

Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Programa de Licenciatura en Ingeniería Electrónica



**Diseño e implementación de un sistema que simule el
comportamiento dinámico de una planta prototipo de control
automático utilizando redes neuronales artificiales**

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

Jorge Andrés Brenes Alfaro

Borrador de 18 de septiembre de 2022

Declaro que los resultados obtenidos en el presente trabajo de investigación, previo a la obtención del título de Licenciado en Ingeniería en Electrónica, son absolutamente originales, auténticos y personales.

Soy consciente de que el hecho de no respetar los derechos de autor y realizar una mala conducta científica; es decir, fabricación de datos falsos y plagio, conlleva sanciones universitarias y/o legales.

En tal virtud, declaro que el trabajo de investigación realizado sujeto a evaluación no ha sido presentado anteriormente para obtener algún grado académico o título, ni ha sido publicado en sitio alguno y los efectos legales y académicos que se puedan derivar del trabajo propuesto de investigación son y serán de mi sola y exclusiva responsabilidad legal y académica.

Jorge Andrés Brenes Alfaro

Cartago, 18 de septiembre de 2022

Céd: 3-0514-0729

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Trabajo Final de Graduación
Acta de Aprobación

Defensa de Trabajo Final de Graduación
Requisito para optar por el título de Ingeniero en Electrónica
Grado Académico de Licenciatura

El Tribunal Evaluador aprueba la defensa del trabajo final de graduación denominado *Diseño e implementación de un sistema que simule el comportamiento dinámico de una planta prototipo de control automático utilizando redes neuronales artificiales*, realizado por el señor Jorge Andrés Brenes Alfaro y, hace constar que cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal Evaluador

M.Sc. José Miguel Barboza Retana
Profesor Lector

Dr. Adolfo Chaves Jiménez
Profesor Lector

Dr. José Pablo Alvarado Moya
Profesor Asesor

Cartago, 18 de septiembre de 2022

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Trabajo Final de Graduación
Tribunal Evaluador
Acta de Evaluación

Defensa del Trabajo Final de Graduación
Requisito para optar por el título de Ingeniero en Electrónica
Grado Académico de Licenciatura

Estudiante: **Jorge Andrés Brenes Alfaro** Carné: 2017101058

Nombre del proyecto: *Diseño e implementación de un sistema que simule el comportamiento dinámico de una planta prototipo de control automático utilizando redes neuronales artificiales*

Los miembros de este Tribunal hacen constar que este proyecto de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica y es merecedor de la siguiente calificación:

Nota de Trabajo Final de Graduación: _____

Miembros del Tribunal Evaluador

M.Sc. José Miguel Barboza Retana
Profesor Lector

Dr. Adolfo Chaves Jiménez
Profesor Lector

Dr. José Pablo Alvarado Moya
Profesor Asesor

Cartago, 18 de septiembre de 2022

Resumen

Palabras clave: Redes neuronales Artificiales, sistemas dinámicos no lineales, NVIDIA Jetson TX2.

Abstract

Same content as the Spanish version, just in English. Check [this site](#) for some help with the translation.

Keywords: Neural networks, non lineal dynamical systems, NVIDIA Jetson TX2.

A mi querida familia por su apoyo incondicional

Agradecimientos

El resultado de este trabajo no hubiese sido posible sin el apoyo de ...

Jorge Andrés Brenes Alfaro

Cartago, 18 de septiembre de 2022

Índice general

Índice de figuras	II
Índice de tablas	III
Revisar	IV
1. Introducción	1
1.1. Objetivos y estructura del documento	8
2. Marco teórico	9
2.1. Péndulo amortiguado a hélice (PAMH)	9
2.1.1. Descripción de la planta	10
2.1.2. Modelo matemático	11
2.1.3. Kit CY8CKIT-059 PSoC 5 LP	12
2.1.4. Controlador de velocidad electrónico (ESC)	13
2.2. Nvidia Jetson TX2	13
2.3. Protocolo de comunicación UART	15
2.4. Redes neuronales artificiales	16
2.4.1. Redes Neuronales Recurrentes	16
2.5. Métricas de evaluación	16
2.5.1. Matriz de confusión	16
3. Red neuronal artificial mimetizadora (RNAM)	17
3.1. Conexión Jetson TX2 y PAHM	17
3.1.1. Implementación de hardware	17
3.1.2. Implementación de software	17
3.2. Planta sintética	17
3.2.1. Modelo empírico	17
3.3. Red neuronal artificial	17
4. Resultados y análisis	18
5. Conclusiones	19
Bibliografía	20

Índice de figuras

1.1. Proceso de diseño para el control de un sistema.	2
1.2. Control directo con redes neuronales artificiales.	4
1.3. Interacción del aprendizaje reforzado.	5
1.4. Diagrama de etapas involucradas en la solución.	6
1.5. Diagrama de control para la PAHM.	6
1.6. Diagrama de etapas involucradas en la red neuronal mimetizadora.	7
2.1. Péndulo Simple.	10
2.2. Péndulo amortiguado a hélice.	11
2.3. Kit CY8CKIT-059 PSoC 5LP	13
2.4. Kit de desarrollo NVIDIA Jetson TX2.	14
2.5. Conexión transmisor-receptor del protocolo UART.	15
2.6. Trama de datos del protocolo UART.	16

Índice de tablas

2.1. Especificaciones generales del kit CY8CKIT-059 PSoC 5 LP.	12
2.2. Especificaciones generales del kit de desarrollo NVIDIA Jetson TX2	14

Revisar

¿Aquí las palabras en inglés también van en itálica?	v
Poner el tamaño de la broca.....	10
¿Puedo referenciar al instructivo? ¿Es informal?	10
Cambiar la imagen de la planta.....	11

¿Aquí las
palabras en
inglés
también van
en itálica?

Capítulo 1

Introducción

Actualmente, el control automático de plantas y procesos tiene una cuota de mercado creciente en la economía global, ya que gran parte de los procesos industriales y de manufactura moderna se encuentran automatizados [1]. En las fábricas modernas e instalaciones industriales se hace cada vez más necesario disponer de sistemas de control que permitan mejorar y optimizar una gran cantidad de procesos, lo que ha repercutido en múltiples beneficios como la reducción de costos, del consumo de energía y tiempo, y el incremento en la calidad y volumen de producción. Además, el logro de una vida con mayores comodidades, aligerar la carga de operaciones manuales repetitivas y rutinarias, la eliminación de errores y aumento en la seguridad de los procesos [2], [3].

Esta área ha demostrado sus múltiples aplicaciones convirtiéndose en parte importante en sistemas de vehículos motrices, espaciales y robóticos, en procesos modernos de fabricación y en cualquier operación industrial que requiera control de temperatura, presión, humedad y flujo [3]. Lo que conlleva a promover la tecnología en áreas como domótica, procesos químicos, ingeniería mecánica, automovilismo, aeronáutica, espacial, entre otras [2]; al igual que el desarrollo y aplicación de técnicas modernas de control, impulsando el desarrollo en avances científicos y tecnológicos.

El control automático ha jugado un papel vital en el avance de la ingeniería y la ciencia, donde los sistemas de estudio son dinámicos y el conocimiento de la teoría de control proporciona una base para entender el comportamiento de dichos sistemas [2]. En los métodos convencionales de control aplicados, cada etapa requiere de intervención manual en la que por medio de experimentación deben ajustarse parámetros, cuya validez depende de qué tan precisos son los modelos determinados previamente. En la figura 1.1 se muestran las etapas para el diseño convencional de control, donde los pasos cinco, seis y siete son procesos iterativos que se pueden optimizar de forma que se reduzca la configuración manual.

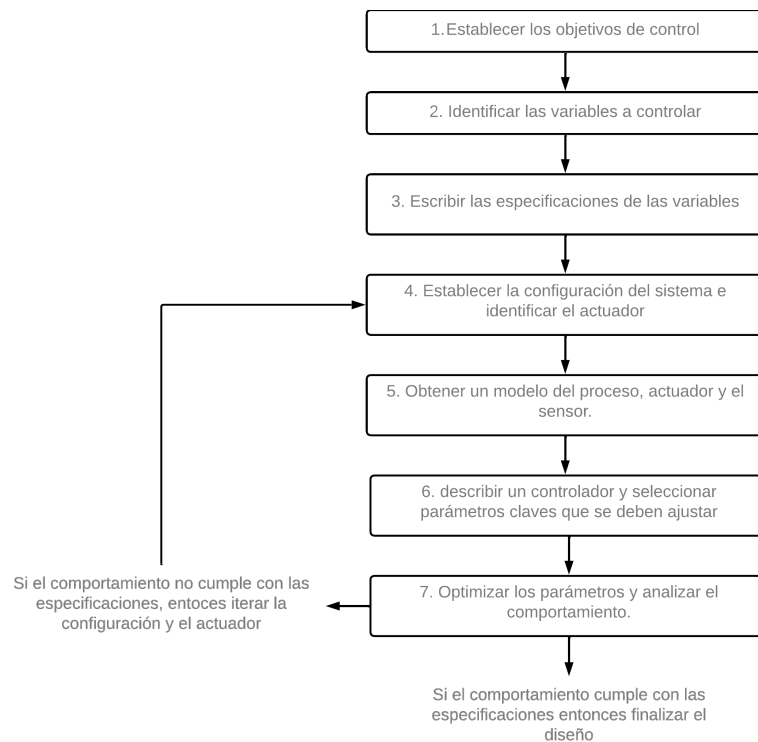


Figura 1.1: Proceso de diseño para el control de un sistema. Fuente: [4].

En sistemas de control, el proceso a controlar requiere de una representación que caracterice el proceso físico; es decir, un modelo matemático dependiente del tiempo que represente su comportamiento. Tal como se observa en el paso cinco de la figura 1.1. Esta representación se puede obtener de forma teórica, lo que se denomina modelado; así como también de forma empírica, mediante experimentos sobre el sistema real, basándose en datos de entrada y salida del proceso, llamado identificación. La ventaja de ello es que permite predecir y estudiar el comportamiento dinámico y estático de los sistemas, además, sin exponer el proceso a daños, se puede determinar que puede ocurrir simulando el modelo con una condición, entrada inusual o perturbación [5].

Sin embargo, debido a la presencia de no linealidades y más de una entrada y salida en las plantas modernas de control, se vuelve más complejo la descripción del sistema, requiriendo una mayor cantidad de ecuaciones haciendo que sea cada vez más difícil derivar modelos a mano que logren un equilibrio aceptable entre precisión y velocidad [3], [6], y a pesar de existir una gran cantidad de métodos clásicos, varios de ellos no conducen a resultados aceptables, por lo tanto, puede ser difícil encontrar modelos matemáticos representativos [7]. La mayoría de los sistemas dinámicos que se encuentran en la naturaleza son no lineales y recientemente, hay sistemas en aplicaciones modernas de ingeniería que presentan dificultades en la ejecución de control, tales como sistemas de comando de vuelo, manipuladores de robot, sistemas de inyección de combustible de alto rendimiento, entre otros [8]. Por lo que se vuelve de importancia algún método o técnica para identificar el sistema, lo que ha llevado al desarrollo de campos que se ocupan de crear

modelos directamente a partir de datos, como la identificación de sistemas, el aprendizaje automático (ML) y el aprendizaje profundo (DL) [6].

El control clásico que trata sistemas con una entrada y una salida pierde su potencialidad cuando se trabaja con sistemas de múltiples entradas y salidas [3]. A pesar de esto, se puede controlar mientras la planta sea linealizable en puntos de operación específicos, ya que para controlar un sistema se debe elegir un punto de operación; no obstante, para aquellos sistemas donde existen puntos de operación variantes en el tiempo y no linealidades en su modelo, la capacidad de controlar eficientemente el sistema se complica [9], lo cual es usual en los sistemas no lineales. Por ello, surge el control moderno, el cual se ha desarrollado para manejar la creciente complejidad de las plantas modernas y los requisitos cada vez más exigentes. Este se basa en el análisis del dominio temporal de los sistemas de ecuaciones diferenciales, simplificando el diseño de los sistemas de control porque se basa en un modelo del sistema real que se quiere controlar. A pesar de presentar alternativas para el control de sistemas no lineales, la estabilidad del sistema depende del error entre el sistema real y su modelo; es decir, cuando el controlador diseñado se aplica al sistema real, éste puede no ser estable [3].

Hoy en día, la industria y aplicaciones en ingeniería tiene dificultades al controlar sistemas dinámicos complejos, pues tienen no linealidades y perturbaciones impredecibles que afectan el desempeño de los controladores [10]. La realización de modelos y controladores requieren experimentación, cuya exhaustividad y resultados están con frecuencia afectados por la no linealidad de los sistemas, lo que conlleva a procesos iterativos que consumen tiempo, exploración y refinamiento de controles lineales para zonas de operación específicas. En consecuencia, surge la necesidad de diseñar e implementar estrategias innovadoras de control automático para los procesos no lineales y controlar el sistema en todo su espacio de operación sin la necesidad de un modelo matemático linealizado en torno de un punto de equilibrio, minimizando la configuración manual y la intervención humana.

En la figura 1.1 se observa que los pasos los pasos cinco, seis y siete referentes a la obtención de un modelo matemático, realizar el controlador y optimizar los parámetros de este, pueden ser optimizados de forma que se reduzca la configuración manual. Por ello, se exploran técnicas recientes involucradas con el aprendizaje automático, ya que utilizan redes neuronales cuyo proceso es iterativo y requieren de mínima intervención humana. Por otro lado, la Escuela de Electrónica ha acumulado experiencia en el diseño de sistemas de control con las metodologías tradicionales, pero en años recientes con el crecimiento del aprendizaje automático han surgido alternativas que aún no han sido exploradas en la escuela.

En años recientes, ha habido un incremento en la aplicación de redes neuronales artificiales (RNA), ya que constituyen una herramienta para el aprendizaje de relaciones complejas a partir de un conjunto de ejemplos. Además, debido a sus capacidades de aproximación, su adaptabilidad y su tolerancia a fallos, las RNAs presentan una alternativa en el modelado e identificación de sistemas no lineales [11], [12]. Son modelos paramétricos que basándose

en una medida del error y en un algoritmo para minimizarlo, aproximan una función no lineal con errores pequeños, además, son capaces de resolver problemas cuya solución por otros métodos convencionales resultan difíciles o insatisfactorios [7], [12].

Por ello, han surgido propuestas del uso de estas mismas para la identificación del comportamiento de un sistema dinámico no lineal desconocido, donde a partir de la salida ante una entrada dada, la red es capaz de modificar sus parámetros aprendiendo una aproximación de la dinámica del sistema hasta conseguir un modelo fiable [13], [14]. De igual forma, las RNAs ofrecen alternativas para el control de sistemas dinámicos no lineales que permite mejorar el desempeño de un sistema en el mundo real en tareas de seguimiento sin la necesidad de un modelo matemático linealizado en un punto de equilibrio. Dichos controladores se pueden aplicar de forma directa al sistema como se observa en la figura 1.2. En este caso la señal con la que la red neuronal aprende corresponde a la diferencia entre la señal de control idónea y la señal calculada por la red neuronal, sin embargo, la señal de control idónea no se conoce y, por tanto, el error no se puede calcular. Donde una solución a ello son los algoritmos de aprendizaje reforzado, que permiten el aprendizaje con señales cualitativas del error en vez de cuantitativas [12].

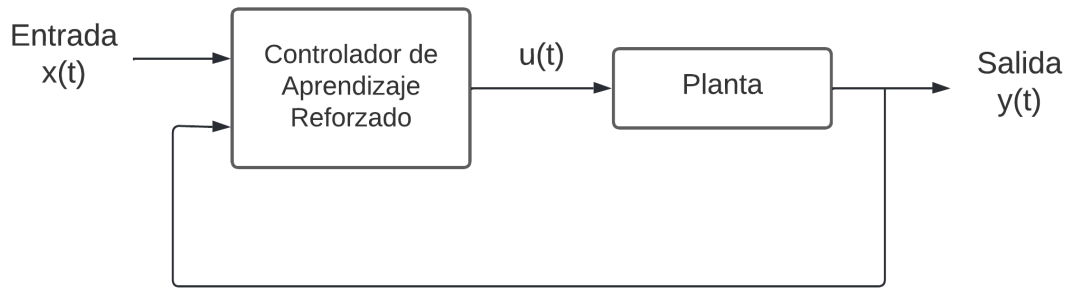


Figura 1.2: Control directo con redes neuronales artificiales. Fuente: [12].

El aprendizaje reforzado (RL), que ha mostrado en los últimos años avances en sistemas como AlphaGo y AlphaFold [15], ofrece algoritmos para buscar y desarrollar controladores óptimos de sistemas con dinámicas no lineales, estocásticas y que son desconocidas o inciertas [16]. Esta metodología busca que un agente sea capaz de encontrar la acción correcta de manera autónoma, explorando un espacio desconocido y determinando la acción mediante prueba y error. Dicho agente aprende por medio de recompensas y penalizaciones que obtiene de sus acciones, con el fin de actuar y crear la mejor estrategia posible, de forma que maximice la recompensa [15], [17]. Esta interacción es representada en la figura 1.3.

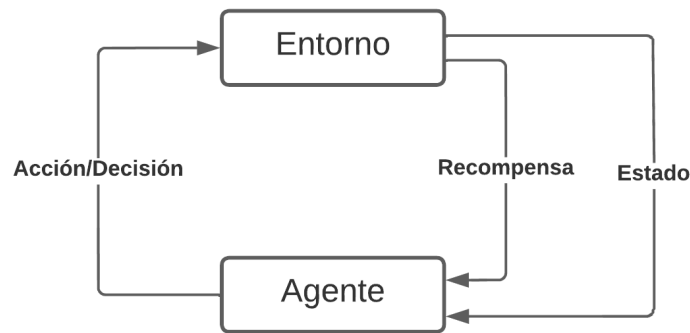


Figura 1.3: Interacción del aprendizaje reforzado.

Este enfoque no solo se ha utilizado para resolver juegos de Atari, sino que ya existen propuestas en sistemas de control, donde el control actual no es 100 % efectivo [15]. Esta metodología ajusta el control ante cambios en el entorno del sistema al aprender con las acciones, permitiendo automatizar el proceso más allá de lo alcanzable con los métodos tradicionales de control.

Por lo tanto, con lo mencionado anteriormente, se propone hacer uso de dos redes neuronales artificiales, una que mimetice el comportamiento del sistema de estudio y otra que lleve a cabo el control del sistema por medio de aprendizaje reforzado. El uso de estas dos RNAs aprovechan la capacidad de aproximación y adaptabilidad, además de experimentar con diversas entradas sin afectar directamente la planta, evitando el riesgo de que los experimentos sean tan violentos en los cambios que se dañe la planta. Usando la red neuronal artificial mimetizadora (RNAM) se puede usar un reloj más rápido que el tiempo real, acelerando los experimentos y eliminando la desventaja de tener que limitar la ejecución al tiempo real de la planta. Por ello, el volumen de datos disponible para el entrenamiento del modelo de aprendizaje reforzado es mayor con la red que modela al sistema real, que el disponible usando la planta en físico. Por último, con la ayuda de las RNA se prescinde de cálculos complejos, así como del planteamiento del modelo que se necesitaría para el control clásico, pues las RNA se adaptan a cualquier tipo de estructura sin necesidad de conocer sus características [13].

La solución se representa con un diagrama de etapas en la figura 1.4. Primeramente, se recopila datos de la planta física para entrenar la red mimetizadora, cuya planta a utilizar es un péndulo amortiguado a hélice (PAHM). Seguido de ello, se implementa y entrena el controlador, el cual ajusta el control ante cambios del entorno del sistema al aprender con acciones. Por último, el controlador se prueba con la red mimetizadora y posteriormente con la planta física.

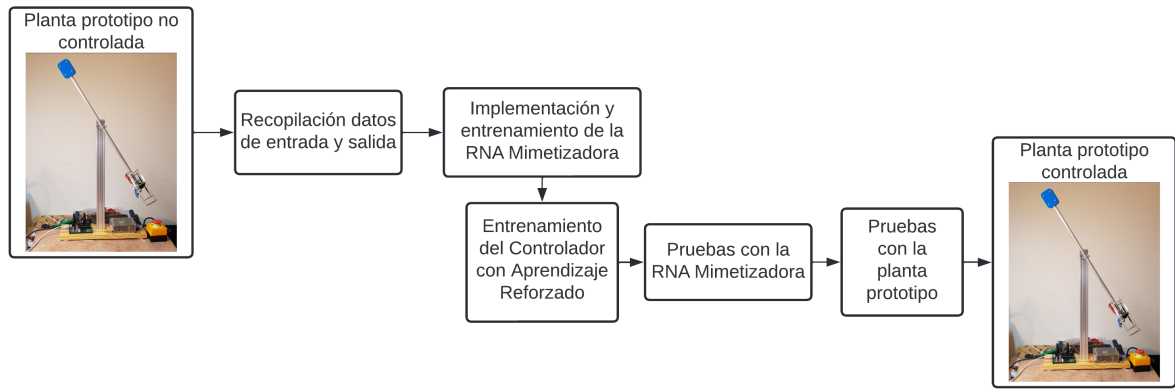


Figura 1.4: Diagrama de etapas involucradas en la solución.

Al conocer las etapas se define un diagrama de control y sus bloques para la PAHM, el cual se observa en la figura 1.5. En dicho diagrama se visualizan 4 bloques: un filtro con el fin de limpiar la señal y eliminar ruido que se pueda adherir, los bloques de las RNAs a utilizar tanto del controlador como de la red mimética, el bloque de la planta cuya observabilidad está dada por el sensor incorporado para medir el ángulo del péndulo, partiendo de que la incertidumbre del sensor es mínima.

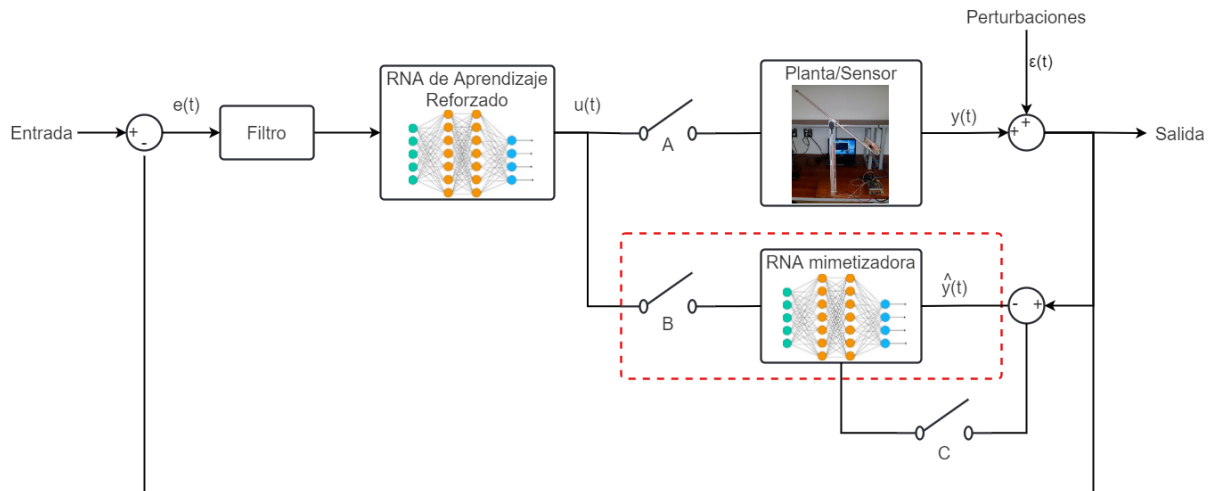


Figura 1.5: Diagrama de control para la PAHM.

También se observa el uso de 3 interruptores: A, B y C. Estos interruptores se cierran o abren de acuerdo a la etapa en la que se encuentre. Para la recopilación de datos y el desarrollo y entrenamiento de la RNA mimetizadora los tres interruptores se encuentran abiertos. Finalizada la RNA mimetizadora se procede a cerrar el interruptor C. La salida $y(t)$ y $\hat{y}(t)$ deben ser aproximadamente iguales, no obstante, la RNA mimetizadora no ve las perturbaciones $\varepsilon(t)$ que tiene la planta, por lo que el objetivo del interruptor C es que la RNA mimetizadora sea capaz de reajustarse. Seguidamente, se abre el interruptor C y se cierra B para entrenar el controlador y realizar las pruebas iniciales de forma segura con la RNA mimetizadora y evaluar la respuesta de este mismo. Por último, se abren los

interruptores B y C y se cierra A para llevar a cabo el control de la planta PAHM física y ver la eficiencia que tiene el controlador desempeñando su función.

Sin embargo, el presente documento está centrado en el desarrollo de la red neuronal mimetizadora, marcada con un cuadro rojo en la figura 1.5, la cual es necesaria para desarrollar e implementar en un futuro el controlador de aprendizaje reforzado. Para ello, se plantea el diagrama de solución de la figura 1.6. Para esta red mimetizadora se recopilará datos de entrada y salida de la PAHM física y de una planta sintética realizada en base a un modelo matemático clásico. Con los conjuntos de datos recopilados se lleva a cabo el entrenamiento y e implementación de la red neuronal mimetizadora. Seguidamente, se aplica un conjunto de datos de prueba de una entrada desconocida con el fin de corroborar si la red neuronal mimetizadora con datos provenientes de la PAHM resulta más óptima que la estimación de un modelo matemático clásico, y de igual forma, determinar qué tan preciso es conforme a la respuesta dada por la PAHM.

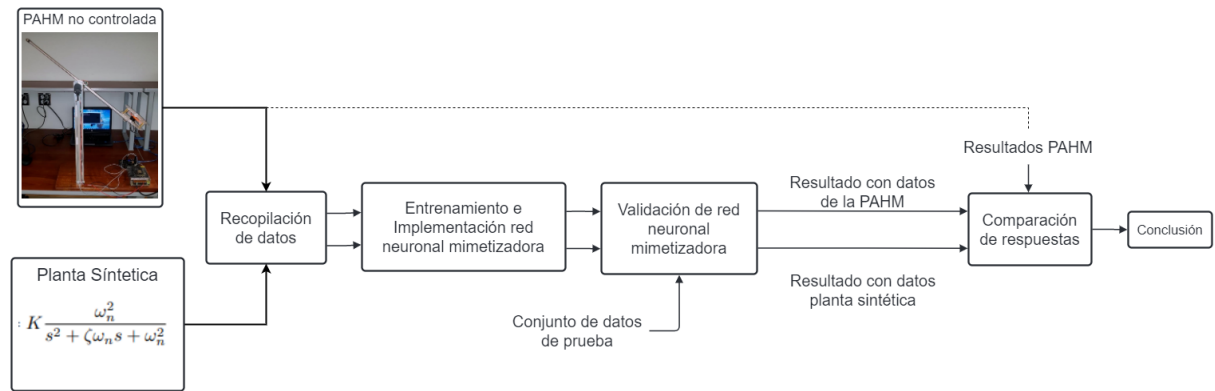


Figura 1.6: Diagrama de etapas involucradas en la red neuronal mimetizadora.

1.1. Objetivos y estructura del documento

El objetivo principal de este proyecto es desarrollar un sistema que mimetice el comportamiento dinámico de la planta prototipo de péndulo amortiguado a hélice (PAHM) mediante el uso de redes neuronales artificiales. Asimismo, esta investigación se basa en cuatro objetivos específicos, donde el primero consiste en acoplar el sistema embebido de altas prestaciones al lazo de control. Como segundo objetivo específico se tiene el preparar un conjunto de datos para el entrenamiento de la red neuronal artificial mimetizadora de la planta real, particionado en datos de entrenamiento, validación y prueba. El tercer objetivo específico consiste en implementar una red neuronal artificial que mimetice el comportamiento del sujeto de estudio. Y el último corresponde a evaluar la respuesta de la red neuronal artificial mimetizadora utilizando una planta sintética y la planta prototipo real como referencia.

Este documento se constituye de los siguientes apartados: en el capítulo 2 se describe en detalle el marco teórico, donde se esbozan los fundamentos teóricos necesarios para explicar la propuesta realizada. En el capítulo 3 se detalla la estrategia desarrollada y el modelo implementado para la solución del problema. En el capítulo 3.3 se lleva a cabo el análisis de los resultados obtenidos. Por último, el capítulo 4 se presenta las conclusiones de la investigación y trabajo realizado, así como recomendaciones y trabajo a futuro por desarrollar.

Capítulo 2

Marco teórico

En este capítulo se presentan los conceptos teóricos a conocer para el entendimiento de la propuesta de diseño e implementación de un sistema que simule el comportamiento dinámico de una planta prototipo de control automático utilizando redes neuronales artificiales. La información expuesta a continuación se deriva tanto de conocimientos propios como de información bibliográfica.

2.1. Péndulo amortiguado a hélice (PAMH)

Un péndulo simple es aquel sistema que consiste en una masa adherida a una barra de longitud fija que se encuentra suspendida en un punto de equilibrio. Esta masa se desplaza a una posición radial θ y se deja caer, generando un movimiento oscilatorio [18]. Dicho sistema se puede observar en la figura 2.1. Los péndulos poseen una riqueza de comportamientos; además, son buenos resonadores y adecuados para el estudio de períodos largos de estabilidad [19]. No obstante, su control se complica al haber cambios o perturbaciones, ya que moverlos de su punto de equilibrio es sencillo, por lo que se necesita controladores robustos para operar en regiones estables [18].

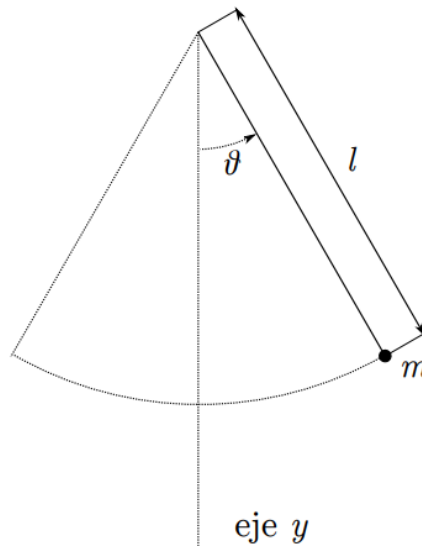


Figura 2.1: Péndulo Simple. Fuente: [20].

El péndulo amortiguado a hélice (PAHM) exhibe un comportamiento idéntico al péndulo simple, con la diferencia de que la masa es una hélice accionada por un motor en corriente directa, la cual desestabiliza el sistema al mover la masa de su punto de equilibrio y cuyo objetivo de dicha planta es realizar un determinado control para estabilizar el sistema en un nuevo punto requerido.

2.1.1. Descripción de la planta

La planta PAHM consta de un soporte rígido de aluminio extruído colocado perpendicularmente sobre una placa de madera. Sobre la parte superior de este soporte se sitúa, montado en rodamientos, el eje de pivote del péndulo. En adición, tiene un sensor de cuadratura óptico incremental, USDigital E5 de 3600 CPR (cuentas por revolución) acoplado al eje, el cual sirve para medir el ángulo. El péndulo mismo está constituido por un tubo hueco de aluminio que al final está unido a un motor sin escobillas, BLDC, Hobbymate 2204, gobernado por un controlador de velocidad, ESC, OneShot125 BLHeli de 15A. El motor tiene adosada una hélice GEMFAN 5045BNR de tres aspas, protegida por rejillas cortadas en acrílico y como contrapeso se hace uso de una broca de (**media**) pulgada [21]. Además, cuenta con una fuente de poder que alimenta el motor y el sensor, también utiliza un sistema embebido PSoC 5 LP y una placa para la conexión de sensores. Este sistema se muestra en la figura 2.2.

Poner el tamaño de la broca

¿Puedo referenciar al instructivo?
¿Es informal?

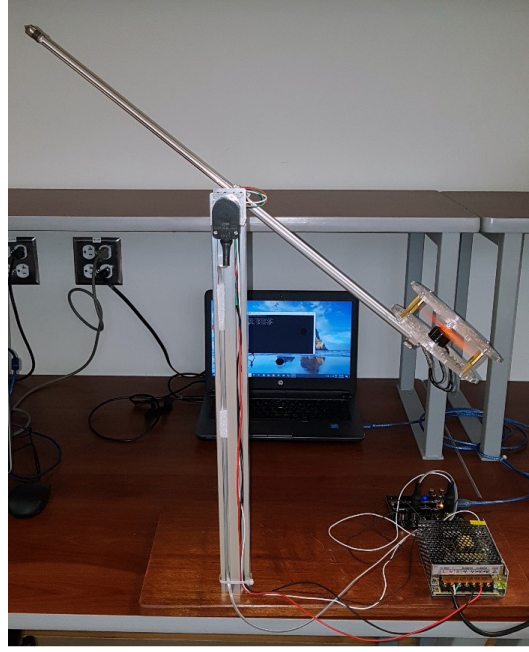


Figura 2.2: Péndulo amortiguado a hélice. Fuente: [21].

Además de ello, la PAHM exhibe las características de ser continua, estable y no lineal. Además, es un sistema SISO (*Single Input, Single Output*), cuya entrada es el valor del PWM (modulación por ancho de pulso) que controla el motor de la hélice a través de un controlador de velocidad electrónico (ESC) que incorpora el protocolo *PWM OneShot125*. Su salida corresponde al ángulo del péndulo medido por el sensor de cuadratura. La PAHM posee solo un grado de libertad, por lo se requiere de solo una ecuación para describir su movimiento y esta no requiere de intervención humana para reiniciarse.

Cambiar la
imagen de la
planta

2.1.2. Modelo matemático

$$\frac{Y(s)}{X(s)} = \frac{K * \omega_n}{s^2 + s + \zeta \omega_n s + \omega_n^2} \quad (2.1)$$

$$\dot{x}(t) = Ax(t) + Bu(t) \quad y(t) = Cx(t) + Dx(t) \quad (2.2)$$

$$A = \begin{bmatrix} 0 & 1 \\ \omega_n^2 & \zeta \omega_n \end{bmatrix} \quad B = \frac{0}{K \omega_n} \quad C = [0 \quad 1] \quad D = [0] \quad (2.3)$$

2.1.3. Kit CY8CKIT-059 PSoC 5 LP

La comunicación con el sensor y el motor de la PAHM se da mediante un microcontrolador, el cual es el kit CY8CKIT-059 PSoC 5 LP. El PSoC es un acrónimo de *programmable system on chip*, el cual es perteneciente a una familia de microcontroladores desarrollados por la empresa *Cypress*, este cuenta con módulos tanto analógicos como digitales de alta precisión, programables y reconfigurables con enrutamiento automático flexible, obteniendo un componente electrónico con gran flexibilidad en su estructura y que permite una gran facilidad para el desarrollo de sistemas embebidos a un costo muy bajo [22], [23]. Esta placa posee un conector micro-USB para permitir el desarrollo de aplicaciones USB, condensadores CMOD integrados para permitir el desarrollo de *CapSense*, condensadores de derivación para garantizar conversiones ADC de alta calidad y condensadores de carga para conectar un oscilador de cristal externo de 32 kHz [22].

Así mismo, en la tabla 2.1 se resumen una serie de especificaciones que posee el kit del PSoC 5 LP.

Tabla 2.1: Especificaciones generales del kit CY8CKIT-059 PSoC 5 LP. Fuente: [23].

Especificaciones generales	
Voltaje de Operación	3.3 V a 5 V
Rendimiento	Controlador DMA. Procesador de filtro digital (DFB). Procesador de 32-bit ARM Cortex M3.
Almacenamiento	Memoria flash: 256 kB. Memoria RAM: 64 KB. EEPROM: 2kB.
Interfaces de Comunicación	UART, I2C, SPI, I2S
Frecuencia	DC a 80 MHz.
Temperatura	-40 °C a 85 °C
Puertos	USB, micro USB, J1 y J2 headers
Fabricante	Cypress Semiconductor

Además, el kit incluye un KitProg que permite la funcionalidad de programación, depuración y conexión en puente sobre USB e interfaces USB, UART e I^2C . Se utiliza como programa de desarrollo un software denominado *PSoC Creator*, el cual presenta varias cualidades como tener una interfaz grafica donde se visualizan los bloques tanto digitales y análogos que posee el microcontrolador y cuales de ellos se han utilizado en la aplicación que se desarrolla [22]. También permite escribir el código de programación para la aplicación en dos lenguajes de programación diferentes los cuales son el lenguaje C y el *Assembler*.

El microcontrolador PSoC 5LP se puede observar en la figura 2.3, donde se observa la tarjeta principal y el kitprog.

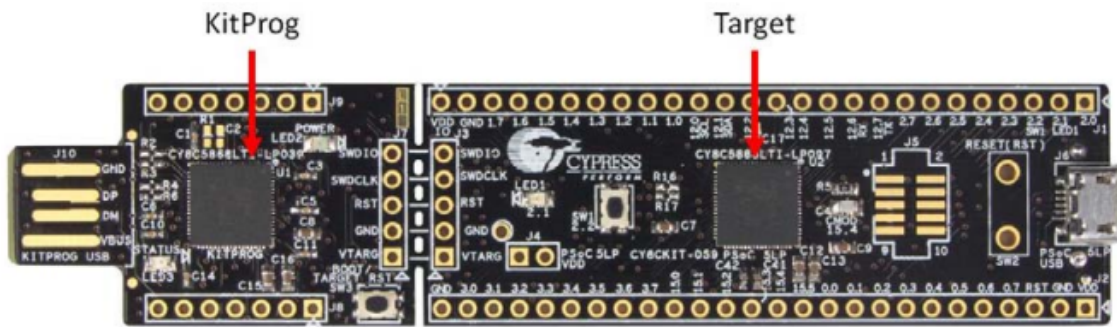


Figura 2.3: Kit CY8CKIT-059 PSoC 5LP. Fuente: [22].

2.1.4. Controlador de velocidad electrónico (ESC)

2.2. Nvidia Jetson TX2

Debido a la exigencia computacional que se requiere se opta por utilizar un kit de desarrollo NVIDIA Jetson TX2. Este sistema embebido de altas prestaciones se encarga de la implementación, entrenamientos y predicciones de las redes neuronales. Además, se encarga de la lectura de los datos y proporciona la alimentación a la planta.

Dentro del kit de desarrollo se encuentra NVIDIA Jetson TX2, la cual es una supercomputadora integrada en un módulo con alta capacidad de procesamiento, brindando una forma rápida para el desarrollo de software y hardware para aplicaciones de alto rendimiento computacional a un bajo consumo energético [24]. El módulo posee un procesador Tegra de NVIDIA que integra una unidad de procesamiento central (CPU) de arquitectura ARM y una unidad de procesamiento gráfico (GPU) de arquitectura NVIDIA Pascal, diseñado para maximizar el rendimiento y eficiencia energética para cargas de trabajo con necesidades de cálculo muy exigentes. Su sistema operativo es Linux4Tegra, basado en Ubuntu 18.04, el cual está disponible a través de la imagen de la tarjeta SD incluida, que está diseñada para ejecutar hardware NVIDIA. Por otro lado, el kit también cuenta con conectores PCIe, SATA, HDMI, USB y micro-USB, Ethernet, expansiones con pines de entrada-salida de propósito general (GPIO), cámara, entre otros que se pueden observar en [25]-[27]. Este kit de desarrollo se puede observar en la figura 2.4.

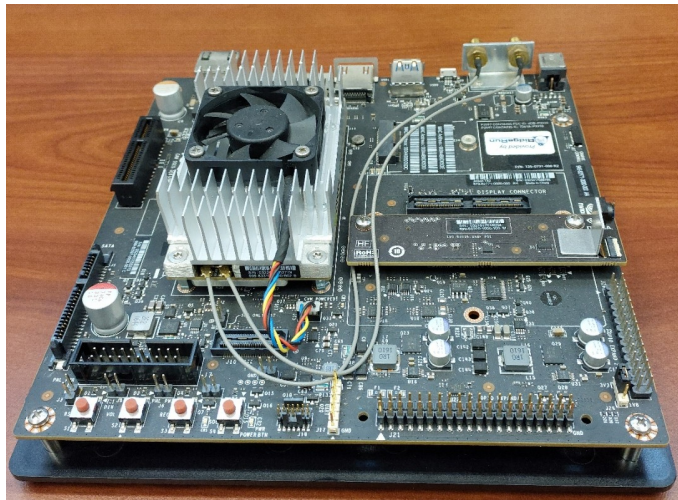


Figura 2.4: Kit de desarrollo NVIDIA Jetson TX2.

Debido a su capacidad computacional la Jetson TX2 se usa para aplicaciones de inteligencia artificial, visión por computadora y aprendizaje automático, siendo una solución a proyectos que requieren involucrar hardware y software. Ejemplo de aplicaciones donde se utiliza dicho embebido es en robots autónomos, drones, dispositivos médicos portátiles, vehículos inteligentes, ente otros.

Así mismo, en la tabla 2.2 se resumen una serie de especificaciones que posee el kit de desarrollo NVIDIA Jetson TX2.

Tabla 2.2: Especificaciones generales del kit de desarrollo NVIDIA Jetson TX2. Fuente: [27].

Especificaciones generales	
CPU	Dual-core Denver 2 64 bit + Quad core ARM Cortex A57
GPU	256 NVIDIA CUDA cores
Memoria	4 GB 128-bit LPDDR4
Almacenamiento	16 GB eMMC 5.1
Red	10/100/1000 BASE-T Ethernet
Video	Decode 2x 4K60 Video Encode 1x 4K60
Camera	12 lanes MIPI CSI-2, D-PHY 1.2 (30 Gbps)
Interfaces periféricas	1x USB 3.0, 3x USB 2.0, 2x PCIe 3x UART, 2x SPI, 4x I2C, 1x CAN, 4 x I2S, GPIOs, 1x SD CARD/SDIO

2.3. Protocolo de comunicación UART

El transmisor-receptor asíncrono universal o UART (por sus siglas en inglés) es un protocolo de comunicación encargado de controlar dos puertos seriales, uno que actúa como receptor y el otro como transmisor, donde se da el intercambio de datos en serie entre dos dispositivos. Dicho protocolo es simple dado que solo utiliza 2 hilos de comunicación, un transmisor y un receptor, donde la comunicación puede ser *simplex* (Los datos se envía en una sola dirección), *half-duplex* (se comunican uno a la vez) o *full duplex* (ambos transmiten y reciben al mismo tiempo) [28]. Esto se puede observar en la figura 2.5.

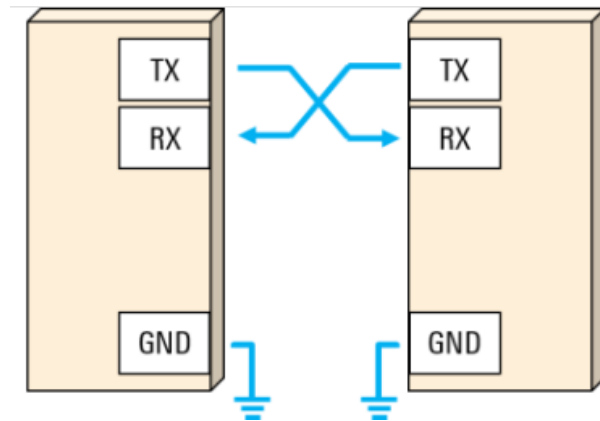


Figura 2.5: Conexión transmisor-receptor del protocolo UART. Fuente: [28], [29]

El transmisor (Tx) se encarga de empaquetar y enviar datos a través del puerto serial, este toma un dato en paralelo de un tamaño establecido y lo empaqueta según el protocolo y lo envía bit a bit a la velocidad estipulada. Por el otro lado, el receptor (Rx) funciona de manera opuesta, se encarga de tomar un dato a través del puerto serial, desempaquetarlo y almacenarlo de manera paralela. La unidad de almacenamiento es en forma de cola, donde el primer dato en entrar es el primer dato en salir (FIFO, por sus siglas en inglés). Este tiene como objetivo evitar un desfase entre la información que se envía y la que se recibe.

El protocolo UART posee parámetros para la transmisión de información, debe definirse el largo de la palabra, es decir, la cantidad de bits que constituye la palabra transmitida, cuyo valor se encuentra entre 0 a 8 bits. El otro parámetro a definir es el *baudrate*, esta variable determina la velocidad con que se transmite los datos a través del sistema, donde los más frecuentes son 9600 bit/s, 19200 bit/s, 38400 bit/s, 57600 bit/s, 115200 bit/s [29]. También se debe definir si existe o no control de flujo ya sea por *software* o *hardware*.

Además, el protocolo también establece como realizar el empaquetamiento de los datos, donde la trama de bits está compuesta por un bit de parada, cuyo valor es un cero lógico. Seguido de ello, viene la palabra de ocho bits, le sigue un bit de paridad, el cual es opcional y su valor depende si se quiere paridad par o impar. Por último, un bit de parada, cuyo valor es un uno lógico. Esta secuencia se puede observar en la figura 2.6

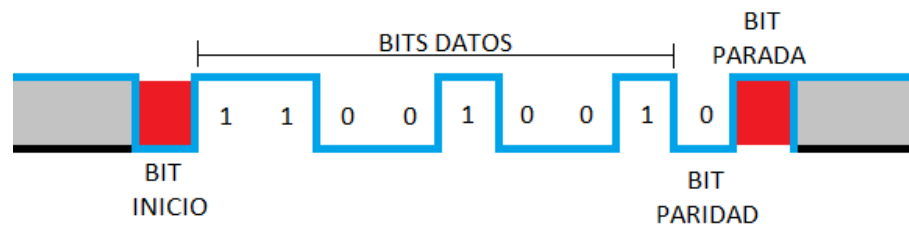


Figura 2.6: Trama de datos del protocolo UART. Fuente: [28], [29]

Otro aspecto a considerar es que si no se está produciendo una transmisión de datos la línea de salida debe estar en nivel alto, denotando que existe comunicación entre dispositivos, pero no existe trasiego de datos [28], [29]. Y de esta manera, el bit de inicio le indica al receptor que una transmisión se ha iniciado.

2.4. Redes neuronales artificiales

2.4.1. Redes Neuronales Recurrentes

Gated Recurrent Units (GRU)

2.5. Métricas de evaluación

2.5.1. Matriz de confusión

Capítulo 3

Red neuronal artificial mimetizadora (RNAM)

3.1. Conexión Jetson TX2 y PAHM

3.1.1. Implementación de hardware

3.1.2. Implementación de software

3.2. Planta sintética

3.2.1. Modelo empírico

3.3. Red neuronal artificial

Capítulo 4

Resultados y análisis

Capítulo 5

Conclusiones

Bibliografía

- [1] Á. Almidón Elescano y E. Julian-Laime, *Sistemas de Control Automático I- Teoría y problemas aplicativos*. 2018. DOI: [10.5281/zenodo.2560185](https://doi.org/10.5281/zenodo.2560185).
- [2] M. Perez, A. Perez Hidalgo y E. Perez Berenguer, *Introducción a los sistemas de control y modelo matemático para sistemas lineales invariantes en el tiempo*, Argentina, 2008.
- [3] K. Ogata, *Ingeniería de control moderna*, 5.^a ed. Madrid: Pearson Educación, 2010.
- [4] W. Corvacho Cárdenas, «Diseño e implementación de un sistema de inmersión temporal automatizado para cultivos de arándano y piña en los laboratorios de Nutriarandanos S.A.C.» Lima, Perú, 2019.
- [5] J. González Godoy, «Modelado de sistemas dinámicos usando redes neuronales recurrentes profundas,» Tesis doct., Centro de investigación y estudios avanzados del instituto politécnico nacional, México, 2019.
- [6] C. Møldrup Legaard, T. Schranz, G. Schweiger y col., «Constructing neural network-based models for simulating dynamical systems,» *ACM Computing Surveys*, vol. 1, n.º 1, 2021.
- [7] J. Acosta, J. Fernández y L. Becerra, *Aplicación de las redes neuronales para la identificación de un sistema no lineal. Un caso práctico*, 2000.
- [8] M. M. Serón y J. H. Braslavsky, «Sistemas no lineales,» *Departamento de Electrónica, Universidad Nacional de Rosario, Primer Cuatrimestre*, 2000.
- [9] J. Gómez Martínez y G. Mendoza Avedaño, *Aplicación del control adaptativo a procesos industriales tipo SISO*, Bucaramanga, Colombia, 2009.
- [10] E. Sarmiento Jurado, «Diseño e implementación de un controlador basado en redes neuronales con entrenamiento rápido para sistemas de control 2x2,» Tesis de mtría., Universidad del Norte, Barranquilla, Colombia, 2006.
- [11] R. Valverde Gil y D. Gachet Páez, «Identificación de sistemas dinámicos utilizando redes neuronales,» *Revista Iberoamericana de Automática e Informática Industrial*, n.º 1697-7912, 2007.
- [12] B. Morcego Seix, «Estudio de redes neuronales modulares para el modelado de sistemas dinámicos no lineales,» Tesis doct., Universitat Politècnica de Catalunya, Barcelona, España, 2000.

- [13] S. Torrubia Caravaca, «Redes neuronales multimodelo aplicadas al control de sistema,» Tesis de mtría., Universidad Autónoma de Barcelona, España, 2010.
- [14] J. F. Hernández Pérez, «Redes neuronales dinámicas para la identificación y control adaptable para sistemas no lineales,» Tesis de mtría., Universidad Autónoma del Estado de Hidalgo, Pachuca, México, 2018.
- [15] A. Díaz Latorre, «Aprendizaje por refuerzo para control de sistemas dinámicos,» Tesis de mtría., Universidad Autónoma de Occidente, Santiago de Cali, Colombia, 2019.
- [16] L. Busoniu, T. de Bruin, D. Tolic, J. Kober e I. Palunko, «Reinforcement learning for control: performance, stability, and deep approximators,» *Annual Reviews in Control*, vol. 46, 2018, ISSN: 1367-5788.
- [17] J. Torres, en *Introducción al aprendizaje por refuerzo profundo. Teoría y práctica en Python*. Publicación Independiente, 2021, Capítulo 1.
- [18] F. Moreno, J. Becerra, Y. Ayala y R. Alarcón, «Desarrollo de un módulo didáctico para control angular de un péndulo suspendido,» Cúcuta, Colombia, inf. téc., 2012.
- [19] A. Kasruddin, «Modeling and controller design for an inverted pendulum system,» Tesis de mtría., Malaysia, 2007.
- [20] V. Gómez, *Práctica 2: Péndulo Simple. Determinación de la aceleración de la gravedad*, Valladolid, España, 2017.
- [21] A. Ruiz y E. Interiano, *Proyecto corto: Identificación y control del péndulo amortiguado a hélice (PAMH)*, Cartago, Costa Rica, 2022.
- [22] C. Semiconductor, *PSoC® 5LP prototyping kit guide*, San José, California, 2016.
- [23] —, *PSoC® 5LP: CY8C58LP family datasheet*, San José, California, 2016.
- [24] NVIDIA, *Harness AI at the edge with the Jetson TX2 Developer kit*, Agosto de 2022. dirección: <https://developer.nvidia.com/embedded/jetson-tx2-developer-kit>.
- [25] —, *NVIDIA Jetson TX1/TX2 Developer Kit Carrier Board*, Santa clara, California, 2017.
- [26] —, *Jetson TX2 Developer Kit*, Santa clara, California, 2020.
- [27] —, *Datasheet NVIDIA Jetson TX2 NX System-on-Module*, Santa clara, California.
- [28] B. Díaz Mulas, «UART: Universal Asynchronous Receiver-Transmitter,» B.S. tesis, 2015.
- [29] C. Álvarez Hernández, «Implementación de un ambiente de verificación UART mediante método UVM,» Lic. Tesis, Tecnológico de Costa Rica, Costa Rica, 2018.