Universidad Michoacana de San Nicolás de Hidalgo

Facultad de Ingeniería Eléctrica

División de Estudios de Posgrado

Aprendizaje Profundo por Refuerzo y Demostraciones Aplicado al Seguimiento del Punto Global de Máxima Potencia en Sistemas Fotovoltaicos

TESIS

que para obtener el grado de:

Doctor en Ciencias en Ingeniería Eléctrica

PRESENTA:

Baldwin Rainiero Cortés Hernández

ASESOR

Dr. Roberto Tapia Sánchez

CO-ASESOR

Dr. Juan José Flores Romero

Morelia, Mich., Octubre de 2022

APRENDIZAJE PROFUNDO POR REFUERZO Y DEMOSTRACIONES APLICADO AL SEGUIMIENTO DEL PUNTO GLOBAL DE MÁXIMA POTENCIA EN SISTEMAS FOTOVOLTAICOS

Los Miembros del Jurado de Examen de Grado aprueban la **Tesis de Doctorado en Ciencias en Ingeniería Eléctrica, Opción en Sistemas de Control** de **Baldwin Rainiero Cortés Hernández**

Dr. Norberto García Barriga
Presidente del Jurado

Dr. Roberto Tapia Sánchez
Director de Tesis

Dr. Juan José Flores Romero
Co-director

Dr. Juan Anzurez Marin
Vocal

Dra. Adriana del Carmen Téllez Aguiano
Revisor Externo (ITM)

Dr. J. Aurelio Medina Rios
Jefe de la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica. UMSNH
(Por reconocimiento de firmas)

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
Septiembre 2022

Resumen

En esta tesis se desarrolla una técnica de *seguimiento del punto global de máxima potencia* (GMPPT) basada en *aprendizaje profundo por refuerzo* (DRL) con uso de demostraciones, aplicada a sistemas *fotovoltaicos* (PV) en condiciones de *sombreado parcial* (PS).

En la actualidad, los algoritmos de DRL no han logrado consolidarse en aplicaciones reales de GMPPT. Esto se debe principalmente a que requieren miles de interacciones con el sistema, antes de obtener un desempeño satisfactorio. Por otra parte, los algoritmos de *seguimiento del punto máximo de potencia* (MPPT) clásicos, como el algoritmo *Perturba y Observa* (P&O), pueden desempeñarse razonablemente bien desde el primer momento desde su implementación. Sin embargo, los algoritmos MPPT clásicos no siempre logran ubicar el *punto global de máxima potencia* (GMPP), lo que provoca pérdidas en la potencia generada.

Esta tesis integra las modalidades de interacción de los algoritmos clásicos en el proceso de aprendizaje de los algoritmos DRL, disminuyendo el número de interacciones requeridas en el entrenamiento y promoviendo un mejor desempeño en el seguimiento. El algoritmo DRL propuesto se denomina TD4, haciendo alusión al algoritmo usado como base: *Gradiente de Política Determinista Profunda con Retraso Gemelo* (TD3) y a la inclusión de *demostraciones* (D) – TD3 + D.

La implementación y entrenamiento del algoritmo TD4 se realiza en Python, utilizando el marco de aprendizaje automático de código abierto PyTorch; mientras que el modelado y simulación del sistema fotovoltaico se realiza en MATLAB/Simulink. Para la validación, se utilizan patrones complejos no uniformes de irradiancia solar. Con el propósito de demostrar las cualidades del método GMPPT TD4 propuesto, se presenta una comparación frente a otras técnicas de seguimiento: un algoritmo MPPT P&O, un algoritmo GMPPT de *Gradiente de Política Determinista Profunda* (DDPG), y un algoritmo GMPPT TD3. Los resultados muestran que el desempeño de TD4 superior, al exhibir una mayor velocidad y eficiencia en el seguimiento, lo que se traduce directamente como un incremento en la cantidad de energía eléctrica generada por el sistema PV bajo las mismas condiciones ambientales.

La base teórica desarrollada en esta tesis abre nuevos caminos de investigación para técnicas GMPPT basadas en DRL, donde los algoritmos de aprendizaje automático pueden beneficiarse de la experiencia e información recolectada por otras técnicas MPPT/GMPPT para adaptarse a los requerimientos de aprendizaje.

Palabras clave: *sistemas fotovoltaicos, aprendizaje profundo por refuerzo, inteligencia artificial, redes neuronales, seguimiento del punto de máxima potencia.*

Abstract

In this thesis, a *global maximum power point tracking* (GMPPT) technique based on *deep reinforcement learning* (DRL) with the use of demonstrations is developed and applied to *photovoltaic* (PV) systems under *partial shading* (PS) conditions.

To date, DRL algorithms have not been able to establish themselves in real applications of GMPPT, mainly because they require thousands of interactions with the system before obtaining satisfactory performance. On the other hand, classic *maximum power point tracking* (MPPT) algorithms, such as the Perturb and Observe (P&O) algorithm, can perform reasonably well out of the box. However, classic MPPT algorithms do not always manage to locate the *global maximum power point* (GMPP), which causes losses in the generated power.

This thesis integrates the interaction modalities of classical algorithms in the learning process of DRL algorithms, reducing the number of interactions required in training and promoting better tracking performance. The proposed DRL algorithm is called TD4, referring to the algorithm used as a base: *Deep Deterministic Policy Gradient with Twin Delay* (TD3) and the inclusion of *demonstrations* (D) – TD3 + D.

The implementation and training of the TD4 algorithm are done in Python, using the open source machine learning framework PyTorch, while the modeling and simulation of the photovoltaic system are done in MATLAB/Simulink. For validation, complex non-uniform patterns of solar irradiance are used. In order to demonstrate the qualities of the proposed GMPPT TD4 method, a comparison is presented against other tracking techniques: an MPPT P&O algorithm, a GMPPT *Deep Deterministic Policy Gradient* (DDPG) algorithm, and a GMPPT TD3 algorithm. The results show that the performance of TD4 is superior, exhibiting greater speed and efficiency in tracking, which directly translates into an increase in the amount of electrical energy generated by the PV system under the same environmental conditions.

The theoretical basis developed in this thesis opens new research paths for DRL-based GMPPT techniques, where machine learning algorithms can benefit from the experience and information collected by other MPPT/GMPPT techniques to adapt to learning requirements.

Keywords: *photovoltaic systems, deep reinforcement learning, artificial intelligence, neural networks, maximum power point tracking.*

Dedicatoria

A la memoria de mi más grande amigo, Gilberto.
Gracias por acompañarme durante más de diez años,
desde que comenzamos nuestra aventura en la universidad.
Tu recuerdo me mantuvo soñando cuando quise rendirme.

Agradecimientos

A mis padres, Angélica y Jesús, por impulsar mis sueños y esperanzas, por estar siempre a mi lado en los días y noches más difíciles, por el apoyo incondicional que siempre me han mostrado. Gracias por creer en mí.

A mi hermana, Carolina, por todo su cariño y apoyo incondicional, por conducirse siempre con integridad en su profesión, por brindarme el ejemplo de un hermano mayor.

A mi novia, Brenda, por todo su amor, apoyo, y sacrificio durante el último año de mi doctorado. Gracias por ayudarme a mantener mi vida en balance y por no me permitirme nunca faltar a un entrenamiento.

A mis amigos y compañeros de viaje, Napoleón y Carlos, por todas las horas de trabajo compartidas a lo largo de nuestra formación y los innumerables gratos recuerdos que guardo de ustedes, por hacer más amena esta (no tan) fugaz experiencia.

Al Dr. Roberto Tapia Sánchez, asesor principal de esta tesis, por brindarme la oportunidad de desarrollar este trabajo bajo su dirección. A pesar de la gran cantidad de trabajo que conlleva ser Jefe de Posgrado y Director de la Facultad de Ingeniería Eléctrica de forma simultánea, siempre mostró disposición en apoyarme y guiarme en esta etapa de mi vida. Gracias por todo el conocimiento y toda la experiencia compartida, por las palabras de aliento cuando las horas de trabajo de hacían largas y la información confusa. Gracias por su orientación y sabios consejos, los llevaré grabados para siempre en la memoria en mi futuro profesional.

Al Dr. Juan José Flores Romero, co-asesor de esta tesis, por acoger cálidamente a un *controlero* dentro de su grupo de investigación en el área de Computación, y permitirme trabajar estrechamente con usted. Gracias por guiarme en el área de *Machine Learning*, por revisar con extrema paciencia mis primeros artículos en inglés, y por siempre contagiarme con esa alegría que lo caracteriza.

A los integrantes de la Mesa Sinodal, el Dr. Norberto García Barriga, el Dr. Juan Anzurez Marín y la Dra. Adriana del Carmen Téllez Anguiano, por el tiempo dedicado a la revisión de esta tesis y por sus valiosas contribuciones a la versión final del documento.

A mi alma mater, la Universidad Michoacana de San Nicolás de Hidalgo por haberme permitido formarme en sus aulas, tanto en licenciatura, maestría, y ahora en doctorado.

Finalmente, al Conacyt por la financiación de mi estudio de doctorado.

Prefacio

En el mundo, existen dos tipos de personas: los optimistas y los pesimistas. Los pesimistas buscan mantener su *status quo* , los optimistas intentan transformarlo.

El optimismo a menudo requiere creer en lo desconocido, en futuros inciertos y fantasiosos; el pesimismo se apega a lo conocido, lo que es prudente y ha sido probado. Un optimista es ingenuo; un pesimista, sabio.

Las soluciones conocidas no pueden resolver los problemas difíciles. Por eso son difíciles, porque requieren disrumpir los pensamientos tradicionales y las formas que han funcionado. Pero algo que ha funcionado bien en el pasado, no significa que no pueda funcionar mejor en el futuro, o incluso, que siquiera pueda seguir funcionando.

Centrarse solo en los sectores establecidos y campos probados, naturalmente, conduce al pesimismo. Para ser optimista, se debe creer que al menos alguna de las especulaciones actuales se harán realidad. ¿Quién hubiera creído que se podían inventar aeroplanos, comunicaciones inalámbricas y televisores?

Una de las principales cuestiones en la actualidad es la de la inteligencia artificial. ¿Puede una máquina exhibir comportamientos inteligentes? ¿Puede una máquina ser capaz de entender el mundo? ¿Puede una máquina pensar? La respuesta más sensata es no. La respuesta optimista, sí.

Si el progreso conseguido en los últimos años fuera meramente una casualidad, éste está destinado a concluir. En cambio, si el progreso es primordialmente una cuestión de esfuerzo humano, depende de nosotros su continuidad.

— Jason Crawford

Contenido

Figuras

Tablas

Algoritmos

Listados

Nomenclatura

α	Factor de aprendizaje del agente en el marco de RL
η	Eficiencia de seguimiento de la técnica MPPT/GMPPT
γ	Factor de descuento del MDP
$\mathbf{k}_B$	Constante de Boltzmann ($\approx 1.38064852 \times 10^{-23}$ J/K)
$\mathbf{T}$	Longitud del episodio en el marco de RL/DRL
$\mathcal{D}$	Conjunto de demostraciones expertas en el marco BC/TD4
$\mathcal{G}$	Recompensa total acumulada en una trayectoria
$\mathcal{L}(\cdot)$	Función de costo
$\mathcal{M}$	MDP
$\mathcal{N}$	Proceso de ruido de regularización en el marco de TD3/TD4
$\mathcal{N}_t$	Proceso de ruido de exploración en el marco de TD3/TD4
$\mathcal{R}$	Búfer de reproducción donde se almacena la experiencia del agente
$\mathcal{T}$	Función de transición del MDP
μ_I	Coeficiente de corriente/temperatura de la celda PV
$\mu(s, \cdot)$	ANN que representa el actor en el marco de DRL
$\mu'(s, \cdot)$	Red de objetivo utilizada por el actor en el marco de DRL
ϕ	Parámetros del crítico en el marco de RL
π	Política del agente en el marco de RL/DRL
π^*	Política óptima
ψ	Pasos de retardo para actualizar la política en el marco de TD3/TD4
ρ	Factor de promedio de Polyak
θ	Parámetros del actor en el marco de RL
θ^μ	Parámetros del actor en el marco de DRL
θ^Q	Parámetros del crítico en el marco de DRL
$\tilde{a}$	Acción con ruido en el marco de TD3/TD4
A	Espacio de acción del MDP

a'	Acción equivalente a a_{t+1}
a_t	Acción determinada por el agente en el tiempo t
D	Ciclo de trabajo del convertidor DC-DC
E_g	Energía de la banda prohibida del semiconductor (1.2 eV para el silicio)
G	Irradiancia solar
$H(\pi)$	Entropía de la política
I_0	Corriente de saturación inversa de la celda PV
I_{mpp}	Corriente de la celda/arreglo PV donde se produce la mayor potencia
I_{ph}	Corriente generada en una celda PV debido al efecto fotovoltaico
I_{pv}	Corriente de la celda/arreglo PV
I_{sc}	Corriente de cortocircuito de la celda/arreglo PV
$J(\cdot)$	Función de desempeño
k	Iteración de entrenamiento en el marco de RL/DRL
M	Número de episodios de entrenamiento en el marco de RL/DRL
N	Tamaño del mini lote de experiencias para aplicar optimización por gradiente
N_p	Número de celdas PV en paralelo
N_s	Número de celdas PV en serie
P_{MPPT}	Potencia del sistema PV utilizando una técnica MPPT/GMPPT
P_{max}	Potencia máxima de la celda/arreglo PV
q	Carga del electrón ($\approx 1.60217662 \times 10^{-19}$ C)
$Q(s,a)$	Función de valor de estado-acción en el marco de RL
$Q(s,a,\cdot)$	ANN que representa el crítico en el marco de DRL
$Q^*(s,a)$	Función de valor de estado-acción óptima en el marco de RL
$Q'(s,a,\cdot)$	Red de objetivo utilizada por el crítico en el marco de DRL
R	Función de recompensa del MDP
R_L	Resistencia de carga
R_p	Resistencia en paralelo de la celda PV
R_s	Resistencia en serie de la celda PV
r_{t+1}	Recompensa obtenida por el agente al evaluar a_t en s_t y pasar a s_{t+1}
S	Espacio de estado del MDP
s'	Estado equivalente a s_{t+1}
s_t	Estado del entorno en el tiempo t
s_{t+1}	Estado del entorno en el tiempo $t+1$, al evaluar a_t en s_t
T	Temperatura de la celda PV

T_ϵ	Número de pasos de desvanecimiento de ruido de exploración en el marco de TD4
T_a	Temperatura ambiente
$V(s)$	Función de valor de estado en el marco de RL
$V^*(s)$	Función de valor de estado óptima en el marco de RL
V_{mpp}	Voltaje de la celda/arreglo PV donde se produce la mayor potencia
V_{oc}	Voltaje de circuito abierto de la celda/arreglo PV
V_{pv}	Voltaje de la celda/arreglo PV
Adam	Estimación adaptativa del momento
AI	Inteligencia Artificial
BC	Clonación Conductual
DDPG	Gradiente de Política Determinista Profunda
DL	Aprendizaje Profundo
DRL	Aprendizaje Profundo por Refuerzo
GMPP	Punto Global de Máxima Potencia
GMPPT	Seguimiento del Punto Global de Máxima Potencia
i.i.d.	Independientes e idénticamente distribuidos
LMPP	Punto Local de Máxima Potencia
MDP	Proceso de Decisión de Markov
MHO	Metaheurísticas de Optimización
ML	Aprendizaje de Máquina
MLP	Perceptrón multicapa
MPP	Punto de Máxima Potencia
MPPT	Seguimiento del Punto de Máxima Potencia
MSE	Error cuadrático medio
P&O	Perturba y Observa
PCE	Eficiencia de Conversión de Potencia
PS	Sombreado Parcial
PV	Fotovoltaico
ReLU	Unidad Lineal Rectificada
RL	Aprendizaje por Refuerzo
SGD	Descenso de gradiente estocástico
TanH	Tangente hiperbólica
TD3	Gradiente de Política Determinista Profunda con Retraso Gemelo
TD4	Gradiente de Política Determinista Profunda con Retraso Gemelo y Demostraciones

Capítulo 1

Introducción

Este capítulo presenta una visión general de la problemática energética mundial en la actualidad, se detallan los inconvenientes de la presente generación de energía eléctrica no renovable, así como las ventajas de la generación renovable, especialmente de la generación fotovoltaica. Respecto a este tipo de generación, se describen las técnicas de control utilizadas actualmente y sus limitantes. Posteriormente, se describe brevemente la historia de la inteligencia artificial y se discute su aplicación en el control de los sistemas fotovoltaicos. Finalmente, se estipulan los objetivos establecidos en la investigación y se especifican las aportaciones y publicaciones derivadas de la misma.

1.1. Justificación

La demanda de energía crece continuamente debido a la explosión demográfica y al desarrollo económico (Chang et al., 2019), y se estima que en el año 2050 la demanda de energía será dos veces mayor que la actual (International Energy Agency, 2021). Además de satisfacer estos requerimientos energéticos, otro reto importante que enfrenta el sector de la generación es atenuar la sobreexplotación de los recursos no renovables para la generación de energía, ya que alrededor del 70 % de la energía mundial en la actualidad se genera a partir de fuentes no renovables (Solar Power Europe, 2021). Este tipo de generación presenta dos inconvenientes principales: la posible escasez de los combustibles y la emisión de gases contaminantes (Harrag & Messalti, 2019). Para mitigar estos problemas, una gran cantidad de países han establecido compromisos para impulsar la adopción de fuentes renovables de energía (Cheng & Yao, 2021; Khan et al., 2020). Así, se espera que en los próximos 20 años, la cantidad de energía producida por fuentes renovables se incremente un 75 % respecto a la generación actual (U.S. Energy Information Administration, 2018), reduciendo a la par la aportación de la generación por parte de las fuentes no renovables.

Entre las fuentes de energía renovables disponibles, la generación *fotovoltaica* (PV, *photovoltaic*) se ha convertido en la opción más popular debido a sus ventajas, como ausencia de combustible, mantenimiento infrecuente, costo asequible, entre otras (Veerachary et al., 2001). Esta popularidad ha impulsado su desarrollo, dado que la aportación

de la generación fotovoltaica al total mundial creció de prácticamente 0 % en el año 2000, a 3.72 % en el año 2021 (Ember Climate, 2022); y se prevé que para el año 2050, la generación PV produzca el 50 % de la energía total proveniente de fuentes renovables, el equivalente al 8 % de la generación total en el mundo (U.S. Energy Information Administration, 2018, 2019).

No obstante, la eficiencia de conversión de las celdas fotovoltaicas de silicio (el material más rentable para su producción) es del 22 % en promedio (Ciulla et al., 2014); y se estima que la eficiencia teórica máxima de este material está cercana al 27 % (Sinke, 2019); es decir, la tecnología PV actual está cerca de su límite de eficiencia de conversión. Aunque se han explorado nuevas estrategias para incrementar la eficiencia, tales como la inclusión de otros materiales semiconductores, el revestimiento de las capas frontales y traseras, sistemas de concentración de la energía solar, entre otras; éstas resultan costosas o inviables de producir en masa con la tecnología actual (Al-Shahri et al., 2021). Por lo tanto, la única alternativa plausible para mejorar la eficiencia de generación de los sistemas PV actuales es optimizar sus técnicas de control, lo que favorecería simultáneamente a todos los sistemas PV, independientemente de su tecnología y sus características añadidas.

En este sentido, los avances en el área del *Aprendizaje Profundo por Refuerzo* (DRL, *Deep Reinforcement Learning*) son prometedores. Recientemente, diversos algoritmos de DRL han solucionado problemas que se consideraban utópicos, tales como el desempeño sobre-humano en distintos videojuegos (Arulkumaran et al., 2017; Mnih et al., 2015), la conducción autónoma de vehículos (Bojarski et al., 2016), y el auto-aprendizaje de movimientos de locomoción en robots a partir únicamente de secuencias de imágenes obtenidas por sensores instalados en la propia estructura de los robots (Levine et al., 2016). Así, en esta tesis se aborda la aplicación de los conceptos de DRL en el área de sistemas PV para optimizar el control de los sistemas PV, y por tanto, incrementar su eficiencia de generación.

1.2. Estado del Arte

1.2.1. La Energía Eléctrica en la Actualidad

A partir de la invención e implementación de la red eléctrica a comienzos del siglo XX, la energía eléctrica se ha convertido en un factor determinante para el desarrollo de las sociedades y economías alrededor del mundo (Schewe, 2007). Prácticamente todas las actividades cotidianas requieren el uso de energía eléctrica: la iluminación residencial, la conservación y procesamiento de los alimentos, el almacenamiento y transmisión de datos a través de medios digitales, etc. Inclusive, algunos sectores como el de las telecomunicaciones y tecnologías de la información, dependen plenamente de la disponibilidad de energía eléctrica para su operatividad (Chochliouros et al., 2021).

La dependencia del estilo de vida moderno a la energía eléctrica ha incrementado considerablemente su demanda: en los últimos 50 años, el uso de energía eléctrica se ha visto incrementado casi un 500 %. En 1974, la demanda mundial de energía eléctrica ascendía a 5,000 TWh (Kelly et al., 2020), mientras que en el 2020, fue de 23,000

TWh. Incluso, existen pronósticos que señalan que la demanda mundial de electricidad en el año 2050 rondará los 45,000 TWh, el doble de la demanda actual (International Energy Agency, 2021). El continuo incremento del consumo de energía eléctrica es ocasionado simultáneamente por dos factores: el aumento en el consumo per cápita de energía eléctrica y el continuo crecimiento de la población mundial (Chang et al., 2019). En referencia al primero de ellos, la demanda mundial promedio de energía eléctrica per cápita por año aumentó de 2.3 kWh a 2.39 kWh, del año 2010 al 2019 (BP, 2021); referente al segundo factor, la población mundial pasó de 7,000 millones, en 2010, a 7,800 millones, en 2020; y se prevé que podría alcanzar 10,100 millones en el año 2050, y 12,700 millones en el año 2100 (Gu et al., 2021).

Además de garantizar la disponibilidad de energía eléctrica para satisfacer la creciente demanda, el sector de la generación debe afrontar otra problemática: detener el uso excesivo de recursos no renovables para la producción de electricidad, en vista de que en el año 2020, el 70 % de la energía eléctrica mundial se generó a partir de fuentes no renovables (Solar Power Europe, 2021). Este tipo de generación, particularmente la basada en la ignición de combustibles fósiles (e.g., petróleo y carbón), contribuye a la degradación ambiental por medio de diversos factores, de entre los cuales se destaca la emisión de *dióxido de carbono* (CO_2) y otros *gases de efecto invernadero* (GHG, *greenhouse gases*) (Khan et al., 2021). La acumulación excesiva de estos gases en la atmósfera afecta a todos los organismos del planeta a través de múltiples procesos, incluyendo el incremento en la temperatura global, fenómenos meteorológicos extremos (e.g., sequías e inundaciones), y el aumento de la contaminación en el aire (Evans, 2019; Mikkelsen et al., 2008; Weissburg & Draper, 2019).

Para afrontar el problema de la emisión de GHG, muchos países y organizaciones han establecido compromisos de protección ambiental bajo diferentes tratados internacionales (Shishlov et al., 2016). El Acuerdo de París, el más reciente de estos tratados, exhorta a los países a desarrollar estrategias de desarrollo a largo plazo para disminuir las emisiones de GHG con el objetivo final de limitar el incremento de la temperatura global a 2°C por encima de niveles pre-industriales, el cual fue firmado por 191 partes[1] en 2015 (Jacquet & Jamieson, 2016). Entre las recomendaciones estipuladas en el tratado, destaca el impulso a los proyectos de investigación referentes a energías renovables a través incentivos económicos y subsidios (Cheng & Yao, 2021; Khan et al., 2020). Satisfactoriamente, estos estudios demuestran que el uso de fuentes renovables de energía como alternativa al uso de combustibles fósiles disminuye las emisiones de GHG, por lo que su adopción se ha visto notablemente favorecida (Bilgili et al., 2016; Koengkan et al., 2021; Zafar et al., 2019). Así, se espera que en 20 años la cantidad de energía producida por fuentes renovables se incremente un 75 % respecto a la generación renovable actual (U.S. Energy Information Administration, 2018).

[1] 190 países más la Unión Europea.

1.2.2. La Generación Fotovoltaica

De entre las fuentes renovables disponibles, la generación PV y la generación eólica se han convertido en las más populares en las últimas décadas, en gran medida a su facilidad de escalamiento (Solar Power Europe, 2021). La capacidad de generación PV instalada durante el 2020 corresponde al 39 % de la capacidad total instalada en este año (incluyendo fuentes renovables y no renovables), mientras que el 33 % corresponde a la generación eólica (International Energy Agency, 2020). No obstante, la construcción y operación de plantas eólicas impacta en mayor medida a la flora y fauna locales. Respecto a la fauna, la instalación de aerogeneradores afecta principalmente a aves y murciélagos, obstaculizando su movimiento e incluso provocando colisiones fatales (Bellebaum et al., 2013; May et al., 2021). Respecto a la flora, un estudio reciente señala que las vibraciones producidas por la operación de las turbinas eólicas puede reducir la población de lombrices en el área, las cuales participan en procesos importantes como la filtración del agua y el reciclado de nutrientes, lo que a su vez afecta el desarrollo de la vegetación (Velilla et al., 2021).

Además de un menor impacto ambiental, la generación PV tiene otras ventajas respecto a la generación eólica y demás fuentes renovables de generación:

1. El recurso solar está disponible prácticamente en cualquier parte del planeta.
2. Los sistemas PV no tienen componentes móviles, por lo que no producen ruido (Rabaia et al., 2021).
3. Los costos relacionados con la operación y mantenimiento de los sistemas PV son mucho menores que el de las turbinas eólicas (Chang & Starcher, 2019).
4. El costo de los sistemas de generación PV disminuye continuamente. Desde el año 2018 es la fuente de producción de energía eléctrica más barata, incluso por debajo de la generación eólica (Solar Power Europe, 2021).
5. La energía solar abarca un amplio espectro de aplicación en los sistemas de potencia, desde sistemas residenciales hasta sistemas de producción a gran escala, incluidas aplicaciones de recolección de energía (*energy harvesting*) (Imran et al., 2020), sistemas aislados (i.e., no conectados a la red de suministro) para electrificación rural (Stojanovski et al., 2017), e incluso aplicaciones en vehículos eléctricos (Mathijsen, 2021).
6. El *tiempo de recuperación de energía*[2] de los sistemas PV es de 13 meses en promedio (Fthenakis & Leccisi, 2021), mientras que el de los sistemas eólicos es de 18 meses (Gao et al., 2019).
7. Los sistemas de generación PV pueden planearse e implementarse con mayor rapidez que los otros sistemas de generación (Barrueto Guzmán et al., 2018).

[2]Definido como el tiempo en el que un sistema genera la misma cantidad de energía que fue utilizada para su producción.

Estas características han convertido a la generación PV en la alternativa más amigable con el ambiente, lo que la ha posicionado como la fuente de generación con más rápido crecimiento en los últimos años (Levenda et al., 2021). De hecho, en el año 2010, el aporte de la generación PV respecto a la energía mundial generada por fuentes renovables fue prácticamente 0 %; mientras que en 2020, la generación PV aportó el 12 % del total de la energía renovable, y se espera que para el año 2050, la generación PV aporte el 50 % de energía entre las fuentes renovables (U.S. Energy Information Administration, 2018, 2019). Si bien las estadísticas de la generación PV entre las fuentes renovables son alentadoras, la comparación con la generación mundial neta es poco más que desoladora: en el año 2020, la aportación de la generación PV a la demanda mundial fue del 3.1 % (Solar Power Europe, 2021). Esto destaca la importancia de continuar con la expansión de los sistemas PV y perfeccionar continuamente su operación.

1.2.3. La Tarea de MPPT

La modesta contribución de la generación PV a la demanda mundial se debe, en gran proporción, a la baja *eficiencia de conversión de potencia* (PCE, *Power Conversion Efficiency*) solar a eléctrica de la tecnología existente. En la actualidad, alrededor del 90 % de los sistemas PV instalados están fabricados de silicio (Raza & Ahmad, 2022; Sutherland, 2020). Este semiconductor presenta una PCE promedio de entre el 20 % y el 22 % (Ciulla et al., 2014; Rühle, 2016), y máxima de 25 % en prototipos de laboratorio (Kato et al., 2019). Aunque algunos estudios muestran que otros materiales semiconductores (e.g., perovskita y arseniuro de galio) permiten alcanzar eficiencias de entre el 32 % (Essig et al., 2017) y el 39 % (France et al., 2022), el costo asociado con su producción es mayor con respecto al costo de producción del silicio; lo cual no es económicamente viable para uso comercial, por lo que estas nuevas tecnologías PV se destinan únicamente a aplicaciones aéreas y espaciales.

Otras estrategias propuestas para aumentar la PCE de las celdas PV han sido estudiadas en años recientes, tales como la adición de compuestos sobre la superficie de los paneles, con el objetivo de optimizar la absorción de la energía solar útil (Kim et al., 2022); la instalación de superficies reflectantes en la parte posterior de los paneles, para permitir que los fotones permanezcan más tiempo dentro de las celdas, al mismo tiempo que disipan el calor de los paneles (Ahmed et al., 2021); la implementación de colectores solares, con el objetivo de concentrar la energía solar difusa (Paul & Smyth, 2020); el montaje de seguidores solares, los cuales ajustan la inclinación de los paneles de tal forma que la luz incidente sea siempre perpendicular (Fernández-Ahumada et al., 2020). Sin embargo, todas estas requieren realizar alguna modificación física al sistema, lo cual puede resultar poco apropiado para algunas aplicaciones.

Por lo tanto, el desafío inmediato a resolver es, maximizar la cantidad de energía generada por los sistemas PV ya implementados, independientemente del material con el que estén fabricados y de las características adicionales que posean. Así, se favorecería tanto a los sistemas actuales, como a los sistemas en vías de ser instalados. En este sentido, la solución consiste en optimizar las técnicas de control utilizadas en ellos.

Los sistemas PV exhiben un único punto de operación (i.e., voltaje y corriente en terminales del sistema) en el cual transfieren la máxima cantidad de potencia a la carga. Este punto, conocido como *punto de máxima potencia* (MPP, *Maximum Power Point*), depende de las condiciones ambientales en cada momento, particularmente de la irradiancia solar y la temperatura. Operar el sistema de manera continua en el MPP es una labor desafiante, ya que las condiciones ambientales cambian constante e imprevisiblemente a lo largo del día, por lo que el algoritmo de control debe ajustar repetidamente el voltaje del sistema PV para cambiar su punto de operación. Esta tarea, la cual recibe el nombre de *seguimiento del punto de máxima potencia* (MPPT, *Maximum Power Point Tracking*), es uno de los temas de investigación más populares hoy en día (Karami et al., 2017), dado que el desarrollo de nuevas estrategias de control que eleven la eficiencia del MPPT, incrementando su velocidad de respuesta y mejorando la exactitud de las acciones de control, es una forma directa de optimizar la generación fotovoltaica (Harrag & Messalti, 2019).

Aunado a la variabilidad de las condiciones ambientales, un fenómeno que dificulta aún más la tarea de MPPT es el denominado *sombreado parcial* (PS, *Partial Shading*). Este fenómeno es producido por la proyección de sombras en el sistema PV a causa de nubes, árboles, edificios y otras estructuras que impiden el paso directo de la luz solar. En circunstancias de PS, el sistema PV exhibe múltiples MPP para cada condición ambiental, en lugar del habitual MPP único, donde solo uno de ellos corresponde con el *punto de máxima potencia global* (GMPP, *Global Maximum Power Point*) (Mohapatra et al., 2017). Dada la complejidad de la tarea de *seguimiento del punto de máxima potencia global* (GMPPT, *Global Maximum Power Point Tracking*) en condiciones de PS, muchos algoritmos son incapaces de ubicarlo correctamente. Se estima que el fenómeno de PS provoca una pérdida de energía anual cercana al 10 %, al margen del efecto de la atenuación de la irradiancia solar directa; es decir, por incapacidad del algoritmo MPPT/GMPPT de ubicar el GMPP (Hanson et al., 2014).

Debido a la importancia y complejidad de la tarea del MPPT, numerosos estudios se han realizado en los últimos años, y una gran cantidad de técnicas MPPT/GMPPT han sido propuestas. Estas técnicas pueden agruparse en tres diferentes categorías: técnicas clásicas, técnicas de *inteligencia artificial* (AI, *Artificial Intelligence*), y técnicas *metaheurísticas de optimización* (MHO, *Meta Heuristic Optimization*) (Bollipo et al., 2020). Dichas técnicas pueden presentar una o varias de las siguientes desventajas: oscilación en estado estable, velocidad lenta de seguimiento, dificultad de implementación, cantidad de sensores necesarios, tiempos extensos de ejecución, restricción de las condiciones de operación, incapacidad de ubicar el GMPP (Motahhir et al., 2020). Las características de cada una de ellas pueden hacer que resulten más adecuadas, o menos, para una aplicación determinada.

No obstante, las técnicas de AI, especialmente las basadas en DRL han mostrado un desempeño superior al resto, mitigando a la vez las pérdidas por PS (Zhang et al., 2019), por lo que en este trabajo se exploran las áreas de oportunidad en dichas técnicas.

1.2.4. Inteligencia Artificial: Aprendizaje Profundo y por Refuerzo

La AI fue descrita por primera vez en 1950 por Alan Turing como "el uso de computadoras para simular el comportamiento inteligente y el pensamiento crítico" (Ramesh et al., 2004), aunque el término fue acuñado hasta 1956 por John McCarthy, quien describió la AI como "la ciencia de fabricar máquinas inteligentes" (Amisha et al., 2019). Si bien la AI comenzó como un conjunto de reglas *Si – Entonces*, durante las últimas décadas se han desarrollado algoritmos complejos que se desempeñan al nivel de un cerebro humano (Kaul et al., 2020; Luchini et al., 2022).

En este sentido, los avances en el área del *Aprendizaje Profundo* (DL, *Deep Learning*), un subconjunto de la AI, ha permitido alcanzar desempeños sobresalientes en distintos sectores como el de la visión computacional, especialmente para clasificación (Krizhevsky et al., 2012) y generación de imágenes (Goodfellow et al., 2014); y el sector del procesamiento de lenguaje natural, en aplicaciones como análisis de sentimientos (Young et al., 2018), generación de secuencias de texto (Lopez & Kalita, 2017), y reconocimiento del habla (Graves et al., 2013).

Por su parte, en el área de *Aprendizaje por Refuerzo* (RL, *Reinforcement Learning*) se han propuesto soluciones a problemas como predicción de series de tiempo y análisis de mercado (Mosavi et al., 2020), en economía; aprendizaje autónomo del juego de tenis de mesa por un brazo robótico simulado, en robótica (Peters et al., 2010); y control de un péndulo invertido, en el área de control automático (Deisenroth & Rasmussen, 2011). No obstante, estos algoritmos requieren modelos matemáticos complejos, de ahí que RL estuvo confinado al área de la investigación desde su surgimiento en 1960 (Sutton & Barto, 2018).

A partir del año 2015, la combinación de las áreas de DL y RL permitieron crear un nuevo paradigma dentro del área de AI: el aprendizaje profundo por refuerzo (François-Lavet et al., 2018). Los logros obtenidos en DRL van desde un desempeño sobre-humano en videojuegos (Mnih et al., 2015; Raiman et al., 2019) y el juego de mesa Go (Silver et al., 2016, 2017), hasta la conducción autónoma de automóviles (Bojarski et al., 2016) y cuadricópteros (Giusti et al., 2016). Estos importantes avances han atraído gran cantidad de atención, lo que ha fomentado su desarrollo y adopción en distintas áreas, particularmente en el área de control y el área de la robótica (Henderson et al., 2018).

El objetivo de los algoritmos de RL/DRL es habilitar a un *agente* (i.e., controlador) para realizar de manera óptima (o cuasi-óptima) alguna tarea mediante la continua interacción con el *entorno* (i.e., sistema). Los algoritmos RL utilizan solamente entradas sensoriales, denominadas *estado*, y una señal de retroalimentación escalar que evalúa su desempeño, denominada *recompensa*. Estos algoritmos no requieren el conocimiento de un modelo del sistema, lo cual les brinda una ventaja considerable frente a otras técnicas de control basadas en el conocimiento detallado del modelo matemático (Kiumarsi et al., 2018). Los algoritmos RL/DRL presentan un desempeño similar al de técnicas de control óptimo como el *Regulador Cuadrático Lineal* (LQR) (Rizvi & Lin, 2020) y *Modelo de Control Predictivo* (MPC) (Ernst et al., 2009; Lin et al., 2021b).

No obstante, los métodos de RL/DRL se basan exclusivamente en la experiencia recopilada por el agente al interactuar con el entorno. En la fase inicial del aprendizaje, las interacciones tienen como finalidad explorar

ampliamente el espacio de estado (i.e., puntos de operación del sistema) con el objetivo de distinguir las acciones óptimas de las sub-óptimas. En esta fase, denominada *fase de exploración*, el desempeño del controlador es deficiente debido a que la finalidad es obtener el mayor conocimiento posible del sistema, por lo que las acciones elegidas no necesariamente son las óptimas. Este problema ha sido ampliamente estudiado, y en la actualidad es considerado un problema abierto (Yang et al., 2021).

Una de las soluciones propuestas para minimizar la pérdida de rendimiento en la fase de exploración es la inclusión de demostraciones en el proceso de entrenamiento del agente (Vecerik et al., 2018). Estas demostraciones se obtienen previamente, al observar el comportamiento de otro agente, denominado *agente experto*, en el mismo entorno. Una premisa importante en este paradigma es el desempeño del agente experto, el cual se considera tiene un comportamiento cercano al óptimo (Nair et al., 2018). En este sentido, se explora la incorporación de demostraciones de un algoritmo MPPT *Perturba y Observa* (P&O, *Perturb and Observe*) al proceso de entrenamiento de una técnica MPPT basada en DRL, con el objetivo de mejorar el desempeño del agente en la fase de exploración.

1.2.5. Trabajos Relacionados

Con el propósito de introducir primeramente los conceptos y definiciones relacionados con el área del Aprendizaje por Refuerzo y el Aprendizaje Profundo por Refuerzo, los trabajos relacionados con esta tesis se detallan en la Sección 4.2, en conjunto con la descripción del caso de estudio.

1.3. Objetivo

Desarrollar un método de seguimiento del punto global de máxima potencia bajo condiciones de sombreado parcial, basado en el paradigma de Aprendizaje Profundo por Refuerzo, que se beneficie del desempeño de otras técnicas de seguimiento, con la finalidad de acelerar la convergencia durante el aprendizaje y mejorar el desempeño en la fase de prueba.

1.4. Hipótesis

El desarrollo de esta tesis está basado en la siguiente hipótesis: la inclusión de demostraciones sub-óptimas, provenientes de un algoritmo MPPT P&O convencional, en el entrenamiento de un algoritmo GMPPT basado en DRL mejoran su eficiencia de seguimiento.

1.5. Metodología

El enfoque de esta tesis es experimental (mediante simulación) y se desarrolla con base en la siguiente metodología:

- **Identificación del problema**: los sistemas fotovoltaicos presentan una baja eficiencia de generación en condiciones de sombreado parcial por lo que se busca desarrollar nuevas estrategias de control que permitan mejorar su desempeño en estas condiciones. Recientemente, el Aprendizaje por Refuerzo ha mostrado avances significativos en el área de control, por esta razón se investiga su aplicación en el área de sistemas PV.
- **Revisión bibliográfica del estado del arte**: se realizó una extensa revisión de las técnicas MPPT y de los algoritmos DRL más novedosos para identificar una contribución que pueda expandir el estado del arte en dichas áreas.
- **Formulación de la hipótesis**: una vez detectada el área de oportunidad, se formuló la hipótesis de la investigación.
- **Establecimiento del objetivo**: se establece el problema a resolver para delimitar el alcance de la tesis.
- **Estudio del modelado de sistemas PV**: se realiza una revisión de los modelos fotovoltaicos y sus características de operación bajo condiciones uniformes de irradiancia y de sombreado parcial.
- **Implementación de los algoritmos DRL**: para la implementación de los algoritmos se elige el lenguaje de programación Python, utilizando la biblioteca de aprendizaje automático PyTorch.
- **Diseño e implementación del caso de estudio**: el sistema PV estudiado se implementa en MATLAB/Simulink, debido a la gran relevancia y aceptación que este software tiene en la comunidad científica, especialmente en el área de control. En esta etapa también se desarrolla la interfaz de comunicación con Python, donde se ejecuta el algoritmo de aprendizaje.
- **Ejecución de los experimentos y recopilación de los datos**: debido a la naturaleza estocástica del entrenamiento de los algoritmos de aprendizaje por refuerzo, es necesario ejecutar varios experimentos independientes, de tal manera que los resultados sean estadísticamente significativos. Durante cada experimento se recolecta la eficiencia de seguimiento a lo largo del día. Para comparar distintos algoritmos de MPPT se utilizan las mismas condiciones ambientales, es decir, los algoritmos se prueban en igualdad de condiciones.
- **Análisis de los resultados**: se realiza un análisis estadístico de los datos recolectados, utilizando la media aritmética y la desviación estándar de la eficiencia de seguimiento de cada algoritmo MPPT a lo largo de un día.

1.6. Contribuciones Científicas

La principal contribución científica de esta tesis es la formulación de un algoritmo DRL que integra un conjunto de demostraciones sub-óptimas (i.e., las demostraciones contienen errores) en el entrenamiento de un algoritmo de

Gradiente de Política Determinista Profunda con Retraso Gemelo (TD3, *Twin-Delayed Deep Deterministic Policy Gradient*). El objetivo de incluir las demostraciones es guiar el proceso de entrenamiento para explorar más eficientemente el espacio de estado del sistema, lo que permite al algoritmo una mayor capacidad de generalización una vez terminado el entrenamiento, y por tanto, logrando un mejor desempeño en la tarea propuesta. Para limitar el aprendizaje de comportamientos sub-óptimos, heredados de las demostraciones sub-óptimas, el algoritmo implementa un filtro de acción, el cual discrimina las demostraciones, utilizando únicamente aquellas que considere beneficiosas de acuerdo a la tarea establecida, y desechando las que muestren un comportamiento errático. Esta configuración novedosa se ha denominado algoritmo de *Gradiente de Política Determinista Profunda con Retraso Gemelo y Demostraciones* (TD4, Twin-Delayed Deep Deterministic Policy Gradient with Demonstrations).

Para comprobar la capacidad de aprendizaje de el algoritmo TD4 propuesto, se ha diseñado un caso de estudio donde se resuelve la tarea de GMPPT en un sistema PV. En la configuración propuesta, las demostraciones sub-óptimas son proporcionadas por un algoritmo MPPT P&O, el cual es conocido por no tener un buen desempeño en tareas de GMPPT en condiciones de sombreado parcial. La simulación del sistema PV y del algoritmo TD4 se realiza utilizando conjuntamente MATLAB/SIMULINK y Python/PyTorch, respectivamente. Nuevamente, en la literatura, este modo de simulación en conjunto no ha sido estudiado, por lo que esta tesis puede abrir una vía prometedora para futuros desarrollos no solo en el área de MPPT, sino en la simulación y control de sistemas dinámicos complejos.

1.6.1. Publicaciones derivadas

Los trabajos publicados en revistas científicas y congresos internacionales que contribuyeron al desarrollo de esta tesis se enlistan a continuación:

- Cortés, B., Sánchez, R. T., & Flores, J. J. (2020). Characterization of a Polycrystalline Photovoltaic Cell Using Artificial Neural Networks. Solar Energy, 196, 157-167.
- Cortés, B., Tapia, R., & Flores, J. J. (2021). System-Independent Irradiance Sensorless ANN-Based MPPT for Photovoltaic Systems in Electric Vehicles. Energies, 14(16), 4820.
- Cortés, B., Tapia, R., & Flores, J. J. (2021, November). A Behavioral Cloning based MPPT for Photovoltaic Systems: Learning Through P&O Demonstrations. In 2021 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (Vol. 5, pp. 1-6). IEEE.

1.7. Esquema general

La tesis está organizada en cinco capítulos cuyo contenido se describe brevemente a continuación.

Capítulo 1

Este capítulo introductorio presenta una descripción general de los antecedentes, así como la motivación y los objetivos de esta tesis. Se presenta una breve introducción a la tecnología PV, su historia y cómo el aumento de las emisiones de GHG han hecho que los sistemas PV se hayan establecido como la principal fuente de energía renovable en la actualidad. Además, este capítulo describe la evolución de los sistemas de AI y detalla sus principales hallazgos.

Capítulo 2

El Capítulo 2 expone el modelo matemático utilizado para la simulación de los sistemas PV, y se detallan sus principales componentes, tales como el convertidor electrónico de potencia DC-DC y los arreglos de celdas PV. Además, en este capítulo, se analiza el efecto de la irradiancia solar y la temperatura en las características eléctricas de los sistemas PV, y se explica la tarea de MPPT en condiciones de irradiancia solar uniforme y PS.

Capítulo 3

El Capítulo 3 describe los conceptos básicos en el paradigma de RL/DRL, tales como el agente, la recompensa y el entorno. Además, en este capítulo, se describen brevemente los tipos de algoritmos de RL y los componentes del paradigma de DRL. Finalmente, se detalla el algoritmo TD4 propuesto.

Capítulo 4

El Capítulo 4 plantea la problemática a resolver en el caso de estudio, un sistema PV compuesto por cuatro módulos PV y un convertidor tipo boost, el cual se implementa en MATLAB/Simulink. Adicionalmente, este capítulo, describe la implementación virtual del algoritmo TD4; y finalmente, se verifica la eficacia de esta técnica GMPPT mediante una serie de simulaciones, y es comparada otras técnicas MPPT/GMPPT.

Capítulo 5

El Capítulo 5 presenta las conclusiones de la tesis y los trabajos futuros que pueden derivarse de ésta.

Capítulo 2

Sistemas Fotovoltaicos

En este capítulo se explica la importancia de la generación fotovoltaica en la actualidad y se discute la relevancia de su desarrollo. Respecto a los sistemas PV, se detalla el modelo matemático de la celda fotovoltaica y se analizan los efectos de la irradiancia solar y la temperatura en sus características eléctricas. Finalmente, se explica el fenómeno de PS y se justifica la necesidad de incorporar un convertidor electrónico DC-DC para realizar la tarea de MPPT.

2.1. Introducción

Se denomina generación PV a la transformación de la energía solar a energía eléctrica mediante dispositivos semiconductores. La relevancia de este tipo de generación radica en que la energía solar es el recurso renovable más abundante sobre la superficie del planeta (Mirza et al., 2020). Se estima que la cantidad de energía solar que impacta la superficie del planeta en una hora es mayor que la cantidad de energía total consumida en el mundo durante el periodo de un año (Dupont et al., 2020). No obstante, al considerar restricciones como la disponibilidad de suelo y la eficiencia de los sistemas de conversión y transmisión de energía, es evidente que solo una fracción de la energía solar puede ser transformada para su uso (Dupont et al., 2020). A pesar de esto, la generación PV tiene el potencial de exceder la demanda actual de energía eléctrica hasta doce veces, de ahí que sea la fuente renovable con mayor crecimiento en los últimos años (Perez & Perez, 2022).

Otras ventajas que presentan los sistemas PV son: facilidad de instalación y escalamiento, mantenimiento infrecuente y poco costoso, nula producción de ruido y desechos en el proceso de transformación de la energía, y, en los últimos años, su costo se ha reducido considerablemente (Kannan & Vakeesan, 2016). Estas características han permitido su adopción en una amplia variedad de sectores, tales como el de la generación (Vartiainen et al., 2020), el industrial (Adesanya & Schelly, 2019), el residencial (Arcos-Vargas et al., 2018), el comercial (Mbungu et al., 2020), y recientemente, el sector del transporte (An, 2021). En consecuencia, la capacidad mundial acumulada de generación fotovoltaica aumentó 17 veces, de 40 GW en 2010, a 707 GW en 2020 (Ritchie et al., 2020), y se prevé

que se agreguen en promedio 444 GW/año hacia el año 2050 (BP, 2021).

La incipiente irrupción de la generación PV en el mercado eléctrico también se ve reflejada en la comunidad científica, ya que de los más de 48,000 estudios publicados desde comienzos del siglo XX sobre generación PV, el 50 % se realizó entre los años 2015 y 2020 (Reyes-Belmonte, 2021). No obstante, al margen de los pronósticos optimistas y alto nivel de aceptación social, la generación PV aún enfrenta desafíos importantes, como la baja PCE de las celdas fotovoltaicas, la incertidumbre en la disponibilidad de generación, las pérdidas por la exposición a patrones no uniformes de irradiancia, y el desajuste de impedancias entre la fuente y la carga. Estos dos últimos, conocidos también como condición de PS y tarea de MPPT, respectivamente, son tratados en esta tesis.

2.2. Clasificación de las tecnologías fotovoltaicas

Los sistemas PV están compuestos, entre otros elementos, por arreglos de celdas fotovoltaicas. Según su tipo de material, las celdas PV se pueden agrupar en tres categorías: silicio, compuestos de semiconductores, y materiales emergentes (Ibn-Mohammed et al., 2017). La Fig. 2.1 muestra a detalle la clasificación general de las celdas fotovoltaicas por material.

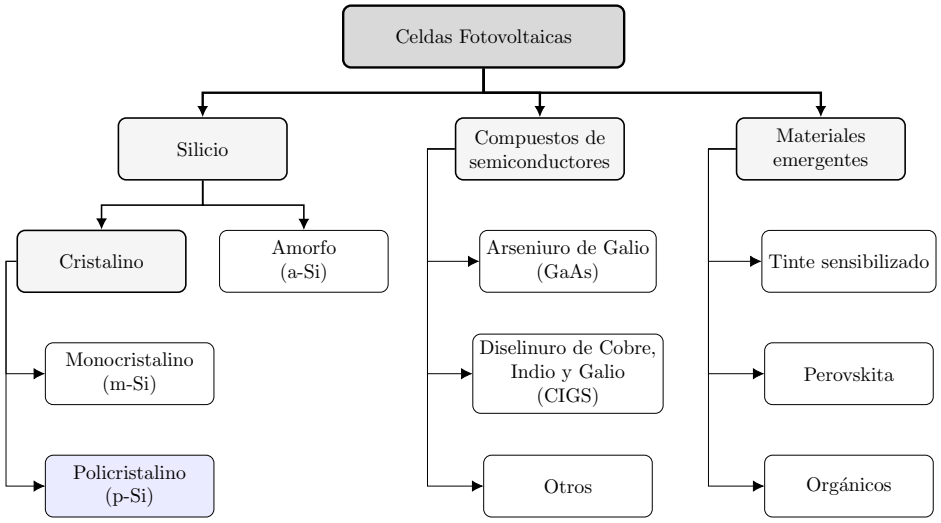


Figura 2.1: Clasificación de las celdas fotovoltaicas por material. Se destaca el silicio policristalino, cuyo modelo se utiliza en esta tesis.

A pesar de la gran cantidad de estudios recientes sobre nuevas tecnologías y materiales, los paneles de silicio gozan de una gran popularidad en el mercado. Hasta el año 2017, la cuota de mercado abarcada por módulos fotovoltaicos de silicio cristalino (c-Si) ascendía a 95 % (Sinke, 2019), y en el 2020 era de alrededor del 90 % (Sutherland, 2020). Este dominio en el mercado se debe principalmente a su relación costo-beneficio, ya que el tiempo de recuperación de la inversión económica es de 6 años, mientras que la vida útil promedio de los paneles es de 25 años; es decir, un sistema PV de silicio puede ser redituable económica y energéticamente hasta por 19 años (Broughton et al., 2022; Nikolic et al., 2022; Rauf et al., 2021).

Asimismo, las celdas fotovoltaicas c-Si se clasifican en dos grupos: celdas de silicio monocristalino (m-Si) y celdas de silicio policristalino (p-Si). Mientras las celdas de m-Si tienen mayor PCE que las celdas de p-Si, siendo 26 % el de las primeras y 20 % el de las últimas (Ciulla et al., 2014), la relación costo-beneficio a largo plazo en ambas es similar (Obeng et al., 2020), con una mejor tendencia hacia el lado de las celdas p-Si por su menor costo inicial (Ameur et al., 2020). Esto coincide con el hecho de que la cuota de mercado ocupada por las celdas p-Si es del 62 % (Sofia et al., 2020). Considerando la popularidad del silicio policristalino, esta tesis se enfoca en sistemas PV compuestos por estas celdas.

2.3. Modelado

El modelo matemático de la celda PV utilizado en esta tesis es el denominado *modelo de diodo único*, el cual ha sido ampliamente estudiado y ofrece un buen compromiso entre simplicidad y exactitud (Messalti et al., 2015; Rhouma et al., 2017; Sera et al., 2007). Este modelo integra una fuente de corriente, un diodo y dos resistencias, como se muestra en la Fig. 2.2. Como se mencionó anteriormente, las celdas PV están compuestas por un material semiconductor al igual que los diodos, por lo que en condiciones ideales, una fuente de corriente y un diodo son eléctricamente equivalentes a una celda PV (Sarkar, 2016). Estos dos elementos simulan la capacidad de absorción de radiación solar de la celda, mientras que los resistores consideran las pérdidas energéticas relacionadas con su fabricación y operación. Específicamente, la resistencia en paralelo modela la corriente de fuga debido a impurezas en el material, en tanto que la resistencia en serie representa las pérdidas internas debidas al flujo de corriente y a los cables de conexión (Villalva et al., 2009).

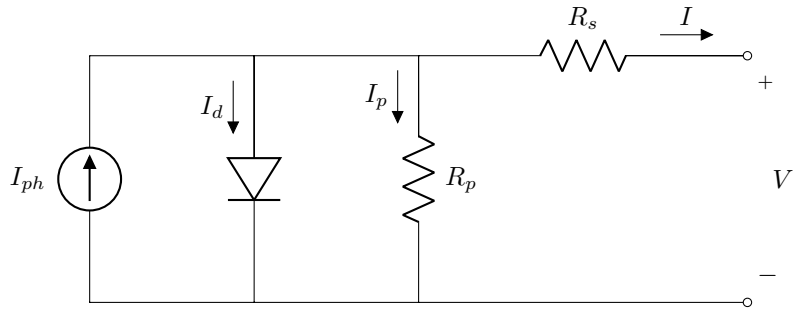


Figura 2.2: Diagrama esquemático del modelo de diodo único de la celda fotovoltaica.

La corriente generada por la celda puede ser determinada directamente de la Fig. 2.2, como:

$$I = I_{ph} - I_d - I_p, \tag{2.1}$$

donde I_{ph} es la corriente total generada debido al efecto fotovoltaico, I_d está determinada por la ecuación Shockley del diodo, e I_p es la corriente que fluye a través de R_p.

Al sustituir los dos últimos términos en la Ec. (2.1) por sus valores correspondientes, se obtiene la ecuación del

modelo de diodo único de la celda fotovoltaica, expresada como:

$$I = I_{ph} - I_0 \left[\exp\left(\frac{V + IR_s}{A\mathbf{k}_B T/q} \right) - 1 \right] - \frac{V + IR_s}{R_p}, \tag{2.2}$$

donde V es el voltaje en terminales de la celda, I_0 es la corriente de saturación inversa del diodo, A es una constante que expresa la idealidad del diodo, T es la temperatura del semiconductor (en kelvin), $\mathbf{k}_B$ es la constante de Boltzmann ($\approx 1.38064852 \times 10^{-23}$ J/K), y q es la carga del electrón ($\approx 1.60217662 \times 10^{-19}$ C).

Las celdas PV pueden agruparse para conformar arreglos (o paneles). Las celdas conectadas en serie proporcionan un voltaje de salida mayor, mientras que las celdas conectadas en paralelo suministran una mayor corriente. De esta manera, el modelo matemático que rige el comportamiento de un sistema PV compuesto es (Gow & Manning, 1999):

$$I = N_p \left\{ I_{ph} - I_0 \left[\exp\left(\frac{VN_p + IR_s N_s}{N_s N_p A\mathbf{k}_B T/q} \right) - 1 \right] - \frac{VN_p + IR_s N_s}{N_s N_p R_p} \right\}, \tag{2.3}$$

donde N_p es el número de arreglos en paralelo de N_s celdas conectadas en serie.

Por otro lado, la irradiancia solar y la temperatura son dos condiciones ambientales que afectan las condiciones de operación en un sistema PV. El modelo matemático que describe la influencia de la irradiancia solar y la temperatura en la corriente foto-generada I_{ph} se describe como (Duffie et al., 2013):

$$I_{ph} = \left[I_{ph,\mathrm{ref}} + \mu_I \left(T - T_{\mathrm{ref}} \right) \right] \frac{G}{G_{\mathrm{ref}}}, \tag{2.4}$$

donde G es la irradiancia solar incidente sobre la superficie de la celda, G_{ref} es la irradiancia de referencia (generalmente 1,000 W/m^2), $I_{ph,\mathrm{ref}}$ es la corriente foto-generada de referencia, μ_I es el coeficiente de corriente/temperatura, T es la temperatura de la celda, y T_{ref} es la temperatura de referencia (generalmente 298.15 K).

Similarmente, la influencia de la temperatura en la corriente de saturación inversa I_0 es (Messenger & Ventre, 2004):

$$I_0 = I_{0,\mathrm{ref}} \left(\frac{T}{T_{\mathrm{ref}}} \right)^3 \exp\left[\frac{qE_g}{A\mathbf{k}_B} \left(\frac{1}{T_{\mathrm{ref}}} - \frac{1}{T} \right) \right], \tag{2.5}$$

donde $I_{0,\mathrm{ref}}$ es la corriente de saturación inversa nominal del diodo, y E_g es la energía de la banda prohibida del semiconductor (1.2 eV para el silicio). Note que la irradiancia no influye en la corriente de saturación.

Finalmente, es importante aclarar que la temperatura de la celda no es igual a la temperatura ambiente. No obstante, es posible aproximar la temperatura de la celda a partir de la irradiancia solar y la temperatura ambiente. Esto es especialmente útil cuando se desea obtener la respuesta del sistema ante condiciones ambientales reales, ya que no resulta práctico ni sencillo obtener mediciones de la temperatura de las celdas, mientras que la temperatura ambiente se puede registrar con extrema facilidad. Así, la temperatura de la celda se puede estimar como (Messenger

& Ventre, 2004):

$$T = T_a + \left(\frac{\text{NOCT} - 293.15}{800}\right) G, \tag{2.6}$$

donde T_a es la temperatura ambiente (K), y NOCT es la temperatura nominal de operación de la celda (318.15 K, generalmente).

Así, las Ec. (2.3)–(2.6) determinan por completo el comportamiento de un sistema PV a partir de sus condiciones ambientales. En la siguiente sección se describe cómo se determina el punto de operación respecto a sus características eléctricas y a la carga conectada.

2.4. Dinámica de un Sistema PV

La zona de operación (i.e., curva I–V) de un sistema PV es caracterizada por una representación no lineal de la corriente en función del voltaje, con temperatura e irradiancia solar constantes. Existen distintos procedimientos para obtener la curva I–V de un sistema PV, algunos se basan en el uso de cargas electrónicas variables, resistencias de carga variables, convertidores electrónicos de potencia, entre otros (Vega et al., 2019). Aún así, todos los métodos se basan en el mismo principio: sensar la magnitud de la corriente al variar el voltaje del sistema desde la condición de cortocircuito hasta la condición de circuito abierto (Zhu & Xiao, 2020).

Considere el sistema PV mostrado en la Fig. 2.3, compuesto por un arreglo PV y una resistencia de carga variable. Los parámetros de este sistema se muestran en la Tabla 2.1.

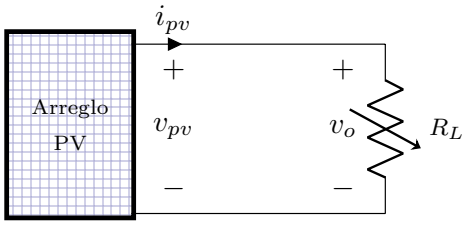
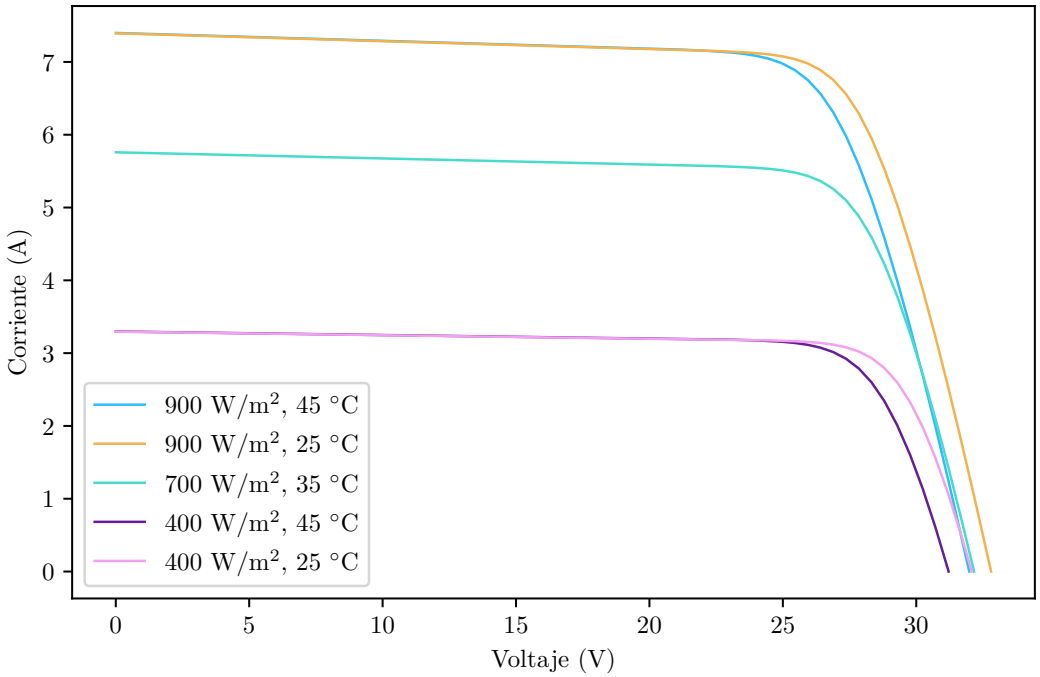


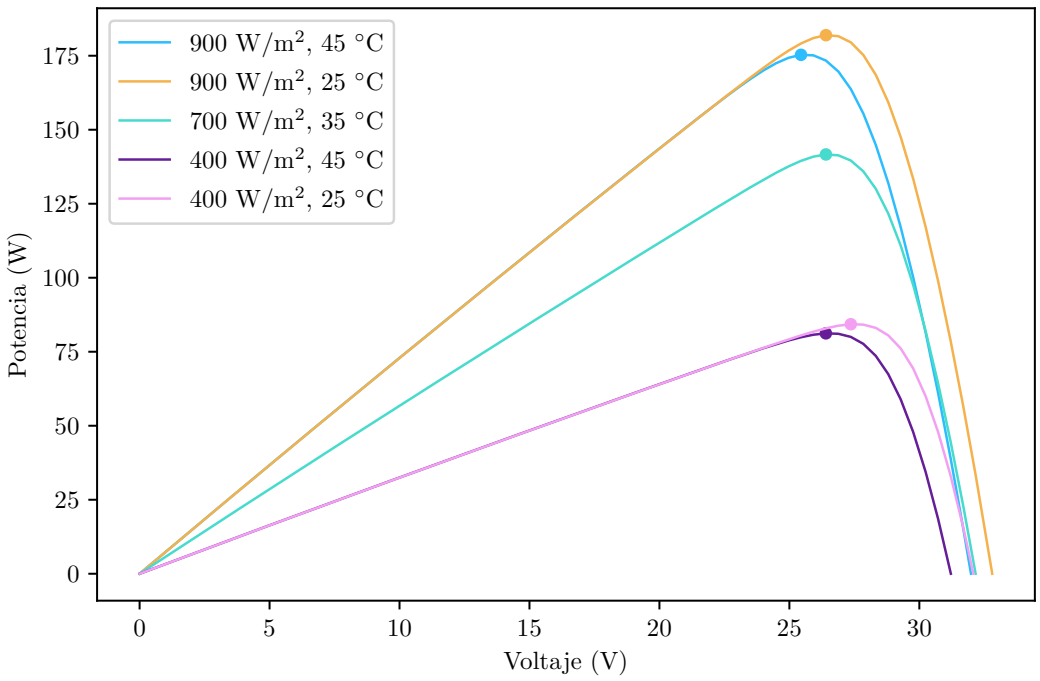
Figura 2.3: Sistema PV simple.

Note que las Ec. (2.3)-(2.6) están en función de parámetros distintos a los mostrados en la Tabla 2.1. Estos últimos son medidos experimentalmente y proporcionados por el fabricante. Antes de realizar la simulación, es necesario obtener los parámetros requeridos por las ecuaciones, lo que se conoce como *estimación paramétrica*. En esta tesis se utilizó el método de estimación paramétrica desarrollado por Cortés et al. (2020).

La curva I–V de este sistema se obtiene al incrementar el valor de la resistencia de carga R_L, desde la resistencia mínima hasta su valor máximo. No obstante, la curva I–V depende de las condiciones ambientales; es decir, existe una curva única para cada par de condiciones de irradiancia solar y temperatura. La Fig. 2.4(a) muestra cinco condiciones ambientales diferentes y la curva I–V asociada con cada una de ellas. Se observa que la irradiancia



(a) Conjunto de curvas I–V.



(b) Conjunto de curvas P–V. El punto de máxima potencia se señala con un círculo sólido.

Figura 2.4: Curvas características del sistema PV mostrado en la Fig. 2.3, en diferentes condiciones ambientales.

Tabla 2.1: Descripción de los parámetros asociados al sistema PV en la Fig. 2.3.

	Parámetro	Valor
Arreglo PV Kyocera KC200GT	P_{max}	200 W
	V_{mpp}	26.3 V
	I_{mpp}	7.61 A
	V_{oc}	32.9 V
	I_{sc}	8.21 A
	N_s	54
Carga	R_L	80 Ω

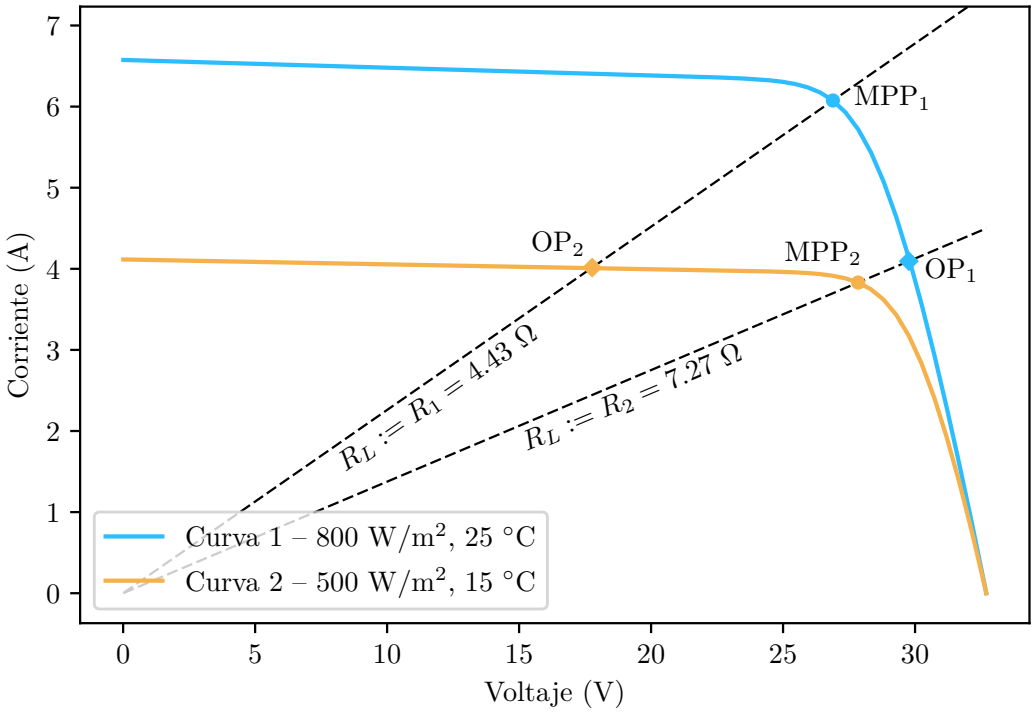
influye notablemente en la corriente de salida, una mayor cantidad de irradiancia produce una mayor cantidad de corriente; mientras que el efecto de la temperatura en la corriente es despreciable. Por otra parte, tanto la irradiancia como la temperatura exhiben un impacto en el voltaje de circuito abierto (i.e., el voltaje cuando no hay flujo de corriente): a mayor temperatura, el voltaje es menor; mientras que una irradiancia mayor, produce un mayor voltaje.

De manera similar, la curva P–V expresa gráficamente la relación entre la potencia producida y el voltaje del sistema, como lo muestra la Fig. Fig. 2.4(b). Los puntos indicados con un círculo sólido señalan el MPP de cada curva. Evidentemente, existe un único punto de operación donde la potencia generada es máxima. La tarea de localizar este voltaje y llevar al sistema a este punto de operación es lo que se conoce como MPPT.

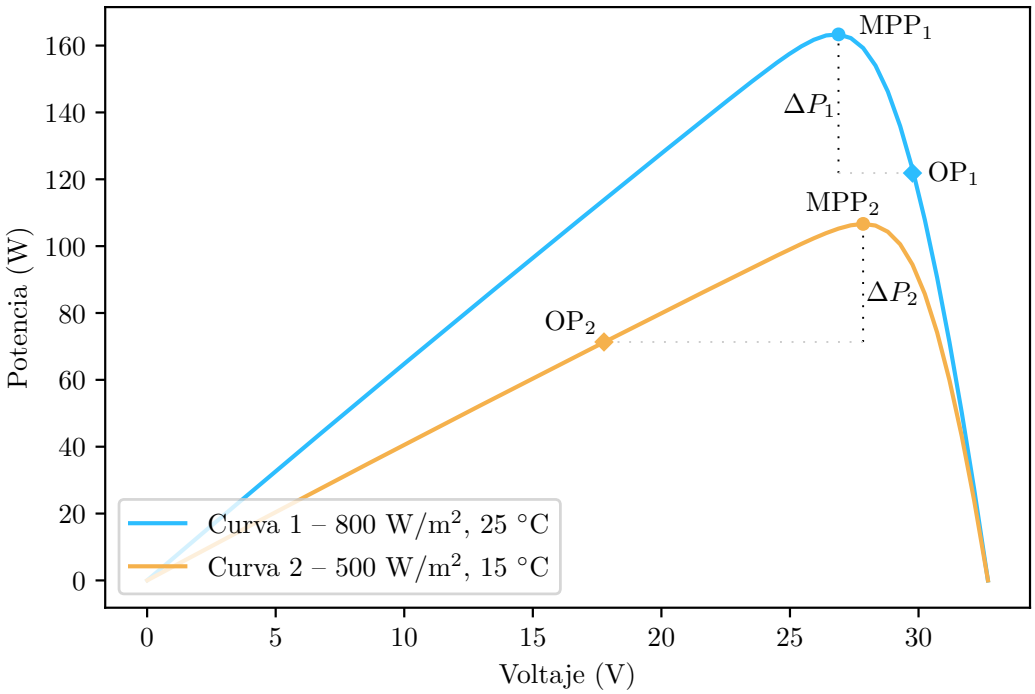
2.5. Convertidor DC-DC

El punto de operación de un sistema PV delimita la cantidad de energía generada en ese instante, y se define como la intersección entre la curva I–V y la curva de carga, como se muestra en la Fig. 2.5(a). Las curvas I–V corresponden al sistema de la Fig. 2.3 en dos condiciones ambientales distintas, y las curvas de carga corresponden a dos valores diferentes de resistencia. Cuando el sistema trabaja en la zona de operación definida por la Curva 1, el punto de operación coincide con el MPP si la resistencia en la carga es R_1; mientras que con la resistencia R_2, el punto de operación del sistema es OP_1. La Fig. 2.5(b) muestra ambos puntos de operación en la curva P–V, donde se observa que la Curva 1 genera alrededor de 160 W con la carga R_1, y 120 W con la carga R_2, esta diferencia se denota como ΔP_1. Similarmente, cuando el sistema PV está en la zona de operación definida por la Curva 2, el MPP se alcanza cuando la resistencia en la carga es equivalente a R_2.

Una forma alternativa de explicar la tarea del MPPT es la siguiente: considere que el sistema está trabajando en el punto MPP_1 mostrado en la Fig. 2.5(a), es decir, la irradiancia solar incidente sobre el sistema es de 800 $\mathrm{W/m^2}$ y la resistencia en la carga tiene un valor de 4.43 Ω. A continuación, la irradiancia solar disminuye a 500 $\mathrm{W/m^2}$ por algún fenómeno atmosférico determinado. En estas condiciones, el punto de operación se movería sobre la pendiente de resistencia de 4.43 Ω hacia la Curva 2, es decir, el sistema operaría en el punto OP_2, con un déficit de producción de potencia de ΔP_2, como se observa en la Fig. 2.5(b). En otras palabras, existe un único valor de



(a) La intersección de la curva I–V y la curva de carga definen el punto de operación del sistema fotovoltaico.



(b) Las curvas P–V demuestran la diferencia en la potencia generada para los puntos de operación definidos por una carga constante.

Figura 2.5: Curvas I–V y P–V de un sistema fotovoltaico con una carga resistiva constante.

resistencia en la carga para la cual el sistema PV produce la máxima potencia para las condiciones ambientales especificadas. De acuerdo con el *teorema de máxima transferencia de potencia*, la potencia suministrada a la carga es máxima cuando la impedancia interna de la fuente es igual a la resistencia en la carga (Vieira & Mota, 2010). Matemáticamente, la máxima transferencia de potencia ocurre cuando se tiene una resistencia en la carga tal que:

$$R_{L,\,mpp} = \frac{V_{mpp}}{I_{mpp}} \tag{2.7}$$

donde V_{mpp} e I_{mpp} son los valores de voltaje y corriente que producen el MPP para las condiciones ambientales actuales.

Por tanto, la tarea del MPPT consiste en determinar el valor de resistencia en la carga de tal manera que coincida con la impedancia interna equivalente del arreglo PV en el MPP. No obstante, la impedancia de la carga raramente está bajo el control del usuario, por lo que se utiliza un convertidor electrónico de potencia DC-DC como una interfaz entre ambas impedancias, como se muestra en la Fig. 2.6. La función del convertidor es modificar la resistencia aparente en la carga (i.e., la resistencia vista por el arreglo PV) de tal manera que coincida con $R_{L,\,mpp}$, descrita en la Ec. (2.7). Entre las diferentes topologías, el convertidor DC-DC tipo boost presenta las mejores características en eficiencia de conversión, regulación de voltaje, facilidad de implementación, y asequibilidad (Başoğlu & Çakır, 2016; Raghavendra et al., 2020; Sarwar et al., 2022).

Para un convertidor tipo boost, la relación entre las impedancias y el ciclo de trabajo está determinado por (Li et al., 2017):

$$R_{pv} = \frac{V_{pv}}{I_{pv}} = (1 - D)^2 R_L \tag{2.8}$$

donde R_{pv} es la resistencia equivalente vista desde el arreglo PV, R_L es la resistencia en la carga, y D es el ciclo de trabajo del convertidor.

De esta manera, el problema de MPPT consiste en determinar el ciclo de trabajo del convertidor que produzca la máxima transferencia de energía entre el arreglo PV y la carga.

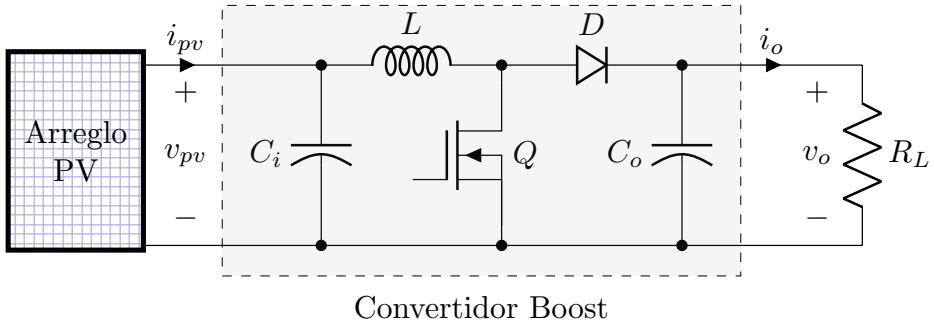


Figura 2.6: Sistema PV con convertidor DC-DC boost. El convertidor actúa como una interfaz entre la fuente y la resistencia en la carga, controlando el punto de operación del sistema al modificar el ciclo de trabajo.

2.6. Tarea de MPPT en condiciones uniformes

La tarea de encontrar el MPP en condiciones de irradiancia solar uniforme es relativamente sencilla, ya que el sistema PV presenta un único punto de operación para el cual la potencia es máxima. En este sentido, numerosos métodos se han propuesto para realizar esta tarea, desde los denominados métodos clásicos, hasta métodos modernos basados en AI y MHO (Mao et al., 2020). Cada uno de los algoritmos tiene sus propias ventajas y limitaciones. Las técnicas clásicas son fáciles de implementar, pero presentan velocidades de rastreo lentas. Las técnicas de AI y MHO presentan una mayor velocidad y eficiencia de rastreo, a cambio de un diseño e implementación más elaborados (Mansoor et al., 2020). De todas las técnicas MPPT propuestas, el algoritmo P&O es el más utilizado, principalmente por su facilidad de implementación (Abdel-Salam et al., 2018; Ishaque et al., 2014; Sera et al., 2013).

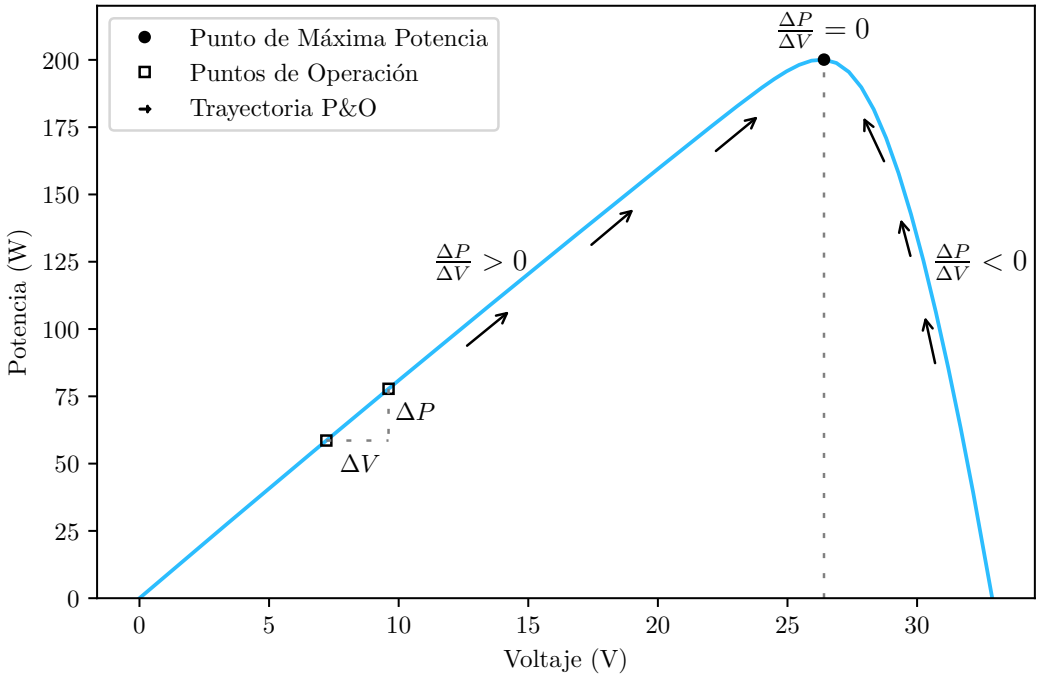


Figura 2.7: Dinámica del algoritmo MPPT P&O convencional en condiciones uniformes de irradiancia solar.

El funcionamiento del algoritmo P&O se basa en observar el cambio causado en la potencia de salida del sistema PV al introducir una perturbación en el punto de operación. Si el cambio en la potencia es positivo, la siguiente perturbación permanece en la misma dirección; si el cambio en la potencia es negativo, la dirección de la perturbación posterior cambia de sentido. Este proceso puede visualizarse en la Fig. 2.7, donde se muestra la trayectoria del algoritmo P&O en la curva P–V. El MPP divide a la curva en dos zonas: la zona izquierda de la curva exhibe una pendiente positiva, mientras que la pendiente en la zona derecha es negativa. En el MPP la pendiente es cero. Si el punto se encuentra en la zona izquierda, la dirección de la siguiente perturbación incrementará el voltaje del arreglo PV, moviendo el punto de operación hacia el MPP. Similarmente, si el punto de operación está a la derecha del MPP, el propósito de la siguiente perturbación es moverlo hacia la izquierda, disminuyendo el voltaje

del arreglo.

Para un convertidor tipo boost, el voltaje de la entrada es inversamente proporcional al ciclo de trabajo; es decir, un incremento del ciclo de trabajo provoca una disminución del voltaje en el arreglo. Por lo tanto, si el punto está a la izquierda del MPP, la perturbación en el ciclo de trabajo debe ser negativa, y viceversa. El diagrama de flujo del algoritmo P&O para un sistema PV con convertidor tipo boost se resume en la Fig 2.8.

A pesar de su simplicidad y confiabilidad, este algoritmo presenta dos inconvenientes principales. El primero, el algoritmo P&O produce oscilaciones en el estado estable, es decir, cerca del MPP. La magnitud de las oscilaciones es directamente proporcional a la magnitud de las perturbaciones y velocidad de seguimiento. El segundo y más severo inconveniente es la incapacidad de distinguir correctamente el GMPP en condiciones de PS, lo que impacta directamente en el rendimiento energético (Ahmed & Salam, 2015).

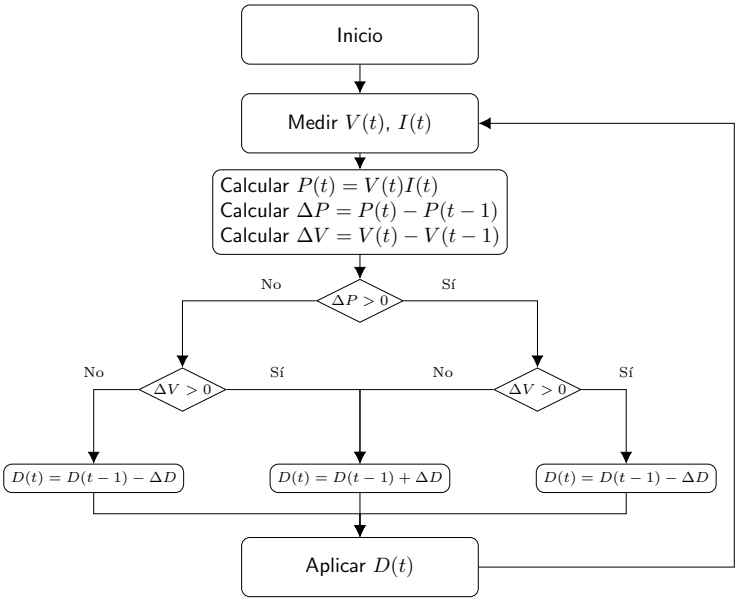
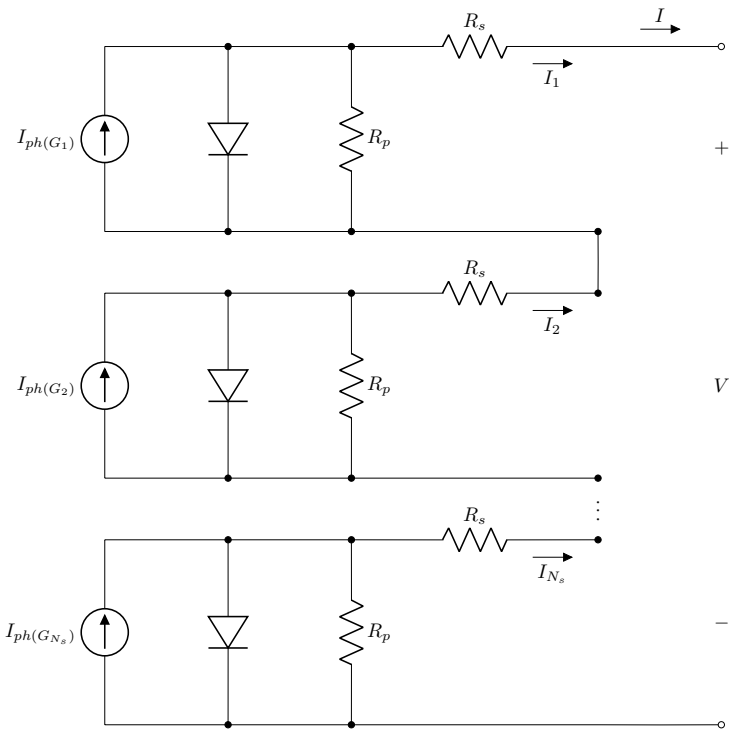


Figura 2.8: Diagrama de flujo del algoritmo MPPT P&O convencional para un sistema PV con convertidor boost.

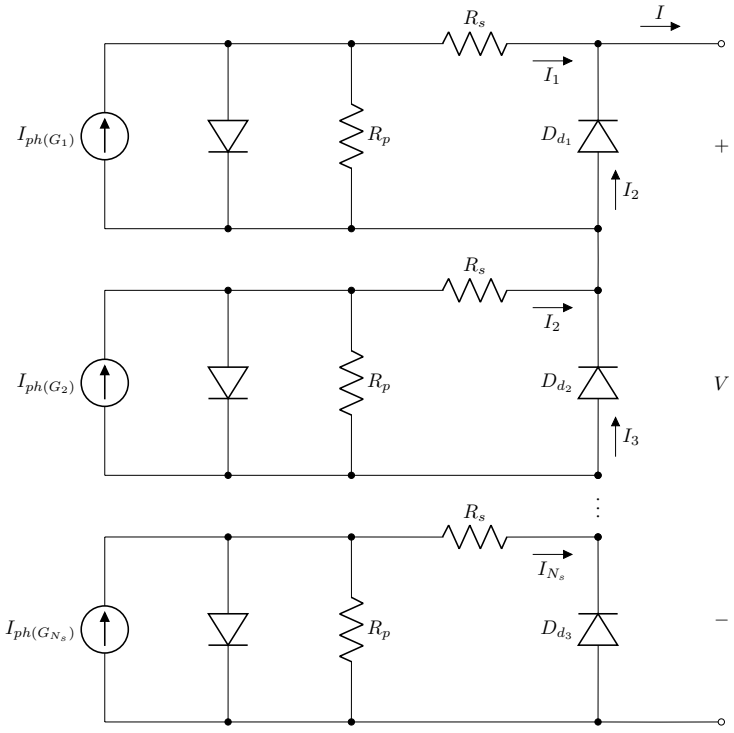
2.7. Sombreado Parcial

El efecto de PS se presenta cuando los módulos que componen el arreglo PV no reciben la misma cantidad de irradiancia. Esta situación se presenta principalmente por la oclusión parcial o total de la irradiancia solar por acción de las nubes, y en menor medida por acción de árboles, edificios y demás estructuras urbanas.

Un módulo PV está compuesto por un conjunto de celdas conectadas en serie, por lo que todas las celdas conducen la misma cantidad de corriente, como se observa en la Fig. 2.9(a). Si una celda se encuentra sombreada, la corriente generada por ésta se reduce; sin embargo, es forzada a conducir la misma cantidad de corriente que las celdas no sombreadas, polarizándose de manera inversa. Debido a que las celdas polarizadas inversamente actúan como una carga, una parte de la potencia generada por el arreglo PV se disipa en forma de calor a través de éstas,



(a) Sin diodos en derivación. Las celdas que generan una menor cantidad de corriente que la de la rama se polarizan inversamente y actúan como una carga, disipando potencia.



(b) Con diodos en derivación. Las celdas polarizadas inversamente hacen que el diodo en derivación entre en conducción, mientras que el diodo de las celdas polarizadas directamente no tiene efecto en el circuito.

Figura 2.9: Arreglo de N_s celdas PV en serie. La corriente que fluye por las celdas es la misma, independientemente de la cantidad de corriente generada por cada una.

lo que se traduce directamente en una pérdida de rendimiento. Más aún, las celdas sombreadas pueden sufrir daños permanentes si alcanzan temperaturas muy altas, inhabilitando a su vez a todo el arreglo PV (Fathy et al., 2020).

Para solucionar este problema se coloca un diodo, denominado *diodo en derivación*, en antiparalelo con la celda o grupo de celdas en serie que se desea proteger, como se muestra en la Fig. 2.9(b). La función del diodo es aislar los módulos sombreados del sistema, al proporcionar una ruta alternativa para el flujo de corriente, de manera que los módulos sombreados no necesiten trabajar en voltaje de polarización inversa. Así, si la corriente generada por la celda sombreada resulta menor a la generada por las otras celdas, el diodo entrará en conducción, desviando el flujo de corriente de la celda sombreada. En cambio, si la celda está polarizada directamente, el diodo en derivación no tiene efecto en el circuito (Tey et al., 2014). En la práctica, utilizar un diodo por cada celda eleva significativamente el costo de los arreglos PV comerciales, por lo que es común que se utilice un diodo por cada grupo de 20-24 celdas en serie, y uno más por módulo (Vieira et al., 2020).

No obstante, la integración de diodos en derivación afecta a la dinámica del sistema PV. En este sentido, la curva P–V muestra una mayor complejidad, al pasar de tener un único MPP a exhibir múltiples MPP, uno por cada diodo en derivación en conducción. Considere el sistema PV mostrado en la Fig. 2.10, el cual está compuesto por un arreglo de cuatro módulos, cada uno con 20 celdas en serie, un convertidor DC-DC y una carga resistiva.

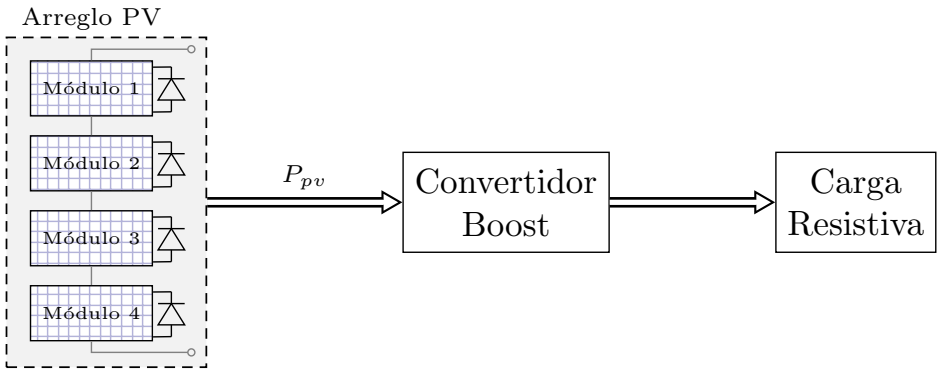
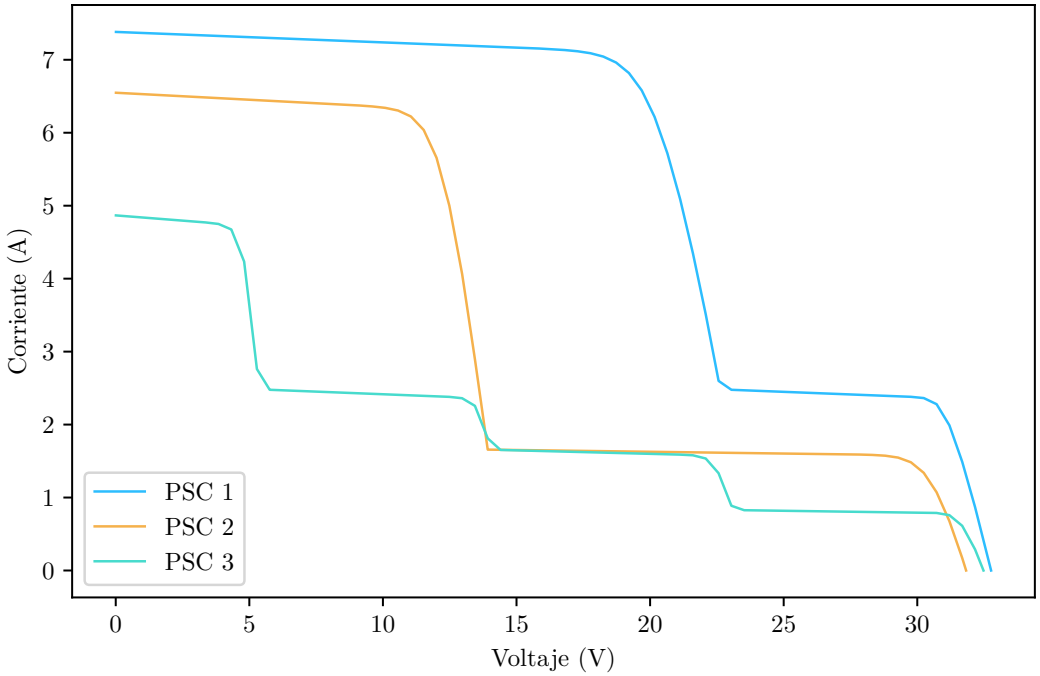


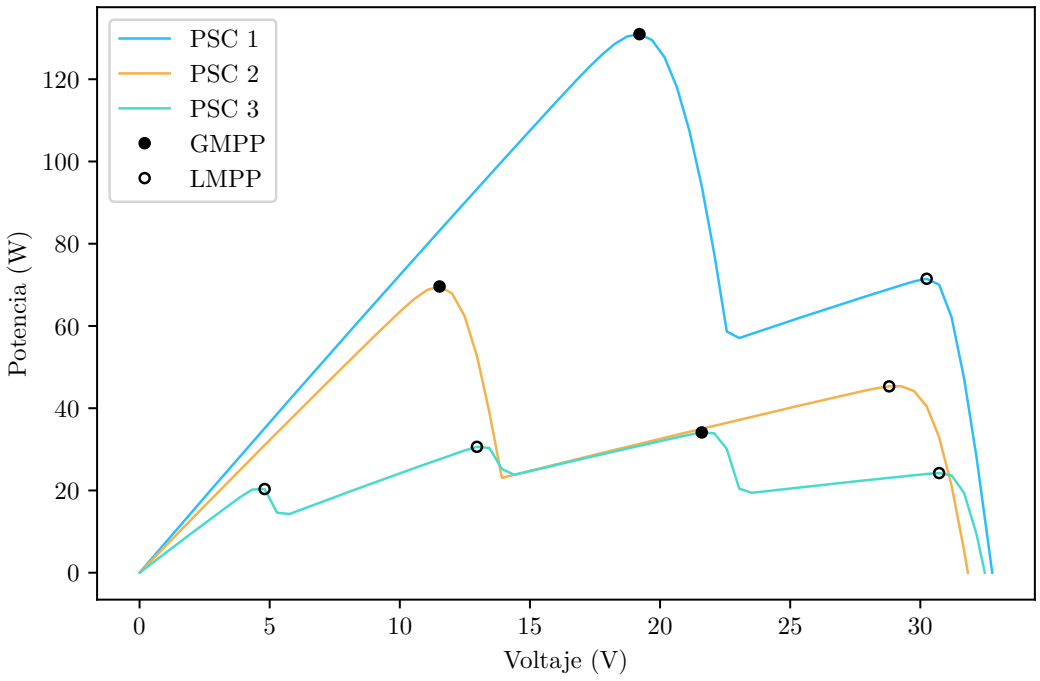
Figura 2.10: Sistema PV con diodos en derivación. El arreglo PV está compuesto por cuatro módulos en serie.

Las curvas características de este sistema en condiciones de PS se muestran la Fig. 2.11. Se puede observar que en cada curva solo existe un punto, denominado GMPP, que produce la máxima cantidad de potencia disponible en ese instante.

Resulta evidente que la tarea de GMPPT en condiciones de PS tiene una mayor complejidad que la tarea de MPPT en condiciones de irradiancia uniforme. Los algoritmos MPPT clásicos suelen quedar atrapados en máximos locales, identificando incorrectamente un *punto local de máxima potencia* (LMPP, *Local Maximum Power Point*) como un GMPP, lo que resulta en una pérdida de rendimiento (Chou et al., 2019). Si bien, los nuevos algoritmos GMPPT basados en AI y MHO permiten ubicar correctamente el GMPP, estos presentan ciertas desventajas, tales como complejidad en el diseño, requieren un amplio conocimiento del sistema o un gran número de iteraciones para lograr la convergencia del algoritmo (Kang et al., 2011).



(a) Conjunto de curvas I–V en PSC.



(b) Conjunto de curvas P–V en PSC. La identificación incorrecta del GMPP reduce el rendimiento del sistema.

Figura 2.11: Conjunto de curvas características en condiciones de sombreado parcial (PSC). Los conjuntos de valores de irradiancia correspondientes a PSC 1, PSC 2 y PSC 3 son $(900, 300, 900, 900)$, $(800, 200, 200, 800)$, y $(100, 200, 300, 600)$ W/m^2, respectivamente. La temperatura para todos los módulos es de 25 °C

.

Para solucionar estos problemas, en esta tesis se desarrolla una técnica GMPPT basada en DRL, la cual no requiere ningún conocimiento previo de la dinámica del sistema PV o sus características. En la siguiente sección se describen los fundamentos y se define formalmente el objetivo de DRL.

2.8. Conclusión

Al ser una tecnología renovable, la generación PV es una alternativa viable para satisfacer una gran parte de la demanda de energía eléctrica mundial en el futuro, a la vez que puede reducir el deterioro ambiental que la generación convencional ha ocasionado en las últimas décadas. No obstante, los sistemas de generación PV enfrentan distintos problemas que limitan su desempeño, y por ende, dificultan su expansión como una tecnología de generación redituable, de entre los cuales destacan los siguientes:

- Las celdas PV actuales tienen una PCE promedio de 20 %; sin embargo, aumentar esta eficiencia implica elevar los costos de producción, lo cual repercutiría directamente en una menor instalación de nuevos sistemas PV, en especial en los sectores residencial y comercial.
- Las condiciones ambientales determinan el punto de operación óptimo del sistema PV, en el cual se genera la máxima potencia en ese instante. Estas condiciones ambientales cambian continuamente a lo largo del día, por lo que el controlador MPPT debe ajustar constantemente el punto de operación del sistema para conseguir el mejor rendimiento. En este sentido, la velocidad del MPPT es una característica importante para la eficiencia del sistema, ya que un menor tiempo de convergencia implica un mayor tiempo de generación cerca del MPP.
- Debido a que el punto de operación del sistema PV está definido por la resistencia de carga, y ésta generalmente no está bajo el control del usuario, se utiliza un convertidor DC-DC para ajustar la resistencia equivalente de carga vista por el arreglo PV. De esta manera, la tarea del MPPT se reduce a encontrar el ciclo de trabajo del convertidor para el cual la resistencia equivalente coincida con la resistencia de carga óptima.
- Distintos fenómenos ambientales pueden afectar la eficiencia del sistema PV. El sombreado parcial es uno de los más frecuentes. En esta condición, la curva P–V del sistema exhibe múltiples MPP, lo que dificulta la tarea de GMPPT. Distintos algoritmos presentan pérdidas de rendimiento importantes al no ser capaces determinar correctamente el GMPP.

Mientras que la mejora de la PCE de las celdas es una labor multidisciplinaria a largo plazo, el desarrollo de técnicas GMPPT que permitan extraer la máxima cantidad de potencia disponible, independientemente de las condiciones ambientales actuales, es una tarea que puede realizarse actualmente. Adicionalmente, la creación de nuevas técnicas GMPPT pueden beneficiar tanto a los sistemas instalados, como a los futuros sistemas de generación PV. En este sentido, los avances en el área de DRL han permitido desarrollar distintas técnicas de control con resultados satisfactorios, las cuales pueden extenderse al área de generación PV.

Capítulo 3

Aprendizaje Profundo por Refuerzo

En este capítulo, se describen los conceptos básicos dentro del campo RL, tales como el *agente*, la *recompensa* y el *entorno*. Se presenta la notación y se define formalmente el problema de RL, los tipos básicos de algoritmos de RL y la transición del aprendizaje *clásico* al *profundo*. Finalmente, se presenta el algoritmo de DRL propuesto en esta tesis, denominado TD4.

3.1. Introducción

La *inteligencia artificial* (AI, *Artificial Intelligence*) comprende el estudio de las técnicas utilizadas por los sistemas para resolver problemas complejos de toma de decisiones, de forma independiente o con una intervención humana mínima. Las primeras investigaciones de AI se centraron en la emulación del proceso de toma de decisiones de los humanos, codificado en lenguajes de programación como reglas de inferencia lógica. Sin embargo, este paradigma enfrenta varias limitaciones, tales como la complejidad de la tarea de abstracción de las reglas a partir del vasto conocimiento del operador, y la adquisición de rasgos inherentes al comportamiento humano (Janiesch et al., 2021).

El *aprendizaje de máquina* (ML, *Machine Learning*) supera dichas limitaciones al liberar al ser humano de la carga de explicar y formalizar su conocimiento en una forma accesible a la máquina y permite desarrollar sistemas inteligentes de manera más eficiente. Las técnicas de ML determinan patrones complejos y relaciones significativas a partir de datos pre-procesados, habilitando a los sistemas a aprender a realizar tareas sin haber sido explícitamente programados para ello (Jordan & Mitchell, 2015).

El área de ML puede dividirse en tres campos: el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. En el primero, los sistemas aprenden a partir de datos etiquetados (con intervención humana), en el segundo, a partir de datos no etiquetados (sin intervención humana), y en el tercero, a partir de datos auto-etiquetados generados al interactuar con el sistema (Li, 2017). Típicamente, el aprendizaje supervisado se utiliza en clasificación (Krizhevsky et al., 2012) y aproximación de funciones (Nabipour et al., 2020), el no

supervisado se emplea en agrupamiento (i.e., *clustering*) (Caron et al., 2018), y el aprendizaje por refuerzo tiene un gran número de aplicaciones, como sistemas de recomendación (e.g., predicción de compras) (Xin et al., 2020b), sistemas de despacho de energía eléctrica (Mason & Grijalva, 2019) y sistemas de navegación autónoma (Kiran et al., 2021).

Conforme a la tarea de aprendizaje, ML ofrece varios algoritmos de aprendizaje, incluyendo modelos de regresión, árboles de decisión, métodos Bayesianos, *redes neuronales artificiales* (ANNs, *Artificial Neural Networks*), entro otros. Las ANNs son de particular interés, debido a que su estructura flexible permite adaptarlas a una amplia variedad de conceptos en los tres campos de ML. En este sentido, DL es una técnica de ML que está basada en el aprendizaje de múltiples niveles de características, las cuales son obtenidas a partir de varias operaciones sucesivas de procesamiento de datos, donde cada operación se denomina una *capa* del modelo (Chollet, 2021). La ventaja de utilizar ANNs como capas de procesamiento sobre otras técnicas como modelos de regresión y árboles de decisión, es que estas últimos requieren la selección manual de las variables dependientes y las explicativas para la construcción del modelo, lo que se conoce como diseño de características. En cambio, las ANNs permiten establecer una relación directa entre los datos y la salida del sistema, permitiendo la extracción de información que incluso puede pasar inadvertida a los humanos. Es decir, la arquitectura DL combina dos procesos de aprendizaje simultáneos: obtención de características y construcción del modelo. Así, los sistemas que implementan DL son conocidos sistemas de extremo a extremo (*end-to-end*) (LeCun et al., 2015).

3.2. Aprendizaje por Refuerzo

El aprendizaje por refuerzo es el tercer paradigma de ML, y es la técnica de aprendizaje que más se asemeja a la forma en la que los humanos aprendemos: a través de un proceso continuo de prueba y error (Kelly et al., 2020). RL es un campo de estudio multidisciplinario, en el cual intervienen distintas ciencias como la Informática, la Ingeniería y la Neurociencia, como se observa en la Fig. 3.1 (Lapan, 2018).

RL está estrechamente relacionado con la teoría del control óptimo clásico. Sin embargo, mientras que el control óptimo supone un conocimiento perfecto del modelo del sistema, RL opera en función de las métricas de rendimiento devueltas como consecuencia de las interacciones con un entorno desconocido (Goecks, 2020).

3.2.1. Bases Teóricas

RL estudia la optimización del desempeño de un agente en un proceso de toma de decisiones, a partir únicamente de observaciones y acciones. Existen dos entidades fundamentales en todo algoritmo de aprendizaje por refuerzo: el *agente* y el *entorno*. El agente se entiende como cualquier entidad capaz de percibir y actuar, donde la *percepción* implica la recepción y el procesamiento de información, y la *acción* define el proceso de elección en un conjunto de trayectorias de comportamiento. El entorno se define como el sistema o proceso donde se desempeña el agente.

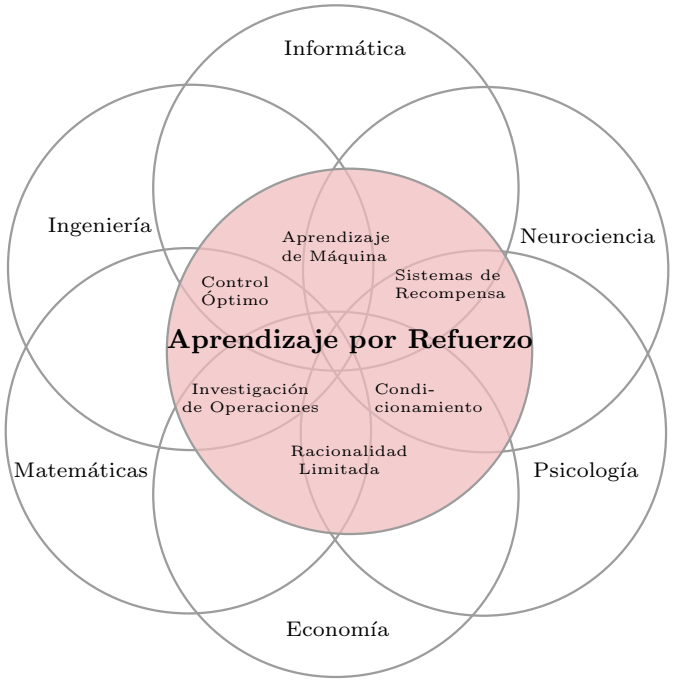


Figura 3.1: Varios campos de estudio intervienen en RL.

El proceso de interacción entre el agente y el entorno puede resumirse en tres etapas recurrentes, como se muestra en la Fig. 3.2. Primero, el agente observa el estado s_t del entorno. Segundo, el agente determina y ejecuta la acción a_t. Tercero, el entorno evoluciona a un nuevo estado s_{t+1} y envía una señal de recompensa r_{t+1} al agente, la cual es utilizada en el proceso de aprendizaje interno del agente. Finalmente, la interacción vuelve a comenzar con la observación del nuevo estado s_{t+1}. Este proceso iterativo de interacción se repite indefinidamente a lo largo de los denominados *episodios*. Dependiendo del entorno, los episodios pueden tener una duración predeterminada, pueden terminar súbitamente al llegar a un estado final, o pueden continuar de forma indefinida. En esta tesis se considera que los episodios tienen una duración predeterminada.

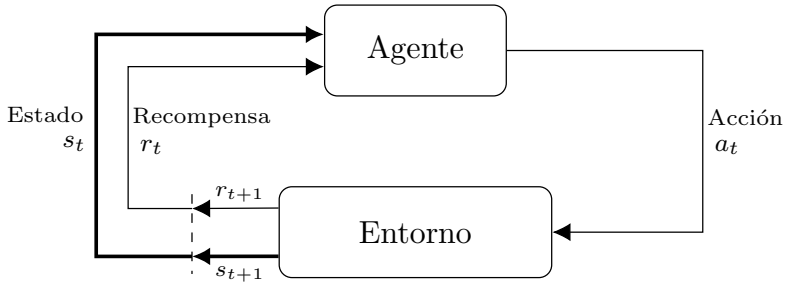


Figura 3.2: Esquema general de RL.

3.2.2. Proceso de Decisión de Markov

Los problemas de RL pueden ser descritos formalmente como un *proceso de decisión de Markov* (MDP, *Markov Decision Process*) (Shuvo et al., 2021), lo que permite encontrar soluciones de manera sistemática. Formalmente,

un MDP se define como (Birhanie et al., 2018):

Definición 3.1 *Un **MDP** está compuesto por cinco elementos, $\mathcal{M} = \{S, A, \mathcal{T}, r, \gamma\}$, donde:*

- *Espacio de estado, S: conjunto de variables observables que representan la condición actual del entorno (donde un vector de estado $s \in S$).*
- *Espacio de acción, A: conjunto de acciones admisibles que inducen un cambio en el entorno (donde un vector de estado $a \in A$).*
- *Función de transición, $\mathcal{T}_{s_{t+1}}^{s_t a} = P(s_{t+1}|s_t, a)$: denota la probabilidad de transición del entorno de un estado s_t a un estado s_{t+1} después de ejecutar una acción a.*
- *Función de recompensa, $R : S \times A \mapsto \mathbb{R}$: define la recompensa inmediata r_{t+1} que el agente recibe por tomar la acción a_t en el estado s_t y pasar a s_{t+1}.*
- *Factor de descuento, γ: valor escalar que indica la ponderación de las recompensas futuras ($0 \leq \gamma \leq 1$). Un valor de 0 considera acciones basadas únicamente en la recompensa inmediata r_{t+1} asociada con el estado s_t, mientras que un valor > 0 considera acciones que pueden llevar a recompensas futuras grandes aún después de obtener recompensas inmediatas menores (Kalogerakis et al., 2020).*

Para que un proceso pueda definirse como un MDP, debe cumplir con la propiedad de Markov, la cual estipula que el estado futuro s_{t+1} depende únicamente del estado actual s_t, de la acción ejecutada a_t y de la dinámica inherente al proceso. Es decir, los estados pasados no influyen en la transición a estados futuros. Además, la función de recompensa R depende únicamente de la transición entre estados (s_t y s_{t+1}) y de la acción ejecutada (a_t) (Shi et al., 2020). La propiedad de Markov se ejemplifica gráficamente en la Fig. 3.3, donde los bloques denotan los estados, acciones y recompensas, y las flechas indican la interdependencia entre ellos, de tal manera que el estado presente depende únicamente del estado inmediato anterior, y no de toda la historia de estados anteriores.

Matemáticamente, la propiedad de Markov se describe como:

$$\mathcal{T}\left(s_{t+1} \mid s_t, a_t\right) = p(s_{t+1} \mid s_0, a_0, \cdots, s_t, a_t) = p(s_{t+1} \mid s_t, a_t), \tag{3.1}$$

$$R(s_{t+1} \mid s_0, a_0, \cdots, s_t, a_t) = R(s_t, a_t, s_{t+1}), \tag{3.2}$$

donde p es una función de probabilidad inherente al proceso, que puede o no ser conocida.

Las Ecuaciones (3.1) y (3.2) expresan que existen funciones que caracterizan completamente la distribución del estado futuro y de la recompensa inmediata a partir únicamente del estado actual y de la acción ejecutada.

La función de recompensa R es una métrica de la pertinencia de la acción ejecutada; es decir, la recompensa r_{t+1} es un valor numérico que indica al agente si la acción a_t fue buena o mala. La única restricción que se impone a

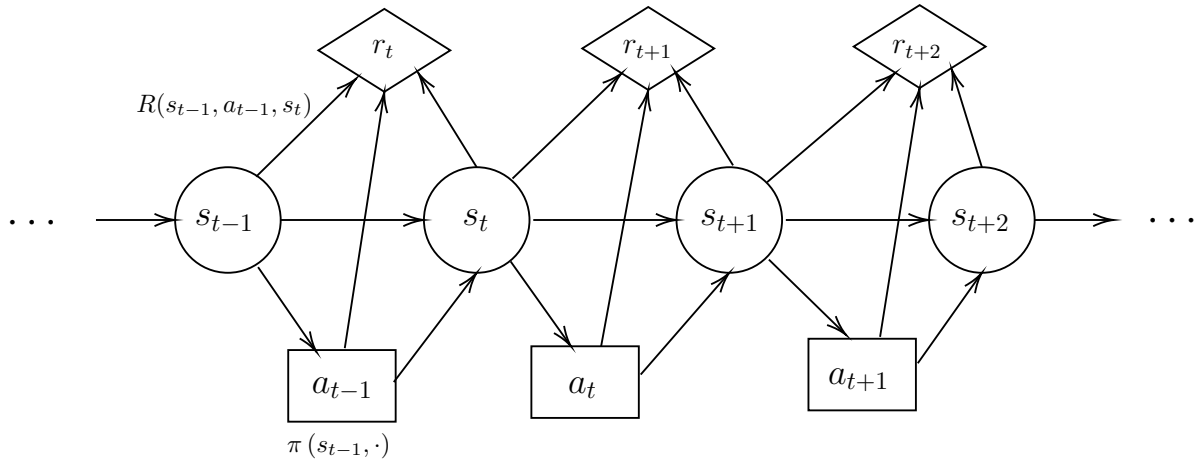


Figura 3.3: Diagrama general de un MDP.

la señal de recompensa es ser un valor escalar, por lo que puede ser positiva, negativa o cero. Note que la función de recompensa puede expresarse de tres formas diferentes, todas ellas equivalentes entre sí: $R(s)$, $R(s, a)$, y $R(s, a, s_{t+1})$. Naturalmente, la tercera forma es la forma más general, pero en ocasiones se utilizan las dos primeras por brevedad en la notación. Además, la función de recompensa puede ser determinista o estocástica. A lo largo de esta tesis se considera únicamente la forma determinista.

Otra propiedad de los MDP es la consideración de que el estado observado por el agente y el estado real del proceso son equivalentes. Es decir, el agente no tiene incertidumbre sobre su estado actual, solo desconoce la función de transición. Un definición más general que considera incertidumbre en la observación es el llamado *MDP Parcialmente Observable* (POMDP, *Partially Observable Markov Decision Process*) (Zhu et al., 2017). En esta tesis se considera que el proceso puede modelarse como un MDP.

3.2.3. Definición Formal del Problema de RL

Bajo la premisa de que el agente RL interactúa con un MDP, el problema de RL puede formularse formalmente como:

Definición 3.2 *El **problema de RL** se puede formalizar como sigue. Un agente RL interactúa con un MDP $\mathcal{M} = \{S, A, \mathcal{T}, r, \gamma\}$ al repetir los siguientes tres pasos, comenzando con $t = 0$:*

1. *El agente observa el estado actual $s_t \in S$ de M.*
2. *El agente elige y ejecuta una acción $a_t \in A$.*
3. *$\mathcal{M}$ pasa a un nuevo estado $s_{t+1} \sim \mathcal{T}(s_{t+1} \mid s_t, a_t)$ y envía una recompensa $r_{t+1} = R(s_t, a_t, s_{t+1})$ al agente, el cual aprende de esta interacción.*

Durante un episodio, este proceso de interacción entre el agente y el entorno genera una secuencia alternativa de estados, acciones, nuevos estados, y recompensas, que se denomina una *trayectoria*. En otras palabras, la trayectoria es el camino que el agente sigue en el entorno durante un episodio.

Definición 3.3 *En un episodio, una* ***trayectoria*** τ *es una secuencia de interacciones que llevan al agente desde el estado inicial* s_0 *hasta el estado final* s_T*:*

$$\tau = \{s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, \cdots, r_{\mathbf{T}}, s_{\mathbf{T}}\}, \tag{3.3}$$

donde $\mathbf{T}$ *es el último instante de tiempo en el episodio.*

Evidentemente, diferentes trayectorias producirán diferentes recompensas. La recompensa total recolectada $\mathcal{G}$ en un episodio, denominada *retorno*, es la suma de todas las recompensas inmediatas recolectadas en la trayectoria.

Definición 3.4 *El* ***retorno****, o recompensa total acumulada, de una trayectoria* τ *es la suma de todas las recompensas inmediatas* r_i*:*

$$\mathcal{G}(\tau) = \sum_{t=0}^{\mathbf{T}} r(s_t, a_t), \tag{3.4}$$

Es importante señalar que la trayectoria, además de depender de la dinámica inherente al MDP y sus funciones de transición, depende de la acción elegida por el agente. En este sentido, el proceso de toma de decisiones del agente es conducido por una *política* π, la cual especifica la acción que el agente debe tomar en un estado. Formalmente, la política es una función que mapea estados a acciones.

Definición 3.5 *La* ***política*** $\pi : S \mapsto A$ *determina la acción elegida por el agente en cualquier estado* $s \in S$ *del MDP.*

De esta manera, la recompensa total acumulada de una trayectoria que sigue una política π se expresa como:

$$\mathcal{G}_\pi = \sum_{t=0}^{\mathbf{T}} \mathop{\mathbb{E}}_{a_t \sim \pi} \left[r\left(s_t, a_t\right) \right]. \tag{3.5}$$

Note que aunque la política π sea una función determinista, la transición entre estados depende de un proceso estocástico inherente al MDP. Por lo tanto, el retorno debe considerarse como un valor esperado.

El objetivo del agente es aprender la política que elija la mejor acción en cualquier estado, es decir, la política que maximice la recompensa esperada. Matemáticamente, la *política óptima* π^* se describe como:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=0}^{\mathbf{T}} \gamma^t r(s_t, a_t) \right], \tag{3.6}$$

donde γ el factor de descuento. Al seleccionar acciones, el agente considerará las recompensas futuras esperadas y maximizará el rendimiento durante un episodio completo. Para tratar con entornos con episodios infinitos, se

introduce un *factor de descuento* $0 \leq \gamma \leq 1$ en la suma de recompensas. Un factor de descuento $\gamma < 1$ hace que las recompensas inmediatas tengan más impacto que las recompensas futuras y también hace posible tratar con episodios infinitos que de otro modo podrían conducir a recompensas infinitas.

El Dilema Exploración–Explotación

Con el fin de obtener π^*, el agente requiere un vasto conocimiento del espacio de estado del entorno y de la función de recompensa (Chen et al., 2021). En una configuración estándar de aprendizaje por refuerzo, el agente no tiene esta información *a priori*, sino que simplemente prueba diferentes acciones y experimenta las consecuencias resultantes con respecto a la recompensa. Así, durante el proceso de aprendizaje el agente debe equilibrar entre explotar con avidez la información recopilada hasta el momento, para elegir acciones que produzcan recompensas más altas a corto plazo, y explorar continuamente el entorno para adquirir más información para lograr beneficios a largo plazo (Wang et al., 2018).

Durante la *fase de exploración*, el agente construye un modelo interno de sus interacciones con el entorno, aprende qué acciones conducen mejores recompensas y qué acciones es mejor evitar. Para promover el posible descubrimiento de nuevas acciones que conduzcan a mejores recompensas, la política suele dictar acciones sub-óptimas para recopilar la mayor cantidad de información del entorno durante la fase de exploración (Masadeh et al., 2018). A medida que la fase de aprendizaje avanza y el modelo interno del agente estima con mayor exactitud la recompensa esperada, la elección de las acciones se sustenta progresivamente en la experiencia acumulada, permitiendo seleccionar con mayor frecuencia aquellas acciones que generan recompensas mayores, pero permitiendo aún la selección de acciones sub-óptimas para evitar que el proceso de aprendizaje quede atrapado en un máximo local.

Una vez que el modelo interno es lo suficientemente adecuado, el agente puede *explotar* el conocimiento aprendido, tomando siempre la acción que genere la mejor recompensa en cada estado.

Desafortunadamente, no existe una regla que determine la proporción adecuada entre la fase de exploración y la fase de explotación. De hecho, el dilema de exploración–explotación es considerado un problema abierto en el paradigma de RL, por lo cual numerosos estudios se han dedicado a la investigación de distintas estrategias de exploración, tales como ruido paramétrico (Shao et al., 2019), estrategia de selección ϵ-greedy (Nguyen & La, 2019), muestreo de Thompson (Coronato et al., 2020), entre otras. Estas estrategias dependen del tipo de algoritmo de aprendizaje, como se detalla en las siguientes secciones.

3.2.4. Tipos de Algoritmos

La clasificación más general de los algoritmos RL distingue dos tipos: los *algoritmos basados en el modelo*, y los *algoritmos basados en datos* (i.e., sin modelo) (Moerland et al., 2020). Los primeros utilizan un modelo interno del entorno que predice estados futuros y evalúa el valor a largo plazo de diferentes acciones simulando sus posibles consecuencias (Plaat et al., 2021). Generalmente, este modelo no está disponible para el agente, por lo que tiene

que aprenderlo mediante la experiencia. No obstante, si el modelo aprendido no es lo suficientemente preciso, la optimización de políticas tiende a sobre-ajustarse a las deficiencias del modelo, lo que lleva a un comportamiento sub-óptimo o incluso provocar fallas catastróficas (Clavera et al., 2018).

Por su parte, el *aprendizaje sin modelo*, almacena estimaciones del valor a largo plazo de estados y acciones, y las actualiza directamente a partir de la experiencia (Akam & Walton, 2021). A pesar de requerir más interacciones con el entorno, los algoritmos RL sin modelo pueden lograr un mejor rendimiento ya que no sufren de sesgos en el modelo que conducen a comportamientos sub-óptimos. En la práctica, esta característica hace que los algoritmos sin modelo sean más utilizados que los basados en el modelo (Tu & Recht, 2019).

En esta sección se presentan cuatro algoritmos básicos de RL. Primero, se introduce el algoritmo inspirado en la función de valor de estado, el cual es un algoritmo basado en el modelo. Posteriormente se presentan dos algoritmos basados en datos: el algoritmo basado en la función de valor de acción y el algoritmo basado en la política. Finalmente, se presenta un algoritmo que combina los dos anteriores, denominado algoritmo actor-crítico.

Algoritmos Basados en Funciones de Valor de Estado

Para construir la política óptima, el agente debe tener la noción de las recompensas que espera recibir en los estados futuros, hasta la conclusión del episodio. Esta cantidad se conoce como el *valor* de un estado.

Definición 3.6 *El **valor** V de un estado s indica el valor esperado de la recompensa acumulada descontada al seguir la política π desde s hasta $s_{\mathbf{T}}$. Matemáticamente se expresa como:*

$$V_\pi(s) = \sum_{a \in A} \pi(a \mid s) \sum_{s_{t+1} \in S} \mathcal{T}\big(s_{t+1} \mid s, a\big)\Big[R(s, a, s_{t+1}) + \gamma V_\pi(s_{t+1})\Big], \tag{3.7}$$

donde γ es el factor de descuento, R la función de recompensa, y $\mathcal{T}$ la función de transición del MDP.

En otras palabras, la *función de valor* $V_\pi(s)$ es una estimación de qué tan deseable es un estado s para el agente que sigue una política π. Note que diferentes políticas podrían tener diferentes valores de $V_\pi(s)$ para el mismo estado. En este sentido, una política π se dice que es mejor que otra política π' si su recompensa esperada es mayor que la de π' para todos los estados. Es decir, $\pi > \pi'$ si y solo $V_\pi(s) > V_{\pi'}(s)$ para todo $s \in S$. La *función de valor óptima* especifica la mayor recompensa esperada para cada estado por cualquier política, es decir:

$$V^*(s) = \max_\pi V_\pi(s), \ \forall s \in S. \tag{3.8}$$

La función de valor óptima no depende de la política, por tanto puede expresarse directamente de la Ec. (3.7) al aplicar el operador de maximización, de la siguiente manera:

$$V^*(s) = \max_a \left[\sum_{s_{t+1}} \mathcal{T}\big(s_{t+1} \mid s, a\big)\Big[R(s, a, s_{t+1}) + \gamma V^*(s_{t+1})\Big]\right]. \tag{3.9}$$

Estos valores no solo indican la mejor recompensa que el agente puede obtener, sino que básicamente dictan la política óptima para obtener esa recompensa. Si el agente conoce el valor de cada estado, puede seleccionar la acción con la recompensa máxima esperada, que es la suma de la recompensa inmediata y la recompensa a largo plazo descontada. De esta manera, es posible obtener la política óptima π^* al seleccionar la acción a que sea *codiciosa* (i.e., la acción que maximice el valor inmediato) con respecto a V^*, es decir:

$$\pi^*(s) = \arg\max_a \left[\sum_{s_{t+1}} \mathcal{T}\big(s_{t+1} \mid s, a\big) \Big[R(s, a, s_{t+1}) + \gamma V^*(s_{t+1}) \Big] \right]. \tag{3.10}$$

Derivar la política óptima a partir de la Ec. (3.10) requiere el conocimiento del modelo completo del MDP (i.e., estados futuros, recompensas y transiciones). Generalmente, el modelo del MDP no se conoce, por lo que se utiliza la interacción del agente con el entorno para estimar el valor de los estados y la probabilidad de transición entre estos. No obstante, la política óptima presenta un comportamiento determinista basado en la estimación del valor de los estados. Si la estimación no es una representación fiel, la política podría no elegir la acción correcta. Más aún, la política podría nunca seleccionar la mejor la acción si su estimación indica que la recompensa esperada para esta acción es menor que para las otras acciones. No elegir nunca esta acción implica que la estimación del valor no se actualizará, y por consiguiente, esta acción seguirá siendo no elegida constantemente.

Una estrategia simple de exploración que resuelve este problema, es la denominada estrategia ϵ-greedy (o exploración epsilon-greedy), que consiste en elegir una acción aleatoria con una probabilidad ϵ y la mejor con una probabilidad $1 - \epsilon$, es decir:

$$a_t = \begin{cases} \pi^*(s_t) & \text{con probabilidad } 1 - \epsilon, \\ a \sim \mathcal{U}(A) & \text{con probabilidad } \epsilon. \end{cases} \tag{3.11}$$

Note que si $\epsilon = 0$, la estrategia ϵ-greedy es equivalente a una estrategia puramente codiciosa (i.e., el agente explota el conocimiento adquirido); y si $\epsilon = 1$, se realiza una estrategia de selección aleatoria de acciones (i.e., el agente explora el entorno para recabar más información). En la práctica, es común que se utilice un valor de $\epsilon = 0.1$, lo que permite beneficiarse de la explotación del conocimiento acumulado y a la vez explorar el entorno con una probabilidad del 10 % (Lapan, 2018).

Algoritmos Basados en Funciones de Valor de Acción

Note que V^* expresa la recompensa máxima esperada para cada estado, pero no indica la acción óptima en cada uno. Para derivar la acción óptima a partir del estado actual, el agente debe simular todas las posibles acciones y elegir aquella que lleve al estado con mayor valor. Es decir, el agente obtiene el valor de la acción a en el estado s, lo que se conoce como el *valor del par estado-acción* (también se conoce simplemente como el *valor de la acción*).

Definición 3.7 *El **valor** Q de un par estado-acción (s, a) indica el valor esperado de la recompensa acumulada descontada al tomar la acción a en el estado s y seguir la política π desde s_{t+1} hasta $s_{\mathbf{T}}$. Matemáticamente se expresa como:*

$$Q_\pi(s,a) = \sum_{s_{t+1} \in S} \mathcal{T}\big(s_{t+1} \mid s, a\big)\Big[R(s,a,s_{t+1}) + \gamma V_\pi(s_{t+1})\Big]. \tag{3.12}$$

La función de valor de acción almacena de manera efectiva los resultados de todas las búsquedas de un paso adelante, proceso adicional requerido para derivar π^* a partir de V^*. La relación entre la función de valor de acción y la función de valor es:

$$V_\pi(s) = \max_a Q_\pi(s,a). \tag{3.13}$$

Similarmente a la Ec. (3.9), la *función de valor de acción óptima* Q^* puede expresarse directamente de la Ec. (3.12) al aplicar el operador de maximización en lugar de la expectativa, de la siguiente manera:

$$Q^*(s,a) = \sum_{s_{t+1}} \mathcal{T}(s_{t+1} \mid s, a)\left[R(s,a,s_{t+1}) + \gamma \max_\pi V_\pi(s_{t+1})\right]. \tag{3.14}$$

Al sustituir las Ec. (3.8) y (3.13) en la Ec. (3.14) se obtiene la función de valor de estado-acción óptima en forma recursiva:

$$Q^*(s,a) = \sum_{s_{t+1}} \mathcal{T}(s_{t+1} \mid s, a)\left[R(s,a,s_{t+1}) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})\right]. \tag{3.15}$$

La función de valor de estado-acción permite encontrar la acción óptima simplemente seleccionando la acción que maximice $Q^\pi(s,a)$. Así, la política óptima se obtiene como:

$$\pi^*(s) = \arg\max_a Q^*(s,a). \tag{3.16}$$

Al igual que la política óptima basada en la función de valor, la política óptima basada en la función de valor de acción es determinista, y las mismas estrategias de exploración son aplicables en este caso. Así, es posible obtener una estimación de Q a través de la interacción con el entorno, de la siguiente manera (van Hasselt et al., 2016):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\Big], \tag{3.17}$$

donde α es un hiperparámetro que indica la velocidad de aprendizaje, y controla la estabilidad del proceso. Un valor de $\alpha = 0.85$ se considera adecuado para la mayoría de aplicaciones (Even-Dar et al., 2003). Esta técnica de estimación de Q se conoce como *bootstrapping*, lo que se puede traducir como "comenzar con los recursos existentes

y mejorarlos conforme avanza el aprendizaje" (Jeong & Kim, 2019).

La función de valor $Q(s, a)$ es más conveniente en la práctica, ya que el proceso de toma de decisiones es más simple que al utilizar $V(s)$. En el caso de $Q(s, a)$, para elegir la acción óptima, el agente solamente necesita calcular el valor de acción para todas las acciones disponibles en el estado actual, y elegir la acción con el mayor valor de Q. Para realizar el proceso equivalente usando los valores únicamente de los estados, el agente necesita conocer no solo $V(s_{t+1})$, sino también las probabilidades de las transiciones. En la práctica, rara vez son conocidos de antemano, por lo que el agente necesita estimar las probabilidades de transición para cada par de acción y estado. Así, los métodos basados en la función de valor de acción $Q(s, a)$ son más populares que los que utilizan la función de valor de estado $V(s)$.

Algoritmos Basados en Políticas Parametrizadas

El algoritmo basado en el valor de la acción (i.e., *Q-learning*) ha sido un algoritmo popular en el aprendizaje por refuerzo debido a su simplicidad y garantías de convergencia (Watkins & Dayan, 1992). No obstante, este algoritmo solo funciona adecuadamente en entornos con espacios de estado y acción discretos y de baja dimensionalidad (Jiang et al., 2019).

Suponiendo que se aprende la función de valor de estado-acción Q en un entorno, la política óptima se obtiene directamente a través de la Ec. (3.16), siempre que el espacio de acción A sea discreto y finito. Sin embargo, tan pronto como el espacio de acción se vuelve continuo (o al menos lo suficientemente grande), esta selección de acción basada en valores deja de ser trivial. En espacios de acción continuos, se utilizan métodos que aprenden una *política parametrizada* que puede seleccionar acciones sin consultar una función de valor. Aún puede ser necesario utilizar una función de valor para aprender los parámetros de la política, pero no es necesaria para la selección de acciones (Sutton & Barto, 2018).

Definición 3.8 *Una **política parametrizada** es una función de estado que calcula la probabilidad de elegir la acción a en el tiempo t dado que el entorno está en un estado s y los parámetros están definidos por θ, esto es:*

$$\pi\big(a \,|\, s, \theta\big) = \Pr\big\{a_t = a \;\big|\; s_t = t,\, \theta_t = \theta\big\}, \tag{3.18}$$

donde θ es el conjunto de parámetros que caracterizan a la política. Para simplificar la notación, se utilizará $\pi_\theta(\cdot)$ para denotar que la política está parametrizada por θ.

El objetivo de estos algoritmos es optimizar el desempeño de la política en base a la recompensa esperada, es decir:

$$J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta}[\mathcal{G}(\tau)] = \mathop{\mathbb{E}}_{s,a \sim \pi_\theta}\left[\sum_{t=0}^{\mathbf{T}} \gamma^t r(s_t, a_t)\right]. \tag{3.19}$$

donde J es la función de desempeño a maximizar. Debido a que la trayectoria τ depende de las acciones elegidas; es decir, depende de la política, el símbolo $\sim$ denota que una trayectoria τ es muestreada (i.e., obtenida) a través de una política π con parámetros θ.

Así, la política óptima se obtiene a partir de los parámetros θ que maximizan la función de desempeño:

$$\pi^* = \arg\max_{\theta} J(\theta) = \arg\max_{\theta} \mathop{\mathbb{E}}_{s,a \sim \pi_\theta} \left[\sum_{t=0}^{\mathbf{T}} \gamma^t r(s_t, a_t) \right]. \tag{3.20}$$

Note que, en el caso de RL, optimizar la política implica maximizar la función de desempeño. Este proceso de optimización puede resolverse mediante distintos métodos como algoritmos genéticos (Sehgal et al., 2019), evolución diferencial (Tan & Li, 2021), enjambre de partículas (Liu et al., 2020a), etc. No obstante, los métodos basados en el gradiente tienen una mayor velocidad de convergencia, por lo que son los más utilizados (Khadka & Tumer, 2018). La regla de actualización de parámetros de la política, utilizando ascenso de gradiente, es la siguiente:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta_k), \tag{3.21}$$

donde θ_k son los parámetros θ en el paso de entrenamiento k, y $\nabla_\theta J(\theta_k)$ es el gradiente de J respecto a los parámetros θ en el instante k.

El *gradiente de la política* define la dirección en la que los parámetros deben actualizarse para mejorar su desempeño en términos de la recompensa total acumulada en la trayectoria. Esta dirección está determinada por la probabilidad de las acciones elegidas, y la escala del gradiente es proporcional a la recompensa recolectada en la trayectoria, es decir:

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\nabla_\theta \pi_\theta(\tau) \mathcal{G}(\tau) \right], \tag{3.22}$$

donde $\nabla_\theta \pi_\theta$ es el gradiente de la función $\pi(\cdot, \theta)$ respecto a los parámetros θ.

En otras palabras, se busca aumentar la probabilidad de trayectorias buenas (i.e., trayectorias con recompensas grandes) y disminuir la probabilidad de trayectorias malas (i.e., trayectorias con recompensas pequeñas o negativas).

Incluso con la política representada como una distribución de probabilidad, el agente puede converger a alguna política óptima local y dejar de explorar el entorno (Haarnoja et al., 2017). La solución propuesta en *Q-learning* fue utilizar la estrategia de selección de acción ϵ-greedy: con una probabilidad ϵ el agente toma alguna acción aleatoria en lugar de la acción dictada por la política actual. Esta estrategia también es aplicable en los algoritmos basados en políticas, sin embargo, estos admiten una mejor estrategia, denominada *bonificación de entropía* (Xin et al., 2020a).

En este sentido, la *entropía* es una métrica de la certeza de la idoneidad de la acción es decir, muestra qué tan seguro está el agente acerca de qué acción tomar. La entropía de la política se define como (Lapan, 2018):

$$H(\pi) = -\sum_{a \in A} \pi(a|s) \log(\pi(a|s)). \tag{3.23}$$

El valor de la entropía es siempre mayor que cero y tiene un único máximo cuando la política es uniforme; en otras palabras, cuando todas las acciones tienen la misma probabilidad de ser elegidas. En contraste, la entropía se vuelve mínima cuando la política tiene una probabilidad de 100 % para alguna acción y 0 % para todas las demás, lo que significa que el agente tiene una certeza total. Para evitar que el agente quede atrapado en mínimos locales, se añade la entropía a la función de desempeño:

$$J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\mathcal{G}(\tau) + H(\pi) \right]. \tag{3.24}$$

Así, se penaliza al agente por tener demasiada certeza en su política, y por tanto se beneficia la exploración.

Además de una mejor estrategia de exploración, una de las principales ventajas de los algoritmos basados en políticas parametrizadas es la habilidad de tratar con espacios de estado continuos, mientras que los algoritmos basados en funciones de valor están limitados a espacios discretos. No obstante, los algoritmos basados en políticas parametrizadas requieren episodios completos para la actualización de los parámetros, mientras que los algoritmos basados en funciones de valor pueden actualizarse en cualquier punto de la trayectoria. Finalmente, los algoritmos basados en política, al requerir trayectorias completas, presentan una alta varianza en la función de costo, lo que puede generar problemas de convergencia (Schulman et al., 2017). Esto se debe a que las trayectorias dependen tanto de los procesos estocásticos de la política, como del entorno. Por otro lado, los algoritmos basados en funciones de valor de acción presentan una varianza baja, al depender únicamente de la transición al siguiente estado.

Como se puede apreciar, ambas familias presentan ciertas ventajas y desventajas. Los algoritmos basados en políticas parametrizadas se prefieren en el caso de los espacios de estado continuos, mientras que los algoritmos basados en funciones de valor son más utilizados en el caso de los espacios discretos. Para obtener lo mejor de ambos, los algoritmos actor-crítico utilizan una combinación de políticas parametrizadas y aproximación de funciones de valor, como se explica a continuación.

Algoritmos Basados en Actor-Crítico

Los algoritmos *actor-crítico* parametrizan tanto las funciones de política, como las funciones de valor, y las actualizan simultáneamente en el entrenamiento. En este sentido, la política es denominada *actor*, ya que indica al agente la acción a realizar, y está parametrizada por θ. La función de valor se denomina *crítico*, pues permite al agente conocer el desempeño de la acción realizada, comparando su valor estimado con el resultado real de la acción. Esta última está parametrizada por ϕ.

En este algoritmo, la función de valor de acción no se utiliza para definir la política, como en el caso de los algoritmos basados únicamente en valores. En lugar de ello, la función de valor de acción define la magnitud del

gradiente de la función de costo. De esta manera, se puede sustituir la recompensa acumulada en la trayectoria, en la Ec. (3.22), por el valor de la acción en el estado actual (Lapan, 2018):

$$\nabla_\theta J(\theta) = \mathbb{E}_{s,a\sim\pi_\theta}\left[\nabla_\theta \pi_\theta(a_t|s_t)\hat{Q}(s_t,a_t)\right]. \tag{3.25}$$

Note que simplemente se sustituye la recompensa acumulada en la trayectoria $\mathcal{G}(\tau)$ por el valor estimado de la acción en el estado actual $\hat{Q}(s,a)$.

La optimización de los parámetros de la política sigue el mismo principio que en los algoritmos basados únicamente en política: se busca incrementar la probabilidad elegir acciones con un valor de estado-acción alto y disminuir la probabilidad de elegir acciones que estén asociadas con un valor de estado-acción bajo (o negativo). De igual manera, las estrategias de exploración descritas para los algoritmos basados únicamente en política pueden ser utilizadas para los algoritmos basados en actor-crítico.

Por otro lado, la optimización de los parámetros de la función de valor de acción tiene como objetivo reducir el error entre la estimación del valor de acción del estado actual y la estimación del valor de la acción en el estado siguiente más la recompensa en la transición. La *función de costo* asociada a los parámetros del crítico está definida como (Mnih et al., 2016):

$$\mathcal{L}(\phi) = \frac{1}{\mathbf{T}}\sum_{t=0}^{\mathbf{T}}\left[\left(r(s_t,a_t) + \gamma\hat{Q}(s_{t+1},a_{t+1}) - \hat{Q}(s_t,a_t)\right)^2\right]. \tag{3.26}$$

El objetivo es encontrar los parámetros ϕ que minimicen la función de costo, es decir:

$$\arg\min_\phi \mathcal{L}(\phi) = \arg\min_\phi \frac{1}{\mathbf{T}}\sum_{t=0}^{\mathbf{T}}\left[\left(r(s_t,a_t) + \gamma\hat{Q}(s_{t+1},a_{t+1}) - \hat{Q}(s_t,a_t)\right)^2\right]. \tag{3.27}$$

Note que, a diferencia del actor, el objetivo de optimización de los parámetros del crítico implica una *minimización* de la función de costo.

Finalmente, la regla de actualización de los parámetros del crítico, utilizando el gradiente de la función de costo, es la siguiente:

$$\phi_{k+1} = \phi_k - \alpha\nabla_\phi\mathcal{L}(\phi_k). \tag{3.28}$$

La ventaja de los métodos actor-crítico sobre los algoritmos basados únicamente en política es un proceso de entrenamiento más rápido y estable. En este sentido, los algoritmos basados únicamente en política utilizan un gradiente por trayectoria. Al no incorporar resultados de realizar una acción diferente en cada estado como punto de comparación, el gradiente estimado puede apuntar en la dirección equivocada. Mientras que en el caso del actor-crítico, el gradiente utilizado incorpora información de cada transición, lo que permite reforzar correctamente las acciones que tengan buenas recompensas. En consecuencia, los métodos actor-crítico pueden requerir muchas menos

interacciones con el entorno que los métodos basados únicamente en política para aprender de manera efectiva, y generalmente muestran mejores resultados empíricos que los métodos basados únicamente en valores y basados únicamente en políticas (Perera & Kamalaruban, 2021).

3.3. Aprendizaje Profundo por Refuerzo

En DRL se utilizan ANNs como representación de las funciones parametrizadas de política y de valor. Una ANN típica en aplicaciones de DRL está compuesta por dos capas ocultas de 64 neuronas cada una (más una entrada adicional que controla el umbral de activación de cada neurona). Para un entorno con M estados y N acciones, la cantidad de parámetros de la función de política es $(N+1) \times 64 + (64+1) \times 64 + (64+1) \times M$. Por ejemplo, para un entorno con doce entradas y tres acciones, la función está representada por 5,187 parámetros (Lapan, 2018).

La ventaja de utilizar ANNs en el esquema de RL es que permite utilizar directamente los conceptos de los algoritmos basados en el valor del estado, en entornos con espacios de estado y acción continuos, ya que elimina la necesidad de utilizar métodos tabulares para almacenar la experiencia del agente (Okafor et al., 2021).

3.3.1. Aproximación de Funciones con Redes Neuronales Artificiales

Las ANNs se utilizan en una amplia variedad de aplicaciones: aproximación de funciones, reconocimiento de patrones, agrupamiento, predicción, entre otras. A pesar de que su desarrollo comenzó en los años ochenta del siglo pasado, solo en los últimos años se han convertido en una tecnología estándar (Hagan et al., 1996). Esto se debe en gran medida a la potencia de cómputo actual de las computadoras y al desarrollo de bibliotecas de aprendizaje automático como TensorFlow[1] o PyTorch[2].

En el paradigma de DRL, la aproximación de funciones es una técnica de aprendizaje muy útil para la optimización de políticas. El *perceptrón multicapa* (MLP, *Multilayer Perceptron*) es la arquitectura de ANNs más común para este tipo de aplicaciones (Chen et al., 2015), ya que puede aproximar cualquier función continua a cualquier grado de precisión (Cybenko, 1989). Un MLP está conformado por, al menos, tres capas de nodos: una capa de entrada, una (o más) capa oculta y una capa de salida. Todas las capas están completamente conectadas con la siguiente capa mediante pesos, y no se permiten conexiones hacia atrás o con capas no adyacentes.

Para aprender la aproximación de una función $f(x)$, el entrenamiento estándar de las ANNs se realiza en un entorno *supervisado*: se introduce un valor de entrada x en la ANN, y la salida $\hat{f}(x, \mathbf{w})$ se compara con el resultado esperado $f(x) = y$. Los parámetros (i.e., pesos) $\mathbf{w}$ se ajustarán de tal manera que la aproximación a $f(x)$ sea lo más cercana posible. Usualmente se utiliza el *error cuadrático medio* (MSE, *Mean Squared Error*) como medida de la exactitud de la aproximación (Heidari et al., 2020):

[1] https://www.tensorflow.org/
[2] https://pytorch.org/

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \left[\hat{f}(x_i) - y_i \right]^2, \tag{3.29}$$

donde x_i es la entrada del ejemplo i, y_i la salida del ejemplo i, y N es el número total de ejemplos.

Los parámetros de la red $\mathbf{w}$ se ajustan para minimizar la función de costo $\mathcal{L}(\mathbf{w})$. Similarmente a la optimización de políticas parametrizadas, el proceso de optimización de los parámetros en una ANN se realiza mediante la técnica de descenso de gradiente: se calcula el gradiente de la función de costo con respecto al vector de pesos y se actualizan los pesos siguiendo la dirección negativa del gradiente (i.e., la dirección en la que la función de costo decae más rápidamente). Utilizar todos los ejemplos del entrenamiento para calcular el gradiente es una técnica muy eficiente, ya que se obtiene el gradiente verdadero, pero es muy costosa computacionalmente, por lo que rara vez se utiliza en la práctica (Ruder, 2016). En contraste, el método de *descenso de gradiente estocástico* (SGD, *Stochastic Gradient Descent*) utiliza una sola muestra para calcular el gradiente, lo que agiliza su velocidad de convergencia. Para SGD, la regla de actualización de los parámetros está dada como (Bottou, 2012):

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_k) = \mathbf{w}_k - \alpha \left[\hat{f}(x, \mathbf{w}_k) - y \right] \nabla_{\mathbf{w}} \hat{f}(x, \mathbf{w}_k). \tag{3.30}$$

Una desventaja que presenta SGD es que no converge realmente al óptimo, ya que la aproximación del gradiente con una sola muestra produce estimaciones ruidosas. En la práctica, se suele hacer uso de la combinación de ambas técnicas, lo que se denomina *SGD en mini lotes*. La función de costo se promedia sobre este mini lote, resultando en una mejor aproximación del gradiente. Esta técnica presenta la ventaja de converger al óptimo verdadero a una velocidad mayor que al utilizar el conjunto completo de ejemplos (Bengio, 2012). Para una buena convergencia, es necesario que las muestras sean independientes e idénticamente distribuidas (i. i. d.) sobre todo el espacio de posibles entradas, lo que se satisface al usar muestreo aleatorio (Liu et al., 2020b).

Existen otras técnicas de optimización más novedosas basadas en gradiente, tales como *estimación adaptativa del momento* (Adam, *Adaptive moment estimation*) (Kingma & Ba, 2014), *gradiente adaptativo* (AdaGrad, *Adaptive Gradient*) (Duchi et al., 2011), *propagación de la raíz cuadrática media* (RMSProp, *Root Mean Square Propagation*) (Hinton et al., 2012), *gradiente acelerado de Nesterov* (NAG, *Nesterov's Accelerated Gradient*) (Sutskever et al., 2013), entre otros. Estos métodos se basan en el mismo principio que SGD por mini lotes, con algunas diferencias de implementación (Dogo et al., 2018).

En algoritmos de DRL, las ANNs se utilizan para aproximar la política del actor, así como la estimación del valor de estado del crítico.

3.3.2. Algoritmo de Gradiente de Política Determinista Profunda (DDPG)

La integración de ANNs al paradigma de RL, permitió el desarrollo de métodos actor-crítico directamente aplicables en entornos con espacios de estado y acción continuos. Uno de ellos, denominado algoritmo de *gradien-*

te de política determinista profunda (DDPG, *Deep Deterministic Policy Gradient*) (Lillicrap et al., 2019), está compuesto por dos ANNs, una que representa la política y otra que representa el crítico. La política está representada por el *actor* $\mu(s|\theta^\mu) \mapsto A$, parametrizado por θ^μ. La función de valor de acción está representada por el *crítico* $Q(s, a|\theta^Q) \mapsto \mathbb{R}$, parametrizado por θ^Q.

A diferencia de los algoritmos anteriores de gradiente de política, que generan una distribución de probabilidad para la selección de acciones que luego se muestrean para obtener la acción real, el actor DDPG produce acciones deterministas que se alimentan directamente al entorno (van Hasselt, 2012). La evaluación de la acción por parte del crítico se utiliza posteriormente para mejorar la política del actor. Este algoritmo es relativamente simple de implementar y se destaca por ser robusto con respecto a sus hiperparámetros, como lo demuestra Lillicrap et al. (2019), donde se utiliza el algoritmo DDPG en más de 20 tareas diversas con la misma configuración de hiperparámetros y obtuvo un buen rendimiento general para todas ellas.

Una de las ventajas que presenta DDPG sobre otros métodos actor-crítico es que puede ser entrenado *off-policy*; es decir, el entrenamiento puede realizarse con datos recolectados por otras políticas. Contrario a esto, los algoritmos *on-policy* solo pueden ser entrenados con datos generados por la política actual, y en cada actualización de esta, nuevos datos deben ser generados mediante nuevas interacciones con el entorno. Así, en DDPG las experiencias resultantes de la interacción se almacenan en un *búfer de reproducción* $\mathcal{R}$. La i-ésima experiencia almacenada en $\mathcal{R}$ es la tupla (s_i, a_i, r_i, s_{i+1}) de estado, acción, recompensa y siguiente estado. Durante el entrenamiento, se realiza un muestreo aleatorio de mini lotes del búfer $\mathcal{R}$ y se utilizan como ejemplos de entrenamiento para optimizar los parámetros de las ANNs del crítico y del actor.

El objetivo del crítico es aproximar la función de valor de acción, por lo que este enfoque está estrechamente relacionado con *Q-learning* y se basa en el mismo principio: si conoce la función de valor de acción óptima $Q^*(s, a)$, entonces, en cualquier estado dado, la acción óptima $a^*(s)$ se puede encontrar resolviendo:

$$a^*(s) = \arg\max_{a \in A} Q^*(s, a). \tag{3.31}$$

En el caso discreto, este proceso de optimización es relativamente simple: se calculan los valores Q para cada acción y se selecciona la acción con mayor valor. No obstante, en el caso continuo, no es posible evaluar exhaustivamente el espacio de acción, y utilizar otras técnicas de optimización resulta inviable, ya que este proceso necesita realizarse en cada iteración, lo cual puede convertirse en un procedimiento exhaustivo (Lillicrap et al., 2019).

Para solucionar este problema, en DDPG se asume que la función $Q^*(s, a)$ es diferenciable con respecto al argumento de acción, lo que permite establecer una regla de aprendizaje basada en gradientes para la política y para la función de valor de acción. Entonces, en lugar de ejecutar una costosa subrutina de optimización cada vez que desea calcular $\max_a Q(s, a)$, se puede aproximar con $\max_a Q(s, a) \approx Q(s, \mu(s))$. Así, la optimización de los parámetros del crítico se realiza mediante el procedimiento estándar de aprendizaje supervisado, descrito en la Ec. (3.29), donde la estimación es la ANN evaluada en el estado actual, y la salida esperada (i.e., objetivo) es la

función descontada de valor de acción estimada en el siguiente estado más la recompensa inmediata (Hou et al., 2017):

$$\mathcal{L}(\theta^Q) = \mathbb{E}\left[\Big(f(s,a) - y(r,s')\Big)^2\right] = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{R}}\left[\left(Q(s,a|\theta^Q) - \Big(r + \gamma \max_{a'} Q(s',a'|\theta^Q)\Big)\right)^2\right], \tag{3.32}$$

donde a' y s' son las acciones y estados siguientes, respectivamente.

Suponiendo que el actor μ es óptimo, es decir $\max_a Q(s,a) \approx Q\left(s,\mu(s)\right)$, entonces la función de pérdida del crítico se puede expresar como (Lillicrap et al., 2019):

$$\mathcal{L}(\theta^Q) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{R}}\left[\left(Q(s,a|\theta^Q) - \Big(r + \gamma Q\big(s',\mu(s')\big)|\theta^Q,\theta^\mu\Big)\right)^2\right]. \tag{3.33}$$

Previo al trabajo presentado por Mnih et al. (2015), el entrenamiento de ANNs para aprender funciones de valor era un proceso inestable, debido a dos problemas principales. Primero, los datos utilizados para el entrenamiento supervisado deben ser independientes e idénticamente distribuidos, sin embargo, las transiciones en el entorno tienen una alta correlación entre ellas. Segundo, la salida esperada, en la función de costo descrita en la Ec. (3.33), depende de los mismos parámetros a optimizar, lo que implica que el objetivo está siempre en movimiento. Para solucionar estos problemas se utiliza el búfer de reproducción $\mathcal{R}$, y las *redes de objetivo* Q' y μ'.

Al muestrear experiencias aleatorias del búfer de reproducción se rompe la correlación entre las experiencias en las trayectorias. Mientras que las redes de objetivo mantienen fija la función de valor de acción que se utiliza para calcular el valor de la función de costo, lo que hace el proceso de entrenamiento más estable. Los parámetros de las redes de objetivo no son actualizados mediante el gradiente, en cambio se actualizan mediante un promedio de Polyak respecto a los parámetros de las redes principales

$$\theta' \leftarrow \rho\theta + (1-\rho)\theta', \tag{3.34}$$

donde $0 \leq \rho \leq 1$ es la velocidad de actualización de los parámetros de las redes de objetivo. En el artículo original (Lillicrap et al., 2019) se recomienda usar $\rho = 0.001$.

Al incluir las redes de objetivo en la Ec. (3.33), la nueva función de costo se expresa como:

$$\mathcal{L}(\theta^Q) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{R}}\left[\left(Q(s,a|\theta^Q) - \Big(r + \gamma Q'\big(s',\mu'(s')\big)|\theta^{Q'},\theta^{\mu'}\Big)\right)^2\right]. \tag{3.35}$$

Note que la estimación y la salida esperada ya no dependen de los mismos parámetros. Así, los parámetros del crítico son optimizados mediante el algoritmo de descenso de gradiente de tal forma que minimicen esta función de costo. La regla de actualización de los parámetros del crítico es:

$$\theta^Q_{k+1} = \theta^Q_k - \alpha_Q \nabla_{\theta^Q}\mathcal{L}(\theta^Q_k), \tag{3.36}$$

donde α_Q es el factor de aprendizaje del crítico, k es el número de iteraciones, y θ_k^Q son los parámetros del crítico en la iteración k. Un factor de aprendizaje recomendado para el crítico es $\alpha_Q = 0.001$ (Islam et al., 2017).

Respecto al entrenamiento del actor, el objetivo de la optimización es maximizar la expectativa de retorno cuando se sigue la política definida por los parámetros de la red. Esta función de desempeño se expresa como (Eckstein, 2020):

$$J(\theta^\mu) = \mathbb{E}\left[Q(s,a)\right] = \mathbb{E}_{s\sim\mathcal{R}}\left[Q\left(s,\mu(s|\theta^\mu)\right)\right]. \tag{3.37}$$

El gradiente de la función de desempeño indica la dirección de actualización de los parámetros del actor, de tal forma que maximice $Q(s,a)$. Debido a que $a = \mu(s)$, la función Q es una función compuesta, por lo que el gradiente es calculado utilizando la regla de la cadena (Lillicrap et al., 2019), es decir:

$$\nabla_{\theta^\mu} J(\theta^\mu) = \mathbb{E}_{s\sim\mathcal{R}}\left[\nabla_{\theta^\mu} Q(s,a|\theta^Q)\right] = \mathbb{E}_{s\sim\mathcal{R}}\left[\nabla_a Q(s,a|\theta^Q)\nabla_{\theta^\mu}\mu(s|\theta^\mu)\right]. \tag{3.38}$$

Intuitivamente, el gradiente de la función se puede dividir en dos partes. El gradiente $\nabla_a Q$ indica la dirección en la que debe moverse la acción para maximizar el valor de la función de acción, mientras que el gradiente $\nabla_{\theta^\mu}\mu$ indica la dirección en la que deben actualizarse los parámetros de la red para maximizar la acción. Al multiplicar ambos gradientes se obtiene la dirección en la que los parámetros deben moverse para maximizar Q.

Así, la regla de actualización de los parámetros del actor es:

$$\theta_{k+1}^\mu = \theta_k^\mu + \alpha_\mu \nabla_{\theta^\mu} J(\theta_k^\mu), \tag{3.39}$$

donde α_μ es el factor de aprendizaje del actor, k es el número de iteraciones, y θ_k^μ son los parámetros del actor en la iteración k. Un factor de aprendizaje recomendado para el actor es $\alpha_\mu = 0.0001$ (Islam et al., 2017).

Al observar las ecuaciones, es evidente que la clave para optimizar la política es una buena aproximación de los valores de acción por parte del crítico, ya que este es el elemento determinante en el gradiente de política utilizado para actualizar al actor.

En cuanto a la exploración del espacio de estado y acción, al ser una política determinista, DDPG requiere una estrategia de exploración explícita. De manera similar a una exploración ϵ-greedy de Q-learning para espacios discretos, en el caso continuo se utiliza una *exploración indirecta*, la cual consiste en agregar un componente de ruido aleatorio, muestreado de un proceso $\mathcal{N}$, a la acción elegida por el actor (Lillicrap et al., 2019), es decir:

$$a_t = \mu(s_t|\theta_t^\mu) + \mathcal{N}_t, \tag{3.40}$$

donde $\mathcal{N}$ se elige de acuerdo al espacio de acción del entorno.

El algoritmo DDPG se presenta en el Algoritmo 1 y se detalla a continuación. Se inicializan dos redes neuronales,

Algoritmo 1: Algoritmo de gradiente de política determinista profunda (DDPG)

Entrada: Tamaño de mini lote N, número de episodios M, factor de descuento γ, tasa de aprendizaje del crítico α_Q, tasa de aprendizaje del actor α_μ, factor de promedio de Polyak ρ

Salida: Actor entrenado $\mu(s|\theta^\mu)$

Inicializar aleatoriamente las redes neuronales $Q(s, a|\theta^Q)$ y $\mu(s|\theta^\mu)$ con pesos θ^Q y θ^μ, respectivamente

Inicializar un búfer de repetición vacío $\mathcal{R}$

Inicializar las redes de objetivo Q', y μ'

$\theta^{Q'} \leftarrow \theta^Q$

$\theta^{\mu'} \leftarrow \theta^\mu$

para *episodio = 1 hasta M* **hacer**

  Inicializar un proceso aleatorio $\mathcal{N}$ para la exploración de acciones

  Recibir el estado inicial s_0

  para *paso t = 0 hasta T* **hacer**

    Seleccionar acción $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}$ de acuerdo a la política actual y al ruido de exploración

    Ejecutar a_t y observar la recompensa r_{t+1} y el nuevo estado s_{t+1}

    Guardar la transición $(s_t, a_t, r_{t+1}, s_{t+1})$ en el búfer de repetición $\mathcal{R}$

    Muestrear un mini lote de N transiciones $(s_t, a_t, r_{t+1}, s_{t+1})$ del búfer de repetición $\mathcal{R}$

    Hacer $y_i = r_{i+1} + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'}|\theta^{Q'}))$

    Calcular la función de costo del crítico, (3.35):

$$\mathcal{L}(\theta^Q) = \frac{1}{N}\sum_i^N \left(y_i - Q(s, a|\theta^Q)\right)^2$$

    Actualizar los parámetros del crítico para minimizar la función de costo, (3.36):

$$\theta^Q_{k+1} = \theta^Q_k - \alpha_Q \nabla^Q_\theta \mathcal{L}(\theta^Q_k)$$

    Calcular la función de desempeño del actor, Ec. (3.37), y su gradiente, (3.38):

$$J(\theta^\mu) = \mathbb{E}\left[Q\left(s, \mu(s|\theta^\mu)\right)\right]$$

$$\nabla_{\theta^\mu} J(\theta^\mu) = \mathbb{E}\left[\nabla_a Q(s, a|\theta^Q) \nabla_{\theta^\mu} \mu(s|\theta^\mu)\right].$$

    Actualizar los parámetros del actor para maximizar la función de desempeño, (3.39):

$$\theta^\mu_{k+1} = \theta^\mu_k + \alpha_\mu \nabla_{\theta^\mu} J(\theta^\mu_k)$$

    Actualizar los parámetros de las redes objetivo, (3.34):

$$\theta^{Q'} \leftarrow \rho\theta^Q + (1-\rho)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \rho\theta^\mu + (1-\rho)\theta^{\mu'}$$

devolver $\mu(\cdot)$

una para trabajar como actor, $\mu(s|\theta^{\mu})$, y la segunda para trabajar como crítico, $Q(s,a|\theta^{Q})$. Se crean dos copias de estas redes, que funcionarán como el actor objetivo y el crítico objetivo, μ' y Q', respectivamente. Se crea un búfer de reproducción vacío $\mathcal{R}$ para almacenar las experiencias del agente al interactuar con el entorno (estados s, acciones a, recompensas r_{t+1}, siguientes estados s_{t+1}). Cada interacción entre el agente y el entorno cuenta como un paso de tiempo, representado por t. Una colección de un número dado de pasos de tiempo $\mathbf{T}$ se llama episodio. El algoritmo DDPG se repite durante un número deseado de episodios M. Al comienzo de cada episodio, se inicializa un proceso aleatorio $\mathcal{N}$ para la exploración del espacio de estado-acción, y el agente recibe la información del estado inicial s_0. Para cada paso de tiempo, se calcula una acción a través de la ANN del actor y se agrega un componente de ruido de exploración (las acciones ruidosas permiten al agente explorar el espacio de estado más allá de su política, lo que conduce a recompensas futuras, probablemente, más altas). Esta acción se ejecuta en el entorno, que devuelve el siguiente estado y la recompensa por la acción. La experiencia resultante (s, a, r_{t+1}, s_{t+1}) se almacena en el búfer de reproducción. Se muestrea un lote de N experiencias del búfer de reproducción para entrenar ambas redes. El entrenamiento consta de los siguientes pasos: el crítico Q evalúa los estados y acciones muestreados, y actualiza sus parámetros θ^{Q} un paso en la dirección opuesta del gradiente respecto del valor objetivo, obtenido de las redes μ' y Q'; los parámetros del actor θ^{μ} se actualizan en la dirección del gradiente que maximice el valor del crítico. Los parámetros de las redes actor y crítico se copian en las redes objetivo, μ' y Q', respectivamente, en función del factor de promedio de Polyak ρ. Siguiendo este proceso, la ANN del actor tiende a aumentar la probabilidad de sugerir acciones que serán mejor evaluadas por el crítico, lo que conducirá a mayores recompensas y a un mejor desempeño del agente.

3.3.3. Algoritmo de Gradiente de Política Determinista Profunda con Retraso Gemelo (TD3)

Un problema en DDPG es que la función Q aprendida puede sobreestimar los valores de acción, lo que conduce al aprendizaje de políticas que explotan estos errores en la estimación. Si bien la política aprendida es óptima respecto a la función Q aprendida, es sub-óptima respecto a los valores reales de la función de acción. El algoritmo de TD3 (Fujimoto et al., 2018) aborda este problema de sobreestimación con tres características clave:

1. Agregar ruido a las acciones del actor durante el entrenamiento.
2. Usar un par de redes críticas (la parte *gemelo* en el título).
3. Actualizar con menos frecuencia los parámetros del actor (la parte *retraso* en el título).

La primera característica de TD3 es la adición de ruido en las acciones calculadas por el actor objetivo, las cuales se utilizan para calcular el objetivo en la actualización de los críticos. Esta característica se denomina *suavizado de la política objetivo* y es una técnica de regularización utilizada para evitar variaciones en la función de valor de

acción. Idealmente, para acciones similares en el mismo estado, los valores Q deberían ser también similares. TD3 reduce la variación entre el valor de acciones similares al agregar una señal pequeña de ruido aleatorio al objetivo y promediarlo sobre mini lotes (Zhou et al., 2021). Para calcular las acciones con ruido $\tilde{a}$ se utilizan dos subprocesos. Primero se muestrea el componente de ruido ε de una distribución normal con media cero y varianza σ^2, se recorta para limitarlo entre el intervalo $[-c, c]$ y se suma a las acciones. Segundo, la acción con ruido se recorta para limitarla a los límites del entorno, definidos por $[a_{\text{Min}},\, a_{\text{Max}}]$:

$$\tilde{a}' = \text{clip}\big(\mu'(s') + \text{clip}\left(\varepsilon, -c, c\right), a_{\text{mín}}, a_{\text{máx}}\big), \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{3.41}$$

donde $\tilde{a}'$ es la acción a tomar en el estado siguiente más el componente de ruido, y $\text{clip}(\varepsilon, -c, c)$ es el equivalente a $\text{máx}(\text{mín}(\varepsilon, c), -c)$.

La segunda característica añadida a TD3 es el uso de dos redes críticas. Ambos críticos se entrenan de la misma manera que el único crítico en DDPG, con una diferencia: para calcular el objetivo se utiliza cualquiera de las dos funciones Q' que proporcione el valor más pequeño:

$$y(r, s') = r + \gamma \min_{i=1,2} Q'_{\theta_i}\left(s', \mu'(s'|\theta^{\mu'})|\theta_i^{Q'}\right), \tag{3.42}$$

y ambos críticos se actualizan conforme a la minimización de su función de pérdida correspondiente:

$$\mathcal{L}(\theta_1^Q) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{R}}\left[\left(Q_{\theta_1}(s, a|\theta_1^Q) - y(r, s')\right)^2\right], \tag{3.43}$$

$$\mathcal{L}(\theta_2^Q) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{R}}\left[\left(Q_{\theta_2}(s, a|\theta_2^Q) - y(r, s')\right)^2\right]. \tag{3.44}$$

Usar el valor de acción más pequeño para el objetivo y optimizar los parámetros hacia él ayuda a evitar la sobreestimación en la función Q (Shehab et al., 2021).

La tercera característica de TD3 es el retraso de la actualización de los parámetros del actor. Esto permite que los críticos se vuelvan más estables y reduzcan los errores antes de que se usen para actualizar al actor (Zhang et al., 2020). En la práctica, la política se actualiza después de un determinado número de iteraciones, mientras que los críticos se actualizan en cada iteración. La actualización de los parámetros en la política sigue el mismo principio que en DDPG, maximizar la expectativa de retorno cuando se sigue la política definida por los parámetros de la red:

$$J(\theta^\mu) = \mathbb{E}_{s\sim\mathcal{R}}\left[Q_{\theta_1}(s, \mu(s)|\theta^\mu)\right]. \tag{3.45}$$

Estas actualizaciones menos frecuentes en los parámetros de la política, estabilizan el entrenamiento, ya que la

política influye en el cálculo del objetivo para la actualización de los parámetros de los críticos. Debido a que ambos críticos se optimizan conforme al mismo valor objetivo, es indiferente usar uno u otro crítico para actualizar los parámetros del actor (Dankwa & Zheng, 2019).

En cuanto a la estrategia de exploración, se utiliza la misma estrategia que en DDPG; es decir, se agrega ruido a las acciones en el momento del entrenamiento:

$$a_t = \mu(s_t|\theta_t^\mu) + \mathcal{N}_t, \tag{3.46}$$

donde $\mathcal{N}$ se elige de acuerdo al espacio de acción del entorno. Note que el ruido de exploración y el ruido de regularización son independientes.

Al estar basado en DDPG, el algoritmo TD3 tiene una estructura muy similar a éste, y se presenta en el Algoritmo 2. A continuación se detalla el proceso de entrenamiento. Se inicializan tres redes neuronales, una para trabajar como actor, $\mu(s|\theta^\mu)$, y las otras dos funcionarán como críticos, $Q_{\theta_1}(s,a|\theta_1^Q)$, $Q_{\theta_2}(s,a|\theta_2^Q)$. Al igual que DDPG, cada una de estas redes principales tiene su respectiva copia, Q'_{θ_1}, Q'_{θ_2} y μ', cuya función es un objetivo fijo durante el proceso de optimización de los pesos. Se crea un búfer de reproducción vacío $\mathcal{R}$ para almacenar las experiencias del agente al interactuar con el entorno (estados s, acciones a, recompensas r_{t+1}, siguientes estados s_{t+1}). Cada interacción entre el agente y el entorno cuenta como un paso de tiempo, representado por t. Una colección de un número dado de pasos de tiempo $\mathbf{T}$ se llama episodio. El algoritmo TD3 se repite durante un número deseado de episodios M. Al comienzo de cada episodio, se inicializa un proceso aleatorio $\mathcal{N}$ para la exploración del espacio de estado-acción, y el agente recibe la información del estado inicial s_0. Para cada paso de tiempo, se calcula una acción a través de la ANN del actor y se agrega un componente de ruido de exploración (las acciones ruidosas permiten al agente explorar el espacio de estado más allá de su política, lo que conduce a recompensas futuras, probablemente, más altas). Esta acción se ejecuta en el entorno, que devuelve el siguiente estado y la recompensa por la acción. La experiencia resultante (s, a, r_{t+1}, s_{t+1}) se almacena en el búfer de reproducción. Se muestrea un lote de N experiencias del búfer de reproducción para entrenar las redes. Para suavizar la política (i.e., prevenir sobreestimación de la función de valor), a las acciones muestreadas del búfer se les añade un componente de ruido, las cuales son usadas para entrenar a los críticos. Ambos críticos calculan el valor de acción objetivo y se selecciona el mínimo de los dos para realizar la regresión, donde los parámetros de ambos críticos, θ_1^Q y θ_2^Q, se optimizan para minimizar su correspondiente función de pérdida. Los parámetros del actor θ^μ se actualizan una vez por cada ψ actualizaciones de los críticos, en la dirección del gradiente que maximice el valor del crítico Q_{θ_1} (ya que los críticos se entrenan con el mismo valor objetivo, ambos presentarán estimaciones similares). Los parámetros de las redes actor y críticos se copian en las redes objetivo, μ', Q'_{θ_1} y Q'_{θ_2}, respectivamente, usando el factor de promedio de Polyak ρ para realizar la actualización de los parámetros. Siguiendo este proceso, la ANN del actor tiende a aumentar la probabilidad de sugerir acciones que serán mejor evaluadas por los críticos, lo que conducirá a mayores recompensas y a un mejor desempeño del agente.

Algoritmo 2: Algoritmo de gradiente de política determinista profunda con retraso gemelo (TD3)

Entrada: Tamaño de mini lote N, número de episodios M, factor de descuento γ, tasa de aprendizaje del crítico α_Q, tasa de aprendizaje del actor α_μ, factor de promedio de Polyak ρ, ruido de exploración $\mathcal{N}$, varianza del ruido de regularización σ^2, pasos de retardo de actualización de la política ψ

Salida: Actor entrenado $\mu(s|\theta^\mu)$

Inicializar aleatoriamente las redes neuronales $Q_{\theta_1}(s,a|\theta_1^Q)$, $Q_{\theta_2}(s,a|\theta_2^Q)$ y $\mu(s|\theta^\mu)$ con pesos θ_1^Q, θ_2^Q y θ^μ, respectivamente

Inicializar un búfer de repetición vacío $\mathcal{R}$

Inicializar las redes de objetivo Q'_{θ_1}, Q'_{θ_2} y μ'

$\theta_1^{Q'} \leftarrow \theta_1^Q$

$\theta_2^{Q'} \leftarrow \theta_2^Q$

$\theta^{\mu'} \leftarrow \theta^{\mu}$

para *episodio = 1 hasta M* **hacer**

  Inicializar un proceso aleatorio $\mathcal{N}$ para la exploración de acciones

  Recibir el estado inicial s_0

  para *paso t = 0 hasta T* **hacer**

    Observar estado s y seleccionar acción de acuerdo a (3.46):

$$a = \mu(s|\theta^\mu) + \mathcal{N}$$

    Ejecutar a_t y observar la recompensa r_{t+1} y el nuevo estado s_{t+1}

    Guardar la transición $(s_t, a_t, r_{t+1}, s_{t+1})$ en el búfer de repetición $\mathcal{R}$

    Muestrear un mini lote de N transiciones $(s_t, a_t, r_{t+1}, s_{t+1})$ del búfer de repetición $\mathcal{R}$

    Calcular las acciones objetivo de acuerdo a (3.41):

$$\tilde{a}' = \text{clip}\big(\mu'(s') + \text{clip}\left(\varepsilon, -c, c\right), a_{\min}, a_{\max}\big), \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

    Calcular los valores de acción objetivo, (3.42):

$$y(r, s') = r + \gamma \min_{i=1,2} Q'_{\theta_i}\left(s', \mu'(s'|\theta^{\mu'})|\theta_i^{Q'}\right)$$

    Actualizar los parámetros del crítico para minimizar la función de pérdida, (3.43) y (3.44):

$$\mathcal{L}(\theta_j^Q) = \frac{1}{N}\sum_i^N \left[\left(Q_{\theta_j}(s,a|\theta_j^Q) - y(r,s')\right)^2\right], \qquad j = 1,2$$

    si $t \bmod \psi = 0$ **entonces**

      Actualizar los parámetros del actor para maximizar la función de desempeño, (3.45):

$$J(\theta^\mu) = \frac{1}{N}\sum_i^N \left[Q_{\theta_1}(s, \mu(s)|\theta^\mu)\right]$$

      Actualizar los parámetros de las redes objetivo, (3.34):

$$\theta_j^{Q'} \leftarrow \rho\theta_j^Q + (1-\rho)\theta_j^{Q'}, \qquad j = 1,2$$
$$\theta^{\mu'} \leftarrow \rho\theta^\mu + (1-\rho)\theta^{\mu'}$$

devolver $\mu(\cdot)$

3.3.4. Algoritmo TD3 con Demostraciones Expertas (TD4)

El algoritmo TD3 no utiliza ningún conocimiento previo y explora constantemente a través de la interacción con el entorno para obtener la política óptima. Sin embargo, el aprendizaje puede requerir una gran cantidad de tiempo, dependiendo de la frecuencia de las interacciones con el entorno, la complejidad de las funciones a aproximar y la dimensionalidad del espacio de estado-acción (Zhang et al., 2020).

Para realizar la exploración del espacio estado-acción, TD3 utiliza las acciones calculadas por la política y agrega un componente de ruido aleatorio $a = \pi(s) + \mathcal{N}$. No obstante, la exploración aleatoria en espacios continuos no es una solución eficiente. En su lugar, es posible realizar una exploración guiada por las demostraciones de un *agente experto*, lo que permite acelerar la velocidad de convergencia y la estabilidad del proceso de entrenamiento (Nair et al., 2018).

El objetivo del aprendizaje por demostración, también denominado *Clonación Conductual* (BC, *Behavioral Cloning*), es entrenar a un agente para imitar el comportamiento de un experto. En este paradigma, el agente aprende únicamente a través de las demostraciones; es decir, no se requiere el uso de recompensas ni interacciones con el entorno. BC se ha convertido en un enfoque ampliamente utilizado para obtener sistemas de control autónomos. En la práctica, es común que se dé preferencia al uso de algoritmos de BC sobre algoritmos RL, ya que las demostraciones de expertos a menudo son más fáciles de conseguir que diseñar las recompensas apropiadas que requiere RL (Sasaki & Yamashina, 2021).

En el paradigma de BC, el agente debe aprender una *política de imitación*, $\pi : S \mapsto A$, usando el conjunto de *demostraciones expertas* $\mathcal{D} = \{(s_0, a_0), (s_1, a_1), \ldots, (s_N, a_N)\}$. Considere que π es una ANN parametrizada por ϕ. El problema de BC consiste en encontrar el conjunto de parámetros ϕ para el cual π_ϕ minimice la función de pérdida (Cortés et al., 2021):

$$\mathcal{L}(\phi) = \mathbb{E}\left[\Big(f(s) - y(s)\Big)^2\right] = \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[\Big(\pi_\phi(s) - a\Big)^2\right]. \tag{3.47}$$

Este problema de optimización consiste en minimizar la diferencia entre las acciones ejecutadas por la política experta y las acciones estimadas por la política aprendida. Note que este problema de optimización es equivalente al problema de optimización de los parámetros en la función de valor de acción en DDPG y TD3, por lo que se utiliza la misma técnica de SGD para la optimización de los parámetros.

En BC se asume que la demostración experta es óptima. Desafortunadamente, en la práctica es difícil obtener demostraciones óptimas para muchas tareas porque el experto puede cometer errores o presentar comportamientos ligeramente separados del óptimo. Estos errores incluyen operaciones innecesarias y/o incorrectas para desempeñar las tareas. Además, la presencia de ruido en los sensores y/o actuadores del sistema pueden degradar la calidad de la demostración. En este caso, los algoritmos de BC son incapaces de aprender la política óptima (Sasaki et al., 2020).

Una posible solución para hacer frente a las demostraciones *ruidosas* es descartar las demostraciones no óptimas entre todas las demostraciones recolectadas. No obstante, este proceso de selección resulta impráctico porque, además de implicar un esfuerzo humano significativo, la cantidad de demostraciones descartadas podría ser considerable (Brown et al., 2019). Otra solución requiere anotar cada demostración con *puntajes de confianza* (Wu et al., 2019), que si bien no supone descartar ninguna experiencia, realizar dichas anotaciones también requiere un esfuerzo humano importante. Por lo tanto, es deseable obtener una solución que pueda hacer frente a las *demostraciones sub-óptimas* sin ningún proceso de detección y anotación asociado con los comportamientos no óptimos.

Para afrontar este problema, en esta tesis se utiliza la técnica propuesta por Nair et al. (2018), la cual consiste en modificar la función de pérdida del algoritmo BC para considerar solo las demostraciones en las que el valor Q de la política experta sea mayor que el valor Q de la política aprendida:

$$\mathcal{L}_{\mathrm{BC}}\left(\theta^{\mu}\right)=\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[\left(\mu\left(s|\theta^{\mu}\right)-a\right)^{2}\mathbb{1}_{Q(s,a)>Q(s,\mu(a))}\right], \tag{3.48}$$

donde el símbolo $\mathbb{1}$ se refiere a la función característica de un subconjunto, la cual indica la pertenencia del elemento a dicho subconjunto.

Teniendo en cuenta que las demostraciones podrían no ser óptimas, es necesario utilizar una técnica de exploración que permita al agente encontrar las acciones óptimas. La técnica de *exploración adaptativa* es una técnica de exploración que permite al agente encontrar las acciones óptimas. La técnica de exploración adaptativa es una versión mejorada de la estrategia ϵ-greedy, en la cual se ajusta dinámicamente el factor de exploración para adecuar la probabilidad de exploración y explotación durante el entrenamiento. Al utilizar esta técnica, la acción ejecutada en el entorno se calcula como:

$$a_t=\mu(s_t|\theta_t^{\mu})+\epsilon_t\mathcal{N}_t, \tag{3.49}$$

donde $\epsilon_t\mathcal{N}_t$ se denomina *ruido decreciente*.

La magnitud del ruido decreciente depende de la cantidad de iteraciones que se hayan ejecutado en el entorno. La magnitud del ruido decreciente se ajusta durante el entrenamiento como (Duan et al., 2020):

$$\epsilon_t \leftarrow \max\left(\epsilon_{\text{mín}}, \epsilon_t\left(1-\frac{t}{T_\epsilon}\right)\right) \tag{3.50}$$

donde $\epsilon_{\text{mín}}$ es el valor final de la magnitud del ruido decreciente, T_ϵ es la cantidad de interacciones necesarias para alcanzar el valor mínimo de la magnitud, y t es la cantidad de interacciones que se han ejecutado en el entorno.

El método de exploración adaptativa puede mejorar la eficiencia de la exploración, reducir la duración de la exploración y acelerar la convergencia del algoritmo (Dong & Zou, 2020).

Al estar basado en TD3, el algoritmo TD4 tiene una estructura muy similar a éste, y se presenta en el Algoritmo 3.

La diferencia principal entre TD3 y TD4 es el uso de demostraciones expertas para el entrenamiento del actor. Así, la función de desempeño a maximizar está determinada como:

$$J(\theta^{\mu}, \mathcal{R}, \mathcal{D}) = J(\theta^{\mu}, \mathcal{R}) - \mathcal{L}_{\mathrm{BC}}\left(\theta^{\mu}, \mathcal{D}\right) = \mathop{\mathbb{E}}_{s\sim\mathcal{R}}\left[Q\left(s, \mu(s|\theta^{\mu})\right)\right] - \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[\left(\mu\left(s|\theta^{\mu}\right) - a\right)^2 \mathbb{1}_{Q(s,a)>Q(s,\mu(a))}\right], \quad (3.51)$$

donde $Q(\cdot)$ puede ser cualquiera de los dos críticos. Note que los parámetros se optimizan en la dirección positiva del gradiente de la función de desempeño respecto a las experiencias recolectadas por el agentes, y en la dirección opuesta al gradiente de la función de pérdida respecto a las demostraciones del experto.

A continuación se detalla el proceso de entrenamiento. Se inicializan tres redes neuronales, una para trabajar como actor, $\mu(s|\theta^{\mu})$, y las otras dos funcionarán como críticos, $Q_{\theta_1}(s, a|\theta_1^Q)$, $Q_{\theta_2}(s, a|\theta_2^Q)$. Al igual que TD3, cada una de estas redes principales tiene su respectiva copia, Q'_{θ_1}, Q'_{θ_2} y μ', cuya función es un objetivo fijo durante el proceso de optimización de los pesos. Se crea un búfer de reproducción vacío $\mathcal{R}$ para almacenar las experiencias del agente al interactuar con el entorno (estados s, acciones a, recompensas r_{t+1}, siguientes estados s_{t+1}). Cada interacción entre el agente y el entorno cuenta como un paso de tiempo, representado por t. Una colección de un número dado de pasos de tiempo $\mathbf{T}$ se llama episodio. El algoritmo TD4 se repite durante un número deseado de episodios M. Al comienzo de cada episodio, se inicializa un proceso aleatorio $\mathcal{N}$ para la exploración del espacio de estado-acción, y el agente recibe la información del estado inicial s_0. Para cada paso de tiempo, se calcula una acción a través de la ANN del actor y se agrega una componente de ruido de exploración decreciente $a = \mu(s|\theta_t^{\mu}) + \epsilon\mathcal{N}$. Esta acción se ejecuta en el entorno, que devuelve el siguiente estado y la recompensa por la acción. La experiencia resultante (s, a, r_{t+1}, s_{t+1}) se almacena en el búfer de reproducción. Se muestrea un lote de N experiencias del búfer de reproducción $\mathcal{R}$, y otro lote del búfer de reproducción del experto $\mathcal{D}$, para entrenar las redes. Para suavizar la política (i.e., prevenir sobreestimación de la función de valor), a las acciones muestreadas del búfer se les añade una señal de ruido, y posteriormente son usadas para entrenar a los críticos. Ambos críticos calculan el valor de acción objetivo y se selecciona el mínimo de los dos para realizar la regresión, donde los parámetros de ambos críticos, θ_1^Q y θ_2^Q, se optimizan para minimizar su correspondiente función de pérdida. Los parámetros del actor θ^{μ} se actualizan una vez por cada ψ actualizaciones de los críticos, en la dirección del gradiente que maximice el valor del crítico Q_{θ_1} y minimice la diferencia entre las acciones estimadas y las acciones del experto. Los parámetros de las redes actor y críticos se copian en las redes objetivo, μ', Q'_{θ_1} y Q'_{θ_2}, respectivamente, usando el factor de promedio de Polyak ρ para realizar la actualización de los parámetros. Al finalizar cada interacción con el entorno, se actualiza la magnitud del ruido decreciente ϵ. Siguiendo este proceso, la ANN del actor tiende a aumentar la probabilidad de sugerir acciones que serán mejor evaluadas por los críticos, al mismo tiempo que intentará imitar las acciones del experto que considere óptimas respecto a los críticos, lo que conducirá a mayores recompensas y a un mejor desempeño del agente.

Algoritmo 3: Algoritmo TD3 con demostraciones expertas (TD4)

Entrada: Demostraciones expertas $\mathcal{D}$, tamaño de mini lote N, número de episodios M, factor de descuento γ, tasa de aprendizaje del crítico α_Q, tasa de aprendizaje del actor α_μ, factor de promedio de Polyak ρ, ruido de exploración $\mathcal{N}$, número de pasos de desvanecimiento de ruido de exploración T_ϵ, magnitud mínima de ruido de exploración $\epsilon_{\text{mín}}$, varianza del ruido de regularización σ^2, pasos de retardo de actualización de la política ψ

Salida: Actor entrenado $\mu(s|\theta^\mu)$

Inicializar redes neuronales y búfer de reproducción, ver inicialización en TD3 (Algoritmo 2)

para *episodio = 1 hasta M* **hacer**

  Inicializar un proceso aleatorio $\mathcal{N}$ para la exploración de acciones

  Recibir el estado inicial s_0

  para *paso t = 0 hasta T* **hacer**

    Observar estado s y seleccionar acción de acuerdo a Ec. (3.49):

$$a = \mu(s|\theta^\mu) + \epsilon\mathcal{N}$$

    Ejecutar la acción a_t y observar la recompensa r_{t+1} y el nuevo estado s_{t+1}

    Guardar la transición $(s_t, a_t, r_{t+1}, s_{t+1})$ en el búfer de repetición $\mathcal{R}$

    Muestrear un mini lote de N transiciones $(s_t, a_t, r_{t+1}, s_{t+1})$ del búfer de repetición $\mathcal{R}$

    Calcular las acciones objetivo de acuerdo a Ec. (3.41):

$$\tilde{a}' = \text{clip}\big(\mu'(s') + \text{clip}\left(\varepsilon, -c, c\right), a_{\min}, a_{\max}\big), \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

    Calcular los valores de acción objetivo, Ec. (3.42):

$$y(r, s') = r + \gamma \min_{i=1,2} Q'_{\theta_i}\left(s', \mu'(s'|\theta^{\mu'})|\theta_i^{Q'}\right)$$

    Actualizar los parámetros del crítico para minimizar la función de pérdida, Ecuaciones (3.43) y (3.44):

$$\mathcal{L}(\theta_j^Q) = \frac{1}{N}\sum_i^N \left[\left(Q_{\theta_j}(s, a|\theta_j^Q) - y(r, s')\right)^2\right], \qquad j = 1, 2$$

    si $t \bmod \psi = 0$ **entonces**

      Muestrear un mini lote de N transiciones $(s_t, a_t, r_{t+1}, s_{t+1})$ del búfer de repetición experto $\mathcal{D}$

      Calcular la función de pérdida de BC, Ec. (3.48):

$$\mathcal{L}_{\text{BC}}\left(\theta^\mu, \mathcal{D}\right) = \frac{1}{N}\sum_i^N \left[\left(\mu\left(s|\theta^\mu\right) - a\right)^2 \mathbb{1}_{Q_{\theta_1}(s,a) > Q_{\theta_1}(s,\mu(a))}\right]$$

      Calcular la función de desempeño del actor, Ec. (3.45):

$$J(\theta^\mu, \mathcal{R}) = \frac{1}{N}\sum_i^N \left[Q_{\theta_1}(s, \mu(s)|\theta^\mu)\right]$$

      Actualizar los parámetros del actor para maximizar la función de desempeño modificada con las demostraciones, Ec. (3.51):

$$J(\theta^\mu, \mathcal{R}, \mathcal{D}) = J(\theta^\mu, \mathcal{R}) - \mathcal{L}_{\text{BC}}\left(\theta^\mu, \mathcal{D}\right)$$

      Actualizar los parámetros de las redes objetivo, Ec. (3.34)

  Actualizar el valor de ruido decreciente, Ec. (3.50):

$$\epsilon \leftarrow \text{máx}\left(\epsilon_{\text{mín}}, \epsilon\left(1 - \frac{t}{T_\epsilon}\right)\right)$$

devolver $\mu(\cdot)$

3.4. Conclusión

El aprendizaje por refuerzo establece un procedimiento formal para realizar tareas de control basadas en el aprendizaje. En este sentido, los algoritmos de RL optimizan la política de toma de decisiones de un agente mediante una función de recompensa definida por el usuario. La función de recompensa define *qué debe hacer* un agente y un algoritmo de aprendizaje por refuerzo determina *cómo hacerlo*.

Existen dos tipos básicos de algoritmos: los basados en la *función de valor* y los basados en el *gradiente de la política*. La ventaja que presentan los algoritmos basados en la función de valor es su facilidad de implementación y rápida convergencia. No obstante, solo son aplicables en espacios discretos. Por su parte, los algoritmos basados en el gradiente de la política admiten espacios continuos, a cambio de una mayor complejidad en la implementación y una menor velocidad de convergencia. En otras palabras, los algoritmos basados en el gradiente de la política necesitan más interacciones con el entorno antes de producir una política satisfactoria.

La aproximación de la función de valor mediante redes neuronales artificiales ha permitido combinar ambos tipos de algoritmos en el denominado algoritmo *actor-crítico*, el cual permite resolver problemas de control en el espacio continuo. Además, las técnicas de *búfer de reproducción* y de *redes objetivo* estabilizan el proceso de entrenamiento.

No obstante, es deseable que la política óptima se obtenga lo más rápido posible, ya que el aprendizaje *desde cero* puede requerir una gran cantidad de tiempo e interacciones, lo que limita en gran medida la aplicación de DRL a muchas tareas del mundo real. Una estrategia para acelerar el aprendizaje es utilizar las demostraciones recopiladas anteriormente por un *experto*. Sin embargo, el rendimiento del agente a partir de la demostración estará limitado por el rendimiento del experto. Es decir, el agente aprenderá a comportarse como el experto, aún cuando el experto cometa errores.

Para solucionar este problema, en esta tesis se propone usar las demostraciones para guiar el aprendizaje, descartando aquellas que muestren un comportamiento sub-óptimo, mientras se permite que el agente interactúe con el entorno para experimentar distintas acciones a las tomadas por el experto, lo que reduce la posibilidad de aprender comportamientos sub-óptimos en el proceso de aprendizaje derivados de las demostraciones.

Capítulo 4

Técnica GMPPT Basada en TD4 para un Sistema Fotovoltaico

En este capítulo se presentan los trabajos de investigación realizados en el área de MPPT para sistemas PV basados en DRL y se describen sus ventajas y limitaciones. A continuación, se plantea la problemática a resolver en el caso de estudio, donde se presenta el sistema PV y la técnica GMPPT propuesta, delimitando las circunstancias de operación del sistema. Posteriormente, se describe la implementación virtual (i.e., mediante simulación) del algoritmo TD4; y finalmente, se verifica la eficacia de esta técnica mediante una serie de simulaciones, en las que se compara el rendimiento con otras técnicas GMPPT basadas en DRL (TD3 y DDPG) y con la técnica MPPT clásica P&O.

4.1. Introducción

Desde la construcción de la primera celda PV en 1941 (Zaidi, 2018) para generar energía eléctrica a partir de energía solar, la investigación sobre esta tecnología no se ha detenido. Más aún, debido a la reciente acentuación de la crisis energética y la contaminación ambiental, el estudio de la tecnología PV ha recibido una gran atención en todo el mundo, particularmente en el desarrollo de técnicas para aumentar la eficiencia de los sistemas PV en general, tales como la introducción de nuevos compuestos semiconductores, el perfeccionamiento de la electrónica de los convertidores e inversores, y distintas formas de ensamblaje y fabricación. No obstante, la forma más accesible para aumentar el rendimiento de los sistemas PV es mejorar su capacidad de GMPPT (Zhang et al., 2021).

DRL es un paradigma de AI, basado en ANNs, que puede utilizarse para resolver la tarea de GMPPT. La ventaja de DRL sobre otras soluciones de AI es clara: su implementación no necesita conjuntos de datos de entrenamiento ni conocimientos previos. Los algoritmos de DRL pueden acumular experiencia basada en la adquisición de datos de la operación del sistema PV, y ajustar los parámetros de control en tiempo real. Distintos estudios han demostrado la

capacidad de los algoritmos de DRL como técnicas GMPPT, superando el rendimiento de técnicas MPPT/GMPPT clásicas y otras técnicas basadas en AI (Lin et al., 2021a). No obstante, la desventaja que presentan los algoritmos de DRL es que requieren un gran número de experiencias (y generalmente, una gran cantidad de tiempo) para aprender a extraer la máxima potencia disponible, ya que los algoritmos necesitan explorar el espacio de estado-acción del sistema para distinguir las acciones de control correctas. En esta tesis se propone utilizar las demostraciones (i.e., experiencias) de un algoritmo MPPT P&O para guiar el proceso de exploración, utilizando el algoritmo denominado TD4.

El problema de la tarea de GMPPT, MPPT generalizando al caso de PS, está caracterizado por un espacio de estado multidimensional continuo: las características eléctricas de operación del sistema PV, como voltaje y corriente de los módulos PV; y por un espacio de acción unidimensional continuo: el ciclo de trabajo del convertidor DC-DC. El objetivo central del controlador TD4 es calcular el cambio requerido en el ciclo de trabajo del convertidor para mover el punto de operación actual del sistema PV hacia el GMPP.

4.2. Trabajos Relacionados

Respecto a los trabajos relacionados, se realiza una distinción entre dos categorías: algoritmos MPPT/GMPPT basados en RL y algoritmos MPPT/GMPPT basados en DRL. La primera categoría utiliza exclusivamente el algoritmo Q-learning, el cual es un método tabular basado en la recompensa media esperada. Aunque este algoritmo tiene un buen desempeño como MPPT/GMPPT, tanto en condiciones de irradiancia uniforme como en condiciones de PS, presenta la desventaja de requerir conjuntos discretos de espacio y de acción, ocasionando errores inherentes a la discretización de señales continuas. Por otro lado, los algoritmos DRL utilizan ANNs como reemplazo a la tabla utilizada por Q-learning, lo que permite estimar la recompensa media esperada para espacios de estado y acción continuos. La integración de ANNs mejora la precisión de la acción, al no requerir la discretización de las señales.

4.2.1. Técnicas MPPT basadas en RL

Hsu et al. (2015) proponen un algoritmo MPPT Q-learning que utiliza mediciones de voltaje y corriente del sistema PV para elaborar un conjunto de 4 estados discretos con codificación *One-Hot*, y un conjunto de 6 acciones discretas para controlar el voltaje del sistema. La función de recompensa se define como una función delta que indica si el punto de operación actual del sistema está en la vecindad del MPP. Los resultados obtenidos en simulación muestran que el algoritmo RL MPPT propuesto tiene una mejor eficiencia de seguimiento que un algoritmo convencional P&O en condiciones uniformes de irradiancia solar. La desventaja de este método es el diseño de la función de recompensa, el cual depende del sistema PV y de los conocimientos del usuario. Youssef et al. (2016) proponen un algoritmo similar, el cual difiere únicamente en la función de recompensa. Esta función define recompensas positivas si la potencia de salida se incrementa, recompensas negativas si disminuye, y recompensas de cero si la potencia

no varía significativamente. Los resultados obtenidos en simulación muestran que el algoritmo propuesto tiene una mayor velocidad de seguimiento y menos oscilaciones en estado estable que un método convencional P&O, para condiciones de irradiancia uniforme.

Lin et al. (2020) utilizan directamente las mediciones de voltaje y corriente como estados, en lugar de utilizar codificación *One-Hot*, lo que proporciona al controlador información más precisa sobre el punto de operación actual del sistema. Los resultados obtenidos en simulación muestran que el algoritmo RL tiene una mejor eficiencia de seguimiento que un algoritmo MPPT de *Lógica Difusa* (FL, *Fuzzy Logic*) y un algoritmo P&O, para un sistema PV aislado en condiciones uniformes de irradiancia.

Kofinas et al. (2017) proponen la inclusión de un estado adicional que señala la pendiente entre el punto de operación actual y el punto de operación en el instante anterior. Una pendiente de cero indica que el punto de operación actual corresponde al MPP. Este estado adicional permite mitigar las oscilaciones en la potencia generada en el estado estable al omitir el uso de perturbaciones una vez que se ha alcanzado el MPP. Los resultados obtenidos por simulación muestran que el algoritmo MPPT RL es capaz de producir mayor energía que un algoritmo convencional P&O en condiciones de irradiancia uniforme.

Lin et al. (2021a) restringen los puntos de operación del sistema PV entre el 60 % y 90 % del voltaje de circuito abierto, ya que generalmente el MPP se encuentra dentro de este rango. Esto permite acelerar la velocidad de convergencia del algoritmo al disminuir el espacio de estado del sistema. Los resultados obtenidos en simulación muestran que el algoritmo propuesto tiene un mejor desempeño que P&O y FL, para un sistema PV aislado en condiciones de irradiancia uniforme. No obstante, esta técnica no puede aplicarse a sistemas en condiciones de PS, ya que no es posible garantizar que el MPP se encuentre únicamente en este rango de operación.

Bavarinos et al. (2021) optimizan el conjunto discreto de acciones del algoritmo RL mediante un *algoritmo genético* (GA, *Genetic Algorithm*). Los resultados obtenidos en simulaciones muestran que el algoritmo RL presenta un mejor desempeño que un algoritmo FL. Sin embargo, la inclusión de GA demora la convergencia del algoritmo MPPT al requerir dos procesos distintos de optimización.

Chou et al. (2020) utilizan mediciones de voltaje de cada uno de los módulos que componen al sistema PV para extender el uso de la técnica MPPT basada en Q-learning a condiciones de PS. Los resultados obtenidos por simulación y experimentalmente muestran que el algoritmo GMPPT es capaz de ubicar correctamente el GMPP, a diferencia de un método P&O, lo que resulta en una mayor generación de energía. El inconveniente de este método es que el entrenamiento se realiza fuera de línea, mediante un conjunto de datos obtenidos para distintos patrones de sombreado y distintas condiciones de irradiancia. Hsu et al. (2020) realizan un estudio similar para condiciones PS. El método propuesto utiliza un conjunto de datos de entrenamiento extraídos del sistema PV para implementar la función de recompensa, por lo que se requiere diseñar una función distinta para cada sistema PV, lo cual no es práctico. Para generalizar el uso de la técnica a cualquier sistema PV, Kalogerakis et al. (2020) proponen utilizar un método híbrido Q-learning – P&O, donde el algoritmo Q-learning ubica la región del GMPP y el algoritmo P&O

realiza el ajuste fino. Los resultados obtenidos en simulación muestran que el método RL tiene una mayor velocidad de convergencia que un método de *optimización de enjambre de partículas* (PSO, *Particle Swam Optimization*).

Zhang et al. (2019) utilizan *múltiples agentes* (i.e., múltiples controladores y sistemas) para acelerar el proceso de aprendizaje de un algoritmo RL GMPPT basado en Q-learning para sistemas PV en condiciones de PS. Los resultados obtenidos muestran que el algoritmo propuesto tiene mejor desempeño que distintas técnicas convencionales y MHO, gracias a la coordinación efectiva entre búsqueda local e intercambio global entre agentes.

Ding et al. (2019) proponen utilizar el concepto de *transferencia de aprendizaje* en el algoritmo Q-learning, lo que permite descomponer el espacio de estado en sub-estados, acelerando el proceso de aprendizaje sin la necesidad de utilizar múltiples agentes. Los resultados obtenidos por simulación muestran que el algoritmo propuesto permite generar una mayor cantidad de energía que algoritmos MPPT como P&O, GA y PSO, entre otros. No obstante, el diseño de este algoritmo es complicado y tiene la limitante de requerir espacios discretos de estado y de acción.

4.2.2. Técnicas MPPT basadas en DRL

Chou et al. (2019) utilizan de una ANN para aproximar la función de recompensa media esperada, lo que soluciona el inconveniente del uso de espacios discretos para el conjunto de estado, sin embargo, requiere de un conjunto finito de acción. Los resultados experimentales muestran que el algoritmo MPPT propuesto permite generar una mayor cantidad de energía que un algoritmo MPPT Q-learning y un algoritmo P&O, en un sistema PV bajo condiciones uniformes de irradiancia.

Okafor et al. (2021) emplean el algoritmo RL DDPG para realizar la tarea de MPPT en un sistema PV en condiciones uniformes de irradiancia solar. Este algoritmo no requiere la discretización de estados ni de acciones, es decir, admite espacios continuos. Los resultados obtenidos mediante simulación muestran que el algoritmo MPPT DDPG tiene una mejor eficiencia de seguimiento y un menor tiempo de convergencia que las técnicas MPPT P&O y PSO. Finalmente, Avila et al. (2020) y Phan et al. (2020) proponen una formulación similar para sistemas PV en condiciones de PS. Los resultados obtenidos por simulación muestran que la técnica GMPPT DDPG es capaz de ubicar correctamente el GMPP, disminuyendo las pérdidas de energía ocasionadas por PS. No obstante, estos algoritmos presentan un desempeño deficiente durante la fase de exploración, lo que reduce la cantidad de energía generada durante ésta.

Una característica compartida por las técnicas MPPT y GMPPT es que, se basan únicamente en la experiencia recolectada por el agente mediante la continua interacción con los sistemas PV. Si bien, se obtienen resultados satisfactorios con este enfoque, ningún estudio plantea el uso de demostraciones expertas en la fase de entrenamiento para mejorar el desempeño del algoritmo, lo que se propone en esta tesis.

4.3. Definición del Caso de Estudio

La arquitectura del sistema PV estudiado está basado en una *nano-red*, la cual se define como un sistema de distribución y generación de energía localizado, con una capacidad instalada menor a 50 kW (Burmester et al., 2017), que se puede usar para alimentar una sola casa o edificio pequeño, o incluso un vehículo eléctrico (Gharibeh et al., 2021). La nano-red puede conectarse a la red eléctrica principal u operar de forma aislada (Cordova-Fajardo & Tututi, 2019).

El sistema PV está constituido por un arreglo PV, integrado por cuatro módulos y su respectivo diodo en derivación, un convertidor DC-DC tipo boost, y una carga DC, constituida por una batería y una resistencia, como se muestra en la Fig. 4.1.

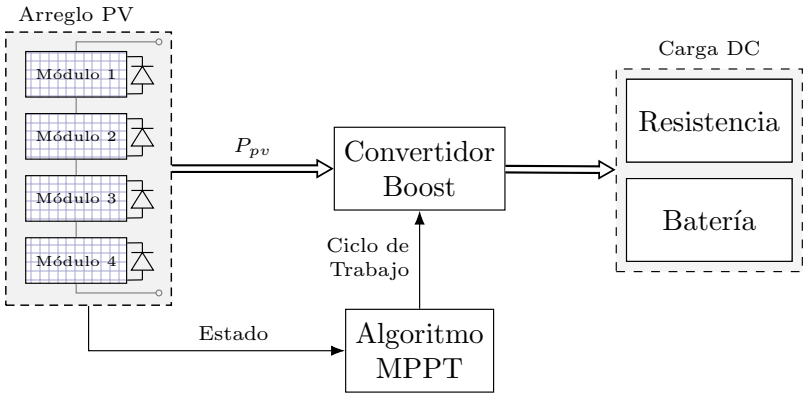


Figura 4.1: Diagrama del sistema PV en condiciones de sombreado parcial para el caso de estudio. Se considera una arquitectura de nano-red aislada.

En este estudio se realizan las siguientes consideraciones:

- El sistema PV opera como una nano-red aislada, es decir, sin conexión con la red eléctrica principal.
- El voltaje del bus de DC (i.e., el voltaje de la batería) es de 48 V, lo cual es típico en arquitecturas de nano-redes (Joseph et al., 2018). Este voltaje se mantiene constante durante todo el tiempo de operación del sistema, es decir, no se considera la descarga de la batería.
- El valor de la resistencia en la carga es de 80 Ω, como se sugiere en diversos estudios (Başoğlu & Çakır, 2016; Chou et al., 2019; Ding et al., 2019).
- El modelo del convertidor DC-DC tipo boost utilizado corresponde al modelo promediado, incluido en MATLAB/Simulink.

El resto de parámetros del sistema PV utilizados en la simulación se muestran en la Tabla 4.1.

Al definir el problema en términos de DRL, los elementos del sistema PV se asocian con la entidad correspondiente en este paradigma. Resulta intuitivo que el arreglo PV, el convertidor boost y la carga DC representan al entorno, mientras que el algoritmo GMPPT representa al agente, basado en el algoritmo TD4. El objetivo del

Tabla 4.1: Parámetros de los componentes del sistema PV en el caso de estudio

	Parámetro	Magnitud
Módulos PV*	P_{max}	50 W
	V_{mpp}	6.57 V
	I_{mpp}	7.61 A
	V_{oc}	8.22 V
	I_{sc}	8.21 A
	μ_I	3.2×10^{-3} A/°C
	N_s	12
Arreglo PV†	P_{max}	200 W
	V_{mpp}	26.3 V
	I_{mpp}	7.61 A
	V_{oc}	32.9 V
	I_{sc}	8.21 A
Carga	R	80 Ω
	V_{bat}	48 V

* Todos los módulos PV comparten las mismas características eléctricas.
† Equivalente al módulo PV comercial Kyocera KC200GT.

agente es determinar la acción de control, definida como el cambio requerido en el ciclo de trabajo del convertidor, que mueva el punto de operación actual del sistema PV hacia el GMPP.

El agente está conformado por una red neuronal artificial que parametriza la política del actor, denotada por μ, y dos redes neuronales artificiales que parametrizan a los críticos, denotados por Q_1 y Q_2. Adicionalmente, cada una de estas *redes principales* tiene su respectiva *red objetivo*, utilizadas para estabilizar el proceso de aprendizaje. Un algoritmo P&O provee las *demostraciones expertas*, en el búfer de reproducción $\mathcal{D}$, utilizadas para guiar el aprendizaje durante la fase de exploración. Durante el aprendizaje, el agente almacena la experiencia obtenida al interactuar con el entorno en el búfer de reproducción $\mathcal{R}$. El objetivo del agente es encontrar los parámetros θ^μ de la política, de tal forma que se maximice la función de desempeño descrita en la Ec. (3.51):

$$J(\theta^\mu) = \mathop{\mathbb{E}}_{s\sim\mathcal{R}}\left[Q\left(s, \mu(s|\theta^\mu)\right)\right] - \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[\left(\mu\left(s|\theta^\mu\right) - a\right)^2 \mathbb{1}_{Q(s,a)>Q(s,\mu(a))}\right]. \tag{4.1}$$

Para utilizar el algoritmo TD4, descrito en el Algoritmo 3, se formula la tarea de GMPPT como un MDP, lo que se detalla a continuación.

4.3.1. Modelo MDP del Sistema PV

Como se explicó anteriormente, para utilizar un algoritmo RL/DRL es necesario definir el problema de optimización como un MDP compuesto por cinco elementos, $\mathcal{M} = \{S, A, \mathcal{T}, r, \gamma\}$.

En el caso de la tarea de GMPPT, la función de transición $\mathcal{T}$ depende de las condiciones ambientales (i.e., condiciones atmosféricas). Dado que la atmósfera es un sistema no linear caótico (Yushou et al., 2008), y el caos

no puede ser descrito por ninguna ecuación (Cai et al., 2021), la función de transición $\mathcal{T}$ no puede conocerse. No obstante, los algoritmos de DRL, al estar basados en la experiencia, no requieren esta información.

El resto de los elementos del MDP, los cuales sí son requeridos por el algoritmo TD4, se definen en las siguientes subsecciones.

Espacio de estado

La propiedad fundamental de un MDP, conocida como la propiedad de Markov, establece que cada transición de estado es independiente de la historia de estados y acciones anteriores y depende solo del estado y la acción actuales; asimismo, cada recompensa es independiente de los estados, acciones y recompensas pasadas y depende únicamente de la transición más reciente (Kalogerakis et al., 2020). En otras palabras, el estado actual del sistema contiene toda la información necesaria para predecir la recompensa actual en base a la acción elegida.

Para cumplir con la propiedad de Markov, el espacio de estado del MDP de GMPPT se define como:

$$S : \{V_{pv}, \Delta V_{pv}, P_{pv}, \Delta P_{pv}, V_j, \Delta V_j\} \in \mathbb{R}, \qquad j = 1, 2, 3, 4, \tag{4.2}$$

donde V_{pv} es el voltaje en terminales del arreglo PV, ΔV_{pv} es la diferencia entre el voltaje actual y el voltaje en el instante anterior, P_{pv} es la potencia generada por el sistema PV, ΔP_{pv} es la diferencia entre la potencia actual y la potencia del instante anterior, y V_j y ΔV_j son el voltaje en terminales y la diferencia de voltaje para el módulo j, respectivamente.

Los voltajes y potencias de cada módulo, y del arreglo en conjunto, son suficientes para inferir la información de los patrones de sombreado y las condiciones meteorológicas (Chou et al., 2020); mientras que la diferencia de las mediciones respecto al instante anterior ($\Delta V_{pv}, \Delta P_{pv}, \Delta V_j$) indican al agente la dirección en la que se ha movido el punto de operación del sistema PV en el instante anterior (Avila et al., 2019).

De esta manera, se requieren 12 estados para describir el MDP del sistema PV mostrado en la Fig. 4.1.

Espacio de acción

El agente determina la magnitud de cambio ΔD del ciclo de trabajo del convertidor para mover el punto de operación del sistema PV hacia el GMPP. Así, el agente decide incrementar, reducir, o no modificar el ciclo de trabajo. Para lograr un seguimiento rápido, el agente debe seleccionar una perturbación pequeña si el punto de operación está cerca del GMPP, y una perturbación mayor si el punto de operación se encuentra distante.

De esta manera, se define el espacio de estado como el conjunto de las perturbaciones permitidas en el ciclo de trabajo del convertidor DC del sistema:

$$A : \{\Delta D \in \mathbb{R} \mid -1 \leq \Delta D \leq 1\}. \tag{4.3}$$

Así, el ciclo de trabajo del convertidor para el instante siguiente se determina como:

$$D_{t+1} = D_t + \Delta D_t, \tag{4.4}$$

donde D_t es el ciclo de trabajo actual y ΔD_t es la acción elegida por el agente.

Función de recompensa

La recompensa es un valor numérico que el ambiente otorga al agente e indica la evaluación del desempeño de la acción elegida. Esta métrica de desempeño juega un papel vital en la fase de aprendizaje de un agente, ya que la optimización de parámetros tiene como objetivo maximizar la recompensa esperada.

En el caso de GMPPT, el agente recibe una recompensa proporcional a la cantidad de potencia generada (Wang et al., 2021). Formalmente, la función de recompensa está definida como:

$$R(s_t, a_t, s_{t+1}) : r_{t+1} = P_{pv,\,t+1}, \tag{4.5}$$

donde $P_{pv,\,t+1}$ es la potencia generada por el sistema PV al evaluar la acción a_t elegida por el agente en el estado s_t.

De esta manera, mientras mayor sea la recompensa acumulada por el agente, mayor será la cantidad de potencia generada por el sistema PV.

Factor de descuento

El factor de descuento γ es un valor que se utiliza para ayudar al agente a encontrar secuencias óptimas de acciones futuras, que conduzcan diligentemente a grandes recompensas posteriores, y no solo a la acción óptima actual. En el sistema PV, la dinámica de la planta es instantánea, ya que la energía generada es utilizada enseguida por la carga. En otras palabras, maximizar la recompensa inmediata es equivalente a maximizar la recompensa futura. Por lo tanto, el factor de descuento utilizado es $\gamma = 0$.

4.3.2. Métrica de evaluación

La métrica de evaluación seleccionada para comparar el desempeño de los modelos es la eficiencia de seguimiento, la cual está definida como (Xu et al., 2020):

$$\eta = \mathbb{E}\left[\sum \frac{P_{\mathrm{MPPT}}}{P_{\max}}\right] \times 100\,\%, \tag{4.6}$$

donde P_{MPPT} es la potencia generada por el sistema PV utilizando el algoritmo MPPT y $P_{\max}$ es la potencia máxima teórica.

4.4. Implementación virtual

La implementación y entrenamiento del algoritmo TD4 se realiza utilizando el lenguaje de programación Python y el marco de aprendizaje automático de código abierto PyTorch, ya que en los últimos años la comunidad de AI (y en especial los usuarios de ML) han popularizado y estandarizado el uso de estas herramientas para la ejecución de algoritmos de aprendizaje automático. Mientras que el modelado y simulación del sistema PV se efectúa en el entorno de programación visual Simulink y la plataforma de cómputo numérico MATLAB, debido a la accesibilidad que brindan estas plataformas para diseñar y simular sistemas dinámicos, además de ser ampliamente utilizado y aceptado en el área de control. La comunicación entre ambos entornos de programación se realiza mediante la librería *MATLAB Engine API for Python*, incluida en el paquete de instalación de MATLAB. Las versiones utilizadas son:

- Python 3.8.10
- PyTorch 1.11.0
- MATLAB 9.9 (R2020b)
- Simulink 10.2

4.4.1. Modelo del sistema PV en Simulink

La Fig. 4.2 muestra el modelo del sistema PV en Simulink. Este modelo incluye cuatro módulos en serie, cada uno con un diodo en derivación, un convertidor DC-DC tipo boost, y una carga DC, compuesta por una resistencia y una batería. Los parámetros de los elementos del modelo están definidos en la Tabla 4.1. El modelo del convertidor boost corresponde al modelo promediado, y está incluido en las librerías de MATLAB/Simulink.

Se utilizan cinco sensores de voltaje, uno para cada módulo, y uno más para el arreglo de módulos; y un sensor de corriente que mide la corriente de salida del arreglo. Esta información es utilizada en Python/PyTorch para formar el estado del MDP.

Los valores de irradiancia, temperatura del módulo y ciclo de trabajo del convertidor se consideran entradas del modelo, y son determinados en Python/PyTorch en cada paso de simulación (i.e., cada interacción entre el agente y el entorno). En este sentido, la irradiancia y la temperatura del módulo son leídas directamente de una base de datos; mientras que el ciclo de trabajo se determina mediante el algoritmo de control TD4, como se explica a continuación.

4.4.2. Modelo MDP del sistema PV en Python

Generalmente, para utilizar algoritmos de DRL en Python se sigue la interfaz *Gym* propuesta por *OpenAI* (Brockman et al., 2016). OpenAI Gym es una plataforma de código abierto implementada en Python para evaluar algoritmos DRL en una variedad de entornos, entre los cuales se incluyen entornos de control clásico como el péndulo

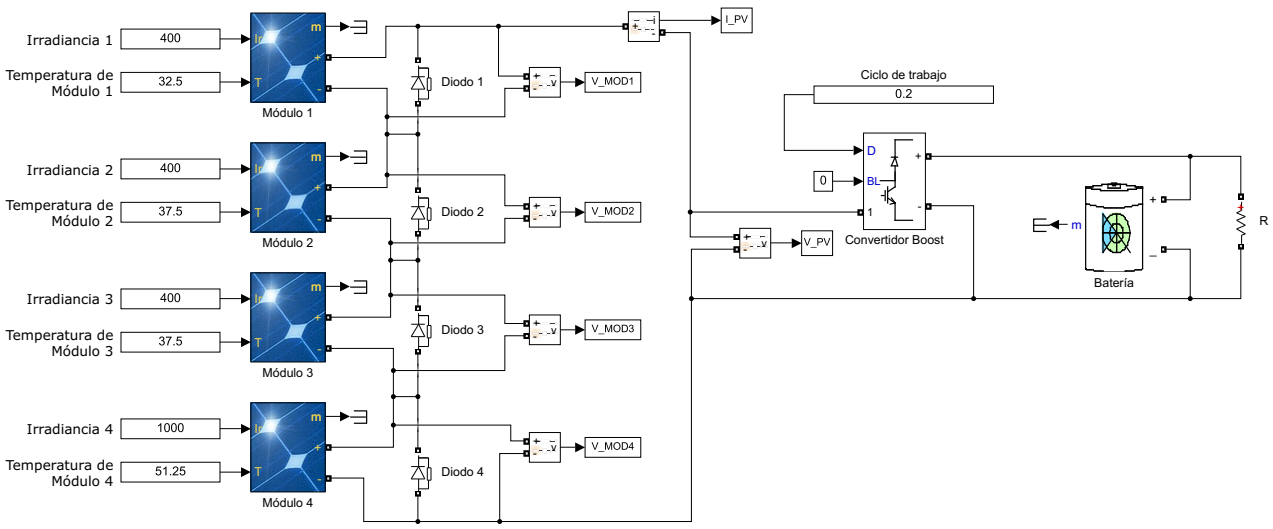


Figura 4.2: Diagrama del sistema PV en MATLAB/Simulink.

invertido. Cada entorno especifica su propio espacio de estado y de acción, es decir, describe el formato de acciones y observaciones válidas. En adición a los entornos integrados, OpenAI Gym permite definir entornos personalizados, como por ejemplo el del sistema PV. La ventaja de utilizar el protocolo de OpenAI Gym es la abstracción de los entornos, permitiendo que sean utilizados indistintamente por cualquier algoritmo de DRL.

Para definir un entorno personalizado, se debe crear una clase que herede de la clase *gym.Env* de la librería *gym*, como se muestra en el siguiente fragmento de código:

```
1  import gym
2
3
4  class SistemaPV(gym.Env):
5      def __init__(self):
6          pass
7
8      def step(self, action):
9          pass
10
11     def reset(self):
12         pass
13
14     def render(self):
15         pass
16
17     def close(self):
18         pass
```

Listado 4.1: Interfaz OpenAI Gym para algoritmos DRL

A continuación se describe cada uno de los métodos que debe implementar la clase personalizada:

- `__init__`: método constructor de la clase. En este método se definen dos atributos, `observation_space` y

`action_space`, que definen el espacio de estado y de acción del entorno. En el caso del sistema PV, el espacio de estado es un vector de doce elementos, mientras que el espacio de acción es un vector de un elemento.

- `step`: método que implementa la acción de un agente en un entorno. En este método se recibe una acción, se ejecuta en el entorno y se devuelve el nuevo estado del entorno y la recompensa. En el caso del sistema PV, la acción indica el cambio en el ciclo del trabajo del convertidor boost. Esta acción se envía a Simulink y se procede a realizar la simulación. Finalmente, Simulink comunica el resultado de la simulación al programa en Python, el cual calcula la recompensa y devuelve el nuevo estado.
- `reset`: método que reinicia el entorno. En este método se devuelve el estado inicial del entorno. Este método es llamado en dos situaciones: cuando se inicia la ejecución del algoritmo, o cuando un episodio ha finalizado. En el caso del sistema PV, un episodio finaliza cuando se han utilizado todos los valores de irradiancia durante un día.
- `render`: método que muestra la representación gráfica del entorno. En este método se muestra el estado del entorno en una ventana de visualización. Este método es opcional, y no se utiliza en el caso del sistema PV.
- `close`: método que finaliza el entorno. En este método se liberan los recursos asociados al entorno. Este método es opcional, y en el caso del sistema PV no se utiliza.

4.4.3. Arquitectura del actor y el crítico

La arquitectura de las redes neuronales del actor y crítico sigue las recomendaciones de Lillicrap et al. (2019) y Lapan (2018). La Fig. 4.3 muestra la arquitectura de ambas.

La arquitectura del actor consta de tres capas: una capa de entrada, una capa oculta y una capa de salida. Para comprobar que esta arquitectura es suficiente para la representación de la política del agente, se realizan distintos experimentos en los que se agregan más capas ocultas, y se observa que la precisión del algoritmo no cambia significativamente, por lo que se utiliza la arquitectura más simple, lo que agiliza el proceso de entrenamiento (Hagan et al., 2014). La capa de entrada recibe un vector de doce observaciones, la capa oculta se compone de 128 neuronas con función de activación ReLU (Unidad Lineal Rectificada, *Rectified Linear Unit*) (Hara et al., 2015), y la capa de salida un vector unidimensional, correspondiente a la acción sobre el ciclo de trabajo del convertidor. Las acciones de salida se transforman con una función de activación tangente hiperbólica (TanH) para comprimir los valores en el rango $(-1, 1)$, los cuales están dentro del espacio de acciones admitidas por el entorno.

Por su parte, la arquitectura del crítico incluye dos rutas separadas: una para el vector de observaciones y otra para el vector de acciones. Esta separación de entradas permite establecer una etapa de preprocesamiento y extracción de características del vector de estados (el más complejo de ambos vectores) (Lapan, 2018), de manera similar a la arquitectura del actor. Posteriormente, la salida de esta etapa se concatena con el vector de acción y se procesa en otra capa oculta, para finalmente transformarse en la salida del crítico, correspondiente a un valor

escalar. De esta manera, el crítico está formado por una capa de entrada con doce observaciones, una capa oculta con 128 neuronas con función de activación ReLU, otra capa oculta de 128 neuronas que concatena la salida de la primera capa oculta y el vector de acciones, con función de activación ReLU, y finalmente una capa de salida con función de activación lineal.

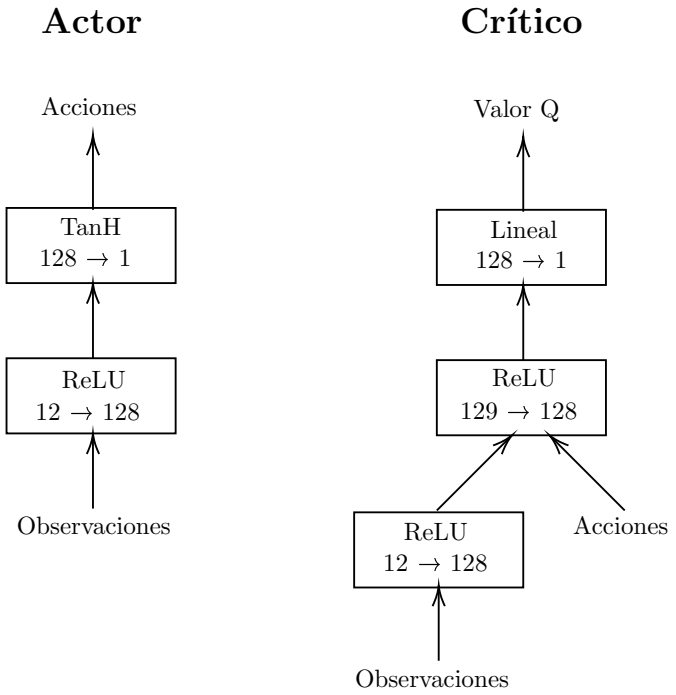


Figura 4.3: Arquitectura del actor y el crítico en la implementación de TD4 para el sistema PV.

4.4.4. Configuración de los episodios

La configuración de los episodios sigue la dinámica de un sistema PV en la vida real. En este caso, se supone que el sistema PV está en funcionamiento desde las 8:00 hasta las 18:00 (i.e., durante el día) y que se leen los valores de irradiancia cada minuto. Así, un episodio tiene una duración de 600 minutos, es decir, el algoritmo DRL puede interactuar 600 veces con el entorno por cada episodio, una interacción por minuto.

Para incluir los efectos de PS, se genera una señal tipo escalón que varía aleatoriamente durante el episodio, tomando valores del conjunto discreto $\{200, 400, 600, 800, 1000\}$ W/m^2 para cada uno de los módulos que componen el arreglo PV; mientras que la temperatura ambiente se mantiene constante en 25°C durante todo el episodio. Esta configuración se determinó adecuada porque la temperatura ambiente tiene una influencia lenta sobre la celda PV, y no está directamente relacionada con la velocidad de respuesta dinámica; mientras, en la práctica, las nubes se mueven rápidamente, lo que provoca un cambio repentino en la potencia de salida del panel PV. (Aashoor & Robinson, 2012). En consecuencia, el algoritmo debe probarse bajo diferentes niveles de irradiación para verificar la velocidad de seguimiento. Algunos investigadores consideran el desempeño de algunas técnicas MPPT/GMPPT propuestas durante la variación de carga. Sin embargo, el sistema PV implementado en esta tesis es un sistema

independiente donde la carga permanece constante, a diferencia de un sistema conectado a la red. Por lo tanto, la variación de carga no se ha tenido en cuenta para probar el rendimiento del método propuesto.

Como es habitual en el paradigma de ML en general, se utilizan dos conjuntos de datos, uno para el entrenamiento y otro para la prueba (i.e., *testing*), los cuales son excluyentes; es decir, los datos en el conjunto de entrenamiento no están presentes en el conjunto de prueba, y viceversa. El conjunto de prueba no se presenta a los modelos durante el entrenamiento. Esta separación de los conjuntos se realiza con el fin de evaluar la capacidad de *generalización* del modelo en datos no vistos durante el entrenamiento.

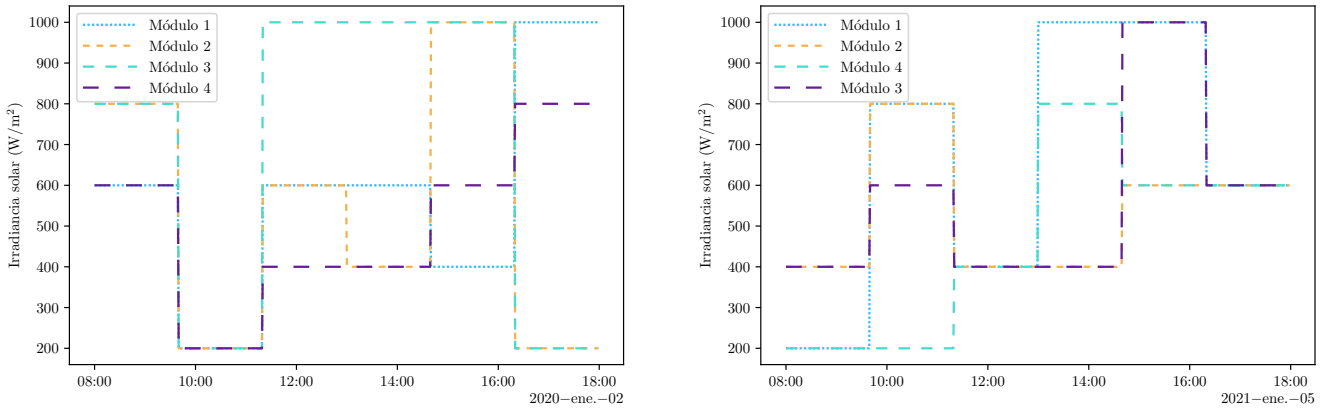
La Fig. 4.4 muestra un día para cada uno de los conjuntos de datos. El *conjunto de datos de entrenamiento* tiene una longitud de 110 días, 100 días reservados para el entrenamiento de los algoritmos y 10 más para la recolección de demostraciones del experto, abarcando el periodo de tiempo del 1 de enero de 2020, al 20 de abril de 2020; mientras que el *conjunto de prueba* tiene una longitud de 60 días, abarcando el periodo de tiempo del 1 de enero de 2021, al 2 de marzo de 2021. La fecha en los datos de ambos conjuntos no contiene información relevante, y solo se utiliza para comparar directamente el desempeño de los algoritmos MPPT/GMPPT en una misma fecha.

Originalmente, el conjunto de datos de entrenamiento estaba compuesto por 1,000 episodios, como lo especifican Phan et al. (2020) en un algoritmo similar. Sin embargo, al realizar los experimentos de prueba, se observó que el algoritmo convergía 10 veces más rápido que el propuesto en Phan et al. (2020), por lo que se realizó una reducción del número de episodios a 100. Respecto a la cantidad de demostraciones, se utiliza el 10 % de episodios de entrenamiento, es decir, un total de 10 episodios de donde se recolecta el comportamiento exhibido por el algoritmo experto, como se recomienda en Huang et al. (2021). Finalmente, en aplicaciones de RL/DRL, es común que el conjunto de prueba contenga al menos el 50 % del número de datos en el conjunto de entrenamiento (Meng & Khushi, 2019), lo cual se satisface con los 60 episodios seleccionados para dicho conjunto en este caso de estudio. La distribución de los episodios en los diferentes conjuntos de demostración, entrenamiento y prueba se resume en la Tabla 4.2. La columna *Distribución (entrenamiento)* especifica el porcentaje de episodios utilizados en cada conjunto respecto al conjunto de entrenamiento, mientras que la columna *Distribución (total)* especifica el porcentaje para cada conjunto respecto al total de episodios utilizados en los tres conjuntos. Para los algoritmos que no requieren demostraciones, la distribución es idéntica, omitiendo únicamente el conjunto de demostración.

Tabla 4.2: Distribución de los episodios para el algoritmo TD4

Conjunto	Episodios	Interacciones	Distribución (entrenamiento)	Distribución (total)
Demostración	10	6,000	10 % †	5.88 %
Prueba	60	36,000	50 % §	35.29 %
Entrenamiento	100	60,000 *	100 %	58.82 %
Total	170	102,000	-	100 %

* Phan et al. (2020).
† Huang et al. (2021).
§ Meng & Khushi (2019).



(a) Datos de entrenamiento.

(b) Datos de prueba.

Figura 4.4: Patrones de irradiancia solar en los conjuntos de datos de entrenamiento y prueba. La temperatura ambiente se considera constante durante todo el día, con valor de 25°C.

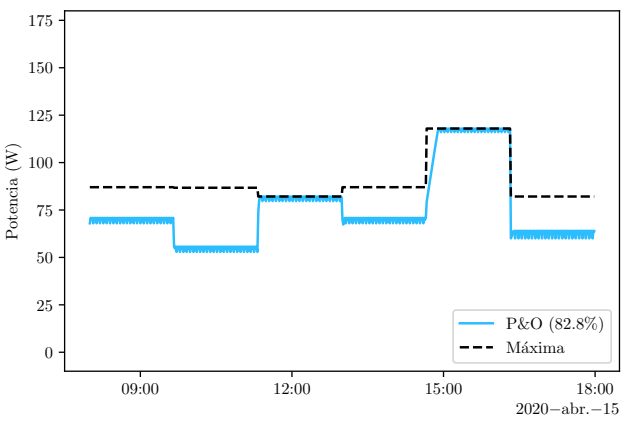
4.4.5. Demostraciones del algoritmo MPPT P&O

Las demostraciones son proporcionadas por un algoritmo MPPT P&O con un paso de perturbación fijo en el ciclo de trabajo del convertidor boost de $\pm 1\,\%$ (Hadji et al., 2018). La longitud del conjunto de demostraciones es de 6,000 experiencias, es decir, diez días para la actual configuración de los episodios. Los días seleccionados para recolectar las demostraciones abarcan del 11 de abril de 2020 al 20 de abril de 2020.

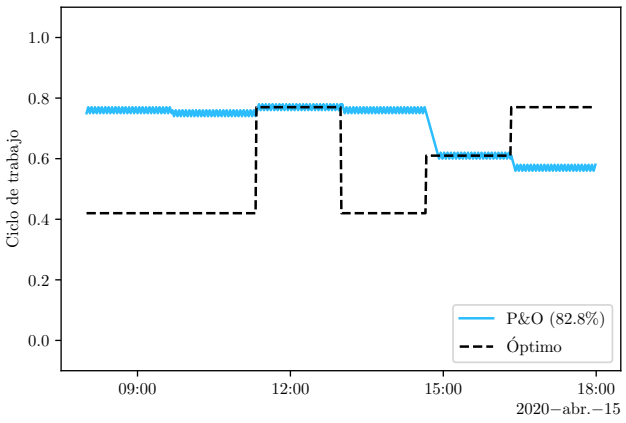
La dinámica del algoritmo MPPT P&O para el día 15 de abril de 2020 se muestra en la Fig. 4.5. La línea punteada corresponde con la potencia máxima disponible, es decir, la potencia producida por el sistema PV cuando se opera en el GMPP teórico. La línea azul indica la potencia producida por el sistema PV cuando es controlado por el algoritmo MPPT P&O. Se observa que, de los seis patrones de sombreado durante el día (i.e., seis GMPPT diferentes), el algoritmo P&O solo es capaz de ubicar correctamente el GMPPT en dos de ellos. Adicionalmente, la baja velocidad de seguimiento del algoritmo P&O se hace evidente alrededor de las 15:00, donde el sistema PV tarda en alcanzar el GMPP alrededor de 20 muestras (una quinta parte de la duración de este patrón de irradiancia), lo que implica una pérdida adicional de rendimiento. Para este día, la eficiencia de seguimiento del MPPT P&O es de $82.5\,\%$. La relativa baja eficiencia de seguimiento se debe a la incapacidad del algoritmo MPPT P&O para detectar el GMPP, como se explicó en la Sección 2.7.

4.4.6. Configuración de los experimentos

Los parámetros de configuración para el entrenamiento del algoritmo TD4 se muestran en la Tabla 4.3. Estos parámetros son seleccionados a través de una *búsqueda de retícula* (*grid search*). Para guiar la búsqueda, se utilizan los valores recomendados por Lillicrap et al. (2019), Lapan (2018) y Nair et al. (2018). El proceso de búsqueda consiste en generar una combinación de parámetros, elegidos mediante la combinación de los distintos valores que



(a) Potencia generated.



(b) Ciclo de trabajo.

Figura 4.5: Dinámica del algoritmo MPPT P&O para el sistema PV del caso de estudio en un día determinado, con una eficiencia de seguimiento de 82.8 %.

se pueden tomar para cada uno de los parámetros, y evaluar el desempeño del algoritmo, eligiendo el conjunto de parámetros que presente un mejor desempeño. Por ejemplo, para el tamaño de mini lote se elige un valor del conjunto discreto $[32, 64, 128, 256, 512]$, y para la tasa de aprendizaje del crítico se elige un valor del conjunto $[0.0001, 0.001, 0.01]$. Los parámetros que presentaron el mejor desempeño se muestran en la Tabla 4.3.

Para los algoritmos DDPG y TD3, se utilizan los mismos parámetros (los aplicables, según el algoritmo). Tanto TD4, TD3 y DDPG utilizan el optimizador Adam (Kingma & Ba, 2014), una versión mejorada de SGD. Debido a la estocasticidad en el proceso de inicialización de los pesos, y por tanto, del proceso de entrenamiento, se realizaron 100 experimentos para cada algoritmo DRL. Esto con la finalidad de realizar una comparación justificada de los resultados de los algoritmos, y no presentar desempeños inconsistentes.

La configuración del algoritmo P&O es similar a la descrita en la Sección 4.4.5; es decir, se utiliza un paso de perturbación fijo en el ciclo de trabajo del convertidor boost de $\pm 1\,\%$. A diferencia de los algoritmos DRL, el algoritmo P&O presenta un comportamiento determinista, por lo que solo se realiza un experimento para determinar su desempeño.

4.5. Resultados de Simulación

Las simulaciones ofrecen una oportunidad para verificar la viabilidad y el rendimiento del algoritmo TD4 propuesto, puesto que permiten hacer comparaciones directas entre dos algoritmos MPPT/GMPPT, que de otra manera sería difícil de realizar en la práctica, ya que las características y condiciones de operación pueden variar entre dos sistemas similares. Además, no es posible conocer con certeza el GMPP en la práctica, por lo que es necesario realizar simulaciones para verificar la eficacia de los algoritmos.

Para el análisis de evaluación y comparación con los otros algoritmos MPPT/GMPPT (TD3, DDPG y P&O), los estudios de simulación se configuraron exactamente en las mismas condiciones, como se describe en la sección

Tabla 4.3: Parámetros de entrenamiento de los algoritmos GMPPT DRL

Parámetro	Valor	Descripción
$\|\mathcal{D}\|$	6,000	Número de demostraciones del experto
N	64	Tamaño de mini lote para aproximar el gradiente
M	100	Número de episodios para el entrenamiento
$\mathbf{T}$	60,000	Número total de interacciones con el entorno durante el entrenamiento
γ	0	Factor de descuento
α_Q	1×10^{-3}	Tasa de aprendizaje del crítico
α_μ	1×10^{-4}	Tasa de aprendizaje del actor
ρ	1×10^{-3}	Factor de promedio de Polyak
$\mathcal{N}$	$\mathcal{N}(0, 0.3)$	Ruido de exploración (distribución normal)
R_ϵ	10,000	Número de interacciones para el desvanecimiento del ruido de exploración
$\epsilon_{\text{mín}}$	0.01	Magnitud final del ruido de exploración
σ^2	0.01	Varianza del ruido de regularización
ψ	2	Pasos de retardo para la actualización de la política

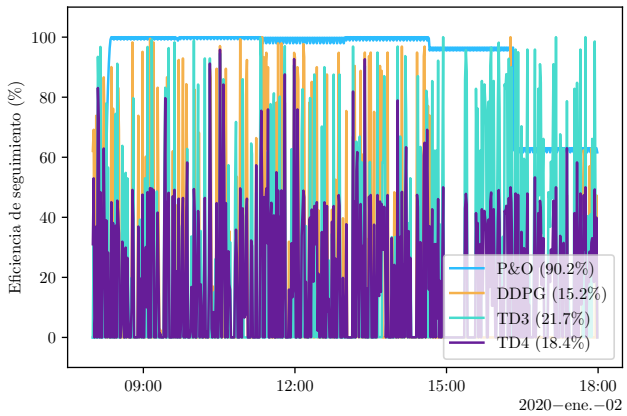
anterior. Así, para comparar el rendimiento de los algoritmos, se utiliza la eficiencia de seguimiento, descrita en la Ec. (4.6), donde la potencia de simulación del conjunto fotovoltaico se compara con la potencia calculada en el GMPP teórico.

4.5.1. Fase de Entrenamiento

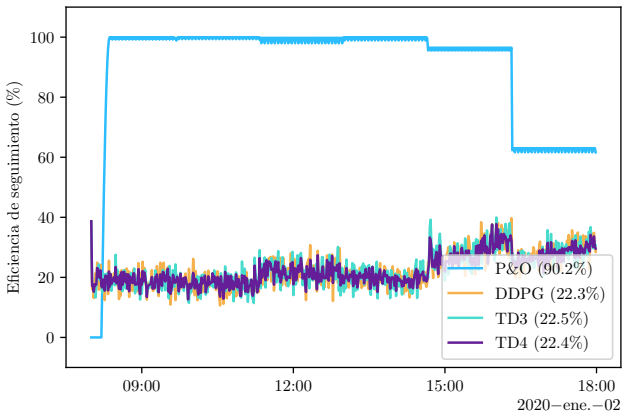
En esta sección se analiza el rendimiento de los algoritmos GMPPT basados en DRL (TD4, TD3 y DDPG) en la fase de entrenamiento y se realiza una comparación con el rendimiento del algoritmo MPPT P&O y con la potencia teórica máxima para cada condición climática de operación.

Durante los primeras interacciones del entrenamiento de los algoritmos DRL con el sistema PV, la eficiencia de seguimiento es muy baja. En el segundo día de entrenamiento, correspondiente a la fecha 2 de enero de 2020, la eficiencia de seguimiento de TD4 es de 18.4 %, y de 21.7 % y 15.2 % en el caso de TD3 y DDPG, respectivamente; mientras que la eficiencia de seguimiento de P&O para este mismo día es de 90.2 %. Estos resultados corresponden a un experimento en particular, cuya dinámica durante el día se observa en la Fig. 4.6(a). La baja eficiencia de seguimiento de los algoritmos DRL durante la fase inicial del entrenamiento se debe a la exploración del espacio de estado y acción, como se explicó en la Sección 3.2.3. Debido a que los pesos en las ANNs del actor y el crítico en los algoritmos DRL se inicializan aleatoriamente, el desempeño de un experimento en específico puede no ser totalmente representativo, por lo que la Fig. 4.6(b) muestra una media de los resultados de 100 experimentos independientes para cada algoritmo. En la Fig. 4.6(b) se observa que el desempeño de los tres algoritmos, en promedio, es extremadamente similar, con una eficiencia de seguimiento de alrededor del 22 %.

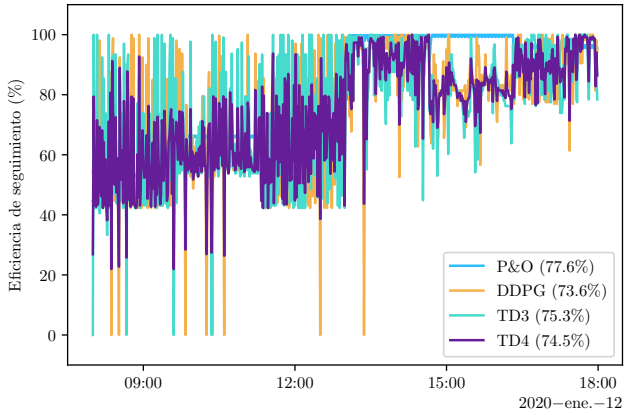
Posteriormente, en el episodio 12 de entrenamiento, correspondiente al día 12 de enero de 2020, la eficiencia de los tres algoritmos DRL alcanza el rendimiento del algoritmo P&O, como se muestra en la Fig 4.6(c) para un experimento en particular y en la Fig. 4.6(d) para una media de 100 experimentos. Particularmente, en esta última figura, la eficiencia de seguimiento de los algoritmos DRL está alrededor del 74.6 %, mientras que la propia



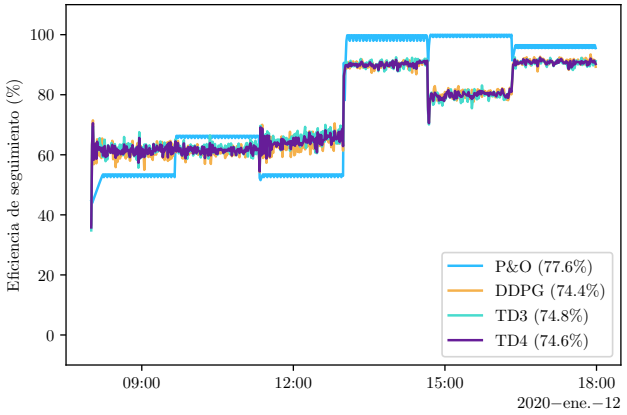
(a) Eficiencia de seguimiento en el segundo día. Resultados de un único experimento por algoritmo.



(b) Eficiencia de seguimiento en el segundo día. Resultados del promedio de 100 experimentos por algoritmo.



(c) Eficiencia de seguimiento en el doceavo día. Resultados de un único experimento por algoritmo.



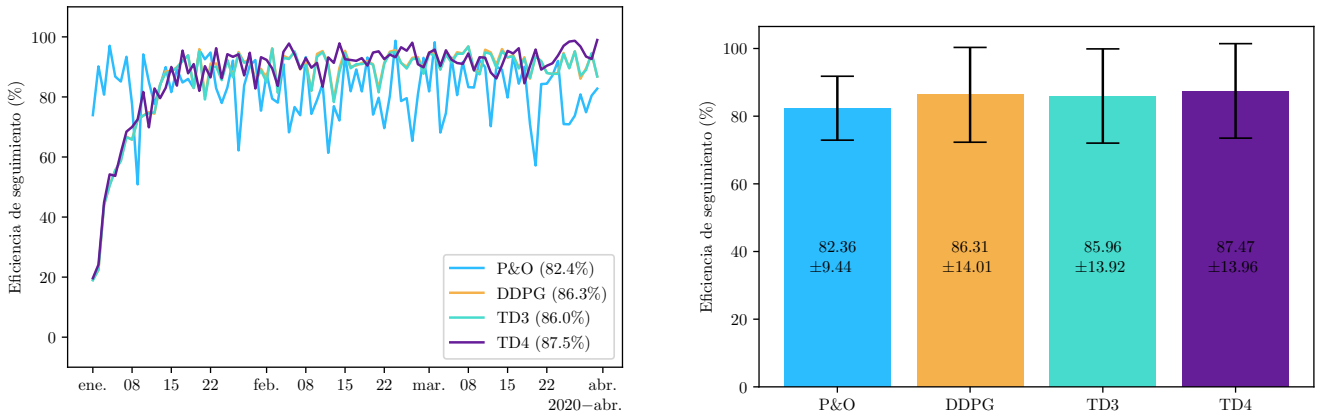
(d) Eficiencia de seguimiento en el doceavo día. Resultados del promedio de 100 experimentos por algoritmo.

Figura 4.6: Promedio de la eficiencia de seguimiento en los datos de entrenamiento al inicio del proceso de aprendizaje.

del algoritmo P&O es de alrededor del 77.6 %. A pesar de que la eficiencia de los algoritmos DRL aumenta, en comparación el día 2 de enero de 2020, la inclusión de demostraciones en el entrenamiento de TD4 no ha mejorado el desempeño de este algoritmo.

En la fase final del entrenamiento, la eficiencia de los algoritmos DRL sobrepasa el rendimiento del algoritmo P&O, como lo muestra la Fig. 4.7(a). Más aún, a partir del día 20 de marzo de 2020, la eficiencia de seguimiento del algoritmo TD4 es mayor que la de los algoritmos DDPG y TD3, lo que indica que la inclusión de demostraciones en el entrenamiento de TD4 mejora el desempeño de este algoritmo en la fase de aprendizaje. Finalmente, la Fig. 4.7(b) muestra la eficiencia de seguimiento de todos los algoritmos MPPT/GMPPT comparados, promediada en 100 experimentos independientes para cada algoritmo y a lo largo de los 100 días de entrenamiento, donde se muestra que la eficiencia de TD4 es superior a la de los otros algoritmos, con una eficiencia de seguimiento promedio de 87.47 % (SD 13.96); mientras que para TD3, DDPG y P&O, la eficiencia de seguimiento promedio

es de 85.96 % (SD 13.92), 86.31 % (SD 14.01) y 82.36 % (SD 9.44), respectivamente. Estos resultados incluyen la fase de exploración, por lo que se espera que en la fase de prueba, la eficiencia de los algoritmos GMPPT DRL se incremente considerablemente.



(a) Promedio de eficiencia por día.

(b) Promedio de eficiencia total por algoritmo.

Figura 4.7: Promedio de eficiencia de seguimiento en el conjunto de datos de entrenamiento, utilizando 100 experimentos por algoritmo.

4.5.2. Fase de Prueba

En esta sección se analiza el rendimiento de los algoritmos GMPPT basados en DRL (TD4, TD3 y DDPG) después de haber concluido la fase de entrenamiento, es decir, los pesos del actor y del crítico ya no se modifican. También se realiza una comparación con el rendimiento del algoritmo MPPT P&O y con la potencia teórica máxima para cada condición climática de operación.

La Fig. 4.8(a) muestra el rendimiento de todos los algoritmos MPPT/GMPPT para el primer día en el conjunto de prueba, correspondiente al día 1 de enero de 2021. Estos son los resultados de un experimento en particular, donde se observa que la eficiencia los algoritmos GMPPT DRL supera ampliamente el rendimiento del algoritmo P&O. Específicamente, la mejor eficiencia de seguimiento para este día en particular es la del algoritmo TD4, la cual es de 98.9 %, seguida por la eficiencia de seguimiento del algoritmo TD3, la cual es de 98 %; en tercer lugar, la eficiencia de seguimiento del algoritmo DDPG, correspondiente al 95 %; y en cuarto lugar, la eficiencia de seguimiento del algoritmo P&O, con el 88.8 %.

En cuanto a la velocidad de seguimiento de los algoritmos, se observa que todos los algoritmos DRL presentan una mayor velocidad que el algoritmo P&O. Esto se hace evidente a las 08:00 horas del primer día de prueba, mostrado en la Fig. 4.8(a), donde se observa que los algoritmos DRL alcanzan el GMPPT en el minuto siguiente, mientras que el algoritmo P&O demora 60 minutos en llegar a éste, coincidiendo con los métodos DRL alrededor de las 09:00 horas.

Respecto al comportamiento en estado estable de los algoritmos MPPT/GMPPT comparados, se observa que

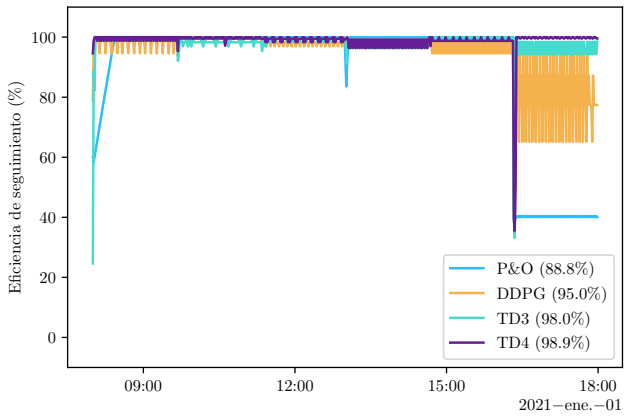
el algoritmo DDPG muestra oscilaciones en la eficiencia de seguimiento, causadas por la oscilación en el ciclo de trabajo, mientras que los algoritmos TD3 y TD4 no presentan oscilaciones. Esto se debe a que la inclusión de otro crítico en el proceso de entrenamiento regulariza los pesos en el actor de estos dos algoritmos, lo que se explica en las secciones 3.3.3 y 3.3.4.

Adicionalmente, la Fig. 4.8(a) muestra un fenómeno en el comportamiento de todos los algoritmos MPPT comparados, donde entre las 16:00 y las 17:00 horas se observa un súbito decremento en la eficiencia de seguimiento. Esto fenómeno, explicado en la Fig. 2.5a y Fig. 2.5b, se debe al cambio repentino en la irradiancia, lo que ocasiona que, tanto la curva I–V del sistema como su GMPPT se desplacen instantáneamente. No obstante, se observa que el algoritmo es capaz de volver a ubicar el GMPPT de manera inmediata, mientras que los otros algoritmos no logran establecerse correctamente en dicho punto.

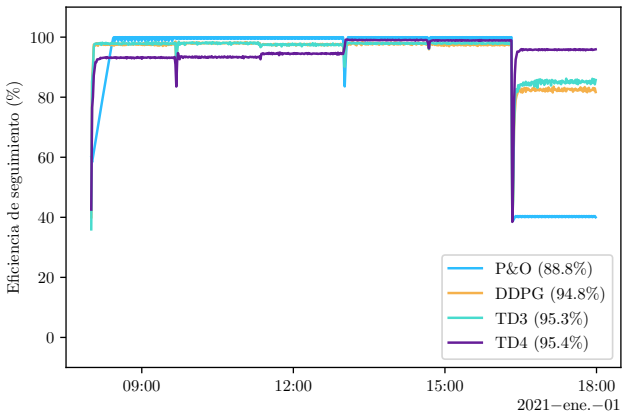
En la Fig. 4.8(b) se muestra el desempeño de los algoritmos GMPPT DRL promediado en 100 experimentos independientes para cada algoritmo, donde se observa que la brecha entre los algoritmos GMPPT DRL se estrecha considerablemente; sin embargo, el algoritmo TD4 sigue presentando una eficiencia de seguimiento superior a la del resto de algoritmos MPPT/GMPPT, con una eficiencia de seguimiento promedio de 95.4 %; mientras que para TD3, DDPG y P&O, la eficiencia de seguimiento promedio es de 95.3 %, 94.8 % y 88.8 %, respectivamente.

Similarmente, las Fig. 4.8(c) y 4.8(d) muestran el rendimiento de los algoritmos MPPT/GMPPT para el día 8 de enero de 2021, correspondiente al octavo día en el conjunto de prueba. En el caso de la Fig. 4.8(c), se presentan los resultados para un experimento en particular, donde TD4 es el algoritmo GMPPT que presenta la mejor eficiencia de seguimiento, correspondiente al 93.7 %; mientras que para TD3, DDPG y P&O, la eficiencia de seguimiento es de 89.8 %, 83.4 % y 65 %, respectivamente. En cuanto a la Fig. 4.8(d), se muestra el rendimiento promedio en 100 experimentos independientes, con resultados similares; el algoritmo TD4 presenta la mejor eficiencia promedio, correspondiente al 89.5 %; mientras que para TD3, DDPG y P&O, la eficiencia promedio es de 86.7 %, 85.9 % y 65 %, respectivamente. Al igual que en el día 01 de enero de 2021, la dinámica de seguimiento de DDPG en el día 08 de enero de 2021 presenta oscilaciones, las cuales se atenúan en TD3 y TD4.

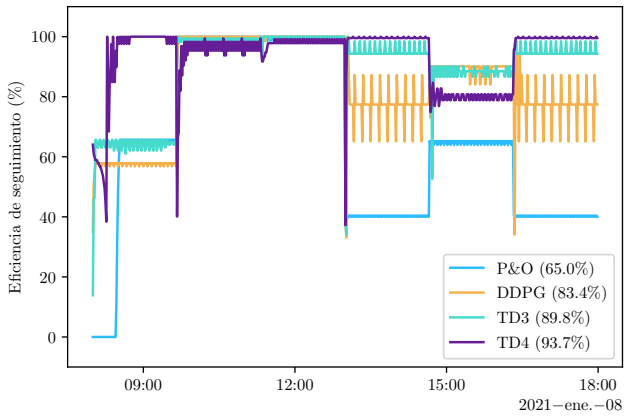
A pesar de que el desempeño promedio durante el día del algoritmo TD4 es mejor que el de los otros algoritmos MPPT comparados, en momentos específicos del día puede verse superado por estos otros algoritmos, como se observa desde las 09:00 hasta pasadas las 12:00 horas en la Fig. 4.8(b), y entre las 10:00 y las 13:00 horas en la Fig. 4.8(d). Este comportamiento se debe a que los algoritmos DRL están basados en tres procesos estocásticos (la inicialización de los pesos de las redes neuronales, la optimización de los pesos durante el entrenamiento, y la selección de acciones de exploración por parte de la política), los cuales pueden converger a distintos óptimos locales dependiendo de las condiciones iniciales. Adicionalmente, los datos utilizados durante la prueba pueden contener información adicional (i.e., estados no vistos durante el entrenamiento), por lo que el desempeño del algoritmo puede degradarse en momentos determinados. Una posible solución a este problema es incrementar el número de episodios de demostración y de entrenamiento, de tal manera que el algoritmo incremente su conocimiento del espacio de



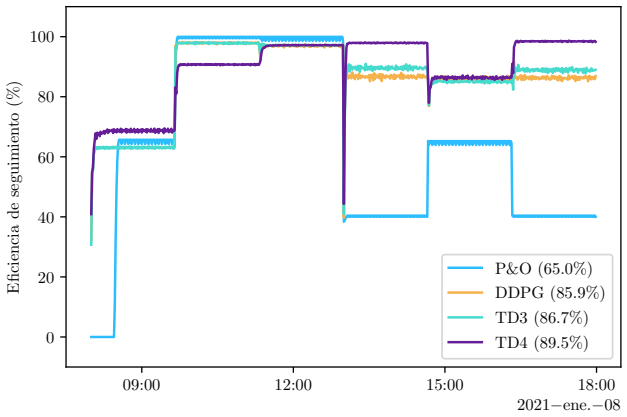
(a) Eficiencia de seguimiento en el primer día. Resultados de un único experimento por algoritmo.



(b) Eficiencia de seguimiento en el primer día. Resultados del promedio de 100 experimentos por algoritmo.



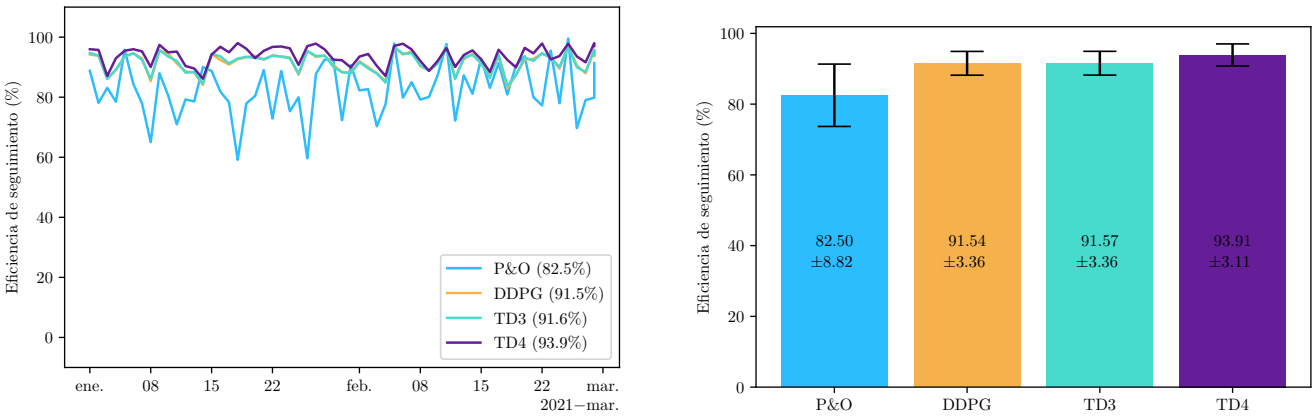
(c) Eficiencia de seguimiento en el octavo día. Resultados de un único experimento por algoritmo.



(d) Eficiencia de seguimiento en el octavo día. Resultados del promedio de 100 experimentos por algoritmo.

Figura 4.8: Eficiencia de seguimiento en los datos de prueba, una vez finalizado el proceso de entrenamiento. El promedio de la eficiencia de seguimiento en el día se indica junto al nombre del algoritmo.

estado y pueda generalizar de mejor manera. No obstante, como se muestra en la Fig. 4.9, el desempeño promedio de TD4 en 100 experimentos independientes a lo largo de los 60 días del conjunto de prueba es mejor que el de los algoritmos TD3, DDPG y P&O. En la Fig. 4.9(a) se presenta el promedio por día, donde se puede observar que los algoritmos DRL sobrepasan el rendimiento del algoritmo P&O, siendo TD4 el algoritmo que presenta la mejor eficiencia en promedio. En la Fig. 4.9(b) se muestra la eficiencia de seguimiento promedio por algoritmo, donde TD4 obtiene el primer lugar, con una eficiencia promedio de 93.91 % (SD 3.11); mientras que el resto de los algoritmos tienen eficiencias de 91.57 % (SD 3.36), 91.54 % (SD 3.36) y 82.5 % (SD 8.82), para TD3, DDPG y P&O, respectivamente.



(a) Promedio de eficiencia por día.

(b) Promedio de eficiencia total por algoritmo.

Figura 4.9: Promedio de eficiencia de seguimiento en el conjunto de datos de prueba, utilizando 100 experimentos por algoritmo.

A partir de estos resultados, se observa que el algoritmo TD4 presenta la mejor eficiencia promedio, tanto en la fase de aprendizaje, como en la fase de prueba. En consecuencia, los sistemas PV controlados por un algoritmo GMPPT TD4 pueden producir una mayor cantidad de potencia en las mismas condiciones ambientales que los otros algoritmos MPPT, lo que resulta de gran relevancia para la constante expansión de la generación PV, y simultáneamente, para promover la descarbonización de la energía.

4.6. Conclusión

La tarea de MPPT en un sistema PV tiene la finalidad de maximizar la cantidad de energía generada. Esta tarea no resulta sencilla ya que el MPP varía continuamente a lo largo del día en función de la irradiancia solar incidente y la temperatura. Por otra parte, PS es una condición de operación frecuente en la que los módulos de un arreglo fotovoltaico no reciben la misma cantidad de irradiancia solar, lo que produce pérdidas en la potencia generada.

Para mitigar las pérdidas en condiciones de PS, se utilizan diodos en derivación para aislar las celdas sombreadas. Sin embargo, la introducción de estos diodos en el sistema PV genera la aparición de múltiples MPP locales (LMPP), de los cuales solo uno corresponde al máximo global (GMPP). Como consecuencia, la tarea de GMPPT en condiciones de PS resulta más compleja, pudiendo incurrir en pérdidas adicionales de potencia si el algoritmo no identifica correctamente el GMPP de los LMPP.

Para enfrentar este problema, en esta tesis se propone el uso de un algoritmo basado en DRL, específicamente una versión modificada del algoritmo TD3, el cual se ha denominado TD4 (TD3+Demostraciones). TD4 incorpora demostraciones de un algoritmo MPPT P&O para guiar el proceso de aprendizaje y mejorar el desempeño del algoritmo en la fase de prueba.

Se realizaron distintas simulaciones para verificar el desempeño del algoritmo propuesto, comparándolo con tres

técnicas MPPT/GMPPT: DDPG, TD3 y P&O. Debido a la naturaleza estocástica del proceso de entrenamiento, en específico en la asignación inicial de los pesos en las ANNs de los algoritmos DRL, se realizaron 100 experimentos para cada algoritmo, y se promedian los resultados. Durante la fase de aprendizaje, la inclusión de demostraciones en TD4 permite que el algoritmo se entrene más rápidamente y mejore el desempeño. La eficiencia de seguimiento promedio durante la fase de aprendizaje para TD4 es de 87.47 %, mientras que para DDPG, TD3 y P&O, es de 86.31 %, 85.96 % y 82.36 %, respectivamente. En la fase de prueba, la eficiencia de seguimiento promedio de de 93.91 %, mientras que para DDPG, TD3 y P&O, es de 91.54 %, 91.57 % y 82.50 %, respectivamente.

A partir de los resultados de la simulación, es evidente que el algoritmo GMPPT TD4 es capaz de localizar el GMPP con mayor exactitud y rapidez que el algoritmo P&O y los otros métodos DRL que no incorporan demostraciones, lo que maximiza la cantidad de energía generada por el sistema PV.

Capítulo 5

Conclusiones y Trabajos Futuros

Esta sección final resume los principales hallazgos y resultados de la tesis, y brinda una perspectiva de los aspectos que merecen ser abordados en trabajos futuros.

5.1. Conclusiones

En esta tesis se demostró que la inclusión de demostraciones de un algoritmo MPPT P&O permite mejorar el desempeño de un algoritmo GMPPT basado en DRL. Esta formulación DRL con demostraciones se ha denominado TD4. La principal aportación científica de este método consiste en el uso de demostraciones sub-óptimas en el entrenamiento de un algoritmo TD3. Para discernir entre las acciones óptimas y sub-óptimas, se implementó un filtro de acción, el cual tiene la finalidad de utilizar unicamente las demostraciones consideradas beneficiosas y desechar aquellas demostraciones no apropiadas (i.e., demostraciones con comportamientos sub-óptimos). Esta configuración de aprendizaje en TD3 no ha sido explorada en la literatura.

Se utilizaron demostraciones de un algoritmo P&O debido a que es la técnica MPPT más popular y sus demostraciones son fáciles de obtener. Si bien, el algoritmo P&O no presenta un desempeño satisfactorio en condiciones de PS, el algoritmo propuesto TD4 logró beneficiarse de estas demostraciones al utilizar el filtro de acción.

Para verificar la eficacia de TD4 como técnica GMPPT, se comparó su desempeño con otras técnicas de seguimiento basadas en DRL (TD3 y DDPG) y con el propio algoritmo P&O utilizado para la generación de las demostraciones. Para realizar esta comparación, se implementó un sistema fotovoltaico compuesto por cuatro módulos en serie en MATLAB/Simulink y se utilizaron distintos patrones complejos de irradiancia solar para emular condiciones de sombreado parcial. Estos datos de irradiancia fueron generados aleatoriamente y separados en dos conjuntos, uno se utilizó para el entrenamiento de los algoritmos DRL y el otro se reservó para validar la eficacia de los métodos ante datos no vistos durante el entrenamiento. Los resultados de la simulación mostraron que TD4 tiene una mejor eficiencia de seguimiento del GMPP que los otros métodos, tanto en la fase de entrenamiento, como en la fase

de prueba. Si bien, el uso de aproximadores de función no lineales anula cualquier garantía de convergencia, los resultados experimentales demostraron que el aprendizaje es estable en todos los casos; es decir, el algoritmo TD4 logró aprender una política de seguimiento del GMPP satisfactoria en cada uno de los experimentos realizados.

A continuación se presentan las ventajas de TD4:

- En comparación con algoritmos con espacios de acción discreto (como P&O y DQN), TD4 utiliza ANNs para estimar la política y la función de valor, admitiendo el uso de espacios continuos de estado y acción. Esto permite que el algoritmo determine las acciones sobre el ciclo de trabajo del convertidor con mayor precisión.
- Con respecto a los algoritmos basados DRL (como DQN y DDPG), la inclusión de un crítico adicional mejora la estabilidad de convergencia en el proceso de optimización de los parámetros de las ANNs, lo que produce fluctuaciones más pequeñas en la potencia de salida. Esto se traduce directamente en un mejor desempeño en la tarea de seguimiento.
- Incorpora conocimiento extra en el entrenamiento a través del uso de demostraciones, lo que mejora razonablemente la eficiencia de seguimiento del GMPP. Por lo tanto, el sistema PV puede generar mayor energía, tanto en la fase de entrenamiento como en la fase de prueba, lo que puede generar un beneficio económico considerable en el funcionamiento a largo plazo.

En resumen, en comparación con el método tradicional P&O, y los métodos basados en DRL, DDPG y TD3, el método GMPPT TD4 propuesto tiene una mejor eficiencia de seguimiento, lo cual se traduce directamente como una mayor cantidad de potencia generada bajo las mismas condiciones ambientales. Los resultados obtenidos abren una vía prometedora para trabajos futuros y desarrollos en el área de DRL para abordar el complejo problema de GMPPT de los sistemas PV en condiciones de PS, lo que permitiría el continuo perfeccionamiento de esta fuente de generación renovable, limitando a la vez la dependencia actual del sector energético a los combustibles fósiles y promoviendo un ambiente libre de contaminación, donde predominen las energías limpias.

5.2. Trabajos futuros

A continuación se enlistan los trabajos que pueden desprenderse de esta investigación:

- En esta tesis, se desarrollaron y utilizaron modelos de simulación para verificar el rendimiento de la técnica GMPPT TD4 propuesta. Se espera que los resultados incentiven futuras investigaciones que involucren la construcción de modelos prototipo para probar experimentalmente la técnica GMPPT desarrollada.
- En el sistema PV estudiado, se utiliza un convertidor boost como interfaz entre los módulos fotovoltaicos y la carga. El efecto de otros convertidores DC-DC como Cuk, SEPIC o buck-boost podría investigarse para determinar si alguno de estos promueve una mejor eficiencia en el seguimiento del GMPP.

- Existen otros algoritmos MPPT/GMPPT que presentan mejor rendimiento que P&O por lo que el uso de éstos como experto puede mejorar la eficiencia de seguimiento de TD4.
- Referente a las redes inteligentes y el mercado eléctrico, se propone realizar una investigación que considere el almacenamiento de la energía generada y la interconexión del sistema con la red eléctrica. En este caso, además de maximizar la potencia generada, el controlador estaría a cargo de administrar el flujo de potencia, de manera que se minimice el costo de la energía obtenida de la red eléctrica.

Bibliografía

Aashoor, F. A. O. & Robinson, F. V. P. A variable step size perturb and observe algorithm for photovoltaic maximum power point tracking. In *2012 47th International Universities Power Engineering Conference (UPEC)*, pages 1–6, Sep. 2012. doi: 10.1109/UPEC.2012.6398612.

Abdel-Salam, M., El-Mohandes, M.-T., & Goda, M. An improved perturb-and-observe based MPPT method for PV systems under varying irradiation levels. *Solar Energy*, 171:547 – 561, 2018. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2018.06.080. URL `http://www.sciencedirect.com/science/article/pii/S0038092X18306315`.

Adesanya, A. A. & Schelly, C. Solar PV-diesel hybrid systems for the nigerian private sector: An impact assessment. *Energy Policy*, 132:196–207, 2019. ISSN 0301-4215. doi: https://doi.org/10.1016/j.enpol.2019.05.038. URL `https://www.sciencedirect.com/science/article/pii/S0301421519303428`.

Ahmed, J. & Salam, Z. An improved perturb and observe (P&O) maximum power point tracking (MPPT) algorithm for higher efficiency. *Applied Energy*, 150:97 – 108, 2015. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2015.04.006. URL `http://www.sciencedirect.com/science/article/pii/S0306261915004456`.

Ahmed, S. R. A., Sunny, A., & Rahman, S. Performance enhancement of sb2se3 solar cell using a back surface field layer: A numerical simulation approach. *Solar Energy Materials and Solar Cells*, 221:110919, 2021. ISSN 0927-0248. doi: https://doi.org/10.1016/j.solmat.2020.110919. URL `https://www.sciencedirect.com/science/article/pii/S092702482030516X`.

Akam, T. & Walton, M. E. What is dopamine doing in model-based reinforcement learning? *Current Opinion in Behavioral Sciences*, 38:74–82, 2021.

Al-Shahri, O. A., Ismail, F. B., Hannan, M., Lipu, M. H., Al-Shetwi, A. Q., Begum, R., Al-Muhsen, N. F., & Soujeri, E. Solar photovoltaic energy optimization methods, challenges and issues: A comprehensive review. *Journal of Cleaner Production*, 284:125465, 2021. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2020.125465. URL `https://www.sciencedirect.com/science/article/pii/S0959652620355116`.

Ameur, A., Berrada, A., Loudiyi, K., & Aggour, M. Forecast modeling and performance assessment of solar pv systems. *Journal of Cleaner Production*, 267:122167, 2020. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2020.122167. URL https://www.sciencedirect.com/science/article/pii/S0959652620322149.

Amisha, P. M., Pathania, M., & Rathaur, V. K. Overview of artificial intelligence in medicine. *Journal of family medicine and primary care*, 8(7):2328, 2019.

An, T. Study of a new type of electric car: Solar-powered car. *IOP Conference Series: Earth and Environmental Science*, 631(1):012118, jan 2021. doi: 10.1088/1755-1315/631/1/012118. URL https://doi.org/10.1088/1755-1315/631/1/012118.

Arcos-Vargas, A., Cansino, J. M., & Román-Collado, R. Economic and environmental analysis of a residential pv system: A profitable contribution to the paris agreement. *Renewable and Sustainable Energy Reviews*, 94:1024–1035, 2018. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2018.06.023. URL https://www.sciencedirect.com/science/article/pii/S136403211830457X.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

Avila, L., De Paula, M., Carlucho, I., & Sanchez Reinoso, C. MPPT for PV systems using deep reinforcement learning algorithms. *IEEE Latin America Transactions*, 17(12):2020–2027, 2019. doi: 10.1109/TLA.2019.9011547.

Avila, L., De Paula, M., Trimboli, M., & Carlucho, I. Deep reinforcement learning approach for mppt control of partially shaded pv systems in smart grids. *Applied Soft Computing*, 97:106711, 2020. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2020.106711. URL http://www.sciencedirect.com/science/article/pii/S1568494620306499.

Barrueto Guzmán, A., Barraza Vicencio, R., Ardila-Rey, J. A., Núñez Ahumada, E., González Araya, A., & Arancibia Moreno, G. A cost-effective methodology for sizing solar PV systems for existing irrigation facilities in Chile. *Energies*, 11(7), 2018. ISSN 1996-1073. doi: 10.3390/en11071853. URL https://www.mdpi.com/1996-1073/11/7/1853.

Bavarinos, K., Dounis, A., & Kofinas, P. Maximum power point tracking based on reinforcement learning using evolutionary optimization algorithms. *Energies*, 14(2), 2021. ISSN 1996-1073. doi: 10.3390/en14020335. URL https://www.mdpi.com/1996-1073/14/2/335.

Başoğlu, M. E. & Çakır, B. Comparisons of mppt performances of isolated and non-isolated DC-DC converters by using a new approach. *Renewable and Sustainable Energy Reviews*, 60:1100–1113, 2016. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2016.01.128. URL https://www.sciencedirect.com/science/article/pii/S1364032116001842.

Bellebaum, J., Korner-Nievergelt, F., Dürr, T., & Mammen, U. Wind turbine fatalities approach a level of concern in a raptor population. *Journal for Nature Conservation*, 21(6):394–400, 2013.

Bengio, Y. Practical recommendations for gradient-based training of deep architectures, 2012. URL `https://arxiv.org/abs/1206.5533`.

Bilgili, F., Koçak, E., & Ümit Bulut. The dynamic impact of renewable energy consumption on CO2 emissions: A revisited environmental Kuznets Curve approach. *Renewable and Sustainable Energy Reviews*, 54:838–845, 2016. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2015.10.080. URL `https://www.sciencedirect.com/science/article/pii/S1364032115011594`.

Birhanie, H. M., Messous, M. A., Senouci, S.-M., Aglzim, E.-H., & Ahmed, A. M. MDP-based resource allocation scheme towards a vehicular fog computing with energy constraints. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018. doi: 10.1109/GLOCOM.2018.8648081.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. End to end learning for self-driving cars, 2016.

Bollipo, R. B., Mikkili, S., & Bonthagorla, P. K. Critical review on PV MPPT techniques: classical, intelligent and optimisation. *IET Renewable Power Generation*, 14(9):1433–1452, 2020.

Bottou, L. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

BP. Statistical review of world energy 2021, Jul 2021. URL `www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2021-full-report.pdf`. Accessed: 2021-10-01.

BP. Statistical review of world energy 2022, Jul 2021. URL `https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2022.pdf`. Accessed: 2022-05-09.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. Openai gym, 2016.

Broughton, J. B., Ybarra, C. E., & Nyer, P. U. The economics of residential solar panels: A comparison of energy charges for different load profiles, rate plans, and panel orientations. *American Journal of Industrial and Business Management*, 12(2):180–194, 2022.

Brown, D., Goo, W., Nagarajan, P., & Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In Chaudhuri, K. & Salakhutdinov, R., editors, *Proceedings of the 36th*

International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 783–792. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/brown19a.html.

Burmester, D., Rayudu, R., Seah, W., & Akinyele, D. A review of nanogrid topologies and technologies. *Renewable and Sustainable Energy Reviews*, 67:760–775, 2017. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2016.09.073. URL https://www.sciencedirect.com/science/article/pii/S1364032116305640.

Cai, X., Cao, H., Fang, X., Sun, J., & Yu, Y. A view for atmospheric unpredictability. *Frontiers in Earth Science*, 9, 2021. ISSN 2296-6463. doi: 10.3389/feart.2021.686832. URL https://www.frontiersin.org/article/10.3389/feart.2021.686832.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Chang, B. & Starcher, K. Evaluation of wind and solar energy investments in Texas. *Renewable Energy*, 132:1348–1359, 2019. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2018.09.037. URL https://www.sciencedirect.com/science/article/pii/S0960148118311005.

Chang, S., Wang, Q., Hu, H., Ding, Z., & Guo, H. An nnwc mppt-based energy supply solution for sensor nodes in buildings and its feasibility study. *Energies*, 12(1), 2019. ISSN 1996-1073. doi: 10.3390/en12010101. URL https://www.mdpi.com/1996-1073/12/1/101.

Chen, A. S., Nam, H., Nair, S., & Finn, C. Batch exploration with examples for scalable robotic reinforcement learning. *IEEE Robotics and Automation Letters*, 6(3):4401–4408, 2021. doi: 10.1109/LRA.2021.3068655.

Chen, J.-F., Do, Q. H., & Hsieh, H.-N. Training artificial neural networks by a hybrid pso-cs algorithm. *Algorithms*, 8(2):292–308, 2015. ISSN 1999-4893. doi: 10.3390/a8020292. URL https://www.mdpi.com/1999-4893/8/2/292.

Cheng, Y. & Yao, X. Carbon intensity reduction assessment of renewable energy technology innovation in China: A panel data model with cross-section dependence and slope heterogeneity. *Renewable and Sustainable Energy Reviews*, 135:110157, 2021. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2020.110157. URL http://www.sciencedirect.com/science/article/pii/S1364032120304482.

Chochliouros, I. P., Kourtis, M.-A., Spiliopoulou, A. S., Lazaridis, P., Zaharis, Z., Zarakovitis, C., & Kourtis, A. Energy efficiency concerns and trends in future 5g network infrastructures. *Energies*, 14(17), 2021. ISSN 1996-1073. doi: 10.3390/en14175392. URL https://www.mdpi.com/1996-1073/14/17/5392.

Chollet, F. *Deep learning with Python*. Simon and Schuster, 2021.

Chou, Yang, & Chen. Maximum power point tracking of photovoltaic system based on reinforcement learning. *Sensors*, 19(22):5054, Nov 2019. ISSN 1424-8220. doi: 10.3390/s19225054. URL http://dx.doi.org/10.3390/s19225054.

Chou, K.-Y., Yang, C.-S., & Chen, Y.-P. Reinforcement learning based maximum power point tracking control of partially shaded photovoltaic system. *Journal of Marine Science and Technology*, 28(5):13, 2020.

Ciulla, G., Lo Brano, V., Di Dio, V., & Cipriani, G. A comparison of different one-diode models for the representation of i–v characteristic of a pv cell. *Renewable and Sustainable Energy Reviews*, 32:684–696, 2014. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2014.01.027. URL https://www.sciencedirect.com/science/article/pii/S1364032114000380.

Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., & Abbeel, P. Model-based reinforcement learning via meta-policy optimization. In Billard, A., Dragan, A., Peters, J., & Morimoto, J., editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 617–629. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/clavera18a.html.

Cordova-Fajardo, M. A. & Tututi, E. S. Incorporating home appliances into a DC home nanogrid. *Journal of Physics: Conference Series*, 1221:012048, jun 2019. doi: 10.1088/1742-6596/1221/1/012048. URL https://doi.org/10.1088/1742-6596/1221/1/012048.

Coronato, A., Naeem, M., De Pietro, G., & Paragliola, G. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109:101964, 2020. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2020.101964. URL https://www.sciencedirect.com/science/article/pii/S093336572031229X.

Cortés, B., Tapia Sánchez, R., & Flores, J. J. Characterization of a polycrystalline photovoltaic cell using artificial neural networks. *Solar Energy*, 196:157 – 167, 2020. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2019.12.012. URL http://www.sciencedirect.com/science/article/pii/S0038092X19312265.

Cortés, B., Tapia, R., & Flores, J. J. A behavioral cloning based MPPT for photovoltaic systems: Learning through P&O demonstrations. In *2021 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, volume 5, pages 1–6, 2021. doi: 10.1109/ROPEC53248.2021.9668084.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL https://doi.org/10.1007/BF02551274.

Dankwa, S. & Zheng, W. *Twin-Delayed DDPG: A Deep Reinforcement Learning Technique to Model a Continuous Movement of an Intelligent Robot Agent*. Association for Computing Machinery, New York, NY, USA, 2019. ISBN 9781450376259. URL https://doi.org/10.1145/3387168.3387199.

Deisenroth, M. & Rasmussen, C. E. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472. Citeseer, 2011.

Ding, M., Lv, D., Yang, C., Li, S., Fang, Q., Yang, B., & Zhang, X. Global maximum power point tracking of pv systems under partial shading condition: A transfer reinforcement learning approach. *Applied Sciences*, 9(13): 2769, Jul 2019. ISSN 2076-3417. doi: 10.3390/app9132769. URL `http://dx.doi.org/10.3390/app9132769`.

Dogo, E., Afolabi, O., Nwulu, N., Twala, B., & Aigbavboa, C. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In *2018 international conference on computational techniques, electronics and mechanical systems (CTEMS)*, pages 92–99. IEEE, 2018.

Dong, Y. & Zou, X. Mobile robot path planning based on improved DDPG reinforcement learning algorithm. In *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*, pages 52–56, 2020. doi: 10.1109/ICSESS49938.2020.9237641.

Duan, J., Shi, D., Diao, R., Li, H., Wang, Z., Zhang, B., Bian, D., & Yi, Z. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Transactions on Power Systems*, 35(1):814–817, 2020. doi: 10.1109/TPWRS.2019.2941134.

Duchi, J., Hazan, E., & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, jul 2011. ISSN 1532-4435.

Duffie, J. A., Beckman, W. A., & Worek, W. *Solar engineering of thermal processes*, volume 3. Wiley Online Library, 2013.

Dupont, E., Koppelaar, R., & Jeanmart, H. Global available solar energy under physical and energy return on investment constraints. *Applied Energy*, 257:113968, 2020. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2019.113968. URL `https://www.sciencedirect.com/science/article/pii/S0306261919316551`.

Eckstein, F. Learning to fly - building an autopilot system based on neural networks and reinforcement learning, 06 2020.

Ember Climate. Global Electricity Review 2022, March 2022. URL `https://ember-climate.org/insights/research/global-electricity-review-2022/`. [Online; accesed 2022-05-07].

Ernst, D., Glavic, M., Capitanescu, F., & Wehenkel, L. Reinforcement learning versus model predictive control: A comparison on a power system problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):517–529, 2009. doi: 10.1109/TSMCB.2008.2007630.

Essig, S., Allebé, C., Remo, T., Geisz, J. F., Steiner, M. A., Horowitz, K., Barraud, L., Ward, J. S., Schnabel, M., Descoeudres, A., et al. Raising the one-sun conversion efficiency of III–V/Si solar cells to 32.8 % for two junctions and 35.9 % for three junctions. *Nature Energy*, 2(9):1–9, 2017.

Evans, G. W. Projected behavioral impacts of global climate change. *Annual Review of Psychology*, 70(1):449–474, 2019. doi: 10.1146/annurev-psych-010418-103023. URL https://doi.org/10.1146/annurev-psych-010418-103023. PMID: 29975596.

Even-Dar, E., Mansour, Y., & Bartlett, P. Learning rates for Q-learning. *Journal of machine learning Research*, 5 (1), 2003.

Fathy, A., Rezk, H., & Yousri, D. A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm. *Solar Energy*, 207:305–316, 2020. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2020.06.108. URL https://www.sciencedirect.com/science/article/pii/S0038092X20307234.

Fernández-Ahumada, L., Ramírez-Faz, J., López-Luque, R., Varo-Martínez, M., Moreno-García, I., & Casares de la Torre, F. Influence of the design variables of photovoltaic plants with two-axis solar tracking on the optimization of the tracking and backtracking trajectory. *Solar Energy*, 208:89–100, 2020. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2020.07.063. URL https://www.sciencedirect.com/science/article/pii/S0038092X20307957.

France, R. M., Geisz, J. F., Song, T., Olavarria, W., Young, M., Kibbler, A., & Steiner, M. A. Triple-junction solar cells with 39.5 % terrestrial and 34.2 % space efficiency enabled by thick quantum well superlattices, 2022. URL https://arxiv.org/abs/2203.15593.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018. ISSN 1935-8245. doi: 10.1561/2200000071. URL http://dx.doi.org/10.1561/2200000071.

Fthenakis, V. & Leccisi, E. Updated sustainability status of crystalline silicon-based photovoltaic systems: Life-cycle energy and environmental impact reduction trends. *Progress in Photovoltaics: Research and Applications*, 29(10): 1068–1077, 2021. doi: https://doi.org/10.1002/pip.3441. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pip.3441.

Fujimoto, S., van Hoof, H., & Meger, D. Addressing function approximation error in actor-critic methods, 2018. URL https://arxiv.org/abs/1802.09477.

Gao, C., Na, H., Song, K., Dyer, N., Tian, F., Xu, Q., & Xing, Y. Environmental impact analysis of power generation from biomass and wind farms in different locations. *Renewable and Sustainable Energy Reviews*, 102:307–317, 2019. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2018.12.018. URL https://www.sciencedirect.com/science/article/pii/S136403211830813X.

Gharibeh, H. F., Yazdankhah, A. S., Azizian, M. R., & Farrokhifar, M. Online energy management strategy for fuel cell hybrid electric vehicles with installed pv on roof. *IEEE Transactions on Industry Applications*, 57(3): 2859–2869, 2021. doi: 10.1109/TIA.2021.3061323.

Giusti, A., Guzzi, J., Cireşan, D. C., He, F., Rodríguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Caro, G. D., Scaramuzza, D., & Gambardella, L. M. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016. doi: 10.1109/LRA.2015.2509024.

Goecks, V. G. *Human-in-the-Loop Methods for Data-Driven and Reinforcement Learning Systems*. PhD thesis, Texas A&M University, 2020. URL `https://arxiv.org/abs/2008.13221`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Gow, J. A. & Manning, C. D. Development of a photovoltaic array model for use in power-electronics simulation studies. *IEE Proceedings - Electric Power Applications*, 146(2):193–200, March 1999. ISSN 1350-2352. doi: 10.1049/ip-epa:19990116.

Graves, A., Mohamed, A.-r., & Hinton, G. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013. doi: 10.1109/ICASSP.2013.6638947.

Gu, D., Andreev, K., & Dupre, M. E. Major trends in population growth around the world. *China CDC Weekly*, 3(28):604, 2021.

Haarnoja, T., Tang, H., Abbeel, P., & Levine, S. Reinforcement learning with deep energy-based policies, 2017. URL `https://arxiv.org/abs/1702.08165`.

Hadji, S., Gaubert, J.-P., & Krim, F. Real-time genetic algorithms-based mppt: Study and comparison (theoretical an experimental) with conventional methods. *Energies*, 11(2), 2018. ISSN 1996-1073. doi: 10.3390/en11020459. URL `https://www.mdpi.com/1996-1073/11/2/459`.

Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. *Neural network design*, volume 20. Pws Pub. Boston, 1996.

Hagan, M. T., Demuth, H. B., & Beale, M. H. *Neural Network Design*. Martin Hagan, 2nd edition, 2014.

Hanson, A. J., Deline, C. A., MacAlpine, S. M., Stauth, J. T., & Sullivan, C. R. Partial-shading assessment of photovoltaic installations via module-level monitoring. *IEEE Journal of Photovoltaics*, 4(6):1618–1624, 2014. doi: 10.1109/JPHOTOV.2014.2351623.

Hara, K., Saito, D., & Shouno, H. Analysis of function of rectified linear unit used in deep learning. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015. doi: 10.1109/IJCNN.2015.7280578.

Harrag, A. & Messalti, S. Ic-based variable step size neuro-fuzzy mppt improving pv system performances. *Energy Procedia*, 157:362 – 374, 2019. ISSN 1876-6102. doi: https://doi.org/10.1016/j.egypro.2018.11.201. URL http://www.sciencedirect.com/science/article/pii/S1876610218311706. Technologies and Materials for Renewable Energy, Environment and Sustainability (TMREES).

Heidari, A. A., Faris, H., Mirjalili, S., Aljarah, I., & Mafarja, M. *Ant Lion Optimizer: Theory, Literature Review, and Application in Multi-layer Perceptron Neural Networks*, pages 23–46. Springer International Publishing, Cham, 2020. ISBN 978-3-030-12127-3. doi: 10.1007/978-3-030-12127-3_3. URL https://doi.org/10.1007/978-3-030-12127-3_3.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11694.

Hinton, G., Srivastava, N., & Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

Hou, Y., Liu, L., Wei, Q., Xu, X., & Chen, C. A novel DDPG method with prioritized experience replay. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 316–321, 2017. doi: 10.1109/SMC.2017.8122622.

Hsu, R. C., Liu, C.-T., Chen, W.-Y., Hsieh, H.-I., & Wang, H.-L. A reinforcement learning-based maximum power point tracking method for photovoltaic array. *International Journal of Photoenergy*, 2015, 2015.

Hsu, R. C., Chen, W.-Y., & Lin, Y.-P. A Q-learning based maximum power point tracking for PV array under partial shading condition. In Arai, K., Kapoor, S., & Bhatia, R., editors, *Intelligent Computing*, pages 155–168, Cham, 2020. Springer International Publishing. ISBN 978-3-030-52246-9.

Huang, J., Juan, R., Gomez, R., Nakamura, K., Sha, Q., He, B., & Li, G. Gan-based interactive reinforcement learning from demonstration and human evaluative feedback, 2021. URL https://arxiv.org/abs/2104.06600.

Ibn-Mohammed, T., Koh, S., Reaney, I., Acquaye, A., Schileo, G., Mustapha, K., & Greenough, R. Perovskite solar cells: An integrated hybrid lifecycle assessment and review in comparison with other photovoltaic technologies. *Renewable and Sustainable Energy Reviews*, 80:1321–1344, 2017. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2017.05.095. URL https://www.sciencedirect.com/science/article/pii/S1364032117307311.

Imran, M., Ghannam, R., & Abbasi, Q. *Engineering and Technology for Healthcare*, chapter 7, pages 129–152. Wiley - IEEE. Wiley, 2020. ISBN 9781119644248.

International Energy Agency. World energy outlook 2020, Oct 2020. URL www.iea.org/reports/world-energy-outlook-2020. Accessed: 2021-08-31.

International Energy Agency. World Energy Outlook 2021, October 2021. URL https://iea.blob.core.windows.net/assets/4ed140c1-c3f3-4fd9-acae-789a4e14a23c/WorldEnergyOutlook2021.pdf. [Online; accesed 2022-05-01].

Ishaque, K., Salam, Z., & Lauss, G. The performance of perturb and observe and incremental conductance maximum power point tracking method under dynamic weather conditions. *Applied Energy*, 119:228 – 236, 2014. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2013.12.054. URL http://www.sciencedirect.com/science/article/pii/S0306261913010635.

Islam, R., Henderson, P., Gomrokchi, M., & Precup, D. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control, 2017. URL https://arxiv.org/abs/1708.04133.

Jacquet, J. & Jamieson, D. Soft but significant power in the Paris Agreement. *Nature Climate Change*, 6(7): 643–646, 2016.

Janiesch, C., Zschech, P., & Heinrich, K. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.

Jeong, G. & Kim, H. Y. Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning. *Expert Systems with Applications*, 117:125–138, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2018.09.036. URL https://www.sciencedirect.com/science/article/pii/S0957417418306134.

Jiang, L., Huang, H., & Ding, Z. Path planning for intelligent robots based on deep Q-learning with experience replay and heuristic knowledge. *IEEE/CAA Journal of Automatica Sinica*, 7(4):1179–1189, 2019.

Jordan, M. I. & Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. doi: 10.1126/science.aaa8415. URL https://www.science.org/doi/abs/10.1126/science.aaa8415.

Joseph, S. C., Mohammed, A. A., Dhanesh, P. R., & Ashok, S. Smart power management for dc nanogrid based building. In *2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 142–146, 2018. doi: 10.1109/RAICS.2018.8635070.

Kalogerakis, C., Koutroulis, E., & Lagoudakis, M. G. Global MPPT based on machine-learning for PV arrays operating under partial shading conditions. *Applied Sciences*, 10(2), 2020. ISSN 2076-3417. doi: 10.3390/app10020700. URL https://www.mdpi.com/2076-3417/10/2/700.

Kang, S.-J., Ko, J.-S., Choi, J.-S., Jang, M.-G., Mun, J.-H., Lee, J.-G., & Chung, D.-H. A novel MPPT control of photovoltaic system using FLC algorithm. In *2011 11th International Conference on Control, Automation and Systems*, pages 434–439, 2011.

Kannan, N. & Vakeesan, D. Solar energy for future world: - a review. *Renewable and Sustainable Energy Reviews*, 62:1092–1105, 2016. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2016.05.022. URL https://www.sciencedirect.com/science/article/pii/S1364032116301320.

Karami, N., Moubayed, N., & Outbib, R. General review and classification of different MPPT techniques. *Renewable and Sustainable Energy Reviews*, 68:1–18, 2017. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2016.09.132. URL https://www.sciencedirect.com/science/article/pii/S1364032116306438.

Kato, T., Wu, J., Hirai, Y., Sugimoto, H., & Bermudez, V. Record efficiency for thin-film polycrystalline solar cells up to 22.9 % achieved by Cs-treated Cu(In,Ga)(Se,S)2. *IEEE Journal of Photovoltaics*, 9(1):325–330, 2019. doi: 10.1109/JPHOTOV.2018.2882206.

Kaul, V., Enslin, S., & Gross, S. A. History of artificial intelligence in medicine. *Gastrointestinal endoscopy*, 92(4): 807–812, 2020.

Kelly, A., O'Sullivan, A., de Mars, P., & Marot, A. Reinforcement learning for electricity network operation, 2020. URL https://arxiv.org/abs/2003.07339.

Khadka, S. & Tumer, K. Evolution-guided policy gradient in reinforcement learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/85fc37b18c57097425b52fc7afbb6969-Paper.pdf.

Khan, I., Hou, F., & Le, H. P. The impact of natural resources, energy consumption, and population growth on environmental quality: Fresh evidence from the United States of America. *Science of The Total Environment*, 754:142222, 2021. ISSN 0048-9697. doi: https://doi.org/10.1016/j.scitotenv.2020.142222. URL https://www.sciencedirect.com/science/article/pii/S004896972035751X.

Khan, R., Khan, L., Ullah, S., Sami, I., & Ro, J.-S. Backstepping based super-twisting sliding mode MPPT control with differential flatness oriented observer design for photovoltaic system. *Electronics*, 9(9):1543, 2020.

Kim, C.-W., Park, G.-Y., Shin, J.-C., & Kim, H.-J. Efficiency enhancement of gaas single-junction solar cell by nanotextured window layer. *Applied Sciences*, 12(2), 2022. ISSN 2076-3417. doi: 10.3390/app12020601. URL https://www.mdpi.com/2076-3417/12/2/601.

Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., & Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–18, 2021. doi: 10.1109/TITS.2021.3054625.

Kiumarsi, B., Vamvoudakis, K. G., Modares, H., & Lewis, F. L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2042–2062, 2018. doi: 10.1109/TNNLS.2017.2773458.

Koengkan, M., Fuinhas, J. A., & Silva, N. Exploring the capacity of renewable energy consumption to reduce outdoor air pollution death rate in Latin America and the Caribbean region. *Environmental Science and Pollution Research*, 28(2):1656–1674, 2021.

Kofinas, P., Doltsinis, S., Dounis, A., & Vouros, G. A reinforcement learning approach for mppt control method of photovoltaic sources. *Renewable Energy*, 108:461 – 473, 2017. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2017.03.008. URL `http://www.sciencedirect.com/science/article/pii/S0960148117301891`.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Lapan, M. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd, 2018.

LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Levenda, A., Behrsin, I., & Disano, F. Renewable energy for whom? a global systematic review of the environmental justice implications of renewable energy technologies. *Energy Research & Social Science*, 71:101837, 2021. ISSN 2214-6296. doi: https://doi.org/10.1016/j.erss.2020.101837. URL `https://www.sciencedirect.com/science/article/pii/S2214629620304126`.

Levine, S., Pastor, P., Krizhevsky, A., & Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, 2016.

Li, S., Liao, H., Yuan, H., Ai, Q., & Chen, K. A MPPT strategy with variable weather parameters through analyzing the effect of the DC/DC converter to the MPP of PV system. *Solar Energy*, 144:175 – 184, 2017. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2017.01.002. URL `http://www.sciencedirect.com/science/article/pii/S0038092X17300117`.

Li, Y. Deep reinforcement learning: An overview, 2017. URL `https://arxiv.org/abs/1701.07274`.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. Continuous control with deep reinforcement learning, 2019.

Lin, D., Li, X., & Ding, S. Maximum power point tracking based on reinforcement learning in photovoltaic system. In *2020 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–6, 2020. doi: 10.1109/PEDES49360.2020.9379644.

Lin, D., Li, X., Ding, S., Wen, H., Du, Y., & Xiao, W. Self-tuning MPPT scheme based on reinforcement learning and beta parameter in photovoltaic power systems. *IEEE Transactions on Power Electronics*, 36(12):13826–13838, 2021a. doi: 10.1109/TPEL.2021.3089707.

Lin, Y., McPhee, J., & Azad, N. L. Comparison of deep reinforcement learning and model predictive control for adaptive cruise control. *IEEE Transactions on Intelligent Vehicles*, 6(2):221–231, 2021b. doi: 10.1109/TIV.2020.3012947.

Liu, T., Li, L., Shao, G., Wu, X., & Huang, M. A novel policy gradient algorithm with pso-based parameter exploration for continuous control. *Engineering Applications of Artificial Intelligence*, 90:103525, 2020a. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2020.103525. URL `https://www.sciencedirect.com/science/article/pii/S0952197620300324`.

Liu, Y., Gao, Y., & Yin, W. An improved analysis of stochastic gradient descent with momentum. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., & Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18261–18271. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper/2020/file/d3f5d4de09ea19461dab00590df91e4f-Paper.pdf`.

Lopez, M. M. & Kalita, J. Deep learning applied to NLP, 2017.

Luchini, C., Pea, A., & Scarpa, A. Artificial intelligence in oncology: current applications and future perspectives. *British journal of cancer*, 126(1):4–9, 2022.

Mansoor, M., Mirza, A. F., & Ling, Q. Harris hawk optimization-based MPPT control for PV systems under partial shading conditions. *Journal of Cleaner Production*, 274:122857, 2020. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2020.122857. URL `https://www.sciencedirect.com/science/article/pii/S0959652620329024`.

Mao, M., Cui, L., Zhang, Q., Guo, K., Zhou, L., & Huang, H. Classification and summarization of solar photovoltaic MPPT techniques: A review based on traditional and intelligent control strategies. *Energy Reports*, 6:1312–1327, 2020. ISSN 2352-4847. doi: https://doi.org/10.1016/j.egyr.2020.05.013. URL `https://www.sciencedirect.com/science/article/pii/S2352484720300512`.

Masadeh, A., Wang, Z., & Kamal, A. E. Reinforcement learning exploration algorithms for energy harvesting communications systems. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018. doi: 10.1109/ICC.2018.8422710.

Mason, K. & Grijalva, S. A review of reinforcement learning for autonomous building energy management. *Computers & Electrical Engineering*, 78:300–312, 2019. ISSN 0045-7906. doi: https://doi.org/10.1016/j.compeleceng.2019.07.019. URL https://www.sciencedirect.com/science/article/pii/S0045790618333421.

Mathijsen, D. The role of composites in getting the solar car to our driveways: Lightyear one. *Reinforced Plastics*, 65(4):178–187, 2021. ISSN 0034-3617. doi: https://doi.org/10.1016/j.repl.2021.06.001. URL https://www.sciencedirect.com/science/article/pii/S0034361721001582.

May, R., Jackson, C. R., Middel, H., Stokke, B. G., & Verones, F. Life-cycle impacts of wind energy development on bird diversity in Norway. *Environmental Impact Assessment Review*, 90:106635, 2021. ISSN 0195-9255. doi: https://doi.org/10.1016/j.eiar.2021.106635. URL https://www.sciencedirect.com/science/article/pii/S0195925521000858.

Mbungu, N. T., Naidoo, R. M., Bansal, R. C., Siti, M. W., & Tungadio, D. H. An overview of renewable energy resources and grid integration for commercial building applications. *Journal of Energy Storage*, 29:101385, 2020. ISSN 2352-152X. doi: https://doi.org/10.1016/j.est.2020.101385. URL https://www.sciencedirect.com/science/article/pii/S2352152X19316962.

Meng, T. L. & Khushi, M. Reinforcement learning in financial markets. *Data*, 4(3), 2019. ISSN 2306-5729. doi: 10.3390/data4030110. URL https://www.mdpi.com/2306-5729/4/3/110.

Messalti, S., Harrag, A. G., & Loukriz, A. E. A new neural networks MPPT controller for PV systems. In *IREC2015 The Sixth International Renewable Energy Congress*, pages 1–6, March 2015. doi: 10.1109/IREC.2015.7110907.

Messenger, R. A. & Ventre, J. *Photovoltaic Systems Engineering*. CRC press, 2004.

Mikkelsen, T. N., Beier, C., Jonasson, S., Holmstrup, M., Schmidt, I. K., Ambus, P., Pilegaard, K., Michelsen, A., Albert, K., Andresen, L. C., Arndal, M. F., Bruun, N., Christensen, S., Danbæk, S., Gundersen, P., Jørgensen, P., Linden, L. G., Kongstad, J., Maraldo, K., Priemé, A., Riis-Nielsen, T., Ro-Poulsen, H., Stevnbak, K., Selsted, M. B., Sørensen, P., Larsen, K. S., Carter, M. S., Ibrom, A., Martinussen, T., Miglietta, F., & Sverdrup, H. Experimental design of multifactor climate change experiments with elevated co2, warming and drought: the climaite project. *Functional Ecology*, 22(1):185–195, 2008. doi: 10.1111/j.1365-2435.2007.01362.x. URL https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2435.2007.01362.x.

Mirza, A. F., Mansoor, M., & Ling, Q. A novel MPPT technique based on Henry gas solubility optimization. *Energy Conversion and Management*, 225:113409, 2020. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2020.113409. URL https://www.sciencedirect.com/science/article/pii/S0196890420309444.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M.,

Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518 (7540):529–533, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning, 2016. URL https://arxiv.org/abs/1602.01783.

Moerland, T. M., Broekens, J., Plaat, A., & Jonker, C. M. Model-based reinforcement learning: A survey, 2020. URL https://arxiv.org/abs/2006.16712.

Mohapatra, A., Nayak, B., Das, P., & Mohanty, K. B. A review on MPPT techniques of PV system under partial shading condition. *Renewable and Sustainable Energy Reviews*, 80:854–867, 2017. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2017.05.083. URL https://www.sciencedirect.com/science/article/pii/S1364032117307256.

Mosavi, A., Faghan, Y., Ghamisi, P., Duan, P., Ardabili, S. F., Salwana, E., & Band, S. S. Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics*, 8(10), 2020. ISSN 2227-7390. doi: 10.3390/math8101640. URL https://www.mdpi.com/2227-7390/8/10/1640.

Motahhir, S., El Hammoumi, A., & El Ghzizal, A. The most used MPPT algorithms: Review and the suitable low-cost embedded board for each algorithm. *Journal of Cleaner Production*, 246:118983, 2020. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2019.118983. URL http://www.sciencedirect.com/science/article/pii/S0959652619338533.

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., & S., S. Deep learning for stock market prediction. *Entropy*, 22(8), 2020. ISSN 1099-4300. doi: 10.3390/e22080840. URL https://www.mdpi.com/1099-4300/22/8/840.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., & Abbeel, P. Overcoming exploration in reinforcement learning with demonstrations, 2018.

Nguyen, H. & La, H. Review of deep reinforcement learning for robot manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 590–595, 2019. doi: 10.1109/IRC.2019.00120.

Nikolic, D., Skerlic, J., Radulovic, J., Miskovic, A., Tamasauskas, R., & Sadauskienė, J. Exergy efficiency optimization of photovoltaic and solar collectors' area in buildings with different heating systems. *Renewable Energy*, 189:1063–1073, 2022. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2022.03.075. URL https://www.sciencedirect.com/science/article/pii/S0960148122003597.

Obeng, M., Gyamfi, S., Derkyi, N. S., Kabo-bah, A. T., & Peprah, F. Technical and economic feasibility of a 50 MW grid-connected solar PV at UENR Nsoatre Campus. *Journal of Cleaner Production*, 247:119159, 2020.

ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2019.119159. URL https://www.sciencedirect.com/science/article/pii/S0959652619340296.

Okafor, E. G., Udekwe, D., Ubadike, O. C., Okafor, E., Jemitola, P. O., & Abba, M. T. Photovoltaic system MPPT evaluation using classical, meta-heuristics, and reinforcement learning-based controllers: A comparative study. *Journal of Southwest Jiaotong University*, 56(3), 2021.

Paul, D. I. & Smyth, M. Enhancing the performance of a building integrated compound parabolic photovoltaic concentrator using a hybrid photovoltaic cell. *International Journal of Renewable Energy Technology*, 11(1):49–69, 2020. doi: 10.1504/IJRET.2020.106519. URL https://www.inderscienceonline.com/doi/abs/10.1504/IJRET.2020.106519.

Perera, A. & Kamalaruban, P. Applications of reinforcement learning in energy systems. *Renewable and Sustainable Energy Reviews*, 137:110618, 2021. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2020.110618. URL https://www.sciencedirect.com/science/article/pii/S1364032120309023.

Perez, M. & Perez, R. Update 2022 - a fundamental look at supply side energy reserves for the planet. *Solar Energy Advances*, 2:100014, 2022. ISSN 2667-1131. doi: https://doi.org/10.1016/j.seja.2022.100014. URL https://www.sciencedirect.com/science/article/pii/S266711312200002X.

Peters, J., Mulling, K., & Altun, Y. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Phan, B. C., Lai, Y.-C., & Lin, C. E. A deep reinforcement learning-based mppt control for pv systems under partial shading condition. *Sensors*, 20(11):3039, May 2020. ISSN 1424-8220. doi: 10.3390/s20113039. URL http://dx.doi.org/10.3390/s20113039.

Plaat, A., Kosters, W., & Preuss, M. High-accuracy model-based reinforcement learning, a survey, 2021. URL https://arxiv.org/abs/2107.08241.

Rabaia, M. K. H., Abdelkareem, M. A., Sayed, E. T., Elsaid, K., Chae, K.-J., Wilberforce, T., & Olabi, A. Environmental impacts of solar energy systems: A review. *Science of The Total Environment*, 754:141989, 2021. ISSN 0048-9697. doi: https://doi.org/10.1016/j.scitotenv.2020.141989. URL http://www.sciencedirect.com/science/article/pii/S0048969720355182.

Raghavendra, K. V. G., Zeb, K., Muthusamy, A., Krishna, T. N. V., Kumar, S. V. S. V. P., Kim, D.-H., Kim, M.-S., Cho, H.-G., & Kim, H.-J. A comprehensive review of DC-DC converter topologies and modulation strategies with recent advances in solar photovoltaic systems. *Electronics*, 9(1), 2020. ISSN 2079-9292. doi: 10.3390/electronics9010031. URL https://www.mdpi.com/2079-9292/9/1/31.

Raiman, J., Zhang, S., & Wolski, F. Long-term planning and situational awareness in OpenAI Five, 2019.

Ramesh, A., Kambhampati, C., Monson, J. R., & Drew, P. Artificial intelligence in medicine. *Annals of the Royal College of Surgeons of England*, 86(5):334, 2004.

Rauf, A., Al-Awami, A. T., Kassas, M., & Khalid, M. Optimal sizing and cost minimization of solar photovoltaic power system considering economical perspectives and net metering schemes. *Electronics*, 10(21), 2021. ISSN 2079-9292. doi: 10.3390/electronics10212713. URL https://www.mdpi.com/2079-9292/10/21/2713.

Raza, E. & Ahmad, Z. Review on two-terminal and four-terminal crystalline-silicon/perovskite tandem solar cells; progress, challenges, and future perspectives. *Energy Reports*, 8:5820–5851, 2022. ISSN 2352-4847. doi: https://doi.org/10.1016/j.egyr.2022.04.028. URL https://www.sciencedirect.com/science/article/pii/S2352484722007934.

Reyes-Belmonte, M. A. Quo vadis solar energy research? *Applied Sciences*, 11(7), 2021. ISSN 2076-3417. doi: 10.3390/app11073015. URL https://www.mdpi.com/2076-3417/11/7/3015.

Rhouma, M. B., Gastli, A., Brahim, L. B., Touati, F., & Benammar, M. A simple method for extracting the parameters of the PV cell single-diode model. *Renewable Energy*, 113:885 – 894, 2017. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2017.06.064. URL http://www.sciencedirect.com/science/article/pii/S0960148117305694.

Ritchie, H., Roser, M., & Rosado, P. Energy. *Our World in Data*, 2020. https://ourworldindata.org/energy.

Rizvi, S. A. A. & Lin, Z. Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback. *IEEE Transactions on Cybernetics*, 50(11):4670–4679, 2020. doi: 10.1109/TCYB.2018.2886735.

Ruder, S. An overview of gradient descent optimization algorithms, 2016. URL https://arxiv.org/abs/1609.04747.

Rühle, S. Tabulated values of the shockley–queisser limit for single junction solar cells. *Solar Energy*, 130:139–147, 2016. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2016.02.015. URL https://www.sciencedirect.com/science/article/pii/S0038092X16001110.

Sarkar, M. N. I. Effect of various model parameters on solar photovoltaic cell simulation: a spice analysis. *Renewables: Wind, Water, and Solar*, 3(1):13, Aug 2016. ISSN 2198-994X. doi: 10.1186/s40807-016-0035-3. URL https://doi.org/10.1186/s40807-016-0035-3.

Sarwar, S., Javed, M. Y., Jaffery, M. H., Arshad, J., Ur Rehman, A., Shafiq, M., & Choi, J.-G. A novel hybrid MPPT technique to maximize power harvesting from PV system under partial and complex partial shading. *Applied Sciences*, 12(2), 2022. ISSN 2076-3417. doi: 10.3390/app12020587. URL https://www.mdpi.com/2076-3417/12/2/587.

Sasaki, F. & Yamashina, R. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zrT3HcsWSAt.

Sasaki, F., Yohira, T., & Kawaguchi, A. Adversarial behavioral cloning. *Advanced Robotics*, 34(9):592–598, 2020. doi: 10.1080/01691864.2020.1729237. URL https://doi.org/10.1080/01691864.2020.1729237.

Schewe, P. F. *The grid: A journey through the heart of our electrified world*. National Academies Press, 2007.

Schulman, J., Chen, X., & Abbeel, P. Equivalence between policy gradients and soft Q-learning, 2017. URL https://arxiv.org/abs/1704.06440.

Sehgal, A., La, H., Louis, S., & Nguyen, H. Deep reinforcement learning using genetic algorithm for parameter optimization. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 596–601, 2019. doi: 10.1109/IRC.2019.00121.

Sera, D., Teodorescu, R., & Rodriguez, P. PV panel model based on datasheet values. In *2007 IEEE International Symposium on Industrial Electronics*, pages 2392–2396, June 2007. doi: 10.1109/ISIE.2007.4374981.

Sera, D., Mathe, L., Kerekes, T., Spataru, S. V., & Teodorescu, R. On the perturb-and-observe and incremental conductance MPPT methods for PV systems. *IEEE Journal of Photovoltaics*, 3(3):1070–1078, July 2013. ISSN 2156-3381. doi: 10.1109/JPHOTOV.2013.2261118.

Shao, K., Tang, Z., Zhu, Y., Li, N., & Zhao, D. A survey of deep reinforcement learning in video games, 2019. URL https://arxiv.org/abs/1912.10944.

Shehab, M., Zaghloul, A., & El-Badawy, A. Low-level control of a quadrotor using twin delayed deep deterministic policy gradient (TD3). In *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–6, 2021. doi: 10.1109/CCE53527.2021.9633086.

Shi, C., Wan, R., Song, R., Lu, W., & Leng, L. Does the Markov decision process fit the data: Testing for the Markov property in sequential decision making. In III, H. D. & Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8807–8817. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shi20c.html.

Shishlov, I., Morel, R., & Bellassen, V. Compliance of the Parties to the Kyoto Protocol in the first commitment period. *Climate Policy*, 16(6):768–782, 2016. doi: 10.1080/14693062.2016.1164658. URL https://doi.org/10.1080/14693062.2016.1164658.

Shuvo, S. S., Gebremariam, H., & Yilmaz, Y. Deep reinforcement learning based optimal perturbation for MPPT in photovoltaics. In *2021 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2021.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of Go without human knowledge. *nature*, 550(7676):354–359, 2017.

Sinke, W. C. Development of photovoltaic technologies for global impact. *Renewable Energy*, 138:911–914, 2019. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2019.02.030. URL https://www.sciencedirect.com/science/article/pii/S0960148119301740.

Sofia, S. E., Wang, H., Bruno, A., Cruz-Campa, J. L., Buonassisi, T., & Peters, I. M. Roadmap for cost-effective, commercially-viable perovskite silicon tandems for the current and future pv market. *Sustainable Energy & Fuels*, 4(2):852–862, 2020.

Solar Power Europe. Global market outlook 2021-2025, Jul 2021. URL www.solarpowereurope.org/global-market-outlook-2021-2025/. Accessed: 2021-08-31.

Stojanovski, O., Thurber, M., & Wolak, F. Rural energy access through solar home systems: Use patterns and opportunities for improvement. *Energy for Sustainable Development*, 37:33–50, 2017. ISSN 0973-0826. doi: https://doi.org/10.1016/j.esd.2016.11.003. URL https://www.sciencedirect.com/science/article/pii/S0973082616310067.

Sutherland, B. R. Perovskite-silicon tandems edge forward. *Joule*, 4(4):722–723, 2020. ISSN 2542-4351. doi: https://doi.org/10.1016/j.joule.2020.03.022. URL https://www.sciencedirect.com/science/article/pii/S2542435120301379.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. On the importance of initialization and momentum in deep learning. In Dasgupta, S. & McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tan, Z. & Li, K. Differential evolution with mixed mutation strategy based on deep reinforcement learning. *Applied Soft Computing*, 111:107678, 2021. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2021.107678. URL https://www.sciencedirect.com/science/article/pii/S1568494621005998.

Tey, K. S., Mekhilef, S., Yang, H.-T., & Chuang, M.-K. A differential evolution based MPPT method for photovoltaic modules under partial shading conditions. *International Journal of Photoenergy*, 2014, 2014.

Tu, S. & Recht, B. The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint. In Beygelzimer, A. & Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 3036–3083. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/tu19a.html.

U.S. Energy Information Administration. International energy outlook 2018, July 2018. URL https://www.eia.gov/pressroom/presentations/capuano_07242018.pdf. [Online; posted 24-July-2018].

U.S. Energy Information Administration. Annual energy outlook 2019, July 2019. URL https://www.eia.gov/outlooks/aeo/pdf/aeo2019.pdf. [Online; posted 24-July-2019].

van Hasselt, H. *Reinforcement Learning in Continuous State and Action Spaces*, pages 207–251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_7. URL https://doi.org/10.1007/978-3-642-27645-3_7.

van Hasselt, H., Guez, A., & Silver, D. Deep reinforcement learning with double Q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi: 10.1609/aaai.v30i1.10295. URL https://ojs.aaai.org/index.php/AAAI/article/view/10295.

Vartiainen, E., Masson, G., Breyer, C., Moser, D., & Román Medina, E. Impact of weighted average cost of capital, capital expenditure, and other parameters on future utility-scale PV levelised cost of electricity. *Progress in Photovoltaics: Research and Applications*, 28(6):439–453, 2020. doi: https://doi.org/10.1002/pip.3189. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pip.3189.

Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., & Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, 2018.

Veerachary, M., Senjyu, T., & Uezato, K. Maximum power point tracking control of idb converter supplied pv system. *IEE Proceedings - Electric Power Applications*, 148(6):494–502, Nov 2001. ISSN 1350-2352. doi: 10.1049/ip-epa:20010656.

Vega, A., Valiño, V., Conde, E., Ramos, A., & Reina, P. Double sweep tracer for i-v curves characterization and continuous monitoring of photovoltaic facilities. *Solar Energy*, 190:622–629, 2019. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2019.07.053. URL https://www.sciencedirect.com/science/article/pii/S0038092X19307169.

Velilla, E., Collinson, E., Bellato, L., Berg, M. P., & Halfwerk, W. Vibrational noise from wind energy-turbines negatively impacts earthworm abundance. *Oikos*, 130(6):844–849, 2021. doi: https://doi.org/10.1111/oik.08166. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/oik.08166.

Vieira, J. A. B. & Mota, A. M. Implementation of a stand-alone photovoltaic lighting system with MPPT battery charging and LED current control. In *2010 IEEE International Conference on Control Applications*, pages 185–190, 2010. doi: 10.1109/CCA.2010.5611257.

Vieira, R. G., de Araújo, F. M. U., Dhimish, M., & Guerra, M. I. S. A comprehensive review on bypass diode application on photovoltaic modules. *Energies*, 13(10), 2020. ISSN 1996-1073. doi: 10.3390/en13102472. URL https://www.mdpi.com/1996-1073/13/10/2472.

Villalva, M. G., Gazoli, J. R., & Filho, E. R. Comprehensive approach to modeling and simulation of photovoltaic arrays. *IEEE Transactions on Power Electronics*, 24(5):1198–1208, May 2009. ISSN 0885-8993. doi: 10.1109/TPEL.2009.2013862.

Wang, H., Zariphopoulou, T., & Zhou, X. Exploration versus exploitation in reinforcement learning: a stochastic control approach, 2018. URL https://arxiv.org/abs/1812.01552.

Wang, K., Ma, J., Man, K. L., Huang, K., & Huang, X. Sim-to-real deep reinforcement learning for maximum power point tracking of photovoltaic systems. In *2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, pages 1–4, 2021. doi: 10.1109/EEEIC/ICPSEurope51590.2021.9584821.

Watkins, C. J. & Dayan, P. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Weissburg, M. & Draper, A. M. Impacts of global warming and elevated co2 on sensory behavior in predator-prey interactions: A review and synthesis. *Frontiers in Ecology and Evolution*, 7:72, 2019.

Wu, Y.-H., Charoenphakdee, N., Bao, H., Tangkaratt, V., & Sugiyama, M. Imitation learning from imperfect demonstration. In Chaudhuri, K. & Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6818–6827. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/wu19a.html.

Xin, B., Yu, H., Qin, Y., Tang, Q., & Zhu, Z. Exploration entropy for reinforcement learning. *Mathematical Problems in Engineering*, 2020:2672537, Jan 2020a. ISSN 1024-123X. doi: 10.1155/2020/2672537. URL https://doi.org/10.1155/2020/2672537.

Xin, X., Karatzoglou, A., Arapakis, I., & Jose, J. M. *Self-Supervised Reinforcement Learning for Recommender Systems*, pages 931–940. Association for Computing Machinery, New York, NY, USA, 2020b. ISBN 9781450380164. URL https://doi.org/10.1145/3397271.3401147.

Xu, L., Cheng, R., & Yang, J. A new mppt technique for fast and efficient tracking under fast varying solar irradiation and load resistance. *International Journal of Photoenergy*, 2020:6535372, Feb 2020. ISSN 1110-662X. doi: 10.1155/2020/6535372. URL https://doi.org/10.1155/2020/6535372.

Yang, T., Tang, H., Bai, C., Liu, J., Hao, J., Meng, Z., & Liu, P. Exploration in deep reinforcement learning: A comprehensive survey, 2021.

Young, T., Hazarika, D., Poria, S., & Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018. doi: 10.1109/MCI.2018.2840738.

Youssef, A., Telbany, M. E., & Zekry, A. Reinforcement learning for online maximum power point tracking control. *J. Clean Energy Technol*, 4:245–248, 2016.

Yushou, K., Lingling, J., Changyu, W., Liguo, L., & Liming, Z. A new technique of non-linear statistic prediction and its application in atmospheric systems. *Kybernetes*, 37(9/10):1417–1424, Jan 2008. ISSN 0368-492X. doi: 10.1108/03684920810907724. URL https://doi.org/10.1108/03684920810907724.

Zafar, M. W., Mirza, F. M., Zaidi, S. A. H., & Hou, F. The nexus of renewable and nonrenewable energy consumption, trade openness, and CO2 emissions in the framework of EKC: evidence from emerging economies. *Environmental Science and Pollution Research*, 26(15):15162–15173, 2019.

Zaidi, B. Introductory chapter: Introduction to photovoltaic effect. *Solar Panels and Photovoltaic Materials*, pages 1–8, 2018.

Zhang, S., Wang, J., Liu, H., Tong, J., & Sun, Z. Prediction of energy photovoltaic power generation based on artificial intelligence algorithm. *Neural Computing and Applications*, 33(3):821–835, Feb 2021. ISSN 1433-3058. doi: 10.1007/s00521-020-05249-z. URL https://doi.org/10.1007/s00521-020-05249-z.

Zhang, X., Li, S., He, T., Yang, B., Yu, T., Li, H., Jiang, L., & Sun, L. Memetic reinforcement learning based maximum power point tracking design for PV systems under partial shading condition. *Energy*, 174:1079 – 1090, 2019. ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2019.03.053. URL http://www.sciencedirect.com/science/article/pii/S0360544219304578.

Zhang, Z., Li, X., An, J., Man, W., & Zhang, G. Model-free attitude control of spacecraft based on PID-guide TD3 algorithm. *International Journal of Aerospace Engineering*, 2020:8874619, Dec 2020. ISSN 1687-5966. doi: 10.1155/2020/8874619. URL https://doi.org/10.1155/2020/8874619.

Zhou, J., Xue, S., Xue, Y., Liao, Y., Liu, J., & Zhao, W. A novel energy management strategy of hybrid electric vehicle via an improved TD3 deep reinforcement learning. *Energy*, 224:120118, 2021. ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2021.120118. URL https://www.sciencedirect.com/science/article/pii/S0360544221003674.

Zhu, P., Li, X., Poupart, P., & Miao, G. On improving deep reinforcement learning for POMDPs, 2017. URL https://arxiv.org/abs/1704.07978.

Zhu, Y. & Xiao, W. A comprehensive review of topologies for photovoltaic i-v curve tracer. *Solar Energy*, 196:346–357, 2020. ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2019.12.020. URL https://www.sciencedirect.com/science/article/pii/S0038092X19312344.