

Evaluación de desempeño de algoritmos de aprendizaje reforzado aplicados al control resistivo de un generador undimotriz

Danilo Pastrana-Mendoza¹, Braulio Neira-Verdugo², Iván Gutiérrez-Pilar³, Fabián Pierart-Vásquez⁴

¹ Departamento de Ingeniería Mecánica, Universidad del Bío-Bío, Concepción, Chile. Email: dpastrana@ubiobio.cl

² Departamento de Ingeniería Mecánica, Universidad del Bío-Bío, Concepción, Chile. Email: braulio.neira2001@alumnos.ubiobio.cl

³ Departamento de Ingeniería Mecánica, Universidad del Bío-Bío, Concepción, Chile. Email: ivan.gutierrez2001@alumnos.ubiobio.cl

⁴ Departamento de Ingeniería Mecánica, Universidad del Bío-Bío, Concepción, Chile. Email: fpierart@ubiobio.cl

Resumen

En la presente investigación tres algoritmos de aprendizaje reforzado fueron aplicados para el control resistivo de un convertidor de energía de las olas del tipo absorbedor puntual. Una evaluación del rendimiento de estos algoritmos del tipo on-policy y off-policy fue llevado a cabo emulando un entorno de olas regulares. Los resultados muestran que los tres algoritmos probados logran alcanzar la máxima potencia disponible. Además, se destaca la importancia del ajuste de hiperparámetros para la implementación efectiva de estos algoritmos, especialmente aquellos que utilizan redes neuronales en su estructura de decisión.

Palabras clave: Generador undimotriz, control resistivo, aprendizaje reforzado

Abstract

In this document, three reinforcement learning algorithms are applied to the resistive control of a point absorber type wave energy converter. The performance of both on-policy and off-policy algorithms is evaluated in a regular wave environment. The results demonstrate that all three tested algorithms achieve maximum power extraction. Furthermore, the importance of hyperparameter tuning is emphasized for effectively implementing these algorithms, particularly those utilizing neural networks in their decision-making structure.

Keywords: Wave energy converter, Resistive control, Reinforcement learning

1. Introducción

El alza creciente de la demanda energética, principalmente impulsada por países en vías de desarrollo como China e India, y el cambio de paradigma mundial hacia un desarrollo sostenible, ha llevado a los investigadores a buscar nuevas fuentes de energía que respondan a ambos requerimientos [1].

Una opción que puede aportar al objetivo de descarbonizar la matriz energética mundial es el desarrollo de la tecnología de generación de a través del mar. EL recurso energético de las olas (energía undimotriz) tiene el potencial de proporcionar una fuente constante y predecible de energía y una elevada densidad energética calculada en 146 TWh/año, lo que las hace especialmente valiosas en un mix energético diversificado [2; 3].

La tecnología undimotriz muestra, dentro de la gama de tecnología marinas, ser una propuesta de alto crecimiento en el futuro. Sin embargo, actualmente aún requiere de mecanismos más eficientes para que sea competitiva[2; 4; 5]. La principal razón de este estancamiento es el alto costo nivelado de energía (LCOE) de estos dispositivos, lo que los hace poco competitivos frente a otras tecnologías y, por consiguiente, económicamente inviables [4; 6].

A pesar de los esfuerzos en utilización de enfoques de optimización estructural, se ha determinado que el uso de sistemas de control más sofisticados puede tener un impacto más significativo en la disminución del coste unitario de generación [7; 8].

Las inherentes no linealidades presentes en la interacción fluido-estructura que describen la dinámica de

los convertidores undimotrices han obligado a la comunidad a trabajar con modelos simplificados, que a pesar de funcionar en la mayor cantidad de casos, carece de realismo. Para evitar errores en los sistemas de control, la estrategia de model predictive control (MPC) ha sido preferido por los investigadores, ya que, a pesar de contener modelos lineales, es capaz de captar las no linealidades de la dinámica de los dispositivos arrojando muy buenos resultados[9; 10]. Sin embargo, esta estrategia es muy sensible errores de predicción, afectando fuertemente su rendimiento en la tarea de maximizar la potencia generado por convertidores de energía de las olas (WEC) [11; 10].

Un importante desafío, necesario va de resolver es la implementación efectiva de estas estrategias de control, para compatibilizar su viabilidad técnica, capacidad de calculo y rendimiento en la tarea de maximizar la potencia generada. Mientras el control resistivo es preferible por su simplicidad, su rendimiento en potencia generada es bajo. Por otro lado, el control reactivo es capaz de mejorar el rendimiento que el control resistivo, sin embargo su implementación es compleja debido a la serie de cálculos necesarios en oleajes irregulares y su necesidad de un PTO flexible que cambie de generador a motor de forma sucesiva. La estrategia de enganche ha otorgado buenos resultados manteniendo en fase el dispositivo y la fuerza de excitación, pero las grandes fuerzas generadas en el sistema de enganche lo hacen poco atractivo. Finalmente, variaciones de MPC se han propuesto en diferentes artículos demostrando sus buenos resultados, pero como ya fue mencionado, su sensibilidad ante errores de predicción, afectando fuertemente su rendimiento [12; 13; 14].

Los avances acelerados de la inteligencia artificial han impactado positivamente diversas áreas del conocimiento y el sector energético no ha sido la excepción [15; 16; 17]. En lo que a convertidores de energía undimotriz se trata, se han utilizado redes neuronales para apoyar procesos de optimización [18], así como también en la determinación del mejor arreglo de granjas de WEC[19]. Recientemente, las técnicas de machine learning han sido aplicadas a los sistemas de control y han resuelto efectivamente la tarea de maximizar la potencia generada. Haider et al[13] ha comparado el rendimiento de MPC y machine learning, donde este ultimo ha demostrado un rendimiento ligeramente superior a MPC, sin embargo su adaptabilidad a diferentes condiciones de oleaje, distintos equipos e incluso a la dinámica cambiante del WEC producto de la incrustación de biofouling, lo posicionan en una técnica digna de investigar. Algoritmos como Q-learning, Deep Q-learning, State-action-reward-state-action (SARSA), Least-Squares Policy Iteration (LSPI), Soft Actor-Critic (SAC) e incluso redes neuronales han aplicados en WEC. A pesar de lo anterior, la literatura carece de comparación entre todos estos algoritmos que permitan vislumbrar el más adecuado en base a criterios como sensibilidad a hiper parámetros, costo computacional y rapidez de conver-

gencia [3; 7; 17; 20; 21; 22; 23; 24; 25].

La contribución de este trabajo consiste en evaluar el desempeño de 3 algoritmos de aprendizaje reforzado cuyo objetivo es maximizar la energía extraída de generador undimotriz del tipo absorbedor puntual, utilizando una estrategia de control resistivo. Estos algoritmos son desarrollados en lenguaje Python y su desempeño es evaluado a partir de factores como, la facilidad de implementación, el tiempo de convergencia y la precisión del algoritmo para encontrar el estado que entregue la máxima potencia. Dentro de los algoritmos estudiados se presenta dos algoritmos off-policy, Least Squares Policy Iteration (LSPI) y Deep Q Network (DQN), y uno del tipo on-policy, Advantage Actor Critic (A2C). Estos resultados además son comparados con el algoritmo Q-learning del tipo off-policy, implementado en trabajos previos, para evaluar las ventajas y desventajas de las distintas clases de algoritmos de aprendizaje reforzado.

2. Metodología

2.1. Modelo hidrodinámico absorbedor puntual

Para el desarrollo de este trabajo se considera un absorbedor puntual de un grado de libertad esquematizado en la figura 1.

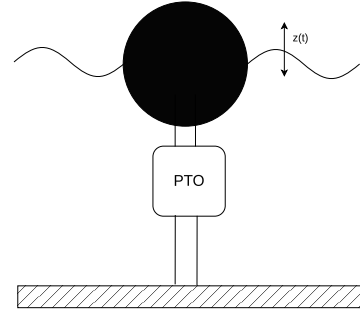


Figura 1: Absorbedor puntual un grado de libertad.

Despreciando el roce entre los elementos mecánicos, los efectos viscosos originados en la interacción fluido-estructura y asumiendo pequeños movimientos del cuerpo flotante, mediante la segunda ley de Newton es posible determinar la ecuación (1) que describe el movimiento de un WEC .

$$(M + m_{add}(\omega))\ddot{z} + C_{rad}(\omega)\dot{z} + K_{hs}(\omega)z = F_{ext}(\omega) + F_{PTO}, \quad (1)$$

donde:

M	: Masa del objeto flotante
$m_{add}(\omega)$: Masa añadida
$C_{rad}(\omega)$: Amortiguamiento radiativo
$K_{hs}(\omega)$: Rigidez hidrostática
$F_{exc}(\omega)$: Fuerzas excitación
F_{PTO}	: Fuerza ejercida por el PTO
ω	: Frecuencia de excitación

La expresión descrita en (1) corresponde a la ecuación de Cummins, que define la dinámica de un cuerpo flotante de sección constante en la línea de flotación expuesta a fuerzas hidrostáticas, hidrodinámicas y aplicadas por un sistema absorbedor de potencia (PTO) [14].

2.2. Control Resistivo

Al desarrollar el equivalente de (1) en el dominio de la frecuencia y considerando una fuerza ejercida por el PTO (\hat{F}_{PTO}) de tipo lineal para cumplir con el principio de superposición, como muestra la expresión (2), se puede demostrar que la máxima potencia se logra mediante el ajuste de dos parámetros asociados a la fuerza del PTO: B_{PTO} y K_{PTO} [14; 26].

$$\hat{F}_{PTO} = -i\omega B_{PTO}\hat{Z}(\omega) - K_{PTO}\hat{Z}(\omega), \quad (2)$$

con $\hat{Z}(\omega)$, B_{PTO} y K_{PTO} , como la amplitud de movimiento compleja, el amortiguamiento y rigidez del PTO respectivamente.

Del control óptimo se puede desprender el esquema de control conjugado complejo (ACC por sus siglas en inglés) que se muestra en la figura 2, que contempla la aplicación de la fuerza del PTO para llevar al sistema a la resonancia. Sin embargo, este esquema de control es impracticable debido a la necesidad de conocimiento de las fuerzas de excitación futuras del oleaje [27].

Una aproximación sub-óptima de este enfoque es el control resistivo, que consiste en determinar el amortiguamiento del PTO óptimo para lograr una máxima potencia de generación considerando un $K_{PTO}=0$. Su fácil implementación lo hace ideal para dispositivos en etapas de prototipado.

El valor de B_{PTO} , que maximiza la energía extraída, depende de la frecuencia de las olas, las características del generador y las condiciones del entorno marino. Aunque su ajuste puede realizarse mediante métodos experimentales o teóricos, estos enfoques suelen ser insuficientes debido a las no linealidades propias de la dinámica de los dispositivos undimotrices y la variabilidad del mar. Una alternativa más flexible para determinar el amortiguamiento óptimo es el uso de técnicas de aprendizaje automático, como Reinforcement Learning, que se detallará en la siguiente sección.

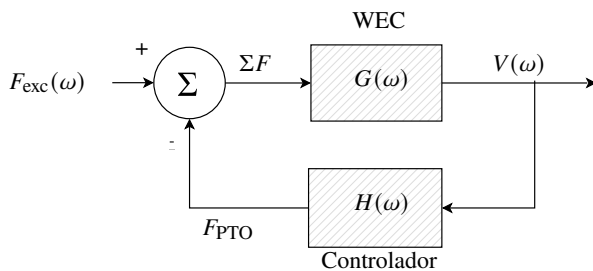


Figura 2: Esquema de control ACC.

2.3. Aprendizaje reforzado

El aprendizaje por refuerzo (RL, por sus siglas en inglés) es una rama del aprendizaje automático donde un agente aprende a tomar decisiones mediante la interacción con su entorno, buscando maximizar una recompensa acumulada a largo plazo. Este enfoque se basa en el concepto de prueba y error, donde el agente explora diferentes acciones y utiliza la retroalimentación obtenida para mejorar su comportamiento futuro.

El funcionamiento de estos algoritmos se puede modelar como un proceso de decisión de Markov como se muestra en la figura 3, donde el agente, que corresponde al algoritmo en cuestión, recibe datos del estado actual del entorno (s) y en base a ello genera una acción (a) sobre el obteniendo una recompensa (r) por el cambio de estado generado en el entorno [28]. Una analogía con el esquema de control retro alimentado es posible al considerar al agente como el controlador utilizado, el entorno como el sistema a controlar, y finalmente la acción, estado y recompensa representan a la acción del actuador, la medición del sensor y la señal de control respectivamente.

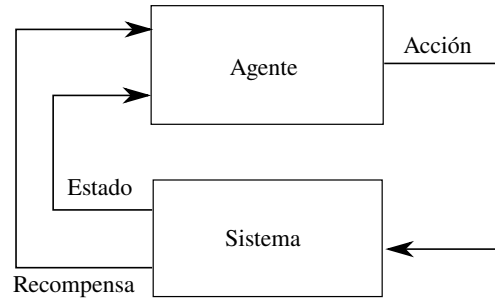


Figura 3: Esquema de decisión de Markov.

Una de las grandes beneficios de la utilización de estos algoritmos es que no necesitan un modelo del sistema a controlar. Esto es útil debido a que los modelos desarrollados en muchas ocasiones carecen de precisión debido a simplificaciones realizadas, que desprecian efectos físicos que afectan la dinámica del sistema, y no tienen en cuenta los cambios en el modelo debido a efectos externos no considerados. Por lo tanto, la aplicación de estos algoritmos de aprendizaje nos otorga adaptabilidad al considerar la dinámica real del sistema.

Los algoritmos de aprendizaje por refuerzo han demostrado ser efectivos en una amplia gama de aplicaciones, desde el control de robots hasta la optimización de sistemas complejos y el desarrollo de estrategias en juegos. Su importancia radica en su capacidad para manejar problemas complejos sin la necesidad de tener un modelo que lo describa, permitiendo a los sistemas aprender y adaptarse de manera autónoma en entornos cambiantes [25; 29].

En los últimos años, se han aplicado diversos algoritmos de RL para maximizar la potencia generada de los WEC. Entre ellos se destaca Q-learning, Least Squart

Policy Iteration (LSPI), Deep Q-network (DQN) y Advantage Actor-Critic (A2C), que serán objeto de estudio en este trabajo.

2.3.1. Q-learning

Q-learning es un algoritmo clásico de aprendizaje por refuerzo, clasificado como model-free y off-policy.

Este algoritmo permite determinar la correcta selección de la acción a en un estado s que maximice la recompensa acumulada R , al lograr estados futuros s' favorables en relación al objetivo del agente. Se basados en una exploración del ambiente mediante la política epsilon-greedy, que selecciona una acción aleatoria una probabilidad de ϵ y selecciona la acción más favorable con una probabilidad de $1 - \epsilon$.

Como es posible determinar en el pseudocódigo 1, el algoritmo actualiza iterativamente un arreglo $\mathbf{Q} \in \mathbb{R}^{|S| \times |A|}$, donde S y A corresponden a el espacio de estados y de acciones respectivamente, denominada Tabla \mathbf{Q} . Esta tabla contiene la función de valor de acción $Q(s, a)$, cuya determinación se hace a través de la ecuación 3, que deriva del método de diferencia temporal.

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a')) \quad (3)$$

El algoritmo es capaz de encontrar la secuencia de acciones óptima mediante el mapeo de pares estado-acción en el entorno. Este proceso se encuentra dominado por parámetros como la tasa de aprendizaje (α), la tasa de descuento (γ) y la tasa de exploración (ϵ), que determinarán la rapidez, la importancia de información futura y la capacidad de exploración del agente.

Debido a la tabulación de los pares estado-acción, Q-learning solo es capaz de trabajar en estados discretos y se vuelve ineficiente en espacios de grandes estados y acciones. A pesar de estas desventajas, ha mostrado su efectividad en la maximizaron de potencia para los WEC en diversos artículos [6; 7; 21; 30].

Algorithm 1 Pseudocódigo Q-learning

```

1: Inicializar Tabla  $Q$ ,  $Q(s, a) \rightarrow 0$ 
2: Inicializar el entorno
3: Inicializar  $\epsilon$ 
4: for  $t = 1, \dots, N_{\text{step}}$  do
5:   if  $\text{rand}() < \epsilon$  then
6:     Seleccionar aleatoriamente  $a \in A$ 
7:   else
8:      $a = \arg \max_{a'} Q(s, a)$ 
9:   end if
10:  Cambiar a estado  $s'$  y observar una recompensa  $r$ 
11:  Determinar los pares acción-estado posibles ( $s', a'$ )
12:  Actualizar valor de  $Q$ :
     $Q(s, a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$ 
13:  Actualizar estado  $s = s'$ 
14:  Actualizar política  $\epsilon$ -greedy:  $\epsilon \leftarrow \epsilon * \epsilon_{\text{decay}}$ 
15: end for
```

2.3.2. Deep Q Network (DQN)

Clasificado como model-free y off-policy, Deep Q-Learning o Deep Q-Network (DQN) es un algoritmo robusto que comparte principios fundamentales con Q-Learning, pero con mejoras significativas en su implementación.

A diferencia de Q-Learning, que utiliza una tabla Q para almacenar cada combinación posible de estado y acción, DQN emplea redes neuronales profunda para parametrizar la función de valor de acción $Q(s, a)$. [31]

Una ventaja destacada de DQN es su capacidad para manejar estados continuos, comunes en muchos problemas donde Q-Learning no sería eficiente debido a la escalabilidad de la tabla Q . Además, gracias al uso de redes neuronales, DQN puede adaptarse a nuevos estados sin necesidad de almacenar explícitamente cada combinación estado-acción, lo cual es una limitación significativa del Q-Learning. Sin embargo, este algoritmo presenta ciertas desventajas asociadas a la sintonización de la red neuronal y sus altos costos computacionales.

Este algoritmo utiliza dos redes neuronales, Q_{main} y Q_{target} , para aproximar la función de valor de acciones $Q(s, a)$ mediante parámetros θ , correspondientes a los pesos de la red principal. Notar que la función de valor solo esta parametrizada por los pesos de solo una red, esto es debido a que Q_{main} es la red encargada de la interacción con el ambiente y Q_{target} tiene la tarea de la predicción de estados futuros.

Inicialmente ambas redes son idénticas en arquitectura y pesos, para posteriormente solo ajustar los pesos de la red principal (Q_{main}) con el fin de evitar divergencias en el entrenamiento. Posteriormente, los pesos ajustado por la red principal a la red objetivo (Q_{target}) luego de N iteraciones.

El entrenamiento de la red neuronal principal se logra mediante la minimización de la función de costo $L(\theta)$, que corresponde al error cuadrático medio de las salidas de la red Q_{target} y Q_{main} generadas por una muestra aleatoria de K transiciones (s, a, r, s') .

$$L(\theta) = \frac{1}{K} \sum_{i=1}^K (Q_{\text{target}, i} - Q(s_i, a_i; \theta))^2 \quad (4)$$

Los pesos son actualizados mediante el uso de gradiente descendente, como muestra la ecuación 5.

$$\theta = \theta - \alpha \nabla_{\theta} L(\theta) \quad (5)$$

Para mayor claridad del funcionamiento referirse al pseudocódigo 2.

2.3.3. Least-Squares Policy Iteration (LSPI)

Desarrollado por Lagoudakis y Parr (2003) [32], LSPI este es un tipo de algoritmo de aprendizaje reforzado de tipo off-policy, diseñado para resolver problemas de control continuo y discreto sin depender de un modelo explícito del sistema (model-free).

En este algoritmo, la política óptima se aproxima mediante la evaluación y mejora iterativa de la función

Algorithm 2 Pseudocódigo Deep Q-Network

```

1: Inicializar el entorno
2: Inicializar  $\epsilon$ 
3: Inicializar la memoria de repetición  $\mathcal{D}$ 
4: Inicializar la red  $Q_{main}$  con parámetros aleatorios  $\theta$ 
5: Inicializar la red objetivo  $Q_{target}$  con pesos  $\theta^* = \theta$ 
6: Obtener el estado inicial  $s_0$ 
7:
8: for iter = 1, ...,  $N_{step}$  do
9:   if rand() <  $\epsilon$  then
10:    Seleccionar aleatoriamente  $a \in \mathcal{A}$ 
11:   else
12:     $a = \arg \max_{a'} Q_{main}(s, a; \theta)$ 
13:   end if
14:   Cambiar a estado  $s'$  y observar recompensa  $r$ 
15:   if longitud de  $\mathcal{D} \geq L_{ote\_size}$  then
16:    Muestrear aleatoriamente un minilote de transiciones  $(s_i, a_i, r_i, s'_i)$  de  $\mathcal{D}$ 
17:    Calcular  $Q_{target, i}$ :
18:     $Q_{target, i} = r_i + \gamma \cdot \max_{a'} Q_{target}(s'_i, a'_i; \theta^*)$ 
19:    Actualizar  $Q_{main}(s_i, a_i; \theta)$ 
20:    Calcular la función de pérdida  $L(\theta)$ 
21:    Actualizar los pesos  $\theta$  de la red  $Q_{main}$ 
22:   end if
23:   Actualizar el estado  $s = s'$ .
24:   Actualizar política  $\epsilon$ -greedy:  $\epsilon \leftarrow \epsilon * \epsilon_{decay}$ 
25:   Cada  $N$  iteraciones, actualizar la red  $Q_{target}$ :  $\theta^* = \theta$ 
26: end for

```

$Q(s, a)$ a partir de una función de aproximación paramétrica. Esto brindando una ventaja en comparación a la tabulación realizada por el algoritmo Q-learning, debido a que es aplicable a espacios de grandes estados.

La función $Q_\theta(s, a)$ que busca ajustar este algoritmo, está parametrizada en términos de un conjunto de parámetros θ , utilizando una combinación lineal de funciones base $\phi(s, a)$ como muestra la expresión siguiente.

$$Q(s, a) \approx \phi(s, a)^T \theta \quad (6)$$

El vector de forma pueden ser calculados mediante una función de forma de tipo tabular o radial. Al igual que en [33], para fines de esta investigación se emplean funciones de base radial descrita en la ecuación (7). Como muestra la figura 4, estas funciones son localizadas, que se activan solo para ciertos estados o entradas cercanas al centro de la función base. Las funciones base radiales son especialmente útiles para capturar la estructura no lineal en la dinámica del sistema, ya que permiten aproximar funciones complejas con un número relativamente pequeño de parámetros, manteniendo una estructura eficiente y escalable.

$$\phi(s, a) = \exp\left(-\frac{|s - s_i|}{2\mu_i}\right) \quad (7)$$

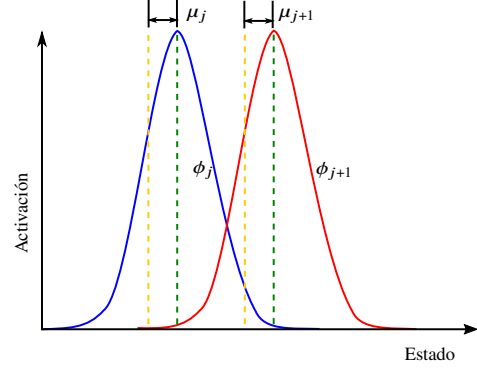


Figura 4: Función de activación de las funciones de base radial [34].

El ajuste de la función de valor de acciones descrita en (6), se obtiene mediante la actualización de los parámetros θ mediante un problema de mínimos cuadrados que se muestra en la ecuación (8). En este contexto, la matriz A acumula la información de las transiciones entre estados ponderada por el factor de descuento, mientras que la matriz b refleja la suma de las recompensas inmediatas. El algoritmo LSPI iterativamente ajusta θ para minimizar el error en la función de valor $Q(s, a)$, y la política se actualiza con la acción que maximiza esta función para cada estado.

$$\theta = A^{-1}b \quad (8)$$

$$A = A + \phi(s)(\phi(s) - \gamma\phi(s', \pi'))^T \quad (9)$$

$$b = b + \phi(s)r \quad (10)$$

$$\pi(s) = \max_{a \in A} Q(s, a) \quad (11)$$

EL algoritmo 3 muestra el pseudo-código de este algoritmo de RL para mayor comprensión de su funcionamiento.

2.3.4. Advantage Actor Critic (A2C)

El algoritmo A2C (Advantage Actor-Critic) es un método de aprendizaje por refuerzo del tipo on-policy, que combina las ventajas de los enfoques de aprendizaje basados ajustes de funciones de valores y basados en ajustes de políticas.

Se compone de dos redes neuronales: actor y crítico, donde el actor intenta aprender una política estocástica $\pi(a|s; \theta)$, parametrizada por factores θ , y crítico que aprende una función de valor $V(s; \phi)$ parametrizada por ϕ [35; 36]. La función en conjunto de estas dos redes neuronales permite la actualización iterativa de ambas redes para lograr la determinación de una distribución de probabilidades sobre el espacio de acciones respecto a un estado actual s con el fin de maximizar la recompensa acumulada en el futuro.

La actualización del actor se basa en la maximización de la función de pérdida $L_{actor}(\theta)$, para maximizar la probabilidad de acciones con ventaja A positiva modificando los pesos θ de la red mediante el gradiente

Algorithm 3 Pseudocódigo Least Squares Policy Iteration

```

1: Inicializar el entorno
2: Inicializar  $\epsilon$ 
3: Definir  $\theta$  albitrariamente
4: Inicializar la memoria de repetición  $\mathcal{D}$ 
5: Definir matriz  $A$  y vector  $b$ :  $A \leftarrow 0, b \leftarrow 0$ 
6:
7: for iter = 1, ...,  $N_{\text{step}}$  do
8:   if rand() <  $\epsilon$  then
9:     Elegir  $a$  aleatoriamente
10:  else
11:     $a = \arg \max_{a'} Q(s, a; \theta)$ 
12:  end if
13:  Ejecutar acción  $a$  en el entorno y observar recompensa  $r$  y nuevo estado  $s'$ 
14:  Almacenar la transición  $(s, a, r, s')$  en memoria  $\mathcal{D}$ 
15:  Calcular vector de forma actual:  $\phi_t \leftarrow \phi(s, a)$ 
16:  Calcular vector de forma siguiente en base a la política actual:  $\phi_{t+1} \leftarrow \phi(s', \pi_{\theta_t}(s'))$ 
17:  Actualizar matriz  $A$  y vector  $b$ 
     $A \leftarrow A + \phi_t(\phi_t - \gamma\phi_{t+1})^\top$ 
     $b \leftarrow b + \phi_t r$ 
18:  Actualizar pesos:  $\theta_{t+1} \leftarrow A^{-1}b$ 
19:  Actualizar política:  $\pi_{\theta_t}(s) = \pi_{\theta_{t+1}}(s)$ 
20:  Actualizar política  $\epsilon$ -greedy:  $\epsilon \leftarrow \epsilon * \epsilon_{\text{decay}}$ 
21: end for

```

positivo de la función de pérdida.

$$\nabla_{\theta} L_{\text{actor}}(\theta) = \nabla_{\theta} \log \pi(a|s; \theta) A(s, a), \quad (12)$$

$$\theta = \theta + \alpha \nabla_{\theta} L_{\text{actor}}(\theta), \quad (13)$$

En las ecuaciones anteriores, $A(s)$ es la función de ventaja, que representa la calidad de tomar la acción a en el estado s en comparación con la acción promedio bajo la política actual.

La ventaja se estima utilizando la siguiente expresión, donde r es la recompensa inmediata y γ es el factor de descuento.

$$A(s) = r + \gamma V(s'; \phi) - V(s; \phi) \quad (14)$$

Notar que en la expresión (14), $V(s'; \phi)$ corresponde a la función de valor de estados parametrizada por los factores ϕ dados por la red crítico. De aquí se puede decir que la red crítico apoya a la red actor para brindarle información de cuan buena es la política que ha ajustado.

Para lograr una buena estimación de la función de valor, se debe minimizar una función de costo correspondiente al error cuadrático medio de la función de ventaja del error cuadrático, para posterior ajustar los parámetros de la red crítico con el gradiente descendiente de dicha función de pérdida. Por otro lado, la actualización del crítico se basa en la siguiente función de pérdida cuadrática entre el valor estimado del estado actual y el objetivo (ventaja).

$$\phi = \phi - \alpha \nabla_{\theta} L_{\text{critico}}(\phi), \quad (15)$$

A2C modifica los pesos de la red actor y crítico simultáneamente en cada iteración utilizando el mismo conjunto de experiencias, lo que mejora la eficiencia del

aprendizaje. Además, el uso de la ventaja como estimador de la función objetivo de política reduce la varianza y mejora la estabilidad del entrenamiento [37].

EL siguiente pseudocódigo detalla con mayor claridad como funciona el algoritmo descrito en esta sección.

Algorithm 4 Pseudocódigo A2C

```

1: Inicializar la red neuronal actor y la red critic con parámetros  $\theta$  y  $\phi$ 
2: Inicializar el entorno
3: Obtener el estado inicial  $s_0$ 
4:
5: for iter = 1, ...,  $N_{\text{step}}$  do
6:   Seleccionar una acción  $a$  según la política  $\pi_{\theta}(a|s)$ 
7:   Ejecutar acción  $a$  en el entorno, obtener recompensa  $r$  y nuevo estado ( $s'$ )
8:   Calcular la ventaja
     $A = r + \gamma V_{\phi}(s') - V_{\phi}(s)$ 
9:   Actualizar la red Actor mediante el gradiente de la política:
     $\theta \leftarrow \theta + \alpha_{\text{actor}} \nabla_{\theta} \log \pi_{\theta}(a|s_t) A$ 
10:  Actualizar la red Critic mediante el gradiente de la función de valor:
     $\phi \leftarrow \phi - \alpha_{\text{critic}} \nabla_{\phi} (A)^2$ 
11:  Actualizar el estado  $s = s'$ 
12: end for

```

2.4. Implementación de RL en el control resistivo

En este artículo se implementará un control resistivo utilizando los algoritmos de aprendizaje reforzado descritos en la sección 2.2. Para ello, se considerará un dispositivo undimotriz del tipo absorbedor puntual propuesto por Pierart et al. (2023) [38].

Como muestra la figura 5, el dispositivo undimotriz se compone de un sistema de adquisición de energía mecánica (piñón-cremallera), con un potenciómetro incluido en el circuito eléctrico, para cumplir el propósito de modificar del amortiguamiento del PTO (B_{PTO}).

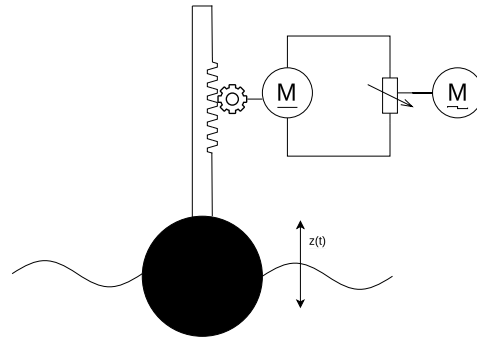


Figura 5: Esquema absorbedor puntual.

Para obtener la respuesta del sistema ante las acciones seleccionadas por los algoritmos, las ecuaciones desarrolladas en [38], que describen las dinámica del

sistema, serán utilizadas y son resumidas a continuación.

$$(M + m_{\text{add}}(\omega) + \frac{1}{r^2})\ddot{z} + (C_{\text{rad}}(\omega) + \frac{D}{r^2} + B_{\text{PTO}}\dot{z} + K_{\text{hs}}z = F_{\text{exc}}(\omega) \quad (16)$$

$$R_L = 0.549 * a \quad (17)$$

$$B_{\text{PTO}} = \frac{1}{r^2} \frac{(k_m)^2}{R_a + R_L} \quad (18)$$

$$P_e = B_{\text{PTO}}\dot{z}^2 \quad (19)$$

Donde:

- a : Ángulo potenciómetro
- r : Radio del piñón
- D : Coeficiente de fricción viscosa del motor
- k_m : Constante de proporcionalidad entre la EMF y velocidad
- R_a : Resistencia interna del motor
- R_L : Resistencia variable
- P_e : Potencia extraída

Como ya fue mencionado anteriormente, la tarea de control asignada a los algoritmos será actuar sobre el potenciómetro, modificando el ángulo que este posea y con ello lograr afectar al amortiguamiento del PTO. Es por esto que se define un espacio de estados \mathcal{S} del agente compuesto por ángulos entre 0 y 90°, discretizado en 1.8°. Por otro lado, se define el espacio de acciones \mathcal{A} con las siguientes acciones posibles sobre los estados: [Disminuir; Mantener; Aumentar].

El objetivo del control del dispositivo undimotriz es maximizar la potencia generada. Para representar este propósito en los algoritmos, la recompensa inmediata r se caracterizará según la por la expresión (20), donde P_{mean} corresponde a la potencia promedio generada por el dispositivo undimotriz en un periodo de tiempo de 10 segundos en un oleaje regular.

$$r = \begin{cases} P_{\text{mean}}^{45}, & \text{si } P_{\text{mean},i-1} \leq P_{\text{mean},i} \\ -10000, & \text{si } P_{\text{mean},i-1} > P_{\text{mean},i} \end{cases} \quad (20)$$

Las valores de variables necesarias para determinar la dinámica del dispositivo undimotriz fueron obtenidas de [21] y son resumidos en la tabla 1.

Los hiperparámetros utilizados en los algoritmo DQN, LSPI y A2C fueron ajustados mediante una metodología de prueba-error. En el caso de Q-learning, las configuraciones implementadas fueron obtenidas de [21].

Un resumen de las características de los algoritmos se detallan en la tabla 2.

Cada uno de los algoritmos mencionados será desarrollados en lenguaje Python, considerando la ejecución de un episodio con una duración de 200 pasos de tiempo, debido a la simplicidad del entorno emulado.

Tabla 1: Valores utilizados para la simulación

Parámetro	Unidad	Valor
r	m	0.00695
D	Nms	0.000039
R_a	Ω	7.2
k_m	V/rad/s	0.05
K_{hs}	N/m	345.65326
M	kg	3.25
A_I (Amplitud de ola)	m	0.0185
ω	Hz	1.3
F_{exc}	N/m	156.815
C_{rad}	Ns/m	6.64
m_{add}	kg	1.2
L_a (Inductancia del motor)	H	0.9

Tabla 2: Tabla de configuraciones para diferentes algoritmos

Algoritmo	Parámetros
LSPI	$\epsilon = 0.95$ $\epsilon_{\text{decay}} = 0.9$ $\gamma = 0.75$ $\mu = 0.5$
DQN	$\epsilon = 0.95$ $\epsilon_{\text{decay}} = 0.98$ $\gamma = 0.5$ $\alpha = 0.001$ Batch size = 100 Arquitectura de redes: Dos capas ocultas de 26 neuronas
A2C	$\gamma = 0.95$ $\alpha_{\text{actor}} = 0.001$ $\alpha_{\text{critic}} = 0.001$ Arquitectura de redes: Dos capas ocultas de 128 neuronas

3. Resultados

De la Figura 6, que muestra la potencia generada para cada ángulo contenido dentro del espacio de estados, se puede determinar que la máxima potencia generada es de 4.019 mW, obtenida al fijar un ángulo de 21.6° en el potenciómetro. Además, se identifica un conjunto de estados favorables, comprendidos entre 14.4° y 30.6°, en los cuales se alcanza al menos el 95 % de la potencia máxima.

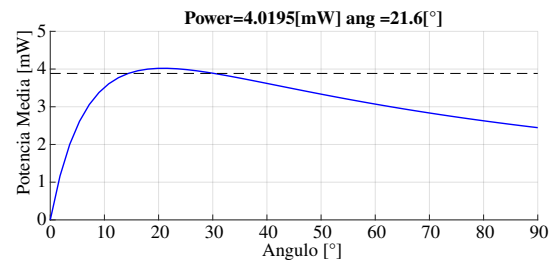


Figura 6: Relación entre ángulos y potencia generada.

Las figuras 7, 8, 9 y 10 presentan los resultados de la implementación de cada uno de los algoritmos de aprendizaje por refuerzo en el sistema estudiado. En la Tabla 3 se resumen los resultados más relevantes de

dichos algoritmos.

Tabla 3: Resultados de la implementación de algoritmos

Algoritmo	Tiempo de cálculo (s)	P_{\max} (mW)
Q-learning	0.014	4.014
DQN	1740.43	4.015
LSPI	0.648	3.824
A2C	1.212	3.994

Es importante destacar la capacidad de cada uno de los algoritmos para encontrar y mantenerse dentro del conjunto de estados favorables, logrando al menos el 95 % de la potencia máxima.

En la Figura 7, se observa el desempeño de Q-learning, donde el sistema converge a una política estable en aproximadamente 180 iteraciones. El algoritmo logra acercarse a los estados favorables en solo 30 iteraciones, lo que le permite explotar la máxima capacidad de generación del dispositivo desde etapas muy tempranas.

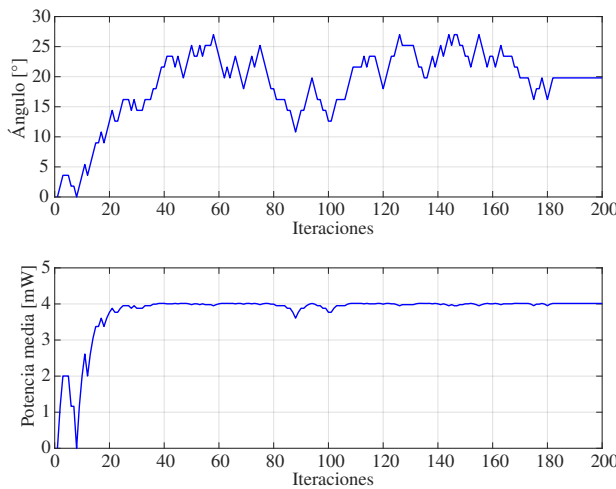


Figura 7: Resultados implementación Q-learning.

Por su parte, la figura 8 muestra los resultados del algoritmo DQN. Aunque logra una potencia máxima solo un poco por debajo de la alcanzada por Q-learning, presenta un consumo computacional significativamente mayor en comparación con los demás algoritmos implementados, producto del proceso de ajuste de parámetros de red. Lo anterior, hace que este algoritmo poco adecuado para estrategias de control en entornos altamente dinámicos, como es el caso del mar.

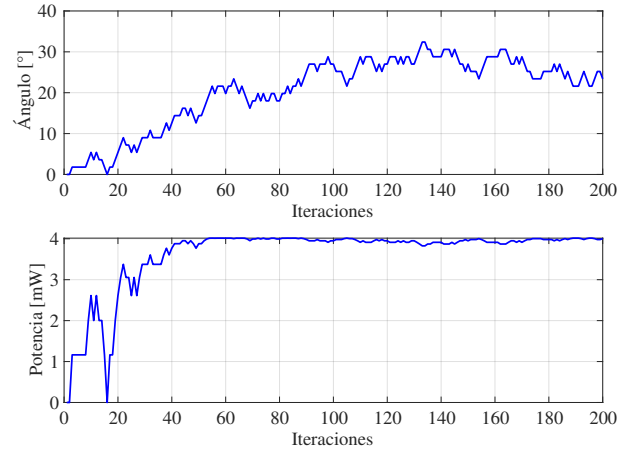


Figura 8: Resultados implementación DQN.

En la Figura 9, que ilustra los resultados de LSPI, se puede notar que este algoritmo también tiende a estancarse en políticas subóptimas. Esto sugiere que los valores iniciales o los hiperparámetros elegidos afectaron negativamente su rendimiento. No obstante, su comportamiento es similar al de Q-learning en cuanto a convergencia.

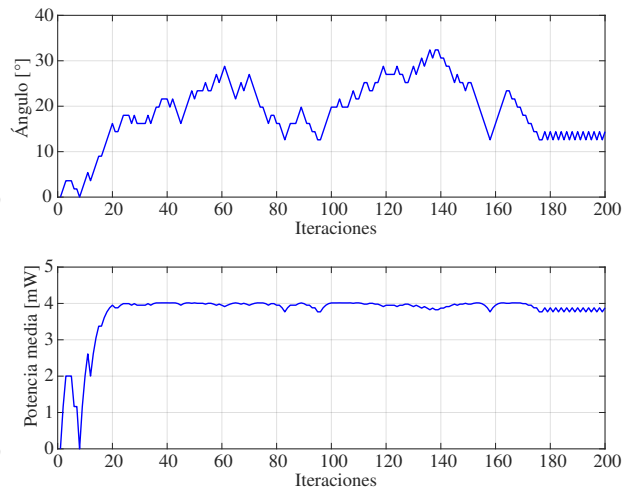


Figura 9: Resultados implementación LSPI.

Finalmente, la Figura 10 muestra el desempeño de A2C. Este algoritmo, a pesar de poseer dos redes neuronales, mostró una mayor rapidez en sus iteraciones en comparación a DQN, a raíz de la interacción de las redes actor y crítico en el proceso de ajuste de política. Su exploración del espacio de estados es más precisa, con menos variaciones antes de alcanzar una política estable, con la que consigue una potencia solo un 0.6 % por debajo del máximo determinado en la figura 6.

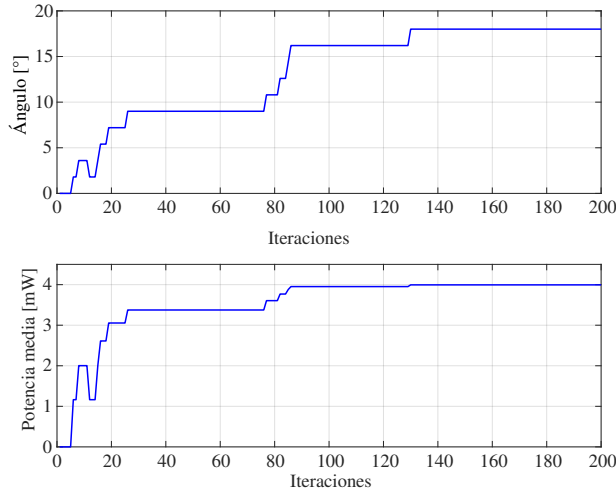


Figura 10: Resultados implementación A2C.

Como se muestra en la Tabla 3, los algoritmos off-policy, a excepción de DQN, lograron tiempos de cálculo menores respecto a los algoritmos on-policy, debido principalmente a que no requieren el entrenamiento de redes neuronales complejas. Adicionalmente, la convergencia de mostrada en los algoritmos basados en valores se logra en un menor número de iteraciones, gracias a su estrategia de exploración. Sin embargo, su rendimiento podría verse afectado en espacios de estados y acciones más grandes.

Un aspecto que pudo haber afectado el rendimiento de los algoritmos en la búsqueda de la máxima potencia es la incorporación de una función de recompensa que no permite interpretar correctamente el entorno, ajustes inadecuados de los hiperparámetros y/o una arquitectura de red ineficiente.

En general, el algoritmo de aprendizaje por refuerzo on-policy muestra mayor estabilidad y suavidad en sus iteraciones, debido a su menor dependencia de la exploración aleatoria del entorno y a sus métodos de optimización basados en parámetros de red, incluidos dentro del algoritmo.

4. Conclusiones

Los algoritmos propuestos lograron maximizar la potencia generada, alcanzando al menos el 95 % de la potencia máxima del sistema. Esto demuestra la efectividad de los métodos de aprendizaje reforzado en el control de generadores undimotrices, optimizando la potencia a través de la manipulación adecuada de las variables del sistema.

Los algoritmos off-policy lograron identificar rápidamente estados favorables debido a su estrategia de exploración aleatoria inicial. Sin embargo, estos algoritmos no logran mantenerse en un estado óptimo, lo que indica la necesidad de realizar ajustes a las variables del sistema por periodos más prolongados en entornos con condiciones cambiantes.

El algoritmo LSPI destacó por su velocidad de implementación en comparación con DQN y A2C, principalmente debido a la menor cantidad de hiperparámetros que requiere ajustar y la ausencia de redes neuronales. La simplicidad de las operaciones en Least-Squares Policy Iteration facilita la detección de problemas que podrían impactar el rendimiento del algoritmo, a diferencia de los métodos basados en redes neuronales, donde su mayor complejidad dificulta tanto el diagnóstico como la optimización de sus parámetros.

En el caso del algoritmo Deep Q-Network (DQN), no mostró una ventaja significativa en comparación con Q-learning o LSPI, y su alto costo computacional lo hace menos viable para su aplicación en estrategias de control en tiempo real.

Por otro lado, el algoritmo A2C, de tipo on-policy, mostró una convergencia más estable y suave, lo que resulta beneficioso, ya que evita cambios bruscos que podrían dañar los componentes del PTO. Aunque A2C requiere más tiempo de cálculo debido a la necesidad de una mayor cantidad de datos para entrenar sus redes neuronales adecuadamente, este incremento en el tiempo de cómputo es compensado por la robustez y fiabilidad del control obtenido.

Es importante destacar que el ajuste manual de los hiperparámetros podría haber introducido sesgos en el rendimiento de los algoritmos, limitando su máximo potencial y prolongando los tiempos de implementación debido a la necesidad de evaluar múltiples combinaciones de parámetros. La sintonización más precisa y automatizada de estos hiperparámetros sería crucial para mejorar la eficiencia de estos algoritmos.

A pesar de estas consideraciones, las estrategias de control basadas en aprendizaje por refuerzo demuestran una gran adaptabilidad a las condiciones variables del entorno marino, permitiendo maximizar la potencia generada mediante la captura de datos reales de la dinámica de los dispositivos de generación undimotriz. Esto es especialmente relevante dado el comportamiento altamente no lineal del sistema, que dificulta el control basado en modelos exactos.

Trabajos futuros en esta área son necesarios, con especial énfasis en la sintonización óptima de los hiperparámetros, la validación experimental y la implementación bajo condiciones de oleajes irregulares.

Referencias

- [1] A. G. Olabi and M. A. Abdelkareem, "Renewable energy and climate change," *Renewable and Sustainable Energy Reviews*, vol. 158, 4 2022.
- [2] Y. Zhang, Y. Zhao, W. Sun, and J. Li, "Ocean wave energy converters: Technical principle, device realization, and performance evaluation," *Renewable and Sustainable Energy Reviews*, vol. 141, 5 2021.
- [3] E. Anderlini, S. Husain, G. G. Parker, M. Abusara, and G. Thomas, "Towards real-time reinforcement learning control of a wave energy converter," *Journal of Marine Science and Engineering*, vol. 8, pp. 1–16, 11 2020.

- [4] S. A. Sirigu, L. Foglietta, G. Giorgi, M. Bonfanti, G. Cervelli, G. Bracco, and G. Mattiazzo, "Techno-economic optimisation for a wave energy converter via genetic algorithm," *Journal of Marine Science and Engineering*, vol. 8, 7 2020.
- [5] D. Curto, V. Franzitta, and A. Guercio, "Sea wave energy. a review of the current technologies and perspectives," 10 2021.
- [6] S. Zhan and J. V. Ringwood, "Model-free linear non-causal optimal control of wave energy converters via reinforcement learning," *IEEE Transactions on Control Systems Technology*, vol. PP, pp. 1–14, 2024.
- [7] E. Anderlini, D. I. Forehand, P. Stansell, Q. Xiao, and M. Abusara, "Control of a point absorber using reinforcement learning," *IEEE Transactions on Sustainable Energy*, vol. 7, pp. 1681–1690, 2016.
- [8] L. Wang, J. Isberg, and E. Tedeschi, "Review of control strategies for wave energy conversion systems and their validation: the wave-to-wire approach," 2018.
- [9] J. V. Ringwood, A. Merigaud, N. Faedo, and F. Fusco, "Wave energy control systems: Robustness issues *," vol. 51, pp. 62–67, Elsevier B.V., 1 2018.
- [10] M. Zhang, S. R. Yu, G. W. Zhao, S. S. Dai, F. He, and Z. M. Yuan, "Model predictive control of wave energy converters," *Ocean Engineering*, vol. 301, 6 2024.
- [11] D. Gallutia, M. T. Fard, M. G. Soto, and J. B. He, "Recent advances in wave energy conversion systems: From wave theory to devices and control strategies," *Ocean Engineering*, vol. 252, 5 2022.
- [12] L. Li, Z. Gao, and Z. M. Yuan, "On the sensitivity and uncertainty of wave energy conversion with an artificial neural-network-based controller," *Ocean Engineering*, vol. 183, pp. 282–293, 7 2019.
- [13] A. S. Haider, K. Bubbar, and A. McCall, "Comparison of advanced control strategies applied to a multiple-degrees-of-freedom wave energy converter: Nonlinear model predictive controller versus reinforcement learning," *Journal of Marine Science and Engineering*, vol. 11, 11 2023.
- [14] J. V. Ringwood, A. Merigaud, N. Faedo, and F. Fusco, "An analytical and numerical sensitivity and robustness analysis of wave energy control systems," *IEEE Transactions on Control Systems Technology*, vol. 28, pp. 1337–1348, 7 2020.
- [15] A. Shadmani, M. R. Nikoo, A. H. Gandomi, R. Q. Wang, and B. Golparvar, "A review of machine learning and deep learning applications in wave energy forecasting and wec optimization," *Energy Strategy Reviews*, vol. 49, 9 2023.
- [16] X. Zhu, M. Li, X. Liu, and Y. Zhang, "A backpropagation neural network-based hybrid energy recognition and management system," *Energy*, vol. 297, 6 2024.
- [17] S. Zou, X. Zhou, I. Khan, W. W. Weaver, and S. Rahman, "Optimization of the electricity generation of a wave energy converter using deep reinforcement learning," *Ocean Engineering*, vol. 244, 1 2022.
- [18] S. K. Poguluri, D. Kim, Y. Lee, J. H. Shin, and Y. H. Bae, "Design optimization of asymmetric wave energy converter using artificial neural network model," *International Journal of Naval Architecture and Ocean Engineering*, vol. 15, 1 2023.
- [19] D. Sarkar, E. Contal, N. Vayatis, and F. Dias, "Prediction and optimization of wave energy converter arrays using a machine learning approach," *Renewable Energy*, vol. 97, pp. 504–517, 2016.
- [20] L. Li, Y. Gao, D. Z. Ning, and Z. M. Yuan, "Development of a constraint non-causal wave energy control algorithm based on artificial intelligence," *Renewable and Sustainable Energy Reviews*, vol. 138, 3 2021.
- [21] F. Pierart, C. Manriquez, and P. Campos, "Reinforcement learning algorithms applied to reactive and resistive control of a wave energy converter," *2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2021*, 2021.
- [22] E. Anderlini, "Control of wave energy converters using machine learning strategies," p. 255, 2017.
- [23] E. Anderlini, D. I. Forehand, E. Bannon, and M. Abusara, "Constraints implementation in the application of reinforcement learning to the reactive control of a point absorber," *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, vol. 10, pp. 1–10, 2017.
- [24] S. Thomas, M. Giassi, M. Eriksson, M. Göteman, J. Isberg, E. Ransley, M. Hann, and J. Engström, "A model free control based on machine learning for energy converters in an array," *Big Data and Cognitive Computing*, vol. 2, pp. 1–15, 2018.
- [25] L. Bruzzzone, P. Fanghella, and G. Berselli, "Reinforcement learning control of an onshore oscillating arm wave energy converter," *Ocean Engineering*, vol. 206, 6 2020.
- [26] J. Falnes and A. Kurniawan, *Ocean waves and oscillating systems: linear interactions including wave-energy extraction*, vol. 8. Cambridge university press, 2020.
- [27] A. Maria-Arenas, A. J. Garrido, E. Rusu, and I. Garrido, "Control strategies applied to wave energy converters: State of the art," *Energies*, vol. 12, no. 16, p. 3115, 2019.
- [28] F. AlMahamid and K. Grolinger, "Reinforcement learning algorithms: An overview and classification," 9 2022.
- [29] A. K. Shakyia, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications*, vol. 231, 11 2023.
- [30] E. Anderlini, D. I. Forehand, E. Bannon, Q. Xiao, and M. Abusara, "Reactive control of a two-body point absorber using reinforcement learning," *Ocean Engineering*, vol. 148, pp. 650–658, 2018.
- [31] K. Chen, X. Huang, Z. Lin, X. Xiao, and Y. Han, "Control of a wave energy converter using model-free deep reinforcement learning," *2024 UKACC 14th International Conference on Control, CONTROL 2024*, pp. 1–6, 2024.
- [32] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *The Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.
- [33] E. Anderlini, D. I. Forehand, E. Bannon, and M. Abusara, "Control of a realistic wave energy converter model using least-squares policy iteration," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1618–1628, 2017.
- [34] R. Waters, "Energy from ocean waves," *Full Scale Experimental Verification of a Wave Energy Converter. Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology*, vol. 580, 2008.
- [35] G. Paczolay and I. Harmati, "A new advantage actor-critic algorithm for multi-agent environments," pp. 1–6, 2020.
- [36] A. Biswas, P. G. Anselma, and A. Emadi, "Real-time optimal energy management of multimode hybrid electric powertrain with online trainable asynchronous advantage actor–critic algorithm," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 2, pp. 2676–2694, 2022.
- [37] E. R. Mageli, "Reinforcement learning in process control," Master's thesis, NTNU, 2019.
- [38] F. G. Pierart, M. Rubilar, and J. Rohten, "Experimental validation of damping adjustment method with generator parameter study for wave energy conversion," *Energies*, vol. 16, no. 14, p. 5298, 2023.