



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO
CAMPUS INSTITUTO TECNOLÓGICO DE CULIACÁN



**TÉCNICAS DE DEEP LEARNING PARA EL CONTROL DE UN
MECANISMO IMPLEMENTANDO VISIÓN ARTIFICIAL**

TESIS

PRESENTADA ANTE EL DEPARTAMENTO ACADÉMICO DE ESTUDIOS DE POSGRADO
DEL INSTITUTO TECNOLÓGICO DE CULIACÁN EN CUMPLIMIENTO PARCIAL DE LOS
REQUISITOS PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

POR:

**ING. DAVID ALEJANDRO TEJEDA OCHOA
INGENIERO EN MECATRÓNICA**

DIRECTORA DE TESIS:

M.C. GLORIA EKATERINE PERALTA PEÑUÑURI

CO-DIRECTOR DE TESIS:

DR. ABRAHAM EFRAIM RODRIGUEZ MATA

CULIACÁN, SINALOA

23 de agosto del 2022



Instituto Tecnológico de Culiacán

Culiacán, Sin., 23 de Agosto del 2022

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
OFICIO: DEPI/264/08/2022

ASUNTO: **Autorización Impresión**

DAVID ALEJANDRO TEJEDA OCHOA
ESTUDIANTE DE LA MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN
PRESENTE.

Por medio de la presente y en virtud de que ha completado los requisitos para el examen de grado de **Maestro en Ciencias de la Computación**, se concede autorización para la impresión de la tesis titulada: **“TÉCNICAS DE DEEP LEARNING PARA EL CONTROL DE UN MECANISMO IMPLEMENTANDO VISIÓN ARTIFICIAL”** bajo la dirección del(a) **M.C. GLORIA EKATERINE PERALTA PEÑUÑURI.**

Sin otro particular reciba un cordial saludo.

ATENTAMENTE
Excelencia en Educación Tecnológica®

M.C. MARÍA ARACELY MARTÍNEZ AMAYA
JEFE(A) DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



INSTITUTO TECNOLÓGICO DE CULIACÁN

DEPARTAMENTO DE DIVISIÓN DE
ESTUDIOS DE POSGRADO E INVESTIGACIÓN

C.c.p. archivo

MAMA/lucy *





"TÉCNICAS DE DEEP LEARNING PARA EL CONTROL DE UN MECANISMO IMPLEMENTANDO VISIÓN ARTIFICIAL"

Tesis presentada por el(a):

DAVID ALEJANDRO TEJEDA OCHOA

Aprobada en contenido y estilo por:

M.C. GLORIA EKATERINE PERALTA PEÑUÑURI
Director de Tesis

DR. ABRAHAM EFRAIM RODRÍGUEZ MATA
Secretario

DR. HÉCTOR RODRÍGUEZ RANGEL
Vocal -1

M.C. ROSALIO ZATARAIN CABADA
Vocal -2

M.C. María Aracely Martínez Amaya
Jefe(a) de la División de Estudios de
Posgrado e Investigación



Dedicatoria

Dedico este trabajo a mis padres Eva Ernestina Ochoa Garcia y Remigio Tejeda, mis mejores amigos, y quienes siempre me han apoyado y han sido mis soportes y pilares más importantes, me han sostenido en los momentos más difíciles de mi vida. Y, sobre todo, por ser mi más grande inspiración para lograr todas mis metas.

A mi hermano, que es mi inspiración y ejemplo a seguir, puesto que ha logrado todas las metas que se ha propuesto, me motiva a seguir preparándome y esforzándome en la vida.

A mis abuelas, tíos, primos y sobrinos, por estar siempre motivándome, ayudándome y procurando mi bienestar.

A mis amigos, quienes se han convertido en mi familia, cada una de esas personas que ponen su confianza en mí, y en los que puedo confiar, sobre todo a los que me han acompañado en mis mejores y peores momentos. Especialmente a Kareli, Laney, Lilieth, Rafael, Roberto y Ana, quienes me han motivado durante el transcurso de este proyecto.

A Dios, por ser quien no me suelta de su mano, por ser mi guía en la oscuridad, y en quien descanso mis problemas.

Y a mí, por no rendirme, por seguir adelante, y por siempre buscar superarme y llenar de orgullo a los míos, esto es por mi padre, por mi madre y por el niño que fui ayer.

Agradecimientos

Agradezco principalmente a **Mi familia** por su apoyo incondicional en cada nuevo paso de mi vida, también a ellos por guiarme en una vida de aprendizaje y superación, siendo mi máxima motivación para afrontar cada uno de mis desafíos. Desde mi niñez formándome y aconsejándome para convertirme en la persona que soy hoy, apoyándome durante mis estudios ante cualquier duda, y alentándome a seguir preparándome.

A mis **Amigos** por ser quienes me alentaban a seguir esforzándome y apoyándome en mis momentos más complicados durante estos dos años.

A mi escuela, el **Instituto Tecnológico de Culiacán** por ser la institución que me ha formado profesionalmente en mi licenciatura y en mi maestría.

Al **Consejo Nacional de Ciencia y Tecnología** por ser quienes me han financiado durante mi maestría

Al **Instituto Tecnológico de Chihuahua** por ser la institución que me recibió y facilitó sus instalaciones y equipo durante mi estancia en la ciudad de Chihuahua, Chihuahua.

A mis **compañeros** por el apoyo dentro y fuera del aula, convirtiéndose en mis amigos.

Al **Dr. Abraham Efraím Rodríguez Mata** por su apoyo antes y durante la maestría, por ser mi asesor y consejero, además por el trabajo que realizamos en conjunto durante mi estancia en el Instituto Tecnológico de Chihuahua.

A la **M.C. Gloria Ekaterine Peralta Peñuñuri** por asesorarme y aconsejarme, por estar siempre al pendiente de mi aprendizaje como coordinadora del posgrado.

A los **profesores** de la Maestría, por su incansable labor de compartir sus conocimientos con sus alumnos, formando a la próxima generación, con un especial agradecimiento al **Dr. Ricardo Quintero** por ser un profesor determinante durante mi paso por esta maestría, demostrando su pasión y vocación como profesor.

Declaración de Autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

David Alejandro Tejeda Ochoa. Culiacán, Sinaloa, México, 2022.

Resumen

Las inteligencia artificial se ha vuelto una de las áreas de conocimiento más importantes del último tiempo, hay empresas como Google, Microsoft o Amazon que ponen cada vez más atención a desarrollar e integrar a sus productos técnicas de esta disciplina. Desde el procesamiento del lenguaje para los asistentes virtuales, hasta la fotografía móvil, en donde los celulares ya han llegado a un punto que con el procesamiento de imagen realizado por la inteligencia artificial ha logrado acercarse al resultado de una cámara fotográfica profesional.

Integrar las tecnologías de esta área con otras áreas se ha convertido en un desafío para los equipos de desarrollo. Unas de las áreas que podría trabajar en conjunto con estas técnicas sería la del control e instrumentación. Por su naturaleza, estas áreas ya cuentan con un algoritmo programado para realizar las tareas, por lo que es posible, dentro de esa programación, integrar algoritmos de inteligencia artificial.

El proyecto aquí presentado hace una integración entre estas áreas de conocimiento, inteligencia artificial, control e instrumentación, adaptando técnicas de inteligencia artificial como lo es la visión artificial o el aprendizaje profundo, a sistemas de control e instrumentación convencionales. Este es capaz de hacer mediciones en tiempo real de la posición y velocidad angular de un mecanismo de péndulo invertido, por medio de un sensor fotográfico, supliendo a los sensores como acelerómetro y giroscopio. Además de implementar una red neuronal en un sistema de control por modos deslizantes, obteniendo un control más robusto.

Este sistema ha sido probado respecto a referencias ideales de posición, velocidad y señales de control, obteniendo una rapidez de muestreo del sistema de visión artificial de entre 1 a 9 milisegundos, y una adaptación a perturbaciones lo suficientemente buena, también con una buena exactitud, acercándose a los valores reales. Además generando una señal de control acercada a la ideal, adaptándose, también, a perturbaciones bruscas que sufrió el sistema durante las pruebas.

Palabras Clave

- Aprendizaje máquina
- Aprendizaje profundo
- Inteligencia artificial
- Mecanismo
- Pendulo invertido
- Sistemas de control
- Sliding Mode Control (SMC)
- Vision artificial

Índice general

Índice de figuras	XI
Índice de tablas	XIII
1. Introducción	1
1.1. Definición del problema	5
1.2. Hipótesis	6
1.3. Objetivo	7
1.4. Objetivos específicos	7
1.5. Justificación	7
1.6. Estructura de la tesis	9
2. Marco teórico	10
2.1. Mecánica	10
2.1.1. Péndulo simple	10
2.1.2. Péndulo invertido	12
2.2. Electrónica	13
2.2.1. Circuito eléctrico	14
2.2.2. Relevador de potencia	14
2.2.3. PWM	15
2.2.4. Puente H	16
2.2.5. NVIDIA Jetson Nano	17
2.3. Control	18
2.3.1. Diagrama de bloques	18
2.3.2. Control ON-OFF	20
2.3.3. Control PID	21
2.3.4. Control de modos deslizantes	24
2.3.5. Teoría de estabilidad de Lyapunov	25
2.4. Computación	25
2.4.1. Inteligencia artificial	26
2.4.1.1. Historia	27
2.4.2. Aprendizaje máquina	32
2.4.3. Aprendizaje profundo	34
2.4.3.1. Redes neuronales	35
2.4.3.2. Perceptrón	36

2.4.3.3.	Redes neuronales multicapa	37
2.4.3.4.	Entrenamiento	38
2.4.3.5.	Algoritmo de propagación hacia atrás (Backpropagation)	39
2.4.4.	Visión artificial	39
2.5.	Tecnologías utilizadas	39
2.5.1.	OpenCV	41
2.5.2.	MATLAB	41
2.5.3.	Python	42
3.	Estado del Arte	44
3.1.	Sistemas convencionales	44
3.2.	Sistemas con implementación de algoritmos de visión artificial	45
3.3.	Sistemas de control con implementación de inteligencia artificial	47
3.4.	Comparación del control PID, PD y SMC	49
4.	Metodología Propuesta	51
4.1.	Hardware	51
4.1.1.	Ensamblaje de componentes	53
4.2.	Desarrollo de subsistemas	54
4.2.1.	Sistema de visión artificial	54
4.2.2.	Sistema de control	54
4.3.	Análisis de requisitos	55
4.3.1.	Requisitos funcionales	55
4.3.2.	Requisitos no funcionales	56
4.3.3.	Actores	56
4.3.4.	Casos de uso	57
4.3.5.	Diagrama de contexto	58
4.3.6.	Arquetipos	59
4.3.7.	Arquitectura	60
4.3.8.	Desarrollo del sistema de visión artificial	61
4.3.8.1.	Detección del ángulo	62
4.3.8.2.	Detección de la velocidad	63
4.3.9.	Desarrollo del sistema de control	64
4.4.	Estancia de investigación	66
5.	Análisis de Resultados	67
5.1.	Resultados del sistema de detección de posición	67
5.1.1.	Sistema de visión artificial	67
5.2.	Resultados del sistema de control de posición	70
5.2.1.	Sistema de control de posición	70
5.3.	Análisis de resultados	71
5.3.1.	Pruebas de visión	72
5.3.2.	Pruebas de control	73

6. Conclusiones	74
6.1. Conclusiones	74
6.2. Aportaciones	75
6.3. Trabajo a futuro	76
Bibliografía	78

Índice de figuras

2.1. Péndulo Simple con una piedra y una cuerda	11
2.2. Dinámica de un péndulo Simple	11
2.3. Diagrama esquemático de un péndulo invertido	13
2.4. Circuito eléctrico sencillo para encender una bombilla	14
2.5. Relevador de potencia con sus terminales	15
2.6. Dispositivo que hace PWM	16
2.7. Gráfica demostrativa de PWM	16
2.8. Puente H y sus posibles estados	17
2.9. Fotografía de una placa NVIDIA Jetson Nano (Fuente: NVIDIA Corporation, https://developer.nvidia.com/embedded/jetson-nano-developer-kit)	18
2.10. Diagramas de bloques demostrativos de un sistema de lazo abierto y un sistema de la cerrado	20
2.11. Gráfica de funcionamiento de un sistema de control On-Off	21
2.12. Gráficas de un sistema de control P, PI y PID	23
2.13. Inteligencia Artificial	27
2.14. Aprendizaje Máquina	34
2.15. Relación entre inteligencia artificial, aprendizaje máquina y aprendizaje profundo	35
2.16. Aprendizaje Profundo	35
2.17. Topología de una red neuronal multicapa	36
2.18. Perceptrón de una sola capa	37
2.19. Neurona artificial del perceptrón multicapa	38
2.20. Patrón para detección en OpenCV	40
2.21. Logotipos de OpenCV (Fuente: OpenCV team)	41
2.22. Logotipos de MathWorks, Matlab y Simulink (Fuente: The MathWorks, Inc.)	42
2.23. Logotipo de Python (Fuente: Python Software Foundation)	42
3.1. Péndulo controlado por PID (Fuente: (Abeysekera & Wanniarachchi, 2018))	45
3.2. Dron estabilizado gracias a visión artificial (Fuente: (Krafes et al., 2020))	46
3.3. Imagen del péndulo con detección de ángulo por visión artificial (Fuente: (Garayalde & Negri, 2014))	47
3.4. Modelo esquemático y físico del helicóptero (Fuente: (Kouro & Musalem, 2002))	48
3.5. Representación esquemática del péndulo invertido con rueda de reacción (Fuente: (Hernández et al., 2013))	49

3.6. Diagrama esquemático de un péndulo invertido de carro	50
4.1. Representación visual de la conexión de los componentes	53
4.2. Actores del sistema	57
4.3. Diagrama de casos de uso	57
4.4. Diagrama de contexto	58
4.5. Diagrama de arquetipos	60
4.6. Arquitectura planteada	61
4.7. Algoritmo seguido por el sistema de visión artificial	62
4.8. Simulación de lo que muestra un fotograma	63
4.9. Diagrama de bloques del diferenciador	63
4.10. Algoritmo seguido por el sistema de control	65
4.11. Diagrama de bloques del sistema de control por modos deslizantes propuesto por Jinkun Liu y Xinhua Wang (fuente (Liu & Wang, 2012))	65
5.1. Señal simulada de entrada para los estimadores y su comparación de veloci- dades calculadas por cada uno de ellos	68
5.2. Señal de entrada para los estimadores y su comparación de velocidades cal- culadas por cada uno de ellos	69
5.3. Prueba de adaptación a perturbaciones del sistema de control	71

Índice de tablas

3.1. Resumen de las características de rendimiento de posición del carro del péndulo invertido entre SMC y PID (Fuente: (Nasir et al., 2010)	50
3.2. Resumen de las características de rendimiento de posición del ángulo del péndulo invertido entre SMC y PID (Fuente: (Nasir et al., 2010)	50

Capítulo 1

Introducción

Actualmente, uno de los tópicos que se encuentra en auge es el de la inteligencia artificial, si bien, empresas directamente relacionadas a la tecnología o a la programación hacen uso de ella cada vez con más frecuencia, hay sectores en donde aún no se implementa como se podría.

En ciertas áreas, sobre todo de procesos industriales, no aprecian que con las técnicas de inteligencia artificial tendrían una mejora proporcional al esfuerzo, o inversión. En otros casos es, simplemente, por el desconocimiento de las mismas. Si bien, hay casos en donde el uso de estas técnicas sería innecesario por tratarse de problemática que se pueden resolver con soluciones sencillas, hay áreas, en este caso la robótica, que encontraría soluciones bastante interesantes en estas herramientas.

La robótica hace uso de la teoría de control y la instrumentación para implementar sus sistemas, tal es el ejemplo de un brazo robótico, que, al simular ser el brazo de una persona normal, cuenta con articulaciones, y esas articulaciones son controladas, cada una, por actuadores, por ejemplo motores.

Continuando con el ejemplo del brazo robótico, hay casos en que se usan 3 motores para controlarlo, y depende sea el caso, las articulaciones deben tener una precisión en sus movimientos casi perfecta, por ejemplo en el robot para cirugías Da Vinci, el cual es controlado por un cirujano para realizar operaciones, por lo que, al tratarse de vidas humanas, la precisión que debe tener en sus movimientos debe ser milimétrica, aquí es donde entra la teoría de control, que como su nombre lo dice, se encarga de indicar a los actuadores cómo y cuándo moverse.

La instrumentación es la parte complementaria al control, a lo que se refiere es a la obtención de datos, ya sea de temperatura, posición, velocidad, o cualquier variable que pueda presentar un sistema. En el mismo caso del robot para cirugías Da Vinci, los sensores que deben usar deben de ser los más precisos para poder no cometer errores al momento de operar, y además de su calidad, también es importante su cantidad, al ser tantas maniobras por parte del cirujano, debe haber una gran cantidad de sensores que capten cada uno de sus micromovimientos (Orihuela, 2015).

Hay dos puntos claves que se tocarán en este proyecto, la detección de posición y determinación de velocidad, y el sistema de control, esto siendo implementado en un mecanismo, el cuál se refiere a un grupo de elementos que trabajan en conjunto para transformar una fuerza de entrada, en un movimiento de salida.

Hay mecanismos sencillos en la vida cotidiana, por ejemplo en una bicicleta, el mecanismo de pedales-ruedas, que cuenta con engranes de diferentes tamaños y una cadena que los ayuda a estar ensamblados y moverse, al momento de girar los pedales, el engrane de estos comienza a moverse, haciendo que la cadena se mueva, produciendo un giro en la rueda, y generando un movimiento. Si bien, físicamente parece algo simple y sin mucho trasfondo, lo que está pasando en ese momento es una transformación de energía a movimiento gracias a ese mecanismo.

Para el caso de este proyecto se utilizará un péndulo invertido, el cuál es un mecanismo muy utilizado para probar nuevas teorías de control, del cuál se conoce muy bien su dinámica y naturaleza, por lo que el sistema actual se adapta a este. Un péndulo simple consta de una masa suspendida por una cuerda, o poste, que tiene un movimiento oscilatorio, el cuál se genera por acción de la gravedad u otra fuerza que le sea aplicada, un péndulo tiene un comportamiento estable, ya que su posición, cuando no recibe fuerzas externas, es una total vertical.

Como se mencionó anteriormente, el mecanismo que se controlará en este sistema es una péndulo invertido, el cual también consta de una masa, sostenida por un poste, pero su finalidad es encontrar una vertical donde la masa sea lo que se encuentre en la parte superior, suspendida por el poste antes mencionado. Contrario a lo mencionado en el péndulo simple, un péndulo invertido tiene un comportamiento inestable, ya que la misma gravedad hace que

la masa tienda a caer, perdiendo la vertical deseada. Como se indicó anteriormente, el péndulo invertido es el sistema mayormente elegido por los investigadores en el área de control para comprobar las nuevas teorías en ese campo. (Grasser et al., 2002).

Para que este mecanismo se mueva necesita un actuador, este es el elemento que transforma la energía eléctrica en otra manifestación, por ejemplo una lámpara, que transforma la energía eléctrica en luz, o un motor, que transforma la energía eléctrica en energía mecánica o movimiento. En el caso del péndulo invertido necesita de un motor al que esté acoplado el poste del péndulo, y que de esta forma se mueva al punto de referencia. Este motor necesita de algo que se llama sistema de control, que le dice cuándo y cómo moverse. Un sistema de control, o controlador, es un procedimiento que intenta regular las variables de un sistema tomando como referencia un punto o estado deseado, donde la comparación de las variables reales y las deseadas, da como resultado el error del sistema, el cuál, siempre busca tender a cero.

Hay distintos sistemas de control, con lógicas diferentes, el más básico se conoce como control “On-Off”, el cuál, como su nombre lo dice, tiene solo dos estados, encendido y apagado, este sería, por ejemplo, en el caso del alumbrado público, que la lámpara se enciende cuando es de noche, o está oscuro, y se apaga de día, o cuando hay luz, de esta forma, la lámpara solo tiene dos estados, es un sistema muy simple y fácil de programar. Pero no todos los problemas se resuelven de esta forma, hay casos en donde se necesitan puntos medios, por ejemplo en la velocidad de un vehículo, o la intensidad luminosa de una bombilla, para eso se utilizan los distintos sistemas de control, cada escenario con el controlador que mejor cubra las necesidades.

Como se mencionó con el péndulo invertido, que es la plataforma para probar las nuevas teorías de control, hay un sistema de control con el cuál se comparan estas nuevas teorías, que es el control Proporcional-Integral-Derivativo (PID por sus siglas), siendo este el sistema de control más utilizado y mejor adaptado a las plataformas de programación de controladores. Lo que hace este sistema de control es tomar la posición, velocidad y aceleración de un sistema, e intentar controlarlos para acercarse lo más posible al punto de referencia deseado. Ese es un sistema que es usado en muchos dispositivos, ya que es fácil de implementar, y hasta cierto punto intuitivo.

Hay más sistemas de control, y que tienen ventajas con respecto al PID, uno de ellos es el control por modos deslizantes (SMC por sus siglas en inglés), el cuál es menos popular, pero con aplicaciones muy interesantes. Este controlador se basa en graficar la posición y la velocidad, haciendo una división a dicha gráfica con una línea, a la que se le llama superficie deslizante, y oscilando las polaridades que se envían al actuador sobre esa superficie, obteniendo una dinámica del sistema muy rápida respecto a otros controladores, el problema es lo complicado que es para los actuadores poderse adaptar a cambios tan bruscos de polaridad, este sistema se explica de forma más extensa más adelante.

Para determinar la posición del mecanismo se necesita la disciplina antes mencionada, instrumentación. Esta es un área de la ingeniería, que como se menciona con antelación, se dedica, principalmente, a los sensores en un sistema. Para el caso de un sistema de control convencional, ya sea un péndulo invertido, se usarían comúnmente sensores como acelerómetros y giroscopios, pero para el caso de este sistema se propone utilizar una cámara de video como sensor, y algoritmos de visión artificial para determinar posición, velocidad o cualquier parámetro que necesite el sistema de control. En este caso se usa una librería del lenguaje de programación Python, que es OpenCV, para encontrar los puntos claves del péndulo, como lo son la masa, el poste y el motor, y cálculos geométricos, determinar el ángulo al que se encuentra, esto sería la posición, y usando un algoritmo de teoría de control, como lo es un diferenciador por modos deslizantes, determinar la velocidad al que se mueve este mecanismo.

Si bien, este algoritmo podría ser innecesario ya que se podría solo determinar la velocidad con la ecuación física de distancia recorrida sobre tiempo, se necesita una mejor precisión de la medición de velocidad, para poder determinar el siguiente movimiento que hará el motor.

Por último se comparan los resultados obtenidos tanto del algoritmo de visión artificial de la posición y velocidad con el movimiento real que realiza el péndulo, y el sistema de control comparado con un controlador PID adaptado al péndulo invertido.

1.1. Definición del problema

Existen muchos sistemas de control, la mayoría basados en modelos matemáticos complejos. Si bien, existen sistemas de control como el “On-Off”, donde solo se tienen parámetros para el encendido y apagado, este no es aplicable para casos más específicos donde se necesite más que solo encender y apagar un actuador (Electrical4U, 2020). Hasta el momento, uno de los sistemas que se utiliza más habitualmente es el control “Proporcional-Integral-Derivativo”(PID) y sus variaciones, este es un sistema que es fácilmente adaptable a cualquier problemática que necesite un sistema de control. Esto debido a que es del sistema de control del que más se encuentra bibliografía, el libro “Ingeniería de control moderna”, escrito por Katsuhiko Ogata es uno de los más importantes en el área, y tiene un capítulo completo dedicado al controlador PID, además de haber una gran cantidad de libros dedicados específicamente para este sistema de control (Ogata & Yang, 2002).

El sistema de control PID tiene sus deficiencias, es lento para adaptarse a perturbaciones, y, más allá de ser lento, es posible que en sistemas de muchas perturbaciones, es posible que nunca encuentre la estabilidad, incumpliendo la razón por la que ha sido implementado, que es darle estabilidad al sistema. Si bien, los investigadores y expertos en el área han tenido acercamientos a sistemas de control PID que llegan a la estabilidad en muy poco tiempo, también hay otros sistemas que han sido implementados para poder tener sistemas estables y adaptables a las perturbaciones. Tal es el caso del control por modos deslizantes, un sistema que si bien, cuenta con bibliografía y literatura en general para saber como implementarlo, no ha sido tan usado debido a que no hay tantos expertos en el área, además de ser un sistema que es muy agresivo con los actuadores, acortando considerablemente su vida útil.

Hay investigadores que han tenido éxito adaptando el control por modos deslizantes a actuadores, tal es el caso de (Rodríguez-Mata et al., 2021), donde se ha adaptado dicho control de forma efectiva en la dinámica de vuelo de un vehículo aéreo no tripulado, consiguiendo una buena estabilidad y tiempo de respuesta, esto gracias a la misma naturaleza de este controlador, que llega rápidamente al punto deseado, adaptándose a perturbaciones y consiguiendo poder reemplazar a los más habitualmente utilizados como es el PID.

Para obtener la posición y velocidad de un mecanismo habitualmente se utilizan unos

dispositivos llamados sensores, por ejemplo un giroscopio o un acelerómetro, estos necesitan ser calibrados para poder usarse, y después de eso se tiene que dar un mantenimiento y nuevas calibraciones para seguir obteniendo lecturas precisas. A pesar de que esto pueda ser tedioso, estos sensores manejan una precisión muy buena, y una velocidad de obtención de valores casi instantánea. Si bien hay algoritmos de visión artificial que pueden sustituir estos sensores, no para todos los casos es necesario, probablemente por su costo de desarrollo o por la negativa a cambiar algo que ya funciona. A pesar de ello, hay casos en que la transición podría ser de gran ayuda.

En el caso de un brazo robótico, cada una de sus articulaciones necesita de un sensor que determine su posición y velocidad, y de un sistema de control que le indique cuando y hacia donde moverse, esto en desarrollo termina siendo complicado por la cantidad de sensores que se necesitan utilizar, son muchas señales que llegan a un controlador y que este a su vez envía igual número de señales para regular el movimiento de dichas articulaciones. Si bien, no es un problema que necesite una solución urgente, hay algoritmos de visión artificial que podrían ayudar a determinar la posición y velocidad del mecanismo, esto permitiría suplir todos los sensores por uno solo, que sería una cámara de video. Esto generaría un desarrollo más extenso en cuanto a los algoritmos de visión artificial, pero eliminando el mantenimiento y calibración que necesitan los sensores actuales.

Hay acercamientos a los dos puntos antes mencionados, hay casos donde se utiliza un punto de referencia visual, que intenta suplir sensores por algoritmos de visión artificial. Y casos donde se intentan implementar sistemas de control convencionales con técnicas de visión artificial, teniendo, en algunos casos, buenos resultados.

1.2. Hipótesis

Las técnicas de aprendizaje profundo, en conjunto con la visión artificial, permitirán que un sistema de control de posición de péndulo invertido sea más preciso que un sistema convencional de control, ya sea PID (Proporcional, Integrativo y Derivativo) o un SMC (Control de Modo Deslizante). Además de mejorar el error de seguimiento de trayectoria con la red neuronal.

1.3. Objetivo

Diseñar e implementar un control de péndulo invertido utilizando aprendizaje profundo, con detección de ángulo mediante visión artificial, montado en una NVIDIA Jetson Nano.

1.4. Objetivos específicos

- Investigar sobre casos en donde se haya implementado inteligencia artificial en sistemas de control.
- Diseñar y construir un mecanismo de péndulo invertido que sea controlado por una NVIDIA Jetson Nano.
- Desarrollar un sistema de visión artificial para identificar la posición angular del péndulo.
- Desarrollar el algoritmo de un diferenciador de modos deslizantes que, a partir de la posición angular del péndulo, determine la velocidad del mismo.
- Implementar un sistema de lazo cerrado para controlar la posición del péndulo, usando el error de seguimiento que se obtenga con el módulo de visión.
- Comparar resultados de los sistemas de visión y de control propuestos respecto a las soluciones más convencionales.

1.5. Justificación

La integración de algoritmos de control convencional y de técnicas de inteligencia artificial para determinar posición y velocidad de un mecanismo y controlar al mismo, representa una opción para áreas como la robótica para desarrollar sus sistemas utilizando este mismo método híbrido, tanto para su control, como para conocer su posición, obteniendo, así, un sistema aún más robusto que lo que existe actualmente.

El uso de algoritmos de visión artificial en conjunto con algoritmos de control automático para suplir sensores, ofrece una poderosa herramienta para la detección de posición y la determinación de la velocidad y aceleración de un mecanismo. Esto ayuda, también, a simplificar sistemas de detección, debido a que, por ejemplo, en el caso de un brazo robótico, este necesita un sensor para cada articulación, y utilizando algoritmos de visión artificial, podría solo necesitarse una cámara de video de manera externa, y de manera interna un algoritmo que determine los parámetros antes mencionados. Haciendo que de manera física, se simplifiquen circuitos.

La combinación de una red neuronal con un sistema de control por modos deslizantes, genera un sistema de control más robusto, ya que un sistema de control por modos deslizantes por si solo ya es robusto y adaptable a perturbaciones, y añadiéndole una red neuronal se obtiene un sistema de control totalmente adaptado a las perturbaciones que se pudieran presentar, lo que corresponde a una gran opción si, por ejemplo, se buscan controlar los motores de las hélices de un dron, ya que es un vehículo que es susceptible a presentar desequilibrio por no poder adaptarse a rachas de viento, o el golpe de algún ave, o simplemente para mejorar su estabilidad, son una serie de variables que este sistema de control puede manejar. También en sistemas como la estabilización de un avión, en busca de la horizontal, y a pesar del viento y peso, este sistema podría asistir a los pilotos debido a la gran capacidad de adaptación a estos cambios bruscos en la variable del sistema. Otro punto a tener en cuenta es en que el uso de los sistemas de control por modos deslizantes suelen no utilizarse en gran medida por lo bruscos que son con los actuadores, en su intento por adaptarse a perturbaciones y tener un tiempo de llegada a meta tan rápido, genera que muchos actuadores presenten fallas al ser controlados por este tipo de sistemas, en esto es en lo que ayuda también la red neuronal, suavizando esa rápida oscilación que los actuadores no resisten, haciendo posible el poder utilizar casi cualquier tipo de actuador con este sistema sin el riesgo de reducir su vida útil.

1.6. Estructura de la tesis

El trabajo actual está organizado por 6 capítulos, donde el primer capítulo es el actual de Introducción.

- Capítulo 2 - Marco Teórico: El capítulo describe la perspectiva en la cual se basa este proyecto, en otras palabras, se explican los fundamentos en los que obtiene fundamento el trabajo, en tópicos de inteligencia artificial, aprendizaje máquina, visión artificial, dinámica, sistemas de control automático, de aquí se extrae la referencia sobre la cual se sustenta el desarrollo del sistema.
- Capítulo 3 - Estado del Arte: Este capítulo expone los trabajos que desarrollan, o han desarrollado, técnicas de inteligencia artificial enfocadas en control, o sistemas en donde la visión artificial aplicada a sistemas mecánicos juegue un rol importante, desplazando a los sensores convencionales.
- Capítulo 4 - Metodología: En este capítulo se describen los pasos que se siguieron para el desarrollo de este proyecto, además de describir el proceso que lleva a cabo el sistema, y de explicar el cómo se llegó a los resultados, tanto metodología, como herramientas utilizadas.
- Capítulo 5 - Análisis de Resultados: Este capítulo muestra los resultados obtenidos a partir de las pruebas realizadas, estos resultados se describen tanto con palabras, como con gráficos, mostrando las comparaciones respecto a sistemas convencionales de detección de posición y velocidad angular, como de sistemas de control.
- Capítulo 6 - Conclusiones: Finalmente, en este capítulo se exponen las conclusiones obtenidas a partir de los resultados y las comparaciones realizadas, determinando si la hipótesis propuesta se cumple, y en qué porcentaje es válida.

Capítulo 2

Marco teórico

Al ser este un proyecto multidisciplinario, hay una parte referente a la mecánica del sistema, otra a la electrónica aplicada en el mismo y al final la parte de computación. A continuación se abordarán cada una de estas disciplinas y como se asocian en este proyecto.

2.1. Mecánica

El concepto de mecánica hace referencia al estudio del movimiento; lo cuál se refiere al cambio de la posición de una partícula o un sistema de partículas respecto al tiempo, dicho sistema de partículas a la par de este movimiento vendrían a ser un sistema mecánico. Un sistema mecánico es un conjunto de elementos que transmiten o transforman energía, como se menciona en (Cosenza, 2015). Para esta investigación hay un sistema mecánico que será el encargado de llevar a cabo la parte dinámica del sistema, el cual lleva por nombre “péndulo invertido”, a continuación se hablará un poco de su origen, composición y aplicación en el sistema desarrollado.

2.1.1. Péndulo simple

Antes de hablar del péndulo invertido es importante aclarar qué es un péndulo simple, o, solamente, péndulo. Un péndulo simple es un sistema mecánico el cual cuenta con una masa suspendida por una cuerda o cualquier otro elemento que le permita hacer movimientos oscilatorios mediante fuerzas aplicadas, un ejemplo sencillo de un péndulo simple es si se toma una piedra, y se cuelga amarrada de un cordón, como se muestra en la figura 2.1, aplicando

una fuerza este péndulo comenzará a oscilar, tendrá un movimiento de vaivén. Para el caso en que alguna fuerza le sea aplicada al péndulo, este se moverá, pero poco a poco va a tender a quedarse quieto de forma vertical por la constante acción de la gravedad actuando sobre él, lo que hace a este un sistema dinámico estable, porque siempre va a tender a volver al origen, como se explica en (Khalil, 2002). En la figura 2.2 se muestra un péndulo simple, y se describe como es dicho movimiento oscilatorio. En (Hewitt, 2007) se dice que durante muchos años se tomó al péndulo invertido para controlar el movimiento de los relojes.



Figura 2.1: Péndulo Simple con una piedra y una cuerda

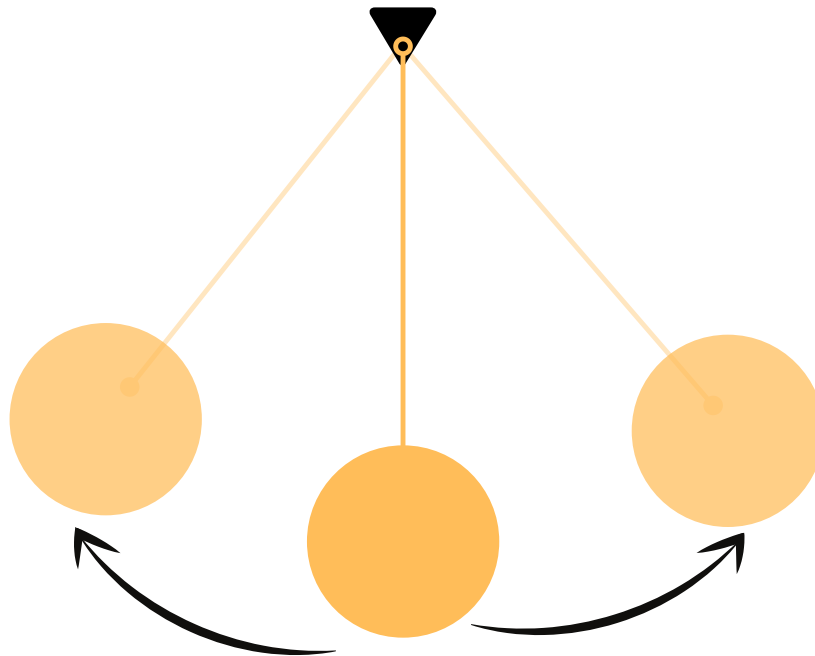


Figura 2.2: Dinámica de un péndulo Simple

2.1.2. Péndulo invertido

En (Tilbury et al., 2017) se explica ampliamente el sistema de péndulo invertido, el cuál consta de una masa sostenida por un poste y, a su vez, éste acoplado a un motor, a un sistema de transmisión de movimiento (caja de engranajes, polea, etc.), a un motor, o a una especie de carro pequeño. El movimiento de estos hace que el péndulo se estabilice en un punto de referencia, comúnmente en una vertical, formando un ángulo de 90° respecto al suelo. En (Khalil, 2002) se ejemplifica su diferencia respecto al péndulo simple, la cual es que éste tiene una dinámica inestable, lo que significa que tiende a alejarse del punto de referencia, en este caso la vertical. Esto significa que es un sistema no lineal, lo cual ofrece una complejidad que ha resultado interesante para los investigadores para experimentar, evaluar y comparar técnicas de control moderna. Todo esto ha desencadenado que el sistema de péndulo invertido aparezca en artículos de investigación y libros de texto sobre sistemas de control. Un ejemplo del mundo real que se relaciona directamente con este sistema de péndulo invertido es el control de la actitud de un cohete impulsor en el despegue, como se menciona en (Tilbury et al., 2017) y (Ooi, 2003).

Uno de los péndulos invertidos más usados en la investigación es en el que se tiene un carro acoplado al poste, al tratarse de un sistema que sin control es muy inestable, el carro necesita estarse moviendo para equilibrar el péndulo, en la figura 2.3 se muestra este tipo de péndulo invertido representado de manera esquemática. θ corresponde al error o ángulo que se forma entre la línea imaginaria vertical, o el punto de referencia de los 90° , y la inclinación real del poste, m corresponde a la masa, y mg corresponde al peso de la masa, P es el punto de acoplamiento entre el poste y el carro, l es la longitud del poste, M es la masa del carro, x es la horizontal sobre la que se mueve el carro, y u corresponden a la acción de control que realizaremos sobre el carro.

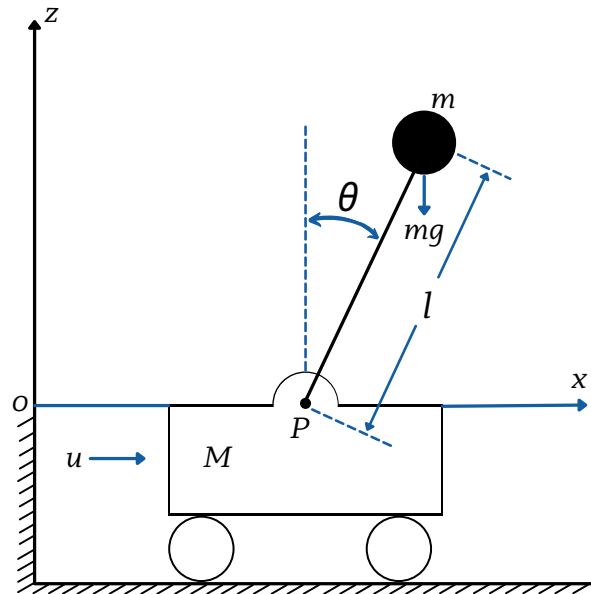


Figura 2.3: Diagrama esquemático de un péndulo invertido

Hay variantes del péndulo invertido, desde su sistema de movimiento, dimensiones y dificultades, pero todos apuntan a ser un sistema inestable, de alto orden, multivariable y fuertemente acoplado. Fue predicho por primera vez por Stephensen, hace aproximadamente 100 años, y en (Acheson, 1993) se amplió la teoría al sistema de péndulo rígido de enlace múltiple, tal como se dice en (Tang & Ren, 2009). En el caso de este proyecto, el sistema de péndulo invertido irá acoplado a una caja de engranes, y esta, a su vez, a un motor de corriente directa a 5 volts.

2.2. Electrónica

Dentro de este sistema se necesita la electrónica para accionar el motor que equilibrará el péndulo, además de controlar el sentido de giro, también se explicará la tarjeta electrónica que controlará todo el sistema.

La electrónica es una rama de la física que se encarga de la emisión, el comportamiento y los efectos de los electrones y de los dispositivos electrónicos, tal como lo se dice en (Scace, 2020). En este proyecto se implementó un circuito electrónico para controlar el giro de un motor, a continuación se explicará a profundidad cada uno de los componentes utilizados y

su funcionamiento.

2.2.1. Circuito eléctrico

Para el encendido y control de giro del motor se necesita un circuito eléctrico, en (Orrego, 2007) se hace referencia a que este es un arreglo cerrado de elementos y dispositivos eléctricos por los cuales fluye una corriente y actúa una fuerza llamada voltaje. Circuito viene de círculo, esto debido a que el inicio está conectado con el final, a esto se refiere con que es un arreglo cerrado. Para que un circuito funcione debe existir un dispositivo que de energía, a este se le conoce como “fuente”, también debe existir un elemento que transforme la energía eléctrica al cual se le conoce como “actuador”, esto puede ser en luz, tal como lo hace una bombilla, o en movimiento, como lo hace un motor, también están los cables o líneas de transmisión, y puede existir un dispositivo de control, como un apagador simple, o dispositivos más complejos como un dimmer, o regulador de voltaje, el cuál puede, por ejemplo, variar la intensidad de luz que emite una bombilla (Britannica, 2018a). La figura 2.4 muestra un circuito eléctrico muy simple que consta de una batería (fuente), una bombilla (actuador), y un interruptor. La función de este circuito es encender la bombilla.

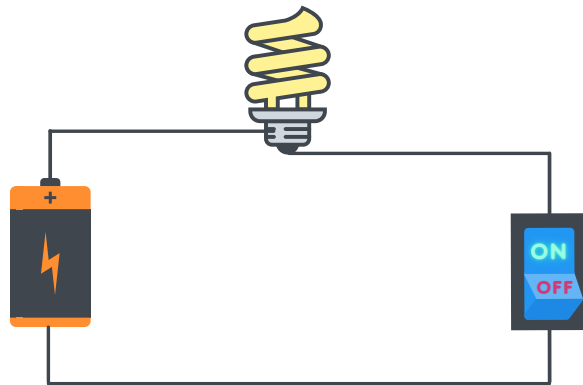


Figura 2.4: Circuito eléctrico sencillo para encender una bombilla

2.2.2. Relevador de potencia

Un relevador, o también llamado “relé”, es un dispositivo electromagnético utilizado para controlar de forma remota o automática la corriente de un circuito, como se menciona en

(Britannica, 2018b), un relé consta de dos circuitos acoplados magnéticamente, uno consta de un común, y dos contactos, uno abierto y otro cerrado, y el otro es un circuito de activación, que consta de una bobina, que cuando se energiza, el contacto abierto del primer circuito se cierra, y el abierto se abre, estos se usan comúnmente en sistemas con voltajes diferentes, y en los últimos años para el internet de las cosas, para circuitos de voltajes limitados, se puedan encender artefactos eléctricos que funcionan a voltajes elevados, por ejemplo encender un aire acondicionado. En la figura 2.5 se muestra el circuito interno de un relevador, además de sus terminales, donde *NA* se refiere a un contacto normalmente abierto, *NC* un contacto normalmente cerrado, *Comun* es el común, el *Activador* es donde entra la señal para energizar la bobina, y *Tierra* es la señal que va a negativo.

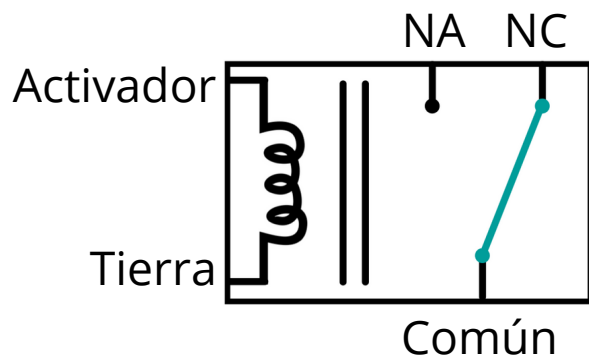


Figura 2.5: Relevador de potencia con sus terminales

2.2.3. PWM

En (Christ & Wernli, 2014) se describe a la modulación por ancho de pulso, o PWM, como una técnica de modulación, la cuál genera pulsos de voltaje con un ancho variable para representar la amplitud de una señal de entrada analógica. Esto para variar la velocidad de un motor, o la intensidad luminosa de una bombilla, ya que variando el voltaje no se genera esto. Hay dispositivos electrónicos que ya generan esta señal, que utilizan un potenciómetro para hacer esta variación del ancho de pulso, en este proyecto se utilizó uno de estos circuitos, en la figura 2.6 se ve el dispositivo que controla la señal con PWM. En la figura 2.7 se muestra una señal de pulsos usando PWM, donde t representa el tiempo.

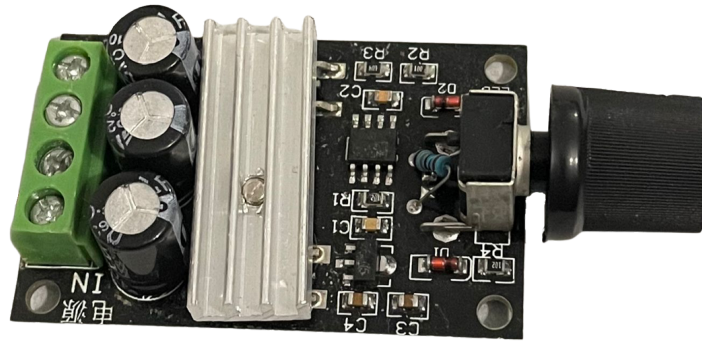


Figura 2.6: Dispositivo que hace PWM

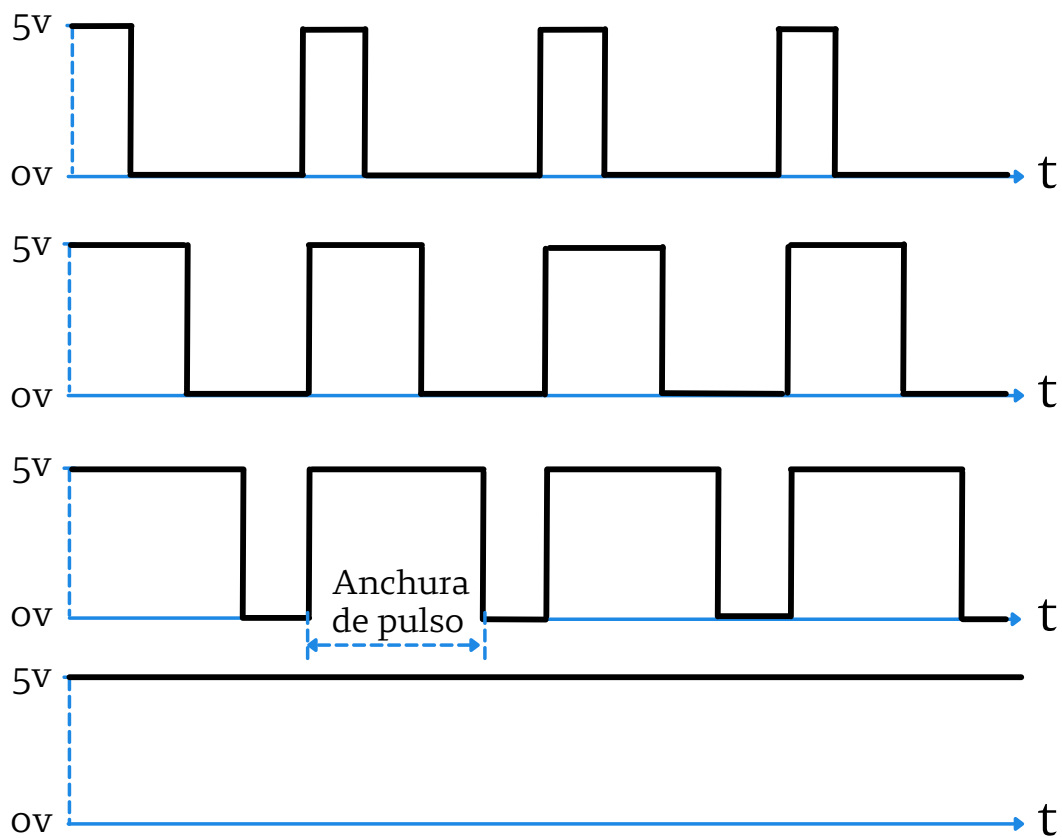


Figura 2.7: Gráfica demostrativa de PWM

2.2.4. Puente H

El control de un motor puede ser por encendido y apagado, regular la velocidad de giro, y también se puede controlar el sentido de giro, ahí es donde entra un circuito como el puente

H. Lo que hace es proveer de dos caminos a la corriente, y depende sea el caso, va a tomar uno de esos caminos, y depende cuál sea el camino que tome la corriente, será el sentido de giro del motor, así como detalla (Christ & Wernli, 2014). En la figura 2.8 se muestra el circuito de puente H, y se aprecia el por qué recibe ese nombre, ya que el arreglo de elementos forma una especie de H, también se muestran cada uno de sus estados.

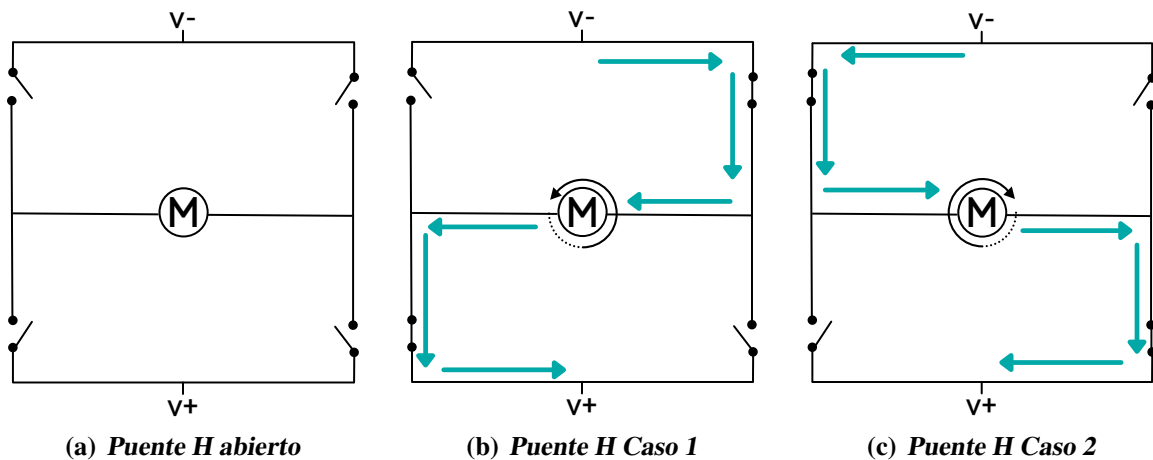


Figura 2.8: Puente H y sus posibles estados

2.2.5. NVIDIA Jetson Nano

Según (NVIDIA, 2021), la NVIDIA Jetson Nano una computadora de tamaño pequeño, que permite ejecutar diferentes tipos de algoritmos de inteligencia artificial, incluso entrar en redes neuronales para aplicaciones como la clasificación de imágenes, la detección de objetos, la segmentación, etc. Todo esto en una plataforma que trabaja a 5 volt. En este caso se implementarán dos algoritmos, uno de detección de inclinación mediante inteligencia artificial, y uno de un sistema de control que implementa redes neuronales. La figura 2.9 muestra una fotografía de la NVIDIA Jetson Nano, que aparece en la página oficial de NVIDIA.



Figura 2.9: Fotografía de una placa NVIDIA Jetson Nano (Fuente: NVIDIA Corporation, <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>)

2.3. Control

Para enviar las señales para encender el motor e indicar el sentido del giro es necesario un sistema, el cuál se le llama “control”. Como se menciona en (Britannica, 2013), el control es el medio por el que una variable o conjunto de ellas se ajustan a una indicación prescrita. Hace, o trata hacer, que dichas variables se mantengan constantes o que tengan variaciones dependiendo los valores de referencia dados. Existen sistemas de control que se basan totalmente en la electricidad, o con medios mecánicos, con la presión de algún gas o fluido, o con una combinación de estos. En el caso de que el núcleo del control, o controlador, sea una computadora, es pertinente que todos los dispositivos a accionar sean eléctricos, aunque también existe el caso de que sean mecánicos. Para el caso de el péndulo invertido, el controlador será la antes mencionada NVIDIA Jetson Nano, y el control se hará sobre un motor de corriente directa.

2.3.1. Diagrama de bloques

Antes de aplicar, o programar, cualquier sistema de control, se necesita hacer una representación gráfica del mismo, para eso existen los diagramas de bloques, que son, como ya se mencionó, una representación visual de las funciones que realiza cada uno de los componentes de un sistema de control, además de las interrelaciones de los mismos y del flujo

que llevan las señales. En (Ogata & Yang, 2002) también se menciona que dicho diagrama muestra, a diferencia de una representación matemática puramente abstracta, una forma más apegada a la realidad los flujos de las señales del sistema real. En los diagramas de bloques todas las variables del sistema están, de alguna forma, vinculadas entre sí, por medio de los bloques funcionales. Dichos bloques son un símbolo de la operación matemática que se aplica sobre la señal de entrada al bloque, y lo que genera esta operación es la salida. Las funciones de transferencias en cada componente se introduce en los bloques correspondientes, estos se unen por medio de flechas, las cuales indican la dirección del flujo de estas señales. La señal solo va en la dirección indicada, esto ayuda para el mejor entendimiento del sistema.

(Ogata & Yang, 2002) también menciona que hay dos tipos de sistemas de control, y que con los diagramas de bloques pueden ser fácilmente comprendidos, uno de lazo abierto y otro de lazo cerrado. El sistema de control de lazo abierto tiene solo una entrada, la cual llega al controlador y este, a su vez, envía la señal de control al proceso, y este, por último, tiene una salida, los sistemas de lazo abierto son sistemas de control finito. Para el caso de los sistemas de lazo cerrado, como su nombre lo indica, hay una línea que regresa como si fuera una nueva entrada, esta representa el estado actual del sistema, y se le llama “realimentación”, y la resta de la referencia, respecto a esta realimentación, nos da una señal de error, que es lo que el controlador intentará regular. Este es un ciclo infinito, por si mismo el sistema se estará regulando. En la figura 2.10 se muestra un diagrama de bloques de un sistema de lazo abierto y uno de lazo cerrado. Para el caso de nuestro sistema, el diagrama de bloques que explica el funcionamiento del mismo será uno de lazo cerrado, debido a que se necesita una constante regulación de la posición del péndulo.

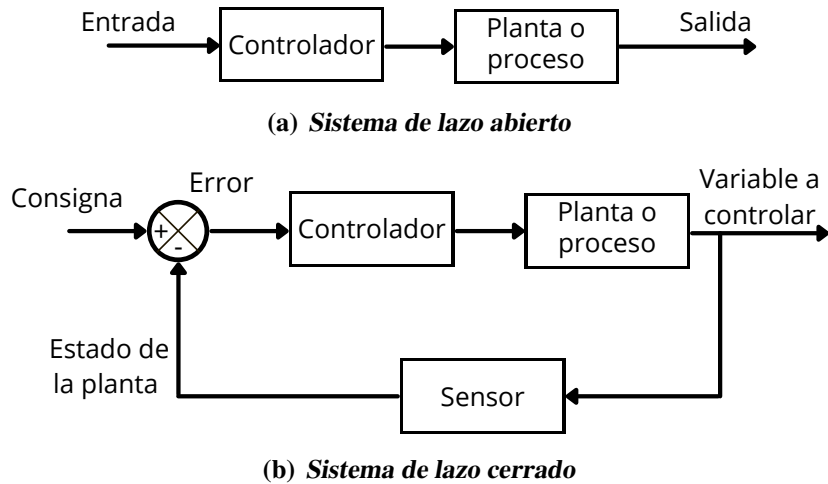


Figura 2.10: Diagramas de bloques demostrativos de un sistema de lazo abierto y un sistema de la cerrado

2.3.2. Control ON-OFF

A continuación se hablará acerca de distintos sistemas de control habitualmente utilizados, desde el más sencillo, hasta uno más complejo, que será el utilizado en este proyecto. El primero de ellos es el control On-Off, o control de dos posiciones, este es un sistema que solamente tiene dos estados, encendido y apagado. (Ogata & Yang, 2002) menciona que estos sistemas de control son muy económicos, por ello son muy utilizados de forma industrial como doméstica. Se usan, por ejemplo, para los sistemas de tinaco, para la iluminación, inclusive los aires acondicionados usan estos sistemas para el encendido del compresor, entre otras muchas aplicaciones. En la figura 2.11 se muestra un control On-Off, y sus variaciones entre *todo* y *nada*, sin puntos medios.

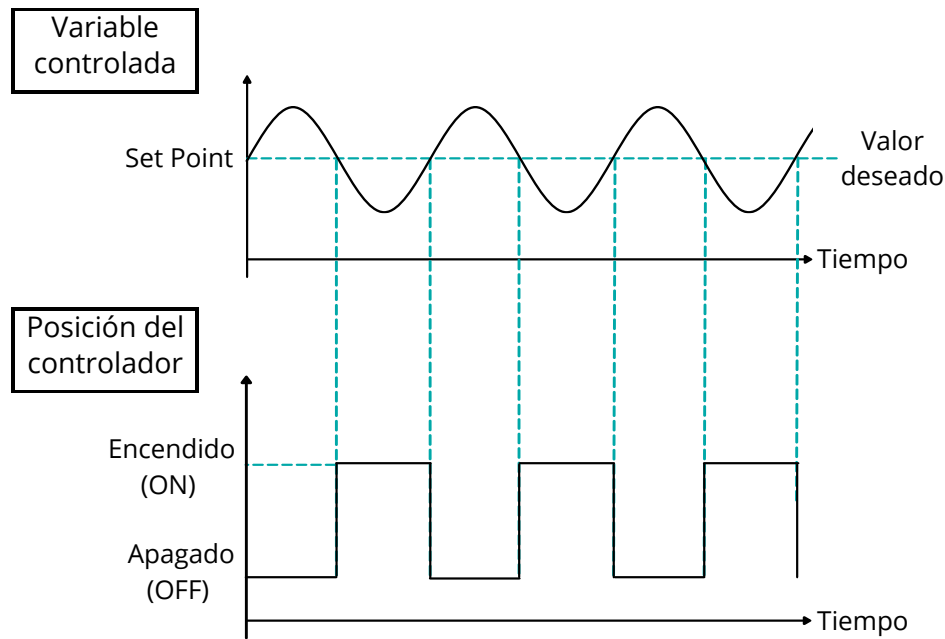
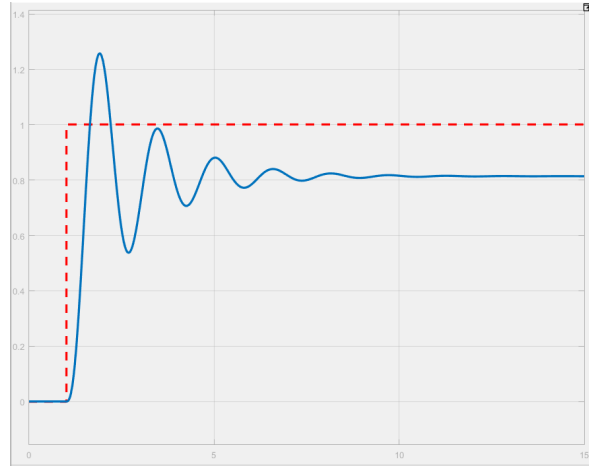


Figura 2.11: Gráfica de funcionamiento de un sistema de control On-Off

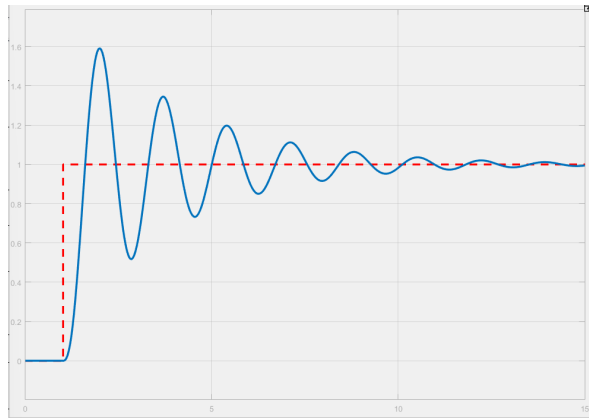
2.3.3. Control PID

A continuación se hablará de uno de los sistemas de control más utilizados, el PID (Proporcional-Integral-Derivativo). El control PID se compone de tres procesos diferentes, la acción de control **Proporcional**, que es una relación proporcional entre la salida del controlador, y el error, como se menciona en (Ogata & Yang, 2002), donde hay una constante K_p , la cual el propio usuario define mediante cálculos. La acción de control **Integral**, la cual tiene una relación proporcional con la integral definida en el tiempo, donde hay una constante K_i , la cual el propio usuario define como en el caso proporcional. Y, finalmente, la acción de control **Derivativa**, que es una relación proporcional con la derivada del error respecto al tiempo, donde hay dos constantes K_p y T_d , donde T_d es el tiempo derivativo. Es importante mencionar que hay un control meramente proporcional y uno meramente integral, pero no existe uno meramente derivativo, solo hay uno en que se mezclan la acción de control proporcional con la acción de control derivativa, por eso mismo una de las constantes de la acción de control derivativa es K_p , por que es la misma constante que en la parte proporcional. La combinación de la acción de control proporcional, la acción de control integral y la acción de

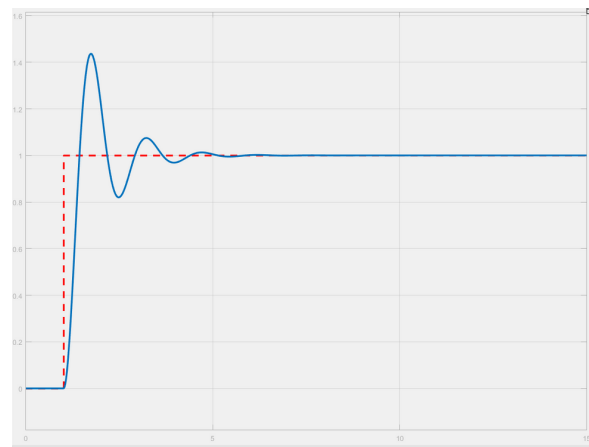
control derivativo se denomina acción de control proporcional más integral más derivativo. Tiene las ventajas de cada una de las tres acciones de control individuales. En la figura 2.12 se muestra un mismo problema, resuelto con tres sistemas de control antes mencionados, primero la acción de control meramente proporcional, después una mezcla del proporcional e integral, y, finalmente, un control PID para el mismo problema. Se puede observar que el control Proporcional no llega a estar tan cerca de la señal a seguir, en el caso del PI, se acerca a la señal pero se mantiene oscilando una gran parte del tiempo de la simulación, y, por último, un control PID, que de inicio tiene 2 oscilaciones, pero se estabiliza muy bien al final.



(a) *Control P*



(b) *Control PI*



(c) *Control PID*

Figura 2.12: Gráficas de un sistema de control P, PI y PID

2.3.4. Control de modos deslizantes

Si el control PID funciona bien, hay sistemas de control más robustos, entre ellos el Control de Modos Deslizantes, o SMC por sus siglas en inglés (Sliding Mode Control), el cuál se basa en la alteración de la dinámica de un sistema no lineal mediante el diseño de múltiples estructuras de control para garantizar que las trayectorias se muevan siempre hacia una condición de conmutación. Por lo tanto, la trayectoria final no existirá completamente dentro de una estructura de control. La ley de control de retroalimentación de estado no es una función continua del tiempo. En su lugar, cambia de una estructura continua a otra en función de la posición actual en el espacio de estado. Por lo tanto, el control en modo deslizante es un método de control de estructura variable. Las múltiples estructuras de control están diseñadas para garantizar que las trayectorias se muevan siempre hacia una condición de conmutación. Por lo tanto, la trayectoria final no existirá completamente dentro de una estructura de control. En cambio, la trayectoria final se deslizará a lo largo de los límites de las estructuras de control. El movimiento del sistema cuando se desliza a lo largo de estos límites se denomina modo de deslizamiento y el lugar geométrico formado por los límites se denomina superficie de deslizamiento (hiper). En el contexto de la teoría de control moderna, cualquier sistema de estructura variable, como un sistema bajo control de modo deslizante, puede verse como un caso especial de sistema dinámico híbrido. Intuitivamente, el control en modo deslizante utiliza una ganancia prácticamente infinita para forzar las trayectorias de un sistema dinámico a deslizarse a lo largo del subespacio restringido de modo deslizante. Las trayectorias de este modo deslizante de orden reducido tienen propiedades deseables (por ejemplo, el sistema se desliza de forma natural a lo largo de él hasta llegar a un equilibrio deseado).

El principal punto fuerte del control en modo deslizante es su robustez. Dado que el control puede ser tan simple como una conmutación entre dos estados, no necesita ser preciso y no será sensible a las variaciones de los parámetros que entran en el canal de control. Además, como la ley de control no es una función continua, el modo deslizante puede alcanzarse en un tiempo finito (es decir, mejor que el comportamiento asintótico). Hay dos pasos en el diseño del SMC.

El primer paso es el diseño de una superficie de deslizamiento para que la planta restringida a la superficie de deslizamiento tenga una respuesta deseada del sistema. Esto significa que las variables de estado de la dinámica de la planta están restringidas para satisfacer otro conjunto de ecuaciones que definen la llamada superficie de conmutación.

El segundo paso consiste en construir las ganancias de retroalimentación conmutadas necesarias para conducir la trayectoria de estado de la planta a la superficie de deslizamiento. Estas construcciones se basan en la teoría de estabilidad de Lyapunov generalizada (Wang, 2012, Edwards & Spurgeon, 1998).

2.3.5. Teoría de estabilidad de Lyapunov

La estabilidad de Lyapunov está relacionada con el comportamiento de las trayectorias de un sistema, cuando su estado inicial se encuentra cerca de un equilibrio. Desde el punto de vista práctico este punto es importante, ya que las perturbaciones que afectan a un sistema tienden a separarlo del equilibrio. Dentro del organigrama general de las diversas teorías de estabilidad, la de Lyapunov se engloba en aquella área que trata con las variables de estado del sistema, y por tanto, tiene que ver con la representación interna de los sistemas. En (Haddad & Bernstein, 1993, Slotine, Li et al., 1991, Vidyasagar, 1993) se encuentran estudios completos sobre esta teoría (García & Barreiro, 2015).

Fue Lyapunov quien introdujo la idea de una función energía ficticia, denominada función de Lyapunov, aplicable a cualquier ecuación diferencial o en diferencias y tanto a sistemas lineales como a no lineales. En función del signo de la función de Lyapunov y de su derivada con respecto al tiempo, se pueden obtener conclusiones sobre el tipo de estabilidad de un estado de equilibrio.

2.4. Computación

En (ACIMED, 1994) se hace referencia a varios conceptos relacionados a la informática y computación, definiendo esta como a la ciencia vinculada al proceso de información con instrumentos creados por la humanidad, y también como el conjunto de técnicas y disciplinas para el manejo automático de la información, por medio del uso de computadoras. Las

computadoras comenzaron dirigiéndose a un consumo principalmente de corporaciones, y tenían un aspecto distinto a como las conocemos ahora, pero relativamente pocos años han bastado para que sean una herramienta fundamental en cada casa, esto por sus múltiples usos para el aprendizaje, trabajo y ocio, siendo uno de los inventos más importantes de la historia, esto mencionado en (Coello, 2003). En la investigación actual, la computación tiene un papel fundamental, debido al uso tanto de visión artificial, como de redes neuronales, a continuación se abordaran estos temas.

2.4.1. Inteligencia artificial

Como se menciona en (Patterson & Gibson, 2017), definir lo que es la inteligencia artificial ha sido un reto para la humanidad, ya que es un termino compuesto de dos palabras, entonces para conceptualizarlo se definen cada una de las palabras que lo componen, y luego se infiere una definición a partir de ellas. Para comenzar, el concepto de inteligencia tiene diferentes vertientes, pero, a su vez, todas apuntan a la capacidad de aprender o comprender algún tipo de información, y a partir de ese conocimiento, manipular el propio entorno, tal como se menciona en (Merriam-Webster, s.f.). Y para el caso de artificial, hay distintas definiciones, una que se adapta bien para este caso sería que se refiere a algo hecho, producido o realizado para que parezca algo natural, como se menciona en (Merriam-Webster, s.f.). Entonces, para la definición de inteligencia artificial, nos referiríamos a la capacidad que tiene un elemento creado por el humano para poder aprender o comprender algún tipo de información, y poder realizar alguna tarea con dicha información. En (McCarthy et al., 2006) se menciona que el término *inteligencia artificial* fue presentado por primera vez en el seminario realizado en el Dartmouth College en 1955, y se definió como “el problema de hacer una máquina que se comporte en maneras que serían llamadas inteligentes si fueran realizadas por un ser humano”.

Dentro de la inteligencia artificial hay conocimiento aún más profundo, como lo es el aprendizaje máquina (machine learning) y el aprendizaje profundo (deep learning), de los cuales más adelante se hablará de forma más puntual. Este proyecto utiliza la inteligencia artificial de dos formas, con la visión artificial para la medición de un ángulo, función que normalmente realiza un giroscopio o acelerómetro, y también en el sistema de control, donde

utiliza una red neuronal como núcleo de este. En la figura 2.13 se muestra una especie de procesador donde de un lado se ve un código, y al otro un cerebro, intentando hacer entender lo que es la inteligencia artificial, una computadora que simula pensar mediante un código.

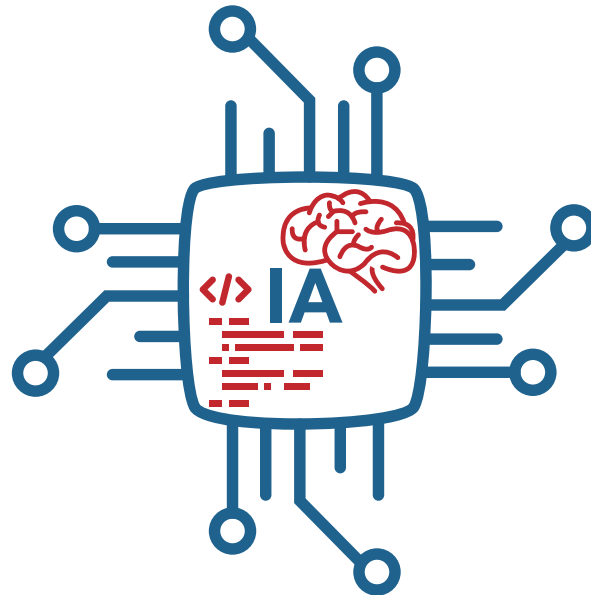


Figura 2.13: Inteligencia Artificial

2.4.1.1. Historia

Como ya se mencionó, el término de *inteligencia artificial* fue propuesto en 1955 en el seminario realizado en Dartmouth College, pero ha habido más eventos importantes en la historia de la inteligencia artificial, (Negnevitsky & Intelligence, 2005) distribuye los eventos importantes del surgimiento de la inteligencia artificial en 7 etapas, que, aunque no sean las únicas, han sido el parteaguas para que, hoy en día, se tenga a la inteligencia artificial tan presente en el día a día.

La primera etapa la llama *El nacimiento de la inteligencia artificial* (1943-1956), primero, en 1943, hubo un trabajo presentado por Warren McCulloch y su coautor Walter Pitts, este trabajo lleva por nombre *A Logical Calculus of the Ideas Immanent in Nervous Activity*, el cual propone un modelo de redes neuronales artificiales en el que se postulaba que cada neurona estaba en un estado binario; es decir, en condición de encendido o apagado. Más adelante, en 1946, John Presper Eckert y John William Mauchly, con la asesoría de John von

Neumann, desarrollaron el Computador e Integrador Numérico Electrónico (ENIAC) por sus siglas en inglés, el cual fue una de las primeras computadoras, sin esta computadora no se hubiesen podido desarrollar las computadoras que se usan hoy en día. En 1950, se escribió y presentó uno de los primeros y más importantes artículos para la inteligencia artificial, llamado “Computing machinery and intelligence”, desarrollado por el Matemático Británico Alan Turing, en 1950. En el mismo año, Claude Shannon, publicó un artículo acerca de las máquinas de ajedrez, donde mencionaba que una partida implicaba una cantidad de movimientos posibles de 10^{120} , y, inclusive considerando que la computadora de tipo von-Neumann tarde un microsegundo por jugada, para realizar su primera jugada esta tardaría alrededor de 3×10^{120} años, para lo que Shannon demostró que era necesario el uso de la heurística. Y finalmente, en 1956, John McCarthy en conjunto con Marvin Minsky y Claude Shannon, organizaron un congreso en el colegio Dartmouth, donde McCarthy trabajó con los graduados de Princeton que estaban interesados en el estudio de la inteligencia máquina, las redes neuronales artificiales y la teoría de autómatas. Los participantes de este congreso estuvieron dominando el área de la inteligencia artificial.

La segunda etapa fue llamada *El surgimiento de la inteligencia artificial* de 1956 a finales de los 60's, donde lo primero relevante que pasó fue la creación del lenguaje de programación LISP, desarrollado por John McCarthy, uno de los organizadores del congreso del colegio Dartmouth y quien acuñó el término de inteligencia artificial, dicho lenguaje es basado en listas y átomos y se convirtió en el estándar para aplicaciones de inteligencia artificial. En 1961, Allen Newell y Herbert Simon desarrollaron un programa de propósito general para simular métodos de resolución de problemas de los humanos, al cual llamaron *Solucionador de Problemas Generales* (GPS por sus siglas en inglés), el cuál fue uno de los proyectos más ambiciosos de la época, y que significó el primer intento de separar el resolver un problema con los datos. Esto lo estuvieron desarrollando desde los 60's hasta los 70's, y aunque fallaba para resolver problemas complejos, fue uno de los grandes avances en la historia de la inteligencia artificial, y una base para desarrollar tecnología más avanzada. Más adelante, en 1962, Frank Rosenblatt se basa en el trabajo que iniciaron McCulloch y Pitts en redes neuronales artificiales, y presenta el Teorema de Convergencia del Perceptrón, demostrando que su algoritmo de aprendizaje podía ajustar las fuerzas de conexión de un perceptrón (Rosenblatt,

1961). También en los 60's Lofti Zadeh publica su artículo llamado "Fuzzy Sets" (L. Zadeh, 1965), que es considerado como el fundador de la teoría de sistemas difusos.

La tercera etapa fue llamada *La desilusión en la inteligencia artificial* de finales de los 60's a principios de los 70's, a pesar de los avances que se habían dado, la mayoría de los fondos de diferentes gobiernos para el desarrollo de la Inteligencia Artificial habían sido cancelados, esto significó una gran dificultad para esta disciplina, además que la percepción era que se encontraba atascada, ya que los primeros programas contenían prácticamente nulo conocimiento sobre el dominio del problema, y para resolver los problemas aplicaban una estrategia de búsqueda probando diferentes combinaciones de pequeños pasos, hasta encontrar la correcta. Y este método solo funcionaba con problemas muy simples, pero para problemas grandes no se podía aplicar esta lógica. Aunado a esto, las proyecciones que se hacían acerca de la evolución de la inteligencia artificial terminaron siendo demasiado optimista, objetivos que no se podían lograr para ese momento y con las herramientas que contaban.

La cuarta etapa fue llamada *El descubrimiento de los sistemas expertos* de principios de los 70's a mediados de los 80's, esta fue una etapa de transición al renacimiento de la inteligencia artificial, aquí se desarrollaron sistemas expertos, un cambio de lógica para la inteligencia artificial, ya que anteriormente se habían desarrollado sistemas buscando inventarse algoritmos de búsqueda y técnicas de razonamiento inteligentes para emular los métodos de resolución de problemas generales, similares a los humanos. Al fracasar los métodos débiles, los investigadores acabaron por darse cuenta de que la única forma de obtener resultados prácticos era resolver casos típicos en áreas de conocimiento reducidas realizando grandes pasos de razonamiento. Lógica que da pie a los sistemas expertos. Uno de los organizadores del congreso del colegio Darmouth fue Marvin Minsky, quien desarrolló, en 1974 una perspectiva antilógica sobre la representación del conocimiento y el razonamiento. Su teoría de los marcos (Minsky, 1974), supuso una importante contribución a la ingeniería del conocimiento. Uno de los primeros sistemas expertos fue DENDRAL, fue el primer sistema basado en conocimiento que resultó exitoso, consistía en un programa para determinar la estructura molecular del suelo marciano, basándose en los datos espectrales de masas proporcionados por un espectrómetro de masas (Solving, s.f.). La NASA fue el principal financiador de dicho sistema. Posterior a DENDAL, Feigenbaum y otros investigadores de Stanford emprendieron

otro proyecto ahora en el área del diagnóstico médico, este fue llamado MYCIN, y posteriormente se convertiría en la tesis doctoral de Edward Shortliffe (Shortliffe, 2012). MYCIN era un sistema experto basado en reglas para el diagnóstico de enfermedades infecciosas de la sangre. También proporcionaba al médico consejos terapéuticos de una forma cómoda y fácil de usar. Otro sistema probabilístico desarrollado a inicios de los 80's PROSPECTOR, un sistema experto para la exploración de minerales (Duda et al., 1981). PROSPECTOR utilizaba una estructura combinada que incorporaba reglas y una red semántica. PROSPECTOR contaba con más de mil reglas para representar un amplio conocimiento del dominio. También contaba con un sofisticado paquete de apoyo que incluía un sistema de adquisición de conocimientos. Otro evento importante en esta etapa es el desarrollo de PROLOG, uno de los lenguajes de programación más utilizados en el desarrollo de Inteligencia Artificial, en este caso de sistemas expertos por su popularidad en esa época.

La quinta etapa fue llamada *El renacimiento de las redes neuronales artificiales* de mediados de los 80's en adelante, aquí hay un resurgimiento de las redes neuronales artificiales, John Hopfield presenta, en 1982, su modelo de redes neuronales con realimentación, llamadas Hopfield networks (Hopfield, 1982). En el mismo año, Teuvo Kohonen presenta su artículo *Self-organised maps*, encontró que para redes simples de elementos físicos adaptativos que reciben señales del espacio de eventos principal, la representación de la señal se asigna automáticamente a un conjunto de respuestas de salida de modo que la respuesta sigue el mismo orden topológico que el conjunto de respuestas. Grandes eventos. Eso quiere decir que se descubrió un principio que facilita la generación automática de gráficos topológicamente correctos de características observables de eventos (Kohonen, 1982). Pero el avance que más impacto tuvo fue el que presentaron Rumelhart y McClelland, quienes, en 1986, tomaron el trabajo anteriormente realizado por Bryson y Ho en 1969 acerca del algoritmo de propagación hacia atrás, y lo reinventaron en el artículo *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, el cual, como su nombre lo dice, se adentraban en las microestructuras para la capacidad de razonar, memorizar y resolver problemas (McClelland et al., 1987). También a finales de la década de los 80's se lleva a cabo la primera conferencia de redes neuronales, organizada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE por sus siglas en inglés). Entrando a los 90's Simon Haykin desarrolla

un artículo llamado *Neural Networks expand SP's horizons*, el cual tenía 3 objetivos; introducir nuevas ideas para métodos estadísticos de procesamiento de señales utilizando redes neuronales que tengan en cuenta el mundo real de la no linealidad, la no estacionariedad y la no gaussianidad; describe tres casos de estudio con datos reales, demostrando claramente que este nuevo enfoque trasciende los enfoques clásicos en el procesamiento estadístico de señales, y discute la información mutua como un criterio esencial para el diseño de una red neuronal no supervisada, que es diferente del criterio del error cuadrático medio (Haykin, 1996). Y, finalmente, en esta etapa, el equipo de MathWorks desarrolló una herramienta dentro del software MATLAB enfocado a las redes neuronales.

La sexta etapa fue llamada *Computación evolutiva* de principios de los 70's en adelante, en esta etapa se aborda desde la década de los 70's porque solo se tratará de computación evolutiva, los avances que hubieron en esa área. La computación evolutiva funciona simulando una población de individuos, evaluando su rendimiento, generando una nueva población y repitiendo este proceso varias veces. El concepto de algoritmos genéticos fue introducido por John Holland a principios de la década de 1970 en su trabajo *Genetic Algorithms and Adaptation* (Holland, 1984). Desarrolló un algoritmo para manipular cromosomas artificiales (cadenas de dígitos binarios) utilizando operaciones genéticas como selección, cruce y mutación. El objetivo de la programación genética es desarrollar no una representación codificada de algún problema, sino un código informático que resuelva el problema. Esto significa que la programación genética genera programas informáticos como solución. John Koza en sus trabajos (Koza, 1994; Koza et al., 1994), utiliza operaciones genéticas para manipular el código simbólico que representa los programas LISP. La programación genética ofrece una solución al principal reto de la informática: hacer que los ordenadores resuelvan problemas sin ser programados explícitamente. Los algoritmos genéticos, las estrategias evolutivas y la programación genética representaron áreas de la Inteligencia Artificial en rápido crecimiento y con un gran potencial.

La sexta etapa fue llamada *Computación con palabras* de finales de los 80's en adelante, donde principalmente se habla de la programación con palabras, como se representa en los conjuntos difusos. Como se mencionó anteriormente, la teoría de conjuntos difusos fue presentada por Lofti Zadeh en 1965 (L. Zadeh, 1965), y desde entonces fue prácticamente

ignorada en occidente, pero muy aprovechada en el oriente, sobre todo en Japón, que comen-
zaron a fabricar, alrededor de 1987, lavavajillas, lavadoras, aparatos de aire acondicionado,
televisores, fotocopiadoras e incluso coches diseñados que tenían un principio de programa-
ción basado en teoría de conjuntos difusos. A pesar de eso, los sistemas expertos basados en
esta lógica terminan siendo difíciles de desarrollar, porque es necesaria una base de conoci-
miento dada por los expertos, y poner ese conocimiento en palabras que el programa pueda
interpretar puede llevar mucho tiempo, por ejemplo los ingenieros de Hitachi tardaron años
en lograr que el Sistema Subterráneo Sendai lograra trabajar con solo 54 reglas difusas.

Algunos de los artículos y libros más importante en la lógica difusa han sido *Fuzzy Sets*,
Neural Networks and Soft Computing (L. A. Zadeh, 1996) , *The Fuzzy Systems Handbook*
(Cox, 1994) , *Fuzzy Engineering* (Kosko, 1996) , *Expert Systems and Fuzzy Systems* (Negoita,
1984) y *Fuzzy Thinking* (Kosko & Toms, 1994).

Otro evento importante fue la incorporación de la herramienta de lógica difusa para el
software MATLAB, desarrollado por MathWorks.

Con esto se terminan las etapas que describen la historia que ha recorrido el campo de la
Inteligencia Artificial.

2.4.2. Aprendizaje máquina

El aprendizaje máquina es un área de la inteligencia artificial que desarrolla métodos para
que las computadoras aprendan a partir de los datos. Esto permite que las computadoras reali-
cen tareas sin necesidad de que se programe el conjunto de reglas explícitas para realizarlas,
y además puedan adaptarse a condiciones ligeramente diferentes a las que aprendió (Russell
& Norvig, 2016).

Un algoritmo de aprendizaje máquina es un algoritmo que puede aprender de los datos.
(Mitchell, 1997) lo explica de manera formal y breve en la siguiente definición: “Se dice que
un programa de computadora aprende de la experiencia E con respecto a una clase de tareas
 T y una medida de rendimiento P , si su rendimiento en las tareas T , medidas por P , mejoran
con la experiencia E ”.

El aprendizaje máquina sirve para realizar tareas que son muy complicadas de resolver
con programas fijos diseñados por humanos. El aprendizaje es el medio para adquirir la ha-

bilidad de realizar la tarea en cuestión, el aprendizaje no es la tarea en sí.

En general los algoritmos de aprendizaje máquina aprenden a partir de conjunto de datos que también es llamado Dataset. Un Dataset es una colección de ejemplos estructurados según el problema que se deseé resolver. Puede estar compuesto de imágenes con su respectiva etiqueta, imágenes sin etiqueta, datos tabulares, series de tiempo, etc (Goodfellow et al., 2016, p. 103).

Las técnicas de aprendizaje máquina se pueden dividir en varios tipos según el nivel de supervisión que requieren durante el aprendizaje y los datos con los que pueden trabajar:

- **Aprendizaje Supervisado:** Se entregan al algoritmo pares de ejemplos entrada-salida y el objetivo es que aprenda una función que relacione la entrada con la salida. Las tareas más comunes con este tipo de aprendizaje son clasificación y regresión.
- **Aprendizaje No Supervisado:** En este, el algoritmo aprende patrones de los datos de entrada sin que se provea retroalimentación, es decir, no hay un valor de salida. La tarea de aprendizaje no supervisado más común es la clusterización, en la cual se busca agrupar elementos por similitud en diferentes grupos.
- **Aprendizaje Semi Supervisado:** Hay casos en que se tienen pocos datos etiquetados y muchos sin etiquetar. Esta clase de algoritmos pueden aprender de los ejemplos etiquetados y después aplicar aprendizaje no supervisado en los datos sin etiquetar. Un ejemplo de esto son los algoritmos de etiquetado de personas en redes sociales.
- **Aprendizaje Por Refuerzo:** El algoritmo aprende de una serie de “refuerzos” que son recompensas o castigos por las decisiones que toma. Es similar a aprender jugando, y se suele utilizar para que el algoritmo aprenda, por ejemplo, a jugar ajedrez.

En la figura 2.14 se muestra una secuencia de pasos que se dan en el aprendizaje máquina tomando como ejemplo la clasificación de una imagen.

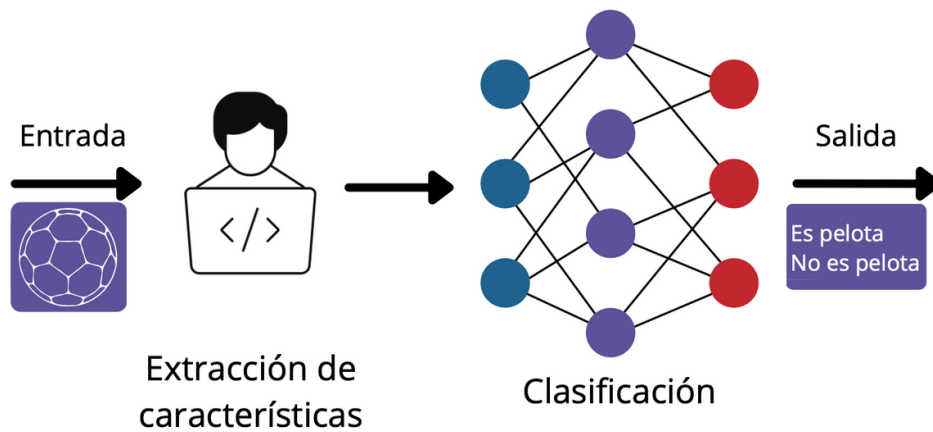


Figura 2.14: Aprendizaje Máquina

2.4.3. Aprendizaje profundo

El aprendizaje profundo ha sido un reto de definir para muchos, porque ha cambiado de forma lentamente en la última década. Una definición útil especifica que el aprendizaje profundo trata de una red neuronal con más de dos capas”. El aspecto problemático de esta definición es que hace que el aprendizaje profundo suene como si existiera desde la década de 1980. Creemos que las redes neuronales han tenido que trascender de forma arquitectónica desde los estilos de red anteriores (junto con mucha más potencia de procesamiento) antes de mostrar los resultados espectaculares que se han visto en años más recientes. A continuación se exponen algunas de las facetas de esta evolución de las redes neuronales:

- Más neuronas que las redes anteriores
- Formas más complejas de conectar capas/neuronas en las redes neuronales
- Explosión en la cantidad de potencia de cálculo disponible para entrenar
- Extracción automática de características

En la figura 2.15 se puede observar un diagrama de Venn de los diferentes campos de estudio de la inteligencia artificial y su relación entre ellos. Y en la figura 2.16 se muestra una secuencia de pasos que se dan en el aprendizaje profundo tomando como ejemplo la clasificación de una imagen.

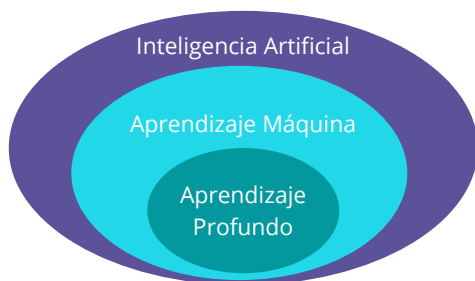


Figura 2.15: Relación entre inteligencia artificial, aprendizaje máquina y aprendizaje profundo

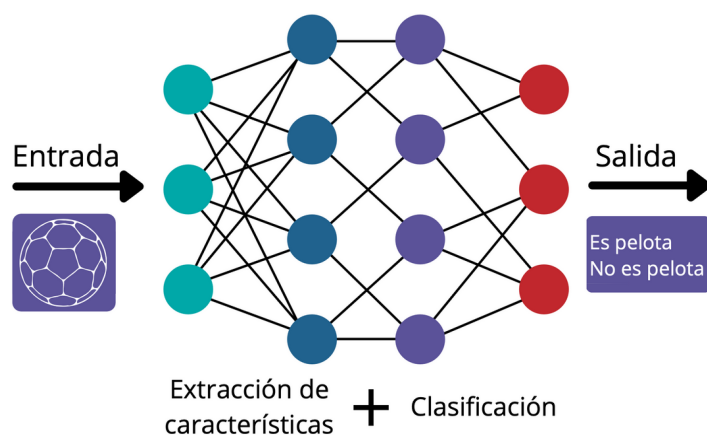


Figura 2.16: Aprendizaje Profundo

2.4.3.1. Redes neuronales

Como se menciona en (Patterson & Gibson, 2017), las redes neuronales son un modelo computacional que comparte algunas propiedades con el cerebro animal en el que muchas unidades simples trabajan en paralelo sin una unidad de control centralizada. Los pesos entre las unidades son el principal medio de almacenamiento de información a largo plazo en las redes neuronales. La actualización de los pesos es la principal forma en que la red neuronal aprende nueva información, esto se abordará más adelante.

Una forma de modelar conjuntos de ecuaciones es en forma de la ecuación $Ax = b$. En el contexto de las redes neuronales, la matriz A son los datos de entrada y el vector de columnas b son las etiquetas o resultados de cada fila de la matriz A . Los pesos de las conexiones de la red neuronal se convierten en x , que sería un el vector de parámetros. El comportamiento de

las redes neuronales viene determinado por su arquitectura de red. La arquitectura de una red puede definirse por:

- Número de neuronas
- Número de capas
- Tipos de conexiones entre capas

La red neuronal más conocida y fácil de entender es la red neuronal multicapa de conexión hacia adelante. Tiene una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa puede tener un número diferente de neuronas, y cada capa está completamente conectada a las capas que le adyacen. Las conexiones entre las neuronas en la capa forman un gráfico acíclico como se muestra en la figura 2.17.

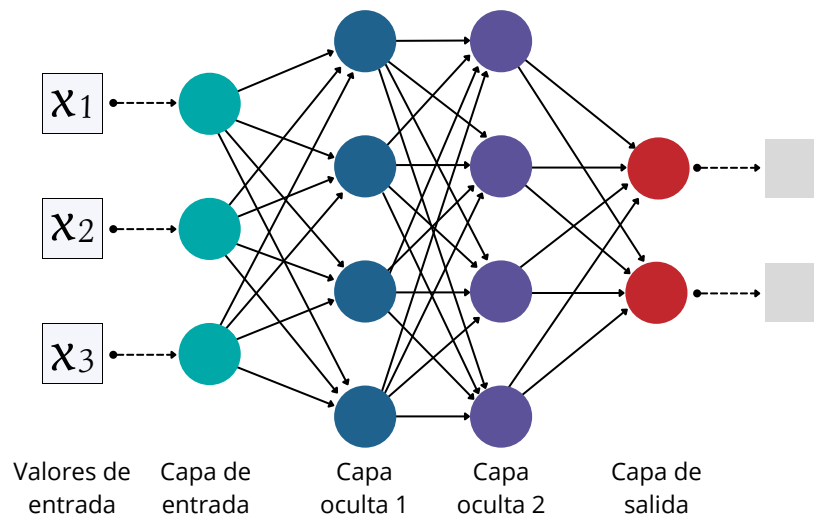


Figura 2.17: Topología de una red neuronal multicapa

2.4.3.2. Perceptrón

Como menciona (Patterson & Gibson, 2017), el perceptrón es un modelo lineal utilizado para la clasificación binaria. En el campo de las redes neuronales, se considera perceptrón a un tipo de neurona artificial que utiliza como función de activación la función escalón de Heaviside, la cual es una función discontinua, donde para todo argumento negativo, su

valor es 0, y para cada argumento positivo, su valor es 1. El predecesor del perceptrón fue la Unidad Lógica de Umbral (TLU, por sus siglas en inglés) desarrollada por Warren McCulloch y Walter Pitts en 1943, que podía aprender las funciones lógicas AND y OR. El algoritmo de aprendizaje de perceptrón se considera un algoritmo de aprendizaje supervisado. Tanto la TLU como el perceptrón están inspirados en neuronas biológicas.

Entrando un poco más en materia, el perceptrón es un clasificador binario de modelo lineal con una relación simple de entrada-salida, como se muestra en la figura 2.18, que muestra que se está sumando n número de entradas por sus pesos asociados y luego enviando esta “entrada neta“ a una función escalonada con un umbral definido. Normalmente, con los perceptrones, se trata de una función escalonada de Heaviside. Esta función dará como resultado un único valor binario real (0 o 1), dependiendo de la entrada.

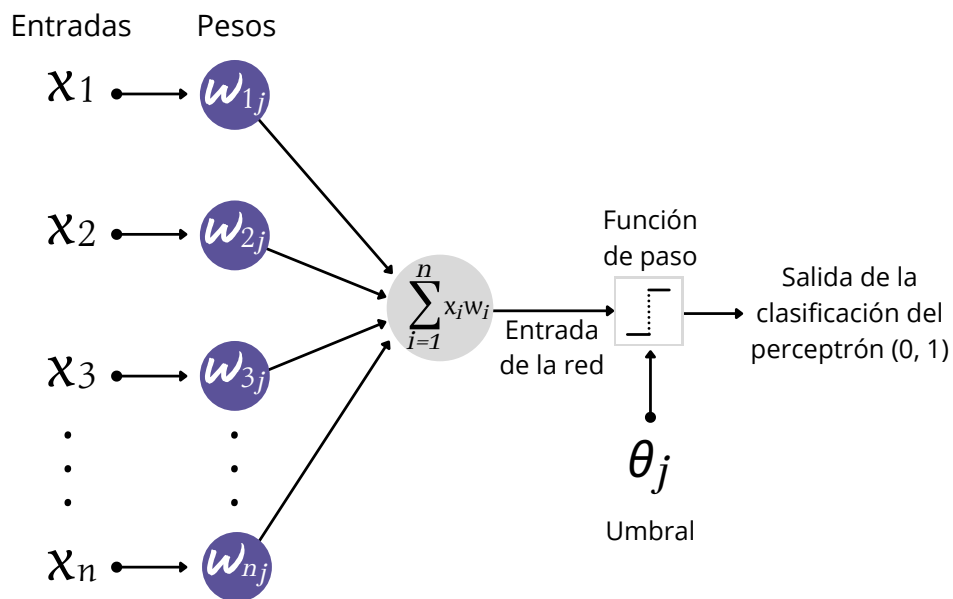


Figura 2.18: Perceptrón de una sola capa

2.4.3.3. Redes neuronales multicapa

En (Patterson & Gibson, 2017) se menciona que el perceptrón es más conocido como “perceptrón de una sola capa” en la investigación de redes neuronales para distinguirlo de su sucesor el “perceptrón multicapa”. La neurona artificial del perceptrón multicapa es similar a la del perceptrón original, pero con la singularidad que este añade flexibilidad en el tipo de

capa de activación que se puede utilizar. La figura 2.19 muestra un diagrama actualizado de la neurona artificial que se basa en el perceptrón.

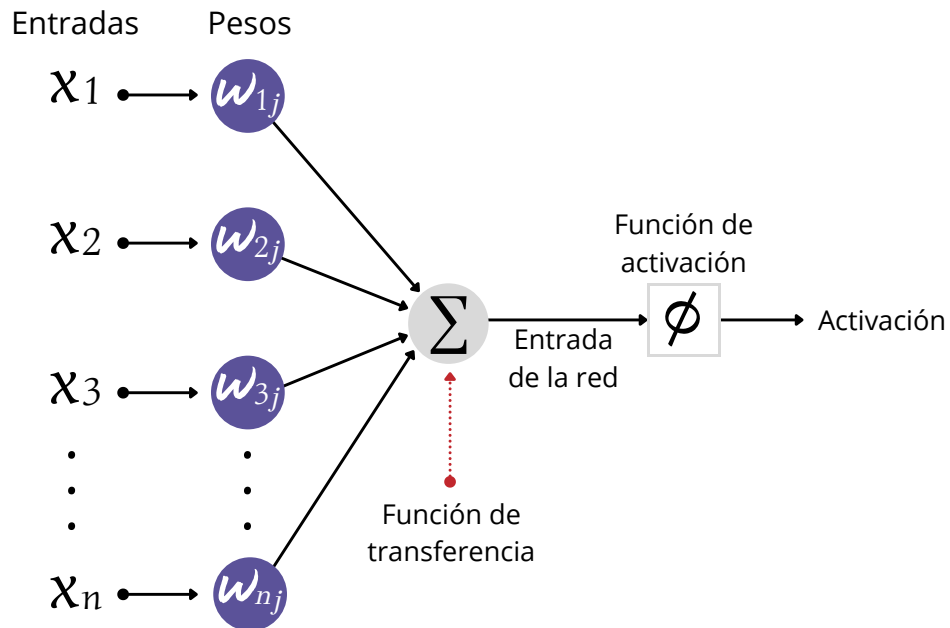


Figura 2.19: Neurona artificial del perceptrón multicapa

2.4.3.4. Entrenamiento

En (Patterson & Gibson, 2017) se explica que una red neuronal artificial debidamente entrenada tiene pesos que amplifican la señal y amortiguan el ruido. Un peso mayor significa una correlación más estrecha entre una señal y el resultado de la red. Las entradas emparejadas con pesos grandes afectarán más a la interpretación de los datos por parte de la red que las entradas emparejadas con pesos más pequeños.

El proceso de aprendizaje de cualquier algoritmo que utilice pesos se trata de reajustar los pesos y sesgos, haciendo algunos más pequeños y otros más grandes, asignando así importancia a ciertos bits de información y minimizando otros bits. Esto ayuda al modelo a aprender qué características están vinculados a qué resultados y ajusta las ponderaciones y los sesgos en consecuencia.

En la mayoría de los conjuntos de datos, ciertas características están fuertemente correlacionadas con ciertas etiquetas (por ejemplo, los metros cuadrados se relacionan con el

precio de venta de una casa). Las redes neuronales aprenden estas relaciones a ciegas, haciendo una conjetura basada en las entradas y los pesos, y después midiendo la precisión de los resultados. Hay algoritmos de optimización que recompensan a la red por las suposiciones acertadas y la penaliza por las equivocadas, esto es una función de pérdida, y ayuda a la red a reducir el error de predicción.

Otra forma de ver el proceso de aprendizaje es considerar las etiquetas como teorías y el conjunto de características como pruebas. Entonces, se puede hacer la analogía de que la red busca establecer la correlación entre la teoría y las pruebas.

2.4.3.5. Algoritmo de propagación hacia atrás (Backpropagation)

La propagación hacia atrás es una parte importante para reducir el error en un modelo de red neuronal. Este es como el algoritmo de aprendizaje del perceptrón. Se requiere calcular la salida del ejemplo de entrada con un pase hacia adelante a través de la red. Si la salida coincide con la etiqueta, no se hace nada nada. Si la salida no coincide con la etiqueta, se deben ajustar los pesos de las conexiones en la red neuronal, esto se explica en (Patterson & Gibson, 2017).

2.4.4. Visión artificial

En (Asad et al., 2020) se menciona que cuando la inteligencia artificial se centra en las imágenes y los vídeos, el ámbito se denomina visión artificial. La cual es un conjunto de algoritmos y técnicas que permiten aplicar los conceptos de aprendizaje profundo, donde se utilizan redes neuronales para completar tareas como la detección de objetos, la clasificación de imágenes o, en este caso, detección de patrones para determinar ángulos.

2.5. Tecnologías utilizadas

Para desarrollar este proyecto, fueron muchas las herramientas de desarrollo y los lenguajes de programación utilizados, además de las librerías y frameworks que ayudaron a simplificar, en cierta medida, algunas tareas que realiza el sistema.

Al desarrollar el sistema para la plataforma de NVIDIA Jetson Nano, era necesario que la implementación final del mismo estuviera programada en el lenguaje de programación Python. Para el caso de la medición del ángulo, se utiliza la librería OpenCV, más precisamente la función *cornerSubPix* la cual localiza en la imagen un patrón de un cuadro dividido en 4, con 2 cuadros negros y dos blancos, como se muestra en la figura 2.20. En el capítulo de Metodología se abordará más a fondo el algoritmo que se utiliza, pero en general lo que se hace es colocar etiquetas a 3 puntos del sistema; la masa, el motor y una referencia horizontal, dicha función los detecta y obtiene sus coordenadas, con esas coordenadas se hacen cálculos matemáticos para determinar el ángulo del péndulo.

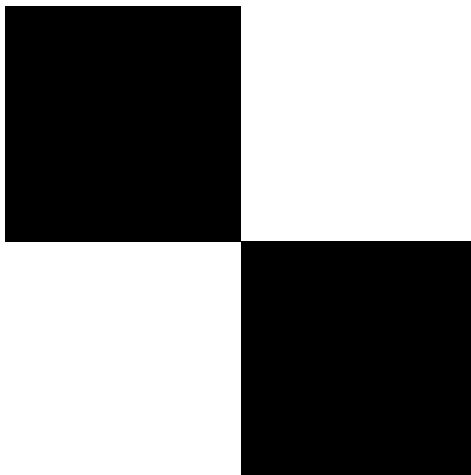


Figura 2.20: Patrón para detección en OpenCV

Para la parte del desarrollo del sistema de control y hacer las simulaciones, se utilizó la plataforma Matlab, con su herramienta Simulink, las cuales se usaron para, en primera instancia, mostrar como trabaja el sistema de manera visual, con bloques de control, y, posteriormente, diseñar un sistema de control que se adapte a las necesidades del problema, en el capítulo de Metodología se hablará más a fondo de estos sistemas.

Finalmente, es importante comentar que gran parte de la programación en Python se desarrolló en las IDE's Spyder y PyCharm, las cuales ayudaron a desarrollar el sistema que posteriormente ejecutaría le NVIDIA Jetson Nano, además de usar las herramientas Git y GitHub para el control de versiones de la programación del sistema.

2.5.1. OpenCV

Open Source Computer Vision Library, o por su abreviatura OpenCV, es una librería de código abierto que contiene una gran cantidad de algoritmos de visión artificial. Originalmente su programación está hecha en C++, en un paradigma orientado a objetos, pero tiene implementación en otros lenguajes como Python, Java, JavaScript, Matlab y Octave, aunque su documentación está mayormente enfocada en C++ y Python. OpenCV fue inicialmente desarrollado por Intel en 1999, y, como su mismo nombre lo dice, tiene la filosofía de ser de código abierto, lo que significa que además de ser gratuito, pone a disposición del público su código fuente. En la figura 2.21 se muestra una imagen del logotipo de OpenCV.



Figura 2.21: Logotipos de OpenCV (Fuente: OpenCV team)

Como se mencionó antes, dentro de esta librería existe una función llamada *cornerSubPix*, la cuál obtiene las coordenadas de un patrón de un cuadro dividido en 4, con 2 cuadros negros y 2 blancos (OpenCV, s.f.).

2.5.2. MATLAB

Como mencionan en (MathWorks, s.f.), MATLAB es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos, esta fue desarrollada por la empresa MathWorks en 1984, su nombre "MATLAB" proviene de 2 palabras *MATrix LABORatory*, y describe lo que es su

principal función, las operaciones con matrices. Pero también es una plataforma utilizada para desarrollar y simular sistemas de control, gracias a su herramienta *Simulink*, la cual permite modelar sistemas de control de forma gráfica con bloques, además de hacer simulaciones y graficar datos de forma muy sencilla (MathWorks, s.f.). En la figura 2.22 se muestra una imagen del logotipo de MathWorks, MATLAB y Simulink.

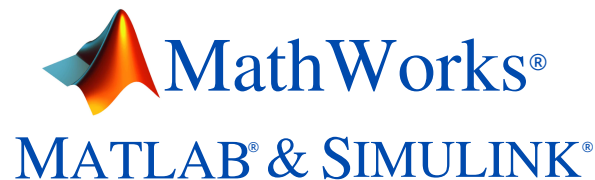


Figura 2.22: Logotipos de MathWorks, Matlab y Simulink (Fuente: The MathWorks, Inc.)

En este proyecto el primer diseño del sistema de control se realizó en MATLAB, y las simulaciones correspondientes se hicieron en Simulink, al tener resultados suficientemente buenos, se hizo la migración de la programación a Python, pero gracias a MATLAB el desarrollo del sistema de control se simplificó en gran medida.

2.5.3. Python

Como se mencionó anteriormente, el lenguaje de programación principal que requería este proyecto es Python, este es un lenguaje de programación interpretado de alto nivel, con una sintaxis muy limpia, semejante a lo que sería un pseudocódigo. En los últimos años, Python ha obtenido una gran popularidad, desde los principiantes para que sea su primer lenguaje de programación al ser muy expresivo y sencillo de entender, y también por programadores con más recorrido que trabajen en áreas de Inteligencia Artificial o Ciencia de Datos. En la figura 2.23 se muestra una imagen del logotipo de Python.



Figura 2.23: Logotipo de Python (Fuente: Python Software Foundation)

Es un lenguaje de programación en el cuál se han desarrollado muchas librerías para prácticamente cualquier tipo de aplicación, lo que lo convierten en una herramienta muy poderosa para la ciencia e ingeniería (Oliphant, 2007). Las características más relevantes son:

- Código abierto: Los sistemas creados en Python se pueden usar, distribuir y vender sin restricciones.
- Multiplataforma: Python puede ejecutarse en la mayoría de sistemas operativos y dispositivos.
- Multiparadigma: Se puede programar con el paradigma orientado a objetos, o cambiar de paradigma según la necesidad.
- Intérprete Interactivo: El intérprete de Python puede mostrar los resultados de forma interactiva e instantánea.
- Extensible: Para tareas que requieran un mayor rendimiento, se pueden agregar módulos escritos en otros lenguajes de bajo nivel.
- Soporte de la comunidad: Existe una gran comunidad que utiliza Python, y resulta fácil obtener ayuda sobre su uso y aplicaciones.
- Repositorio de librerías: Se pueden instalar fácilmente librerías adicionales desde el repositorio PyPi que contiene una gran multitud de herramientas.

Capítulo 3

Estado del Arte

Ha habido acercamientos a soluciones que involucren técnicas de deep learning, o inteligencia artificial, como sistema de control para plataformas convencionales, como lo sería el péndulo invertido, también hay alternativas usando técnicas de visión artificial en lugar de los sensores como acelerómetro o giroscopio, a continuación se verán cuales han sido, además de mostrar los sistemas de control convencional para este tipo de casos.

3.1. Sistemas convencionales

Para comenzar, la plataforma más convencional para comparar nuevos sistemas de control es el péndulo invertido con un sistema de control PID y con sensores como acelerómetro y giroscopio para determinar la posición de este, desde hace muchos años que existe esta implementación, y hay muchos artículos tanto nuevos, como antiguos que hablan y hacen pruebas acerca de esto, pero del que se hablará aquí es uno desarrollado en Noviembre de 2018, Buddhika Abeysekera y su co-autor Indika Wanniarachchi desarrollaron el artículo llamado *Modelling and Implementation of PID Control for Balancing of an Inverted Pendulum*, el cuál es una muestra muy básica del funcionamiento de un péndulo invertido acoplado a una especie de carro, movido por poleas, y del control PID y usa un giroscopio para determinar la inclinación, además de implementar el algoritmo de predicción de Kalman (Abeysekera & Wanniarachchi, 2018). En la figura 3.1 se muestra la parte física del sistema que diseñaron Abeysekera y Wanniarachchi.

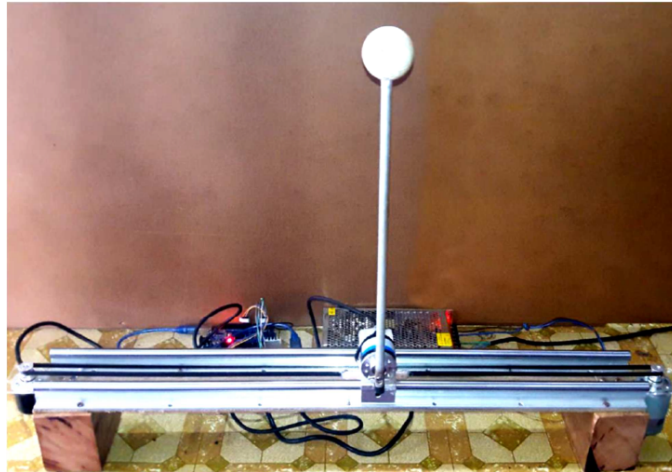


Figura 3.1: Péndulo controlado por PID (Fuente: (Abeysekera & Wanniarachchi, 2018))

Abeysekera y Wanniarachchi comentan haber obtenido buenos resultados, aun cuando insertaban perturbaciones intencionalmente, se lograba la estabilidad, y gracias al algoritmo de Kalman se redujeron las señales de ruido innecesarias y la discretización de los datos, y dicen que su sistema podía ser replicado en robots auto-equilibrados, aviones no tripulados (quad copters o drones) e incluso, en sistema de guiado de misiles. El problema de este sistema es que no se adapta a variaciones, trabaja bien con funciones cíclicas y hasta cierto punto sencillas, además que es mejor solo utilizarlo en sistemas informáticos de baja potencia, como los microcontroladores.

3.2. Sistemas con implementación de algoritmos de visión artificial

Como se menciona anteriormente, los péndulos invertidos deben tener un sensor que indique, mínimamente, su posición. Habitualmente se usan giroscopios, pero gracias al auge de la visión artificial o la visión computacional, hay personas que han intentado emular lo que hacen estos sensores, pero obteniendo dichos datos con técnicas de visión artificial, un artículo que relata esto es *Vision-based control of a flying spherical inverted pendulum on a Quadrotor* desarrollado en 2020 por Soukaina Krafes, Zakaria Chalh y Abdelmjid Saka. Este artículo, tal como mencionaba el escrito por Abeysekera y Wanniarachchi, usan el sistema de

control sobre un avión no tripulado del tipo Quad copter, poniendo una cámara en la parte superior, y su función es estabilizar dicho avión no tripulado, debido a que tiene 4 helices, y para siempre conservar una tendencia horizontal, tal como muestra la figura 3.2. Este dron pudiera verse como un péndulo invertido esférico, y, a pesar de usar la visión artificial para la obtención de la posición, el sistema de control sigue siendo uno convencional (Krafes et al., 2020).

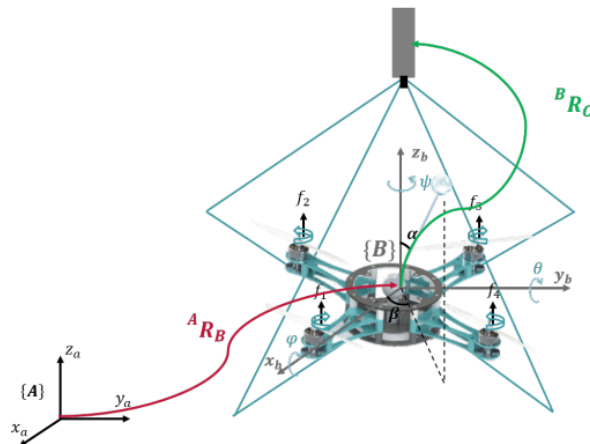


Figura 3.2: Dron estabilizado gracias a visión artificial (Fuente: (Krafes et al., 2020))

En 2014, Damian Garayalde y Pabo Negri presentaron un artículo donde desarrollan un sistema de visión artificial para detectar el ángulo de un péndulo invertido, ayudándose del filtro de Kalman, en este caso usaron un control de realimentación lineal de estados, los errores obtenidos por el sistema de medición fueron de 0.23° , lo que lo vuelve algo prometedor, el problema es que hubo múltiples complicaciones como la disposición geométrica de los puntos de referencia, porque depende de ellos la precisión del ángulo detectado, la iluminación que se cuenta en la sala de pruebas, la detección de falsos positivos y la velocidad de procesamiento (en este caso se obtenían velocidades de entre 5 y 10 milisegundos) (Garayalde & Negri, 2014). En la figura 3.3 se muestra como se ve el prototipo de péndulo del que se detecta el ángulo.

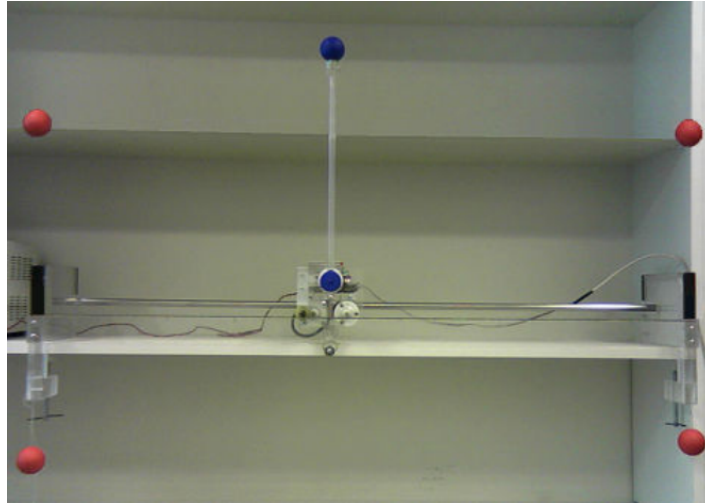


Figura 3.3: Imagen del péndulo con detección de ángulo por visión artificial (Fuente: (Garayalde & Negri, 2014))

3.3. Sistemas de control con implementación de inteligencia artificial

La idea de incluir técnicas de Inteligencia Artificial en sistemas de control clásicos ha estado presente por décadas, y una de las teorías más rápido adaptadas fue la lógica difusa. El control por lógica difusa ya forma parte de la teoría de control moderna, como se menciona anteriormente desde los años 80 ´s en Japón ya había electrodomésticos que funcionaban con esta lógica. En 2002, Samir Kouro y Rodrigo Musalem desarrollaron un control usando lógica difusa par controlar un prototipo de helicóptero, por su naturaleza, el control de estabilidad de un helicóptero podría ser visto como un péndulo invertido con 2 actuadores, las hélices, una principal y una de apoyo como se muestra en la figura 3.4. Para este proyecto desarrollaron un artículo, en el que describen su proceso y características de su sistema, en este caso usaron un sensor óptico para determinar su inclinación, este sensor también puede ser llamado encoder (Kouro & Musalem, 2002). Sobre este proyecto se rescatan una ventaja y una desventaja, como los mismos autores comentan, el control por lógica difusa resulta más sencillo de diseñar que un método de control clásico, ya que este no necesita de un modelo riguroso de la planta, pero una de las desventajas son los altos costos de procesamiento, ya que

el muestreo, fusificador y defusificador resultan procesos más complejos que una operación matemática que realizan los sistemas de control convencionales.

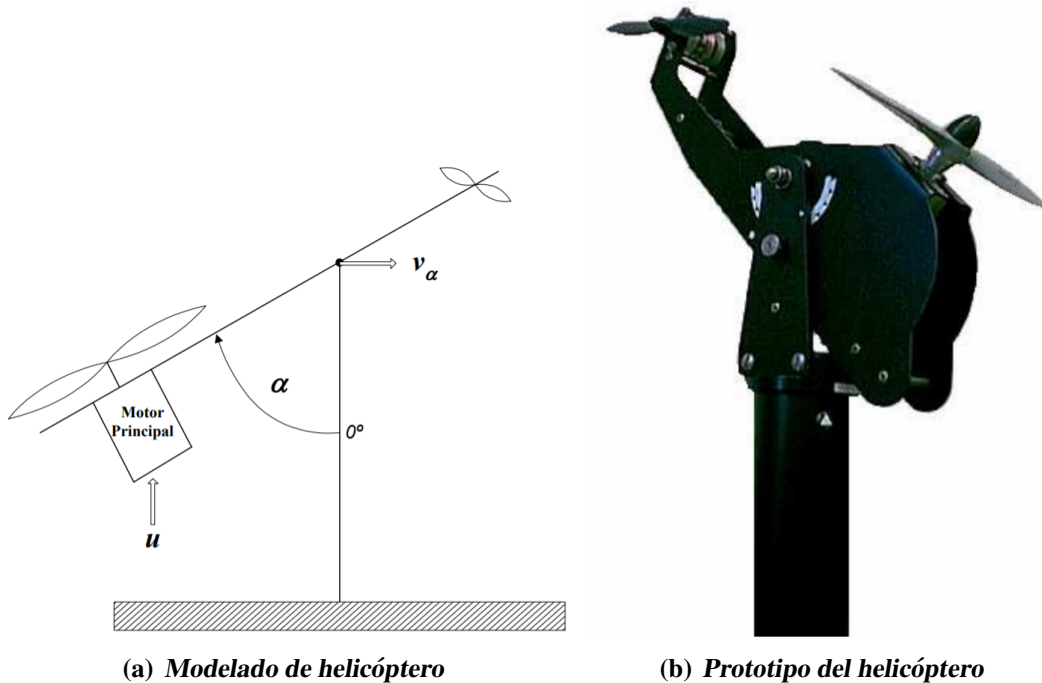


Figura 3.4: Modelo esquemático y físico del helicóptero (Fuente: (Kouro & Musalem, 2002))

Para 2013, Juan Guillermo Valenzuela Hernández, Oscar Danilo Montoya Giraldo y Didier Giraldo Buitrago, realizaron un modelado de un sistema de control de un péndulo invertido también basado en lógica difusa, de este proyecto no se realizó un prototipo físico, pero en su artículo describen como se realizaron las simulaciones y con qué parámetros probaron el sistema (Hernández et al., 2013). Este péndulo tenía la peculiaridad de ser del tipo de rueda de reacción, en este caso el actuador está incluido en la masa del péndulo, como muestra la figura 3.5. Similar a lo que se comenta en (Kouro & Musalem, 2002), una de las bondades de usar un sistema de control con base en la lógica difusa es que no es necesario un modelo matemático riguroso de la planta, eso hace que sea más sencillo aplicarlo. Pero una de las complejidades es que la eficacia de este sistema de control recae directamente en el conocimiento del experto que describe la dinámica del sistema mediante sentencias lingüísticas.

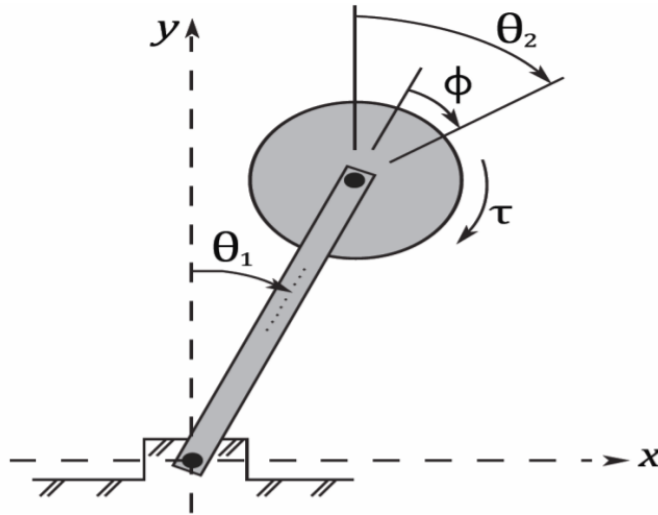


Figura 3.5: Representación esquemática del péndulo invertido con rueda de reacción (Fuente: (Hernández et al., 2013))

3.4. Comparación del control PID, PD y SMC

Finalmente, en el año 2010, A. N. K. Nasir, R. M. T. Raja Ismail y M. A. Ahmad, publicaron un artículo para comparar el control convencional PD y PID, con el Control por Modos Deslizantes (SMC, por sus siglas en inglés) (Nasir et al., 2010), probando estos 2 sistemas de control en una plataforma de péndulo invertido de carro, como lo muestra la figura 3.6. Sus resultados mostraron que ambos controladores lograron controlar la posición del péndulo de manera correcta, pero el SMC produce una mejor respuesta en comparación con el PID, ya que se adapta mejor a las perturbaciones, como se muestra en las Tablas 3.1 y 3.2. También es importante destacar que todo el estudio realizado en (Nasir et al., 2010) se basó puramente en simulaciones hechas en MATLAB y Simulink .

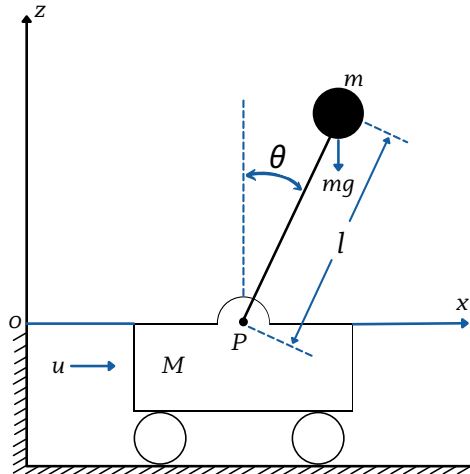


Figura 3.6: Diagrama esquemático de un péndulo invertido de carro

Tabla 3.1: Resumen de las características de rendimiento de posición del carro del péndulo invertido entre SMC y PID (Fuente: (Nasir et al., 2010))

Especificación del tiempo de respuesta	SMC	PID
Tiempo de subida	1.01 sec	1.02 sec
Tiempo de estabilización	2.33 sec	3.52 sec
Error de estado estacionario	0.00	0.00
Porcentaje de rebase máximo	0.5 %	5 %

Tabla 3.2: Resumen de las características de rendimiento de posición del ángulo del péndulo invertido entre SMC y PID (Fuente: (Nasir et al., 2010))

Especificación del tiempo de respuesta	SMC	PID
Tiempo de subida	0.14 sec	0.16 sec
Tiempo de estabilización	3.03 sec	2.93 sec
Error de estado estacionario	0.00	0.00
Rebase máximo	0.5 radianes	0.55 radianes

Si bien, esta última investigación se asemeja a lo que busca el proyecto actual, difiere por el hecho de que, además de hacer pruebas de los diferentes sistemas de control en un prototipo de péndulo invertido físico, este pondrá a prueba distintos sistemas de control, alterando diferentes parámetros, hasta conseguir una comparativa más clara de las diferencias, ventajas y desventajas de cada uno de los sistemas de control.

Capítulo 4

Metodología Propuesta

Para el desarrollo de este proyecto fue necesaria la utilización de hardware específico, a continuación se hará una explicación de este equipo y componentes electrónicos, y, posterior a eso, se describirá el desarrollo del sistema a nivel computacional.

4.1. Hardware

Como se comentó anteriormente, hay equipo necesario para este proyecto, su misma naturaleza indica cuáles son sus componentes fundamentales. A continuación se hace un listado de este hardware necesario para el funcionamiento de este sistema. También es importante aclarar que en el caso de escalar este sistema, sería necesario cambiar componentes como el motor, y por ende, la fuente de alimentación.

- **Cámara:** Al tratarse de un proyecto que involucra la visión artificial, es necesario usar una cámara, para este caso se utilizó una IMX219, la cual es un sensor de imagen de tipo píxel activo CMOS diagonal de 4.60 mm (Tipo 1/4.0), con una matriz de píxeles cuadrados, y 8.08M de píxeles efectivos. Este chip funciona con tres fuentes de alimentación, una analógica de 2.8 Volts, otra digital de 1.2 Volts y una IF de 1.8 Volts. Esta cámara es ideal para este proyecto por su compatibilidad con el siguiente dispositivo del hardware, la NVIDIA Jetson Nano, además de existir una comunidad a la cual recurrir en caso de algún inconveniente o duda con el equipo.
- **NVIDIA Jetson Nano:** Este es el núcleo de sistema, es donde se ejecutan los algoritmos de visión artificial y también del sistema de control, aunque este último lo comparte

con el siguiente dispositivo de la lista, el Arduino UNO. En cuanto a la NVIDIA Jetson Nano, es una pequeña computadora, la cual ofrece 472 giga-FLOPS de rendimiento, con una CPU con arquitectura ARM de 64 bits de cuatro núcleos y una GPU NVIDIA integrada de 128 núcleos, lo que la hace potente e ideal para tareas de visión artificial y redes neuronales comparándola con otras computadoras del mismo tipo. También incluye memoria LPDDR4 de 4 GB en un paquete eficiente de bajo consumo con modos de alimentación de 5 y 10 watts dependiendo del uso que se le vaya a dar, y entrada de 5 volts de corriente directa.

- **Arduino UNO:** Como se mencionó anteriormente, la NVIDIA Jetson Nano comparte las tareas de control en algunas pruebas y aplicaciones con el Arduino UNO. Esta es una placa de desarrollo que su núcleo es un microcontrolador ATmega328P, la cual cuenta con terminales para entrada y salida de datos, y se utilizan para recibir información de sensores, o bien, para enviar señales de encendido, apagado y PWM. El Arduino UNO tiene un voltaje de operación de 5 volts, 14 pines digitales de entrada y salida, de los cuales 6 pueden generar una salida PWM, y 6 pines analógicos de entrada.
- **Relevador de potencia:** Este es un dispositivo que se usa en electrónica de potencia para acoplar dos circuitos que trabajan a diferentes voltajes, inclusive a diferentes tipos de corrientes, ya sea alterna o continua. La función principal de este componente es el de distribuir una señal eléctrica, esto con un pin de "activación", un pin de entrada donde se conecta la fuente de alimentación y un pin de salida a donde se dirige el voltaje/corriente de esta fuente, para el caso de este sistema, esa salida va al motor CD del que se hablará a continuación.
- **Motor CD:** Este es el dispositivo a controlar, entonces es el necesario para ejecutar los movimientos del mecanismo. Lo que hace este dispositivo es transformar la energía eléctrica en energía mecánica, en este caso generando una rotación o desplazamiento angular. Esto es debido a que está compuesto de un campo magnético en su parte estática, generalmente compuesto por imanes, y en su parte móvil tiene un bobinado, el cual cuando es energizado genera otro campo magnético, esto hace que ambos campos se atraigan y se repelen, generando así un giro. En particular, el motor utilizado en este

proyecto, funciona con corriente directa a 5 volts, que es la que proveen las baterías comunes, dependo la polaridad a la que se haga la conexión, se obtiene un giro a favor o en contra de las manecillas del reloj, y bajo ciertas condiciones se puede obtener un giro más o menos veloz.

- **Batería:** Este es el dispositivo que sirve como fuente de alimentación para el motor, y en este proyecto las características necesarias con las que debía contar la batería para que el motor pudiera girar era de 5 voltios y, al menos, 1 ampere.

4.1.1. Ensamblaje de componentes

En esta sección se muestra la forma en que se conectaron los componentes. En la figura 4.1 se puede apreciar una representación visual de los componentes ensamblados, se aprecia la cámara y su conexión a la NVIDIA Jetson Nano, la cual a su vez está conectada al Arduino UNO, y finalmente este manda la señal al motor.

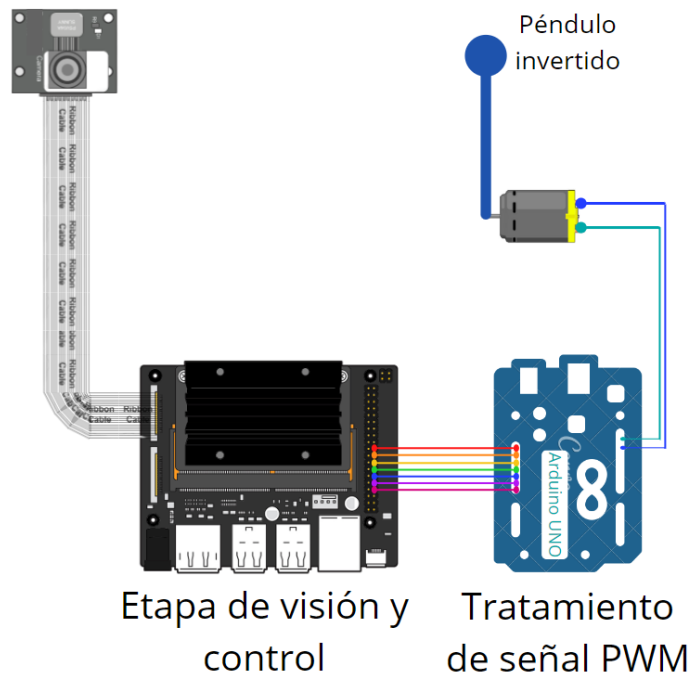


Figura 4.1: Representación visual de la conexión de los componentes

4.2. Desarrollo de subsistemas

Desde un inicio se propuso desarrollar dos subsistemas aislados que trabajaran en conjunto para un sistema general, como ya se ha comentado con anterioridad, dichos subsistemas son el sistema de visión artificial para determinar la posición y velocidad angular de un mecanismo, en este caso péndulo invertido. Y el otro subsistema sería el sistema de control por modos deslizantes apoyado por una red neuronal, a continuación se describe el desarrollo de estos subsistemas.

4.2.1. Sistema de visión artificial

El sistema de visión artificial propuesto tiene como entrada las imágenes obtenidas, y da como resultado un ángulo y un valor de velocidad.

Hubo una serie de posibilidades en cuanto a la forma en que el algoritmo iba a detectar estos datos, desde visión artificial con reconocimiento de patrones adaptando cálculos geométricos, una red neuronal con un dataset de los diferentes ángulos, una red neuronal de reconocimiento de los puntos importantes siendo el motor y la masa del péndulo agregando cálculos de geometría.

Tomando en cuenta los requisitos de usuario de los que se hablará posteriormente, se decidió cuál sería la lógica a la que se adaptaría el algoritmo de visión.

4.2.2. Sistema de control

Para el caso del controlador, la función principal era que tuviera el apoyo de una red neuronal o técnicas de inteligencia artificial.

Lo que hacía que fuera una tarea muy general, ya que podría adaptarse la red neuronal para que fuera directamente la que controlara el mecanismo determinando sentido y velocidad, pero esto conllevaba realizar una serie de pruebas y adaptar el modelo a este tipo de control. Otra opción era un control dual, que a la vez que funcionara un control convencional, la red neuronal aprendiera los movimientos para posteriormente el mecanismo solo fuera manipulado por esta. Y también estaba la opción de utilizar un sistema de control convencional y que

se le adaptara una red para predecir el movimiento del mecanismo, y de este modo, ejecutar movimientos más precisos y ágiles.

Como se mencionó en el subsistema anterior, la decisión de qué alternativa utilizar se determinó por las necesidades del proyecto obtenidas por los requisitos de usuario.

4.3. Análisis de requisitos

En esta sección, a partir de las necesidades e información del problema a resolver con este proyecto, se generan una serie de requisitos de usuario, con los cuales se obtienen cuáles corresponden a los requisitos funcionales y cuáles a los requisitos no funcionales, o de calidad.

4.3.1. Requisitos funcionales

Como su nombre lo dice, los requisitos funcionales describen funciones específicas que debe realizar el sistema, para este sistema los requisitos funcionales planteados son:

- RF-01: El sistema deberá captar una imagen para posteriormente analizarla.
- RF-02: El sistema deberá analizar la imagen captada y determinar la posición a la que se encuentra el mecanismo.
- RF-03: El sistema deberá estimar la velocidad angular mediante la posición angular calculada.
- RF-04: El sistema deberá determinar una señal de control a partir de los datos de posición y velocidad angular.
- RF-05: El sistema deberá manipular un motor CD por medio de una señal de control enviada.
- RF-06: El sistema deberá mantener la posición angular del mecanismo según sea la necesidad.

4.3.2. Requisitos no funcionales

Los requisitos no funcionales, o de calidad, son los que, si bien, no interfieren directamente con las funciones del sistema, si terminan definiendo si un sistema cumple, o no, correctamente con las necesidades.

- RC-01: El sistema de visión artificial deberá captar correctamente una imagen bajo condiciones lumínicas adecuadas.
- RC-02: El sistema deberá ser ejecutado en una distribución de Linux y programado en lenguaje Python.
- RC-03: El sistema de visión artificial deberá ser capaz de detectar un ángulo en menos de 0.005 segundos.
- RC-04: El sistema de control deberá determinar y enviar la señal de control en menos de 0.005 segundos.
- RC-05: El sistema deberá permitir mejoras de manera sencilla.

4.3.3. Actores

A continuación se describen los actores, los cuales son la representación de todo lo que interactúa con el sistema para su correcto funcionamiento. Este sistema solo cuenta un actor, el cual es el operador, no hay otro usuario que interactúe directamente con el sistema para que este pueda funcionar. En la figura 4.2 se muestra el actor del sistema.

- Operador: Se encarga de realizar los procedimientos para que el sistema inicie sus funciones.

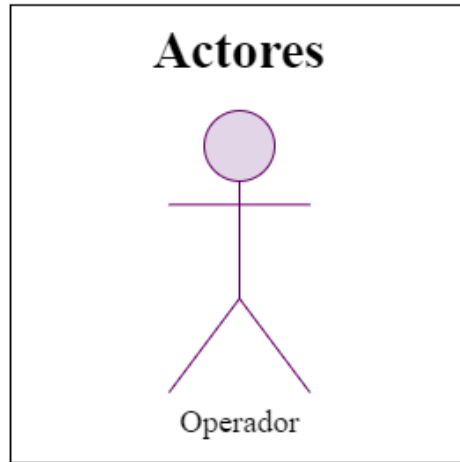


Figura 4.2: Actores del sistema

4.3.4. Casos de uso

A partir de los requisitos funcionales que fueron detectados con anterioridad, se definen cuáles serán los casos de uso, los cuales son escenarios donde los usuarios interactúan con las funcionalidades del sistema al llevar a cabo el proceso. A continuación se muestra el único caso de uso del sistema, y posterior a eso, en la figura 4.3 el diagrama de este.

- Iniciar: Es la acción donde se pone en marcha el sistema.

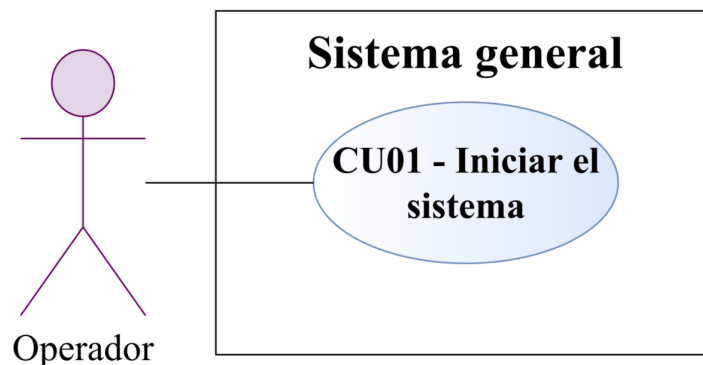


Figura 4.3: Diagrama de casos de uso

4.3.5. Diagrama de contexto

En esta sección se analizará el diagrama de contexto, y es mediante el cuál se describe el flujo de la información dentro del sistema, y de entender la forma en que interactúan los componentes y métodos. En la figura 4.4 se puede observar el diagrama de componentes que describe a este sistema. Si bien, hay diagramas en donde hay múltiples niveles, para este sistema solo fueron necesarios dos; “Usado por” y “Usa a”, por la misma naturaleza del proyecto. El nivel de “Usado por” se refiere directamente a los usuarios, en este caso solo sería un operador, y para el nivel de “Usa a” se refiere a los componentes que son usados o interaccionan con el sistema nuclear, en este caso sería la cámara para la etapa de detección de posición y velocidad angular, y el motor, el cual lleva a cabo los movimientos dictados por la etapa de control.

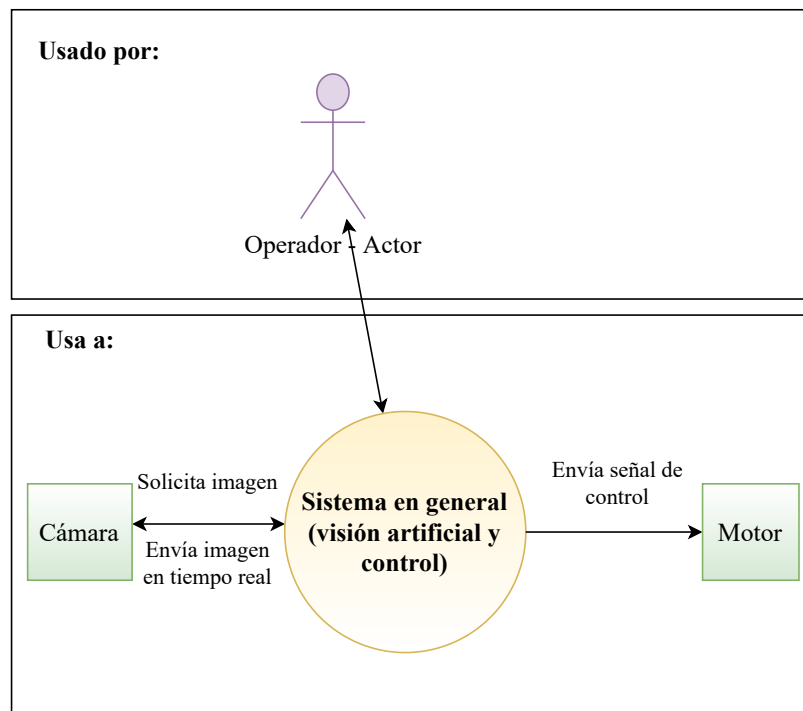


Figura 4.4: Diagrama de contexto

- **Operador:**

Es el encargado de encender y/o apagar el sistema.

- **Cámara:**

Es el dispositivo encargado de recopilar las imágenes en tiempo real para enviarlas al sistema de visión artificial.

- **Motor:**

Es el dispositivo que ejecuta las señales de control dictadas por la etapa de control.

4.3.6. Arquetipos

Los arquetipos corresponden a un patrón o modelo. Son los elementos más importantes para el sistema, al tratarse de la representación más abstracta de las entidades dentro del sistema, de este modo se puede explicar de manera general el cómo se comporta el sistema. A continuación se describen los arquetipos para representar el sistema actual.

- **Operador:**

Es la representación de la persona que encenderá el sistema.

- **Visión:**

Es la abstracción del subsistema que determina la posición y velocidad angular del mecanismo.

- **Control:**

Es la abstracción del subsistema que determina la señal de control que será enviada al motor acoplado al mecanismo para manipular su movimiento.

- **Péndulo:**

Es la representación visual de lo que capta la cámara y será utilizada como entrada para el algoritmo de visión.

- **Ángulo y velocidad:**

Es la representación del resultado emitido por el algoritmo de visión artificial que indica la posición y velocidad angular del mecanismo.

- Señal:

Es la representación de la señal emitida por el sistema de control, y que se envía al motor.

En la figura 4.5 se muestra el diagrama generado a partir de los arquetipos propuestos y sus interacciones.

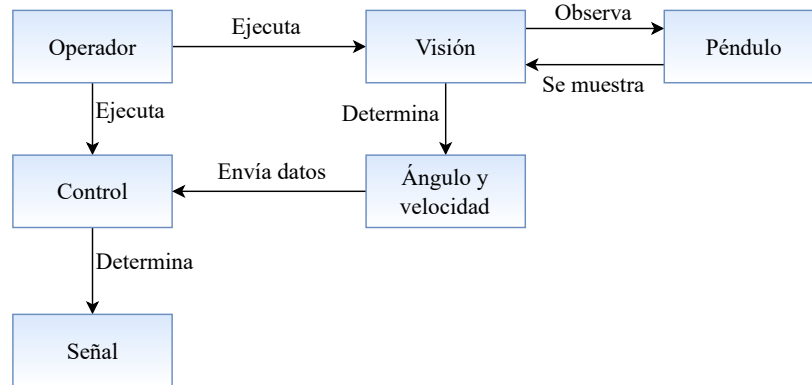


Figura 4.5: Diagrama de arquetipos

4.3.7. Arquitectura

La arquitectura planteada para este proyecto es la de tubos y filtros, debido a la naturaleza misma del sistema que busca que los datos de entrada se transformen en cada etapa, refiriéndose a los subsistemas el sistema general, y obteniendo un dato de salida que haya pasado por todas las transformaciones. En la figura 4.6 se muestra la representación visual de la Arquitectura que describe el sistema general.

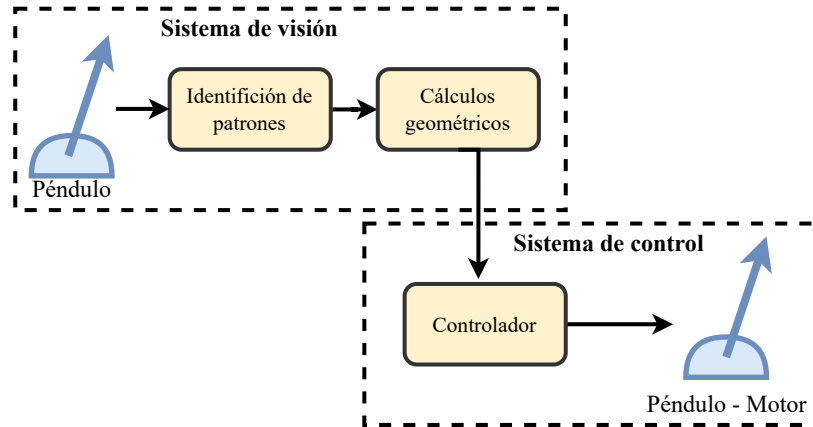


Figura 4.6: Arquitectura planteada

Como se ha mencionado con anterioridad, este sistema es posible dividirlo en 2 subsistemas, la etapa de visión artificial, y la etapa de control. Y esto se logra plasmar en el diagrama anteriormente visto de la Arquitectura planteada.

4.3.8. Desarrollo del sistema de visión artificial

Para la realización de este subsistema de detección de posición y velocidad angular se desarrollaron varios pasos, y, el mismo algoritmo cuenta con varias etapas. Analizando los requisitos, se determinó que la forma más conveniente de hacer la detección era con la ayuda de un patrón, el cual es detectado por el sistema, y posterior a eso, se realizan una serie de cálculos geométricos para determinar el ángulo, luego enviar ese valor a otro algoritmo dentro de este subsistema, para que este determine la velocidad angular, a continuación se describirá cuál es el proceso que sigue dicho subsistema. En la figura 4.7 se aprecia el algoritmo que sigue este sistema para determinar los valores de posición y velocidad.

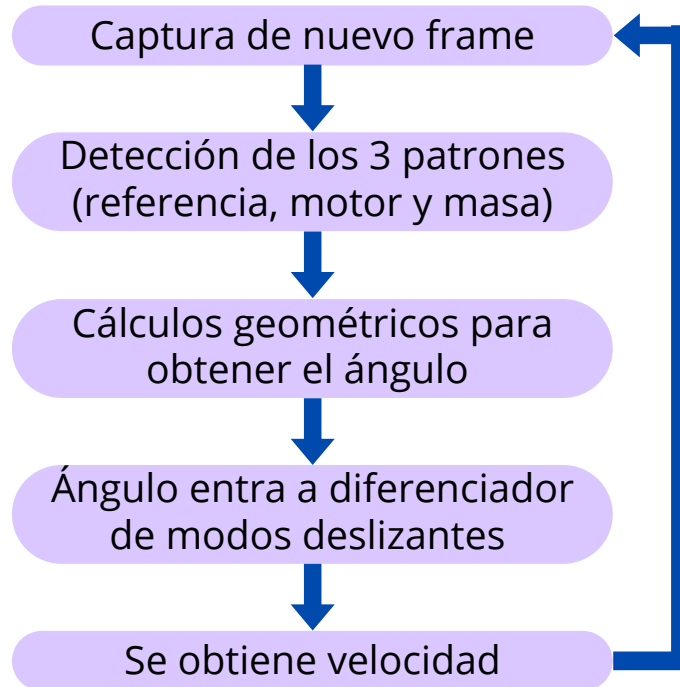


Figura 4.7: Algoritmo seguido por el sistema de visión artificial

4.3.8.1. Detección del ángulo

Se comienza capturando un video con calidad de 1280x720 píxeles, de ese video se va analizando cada uno de los fotogramas en tiempo real. En esos fotogramas se apreciará algo similar a la figura 4.8, que es el péndulo acompañado de esos tres patrones, los cuales son identificados por la función de OpenCV llamada *cornerSubPix*, con la cuál se obtienen las coordenadas de cada patrón, con lo que se tendrían 3 puntos.

Posterior a esto se obtienen dos líneas y pendientes, la primera entre los dos puntos que tienen mismo valor en y (motor y punto de referencia), y la segunda de los dos puntos que tiene valor mayor en x (motor y masa), con esos valores ya se puede obtener el ángulo entre las dos rectas, con la fórmula del ángulo entre dos rectas.

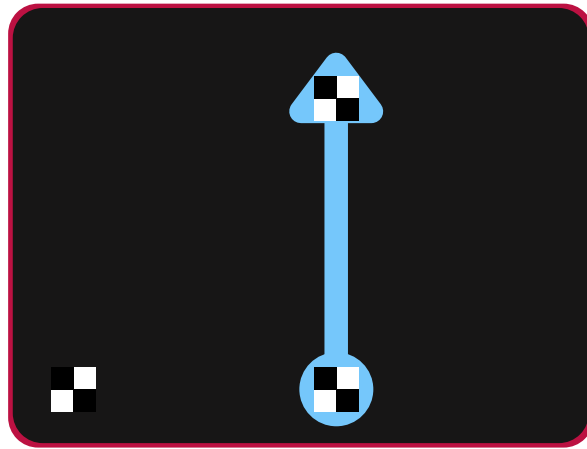


Figura 4.8: Simulación de lo que muestra un fotograma

4.3.8.2. Detección de la velocidad

A partir del ángulo se puede obtener la velocidad, esta se hace con un diferenciador de alta ganancia, en la figura 4.9 se muestra una especie de diagrama de bloques de control, debido a que este es un algoritmo de teoría de control adaptado a esta aplicación. Y posterior a esto se muestra en la ecuación 4.1 el cómo se obtiene $dz2$ que sería la velocidad, esas ecuaciones son lo que está dentro de la programación del diferenciador, y donde se declaran cada uno de los parámetros, tales como la gravedad, masa, longitud de poste, etc.

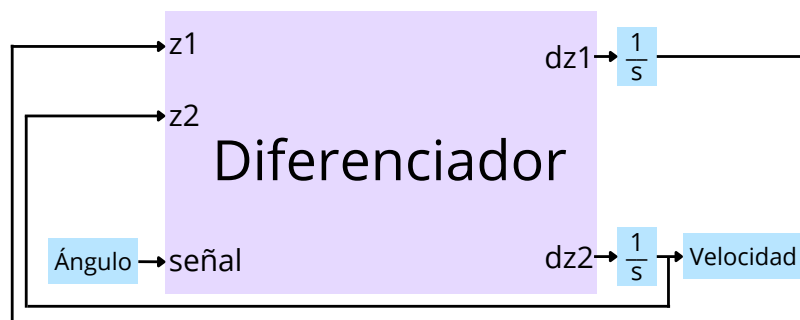


Figura 4.9: Diagrama de bloques del diferenciador

$$\begin{aligned} dz_1 &= z_2 + r * l_1 * e \\ dz_2 &= -\frac{g}{l} * \sin z_1 - \frac{k}{m} * z_2 + \frac{1}{ml^2} * u(t) + r^2 l_2 e \end{aligned} \tag{4.1}$$

4.3.9. Desarrollo del sistema de control

Para la realización de este subsistema de control se implementó un control por modos deslizantes propuesto en el libro “Advanced Sliding Mode Control for Mechanical Systems”, de Jinkun Liu y Xinhua Wang, los cuales hacen el desarrollo de este sistema en MATLAB y añaden una red neuronal para que trabaje con este sistema, este sistema se adaptó a las características del prototipo de péndulo invertido físico que se tenía (Liu & Wang, 2012). Además se modificaron las ecuaciones de la dinámica del sistema a la del péndulo de este proyecto.

La tarea de la red neuronal es para obtener las funciones dinámicas del modelo no lineal, que se representan como funciones $f(x)$ y $g(x)$, con estas funciones se obtiene la señal de control que va al actuador. En la figura 4.10 se puede apreciar la secuencia de pasos que sigue el algoritmo de control.

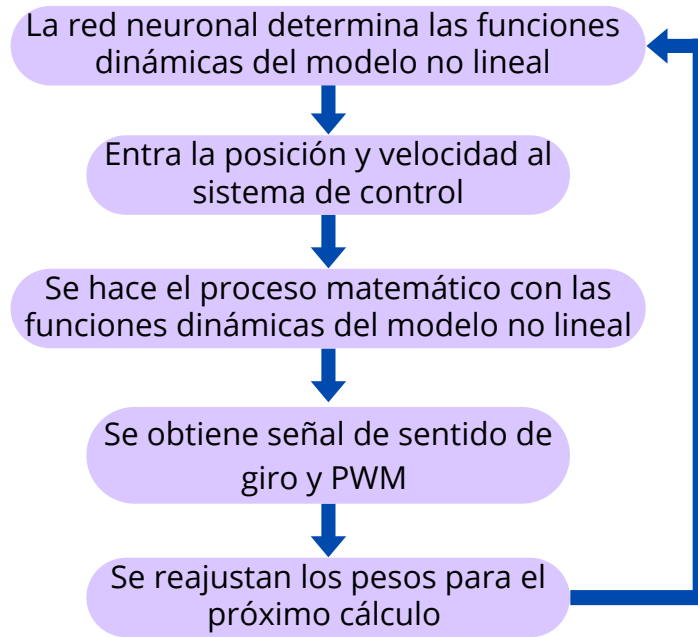


Figura 4.10: Algoritmo seguido por el sistema de control

En la figura 4.11 se puede apreciar el diagrama de bloques del sistema de control apoyado por una red neuronal presentado por Jinkun Liu y Xinhua Wang en su libro, del cuál se basó el desarrollo de este sistema.

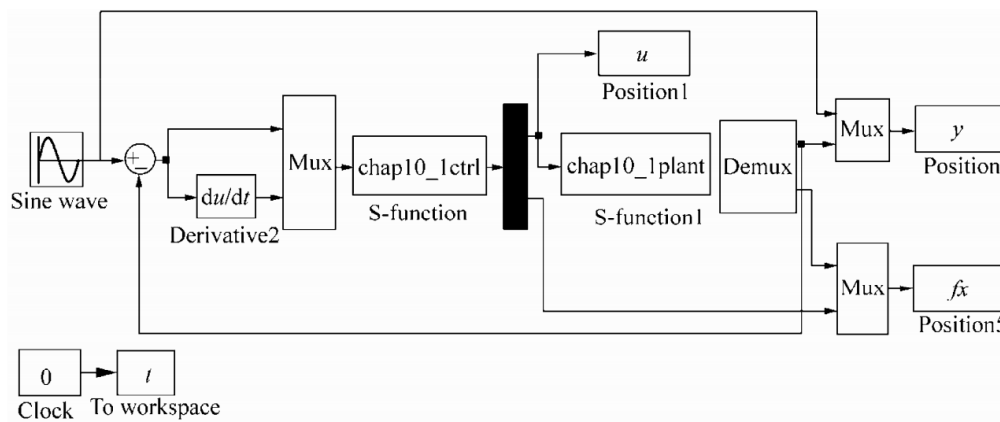


Figura 4.11: Diagrama de bloques del sistema de control por modos deslizantes propuesto por Jinkun Liu y Xinhua Wang (fuente (Liu & Wang, 2012))

En este diagrama el bloque más importante es “chap10_1ctrl”, ya que es el que se refiere al controlador en sí, e internamente es un programa de un controlador por modos deslizantes

convencional, añadiéndole una red neuronal de función de base radial, por lo que tiene un método híbrido de entrenamiento. La estructura de esta red es la propuesta por los mismos autores del sistema de control, y consta de 2 entradas, 5 capas ocultas y una salida, y el valor inicial del peso es de 0.10. En la sección de resultados se habla más a fondo de la prueba que se le hizo a este controlador.

Y la secuencia que se sigue es que la señal de posición y velocidad entran al sistema, y entrega como salida una polaridad a la cuál se moverá el motor, y una señal de PWM, que es un número analógico entre 0 y 255, que determina la velocidad a la que se moverá el motor.

Internamente lo que hace este controlador es que se entrena la red, posterior a esto entran a la misma red los valores de posición y velocidad, y al valor que sale de la red entra al controlador, que lo que hace es ejecutar una ecuación de modos deslizantes, lo que determina la señal de control antes mencionada.

4.4. Estancia de investigación

Del 2 de marzo al 10 de mayo del presente año se realizó una estancia de investigación en el Tecnológico Nacional de México campus Instituto Tecnológico de Chihuahua (TECNM-ITCH), a cargo del Dr. Abraham Efraím Rodríguez Mata, en donde se adaptó el sistema de control y de visión artificial a la planta, y posterior a eso se realizaron las pruebas antes mencionadas, comparando el sistema propuesto con sistemas convencionales, y desarrollando dos artículos de investigación para la revista de La Salle y para el Congreso Nacional de Control Automático.

Capítulo 5

Análisis de Resultados

Para esta sección se presentan los resultados; desde la plataforma física con su sistema de detección con visión artificial, sistema de control con una red neuronal, hasta las comparaciones obtenidas contra los sistemas convencionales, obtenidas en el TECNM-ITCH.

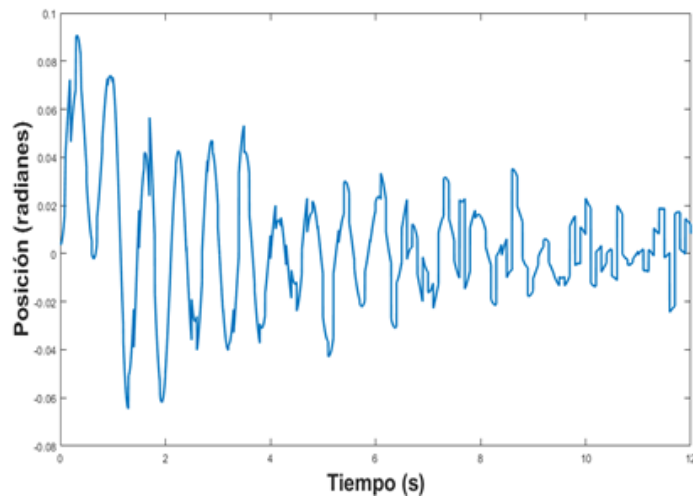
5.1. Resultados del sistema de detección de posición

En esta sección se muestran y analizan los resultados que se obtuvieron mediante el algoritmo de visión artificial para la detección de posición y velocidad angular ejecutado por la NVIDIA Jetson Nano. Como se menciona anteriormente, este algoritmo detecta un patrón, genera dos líneas imaginarias, determina sus pendientes y calcula el ángulo de la intersección de esas dos líneas mediante sus pendientes.

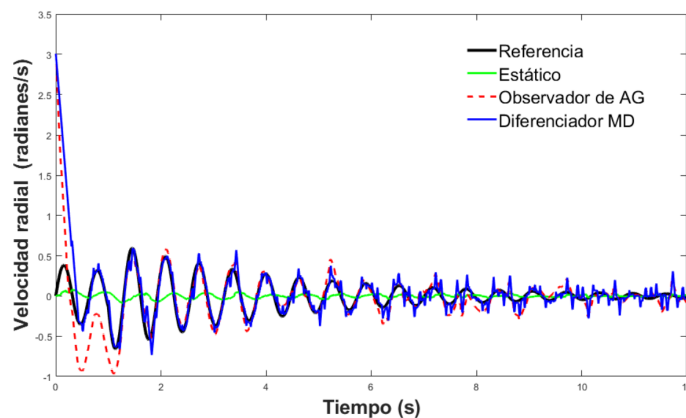
5.1.1. Sistema de visión artificial

El sistema de visión artificial, como se menciona en los capítulos anteriores, consta de una cámara conectada a una NVIDIA Jetson Nano, se genera un video en tiempo real, y en cada cuadro capturado de ese video, se detecta, gracias a la librería OpenCV, un patrón, mediante cálculos geométricos se generan dos líneas imaginarias, se determinan sus pendientes y se obtiene el ángulo entre esas pendientes, así se realiza la obtención del ángulo del péndulo invertido. Para el caso de la velocidad, normalmente se usaría derivada de la posición respecto al tiempo, pero para tener una mejor precisión a altas velocidades, se utiliza un diferenciador-observador de alta ganancia, y con él se obtiene la velocidad.

Los resultados obtenidos a partir de las pruebas de este sistema, indican que si bien, no existe una mejoría respecto a los dispositivos de detección convencionales, ya sea un acelerómetro o giroscopio, si hay un buen acercamiento a ellos, el algoritmo de detección de ángulo y velocidad muestra un acercamiento muy cercano a los datos reales. Se puso a prueba en una simulación el diferenciador-observador de alta ganancia respecto a un derivador común y a la señal simulada, como se muestra en la figura 5.1, obteniendo que el diferenciador-observador de alta ganancia tiene una mejor precisión que un derivador estático común y un observador simple de alta ganancia, siendo este último el que más se acerca a lo obtenido por el diferenciador-observador de alta ganancia.



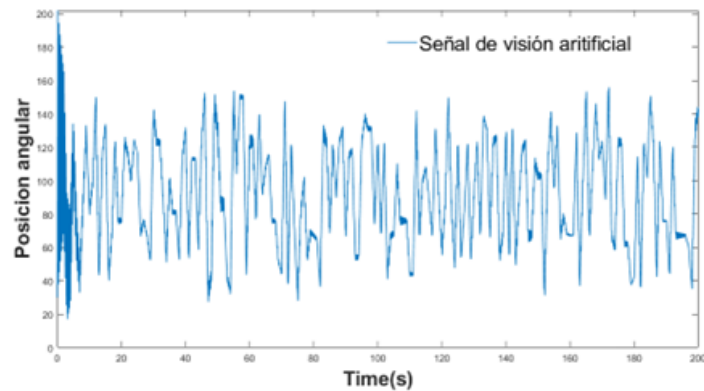
(a) *Señal simulada con ruido Gaussiano*



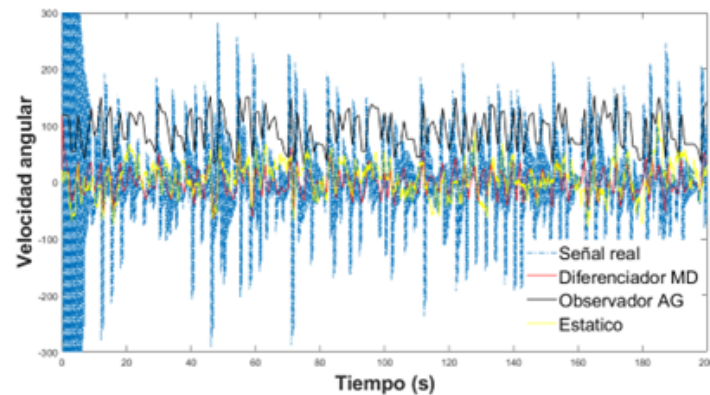
(b) *Comparación de estimadores*

Figura 5.1: Señal simulada de entrada para los estimadores y su comparación de velocidades calculadas por cada uno de ellos

A pesar de ser el mejor en esta comparación, sigue sin ser totalmente equiparable a un sensor o dispositivo convencional de detección de posición y velocidad angular. Posterior a esto, se hizo una prueba con datos reales, y se seguían obteniendo mejores resultados con el diferenciador-observador de alta ganancia que con estimadores más convencionales, como se muestra en la figura 5.2, a pesar de haber muchas perturbaciones causadas por temas físicos como la luz o el movimiento de la cámara, el sistema termina determinando de buena forma la velocidad, siendo este sistema el que destaque por sobre los demás estimadores.



(a) Señal real de posición angular



(b) Comparación de estimadores

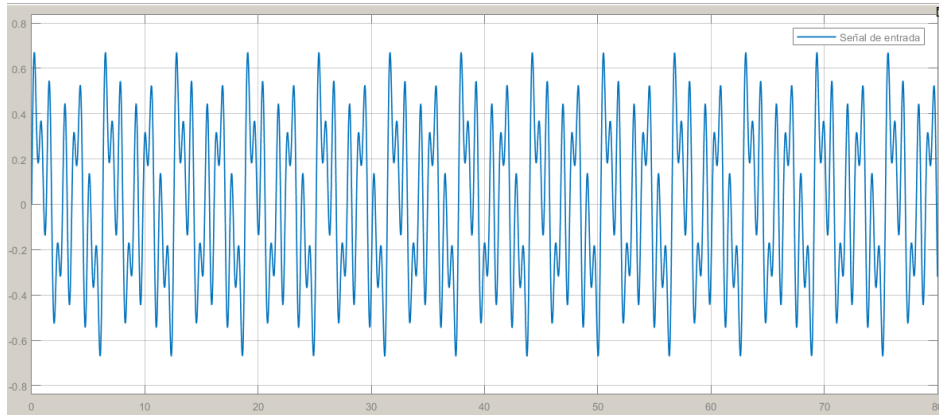
Figura 5.2: Señal de entrada para los estimadores y su comparación de velocidades calculadas por cada uno de ellos

5.2. Resultados del sistema de control de posición

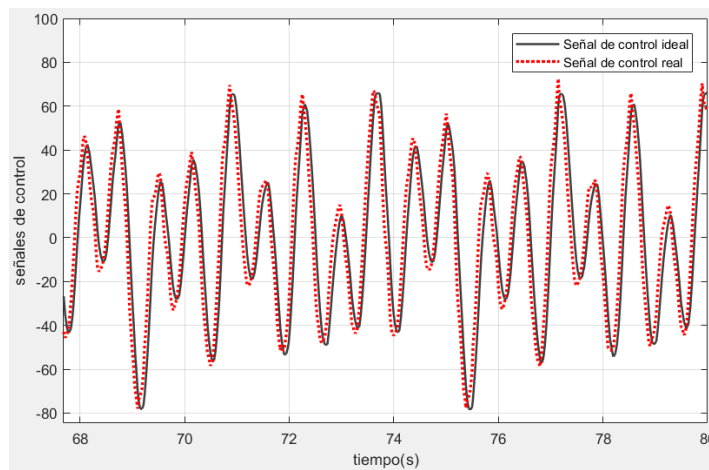
A continuación, se muestran y analizan los resultados obtenidos en el algoritmo de control de posición angular del péndulo invertido ejecutado por la NVIDIA Jetson Nano. Este algoritmo recibe datos del sistema de visión artificial y los convierte en una señal polar PWM, para que el mecanismo ejecute un movimiento.

5.2.1. Sistema de control de posición

Para el caso de la posición angular se utilizó un control por modos deslizantes apoyado por una red neuronal, mejorando en gran medida los resultados obtenidos por los sistemas convencionales de control, el sistema se adapta perfectamente a la señal y predice de manera correcta el siguiente movimiento del mecanismo. En cuando a la comparación de este sistema contra un control convencional PID se obtuvieron buenos resultados, sobre todo en situaciones donde la planta presenta muchas variaciones o perturbaciones, tal podría ser el caso del sistema de estabilización de un dron, que, dependiendo la altitud o el viento de cada momento, varía la velocidad de giro en cada hélice. Para los casos donde el movimiento es más simple, no existen diferencias perceptibles entre un sistema de control y otro. En la figura 5.3 se muestra una comparativa de una simulación en donde el control por modos deslizantes se adapta muy bien a la señal ideal de control a pesar de las múltiples perturbaciones. Es aquí donde se demuestra lo antes dicho de los sistemas donde sería fácilmente aplicable.



(a) Señal de entrada del control



(b) Comparación de señales de salida del controlador (ideal vs real)

Figura 5.3: Prueba de adaptación a perturbaciones del sistema de control

5.3. Análisis de resultados

A continuación se desglosan las pruebas anteriores y se analizan los resultados obtenidos en las mismas, comenzando por las del sistema de visión artificial para la detección de posición y velocidad angular.

5.3.1. Pruebas de visión

En cuanto a las pruebas de la estimación de la posición y velocidad angular del péndulo, como se menciona anteriormente, se hicieron pruebas de simulación y también con resultados reales, en cuanto a la posición, el algoritmo logró una velocidad de entre 1 a 9 milisegundos, siendo lo suficientemente rápido, y comparándose con giroscopios usados comúnmente, que rondan su muestreo al rededor de los 2 a 3 milisegundos, en algunos casos este sistema puede llegar a ser más rápido, pero los sensores siguen teniendo velocidades de muestreo más estables. En el caso de la precisión al medir, son muy similares, el sistema de visión no tuvo casi errores al estimar posición, únicamente en los momentos en que la luz no favorecía a la escena, factor que no afecta a los sensores convencionales.

En el caso de la velocidad, como se menciona anteriormente se probó el diferenciador con una señal con ruido y después con una señal real, comparándolo con un observador de alta ganancia y un derivador estático, obteniendo los mejores resultados el diferenciador por modos deslizantes. La tarea consistía en acercarse lo más posible a la señal de velocidad generada a la que se le añadió ruido blanco, el diferenciador por modos deslizantes después de menos de un segundo logra adaptarse y estimar con buena precisión la velocidad, cerca de este está el observador de alta ganancia, el cuál se adapta después del segundo uno, y hace un buen trabajo en adelante para estimar la velocidad. En el caso del diferenciador estático le cuesta calcular debido al ruido blanco, esas perturbaciones no cuadran en la formula, lo que genera que se obtengan resultados poco útiles.

Al momento de probar este sistema con una señal real se obtienen resultados con conclusiones tanto buenas, como malas. Para el caso de estimar, lo hace bien, pero cuando hay perturbaciones muy grandes, que generan un valor que es el doble o más de la muestra anterior, le cuesta estimar la velocidad con exactitud, a pesar de eso, con valores con un incremento más natural a la dinámica del sistema, se logra adaptar y estimar la velocidad de forma más precisa. El diferenciador por modos deslizantes terminó siendo el que obtuvo mejores resultados, con un observador de alta ganancia y un diferenciador estático que no se pudieron adaptar a los cambios tan bruscos en los valores de posición. Es importante aclarar que esta prueba fue buscando el fallo, generando movimientos muy bruscos y poniendo a prueba la

capacidad de adaptación de los estimadores.

5.3.2. Pruebas de control

En el caso de los sistemas de control, se hizo una simulación en donde la señal que necesitaba el sistema estaba llena de perturbaciones, se generó esa señal combinando valores de senos y cosenos a diferentes ganancias, generando una señal con picos tanto negativos como positivos, que si bien, era cíclica, los cálculos se hacen para cada señal, lo que termina haciendo que el controlador que se pruebe con esta señal tenga difícil el adaptarse.

Como se esperaba, el control por modos deslizantes es muy preciso y de fácil adaptación a perturbaciones, y la red neuronal mejora aún más su efectividad, como se menciona, la señal de entrada tiene muchos picos repentinos, y el controlador se adapta muy bien a esto, generando una señal de control suave y muy fiel a la ideal. Este control es más robusto que un PID, debido a su buena adaptación y rapidez para encontrar la señal de control adecuada. Lo principal en esta prueba es percatarse de lo robusto que es este sistema de control por modos deslizantes apoyado por la red neuronal de función de base radial, y sus aplicaciones tomando en cuenta esto, como antes se mencionó en el área de los vehículos aéreos este controlador podría encontrar un gran campo de exploración.

Capítulo 6

Conclusiones

En este capítulo se describen tres apartados: las conclusiones que se obtienen a partir de esta investigación, las aportaciones que se obtuvieron al desarrollar este proyecto, y el trabajo a futuro para mejorar el proyecto y obtener un mejor producto final, ya sea con el autor actual o uno diferente. Por lo que a continuación se desarrollarán cada uno de estos puntos.

6.1. Conclusiones

La integración de teoría y técnicas de distintas áreas del conocimiento es la tendencia a la que apunta el mundo en los próximos años, y poder materializar en una aplicación esta conjunción corresponde a un gran paso en la dirección correcta. El estancamiento de una industria es la razón de su obsolescencia, en este sistema se integran técnicas de inteligencia artificial para resolver un problema que comúnmente solo se resolvería con métodos electrónicos y teoría de control automático, lo que abre un nuevo abanico de posibilidades para las dos áreas. La inteligencia artificial, en la rama de la visión computacional, es un área muy atractiva por sus múltiples aplicaciones, puntualmente en este caso su uso es importante para evitar el uso de sensores convencionales que necesiten una calibración a periodos cortos, haciendo a esta una alternativa atractiva para no necesitar una constante calibración ni mantenimiento. Gracias a esto, la visión computacional desarrollada en este proyecto es un parte aguas para el desarrollo de algoritmos que puedan competir, superar y sustituir a sensores convencionales.

El sistema de control más convencional y utilizado es el control PID, por su fácil aplicación y versatilidad, pero para ciertas ocasiones termina siendo no lo suficientemente robusto,

debido a que es malo adaptándose a perturbaciones, en el caso de esta investigación se propuso un control por modos deslizantes, que si bien, ya existe desde hace tiempo, se utiliza una técnica en donde se le acopla una red neuronal, esto para predecir el siguiente movimiento del mecanismo a controlar, haciendo a esta combinación más rápida y robusta respecto a los sistemas de control convencionales.

Dentro de un sistema de control, las perturbaciones juegan un papel muy importante, al momento de simular como se comportará un sistema de control, se simulan también ciertas perturbaciones, para ver si este controlador será capaz de llegar al objetivo de manera adecuada. Y, en el caso de algunos sistemas convencionales, las perturbaciones afectan en gran medida a su comportamiento, lo que se consiguió con este sistema es poder adaptarse a las perturbaciones sin tener demasiada pérdida de precisión ni tiempo, objetivo que se ha cumplido.

Con lo anterior mencionado, se llega a la conclusión que la integración de diferentes tecnologías logra encontrarle solución a un problema sin solución, o con solución complicada para un área acotada del conocimiento.

6.2. Aportaciones

Esta investigación consistió en integrar técnicas de inteligencia artificial a sistemas de control para controlar mecanismos, convirtiéndose en un proyecto con múltiples contribuciones. Estas contribuciones son el desarrollo de un sistema de visión artificial para detectar posición angular, un diferenciador-observador de alta ganancia para de terminar la velocidad de un mecanismo a partir de su posición y un sistema de control que integra una red neuronal para predecir el movimiento de un mecanismo. A continuación se desarrollan cada una de estas aportaciones:

- Sistema de visión artificial: Como parte del desarrollo de visión artificial, se logró obtener un algoritmo que pueda realizar mediciones estimadas de la posición angular de un péndulo invertido, algoritmo desarrollado reconociendo un patrón en la imagen mostrada y haciendo una serie de cálculos geométricos.

- **Diferenciador-observador de alta ganancia:** En cuanto a la velocidad angular, se desarrolló un diferenciador-observador de alta ganancia, que usando como entrada las posiciones angulares obtenidas en el sistema de visión artificial, calculaba la velocidad estimada respecto al tiempo, si bien, este algoritmo ya existía, se le hicieron adaptaciones para que pudiera ser aplicable en el proyecto actual.
- **Sistema de control con redes neuronales:** Otra contribución obtenida en esta investigación fue el desarrollo de un sistema de control por modos deslizantes que tuviera integrada una pequeña red neuronal. Al igual que la aportación anterior, este algoritmo ya había sido desarrollado por Hassan K. Khalil, la contribución y diferenciación en este proyecto fue el adaptar este algoritmo a un sistema con diferente comportamiento al que describe él en su texto, además de poder aplicar este mismo algoritmo a otros sistemas.

6.3. Trabajo a futuro

Como trabajo futuro se espera robustecer el sistema en general. Para el caso del sistema de control, hacer pruebas con diferentes redes neuronales y observar su comportamiento en este tipo de situaciones en donde tengan que controlar un mecanismo, además de lograr que el sistema de control sea más preciso y rápido, pero sin sacrificar el funcionamiento y vida útil de los componentes, ya que este tipo de controlador (Control por Modos Deslizantes) tiene a acortar considerablemente la vida útil de los actuadores.

Y, para el caso del sistema de visión artificial, es necesario volverlo más genérico, preciso y de múltiples detecciones. En cuanto a lo genérico, se hace referencia por su aplicación tan específica de detectar un poste y masa de un péndulo invertido, pero lo ideal es detectar ángulos sin tener ningún tipo de referencias en postes, esto haría que, por ejemplo, en un brazo robótico pueda detectar el ángulo de un eslabón desde cualquier punto de visión. En el caso de que sea más preciso, más que hacer referencia a los grados que detecta, es por su aplicación cuando las condiciones no son las apropiadas, cuando hay mucha o poca iluminación, actualmente hay momentos en que no hace las mediciones correctas, entonces sería importante solucionar ese problema para mejorar la precisión en todas las muestras. Y, por último,

las múltiples detecciones podría ser aplicable en un brazo robótico de múltiples eslabones y grados de libertad, esta sería la aplicación más útil en cuanto a lo monetario, ya que solo se necesitaría una cámara y no un sensor por cada articulación, volviéndose una solución más económica. La combinación de estos tres avances lograría tener un sistema de detección y control más robustos.

Bibliografía

Abeyssekera, B. & Wanniarachchi, I. (2018). Modelling and implementation of pid control for balancing of an inverted pendulum (pages 44, 45).

Acheson, D. (1993). A pendulum theorem. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 443(1917), 239-245 (page 13).

ACIMED. (1994). Apuntes sobre los términos computación.ºinformática”. *ACIMED*, 2, 56-57. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94351994000300009&nrm=iso (page 25)

Asad, H., Ravi, V. & Singh, N. (2020). *The Computer Vision Workshop: Develop the skills you need to use computer vision algorithms in your own artificial intelligence projects*. ”Packt Publishing Ltd.” (Page 39).

Britannica, T. E. o. E. (2013). Control system. Encyclopedia Britannica. <https://www.britannica.com/technology/control-system>. (Page 18)

Britannica, T. E. o. E. (2018a). Electric circuit. Encyclopedia Britannica. <https://www.britannica.com/technology/electric-circuit>. (Page 14)

Britannica, T. E. o. E. (2018b). Relay. Encyclopedia Britannica. <https://www.britannica.com/science/relay-electronics>. (Page 15)

- Christ, R. D. & Wernli, R. L. (2014). Chapter 7 - Power and Telemetry. En R. D. Christ & R. L. Wernli (Eds.), *The ROV Manual (Second Edition)* (Second Edition, pp. 141-161). Butterworth-Heinemann. [https://doi.org/https://doi.org/10.1016/B978-0-08-098288-5.00007-5](https://doi.org/10.1016/B978-0-08-098288-5.00007-5). (Pages 15, 17)
- Coello, C. A. C. (2003). *Breve historia de la computación y sus pioneros*. Fondo de cultura económica México. (Page 26).
- Cosenza, M. (2015). Mecánica Clásica. *Universidad de los Andes (ULA). Facultad de Ciencias. Departamento de Física. Mérida-Venezuela (Publicación Electrónica)*, 133-141 (page 10).
- Cox, E. (1994). *The fuzzy systems handbook: a practitioner's guide to building, using, and maintaining fuzzy systems*. Academic Press Professional, Inc. (Page 32).
- Duda, R., Gaschnig, J. & Hart, P. (1981). Model design in the PROSPECTOR consultant system for mineral exploration. *Readings in Artificial Intelligence* (pp. 334-348). Elsevier. (Page 30).
- Edwards, C. & Spurgeon, S. (1998). *Sliding mode control: theory and applications*. Crc Press. (Page 25).
- Electrical4U. (2020). On Off Control Controller: What is it? (Working Principle). <https://www.electrical4u.com/on-off-control-theory-controller/>. (Page 5)
- Garayalde, D. A. & Negri, P. (2014). Visión artificial y Filtro de Kalman aplicados al control de un péndulo invertido. *2014 IEEE Biennial Congress of Argentina (ARGENCON)*, 370-375. <https://doi.org/10.1109/ARGENCON.2014.6868521> (pages 46, 47)
- García, M. & Barreiro, A. (2015). Análisis de la Estabilidad según Lyapunov de un Control Borroso en Tiempo Discreto (page 25).

- Goodfellow, I., Bengio, Y. & Courville, A. (2016). Regularization for deep learning. *Deep learning*, 216-261 (page 33).
- Grasser, F., D'Arrigo, A., Colombi, S. & Rufer, A. (2002). JOE: A mobile, inverted pendulum. *Industrial Electronics, IEEE Transactions on*, 49, 107-114. <https://doi.org/10.1109/41.982254> (page 3)
- Haddad, W. M. & Bernstein, D. S. (1993). Explicit construction of quadratic Lyapunov functions for the small gain, positivity, circle, and Popov theorems and their application to robust stability. Part I: continuous-time theory. *International Journal of Robust and Nonlinear Control*, 3(4), 313-339 (page 25).
- Haykin, S. (1996). Neural networks expand SP's horizons. *IEEE Signal Processing Magazine*, 13(2), 24-49 (page 31).
- Hernández, J. G. V., Giraldo, O. D. M. & Buitrago, D. G. (2013). Lógica Difusa Aplicada al Control Local del Péndulo Invertido con Rueda de Reacción. *Scientia et Technica*, 18(4), 623-632 (pages 48, 49).
- Hewitt, P. G. (2007). *Vibraciones y ondas*. (Page 11).
- Holland, J. H. (1984). Genetic algorithms and adaptation. *Adaptive control of ill-defined systems* (pp. 317-333). Springer. (Page 31).
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558 (page 30).
- Khalil, H. K. (2002). Nonlinear systems third edition. *Patience Hall*, 115 (pages 11, 12).
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1), 59-69 (page 30).

- Kosko, B. (1996). *Fuzzy engineering*. Prentice-Hall, Inc. (Page 32).
- Kosko, B. & Toms, M. (1994). *Fuzzy thinking: The new science of fuzzy logic*. Flamingo London. (Page 32).
- Kouro, S. & Musalem, R. (2002). Control mediante lógica difusa. *Técnicas Modernas en Automática, 1*, 1-7 (pages 47, 48).
- Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4(2), 87-112 (page 31).
- Koza, J. R. et al. (1994). *Genetic programming II: Automatic Discovery of Reusable Programs* (Vol. 17). MIT press Cambridge. (Page 31).
- Krafes, S., Chalh, Z. & Saka, A. (2020). Vision-based control of a flying spherical inverted pendulum on a Quadrotor. *2020(1)*, **jourarticle** 11. <https://doi.org/https://doi.org/10.5339/qproc.2019.imat3e2018.11> (page 46)
- Liu, J. & Wang, X. (2012). *Advanced sliding mode control for mechanical systems*. Springer. (Pages 64, 65).
- MathWorks. (s.f.). MATLAB - El lenguaje del cálculo técnico. <https://la.mathworks.com/products/matlab.html>. (Pages 41, 42)
- McCarthy, J., Minsky, M. L., Rochester, N. & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4), 12-12 (page 26).
- McClelland, J. L., Rumelhart, D. E., Group, P. R. et al. (1987). *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models* (Vol. 2). MIT press. (Page 30).

- Merriam-Webster. (s.f.). Intelligence. <https://www.merriam-webster.com/dictionary/intelligence/>. (Page 26)
- Minsky, M. (1974). A framework for representing knowledge. (Page 29).
- Mitchell, T. (1997). Machine learning (page 32).
- Nasir, A., Ismail, R. & Ahmad, M. (2010). Performance comparison between sliding mode control (SMC) and PD-PID controllers for a nonlinear inverted pendulum system. *World Academy of Science, Engineering and Technology*, 71, 400-405 (pages 49, 50).
- Negnevitsky, M. & Intelligence, A. (2005). A guide to intelligent systems. *Artificial Intelligence* (page 27).
- Negoita, C. V. (1984). *Expert systems and fuzzy systems*. Benjamin-Cummings Publishing Co., Inc. (Page 32).
- NVIDIA. (2021). Jetson Nano Developer Kit. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. (Page 17)
- Ogata, K. & Yang, Y. (2002). *Modern control engineering* (Vol. 4). Prentice hall India. (Pages 5, 19-21).
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in science & engineering*, 9(3), 10-20 (page 43).
- Ooi, R. C. (2003). Balancing a two-wheeled autonomous robot. *University of Western Australia*, 3 (page 12).
- OpenCV. (s.f.). OpenCV: Introduction. <https://docs.opencv.org/4.5.5/d1/dfb/intro.html>. (Page 41)
- Orihuela, R. J. P. (2015). Robot da Vinci: El quirófano del futuro. *Perfiles de Ingeniería*, 11(11) (page 2).

- Orrego, J. J. M. (2007). *Electricidad I. Teoría básica y prácticas* (Vol. 1). Marcombo. (Page 14).
- Patterson, J. & Gibson, A. (2017). *Deep learning: A practitioner's approach*. .°Reilly Media, Inc.” (Pages 26, 35-39).
- Rodríguez-Mata, A., Aurélien Cabarbaye, R., Baray-Arana, R., De Los Ríos, P. A.-C. & Rodríguez-Rangel, H. (2021). Neuronal Sliding Mode Output Control Application in a Attitude for Quadrotor. *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 1-7. <https://doi.org/10.1109/CCE53527.2021.9633073> (page 5)
- Rosenblatt, F. (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms* (inf. téc.). Cornell Aeronautical Lab Inc Buffalo NY. (Page 28).
- Russell, S. & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*, Global Edition. *Pearson* (page 32).
- Scace, R. I. (2020). Electronics. *Encyclopedia Britannica*. <https://www.britannica.com/technology/electronics>. (Page 13)
- Shortliffe, E. (2012). *Computer-based medical consultations: MYCIN* (Vol. 2). Elsevier. (Page 30).
- Slotine, J.-J. E., Li, W. et al. (1991). *Applied nonlinear control* (Vol. 199). Prentice hall Englewood Cliffs, NJ. (Page 25).
- Solving, P. (s.f.). A Case Study Using the DENDRAL Program,” *Machine In* (page 29).
- Tang, J. & Ren, G. (2009). Modeling and simulation of a flexible inverted pendulum system. *Tsinghua Science and Technology*, 14(S2), 22-26 (page 13).

- Tilbury, D., Messner, B., MathWorks, Messner, P., Hill, R. & Taylor, J. (2017). Control Tutorials for MATLAB and Simulink - Inverted Pendulum: System Modeling. <https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>. (Page 12)
- Vidyasagar, M. (1993). Nonlinear Systems Theory. (Page 25).
- Wang, X. (2012). *Advanced sliding mode control for mechanical systems: design, analysis and MATLAB simulation*. Springer Berlin Heidelberg. (Page 25).
- Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353. [https://doi.org/https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/https://doi.org/10.1016/S0019-9958(65)90241-X) (pages 29, 31)
- Zadeh, L. A. (1996). Fuzzy logic, neural networks, and soft computing. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh* (pp. 775-782). World Scientific. (Page 32).