



Universidad Nacional
de Educación a Distancia

Escuela Técnica Superior de
Ingeniería Informática



Universidad
Complutense de Madrid

Facultad de Informática

Control Inteligente del ángulo de palas de un aerogenerador mediante aprendizaje por refuerzo

Luis Inazio Cavero Juste

Directora: Matilde Santos Peñas

Co-director: Jesús Enrique Sierra-García

Trabajo de Fin de Máster

Máster Universitario
en Ingeniería de Sistemas y Control

Junio 2024



Máster en Ingeniería de Sistemas y Control.

**Control Inteligente del ángulo de palas de un
aerogenerador mediante aprendizaje por refuerzo**

Proyecto Tipo A

Luis Inazio Cavero Juste

Directora: Matilde Santos Peñas

Co-director: Jesús Enrique Sierra-García



Calificaciones



Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

A handwritten signature in black ink, consisting of several overlapping loops and a final flourish that extends upwards and to the right.

Firma del alumno



Resumen

La energía eólica se presenta como una fuente renovable clave para satisfacer la creciente demanda energética global. Sin embargo, la variabilidad inherente del viento plantea desafíos significativos para mantener una generación de energía estable y prevenir sobrecargas eléctricas y mecánicas en los aerogeneradores. El control eficiente de la turbina, y en concreto el control del ángulo de las palas es fundamental para optimizar la captura de energía eólica y garantizar la integridad estructural de las turbinas.

En este trabajo, se explora la aplicación del aprendizaje por refuerzo (RL) para desarrollar un controlador inteligente del ángulo de palas en aerogeneradores. El objetivo es ajustar el ángulo de alineación de las palas, en respuesta a las variaciones en la velocidad del viento, para permitir mantener una velocidad del rotor constante y, en consecuencia, una potencia constante. Se ha utilizado el algoritmo Deep Deterministic Policy Gradient (DDPG) para entrenar el controlador.

El controlador propuesto se validó mediante simulaciones en un modelo computacional de una turbina eólica pequeña y sus resultados se compararon con los de un controlador Proporcional-Integral-Derivativo (PID) convencional. Los resultados demuestran que el controlador basado en RL es capaz de generar ajustes precisos en el ángulo de palas para mantener una velocidad de rotor constante frente a las fluctuaciones del viento. Además, el controlador RL superó el rendimiento del controlador PID, lo que subraya el potencial del aprendizaje por refuerzo para el control de aerogeneradores, ya que no requiere un conocimiento detallado del modelo de la turbina y se adapta eficazmente a las no linealidades del sistema.

Palabras clave: Aprendizaje por refuerzo, Control inteligente, Turbina eólica, Ángulo de pitch, Deep Deterministic Policy Gradient.





Abstract

Wind energy is presented as a key renewable source to satisfy the growing global energy demand. However, the inherent variability of the wind creates significant challenges to maintaining stable energy generation while preventing electric and mechanical overloads in wind turbines. Efficient control of the Blade pitch angle is fundamental to optimize wind energy capture and ensuring the structural integrity of the turbines.

In this work, the application of reinforcement learning (RL) to develop an intelligent controller of the pitch angle in wind turbines is explored. The objective is to adjust the alignment angle of the blades, in response to variations in wind speed, to allow maintaining a constant rotor speed, and consequently a constant power output. The Deep Deterministic Policy Gradient (DDPG) algorithm was implemented to train the controller, designing various reward functions to optimize its performance.

The proposed controller was validated by means of simulations in a computational model of a small wind turbine and its results were compared with those of a conventional Proportional-Integral-Derivative (PID) controller. The results demonstrate that the RL-based controller is capable of generating precise adjustments in the blade angle to maintain a constant rotor speed subject to wind fluctuations. Furthermore, the RL controller outperformed the performance of the PID controller, what underlines the potential of reinforcement learning for the control of wind turbines, since it does not require detailed knowledge of the turbine model and adapts effectively to the non-linearities of the system.

Keywords: Reinforcement Learning, Intelligent Control, Wind Turbine, Pitch Angle, Deep Deterministic Policy Gradient.





Índice general

Abstract.....	6
Índice general.....	8
Índice de figuras	11
Índice de tablas.....	13
Capítulo 1 Introducción.....	15
1.1. Motivación.....	15
1.2. Propuesta y objetivos	16
1.3. Estructura del documento.....	17
1.4. Publicaciones derivadas	18
1.5. Repositorios de código y recursos desarrollados	18
Capítulo 2 Fundamentos y Estado del Arte	19
2.1. Fundamentos de las turbinas eólicas	19
2.2. Control de aerogeneradores	20
2.3. Aprendizaje automático	22
2.3.1. Redes neuronales	23
2.3.2. Aprendizaje por refuerzo	25
Capítulo 3 Modelo de aerogenerador y arquitectura de control	28
3.1. Modelo del aerogenerador.....	28
3.1.1. Implementación del modelo de aerogenerador.....	30
3.1.2. Validación del modelo	31
3.2. Arquitectura de control	34
3.2.1. Implementación del entrenamiento de aprendizaje por refuerzo.....	36
Capítulo 4 Control del aerogenerador mediante aprendizaje por refuerzo	38
4.1. Controlador 1. RL-Sin Referencia	39
4.2. Controlador 2. RL-Básico	42
4.3. Controlador 3. RL-Recompensa Integral	46
4.4. Controlador 4. RL- Integral.....	49



4.5. Comparativa de los controladores	52
Capítulo 5 Conclusiones y trabajos futuros	53
Bibliografía	54





Índice de figuras

Figura 1.1. Consumo de energía mundial hasta 2023 en Mtoe (Mega Tonelada equivalente de petróleo), por regiones del mundo y tipo de energía. (Enerdata, 2024)	15
Figura 1.2. Reparto energético de la electricidad en España en 2023. Fuente: (International Energy Agency, 2024).....	16
Figura 2.1. Molinos de viento. Autor: Jaime González. Fuente: Wikimedia Commons	19
Figura 2.2. Partes de un aerogenerador de eje horizontal.....	20
Figura 2.3. Modos de operación del control del aerogenerador.	21
Figura 2.4. Clasificación de métodos de aprendizaje automático (Mathworks, 2023).....	23
Figura 2.5. Neurona artificial	23
Figura 2.6: Funciones de activación.....	24
Figura 2.7: Representación de una red neuronal multicapa.....	24
Figura 2.8: Representación de los elementos básicos en Aprendizaje por Refuerzo	25
Figura 2.9: Clasificación de algoritmos de aprendizaje por refuerzo (OpenAI, 2020).	26
Figura 3.1: Entorno de Turbina Eólica con interfaz de Open AI Gym.....	31
Figura 3.2: Simulación de validación del modelo de aerogenerador.....	32
Figura 3.3: Respuesta al escalón de pitch de referencia de 10°	33
Figura 3.4: Respuesta al escalón de pitch de referencia de 40°	33
Figura 3.5: Simulación de escalera de pitch. Comparación de modelos	34
Figura 3.6: Arquitectura de control de ángulo de palas.....	35
Figura 3.7: División de arquitectura de control para entrenamiento	37
Figura 4.1: Arquitectura de control del Controlador 1.	39
Figura 4.2: Simulación de validación Controlador 1	40
Figura 4.2: Arquitectura de control del Controlador 2.	42
Figura 4.3: Métricas del entrenamiento Controlador 2.....	43
Figura 4.4: Simulación de validación Controlador 2.....	44
Figura 4.5: Histograma de exploración del error. Controlador 2.	45
Figura 4.6: Viento en las dos simulaciones de validación.....	47
Figura 4.7: Resultados simulación escalones de viento. Controlador 3	47
Figura 4.8: Resultados simulación viento sinusoidal. Controlador 4	48
Figura 4.9: Arquitectura Controlador 4	49
Figura 4.10: Evolución del RMSE Controlador 4	50
Figura 4.11: Simulación con escalones de viento Controlador 4	51
Figura 4.12: Simulación con viento sinusoidal Controlador 4	51





Índice de tablas

Tabla 3.1. Parámetros para una turbina de 7KW (Sierra-García, 2020).....	29
Tabla 3.2. Variables de estado del controlador	36
Tabla 4.1. Diferencias entre los controladores propuestos.	38
Tabla 4.2. Hiper parámetros de DDPG utilizados para el control de pitch del aerogenerador.....	39
Tabla 4.3. Variables de estado del Controlador 1.....	39
Tabla 4.4. Variables de estado del Controlador 2.....	42
Table 4.5. Variables de estado del Controlador 1.....	46
Table 4.6. Variables de estado del Controlador 4.....	49
Tabla 4.7. Comparativa RMSE	52





Capítulo 1

Introducción

1.1. Motivación

Debido al modelo de consumo, y el crecimiento de los países emergentes, el consumo energético ha tenido una tendencia al alza durante todo el siglo XX y sigue acelerándose durante el siglo XXI, como puede observarse en la Figura 1.1. El crecimiento del consumo mundial de energía aumentó en 2023 (2,2 %), mucho más rápido que la tasa de crecimiento promedio de 2010-2019 (1,5 % anual). Además, en la distribución del consumo energético, siguen pesando especialmente los combustibles fósiles: petróleo y carbón.

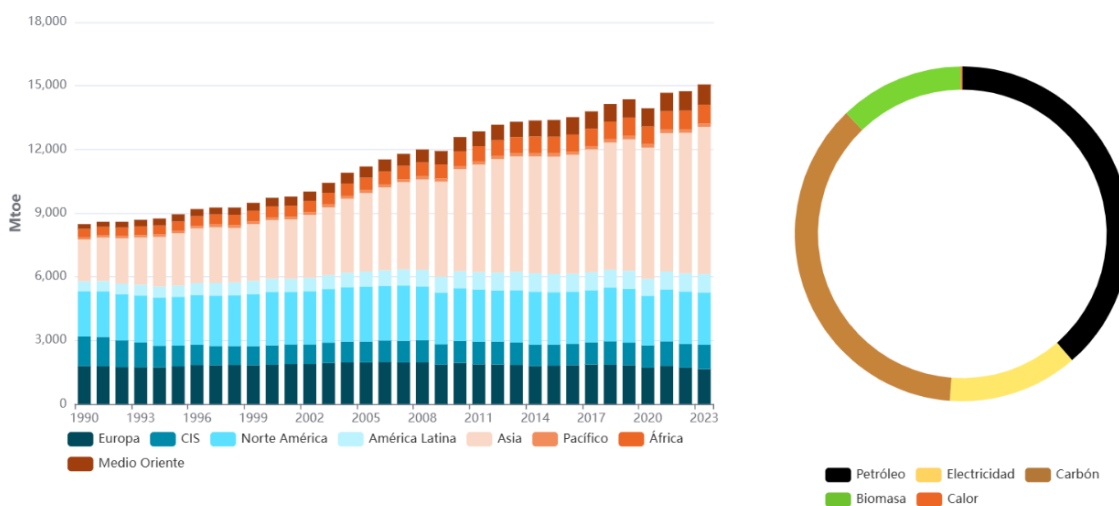


Figura 1.1. Consumo de energía mundial hasta 2023 en Mtoe (Mega Tonelada equivalente de petróleo), por regiones del mundo y tipo de energía. (Enerdata, 2024)

Tanto por motivos ecológicos, debido especialmente a la emisión de CO₂ por parte de los combustibles fósiles, como productivos en el caso de Europa, por carecer de dichos recursos naturales, se hace imprescindible impulsar la producción de energías renovables.

En concreto en España, en 2023, el consumo total de electricidad representó el 23.7% del consumo total de energía (International Energy Agency, 2024). Dentro de la electricidad consumida, las principales fuentes fueron la eólica y el gas natural, como puede observarse en la Figura 1.2.

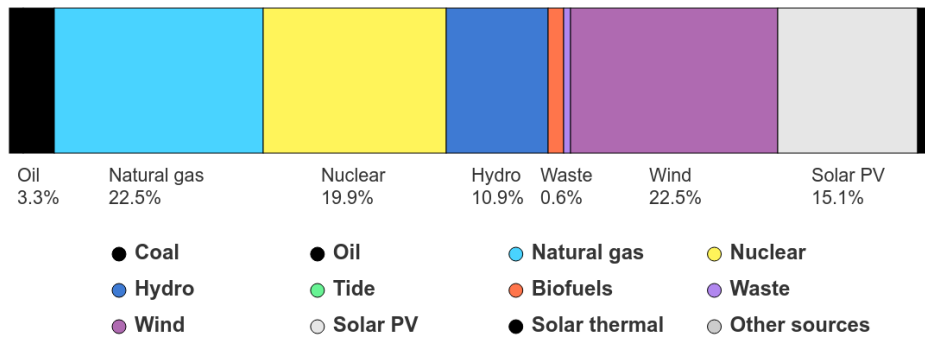


Figura 1.2. Reparto energético de la electricidad en España en 2023. Fuente: (International Energy Agency, 2024)

Las turbinas eólicas generan electricidad aprovechando la energía del viento para hacer girar un generador. Esta fuente de energía es altamente variable, lo que hace difícil obtener una salida de energía estable (Manwell, 2010), además de provocar grandes esfuerzos mecánicos en la estructura (Bossanyi, 2003). El diseño del aerogenerador se dimensiona para optimizar la extracción de energía en un rango de velocidades de viento, pero para maximizar su extracción, sin sobrepasar los límites mecánicos y eléctricos, es necesario disponer de un sistema de control adecuado (Pandit, 2022).

En concreto, las turbinas eólicas modernas cuentan con palas motorizadas, que permiten modificar la orientación del ángulo de palas respecto al viento, variando la potencia extraída. El control de este ángulo de palas es fundamental para mantener la extracción a la máxima potencia de diseño de la turbina, sin sobrepasar la velocidad permitida del rotor. Habitualmente se comandan todas las palas del aerogenerador con el mismo ángulo (Collective Pitch Control). Para escoger dicho ángulo, se suele utilizar un controlador PID del error de angular del generador. Sin embargo, como el sistema es altamente no lineal, se debe establecer técnicas de PID con ganancias variables en función del punto de operación. No obstante, debido a las mencionadas no linealidades, así como la gran variabilidad del viento, este problema es complejo y todavía queda rango de mejora, para conseguir una obtención de energía más suave, mitigando las cargas mecánicas. Por ello, se trata de un campo de investigación abierto, y se han propuesto métodos de control más modernos, tales como controladores robustos y adaptativos, control predictivo (López-Queija, 2024), o técnicas de control inteligente, como el control difuso (Serrano, 2022) o basado en redes neuronales, mediante aprendizaje no supervisado (Narayanan, 2024), o aprendizaje por refuerzo (Sierra-García, 2020).

1.2. Propuesta y objetivos

En este trabajo se propone utilizar aprendizaje por refuerzo para entrenar un controlador del ángulo colectivo de palas de un aerogenerador, que permita regular la velocidad del rotor pese al viento variable. El objetivo general es desarrollar un controlador inteligente que mejore la eficiencia en la regulación de la velocidad del rotor en comparación con los controladores tradicionales, a la vez que se mantiene la estabilidad del sistema. Para lograr esto, se explorarán y compararán diversas configuraciones de control basadas en aprendizaje por refuerzo, donde cada configuración se diseñará para abordar las limitaciones encontradas en las anteriores. Específicamente, se iterará sobre



el diseño de la función de recompensa y las variables de estado del controlador, con el fin de optimizar su capacidad de respuesta ante las variaciones del viento y reducir el error estacionario.

Para ello, se han seguido los siguientes pasos:

- Se ha implementado y validado un modelo de simulación de una turbina de 7kW.
- Se ha implementado y calibrado un controlador PI del ángulo de palas
- Se han diseñado y entrenado 4 controladores mediante aprendizaje por refuerzo, corrigiendo y mejorando su comportamiento en cada uno de ellos.
- Se ha realizado la validación de los controladores entrenados mediante aprendizaje por refuerzo en simulación frente al controlador PI, demostrando el éxito de la técnica de aprendizaje por refuerzo.

1.3. Estructura del documento

Este Trabajo de Fin de Máster se organiza en cinco capítulos, estructurados para guiar al lector desde los fundamentos teóricos hasta los resultados obtenidos y las conclusiones extraídas del estudio:

En el Capítulo 1 se presenta la motivación del trabajo, sus objetivos principales, las publicaciones derivadas y los recursos desarrollados. Se contextualiza el problema de control en aerogeneradores y se justifica el uso de técnicas de aprendizaje por refuerzo para abordarlo.

En el Capítulo 2 se revisan los conceptos esenciales relacionados con la energía eólica y el funcionamiento de los aerogeneradores. Asimismo, se introduce el aprendizaje automático, con especial énfasis en el aprendizaje por refuerzo, y se analizan las principales estrategias de control existentes en la literatura.

A continuación, en el Capítulo 3 se describe en detalle el modelo computacional de la turbina eólica utilizado en las simulaciones, así como la arquitectura general del sistema de control basada en aprendizaje por refuerzo. También se detallan las herramientas empleadas y la implementación del entorno de simulación.

En el Capítulo 4 se explican las distintas configuraciones de control desarrolladas, justificando las decisiones de diseño adoptadas en cada iteración. Se presentan los resultados obtenidos en simulaciones y se realiza una comparación con un controlador clásico de tipo PI.

Finalmente, en el Capítulo 5 se resumen los principales logros alcanzados y se discuten posibles líneas de investigación futura, como la mejora del entorno de simulación o la exploración de otros algoritmos de aprendizaje por refuerzo.



1.4. Publicaciones derivadas

Este Trabajo de Fin de Máster ha dado lugar a las siguientes publicaciones en congresos de carácter nacional e internacional.

- I. Cavero, J.E. Sierra-García, M. Santos. First Approximation of Reinforcement Learning Applied to a Wind Turbine Pitch Control. 13 European Conference on Renewable Energy Systems ECRES 2025, Perugia, Italy, May 26-27, 2025
- Inazio Cavero, Matilde Santos, J. Enrique Sierra-García. Control Inteligente del ángulo de palas de un aerogenerador mediante aprendizaje por refuerzo. WWWE 2024/ VI Workshop on Wind (and) Water Energy, 16/12/2024 UCM. Lecture Notes on Wind and Marine Energy, 37-42, 2025

1.5. Repositorios de código y recursos desarrollados

El código fuente desarrollado en este trabajo se ha publicado en formato abierto, organizado en los siguientes repositorios públicos:

- [nach96 / simpleWT-gym](#)
Contiene el entorno de simulación de la turbina eólica con interfaz compatible con Open AI Gym, descrito en el Capítulo 3.
- [nach96 / WindTurbine-RL](#)
Incluye el software empleado para el entrenamiento de los controladores mediante aprendizaje por refuerzo, explicado en el Capítulo 4.
- [nach96 / openfast-gym](#)
Proporciona una interfaz en Python para el simulador de alta fidelidad *OpenFAST*, permitiendo su integración con OpenAI Gym y su uso en entornos de aprendizaje por refuerzo.



Capítulo 2

Fundamentos y Estado del Arte

Este capítulo se estructura en dos secciones. En primer lugar, se realiza una introducción a las turbinas eólicas y al estudio de la literatura de los sistemas de control empleados en los aerogeneradores. En segundo lugar, se dedica una sección a explicar las bases del aprendizaje por refuerzo.

2.1. Fundamentos de las turbinas eólicas

Durante siglos, la energía del viento ha sido aprovechada por la humanidad a través de los molinos. Estas estructuras, que datan de la antigua Persia entre los siglos VI y VIII, convertían la fuerza del viento en energía mecánica para realizar tareas como moler grano o bombear agua. Su desarrollo y expansión a lo largo de la historia demuestran la larga tradición de la humanidad en utilizar el viento como un recurso energético valioso y sostenible. Estos primeros ingenios mecánicos, aunque rudimentarios en comparación con las turbinas modernas, sentaron las bases para la posterior evolución de la tecnología eólica.



Figura 2.1. Molinos de viento (Jaime González. [Wikimedia Commons](#))

El concepto de generar electricidad a partir del viento comenzó a tomar forma a finales del siglo XIX y principios del XX, dando lugar a los primeros aerogeneradores. Inicialmente, estos sistemas solían emplear máquinas síncronas conectadas directamente a la red eléctrica. Este enfoque, si bien permitía la generación de electricidad, presentaba limitaciones en cuanto a la flexibilidad de operación y la necesidad de sincronización precisa con la frecuencia de la red. La velocidad de rotación del generador estaba directamente ligada a la frecuencia de la red, lo que imponía restricciones en la eficiencia de la captación de energía eólica, que es inherentemente variable.

Los aerogeneradores modernos más utilizados son de eje horizontal. Constan de varios componentes clave, representados en la Figura 2.2: las palas (entre una y cinco, pero habitualmente tres), están diseñadas con perfiles aerodinámicos, que capturan la energía del viento y la convierten en un movimiento de rotación del rotor. Además, las palas suelen contar con un mecanismo de orientación de éstas respecto al viento. El rotor está conectado a una góndola, situada en lo alto de una torre, que eleva las palas para acceder a vientos más fuertes y constantes. Los sistemas de orientación permiten a la turbina girar sobre su eje vertical para encarar el viento.

La góndola alberga la multiplicadora y el generador, donde la energía mecánica se transforma en energía eléctrica. La aparición de los inversores de potencia supuso un avance significativo en la tecnología eólica ya que estos dispositivos electrónicos permiten desacoplar la velocidad de rotación del generador de la frecuencia de la red, lo que ofrece una mayor flexibilidad en el diseño y operación de los aerogeneradores.

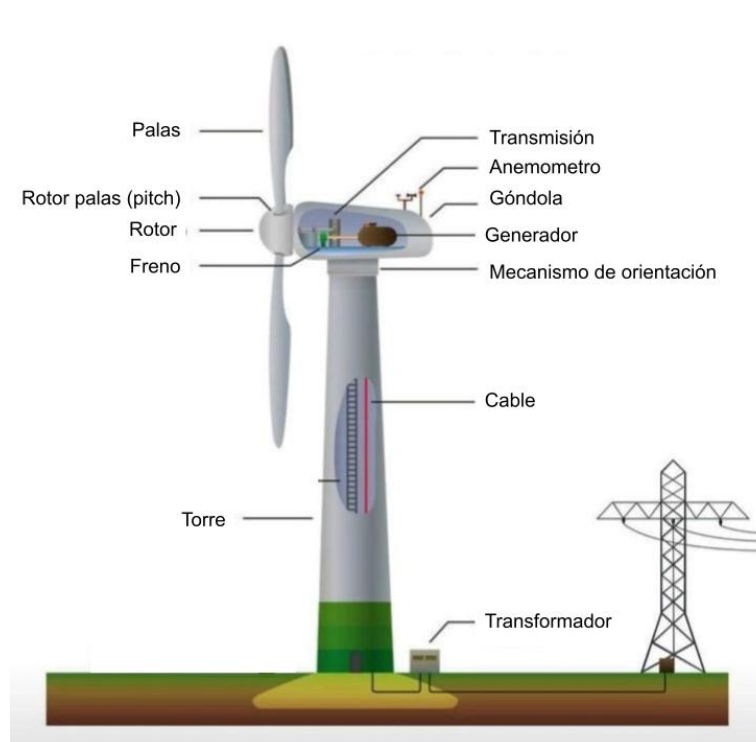


Figura 2.2. Partes de un aerogenerador de eje horizontal

2.2. Control de aerogeneradores

El objetivo del sistema de control del aerogenerador es maximizar la potencia extraída del viento, dentro de los límites del aerogenerador. La potencia del viento P_v [W] que atraviesa cierta área viene dada por la ecuación Ec. (1):

$$P_v = \frac{1}{2} \rho A v^3 \quad (1)$$

Donde ρ [Kg/m³] es la densidad del aire, A [m²] el área barrida por las palas y v [m/s] la velocidad del viento. Sin embargo, la turbina no es capaz de extraer toda esta energía, sino que la potencia extraída depende de la eficiencia de cada turbina. El límite teórico de eficiencia se conoce



como límite de Betz y es de un 59 %. La potencia extraída del viento después de aplicar el coeficiente de potencia $C_p(\beta, \lambda)$ se muestra en la ecuación Ec. (2):

$$P_w = \frac{1}{2} \rho A C_p v^3 \quad (2)$$

El coeficiente de potencia $C_p(\beta, \lambda)$ depende del diseño de cada turbina, así como del ángulo de ataque de las palas (β [rad]) y del ratio de velocidad en punta (λ). Para optimizar la potencia extraída se debe aumentar el coeficiente de potencia todo lo posible, por lo que en los aerogeneradores modernos hay dos mecanismos principales de actuación: el par del generador y el ángulo de palas. El par del generador permite actuar directamente sobre el rotor para variar su velocidad hacia el régimen de trabajo que optimice la extracción de energía. Por otro lado, el ángulo de palas influye directamente en la potencia extraída del viento, variando la alineación de las palas frente a la dirección del viento, para permitir una obtención de energía constante.

Desde el punto de vista de control, el aerogenerador es un sistema altamente no lineal, multivariable y multiobjetivo (Elkodama, 2023). Controladores tradicionales han tenido éxito en realizar esta tarea, dividiendo el problema para los distintos modos de operación en función de la velocidad del viento (Figura 2.3), con controladores específico para cada región de trabajo (Muñoz-Palomeque, 2024).

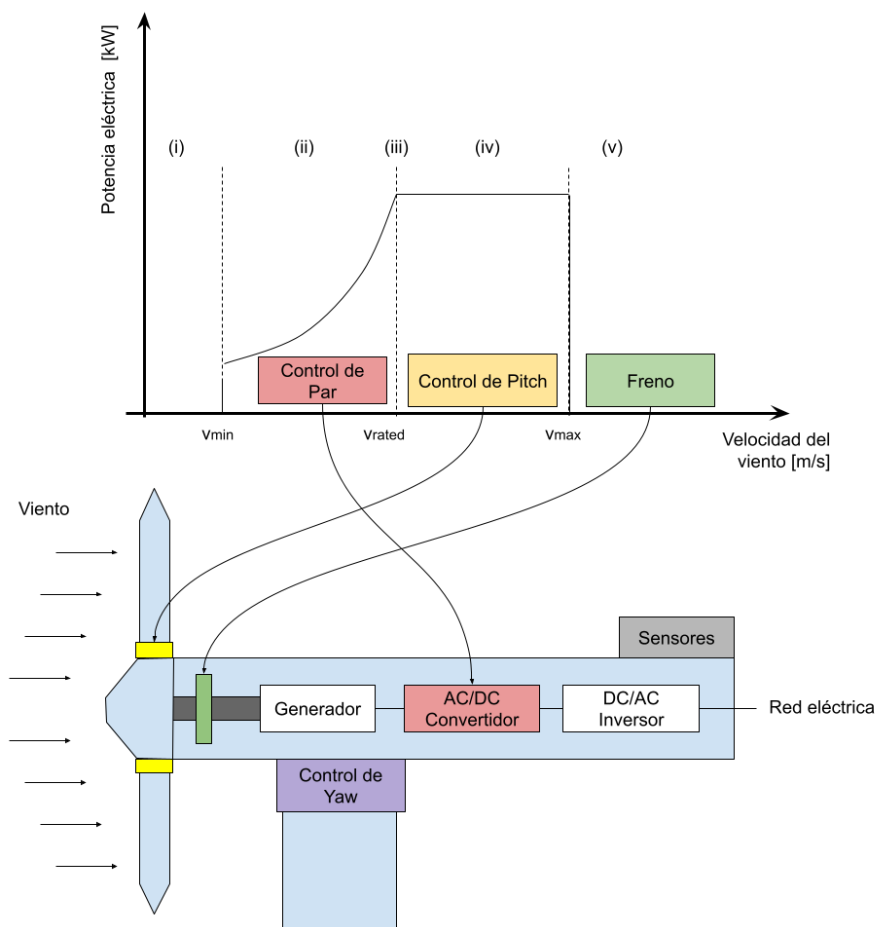


Figura 2.3. Modos de operación del control del aerogenerador.



Regiones de trabajo de un aerogenerador (Figura 2.3, arriba):

- I. El aerogenerador permanece parado debido a que la velocidad del viento es demasiado baja (velocidad del viento inferior a v_{\min}). La velocidad de conexión suele estar en torno a los 3-4 m/s.
- II. Carga parcial (velocidad del viento entre v_{\min} y v_{rated}). A velocidad de viento subnominal, el objetivo es maximizar la producción de energía. El ángulo de palas del aerogenerador permanece a un valor fijo que extrae la máxima potencia del viento, y se actúa sobre el par del aerogenerador para seguir los puntos de velocidad óptimos. Estos algoritmos se denominan de seguimiento del punto de máxima potencia, MPPT (Maximum Power Point Tracking).
- III. Esta es la región de transición entre las regiones (ii) y (iv) (velocidad de viento cercana a v_{rated}). Se emplean estrategias de control que consigan una conmutación suave entre los controladores de estas dos regiones.
- IV. Región nominal o a plena carga (velocidad del viento entre v_{rated} y v_{\max}). En esta sección la velocidad del viento es alta, y se debe conseguir que el aerogenerador trabaje a potencia nominal, variando el ángulo de palas para regular el par de entrada, manteniendo la velocidad del aerogenerador constante.
La variación del ángulo de ataque de las palas se puede conseguir mediante técnicas pasivas, ajustando la flexibilidad de las palas para que varíen su ángulo de ataque en función de la velocidad del viento; o mediante actuadores que permitan ajustar dicho ángulo en tiempo real en base a un algoritmo de control.
- V. La velocidad del viento es demasiado alta (superior a v_{\max}) y puede dañar la turbina, así que el ángulo de palas se establece paralelo al viento (ángulo de ataque 0°) y se utilizan los frenos mecánicos para parar la turbina.

2.3. Aprendizaje automático

Desarrollar controladores clásicos basados en modelos para sistemas no lineales y/o multivariables como el control de un aerogenerador es una tarea compleja, y su comportamiento está limitado por las hipótesis tomadas durante su diseño. Por otro lado, las aproximaciones basadas en datos prometen ser una alternativa útil, superando estas hipótesis al aprender directamente en base a datos reales del sistema.

Las tecnologías de aprendizaje automático se pueden clasificar en tres grandes grupos: Aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo (Figura 2.4).

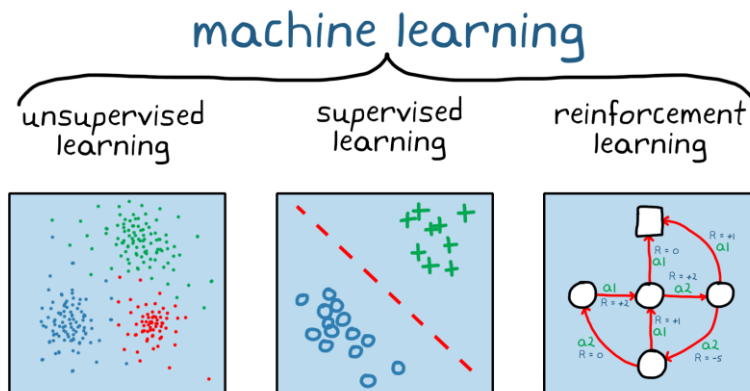


Figura 2.4. Clasificación de métodos de aprendizaje automático (Mathworks, 2023)

En aprendizaje supervisado se parte de un conjunto de datos, etiquetados previamente, y a partir de los mismos se ajusta un modelo de aprendizaje mediante optimización tal que minimice el error de la salida predicha por el modelo ante una entrada dada.

El inconveniente de esta estrategia es que requiere mucho trabajo previo para identificar dichos datos. Para evitarlo, el aprendizaje no supervisado aprende a realizar estimaciones únicamente en base a la estructura de los datos, sin necesidad de esa identificación previa.

Por último, el aprendizaje por refuerzo no parte de un conjunto de datos estático, sino que recopila los datos mediante prueba y error entre un agente y el entorno, automáticamente actualizando el modelo en función de la calidad de los resultados. Esta técnica es especialmente útil para problemas de control y será la empleada en este trabajo.

Por otro lado, existen multitud de modelos de aprendizaje capaces de realizar un mapeo entre entradas y salidas. La estructura de estos modelos da lugar a otras clasificaciones del aprendizaje automático. Uno de los modelos más extendidos son las denominadas redes neuronales.

2.3.1. Redes neuronales

Las redes neuronales artificiales están basadas en la forma en que el cerebro humano procesa la información, mediante la conexión de múltiples unidades individuales de procesamiento. Por analogía, la unidad de procesamiento computacional se denomina neurona. Esta realiza una combinación ponderada de todas sus entradas, tal como se muestra en la figura 2.5. La combinación de funciones lineales continúa siendo una función lineal, por lo que, para poder representar funciones no lineales, se añade una función de activación a la salida de cada neurona. Se utilizan diferentes funciones matemáticas como función de activación (ver Figura 2.6), tales como sigmoid, tanh o ReLU.

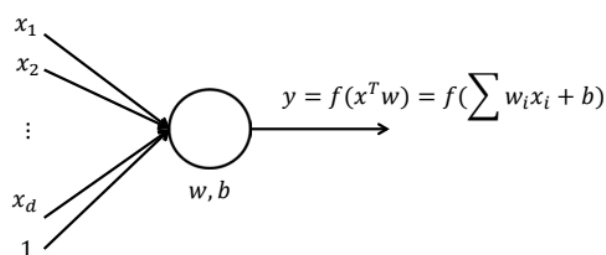


Figura 2.5. Neurona artificial

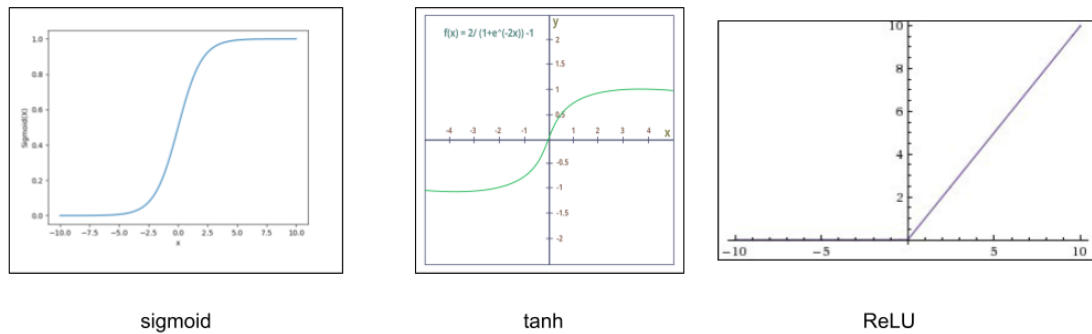


Figura 2.6: Funciones de activación

Finalmente, las neuronas se conectan entre sí en capas, formando la denominada red neuronal (Figura 2.7). Con un número suficiente de neuronas y al menos una capa intermedia es posible aproximar cualquier función.

Existen diferentes formas de conectar las neuronas que se han ido haciendo más complejas a medida que los algoritmos de entrenamiento han permitido ajustar los pesos en capas más profundas, dando lugar a lo que se conoce como Deep Learning. Entre algunos de los modelos más conocidos están desde las más básicas redes neuronales multicapa (Multi Layer Perceptron, MLP) como las de la figura 2.5; las redes neuronales recurrentes (Sherstinsky, 2020), especialmente útiles para analizar datos secuenciales; redes convolucionales (Albawi, 2017), que han demostrado gran éxito en el campo de análisis de imágenes; redes neuronales ODE (Chen, 2018), para modelar sistemas físicos en ecuaciones diferenciales, o los famosos Transformers, empleados en los grandes modelos de lenguaje (Vaswani, 2017).

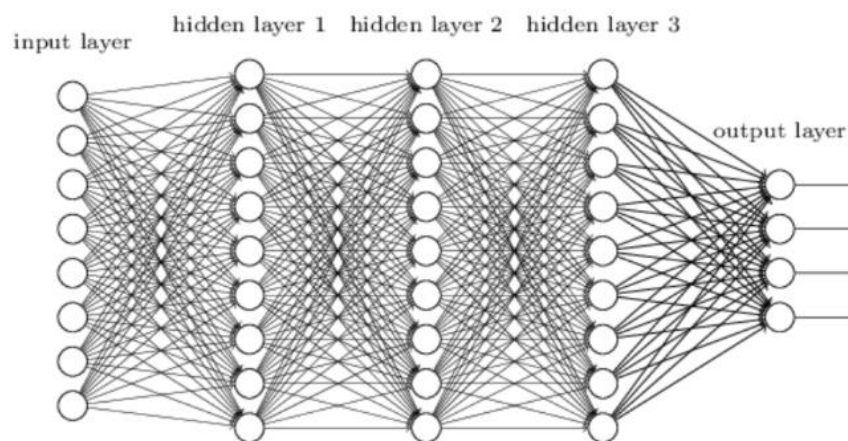


Figura 2.7: Representación de una red neuronal multicapa

Una vez escogido un modelo, el entrenamiento de la red neuronal consiste en ajustar las ganancias ω , b de cada neurona para producir la salida deseada a partir de un conjunto de entradas.



2.3.2. Aprendizaje por refuerzo

El aprendizaje por refuerzo es un área del aprendizaje automático inspirado en la psicología constructivista. Plantea resolver problemas de control mediante agentes interactuando con el entorno, mejorando sucesivamente su comportamiento en base a las recompensas obtenidas en el proceso (Shakya, 2023).

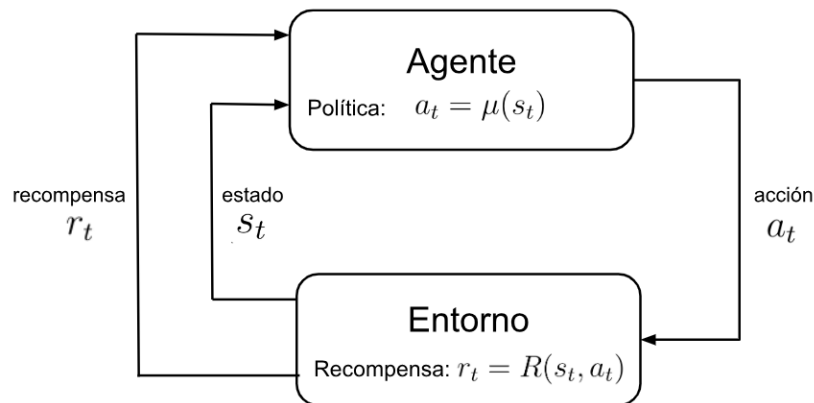


Figura 2.8: Representación de los elementos básicos en Aprendizaje por Refuerzo

El esquema básico del aprendizaje por refuerzo se muestra en la Figura 2.8. El agente, en el tiempo t y en el estado s_t , escoge realizar una acción a_t en función de la política $\gamma(s_t)$. La calidad de realizar esa acción, para ese estado, se evalúa mediante la recompensa r_t . En base a esas recompensas se deberá actualizar la política.

El objetivo final del controlador no es maximizar la recompensa en cada paso, sino la recompensa total de la *trayectoria* ($\tau = (s_0, a_0, s_1, a_1, \dots)$):

$$R(\tau) = \sum_{t=0}^T r_t \quad (3)$$

Como antes de finalizar la trayectoria no es posible conocer la recompensa, es útil estimar la esperanza de recompensa al comenzar en un estado dado y aplicando cierta política. A esta estimación se le denomina “On-Policy Action-Value Function” $Q^\mu(s, a)$.

$$Q^\mu(s, a) = E_{\tau \sim \mu}[R(\tau) | s_0 = s, a_0 = a] \quad (4)$$

A partir del concepto de optimalidad de Bellman, se puede formular dicho valor en dos términos, la recompensa inmediata y la recompensa futura (la Q-function comenzando en el estado siguiente (s') obtenido de aplicar la política), descontada con un factor γ :

$$Q^\mu(s, a) = E[r(s, a) + \gamma E[Q^\mu(s', a')]] \quad (5)$$

De entre las Q-functions, la óptima es la que maximiza la esperanza de recompensa. En los algoritmos de Q-learning se denomina Crítico al componente que calcula la estimación de este valor de recompensa.

$$Q^*(s, a) = E[r(s, a) + \gamma \max_{a' \sim \mu} Q^*(s', a')] \quad (6)$$

Para conseguir esa recompensa máxima, la acción óptima es la que maximiza dicha Q-function. En los algoritmos basados en el cálculo de la política, se suele denominar Actor al componente que calcula la acción.

$$a^*(s) = \operatorname{argmax}(Q^*(s, a)) \quad (7)$$

A partir de esta formulación del problema, existen algoritmos que intentan aprender los distintos elementos, o en ocasiones varios de ellos (Figura 2.9). Ordenados de más bajo a alto nivel, se puede aprender:

- **La política (Actor):** El actor calcula directamente la acción óptima.
- **La estimación de la recompensa (Q-learning, Crítico).** Conocido el valor de la recompensa, en un sistema discreto, se pueden probar todas las acciones posibles hasta dar con la óptima. En un sistema continuo, puede resolverse un problema de optimización que encuentre las acciones óptimas.
- **El modelo del entorno.** Conocido el modelo del entorno puede resolverse un problema de optimización para encontrar la política óptima mediante técnicas como control predictivo basado en modelo (MPC).

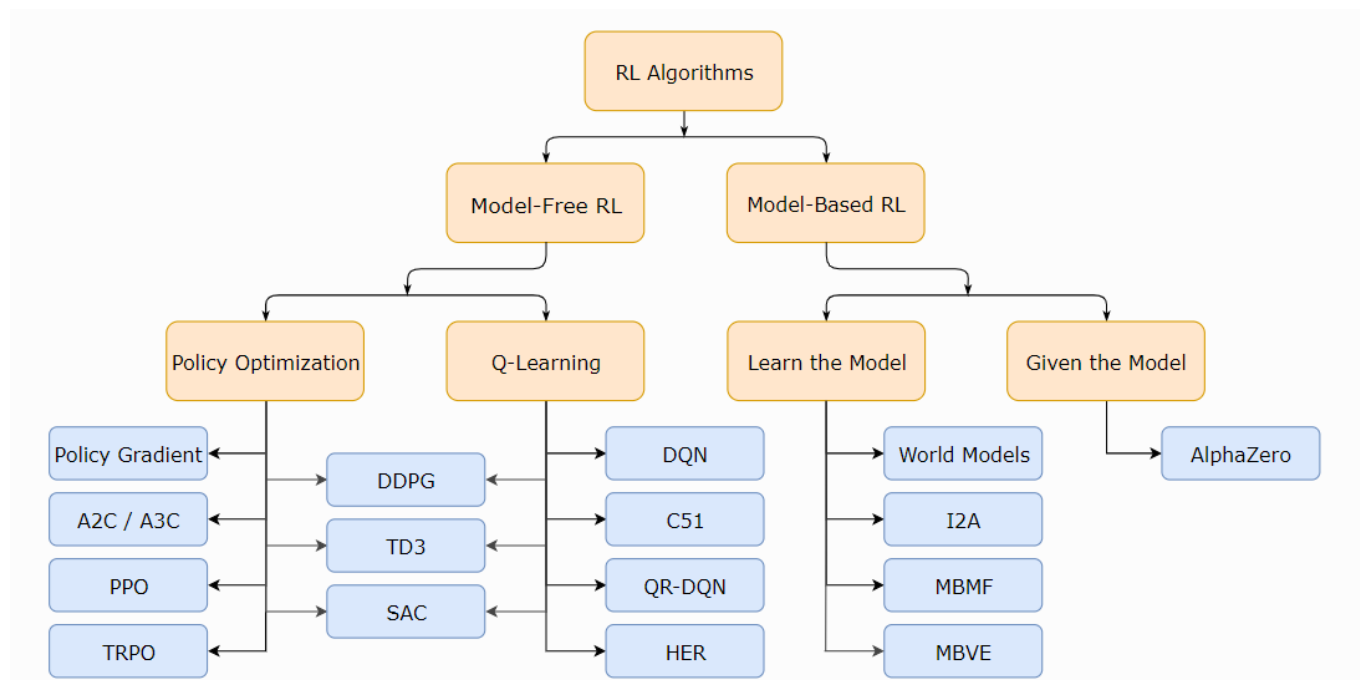


Figura 2.9: Clasificación de algoritmos de aprendizaje por refuerzo (OpenAI, 2020).



De entre estos algoritmos se ha seleccionado *Deep Deterministic Policy Gradient DDPG* (Lillicrap, 2015). Fue el primer algoritmo de tipo Actor-crítico para variables continuas. Es decir, combina ambos métodos ya que tiene un Actor que calcula directamente la acción, pero también tiene un crítico cuyo valor se utiliza para optimizar la política. Para adaptarlo a variables continuas, tanto el actor como el crítico están formados por redes neuronales.

Después de DDPG surgieron otros algoritmos más robustos basados en la misma filosofía, como *Soft Actor Critic, SAC* (Haarnoja, 2018) y *Twin Delayed Deep Deterministic Policy Gradient, TD3* (Fujimoto, 2018), pero según la literatura son más complicados de calibrar, por lo que se recomienda comenzar en nuevos problemas con DDPG.

Capítulo 3

Modelo de aerogenerador y arquitectura de control

En esta sección se presentan las técnicas y herramientas de software empleadas en el desarrollo del trabajo. En primer lugar se define el modelo del aerogenerador utilizado para entrenar el controlador mediante aprendizaje por refuerzo en simulación. En segundo lugar, la arquitectura de control empleada y, en tercer lugar, las técnicas de entrenamiento.

3.1. Modelo del aerogenerador

Realizar el entrenamiento en un aerogenerador real es inviable, debido al riesgo de dañarlo, y al elevado número de horas de recopilación de datos que requeriría. Por ese motivo, se ha implementado un modelo de aerogenerador basado en (Sierra-García, 2020).

La turbina se modela como un rotor con inercia $J [Kg \cdot m^2]$, sujeto al par del viento $T_m [N]$, el par electromagnético del generador $T_{em} [N]$ y fricción $K_f [\frac{Nm}{rad/s}]$, cuya variable de estado es la velocidad del rotor $\omega [\frac{rad}{s}]$

$$J \frac{d\omega}{dt} = T_m - T_{em} - K_f \cdot \omega \quad (8)$$

El par electromagnético T_{em} se calcula mediante el modelo de un generador de corriente continua, a partir de la corriente $I_a [A]$ y definido por los parámetros: $R_a [\Omega]$ (Resistencia de armadura), $L_a [\Omega]$ (Inductancia de armadura), K_g (constante adimensional del generador), $K_\phi [\frac{V}{rad/s}]$ (constante de flujo electromagnético) y $R_L [\Omega]$:

$$T_{em} = K_g K_\phi I_a \quad (9)$$

$$L_a \frac{dI_a}{dt} = E_a - R_L I_a - R_a I_a \quad (10)$$

$$E_a = K_g K_\phi \omega \quad (11)$$

$$P_{out} = R_L I_a^2 \quad (12)$$

Por otro lado, el par aerodinámico T_m (Ec.13) se calcula partiendo de la potencia máxima del viento en la sección $A [m^2]$ barrida por el aerogenerador, reducida en un coeficiente de potencia C_p .



Este puede modelarse mediante expresiones empíricas (Ecs. (14) – (16)) en función del ángulo de palas θ [rad] y del ratio de velocidad en punta λ (Ec. 14).

$$T_m = \frac{C_p(\lambda, \theta) \rho A v^3}{2\omega} \quad (13)$$

$$\lambda = \omega \frac{R}{v} \quad (14)$$

$$\lambda_i(\lambda, \theta) = \left(\frac{1}{\lambda + c_8} \right) - \frac{c_9}{\theta^3 + 1} \quad (15)$$

$$C_p(\lambda_i, \theta) = c_1 \left[\frac{c_2}{\lambda_i} - c_3\theta - c_4\theta^{c_5} - c_6 \right] e^{c_7/\lambda_i} \quad (16)$$

Por último, la única modificación del modelo implementado respecto al artículo de referencia (Sierra-García, 2020) está en el modelo de la dinámica de las palas. Al modelo de primer orden que representa el sistema en bucle cerrado de la pala, actuador y control de posición, se le ha añadido saturación en la derivada. La saturación es esencial para hacer el modelo más realista. En un sistema real, los actuadores que controlan el ángulo de las palas tienen limitaciones físicas en la velocidad a la que pueden moverse. No pueden cambiar el ángulo instantáneamente. La saturación en la ecuación 17 modela estas limitaciones, evitando que el modelo prediga cambios de ángulo de palas que son físicamente imposibles.

$$\dot{\theta} = \max \left(-\dot{\theta}_{max}, \min \left(\dot{\theta}_{max}, \frac{1}{\tau_\theta} (\theta_{ref} - \theta) \right) \right) \quad (17)$$

Los parámetros del modelo son los de una turbina de 7 kW, y se muestran en la Tabla 3.1.

Tabla 3.1. Parámetros para una turbina de 7KW (Sierra-García, 2020).

Parámetro	Descripción	Valor	Unidad
L_a	Inductancia de armadura	13.5	mH
K_g	Constante adimensional del generador	23.31	
K_{phi}	Constante de flujo electromagnético	0.264	V/rad/s
R_a	Resistencia de la armadura	0.275	Ω
R_L	Resistencia de la carga	8	Ω
J	Inercia del rotor	6.53	Kg m^2
R	Radio de las palas	3.2	m
ρ	Densidad del aire	1.223	Kg/m^3
K_f	Fricción del rotor	0.025	Nm/rad/s
$(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9)$	Factores de eficiencia (C_p)	(0.73, 151, 0.58, 0.002, 2.14, 13.2, 18.4, -0.02, -0.03)	-
τ_θ	Constante dinámica del pitch	2	-
$\dot{\theta}_{max}$	Máxima tasa de variación del pitch	0.087	rad/s



3.1.1. Implementación del modelo de aerogenerador

Este modelo se ha implementado en Python con la interfaz de Open AI Gym (Brockman, 2016) que permite utilizarlo con algoritmos de aprendizaje por refuerzo genéricos. En la Figura 3.1 se muestra la estructura del entorno de la turbina eólica con la interfaz de Open AI Gym. Está formada principalmente por tres funciones que son llamadas por el algoritmo de aprendizaje por refuerzo.

- **init**
En esta función deben definirse la estructura, número de las acciones y las observaciones, si son discretas o continuas y el valor mínimo y máximo que pueden tener, además de inicializar el simulador de la turbina.
- **Reset**
Esta función se llama al comienzo de cada episodio, por lo que debe reiniciar las condiciones iniciales y generar la primera observación.
- **Step**
Esta función se llama en cada paso de control. Debe correr la simulación durante el periodo de control definido, aplicando las acciones dadas y generando tanto las observaciones como la recompensa.

La integración numérica del sistema de ecuaciones diferenciales ordinarias (ODE) definido en las ecuaciones Ec. (8)-(17), se realiza mediante la función *solve_ivp*, del módulo *integrate* de la librería *scipy*. Se ha escogido el método RK45 (Ruge Kutta de orden 4 con paso variable).

En formulación compacta, el sistema ODE se define en la ecuación (18).

$$\dot{x} = f(x, u) \quad (18)$$

Donde las variables de estado x son la velocidad angular del rotor, la corriente, el ángulo de palas y la velocidad del ángulo de palas.

$$x = [\omega, I_a, \theta, \dot{\theta}], \quad (19)$$

Y las entradas u son el ángulo de palas de referencia y la velocidad del viento.

$$u = [\theta_{ref}, v_x] \quad (20)$$

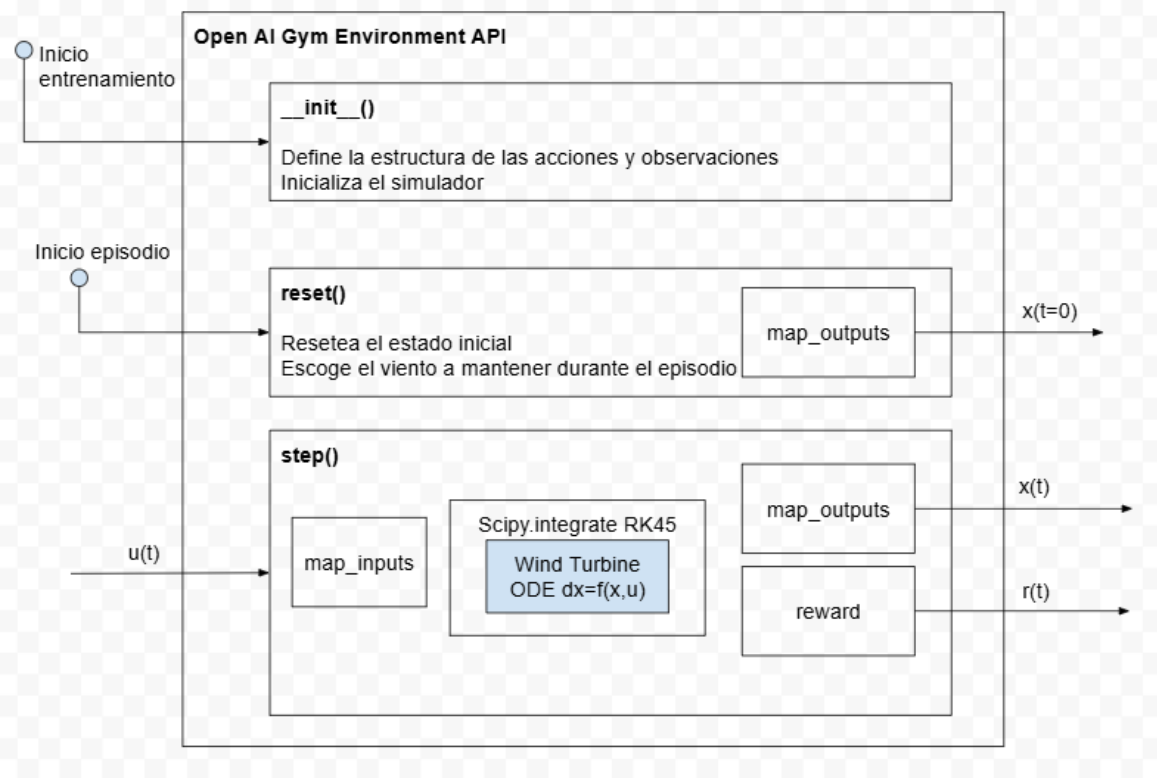


Figura 3.1: Entorno de Turbina Eólica con interfaz de Open AI Gym

3.1.2. Validación del modelo

Validación frente al modelo de referencia

En primer lugar, se valida el modelo sin modificaciones, comparando la implementación en Python del modelo existente en Simulink, utilizado con éxito en (Sierra-García, 2020). La dinámica de pitch se modela como un sistema de primer orden según la ecuación 21:

$$\dot{\theta} = \frac{1}{\tau_{\theta}} (\theta_{ref} - \theta) \quad (21)$$

Se realiza una simulación con viento constante de 12.3 m/s, en bucle abierto (sin controlador), realizando escalones de ángulo de palas de referencia, entre 0 rad y $\pi/2$ rad.

Entradas:

- v_x (Velocidad del viento) = 12.3 m/s
- θ_{ref} (Ángulo de palas de referencia) = [0, $\pi/4$, $\pi/3$, $\pi/2$, 0] rad

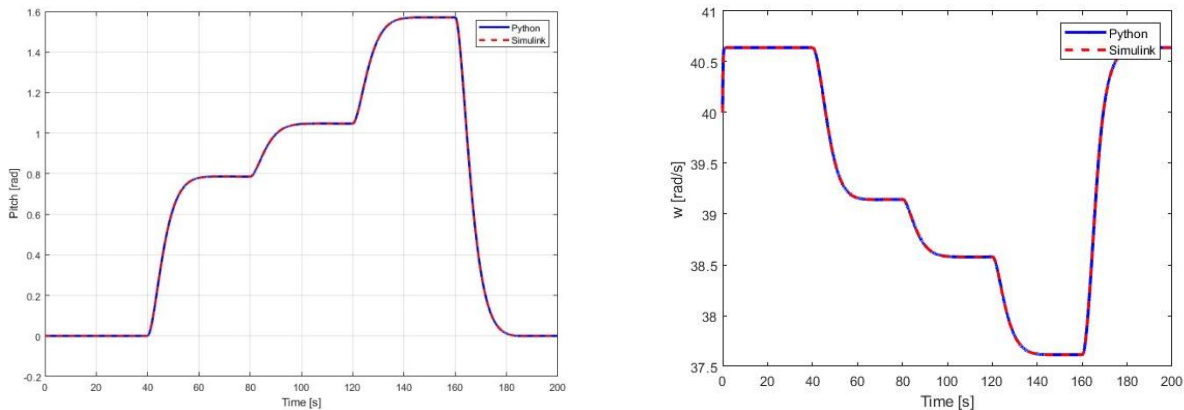


Figura 3.2: Simulación de validación del modelo de aerogenerador

En la Figura 3.2 se compara la respuesta ante los escalones del ángulo de palas de referencia del simulador en Simulink (línea roja punteada) y en Python (línea azul). Se puede observar que ambas respuestas se superponen completamente para los distintos valores de entrada.

En la figura 3.2 izquierda se observa la evolución del ángulo de palas con su dinámica de primer orden en respuesta a los escalones del ángulo de palas de referencia. Por su parte, en la figura 3.2 derecha, se observa la evolución de la velocidad del rotor ante esos cambios en el ángulo de palas. Cuanto mayor es el ángulo, menor es la velocidad del rotor. Se observa también cómo la influencia del ángulo de palas es limitada. Para una velocidad de viento concreta (12.3 m/s en esta simulación), es posible modificar el ángulo de palas en el rango $[0, \pi/2]$ rad, obteniendo una variación en la velocidad del rotor entre $[40.5, 37.5]$ rad/s. Ambas gráficas se solapan durante toda la simulación, por lo que puede darse la implementación del modelo de la turbina en Python por validada.

Una limitación importante de este modelo es que, al utilizar una aproximación lineal de primer orden para la respuesta del ángulo de palas, solamente es válido para pequeñas variaciones de ángulo de palas respecto al punto de operación, mientras que necesitamos que el modelo sea operativo en todo el rango $[0, \pi/2]$ rad. En concreto, el problema es que el modelo de primer orden tiene el mismo tiempo de respuesta ante cualquier tamaño de escalón. Por ello, si se da un escalón grande al ángulo de referencia de palas, la velocidad de éstas será mucho mayor a la que en la realidad pueden moverse, por las limitaciones de par, fricción mecánica, fricción aerodinámica etc. que no están contempladas en este modelo.

Como desconocemos muchos de esos datos, pero sí conocemos la velocidad máxima a la que pueden rotar las palas, una solución sencilla y elegante es incluir una saturación en la derivada del ángulo de palas (ecuación 17). De modo que, ante escalones pequeños, la respuesta será igual que el modelo existente, pero ante escalones grandes, no superará la limitación de velocidad conocida. Se decide mantener el modelo de primer orden ya que se considera que el controlador del actuador, que se incluye también implícitamente en este modelo, se encargará de realizar un seguimiento de la posición requerida sin oscilaciones.



Modelo con dinámica de pitch con saturación del ratio del ángulo de paso

Se compara ahora la respuesta al escalón del ángulo de palas del modelo en Simulink (Sierra-García, 2020), con una nueva propuesta incluyendo la saturación de 5°/s del ratio del ángulo de paso.

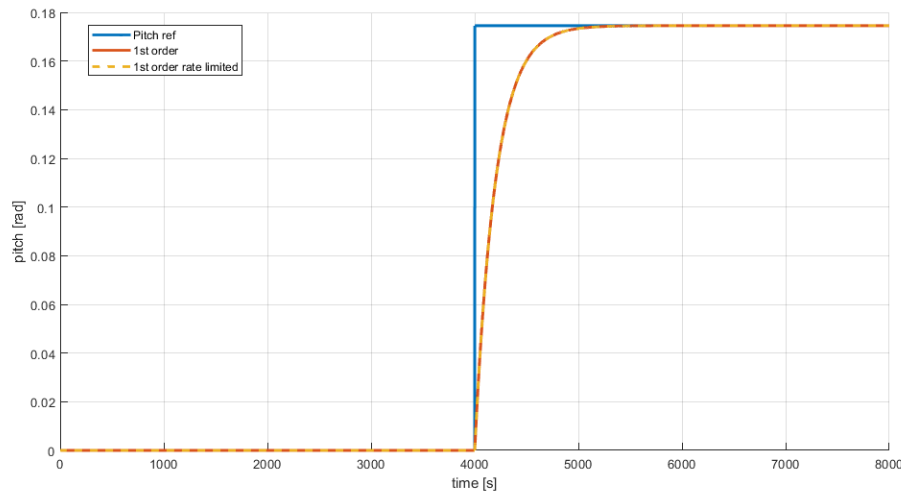


Figura 3.3: Respuesta al escalón de pitch de referencia de 10°

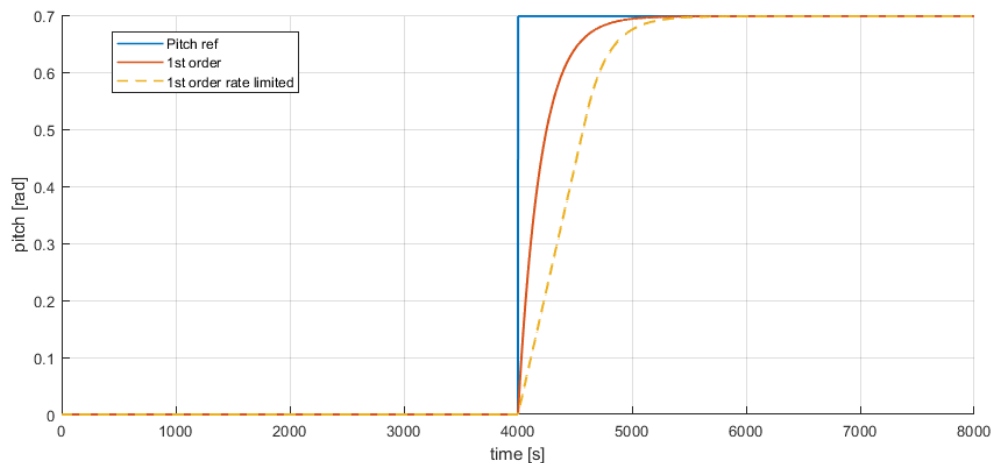


Figura 3.4: Respuesta al escalón de pitch de referencia de 40°

Como se indicaba en el objetivo para este nuevo modelo, en la Figura 3.3 se observa como para escalones pequeños la respuesta de ambos modelos coincide, mientras que en la Figura 3.4 se observa cómo para escalones más grandes, por ejemplo de 0.7 rad, el nuevo modelo con saturación es más lento que el original, respetando las limitaciones reales del sistema.

A continuación, se presenta la comparativa en las simulaciones de ambos modelos en las mismas condiciones de simulación de la Figura 3.2, es decir, las respuestas a escalones del ángulo de pitch de referencia:

Entradas:

- v_x (Velocidad del viento) = 12.3 m/s
- θ_{ref} (Angulo de palas de referencia) = $[0, \pi/4, \pi/3, \pi/2, 0]$

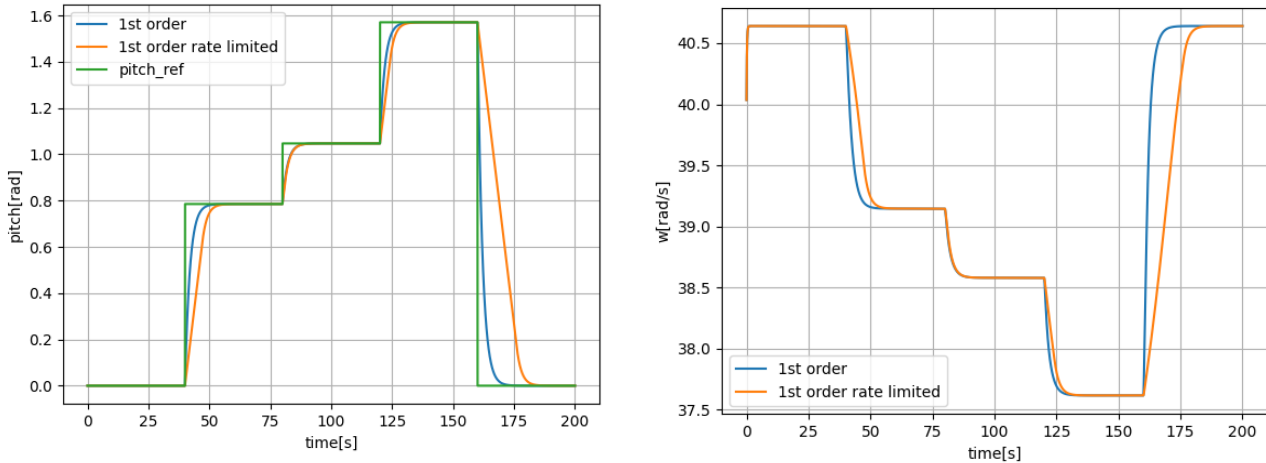


Figura 3.5: Simulación de escalones de ángulo de pitch. Comparación de la respuesta de los dos modelos

En la Figura 3.5, se aprecia la diferencia de entre los dos modelos. La referencia de pitch es la línea verde, mientras que la línea azul es el modelo original, y la naranja la nueva propuesta. En los escalones grandes, especialmente el primero y el último, el nuevo modelo es más lento (y realista) en realizar el movimiento de las palas, lo que también hace que sea más lento en generar el cambio de velocidad del rotor.

3.2. Arquitectura de control

En esta sección se explica con detalle la arquitectura de control, así como su implementación software. En el capítulo 4 se explicarán las particularidades de los distintos controladores que se han diseñado y se especifican las configuraciones aplicadas con respecto a la estructural general.

La arquitectura de control del ángulo de palas propuesta se muestra en la Figura 3.6.

A partir de la potencia nominal (P_{Nom}), se calcula la velocidad nominal del rotor como referencia para el controlador (ω_{ref}).

El controlador (Política de control) se basa en una red neuronal, que calcula el incremento de pitch ($\Delta\theta$) como acción, a partir del estado $x(t)$. El propósito de utilizar el incremento es incorporar las restricciones de la tasa de variación de pitch en la formulación de forma intrínseca.

El incremento de pitch se convierte a continuación en la referencia de ángulo de palas (θ_{ref}), que se utiliza como entrada del modelo de aerogenerador.



El entrenamiento de la política del controlador se realiza mediante aprendizaje por refuerzo, utilizando el algoritmo DDPG. El Crítico es nuevamente una red neuronal entrenada a partir de la recompensa inmediata (r_t), que es una función del error de velocidad del rotor.

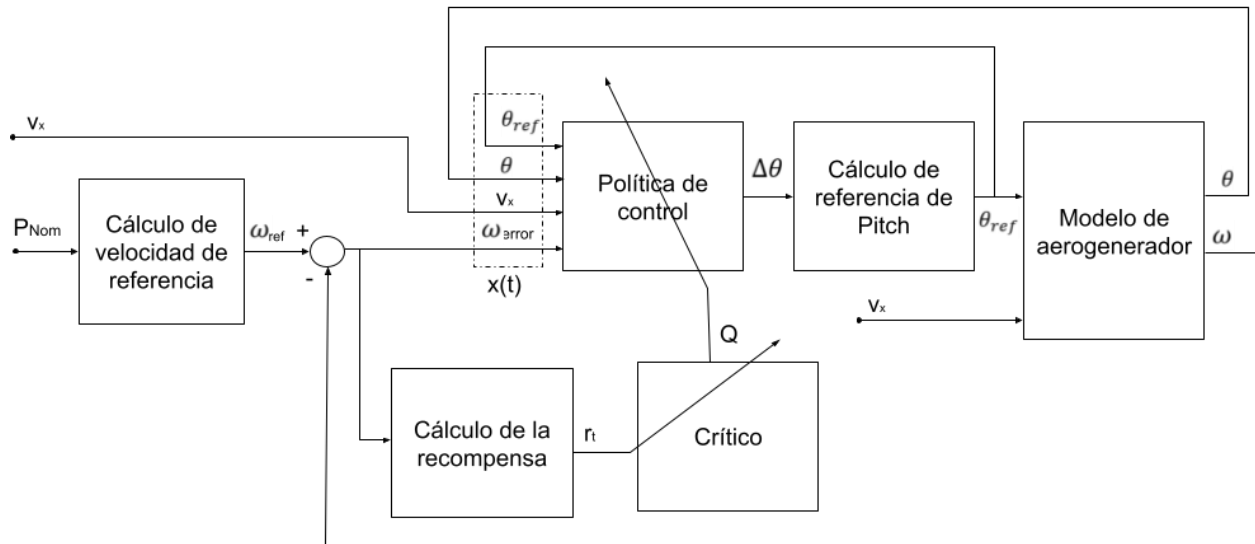


Figura 3.6: Arquitectura de control de ángulo de palas.

A continuación se describen por partes los distintos bloques y variables de la Figura 3.6.

Cálculo de la velocidad de referencia del rotor

De la ecuación (13), se despeja la corriente nominal:

$$I_{nom} = \sqrt{\frac{P_{nom}}{R_L}} \quad (22)$$

Y sustituyendo en (8), en estado estacionario, $\dot{I}_a = 0$, la velocidad nominal se calcula cómo:

$$\omega_{ref} = \omega_{nom} = \frac{R_a + R_L}{K_g K_\phi} I_{nom} \quad (23)$$

Con los datos de una turbina real de 7kW presentados en la Tabla 3.1, la velocidad nominal de referencia es $\omega_{ref} = 39.77 \text{ rad/s}$.

Política de control

La política de control se implementa mediante una red neuronal multicapa (MLP), con 2 capas ocultas de 32 neuronas.



Las variables de estado del controlador se definen en la tabla 3.2. Es importante definir correctamente el rango, porque el algoritmo de aprendizaje realiza una normalización de las variables de estado para ponderar la influencia de cada una de ellas correctamente.

Tabla 3.2. Variables de estado del controlador

Variable	Definición	Unidad	Rango
$\theta_{ref}(t)$	Referencia de pitch anterior	rad	$[0, \pi]$
$\theta(t)$	Angulo de pitch actual	rad	$[0, \pi]$
$v(t)$	Velocidad del viento	m/s	$[12, 13]$
$\omega_{error}(t)$	Error de velocidad del rotor	rad/s	$[-10, 10]$

La acción, como se ha comentado, es el incremento de pitch normalizado ($\Delta\theta$) en el rango $[-1, 1]$.

Cálculo de referencia de pitch

El incremento de pitch debe convertirse en la referencia de pitch, que es la acción del controlador a enviar al aerogenerador. Se calcula como:

$$\theta_{ref}(t) = \theta_{ref}(t - 1) + \Delta\theta \dot{\theta}_{max} \Delta t_{step} \quad (45)$$

Donde $\dot{\theta}_{max}$ es la máxima tasa de variación del ángulo de palas, que deberá convertirse de valor normalizado a absoluto. Su valor es de 0.087 rad/s. Δt_{step} es el tiempo del paso de control, de 0.2s. Por último, $\theta_{ref}(t - 1)$ es el ángulo de referencia de pitch anterior, al que se debe sumar el nuevo incremento, para calcular la nueva referencia de pitch.

Crítico

El crítico se modela también con una red neuronal multicapa, con 2 capas ocultas de 32 neuronas.

Cálculo de la recompensa

La definición de la recompensa, como en todo problema de optimización, es uno de los factores más importantes para el aprendizaje de la política correcta. En los controladores propuestos se utilizan distintas funciones de recompensa, siempre en función del error de velocidad del generador.

$$r_t = f(\omega_{error}) \quad (25)$$

3.2.1. Implementación del entrenamiento de aprendizaje por refuerzo

El controlador se ha entrenado utilizando la librería Stable Baselines 3 (Raffin, 2021). Esta librería consta de un set de algoritmos de aprendizaje por refuerzo con una estructura unificada, para escoger el que más se adapte a cada aplicación, o para poder probar varios de forma rápida, al contar con la misma interfaz.

Para esta aplicación, se ha escogido el algoritmo DDPG, explicado en la sección 2.2.

Esta librería engloba solamente la política de control y su entrenamiento, y en el caso del algoritmo DDPG, también el del crítico. Pero, tanto el resto de la arquitectura de control de la



Figura 3.2, así como el modelo de simulación, deben englobarse como parte del entorno (“Gym”). Esta agrupación se observa en la Figura 3.3.

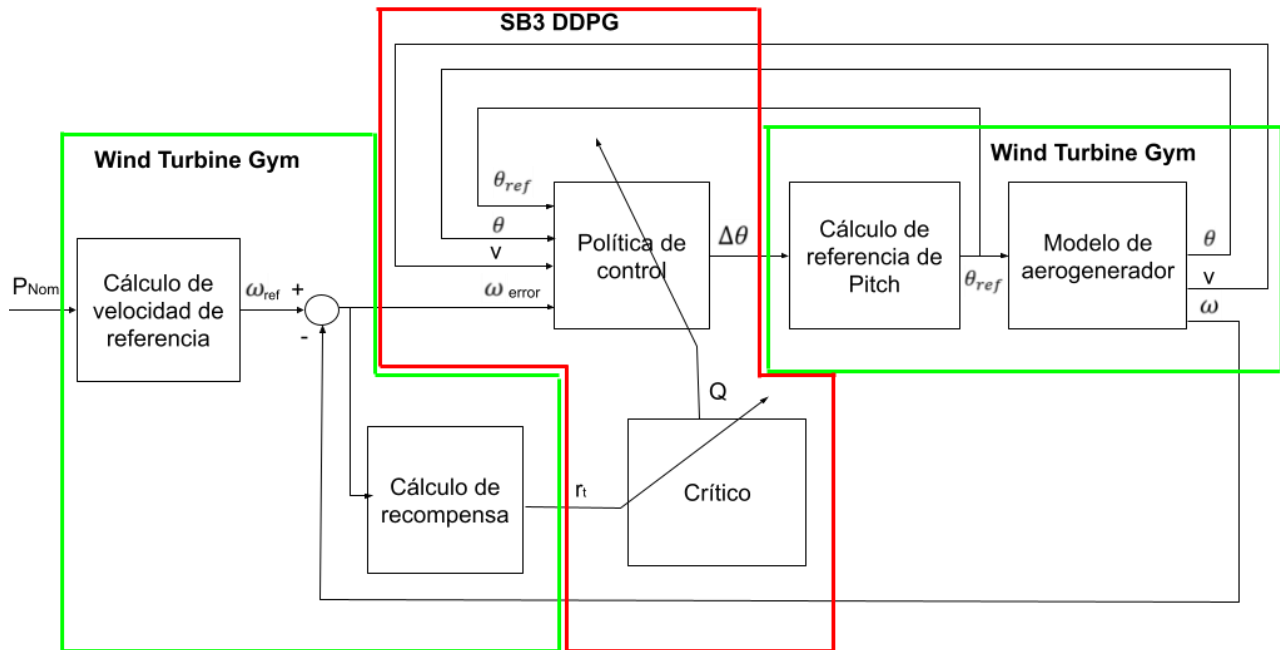


Figura 3.7: División de arquitectura de control para el entrenamiento RL

El “entorno”, bloque verde, engloba parte del controlador: cálculo de velocidad de referencia, el cálculo de la recompensa, y el cálculo de referencia de pitch, así como el modelo de simulación de aerogenerador propiamente dicho.

Las figuras 3.1 (diagrama de bloques de la arquitectura de software) y 3.7 (diagrama de bloques de la arquitectura del sistema de control) se relacionan de la siguiente manera:

- La función *map_inputs* = Cálculo de velocidad de referencia.
- La función *map_outputs* = Cálculo de referencia de pitch
- La función *reward* = Cálculo de recompensa

Capítulo 4

Control del aerogenerador

mediante aprendizaje por refuerzo

En este capítulo se realiza un estudio de las distintas configuraciones de control con aprendizaje por refuerzo implementadas para resolver el problema propuesto, explicando por qué se han introducido las sucesivas modificaciones que han ayudado a mejorar el desempeño del controlador.

Todos los controladores siguen una arquitectura similar a la presentada en el capítulo anterior y varían las variables de estado, o la recompensa utilizada, así como el tipo de viento durante el entrenamiento. En la Tabla 4.1 se resumen las principales diferencias entre los cuatro controladores propuestos.

Tabla 4.1. Diferencias entre los controladores propuestos.

	Controlador 1 RL-Sin Referencia	Controlador 2 RL-Básico	Controlador 3 RL-Recompensa integral	Controlador 4 RL-Integral
Estado	$[\theta, v, \omega_{error}]$	$[\theta, v, \omega_{error}, \theta_{ref}]$	$[\theta, v, \omega_{error}, \theta_{ref}]$	$[\theta, v, \omega_{error}, \theta_{ref}, \int_{t-T}^t \omega_{error}]$
Recompensa	$-\omega_{error}^2$	$-\omega_{error}^2$	$= - \left[(1 - K_1) \omega_{error}^2 + K_1 \left\ \int_{t-T}^t \omega_{error} \right\ \right]$	$= - \left[(1 - K_1) \omega_{error}^2 + K_1 \left\ \int_{t-T}^t \omega_{error} \right\ \right]$
Tipo de viento	Constante	Constante	Variable	Variable

Los parámetros del modelo DDPG han sido los mismos en todos los controladores, y se han escogido por semejanza con otros problemas de control ya resueltos con éxito mediante técnicas de aprendizaje por refuerzo, y que pueden encontrarse en la colección RL Baselines 3 Zoo (Raffin, 2020). Los hiperparámetros seleccionados se muestran en la Tabla 4.2.



Tabla 4.2. Hiper parámetros de DDPG utilizados para el control de pitch del aerogenerador.

Parámetro	Valor	Descripción
learning_rate	$1e^{-3}$	Tasa de aprendizaje del optimizador ADAM
buffer_size	$1e^{-5}$	Tamaño del replay buffer
learning_starts	500	Nº pasos recopilando experiencias antes entrenar
batch_size	256	Minibatch para cada actualización del gradiente
gamma	0.98	Factor de descuento de recompensas futuras
train_freq	1	Actualización del modelo cada train_freq pasos
gradient_dsteps	1	Nº de actualizaciones de gradiente cada train_freq
action_noise	0.1	Ruido de la acción para explorar

4.1. Controlador 1. RL-Sin Referencia

Variables de estado

Las variables de estado de este controlador son el error de velocidad del rotor, la velocidad del viento, y el ángulo de las palas.

Tabla 4.3. Variables de estado del Controlador 1

Variable	Definición	Unidad	Rango
$\theta(t)$	Angulo de pitch actual	rad	$[0, \pi]$
$v(t)$	Velocidad del viento	m/s	$[12, 13]$
$\omega_{error}(t)$	Error de velocidad del rotor	rad/s	$[-10, 10]$

Recompensa

Error cuadrático de la velocidad del rotor

$$r_t = -\omega_{error}^2 \quad (26)$$

Arquitectura

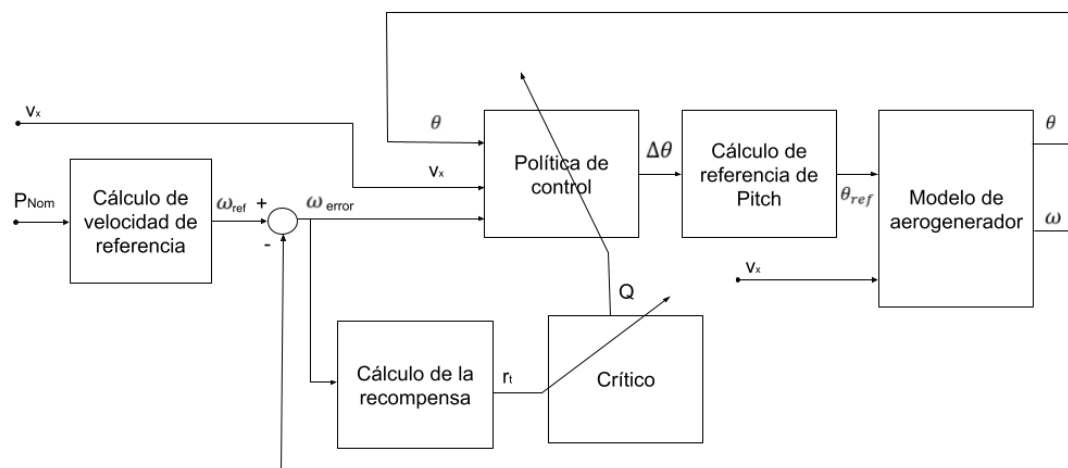


Figura 4.1: Arquitectura de control del Controlador 1.

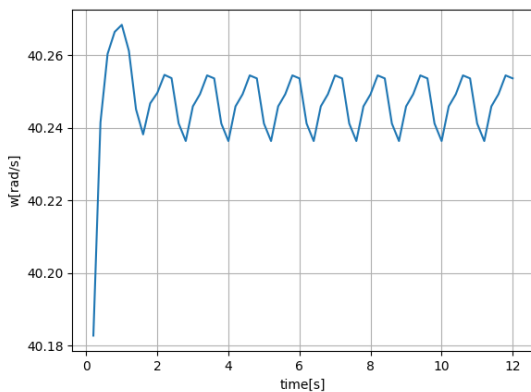
Entrenamiento

Estos primeros controladores se entrena únicamente con viento constante de 12.3 m/s para simplificar el problema. Una vez el controlador es capaz de realizar el seguimiento correctamente, se puede continuar el entrenamiento en escenarios más complejos, con viento variable. A esta técnica se le conoce como aprendizaje por currículo.

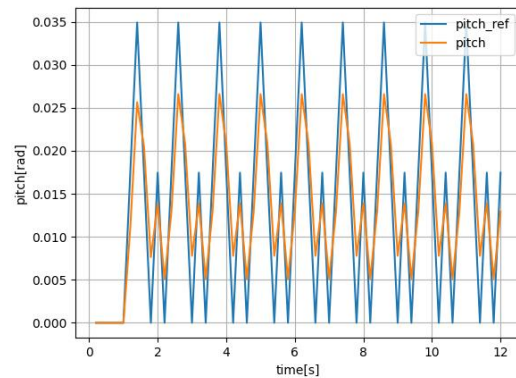
Se ha entrenado durante 8.000s de simulación (40.000 pasos de control y entrenamiento), lo cual ha requerido 22 minutos de computación (x6) en un portátil de gama media con Intel Core i7. Se ha entrenado con los parámetros indicados en la Tabla 4.1.

Simulación de validación

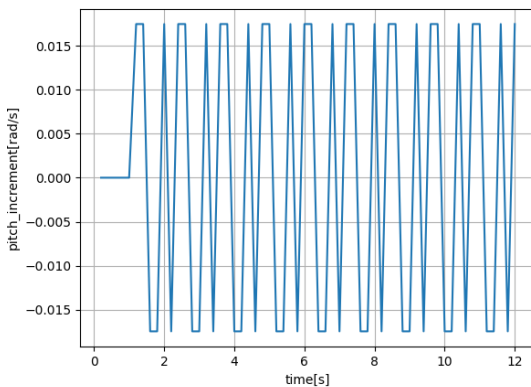
Tras el entrenamiento, se realiza una simulación de validación con viento constante igual al de entrenamiento (12.3 m/s). Los resultados se observan en la Figura 4.2



a) Velocidad del rotor



b) Angulo de pitch



c) Acción de control

Figura 4.2: Simulación de validación Controlador 1

La Figura 4.2. a) muestra cómo el Controlador 1 ha aprendido a llevar una velocidad que oscila alrededor de un punto cercano al de consigna, de 40 rad/s. No obstante, tiene un offset respecto al valor deseado, además de una oscilación tampoco deseable.



En la figura 4.2. c) se puede ver que el controlador solamente realiza acciones máximas o mínimas, y por ello la velocidad resultante es oscilatoria. El máximo pitch incremental en cada paso de simulación se calcula cómo el ángulo máximo de palas permitido por segundo, multiplicado por el paso de simulación. En este modelo, se ha escogido una variación máxima de 5°/s, y un paso de simulación de 0.2s:

$$\Delta\theta_{max} = 5 \cdot \frac{\pi}{180} \cdot 0.2 = 0.01745 \text{ rad}$$

En la gráfica 4.2. b) se observan las rampas del ángulo de palas de referencia (línea azul), resultado de acumular el pitch incremental de la Figura 4.2. c), así cómo el ángulo de palas real (línea roja), cuyos picos son menores que la referencia, ya que la oscilación no le da tiempo a estabilizarse.

Se sospecha que esta oscilación puede estar favorecida por el retardo en la aplicación de la acción debido a la dinámica del rotor, y el hecho de que se acumule la acción incremental sobre una variable, θ_{ref} , que es desconocida para el Controlador 1. Por ello, dicha variable se añadirá al conjunto de variables de estado en el Controlador 2.

4.2. Controlador 2. RL-Básico

Variables de estado

A las variables de estado del Controlador 1 se añade la referencia de ángulo de palas θ_{ref} . Es decir, la acción final del controlador.

Tabla 4.4. Variables de estado del Controlador 2

Variable	Definición	Unidad	Rango
$\theta(t)$	Angulo de pitch actual	rad	$[0, \pi]$
$v(t)$	Velocidad del viento	m/s	$[12, 13]$
$\omega_{error}(t)$	Error de velocidad del rotor	rad/s	$[-10, 10]$
$\theta_{ref}(t)$	Referencia de pitch anterior	rad	$[0, \pi]$

Recompensa

Igual que Controlador 1. Error cuadrático de velocidad del rotor

$$r_t = -\omega_{error}^2 \quad (26)$$

Arquitectura

En la Figura 4.2 se muestra la arquitectura del Controlador 2, con la nueva variable de estado.

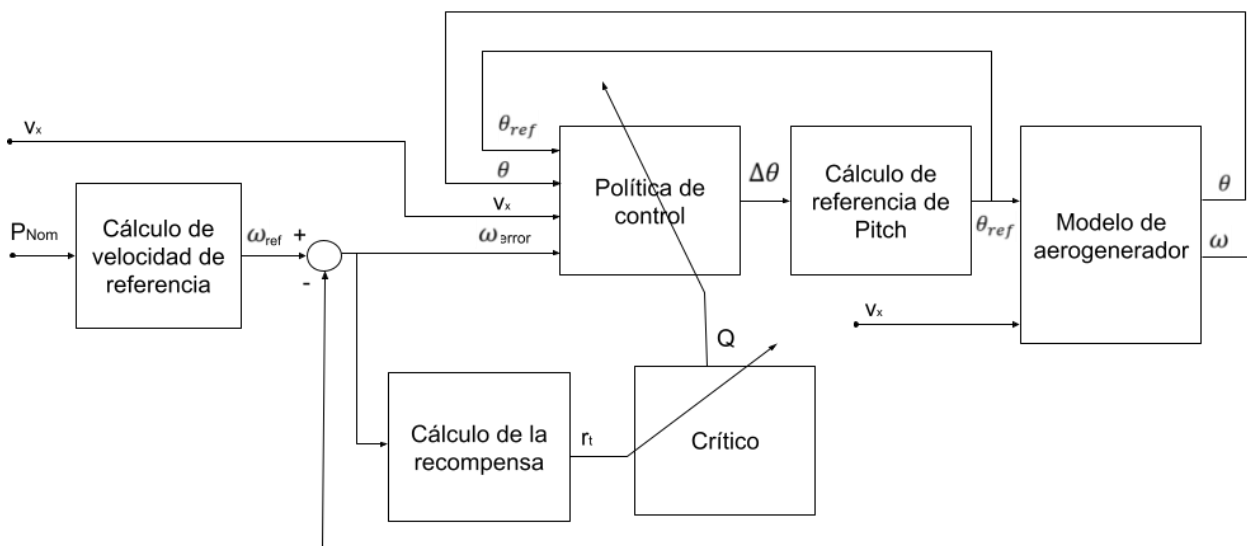


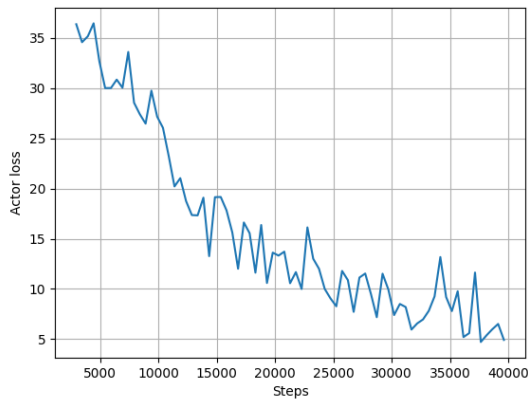
Figura 4.2: Arquitectura de control del Controlador 2.

Entrenamiento

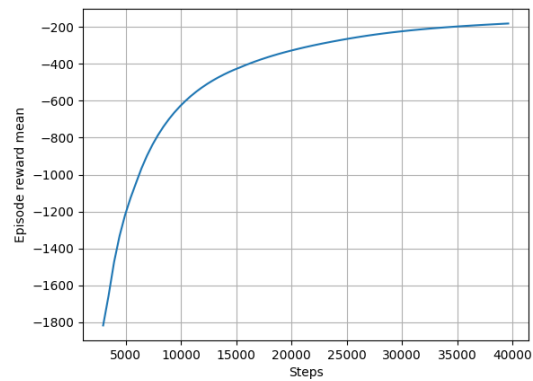
En este controlador se entrena únicamente con viento constante de 12.3 m/s.

Se ha entrenado durante 8.000s de simulación (40.000 pasos de control y entrenamiento).

En la figura 4.3 a) se aprecia la evolución descendente del coste del actor, y en 4.3 b) se observa el incremento de la recompensa media en cada episodio.



a) Coste



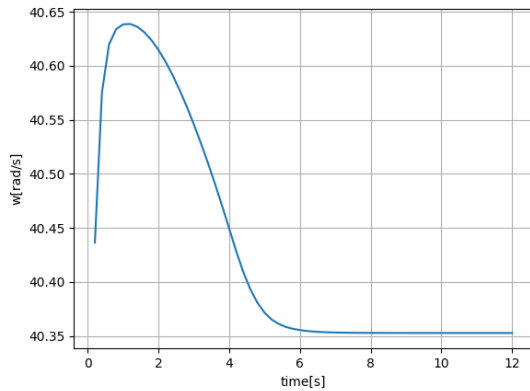
b) Recompensa

Figura 4.3: Métricas del entrenamiento Controlador 2.

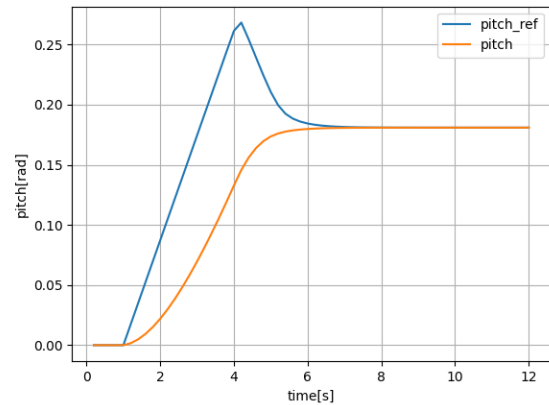
La tendencia de ambas curvas de la Figura 4.3 indica que el controlador está aprendiendo correctamente. Empezando por la 4.3-b, la media de la recompensa en cada episodio es cada vez mayor a lo largo del entrenamiento. De hecho, parece que la tendencia todavía tiene recorrido, pero se ha probado a continuar el entrenamiento desde este punto y sin embargo se produce un empeoramiento de la recompensa. En la gráfica izquierda (4.3-a) se muestra el coste del actor a lo largo del entrenamiento y se observa cómo va reduciéndose, lo cual es el comportamiento deseado, minimizar dicho coste.

Simulación de validación

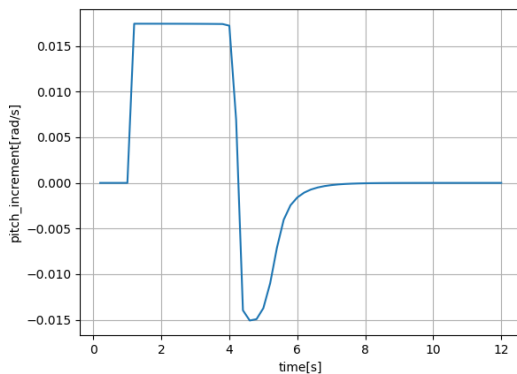
Se realiza la misma simulación de validación que en el controlador anterior, con viento constante de 12.3 m/s. El resultado puede observarse en las gráficas de la Figura 4.4.



a) Velocidad del rotor



b) Angulo de pitch



d) Acción de control

Figura 4.4: Simulación de validación Controlador 2

En la Figura 4.4. a) se observa la velocidad del rotor, la cual tras un transitorio inicial, es capaz de estabilizarse en una velocidad cercana a la de referencia. No obstante, la velocidad de referencia es de 40 rad/s, por lo que hay un error estacionario de 0.35 rad/s.

En la Figura 4.4. b) se observa la evolución del ángulo de palas de referencia (línea azul) y real (línea naranja). Para un viento dado, hay un ángulo de palas en estacionario que permite llevar la velocidad del rotor a la deseada. Sin embargo, es destacable cómo el ángulo de referencia de palas (línea azul) no se estabiliza en dicho punto (0.18 rad) y espera a que le alcance el ángulo de palas según la dinámica de pitch, sino que continúa aumentando, para después disminuir hasta el valor de estacionario, realizando el pico en el segundo 4.

Los resultados de este controlador son prometedores, ya que consigue realizar una acción de control coherente y sin oscilaciones, llevando la velocidad del rotor a una cercana a la de referencia. Sin embargo, debe corregirse el error permanente, así como probar que generaliza con viento variable.

Los resultados de este controlador se presentaron en el congreso "VI WWWE 2024 Workshop en Energía Eólica y Marina", con el título del artículo "Control inteligente del ángulo de palas de un aerogenerador mediante aprendizaje por refuerzo"



Análisis del error permanente

Para comprobar si el error permanente es debido a que no se ha explorado suficientemente la zona sin error, se grafica el histograma del error durante el entrenamiento.

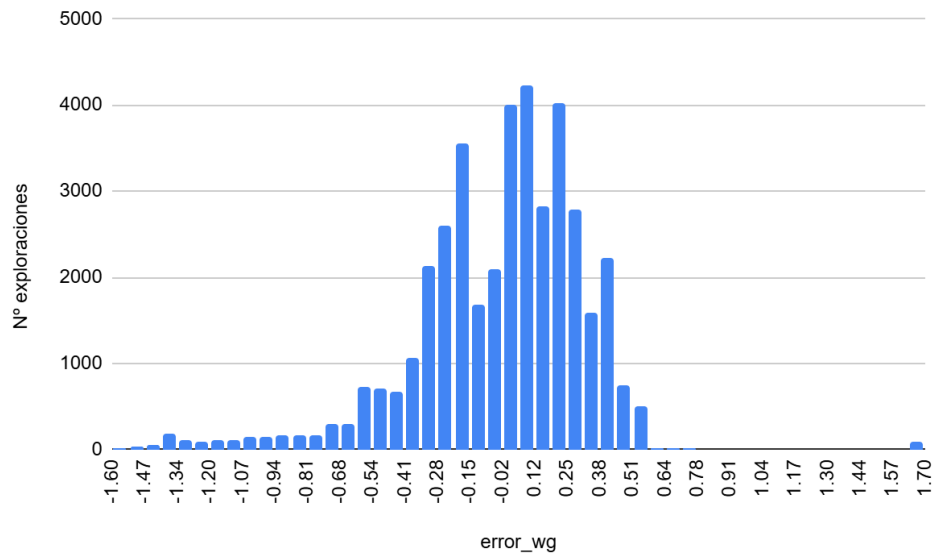


Figura 4.5: Histograma de exploración del error. Controlador 2.

Como se observa en la figura 4.5, sí se ha explorado correctamente esa zona, lo que significa que no se ha recompensado correctamente el anular el error permanente. Esto se intentará corregir en el Controlador 3.

4.3. Controlador 3. RL-Recompensa Integral

VARIABLES DE ESTADO

Se mantienen las variables de estado del Controlador 2.

Table 4.5. Variables de estado del Controlador 1

Variable	Definición	Unidad	Rango
$\theta(t)$	Angulo de pitch actual	rad	$[0, \pi]$
$v(t)$	Velocidad del viento	m/s	$[12, 13]$
$\omega_{error}(t)$	Error de velocidad del rotor	rad/s	$[-10, 10]$
$\theta_{ref}(t)$	Referencia de pitch anterior	rad	$[0, \pi]$

RECOMPENSA

Como en los controladores anteriores se observaba un error estacionario que no mejora con el entrenamiento, se decide incluir un término de la integral del error en la recompensa. Para evitar que el término integral de la recompensa crezca indefinidamente y domine la función de coste, se ha optado por utilizar una **integral móvil** del error de velocidad angular, calculada sobre una ventana de longitud T . Esta aproximación permite capturar el comportamiento reciente del sistema y penalizar errores sostenidos a corto plazo, sin afectar negativamente el aprendizaje a largo plazo. Una integral durante todo el episodio podría penalizar de forma desproporcionada errores antiguos que ya no son relevantes y dificultar la convergencia del controlador.

$$r_t = - \left[(1 - K_1) \omega_{error}^2 + K_1 \left\| \int_{t-T}^t \omega_{error} \right\| \right] \quad (27)$$

Se desea que el término integral tenga más influencia hacia el final del episodio, cuando debe anularse el error estacionario, por lo que se pondera mediante una función exponencial del tiempo.

$$K_1 = K_{max} - K_{max} e^{-\frac{t}{T}} \quad (28)$$

ARQUITECTURA

Igual que Controlador 2.

ENTRENAMIENTO

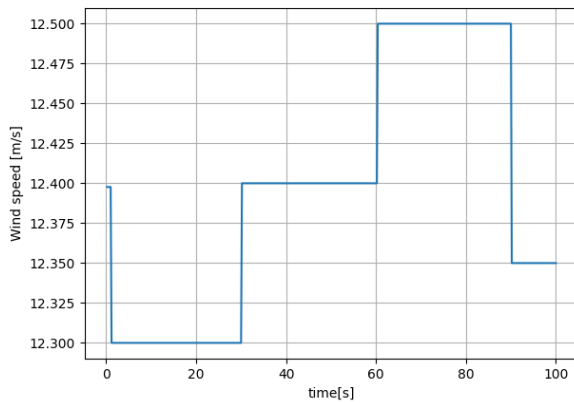
El entrenamiento se realiza con una velocidad de viento escogida al inicio del episodio según una distribución uniforme entre 12 y 13 m/s, y que se mantiene constante durante todo el episodio. El objetivo de mantener la velocidad de viento constante durante cada episodio es que el algoritmo pueda recoger un número suficiente de datos para cada velocidad de viento.

Se ha entrenado durante 8.000s de simulación (40.000 pasos de control y entrenamiento).

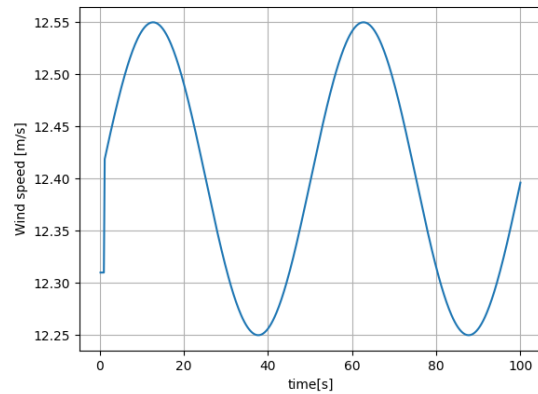


Simulación de validación

Se realizan dos simulaciones de validación. Una con un perfil de entrada de escalones de viento, Figura 4.6. a), y otra con un perfil de viento sinusoidal, Figura 4.6. b).



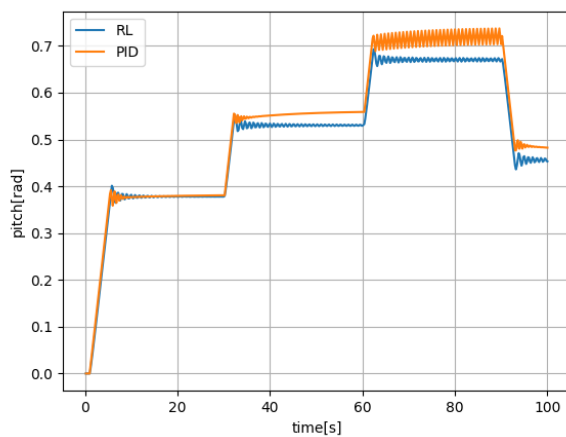
a) Perfil de velocidad de viento escalonada



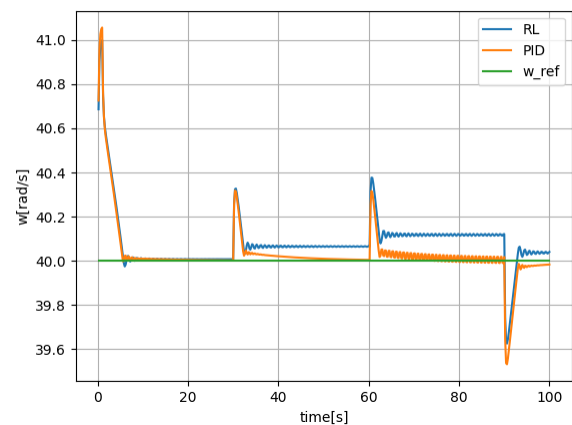
b) Perfil de velocidad de viento sinusoidal

Figura 4.6: Viento en las dos simulaciones de validación

En la Figura 4.7 se muestra la simulación con escalones de viento, comparada además con un controlador PI de referencia (línea naranja), con ganancias: $K_p = 1$, $K_i = 0.1$, $K_d = 0$. Dicho controlador se ha sintonizado mediante prueba y error.



a) Ángulo de pitch



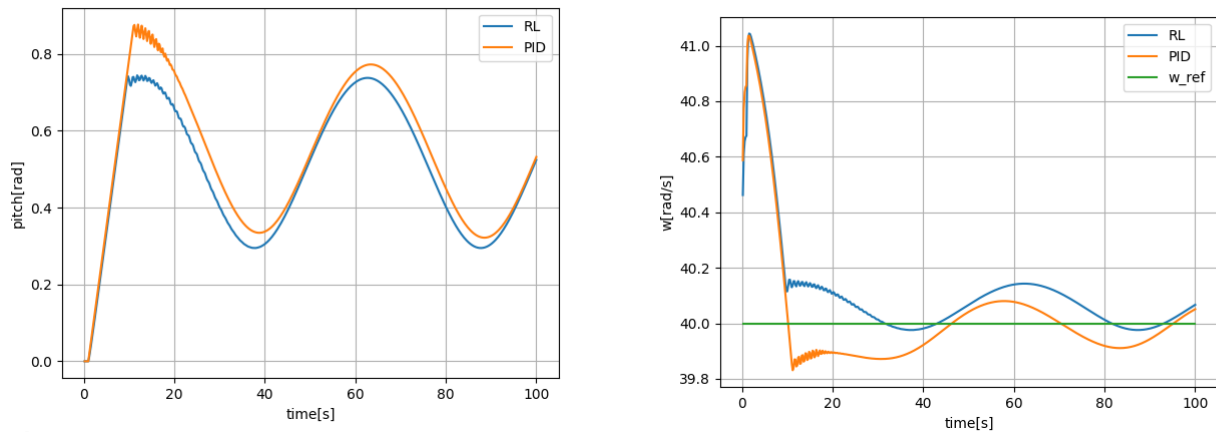
b) Velocidad del rotor

Figura 4.7: Resultados simulación escalones de viento. Controlador 3

Como se puede observar, con viento a velocidad de 12.3 m/s, se consigue un error muy cercano a 0, mejorando el comportamiento del Controlador 2. Además, generaliza bastante bien, aprendiendo a realizar el seguimiento para vientos diferentes dentro de la misma simulación. Sin embargo, para los vientos más altos, el error aumenta, y el comportamiento del controlador todavía es inferior al del PI de referencia.

En la Figura 4.8 se muestran los resultados de la simulación con perfil de viento sinusoidal (línea azul). El controlador realiza las acciones necesarias para llevar un ángulo de palas también

sinusoidal, que intenta compensar al viento y de hecho consigue llegar a una velocidad cercana a la de referencia, aunque sin conseguir estabilizar por completo la velocidad. El comportamiento es muy similar al del controlador PID de referencia (línea naranja).



a) Ángulo de pitch

b) Velocidad del rotor

Figura 4.8: Resultados simulación viento sinusoidal. Controlador 4

Los resultados de este controlador se presentaron en el congreso “ECRES 2025. European Conference on Renewable Energy Systems”, con el título del artículo: “First Approximation of Reinforcement Learning Applied to a Wind Turbine Pitch Control.”



4.4. Controlador 4. RL- Integral

Variables de estado

El Controlador 3 mejoró el error en estado estacionario, pero no lo resuelve completamente. Emulando el concepto de un controlador PI, se añade la integral del error como variable de estado.

Table 4.6. Variables de estado del Controlador 4

Variable	Definición	Unidad	Rango
$\theta(t)$	Angulo de pitch actual	rad	$[0, \pi]$
$v(t)$	Velocidad del viento	m/s	$[12, 13]$
$\omega_{error}(t)$	Error de velocidad del rotor	rad/s	$[-10, 10]$
$\theta_{ref}(t)$	Referencia de pitch anterior	rad	$[0, \pi]$
$\int_{t-T}^t \omega_{error}$	Integral del error	rad/s	$[0, 20]$

Si se escogiese una tipología de red neuronal con memoria, distinta del MLP, quizá podría llegar a aprender las dependencias temporales por sí misma, pero al no tenerlo, la única forma de forzarlo es añadiéndolo en las variables de estado.

Recompensa

Se mantiene la misma recompensa que en el Controlador 3.

$$r_t = - \left[(1 - K_1) \omega_{error}^2 + K_1 \left\| \int_{t-T}^t \omega_{error} \right\| \right] \quad (27)$$

$$K_1 = K_{max} - K_{max} e^{-\frac{t}{T}} \quad (28)$$

Arquitectura

Se añade la integral del error como variable de estado (Figura 4.9)

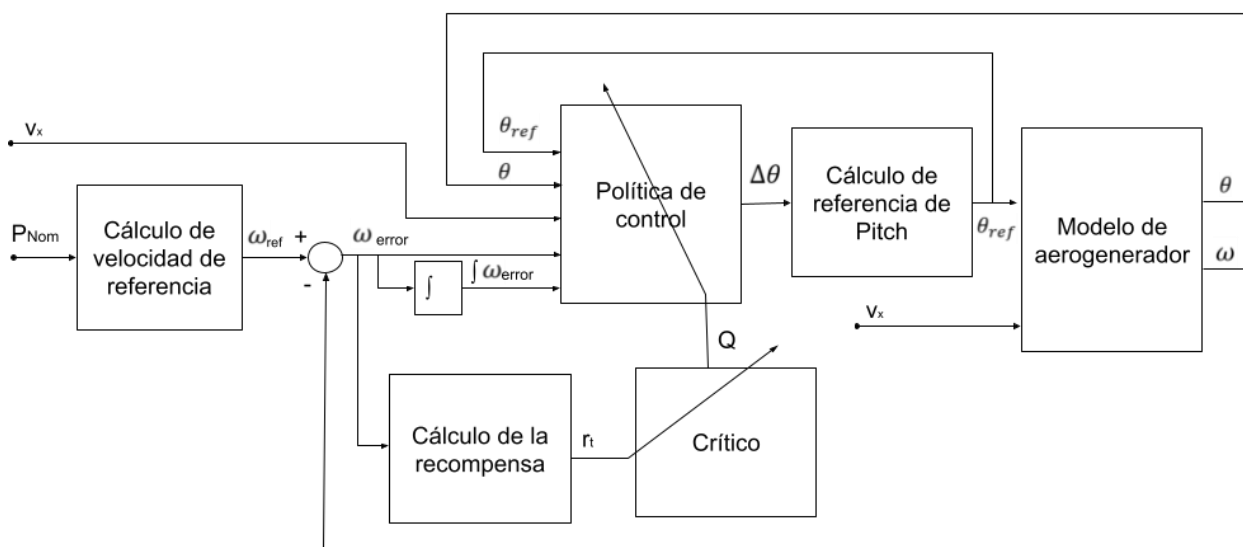


Figura 4.9: Arquitectura Controlador 4

Entrenamiento

Al igual que en el Controlador 3, el entrenamiento se realiza con viento constante durante un episodio, pero diferente en cada episodio.

Inicialmente, se ha entrenado durante 8.000s de simulación (40.000 pasos de control y entrenamiento), como en los controladores anteriores, pero el resultado obtenido era peor que en el Controlador 3, e incluso que el Controlador 2. No obstante, con el número de variables de estado aumenta la complejidad del problema del entrenamiento, por lo que es razonable que se necesite un mayor número de muestras para dar buenos resultados.

Se ha reentrenado la red neuronal con 3 entrenamientos adicionales de 4.000s cada uno, hasta que se ha observado que el comportamiento dejaba de mejorar.

En la figura 4.10 se muestra la evolución de los resultados del Controlador 4 con el tiempo de entrenamiento.

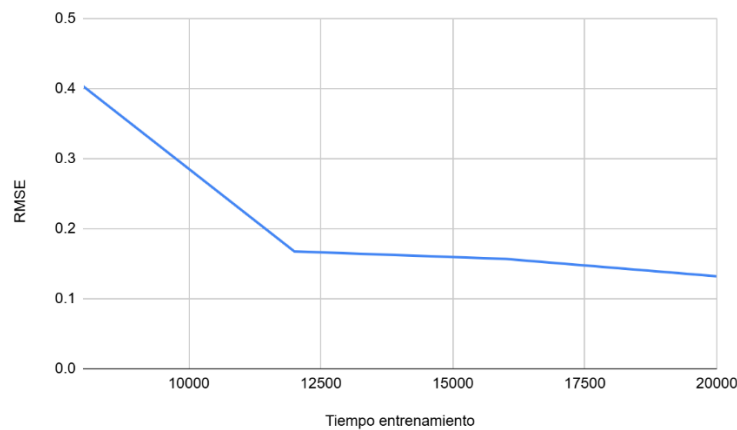
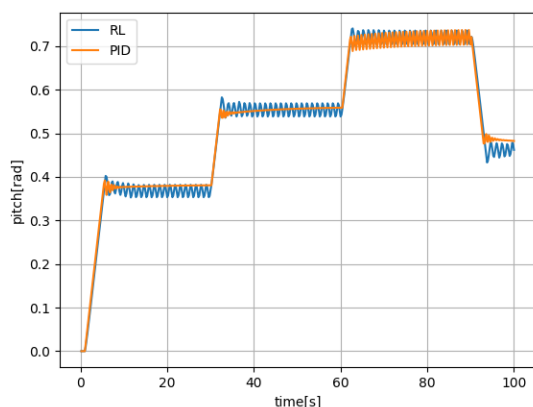


Figura 4.10: Evolución del RMSE Controlador 4

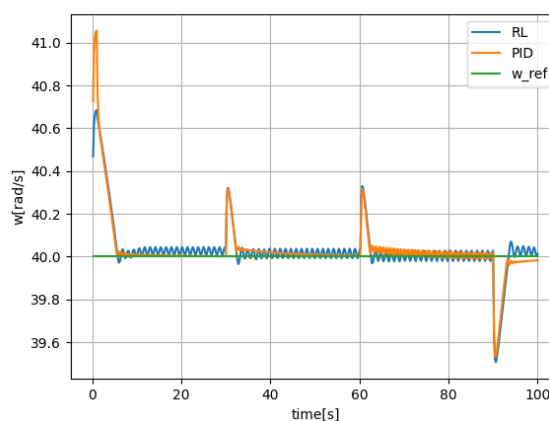
En la gráfica 4.10. se muestra el RMSE (root mean square error) del Controlador 4 después de cada entrenamiento, calculado en la misma simulación de validación con el perfil de viento escalonado de la Figura 4.7. Se observa una gran mejoría con el segundo re-entrenamiento, y en menor medida con el tercero y cuarto re-entrenamientos.

Simulación de validación

En este apartado se muestran los resultados obtenidos mediante el Controlador 4, tras los 4 entrenamientos.

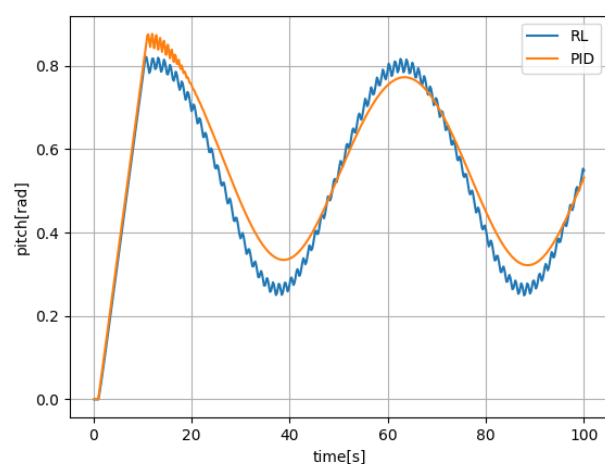


a) Angulo de pitch

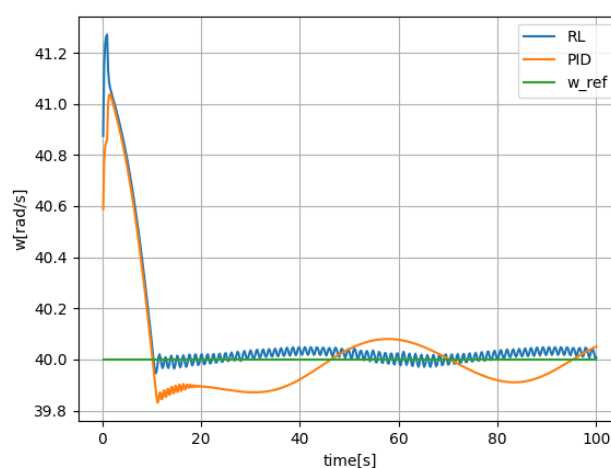


b) Angulo de pitch

Figura 4.11: Simulación del perfil de viento escalonado con el Controlador 4



a) Angulo de pitch



b) Angulo de pitch

Figura 4.12: Simulación del perfil de viento sinusoidal con el Controlador 4

Como se puede observar, tanto en la figura 4.9 como en la 4.10, el Controlador 4 (línea azul) consigue eliminar el error permanente para todas las condiciones de viento, mejorando el resultado del Controlador 3, así como el del PI (línea naranja), especialmente en el caso del viento sinusoidal. No obstante, el controlador resultante presenta una oscilación de poca amplitud.

4.5. Comparativa de los controladores

En resumen, se han implementado cuatro controladores mediante aprendizaje por refuerzo, que han ido mejorando sus resultados gracias a las modificaciones realizadas, hasta superar el controlador PI de referencia.

El Controlador 1 consiguió aprender a realizar una acción de seguimiento coherente, aunque oscilatoria. El Controlador 2 resuelve la oscilación del controlador, pero tiene error permanente. Este error se corrige en el Controlador 3, al añadir la integral del error de velocidad en la recompensa. Además, se entrena con éxito en viento variable. Finalmente, en el Controlador 4 se añade la integral del error también en el estado, y se consigue mejorar el comportamiento en todas las condiciones de viento, superando a los controladores antes planteados, así como al controlador PI de referencia

En la Tabla 4.5. se muestra el RMSE, de los Controladores 2 al 4, en los dos escenarios de viento planteados en la Figura 4.6, con escalones de viento y con viento sinusoidal. El Controlador 1 no se ha incluido en la comparativa ya que su comportamiento no era bueno con viento constante, y no se han realizado más simulaciones con él.

El RMSE se calcula según la ecuación (29), donde N es el número de pasos de control en la simulación:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\omega_i - \omega_{ref})^2} \quad (29)$$

Tabla 4.7. Comparativa RMSE

Controlador	RMSE Escalera	RMSE Sinusoidal
PID	0.1473	0.2392
Controlador 2	0.3543	1.0830
Controlador 3	0.1605	0.3991
Controlador 4	0.1320	0.2303



Capítulo 5

Conclusiones y trabajos futuros

En este Trabajo de Fin de Máster se ha explorado la aplicación del aprendizaje por refuerzo para el control del ángulo de palas en aerogeneradores, con el objetivo de mejorar la regulación de la velocidad del rotor ante las variaciones del viento, permitiendo mantener constante la potencia de salida de la turbina en la región de carga nominal. Se ha demostrado la viabilidad de utilizar algoritmos de aprendizaje por refuerzo, específicamente DDPG, para diseñar controladores en los que, mediante experimentación, es capaz de aprender la ley de control, proporcionando una mejor reactividad ante cambios del sistema, incertidumbres y dinámicas no modeladas.

Los resultados obtenidos en las simulaciones muestran que los controladores basados en aprendizaje por refuerzo son capaces de generar ajustes precisos en el ángulo de las palas, mejorando los resultados de un controlador PI convencional. Esto subraya el potencial del aprendizaje por refuerzo como una herramienta eficaz para el control de aerogeneradores, ofreciendo la posibilidad de optimizar la generación de energía.

Tanto el campo de la energía eólica como el aprendizaje por refuerzo tienen un gran potencial, por lo que hay diversos trabajos futuros que podrían realizarse partiendo de la base de este trabajo.

En primer lugar, se podría continuar el entrenamiento del controlador en un simulador de mayor fidelidad como OpenFAST. Esto puede utilizarse para realizar directamente el entrenamiento del controlador en un entorno más realista, o para estudiar la capacidad de transferencia del controlador entrenado con un modelo sencillo y validado sobre uno más complejo. En el segundo caso, serviría para emular la posterior transferencia del controlador en un aerogenerador real, entendiendo cómo afectan las aproximaciones del entorno de simulación a la capacidad del controlador resultante.

Por otro lado, se puede continuar experimentando con otros algoritmos de aprendizaje por refuerzo para el mismo problema de control del ángulo de palas, evaluando tanto el rendimiento obtenido de cada uno como comparando el coste computacional de las distintas alternativas.

Además, en este trabajo se ha manejado una sola salida de control, pero el aerogenerador es un sistema multivariable y con las técnicas de aprendizaje por refuerzo se podría estudiar la posibilidad de unificar en un solo controlador varios modos de operación, manejando varias acciones de control, como el ángulo de palas y el par del generador al mismo tiempo.



Bibliografía

- Bossanyi, E. A. (2003). Wind turbine control for load reduction. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 6(3), 229-244.
- Brockman, G., Cheung V., Pettersson, L, Schneider, J. Schulman,J., Tang, J. and Zaremba, W. Openai gym. arXiv preprint arXiv:1606.01540, 2016.
- Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.
- Elkodama, A., Ismaiel, A., Abdellatif, A., Shaaban, S., Yoshida, S., & Rushdi, M. A. (2023). Control methods for horizontal axis wind turbines (HAWT): State-of-the-art review. *Energies*, 16(17), 6394.
- Eneredata, (2024). <https://datos.enerdata.net/energia-total/datos-consumo-internacional.html> [Online accessed 08/05/2025];
- Haarnoja,T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Offpolicy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.
- International Energy Agency (2024). <https://www.iea.org/>
- Lillicrap, T. P., Hunt, J. P., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.
- López-Queija, J., Jugo, J., Tena, A., Robles, E., & Sotomayor, E. (2024). Floating offshore wind turbine nonlinear model predictive control optimisation method. *Ocean Engineering*, 314, 119754.
- Manwell, J.F., McGowan, J.G., Rogers, A.L., 2010. *Wind Energy Explained: Theory, Design and Application*. Wiley.
- MathWorks. Reinforcement learning, 2023. [Online; accessed 12/04/2025].
- Mikati, M., Santos, M., & Armenta, C. (2013). Electric grid dependence on the configuration of a small-scale wind and solar power hybrid system. *Renewable energy*, 57, 587-593.
- Muñoz-Palomeque, E., Sierra-García, J. E., & Santos, M. (2024). Enhancing Offshore Wind Turbines Performance with Hybrid Control Strategies Using Neural Networks and Conventional Controllers. *Journal of Computational Design and Engineering*, qwae103.
- Narayanan, V. L., Dhaked, D. K., & Sitharthan, R. (2024). Improved machine learning-based pitch controller for rated power generation in large-scale wind turbine. *Renewable Energy Focus*, 50, 100603.
- OpenAI. Openai spinning up reinforcement learning, 2020. <https://spinningup.openai.com/en/latest/user/introduction.html> [Online; accessed 12/04/2025].



Pandit, R., Infield, D., & Santos, M. (2022). Accounting for environmental conditions in data-driven wind turbine power models. *IEEE Transactions on Sustainable Energy*, 14(1), 168-177.

Raffin, A. (2020). RL Baselines3 Zoo. GitHub Repository. Retrieved from <https://github.com/DLR-RM/rl-baselines3-zoo>

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268), 1–8. Retrieved from <http://jmlr.org/papers/v22/20-1364.html>

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET), pages 1–6. Ieee, 2017.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Serrano, C., Sierra-García, J. E., & Santos, M. (2022). Hybrid optimized fuzzy pitch controller of a floating wind turbine with fatigue analysis. *Journal of Marine Science and Engineering*, 10(11), 1769.

Shakya, A. K., Pillai, G., & Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231, 120495.

Sherstinsky, Alex. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

Sierra-García, J. E., & Santos, M. (2020). Exploring reward strategies for wind turbine pitch control by reinforcement learning. *Applied Sciences*, 10(21), 7462.

Sierra-García, J. E., & Santos, M. (2020). Performance analysis of a wind turbine pitch neurocontroller with unsupervised learning. *Complexity*, 2020(1), 4681767.