

Análisis comparativo entre un controlador difuso, un controlador LQR con filtro de Kalman y un controlador PID para un robot móvil pendular



**Karen Viviana Chate García
Oscar Eduardo Prado Ramírez**

Director: Ph.D. Carlos Felipe Rengifo

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Programa de Ingeniería en Automática Industrial
Popayán, Cauca
2016**

Agradecimientos

Damos gracias a Dios por el tiempo y la disposición que proveyó para la realización del presente trabajo de grado.

Gracias a nuestras familias y especialmente a nuestros padres y hermanos, por siempre brindarnos su apoyo constante y sincero, por ser pilares fundamentales en nuestra vida y darnos las fuerzas y ánimo necesarios para alcanzar nuestros sueños.

Gracias a nuestros compañeros y amigos que nos ofrecieron su valiosa amistad, por compartir con nosotros su tiempo y espacio, sus conocimientos, alegrías y tristezas.

Igualmente, agradecemos de forma especial a nuestro director, el Dr. Carlos Felipe Rengifo, por los conocimientos compartidos a lo largo de este proyecto, por la paciencia y la labor que realiza con esmero.

Índice general

Lista de figuras	V
Lista de tablas	VIII
Estado del Arte	IX
1. Modelo físico de un robot móvil pendular	1
1.1. Formalismo de <i>Lagrange</i>	1
1.2. Linealización del modelo matemático	6
1.2.1. Modelo dinámico subsistema eléctrico	9
1.3. Linealización del modelo matemático incluyendo la dinámica del actuador .	12
2. Robot móvil pendular <i>InstaBot SRAT-2</i>	15
2.1. Descripción	15
2.2. Características de los componentes	17
2.3. Parámetros o constantes del robot móvil pendular	21
2.3.1. Resumen de los parámetros del robot móvil pendular <i>InstaBot SRAT-2</i>	24

3. Control PID	25
3.1. Control lineal	25
3.2. Control PID	26
3.2.1. Acción proporcional	26
3.2.2. Acción integral	27
3.2.3. Acción derivativa	27
3.3. Implementación control PID	28
3.3.1. Identificación de variables	28
3.3.2. Ajuste de parámetros del controlador PID	29
3.4. Resultados	31
3.4.1. Índices de desempeño	33
3.4.2. Ángulo crítico de control	36
4. Control LQR con filtro de Kalman	38
4.1. Control LQR	38
4.2. Controlabilidad del modelo linealizado	39
4.3. Cálculo de la ganancia de realimentación del vector de estados	40
4.4. Ecuaciones generales del filtro de Kalman	43
4.5. Estimación de orientación con filtro de Kalman	45
4.6. Resultados	47
4.6.1. Índices de desempeño	49
4.6.2. Ángulo crítico de control	51

5. Control difuso	53
5.1. Control no Lineal	53
5.2. Control Difuso	54
5.3. Control PD difuso aplicado a un robot móvil pendular	55
5.3.1. Definición de conjuntos difusos	56
5.3.2. Base de Reglas	59
5.4. Resultados	60
5.4.1. Índices de desempeño	62
5.4.2. Ángulo crítico de control	64
6. Comparación del desempeño de los diferentes controladores implementados	65
6.1. Comparación con los índices de desempeño obtenidos	65
6.1.1. Resumen de los ángulos de convergencia de los controladores implementados	67
7. Conclusiones y trabajos futuros	69
7.1. Conclusiones	69
7.2. Trabajos futuros	71

Índice de figuras

1.1. Esquema gráfico de un robot móvil pendular	3
1.2. Diagrama circuital de un motor DC	10
2.1. Robot móvil pendular <i>InstaBot SRAT-2</i>	16
2.2. Sensor giroscopio + acelerómetro (MPU6050)	18
2.3. Motor con encoder JGA25-371-21	19
2.4. <i>SainSmart Protoshield Shield V3.0</i>	19
2.5. Módulo <i>bluetooth HC-05</i>	20
2.6. Resultados de la correlación de los datos	22
2.7. Ajuste entre datos obtenidos del experimento y modelo establecido	23
3.1. Diagrama en bloques del control PID para el <i>InstaBot SRAT-2</i>	29
3.2. Resultados del control PID (posición y esfuerzo de control)	31
3.3. Resultados del control PID (velocidad angular)	32
3.4. Histograma de desviaciones estándar del error para el controlador PID	35
3.5. Histograma de desviaciones estándar del esfuerzo de control para el controlador PID	35

3.6.	Experimento (a): Ángulo inicial utilizando controlador PID	37
3.7.	Experimento (b): Ángulo inicial utilizando controlador PID	37
4.1.	Diagrama en bloques del filtro de Kalman	44
4.2.	Resultados de la implementación de un controlador LQR con filtro de Kalman (posición y esfuerzo de control)	48
4.3.	Resultados del control LQR con filtro de Kalman (Velocidad angular) . . .	48
4.4.	Histograma de desviaciones estándar del error para el controlador LQR con filtro de Kalman	50
4.5.	Histograma de desviaciones estándar del esfuerzo de control para el controlador difuso	50
4.6.	Experimento (a): Ángulo inicial utilizando controlador LQR con filtro de Kalman	51
4.7.	Experimento (b): Ángulo inicial utilizando controlador LQR con filtro de Kalman	52
5.1.	Diagrama en bloques de un controlador difuso	55
5.2.	Diagrama de bloques de un controlador difuso PD	55
5.3.	Funciones de membresía para la entrada <i>Error</i>	56
5.4.	Funciones de membresía para la entrada <i>Error Change</i>	57
5.5.	Funciones de membresía para la salida <i>Control Law</i>	57
5.6.	Diagrama de superficie del controlador difuso	60
5.7.	Resultados del control difuso (posición y esfuerzo de control)	61
5.8.	Resultados del control difuso (Velocidad angular)	61
5.9.	Histograma de desviaciones estándar del error para el controlador PD difuso	63

5.10. Histograma de desviaciones estándar del esfuerzo de control para el controlador difuso	63
5.11. Experimento (b): Ángulo inicial utilizando un controlador PD difuso	64

Índice de tablas

1.	Resumen estado del arte.	XV
2.1.	Parámetros del robot móvil pendular <i>InstaBot SRAT-2</i>	24
3.1.	Resultados de los índices de desempeño obtenidos con la implementación de un controlador PID	34
4.1.	Resultados de los índices de desempeño obtenidos con la implementación de un controlador LQR con filtro de Kalman	49
5.1.	Definición de conjuntos para un controlador PD difuso.	58
5.2.	Matriz de Reglas del controlador difuso PD	59
5.3.	Resultados de los índices de desempeño obtenidos con la implementación de un controlador PD difuso	62
6.1.	Comparación de promedios de desviaciones estándar	66
6.2.	Comparación de las regiones de convergencia de los controladores	67

Estado del Arte

El péndulo invertido es un ejemplo académico clásico utilizado para introducir los principios de la dinámica de sistemas y del control realimentado [1]. Dicho sistema es no lineal, inherentemente inestable, que presenta claramente la necesidad de control. Debido a esta naturaleza, el péndulo invertido se ha convertido en la fuente básica de estudio para el diseño e implementación de diferentes tipos de controladores.

Un robot móvil pendular de dos ruedas comparte las mismas características del sistema dinámico antes mencionado. Es no lineal, inestable, multivariable y está fuertemente acoplado [2], tales particularidades han incrementado el interés de muchos investigadores en proporcionar herramientas de enseñanza e investigación que permitan afianzar los conocimientos relacionados con las diferentes técnicas de control, el procesamiento de señales y la obtención de modelos dinámicos, además de ampliar el campo de aplicación de estos sistemas y mejorar el rendimiento de este tipo de vehículos [3]. Actualmente se cuentan con algunos modelos de este tipo de robots como el Yamabico Kurara, Joe y Segway, este último comercialmente disponible como un vehículo de transporte personal [4].

El objetivo de control para el robot móvil pendular es mantener el balance del robot, para ello, se controlan las 2 ruedas del robot con dos motores independientes [5]. En general, este tipo de vehículos son afectados por disturbios externos, dinámicas no modeladas, errores en la estimación de los parámetros y el ruido adicional en las medidas obtenidas por los sensores, entre otros [6], por tanto, en el diseño del control se tiene una mayor exigencia de precisión y robustez.

Dentro de la literatura consultada se encuentra la aplicación de diferentes técnicas de

control para estos sistemas, tanto lineales como no lineales. Ambas técnicas requieren de una buena comprensión del modelo matemático [3] que puede ser obtenido mediante las ecuaciones de Lagrange, las ecuaciones de Newton-Euler, la ecuación de Gibbs-Appell y el método de Kane [5]. Para la estrategia de control lineal, el modelo que se obtiene con cualquiera de las anteriores técnicas debe ser linealizado, caso contrario al control no lineal, donde se utiliza directamente el modelo matemático obtenido.

A continuación se proponen dos ejes temáticos alrededor de los cuales se han clasificado las publicaciones consultadas sobre el robot móvil pendular.

Control lineal

Los controladores lineales requieren un modelo matemático de la planta, que es definido por cualquiera de los métodos anteriormente mencionados. Una vez obtenido el modelo, se linealiza para posteriormente continuar con el diseño del controlador. Entre los controladores lineales comúnmente conocidos para hacer control del robot móvil pendular se encuentran el PID [5], LQR [7],[8] y LQG (*Linear Quadratic Gaussian*) [9], entre otros.

En [5] se realiza la comparación entre dos métodos de control lineales, el LQR y el PID, y uno no lineal, FLC (*Fuzzy Logic Controller*), donde se concluye que este último presenta una respuesta más rápida y un menor sobreimpulso en comparación con los otros dos, pero tiene un mayor consumo de energía. Por tanto, se han propuesto mejoras a estos controladores, tal como se evidencia en [1], donde se propone una estrategia de control en espacio de estados combinada con dos lazos de control PI adicionales que permite tener un controlador lo suficientemente robusto.

En [9] se estudia el diseño de un controlador LQR con filtro de Kalman simplificado, donde el procesamiento de la señal se realiza utilizando menos recursos del sistema, reduciendo la carga del mismo. Con este controlador se tiene un buen desempeño en la velocidad de respuesta y una alta precisión, como se comprueba en las simulaciones realizadas en Matlab. De igual forma en [2], se describe el diseño y control de un robot tipo péndulo invertido de dos ruedas, donde se implementa un control LQR con compensación de fricción para garantizar una mejor estabilidad del robot. Dicha compensación de fric-

ción es calculada usando el modelo de fricción de Coulumb y finalmente se añade a la ley de control. Para el diseño del controlador fue necesario obtener estimaciones precisas de los estados del sistema.

En [10] se propone un controlador basado en la combinación de un PID con un LQR, donde el modelo dinámico del robot se estima por el método de *Lagrange* y posteriormente se linealiza. Se comprueba en primer lugar, el desempeño del controlador LQR mediante diferentes condiciones iniciales en el ángulo de inclinación, concluyéndose que entre mayor sea este ángulo, mayor será el error de aproximación de la linealización y por tanto es imposible alcanzar el equilibrio dinámico mediante este controlador. Por esta razón, se presenta el diseño de un controlador híbrido entre el LQR y el PID, donde el algoritmo de control del PID se utiliza para regresar el ángulo a los rangos de operación del LQR, dicho controlador permite obtener un buen equilibrio y robustez ante disturbios.

Control no lineal

Otra de las estrategias abordadas en la literatura para controlar un robot móvil pendular o péndulo invertido sobre ruedas son los controladores no lineales. Los controladores basados en redes neuronales [11],[12],[3] y lógica difusa [13],[14],[15] son comúnmente aplicados a este tipo de sistemas dado que permiten implementar un control sin necesidad de modelar dinámicamente la planta, dependiendo únicamente de la sensorica presente en la misma. Las distintas variaciones de las redes neuronales, tales como *Adaptive Output Recurrent Cerebellar Model Articulation Controller* (AORCMAC) planteado en [11] y [12] o *Self-tuning output recurrent cerebellar model articulation controller* (SORCMAC) abordado en [3] ofrecen algoritmos adaptativos con una alta tolerancia a fallos e inmunidad al ruido diseñados para operar en tiempo real. Los controladores basados en lógica difusa, por su parte, ofrecen velocidad, gran precisión, estabilidad y tolerancia al ruido, siendo controladores sencillos de modelar e implementar pero que pueden requerir de altas cargas computacionales y mayor consumo de energía como se evidencia en [5]. Otro método de control no lineal propuesto en la literatura es el controlador H_∞ realizado en [6], el cual aborda el problema de control como una optimización matemática para encontrar los parámetros que minimizan el valor del costo funcional para describir la trayectoria óptima de un robot móvil pendular. Este tipo de técnicas permite minimizar el impacto

de perturbaciones añadiendo robustez al sistema y mejorando la velocidad de respuesta como se concluye en [6], sin embargo la carga computacional para la implementación de este tipo de algoritmos es elevada, debido a la complejidad de las operaciones requeridas. Cada uno de los controladores anteriormente mencionados fueron aplicados a diferentes prototipos de robots.

En la Tabla 1 se presenta una clasificación de las referencias bibliográficas consultadas según el tipo de modelado y control realizados en cada uno de ellos.

Referencia	Tipo de modelado	Tipo de control	Observación
A comparison of controllers for balancing two wheeled inverted pendulum robot [5]	Ecuaciones de Euler-Lagrange	LQR, PID, FLC	Se realiza una comparación entre los controladores mencionados, analizando el desempeño de cada uno de ellos.
A design for two-wheeled self-Balancing robot based on Kalmar Filter and LQR [9]	Se utilizan las leyes de Newton para establecer el modelo de movimiento físico	LQR	Se propone un filtro de Kalman simplificado para realizar el procesamiento de la señal y el control mediante LQR.
Design and control of YAIP - an inverted pendulum on two wheels robot [2]	Ecuaciones de Euler-Lagrange	LQR	Se adiciona a la ley de control establecida por LQR una compensación por fricción basada en el modelo de fricción de Coulumb.
Dynamic model and balancing control for two-wheeled self-balancing mobile robot on the slopes [16]	Ecuaciones de Lagrange	LQR	Se utiliza un modelo cinemático y el método de Lyapunov para determinar la estabilidad y controlabilidad del robot. Además el diseño del LQR se realiza para zonas con pendientes.
Intelligent backstepping control for wheeled inverted pendulum [11]	No realizan modelado dinámico de la planta de ningún tipo	AORCMAC	Las leyes de adaptación se derivan del análisis de estabilidad de Lyapunov y técnicas de linealización de Taylor.

Just in time predictive control for a two wheeled robot [17]	Se utiliza una base de datos en lugar de un modelo. En la base de datos se tienen las entradas y salidas de la planta controlada.	Control predictivo <i>Just in time</i> (JIT)	Se almacenan los valores de entrada y salida del robot y utilizando el algoritmo JIT se predice una entrada de control apropiada para comparar los datos de entrada y salida con la base de datos.
Mathematical model and control strategy of a two wheeled self balancing robot [1]	Ecuaciones de Lagrange	LQR y dos controladores PI	Se propone un control en espacio de estados a través de un regulador lineal cuadrático con dos lazos de control PI adicionales para permitir el control de la velocidad y la velocidad diferencial de las ruedas.
Model predictive control for a two wheeled self balancing robot [18]	Ecuaciones de movimiento. Se desacoplan las ecuaciones dinámicas del robot	<i>Model predictive control</i> (MPC)	Se realiza una comparación entre el controlador PD y el MPC ante la presencia de diferentes disturbios.
Modeling control of a two wheeled self balancing robot [8]	Ecuaciones cinemáticas por el método de Lagrange del cuerpo del robot y las ruedas	LQR	Se realiza una comparación entre el controlador PID dual y el controlador LQR. Para el primero se diseñan dos controladores PID, uno para controlar el balance del cuerpo y el otro para la velocidad de los motores.
Motion control for a two-wheeled vehicle using a self-tuning PID controller [19]	Análisis dinámico de un Diagrama de Fuerzas	PID Autosintonizado	Se aborda el problema de la autosintonización de forma similar a como lo haría una Red Neuronal estableciendo tasas de aprendizaje y entrenamiento al algoritmo de autosintonía.

Nonlinear control of mobile inverted pendulum [4]	Diagrama de fuerzas y Modelado en Espacio de Estados	Control no Lineal Modular	Se diseñan 3 controladores no lineales, encargándose cada uno del seguimiento de la referencia de una variable distinta en el mismo robot.
Practical stabilization of Two-Wheel mobile robot with velocity limitations using time-varying control law [20]	Ecuaciones de Lagrange	<i>Time-varying control law</i>	Se aborda un problema de control ya desarrollado por otros autores buscando mejorar al escoger una precisión adecuada de la trayectoria para limitar la frecuencia de la señal de control y mejorar la calidad del estado transitorio del robot.
Real-time control of a wheeled inverted pendulum based on an intelligent model free controller [12]	No realizan modelado dinámico de la planta de ningún tipo	AORCMAC	Utilizan un sistema de redes neuronales para desarrollar un algoritmo de aprendizaje en línea, donde según las lecturas de los sensores los pesos de la red neuronal se ajustan automáticamente, permitiendo así controlar el robot sin conocer el modelo dinámico del mismo.
Researching Of Two-Wheeled Self-Balancing Robot Base On LQR Combined With PID [10]	Modelado de Lagrange, linealizando alrededor del punto de equilibrio	PID y LQR	Se utiliza el controlador LQR como un complemento al controlador PID, logrando así mejorar la respuesta del control en la vecindad del punto de equilibrio en el que fue linealizado.
The System Design and LQR control of a Two-wheels Self-balancing Mobile Robot [7]	Modelado de Lagrange	LQR	Se utiliza un regulador lineal cuadrático para mantener el balance del robot cuando se reciben señales de disturbio, minimizando la dinámica del error y el consumo de energía.
Two-wheeled Robot Control Based on Self-tuning Output Recurrent CMAC [3]	Análisis de Diagrama de fuerzas	SORCMAC	Utilizan el método del gradiente dinámico para realizar un ajuste en línea de los parámetros del SORCMAC.

Two-wheeled self-balanced pendulum workspace improvement via underactuated robust nonlinear control [6]	Modelado de Lagrange	Controlador no lineal H_∞	Se recrea la solución de un problema de optimización de parámetros para implementar un controlador no lineal H_∞ en función del mejor valor posible de costo funcional.
---	----------------------	----------------------------------	--

Tabla 1: Resumen estado del arte.

Objetivos

Objetivo General

Comparar el desempeño de un controlador difuso con respecto a dos controladores lineales: un PID y un LQR cuando estos se aplican a un robot tipo péndulo invertido sobre ruedas.

Objetivos Específicos

- Obtener el modelo no lineal en espacio de estados de un robot móvil pendular a partir del formalismo de *Lagrange*.
- Implementar un controlador difuso PD, un controlador PID clásico y un controlador en espacio de estados LQR para el robot móvil pendular *InstaBot SRAT-2*.
- Evaluar el desempeño de los controladores implementados con base en el error cuadrático medio de orientación, el esfuerzo de control y el rango de valores angulares iniciales para el cual se tiene convergencia al punto de equilibrio deseado (inclinación cero).

Estructura de la Monografía

Además del estado del arte, este trabajo se compone de siete capítulos:

En el capítulo 1 se hace una recopilación general de las leyes físicas de un péndulo invertido, tomando este sistema como referencia para el modelado matemático en espacio de estados del robot móvil pendular.

En el capítulo 2 se realiza un acercamiento al robot móvil pendular *InstaBot SRAT-2*, detallándose cada uno de los componentes físicos y electrónicos de este sistema.

En el capítulo 3 se aborda la estrategia de control lineal PID utilizada para garantizar el equilibrio del robot móvil pendular, detallando su funcionamiento y aplicación a la planta estudiada.

Se presenta la técnica de control lineal LQR con filtro de Kalman en el capítulo 4 donde se realiza una breve descripción del controlador utilizado así como la implementación del mismo en el robot móvil pendular *InstaBot SRAT-2*.

En el capítulo 5 se presenta el diseño del controlador difuso aplicado al robot, detallando los conjuntos difusos utilizados, la base de reglas establecidas y el proceso de inferencia realizado.

En el capítulo 6 se presentan los resultados finales comparando los índices de desempeño obtenidos con la implementación de cada controlador en el robot móvil pendular *InstaBot SRAT-2*.

Finalmente en el capítulo 7 se muestran las ventajas y desventajas de cada técnica de control desarrollada para así obtener un resultado que concluya la pregunta de investigación realizada en este trabajo. Además se presentan los temas de interés para proyectos futuros.

La información adicional a este trabajo se puede encontrar en:

- Manual del robot móvil pendular
- Código en C de la implementación de cada controlador en el robot móvil pendular *InstaBot SRAT-2*
- *Scripts* de *MATLAB* del procesamiento de datos realizado.
- Fotos y videos de los componentes del robot, así como del funcionamiento de cada uno de los controladores implementados.

Capítulo 1

Modelo físico de un robot móvil pendular

En el presente capítulo se deduce el modelo matemático de un robot móvil pendular a partir del formalismo de *Lagrange*, considerando este sistema como un péndulo invertido. Adicionalmente se analiza el subsistema eléctrico asociado a los motores de las ruedas con el objetivo de obtener todas las dinámicas que describen el comportamiento de la planta y finalmente presentar un modelo en espacio de estados. Dado que el modelo matemático será obtenido mediante el formalismo de *Lagrange*, inicialmente se abordarán algunos conceptos físicos relacionados con la ecuación que define este formalismo.

1.1. Formalismo de *Lagrange*

La ecuación de *Lagrange* permite obtener el modelo matemático de un sistema dinámico a partir de las expresiones que describen sus energías cinética y potencial. La dinámica presentada por *Lagrange* es un procedimiento alternativo de la mecánica newtoniana para modelar sistemas, fundamentado en la mecánica analítica donde se manejan magnitudes asociadas al cuerpo (energías)[21].

1. Energía cinética: Es una cantidad escalar que representa la energía asociada con el movimiento de un objeto o partícula [21] . Se define por la ecuación (1.1), donde se

relacionan la masa m , la inercia J , la velocidad lineal v y la velocidad angular ω del objeto o partícula.

$$T = \frac{1}{2}m|v|^2 + \frac{1}{2}J|\omega|^2 \quad (1.1)$$

2. Energía potencial: Es una cantidad escalar que representa la energía asociada con la posición de un objeto, se define por la ecuación (1.2), donde m es la masa del objeto, h , la altura del objeto y g es la gravedad [21].

$$U = mgh \quad (1.2)$$

Tanto las velocidades v y ω , como la altura h se miden con respecto a un sistema de referencia común denominado galileano o inercial.

La ecuación de *Lagrange* [21] se define como:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} + \frac{\partial R(q, \dot{q})}{\partial \dot{q}} = e(t) \quad (1.3)$$

Donde: $L(q, \dot{q}) = T(q, \dot{q}) - U(q, \dot{q})$, es conocido como el *lagrangiano* y corresponde a la diferencia entre las energías cinética (T) y potencial (U) del sistema.

En la ecuación (1.3), $R(q, \dot{q})$ contiene los términos de disipación de energía, $e(t)$ es el vector de esfuerzos generalizados aplicados al sistema.

La función de *Lagrange* depende de coordenadas generalizadas y de velocidades generalizadas, las cuales se encuentran definidas por el vector q y su derivada \dot{q} con respecto al tiempo respectivamente. Cada grado de libertad se expresa mediante una coordenada generalizada [21].

La aproximación matemática del robot móvil pendular se realiza considerando las semejanzas existentes entre éste y el sistema no lineal del péndulo invertido. En la figura 1.1 se presenta un esquema gráfico del robot móvil pendular en dos dimensiones.

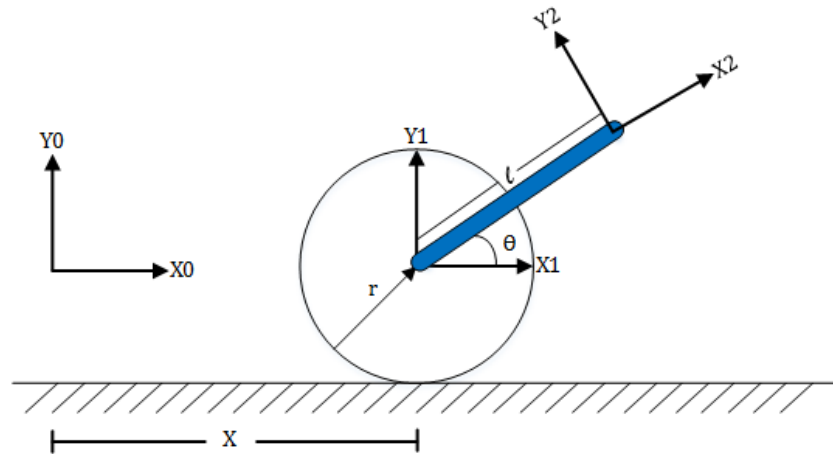


Figura 1.1: Esquema gráfico de un robot móvil pendular. [Elaboración propia]

Para el desarrollo del modelo matemático se utilizará la siguiente notación:

C_{g_1} : Vector de coordenadas del centro de gravedad de la rueda del robot con respecto al sistema de ejes coordenados $\langle X_0, Y_0 \rangle$

C_{g_2} : Vector de coordenadas del centro de gravedad del cuerpo del robot con respecto al sistema de ejes coordenados $\langle X_0, Y_0 \rangle$

J_1 : Momento de inercia de la rueda del robot móvil pendular

J_2 : Momento de inercia del péndulo (cuerpo del robot)

M : Masa de la rueda del robot móvil pendular

m : Masa del péndulo (cuerpo del robot)

r : Radio de las ruedas del robot móvil pendular

θ : Ángulo de inclinación del péndulo

l : Longitud del péndulo (cuerpo del robot)

Con referencia en la figura 1.1 se calculan las posiciones de los centros de masa de la rueda y cuerpo del robot y posteriormente se derivan en función del tiempo con el fin de obtener las velocidades de los centros de masa con respecto al referente local 0, los cuales se encuentran definidos por los vectores que se muestran a continuación.

$$C_{g_1} = \begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}, C_{g_2} = \begin{bmatrix} x + l \cos \theta \\ l \sin \theta \\ 0 \end{bmatrix}, \dot{C}_{g_1} = \begin{bmatrix} \dot{x} \\ 0 \\ 0 \end{bmatrix}, \dot{C}_{g_2} = \begin{bmatrix} \dot{x} - l \sin \theta \dot{\theta} \\ l \cos \theta \dot{\theta} \\ 0 \end{bmatrix}$$

Se procede a calcular el *lagrangiano* del sistema considerando las energías cinética y potencial del mismo.

- Con base en la ecuación (1.1), la energía cinética del sistema es:

$$T = \frac{1}{2}M [\dot{C}_{g_1}]^2 + \frac{1}{2}m [\dot{C}_{g_2}]^2 + \frac{1}{2}J_1 \left(\frac{\dot{x}}{r}\right)^2 + \frac{1}{2}J_2\dot{\theta}^2 \quad (1.4)$$

Reemplazando la magnitud de \dot{C}_{g_1} y \dot{C}_{g_2} , la ecuación (1.4) se puede plantear como:

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m \left((\dot{x} - \dot{\theta}l \sin \theta)^2 + \dot{\theta}^2 l^2 \cos^2 \theta \right) + \frac{1}{2}J_1 \left(\frac{\dot{x}}{r}\right)^2 + \frac{1}{2}J_2\dot{\theta}^2 \quad (1.5)$$

Reordenando los términos de la ecuación (1.5) y aplicando la identidad trigonométrica fundamental¹ se obtiene:

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m \left(\dot{x}^2 - 2\dot{x}\dot{\theta}l \sin \theta + \dot{\theta}^2 l^2 \right) + \frac{1}{2} \left(\frac{J_1}{r^2}\right) \dot{x}^2 + \frac{1}{2}J_2\dot{\theta}^2$$

$$T = \frac{1}{2}\dot{x}^2 \left(M + m + \frac{J_1}{r^2} \right) + \frac{1}{2}\dot{\theta}^2 (ml^2 + J_2) - \dot{x}\dot{\theta}ml \sin \theta \quad (1.6)$$

- Con base en la ecuación (1.2), la energía potencial del sistema es:

$$U = mgl \sin \theta \quad (1.7)$$

Finalmente el *lagrangiano* del sistema está dado por:

$$L = \frac{1}{2}\dot{x}^2 \left(M + m + \frac{J_1}{r^2} \right) + \frac{1}{2}\dot{\theta}^2 (ml^2 + J_2) - \dot{x}\dot{\theta}ml \sin \theta - mgl \sin \theta \quad (1.8)$$

Se definen como coordenadas generalizadas a la posición lineal x y a la posición angular θ :

¹ $\sin^2\theta + \cos^2\theta = 1$

$$q = \begin{bmatrix} x \\ \theta \end{bmatrix}, \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix}$$

De acuerdo con (1.3) las ecuaciones de *Lagrange* son descritas como:

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{x}} \right] - \frac{\partial L}{\partial x} = \frac{\Gamma}{r} \quad (1.9)$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] - \frac{\partial L}{\partial \theta} = 0$$

Siendo Γ , el torque aplicado a los motores de las ruedas del robot móvil pendular.

Reemplazando el *lagrangiano* de la ecuación (1.8) en la ecuación (1.9) se obtiene:

$$\frac{\partial L}{\partial \dot{x}} = \dot{x} \left(M + m + \frac{J_1}{r^2} \right) - \dot{\theta} ml \sin \theta$$

$$\frac{\partial L}{\partial \dot{\theta}} = \dot{\theta} (ml^2 + J_2) - \dot{x} ml \sin \theta$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{x}} \right] = \ddot{x} \left(M + m + \frac{J_1}{r^2} \right) - \ddot{\theta} ml \sin \theta - \dot{\theta}^2 ml \cos \theta \quad (1.10)$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] = \ddot{\theta} (ml^2 + J_2) - \ddot{x} ml \sin \theta - \dot{x} \dot{\theta} ml \cos \theta$$

Derivando el *lagrangiano* con respecto a las coordenadas generalizadas se obtiene:

$$\frac{\partial L}{\partial x} = 0 \quad (1.11)$$

$$\frac{\partial L}{\partial \theta} = -\dot{x} \dot{\theta} ml \cos \theta - mgl \cos \theta$$

Expresando el modelo en la forma *lagrangiana* según la ecuación (1.3), se tiene que:

$$\begin{bmatrix} \ddot{x} \left(M + m + \frac{J_1}{r^2} \right) - ml\ddot{\theta} \sin \theta - ml\dot{\theta}^2 \cos \theta \\ \ddot{\theta} (ml^2 + J_2) - ml\ddot{x} \sin \theta - ml\dot{x}\dot{\theta} \cos \theta \end{bmatrix} - \begin{bmatrix} 0 \\ -ml\dot{x}\dot{\theta} \cos \theta - mgl \cos \theta \end{bmatrix} = \begin{bmatrix} \frac{1}{r} \\ 0 \end{bmatrix} \Gamma \quad (1.12)$$

Reescribiendo de forma matricial la ecuación (1.12) se obtiene:

$$\underbrace{\begin{bmatrix} M + m + \frac{J_1}{r^2} & -ml \sin \theta \\ -ml \sin \theta & ml^2 + J_2 \end{bmatrix}}_{I(q)} \underbrace{\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}}_{\ddot{q}} + \underbrace{\begin{bmatrix} -ml\dot{\theta}^2 \cos \theta \\ 0 \end{bmatrix}}_{H(q,\dot{q})} + \underbrace{\begin{bmatrix} 0 \\ -mgl \cos \theta \end{bmatrix}}_{G(q)} = \underbrace{\begin{bmatrix} \frac{1}{r} \\ 0 \end{bmatrix}}_E \Gamma \quad (1.13)$$

Siendo $I(q)$ la matriz de inercia, $H(q, \dot{q})$ el vector de efectos centrífugo y de Coriolis y $G(q)$, el vector de fuerzas gravitacionales.

1.2. Linealización del modelo matemático

La ecuación (1.13) se puede representar mediante las ecuaciones de estado en la forma matricial siguiente:

$$\dot{x} = f(x, u) \quad (1.14)$$

En donde x representa el vector de estados de $n \times 1$, u es el vector de entrada de $p \times 1$ y $f(x, u)$ es una función de $\mathbb{R}^n \times \mathbb{R}^p$ en \mathbb{R}^n . En general, f es una función dependiente del vector de estado y del vector de entrada [22].

Debido a que las ecuaciones son no lineales, se realizará una aproximación lineal del modelo que permita analizar y diseñar más fácilmente el sistema alrededor de los puntos de interés.

En la siguiente ecuación se muestra la representación general de un modelo lineal en

espacio de estados para una ecuación diferencial de orden n .

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{1.15}$$

Siendo:

- x : Vector de estados $x \in \mathbb{R}^n$
- A : Matriz constante $A \in \mathbb{R}^{n \times n}$
- B : Matriz constante $B \in \mathbb{R}^{n \times p}$
- C : Matriz constante $C \in \mathbb{R}^{q \times n}$
- D : Matriz constante $D \in \mathbb{R}^{q \times p}$
- u : Vector de entradas $u \in \mathbb{R}^p$
- y : Combinación del vector de estados $y \in \mathbb{R}^n$

La linealización se realiza en un punto de operación o equilibrio (\bar{x}) , donde la derivada del vector de estados es igualada a cero. El proceso de linealización depende de la expansión de las ecuaciones de estado no lineales en una serie de Taylor alrededor de un punto o trayectoria de operación nominal. Para la linealización se realiza una aproximación de primer orden,

$$f(\bar{x} + \Delta x, \bar{u} + \Delta u) \cong f(\bar{x}, \bar{u}) + \left(\frac{\partial f}{\partial x} \bigg|_{\substack{u=\bar{u} \\ x=\bar{x}}} \right) \Delta x + \left(\frac{\partial f}{\partial u} \bigg|_{\substack{u=\bar{u} \\ x=\bar{x}}} \right) \Delta u\tag{1.16}$$

Dado que $f(\bar{x}, \bar{u}) = 0$ debido a que (\bar{x}, \bar{u}) es un punto de equilibrio, se obtiene

$$\dot{x} = \left(\frac{\partial f}{\partial x} \bigg|_{\substack{u=\bar{u} \\ x=\bar{x}}} \right) \Delta x + \left(\frac{\partial f}{\partial u} \bigg|_{\substack{u=\bar{u} \\ x=\bar{x}}} \right) \Delta u\tag{1.17}$$

Dado que $\Delta x = x - \bar{x}$, su derivada es $\dot{x} = \dot{\Delta x}$ ya que \bar{x} es una constante. Se tiene entonces que:

$$\dot{\Delta x} = A \Delta x + B \Delta u\tag{1.18}$$

Siendo,

$$A = \left. \frac{\partial f}{\partial x} \right|_{\substack{u=\bar{u} \\ x=\bar{x}}}, \quad B = \left. \frac{\partial f}{\partial u} \right|_{\substack{u=\bar{u} \\ x=\bar{x}}} \quad (1.19)$$

Teniendo en cuenta lo anterior se procederá a realizar la linealización del modelo matemático, para ello se reescribe la ecuación (1.13) en términos de las variables de las cuales depende el modelo.

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}, \quad F = \begin{bmatrix} x_3 \\ x_4 \\ I^{-1}(x_2) (E \Gamma - H(x_2, x_4) - G(x_2)) \end{bmatrix} \quad (1.20)$$

Donde el vector de estados se encuentra definido como:

$$\begin{aligned} x_1 &= x \\ x_2 &= \theta \\ x_3 &= \dot{x} \\ x_4 &= \dot{\theta} \end{aligned} \quad (1.21)$$

Como se aprecia en la ecuación (1.20), el modelo dinámico del robot es independiente de x_1 , lo que implica que colocar el robot en cualquier posición lineal es irrelevante para la dinámica del mismo debido a que físicamente el robot tendrá el mismo comportamiento.

El punto de equilibrio definido para realizar la linealización es:

$$\begin{aligned} \bar{x}_1 &= 0 \\ \bar{x}_2 &= \frac{\pi}{2} \\ \bar{x}_3 &= 0 \\ \bar{x}_4 &= 0 \\ \bar{\Gamma} &= 0 \end{aligned} \quad (1.22)$$

Las matrices en espacio de estados que definen el comportamiento dinámico de la planta se muestran en la ecuación 1.2

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{g l^2 m^2 r^2}{d_n} & 0 & 0 \\ 0 & -\frac{g l m c_n}{d_n} & 0 & 0 \end{bmatrix} \quad (1.23)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{r(m l^2 + J_2)}{d_n} \\ \frac{l m r}{d_n} \end{bmatrix} \quad (1.24)$$

Siendo:

$$c_n = J_1 + M r^2 + m r^2$$

$$d_n = J_1 J_2 + J_2 M r^2 + J_1 l^2 m + J_2 m r^2 + M l^2 m r^2$$

Las matrices C y D del modelo linealizado corresponden a:

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

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

Con las matrices A , B , C y D anteriores se obtiene una representación lineal en espacio de estados del robot móvil pendular alrededor del punto de equilibrio definido por (1.22).

1.2.1. Subsistema eléctrico de los motores del robot móvil pendular

El control del robot móvil pendular se realiza mediante una compensación en el movimiento de la base del robot, por medio de variaciones en el voltaje de alimentación de los motores de corriente directa de las ruedas del robot, por tanto es necesario considerar

las dinámicas que se presentan en los motores, para ello se muestra en la figura 1.2, un diagrama circuital básico para un motor de corriente directa.

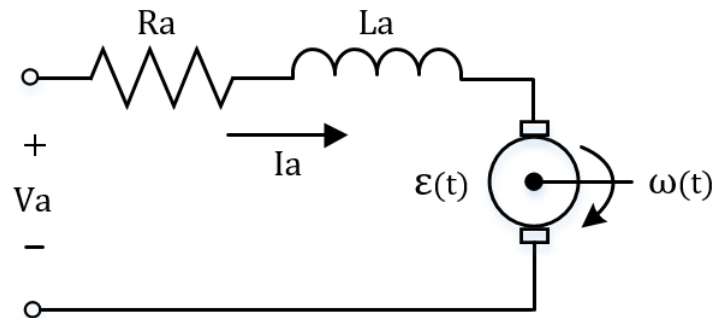


Figura 1.2: Diagrama circuital equivalente del motor de las ruedas del robot. [Elaboración propia]

Donde:

- V_a : Voltaje suministrado al motor
- I_a : Corriente que circula por el motor
- R_a : Resistencia de armadura del motor
- L_a : Inductancia de armadura del motor
- $\omega(t)$: Velocidad rotacional del motor
- $\varepsilon(t)$: *f.e.m.* inducida en el motor

El subsistema eléctrico del motor presentado en la figura 1.2 se describe por la siguiente ecuación:

$$-V_a + I_a R_a + L_a \dot{I}_a + \varepsilon(t) = 0 \quad (1.25)$$

Despejando \dot{I}_a de la ecuación (1.25) se obtiene:

$$\dot{I}_a = -\frac{1}{L_a} \left(I_a R_a - V_a + \varepsilon(t) \right) \quad (1.26)$$

Donde ε se define según la ley de Lenz como:

$$\varepsilon = K_m \omega(t) \quad (1.27)$$

Siendo K_m , la constante de máquina del motor y $\omega(t)$, la velocidad angular del motor que se describe por:

$$\omega(t) = \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad (1.28)$$

Donde:

\dot{x} : Velocidad lineal del robot

r : Radio de las ruedas del motor

$\dot{\theta}$: Velocidad rotacional del péndulo

Reemplazando la expresión de $\omega(t)$ en la ecuación 1.27 se tiene:

$$\varepsilon = K_m \left(\frac{\dot{x}}{r} - \dot{\theta} \right) \quad (1.29)$$

Sustituyendo ε en la ecuación (1.26) se obtiene la ecuación dinámica del motor descrita por:

$$\dot{I}_a = -\frac{1}{L_a} \left(K_m \left(\frac{\dot{x}}{r} - \dot{\theta} \right) + R_a I_a - V_a \right) \quad (1.30)$$

$$\dot{I}_a = -\frac{K_m}{rL_a} \dot{x} + \frac{K_m}{L_a} \dot{\theta} - \frac{R_a}{L_a} I_a + \frac{1}{L_a} V_a$$

El modelo matemático del robot móvil pendular está definido conjuntamente por las ecuaciones (1.13) y (1.30). La relación entre el sistema mecánico y el subsistema eléctrico se define por:

$$\Gamma = k I_a \quad (1.31)$$

Donde k es una constante proporcional a la corriente que circula por el motor.

Teniendo en cuenta las ecuaciones (1.13), (1.30) y (1.31) se describe el modelo dinámico

no lineal en espacio de estados del robot móvil pendular por la ecuación (1.32).

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix}, \quad F = \begin{bmatrix} x_3 \\ x_4 \\ I(x_2)^{-1} (E k x_5 - H(x_2, x_4) - G(x_2)) \\ -\frac{K_m}{rL_a} x_3 + \frac{K_m}{L_a} x_4 - \frac{R_a}{L_a} x_5 + \frac{1}{L_a} V_a \end{bmatrix} \quad (1.32)$$

Donde se consideran los siguientes estados:

$$\begin{aligned} x_1 &= x \\ x_2 &= \theta \\ x_3 &= \dot{x} \\ x_4 &= \dot{\theta} \\ x_5 &= I_a \end{aligned} \quad (1.33)$$

Siendo igualmente equivalente \dot{x}_3 y \dot{x}_4 a \ddot{x} y $\ddot{\theta}$ respectivamente.

1.3. Linealización del modelo matemático incluyendo la dinámica del actuador

En la ecuación (1.34) se reescribe el modelo dinámico no lineal en espacio de estados del robot móvil pendular en términos de los estados de los cuales depende de acuerdo con la ecuación (1.32).

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix}, \quad F = \begin{bmatrix} x_3 \\ x_4 \\ I^{-1}(x_2) (E k x_5 - H(x_2, x_4) - G(x_2)) \\ -\frac{K_m}{rL_a} x_3 + \frac{K_m}{L_a} x_4 - \frac{R_a}{L_a} x_5 + \frac{1}{L_a} V_a \end{bmatrix} \quad (1.34)$$

El punto de equilibrio definido para realizar la linealización es:

$$\begin{aligned}
 \bar{x}_1 &= 0 \\
 \bar{x}_2 &= \frac{\pi}{2} \\
 \bar{x}_3 &= 0 \\
 \bar{x}_4 &= 0 \\
 \bar{x}_5 &= 0 \\
 \bar{V}_a &= 0
 \end{aligned} \tag{1.35}$$

Con la herramienta software de *Matlab* se realiza la linealización, obteniéndose el modelo lineal en espacio de estados descrito por la ecuación (1.15). Las matrices A, B, C y D obtenidas en la linealización alrededor del punto de equilibrio definido son mostradas en la ecuación 1.38

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\frac{g l^2 m^2 r^2}{d_n} & 0 & 0 & \frac{r(k m l^2 + J_2 k)}{d_n} \\ 0 & -\frac{g l m c_n}{d_n} & 0 & 0 & \frac{k l m r}{d_n} \\ 0 & 0 & -\frac{K_m}{L R} & \frac{K_m}{L} & -\frac{R}{L} \end{bmatrix} \tag{1.36}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \tag{1.37}$$

Siendo:

$$\begin{aligned}
 c_n &= J_1 + M r^2 + m r^2 \\
 d_n &= J_1 J_2 + J_2 M r^2 + J_1 l^2 m + J_2 m r^2 + M l^2 m r^2
 \end{aligned}$$

Las matrices C y D del modelo linealizado corresponden a:

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

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (1.38)$$

De esta manera es posible obtener una aproximación matemática lineal del comportamiento dinámico de un péndulo invertido móvil, en la vecindad de los puntos de equilibrio establecidos.

Capítulo 2

Robot móvil pendular *InstaBot* *SRAT-2*

En el presente capítulo se realiza una descripción del robot *InstaBot SRAT-2* utilizado en este trabajo, detallando cada uno de los componentes electrónicos de esta plataforma móvil.

2.1. Descripción

Con el incremento de las aplicaciones del robot móvil pendular para diferentes campos [18], el interés por estudiar sus dinámicas y diseñar controladores robustos para el mismo, se ha igualmente acrecentado. La dinámica presente en este tipo de robots es semejante a la del sistema dinámico del péndulo invertido como se analizó en el capítulo 1, donde el principal interés consiste en mantener en equilibrio el péndulo, para ello deben considerarse las no linealidades presentes en el sistema mediante el modelado del mismo para tener una comprensión clara y precisa del sistema. El objetivo principal del presente trabajo es por tanto definir indicadores de desempeño que permitan establecer las ventajas y desventajas de la implementación de dos controladores lineales y uno no lineal en el robot móvil pendular y de esta forma brindar información a estudiantes y profesionales sobre los principales aspectos que deben considerarse al momento de diseñar controladores para

este tipo de robots.

El *SainSmart InstaBots Upright Rover Kit Pro Updated 2-Wheel Self-Balancing Arduino Robot Kit* es el kit adquirido para la realización del presente trabajo de grado, que consta de una plataforma móvil que corresponde al robot y un control remoto para el mismo como se visualiza en la figura 2.1.

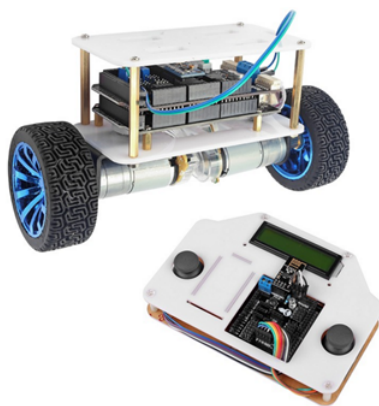


Figura 2.1: Robot móvil pendular *InstaBot SRAT-2*. [Tomado de: Sainsmart Instabot upright rover kit pro]

Los componentes del robot móvil pendular *InstaBot SRAT-2* son:

- 1 × Tarjeta Arduino (Ref: MEGA 2560 R3)
- 1 × Cable USB
- 1 × *SainSmart Protoshield Shield V3.0*
- 1 × Módulo de comunicación inalámbrico (Ref: NRF24L01)
- 2 × Motor con enconder (Ref: JGA25-371-21)
- 2 × Ruedas
- 2 × Soporte para el motor

- 1 × Módulo de puente H dual (Ref: *SainSmart* L298N)
- 1 × Sensor giroscopio + acelerómetro (Ref: MPU6050)
- 1 × Plataforma del robot móvil pendular
- 1 × Conector para batería 9V
- 2 × Coupling

2.2. Características de los componentes utilizados en el robot móvil pendular

A continuación se detallarán las características de los componentes que conforman el *InstaBot SRAT-2*.

1. Sensor giroscopio + acelerómetro MPU6050:

También conocida como unidad de medición inercial, es un sensor de movimiento que permite obtener mediciones de velocidad, rotación y fuerzas gravitacionales de forma autónoma. Posee un acelerómetro y un giroscopio en los 3 ejes (x, y, z) con una alta precisión.

Las características electrónicas de la MPU6050 mostrada en la figura 2.2 son:

- Voltaje de alimentación: 3-5 V
- Protocolo de comunicación: I2C
- Rango del giroscopio: 250/500/1000/2000 $^{\circ}/s$
- Rango del acelerómetro: 2/4/8/16 g
- Convertidor AD 16 bits en todos los ejes

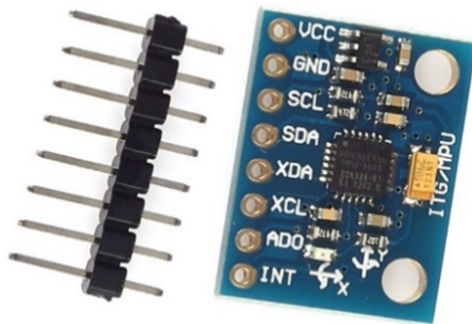


Figura 2.2: Sensor giroscopio + acelerómetro MPU6050. [Tomado de: Sainsmart *MPU6050*]

2. Motor con encoder JGA25-371-21:

Es un motor de corriente directa utilizado para plataformas robóticas debido a su alto desempeño, con un torque alto y bajo ruido. Este motor cuenta con un codificador integrado que tiene una resolución de 12 pulsos por revolución, lo que permite un control preciso de la velocidad del motor.

Las características de este motor son:

- Voltaje de alimentación: 6-12 V
- Velocidad sin carga: 201 rpm
- Velocidad con carga: 168 rpm
- Corriente sin carga: 46 mA
- Corriente con carga: 250 mA
- Potencia de salida: 1.25 W
- Torque de carga: 0.53 Kg-cm
- Temperatura de operación: $-40^{\circ}\text{C} \sim 120^{\circ}\text{C}$
- Peso: 101 g



Figura 2.3: Motor con encoder JGA25-371-21. [Tomado de: Sainsmart JGA25 Gearmotor]

3. *SainSmart Protoshield Shield V3.0*

Es una tarjeta de expansión de la unidad de motor compatible con Arduino Mega2560, que permite accionar motores de corriente continua con un máximo de corriente de 2A mediante el puente H, L298P que tiene incorporado. Además cuenta con dos canales para la conexión de los motores con encoders y tiene 3 puertos de comunicación *I2C* para la conexión de la MPU6050 y el módulo de comunicación NRF24L01 y tiene 3 potenciómetros integrados. De igual forma posee un regulador de voltaje, el LM1084 para proporcionar una salida de 5V con una corriente máxima de 4A. Posee 12 puertos para conectar servos y otros sensores según requiera la aplicación. El voltaje de alimentación de la tarjeta es de 10-14V, cuenta con un botón de reset y de encendido/apagado

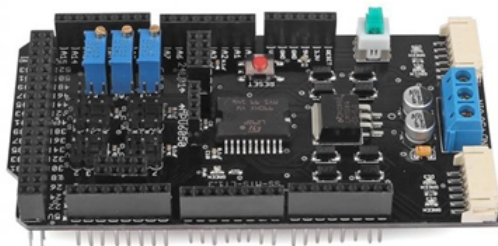


Figura 2.4: *SainSmart Protoshield Shield V3.0*. [Tomado de: SainSmart Protoshield Shield V3.0]

4. Módulo *Bluetooth* HC-05

Aunque el kit cuenta con el módulo de comunicación *bluetooth* NRF24L01, se optó por utilizar el módulo HC-05 por la facilidad de configuración y comunicación con otros dispositivos.

El HC-05 es un módulo *Bluetooth* maestro-esclavo, es decir que genera o recibe conexiones con otros dispositivos con *Bluetooth* mediante comandos *AT* a través de un puerto serie. Este módulo utiliza el protocolo *UART RS 232 serial*.

Las características de estos módulos son:

- Chipset CSR BC417143
- Bluetooth versión V2.0 + EDR
- Voltaje de alimentación: 3.3 V
- Frecuencia: 2.4 GHz banda ISM
- Modulación: GFSK (*Gaussian Frequency Shift Keying*)
- Velocidad asíncrona: 2.1 Mbps(máx)/160 Kbps
- Velocidad síncrona: 1 Mbps/ 1 Mbps
- Temperatura de trabajo: $-20^{\circ}C$ a $75^{\circ}C$

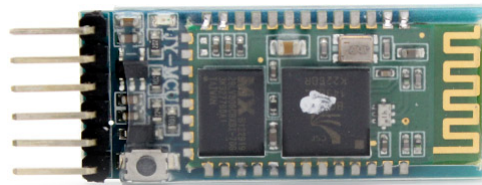


Figura 2.5: Módulo *bluetooth* HC-05. [Tomado de: Módulo HC-05]

2.3. Parámetros o constantes del robot móvil pendular

Para obtener el modelo linealizado presentado en la subsección 1.2 del capítulo 1 fue necesario establecer los parámetros o constantes del robot móvil pendular, para ello se procedió a realizar las mediciones correspondientes por medio de instrumentos de laboratorio tales como la balanza y el vernier o pie de rey. Los parámetros del robot medidos se listan a continuación:

$$M = 0.05304 \text{ Kg}$$

$$m = 0.39277 \text{ Kg}$$

$$l = 0.08530 \text{ m}$$

$$r = 0.03310 \text{ m}$$

Cada una de estas constantes se encuentra definida por la figura 1.1, donde se establecen los parámetros del modelo físico del robot móvil pendular.

El momento de inercia de la rueda del robot (J_1) fue calculado mediante la ecuación (2.1), que corresponde al momento de inercia de un cilindro.

$$J_1 = M r^2 \tag{2.1}$$

Donde:

M : Masa de la rueda del robot

r : Radio de la rueda del robot

Reemplazando los valores numéricos obtenidos anteriormente se encuentra que el momento de inercia para la rueda del robot es aproximadamente $5.8111 \times 10^{-5} \text{ kg} \cdot \text{m}^2$.

Para calcular el momento de inercia del cuerpo del robot J_2 , se realizó un experimento mediante el cual se posicionó la plataforma boca abajo en una base fija, con el propósito de representar un péndulo simple y así obtener los datos necesarios para la deducción del momento de inercia. El comportamiento oscilatorio del robot se obtuvo mediante el envío de valores aleatorios de la ley de control, los cuales definían el ancho de pulso para

los motores junto con el sentido de giro de los mismos. Los datos de posición, velocidad angular y ley de control fueron almacenados mediante una interfaz *Bluetooth* y procesados mediante el software *Matlab*. Se propuso inicialmente, con los datos de entrada y de salida obtenidos de la planta, realizar la identificación de la misma, mediante una aproximación a un modelo ARMAX (*Autoregressive Moving average model*), aunque se tuvieron buenos datos en la correlación, el ajuste entre el modelo real de la planta y el modelo predicho fue demasiado bajo (6.05%), como se observa en las gráficas 2.6 y 2.7.

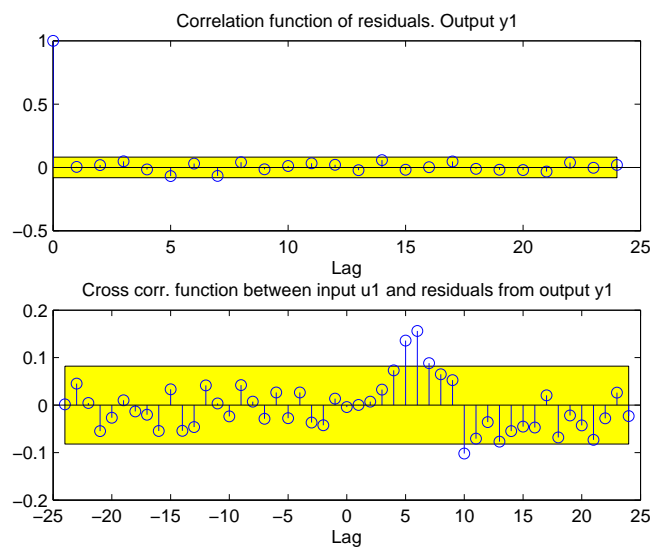


Figura 2.6: Resultados de la correlación de los datos.

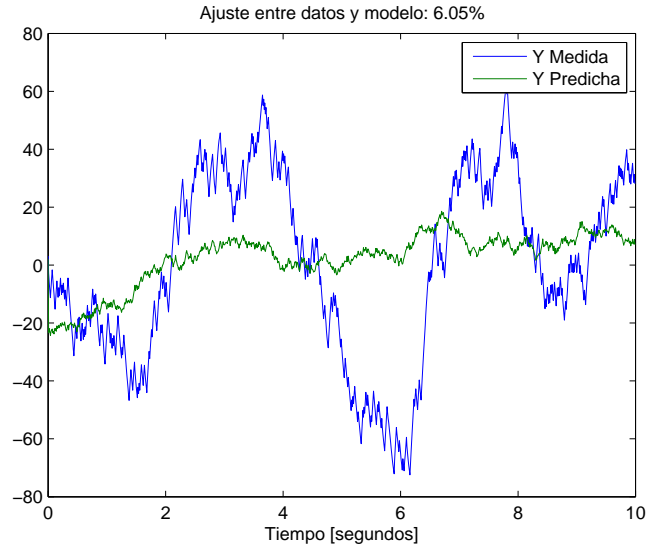


Figura 2.7: Ajuste entre datos obtenidos del experimento y modelo establecido.

Con el modelo obtenido se buscaba establecer el periodo de oscilación del péndulo. De esta manera, conociendo el periodo de oscilación (T_o), la masa (m), la longitud del péndulo (l) y la gravedad (g), es posible calcular el momento de inercia, despejándolo de la ecuación 2.2.

$$T_o = 2\pi \sqrt{\frac{J_2}{mgl}} \quad (2.2)$$

Sin embargo debido a la presencia de fricción seca en los motores producida por la caja reductora de los mismos, no fue posible realizar un experimento en el cual se pudiera evidenciar de forma clara el valor del periodo de oscilación del péndulo. Por este motivo se optó por calcular el momento de inercia J_2 utilizando la siguiente expresión:

$$J_2 = m \left(\frac{l}{3} \right)^2 \quad (2.3)$$

Donde:

l : longitud del péndulo

m : masa del cuerpo del péndulo

Debido a que la masa del cuerpo del péndulo no está uniformemente distribuida, se tomó como centro de gravedad la tercera parte de la longitud total del péndulo, región en la cual se encuentra concentrada su mayor cantidad de masa. El valor numérico calculado para el momento de inercia del cuerpo del péndulo es de $3.1754 \times 10^{-4} \text{ kg} \cdot \text{m}^2$.

2.3.1. Resumen de los parámetros del robot móvil pendular *InstaBot SRAT-2*

En la tabla 2.1 se realiza una recopilación de los parámetros encontrados a lo largo del capítulo:

Parámetro	Símbolo	Valor	Unidad
Longitud del cuerpo del péndulo	l	0.08530	m
Radio de la rueda	r	0.03310	m
Masa de la rueda	M	0.05304	Kg
Masa del péndulo	m	0.39277	Kg
Momento de inercia de la rueda	J_1	5.8111×10^{-5}	$Kg \text{ m}^2$
Momento de Inercia del péndulo	J_2	3.1754×10^{-4}	$Kg \text{ m}^2$
Gravedad	g	9.8	m/s^2

Tabla 2.1: Parámetros del robot móvil pendular *InstaBot SRAT-2*.

Capítulo 3

Control PID

En el presente capítulo se realizará una descripción del controlador PID implementado, se detallará el funcionamiento y la implementación de ésta técnica de control en el robot móvil pendular *InstaBot SRAT-2*.

3.1. Control lineal

En la teoría de control clásica es común iniciar con modelos matemáticos lineales debido a la facilidad de control que éstos presentan; sin embargo, los sistemas reales exhiben comportamientos no lineales, que por lo general son difíciles de modelar. Una solución parcial a este inconveniente es linealizar el modelo obtenido alrededor de uno o varios puntos de equilibrio, restringiendo su funcionamiento a un número limitado de puntos dentro de su vecindad, como es mostrado en el capítulo 1, sección 1.2. Los controladores lineales tales como el PID convencional, las redes de adelanto-atraso (*lag-lead*), los reguladores lineales (LQR, LQG), entre otros, presentan también este tipo de limitaciones, debido a que constituyen sistemas dinámicos sujetos al comportamiento de la planta a controlar. Por este motivo se hace necesario operar con un modelo linealizado de la planta, el cual permita lograr el objetivo de control alrededor de los puntos de interés.

3.2. Control PID

Los controladores PID son ampliamente utilizados debido a su simplicidad de implementación, la facilidad de ajuste de sus parámetros y su buen desempeño en un amplio rango de condiciones de operación [22], lo cual lo convierte en un controlador que puede ser aplicado en la mayoría de los problemas de control realimentado [22]. Un PID opera con base en el error, el cual es calculado a través de la diferencia entre el *set point* o referencia y la medición de la variable controlada. La ley de control asociada a un PID se define en la ecuación 3.1.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (3.1)$$

Donde:

$e(t)$: Valor actual del error

K_p : Constante proporcional del error

T_i : Tiempo integral

T_d : Tiempo derivativo

De la ecuación (3.1) se distinguen las 3 acciones que componen un controlador PID: acción proporcional (P), acción integral (I) y acción derivativa (D), las cuales son detalladas a continuación:

3.2.1. Acción proporcional

Una acción de control proporcional consiste en un valor ganancia el cual es multiplicado por la función de error. En sistemas estables en lazo abierto implementar un control proporcional grande implica introducir oscilaciones en la respuesta e incluso puede llevar a la inestabilidad si el valor de la ganancia es demasiado elevado; sin embargo, con valores proporcionales grandes se reduce el error de estado estacionario y se acelera la respuesta del sistema para llegar al valor deseado [23]. Para el control de procesos industriales la acción proporcional es acompañada de un valor de *offset* o *bias* el cual corresponde al valor de salida del controlador cuando el error es 0 [23]. Es poco común encontrar controladores puramente proporcionales implementados en sistemas de control, debido a que éstos no

garantizan la eliminación del error de estado estacionario, solo una atenuación del mismo y la mayoría de problemas de aplicación necesitan de controladores precisos que aseguren un seguimiento de la referencia.

3.2.2. Acción integral

Es la encargada de eliminar el error de estado estacionario a través de una acumulación continua de errores pasados. No existen controladores que operen únicamente con acción integral, debido a que la respuesta entregada tarda en hacer efecto, por lo que son complementados con controladores proporcionales que brindan una respuesta instantánea con base en el error. El efecto de la acción integral se hace presente en el comportamiento de la señal de control, debido a que la presencia de un mínimo error positivo conduce a un esfuerzo de control creciente, mientras que un error negativo resultará en un esfuerzo decreciente, sin importar la magnitud de la señal de error [23]. Para valores elevados del tiempo de integración T_i , la señal de salida tiende gradualmente hacia la referencia, por lo que el efecto de la acción integral es más rápido para valores pequeños de T_i , pero introduce oscilaciones no deseadas en la respuesta obtenida [23]. En la práctica los controladores con acción proporcional e integral (PI) se ven afectados por fenómenos como el *Windup* producido por la saturación de los actuadores ante un cambio brusco en la referencia y las oscilaciones de régimen permanente asociadas a la cuantización del conversor analógico-digital (DAC) del controlador, no obstante, a pesar de estos inconvenientes los controladores PI son los más utilizados en aplicaciones prácticas en la industria [24].

3.2.3. Acción derivativa

El propósito de la acción derivativa es reducir las oscilaciones en la salida, resultado de las acciones proporcional e integral, mejorando así la estabilidad de la respuesta en lazo cerrado [23]. La parte derivativa actúa en proporción a los cambios en la pendiente de la señal de error, por lo que es preciso afirmar que este componente opera de forma proporcional a una estimación del valor predicho del error T_d unidades de tiempo más adelante, obtenido a partir de una extrapolación lineal [23]. Es común encontrar controladores que no hagan

uso de la acción derivativa, debido a que amplifica los componentes de alta frecuencia del ruido de medida, lo que causa variaciones abruptas en la señal de control [24].

3.3. Implementación de un controlador PID a un robot móvil pendular

3.3.1. Identificación de variables

Para lograr el objetivo de control es preciso conocer el comportamiento de la planta a controlar así como el número de variables de entrada y de salida de la misma, lo cual permite clasificar el sistema dinámico a modelar como SISO (*Single Input - Single Output*), MIMO (*Multiple Input - Multiple Output*), SIMO (*Single Input - Multiple Output*) o MISO (*Multiple Input - Single Output*). En esta sección se busca identificar las variables involucradas en el proceso de control del robot móvil pendular *InstaBot SRAT-2*.

Diversos trabajos se han publicado abordando el control de un péndulo invertido móvil como un sistema MIMO, como se puede apreciar en [1],[4],[6],[8],[9] y [19]; sin embargo, a lo largo de este capítulo el sistema dinámico relacionado al robot móvil pendular *InstaBot SRAT-2* será planteado como un sistema SISO, donde:

- θ es la variable controlada, correspondiente al valor de la lectura del acelerómetro de la MPU6050 entregada en grados.
- V es la variable manipulada, equivalente al voltaje de alimentación de los motores DC entregado por la tarjeta Arduino *Shield*, el cual será traducido a un par en las ruedas del robot.

De lo anterior se deduce que V es la variable de entrada y θ es la variable de salida del sistema.

Para realizar la implementación del controlador PID, como se aprecia en la figura 3.1, es necesario determinar:

- r es la referencia o *set point* deseado del sistema, correspondiente a 0 grados que define el equilibrio del robot móvil pendular *InstaBot SRAT-2*.
- $e(t)$ es el error utilizado para definir la ley de control que se aplicará al robot, corresponde por tanto a la diferencia entre la referencia y el valor de salida obtenido de la lectura de la MPU6050.
- $u(t)$ es la ley de control definida por la ecuación 3.1.

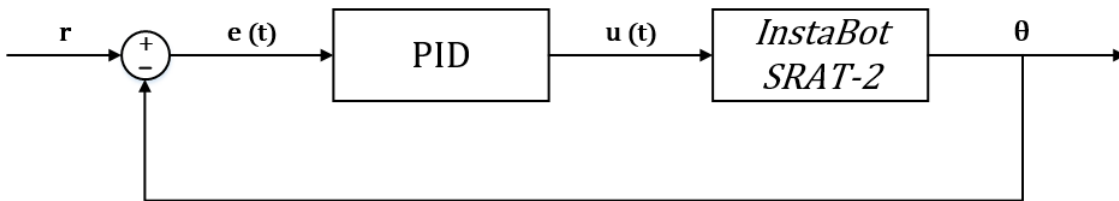


Figura 3.1: Diagrama en bloques del control PID para el robot móvil pendular *InstaBot SRAT-2*. [Elaboración propia]

3.3.2. Ajuste de parámetros del controlador PID

El controlador PID implementado en el robot móvil pendular *InstaBot SRAT-2* fue obtenido por el método de ensayo y error, mediante la variación de los tres parámetros de ajuste descritos en la subsección 3.2, siendo estos valores definidos por las constantes K_p , K_i y K_d . De acuerdo con la ecuación (3.1) se tiene que:

$$\begin{aligned}
 K_p &= K_p \\
 K_i &= \frac{1}{T_i} \\
 K_d &= T_d
 \end{aligned}$$

Conociendo el comportamiento que introducen cada uno de estos parámetros en la planta, se procedió inicialmente a limitar el rango de valores para estas constantes a $(0, 100]$, $(0, 10]$ y $(0, 10]$ respectivamente, dado que introducir ganancias muy altas puede llevar al

sistema a oscilaciones abruptas que pueden afectar el desempeño de los motores de las ruedas del robot. Además, para el caso de la acción derivativa es necesario considerar que se requiere una acción de control rápida que lleve nuevamente el sistema a la referencia inmediatamente se presente una desviación en el equilibrio del robot, por tanto se debe considerar un valor de K_d pequeño debido a que la acción derivativa puede hacer la respuesta del sistema más lenta.

De igual forma, debido al fenómeno de *windup* que se presenta en la acción integral fue necesario limitar aún más el rango de selección de K_i , entre 0 y 1 ya que esto afecta el desempeño del controlador.

Para la sintonización del controlador se inicializaron los valores de K_p , K_i y K_d , posteriormente se varió la acción proporcional, obteniéndose oscilaciones considerables y un equilibrio del robot de unos pocos segundos. Por tanto se procedió a incluir la acción integral de aproximadamente 0.1, donde se evidenció un mayor control para el seguimiento de la referencia aunque persistían las oscilaciones en el movimiento del robot móvil pendular. Debido a lo anterior, se introdujo un valor de K_d que permitiera atenuar estas oscilaciones. La variación de los parámetros se realizó en el orden descrito anteriormente, considerando los rangos establecidos y los resultados que se obtenían en el comportamiento del robot cada vez que se cambiaba uno de estos valores.

Los valores de las constantes del controlador PID con los cuales se obtuvo una mejor respuesta en el equilibrio del robot fueron:

$$K_p = 47.5, \quad K_i = 0.05, \quad K_d = 0.2 \quad (3.2)$$

Para realizar esta sintonización, la plataforma móvil cuenta dentro de su hardware con 3 potenciómetros acoplados a la tarjeta Arduino *Shield*, que tienen por objetivo modificar el valor de las constantes del PID de forma manual. Sin embargo, este proceso no es el más adecuado para esta clase de sintonización, debido a que no se conoce el valor exacto de los parámetros que se están variando ni los cambios que éstos puedan sufrir a lo largo del experimento. Por este motivo se implementó una interfaz de comunicación vía *Bluetooth* que permitiera modificar de forma precisa y en línea estas constantes durante el funcionamiento del robot y así verificar el mejor comportamiento introducido por

los tres parámetros. La interfaz hace uso del módulo *Bluetooth* HC-05 de Arduino y de la aplicación móvil *Blue serial beta*, la cual permite una comunicación *serial* jerárquica Maestro-Esclavo entre el *InstaBot SRAT-2* y un dispositivo móvil con *Bluetooth*, siendo éste último el maestro y el robot el esclavo.

3.4. Resultados

La implementación del controlador PID se realizó en lenguaje C utilizando el entorno *Arduino*. La adquisición de los datos se llevó a cabo mediante la interfaz *Bluetooth* comunicando *Arduino* y *MATLAB* y en el procesamiento de los mismos se utilizó el software *MATLAB*. A continuación se muestran las curvas de respuesta obtenidas de los sensores al implementar el control PID en la plataforma móvil *InstaBot SRAT-2*:

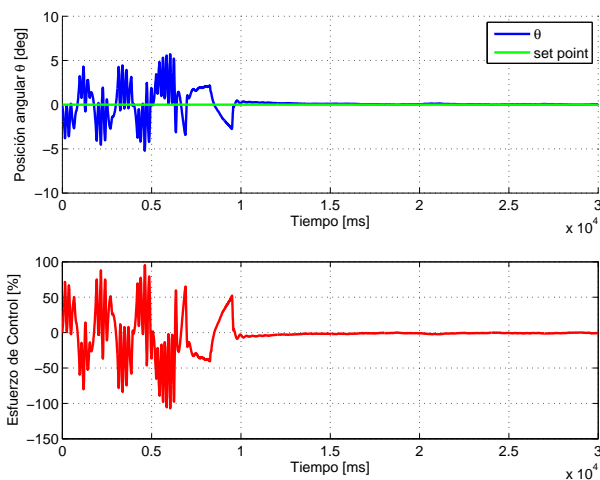


Figura 3.2: Resultados de la implementación de un controlador PID en el robot móvil pendular *InstaBot SRAT-2*. *i)* Arriba: Posición angular obtenida (línea azul) y referencia establecida (línea verde). *ii)* Abajo: Esfuerzo de control.

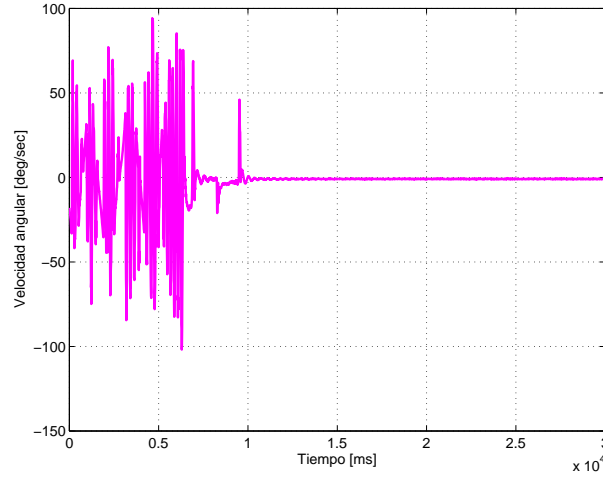


Figura 3.3: Resultados de la implementación de un controlador PID en el robot móvil pendular *InstaBot SRAT-2* (Velocidad angular).

La figura 3.3-(ii) correspondiente al esfuerzo de control fue escalizada mediante la siguiente ecuación:

$$u_s = \text{sign}(u_{pid}) \frac{100 \times |u_{pid}|}{255} \quad (3.3)$$

Siendo $|u_{pid}|$ un número entre 0 y 255 que define el ancho del pulso de la señal de voltaje que se le envía al motor y u_s un número entre -100 % y 100 %.

La respuesta obtenida en la figura 3.2 presenta oscilaciones iniciales causadas por el valor elevado de la constante proporcional K_p y la inercia del sistema, sin embargo a partir del segundo 10 la respuesta se estabiliza y se acerca al valor de la referencia.

El desempeño del controlador se ve afectado por la superficie, lisa o rugosa, donde se desplaza el robot, esto debido a la fricción que hay entre ésta y las ruedas del robot. Para una superficie rugosa existe un mejor agarre para las ruedas por lo que es más fácil alcanzar la estabilidad deseada en menos tiempo y tener un mayor tiempo de equilibrio, caso contrario si el robot se trasladara en una superficie lisa, donde la fricción es menor y el control debe ser más preciso. Los experimentos fueron realizados en superficies rugosas donde se obtuvieron mejores respuestas.

3.4.1. Índices de desempeño

En esta subsección se procederá a realizar un análisis matemático de los resultados obtenidos con la implementación del controlador PID en el robot móvil pendular *InstaBot SRAT-2*, para ello se utilizarán índices de desempeño tales como la desviación estándar del error y del esfuerzo de control y los rangos de valores angulares iniciales para el cual el robot móvil pendular tiene convergencia al punto de equilibrio deseado.

- Desviación estándar del error

$$\varepsilon_e = \sqrt{\frac{1}{N} \sum_{k=1}^N [r(k) - y(k)]^2} \quad (3.4)$$

Siendo ε_e la desviación estándar de la señal de error, N el número de muestras tomadas, $r(k) \triangleq 0$ la señal de referencia y $y(k)$ la señal de salida obtenida.

- Desviación estándar del esfuerzo de control

$$\varepsilon_u = \sqrt{\frac{1}{N} \sum_{k=1}^N u^2(k)} \quad (3.5)$$

Donde ε_u es la desviación estándar del esfuerzo de control $u(k)$.

En el capítulo 6 se presentan los resultados de la comparación de los controladores implementados en el robot de acuerdo con los índices de desempeño (3.4) y (3.5). Estos dos índices permiten establecer cuál de los controladores implementados presenta una mejor respuesta frente a los demás.

En la tabla 3.1 se aprecian los resultados de los índices de desempeño calculados para once (11) experimentos entre los que se incluye los datos presentados en la gráfica 3.2. El número total de datos tomados en cada prueba fue de 3000 con una referencia de 0° , que indica los grados de inclinación que debe tener el robot para estar en equilibrio. Posteriormente, se realizó un promedio de estos índices con el fin de obtener un valor estándar de los mismos. El promedio de la desviación del error calculado para los once experimen-

tos realizados para el controlador PID es de 1.95° que indica los grados desviación de la posición angular del robot móvil pendular con respecto al *set point* establecido.

Experimento	$\varepsilon_{e_{PID}}(^{\circ})$	$\varepsilon_{u_{PID}}(\%)$
1	1.2006	22.4221
2	4.3408	64.1187
3	2.3096	41.9113
4	2.5081	47.7092
5	2.3408	47.2018
6	1.2140	24.1228
7	1.8199	34.0127
8	1.9620	36.8070
9	1.7442	45.9195
10	0.6451	13.5123
11	1.3649	37.7534
Promedio	1.95	37.7719

Tabla 3.1: Resultados de los índices de desempeño obtenidos con la implementación de un controlador PID

En las figuras 3.4 y 3.5 se aprecian los histogramas de desviaciones estándar del error y del esfuerzo de control respectivamente para el controlador PID

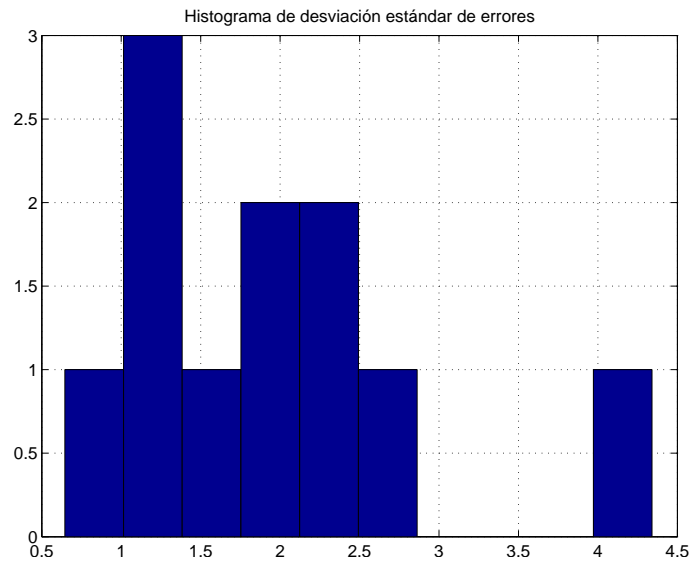


Figura 3.4: Histograma de desviaciones estándar del error para el controlador PID.

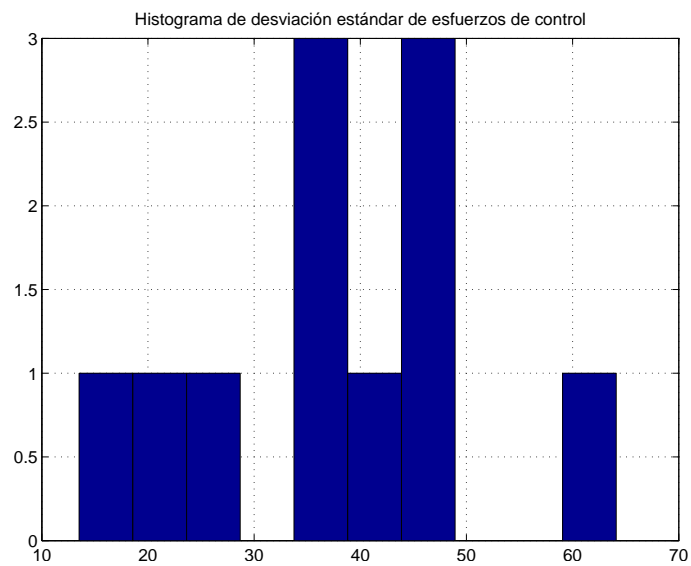


Figura 3.5: Histograma de desviaciones estándar del esfuerzo de control para el controlador PID.

En los histogramas anteriores se aprecia gráficamente la relación existente entre cada uno de los experimentos realizados debido a que asocia los resultados que presentan similares

características. Se puede observar que en 3 de los experimentos realizados la desviación estándar del error presentó un valor dentro del rango $[1, 1.5]$, sin embargo la mayor cantidad de desviaciones estándar se concentraron en el rango $[1.7, 2.5]$, dentro del cual se encuentra el valor medio de la desviación estándar del error. En el caso de la desviación estándar del esfuerzo de control, se presentaron 3 experimentos en los cuales el índice de desempeño se encontró en el rango $[35, 40]$ y otros 3 experimentos en donde las desviaciones se ubicaron en el rango $[45, 49]$. El promedio de las desviaciones del esfuerzo de control se presentó en el valor de 37.7719.

3.4.2. Ángulo crítico de control

El ángulo crítico de control es el valor máximo de la posición angular del robot, para el cual el controlador es capaz de realizar la convergencia a la referencia dada. Para determinar este valor se llevó a cabo un experimento, el cual consistió en poner en funcionamiento el robot con una inclinación inicial entre -1° y $+1^\circ$, la cual fue incrementada de manera gradual hasta que la plataforma no fuera capaz de seguir la referencia establecida, por tanto, el valor de ángulo crítico corresponderá a la primera lectura de inclinación cuando el robot perdió el equilibrio. Para el controlador PID, el ángulo de inclinación inicial máximo obtenido para este experimento fue de $[-4, 4]^\circ$ debido a que la acción integral acumula error a causa de la posición angular en la que se encuentra el robot al iniciar la prueba, por tanto la ley de control sobreactúa en éste haciendo que pierda el equilibrio con valores mayores a los descritos. Los resultados conseguidos para este experimento se muestran en las figura 3.6 y 3.7.

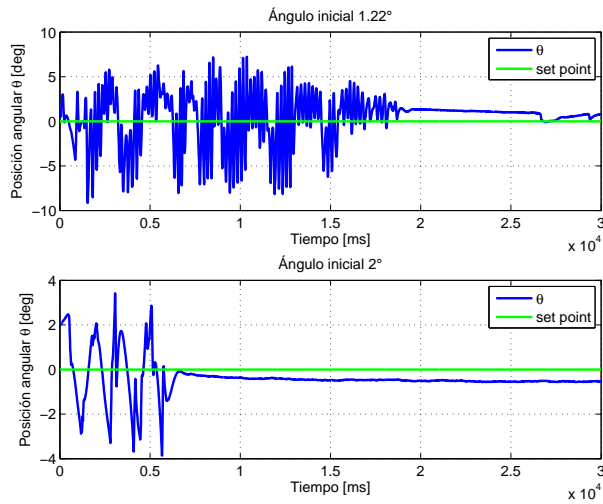


Figura 3.6: Experimento donde se coloca el robot móvil pendular *InstaBot SRAT-2* en diferentes posiciones iniciales. *i)* Arriba: Posición inicial $+1.22^\circ$. *ii)* Abajo: Posición inicial $+2.0^\circ$.

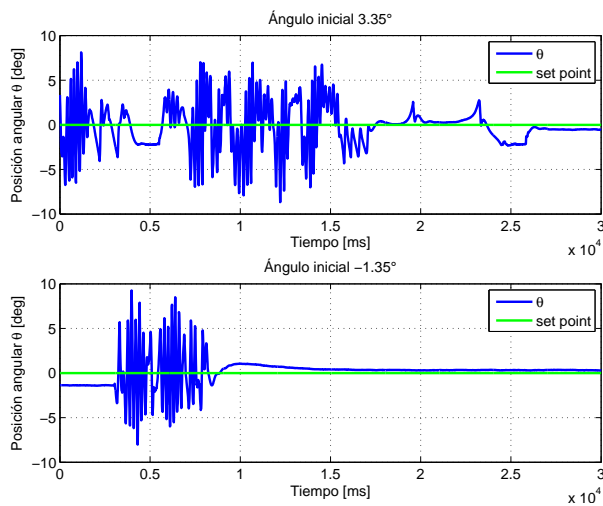


Figura 3.7: Experimento donde se coloca el robot móvil pendular *InstaBot SRAT-2* en diferentes posiciones iniciales. *i)* Arriba: Posición inicial $+3.35^\circ$. *ii)* Abajo: Posición inicial -1.35° .

Capítulo 4

Control LQR con filtro de Kalman

En este capítulo se hará una descripción del funcionamiento e implementación del controlador LQR (*Linear Quadratic Regulator*) en el robot móvil pendular *InstaBot SRAT-2* junto con la implementación del filtro de Kalman para la estimación del ángulo de orientación del robot móvil pendular.

4.1. Control LQR

Como se mencionó en la sección 3.1, el controlador LQR es lineal al igual que el PID clásico. El diseño de este controlador se realiza con base en un modelo lineal en espacio de estados de la planta o proceso a controlar. Para este trabajo en particular se utilizó el modelo descrito por la ecuación 1.15, donde cada uno de los estados del sistema debe ser conocido o estimado a partir de las variables medidas [1]. La ley de control está definida mediante la siguiente ecuación:

$$u = -Kx \tag{4.1}$$

Donde:

K : Matriz de ganancia de realimentación de estados

x : Vector de estados del sistema

Reemplazando la ecuación 4.1 en 1.15 se obtiene un nuevo modelo en espacio de estados del sistema descrito por:

$$\dot{x} = (A - BK)x \quad (4.2)$$

El objetivo del controlador LQR es por tanto ubicar cada uno de los valores propios de la matriz $(A - BK)$ en el semiplano izquierdo del plano s de tal forma que la dinámica de 4.2 sea estable. Para el cálculo de las componentes de K se utiliza la siguiente función de costo cuadrático, que representa la energía total del sistema en lazo cerrado.

$$J = \frac{1}{2} \int_0^{\infty} x^T(t)Wx(t) + u^T(t)Ru(t) dt \quad (4.3)$$

Siendo W una matriz de $n \times n$ semidefinida positiva y R de $m \times m$ definida positiva¹.

Valores grandes en la matriz W implican un decaimiento rápido de los estados a cero ya que ubican los autovalores del sistema en lazo cerrado muy a la izquierda en el plano s , penalizándose así los estados del sistema. Por otro lado, cuando el valor de la matriz R es demasiado grande con respecto a W , se penaliza la ley de control, haciendo que se tenga un esfuerzo de control menor. Por lo anterior deben considerarse cuidadosamente los valores de estas matrices ya que determinarán finalmente la ubicación de los polos en lazo cerrado del sistema mediante la matriz K .

4.2. Controlabilidad del modelo linealizado

Los valores propios de $A - KB$ pueden imponerse a partir de una adecuada selección de K siempre y cuando la matriz de controlabilidad definida en función de A y B sea de rango completo. Para el modelo lineal de cuarto orden obtenido en la sección 1.2, dicha matriz corresponde a:

$$M_c = \begin{bmatrix} A^3 B & A^2 B & AB & B \end{bmatrix} \quad (4.4)$$

¹Se resalta que una matriz M de $n \times n$ es definida positiva si para todo vector x en \mathbb{R}^n no nulo, el escalar $x^T M x$ es siempre positivo.

Al reemplazar las matrices A y B descritas respectivamente por las ecuaciones (1.23) y (1.24), se obtiene:

$$M_c = \begin{bmatrix} -\frac{g l^3 m^3 r^3}{d_n^2} & 0 & \frac{r(m l^2 + J_2)}{d_n} & 0 \\ -\frac{g l^2 m^2 r c_n}{d_n^2} & 0 & \frac{l m r}{d_n} & 0 \\ 0 & -\frac{g l^3 m^3 r^3}{d_n^2} & 0 & \frac{r(m l^2 + J_2)}{d_n} \\ 0 & -\frac{g l^2 m^2 r c_n}{d_n^2} & 0 & \frac{l m r}{d_n} \end{bmatrix} \quad (4.5)$$

Siendo:

$$c_n = J_1 + M r^2 + m r^2$$

$$d_n = J_1 J_2 + J_2 M r^2 + J_1 l^2 m + J_2 m r^2 + M l^2 m r^2$$

Una inspección detallada de M_c pone en evidencia que sus cuatro columnas son linealmente independientes. Esta afirmación puede corroborarse con ayuda de la función `rank` del *toolbox* de matemática simbólica de *Matlab*. De esta manera se concluye que el robot móvil pendular es controlable alrededor del punto de equilibrio $x_1 = 0$, $x_2 = \pi/2$, $x_3 = 0$ y $x_4 = 0$.

4.3. Cálculo de la ganancia de realimentación del vector de estados

Para el cálculo de la ganancia de realimentación K se parte del modelo en espacio de estados definido por las ecuaciones (1.23) y (1.24) y los valores de las constantes del robot móvil pendular encontradas en el capítulo 2. Teniendo en cuenta la forma en que la función de coste definida por la ecuación (4.3) penaliza los estados y el esfuerzo de control, se procedió a establecer una penalización mayor para el estado relacionado con la posición angular debido a que es el estado de mayor interés para la estabilización del robot móvil pendular. De igual forma se penalizó el esfuerzo de control definiendo un valor pequeño

para R , permitiendo así que la ley de control opere con grandes valores de esfuerzo de control. Para el control LQR del robot móvil pendular *InstaBot SRAT-2* se definieron los siguientes valores para W y R :

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 190 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.7 \end{bmatrix} \quad R = 0.1$$

Mediante la utilización de la función `lqr` de *Matlab* se procedió a calcular la ganancia de realimentación siendo esta:

$$K = \begin{bmatrix} 3.1623 & 44.1759 & 6.2140 & 0.3223 \end{bmatrix}$$

Con este valor de ganancia los autovalores del sistema en lazo cerrado se ubican en:

$$1.0E + 03 \begin{bmatrix} -1.9481 + 0.0000i \\ -0.0488 + 0.0000i \\ -0.0007 + 0.0005i \\ -0.0007 - 0.0005i \end{bmatrix}$$

Se comprueba que la parte real de todos los autovalores de la matriz que describe la dinámica de la planta se ubican en el semiplano izquierdo del plano s , con lo cual se concluye su estabilidad en lazo cerrado. Con respecto a la estimación de los estados se resalta que si se consideran como salidas medidas la posición angular y su derivada con respecto al tiempo, el sistema no será completamente observable. Esto se corrobora mediante la matriz de observabilidad definida por:

$$M_o = \begin{bmatrix} C \\ C A \\ C A^2 \\ C A^3 \end{bmatrix} \quad (4.6)$$

Reemplazando las matrices A y C descritas respectivamente por las ecuaciones (1.23) y

(1.2), se obtiene la matriz de observabilidad descrita por 4.7. Mediante el comando `rank` de *Matlab* se verifica el número de columnas linealmente independientes de esta matriz de observabilidad, siendo este valor (2) menor al número de estados del sistema (4), con lo cual se concluye que existen dos estados que no son observables con la información obtenida a partir de las salidas medidas. Los estados no observables corresponden a la posición lineal x y la velocidad lineal \dot{x} como se observa en la matriz (4.7).

$$M_o = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{g l m c_n}{d_n} & 0 & 0 \\ 0 & -\frac{g l m c_n}{d_n} & 0 & 0 \\ 0 & 0 & 0 & -\frac{g l m}{d_n} \\ 0 & 0 & 0 & -\frac{g l m c_n}{d_n} \\ 0 & \frac{g l^2 m^2 c_n^2}{d_n} & 0 & 0 \end{bmatrix} \quad (4.7)$$

Donde:

$$c_n = J_1 + M r^2 + m r^2$$

$$d_n = J_1 J_2 + J_2 M r^2 + J_1 l^2 m + J_2 m r^2 + M l^2 m r^2$$

Para implementar la ley de control por realimentación del vector de estados se supondrán x_1 y x_3 idénticamente iguales a cero. De acuerdo con lo anterior, la ley de control correspondiente al controlador LQR se puede redefinir como:

$$U_{LQR} = -K_2 x_2 - K_4 x_4 \quad (4.8)$$

Siendo K_2 y K_4 las posiciones 2 y 4 del vector K calculado anteriormente. Es importante

tener presente que la ley de control (4.8) es válida solo si el robot se inicia con velocidad lineal igual a cero ($\dot{x}(0) = 0$). Así, el sistema descrito anteriormente solo controla la orientación del robot pero no su posición lineal, por lo que el robot siempre finaliza con una posición lineal constante pero en general diferente de la del punto de partida.

4.4. Ecuaciones generales del filtro de Kalman

El filtro de Kalman es un algoritmo recursivo que utiliza un conjunto de ecuaciones en diferencias y datos de entrada (mediciones) para estimar los estados reales de un sistema dinámico sujeto a observación, mediante la minimización de la varianza estimada del error utilizando mínimos cuadrados.

El proceso matemático que realiza el filtro de Kalman para obtener la estimación del nuevo estado, se basa en un mecanismo de predicción y corrección, donde mediante una compensación de ganancia entre la estimación anterior y la observación actual, es posible lograr una convergencia a los estados reales del sistema, los cuales son desconocidos. Una explicación más detallada del funcionamiento del filtro de Kalman se puede apreciar en la figura 4.1.

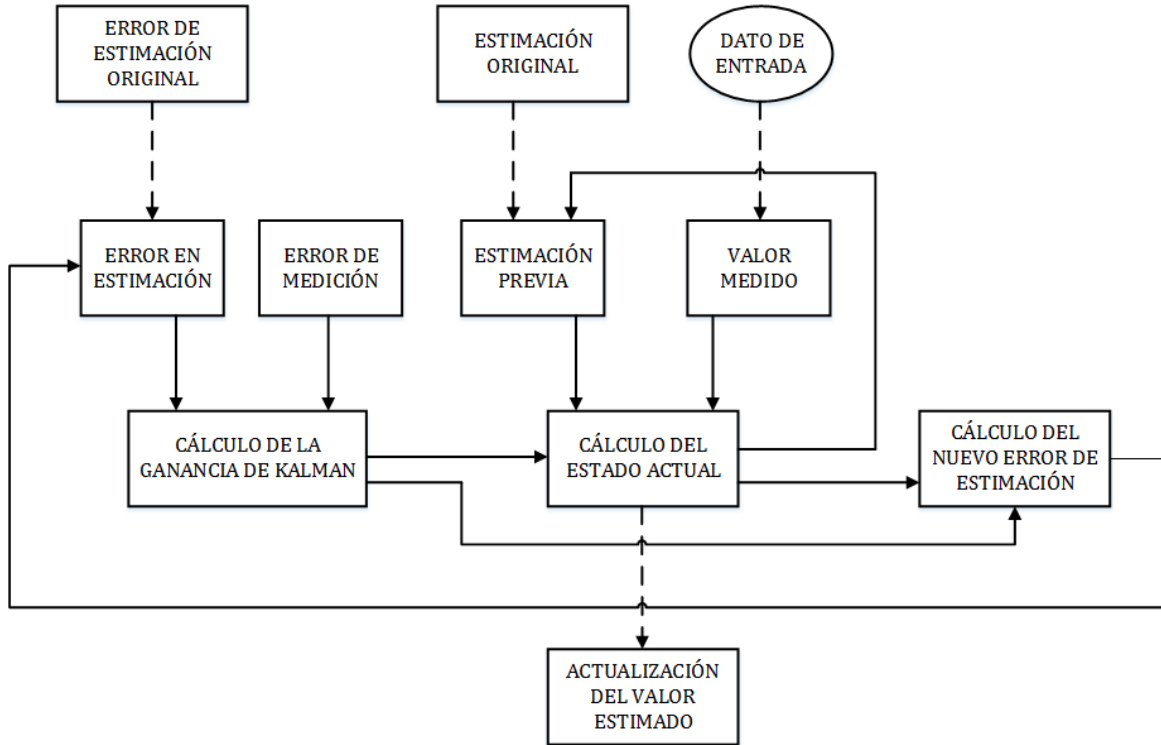


Figura 4.1: Diagramas en bloques del filtro de Kalman. [Elaboración propia]

El funcionamiento del filtro de Kalman se centra alrededor de 3 bloques, que aseguran su comportamiento como un algoritmo recursivo, siendo estos: (i) el cálculo de la ganancia de Kalman, (ii) cálculo del estado actual y (iii) cálculo del nuevo error de estimación. De forma general, el filtro de Kalman puede definirse mediante dos procesos [25] : En el primero se realiza la actualización de las variables en el tiempo (ecuaciones 4.9 y 4.10) mientras que en el segundo se lleva a cabo la actualización de los valores de medición y estimación (ecuaciones 4.11 , 4.12 y 4.13).

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (4.9)$$

$$\sigma_k^- = A\sigma_{k-1}A^T + Q \quad (4.10)$$

$$G_k = \sigma_k^- C^T (C\sigma_k^- C^T + \rho)^{-1} \quad (4.11)$$

$$\hat{x}_k = \hat{x}_k^- + G_k(z_k - C\hat{x}_k^-) \quad (4.12)$$

$$\sigma_k = (I - G_k C)\sigma_k^- \quad (4.13)$$

Donde:

$A \in \mathbb{R}^{n \times n}$: Matriz de estados del sistema

$B \in \mathbb{R}^{n \times p}$: Matriz de entradas del sistema

$C \in \mathbb{R}^{m \times n}$: Matriz de salidas del sistema

$Q \in \mathbb{R}^{n \times n}$: Matriz de covarianza del ruido del proceso

$I \in \mathbb{R}^{n \times n}$: Matriz identidad

$G_k \in \mathbb{R}^{n \times m}$: Matriz de ganancia dinámica de Kalman

$\sigma_k^- \in \mathbb{R}^{n \times n}$: Matriz de covarianza de error de estimación *a priori*

$\sigma_k \in \mathbb{R}^{n \times n}$: Matriz de covarianza de error de estimación *a posteriori*

$\rho \in \mathbb{R}^{n \times n}$: Matriz de covarianza de ruido de medida

$k \in \mathbb{Z}^+$: Índice de tiempo

$\hat{x}_k^- \in \mathbb{R}^n$: Vector de estados estimados *a priori*

$\hat{x}_k \in \mathbb{R}^n$: Vector de estados estimados *a posteriori*

$u_k \in \mathbb{R}^p$: Vector de control

$z_k \in \mathbb{R}^m$: Vector de mediciones

A través de estos procesos, el filtro de Kalman logra una minimización de la varianza de los errores de estimación del vector de estados.

4.5. Estimación de la orientación del robot móvil pendular *InstaBot SRAT-2* utilizando un filtro de Kalman

Para la presente sección, el filtro de Kalman se utilizará con el propósito de mejorar la estimación del ángulo de orientación del robot móvil pendular *InstaBot SRAT-2*, mediante una fusión sensorial del giroscopio y del acelerómetro. Cada uno de estos sensores aporta datos diferentes que pueden combinarse para obtener una medición más precisa y confiable.

Los datos medidos con el acelerómetro están relacionados con la orientación angular y los datos entregados por el giroscopio son velocidades angulares. De acuerdo con lo anterior, se define el siguiente modelo matemático:

$$\begin{aligned}\dot{x} &= u \\ \hat{y} &= x\end{aligned}\tag{4.14}$$

Donde :

\dot{x} :Velocidad angular
 u :Lectura del giróscopo
 \hat{y} :Posición angular
 x :Posición angular

Realizando una discretización utilizando un mantenedor de orden cero (*zoh*) se obtiene la siguiente ecuación en diferencias:

$$\begin{aligned}x(kh + h) &= Ax(kh) + Bu(kh) \\ y(kh) &= Cx(kh) + Du(kh)\end{aligned}\tag{4.15}$$

Donde:

$$\begin{aligned}A &= e^{A_c h} \\ B &= \int_0^h e^{A_c w} dw B_c\end{aligned}$$

Siendo: h el período de muestreo, A_c y B_c , las matrices dinámicas y de entrada del modelo de tiempo continuo que se va a discretizar.

Aplicando la discretización al sistema dinámico en tiempo continuo definido en 4.14 para el cual $A_c = 0$ y $B_c = 1$ se obtiene como resultado la ecuación en diferencias mostrada en

4.16.

$$\begin{aligned}x(kh + h) &= x(kh) + hu(kh) \\ y(kh) &= x(kh)\end{aligned}\tag{4.16}$$

Al reemplazar $A = 1$, $B = h$ y $C = 1$ en las ecuaciones del filtro de Kalman descritas por (4.9), (4.10), (4.11), (4.12) y (4.13) se tiene lo siguiente:

- Predicción:

$$\begin{aligned}\hat{x}_k^- &= \hat{x}_{k-1} + h u_k \\ \sigma_k^- &= \sigma_{k-1} + Q\end{aligned}\tag{4.17}$$

- Actualización:

$$\begin{aligned}G_k &= \sigma_k^- (\sigma_k^- + \rho)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + G_k (z_k - \hat{x}_k^-) \\ \sigma_k &= (1 - G_k) \sigma_k^-\end{aligned}\tag{4.18}$$

Para este sistema en particular, u_k es la velocidad angular medida con el gir6scopo, z_k es la orientaci6n estimada a partir de las medidas entregadas por el aceler6metro y \hat{x}_k es la orientaci6n estimada por el filtro de Kalman. Las dem6s variables tienen el significado descrito en la secci6n precedente.

4.6. Resultados

La l6gica del controlador LQR fue realizada en c6digo C e implementada en el entorno *Arduino*. La sintonizaci6n de los par6metros del controlador se llev6 a cabo utilizando el software *Matlab* y el modelo linealizado en espacio de estados obtenido en el cap6tulo 1 (ecuaci6n (1.2)), reemplazando las constantes del modelo por las encontradas en el cap6tulo 2. Para la implementaci6n del filtro de Kalman se utiliz6 la librer6a *KalmanFilter*² de *TKJElectronics*. Los resultados mostrados en las gr6ficas 4.2 y 4.3 corresponden al control LQR con el filtro de Kalman implementado.

² *Kalman filter* . Disponible en <https://github.com/TKJElectronics/KalmanFilter>

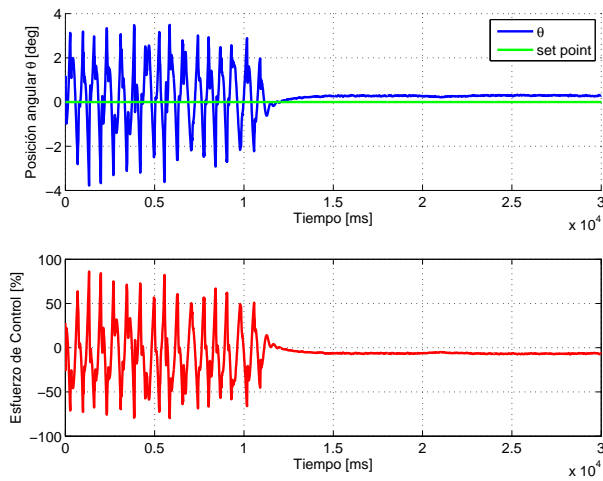


Figura 4.2: Resultados de la implementación de un controlador LQR con filtro de Kalman en el robot móvil pendular *InstaBot SRAT-2*. *i)* Arriba: Posición angular obtenida (línea azul) y referencia establecida (línea verde). *ii)* Abajo: Esfuerzo de control.

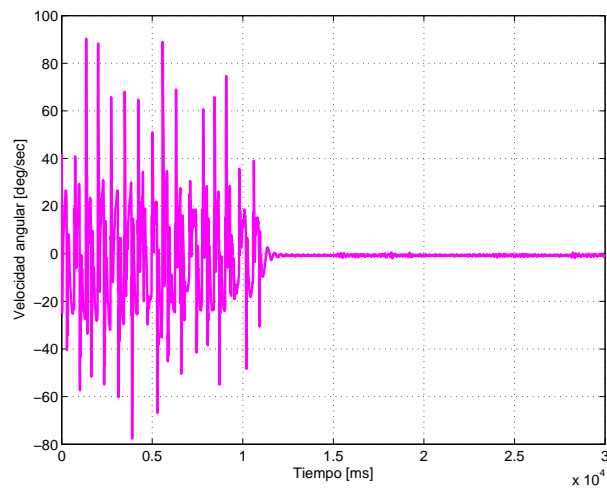


Figura 4.3: Resultados de la implementación de un controlador LQR con filtro de Kalman en el robot móvil pendular *InstaBot SRAT-2* (Velocidad Angular).

4.6.1. Índices de desempeño

En esta subsección se presentan los valores numéricos de los índices de desempeño definidos en la subsección 3.4.1, para tal fin se hará un análisis estadístico de los datos obtenidos en once (11) experimentos, siendo estos resultados semejantes a los mostrados en la figura 4.2. Como se muestra en la tabla 4.1, el error medio de desviación con respecto al seguimiento de la referencia para el controlador LQR con filtro de Kalman es aproximadamente 1.1559° con un esfuerzo de control medio del 26.4835% .

Experimento	$\varepsilon_{e_{LQR}}(^{\circ})$	$\varepsilon_{u_{LQR}}(\%)$
1	1.5647	35.8108
2	1.1908	27.284
3	0.5838	13.2977
4	1.4190	32.5499
5	1.5802	36.2453
6	0.6245	14.3426
7	1.0562	24.2305
8	0.9478	21.7034
9	0.9266	21.2190
10	1.4332	32.8341
11	1.3886	31.8014
Promedio	1.1559	26.4835

Tabla 4.1: Resultados de los índices de desempeño obtenidos con la implementación de un controlador LQR con filtro de Kalman.

Los índices de desempeño mostrados en la tabla 4.1 fueron calculados aplicando las ecuaciones 3.4 y 3.5 definidas en en la subsección 3.4.1 del capítulo 3. A continuación se muestran en las figuras 4.4 y 4.5 los histogramas de desviaciones estándar del error y esfuerzo de control respectivamente para el controlador LQR con filtro de Kalman.

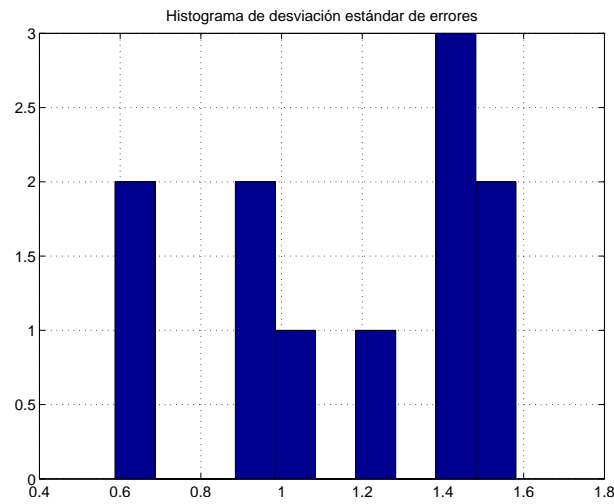


Figura 4.4: Histograma de desviaciones estándar del error para el controlador LQR con filtro de Kalman

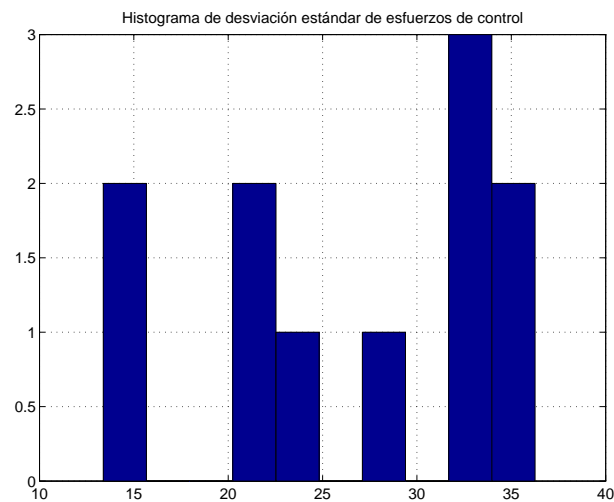


Figura 4.5: Histograma de desviaciones estándar del esfuerzo de control para el controlador LQR con filtro de Kalman

En los histogramas 4.4 y 4.5 se muestra la distribución estadística de los índices de desempeño propuestos. Para el caso de la desviación estándar del error se observó que en 3 de los experimentos, su valor se encontró en el rango $[1.4, 1.5]$, sin embargo, su promedio se

presentó en el valor de 1.1559. En cuanto a la desviación estándar del esfuerzo de control, se presentó una mayor concentración de datos en $[25, 40]$, rango dentro del cual se encontró su valor promedio en 26.4835.

4.6.2. Ángulo crítico de control

Para encontrar el ángulo crítico de control se llevó a cabo el experimento descrito en la subsección 3.4.2. A diferencia del control PID, el LQR no presenta el inconveniente de la saturación de la ley de control debido a que la realimentación de estados es similar a un control PD, en donde no existe un componente acumulativo que actúe en función de los errores pasados, por este motivo fue posible realizar la prueba del ángulo inicial, sin comprometer la calidad de la ley de control entregada. En las figuras 4.6 y 4.7 se aprecian las curvas de respuesta obtenidas para los experimentos de ángulo inicial, para valores positivos y negativos.

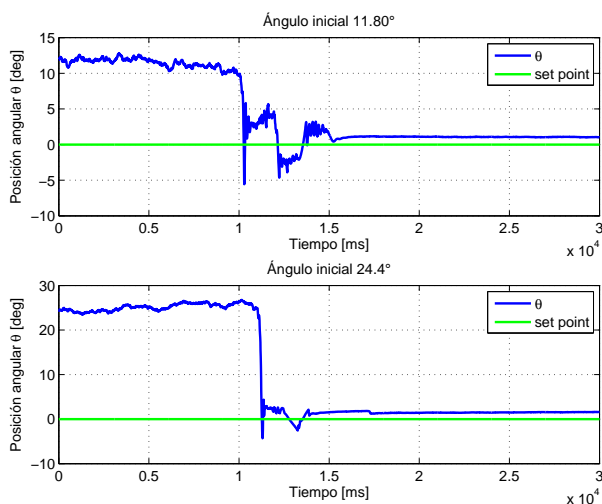


Figura 4.6: Experimento donde se coloca el robot móvil pendular *InstaBot SRAT-2* en diferentes posiciones iniciales. *i)* Arriba: Posición inicial $+11.8^\circ$. *ii)* Abajo: Posición inicial $+24.4^\circ$.

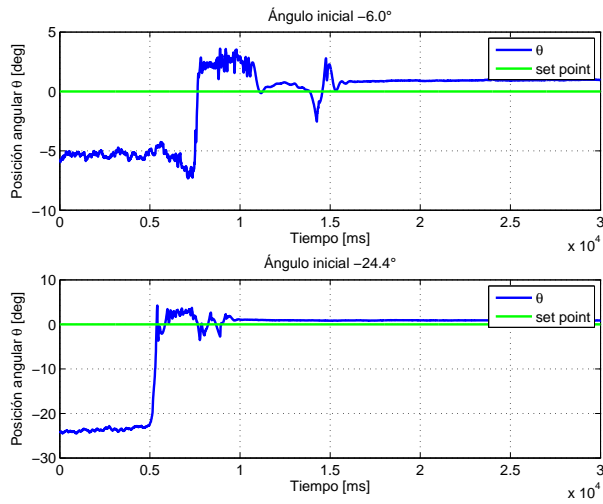


Figura 4.7: Experimento donde se coloca el robot móvil pendular *InstaBot SRAT-2* en diferentes posiciones iniciales. *i)* Arriba: Posición inicial -6.0° . *ii)* Abajo: Posición inicial -24.4° .

El ángulo crítico de control encontrado con este experimento oscila alrededor de $[-25, 25]^\circ$. Con esto se comprueba el rango de valores para los cuales el controlador LQR con filtro de Kalman es capaz de realizar la convergencia al *set point* establecido.

Capítulo 5

Control difuso

En este capítulo se presentará una descripción del controlador difuso implementado en la plataforma móvil *InstaBot SRAT-2*, detallando los conjuntos difusos definidos, la base de reglas establecidas, los métodos de fusificación y defusificación utilizados y el proceso de inferencia realizado.

5.1. Control no Lineal

La implementación de cualquier sistema de control requiere de un conocimiento previo del comportamiento de la planta, que generalmente se describe por modelos matemáticos no lineales representados por ecuaciones en diferencias o diferenciales. Dichos modelos son linealizados alrededor de puntos de operación o equilibrio y limitan el funcionamiento del controlador, por tanto se han estudiado técnicas de control no lineal que permitan solventar este problema, entre dichas técnicas se pueden mencionar: las redes neuronales, los controladores difusos, los algoritmos genéticos, entre muchas otras. Cada una de estas técnicas aborda el problema del control desde diferentes perspectivas, aportando así características únicas en el diseño del controlador. Este capítulo se centrará en el diseño de un controlador no lineal basado en lógica difusa para el robot móvil pendular *InstaBot SRAT-2*.

5.2. Control Difuso

El control difuso sienta sus bases en la lógica difusa, desarrollada en 1965 por Lotfi Asker Zadeh, un ingeniero y matemático azerbaiyano de la Universidad de Columbia. La lógica difusa presenta una alternativa al razonamiento clásico y la teoría de conjuntos, introduciendo los *grados de verdad* en una proposición y el *grado de pertenencia* a un conjunto como valores reales en el rango $[0, 1]$. Esto con el fin de lograr una aproximación más realista del razonamiento humano donde los límites no están definidos [26]. Con base en este raciocinio es posible plantear controladores no lineales que operen con reglas heurísticas de la forma *If X Then Y*, siendo X el antecedente y Y el consecuente, ambas variables lingüísticas traducidas a conjuntos difusos. La arquitectura de un controlador difuso está definida por 4 módulos:

1. **Fusificador:** Este módulo es el encargado de convertir las entradas numéricas reales, asociadas a variables lingüísticas en conjuntos difusos admisibles por el controlador.
2. **Reglas:** Se encarga de almacenar la base de reglas heurísticas del controlador con sus respectivos antecedentes y consecuentes.
3. **Motor de inferencia:** Encargado de simular el proceso de razonamiento, realizando una inferencia difusa a partir de la base de reglas establecidas para entregar el conjunto difuso de salida.
4. **Defusificador:** Módulo encargado de transformar el conjunto difuso de salida en el valor numérico real que mejor lo represente a partir de diferentes métodos (centroide de área, bisector de área, promedio de máximos, entre otros).

El diagrama de bloques de un controlador difuso se puede apreciar en la figura 5.1 :

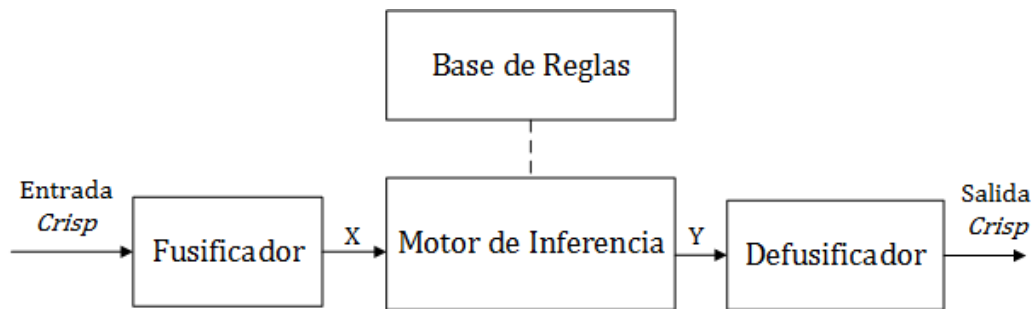


Figura 5.1: Diagrama en bloques de un controlador difuso. [Elaboración Propia]

5.3. Control PD difuso aplicado a un robot móvil pendular

En esta sección se detalla el diseño del controlador difuso utilizado en el robot móvil pendular *InstaBot SRAT-2*, las variables lingüísticas, los conjuntos difusos y la base de reglas definidas. La variable manipulada y la variable controlada utilizadas para el diseño de este controlador son las mismas que se estipularon en la sección 3.3.1.

Un control PD difuso, al igual que un PD convencional opera con base en el valor actual de la señal de error y de su derivada con respecto al tiempo, ambas variables entregadas respectivamente por el acelerómetro y el giroscopio de la MPU6050 del robot, como resultado se obtiene un controlador MISO con 2 entradas y una salida, tal como lo muestra la figura 5.2.

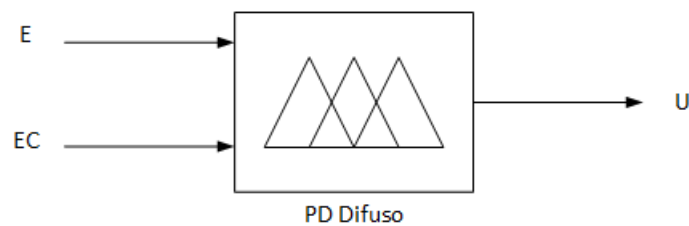


Figura 5.2: Diagrama de bloques de un controlador difuso PD. [Elaboración Propia]

La variable E corresponde al valor del error de posición actual mientras que EC se refiere

al cambio o derivada del error y la variable de salida U es el esfuerzo de control aplicado a los motores de las ruedas del robot.

5.3.1. Definición de conjuntos difusos

En los controladores difusos, las variables lingüísticas juegan un papel fundamental a la hora de establecer el número de conjuntos difusos requeridos para modelar el comportamiento de la ley de control deseada. Para el caso del robot móvil pendular *InstaBot SRAT-2* se definieron 3 conjuntos difusos para cada una de las variables de entrada, cuyas funciones de membresía se pueden apreciar en las figuras 5.3 y 5.4 y 4 conjuntos difusos para la variable de salida, tal como se muestra en la figura 5.5. En la tabla 5.1 se muestran de forma detallada las variables lingüísticas utilizadas y los conjuntos difusos definidos para el control PD del robot.

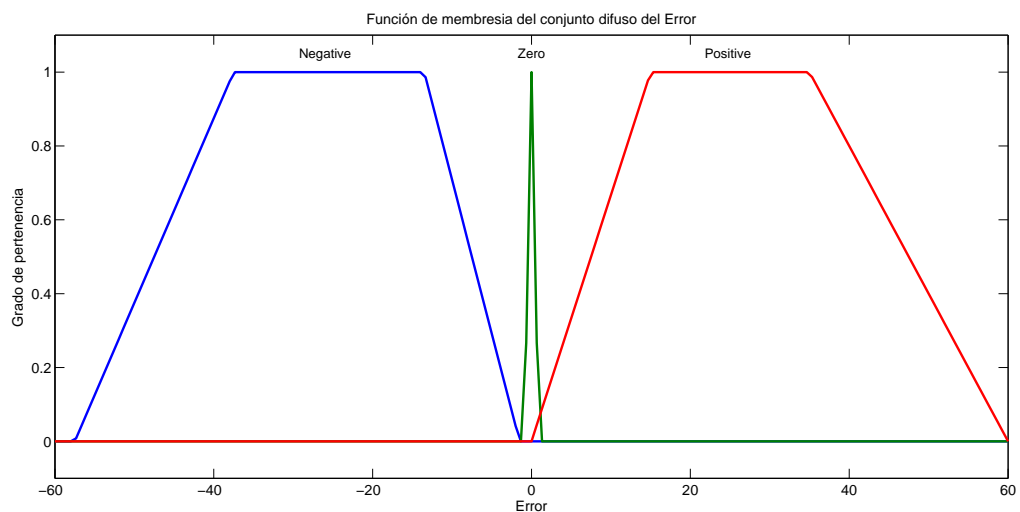


Figura 5.3: Funciones de membresía asociadas a los conjuntos difusos de la entrada *Error*.

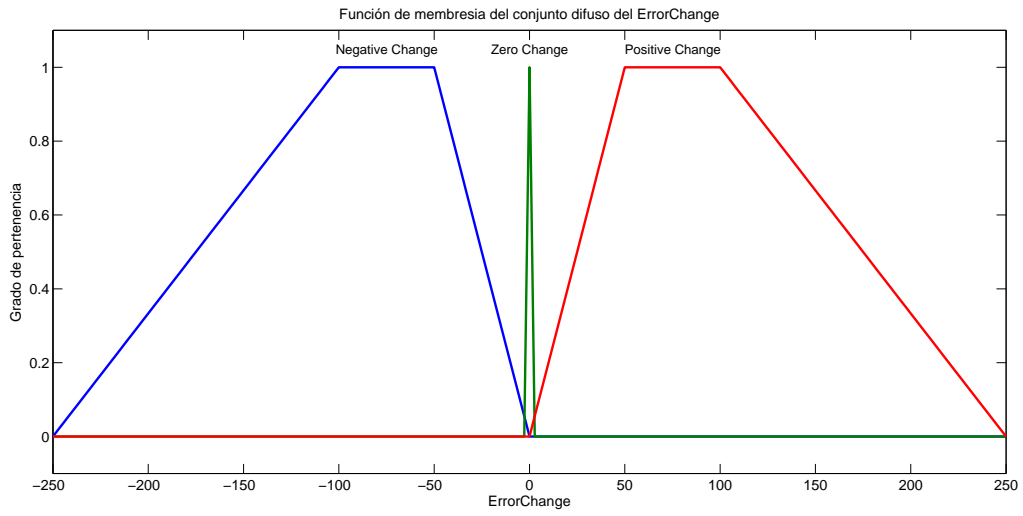


Figura 5.4: Funciones de membresía asociadas a los conjuntos difusos de la entrada *Error-Change*.

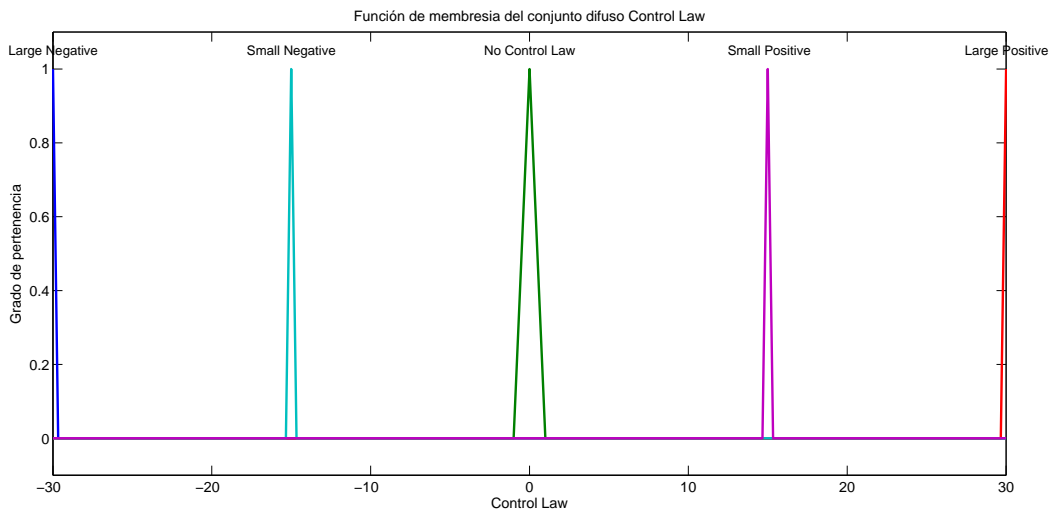


Figura 5.5: Funciones de membresía asociadas a los conjuntos difusos de la Salida *Control Law*.

Nombre de la Variable	Universo de discurso	Variables lingüísticas asociadas	Tipo de función de membresía	Valores Límite de la función de membresía	Descripción
Error (E)	[-60, 60] deg	Negative(N)	Trapezoidal	[-60, -35, -15, 0]	La diferencia entre la posición medida y la referencia es negativa
		Zero(Z)	Trapezoidal	[-0.8, -0.3, 0.3, 0.8]	El robot se encuentra en la posición deseada
		Positive(P)	Trapezoidal	[0, 15, 35, 60]	La diferencia entre la posición medida y la referencia es positiva
Error Change (EC)	[-250, 250] deg/sec	Negative Change(NC)	Trapezoidal	[-250, -100, -50, 0]	La derivada del error es negativa
		Zero Change(ZC)	Trapezoidal	[-0.5, -0.2, 0.2, 0.5]	El valor del error se mantiene constante
		Positive Change(PC)	Trapezoidal	[0, 50, 100, 250]	La derivada del error es positiva
Control Law (U)	[-30, 30] %	Large Negative (LN)	Singleton	-30	Señal de control negativa de alta magnitud
		Small Negative (SN)	Singleton	-15	Señal de control negativa de baja magnitud
		No Control Law(NCL)	Triangular	[-1, 0, 1]	No se aplica control
		Small Positive(SP)	Singleton	15	Señal de control positiva de baja magnitud.
		Large Positive(LP)	Singleton	30	Señal de control positiva de alta magnitud.

Tabla 5.1: Definición de conjuntos para un controlador PD difuso.

Se utilizaron funciones *singleton* para caracterizar el universo de discurso de la ley de

control con el fin de facilitar el proceso de sintonización del controlador. De acuerdo con lo presentado en la figura 5.5 el único parámetro a sintonizar es la ubicación del conjunto *singleton Large Positive*. La ubicación del conjunto difuso *Small Positive* corresponderá a la mitad de la distancia entre cero y el conjunto *Large Positive*. Las distancias de los conjuntos restantes se determinan por simetría con respecto a los dos conjuntos anteriormente mencionados. El conjunto triangular *No Control Law* se ubica siempre alrededor de 0.

5.3.2. Base de Reglas

Las reglas establecidas en el controlador difuso son las encargadas de definir el comportamiento de la ley de control, escogiendo el conjunto difuso de salida (consecuente) más indicado ante las entradas verificadas (antecedentes) del sistema. Para el caso del robot móvil pendular *InstaBot SRAT-2* el controlador difuso PD diseñado cuenta con una base de 9 reglas, mostradas de forma matricial en la tabla 5.2.

	PC	ZC	NC
P	LP	SP	NCL
Z	SP	NCL	SN
N	NCL	SN	LN

Tabla 5.2: Matriz de Reglas del controlador difuso PD

Las celdas de color gris están asociadas a las entradas del control difuso, siendo las horizontales las correspondientes a la entrada *ErrorChange(EC)* y las verticales a la entrada *Error(E)*, las celdas sin color se asocian a la ley de control estipulada ante el par ordenado de entradas (EC, E) , según la notación definida en la tabla 5.1. Como resultado de los conjuntos difusos definidos y las reglas heurísticas establecidas se obtiene un diagrama de superficie, mostrado en la figura 5.6 .

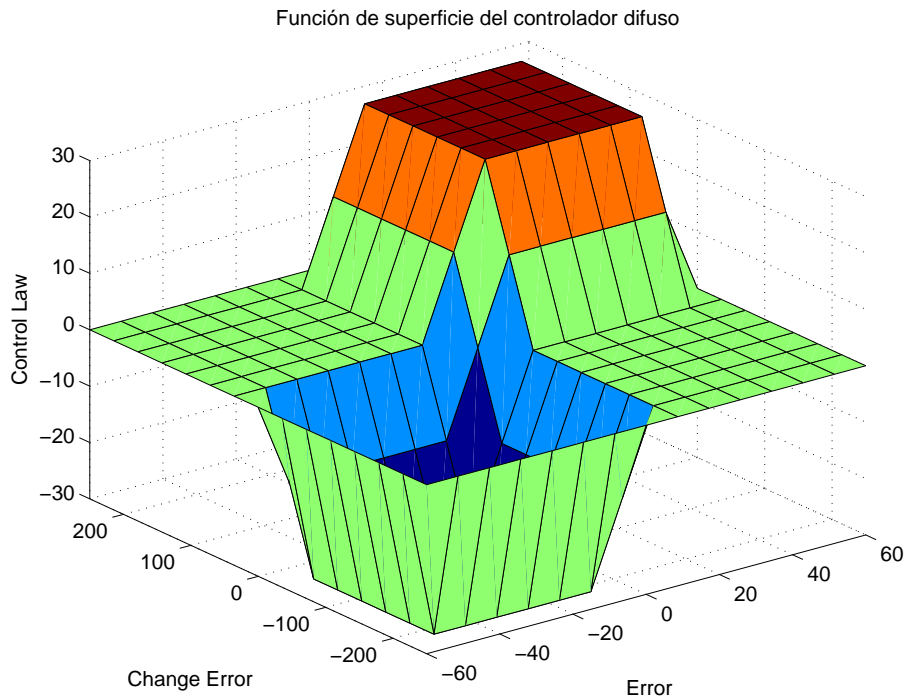


Figura 5.6: Diagrama de superficie del controlador difuso aplicado al robot móvil pendular *InstaBot SRAT-2*. Variables de entrada: *Error* (eje x), *Error Change*(eje y), variable de salida: *Control Law*(eje z)

5.4. Resultados

La implementación de la lógica difusa del controlador fue codificada en lenguaje *C* dentro del entorno Arduino. Para la definición de variables lingüísticas, conjuntos difusos y reglas se utilizó la librería *eFLL*¹. La librería hace uso del método de máximos y mínimos de Mandami para el motor de inferencia y la composición difusa y el método del centroide de área para la defusificación de conjuntos. Los resultados experimentales del controlador difuso aplicado al *InstaBot SRAT-2* se evidencian en la figura 5.7, donde se muestra el seguimiento a la referencia, de igual forma en la figura 5.8 se presenta el esfuerzo de control requerido para realizar lograr la estabilización del robot móvil pendular.

¹Embedded Fuzzy Logic Library. Disponible en <https://github.com/zerokol/eFLL>

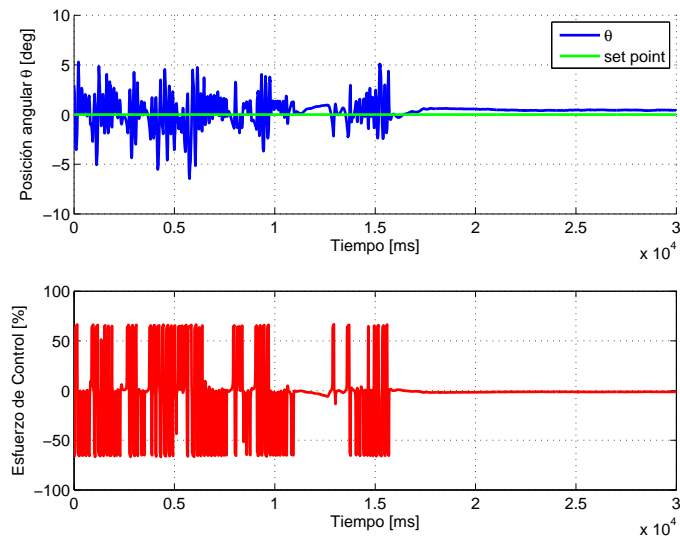


Figura 5.7: Resultados de la implementación de un controlador difuso en el robot móvil pendular *InstaBot SRAT-2*. *i)* Arriba: Posición angular obtenida (línea azul) y referencia establecida (línea verde). *ii)* Abajo: Esfuerzo de control.

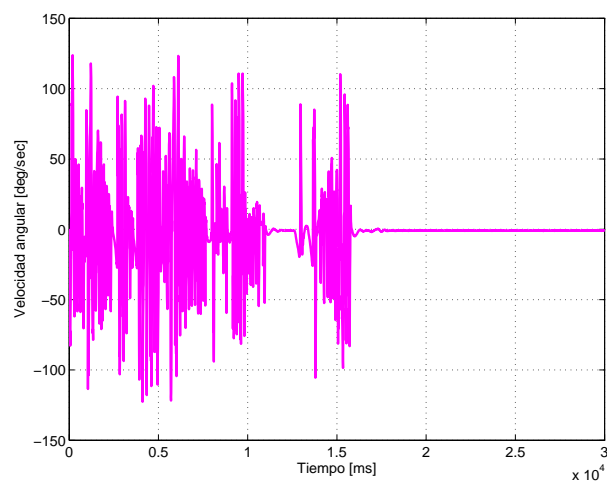


Figura 5.8: Resultados de la implementación de un controlador difuso en el robot móvil pendular *InstaBot SRAT-2* (Velocidad Angular).

5.4.1. Índices de desempeño

Al igual que en los capítulos 3 y 4 en esta subsección se abordarán los índices de desempeño definidos en 3.4.1 para el controlador difuso, para tal fin se llevaron a cabo once (11) experimentos para los cuales se evaluó el seguimiento a la referencia del robot. La tabla 5.3 muestra los índices de desempeño para cada uno de los experimentos, así como su promedio, siendo este valor 0.7689° para la desviación estándar del error y 12.4778% para la desviación estándar del esfuerzo de control.

Experimento	$\varepsilon_{e_{Fuzzy}} (^\circ)$	$\varepsilon_{u_{Fuzzy}} (\%)$
1	0.8088	20.1843
2	0.6166	13.6814
3	0.9268	14.7390
4	1.0976	28.5373
5	0.7386	17.5151
6	0.6166	12.9436
7	0.6405	2.8881
8	0.8045	5.0954
9	0.7299	10.5461
10	0.6900	5.1826
11	0.7881	5.9434
Promedio	0.7689	12.4778

Tabla 5.3: Resultados de los índices de desempeño obtenidos con la implementación de un controlador PD difuso.

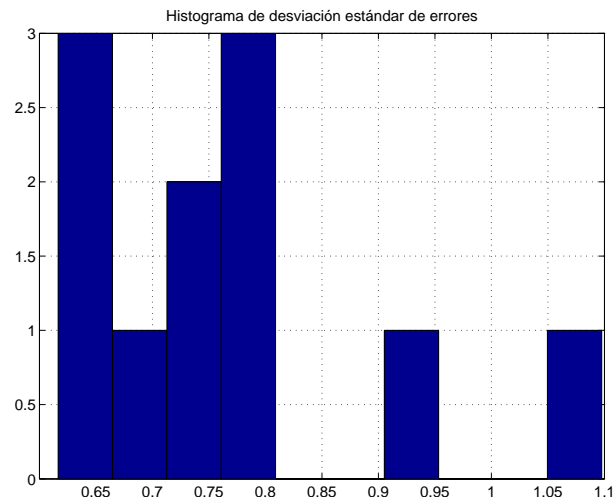


Figura 5.9: Histograma de desviaciones estándar del error para el controlador PD difuso

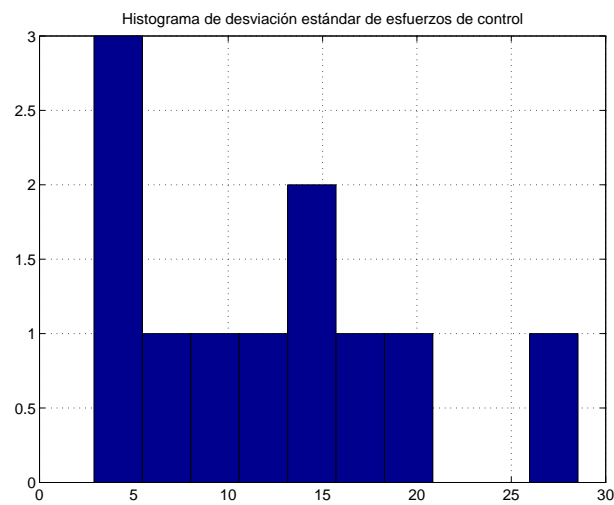


Figura 5.10: Histograma de desviaciones estándar del esfuerzo de control para el controlador PD difuso

En los histogramas anteriores se aprecian los índices de desempeño de los once experimentos realizados. La mayoría de las desviaciones estándar del error se concentraron en el rango $[0.5, 0.8]$ con un promedio de 0.7689. Para el caso de la desviación estándar del esfuerzo de control la distribución fue relativamente uniforme, con 3 desviaciones en el rango

[3, 5] y 2 desviaciones en el rango [13, 15] el valor medio de las desviaciones se produjo en 12.4778.

5.4.2. Ángulo crítico de control

Para hallar el ángulo crítico del controlador difuso se realizó el experimento descrito en la subsección 3.4.2 del capítulo 3. Los resultados obtenidos se presentan en la figura 5.11 tanto para ángulos positivos como negativos.

Dado que el controlador PD difuso diseñado no posee ningún componente con memoria, tal como la acción integral, es posible realizar el experimento del ángulo inicial sin comprometer la calidad de la ley de control entregada. Para el controlador difuso se obtuvieron mejores resultados con respecto a esta prueba como se evidencia en la figura 5.11 donde el ángulo inicial osciló entre $[-28.0, 28.0]^\circ$.

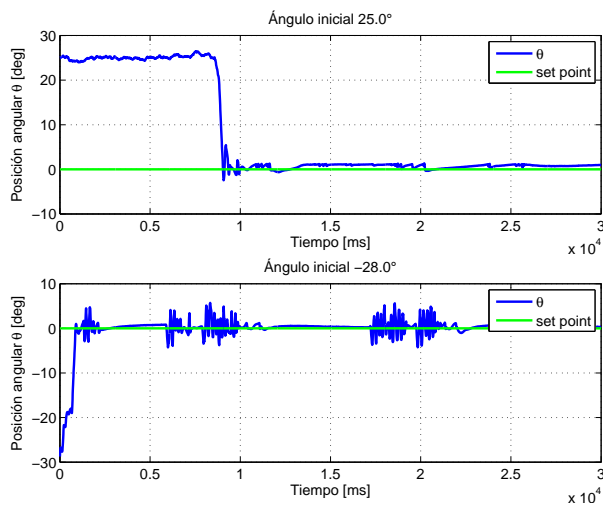


Figura 5.11: Experimento donde se coloca el robot móvil pendular *InstaBot SRAT-2* en diferentes posiciones iniciales. *i)* Arriba: Posición inicial $+25.0^\circ$. *ii)* Abajo: Posición inicial -28.0° .

Capítulo 6

Comparación del desempeño de los diferentes controladores implementados

En este capítulo se realiza la comparación de los resultados obtenidos con la implementación de los controladores presentados en los capítulos 3, 4 y 5. La comparación de los mismos se realizó teniendo en cuenta los índices de desempeño presentados en el capítulo 3, así como el rango de valores angulares iniciales para el cual el *InstaBot SRAT-2* tuvo convergencia al punto de equilibrio deseado.

6.1. Comparación con los índices de desempeño obtenidos

En la tabla 6.1 se muestran los índices de desempeño obtenidos al evaluar los controladores implementados.

Controlador	$\varepsilon_e(^{\circ})$	$\varepsilon_u(\%)$
PID	1.95	37.7719
LQR	1.1559	26.4835
Fuzzy	0.7689	12.4778

Tabla 6.1: Comparación de los promedios de las desviaciones estándar obtenidas para los 11 experimentos realizados con cada controlador.

Para obtener la relación de un controlador con respecto al otro se calculan los cocientes $\varepsilon_e/\varepsilon_{eFuzzy}$ y $\varepsilon_u/\varepsilon_{uFuzzy}$, siendo las desviaciones estándar del error y del esfuerzo de control del controlador difuso el marco de referencia establecido para la comparación de los controladores lineales. De acuerdo a lo anterior, se obtuvieron los siguientes cocientes:

- Cociente entre el promedio de la desviación estándar del error del controlador PID y el promedio de la desviación estándar del error del controlador difuso

$$\frac{1.95}{0.7689} = 2.5361$$

El resultado obtenido significa que la desviación estándar del error del controlador PID es 2.5361 veces mayor que la del error del controlador difuso.

- Cociente entre el promedio de la desviación estándar del error del controlador LQR con filtro de Kalman y el promedio de la desviación estándar del error del controlador difuso

$$\frac{1.1559}{0.7689} = 1.5033$$

El cociente calculado indica que la desviación estándar del error del controlador LQR es 1.503 veces mayor en relación con la del error del controlador difuso.

Para la comparación del esfuerzo de control los cocientes calculados se muestran a continuación:

- Cociente entre el promedio de la desviación estándar del esfuerzo de control del controlador PID y promedio de la desviación estándar del esfuerzo de control del

controlador difuso

$$\frac{37.7719}{12.4778} = 3.0271$$

Esto significa que la desviación estandar del la ley de control PID es 3.0271 veces más grande que la del controlador difuso.

- Cociente entre la desviación estándar promedio del esfuerzo de control del controlador LQR con filtro de Kalman y la desviación estándar promedio del esfuerzo de control del controlador difuso

$$\frac{26.4835}{12.4778} = 2.1224$$

Esto significa que la desviación estandar del la ley del controlador LQR es 2.1224 veces más grande que la del controlador difuso.

Los resultados de la tabla 6.1 muestran que el controlador difuso presenta un desempeño mejor en relación con los controladores lineales implementados debido a que tiene el menor error de desviación y un gasto energético menor.

6.1.1. Resumen de los ángulos de convergencia de los controladores implementados

En la tabla 6.2 se presenta la comparación de las regiones de convergencia encontradas para los 3 controladores implementados.

<i>Convergencia_{PID}</i>	<i>Convergencia_{LQR}</i>	<i>Convergencia_{Fuzzy}</i>	Mayor región convergencia
$[-4^\circ, 4^\circ]$	$[-25^\circ, 25^\circ]$	$[-28^\circ, 25^\circ]$	Fuzzy

Tabla 6.2: Comparación de las regiones de convergencia encontradas en los controladores implementados.

De acuerdo a lo evidenciado en la tabla 6.2, se determinó que el controlador difuso presenta una mayor región de atracción al punto de equilibrio para los puntos de operación establecidos. En cuanto al controlador LQR con filtro de Kalman, los resultados presentados

fueron satisfactorios, se obtuvo un mayor ángulo de convergencia, menor gasto energético y menor error que el control PID. El controlador PID por su parte, mostró un bajo desempeño en los experimentos relacionados al ángulo inicial, descritos en la subsección 3.4.2, debido a la presencia del componente integral, el cual depende de la acumulación de errores pasados. Al permanecer la plataforma móvil durante un tiempo considerable en un grado de inclinación, como lo exigía el experimento, la ley de control resultante se saturaba, produciendo que el robot perdiera el equilibrio limitando considerablemente la región de convergencia a la referencia establecida.

Capítulo 7

Conclusiones y trabajos futuros

El presente capítulo busca resolver la pregunta de investigación planteada en el inicio del trabajo, concluyendo con base en los resultados obtenidos, cuál es el mejor controlador para el robot móvil pendular *Instabot SRAT-2*.

7.1. Conclusiones

A continuación se muestran las conclusiones del presente trabajo:

- Se encontró el modelo dinámico en espacio de estados de un robot móvil pendular teniendo en cuenta el modelo físico del mismo y el subsistema eléctrico asociado a motores de las ruedas del robot. El modelo físico fue hallado mediante el formalismo de *Lagrange*, y el modelo del subsistema eléctrico se calculó mediante un diagrama circuital básico para un motor de corriente directa. Se realizó la implementación de tres controladores en la plataforma móvil *InstaBot SRAT-2*, los cuales correspondieron a dos controladores lineales (PID y LQR) y uno no lineal (*Fuzzy*). El diseño del controlador LQR se realizó con el modelo dinámico obtenido y se implementó un filtro de Kalman para mejorar la estimación del ángulo de orientación del robot móvil pendular.

- Para obtener el modelo en espacio de estados fue necesario hallar los valores de las constantes del robot, la mayor parte de los parámetros se encontraron mediante mediciones, sin embargo constantes como el momento de inercia fue necesario calcularlas por ecuaciones matemáticas ya que no fue posible obtener resultados satisfactorios del proceso de identificación que se quería realizar debido a la alta fricción seca presente en los motores del robot.
- La sintonización del controlador PID fue la más dispendiosa de realizar puesto que fue necesario configurar 3 variables, Kp , Ki y Kd , cada una de las cuales afectaba de manera distinta el desempeño del robot, mientras que en el controlador LQR la sintonización solo correspondió al cálculo del vector de ganancia asociado a la ley de control y finalmente con el controlador difuso solo se necesitó establecer adecuadamente las reglas y los rangos de los conjuntos difusos utilizados.
- Las pruebas realizadas para determinar la convergencia al ángulo deseado permitieron establecer un criterio adicional para determinar la ventaja de la utilización de un controlador con respecto a otro. En este trabajo de grado se halló que el controlador PD difuso tiene un mejor desempeño con respecto a los dos controladores lineales también implementados, verificado a través de los índices de desempeño y el ángulo inicial de control que fue superior al de los controladores lineales.
- Tanto el controlador PID como el PD difuso se implementaron fácilmente debido a que no se necesitó directamente el modelo matemático de la planta, mientras que para el LQR fue necesario obtener el modelo dinámico del robot. En algunos casos estos modelos pueden resultar excesivamente complejos y difíciles de identificar.
- Además de obtener una inspección visual de los resultados obtenidos de la implementación de cada controlador se calcularon índices de desempeño que permitieran conocer la desviación estándar del error y el gasto de energía empleado por los controladores para definir así criterios más sólidos sobre el mejor resultado encontrado con la implementación de un determinado controlador.
- En cada una de las pruebas realizadas se encontró una mejor estabilidad en superficies rugosas debido a que las ruedas del robot se adhieren mejor a la superficie, evitando el deslizamiento continuo que se presentaría en superficies completamente lisas.

- De acuerdo a los resultados obtenidos, se concluyó que el controlador PD difuso es el mejor controlador para la plataforma móvil *Instabot SRAT-2* debido a que presenta una menor desviación estándar del error, un menor gasto energético y una mayor región de convergencia en comparación a los controladores lineales PID y LQR.

7.2. Trabajos futuros

Con el desarrollo del presente trabajo de grado es posible plantear nuevos trabajos de investigación tales como:

- Incorporar al robot móvil pendular *InstaBot SRAT-2* un sistema de visión artificial para evadir obstáculos.
- Diseñar un controlador que proporcione una buena estabilidad al robot en superficies o terrenos con diferentes grados de inclinación y tipo de rugosidades.
- Implementar nuevas técnicas de control lineal y no lineal tales como redes neuronales o redes de adelanto-atraso, así como controladores híbridos que permitan mejorar aún más el desempeño de este tipo de plataformas móviles.
- Diseñar una fuente de alimentación externa que proporcione la corriente necesaria para el funcionamiento de los motores y alimentación de las tarjetas que están incorporadas en el *InstaBot SRAT-2*.
- Implementar un control de velocidad diferencial para permitir la rotación del robot y su desplazamiento.
- Diseñar una plataforma móvil para el control remoto (cambio de posición y velocidad) del robot y visualización en línea de los datos de la curva de respuesta del mismo.
- Realizar el modelado dinámico del robot móvil pendular con otros métodos y ecuaciones matemáticas para determinar la mejor aproximación matemática para estas plataformas móviles.

Bibliografía

- [1] B. Mahler and J. Haase, “Mathematical model and control strategy of a two-wheeled self-balancing robot,” in *39th Annual Conference of the IEEE Industrial Electronics Society, (IECON)*, Nov 2013, pp. 4198–4203.
- [2] J. Akeson, A. Blomdell, and R. Braun, “Design and control of yaip- an inverted pendulum on two wheels robot,” in *Proceedings of the 2006 IEEE International Conference on Control Applications*, Munich, Oct 2006.
- [3] C.-H. Chiu, W.-R. Tsai, M.-H. Chou, and Y.-F. Peng, “Two - wheeled robot control based on self-tuning output recurrent CMAC,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, Hong Kong, March 2009.
- [4] R. Brisilla and V. Sankaranarayanan, “Nonlinear control of mobile inverted pendulum,” *Robotics and Autonomous Systems*, vol. 70, pp. 145 – 155, 2015.
- [5] A. A. Bature, S. Buyamin, M. N. Ahmad, and M. Muhammad, “A comparison of controllers for ballancing two wheeled inverted pendulum robot,” *International Journal of Mechanical & Mechatronic Engineering IJMME - IJENS*, 2014.
- [6] G. V. Raffo, M. G. Ortega, V. Madero, and F. R. Rubio, “Two-wheeled self-balanced pendulum workspace improvement via underactuated robust nonlinear control,” *Control Engineering Practice*, vol. 44, pp. 231 – 242, 2015.
- [7] C. Xu, M. Li, and F. Pan, “The system design and lqr control of a two-wheels self-balancing mobile robot,” in *Electrical and Control Engineering (ICECE), 2011 International Conference on*, Sept 2011, pp. 2786–2789.

- [8] O. Jamil, M. Jamil, Y. Ayaz, and K. Ahmad, "Modeling, control of a two-wheeled self-balancing robot," in *Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on*, April 2014, pp. 191–199.
- [9] F. Sun, Z. Yu, and H. Yang, "A design for two-wheeled self-balancing robot based on kalman filter and LQR," in *International Conference on Mechatronics and Control (ICMC)*, Jinzhou., Jul. 2014.
- [10] L. Sun and J. Gan, "Researching of two-wheeled self-balancing robot base on LQR combined with PID," in *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*, May 2010, pp. 1–5.
- [11] C.-H. Chiu, Y.-F. Peng, and Y.-W. Lin, "Intelligent backstepping control for wheeled inverted pendulum," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3364 – 3371, 2011.
- [12] C.-H. Chiu, Y.-W. Lin, and C.-H. Lin, "Real-time control of a wheeled inverted pendulum based on an intelligent model free controller," *Mechatronics*, vol. 21, no. 3, pp. 523 – 533, 2011.
- [13] W.-J. Wang and C.-H. Huang, "Model-based fuzzy control application to a self-balancing two-wheeled inverted pendulum," in *IEEE Control Applications, (CCA) and Intelligent Control, (ISIC)*, July 2009, pp. 1158–1163.
- [14] Y.-H. Wen, Y.-S. Lin, and Y.-G. Leu, "Design and implementation of the balance of two-wheeled robots," in *Advanced Robotics and Intelligent Systems (ARIS), 2013 International Conference on*, May 2013, pp. 130–134.
- [15] J. Wu and W. Zhang, "Design of fuzzy logic controller for two-wheeled self-balancing robot," in *Strategic Technology (IFOST), 2011 6th International Forum on*, vol. 2, Aug 2011, pp. 1266–1270.
- [16] K. Peng, X. Ruan, and G. Zuo, "Dynamic model and balancing control for two-wheeled self-balancing mobile robot on the slopes," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Beijing., Jul. 2012.

- [17] N. Nakpong and S. Yamamoto, “Just-in-time predictive control for a two-wheeled robot,” in *ICT and Knowledge Engineering (ICT Knowledge Engineering), 2012 10th International Conference on*, Nov 2012, pp. 95–98.
- [18] M. Azimi and H. Koofgar, “Model predictive control for a two wheeled self balancing robot,” in *Proceeding of the International Conference on Robotics and Mechatronics (RSIIISM)*, Tehran, Feb. 2013.
- [19] T.-J. Ren, T.-C. Chen, and C.-J. Chen, “Motion control for a two-wheeled vehicle using a self-tuning PID controller,” *Control Engineering Practice*, vol. 16, no. 3, pp. 365 – 375, 2008.
- [20] D. Pazderski and K. Kozlowski, “Practical stabilization of two - wheel mobile robot with velocity limitations using time-varying control law,” in *Proceedings of the Fifth International Workshop on Robot Motion and Control (RoMoCo)*, Jun. 2005, pp. 205 – 212.
- [21] H. Baruh, *Analytical Dynamics*, I. Editions, Ed. McGraw-Hill, 1999.
- [22] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 12th ed., P. P. Hall, Ed. Prentice Hall, 2011.
- [23] K. J. Astrom and T. Hagglund, *Control PID avanzado*. Pearson Prentice Hall, 2009.
- [24] F. Haugen, “Comparing PI tuning methods in a real benchmark temperature control system,” *Modeling, Identification and Control*, vol. 31, no. 3, pp. 79–91, 2010.
- [25] G. Welch and G. Bishop, “An introduction to the kalman filter,” Chapel Hill, NC, USA, Tech. Rep., 2001.
- [26] R. Yager and L. Zadeh, *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 1992.