



UNIVERSIDAD NACIONAL
DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de
Ingeniería Informática



UNIVERSIDAD
COMPLUTENSE DE MADRID

Facultad de
Informática

APRENDIZAJE FEDERADO Y Q-LEARNING PARA EL CONTROL DE TURBINAS EÓLICAS: OPTIMIZACIÓN MEDIANTE DISCRETIZACIÓN DE ESTADOS

Alberto Gil Maciá

Director/a: Dr. Matilde Santos Peñas

Co-Director/a: Dr. Jesús Enrique Sierra García

Trabajo de Fin de Máster

Máster Universitario
en Ingeniería de Sistemas Y Control

Curso 2024-2025 / Convocatoria de Junio

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 that reads "Alberto". The signature is written in a cursive style with a long horizontal stroke at the end.

Alberto Gil Maciá

Resumen

La energía eólica ha experimentado una gran transformación tecnológica en los últimos años. El empleo de turbinas de mayor tamaño, el uso de materiales compuestos y la mejora en los sistemas de transmisión han permitido que los aerogeneradores puedan producir grandes cantidades de energía de forma eficiente, consolidándose así como una de las principales fuentes de energía renovable.

Sin embargo, el diseño de controladores eficientes que permitan maximizar la potencia extraída del viento a la vez que se minimizan las cargas estructurales sobre el sistema sigue siendo un desafío de control, donde las dinámicas no lineales, el acoplamiento entre variables, la incertidumbre del modelo y las perturbaciones externas hacen necesario el empleo de técnicas de control avanzadas. Los controladores basados en aprendizaje reforzado (RL, del inglés, Reinforcement Learning) han demostrado grandes capacidades en esta área, pero su aplicación práctica se ha visto limitada debido a la cantidad de datos requeridos para su entrenamiento.

En este trabajo se desarrolla un controlador que combina Q-learning con Aprendizaje Federado (FL, del inglés, Federated Learning) con el objetivo de maximizar la potencia de salida de una turbina eólica, abordando los siguientes temas:

- Estudio de funciones de recompensa para minimizar la actuación del controlador a la vez que se maximiza la producción de energía.
- Propuesta de métodos de discretización del espacio de estados para reducir el tiempo de entrenamiento y disminuir el error en el estado estacionario.
- Empleo de técnicas de aprendizaje federado en una granja eólica para el aprendizaje colaborativo de un modelo global sin compartir datos sensibles en el proceso.

Las técnicas propuestas son simuladas, demostrando que el empleo de las técnicas de discretización de estados reduce significativamente el tiempo de entrenamiento y mejora el error en el estado estacionario. Además, el empleo de aprendizaje federado mejora significativamente el tiempo de convergencia del controlador en comparación con las estrategias de control sin aprendizaje federado. Los resultados demuestran cómo el aprendizaje federado tiene un gran potencial para mejorar la eficacia de los controladores de turbinas eólicas, manteniendo la privacidad y seguridad de los datos empleados en el entrenamiento.

Palabras clave: Q-Learning, Aprendizaje Reforzado, Control Inteligente, Aprendizaje Federado, Energía eólica.

Abstract

Wind energy has undergone a major technological transformation in recent years, where larger wind turbines, the use of composite materials, and improved transmission systems have enabled wind turbines to produce large amounts of energy efficiently, thus consolidating their position as one of the main sources of renewable energy.

However, the design of efficient controllers that maximize the power extracted from the wind while minimizing structural loads on the system remains a challenge, where non-linear dynamics, coupling between variables, model uncertainty, and external disturbances require the use of advanced control techniques. Reinforcement learning (RL) based controllers have demonstrated great capabilities in this area, but their practical application has been limited due to the amount of data required for training.

In this work, a controller which combines Q-learning and federated learning (FL) is developed with the objective of maximizing the power output of a wind turbine, addressing the following key aspects:

- Study of reward functions to minimize control actuation while maximizing energy production.
- Proposal of state-space discretisation methods to reduce the training time and minimize the error in the steady-state.
- Use of federated learning (FL) techniques in a wind farm for collaborative learning of a global model without sharing sensitive data in the process.

The proposed techniques are simulated, demonstrating that the use of state discretisation techniques significantly reduces the training time and improves the steady-state error. Furthermore, the use of Federated Learning significantly improves the convergence time of the controller when compared to control strategies without Federated Learning. Results demonstrate how Federated Learning has great potential to improve the efficiency of wind turbine controllers while maintaining the privacy and security of the data used in training.

Keywords: Q-Learning, Federated Learning, Intelligent Control, Reinforcement Learning, Wind Turbine.

Glosario de símbolos

P_{error} Diferencia entre la potencia eléctrica generada por la turbina y su potencia nominal (W).

P_{out} Potencia eléctrica generada por la turbina (W).

P_{rated} Potencia nominal de la turbina en condiciones de operación estandar (W).

R Función de recompensa en aprendizaje por refuerzo, empleada para determinar el rendimiento inmediato de una acción en el entorno (adimensional).

V_{wind} Velocidad del viento incidente en el rotor de la turbina (m/s).

α Tasa de aprendizaje en Q-Learning. Define el peso de nuevas experiencias en la actualización de la tabla Q (adimensional).

ϵ Tasa de exploración en Q-Learning. Define la probabilidad de que el agente tome una acción aleatoria durante el proceso de aprendizaje (adimensional).

η Parámetro de escala en la distribución de Weibull (adimensional).

γ Factor de descuento en Q-Learning. Define la importancia de las recompensas futuras, frente a las inmediatas (adimensional).

γ_ϵ Tasa de decaimiento en Q-Learning. Define como disminuye la tasa de exploración con el tiempo (adimensional).

λ Relación de velocidad de punta. Ratio entre la velocidad de las puntas de las palas y la velocidad del viento (adimensional).

θ Ángulo de paso de las palas de la turbina ($^\circ$).

θ_{opt} Ángulo de paso que maximiza la eficiencia de la turbina para unas condiciones de operación determinadas ($^\circ$).

k Parámetro de forma en la distribución de Weibull (adimensional).

Índice general

Glosario de símbolos	IX
1. Introducción	1
1.1. Propuesta y objetivos	2
1.2. Estructura del documento	4
1.3. Publicaciones derivadas	4
2. Fundamentos Teóricos y Estado del Arte	7
2.1. Funcionamiento básico de una turbina eólica	7
2.2. Regiones de operación de una turbina eólica	8
2.2.1. Curvas de potencia óptimas	8
2.3. Estrategias de control de una turbina eólica	10
2.3.1. Control por pérdida aerodinámica	10
2.3.2. Control de paso	11
2.3.3. Estrategias de control basadas en aprendizaje reforzado	16
2.3.4. Aplicación de aprendizaje reforzado en el control de turbinas eólicas .	25
2.4. Aprendizaje federado	26
2.4.1. Paradigmas de aprendizaje federado.	27
2.4.2. Estrategias de aprendizaje federado	27
2.4.3. Aplicación del aprendizaje federado en el control de turbinas eólicas .	30
2.5. Modelo de turbina eólica	31
2.6. Modelo de viento	33
3. Propuesta de Control	37
3.1. Introducción a la propuesta de control	37
3.2. Análisis de las correlaciones de entrada y salida del sistema	38
3.2.1. Correlación entre en ángulo de las palas y la potencia generada . . .	38
3.2.2. Correlación entre la velocidad del viento y la potencia generada . . .	39
3.3. Requisitos de control	40
3.4. Control de paso de una turbina eólica	41
3.4.1. Discretización del espacio de estados y acciones	42

3.4.2.	Funciones de recompensa	48
3.4.3.	Configuración de los hiperparámetros del modelo	49
3.5.	Control multiagente mediante aprendizaje reforzado	52
3.5.1.	Arquitectura del sistema federado	52
3.5.2.	Función de agregación	53
3.5.3.	Sincronización y comunicación	54
4.	Resultados	57
4.1.	Descripción del entorno de simulación	57
4.2.	Métricas de evaluación	57
4.3.	Evaluación de las técnicas de discretización de estados	59
4.4.	Evaluación de los resultados de aprendizaje federado	64
4.4.1.	Evaluación del error de potencia	64
4.4.2.	Comparación de las respuestas temporales de los sistemas federados y no federados	66
4.4.3.	Evaluación de la señal de control	68
4.4.4.	Análisis del comportamiento de las turbinas entrenadas mediante apren- dizaje federado	69
4.4.5.	Análisis de la respuesta del sistema tras la fase de entrenamiento . . .	71
5.	Conclusiones y trabajos futuros	73
5.1.	Conclusiones y objetivos alcanzados	73
5.2.	Aplicaciones	74
5.3.	Trabajos futuros	74
	Bibliografía y referencias	76

Índice de figuras

2.1. Partes de una turbina eólica. Imagen obtenida del artículo [19].	7
2.2. Regiones de operación de un aerogenerador [28].	9
2.3. Coeficientes de potencia de una turbina eólica dependiendo del ángulo de ataque [12].	9
2.4. Control por pérdida aerodinámica de una turbina eólica [8].	11
2.5. Control Adaptativo de Referencia Directa (DMRAC) [9].	12
2.6. Estrategia de control de paso mediante IPC [68].	15
2.7. Arquitectura de un modelo de aprendizaje reforzado. Figura adaptada del artículo de Amiri et al. [7].	16
2.8. Diagrama básico de una red neuronal [22].	20
2.9. Modelo de interacción en un sistema de aprendizaje por refuerzo mediante Deep Policy Network. Imagen adaptada de [16].	21
2.10. Arquitectura de red actor-crítico. Figura adaptada del artículo de Quang Vinh Do et al. [25]	22
2.11. Arquitectura del método Deep Deterministic Policy Gradient. Figura adaptada del artículo de Guo, Y et al. [33]	23
2.12. Arquitectura de aprendizaje federado.	26
2.13. Modelo dinámico de una turbina eólica [15].	31
2.14. Distribución probabilística de Weibull con $\eta = 12$ y $k = 2$	34
2.15. Simulación del viento empleando la distribución probabilística de Weibull con $\eta = 12$ y $k = 2$. Empleando factores de ponderación de $\alpha = 0,9$ y $\beta = 0,1$	34
3.1. Función de correlación cruzada entre el ángulo de las palas (θ) y la potencia de salida (P_{out}).	39
3.2. Función de correlación cruzada entre la velocidad del viento V_{wind} y la potencia de salida P_{out}	40
3.3. Diagrama de las entradas y salidas del sistema.	41
3.4. Funciones de discretización de la variable de estado V_{wind}	43
3.5. Representación de la discretización de la variable P_{error} utilizando la función de discretización 3.4 con $\delta = 0,001$ y $\psi = 0,477$	44

3.6. Representación de la discretización de la variable P_{error} mediante la función de discretización 3.5 con $\delta = 1, 15$	45
3.7. Representación de la discretización de la variable P_{error} mediante la función de discretización 3.6, con $\mu = 4096$	46
3.8. Representación de la discretización de la variable P_{error} mediante la función de discretización 3.7, con $A = 0, 467$	47
3.9. Discretización de la variable de salida a mediante la función $Q(i) = 0,5 \cdot i$	48
3.10. Decaimiento del valor de la recompensa futura para distintos valores de γ	50
3.11. Arquitectura del sistema de control federado.	52
3.12. Diagrama secuencial de aprendizaje federado de las interacciones entre los aerogeneradores y el servidor federado.	55
4.1. Potencia media por episodio de cada método de discretización durante la primera ejecución.	59
4.2. Potencia media de salida por episodio de cada método de discretización durante la segunda ejecución.	60
4.3. Potencia media de salida por episodios de cada método de discretización durante la tercera ejecución.	60
4.4. Potencia media de salida por episodio de cada método de discretización durante la cuarta ejecución.	61
4.5. Potencia media por episodio durante 20 episodios de entrenamiento.	62
4.6. Respuesta temporal de los métodos de discretización tras 40 episodios de entrenamiento.	62
4.7. Respuesta temporal en estado estacionario del método de discretización tras 40 episodios de entrenamiento.	63
4.8. Comparación del error cuadrático medio entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.	64
4.9. Comparación del error absoluto integral entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.	65
4.10. Comparación del Error Absoluto Ponderado por el Tiempo (ITAE) entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.	66
4.11. Comparación de la potencia de salida media entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.	67
4.12. Comparación de la potencia de salida de las turbina individual con la potencia de salida media de las turbinas del sistema federado.	68
4.13. Comparación de la variación total de la señal de control entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.	69

4.14. Evolución del valor del Error Cuadrático Medio para cada turbina del sistema de Aprendizaje Federado durante el entrenamiento.	70
4.15. Potencia de las turbinas del sistema de aprendizaje federado en la episodio número 10 del entrenamiento.	70
4.16. Respuesta temporal de los sistemas de control entrenados mediante aprendizaje federado una vez finalizada la fase de entrenamiento.	71
4.17. Respuesta temporal obtenida tras la simulación una vez ha finalizado el entrenamiento.	72

Índice de tablas

2.1. Parámetros de la turbina eólica	33
--	----

Capítulo 1

Introducción

En los últimos años, la energía eólica se ha convertido en una de las principales fuentes de energía renovable debido a su capacidad para generar energía de manera eficiente y a los avances tecnológicos en este campo, contribuyendo así a la transición hacia energías sostenibles [34].

Sin embargo, las dinámicas no lineales, el acoplamiento entre variables y la incertidumbre en el modelado de los aerogeneradores suponen retos importantes a la hora de desarrollar estrategias de control eficaces y robustas, que son, por otro lado, esenciales para que estos sistemas sean eficientes y rentables [82].

A pesar de estas complicaciones, gran parte de las turbinas eólicas en la actualidad emplean estrategias de control clásicas basadas en controladores de tipo PI. Estos no son capaces de capturar las no linealidades de la dinámica o adaptarse a cambios significativos en la planta [13], dando como resultado una pérdida en la eficiencia de la turbina cuando ésta se ve sometida a perturbaciones externas.

Para mejorar los resultados obtenidos con las técnicas de control clásicas, en los últimos años se han propuesto enfoques basados en técnicas de inteligencia artificial (IA), los cuales permiten al controlador aprender leyes de control óptimas sin la necesidad de disponer de un modelo preciso de la planta [49]. Entre estas técnicas, el aprendizaje reforzado (del inglés, reinforcement learning, RL) ha mostrado buenas características de control con respecto a la estabilización de la potencia de salida y la reducción de cargas estructurales [76]. Estos resultados se deben principalmente a la capacidad de los controladores de aprendizaje por refuerzo para adaptarse a cambios en su entorno de operación a partir de su interacción con el mismo [26].

Entre los enfoques de aprendizaje reforzado, las técnicas discretas ofrecen ventajas sobre los

métodos continuos, como mejor interpretabilidad del modelo, mayor estabilidad y menores requisitos computacionales [24], siendo estas características deseadas para el control de un aerogenerador.

Sin embargo, el uso de modelos discretos también tiene desventajas significativas, como restricciones en el error mínimo de control debido a la granularidad de la discretización del espacio de estados, y el crecimiento exponencial en el número de estados discretos a medida que aumenta la dimensionalidad del problema [64], lo cual está directamente relacionado con la cantidad de datos necesarios para entrenar el controlador.

A pesar de los buenos resultados obtenidos al aplicar técnicas de aprendizaje por refuerzo al control de la potencia de salida en turbinas eólicas, su empleo en la industria se ha visto limitado debido a la gran cantidad de datos necesarios para que el controlador aprenda la estrategia de control óptima [57]. Aunque una posible solución a este problema sería combinar los datos de múltiples aerogeneradores para entrenar un modelo global, disminuyendo así los tiempos de entrenamiento, la sensibilidad de los datos operativos de los aerogeneradores y los riesgos de ciberseguridad derivados de transmitir los datos por la red hacen que esta solución no pueda ser implementada en la práctica.

Las técnicas de aprendizaje federado (del inglés, federated learning, FL) surgen como una posible solución a este problema. El aprendizaje federado es un tipo de aprendizaje automático descentralizado en el que varios agentes entrenan un modelo global de forma colaborativa, sin la necesidad de compartir datos sensibles en el proceso. En su lugar, cada agente entrena el modelo localmente utilizando sus propios datos de entrenamiento, y sólo las actualizaciones del modelo se envían a un servidor central para su agregación, preservando así la privacidad de los datos y reduciendo la cantidad de información que debe ser transmitida por la red [81]. Finalmente, el servidor central agrega los datos obtenidos de todas las turbinas y actualiza el modelo global.

Además, el empleo de técnicas de discretización del espacio de estados permite reducir el conjunto de estados que deben ser aprendidos por el controlador, reduciendo así la cantidad de datos necesarios para su entrenamiento y disminuyendo el error en el estado estacionario [61].

1.1. Propuesta y objetivos

Basándose en el trabajo presentado en [56], en este trabajo se propone la aplicación de técnicas de aprendizaje federado para entrenar un modelo de control global basado en aprendizaje reforzado, mediante el uso de datos provenientes de todos los aerogeneradores presentes en

una granja eólica. Este método permite preservar la privacidad y seguridad de los datos, al mismo tiempo que se aprovecha la información de múltiples aerogeneradores para mejorar el rendimiento del controlador y reducir los tiempos de entrenamiento.

Además, se proponen y evalúan varios métodos de discretización de la entrada con el objetivo de reducir la cantidad de datos de entrenamiento necesarios para obtener la política óptima y mejorar el error en estado estacionario de los controladores.

De esta forma, este trabajo pretende demostrar cómo el empleo de técnicas de aprendizaje federado y discretización de estados permite reducir los tiempos de aprendizaje de sistemas de control basados en aprendizaje reforzado y reducir su error en el estado estacionario, cuando estos son aplicados a la estabilización de potencia en turbinas eólicas.

Para ello, se desarrollará un sistema de control de paso colectivo (CPC) basado en la técnica de aprendizaje por refuerzo de Q-Learning, con el objetivo de estabilizar la potencia de una turbina eólica de 800KW de potencia. Este controlador será entrenado y optimizado para el control de una turbina eólica individual, analizando el rendimiento de los métodos de discretización propuestos. Posteriormente, se realizará el mismo entrenamiento empleando aprendizaje federado con un sistema compuesto por tres turbinas ¹.

Finalmente, se realizará una comparación tanto de los tiempos de aprendizaje como de los objetivos de control entre el método de control federado y el no federado.

Se definen los siguientes objetivos específicos del trabajo:

- Diseño e implementación de un controlador basado en Q-learning para la estabilización de potencia en una turbina eólica. Para ello:
 - Se estudiarán distintas funciones de recompensa para mejorar el rendimiento del sistema y reducir la fatiga en el rotor de la turbina.
 - Se propondrán métodos de discretización del espacio de estados para mejorar los tiempos de aprendizaje y reducir el error en el estado estacionario.
- Diseño e implementación de un algoritmo de aprendizaje federado para entrenar el controlador basado en Q-learning utilizando datos de múltiples turbinas eólicas.
- Evaluación de las técnicas propuestas. Para ello:
 - Se compararán las técnicas de discretización del espacio de estados propuestas, analizando los tiempos de aprendizaje del controlador y error en el estado estacionario.

¹Las técnicas de control se tratarán con mayor profundidad en el **Capítulo 2**.

- Se compararán los tiempos de aprendizaje y objetivos de control obtenidos entre una turbina individual y una granja de turbinas empleando aprendizaje federado.

Como resultado de este proyecto, se pretende demostrar cómo las técnicas de aprendizaje federado y discretización del espacio de estados pueden reducir los tiempos de aprendizaje de controladores basados en Q-Learning aplicados a la estabilización de potencias de turbinas eólicas, a la vez que se realiza un entrenamiento descentralizado en el que se preserva la privacidad de los datos operacionales de las turbinas, de forma que se demuestre la viabilidad de este tipo de controladores en sistemas comerciales.

1.2. Estructura del documento

Este documento comienza por el **Capítulo 1** con la contextualización del tema y la presentación de la propuesta del trabajo, junto con los objetivos que se desean obtener en el mismo.

En el **Capítulo 2** se realiza un análisis de las técnicas de control actuales aplicadas al control de turbinas eólicas, así como tendencias en la investigación. En este capítulo también se realiza una revisión de las técnicas de control basadas en aprendizaje por refuerzo y aprendizaje federado. Finalmente, se presenta el modelo de turbina empleado en la simulación.

En el **Capítulo 3** se detalla el sistema de control desarrollado en este trabajo. Este capítulo comienza con un análisis de la dinámica de la turbina. Posteriormente, se presenta el algoritmo de control Q-learning empleado y se define la arquitectura del sistema de aprendizaje federado.

En el **Capítulo 4** se muestran y analizan los resultados obtenidos para distintos métodos de discretización de las variables de entrada. Además, se realiza una comparación entre los resultados obtenidos con el modelo federado y el modelo no federado.

En el **Capítulo 5** se enuncian las conclusiones obtenidas tras el análisis de los resultados y se proponen posibles mejoras y trabajos futuros.

Este documento finaliza con la bibliografía y referencias empleadas para su realización.

1.3. Publicaciones derivadas

Este trabajo de final de máster (TFM) ha dado lugar a las siguientes publicaciones científicas:

- Alberto Gil, M. Santos, J. Sierra-García. First Approximation of Application of Federated Learning to Wind Turbines. 23rd International Conference on Renewable Energies and Power Quality (ICREPQ'25), University of La Laguna, Tenerife, Spain, June 25-27, 2025 (aceptado, pendiente de publicación)
- Alberto Gil, M. Santos, J. Sierra-García. Exponential and quadratic state discretization for Wind Turbine Pitch Control based on Reinforcement learning. 1st IFAC Joint Conference on Computers, Cognition, and Communication, (J3C 2025 (EAAS2025, ICONS2025, TA2025)), International Federation of Automatic Control, Padova, ITALY, September 15, 2025 (enviado, pendiente de revisión)

Partes de este TFM se basan en las publicaciones derivadas, profundizando en ciertos aspectos de los mismos y ampliando los resultados previos con nuevos análisis.

Capítulo 2

Fundamentos Teóricos y Estado del Arte

En este capítulo se presentan los conceptos fundamentales relacionados con las condiciones de operación de una turbina eólica y se analizan las principales técnicas empleadas para su control. Finalmente, se hace una revisión del estado actual de las técnicas de aprendizaje por refuerzo aplicadas al control y su entrenamiento mediante aprendizaje federado.

2.1. Funcionamiento básico de una turbina eólica

Una turbina eólica o aerogenerador es un dispositivo que convierte la energía del viento en electricidad a partir del movimiento de sus palas [42].

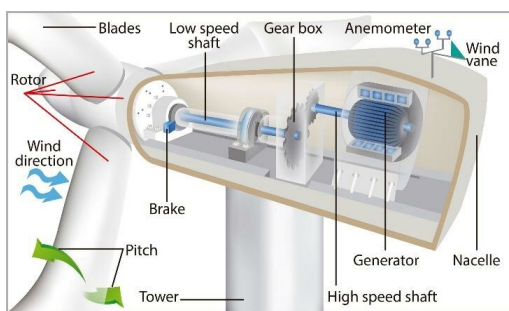


Figura 2.1: Partes de una turbina eólica. Imagen obtenida del artículo [19].

La figura 2.1 muestra los componentes principales de una turbina eólica.

El viento hace girar las palas de la turbina, las cuales se encuentran conectadas al eje de baja velocidad a través del buje. Con tal de generar electricidad a la frecuencia deseada, se

emplea una caja de cambios que incrementa la velocidad angular en el eje de alta velocidad a costa de reducir el par transmitido.

Finalmente, el eje de rotación de alta velocidad está conectado a un generador, el cual transforma la energía mecánica del eje de rotación en energía eléctrica.

Por otro lado, la turbina contiene un sistema de frenado, el cual permite detener la rotación del rotor cuando las condiciones del viento hacen peligrar la integridad del sistema o se requiere mantenimiento [2].

2.2. Regiones de operación de una turbina eólica

Las regiones de operación de una turbina eólica corresponden a los rangos de velocidad del viento en los que esta puede operar, los cuales se muestran en la figura 2.2. De esta forma, es posible distinguir cuatro regiones de operación [8]:

- Región 1: Velocidades de viento comprendidas entre $0m/s$ y la velocidad de corte v_{corte1} . En esta región, la velocidad del viento no es suficiente para que la turbina pueda generar energía útil, de modo que esta permanece inactiva y no se produce energía.
- Región 2: Velocidades de viento comprendidas entre v_{corte1} y v_{nominal} . En esta región la turbina comienza a generar energía, pero la velocidad del viento no es la suficiente como para alcanzar la potencia nominal, de modo que el objetivo de control es maximizar la energía producida.
- Región 3: Velocidades de viento comprendidas entre v_{nominal} y v_{corte2} . En esta región la turbina opera a su capacidad nominal, siendo el objetivo de control mantener la potencia generada lo más cercana posible a la potencia nominal, evitando así daños estructurales.
- Región 4: Velocidades de viento superiores v_{corte2} . En esta región la turbina no puede operar de forma segura, por lo que se detiene para evitar daños estructurales.

2.2.1. Curvas de potencia óptimas

La curva de potencia óptima de una turbina eólica describe cuál es la potencia máxima que esta puede generar en función de la velocidad del viento. La curva de potencia óptima está directamente relacionada con las regiones de operación de la turbina, correspondiéndose cada región de operación a una sección de la curva.

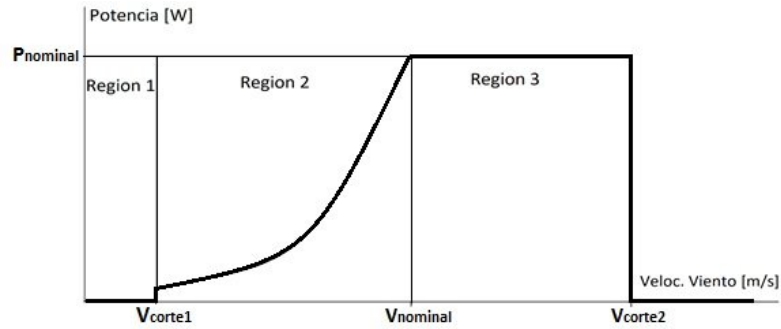


Figura 2.2: Regiones de operación de un aerogenerador [28].

Para una velocidad de viento v_{viento} (en m/s), la energía del viento viene definida por la siguiente expresión:

$$P_{viento} = \frac{1}{2} \sigma A v_{viento}^3 \quad (2.1)$$

Donde σ es la densidad del aire (en kg/m^3) y A es el área barrida por las aspas de la turbina (en m^2).

Sin embargo, la potencia máxima que la turbina puede extraer del viento queda limitada por su coeficiente de potencia C_p , el cual indica la fracción de potencia que puede ser capturada por la turbina con respecto a la energía total del viento [36]. Este valor queda determinado por la velocidad del viento v_{viento} y el ángulo de ataque de las palas de la turbina θ (en $^\circ$).

La figura 2.3 muestra la variación del coeficiente de potencia de una turbina eólica para distintos ángulos de las palas.

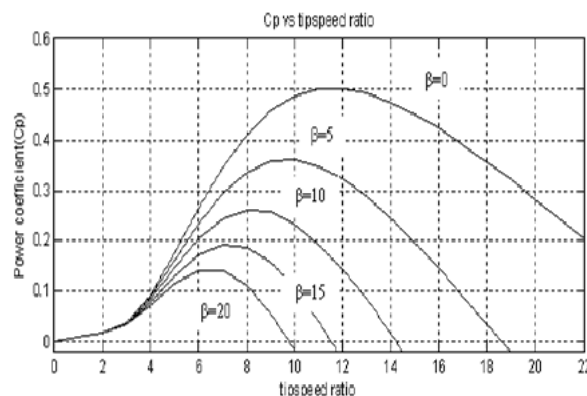


Figura 2.3: Coeficientes de potencia de una turbina eólica dependiendo del ángulo de ataque [12].

De esta forma, la curva de potencia óptima de una turbina eólica vendrá dada por la siguiente

expresión:

$$P_{maxima} = \frac{1}{2} \sigma A v_{viento}^3 C_p(\lambda, \theta) \quad (2.2)$$

Donde λ es la relación de velocidad de punta (del inglés, Tip Speed Ratio, TSR o λ) de la turbina (en m/s).

Por último, cabe destacar que el valor máximo teórico de C_p queda definido por el límite de Betz [23], el cual establece que $C_{p_{max}} = 0,59$.

2.3. Estrategias de control de una turbina eólica

El objetivo de control en una turbina eólica es maximizar la producción de energía a la vez que se reducen las cargas estructurales, prolongando así la vida útil del sistema [48]. Para ello, existen múltiples tipos de control, entre los que destacan el control de paso y el control por pérdida aerodinámica.

2.3.1. Control por pérdida aerodinámica

Los sistemas de control por pérdida aerodinámica limitan la potencia de salida de la turbina a partir de la modificación de la eficiencia aerodinámica de sus palas, lo cual puede realizarse mediante control pasivo o activo [8].

En los sistemas de control pasivos, el perfil aerodinámico de las palas se diseña de forma que entren en pérdida aerodinámica ante condiciones de viento adversas. Esto permite ajustar la potencia de salida de la turbina sin la necesidad de partes móviles.

Por otro lado, los sistemas de control activos basados en pérdida aerodinámica controlan la potencia generada mediante el ajuste del ángulo de ataque de las palas, provocando una pérdida aerodinámica controlada cuando la velocidad del viento supera la velocidad de corte, lo cual permite controlar la potencia generada de una forma mucho más precisa que en el control pasivo.

En la figura 2.4 se compara la respuesta temporal obtenida por un controlador de pérdida aerodinámica activo y otro pasivo. Como se puede observar, el método de control pasivo provoca que las palas de la turbina entren en pérdida aerodinámica cuando la potencia producida excede los límites certificados para la turbina, provocando así una caída en la

potencia generada. Por otro lado, el control activo permite controlar la fuerza de sustentación de las palas de forma mucho más precisa, provocando una caída menor en la potencia de salida.

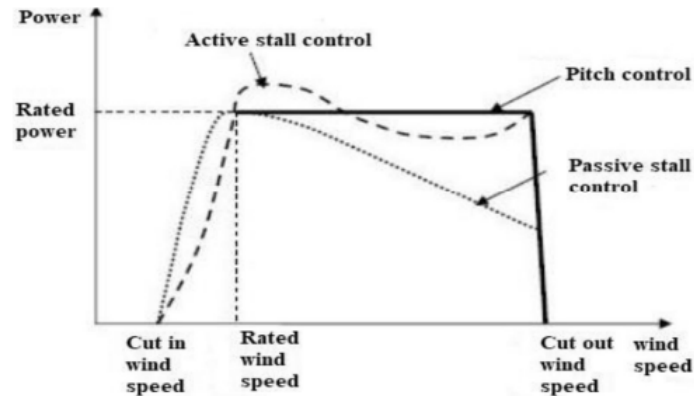


Figura 2.4: Control por pérdida aerodinámica de una turbina eólica [8].

La principal ventaja del control por pérdida aerodinámica es su simplicidad y robustez. Sin embargo, la baja eficiencia de este tipo de control, junto con la complejidad de diseño de las palas, provoca que su aplicación esté limitada a turbinas eólicas antiguas o de velocidad fija [8].

2.3.2. Control de paso

Los métodos de control de paso ajustan el ángulo de las palas a partir de la información sobre la potencia generada por la turbina, con el objetivo de reducir el error entre la potencia de salida y una potencia de referencia, a la vez que se minimizan las cargas estructurales sobre el sistema [8].

Es posible dividir el control de paso en control de paso individual o paso colectivo.

Control de paso colectivo

El control de paso colectivo (Collective Pitch Control, CPC) realiza el control de la turbina mediante el ajuste simultáneo del ángulo de todas sus palas [48].

La implementación más ampliamente utilizada de este tipo de control es mediante control proporcional-integral (PI) con ganancia programada, el cual es empleado para seguir una velocidad angular de referencia w_{ref} (en rad/s) [35], limitando así la potencia generada.

$$\theta = K_p \left(1 + \frac{K_i}{s}\right) (w_{ref} - w) \quad (2.3)$$

Donde θ es el ángulo de paso colectivo, w es la velocidad angular del rotor (en rad/s) y K_p y K_i son las constantes proporcional e integral del controlador PI.

El principal problema de los controladores de tipo PI es su dificultad para manejar las no linealidades en la dinámica del sistema, lo cual provoca una pérdida de eficiencia en la turbina. Por otro lado, estos métodos de control suponen que todas las palas de la turbina se ven sometidas a las mismas cargas aerodinámicas, causando un estrés en los componentes de la turbina que puede reducir su vida útil [47]. Para resolver estos problemas, se han propuesto nuevas técnicas basadas en control adaptativo y control robusto.

El control adaptativo es una estrategia de control que ajusta los parámetros del modelo en tiempo real a partir de variaciones en el modelo de la planta o perturbaciones externas [20]. Debido a la no linealidad en la dinámica de las turbinas eólicas, estos controladores permiten adaptarse a cambios en la planta que no hayan sido capturados durante la linealización, mejorando así el rendimiento y la robustez del sistema de control.

Un ejemplo de control adaptativo aplicado al control de paso colectivo puede encontrarse en [29], donde se utiliza un Control Adaptativo de Referencia Directa (DMRAC) para el control de la potencia y minimización de perturbaciones en una turbina eólica. Este tipo de control emplea un modelo de referencia para ajustar los parámetros del controlador, con el objetivo de minimizar el error entre el modelo real y la salida del modelo de referencia. En la figura 2.5 se muestra el diagrama de bloques empleado por DMRAC. En él se puede observar cómo el mecanismo de ajuste adaptará los parámetros del controlador a partir de los datos de la acción de control U , la salida del modelo de referencia Y_m y la salida real de la planta Y_p .

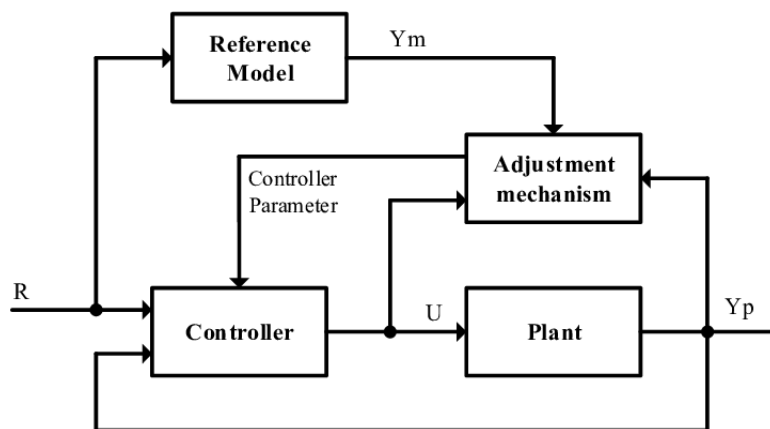


Figura 2.5: Control Adaptativo de Referencia Directa (DMRAC) [9].

Otra forma de manejar la incertidumbre en el modelo es mediante control robusto [27]. Este consiste en una estrategia de control diseñada para garantizar el rendimiento y la estabilidad del sistema. De este modo, en [45] se propone un control basado en control H_∞ para reducir las cargas en la estructura de una turbina eólica.

Por otro lado, existe abundante literatura en la que se estudia el empleo de técnicas de control no lineal para el control de turbinas eólicas [65]. Entre estos métodos destacan los controladores basados en lógica difusa, redes neuronales artificiales y aprendizaje por refuerzo.

En los controladores basados en lógica difusa, las acciones de control se obtienen a partir de un conjunto de reglas lingüísticas que describen cuál debe ser el comportamiento del sistema para un conjunto de entradas determinadas. Estos controladores emplean un conjunto de funciones de pertenencia para asignar los valores de las variables de entrada a una categoría difusa. A continuación, se emplea un proceso de inferencia difusa para combinar las distintas reglas de control y determinar la acción de control difusa más adecuada. Finalmente, se traduce el valor de la salida difusa a una acción de control concreta, la cual corresponderá a la entrada de la planta. Los sistemas de control difuso permiten controlar las dinámicas no lineales de una turbina eólica sin la necesidad de disponer de un modelo matemático detallado del sistema [5].

En cuanto a las técnicas de control basadas en redes neuronales, destacan los resultados obtenidos mediante redes neuronales RBF (Radial Basis Function Networks) [46] y redes neuronales recurrentes ¹.

Las redes neuronales RBF emplean funciones de base radial como funciones de activación, lo que les permite aproximar cualquier función no lineal con una precisión arbitraria, reduciendo así el tiempo de entrenamiento en redes neuronales profundas [1]. Estas funciones de activación dependen de la distancia entre el vector de entrada y el centro asociado a cada neurona, siendo una de las funciones más comunes la función gaussiana [50] mostrada en la ecuación 2.4.

$$h_j(t) = \exp\left(-\frac{\|X - C_j\|^2}{2b_j^2}\right), j = 1, 2, \dots, m \quad (2.4)$$

Donde X es la entrada a la neurona, C es el centro de la base radial alrededor del cual se evalúa la función y $B = [b_1, b_2, \dots, b_m]^T$ es el ancho del vector de bases.

De esta forma, es posible emplear una red neuronal RBF junto con alguna técnica de control

¹En la sección 2.3.3 se aporta una explicación más detallada del funcionamiento de una red neuronal.

lineal, de forma que la red RBF tenga la función de aprender la parte de la dinámica no lineal de la turbina eólica [46][58]. Por otro lado, en [71] se presenta un método de control adaptativo basado en RBF-PID, donde una red neuronal RBF es empleada para actualizar los parámetros del controlador PID.

Otro tipo de redes neuronales con las que se han obtenido resultados relevantes en el control de turbinas eólicas son las redes neuronales recurrentes (RNN). En [11] se muestra cómo el empleo de RNN para el control de paso mejora los resultados obtenidos mediante técnicas de control tradicionales. Esto se debe a la capacidad de las RNN para manejar datos temporales y patrones secuenciales, los cuales son de gran relevancia en el control de una turbina eólica.

Por otro lado, también es posible emplear sistemas de control híbridos, los cuales combinan métodos de redes neuronales y aprendizaje difuso. Esto se demuestra en [55], donde se emplea una red neuronal para la estimación de las condiciones de viento futuras. Mejoran así las capacidades de un controlador basado en lógica difusa.

Por último, en [63] se muestra cómo el empleo de controladores basados en aprendizaje por refuerzo consigue mejorar significativamente los resultados obtenidos por un controlador PID. Por otro lado, en [59] se analizan distintas técnicas de aprendizaje por refuerzo para el control de turbinas eólicas. Los controladores basados en aprendizaje por refuerzo han demostrado resultados significativos en el control de turbinas eólicas y serán analizados en mayor profundidad en la sección 2.3.3.

Control de paso individual

El control de paso individual (Individual Pitch Control, IPC) permite ajustar el ángulo de cada una de las palas de la turbina de forma independiente, lo cual le permite reducir las cargas asimétricas en la turbina, mejorando así su vida útil [68]. IPC emplea sensores y actuadores individuales para cada una de las palas, tratándose de un problema de control de entrada y salida múltiple (MIMO).

Gran parte de la investigación en este tipo de control se basa en la aplicación de la transformada de Coleman, la cual se emplea para expresar las dinámicas del sistema con respecto a la rotación a un sistema de referencia ortogonal fijo compuesto por ejes directos (d) y cuadráticos (q) (los cuales representan los momentos de flexión de la guiñada e inclinación) [41].

$$\begin{pmatrix} M_d \\ M_q \end{pmatrix} = \frac{2}{3} \begin{pmatrix} \cos(\psi) & \cos(\psi + \frac{2\pi}{3}) & \cos(\psi + \frac{4\pi}{3}) \\ \sin(\psi) & \sin(\psi + \frac{2\pi}{3}) & \sin(\psi + \frac{4\pi}{3}) \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} \quad (2.5)$$

Donde M_1 , M_2 y M_3 son las cargas en la base de las palas, las cuales son transformadas en las señales ortogonales de los ejes directos y cuadráticos M_d y M_q .

Una vez aplicada la transformada de Coleman, es posible realizar el control en este nuevo espacio de coordenadas. Este control suele realizarse empleando dos bucles de control PI independientes, los cuales cancelan las cargas en los ejes q y d respectivamente. Sin embargo, también se han estudiado el empleo de técnicas de control óptimas como Linear Quadratic Regulator [32] o H_∞ [30].

Finalmente, se emplea la transformada inversa de Coleman para obtener el ángulo de paso requerido por cada pala.

$$\begin{pmatrix} \Delta\beta_1 \\ \Delta\beta_2 \\ \Delta\beta_3 \end{pmatrix} = \frac{2}{3} \begin{pmatrix} \cos(\psi) & \sin(\psi) \\ \cos(\psi + \frac{2\pi}{3}) & \sin(\psi + \frac{2\pi}{3}) \\ \cos(\psi + \frac{4\pi}{3}) & \sin(\psi + \frac{4\pi}{3}) \end{pmatrix} \begin{pmatrix} \beta_d \\ \beta_q \end{pmatrix} \quad (2.6)$$

En la figura 2.6 se puede observar el diagrama de control de un sistema IPC. En él se observa cómo el sistema de control CPC es empleado para el control del ángulo de ataque de la turbina con el fin de maximizar la potencia. Por otro lado, el sistema de control IPC se emplea para reducir las cargas a las que se ven sometidas las palas.

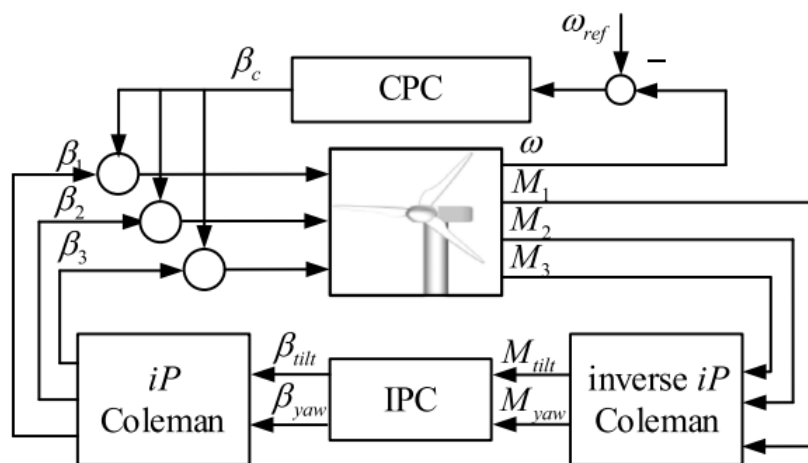


Figura 2.6: Estrategia de control de paso mediante IPC [68].

Si bien se han realizado grandes avances en el control de paso individual en las últimas

décadas, su aplicación en la actualidad está limitada al ámbito académico. Es muy probable su implementación en las próximas generaciones de aerogeneradores, donde las palas más grandes y flexibles requieran técnicas de control más complejas.

2.3.3. Estrategias de control basadas en aprendizaje reforzado

Como ya se mencionó en la sección 2.3.2, las dinámicas no lineales de una turbina eólica hacen que sea de gran conveniencia el empleo de sistemas de control basados en técnicas de inteligencia artificial. En particular, las técnicas de aprendizaje reforzado han demostrado una gran eficacia para el control de turbinas eólicas.

La figura 2.7 muestra la arquitectura general de un controlador basado en aprendizaje por refuerzo. En este enfoque, la estrategia de control óptima se aprende a partir de la interacción entre el controlador y su entorno. El controlador, representado por el agente, modifica el entorno realizando una acción a_t . La ejecución de esta acción dará como resultado un cambio en el estado del entorno s_{t+1} y un valor de recompensa r_t , que representa lo beneficiosa que ha sido la acción con respecto a los objetivos de control. Por último, el algoritmo de aprendizaje por refuerzo utiliza esta información para adaptar su política π con el fin de promover acciones futuras que maximicen la recompensa a lo largo del tiempo [59].

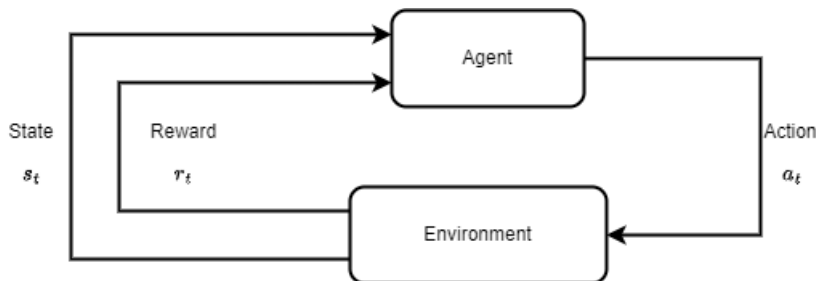


Figura 2.7: Arquitectura de un modelo de aprendizaje reforzado. Figura adaptada del artículo de Amiri et al. [7].

Los algoritmos de aprendizaje por refuerzo están estrechamente relacionados con los procesos de decisión de Markov (MDP), donde el estado futuro del sistema depende únicamente de su estado actual. En el caso del aprendizaje por refuerzo, esta definición se extiende para incluir también acciones y recompensas.

En este contexto, la política π se encarga del mapeado de cada estado $s \in S$ del entorno a una acción $a \in A(s)$ a partir de la función $\pi(s, a)$, siendo el objetivo del entrenamiento el de determinar la política $\pi(s, a)$ que maximice la recompensa. Para ello se define la función de valor $V_\pi(s)$, la cual permite conocer el valor de recompensa futura esperada de encontrarse

en un estado s con la política π [67]. Cabe destacar que $V_\pi(s)$ es una función probabilística, ya que el estado del entorno queda descrito por un proceso de decisiones de Markov [16].

$$V^\pi(s) = E_\pi \left(\sum_k \gamma^k r_k | s_0 = s \right) \quad (2.7)$$

En la ecuación 2.7 el símbolo γ es un factor de descuento que se emplea para priorizar la obtención de recompensas cercanas en el tiempo.

Ecuación de Bellman

En el sistema de aprendizaje por refuerzo descrito anteriormente, tanto la política π como la función valor $V_\pi(s)$ son desconocidas. El objetivo del algoritmo de entrenamiento es encontrar la política y la función de valor que maximicen la recompensa.

Si se considera que se está empleando una política π óptima en todo momento, es posible expresar la ecuación 2.7 como:

$$V(s) = \max_\pi E \left(\sum_k \gamma^k r_k | s_0 = 0 \right) \quad (2.8)$$

Esta función se puede descomponer de modo que quede expresada como la suma de la recompensa actual y la recompensa futura, obteniendo así la ecuación de Bellman [16].

$$V(s) = \max_\pi E (r_0 + \gamma V(s')) \quad (2.9)$$

Donde s' representa el siguiente estado del sistema.

De esta forma, es posible encontrar la función $V(s)$ de forma recursiva a partir de la recompensa actual r_t y la recompensa obtenida en el futuro r_{t+1} .

Por último, una vez obtenida la función $V(s)$, la política óptima será aquella que maximice la recompensa esperada.

$$\pi = \operatorname{argmax}_\pi E (r_0 + \gamma V(s')) \quad (2.10)$$

De este modo, el objetivo del algoritmo de entrenamiento será el de hallar el valor de la

función de valor, ya que a partir de esta se puede obtener la política óptima. El método empleado para hallar la función de valor da lugar a las distintas técnicas de aprendizaje por refuerzo.

Método de iteración de valores

El Método de iteración de valores permite obtener la función de valor a partir del modelo del sistema [79].

Existen ciertos sistemas en los que es posible describir el entorno mediante un proceso de decisión de Markov de la forma [16]:

$$P(s', s, a) = Pr(s_{k+1} = s' | s_k = s, a_k = a) \quad (2.11)$$

Del mismo modo, existirá también un modelo para la recompensa expresado como:

$$R(s', s, a) = Pr(r_{k+1} | s_{k+1} = s', s_k = s, a_k = a) \quad (2.12)$$

En este tipo de sistemas, es posible aprovechar el conocimiento de los modelos para calcular el valor de la función $V(s)$.

Una aproximación ampliamente utilizada para la obtención de la función $V(s)$ es el método de iteración de valores. Este emplea el conocimiento sobre el modelo para, aplicando la función 2.9, hallar la función de valor para la política óptima [67].

$$V(s) = \max_a \sum_{s'} P(s'|s, a)(R(s', s, a) + \gamma V(s')) \quad (2.13)$$

Como se puede observar, la expresión anterior obtiene el valor de valor $V(s)$ a partir de los modelos del entorno $P(s'|s, a)$ y de la recompensa $R(s', s, a)$. Desafortunadamente, las dinámicas complejas de las turbinas eólicas hacen imposible que estas funciones puedan ser obtenidas, de modo que es necesario emplear otros métodos de aprendizaje reforzado que permitan encontrar la función de valor $V(s)$ y la política π sin la necesidad de disponer de un modelo preciso del entorno o función de recompensa.

Q-learning

Otra forma comúnmente empleada para la obtención de la política óptima es mediante la función de calidad $Q(s, a)$. Esta expresa el valor de recompensa esperable por una acción, estando en un determinado estado, si se sigue la política óptima a partir de ese punto [17]. La principal ventaja de esta función es que contiene información de la calidad de un estado dependiendo de cada una de las posibles acciones futuras, lo cual facilita la selección de la acción óptima [16].

$$Q(s, a) = E(R(s', s, a) + \gamma V(s')) = \sum_{s'} P(s'|s, a)(R(s', s, a) + \gamma V(s')) \quad (2.14)$$

Cabe destacar que la función de valor $V(s)$ se relaciona con la función de calidad $Q(s, a)$ mediante la siguiente expresión:

$$V(s) = \max_a Q(s, a) \quad (2.15)$$

Por otro lado, la política óptima obtenida a partir de $Q(s, a)$ queda definida como:

$$\pi(s, a) = \operatorname{argmax}_a Q(s, a) \quad (2.16)$$

Al igual que sucedía en el método de iteración de valores, la función 2.14 únicamente puede ser obtenida si se dispone del modelo del entorno $P(s'|s, a)$ y de la recompensa $R(s', s, a)$. Cuando no se dispone de un modelo específico del entorno, es necesario aproximar el valor de la función de calidad de forma iterativa, mediante las interacciones del agente con el entorno. Así, la función $Q(s, a)$ es actualizada dependiendo de la recompensa r , obtenida al realizar una determinada acción a , mediante la siguiente ecuación [72]:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right) \quad (2.17)$$

Donde $r + \gamma \max_{a'} Q(s', a')$ representa la recompensa esperable por estar en el estado s . Si existe una discrepancia entre el valor de $Q(s, a)$ previamente estimado y el nuevo valor obtenido, la función $Q(s, a)$ será actualizada.

Cabe destacar que Q-learning es un método de aprendizaje fuera de política, lo cual implica que se puede realizar el aprendizaje incluso cuando las acciones tomadas por el agente son

subóptimas. Esto permite que se puedan seleccionar acciones aleatorias con tal de explorar nuevos estados en el espacio de búsqueda, reduciendo así la posibilidad de que el algoritmo converja a un mínimo local [17].

Deep Q-Learning

Uno de los principales problemas de Q-Learning es la necesidad de disponer de un espacio de estados y acciones finitos para que el algoritmo converja a la política óptima. Este problema es resuelto por Deep Q-Learning mediante el empleo de redes neuronales para la estimación de la función $Q(s, a)$, generalizando así el concepto de Q-Learning para sistemas con espacios de estado o acciones continuos.

Una red neuronal puede ser descrita como un conjunto de capas formadas por neuronas conectadas entre sí, donde cada neurona procesa la información como la suma ponderada de las salidas de las neuronas de la capa anterior por los pesos de los enlaces, y a cuyo resultado se añade el valor del sesgo de la neurona. El resultado de esta operación sirve como entrada para una función de activación no lineal, la cual da como resultado la salida de la neurona [53]. En la figura 2.8 se muestra el diagrama básico de una red neuronal.

El algoritmo de backpropagation permite a las redes neuronales aprender a partir de un conjunto de datos mediante el ajuste iterativo de sus pesos y sesgos [51]. Esto se hace con el objetivo de minimizar el error entre la salida predicha y la esperada.

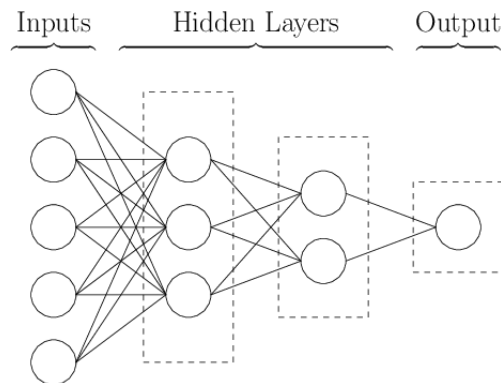


Figura 2.8: Diagrama básico de una red neuronal [22].

En el ámbito del control mediante aprendizaje reforzado, es posible considerar una red neuronal como un aproximador universal para cualquier función.

Una de las primeras aproximaciones en el uso de redes neuronales en el ámbito de aprendizaje reforzado es mediante el método de Deep Policy Network, cuya arquitectura se muestra en la figura 2.9. Este emplea una red neuronal que toma el estado del sistema como entrada

y produce la acción a realizar como salida [16]. La red neuronal parametriza entonces la función de la política como $\pi(s, a) \approx \pi(s, a, \theta)$.

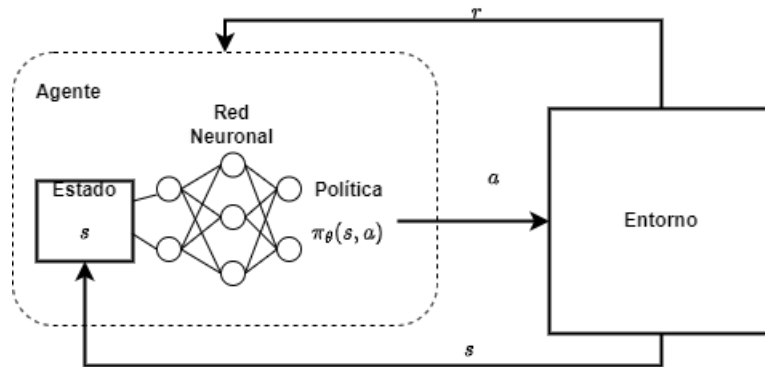


Figura 2.9: Modelo de interacción en un sistema de aprendizaje por refuerzo mediante Deep Policy Network. Imagen adaptada de [16]

Los pesos de la red en esta técnica son actualizados a partir de la siguiente función:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} r_{\theta} \quad (2.18)$$

Donde r_{θ} representa la recompensa obtenida para una configuración θ de los pesos de la neurona, de forma que la red neuronal aprenderá a maximizar la recompensa.

Otra aproximación al uso de redes neuronales es Deep Q-Learning, el cual trata de aproximar la función de calidad $Q(s, a)$ mediante una red neuronal $Q(s, a, \theta)$. Teniendo en cuenta la definición de la función de calidad mostrada en 2.17, es posible derivar la función de pérdida de la red como el error cuadrático del valor $Q(s, a)$ predicho y el valor obtenido [17]:

$$L = E \left[\left(r_k + \gamma \max_{a'} Q(s', a', \theta) - Q(s, a, \theta) \right)^2 \right] \quad (2.19)$$

La red neuronal aprenderá a aproximar la función $Q(s, a)$ de forma que se minimice el error entre el valor predicho y el obtenido. Una vez obtenida la función $Q(s, a)$, es posible obtener el valor de la política óptima $\pi(s, a)$ empleando la expresión 2.16. Cabe destacar que, al igual que sucedía con Q-Learning, Deep Q-Learning requiere un espacio de acciones discreto para obtener la política óptima.

Deep Deterministic Policy Gradient

Deep Q-Learning (DQN) resuelve el problema de tener que emplear un conjunto de estados finitos para la representación del entorno. Sin embargo, este método de aprendizaje sigue requiriendo que el sistema tenga un espacio de acciones finito.

Dado que la variable de actuación de una turbina eólica corresponde al ángulo de las palas θ , que se trata de una acción continua, conviene emplear métodos de aprendizaje que permitan representar tanto estados como acciones continuas. En este sentido, Deep Deterministic Policy Gradient (DDPG) es un método actor-crítico basado en DQN, el cual permite su empleo para sistemas con un espacio de acciones continuo [64].

Antes de describir el funcionamiento del método, es importante mostrar cómo funcionan las arquitecturas de red actor-crítico. En este tipo de sistemas existe una red neuronal denominada actor que intenta aproximar la política óptima $\pi(s, a) \approx \pi(s, a, \theta)$, y otra red neuronal denominada crítico que intenta aproximar el valor de la función de valor $V(s, \theta)$ o la función de valor acción-estado $Q(s, a, \theta)$ [16].

De este modo, el actor predice la acción a tomar a partir del estado actual del entorno. Por otro lado, el crítico proporciona retroalimentación al actor sobre las últimas acciones tomadas, como se puede observar en la figura 2.10.

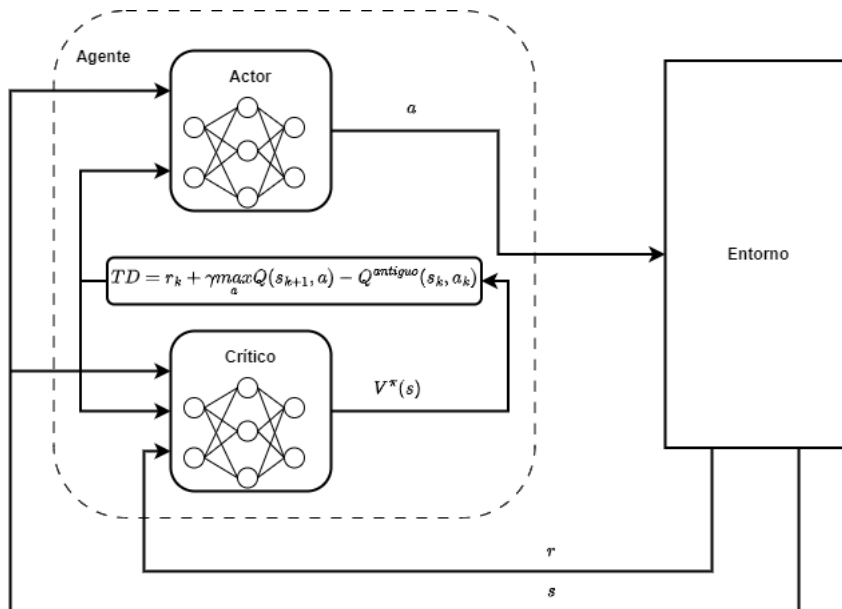


Figura 2.10: Arquitectura de red actor-crítico. Figura adaptada del artículo de Quang Vinh Do et al. [25]

Si bien el método de Q-Learning permite obtener la acción óptima a partir de la función

$Q(s, a)$, como aquella que maximiza la recompensa esperada según la expresión 2.16, esto no es generalizable para acciones en el espacio continuo. Empleando una arquitectura actor-crítico, es posible utilizar la red neuronal del actor para aproximar esta función $\pi(s, \theta) \approx \operatorname{argmax}_a Q(s, a)$, dando como resultado un método generalizable a conjuntos de estados y acciones continuos.

Deep Deterministic Policy Gradient(DDPG), cuya arquitectura general se muestra en la figura 2.11, es un método de aprendizaje basado en la técnica de actor-crítico, donde el crítico (Q) se encarga de aprender la función de calidad $Q(a, s, \theta)$ y el actor (π) emplea esta función para aprender la política óptima $\pi(s, \theta)$ que maximice la recompensa. Además, el método emplea una red neuronal adicional para el crítico Q' y otra para el actor (π'), denominadas redes objetivo, las cuales son empleadas para estabilizar el proceso de aprendizaje [18].

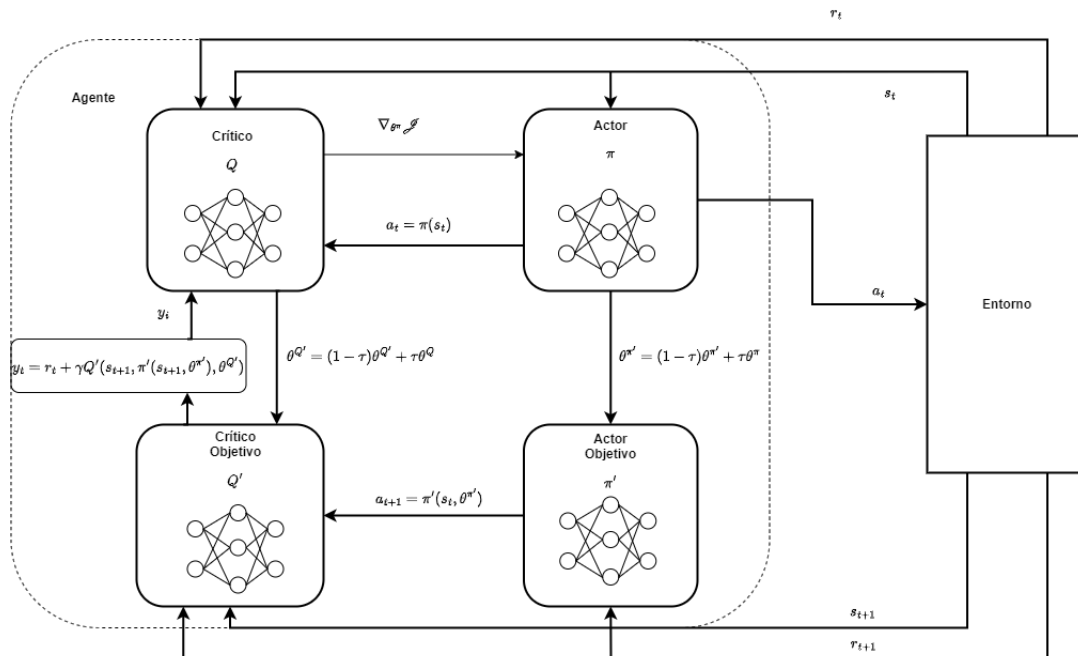


Figura 2.11: Arquitectura del método Deep Deterministic Policy Gradient. Figura adaptada del artículo de Guo, Y et al. [33]

Dado un determinado estado s_t , la red neuronal del actor $\pi(s_t, \theta)$ determinará la siguiente acción del agente a_t . Como resultado de esta acción, el entorno devolverá el siguiente estado s_{t+1} y la recompensa r_t ².

Basándose en la recompensa obtenida por el entorno r_t y el siguiente estado s_{t+1} , se obtiene el valor de y_t , el cual representa la suma de la recompensa actual más la esperada por encontrarse en el estado s_{t+1} [43].

²A diferencia de los apartados anteriores, esta sección hace uso del subíndice temporal en las variables para facilitar la comprensión del algoritmo.

$$y_t = r_t + \gamma Q'(s_{t+1}, \pi'(s_{t+1}, \theta^{\pi'}), \theta^{Q'}) \quad (2.20)$$

Donde $Q'(s_{t+1}, \pi'(s_{t+1}, \theta^{\pi'}), \theta^{Q'})$ es la estimación del crítico objetivo para el par estado-acción (s_{t+1}, a_{t+1}) , con $a_{t+1} = \pi'(s_{t+1}, \theta^{\pi'})$.

Una vez obtenido el valor de y_i , el crítico actualiza sus parámetros para minimizar el error entre el valor $Q(s_i, a_i, \theta^Q)$ predicho y el valor objetivo y_i . Para ello se emplea la siguiente función de pérdida [43]:

$$L = \frac{1}{N} \sum_i (Q(s_i, a_i, \theta^Q) - y_i)^2 \quad (2.21)$$

Una vez entrenado el crítico, se emplea su estimación para actualizar los pesos del actor. Para ello, el actor debe maximizar la función de calidad $Q(s, a)$ predicha por el crítico π durante un tiempo i .

$$\mathcal{J}(\theta^\pi) = \frac{1}{N} \sum_i Q(s_i, \pi(s_i, \theta^\pi), \theta^Q) \quad (2.22)$$

De esta forma, se puede demostrar que los parámetros del actor θ^π deben ser entrenados empleando el gradiente de la salida del crítico con respecto a la acción, como se muestra en el artículo [40].

$$\nabla_{\theta^\pi} \mathcal{J} = \frac{1}{N} \sum_i \nabla_a Q(s_i, a, \theta^Q)|_{a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s_i, \theta^\pi) \quad (2.23)$$

Por último, los parámetros del crítico objetivo y del actor objetivo son actualizados de modo que converjan a los valores de Q y π mediante la siguiente expresión [4]:

$$\theta^{\pi'} \leftarrow \tau \theta^\pi + (1 - \tau) \theta^{\pi'} \quad (2.24)$$

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (2.25)$$

Donde τ es una ganancia pequeña.

Dado que en el control de una turbina eólica tanto la entrada como la salida son continuas,

esta arquitectura de red es de especial relevancia para realizar el control sin necesidad de discretizar las acciones de control.

2.3.4. Aplicación de aprendizaje reforzado en el control de turbinas eólicas

El empleo de estrategias de aprendizaje reforzado para el control de turbinas eólicas ha sido ampliamente investigado en el ámbito académico. [63] presenta un sistema de control basado en Deep Q-learning para maximizar la potencia generada en una turbina eólica. Por otro lado, en [59] se comparan distintas funciones de recompensa para el control de una turbina eólica mediante Q-Learning.

El trabajo presentado en [63] explora el uso del aprendizaje por refuerzo para optimizar el control de aerogeneradores, con el objetivo de maximizar la captura de energía y minimizar las cargas estructurales. En concreto, los autores aplican Double Deep Q-Learning para controlar la potencia de salida del aerogenerador, utilizando como variables de control la velocidad del rotor, el ángulo de las palas y la guiñada de la orientación de la góndola. El trabajo de Xie et al. [77] aborda los problemas de control del par y el paso asociados a los convertidores de energía eólica. Su diseño aplica un sistema de control en tiempo real basado en aprendizaje por refuerzo, el cual combina redes neuronales profundas (DNN, del inglés, Deep Neural Network) y control predictivo por modelo (MPC, del inglés, Model Predictive Control).

Por otro lado, cuando se emplean métodos de aprendizaje reforzado discretos, es posible emplear técnicas adaptativas de discretización del espacio de estados.

En la literatura se han propuesto técnicas de discretización del espacio de estados con el objetivo de encontrar la partición óptima del espacio de estados durante el proceso de entrenamiento de algoritmos de aprendizaje por refuerzo. Sin embargo, el uso de estas técnicas aumenta los requisitos computacionales del controlador durante el proceso de entrenamiento y hace que los resultados sean menos interpretables.

Varios estudios destacan las ventajas de la discretización de estados en el aprendizaje por refuerzo. Por ejemplo, el trabajo presentado en [38] utiliza la técnica de discretización de estados de tile coding para reducir las dimensiones del espacio de estados en una aplicación robótica. Del mismo modo, el artículo en [60] muestra cómo el uso de técnicas de discretización adaptativas mejora el rendimiento de controladores basados en Q-Learning, en comparación con una discretización uniforme. Sin embargo, el uso de estas técnicas en el control de potencia de un aerogenerador permanece en gran medida inexplorado.

2.4. Aprendizaje federado

El empleo de técnicas de aprendizaje por refuerzo permite realizar el control de sistemas sin la necesidad de disponer de un modelo matemático exacto de su dinámica. Sin embargo, requieren de una gran cantidad de datos para que el algoritmo converja a la política óptima.

Dado que, por lo general, los aerogeneradores son empleados en granjas eólicas donde todos los sistemas operan en condiciones muy similares, una posible solución para aumentar los datos de entrenamiento es emplear información proveniente de múltiples turbinas del mismo tipo para el entrenamiento de un modelo global, el cual pueda ser posteriormente distribuido a las turbinas para realizar el control.

Sin embargo, las políticas de datos de los fabricantes de turbinas hacen que los datos necesarios para el entrenamiento de la turbina no puedan ser compartidos, ya que estos pueden contener información sensible sobre el sistema.

El aprendizaje federado (FL, del inglés, Federated Learning) es una técnica de aprendizaje automático en la que agentes colaboran con tal de entrenar un modelo global de forma descentralizada y sin la necesidad de compartir datos sensibles en el proceso, garantizando así la seguridad de los datos y modelos. La figura 2.12 muestra la arquitectura general de un sistema de entrenamiento mediante aprendizaje federado.

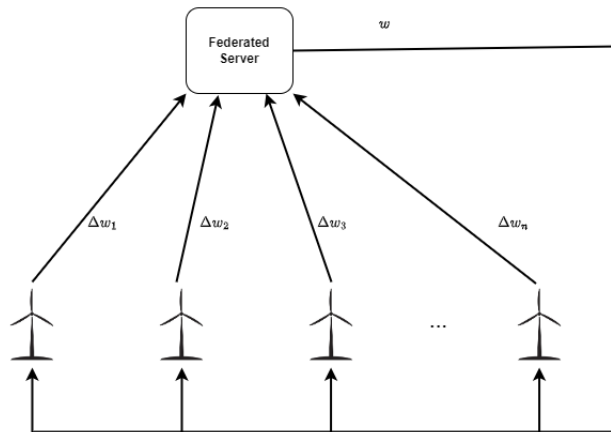


Figura 2.12: Arquitectura de aprendizaje federado.

Desde su introducción original en 2016, el aprendizaje federado ha ganado cada vez más popularidad en múltiples sectores debido a su capacidad para entrenar modelos de aprendizaje automático de forma distribuida, a la vez que se preserva la privacidad y seguridad de los datos de entrenamiento [44]. Sin embargo, a pesar de su creciente adopción en múltiples campos [74], incluso en energías renovables [31], su aplicación en el campo del control es aún relativamente limitada. En esta sección se estudiarán los paradigmas y estrategias de

aprendizaje federado más empleados en la actualidad.

2.4.1. Paradigmas de aprendizaje federado.

Dependiendo de cómo se distribuyan los datos entre las entidades del sistema, es posible distinguir entre aprendizaje federado vertical y horizontal [80].

En el paradigma de aprendizaje federado horizontal, todas las entidades del sistema tienen datos con las mismas características, pero la información es distinta para cada individuo. De esta forma, los participantes entrenan localmente el modelo empleando sus datos individuales. Posteriormente, los parámetros de los modelos locales son enviados al servidor federado, el cual los agrega para obtener el modelo global. Finalmente, el modelo global es distribuido entre los participantes. Este proceso se repite hasta que los modelos individuales convergen al modelo global.

Por otro lado, en el aprendizaje federado vertical, todos los nodos del sistema tienen datos sobre las mismas entidades, pero con características distintas, de modo que es necesaria la colaboración entre los nodos del sistema para entrenar el modelo.

En el caso de la granja eólica, todas las turbinas compartirán las mismas características, variando únicamente los datos recopilados durante su funcionamiento. De esta forma, el paradigma empleado será aprendizaje federado horizontal.

2.4.2. Estrategias de aprendizaje federado

Dependiendo del paradigma empleado, existen multitud de estrategias para la implementación de aprendizaje federado, cuyo objetivo es entrenar a un modelo global a partir de los datos distribuidos entre las entidades del sistema, sin compartir información sensible en el proceso [39]. En esta sección se tratarán las estrategias de aprendizaje más relevantes para el paradigma de aprendizaje federado horizontal.

FedAvg (Federated Averaging)

En la estrategia de aprendizaje de Federated Averaging (FedAvg), las entidades del sistema entrenan copias locales del modelo global con sus datos de operación individuales. A continuación, se envían los parámetros de su modelo local a un servidor central, denominado servidor federado. Una vez recibidos los parámetros de todas las turbinas, el servidor federa-

do los agrega realizando un promedio ponderado de los mismos y actualiza el modelo global. Finalmente, el modelo actualizado es distribuido entre las entidades del sistema, repitiéndose el proceso hasta lograr la convergencia [70].

La función de agregación empleada en FedAvg es la siguiente:

$$w_{t+1} \leftarrow \sum_{k=0}^N \frac{n_k}{n} w_{t+1}^k \quad (2.26)$$

Donde:

- w_{t+1} : Parámetros del modelo global actualizado.
- w^k : Parámetros del modelo de la entidad k .
- n_k : Tamaño del conjunto de datos de la entidad k .
- $n = \sum_{i=1}^N n_k$: Tamaño del conjunto de datos total de todas las entidades.

FedAvg consiste en una de las estrategias de aprendizaje federado más ampliamente utilizadas debido a la escalabilidad y robustez de su arquitectura [80].

FedProx (Federated Proximal)

La estrategia de Federated Proximal (FedProx) es una extensión de la técnica de FedAvg empleada en entornos con heterogeneidad entre los datos de los nodos del sistema. De esta forma, FedProx introduce un término de regulación que limita cuánto puede desviarse un modelo local w_i del modelo global w [39].

La función de agregación empleada en FedProx es la siguiente:

$$\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2 \quad (2.27)$$

Donde:

- $F_k(w)$: Es la función de pérdida que el cliente k trata de minimizar.
- w : Son los parámetros del modelo de la entidad i .
- w^t : Son los parámetros del modelo global en el instante t .

- μ : Son los parámetro que determina la penalización por desviarse del modelo global.
- $h_k(w; w^t)$: Es la función objetivo que el cliente k trata de minimizar.

Como resultado, este método de aprendizaje permite mejorar la eficiencia en la comunicación y reducir los tiempos de entrenamiento en comparación con FedAvg [14].

Scaffold (Stochastic Controlled Averaging for Federated Learning)

Al igual que FedProx, la estrategia de entrenamiento de Scaffold mejora los tiempos de convergencia del algoritmo FedAvg para sistemas con datos heterogéneos. Para ello, esta estrategia emplea un conjunto de variables de control que corrigen las desviaciones del modelo local con respecto al modelo global [37]. Las variables de control representan cuánto diverge el modelo local del gradiente global esperado.

$$w_t^k \leftarrow w_t - \eta(\nabla f_k(w_t) + c - c_k) \quad (2.28)$$

Donde:

- w_t : Son los parámetros del modelo global.
- $f_k(w_t)$: Es el gradiente local del cliente k .
- η : Tasa de aprendizaje.
- c_k : Variable de control del cliente k .
- c : Variable de control del servidor que captura la desviación entre los clientes.

Scaffold consigue que los modelos locales no difieran significativamente del modelo global, lo cual mejora los tiempos de entrenamiento en sistemas para los que existe una gran heterogeneidad de datos entre los nodos.

Clustered Federated Learning (CFL)

Al igual que sucedía con las técnicas anteriores, Clustered Federated Learning (CFL) se emplea en sistemas con una gran heterogeneidad entre los datos de los nodos. CFL realiza agrupaciones de nodos con datos similares y entrena modelos separados para cada grupo [78].

Al inicio del entrenamiento, todos los nodos contribuyen en el entrenamiento de un modelo global. Conforme el entrenamiento avanza, los nodos son divididos en subgrupos dependiendo de la similitud de sus modelos locales. A partir de ese momento, se entrenan modelos separados para cada subgrupo.

Una posible separación de los grupos se puede realizar teniendo en cuenta la distancia entre los gradientes locales y el rendimiento del modelo en los datos locales, como se muestra en [52].

2.4.3. Aplicación del aprendizaje federado en el control de turbinas eólicas

En el artículo presentado en [73], los autores analizan los últimos avances relacionados con la aplicación del aprendizaje federado en el campo del control, donde se destacan los beneficios potenciales de estas técnicas en cuanto a adaptabilidad, escalabilidad, generalización y preservación de privacidad. Del mismo modo, se realiza un análisis de los problemas actuales en relación a la sobrecarga en la comunicación, los datos independientes e idénticamente distribuidos y las limitaciones en la robustez de los métodos de agregación.

En el trabajo presentado en [56], técnicas de aprendizaje reforzado son empleadas en combinación con aprendizaje por refuerzo para mejorar el rendimiento en Vehículos Guiados Autónomos (AGVs, del inglés, Autonomous Guided Vehicles). El artículo demuestra cómo pueden emplearse técnicas de Aprendizaje Federado para aumentar el rendimiento en controladores basados en aprendizaje por refuerzo sin necesidad de compartir datos sensibles en el proceso. El artículo sugiere posibles aplicaciones del FL en otros dominios de control, como los sistemas de control de turbinas eólicas.

FL también ha sido aplicado para la predicción en el campo de la energía eólica [10]. Como ejemplo, en el artículo [6] se muestra un modelo basado en el aprendizaje federado para la predicción de la energía eólica a partir de la velocidad y la dirección del viento.

Aunque hay estudios recientes que exploran el uso del Aprendizaje Federado en sistemas de control, así como el uso de controladores de Aprendizaje por Refuerzo para la estabilización de la potencia y la reducción de la carga de las turbinas eólicas. La combinación de estas técnicas en el ámbito de los aerogeneradores permanece en gran medida inexplorada.

2.5. Modelo de turbina eólica

Las dinámicas de la turbina eólica son simuladas empleando el modelo de turbina de 1,5MW presentado en [69], el cual consiste en una simplificación del modelo dinámico de una turbina eólica similar a la mostrada en la figura 2.13, donde se desprecia el amortiguamiento externo y se simplifica la dinámica del generador.

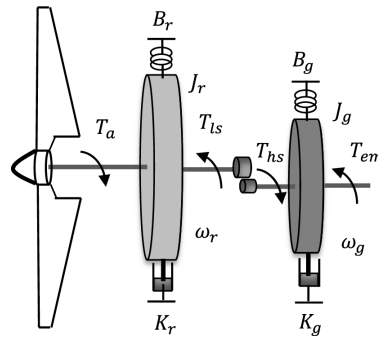


Figura 2.13: Modelo dinámico de una turbina eólica [15].

El Tip Speed Ratio (λ), que representa la fracción de la potencia del viento que es extraída por el rotor [62], se calcula a partir de la velocidad angular del rotor w_r y la velocidad del viento (V_{viento}) mediante la siguiente expresión:

$$\lambda = \frac{R \cdot w_r}{V_{viento}} \quad (2.29)$$

Donde R es el radio del motor expresado en metros y w es la velocidad angular del rotor en rad/s .

A partir de los valores del Tip Speed Ratio (λ) y del ángulo de pala θ (en grados, $^\circ$), se calcula el coeficiente de potencia C_p . Para ello se emplea una tabla de consulta que define la eficiencia del aerogenerador para cada posible configuración de funcionamiento.

Una vez determinado el coeficiente de potencia C_p del aerogenerador, se calcula la potencia aerodinámica P_{aero} , expresada en W , mediante la siguiente expresión:

$$P_{aero} = 0,5\rho AV^3 C_p(\lambda, \theta) \quad (2.30)$$

Donde ρ es la densidad del aire en Kg/m^3 y $A = \pi R^2$ es el área de barrido del rotor en m^2 .

La potencia aerodinámica P_{aero} (en vatios, W), junto con la velocidad del rotor w_r (en radianes

por segundo, rad/s), puede utilizarse para calcular el par aerodinámico T_a (en newton-metro, $N \cdot m$). Para ello, se utiliza la siguiente ecuación:

$$T_a = \frac{P_{aero}}{w_r} \quad (2.31)$$

El par de reacción del generador T_{ls} (en newton-metro, $N \cdot m$) está relacionado con el par aerodinámico y la velocidad del rotor mediante la siguiente expresión:

$$J_r \dot{w}_r = T_a - T_{ls} \quad (2.32)$$

Donde J_r es la inercia del motor en $Kg \cdot m^2$ y T_{ls} es el par de reacción del generador en $N \cdot m$.

Por otro lado, el par del generador T_{hs} se define como el par de reacción escalado por la relación de transmisión N de la caja de engranajes:

$$T_{ls} = \frac{T_{hs}}{N} \quad (2.33)$$

Del mismo modo, la velocidad angular del rotor w_r y del generador w_g (en radianes por segundo, rad/s) queda relacionada por la siguiente expresión:

$$w_g = N w_r \quad (2.34)$$

Por último, la potencia de salida generada por la turbina se calcula mediante la siguiente expresión:

$$P_{gen} = \eta_g P_{mech} \quad (2.35)$$

Donde η_g es la eficiencia del generador y P_{mech} (en vatios, W) es la potencia mecánica de entrada al generador, definida como:

$$P_{mech} = T_{hs} w_g \quad (2.36)$$

Además, se añade un limitador de velocidad dinámico de $0,17rad/s$ en la entrada de la

planta con el objetivo de simular cierto retardo del actuador correspondiente a la variable de entrada θ .

$$|\dot{\theta}| \leq 0,17 \quad (2.37)$$

Se utilizan los mismos parámetros de modelo y valores de C_p presentados en el modelo base [69]. Los más relevantes de estos parámetros se resumen en la siguiente tabla:

Tabla 2.1: Parámetros del modelo de turbina eólica.

Parámetro	Valor	Unidades	height	R	35	m
J	3410432	Kgm^2				
N	88	—				
ρ	1,225	Kgm^3				
η_g	0,95	—				

2.6. Modelo de viento

Para representar correctamente las condiciones del entorno de operación de la turbina, es necesario implementar un modelo del viento.

El modelo de viento se realiza empleando la distribución de Weibull [54]. Es una distribución de probabilidad continua empleada comúnmente para el modelado de la velocidad del viento, y que queda definida por la siguiente expresión:

$$f(x; \eta, k) = \frac{k}{\eta} \left(\frac{x}{\eta} \right)^{k-1} e^{-\left(\frac{x}{\eta}\right)^k} \quad (2.38)$$

Donde x es la variable aleatoria, η determina la escala de la distribución y k su forma.

En la figura 2.14 se representa la distribución estadística de 1000 variables generadas aleatoriamente mediante la distribución de probabilidad de Weibull para un valor de $\eta = 12$ y $k = 2$. Estos resultados también pueden ser interpretados como un entorno en el que la velocidad del viento más común es de $9m/s$, y existe la posibilidad de ráfagas comprendidas entre $0m/s$ hasta $30m/s$.

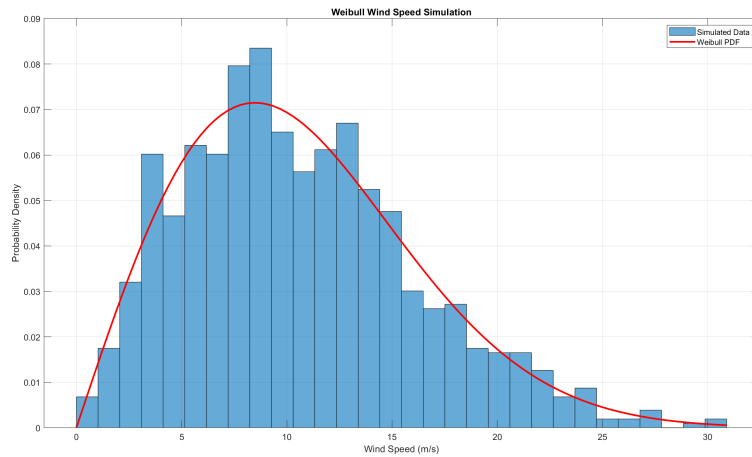


Figura 2.14: Distribución probabilística de Weibull con $\eta = 12$ y $k = 2$.

La velocidad del viento en un instante de simulación se actualiza a partir del valor ponderado entre la velocidad actual y el nuevo valor aleatorio obtenido mediante la distribución de Weibull.

$$V_{wind} = \alpha V_{wind} + \beta f(x; \eta, k) \quad (2.39)$$

Donde $f(x; \eta, k)$ es la distribución de Weibull con parámetros de escala η y forma k . Y α y β son factores de ponderación.

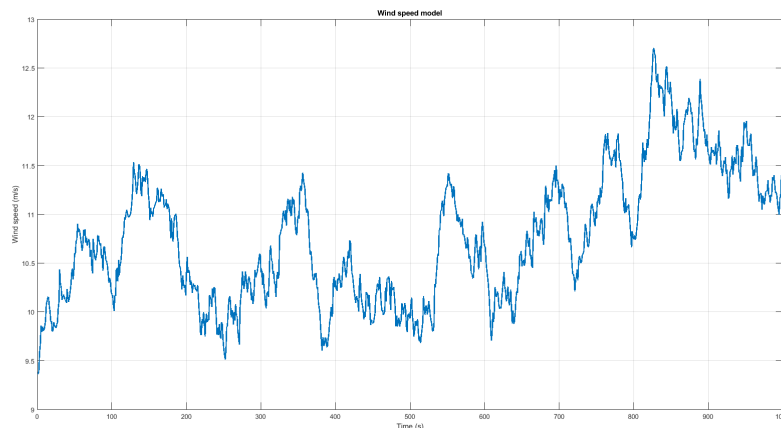


Figura 2.15: Simulación del viento empleando la distribución probabilística de Weibull con $\eta = 12$ y $k = 2$. Empleando factores de ponderación de $\alpha = 0,9$ y $\beta = 0,1$

La figura 2.15 muestra el resultado de la simulación del modelo durante un periodo de 1000s para unos factores de ponderación de $\alpha = 0,9$ y $\beta = 0,1$. Como se puede observar, el modelo

del viento aproxima de forma adecuada las condiciones de viento a las que se puede ver sometida una turbina eólica.

Capítulo 3

Propuesta de Control

En este capítulo se propone un controlador basado en Q-learning para la estabilización de potencia de una turbina eólica, aplicando distintas discretizaciones del espacio de estados y funciones de recompensa. Adicionalmente, se emplea aprendizaje federado para el entrenamiento distribuido de un modelo global, garantizando la privacidad de los datos empleados en el entrenamiento.

3.1. Introducción a la propuesta de control

El objetivo de este trabajo es demostrar la eficacia del aprendizaje federado para mejorar el rendimiento de controladores basados en aprendizaje reforzado aplicados a la estabilización de potencia en turbinas eólicas.

Para ello, se implementa un sistema de control basado en Q-learning para el control de paso de una turbina eólica. Para reducir los tiempos de entrenamiento y minimizar el error en el estado estacionario, se proponen varios métodos de discretización del espacio de estados. Posteriormente, se implementa un algoritmo de aprendizaje federado con el objetivo de entrenar un modelo de control global basado en Q-learning a partir de los datos de operación de tres turbinas, a la vez que se mantiene la privacidad de los datos.

3.2. Análisis de las correlaciones de entrada y salida del sistema

Los sistemas de control mediante aprendizaje por refuerzo no necesitan disponer de un modelo exacto de la planta para realizar el control. Sin embargo, el conocimiento de ciertas características dinámicas del sistema proporciona información útil a la hora de determinar los parámetros de los modelos. En esta sección se realiza un análisis de correlación entre los datos de entrada y salida del sistema mostrado en la sección 2.5, con el objetivo de obtener información sobre su comportamiento.

Una primera aproximación en el diseño de la señal de entrada empleada para excitar el sistema es una señal PRBS (del inglés, Pseudo-Random Binary Sequence) del tipo MLBS (del inglés, Maximum Length Binary Sequence). Consiste en una señal binaria pseudoaleatoria, periódica y de excitación persistente, la cual presenta un espectro de potencia que le permite excitar la mayor parte de los componentes frecuenciales del sistema a la vez que es amigable con la planta.

Sin embargo, dado que se dispone de un modelo aproximado de la planta y el índice de amigabilidad con la planta no es relevante en este caso, se opta por emplear una señal de entrada aleatoria, la cual consta de un espectro de potencia constante para todas las frecuencias, convirtiéndola en una señal de entrada ideal para la identificación de la planta.

3.2.1. Correlación entre en ángulo de las palas y la potencia generada

Los datos empleados para el estudio de la correlación entre el ángulo de las palas y la potencia de salida se obtienen fijando la velocidad de viento a un valor constante de $13m/s$, y excitando la planta con una señal aleatoria de ángulo de ataque, con una media y varianza de 5 grados.

A partir de los datos obtenidos durante la experimentación, es posible estudiar la correlación cruzada entre el valor del ángulo de las palas θ y la potencia de salida P_{out} , obteniendo los resultados mostrados en la figura 3.1.

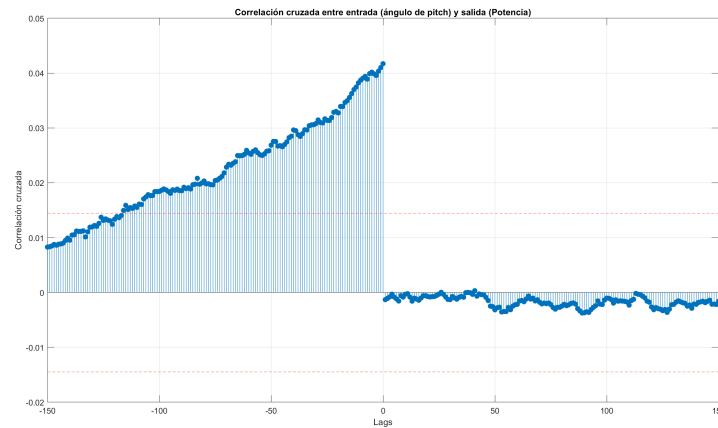


Figura 3.1: Función de correlación cruzada entre el ángulo de las palas (θ) y la potencia de salida (P_{out}).

Analizando los resultados obtenidos, se puede observar que la función de correlación presenta su pico máximo para el valor de retardo $\tau = 0$, de modo que se trata de un sistema sin retardo.

Por otro lado, existe una correlación entre el valor de P_{out} y los valores de θ pasados, indicando que los valores de pitch pasados tienen un impacto acumulativo en la potencia.

La correlación de la entrada y la salida para valores tan altos de retardo se debe a la presencia de inercia en el rotor de la turbina, lo cual provoca que la acción del actuador persista en el tiempo.

3.2.2. Correlación entre la velocidad del viento y la potencia generada

Los datos empleados para el estudio de la correlación entre el viento y la potencia se obtienen fijando el ángulo de las palas a un valor constante de 7 grados, y estudiando los datos de la potencia de salida para una señal de viento aleatoria de media $10m/s$ y una varianza de $7m/s$, obteniendo así la función de correlación mostrada en la figura 3.2.

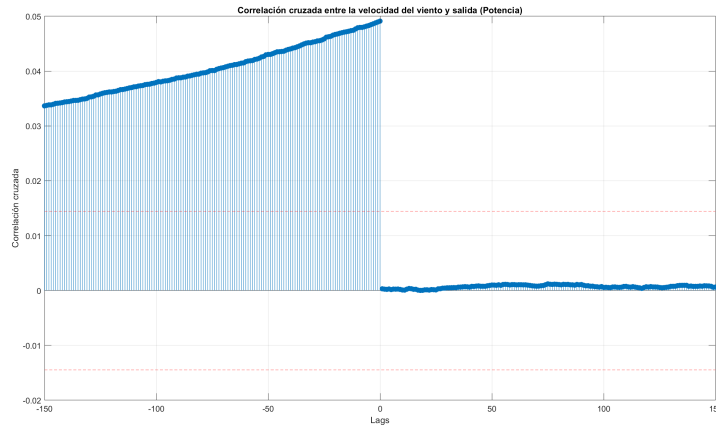


Figura 3.2: Función de correlación cruzada entre la velocidad del viento V_{wind} y la potencia de salida P_{out} .

Los resultados obtenidos muestran un comportamiento similar a la relación entre el ángulo de las palas y la potencia generada. Sin embargo, se puede observar cómo la correlación negativa entre la velocidad del viento y la potencia decae más lentamente que en el caso anterior. Esto indica que la influencia del viento en la salida persiste durante un tiempo mayor que la influencia del ángulo de las palas.

3.3. Requisitos de control

Tal y como se ha explicado anteriormente, se emplea un sistema de control de paso basado en Q-learning para la estabilización de la potencia de salida de la turbina.

Dado que el objetivo fundamental de este trabajo es la evaluación de la mejora del controlador al aplicar aprendizaje federado y métodos de discretización del espacio de estados, no se considera necesaria la definición de requisitos de control específicos en relación con la estabilidad y robustez del sistema.

De esta forma, los objetivos fundamentales de control del sistema serán la minimización del error P_{error} entre la potencia generada por la turbina P_{out} y la potencia deseada P_{rated} , a la vez que se minimiza la acción del control. Se definen así los principales objetivos de control:

- El controlador minimizará la desviación de potencia $|P_{rated} - P_{out}|$ con el objetivo de maximizar la captación de energía de la turbina.
- El controlador mantendrá la velocidad del rotor dentro de sus límites de actuación con el objetivo de evitar daños estructurales en la turbina.

Cabe destacar que uno de los objetivos de control deseables para cualquier turbina eólica es la minimización de las cargas mecánicas en el rotor. Sin embargo, dado que el modelo descrito en la sección 2.5 no contiene esta dinámica, este requisito no se incluye de forma explícita en los objetivos de control ¹.

3.4. Control de paso de una turbina eólica

El sistema de control permite mantener la potencia de salida deseada mediante el ajuste del ángulo de ataque de las palas de la turbina. En este caso, el controlador es empleado para la estabilización de potencia de una turbina eólica operando en la región 3 de operación.

La variable de control es el ángulo de las palas y el objetivo de control es mantener la potencia de salida de la turbina lo más cerca posible de la potencia nominal, de forma que se maximice la potencia de salida sin comprometer los límites de seguridad de la turbina.

La figura 3.3 muestra la arquitectura del controlador de tipo Q-Learning propuesto. En este modelo, el estimador de estado determina el estado actual del aerogenerador s_t a partir del error de potencia del aerogenerador P_{error} y la velocidad actual del viento V_{wind} .

Una vez obtenido el estado del sistema, el selector de acciones determina la acción de control a_t que maximiza la recompensa esperada a partir de los valores de la tabla Q. Además, durante el entrenamiento, la acción de control no siempre es la óptima, con tal de permitir al agente explorar nuevas partes del espacio de soluciones. El concepto de exploración y explotación será tratado con más detalle en la sección 3.4.3.

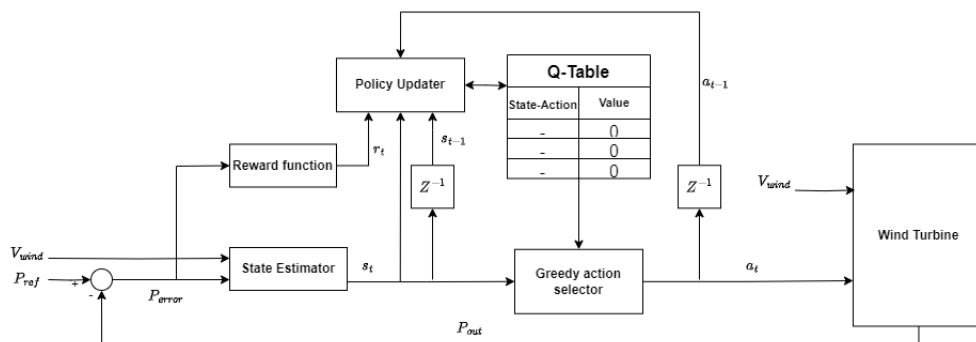


Figura 3.3: Diagrama de las entradas y salidas del sistema.

La actualización de la política de control, representada por la tabla Q, se realiza basándose en la señal de recompensa recibida, así como en la acción y el estado anteriores y actuales

¹La minimización de la acción de control tiene como objetivo reducir el estrés en la turbina, lo cual implica la reducción de las cargas sobre el rotor.

del sistema [59]. De esta forma, tras realizar una acción, el sistema de actualización de la política actualiza la tabla Q de forma análoga a como se explicó en la sección 2.3.3.

3.4.1. Discretización del espacio de estados y acciones

En la sección 2.3.3 se detallaron las características de las distintas técnicas de control basadas en aprendizaje por refuerzo en relación al espacio de estados y acciones. En Q-learning, el empleo de la tabla Q hace necesario el uso de estados y acciones discretos.

En el caso del control de una turbina eólica, el espacio de estados consta del conjunto de valores posibles de las variables de entrada correspondientes al error de potencia P_{error} y a la velocidad del viento V_{wind} , mientras que la variable de salida son los posibles valores del ángulo de ataque de las palas θ .

Discretización de la variable V_{wind}

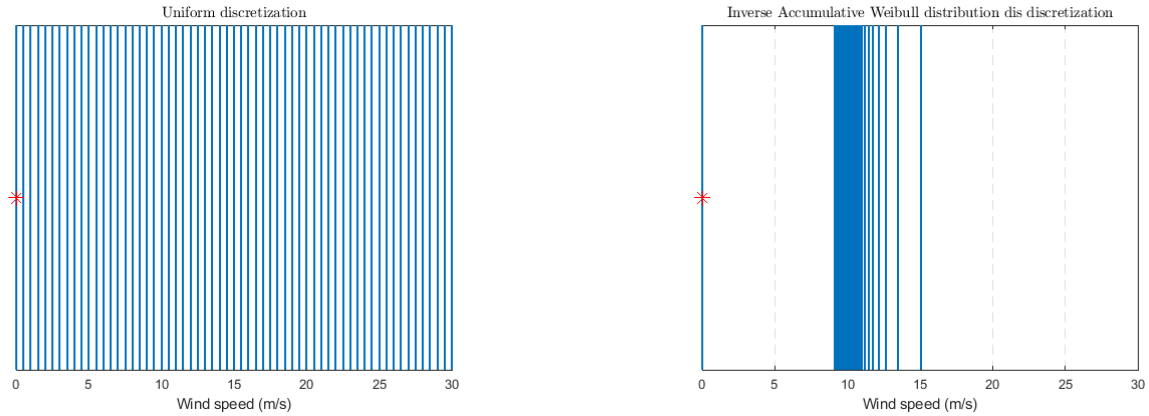
La variable V_{wind} presenta una distribución de probabilidad de Weibull definida para valores de entre $0m/s$ y $30m/s$. Una primera aproximación para la discretización de la variable es emplear una distribución uniforme con incrementos de $0,5m/s$ para el intervalo de viento esperado.

$$Q(i; \delta) = \delta \cdot i = 1, 2, \dots, n \quad (3.1)$$

Por otro lado, es posible emplear el conocimiento del modelo del viento con el objetivo de definir una función de discretización más adecuada. Se puede usar la función de distribución acumulativa inversa de Weibull para discretizar la variable V_{wind} [3].

$$Q(i; k, \eta) = \eta(-\ln(1 - i))^{1/k}, i = 1, 2, \dots, n \quad (3.2)$$

A continuación se muestran de forma gráfica ambas discretizaciones:



(a) Distribución uniforme de valores con $\delta = 0,5$.

(b) CDF inversa de Weibull con $\eta = 12$ y $k = 2$.

Figura 3.4: Funciones de discretización de la variable de estado V_{wind} .

Como se puede observar en los resultados obtenidos, la transformada inversa de Weibull produce una distribución mucho más representativa del modelo de viento empleado, reduciendo significativamente el espacio de búsqueda. Por lo tanto, esta es la función empleada para la discretización de la variable V_{wind} .

Discretización de la variable P_{error}

La naturaleza de la variable del error de potencia P_{error} hace posible el empleo de múltiples técnicas de discretización.

En este caso no es posible conocer a priori la distribución estadística de la P_{error} , de forma que pueda emplearse para discretizar la variable. Se emplean entonces otras opciones de discretización.

El objetivo del controlador Q-Learning es reducir el error entre la potencia nominal y la potencia de salida. La variable de error de potencia de salida debe discretizarse, por lo tanto, de forma que los valores cercanos a cero contengan un conjunto más denso de estados posibles, permitiendo al controlador realizar ajustes más finos para mantener el error de potencia en el punto de equilibrio ($P_{error} = 0$). Por otro lado, a medida que aumenta el error de potencia de salida, la necesidad de ajustes de control específicos se vuelve menos relevante, lo que permite reducir la granularidad de los estados discretos.

Una primera aproximación a la discretización de la variable de entrada P_{error} consiste en aplicar una discretización uniforme con una granularidad dada δ , para un valor máximo de P_{error} de $1000kW$.

$$Q(i; \delta) = \delta \cdot i = -1000, -999, \dots, 0, \dots, 999, 1000 \quad (3.3)$$

Este enfoque produce 2001 estados discretos que van de $-1000kW$ a $1000kW$, cuando se utiliza un valor de $\delta = 1kW$.

Para reducir el número de estados discretos y el error en el estado estacionario, se emplean técnicas de discretización más avanzadas.

Una posible técnica de discretización de la variable de entrada P_{error} es utilizar una función exponencial. De esta forma, el valor de P_{error} se aproxima al valor más cercano de una serie exponencial que parte de un valor de error mínimo δ .

$$Q(i; \delta, k) = \text{sign}(i) \cdot \delta \cdot e^{\psi \cdot (|i|-1)}, \quad i \in \{-n, \dots, n\} \quad (3.4)$$

Esta aproximación reduce significativamente la cantidad de estados necesarios para representar la variable P_{error} en comparación con las distribuciones uniformes, lo que permite un tiempo de convergencia más rápido a la vez que mantiene la precisión alrededor del punto de equilibrio.

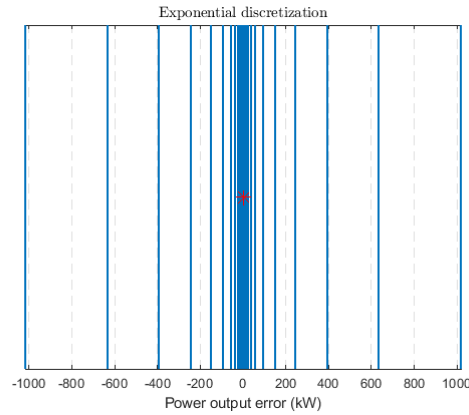


Figura 3.5: Representación de la discretización de la variable P_{error} utilizando la función de discretización 3.4 con $\delta = 0,001$ y $\psi = 0,477$.

La figura 3.5 muestra cómo los estados se agrupan en torno al valor 0 en el error de potencia, volviéndose más dispersos a medida que aumenta el error de salida de potencia. El error mínimo en estado estacionario viene determinado por el valor de δ , siendo $1W$ en este caso.

Otra posibilidad para realizar la discretización es utilizar una función cuadrática.

$$Q(i; \delta) = \text{sgn}(i) \cdot \delta \cdot i^2, \quad i \in \{-n, \dots, n\} \quad (3.5)$$

En comparación con la discretización exponencial, este método produce una mayor densidad de estados para valores grandes de error P_{error} , lo que permite al controlador tener más capacidad de actuación en el transitorio. Sin embargo, esto también aumenta el número de estados que el controlador debe aprender, aumentando así el tiempo de entrenamiento.

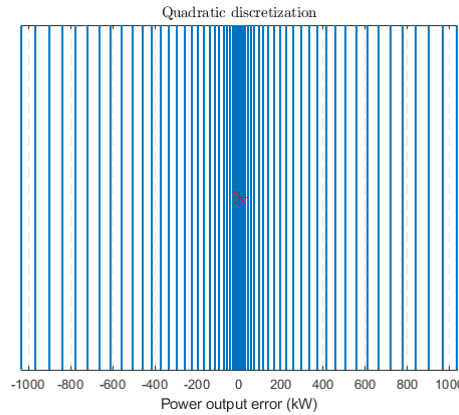


Figura 3.6: Representación de la discretización de la variable P_{error} mediante la función de discretización 3.5 con $\delta = 1, 15$.

La figura 3.6 muestra cómo la discretización cuadrática obtiene una distribución más uniforme para los posibles valores de P_{error} , enfatizando los valores alrededor del punto de equilibrio.

El error mínimo obtenido por el método de discretización cuadrático depende del valor de δ , siendo $1, 2kW$ en la discretización propuesta.

Además de los métodos descritos anteriormente, se estudia el uso de métodos de compansión para realizar la discretización de estados.

Los métodos de compansión son técnicas no lineales de compresión de datos ampliamente utilizadas en sistemas PCM (del inglés, Pulse Code Modulation) para reducir el número de bits del código digital de salida [75]. Aplicados a Q-Learning, estos métodos proporcionan un enfoque estructural para discretizar el espacio de estados. Los métodos de companding explorados son la ley μ y la ley A.

Mientras que las funciones de compansión como la ley μ se utilizan para comprimir la información, en este caso, el método de expansión (o ley μ inversa) el que se aplica para discretizar la variable de entrada P_{error} .

$$Q(i; \mu) = \operatorname{sgn}(i) \frac{(1 + \mu)^{|i|} - 1}{\mu}, \quad -1 \leq i \leq 1 \quad (3.6)$$

En comparación con los métodos de discretización anteriores, la discretización basada en la ley μ logra una buena resolución tanto para valores cercanos al equilibrio como para valores grandes de error, como se muestra en la figura 3.7.

El error mínimo en estado estacionario para la discretización mediante la ley μ , con $\mu = 4096$, es de $0,35kW$.

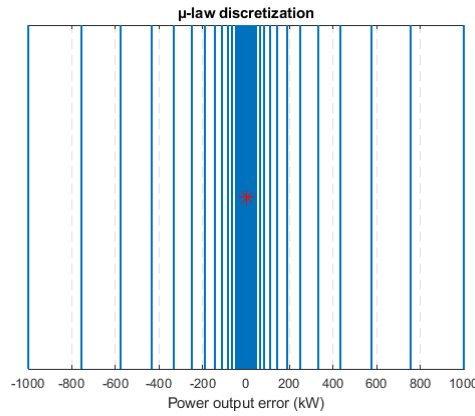


Figura 3.7: Representación de la discretización de la variable P_{error} mediante la función de discretización 3.6, con $\mu = 4096$.

De forma similar al método de discretización basado en la ley μ , la transformada inversa del método de compansión de la ley A también proporciona una forma de realizar la discretización de la variable de entrada P_{error} .

$$Q(i; A) = \operatorname{sgn}(i) \begin{cases} \frac{|i|(1+\ln A)}{A}, & |i| < \frac{1}{1+\ln A} \\ \frac{e^{-1+|i|(1+\ln A)}}{A}, & \frac{1}{1+\ln(A)} \leq |i| < 1 \end{cases} \quad (3.7)$$

La figura 3.8 muestra la discretización de la variable P_{error} tras aplicar la inversa de la ley A. Los resultados obtenidos para un valor de $A = 0,467$ son similares a la discretización obtenida utilizando la ley μ , teniendo una buena distribución de estados para todos los valores posibles de P_{error} , con un error mínimo de $0,5kW$.

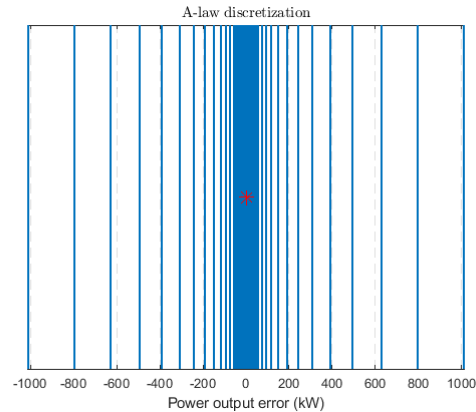


Figura 3.8: Representación de la discretización de la variable P_{error} mediante la función de discretización 3.7, con $A = 0,467$.

Por último, cabe señalar que la variación de los parámetros en cada uno de los métodos de discretización mencionados modificará también las características de la función resultante y, por tanto, la discretización de los estados. Así, el mejor método de discretización dependerá de los objetivos de control y de las características del sistema. En la sección 4.3 se realizará una comparación entre los distintos métodos de discretización.

Discretización de la variable de control θ

La variable de actuación del ángulo de ataque de las palas θ determina el ángulo de ataque de las palas que produce una estabilización de la potencia de salida de la turbina.

En este caso, dadas las restricciones de control y el conocimiento de la dinámica del sistema, se restringe la actuación de las palas a ángulos dentro del intervalo de $[0, 30]$ grados. Esta restricción en el espacio de acción se realiza con el objetivo de reducir la complejidad del problema y disminuir el tiempo necesario para aprender la política óptima.

Teniendo en cuenta los valores medios de la distribución del viento, es posible definir una función de discretización que enfatice los ángulos cercanos a las condiciones de operación que estabilizan la potencia del sistema en estas condiciones. Sin embargo, dado que no se dispone de este conocimiento a priori, y el sistema de control debe funcionar correctamente para cualquier condición de viento, no se realiza ninguna discretización específica para el problema de control.

De esta forma, la variable de salida θ es discretizada en valores uniformes de 0,5 grados en el intervalo de valores $[0, 30]$:

$$Q(i) = 0,5 \cdot i, i = 1, 2, \dots, n \quad (3.8)$$

Obteniéndose la representación gráfica mostrada en la figura 3.9.

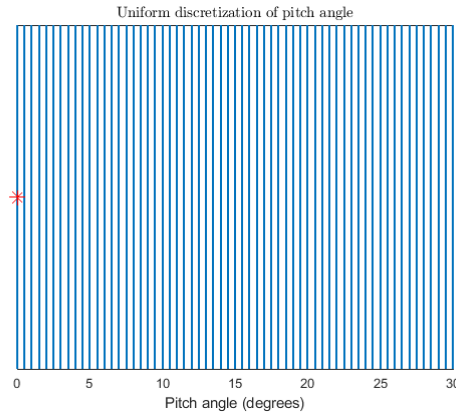


Figura 3.9: Discretización de la variable de salida a mediante la función $Q(i) = 0,5 \cdot i$.

3.4.2. Funciones de recompensa

La función de recompensa se utiliza para incentivar al agente a aprender estrategias de control óptimas, fomentando comportamientos que minimicen el error entre la potencia de salida P_{out} y la potencia nominal P_{target} , a la vez que se minimiza el esfuerzo de control.

La naturaleza de los sistemas de aprendizaje reforzado dificulta la obtención de una función de recompensa que cumpla con los objetivos de control, dado que no es posible establecer una relación directa entre ambos. Por lo tanto, es conveniente tener en cuenta las siguientes consideraciones generales a la hora de diseñar una función de recompensa:

- Debe reflejar claramente los objetivos de control. En este caso será la estabilización de la potencia en su valor nominal, y la minimización del esfuerzo de actuación.
- Debe existir un balance entre recompensas densas y dispersas. Las recompensas densas deben guiar al agente en el proceso de búsqueda, mientras que las recompensas dispersas indican hitos importantes.
- Debe existir una diferenciación clara entre buenas y malas acciones.
- Debe ser robusta a variaciones en el entorno.

Una primera aproximación a la función de recompensa es el empleo del error absoluto [59], es decir:

$$R = -|P_{error}| \quad (3.9)$$

El principal problema de la función anterior es que no tiene en cuenta la actuación del sistema, de modo que el controlador no aprende estrategias de control que minimicen la actuación del sistema. Es posible ampliar la función para que se penalicen cambios bruscos en el ángulo de las palas.

$$R = -\alpha|P_{error}| - \beta|\Delta\theta| \quad (3.10)$$

Donde θ es el ángulo de las palas, y α y β son factores de ponderación.

Por último, con el objetivo de introducir recompensas discretas que permitan al controlador encontrar la política de control óptima, se introduce el valor de recompensa máximo R_{max} cuando la potencia de salida de la turbina se encuentra dentro de un intervalo $[-\epsilon, \epsilon]$ de la potencia objetivo. Esta es la función de recompensa finalmente usada en el controlador.

$$R = \begin{cases} R_{max} & |P_{error}| < \epsilon \\ -\alpha|P_{error}| - \beta|\Delta\theta| & |P_{error}| \geq \epsilon \end{cases} \quad (3.11)$$

3.4.3. Configuración de los hiperparámetros del modelo

Una vez definido el modelo, es necesario determinar el valor del conjunto de hiperparámetros que serán empleados para el entrenamiento del mismo. De forma general, los controladores basados en Q-learning constan del siguiente conjunto de parámetros:

- Tasa de aprendizaje (α): Determina el peso de nuevas experiencias en la actualización de la tabla Q.
- Factor de descuento (γ): Define la importancia de recompensas futuras frente a las inmediatas.
- Tasa de exploración (ϵ): Define la probabilidad de que el agente tome una acción aleatoria con el objetivo de explorar nuevas partes del espacio de soluciones.
- Tasa de decaimiento (γ_e): Define como disminuye el valor de ϵ con el tiempo.
- Número de iteraciones (n): Define el numero total de iteraciones que serán empleadas durante el entrenamiento.

Parámetros de actualización de la tabla Q

Tras la obtención de los valores de una nueva experiencia, la correspondiente celda en la tabla Q se actualiza mediante la función:

$$Q_t \leftarrow Q_t + \alpha(r_t + \gamma Q_{max} - Q_t) \quad (3.12)$$

Donde α corresponde a la tasa de aprendizaje, γ al factor de descuento y Q_{max} es la recompensa futura esperada por encontrarse en un determinado estado.

De esta forma, el valor de α determina la importancia que tienen las nuevas experiencias en la actualización de la tabla Q en relación a valores anteriores de la tabla. Se debe tener en cuenta que un valor excesivo de α puede hacer que el sistema sea sensible al ruido. Con estas consideraciones, se escoge una tasa de aprendizaje de $\alpha = 0,01$.

Por otro lado, γ determina la importancia de la recompensa futura por encontrarse en el nuevo estado s_{t+1} .

Para la obtención del valor de γ se tienen en cuenta los resultados del análisis de correlación realizado en la sección 3.2, de modo que la importancia de la recompensa futura decaiga de forma similar a la correlación entre el valor de actuación θ y el error de potencia P_{error} . Como resultado de este análisis, se determina que el factor de descuento será $\gamma = 0,95$. La figura 3.10 muestra el decaimiento de una recompensa inicial de una unidad para distintos valores de γ .

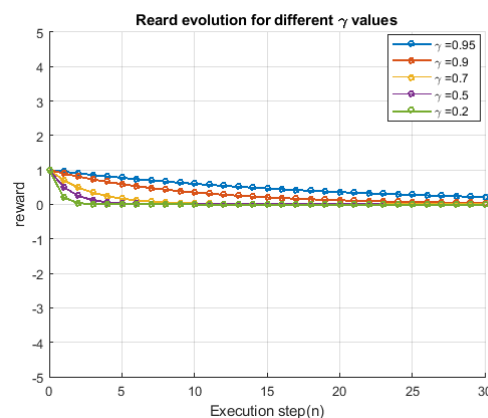


Figura 3.10: Decaimiento del valor de la recompensa futura para distintos valores de γ .

Dado que el objetivo del trabajo es realizar un estudio comparativo del tiempo de aprendizaje del modelo, el número de iteraciones (n) empleadas no es relevante en este caso.

Estrategia de exploración y explotación

La combinación del valor de ϵ y γ_e determina cómo el agente explorará el espacio de búsqueda a lo largo del entrenamiento, lo que es conocido como la estrategia de exploración y explotación.

Tal y como se explicó en la sección 2.3.3, la estrategia de aprendizaje empleada por los algoritmos de Q-learning toma acciones subóptimas con respecto a los valores de la tabla Q con una cierta probabilidad, con el objetivo de explorar nuevas partes del espacio de soluciones.

Si se define el factor de exploración ϵ como la probabilidad de que el agente tome una acción subóptima, la acción de control empleada será:

$$a_t = \begin{cases} \operatorname{argmax}_a Q(s_t, a) & \text{Con probabilidad } \epsilon - 1 \\ a \sim \mathcal{U}(0, 30) & \text{Con probabilidad } \epsilon \end{cases} \quad (3.13)$$

Donde $\mathcal{U}(0, 30)$ es la distribución uniforme de valores entre 0 y 30 grados.

De esta forma, un valor de ϵ elevado provocará que el agente explore nuevas estrategias de control en el espacio de soluciones de forma más eficaz a costa de un detrimento en la respuesta obtenida durante el entrenamiento. Esto puede llevar a que el sistema se vuelva inestable y no sea posible explorar los estados cercanos a la solución objetivo.

Por otro lado, se define la tasa de decaimiento de γ_e como el factor con el que se reducirá el valor de ϵ durante el proceso de aprendizaje. Así, ϵ se reduce tras cada paso de ejecución mediante la siguiente expresión:

$$\epsilon \leftarrow \epsilon \cdot \gamma_e \quad (3.14)$$

Esta reducción se realiza debido a que, conforme aumenta la confianza en el controlador, debido a un aumento en los datos de entrenamiento, la necesidad de explorar nuevas partes del espacio de soluciones se hace cada vez menos relevante.

El objetivo de este trabajo es la evaluación de los tiempos requeridos al emplear un método de entrenamiento con y sin aprendizaje federado. Por lo tanto, se escoge un valor de $\epsilon = 0,1$, y un valor de $\gamma_e = 1$ con el objetivo de que no exista decaimiento en la tasa de exploración.

3.5. Control multiagente mediante aprendizaje reforzado

Una vez definido el sistema de control de paso de cada turbina mediante aprendizaje reforzado, a continuación se detallará el esquema de aprendizaje federado. Este tiene como objetivo reducir el tiempo de entrenamiento y mejorar el rendimiento del controlador mediante el entrenamiento descentralizado de un modelo global a partir de los datos individuales de cuatro turbinas eólicas, a la vez que se conserva la privacidad de los datos.

3.5.1. Arquitectura del sistema federado

El paradigma de aprendizaje reforzado permite entrenar un modelo global de forma distribuida, sin la necesidad de compartir datos en el proceso. En su lugar, los parámetros de los modelos son transmitidos a un servidor central, el cual los agrega en un modelo global que es posteriormente distribuido a cada una de las turbinas. La figura 3.11 muestra la arquitectura de control multiagente del sistema.

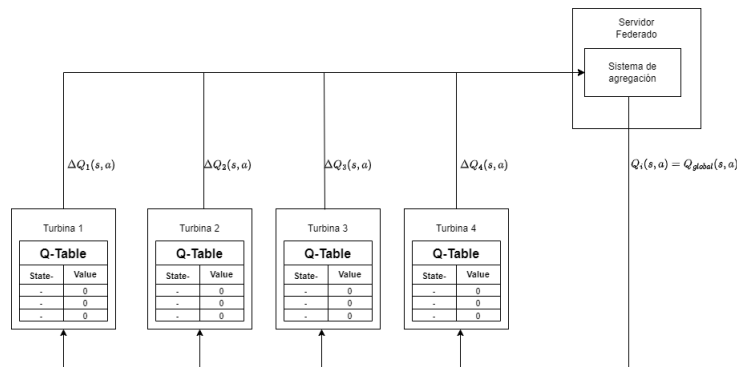


Figura 3.11: Arquitectura del sistema de control federado.

Cada aerogenerador representa un agente independiente controlado por el controlador de aprendizaje por refuerzo mediante Q-learning descrito en la sección 3.4.

Cuando un aerogenerador recibe una nueva señal de recompensa, este actualizará su tabla local $Q(s,a)$, almacenando también los cambios en una tabla temporal $\Delta Q(s,a)$.

Transcurrido un tiempo determinado, los sistemas individuales envían su tabla temporal $\Delta Q(s,a)$ al servidor federado, que agrega la información y actualiza el modelo global $Q_{global}(s,a)$. Por último, el modelo global se distribuye a todos los aerogeneradores individuales, los cuales sustituyen sus tablas locales $Q(s,a)$ por el modelo global $Q_{global}(s,a)$.

De esta forma, el sistema de aprendizaje federado queda definido por la función de agregación

y el protocolo de comunicación entre los agentes y el servidor federado.

3.5.2. Función de agregación

En la sección 2.4 se mostraron las principales funciones de agregación empleadas en aprendizaje federado para la agregación de los modelos de los agentes individuales.

Dado que en el sistema de interés existe homogeneidad en los datos de entrenamiento, y los tiempos de actualización del modelo global provocan que no exista una gran divergencia entre los modelos locales y el modelo global, una primera aproximación es emplear la función de agregación de FedAvg. Además, es la función de agregación más comúnmente empleada en aprendizaje federado.

De esta forma, la tabla Q del modelo global es actualizada a partir de la combinación lineal de las tablas Q locales de cada una de las turbinas.

$$Q_{global}(s, a) \leftarrow \frac{1}{N} \sum_{i=0}^N Q_i(s, a) \quad (3.15)$$

El principal problema de esta aproximación es que, debido a la información representada en la tabla Q , los datos de las tablas Q de los aerogeneradores no pueden ser agregados promediando los valores de cada aerogenerador. Esto implicaría que la recompensa futura esperada para un determinado estado dependería del promedio con turbinas que pueden no haber explorado todavía este estado.

Esto se debe a que, en una determinada iteración, la política óptima para un determinado estado puede haber sido explorada por una única turbina. En este caso, promediar los resultados de todos los agentes del sistema daría lugar a valores de recompensa incorrectos. Estos efectos pueden mitigarse inicializando los parámetros de las tablas Q a cero. Sin embargo, en esta aproximación, los estados explorados por múltiples turbinas serán siempre escogidos sobre los estados no explorados o explorados por pocos agentes.

Como resultado, el uso de técnicas tradicionales de agregación de aprendizaje por refuerzo como Federated Averaging [66] no produce resultados satisfactorios.

Para solucionar este problema es necesario modificar la función de agregación de modo que el valor de la tabla resultante corresponda a la acción con la mayor recompensa. Para ello se define una función de agregación en la que la recompensa esperada para un conjunto de estados y acciones corresponde a la suma de las recompensas estimadas por todos los

agentes para este determinado conjunto. Dado que, tal y como se mostró en la sección 3.4.2, la recompensa del sistema es negativa, esta función de agregación incentiva a los agentes a explorar el espacio de soluciones, mejorando así la convergencia del sistema.

$$Q_{global}(s, a) \leftarrow \sum_{i=0}^N \Delta Q_i(s, a) \quad (3.16)$$

Donde N es el número de turbinas del sistema y $\Delta Q_i(s, a)$ representa las actualizaciones locales del aerogenerador i -ésimo.

Este método de agregación añade los conocimientos individuales adquiridos por cada una de las turbinas a un modelo global $Q_{global}(s, a)$ únicamente si este valor de la tabla ha sido modificado en la última iteración, sin necesidad de compartir los datos operativos brutos del sistema. Durante el proceso de actualización, los únicos datos que se comparten con el aprendizaje federado son las actualizaciones de la tabla Q . Por lo tanto, se garantiza la privacidad de los datos confidenciales a la vez que se aprovecha el conocimiento colectivo de todos los sistemas.

3.5.3. Sincronización y comunicación

Otro aspecto importante en la definición de un sistema de aprendizaje federado es determinar el período de actualización del modelo global y la distribución del mismo a los agentes individuales. En general, es posible distinguir entre estrategias de sincronización síncronas y asíncronas.

En las estrategias síncronas, el servidor espera a que todos los agentes individuales hayan terminado su entrenamiento antes de agregar sus parámetros. Una vez que el entrenamiento finaliza, las actualizaciones de los agentes son enviadas al servidor, el cual las agrega y distribuye el modelo global.

Por otro lado, en las estrategias asíncronas, el servidor federado recibe las actualizaciones de los agentes tan pronto como el entrenamiento finaliza. Una vez recibidas, este actualiza el modelo global de forma incremental a partir de una tasa de aprendizaje α y distribuye el nuevo modelo a los agentes.

Por último, también es posible el empleo de estrategias híbridas, en las cuales el servidor federado actualiza el modelo global de forma incremental a partir de los datos obtenidos por subconjuntos de agentes individuales.

En la estrategia implementada, el modelo global es actualizado de forma incremental cada vez que se reciben los parámetros de un agente individual, enviados en intervalos de tiempo definidos. Sin embargo, la distribución del modelo global únicamente se realiza en intervalos de tiempo dados.

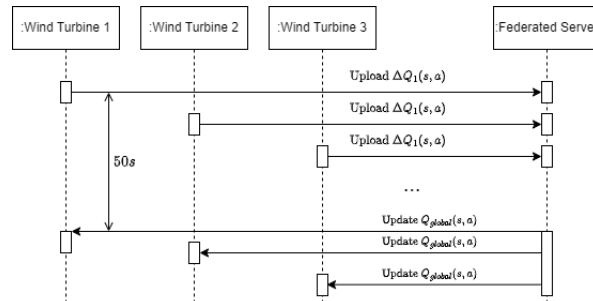


Figura 3.12: Diagrama secuencial de aprendizaje federado de las interacciones entre los aerogeneradores y el servidor federado.

La 3.12 muestra un diagrama de secuencia que representa la comunicación entre las turbinas y el servidor federado. En este, cada turbina envía sus actualizaciones de la tabla Q al servidor federado con una frecuencia de $f_{sync} = \frac{1}{t_s} Hz$. Por otro lado, el modelo global se distribuye a las turbinas con una frecuencia de $f_{update} = \frac{1}{t_u} Hz$. En la implementación se emplean valores de $t_s = 10s$ y $t_u = 50s$.

Capítulo 4

Resultados

En este capítulo se evalúa la mejora en los tiempos de entrenamiento al aplicar aprendizaje reforzado en una granja de 3 turbinas. Adicionalmente, se realiza un análisis de los tiempos de entrenamiento y error en el estacionario para los métodos de discretización de estados propuestos.

4.1. Descripción del entorno de simulación

La simulación del modelo se realiza empleando el entorno de modelado y simulación de Simulink, donde se emplea el modelo de la turbina descrito en la sección 2.5 con una potencia de referencia (P_{ref}) de $800KWh$ y una frecuencia de control de $1Hz$, y el modelo de viento detallado en la sección 2.6. Por otro lado, los resultados de la simulación son analizados empleando Matlab.

El modelo es evaluado al final de cada episodio. Durante este proceso de evaluación, la tabla Q de los agentes no se modifica y el agente toma las decisiones de control que maximizan la recompensa esperada independientemente del valor de la tasa de exploración (ϵ). Dado que la tasa de exploración puede alterar los resultados obtenidos de forma aleatoria, los resultados mostrados serán siempre los correspondientes a la evaluación al final del episodio.

4.2. Métricas de evaluación

Los requisitos de control enunciados en la sección 3.3 son evaluados mediante el conjunto de métricas descritas en esta sección.

En primer lugar, se emplea el error cuadrático medio (MSE, del inglés, mean squared error) para la evaluación de la diferencia entre la potencia de salida y la potencia de referencia.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.1)$$

MSE penaliza grandes errores de seguimiento, lo cual es especialmente útil para evaluar errores en el régimen transitorio. Sin embargo, es beneficioso el uso de otra métrica que permita conocer el error absoluto entre la salida de potencia obtenida y la potencia de referencia. Para ello se emplea el error medio absoluto (IAE, del inglés, integral absolute error), el cual permite calcular el error acumulado.

$$IAE = \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.2)$$

Por otro lado, uno de los principales problemas de las métricas descritas anteriormente es la penalización de valores que se desvían de la potencia de referencia sin importar el tiempo de ejecución. De esta forma, el error en régimen transitorio domina el resultado de la métrica sin tener en cuenta el error en tiempo estacionario.

Para solucionar este problema se emplea también la métrica de la integral del error absoluto ponderado por el tiempo (ITAE, del inglés, integral time absolute error), la cual pondera los errores en función del tiempo, penalizando en mayor medida los errores que ocurren en estado estacionario. La métrica ITAE es empleada para penalizar errores persistentes en el tiempo.

$$ITAE = \sum_{i=1}^n t_i \cdot |y_i - \hat{y}_i| \quad (4.3)$$

Además, se utiliza una métrica para evaluar el uso de acciones de control agresivas por parte del controlador. Para ello, se emplea la variación total (TV, del inglés, total variation) de la señal de control, la cual penaliza cambios bruscos en la señal de control.

$$TV = \sum_{i=1}^N |u_i - u_{i-1}| \quad (4.4)$$

Las métricas mostradas permiten analizar tanto los errores en la potencia de salida como el uso excesivo de actuación.

4.3. Evaluación de las técnicas de discretización de estados

Los métodos de discretización de la variable P_{error} descritos en la sección 3.4.1 son simulados, comparando su tiempo de convergencia y error en estado estacionario. El método con los mejores resultados es empleado en la evaluación del sistema multiagente mediante aprendizaje por refuerzo presentado en la sección 4.4.

Se realizan 4 ejecuciones, en las que cada ejecución consiste en 20 episodios. Cada episodio incluye 1000 segundos de simulación, evaluando los controladores al final de cada episodio durante otros 1000 segundos. Debido a la naturaleza probabilística del algoritmo de Q-Learning, se obtienen resultados distintos en cada una de las ejecuciones.

Todas las simulaciones se realizan empleando el mismo perfil de viento, generado de forma aleatoria.

Los métodos de discretización propuestos son implementados con un valor de $n = 30$, dando como resultado un tamaño de discretización de $2n + 1 = 61$ estados.

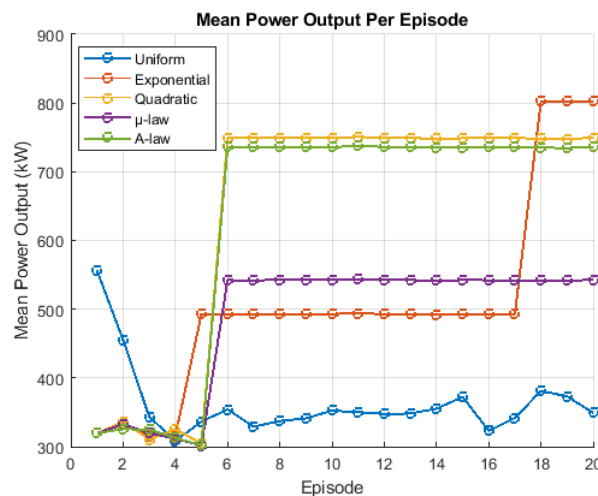


Figura 4.1: Potencia media por episodio de cada método de discretización durante la primera ejecución.

La figura 4.1 muestra los resultados del entrenamiento durante la primera ejecución. Como se puede observar, el método de discretización exponencial converge a la salida deseada después de 18 episodios, mientras que el resto de los métodos de discretización no son capaces de converger a la potencia de referencia.

Todos los métodos de discretización obtienen resultados significativamente mejores que la

discretización uniforme.

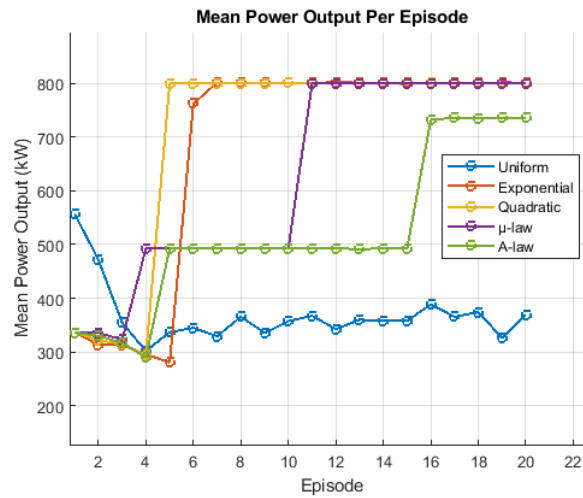


Figura 4.2: Potencia media de salida por episodio de cada método de discretización durante la segunda ejecución.

La figura 4.2 muestra los resultados del entrenamiento de la segunda ejecución. La convergencia a la potencia de salida deseada es alcanzada en primer lugar por el controlador utilizando la discretización cuadrática después de 5 episodios de entrenamiento, mientras que la discretización exponencial converge tras 7 episodios de entrenamiento.

El método de discretización basado en la ley μ converge tras 11 episodios de entrenamiento, mientras que la ley A no es capaz de converger en los 20 episodios analizados.

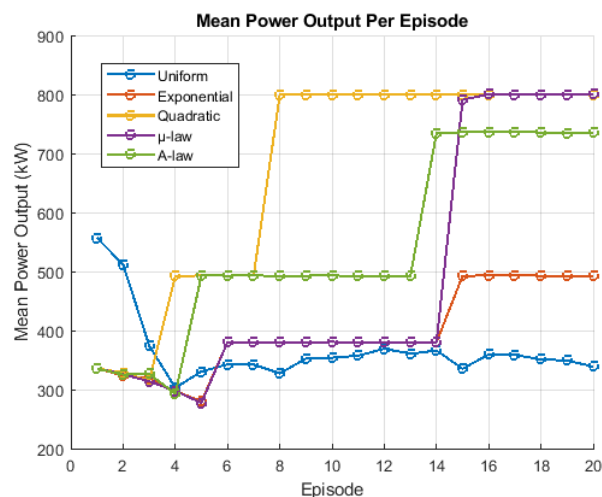


Figura 4.3: Potencia media de salida por episodios de cada método de discretización durante la tercera ejecución.

La figura 4.3 muestra los resultados del entrenamiento durante la tercera ejecución. El método

de discretización cuadrática logra el mejor tiempo de convergencia en 8 episodios, seguido por el método basado en la ley μ , el cual converge tras 16 iteraciones de entrenamiento.

Ni los métodos de discretización exponencial ni el basado en la ley A convergen en los 20 episodios de entrenamiento, aunque logran resultados significativamente mejores que el método de discretización uniforme.

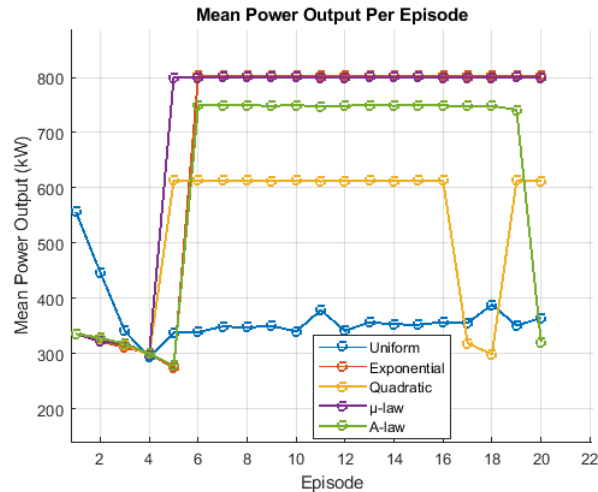


Figura 4.4: Potencia media de salida por episodio de cada método de discretización durante la cuarta ejecución.

La figura 4.4 muestra los resultados del entrenamiento durante la última ejecución. Donde la discretización basada en la ley μ converge a la potencia de salida deseada después de 5 episodios, seguida de la discretización exponencial, la cual converge tras 6 episodios de entrenamiento. Los otros métodos de discretización logran converger.

Todas las ejecuciones realizadas muestran cómo los controladores que utilizan métodos de discretización avanzados logran resultados significativamente mejores que el controlador con discretización uniforme. Sin embargo, no existen diferencias significativas en el tiempo de convergencia entre los métodos propuestos. Esto se explica porque todos los métodos de discretización tienen el mismo número de estados.

Por otra parte, todos los métodos, excepto la discretización uniforme, son capaces de converger a la potencia de salida deseada después de suficientes episodios de entrenamiento, como se observa en la figura 4.5.

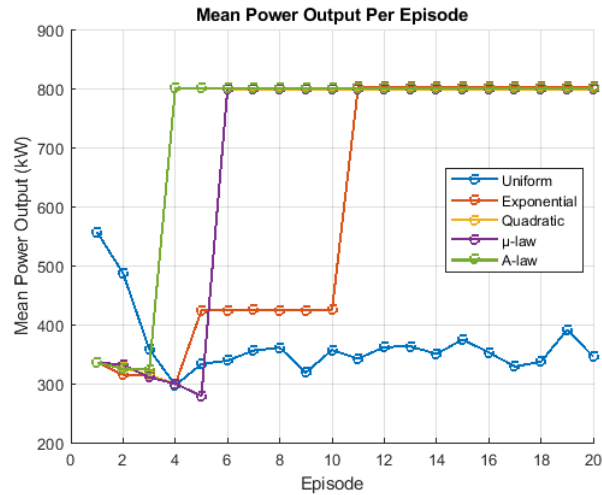


Figura 4.5: Potencia media por episodio durante 20 episodios de entrenamiento.

A continuación, se estudia la respuesta temporal de los distintos métodos una vez finalizado el entrenamiento correspondiente a la ejecución mostrada en la figura 4.5, siendo esta la ejecución en la que se obtienen los mejores resultados.

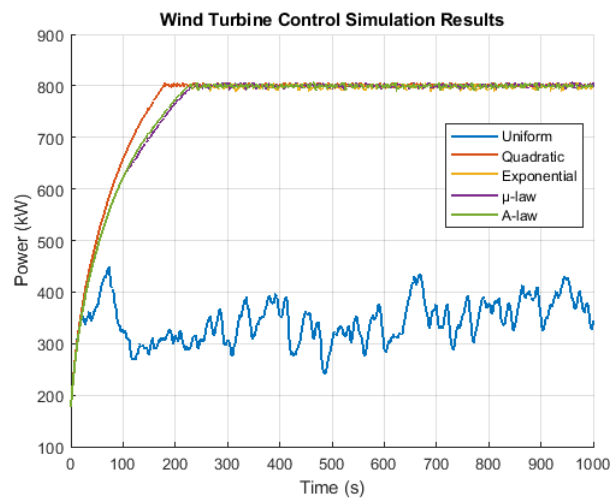


Figura 4.6: Respuesta temporal de los métodos de discretización tras 40 episodios de entrenamiento.

La figura 4.6 muestra la respuesta temporal del sistema tras 40 episodios de entrenamiento. Todos los métodos de discretización presentan una respuesta casi idéntica, a excepción de la distribución uniforme, que no logra converger a la potencia deseada.

Entre las discretizaciones comparadas, el método cuadrático obtiene una respuesta temporal ligeramente mejor que el resto de los métodos.

Para evaluar el error de los métodos de discretización propuestos, se analiza con más detalle

la respuesta en estado estacionario.

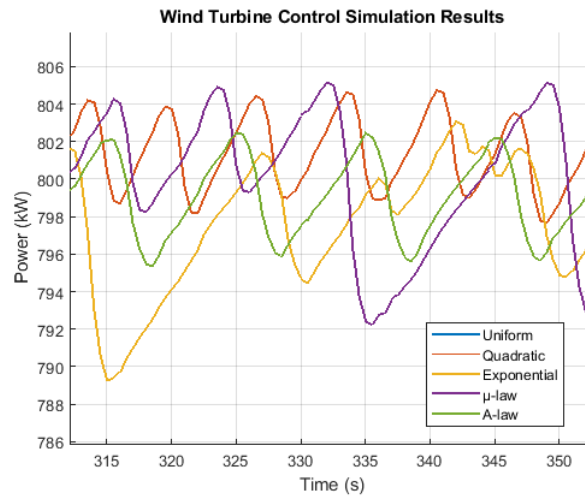


Figura 4.7: Respuesta temporal en estado estacionario del método de discretización tras 40 episodios de entrenamiento.

En la figura 4.7 se compara la potencia de salida en el estado estacionario de los distintos métodos de discretización una vez que el sistema ha alcanzado el estado estacionario. Como se observa, la discretización basada en la ley μ obtiene un error en estado estacionario de aproximadamente $8kW$, mientras que el método exponencial presenta un error en estado estacionario de $11kW$.

Por otro lado, los métodos de discretización cuadrático y de ley A presentan resultados significativamente mejores, obteniendo este último una estabilización de potencia más consistente alrededor del punto de equilibrio. De esta forma, es la discretización mediante la ley A la empleada en el control de paso del sistema multiagente mediante aprendizaje federado.

Los resultados obtenidos muestran que todos los métodos de discretización propuestos obtienen tiempos de convergencia comparables.

Además, en este caso, el retardo de 1 segundo en la actuación provoca que el error en estado estacionario del sistema sea diferente del límite teórico de la discretización, determinado por el menor valor posible de P_{error} . Como consecuencia, los métodos con una discretización más densa alrededor del punto de equilibrio obtienen mejores errores en estado estacionario.

4.4. Evaluación de los resultados de aprendizaje federado

El objetivo de la investigación llevada a cabo en este trabajo es evaluar el impacto de la aplicación de técnicas de aprendizaje federado en el tiempo de aprendizaje de sistemas de control basados en Q-learning, cuando estos son aplicados a la estabilización de la potencia de salida en turbinas eólicas en la región de viento nominal. En esta sección se realiza el análisis de los resultados obtenidos.

A diferencia del perfil de viento empleado en la sección 4.3, la evaluación del modelo de aprendizaje federado se realiza empleando un perfil de viento distinto para cada episodio.

Como se describió en la sección 3.5, el sistema de aprendizaje federado simulado se compone de tres turbinas que entrenan de forma distribuida un modelo común. Por otro lado, estos resultados son comparados con la evolución de una turbina aislada con el mismo controlador basado en Q-Learning, pero sin implementar la estrategia de aprendizaje federado.

Se realiza el entrenamiento de los modelos durante 130 episodios, donde cada uno de los episodios consiste en 800 segundos de simulación.

4.4.1. Evaluación del error de potencia

En primer lugar, se muestra la comparación del error cuadrático medio (MSE) obtenido por la turbina sin emplear aprendizaje federado con la media del MSE de las turbinas que forman parte del sistema federado para cada una de las simulaciones de evaluación realizadas durante el entrenamiento.

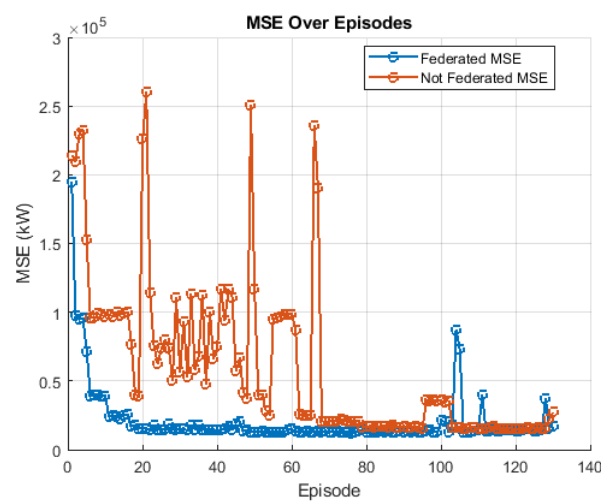


Figura 4.8: Comparación del error cuadrático medio entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.

La figura 4.8 muestra cómo el control basado en el aprendizaje federado es capaz de aprender una estrategia de control que minimiza el error cuadrático entre la potencia de salida de la turbina (P_{out}) y la potencia de referencia (P_{ref}) en un número significativamente menor de episodios en comparación con el enfoque no federado.

Del mismo modo, es posible observar cómo los sistemas empleando aprendizaje federado muestran una menor variación en el error de potencia durante el entrenamiento, mientras que el sistema no federado fluctúa durante los primeros 70 episodios del mismo.

Las fluctuaciones en los valores del MSE entre episodios después de que el algoritmo haya convergido pueden explicarse debido a la presencia de condiciones de viento nuevas para el controlador. Es decir, si el sistema experimenta un nuevo valor de velocidad de viento (V_{viento}), este corresponderá a una celda en la tabla Q que no ha sido explorada con anterioridad, haciendo necesario que el agente encuentre la política óptima.

El error cuadrático medio penaliza de forma severa grandes errores de seguimiento. De esta forma, los resultados obtenidos pueden verse afectados por la presencia de alguna inestabilidad del controlador. Del mismo modo, esto significa que la métrica queda dominada por la respuesta del sistema en el régimen transitorio, haciendo difícil evaluar el rendimiento del controlador en régimen estacionario. Para ello se usan métricas que muestran la magnitud total del error, como es el caso del error absoluto integral (IAE).

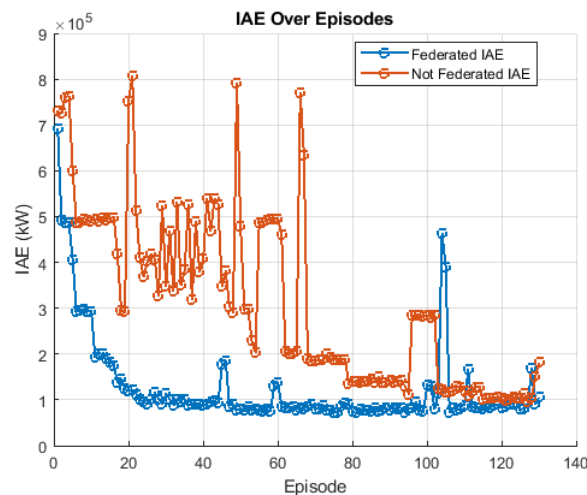


Figura 4.9: Comparación del error absoluto integral entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.

En la figura 4.9 se puede observar cómo las turbinas del sistema federado son capaces de reducir el error integral absoluto en un número significativamente menor de episodios que el sistema no federado. En este caso, los sistemas de control entrenados mediante aprendizaje federado son capaces de encontrar la estrategia de control que minimiza la diferencia con

la potencia objetivo tras 22 episodios. Sin embargo, el sistema no federado requiere 115 episodios para converger a la misma solución.

Uno de los principales inconvenientes de las métricas anteriores es la influencia del régimen transitorio en el error. Dado que el entrenamiento y evaluación del sistema se realizan durante 800 segundos, los valores en el tiempo de asentamiento tienen demasiada relevancia en el valor del error, dificultando la evaluación de errores en el estado estacionario.

Se emplea entonces el Error Absoluto Ponderado por el Tiempo (ITAE) para obtener una mayor información sobre los errores estacionarios en la respuesta.

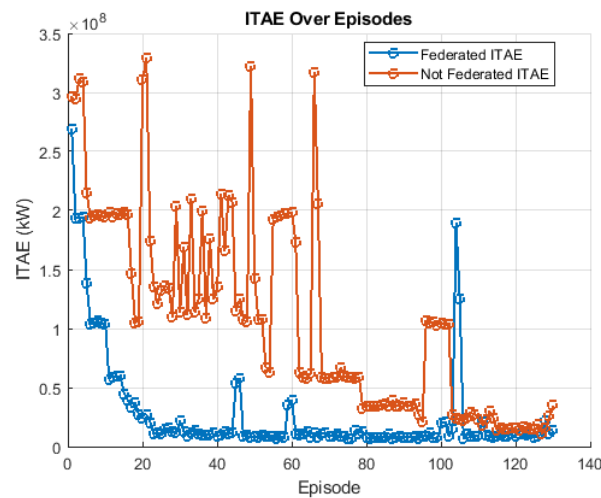


Figura 4.10: Comparación del Error Absoluto Ponderado por el Tiempo (ITAE) entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.

La figura 4.10 muestra la comparación entre los sistemas federados y no federados de la métrica ITAE para cada episodio del entrenamiento. Al igual que en los casos anteriores, la métrica muestra cómo los sistemas empleando técnicas de aprendizaje federado son capaces de converger a la estrategia de control óptima en muchos menos episodios que el sistema no federado.

4.4.2. Comparación de las respuestas temporales de los sistemas federados y no federados

Una vez analizadas las métricas del error de potencia, a continuación se muestra un conjunto de gráficas con el objetivo de ilustrar la potencia de salida de las turbinas durante el proceso de entrenamiento.

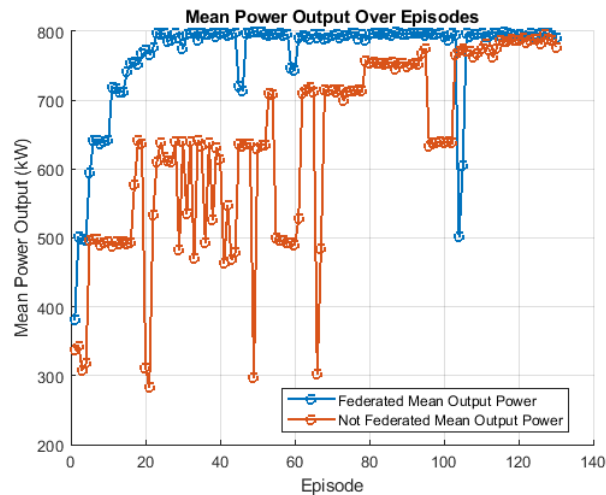


Figura 4.11: Comparación de la potencia de salida media entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.

La figura 4.11 muestra la comparación de la potencia de salida media de las turbinas federadas y no federadas en cada episodio del entrenamiento. En esta se puede observar cómo la potencia media de las turbinas que forman parte del sistema federado alcanza la potencia objetivo de 800KW en un número de episodios significativamente menor que el enfoque no federado. En particular, las turbinas empleando aprendizaje federado son capaces de estabilizar la salida de potencia en torno a la potencia nominal tras 22 episodios. Mientras tanto, el sistema no federado necesita 115 episodios para lograr resultados similares de forma constante.

Del mismo modo, es posible observar cómo la potencia media de salida del sistema no federado varía significativamente durante el entrenamiento.

Por otro lado, también es posible observar la evolución de la potencia de salida de las turbinas en distintos episodios del entrenamiento.

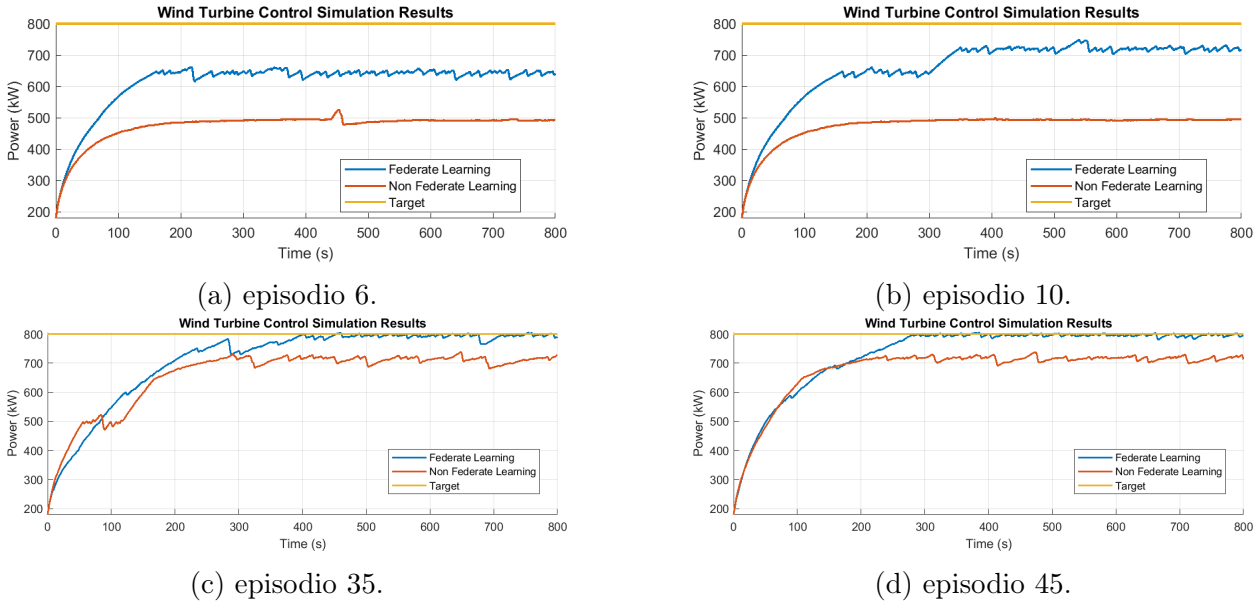


Figura 4.12: Comparación de la potencia de salida de las turbina individual con la potencia de salida media de las turbinas del sistema federado.

Como se puede observar en la figura 4.12, la potencia de salida media de los sistemas federados se encuentra más cercana a la potencia de referencia que el sistema no federado en todos los episodios mostrados.

Por otro lado, cabe destacar que el valor en el estacionario que se muestra en las figuras coincide con la discretización de la variable de estado P_{error} descrita en la sección 3.4.1. Para un determinado conjunto de viento y error de potencia, el controlador debe aprender la acción de control óptima que permita reducir el error de potencia, y por tanto avanzar al siguiente nivel de potencia. Mientras esta acción no haya sido encontrada, la potencia del controlador permanece en valores cercanos a la discretización de la variable.

4.4.3. Evaluación de la señal de control

Una vez se ha demostrado que el empleo de estrategias de aprendizaje federado permite reducir el tiempo de aprendizaje requerido para aprender la política de control que minimiza el error de la potencia de salida, a continuación se evalúan los efectos de estas estrategias en la disminución de la acción de control necesaria para controlar el sistema.

Para ello, se emplea como métrica de variación temporal(TV) de la señal de control.

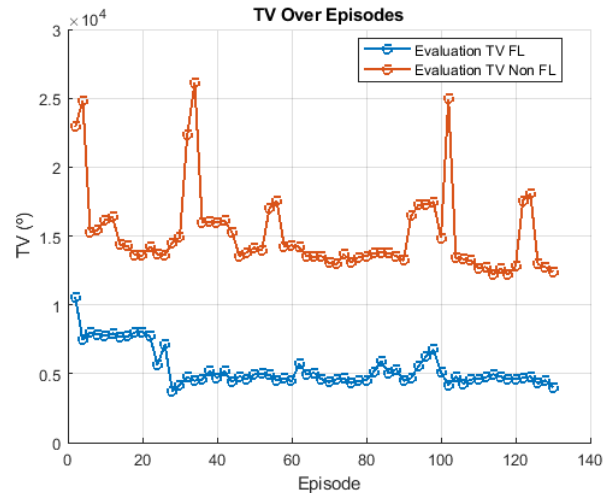


Figura 4.13: Comparación de la variación total de la señal de control entre sistemas federados (línea azul) y no federados (línea roja) a lo largo de 130 episodios.

La figura 4.13 muestra cómo los sistemas que emplean aprendizaje federado son capaces de reducir significativamente las variaciones de la señal de actuación con respecto al sistema no federado en la mayoría de los episodios del entrenamiento.

Por otro lado, cabe destacar que la variación en la actuación aumenta ligeramente en los últimos episodios del entrenamiento. Esto puede deberse a los pesos empleados en la función de recompensa, la cual prioriza la reducción del error de la salida sobre la minimización de la señal de actuación.

4.4.4. Análisis del comportamiento de las turbinas entrenadas mediante aprendizaje federado

En las secciones anteriores se ha realizado una comparación de los resultados obtenidos por turbinas entrenadas mediante técnicas de aprendizaje federado, y una turbina sin aprendizaje federado. Sin embargo, con el objetivo de comprender mejor la evolución de los distintos agentes del sistema de aprendizaje federado, en esta sección se analizará el comportamiento de las turbinas del sistema federado durante el entrenamiento.

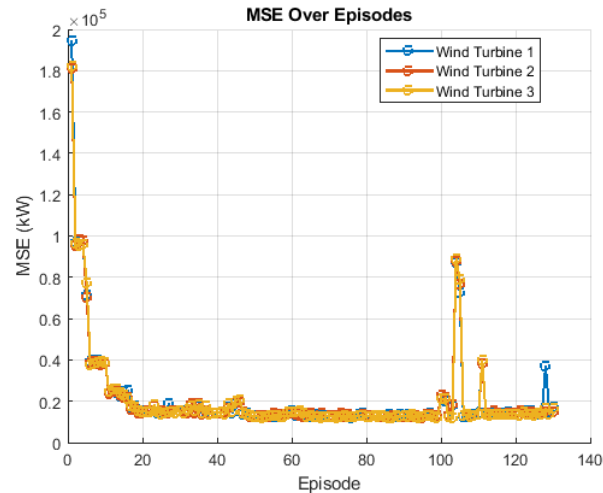


Figura 4.14: Evolución del valor del Error Cuadrático Medio para cada turbina del sistema de Aprendizaje Federado durante el entrenamiento.

Una de las características principales de los sistemas de aprendizaje federado es el entrenamiento de forma distribuida de un modelo global, el cual se distribuye al conjunto de agentes del sistema para que actualicen su modelo.

De esta forma, en cualquier instante de la simulación, las tablas Q de las turbinas que forman parte del sistema federado no diferirán significativamente entre sí. Esto se puede observar claramente en la figura 4.14, donde se muestra cómo el Error Cuadrático Medio de todas las turbinas del sistema federado es similar para cada episodio del entrenamiento.

Esta propiedad se observa de forma más clara analizando la respuesta temporal de las turbinas del sistema federado durante el proceso de entrenamiento.

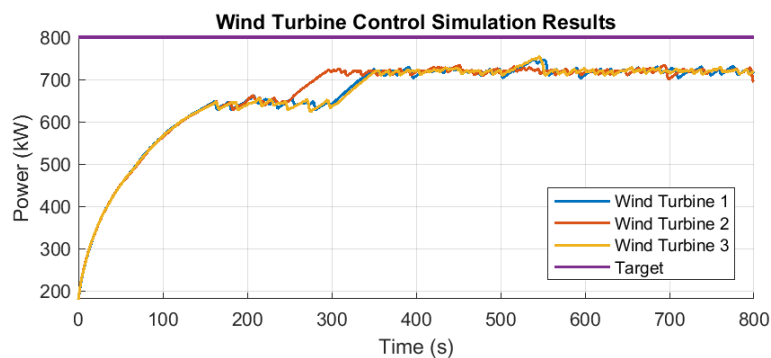


Figura 4.15: Potencia de las turbinas del sistema de aprendizaje federado en la episodio número 10 del entrenamiento.

La figura 4.15 muestra la respuesta temporal de las turbinas del sistema federado durante la episodio número 10 del entrenamiento.

Como se puede observar, en torno al segundo 150 de la simulación, la turbina 2 encuentra una nueva estrategia de control que mejora el error de la potencia. A continuación, los parámetros del modelo son enviados al servidor federado, el cual los agrega al modelo global.

Una vez el modelo global es distribuido al resto de las turbinas en torno al segundo 300, las turbinas 1 y 3 agregan la política óptima encontrada por la turbina 2 a su tabla Q , haciendo que sean capaces de mejorar la potencia de salida.

4.4.5. Análisis de la respuesta del sistema tras la fase de entrenamiento

Si bien el objetivo de este trabajo es demostrar cómo estrategias de aprendizaje federado pueden ser empleadas para reducir los tiempos de aprendizaje de sistemas de control basados en Q-learning, es interesante analizar la respuesta temporal obtenida por las turbinas entrenadas mediante aprendizaje federado tras finalizar la fase de entrenamiento.

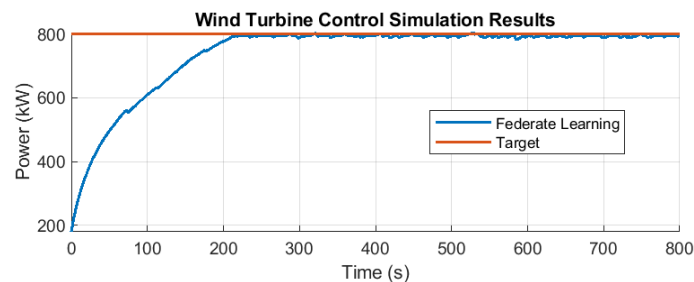


Figura 4.16: Respuesta temporal de los sistemas de control entrenados mediante aprendizaje federado una vez finalizada la fase de entrenamiento.

Los resultados mostrados en la figura 4.16 corresponden a la respuesta temporal de una de las turbinas entrenadas empleando aprendizaje federado una vez finalizado el entrenamiento. Se trata de un sistema subamortiguado o con amortiguamiento crítico, con un tiempo de asentamiento de 200 segundos.

Por otro lado, el error de potencia varía un máximo de 30KW en estado estacionario, lo cual puede ser explicado por la presencia de nuevas condiciones de viento que perturben el sistema.

Por último, cabe destacar el comportamiento ligeramente oscilatorio de la respuesta en el estacionario, lo cual puede producir vibraciones en el rotor de la turbina. Esto queda explicado debido a que la función de recompensa del sistema no ha sido diseñada para la minimización de vibraciones, del mismo modo que la discretización de los estados del sistema produce que la acción de control oscile en torno al punto de equilibrio.

En la figura 4.17 se comparan las respuestas temporales de los sistemas federado y no federado tras la fase de entrenamiento, obteniendo resultados muy similares para ambos sistemas.

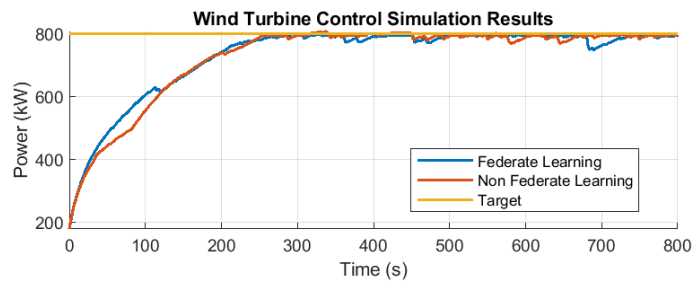


Figura 4.17: Respuesta temporal obtenida tras la simulación una vez ha finalizado el entrenamiento.

Los resultados obtenidos tras la simulación de los sistemas demuestran la eficacia del aprendizaje federado para mejorar el rendimiento y reducir los tiempos de entrenamiento en controladores basados en Q-Learning, cuando estos son aplicados a la estabilización de potencia en turbinas eólicas, mostrando su potencial aplicación en este campo.

Capítulo 5

Conclusiones y trabajos futuros

En este capítulo se revisan las conclusiones obtenidas como resultado del desarrollo del controlador de paso y de la implementación del método de aprendizaje federado. Finalmente, se presentan posibles líneas de trabajo futuras.

5.1. Conclusiones y objetivos alcanzados

Este trabajo ha demostrado cómo el empleo de técnicas de aprendizaje federado (FL) permite reducir los tiempos de entrenamiento de controladores basados en Q-Learning aplicados a la estabilización de la potencia de salida de turbinas eólicas.

Para ello, se ha diseñado y sintonizado un sistema de control basado en Q-Learning con el objetivo de realizar la estabilización a la potencia nominal de una turbina aislada.

Se ha demostrado cómo el empleo de técnicas de discretización del espacio de estados permite reducir considerablemente el tiempo de entrenamiento de controladores basados en Q-Learning. Se han propuesto y analizado cuatro métodos de discretización basados en funciones exponenciales, cuadráticas y métodos de compansión como la ley μ y la ley A, demostrando cómo las técnicas propuestas obtienen resultados significativamente mejores a la distribución uniforme. Los resultados de la simulación han mostrado que la discretización cuadrática y la basada en la ley A obtienen el menor error en estado estacionario, mientras que todos los métodos propuestos obtienen una respuesta transitoria similar.

Además, se han estudiado distintas funciones de recompensa para minimizar la acción de control, reduciendo así la fatiga en la turbina a la vez que se minimiza el error de potencia. Los resultados obtenidos muestran cómo los controladores son capaces de aprender políticas

que cumplen con ambos objetivos de control.

Posteriormente, este controlador ha sido extendido para su utilización en una granja de tres turbinas, empleando aprendizaje federado para entrenar un modelo común sin compartir datos sensibles en el proceso.

La comparación del sistema federado y no federado ha mostrado cómo el enfoque de aprendizaje federado permite aprender una estrategia de control óptima en menos iteraciones que el enfoque no federado. En concreto, la potencia media de salida del enfoque de aprendizaje federado converge a la potencia de salida de referencia tras 22 episodios de entrenamiento, en comparación con los 115 episodios requeridos por el enfoque no federado.

5.2. Aplicaciones

Una posible aplicación de esta tecnología es el empleo de los datos distribuidos de todos los aerogeneradores disponibles del mismo modelo para entrenar un sistema de control óptimo basado en datos, el cual aproveche la información de todos los sistemas disponibles, a la vez que se mantiene la seguridad y privacidad de los datos de entrenamiento.

El enfoque de aprendizaje por refuerzo, combinado con técnicas de discretización de estados, tiene el potencial de reducir los tiempos de entrenamiento de los controladores Q-Learning utilizados para la estabilización de potencia de los aerogeneradores, facilitando la adopción de este tipo de técnicas en el control de turbinas eólicas.

Además, la mejora en los tiempos de entrenamiento, junto con el uso de funciones de recompensa que minimicen las cargas en el rotor, permitiría la reducción de la fatiga de las turbinas, reduciendo así los costos de mantenimiento y prolongando la vida útil de las mismas.

5.3. Trabajos futuros

Como trabajos futuros, se propone realizar un análisis de robustez de los controladores obtenidos, así como el uso de técnicas de aprendizaje por refuerzo con espacios de acciones y estados continuos, como Deep Deterministic Policy Gradient (DDPG).

Además, factores externos a la turbina, como las condiciones atmosféricas o los cambios de dinámica debidos a problemas de mantenimiento, pueden causar discrepancias en el comportamiento dinámico de los aerogeneradores del mismo modelo. Para abordar este problema, se propone emplear Aprendizaje Federado Agrupado (CFL, del inglés, Clustered Federated

Learning), con el objetivo de agrupar automáticamente los aerogeneradores en grupos con dinámicas similares, entrenando modelos independientes para cada uno de estos grupos.

Se propone la extensión de los modelos descritos en este trabajo, empleando una función de Lyapunov para imponer restricciones adicionales en la evaluación de la acción del agente con el objetivo de garantizar la estabilidad del sistema [21].

Por último, se propone realizar una comparación de las técnicas de discretización propuestas en este trabajo con técnicas de discretización del espacio de estados adaptativas, las cuales son capaces de encontrar la partición óptima del conjunto de estados durante el proceso de entrenamiento [61].

Bibliografía

- [1] *A Review of Radial Basis Function (RBF) Neural Networks*, pages 1–19. doi: 10.1142/9789812812506_0001. URL https://www.worldscientific.com/doi/abs/10.1142/9789812812506_0001.
- [2] How a wind turbine Works - text version. URL <https://www.energy.gov/eere/wind/how-wind-turbine-works-text-version>.
- [3] wblinv. URL <https://nl.mathworks.com/help/stats/wblinv.html>.
- [4] Deep Deterministic Policy Gradient — Spinning Up documentation. URL <https://spinningup.openai.com/en/latest/algorithms/ddpg.html>.
- [5] Abdel Ghani Aissaoui, Ahmed Tahour, Mohamed Abid, Najib Essounbouli, and Frederic Nollet. Power control of wind turbine based on fuzzy controllers. *Energy Procedia*, 42:163–172, 2013. ISSN 1876-6102. doi: <https://doi.org/10.1016/j.egypro.2013.11.016>. URL <https://www.sciencedirect.com/science/article/pii/S1876610213017189>. Mediterranean Green Energy Forum 2013: Proceedings of an International Conference MGEF-13.
- [6] Amal Alshardan, Sidra Tariq, Rab Bashir, Oumaima Saidani, and Rashid Jahangir. Federated learning (fl) model of wind power prediction. *IEEE Access*, PP:1–1, 01 2024. doi: 10.1109/ACCESS.2024.3415781.
- [7] Roohollah Amiri, Hani Mehrpouyan, Lex Fridman, Ranjan Mallik, Arumugam Nallanathan, and David Matolak. A machine learning approach for power allocation in hetnets considering QoS. *arXiv preprint*, 2018. doi: 10.48550/arXiv.1803.06760. URL <https://arxiv.org/abs/1803.06760>.
- [8] O. Apata and D.T.O. Oyedokun. An overview of control techniques for wind turbine systems. *Scientific African*, 10:e00566, 2020. ISSN 2468-2276. doi: <https://doi.org/10.1016/j.sciaf.2020.e00566>. URL <https://www.sciencedirect.com/science/article/pii/S2468227620303045>.

- [9] Lunde Ardhenta and Ramadhani Subroto. Application of direct mrac in pi controller for dc-dc boost converter. *International Journal of Power Electronics and Drive Systems (IJPEDS)*, 11:851, 06 2020. doi: 10.11591/ijpedsv11.i2.pp851-858.
- [10] Qumrishi Arooj. FedWindT: Federated learning assisted transformer architecture for collaborative and secure wind power forecasting in diverse conditions. *Energy*, 309 (C), 2024. doi: 10.1016/j.energy.2024.133072. URL <https://ideas.repec.org/a/eee/energy/v309y2024ics0360544224028470.html>.
- [11] Aamer Asghar, Raza Ehsan, Khazina Naveed, Essam Al-Ammar, Krzysztof Ejsmont, and Mirosław Nejman. Recurrent neural network for pitch control of variable-speed wind turbine. *Science Progress*, 107:1–22, 04 2024. doi: 10.1177/00368504241243160.
- [12] Ramesh Babu.N and Arulmozhivarman Pachiyappan. Wind energy conversion systems-a technical review. *Journal of Engineering Science and Technology(SNIP: 0.477)*, Volume 8:493–507, 08 2013.
- [13] M. Bahgat, M. Ezzat, M. A. Attia, et al. Comparative analysis of PI and fuzzy logic controller for grid connected wind turbine under normal and fault conditions. *Sci Rep*, 15:1954, 2025. doi: 10.1038/s41598-024-85073-w. URL <https://doi.org/10.1038/s41598-024-85073-w>.
- [14] Charles Beauville. FL Starter Pack: FedProx on MNIST using a CNN, 2 2023. URL <https://flower.ai/blog/2023-02-16-fl-starter-pack-fedprox-mnist-cnn/>.
- [15] Billel Bouchemal and Mohamed Chemachema. Model-free control for variable-speed wind energy conversion systems based on power tracking. *Electrical Engineering*, 107: 179–189, 06 2024. doi: 10.1007/s00202-024-02514-5.
- [16] Steven L. Brunton. Overview of methods. Cassyni, 2022/01/03, 2022. URL <https://doi.org/10.52843/cassyni.jcgdvc>. Accessed: 2025-02-22.
- [17] Steven L. Brunton. Q-learning: Model free reinforcement learning and temporal difference learning. Cassyni, 2022/01/14, 2022. URL <https://doi.org/10.52843/cassyni.ss11hp>. Accessed: 2025-02-22.
- [18] Gabriel Cassimiro. A Deep Dive into Actor-Critic methods with the DDPG Algorithm. 4 2023. URL <https://medium.com/geekculture/a-deep-dive-into-the-ddpg-algorithm-for-continuous-control-2718222c333e>.
- [19] Ernesto Chavero-Navarrete, Mario Perea, Juan Jauregui, R.V. Serrano, and G.J. Moreno. Expert control systems implemented in a pitch control of wind turbine: A review. *IEEE Access*, PP:1–1, 01 2019. doi: 10.1109/ACCESS.2019.2892728.

- [20] Karim Cherifi. An overview on recent machine learning techniques for port hamiltonian systems. *Physica D: Nonlinear Phenomena*, 411:132620, 2020. ISSN 0167-2789. doi: <https://doi.org/10.1016/j.physd.2020.132620>. URL <https://www.sciencedirect.com/science/article/pii/S0167278919307705>.
- [21] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning, 2018. URL <https://arxiv.org/abs/1805.07708>.
- [22] Lucas Cinelli, Gabriel Chaves, and Markus Lima. Vessel classification through convolutional neural networks using passive sonar spectrogram images. 05 2018. doi: 10.14209/sbrt.2018.340.
- [23] Paulo Coelho. The betz limit and the corresponding thermodynamic limit. *Wind Engineering*, 47(2):491–496, 2023. doi: 10.1177/0309524X221130109. URL <https://doi.org/10.1177/0309524X221130109>.
- [24] Olivier Delalleau, Maxim Peter, Eloi Alonso, and Adrien Logut. Discrete and continuous action representation for practical RL in video games. *CoRR*, abs/1912.11077, 2019. URL <http://arxiv.org/abs/1912.11077>.
- [25] Quang Vinh Do and Insoo Koo. Actor-critic deep learning for efficient user association and bandwidth allocation in dense mobile networks with green base stations. *Wireless Networks*, 25(8):5057–5068, nov 2019. ISSN 1572-8196. doi: 10.1007/s11276-019-02117-0. URL <https://doi.org/10.1007/s11276-019-02117-0>.
- [26] H. Dong, J. Xie, and X. Zhao. Wind farm control technologies: From classical control to reinforcement learning. *Progress in Energy*, 4(3):032006, 2022. doi: 10.1088/2516-1083/ac7a1c.
- [27] John Doyle, Keith Glover, P. Khargonekar, and Bruce Francis. State space solution to standard h2 and h control problem. *Automatic Control, IEEE Transactions on*, 34:831 – 847, 09 1989. doi: 10.1109/9.29425.
- [28] Gaston Frias, Maximiliano Asensio, and Guillermo Catuogno. Implementación de un seguidor del punto de máxima potencia en un aerogenerador con dspic. 06 2018.
- [29] Susan Frost, Mark Balas, and Alan Wright. Direct adaptive control of a utility-scale wind turbine for speed regulation. *International Journal of Robust and Nonlinear Control*, 19:59 – 71, 01 2009. doi: 10.1002/rnc.1329.
- [30] M. Geyler and P. Caselitz. Robust multivariable pitch control design for load reduction on large wind turbines. *Journal of Solar Energy Engineering*, 130(3):031014, 07 2008. ISSN 0199-6231. doi: 10.1115/1.2931510. URL <https://doi.org/10.1115/1.2931510>.

- [31] Albin Grataloup, Stefan Jonas, and Angela Meyer. A review of federated learning in renewable energy applications: Potential, challenges, and future directions. *Energy and AI*, 17:100375, 2024. ISSN 2666-5468. doi: <https://doi.org/10.1016/j.egyai.2024.100375>. URL <https://www.sciencedirect.com/science/article/pii/S2666546824000417>.
- [32] Bir GS. Multiblade coordinate transformation and its application to wind turbine analysis. 01 2008. doi: 10.2514/6.2008-1300.
- [33] Y. Guo, Zhibin Zhang, Weiqing Zheng, and Du Yanhua. An autonomous path planning model for unmanned ships based on deep reinforcement learning. *Sensors*, 20:426, 01 2020. doi: 10.3390/s20020426.
- [34] Q. Hassan, P. Viktor, T. J. Al-Musawi, B. M. Ali, S. Algburi, H. M. Alzoubi, others, and M. Jaszczur. The renewable energy role in the global energy transformations. *Renewable Energy Focus*, 48:100545, 2024. doi: 10.1016/j.ref.2024.100545.
- [35] Qusay Hawari, Taeseong Kim, Christopher Ward, and James Fleming. A robust gain scheduling method for a pi collective pitch controller of multi-mw onshore wind turbines. *Renewable Energy*, 192:443–455, 2022. ISSN 0960-1481. doi: <https://doi.org/10.1016/j.renene.2022.04.117>. URL <https://www.sciencedirect.com/science/article/pii/S0960148122005857>.
- [36] Parikshit Jamdade, Santosh Patil, and Shrinivas Jamdade. Assessment of power coefficient of an offline wind turbine generator system. *Electronic Journal of Energy Environment*, 1, 12 2013. doi: 10.7770/ejee-V1N3-art683.
- [37] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning, 2021. URL <https://arxiv.org/abs/1910.06378>.
- [38] Hajime KIMURA. Reinforcement learning in multi-dimensional state-action space using random tiling and gibbs sampling. *Transactions of the Society of Instrument and Control Engineers*, 42:1336–1343, 01 2006. doi: 10.9746/sicetr1965.42.1336.
- [39] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020. URL <https://arxiv.org/abs/1812.06127>.
- [40] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. URL <https://arxiv.org/abs/1509.02971>.

- [41] Q. Lu, R. Bowyer, and B.Ll Jones. Analysis and design of coleman transform-based individual pitch controllers for wind-turbine load reduction. *Wind Energy*, 18, 06 2014. doi: 10.1002/we.1769.
- [42] J.F. Manwell. Hybrid energy systems. In Cutler J. Cleveland, editor, *Encyclopedia of Energy*, pages 215–229. Elsevier, New York, 2004. ISBN 978-0-12-176480-7. doi: <https://doi.org/10.1016/B0-12-176480-X/00360-0>. URL <https://www.sciencedirect.com/science/article/pii/B012176480X003600>.
- [43] Amaresh Marekar. How DDPG (Deep Deterministic Policy Gradient) Algorithms works in reinforcement learning ? 6 2022. URL <https://medium.com/@amaresh.dm/how-ddpg-deep-deterministic-policy-gradient-algorithms-works-in-reinforcement-learn>
- [44] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023. URL <https://arxiv.org/abs/1602.05629>.
- [45] Eduardo J.N. Menezes and Alex Maurício Araújo. Wind turbine structural control using h-infinity methods. *Engineering Structures*, 286:116095, 2023. ISSN 0141-0296. doi: <https://doi.org/10.1016/j.engstruct.2023.116095>. URL <https://www.sciencedirect.com/science/article/pii/S0141029623005096>.
- [46] El Kabira El Mjabber, Abdellah El Hajjaji, and Abdellatif Khamlichi. Analysis of a rbf neural network based controller for pitch angle of variable-speed wind turbines. *Procedia Engineering*, 181:552–559, 2017. ISSN 1877-7058. doi: <https://doi.org/10.1016/j.proeng.2017.02.433>. URL <https://www.sciencedirect.com/science/article/pii/S1877705817310172>. 10th International Conference Interdisciplinarity in Engineering, INTER-ENG 2016, 6-7 October 2016, Tirgu Mures, Romania.
- [47] Jackson G. Njiri and Dirk Söffker. State-of-the-art in wind turbine control: Trends and challenges. *Renewable and Sustainable Energy Reviews*, 60:377–393, 2016. ISSN 1364-0321. doi: <https://doi.org/10.1016/j.rser.2016.01.110>. URL <https://www.sciencedirect.com/science/article/pii/S1364032116001404>.
- [48] Eduardo José Novaes Menezes, Alex Maurício Araújo, and Nadège Sophie Bouchonneau da Silva. A review on wind turbine control and its associated methods. *Journal of Cleaner Production*, 174:945–953, 2018. ISSN 0959-6526. doi: <https://doi.org/10.1016/j.jclepro.2017.10.297>. URL <https://www.sciencedirect.com/science/article/pii/S0959652617326021>.
- [49] R. Pandit, D. Infield, and M. Santos. Accounting for environmental conditions in data-driven wind turbine power models. *IEEE Transactions on Sustainable Energy*, 14(1): 168–177, 2022. doi: 10.1109/TSTE.2022.3201234.

- [50] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991. doi: 10.1162/neco.1991.3.2.246.
- [51] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 10 1986. ISSN 1476-4687. doi: 10.1038/323533a0. URL <https://doi.org/10.1038/323533a0>.
- [52] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints, 2019. URL <https://arxiv.org/abs/1910.01991>.
- [53] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *CoRR*, abs/1404.7828, 2014. URL <http://arxiv.org/abs/1404.7828>.
- [54] J.V. Seguro and T.W. Lambert. Modern estimation of the parameters of the weibull wind speed distribution for wind energy analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 85(1):75–84, 2000. ISSN 0167-6105. doi: [https://doi.org/10.1016/S0167-6105\(99\)00122-1](https://doi.org/10.1016/S0167-6105(99)00122-1). URL <https://www.sciencedirect.com/science/article/pii/S0167610599001221>.
- [55] J. Sierra-Garcia and Matilde Santos Peñas. Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control. *Neural Computing and Applications*, 34:1–15, 07 2022. doi: 10.1007/s00521-021-06323-w.
- [56] J. E. Sierra-García and M. Santos. Federated discrete reinforcement learning for automatic guided vehicle control. *Future Generation Computer Systems*, 150:78–89, 2024. doi: 10.1016/j.future.2023.08.021. URL <https://doi.org/10.1016/j.future.2023.08.021>.
- [57] J. Enrique Sierra-Garcia, Matilde Santos, and Ravi Pandit. Wind turbine pitch reinforcement learning control improved by pid regulator and learning observer. *Engineering Applications of Artificial Intelligence*, 111:104769, 2022. ISSN 0952-1976. doi: <https://doi.org/10.1016/j.engappai.2022.104769>. URL <https://www.sciencedirect.com/science/article/pii/S0952197622000598>.
- [58] Jesus Enrique Sierra-García and Matilde Santos. Wind turbine pitch control with an rbf neural network. In Álvaro Herrero, Carlos Cambra, Daniel Urda, Javier Sedano, Héctor Quintián, and Emilio Corchado, editors, *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020)*, Cham, 2021. Springer International Publishing.
- [59] Jesús Sierra-García and Matilde Santos Peñas. Exploring reward strategies for wind turbine pitch control by reinforcement learning. *Applied Sciences*, 10:7462, 10 2020. doi: 10.3390/app10217462.

- [60] Sean R. Sinclair, Siddhartha Banerjee, and Christina Lee Yu. Adaptive discretization for episodic reinforcement learning in metric spaces. *CoRR*, abs/1910.08151, 2019. URL <http://arxiv.org/abs/1910.08151>.
- [61] Sean R. Sinclair, Tianyu Wang, Gauri Jain, Siddhartha Banerjee, and Christina Lee Yu. Adaptive discretization for model-based reinforcement learning. *CoRR*, abs/2007.00717, 2020. URL <https://arxiv.org/abs/2007.00717>.
- [62] Mohit Singh, Surya Santoso, The University of Texas at Austin, and National Renewable Energy Laboratory. Dynamic models for wind turbines and wind power plants. Technical report, 10 2011. URL <https://www.nrel.gov/docs/fy12osti/52780.pdf>.
- [63] Daniel Soler, Oscar Mariño, David Huergo, Martín de Frutos, and Esteban Ferrer. Reinforcement learning to maximize wind turbine energy generation. *Expert Systems with Applications*, 249:123502, 2024. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2024.123502>. URL <https://www.sciencedirect.com/science/article/pii/S0957417424003671>.
- [64] Avanthikaa Srinivasan. Reinforcement learning: Advancements, limitations, and real-world applications. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 07, 08 2023. doi: 10.55041/IJSREM25118.
- [65] Adrian Stetco, Fateme Dinmohammadi, Xingyu Zhao, Valentin Robu, David Flynn, Mike Barnes, John Keane, and Goran Nenadic. Machine learning methods for wind turbine condition monitoring: A review. *Renewable Energy*, 133:620–635, 2019. ISSN 0960-1481. doi: <https://doi.org/10.1016/j.renene.2018.10.047>. URL <https://www.sciencedirect.com/science/article/pii/S096014811831231X>.
- [66] T. Sun, D. Li, and B. Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4289–4301, April 2023. doi: 10.1109/TPAMI.2022.3196503.
- [67] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018. URL <https://mitpress.mit.edu/books/reinforcement-learning-second-edition>.
- [68] Shize Tang, De Tian, Jianju Fang, Feng Liu, and Chenkai Zhou. Individual pitch controller characteristics analysis and optimization under aerodynamic imbalanced loads of wind turbines. *Energy Reports*, 7:6489–6500, 2021. ISSN 2352-4847. doi: <https://doi.org/10.1016/j.egyr.2021.09.114>. URL <https://www.sciencedirect.com/science/article/pii/S2352484721009197>.

- [69] The MathWorks, Inc. Control design for wind turbine. <https://nl.mathworks.com/help/control/ug/wind-turbine-control-design.html>, 2025. URL <https://nl.mathworks.com/help/control/ug/wind-turbine-control-design.html>. [Online]. Available: <https://nl.mathworks.com/help/control/ug/wind-turbine-control-design.html>. Accessed: February 12, 2025.
- [70] Anh Tuan. Paper Review: “Communication-Efficient Learning of Deep Networks from Decentralized Data”. 5 2022. URL https://medium.com/@anhtuan_40207/paper-review-communication-efficient-learning-of-deep-networks-from-decentralized-d
- [71] Zhanshan Wang, Zhengwei Shen, Chao Cai, and Kaili Jia. Adaptive control of wind turbine generator system based on rbf-pid neural network. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 538–543, 2014. doi: 10.1109/IJCNN.2014.6889538.
- [72] Christopher Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 05 1992. doi: 10.1007/BF00992698.
- [73] Jakob Weber, Markus Gurtner, Amadeus Lobe, Adrian Trachte, and Andreas Kugi. Combining federated learning and control: A survey. *IET Control Theory amp; Applications*, 18(18):2503–2523, November 2024. ISSN 1751-8652. doi: 10.1049/cth2.12761. URL <http://dx.doi.org/10.1049/cth2.12761>.
- [74] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14:1–23, 11 2022. doi: 10.1007/s13042-022-01647-y.
- [75] Jerry C. Whitaker. *Microelectronics*. 10 2018. doi: 10.1201/9781315220482. URL <https://doi.org/10.1201/9781315220482>.
- [76] J. Xie, H. Dong, and X. Zhao. Power regulation and load mitigation of floating wind turbines via reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 2023. doi: 10.1109/TASE.2023.XXXXXXX.
- [77] Jingjie Xie, Hongyang Dong, and Xiaowei Zhao. Power regulation and load mitigation of floating wind turbines via reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, PP:1–12, 01 2023. doi: 10.1109/TASE.2023.3295576.
- [78] Bo Xu, Wenchao Xia, Haitao Zhao, Yongxu Zhu, Xinghua Sun, and Tony Q. S. Quek. Clustered federated learning in internet of things: Convergence analysis and resource optimization. *IEEE Internet of Things Journal*, 11(2):3217–3232, 2024. doi: 10.1109/JIOT.2023.3298366.

-
- [79] Huizhen Yu. On convergence of value iteration for a class of total cost markov decision processes. *SIAM Journal on Control and Optimization*, 53(4):1982–2016, January 2015. ISSN 1095-7138. doi: 10.1137/141000294. URL <http://dx.doi.org/10.1137/141000294>.
- [80] Betul Yurdem, Murat Kuzlu, Mehmet Kemal Gullu, Ferhat Ozgur Catak, and Maliha Tabassum. Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon*, 10(19):e38137, 2024. ISSN 2405-8440. doi: <https://doi.org/10.1016/j.heliyon.2024.e38137>. URL <https://www.sciencedirect.com/science/article/pii/S2405844024141680>.
- [81] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021. doi: 10.1016/j.knosys.2021.106775.
- [82] Bowen Zhou, Zhibo Zhang, Guangdi Li, Dongsheng Yang, and Matilde Santos. Review of key technologies for offshore floating wind power generation. *Energies*, 16(2):710, 2023. doi: 10.3390/en16020710.