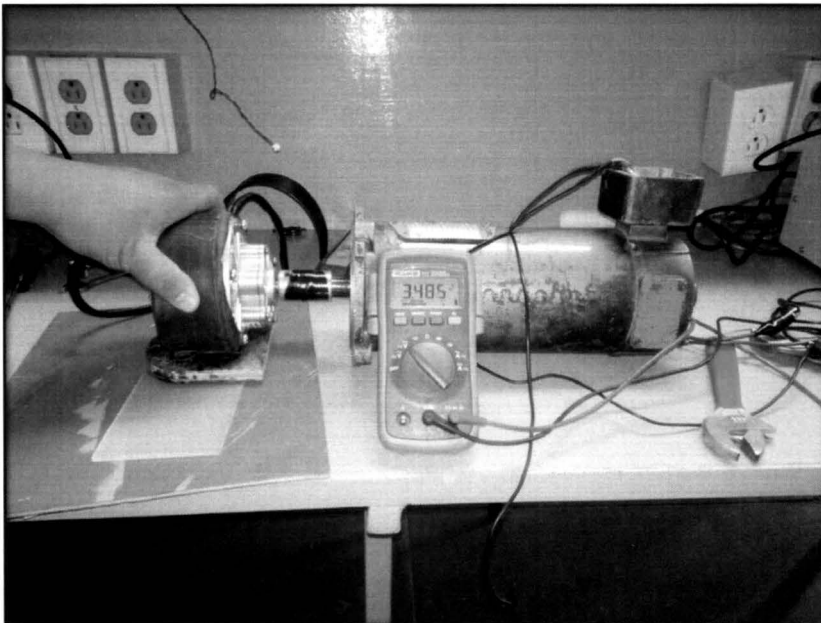


Figura 5.20: Acoplamiento entre el motor DC del sistema ISEWAY y el motor DC para pruebas

Parámetros del motor DC

La medición de la resistencia de armadura se puede hacer mediante un multímetro conectado al devanado de la armadura (cable rojo y negro).

$$R_a = 1.14\Omega \quad (5.1)$$

Otra forma de medir la resistencia de armadura es aplicando un voltaje mínimo de DC (0.94V) de tal manera que se mida la corriente de armadura y se aplica la ley de ohm.

$$V = IR_a \implies R_a = \frac{V}{I} \implies R_a = \frac{0.94V}{0.96I} = 0.98\Omega \quad (5.2)$$

Se observa que es muy similar el valor de la resistencia R_a en ambos métodos, por otro lado se requiere el calculo de la inductancia. En forma experimental se coloca un LCR/ESR meter para obtener la inductancia. Ver Figura 5.21.

$$L_a = 236\mu H. \quad (5.3)$$

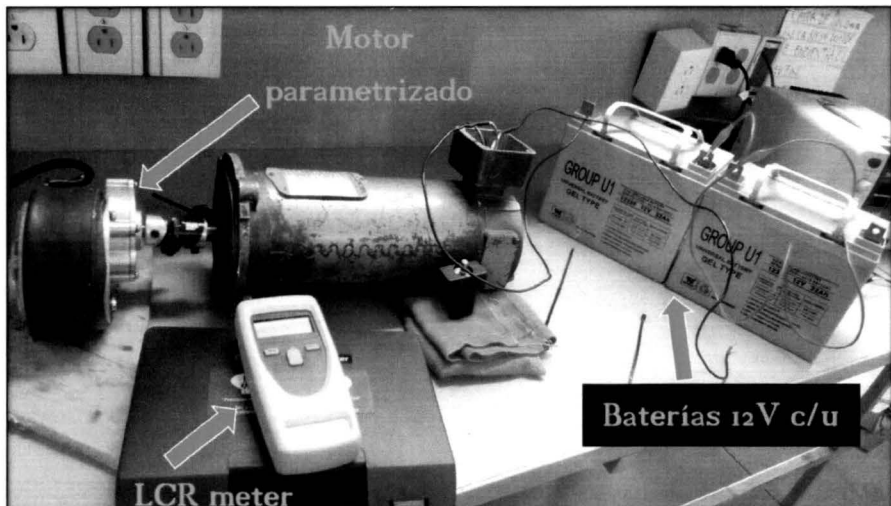


Figura 5.21: Parametrización del motor de DC de 24V

5. Resultados

Para calcular la constante de torque se procede a acoplar la flecha de un motor de DC de 90V con la flecha del motor a parametrizar, el voltaje se fue variando por medio de una fuente de poder variable (Bk Precision 1673). Se tomaron diferentes medidas de voltaje en las terminales del motor de interés y se midió la velocidad en rpm (revoluciones por minuto) mediante un tacómetro, el Cuadro 5.1 muestra los datos obtenidos y la relación entre ellos. Ver Cuadro 5.1.

Voltaje aplicado [V]	Voltaje generado [V]	Velocidad [RPM]	Velocidad ang. [Rad/seg]
6.8	0.62	14.07	1.47
8.2	1.68	36.81	3.85
10.1	3	68.76	7.20
12	4.68	107.3	11.23
14	6.095	139.2	14.57
16	7.77	177.5	18.58
18	9.45	216.5	22.67
20	11.4	262.3	27.46
22	13.6	313.3	32.8
24	15.4	354.3	37.10

Cuadro 5.1: Tabla valores experimentales del motor dc.

La gráfica de la Figura 5.21 muestra el voltaje generado contra la velocidad angular y de ahí se calcula la pendiente de la recta que representará la constante de torque.

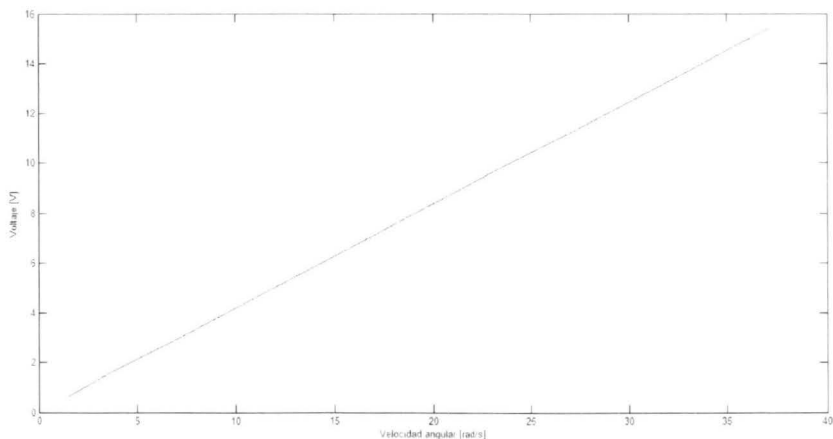


Figura 5.22: Gráfica de constante de torque

$$K_i = 0.4139 \frac{V}{rad/s} \quad (5.4)$$

5.2.1. Modelo a escala

En el modelo a escala se utilizaron diversos elementos como son: dos motores de 12V a los cuales se les acopló una rueda a cada uno de ellos, los soportes de cada nivel se hicieron de acrílico y se adaptaron con un tornillo largo a una distancia simétrica de 15 cm entre un nivel y otro. Se fijaron los motores a la base principal con cinturones de plástico a una distancia intermedia entre uno y otro tomando como referencia la base del sistema. Se aseguró cada nivel de acrílico con roldanas y tuercas, dando como resultado un diseño funcional el cual permitió implementar el modelo y el control anterior en la tarjeta de desarrollo ARDUINO® que permitió hacer la comparación entre las simulaciones y un sistema real.

Los motores se alinearon lo mejor posible desde el eje de giro hacia las ruedas para permitir que las mismas giren concéntricamente respecto al eje. Se conectaron los cables de los motores de DC y se acomodaron sobre la superficie para hacer más fácil la alimentación de los mismos. La Figura 5.23 muestra el acoplamiento entre la rueda izquierda y el motor, de igual forma se observa la colocación entre éste motor y la base principal del prototipo a escala.

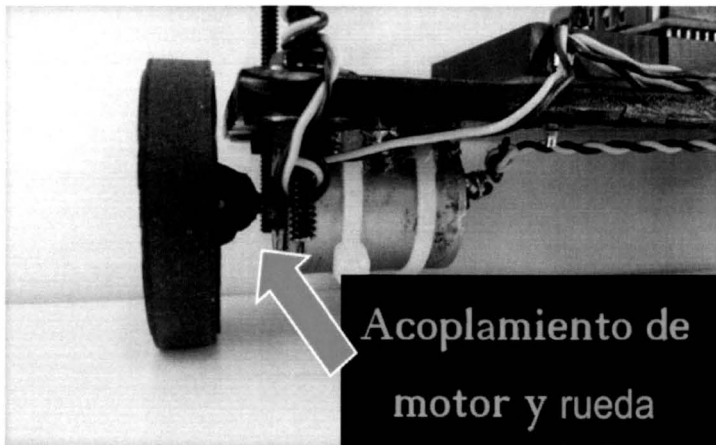


Figura 5.23: Acoplamiento motor-rueda

Los motores se ajustaron a la base del sistema considerando que el espacio sea el mismo entre éstos y el acrílico, se usaron cinturones de plástico para la sujeción y así evitar que se muevan en direcciones incorrectas. Ver Figura 5.24.

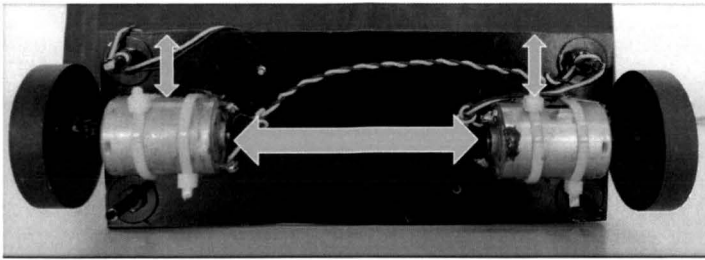


Figura 5.24: Acoplamiento motor-base

Se colocaron los componentes electrónicos atornillando cada elemento para su mejor funcionamiento como se muestra en la Figura 5.25.

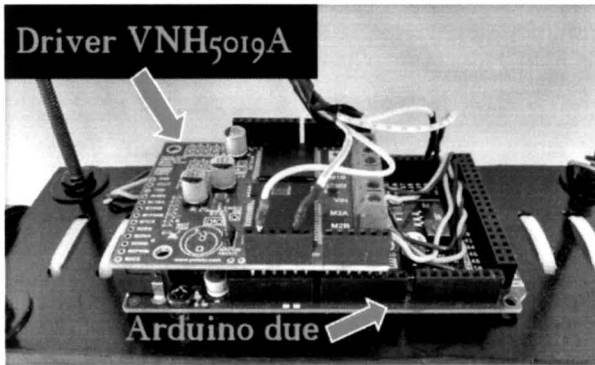


Figura 5.25: Arduino due y driver POLOLU® en el ISEIWAY-MINI

Se colocó el sensor IMU en la parte intermedia del prototipo a escala. Ver Figura 5.26.

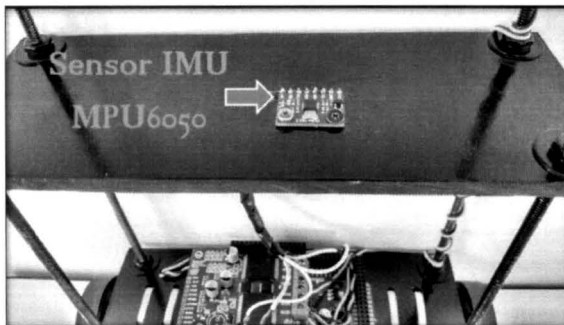


Figura 5.26: Sensor IMU MPU6050 sobre la base del ISEIWAY-MINI

Por último se adaptó la batería en la parte superior del vehículo la cual suministra el consumo total de energía al sistema como se puede observar en la Figura 5.27.

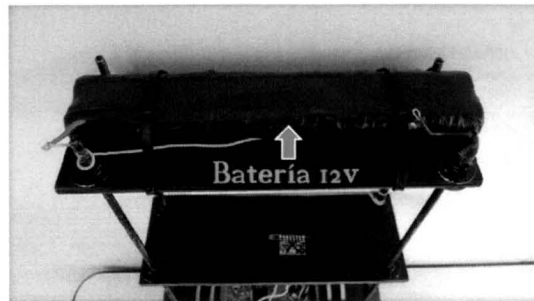


Figura 5.27: Batería de 12V

El prototipo se fue corrigiendo debido a ciertos desperfectos como el acoplamiento de las llantas, se modificaron algunos materiales para un mejor desempeño dando como resultado el diseño final que se presenta en la Figura 5.28, obteniendo el comportamiento deseado, el equilibrio del sistema a escala.

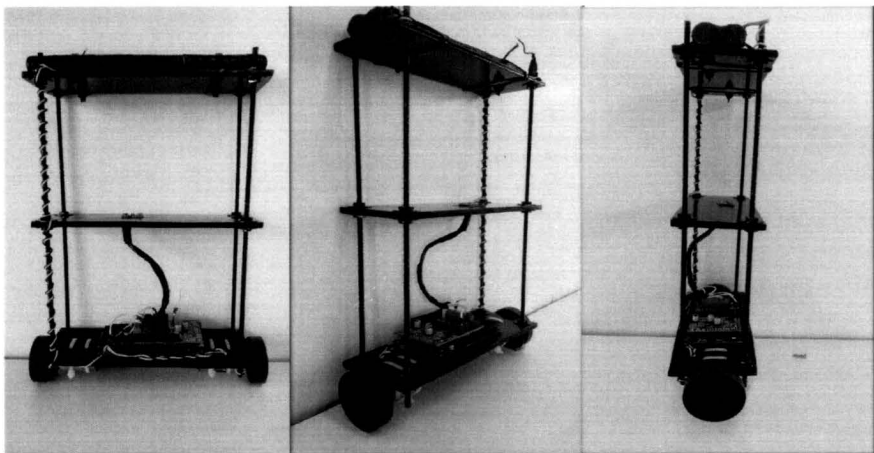


Figura 5.28: ISEIWAY mini, diseño final

5.2.2. Observador Adaptable

En el Apéndice A se describe el código utilizado para el observador adaptable implementado en la tarjeta ARDUINO® en el prototipo a escala. A continuación se

presentan las gráficas de la Figura 5.29, la Figura 5.30, la Figura 5.31 y la Figura 5.32 obtenidas de forma experimental, donde se observa la convergencia de los estados y parámetros del sistema.

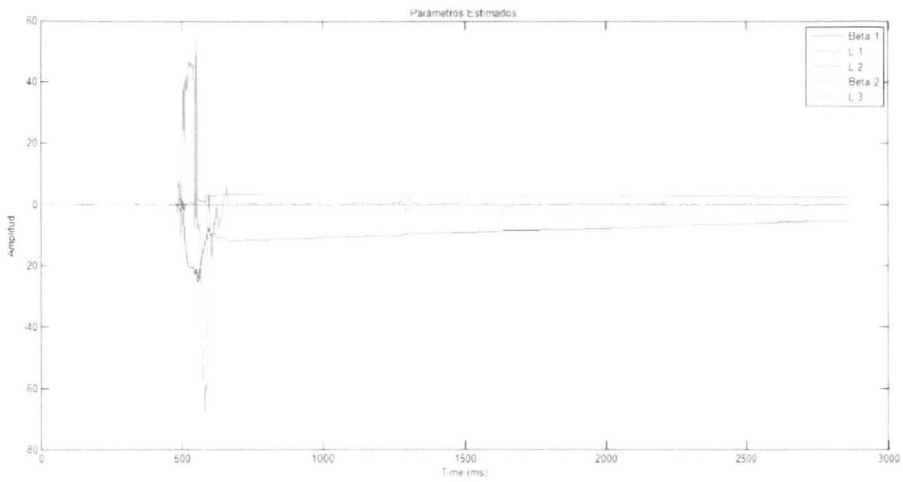


Figura 5.29: Desempeño del observador adaptable para los parámetros del prototipo ISEIWAY mini

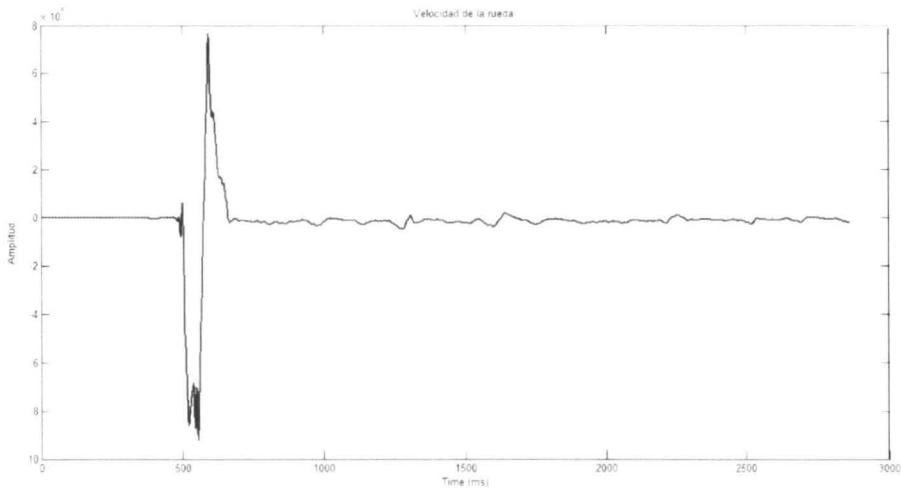


Figura 5.30: Desempeño del observador adaptable para la velocidad de las ruedas del prototipo ISEIWAY mini

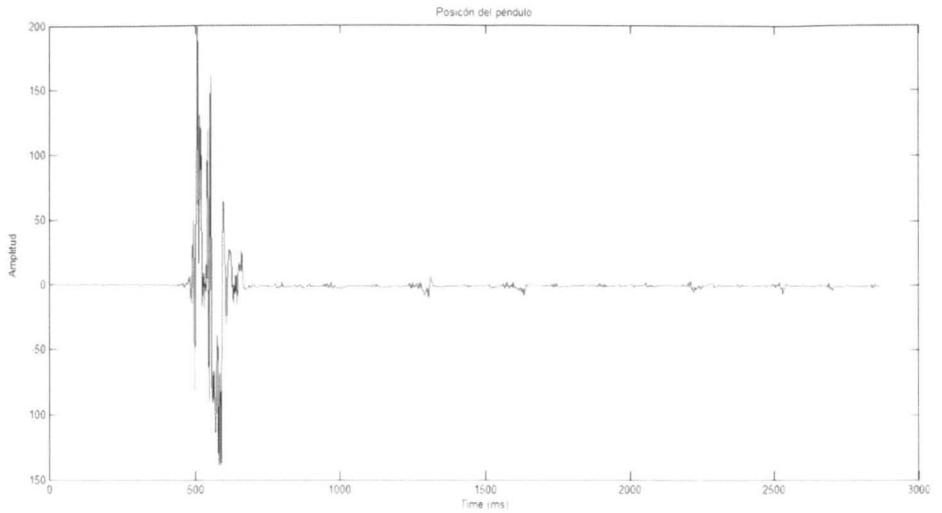


Figura 5.31: Desempeño del observador adaptable para la posición del prototipo ISEIWAY mini

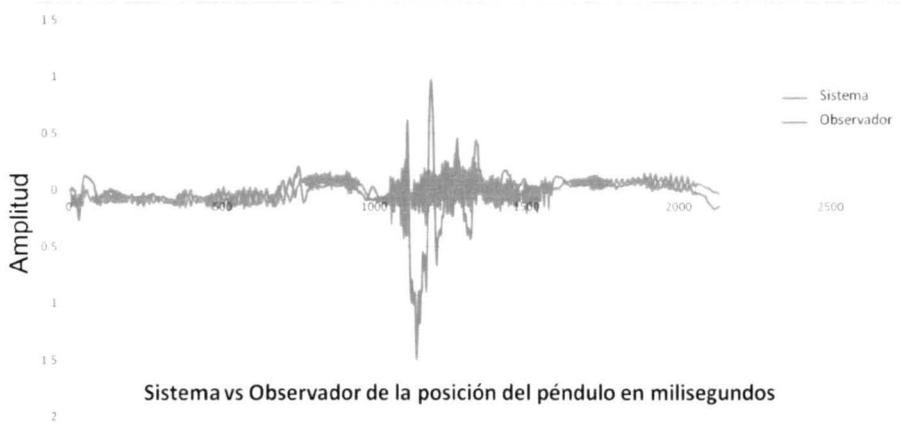


Figura 5.32: Comparación del desempeño entre el observador adaptable y la respuesta del sistema para la posición del prototipo ISEIWAY mini

5.2.3. Prototipo final

Las mejoras en el prototipo final se llevaron a cabo para garantizar una mejor funcionalidad del sistema: el ajuste de las ruedas, el manubrio, la cadena hacia el motor entre otros, fueron aspectos muy importantes a lo largo del desarrollo de este trabajo, logrando así un diseño optimizado como el que se muestra en la Figura 5.33.



Figura 5.33: Sistema ISEIWAY final

La unión de la cadena hacia el motor y las ruedas fue un problema que se logró solucionar, se realizaron varias pruebas para conseguir que el giro en ambos sentidos fuera el correcto. Se puede observar como quedó la estructura en la Figura 5.34.

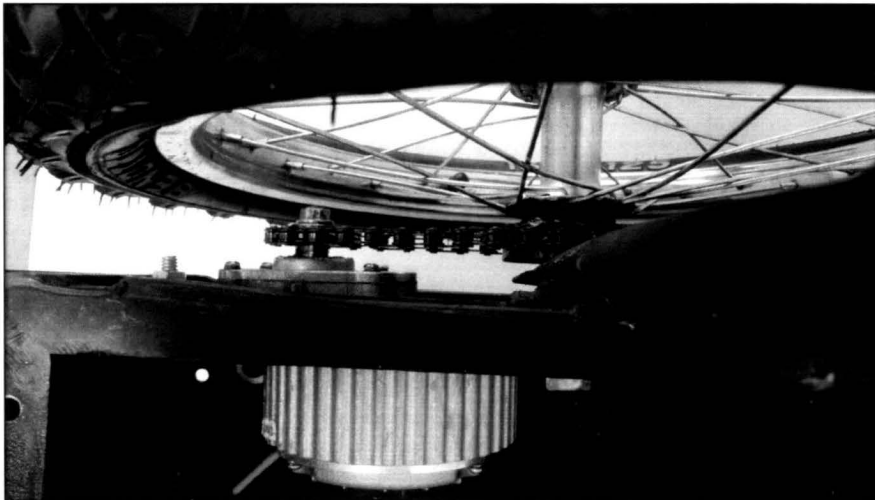


Figura 5.34: Acoplamiento motor-rueda

El sensor MPU6050 se adaptó a la base del sistema de forma estratégica y para su protección se colocó de bajo de la base que sujeta al manubrio. Se soldaron los pines correspondientes para realizar la conexión con la placa ARDUINO®, poste-

riormente se verificó la respuesta del sensor colocándolo en una posición estratégica para configurar el punto de equilibrio deseado (el cero). En la Figura 5.35 se observa la colocación de este sensor.

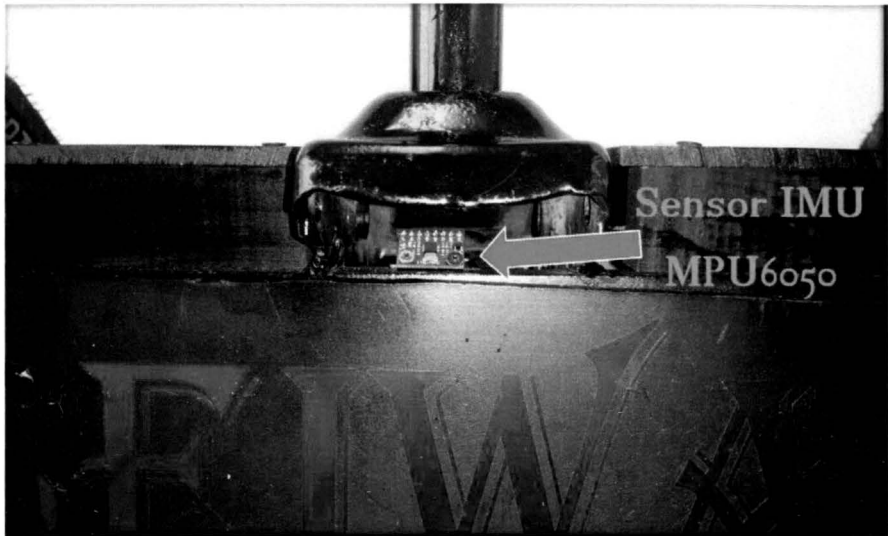


Figura 5.35: Acoplamiento sensor IMU

En la Figura 5.36 se observa el acoplamiento rueda-motor a la base.

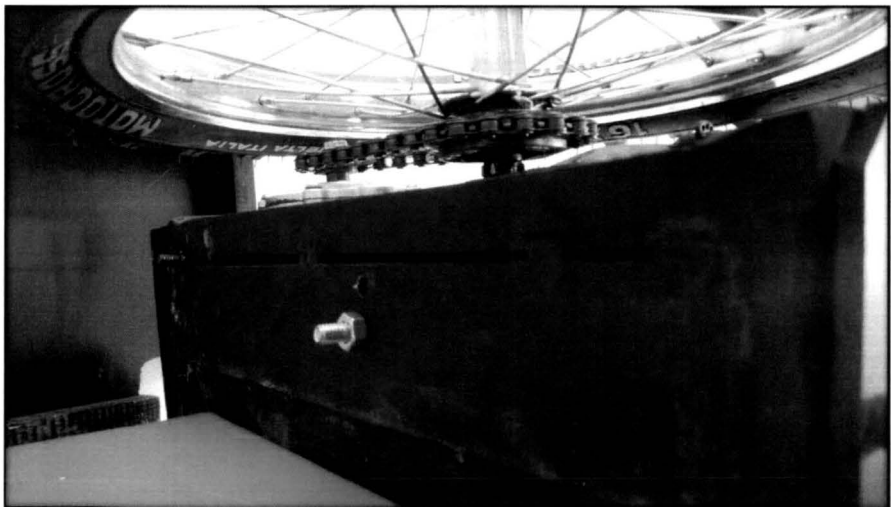


Figura 5.36: Acoplamiento de la rueda y motor a la base

En la Figura 5.37 se observa el ordenamiento de los componentes en la base del vehículo.

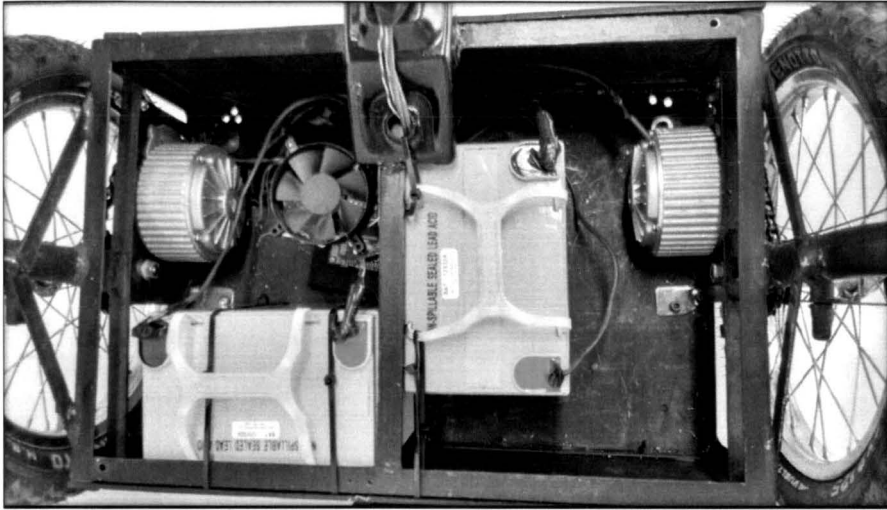


Figura 5.37: Vista superior de componentes

La Figura 5.38 muestra la conexión de la tarjeta de desarrollo ARDUINO® con el (shield) driver POLOLU® y demás componentes.

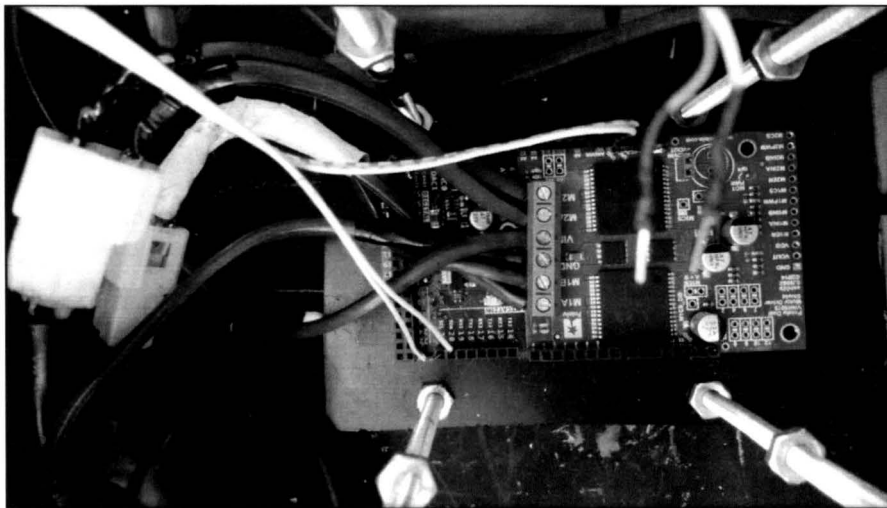


Figura 5.38: Vista superior de driver y conexión a la tarjeta Arduino Due

La Figura 5.39 muestra la adaptación de un ventilador para el enfriamiento de la etapa electrónica.

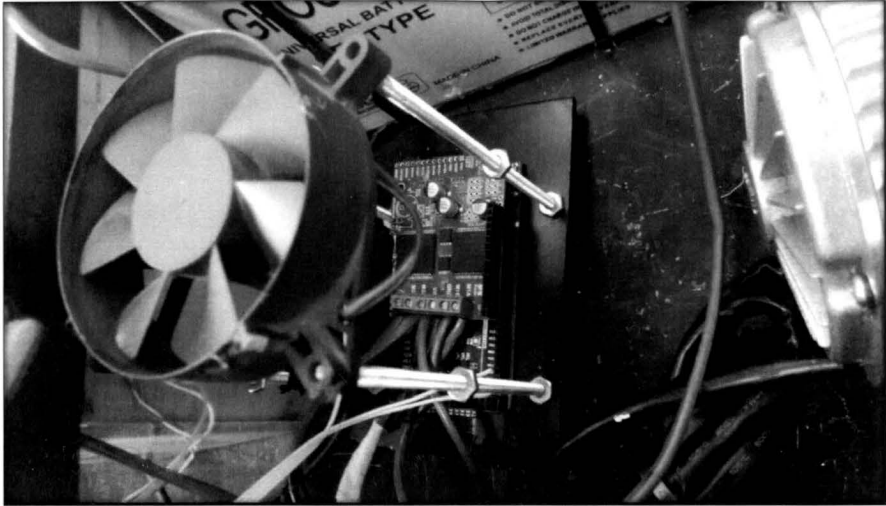


Figura 5.39: Acoplamiento de ventilador

El sistema en proceso de desarrollo se muestra en la Figura 5.40.

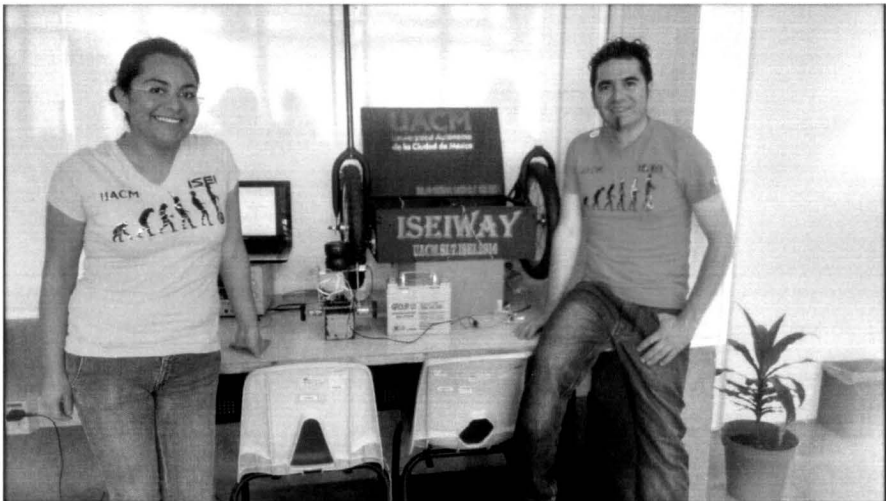


Figura 5.40: Integrantes del sistema ISEIWAY

El sistema final se presenta en la Figura 5.41.



Figura 5.41: Vista superior ISEIWAY

La Figura 5.42 y la Figura 5.43 muestran al sistema en operación.



Figura 5.42: Funcionamiento del sistema ISEIWAY



Figura 5.43: Funcionamiento del sistema ISEIWAY

6

Conclusiones

6.1. Conclusiones

El sistema como tal es inestable en el punto de equilibrio (cero grados) respecto al eje y , por lo tanto presenta un grado de dificultad considerable para la obtención del modelo matemático.

En la parte teórica se fue desarrollando el proceso matemático en conjunto con el análisis de control adquiridos a lo largo de la carrera Ingeniería en Sistemas Electrónicos Industriales. El desarrollo de dos diseños de controladores, un control PD, y un control LQR permitieron una mejor perspectiva respecto a lo requerido en el sistema, en este caso, que mantuviera su equilibrio para no caer al piso. Como se mostró en el Capítulo 5 los controles diseñados muestran su desempeño entre sí, logrando que los dos sean capaz de compensar correctamente tanto perturbaciones como variaciones en los parámetros del sistema.

Otro de los mas grandes retos fue el diseño y la construcción de la estructura en donde se instalaron los componentes mecánicos y electrónicos, posteriormente se llevaron a cabo las pruebas físicas para así determinar la funcionalidad en el prototipo real. La adaptación correcta de las llantas a los motores, la ubicación correcta del sensor IMU, así como la colocación del manubrio a la base del sistema, fueron aspectos muy importantes para garantizar que el vehículo funcione. Se enfrento a diversos factores en la fabricación de la base y la colocación de todos los componentes físicos. Al final se aumentó la altura de la base, esto debido al tamaño de las baterías.

La parte teórica obtenida a lo largo de nuestra preparación como estudiantes de ingeniería fue de gran relevancia para la realización del sistema **ISEIWAY**. Cabe destacar que las simulaciones que se llevaron a cabo en MATLAB® siempre van a presentar ciertas diferencias respecto al resultado obtenido en un sistema real, ya que las condiciones de pruebas no son las mismas, físicamente nuestros modelos o nuestros sistemas no son ideales como los que existen en una simulación.

El diseño de un observador adaptable lineal fue uno de los principales objetivos en este trabajo, ya que la finalidad de dicho observador permite analizar y estimar a profundidad parámetros o estados ocultos en un sistema dado. Se puede decir que aplicando un observador adaptable muy bien estructurado a algún sistema que lo

permita, será muy posible adquirir los parámetros que a simple vista son complicados de determinar.

6.2. Trabajo futuro

El sistema puede mejorarse sin duda alguna, en especial para disminuir el peso del vehículo y hacerlo más ligero y cómodo, se ha pensado adaptar un asiento para mayor comodidad y evitar ir de pie en largos recorridos. Este análisis puede aplicarse a prototipos semejantes como una silla de ruedas, un monociclo, o un patín, tomando las consideraciones electrónicas, los controladores a usar y los sensores correspondientes. Esto puede ser de gran apoyo para personas con capacidades especiales o individuos que se encuentran en rehabilitación en algún hospital.

El prototipo inicial pretende dar un gran salto al innovar en un sistema de dos ruedas de tamaño real, que permita trasladar a los usuarios de un punto a otro con tan solo manipular su dirección simplemente con inclinarse un poco o presionar el sentido de giro para la vuelta.

Una característica a futuro, pudiera ser el uso de cámaras de seguridad acondicionadas al vehículo para que se permita grabar sucesos que afecten a nuestra sociedad y así capturar imágenes y/o videos de todo lo que se presente al frente del sistema **ISEIWAY**. Otra característica sería adaptar una etapa que permita enviar todos los datos capturados a algún servidor en forma remota.

6.3. Resultados

Las diferentes fuerzas que interactúan entorno al sistema fueron analizadas minuciosamente y esto fue un gran reto ya que las primeras simulaciones no arrojaron los resultados esperados. Las fuerzas de fricción y la inercia del sistema se lograron calcular y así cubrir los objetivos planteados en el proceso de la identificación de parámetros teniendo esa gran ventaja para obtener el modelo del sistema. Los análisis tanto en simulación como experimentales proporcionaron un gran avance para lograr así la culminación de este trabajo de forma satisfactoria.

El sistema **ISEIWAY** logra mantener el autobalance, tanto en el sistema grande como en el pequeño, con tan solo usar un sensor IMU y no incluir el uso de encoders, podemos concluir que se logró el objetivo deseado usando solamente un giroscopio y un acelerómetro como sensores.

El diseño del control PD funciona correctamente y está garantizado en un intervalo de $\pm 20^\circ$. Si el sistema se sale del rango definido, el sistema deja de funcionar desactivando los motores para evitar accidentes.

La velocidad del sistema se limitó por motivos de seguridad.

El sistema se desarrolló en un tiempo de 11 meses con un horario de 25 horas a la semana aproximadamente.

El costo de dicho sistema puede aproximarse a los \$22,000.00 pesos, esto debido a gastos de envío e impuestos de aduana. Se podría disminuir el costo utilizando componentes más económicos pero no se garantiza un desempeño óptimo hasta realizar las pruebas correspondientes.

7

Apendice

MATLAB®

Archivo .m para la identificación del motor.

```
clc
%clear all
close all

%***** Datos de entrada *****

V = 12;           % Voltaje que se le aplica al motor (Escalon).
vf = 221.1;      % Velocidad del motor cuando el tiempo tiende al
                % infinito.
tau = 1.69;      % Constante de tiempo cuando alcanza el 63.2%
R = 10.462;
L = 4.52e-3;

%*****

vf_63_3 = (vf)*(.632);
K_cosc_B = vf/V;
B = ( log(1-(vf_63_3/vf)) ) / (-tau);
K = (K_cosc_B)*(B);

Kbf = B/K;
J = Kbf/(K*R);
feq = 0; % Coeficiente de friccion viscosa, se considera cero.

% Funcion de transferencia de primer orden
num1 = [K];
den1 = [1 B];
Gs1 = tf(num1,den1)
% Funcion de transferencia de segundo orden (Aproximacion).
num2 = [0 0 Kbf];
den2 = [L*J (R*J + L*feq) (feq*R + (Kbf)^2)];
Gs2 = tf(num2,den2)

A = [V;vf;tau;R;L;vf_63_3;B;K;Kbf;J;]
```

Archivo .m del modelo para el sistema ISEWAY.

```

syms V g R r m1 m2 I1 I2 l Kt Kw Jm Ra B ddth1 dth1 th1 ddth2 dth2 th2

Eq=[(((m1+m2)*R^2)+I1+(Jm*Ra*r^2))*ddth1+(m2*R*l*cos(th2))*ddth2+((r*B)
+(Kw*Kt*r/Ra))*r*dth1-((r*B)+(Kw*Kt*r/Ra))*dth2-(V*r*Kt/Ra);((m2*R*
l*cos(th2))-(Jm*Ra*r^2))*ddth1+((m2*l^2)+I2)*ddth2-(m2*g*l)*th2-((r
*B)+(Kw*Kt*r/Ra))*r*dth1+((r*B)+(Kw*Kt*r/Ra))*dth2+((r*Kt)/Ra)*V];

[DDth1 DDth2]=solve(Eq,ddth1,ddth2);

f2=simplify(DDth1);
f4=simplify(DDth2);

%% Parametros
g=9.81; R=0.026; %%0.051;
m1=0.03; m2=0.781; %%1.13;
I1=0.00002028; %%0.000039;
I2=0.00765458; %%0.0041;
l=0.099; %%0.07;
Kt=0.006123; Kw=0.006087;
Jm=0.0001234;
Ra=3; r=1;
B=0.00234;

%% Funcion
f1=dth1;

f22=f2;

f3=dth2;

f44=f4;

dx=[f1; f22; f3; f44];

a1=jacobian(dx,[th1;dth1;th2;dth2]);
aa1=eval(a1);
A1=subs(jacobian(dx,[th1;dth1;th2;dth2]),{th1,th2,dth1,dth2},{0,0,0,0});
;
AA1=eval(A1)

b2=jacobian(dx,V);
bb2=eval(b2);
B2=subs(jacobian(dx,V),{th1,th2,dth1,dth2},{0,0,0,0});
BB2=eval(B2)

%% Modelo de 4 estados
A=[0 1 0 0;0 -1.5592 58.5689 1.5592; 0 0 0 1;0 0.2322 -52.5154
-0.2322];
B=[0;0.4073;0;-0.0607];
C=[0 0 1 0];

```

```

%% Modelo con tres estados
A1=[-1.5592 58.5689 1.5592;0 0 1;0.2322 -52.5154 -0.2322];
B1=[0.4073;0;-0.0607];
C1=[0 1 0];
D1=[0];

%% Modelo en tiempo discreto

Ts=0.01;
A1d=Ts*A1+eye(3,3);
B1d=Ts*B1;
Cd=C1;

Q=diag([1000 100 1000]);
R=1;
K=d1qr(A1d,B1d,Q,R);

%% diceo de L
Lk=place(A1d',Cd',[0.8 0.4 0.7])';
eig(A1d-Lk*Cd);

%%Matriz Ak en modo discreto
Ak=A1d-Lk*Cd;

%% ganancia del observador

Kd1=place(Ak',Cd',[0.79 0.13 0.65])';

```

Código de la Figura 5.2: Diagrama de bloques de control PD y observador

```

////////////////////////////////////
%%Embedded MATLAB Function1

function [dx,y] = fcn(x,u)
%#eml
x1=x(1);
x2=x(2);
x3=x(3);

X=[x1;x2;x3];

A1=[-1.5592 58.5689 1.5592;0 0 1;0.2322 -52.5154 -0.2322];
B1=[0.4073;0;-0.0607];
C1=[0 1 0];

dx=A1*X+B1*u;

y = C1*X;

////////////////////////////////////
%%Observador

function [X_1st, thet_1st , ups_1st ] = fcn(u,y,Cd,Ak,Kd1,miu,xkest , tkest ,
upsk)

```

```

x2kest=xkest(1);
x3kest=xkest(2);
x4kest=xkest(3);

Xkest=[x2kest;x3kest;x4kest];

Psi=[u y 0 0 0;0 0 y 0 0;0 0 0 u y];

ups_1st=(Ak-Kd1*Cd)*upsk+Psi;
thet_1st=tkest+miu*upsk'*Cd'*(y-Cd*Xkest);

X_1st=Ak*Xkest+Psi*tkest+Kd1*(y-Cd*Xkest)+ups_1st*(thet_1st-tkest);

```

Código de la Figura 5.12: Diagrama de bloques del control LQR en Simulink

```

%%Modelo de ISEIWAY

function [dx,y] = fcn(x,u)
%%eml
x1=x(1);
x2=x(2);
x3=x(3);

X=[x1;x2;x3];

A1=[-1.5592 58.5689 1.5592;0 0 1;0.2322 -52.5154 -0.2322];
B1=[0.4073;0;-0.0607];
C1=[0 1 0];

dx=A1*X+B1*u;

y = C1*X;
////////////////////////////////////
%%LQR
function V = fcn(x,Kd)
%%eml
V = (-Kd'*x);

```

Código de la Figura 5.15: Diagrama de bloques del control LQR con el Observador Adaptable

```

%%Observador
function [DXest,Dtest,Dups] = fcn(u,C,A,y,K1,xest,test,ups)

y2=y(2);
x1est=xest(1);
x2est=xest(2);
x3est=xest(3);
x4est=xest(4);

```

```

Gamma=5e2*diag([2,2.4,2,2.4]);
Sigma=100*diag([1,1]);
Xest=[x1est;x2est;x3est;x4est];

Psi=[0 0 0 0;y2 u 0 0;0 0 0 0;0 0 y2 u];

Dups=(A-K1*C)*ups+Psi;

DXest=A*Xest+Psi*test+(K1+ups*Gamma*ups'*C'*Sigma)*(y-C*Xest);
Dtest=Gamma*ups'*C'*Sigma*(y-C*Xest);

```

ARDUINO®

Codigo del ISEIWAY a escala con observador

```

////////////////// librerias ////////////////////
#include <Wire.h>
#include "DualVNH5019MotorShield.h"
#include <DueFlashStorage.h>
DueFlashStorage dueFlashStorage;
////////////////////////////////////

////// variables del acelerometro y giroscopio con filtro
////////////////////////////////////
DualVNH5019MotorShield md;
//Direccion I2C de la IMU
#define MPU 0x68

//Ratios de conversion
#define A_R 16384.0
#define G_R 131.0

//Conversion de radianes a grados 180/PI
#define RAD_A_DEG = 57.295779

//MPU-6050 da los valores en enteros de 16 bits
//Valores sin refinar
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;
unsigned long interval;

float pos_leida_v, ref, pos_leida_theta, error, error1, MKT, kp, QKT, PKT, ki
,PKT_ANT=0,kd, error_ant=0,UKT,UKT_AJUST, Kajust;
float vin, R, i, V;
//Angulos
float Acc[2];
float Gy[2];
float Angle[2];
float Acyest;
////////////////////////////////////

//////////////////variables de loop//////////////////

```

```

const int button1 = 52;    // interruptor azul
const int button2 = 50;    // interruptor anaranjado
const int led = 53;
int buttonState1 = 0;
int buttonState2 = 0;
/////////////////////////////////////////////////////////////////

////////////////variables de interrupcion////////////////

volatile boolean ledon;
volatile unsigned long lasttime;
volatile unsigned long now;
int FREQ_1Hz =500; //100=10us 1=1s;
/////////////////////////////////////////////////////////////////

//////////////// funcion de inerrupcion////////////////
void TC3_Handler() {
    TC_GetStatus(TC1, 0);
    // now = millis();

    //Leer los valores del Acelerometro de la IMU
    Wire.beginTransmission(MPU);
    Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,6,true); //A partir del 0x3B, se piden 6
        registros
    AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
    AcY=Wire.read()<<8|Wire.read();
    AcZ=Wire.read()<<8|Wire.read();

    //Se calculan los angulos Y, X respectivamente.
    Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2))
        )*RAD_TO_DEG;
    Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*
        RAD_TO_DEG;

    //Leer los valores del Giroscopio
    Wire.beginTransmission(MPU);
    Wire.write(0x43);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,4,true); //A diferencia del Acelerometro, solo
        se piden 4 registros
    GyX=Wire.read()<<8|Wire.read();
    GyY=Wire.read()<<8|Wire.read();

    //Calculo del angulo del Giroscopio
    Gy[0] = GyX/G_R;
    Gy[1] = GyY/G_R;

    //Aplicar el Filtro Complementario
    Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0];
    Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1];

    float Acx=Acc[0];
    float Acxest=Angle[0];

```

```

float Acy=Acc[1];
Acyest=Angle[1];

error1 = (ref - Acyest);

error = abs(error1);
MKT = error*kp;
PKT = (error*ki)+PKT_ANT;
QKT = (error*kd)-error_ant*kd;
UKT = MKT + QKT + PKT;
error_ant = error;
PKT_ANT = PKT;
Kajust = 3.75; //12.75; //Tomando en cuenta solo el error, el
              //cual se considera maximo de 20 grados
UKT_AJUST = UKT*Kajust;
UKT_AJUST = (int)floor(UKT_AJUST);
//analogWrite(pin_pwm,UKT_AJUST);
// Salida PWM

if(error1 > 0){
    // md.setM1Speed(-UKT_AJUST);
    md.setM2Speed(-UKT_AJUST);
}
if(error1 < 0){
    // md.setM1Speed(UKT_AJUST);
    md.setM2Speed(UKT_AJUST);
}

i=md.getM2CurrentMilliamps();
i=i/1000;
V=(vin-(R*i))/0.006087;
//digitalWrite(led, ledon = !ledon);

//Serial.println(now - lasttime);
// lasttime = now;
}
////////////////////////////////////
////////////////////////////////////Configuracion de Interrupcion////////////////////////////////////
void startTimer(Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t
frequency){

//Enable or disable write protect of PMC registers.
pmc_set_writeprotect(false);
//Enable the specified peripheral clock.
pmc_enable_periph_clk((uint32_t)irq);

TC_Configure(tc, channel, TC_CMR_WAVE|TC_CMR_WAVSEL_UP_RC|
TC_CMR_TOCLKS_TIMER_CLOCK4);
uint32_t rc = VARIANT_MCK/128/frequency;

```

```

TC_SetRA(tc, channel, rc/2);
TC_SetRC(tc, channel, rc);
TC_Start(tc, channel);

tc->TC_CHANNEL[channel].TC_IER = TC_IER_CPCS;
tc->TC_CHANNEL[channel].TC_IDR = ~TC_IER_CPCS;
NVIC_EnableIRQ(irq);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
const float Ak[3][3]={{0.9844, -1321.6676,0.0155},{0, -0.0820,0.0100},
                    {0.0023, -34.0615,0.9976}};
const float Ck[1][3]={0,1,0};
//const float Kk[3][1]={{-81.5049},{-0.02000},{-1.6417}};
const float Kk[3][1]={{-819.5049},{-0.2000},{-12.6417}};
const float miu=0.0005;
const float C_trans[3][1]={{0},{1},{0}};

float psi[3][5]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};

float epsilon_est_1[3][5]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};
float epsilon_est_1a[3][5]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};
float epsilon_est[3][5]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};
float copy_epsilon_est[3][5]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};
float epsilon_est_trans
[5][3]={{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0}};

float xk_est_1[3][1]={{0},{0},{0}};// [velocidad del motor,posicion del
pendulo, velocidad del pendulo]
float xk_est[3][1]={{0},{0},{0}};// [ velocidad del motor,posicion del
pendulo, velocidad del pendulo]

float theta_est_1[5][1]={{0},{0},{0},{0},{0}};
float theta_est[5][1]={{0},{0},{0},{0},{0}};

float K_C[3][3]={{0,0,0},{0,0,0},{0,0,0}};
float A_K_C[3][3]={{0,0,0},{0,0,0},{0,0,0}};

float miu_epsilon_est_trans
[5][3]={{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0}};
float miu_epsilon_tran_Ctrans
[5][3]={{0,0,0},{0,0,0},{0,0,0},{0,0,0},{0,0,0}};
float Cd_Xkest[1][1]={0};
float mu_x_ipstran_x_Ctran_x_y[5][1]={{0},{0},{0},{0},{0}};
float mu_x_ipstran_x_Ctran_x_CdxXtran[5][1]={{0},{0},{0},{0},{0}};
float Ak_x_Xest[3][1]={{0},{0},{0}};
float psi_x_thetaest[3][1]={{0},{0},{0}};
float K_x_y[3][1]={{0},{0},{0}};
float K_x_C_x_Xest[3][1]={{0},{0},{0}};
float ips1est_x_the1[3][1]={{0},{0},{0}};
float ips1est_x_the[3][1]={{0},{0},{0}};
float y[1][1]={0};// salida posicion delpendulo

int ii,w,Tem;
int j,l;

```

```

int a , b , c ;
float x1,x2,x3,t1,t2,t3,t4,t5;
void setup() {

  //PWM->PWM_CLK = 0x03;
  //PWMC_ConfigureClocks(4000 * PWM_MAX_DUTY_CYCLE , 1, VARIANT_MCK);
  pinMode(led , OUTPUT);
  pinMode(button1 , INPUT);
  pinMode(button2 , INPUT);
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);

  Serial.begin(250000);
  md.init();
  startTimer(TC1, 0, TC3_IRQn, FREQ_1Hz);
  lasttime = 0;

  ref =1.12;
  kp = 300;
  ki = 0;
  kd = 0.01;
  vin=10;
  R=3;

  //dueFlashStorage.write(0,x1);
  //dueFlashStorage.write(1,x2);
  //////////////////////////////////////////////////multiplicacion de K*C
  //////////////////////////////////////////////////
  for ( c=0; c < 3 ; c++ ) {
    K_C[c][b]=0;
    for ( b=0; b < 3 ; b++ ) {
      for ( a=0; a < 1 ; a++ ) {

        K_C[c][b] += Kk[c][a]*Ck[a][b];
      }
    }
  }

  //////////////////////////////////////////////////
  //epsilon_est_1[ii][j]=(Ak[ii][j]-(Kk[ii][j]*Ck[ii][j]))
  for ( ii=0; ii < 3; ii++ ) {
    for ( j=0; j < 3; j++){

      A_K_C[ii][j]=(Ak[ii][j]-K_C[ii][j]);
    }
  }
}

void loop() {

  for ( ii=0; ii < 1; ii++){

```

```

        for (j=0; j<1; j++){
            y[0][0]=Acyest;
        }
}

////////// matriz Psi////////////////////////////////////
//Psi=[u y 0 0 0;0 0 y 0 0;0 0 0 u y];

    for (ii=0; ii<3; ii++){
        for (j=0; j<5; j++){
            psi[0][0]=UKT_AJUST ;
            psi[0][1]=y[0][0] ;
            psi[1][2]=y[0][0] ;
            psi[2][3]=UKT_AJUST ;
            psi[2][4]=y[0][0] ;
        }
    }

////////////////////////////////////

//////////imprime matriz Psi////////////////////////////////////
/*
    Serial.print( " Matriz Psi\n");
    for( ii=0; ii<3; ii++){
        Serial.print(" ");
        for( j=0; j<5; j++){
            Serial.print( psi[ii][j] );
            Serial.print( ",");
        }
        Serial.print( " ");
        Serial.print( "\n");
    }
*/
////////////////////////////////////

////////// matriz Upsilon////////////////////////////////////
//ups_1st=(Ak-Kd1*Cd)*upsk+Psi;

    for ( c =0; c < 3 ; c++ ) {
        for ( b=0; b < 5 ; b++ ) {
            ipsilon_est_1a[c][b]=0;
            for ( a=0; a < 3 ; a++ ) {

                ipsilon_est_1a[c][b] += A_K_C[c][a] * ipsilon_est[a][b];
                //ipsilon_est_1[ii][j]=(Ak[ii][j]-Kk[ii][j]*Ck[ii][j])*ipsilon_est[
                //ii][j]+psi[ii][j];
            }
        }
    }

for (ii=0; ii<3; ii++) {
    for (j=0; j<5; j++){
        ipsilon_est_1[ii][j] = ipsilon_est_1a[ii][j] + psi[ii][j];
        //ipsilon_est[ii][j] = ipsilon_est_1[ii][j];
    }
}

```

```

}

//////////imprime matriz epsilon////////////////////////////////////
/* Serial.print( " Matriz epsilon\n");
for( ii=0; ii<3; ii++){
  Serial.print(" ");
  for( j=0; j<5; j++){
    Serial.print( epsilon_est_1[ii][j] );
    Serial.print( " ");
  }
  Serial.print( "          ");
  Serial.print( "\n");
}*/

for (ii=0; ii<3; ii++) {
  for (j=0; j<5; j++){

    epsilon_est[ii][j] = epsilon_est_1[ii][j];
  }
}

/* Serial.print( " Matriz epsilon\n");
for( ii=0; ii<3; ii++){
  Serial.print(" ");
  for( j=0; j<5; j++){
    Serial.print( epsilon_est[ii][j] );
    Serial.print( " ");
  }
  Serial.print( "          ");
  Serial.print( "\n");
}*/
//////////

//////////transpuesta de epsilon////////////////////////////////////

for( ii=0; ii<3; ii++){
  for( j=0; j<5; j++){
    copy_epsilon_est[ii][j]=epsilon_est[j][ii];
  }
}

for( ii=0; ii<3; ii++){
  for( j=0; j<5; j++){

    epsilon_est_trans[j][ii]=copy_epsilon_est[ii][j];

  }
}

//////////imprime matriz epsilon transpuesta////////////////////////////////////

```

```

/* Serial.print( '   Matriz ipsilon '\n');
for( ii=0; ii<5; ii++){
  Serial.print(' ');
  for( j=0; j<3; j++){
    Serial.print( ipsilon_est_trans[ii][j] );
    Serial.print( ', ');
  }
  Serial.print( '          ');
  Serial.print( '\n');
}*/
/////////////////////////////////////////////////////////////////

//////////////////THETA////////////////////////////////////
//thet_1st=tkest+miu*upsk'*Cd'*(y-Cd*Xkest);

//////////////////miu * ups'/////////////////////////////////

    for ( b=0; b < 5 ; b++ ) {
      for ( a=0; a < 3 ; a++ ) {

        miu_ipsilon_est_trans[b][a] = miu * ipsilon_est_trans[b][a];
      }
    }

/*
  Serial.print( '   Matriz ipsilon ' * miu\n');
for( ii=0; ii<5; ii++){
  Serial.print(' ');
  for( j=0; j<3; j++){
    Serial.print( miu_ipsilon_est_trans[ii][j] );
    Serial.print( ', ');
  }
  Serial.print( '          ');
  Serial.print( '\n');
}
*/
/////////////////////////////////////////////////////////////////

//////////////////miu*ups'*c'/////////////////////////////////

    for ( c =0; c < 5 ; c++ ) {
      for ( b=0; b < 1 ; b++ ) {
        miu_ipsilon_tran_Ctrans[c][b]=0;
        for ( a=0; a < 3 ; a++ ) {

          miu_ipsilon_tran_Ctrans[c][b] += miu_ipsilon_est_trans[c][a]*
            C_trans[a][b];

        }
      }
    }

/*
  Serial.print( '   Matriz miu*ipsilon '*C'\n');

```

```

for( ii=0; ii<5; ii++){
  Serial.print(" ");
  for( j=0; j<1; j++){
    Serial.print( miu_ipsisilon_tran_Ctrans[ii][j] );
    Serial.print(" , ");
  }
  Serial.print(" ");
  Serial.print( "\n");
}*/
////////////////////////////////////
//////////////////////////////////// Cd*Xest////////////////////////////////////

for ( c =0; c < 1 ; c++ ) {
  for ( b=0; b < 1 ; b++ ) {
    Cd_Xkest[c][b]=0;
    for ( a=0; a < 3 ; a++ ) {

      Cd_Xkest [c][b] +=Ck[c][a]*xk_est[a][b];

    }
  }
}

/*
Serial.print( " Matriz C*xest\n");
for( ii=0; ii<1; ii++){
  Serial.print(" ");
  for( j=0; j<1; j++){
    Serial.print( Cd_Xkest[ii][j] );
    Serial.print(" , ");
  }
  Serial.print( " ");
  Serial.print( "\n");
}*/

////////////////////////////////////
//////////////////////////////////// miu*ipsisilon '*c '*y////////////////////////////////////

for ( c =0; c < 5 ; c++ ) {
  for ( b=0; b < 1 ; b++ ) {
    mu_x_ipstran_x_Ctran_x_y[c][b]=0;
    for ( a=0; a < 1 ; a++ ) {

      mu_x_ipstran_x_Ctran_x_y[c][b] += miu_ipsisilon_tran_Ctrans[c][a]*y
      [a][b];

    }
  }
}

/*
Serial.print( " Matriz mu*ups '*c '*y\n");
for( ii=0; ii<5; ii++){
  Serial.print(" ");

```

```

for( j=0; j<1; j++){
    Serial.print( mu_x_ipstran_x_Ctran_x_y[ii][j] );
    Serial.print( ', ');
}
Serial.print( ' ');
Serial.print( "\n");
}*/
////////////////////////////////////

//////////////////////////////////// miu*ipsilon '*\ '{c}*C*Xest////////////////////////////////////

for ( c =0; c < 5 ; c++ ) {
for ( b=0; b < 1 ; b++ ) {
mu_x_ipstran_x_Ctran_x_CdxXtran[c][b]=0;
for ( a=0; a < 1 ; a++ ) {

mu_x_ipstran_x_Ctran_x_CdxXtran[c][b] += miu_ipsilon_tran_Ctrans[
c][a]*Cd_Xkest[a][b];

}
}
}

/*
Serial.print( ' Matriz mu*ups '*c '*C*x\n');
for( ii=0; ii<5; ii++){
Serial.print(" ");
for( j=0; j<1; j++){
Serial.print( mu_x_ipstran_x_Ctran_x_CdxXtran[ii][j] );
Serial.print( ', ');
}
Serial.print( ' ');
Serial.print( "\n");
}
}*/
////////////////////////////////////
//theta_est_1[ii][j] = theta_est[ii][j]+(miu*ipsilon_est_trans[ii][j]*
C_trans[ii][j]*(y[ii][j]-Ck[ii][j]*xk_est[ii][j]));

for( ii=0; ii<5; ii++){
for( j=0; j<1; j++){

theta_est_1[ii][j] = theta_est[ii][j]+mu_x_ipstran_x_Ctran_x_y[ii][
j]-mu_x_ipstran_x_Ctran_x_CdxXtran[ii][j];

//theta_est[ii][j]=theta_est_1[ii][j];
}
}

/*
Serial.print( ' Matriz theta estimada k+1\n");
for( ii=0; ii<5; ii++){
Serial.print(" ");
for( j=0; j<1; j++){
Serial.print( theta_est_1[ii][j] );
Serial.print( ', ');
}
}

```

```

    }
    Serial.print( '      ');
    Serial.print( '\n');
}
*/
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

for( ii=0; ii <5; ii++){
  for( j=0; j <1; j++){

    theta_est [ ii ][j]=theta_est_1 [ ii ][j];
  }
}
/*
Serial.print( '  Matriz theta estimada\n');
for( ii=0; ii <5; ii++){
  Serial.print( ' ');
  for( j=0; j <1; j++){
    Serial.print( theta_est[ii][j] );
    Serial.print( ', ');
  }
  Serial.print( '      ');
  Serial.print( '\n');
}
*/
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//X_1st=Ak*Xkest+Psi*tkest+Kd1*(y-Cd*Xkest)+ups_1st*(thet_1st-tkest);

for ( c =0; c < 3 ; c++ ) {
  for ( b=0; b < 1 ; b++ ) {
    Ak_x_Xest[c][b]=0;
    for ( a=0; a < 3 ; a++ ) {

Ak_x_Xest[c][b] += Ak[c][a]*xk_est[a][b];

    }
  }
}
/*
Serial.print( '  Matriz Ak*Xst \n');
for( ii=0; ii <3; ii++){
  Serial.print( ' ');
  for( j=0; j <1; j++){
    Serial.print( Ak_x_Xest[ii][j] );
    Serial.print( ', ');
  }
  Serial.print( '      ');
  Serial.print( '\n');
}
*/
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

for ( c =0; c < 3 ; c++ ) {
  for ( b=0; b < 1 ; b++ ) {

```

```

    psi_x_thetaest[c][b]=0;
    for ( a=0; a < 5 ; a++ ) {
psi_x_thetaest[c][b] += psi[c][a] * theta_est[a][b];
    }
}

/*
Serial.print( " Matriz psi*thetaest \n");
for( ii=0; ii<3; ii++){
    Serial.print(" ");
    for( j=0; j<1; j++){
        Serial.print( psi_x_thetaest[ii][j] );
        Serial.print( ", ");
    }
    Serial.print( " ");
    Serial.print( "\n");
}
*/
////////////////////////////////////

for ( c =0; c < 3 ; c++ ) {
    for ( b=0; b < 1 ; b++ ) {
        K_x_y[c][b]=0;
        for ( a=0; a < 1 ; a++ ) {

            K_x_y[c][b] +=Kk[c][a]*y[a][b];

        }
    }
}

/*
Serial.print( " Matriz K*y \n");
for( ii=0; ii<3; ii++){
    Serial.print(" ");
    for( j=0; j<1; j++){
        Serial.print( K_x_y[ii][j] );
        Serial.print( ", ");
    }
    Serial.print( " ");
    Serial.print( "\n");
}
*/
////////////////////////////////////

for ( c =0; c < 3 ; c++ ) {
    for ( b=0; b < 1 ; b++ ) {
        K_x_C_x_Xest[c][b]=0;
        for ( a=0; a < 1 ; a++ ) {

            K_x_C_x_Xest[c][b] += Kk[c][a]*Cd_Xkest[a][b];

        }
    }
}

```

```

}

/*
Serial.print( '   Matriz K*Cd*xest \n');
for( ii=0; ii<3; ii++){
  Serial.print(" ");
  for( j=0; j<1; j++){
    Serial.print( K_x_C_x_Xest[ii][j] );
    Serial.print( ', ');
  }
  Serial.print( "          ");
  Serial.print( "\n");
}
*/
////////////////////////////////////

for ( c =0; c < 3 ; c++ ) {
  for ( b=0; b < 1 ; b++ ) {
    ipslest_x_the1[c][b]=0;
    for ( a=0; a < 5 ; a++ ) {

      ipslest_x_the1[c][b] += epsilon_est_1[c][a]*theta_est_1[a][b];
    }
  }
}
/*
Serial.print( '   Matriz ipsik+1*theta k+1 \n');
for( ii=0; ii<3; ii++){
  Serial.print(" ");
  for( j=0; j<1; j++){
    Serial.print( ipslest_x_the1[ii][j] );
    Serial.print( ', ');
  }
  Serial.print( "          ");
  Serial.print( "\n");
}
*/
////////////////////////////////////

for ( c =0; c < 3 ; c++ ) {
  for ( b=0; b < 1 ; b++ ) {
    ipslest_x_the[c][b]=0;
    for ( a=0; a < 5 ; a++ ) {

      ipslest_x_the[c][b] += epsilon_est_1[c][a]*theta_est[a][b];
    }
  }
}
/*
Serial.print( '   Matriz ipsik+1*theta \n');
for( ii=0; ii<3; ii++){
  Serial.print(" ");
  for( j=0; j<1; j++){
    Serial.print( ipslest_x_the[ii][j] );

```

```

        Serial.print( ' ', ' ');
    }
    Serial.print( '          ');
    Serial.print( "\n");
}
*/
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
for ( ii=0; ii <3; ii++) {

    for ( j=0; j <1; j++){

        xk_est_1[ii][j] = Ak_x_Xest[ii][j] + psi_x_thaest[ii][j] + K_x_y[ii
            ][j] - K_x_C_x_Xest[ii][j] + ipslest_x_the1[ii][j] -
            ipslest_x_the[ii][j];
        //xk_est[ii][j]=xk_est_1[ii][j];

    }
}
/*
Serial.print( '   Matriz X_est k+1 \n');
for( ii=0; ii <3; ii++){
    Serial.print( ' ');
    for( j=0; j <1; j++){
        Serial.print( xk_est_1[ii][j] );
        Serial.print( ' ', ' ');
    }
    Serial.print( '          ');
    Serial.print( "\n");
}*/
for ( ii=0; ii <3; ii++) {

    for ( j=0; j <1; j++){

        xk_est [ ii ][j]=xk_est_1 [ ii ][j];

    }
}
/*
Serial.print( '   Matriz X_est \n');
for( ii=0; ii <3; ii++){
    Serial.print( ' ');
    for( j=0; j <1; j++){
        Serial.print( xk_est[ii][j] );
        Serial.print( ' ', ' ');
    }
    Serial.print( '          ');
    Serial.print( "\n");
}
*/

t1=theta_est [0][0];
t2=theta_est [1][0];
t3=theta_est [2][0];
t4=theta_est [3][0];

```

```

t5=theta_est[4][0];
x1=xk_est[0][0];
x2=xk_est[1][0];
x3=xk_est[2][0];
Serial.print(millis());
Serial.print(":");
Serial.print(t1);
Serial.print(":");
Serial.print(t2);
Serial.print(":");
Serial.print(t3);
Serial.print(":");
Serial.print(t4);
Serial.print(":");
Serial.print(t5);
Serial.print(":");
Serial.print(x1);
Serial.print(":");
Serial.print(x2);
Serial.print(":");
Serial.println(x3);
}

```

Codigo de ISEWAY tamaño real

```

////////////////// librerias ////////////////////
#include <Wire.h>
#include "DualVNH5019MotorShield.h"
////////////////////////////////////

////// variables del acelerometro y giroscopio con filtro
////////////////////////////////////
DualVNH5019MotorShield md;
//Direccion I2C de la IMU
#define MPU 0x68

//Ratios de conversion
#define A_R 16384.0
#define G_R 131.0

//Conversion de radianes a grados 180/PI
#define RAD_A_DEG = 57.295779

//MPU-6050 da los valores en enteros de 16 bits
//Valores sin refinar
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;
unsigned long interval;

float pos_leida_v, ref, pos_leida_theta, error,error1,MKT,kp,QKT,PKT,ki
,PKT_ANT=0,kd,error_ant=0,UKT,UKT_AJUST,Kajust;
float vin, R, i, V;
//Angulos
float Acc[2];
float Gy[2];
float Angle[2];

```

```

////////////////////////////////////
//variables de loop//
const int button1 = 52; // interruptor azul
const int button2 = 50; // interruptor anaranjado
const int led = 53;
int buttonState1 = 0;
int buttonState2 = 0;
////////////////////////////////////

////////////////////////////////////
//variables de interrupcion//

volatile boolean ledon;
volatile unsigned long lasttime;
volatile unsigned long now;
int FREQ_1Hz =300; //10us;
////////////////////////////////////

////////////////////////////////////
//funcion de inerrupcion//
void TC3_Handler(){
    TC_GetStatus(TC1, 0);
    // now = millis();

    //Leer los valores del Acelerometro de la IMU
    Wire.beginTransmission(MPU);
    Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,6,true); //A partir del 0x3B, se piden 6
        registros
    AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
    AcY=Wire.read()<<8|Wire.read();
    AcZ=Wire.read()<<8|Wire.read();

    //Se calculan los angulos Y, X respectivamente.
    Acc[1] = atan(-1*(AcX/A_R)/sqrt(pow((AcY/A_R),2) + pow((AcZ/A_R),2))
        )*RAD_TO_DEG;
    Acc[0] = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) + pow((AcZ/A_R),2)))*
        RAD_TO_DEG;

    //Leer los valores del Giroscopio
    Wire.beginTransmission(MPU);
    Wire.write(0x43);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,4,true); //A diferencia del Acelerometro, solo
        se piden 4 registros
    GyX=Wire.read()<<8|Wire.read();
    GyY=Wire.read()<<8|Wire.read();

    //Calculo del angulo del Giroscopio
    Gy[0] = GyX/G_R;
    Gy[1] = GyY/G_R;

    //Aplicar el Filtro Complementario
    Angle[0] = 0.98 *(Angle[0]+Gy[0]*0.010) + 0.02*Acc[0];
    Angle[1] = 0.98 *(Angle[1]+Gy[1]*0.010) + 0.02*Acc[1];

```

```

float Acx=Acc[0];
float Acxest=Angle[0];
float Acy=Acc[1];
float Acyest=Angle[1];

error1 = (ref - Acyest);

error = abs(error1);
MKT = error*kp;
PKT = (error*ki)+PKT_ANT;
QKT = (error*kd)-error_ant*kd;
UKT = MKT + QKT + PKT;
error_ant = error;
PKT_ANT = PKT;
Kajust = 3.75; //12.75; //Tomando en cuenta solo el error el cual
              se considera el maximo de 20 grados
UKT_AJUST = UKT*Kajust;
UKT_AJUST = (int)floor(UKT_AJUST);
//analogWrite(pin_pwm,UKT_AJUST);
// Salida PWM

if(error1 > 0){
    md.setM1Speed(-UKT_AJUST*0.017);
    md.setM2Speed(-UKT_AJUST*0.017);
}
if(error1 < 0){
    md.setM1Speed(UKT_AJUST*0.017);
    md.setM2Speed(UKT_AJUST*0.017);
}
//digitalWrite(led, ledon = !ledon);

//Serial.println(now - lasttime);
// lasttime = now;
}
////////////////////////////////////
////////////////////////////////////Configuracion de Interrupcion////////////////////////////////////
void startTimer(Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t
frequency){

//Enable or disable write protect of PMC registers.
pmc_set_writeprotect(false);
//Enable the specified peripheral clock.
pmc_enable_periph_clk((uint32_t)irq);

TC_Configure(tc, channel, TC_CMR_WAVE|TC_CMR_WAVSEL_UP_RC|
TC_CMR_TCCLKS_TIMER_CLOCK4);
uint32_t rc = VARIANT_MCK/128/frequency;

```

```

TC_SetRA(tc , channel , rc/2);
TC_SetRC(tc , channel , rc);
TC_Start(tc , channel);

tc->TC_CHANNEL[channel].TC_IER = TC_IER_CPCS;
tc->TC_CHANNEL[channel].TC_IDR = ~TC_IER_CPCS;
NVIC_EnableIRQ(irq);
}
////////////////////////////////////

void setup() {

  //PWM->PWM_CLK = 0x03;
  //PWMC_ConfigureClocks(4000 * PWM_MAX_DUTY_CYCLE , 1, VARIANT_MCK);
  pinMode(led , OUTPUT);
  pinMode(button1 , INPUT);
  pinMode(button2 , INPUT);
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);

  Serial.begin(115200);
  md.init();
  startTimer(TC1, 0, TC3_IRQn, FREQ_1Hz);
  lasttime = 0;

  ref = 1.12;
  kp = 300;
  ki = 0;
  kd = 0.01;
}

void loop() {

  //boton azul
  if(error1 < 0){
  buttonState1 = digitalRead(button1);
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState1 == HIGH) {
    // turn LED on:
    digitalWrite(led ,LOW);
    md.setM1Speed((UKT_AJUST*0.0155)-35);//mortor izquierdo ****tiene
    menor torque****
    md.setM2Speed((UKT_AJUST*0.0155)+60);//motor derecho
  }
  else {
    digitalWrite(led ,HIGH);
  }
  //bonton naranja
  buttonState2 = digitalRead(button2);
  if (buttonState2 == HIGH){

```

```

    // turn LED off:
    digitalWrite(led ,LOW);
    md.setM1Speed((UKT_AJUST*0.0155)+70);//motor izquierdo ****tiene
    menor torque****
    md.setM2Speed((UKT_AJUST*0.0155)-30);// motor derecho
}
else{
digitalWrite(led ,HIGH);
}
}
//boton azul
if(error1 > 1){
buttonState1 = digitalRead(button1);
// check if the pushbutton is pressed.
// if it is, the buttonState is HIGH:
if (buttonState1 == HIGH) {
    // turn LED on:
    digitalWrite(led ,LOW);
    md.setM1Speed(-((UKT_AJUST*0.0155)-35));//mortor izquierdo ****tiene
    menor torque****
    md.setM2Speed(-((UKT_AJUST*0.0155)+60));//motor derecho
}
else {
    digitalWrite(led ,HIGH);
}
}
//bonton naranja
buttonState2 = digitalRead(button2);
if (buttonState2 == HIGH){
    // turn LED off:
    digitalWrite(led ,LOW);
    md.setM1Speed(-((UKT_AJUST*0.0155)+70));//mortor izquierdo ****
    tiene menor torque****
    md.setM2Speed(-((UKT_AJUST*0.0155)-30));//motor derecho
}
else{
digitalWrite(led ,HIGH);
}
}
}
}

```

Bibliografía

- [1] NGUYEN, Hoa G., et al. Segway robotic mobility platform. En Optics East. International Society for Optics and Photonics, 2004. p. 207-220.
- [2] Lennart. System identification. Birkhäuser Boston, 1998.
- [3] Katsuhiko; YANG, Yanjuan. Modern control engineering. 1970.
- [4] Benjamin C. Automatic control systems. Prentice Hall PTR, 1981.
- [5] N. S. (2002). Sistemas de control para ingeniería.
- [6] Katsuhiko. Sistemas de control en tiempo discreto. Pearson educación, 1996.
- [7] Sergio; SEBASTIÁN, J. M.; CAMPOY, P. Control en el espacio de estado. Universidad Politécnica de Madrid, 2000.
- [8] Arnaud; ZHANG, Qinghua. Adaptive observer for discrete time linear time varying systems. En 13th IFAC IFORS Symp. on Sys. Identification (SYSID). Rotterdam. 2003.
- [9] Qinghua. Adaptive observer for multiple-input-multiple-output (MIMO) linear time-varying systems. Automatic Control, IEEE Transactions on, 2002, vol. 47, no 3, p. 525-529.
- [10] MANTZ, Ricardo Julián. Introducción al control óptimo. Cátedra de control Moderno-Universidad Nacional de La Plata, 2003, p. 19.
- [11] Maia Reynolds. Balancing a two-wheeled Segway robot. 2011. Tesis Doctoral. Massachusetts Institute of Technology.
- [12] LI, Lanjing. MATLAB User Manual.
- [13] David, et al. Self-Balancing Two Wheeled Robot. 2015.
- [14] Brian; BUSHNELL, Linda. Enhancing Linear System Theory Curriculum with an Inverted Pendulum Robot.
- [15] Manuel Sebastián Álvarez. Modelo matemático de un motor de corriente continua separadamente excitado: Control de velocidad por corriente de armadura. Lat. Am. J. Phys. Educ. Vol, 2012, vol. 6, no 1, p. 155.

-
- [16] Oscar Danilo Montoya; HERNÁNDEZ, Juan Guillermo Valenzuela; BUITRAGO, Didier Giraldo. Control global del Péndulo Rotacional Invertido empleando modelos de energía. *Scientia et Technica*, 2012, vol. 1, no 52, p. 16-25.
- [17] Virginia; MALLO, Sebastián; SACO, Roberto. CONTRIBUCIONES A UN LABORATORIO DE CONTROL DE BAJO COSTO: EL PENDULO INVERTIDO ROTANTE.
- [18] http://alojamientos.uva.es/guia_docente/uploads/2011/493/46447/1/Documento1.pdf
- [19] <http://sine.ni.com/nips/cds/print/p/lang/es/nid/207405>
- [20] <http://pdf.datasheetcatalog.com/datasheet/fairchild/LM331.pdf>
- [21] <http://pdf1.alldatasheet.es/datasheet-pdf/view/432662/CUI/AMT102-V.html>
- [22] <https://cdn.sparkfun.com/datasheets/Components/General>
- [23] Walter T. A comparison of complementary and Kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 1975, vol. 11, no 3, p. 321-325.
- [24] www.arduino.cc/en/Main/arduinoBoardDue
- [25] www.pololu.com/product/2507
- [26] http://robots-argentina.com.ar/Comunicacion_busI2C.html
- [27] robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/
- [28] https://en.wikiversity.org/wiki/SerialChart_Tutorial

