Control of a Wave Energy Converter using Reinforcement Learning

José Carlos Mota Trigueiro

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. Miguel Afonso Dias de Ayala Botto
Prof. Susana Margarida da Silva Vieira

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Prof. Susana Margarida da Silva Vieira
Member of the Committee: Prof. João Carlos de Campos Henriques

December 2021

Acknowledgments

First and foremost I would like to thank my two thesis supervisors, Professors Miguel Ayala Botto and Susana Vieira, for their support, expertise and feedback for the last half a year. Without their orientation this thesis wouldn't have been possible.

I also must express my gratitude to Jorge Marques Silva for taking the time to explain to me all the fundamentals of Wave Energy Converter modelling, and kindly providing such an extensive amount of *Simulink* and *MATLAB* code to use as a basis for my work, as well as being available to explain any doubts I had throughout the development of the thesis

I would also like to thank Professor João Henriques for his enthusiasm in introducing the challenges behind wave energy conversion to us, as well as freely providing his research papers and simulation models as an introduction to the topic.

For all the support in the journey to become a mechanical engineer that culminated in this thesis, I also have to thank my parents who always had the patience to help me in any way they could and to encourage me to move forward in this challenging and arduous path.

Last but not least, I'd like to thank all of my friends at Técnico, who were responsible for the best moments of my five years at this university, and were always there to support, encourage and make me laugh in the difficult times.

Resumo

O desenvolvimento de estratégias de controlo para maximização da geração de energia em conversores de energia das ondas é fundamental para tornar a exploração das ondas do mar um elemento economicamente viável do cabaz energético. As técnicas de controlo clássicas, baseadas em modelação, apresentam limitações significativas para o cumprimento este requisito, dada a sua dependência da precisão de modelação e incapacidade de adaptação a alterações na dinâmica do sistema ao longo do tempo. Nesta tese é apresentado um esquema de controlo baseado em *Deep Reinforcement Learning* (DRL), utilizando um modelo em *MATLAB* e *Simulink* da coluna de água oscilante de Mutriku como ambiente de treino. O controlador proposto atua tanto no momento eletromagnético exercido pelo sistema de tomada de potência como na abertura da válvula de alívio e utiliza exclusivamente dados medidos na própria instalação como sinais de observação, sem necessitar de um meio de medição externo para estimação do estado do mar. Foram treinadas e testadas três arquiteturas distintas de DRL: *Deep Deterministic Policy Gradient* (DDPG), *Twin Delayed DDPG* (TD3) e *Soft Actor-Critic* (SAC). Usando como base de comparação uma lei de controlo exponencial desenvolvida por outros investigadores, estes agentes são comparados em termos da sua produção anual de energia elétrica. Para além disso, o comportamento do tipo ”caixa negra” do controlador é analisado, de forma a clarificar o tipo de lei de controlo aprendida que é implementada.

Palavras-chave: conversor de energia das ondas, coluna de água oscilante, Mutriku, controlo por torque eletromagnético, *deep reinforcement learning*

Abstract

The development of control strategies that maximize power generation in Wave Energy Converters is fundamental in making the exploitation of sea waves an economically viable element of the energy mix. Classical, model-based control techniques have significant limitations in achieving this goal, due to their dependency on modelling accuracy and inability to adapt to changing system dynamics over time. In this thesis a control scheme based on Deep Reinforcement Learning (DRL) is presented, using a *MATLAB* and *Simulink* model of the Mutriku Oscillating Water Column plant as a training environment. This controller acts on the power take-off electromagnetic torque and relief valve aperture simultaneously, and exclusively uses data measured in the plant itself as observation signal, without requiring an external measuring tool for estimation of the sea state. Three different agent architectures are trained and tested: Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3) and Soft Actor-Critic (SAC). Using as a baseline a power control law developed by previous authors, these agents are compared in terms of their expected yearly electric power production. The black box behaviour of the controller is also analysed, in an effort to gain insight into the type of learned control law it implemented.

Keywords: wave energy converter, oscillating water column, Mutriku, power take-off control, deep reinforcement learning

Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

ANN	Artificial Neural Network.
DDPG	Deep Deterministic Policy Gradient.
DNN	Deep Neural Network.
DQN	Deep Q-Network.
DRL	Deep Reinforcement Learning.
FC	Fully Connected
HSSV	High-Speed Safety Valve
IRENA	International Renewable Energy Agency.
JONSWAP	Joint North Sea Wave Observation Project.
LCOE	Levelised Cost of Energy.
MDP	Markov Decision Process.
MIBEL	Mercado Ibérico de Electricidade
OTEC	Ocean Thermal Energy Conversion.
OWC	Oscillating Water Column.
PTO	Power Take-Off.
ReLU	Rectified Linear Unit
RL	Reinforcement Learning.
SAC	Soft Actor-Critic.
SARSA	State-Action-Reward-State-Action.
TD3	Twin Delayed Deep Deterministic Policy Gradient.
WEC	Wave Energy Converter.

Reinforcement Learning Symbols

α	Entropy term weight.
α_l	Learning rate.
ϵ	Greedy exploration factor.
$\gamma(s)$	Reward discount factor.
λ	L2 regularisation factor.
$\mathcal{D}$	Experience replay buffer.
$\mathcal{H}(\pi)$	Entropy of policy π.
$\mathcal{L}(\theta)$	Loss function.
$\mathcal{N}$	Added exploration noise.
$\mu(s)$	Deterministic policy function.
μ_π, σ_π	Mean and standard deviation of policy π.
ρ	Polyak averaging weight.
σ	Standard deviation of the Gaussian noise.
θ	Set of approximator parameters (for example, neural network weights).
A	Set of possible agent actions.
a	Action chosen by the agent.
c	Noise clipping amplitude.
E	Expected value operator.
G	Expected return.
J	Objective function for expected return.
n	Sample from Gaussian noise.
$Q(s,a)$	Q-function or action-value function.
r	Reward signal.
S	Set of environment states.
s	Environment state.
U	Minibatch of experiences.
$V(s)$	V-function or state-value function.

Wave Energy Conversion Symbols

α_k	Prony method constants.
β_k	Prony method exponents.
Δp	Stagnation pressure head between air chamber and atmosphere.
$\Delta\omega$	Frequency discretisation interval.
Δ_t	Periodogram sampling interval.
$\dot{m}$	Air mass flow rate.
η	Efficiency.
Γ	Imaginary piston heave excitation response.
γ	Specific heat ratio of air.
Λ	Dimensionless generator load.
Ω	Turbine rotation velocity.
ω	Wave frequency.
ω_p	Peak wave frequency.
Φ	Turbine dimensionless flow rate.
ϕ	Excitation response to the wave component.
ϕ_r	Random wave phase.
Π	Turbine dimensionless power coefficient.
Ψ	Turbine dimensionless pressure head.
ρ	Density.
$\varphi_{Mutriku}$	Local Mutriku attenuation function.
A	Wave frequency component amplitude.
a	Power control law constant.
A^{∞}	Added mass at infinite frequency.
A_v	Effective valve area.
A_{γ_s}	Wave spectrum normalising factor.
b	Power control law exponent.
D	Turbine rotor diameter.

E_{kin}	Kinectic energy.
F_{exc}	Wave excitation force.
g	Acceleration of gravity.
H_s	Significant wave height.
I	Turbine-generator rotor inertia.
I_k	State variable in the Prony method model.
K	Impulse response function of the wave radiation.
k_v	Valve aperture state.
m	Mass of the imaginary water piston.
N	Number of data points in periodogram.
P	Power.
p	Absolute air chamber pressure.
p^*	Dimensionless air chamber pressure.
p_o	Probability of sea state occurrence.
p_{at}	Absolute atmospheric pressure.
R	Wave radiation memory term.
S	Surface area of the water column.
S_J	JONSWAP wave spectrum.
$S_{Mutriku}$	Mutriku wave spectrum.
S_{PM}	Pierson-Moskowitz spectrum.
T	Torque.
t	Time.
T_e	Wave energy period.
T_p	Wave peak period.
$t_{spectrum}$	Periodogram computation interval.
u	Safety valve aperture.
V_0	Air chamber volume at hydrostatic conditions.
V_c	Instantaneous air chamber volume.

$w(t)$	Periodogram window function.
z	Water surface height.

Subscripts

at	Atmospheric air.
c	Air chamber.
$elec$	Electric.
gen	Generator.
in	Inlet.
max	Rated quantity.
opt	At the turbine optimal operating point.
$pneu$	Pneumatic.
$turb$	Turbine.
v	Valve.
w	Water.

Superscripts

$'$	Indicates a subsequent state or action.
$-$	Related to a target actor/critic.
$\hat{}$	Estimate of value.
$\star$	Optimal.
$rated$	Rated quantity.
*	Dimensionless.

Chapter 1

Introduction

The need for the mitigation of man-made climate change caused by greenhouse gases in conjunction with the future depletion of the world's fossil fuel reserves means that one of humanity's most difficult challenges for the 21st century is performing a successful transition from a fossil fuel-based energy sector to renewable, carbon-free energy sources, without compromising the current standard of living and economic growth, which have historically been correlated with an increase in energy consumption [1]. In 2019 alone, the world's gross electricity production was 26 730 TWh, an increase of 3.3% from the previous year [2].

An alternative to address this issue is the widespread adoption of renewable energy sources. In this chapter, the case is made for wave energy as an element of the world's energy mix in the future. Some current approaches to the problem of maximising their energy output using multiple control strategies are also explored. Finally, the objectives and a brief outline of the thesis are presented.

1.1 Ocean Energy

Ocean energy is a type of renewable energy that describes all technologies that use the ocean as a clean, renewable energy source to produce electricity. The International Renewable Energy Agency (IRENA) highlights four main solutions that fit into this category: tidal energy, salinity gradient energy, ocean thermal energy conversion (OTEC) and wave energy [3].

Salinity gradient energy extracts energy from the difference in salt concentration between two fluids, usually sea and river water, by using the osmotic pressure to force water to pass through a selectively permeable membrane, while OTEC takes advantage of the temperature gradient between superficial and deeper water in the ocean to power a Rankine based cycle [4]. As of 2020, both of these technologies are still not in commercial deployment, with the only existing installations being small scale models for scientific research, totalling 0.28 MW of installed capacity [3].

Tidal energy is exploited in two different ways, either taking advantage of the potential energy of the sea level variation in low and high tide, using the ebb and flow of the ocean in the same way as a traditional hydroelectric dam, called tidal barrage, or by using the kinetic energy of tidal currents in open sea, called tidal stream energy [5]. The high predictability of the gravitational variations that are responsible for tides [4] means that this technology has been more widely adopted commercially, with 521.5 MW of installed capacity for tidal barrages, divided between two large installations in La Rance, France (240 MW, opened in 1966) and Sihwa Lake, South Korea (254 MW, opened in 2011) totalling 494 MW and multiple smaller installations with a combined installed capacity of 27.5 MW. In contrast, tidal stream energy is still in an early implementation stage, with only 10.60 MW of installed capacity currently in operation. [3, 5].

Wave energy technologies will be further explored in the next section, but in terms of market readiness, it is situated between OTEC and tidal energy, with multiple sites already functioning, totalling 2.31 MW of installed capacity, but no convergence either in the type of use case (larger plants to obtain economies of scale or smaller ones to serve niche applications like islands and other isolated communities) or the type of mechanism used to generate energy, as will be seen in the next section.

Figure 1.1 displays the breakdown between all types of ocean energy technologies, showing the current dominance of tidal energy compared to the others. Even when adding up all technologies, they only make up a small part of the 2351 GW of total renewable energy installed capacity, most of which is hydro-power (1293 GW), wind (564 GW) and solar (486 GW) [6].

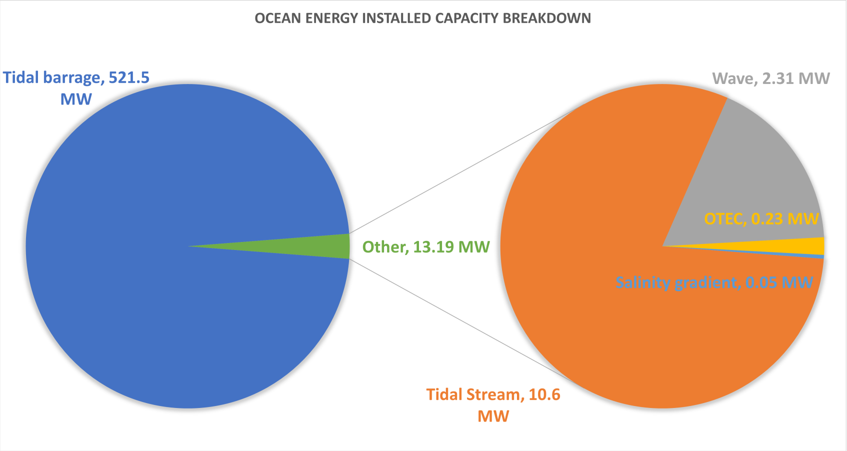


Figure 1.1: Breakdown of ocean energy installed capacity (adapted from IRENA [3]).

1.2 Wave Energy

Extracting energy from the oscillation of the ocean's waves has for long been a subject of interest, from the first recorded patent in the field, registered in France by a father and son named Girard in 1799 [7], to the invention of modern Wave Energy Converters (WEC) in the 1940's by Japanese naval commander Yoshio Matsuda , who powered floating navigation buoys with an air turbine and by doing so created the first of what was later named Oscillating Water Column (OWC) [8]. Interest in wave energy as a competitor to fossil fuels was renewed in response to the 1973 oil crisis, with researchers like Salter [9] or Budar and Falnes [10] publishing their pioneering research on the subject, the latter two being responsible for the definition of point absorber used later in this thesis, as well as the first application of optimal control to wave energy conversion. From that point, multiple methods to harness energy from the waves were developed, and convergence to a dominant technology has still not materialised. Falcão [8] developed a system of classification for Wave Energy Converters through their working principle and type of structure, shown in Figure 1.2.

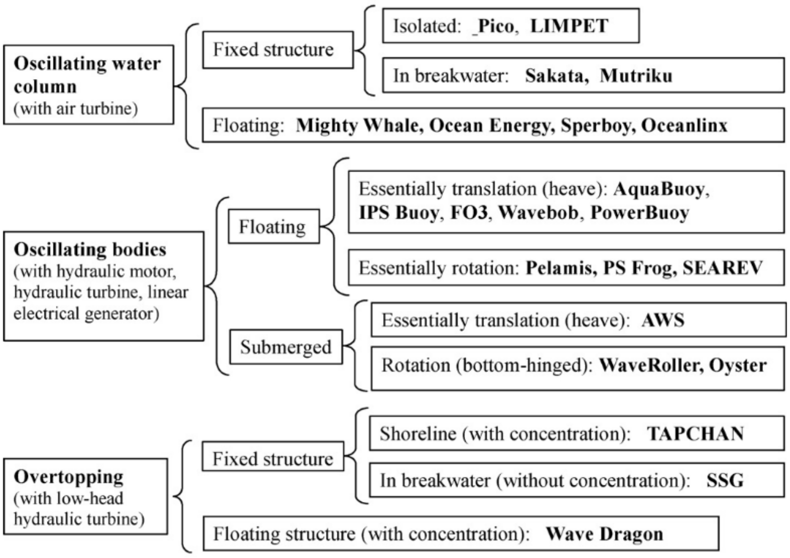

Oscillating water column (with air turbine)
Fixed structure
Isolated: _Pico, LIMPET
In breakwater: Sakata, Mutriku
Floating: Mighty Whale, Ocean Energy, Sperboy, Oceanlinx
Oscillating bodies (with hydraulic motor, hydraulic turbine, linear electrical generator)
Floating
Essentially translation (heave): AquaBuoy, IPS Buoy, FO3, Wavebob, PowerBuoy
Essentially rotation: Pelamis, PS Frog, SEAREV
Submerged
Essentially translation (heave): AWS
Rotation (bottom-hinged): WaveRoller, Oyster
Overtopping (with low-head hydraulic turbine)
Fixed structure
Shoreline (with concentration): TAPCHAN
In breakwater (without concentration): SSG
Floating structure (with concentration): Wave Dragon

Figure 1.2: Classification system for Wave Energy Converters [8].

Another system of classifying WEC technologies is by the type of Power Take-Off (PTO) system they use to transmit power to the generator [11], as shown in Figure 1.3. In this case, wave energy may be converted to mechanical energy, which either drives the generator's rotation directly or is transmitted through an hydraulic system, to pneumatic energy, through the compression of air in an air chamber, which then drives an air turbine (Wells or biradial), or directly as potential energy by storing the water moved by the waves in a reservoir which then powers a water turbine when full. This classification is closely correlated to the one in figure 1.2, since OWCs extract energy using air turbine PTOs, oscillating bodies use hydraulic PTOs, and over-topping devices use the ocean water accumulated in a reservoir to powera water turbine PTO.

1.2.1 Oscillating Water Column

OWC devices may be either installed in a fixed structure, or in a floating platform. Fixed structure OWCs may be installed in a purpose-built structure, but since construction of the structure that houses the device is an expensive and complex step in the deployment of OWC, it's often advantageous to integrate these devices in a breakwater or other preexisting structure in order to split infrastructure costs and share access for building and maintenance [12]. Existing examples of purpose-built fixed structure OWC devices are the Pico OWC in the Azores or the LIMPET plant in the island of Islay, Scotland, both of which have already been decommissioned, while the Mutriku OWC, the subject of this work, is a currently functioning example of an OWC that was integrated in a breakwater structure. A recent development in this type of OWC is the integration in monopile offshore wind turbines, forming a hybrid

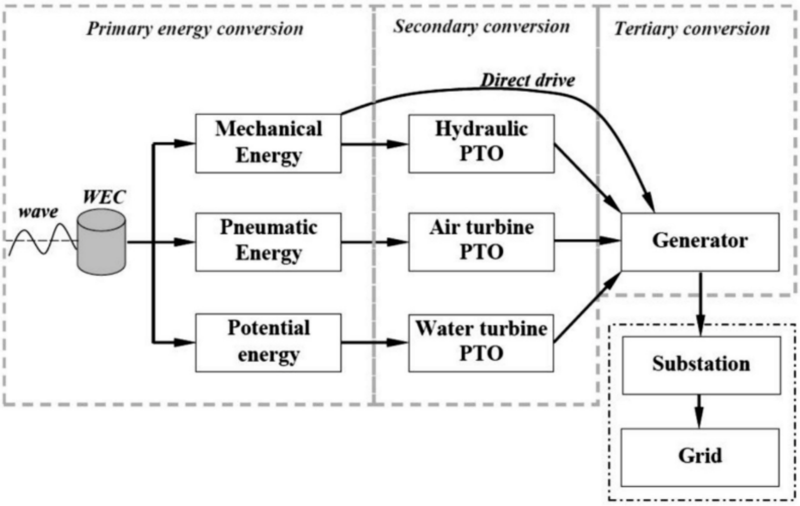


Figure 1.3: Classification system for Wave Energy Converters based on PTO [11].

wind-wave system which may share not only the structural elements but also the connection to the electric grid [13].

Floating platform OWCs are usually deployed at higher sea depths, where more energetic sea states may be found [12]. An application example of this type of technology is the aforementioned navigation buoy developed by Matsuda, but the integration of this technology in large floating platforms like oil storage facilities or floating docks is also a possibility [14].

1.2.2 Oscillating bodies

Oscillating body WECs are devices that are deployed off-shore and, consequently, take advantage of more powerful waves, but present additional drawbacks related to complexity, mooring, maintenance and connection to the grid. A large variety of these devices exists, including single, multiple body or submerged heaving systems, where the mechanical energy extracted is mostly derived from the translation motion of floating bodies, and floating or bottom hinged pitching systems, where rotational mechanical energy is extracted from the rotation of a moving body [8].

1.2.3 Overtopping devices

Overtopping devices are systems where wave energy is not converted directly to electricity, but instead is stored as potential energy in a reservoir with a higher level that the surrounding average sea height. The stored water is then used to drive a low-head water turbine. These systems usually integrate tapered channels, ramps and reflecting walls to concentrate the waves at a single entry point [8].

1.3 Motivation

As mentioned, wave energy is still in an early stage in its implementation for commercial use. However, its potential as an element of the world's energy mix has been mentioned by multiple authors before, with an estimated theoretical installed capacity of 2.11 TW [15] or a total yearly energy production of 29500 TWh [3], which would be enough to cover the worlds current electricity needs. The critical factor identified by the European Commission that will determine the fulfilment of this potential is the reduction of the Levelised Cost of Energy (LCOE) of WECs, mostly through the improvement of device reliability and survivability in agitated sea states. For wave energy this means a LCOE target of 0.20€/kWh in 2025, 0.15€/kWh in 2030 and 0.10€/kWh in 2035 [16] (current LCOE is estimated to be 0.30 to 0.50€/kWh [3]). One of the critical factors in achieving this reduction was identified by the European Technology & Innovation Platform for Ocean Energy to be the improvement of PTO control systems, increasing their adaptability to ocean conditions [17]. This thesis aims to use Reinforcement Learning (RL) to tackle this problem, providing a model-free algorithm that adapts to changing conditions, in order to maximise energy production while mitigating the damage that may result from extreme events.

1.4 Objectives and Deliverables

The main objective of this thesis is the development of control strategies for an OWC plant, using an adaptation of a preexisting simulation model of the Mutriku wave power plant in *Simulink* in as a training environment. This control scheme should be model-free, meaning that it won't use an explicit model of the plant directly, but instead relies only on data, which in this case will be measured from the simulation model. An additional restriction was imposed for the source of this data, where only data available locally at the plant may be considered, avoiding the reliance on other systems such as wave measuring buoys or satellites. This control scheme should be adaptable to different sea states and respect the safety constraints of the plant while maximising power production, and should act on the two control variables available at the plant: the generator torque and the aperture of a relief valve. Performance of this control scheme will be benchmarked against a state of the art method currently in use at the plant, developed based on the turbine performance curve.

To achieve these objectives, reinforcement learning controllers using three different architectures will be presented and compared against each other: Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3) and Soft Actor Critic (SAC). To the author's best knowledge, this work is the first application of these continuous Deep Reinforcement Learning (DRL) algorithms to the control of OWC devices, and the second for WEC devices in general following the work of Anderlini et al. [18] in an oscillating body WEC. It is also the first reinforcement learning WEC control scheme to include the aperture of a relief valve as an additional control action to the PTO force.

1.5 Thesis Outline

Chapter 2 of this thesis presents a state of the art report in the field of Wave Energy Converter and, more specifically, Oscillating Water Column control, ranging from the classical control techniques both in frequency (latching, impedance matching) and time (rotational speed control, air flow control) to control schemes based on computational intelligence and machine learning, with a focus on the existing Reinforcement Learning solutions.

Chapter 3 provides the background concepts behind RL and introduces the main state-of-the-art DRL algorithms, providing the reasoning for the choice of DDPG, TD3 and SAC as possible control schemes for the Mutriku power plant.

Chapter 4 contains a description of the Mutriku WEC and the respective model, listing all the equations and relevant parameters that constitute the *Simulink* model that will be used as a training environment for the controller. This includes a model for the wave force, the water column hydrodynamics, the air chamber compression process and turbine, valve and generator dynamics, constituting a complete wave-to-wire model.

Chapter 5 formulates the OWC control problem in an RL framework and details the structure of the Neural Networks (NN) that will encode the control scheme, as well as all relevant controller hyper-parameters. The parameters for the online sea state estimation using spectral analysis on the air chamber pressure signal are also defined.

Chapter 6 presents the results of the application of each of the tested RL controllers and a discussion on their performance.

In Chapter 7, conclusions are drawn from the results of the work developed, and a set of proposals to further improve the control of the Mutriku OWC is presented, along with suggestions on how to implement the suggested controller on a physical prototype of the system.

Chapter 2

State of the Art in OWC Control

The control problem in OWC devices may be formulated with multiple objectives in mind, such as maximizing electrical power, keeping the air turbine close to its optimal operating point or minimising undesirable events such as turbine stalling.

In this chapter an overview of the main control strategies for WEC and, in particular, OWC devices, identified in the scientific literature, ranging from research problems tested in simulation models to the strategies used in the currently existing prototypes and commercial installations of OWC. In the first section, classical control schemes based on physical modelling and control theory principles are presented, while the latter focuses on the use of computational intelligence and data-based approaches for control and prediction in WEC's.

2.1 Classical Control

As identified in multiple review papers [19–24], modelling and control in WEC devices may be classified into two main categories: frequency domain and time domain control.

2.1.1 Frequency Domain Control

The first applications of control to WEC devices were performed in the frequency domain, using the optimal conditions for oscillation amplitude and phase. Falnes [20] states that the optimal magnitude is the one where the absorbed power equals the power reradiated into the sea by the oscillating system, while the optimal phase condition requires that the oscillation velocity is in phase with the excitation force.

In OWC devices achieving both conditions means that excitation volume flux must be in phase with air chamber pressure. With regular, sinusoidal waves this condition is satisfied when the oscillation velocity is in resonance with the wave excitation force. However real, irregular waves cause the dynamics of the system to be non-causal, requiring a prediction model for the excitation force to achieve optimality. Another issue is that impedance matching requires extraction of reactive power from the grid.

To adapt the optimal conditions to irregular waves, a technique called complex conjugate control may be used, where the impedance of the PTO system must match the mechanical impedance of the OWC for the frequencies found in a spectral model of the waves. This optimization has been achieved in simulation by adjusting the rotation velocity of an air turbine using a stochastic approach [25] or by changing the pitch angle on a Wells turbine's blades through the use of a PID feedback controller [26] .

2.1.2 Time Domain Control

A way to bridge the gap between frequency domain control objectives and time domain control is through latching, a simpler strategy to approximate the optimal phase condition is through latching and unlatching. This approach locks the WEC in position when it reaches a velocity of zero in its oscillation. In OWC's, the latching mechanism is a valve place in series between the air chamber and the turbine, which stops the air from escaping the chamber. This strategy has been successfully tested both in simulation [27, 28] and in a laboratory model [29]. While this type of control scheme does not require reactive power, the actuation time and structural resistance requirements for the valve, coupled with the compressibility of the air means the pressure variation will never reach zero, making the control suboptimal [28]. The time to unlatch will also be a relevant control variable that requires a predictive model of the system to optimize [23, 29].

The latching problem may also be formulated in a Model Predictive Control (MPC) framework. The MPC acts in time domain to compute the optimal latching times that maximize future power production through the approximation of the optimality conditions [30].

Even considering these alternatives, the optimality conditions for frequency domain control are only appropriate when the PTO system is linear. When including non-linearities in the model such as stalling in Wells turbines or the thermodynamics of air compression, a time domain approach may be preferred [19]. Specifically for OWCs, two main modes of control are identified: turbine rotational speed control, through regulation of the generator's electromagnetic torque, or airflow control, through the use of valves. Two types of valve may be used for airflow control: a relief valve, mounted in parallel with the turbine duct, reducing the pressure head in the turbine or a high speed safety valve mounted in series before the turbine inlet, cutting off the flow of air to the turbine [12, 24].

Rotational speed control focuses on determining a control law for the generator electromagnetic torque to operate the turbine-generator set at its optimal rotation velocity. Justino and Falcão [31] experimented in simulation with piecewise constant torque control aiming to keep the rotation velocity under a certain threshold of deviation from a reference value, calculated using the model, but this strategy failed after finding that it introduces unacceptable oscillations in either the rotation velocity or generator torque.

The common quadratic relationship between generator torque and rotation velocity Ω, or cubic for the generator power ($P_{gen} = a\Omega^3$) was developed by the same researchers as an alternative. Later, multiple authors extended this approach to a more general exponential law of the type $P_{gen} = a\Omega^b$, [32–34] where a and b are manually tuned scalar parameters. This approach allows the controller to take into account friction losses, non-negligible rotor inertia and coupling between turbine aerodynamics and water column hydrodynamics. These authors focused on OWC simulation models using both biradial and Wells turbines and included limitations on maximum generator torque and power.

A different approach to modelling for rotational speed control focuses on the generator dynamics, having the turbine torque and rotation velocity as inputs to the model. Examples include a purely resistive (proportional to rotational speed) generator torque achieved by adding resistances in series with the generator rotor windings [35], or by independent PID control of active and reactive generator power through variation of the rotor currents as a function of turbine pressure drop [36]. Both these strategies rely on accurate modelling of the whole plant to generate either a lookup table for the resistance value or a reference signal for the PID controller.

Airflow control through the use of valves both in series and in parallel was first introduced by Falcão and Justino [37].

Relief valves have been shown in a simulation of the Pico OWC to significantly increase energy production when combined with the previously defined cubic power law [38], even when only allowing discrete variations of valve effective area.

A simple control law that adjusts valve aperture based on a forecast of future wave elevation using an autorregressive (AR) model was then implemented in the Pico plant, showing a 15% increase in production when using a sub-optimal approximation of the power law [39, 40]. Sequentially opening multiple release valves has also been used in a model of the Rocella Jonica OWC to limit the rotation velocity of the turbine and keep it operating in highly energetic sea states[41].

Safety valves in series with the air turbine have mostly been used to cut off air flow from the turbine when it reaches a threshold rotation velocity [33, 34] in a binary on-off control scheme, but Amundarain et al. [36] used the difference between the cubic power law and the actual generator power output as an error signal to continuously control the valve aperture.

2.2 Applications of Computational Intelligence to Wave Energy

Computational intelligence and machine learning is a growing interest for energy systems research, both due to increased data and computing power availability and higher complexity of energy systems with the inclusion of intermittent renewable power sources [42, 43], for applications as diverse as demand and generation forecasting, dispatching and control, among others [44].

One relevant field of application of machine learning to wave energy is in forecasting either the sea state or the wave elevation time series, since, as previously mentioned, the theoretical optimal control requires advanced information about the excitation force.

Shallow neural network models have been previously used to predict pressure oscillations in an OWC chamber up to 3 seconds ahead [45, 46], although less computationally expensive AR models may be able to achieve similar results [47]. A possible improvement would be adding extra features to the network, such as data measurements gathered at high seas[39]. In the field of WEC control, neural networks have also been used to generate a rotation velocity reference for generator control in an OWC [48], to approximate the optimal phase and amplitude using Internal Model Control [49] and to design a controller that performs real time tuning of a lower level reactive controller. [50].

The main topic of this thesis, Reinforcement Learning (RL), has also been approached before as a strategy to control WEC devices. After an exhaustive revision of existing literature on this topic, most works published have focused on the analysis of oscillating body WEC devices [18, 51–56], although some research papers have been written on the RL control of the Mutriku OWC [57, 58]. From the previously mentioned works, none have been tested on operations in real waves, and only the work by [58] was tested in a dry lab with a motor simulating the turbine torque.

In oscillating body WEC's, using only significant height and energy period as state inputs, RL control has been shown to converge to the theoretically predicted optimal proportionality constant B_{PTO} in a resistive PTO force control law $F_{PTO} = B_{PTO}\dot{z}$, where $\dot{z}$ is the displacement velocity of the oscillating body. This goal was achieved using either a tabular Q-Learning method [51] or a policy iteration algorithm [52, 53].

Q-Learning has also successfully used to approximate the theoretical optimal resistive force for more complex systems such as an oscillating arm energy converter [55] and a two oscillating body WEC with a multidimensional action space, adding a proportionality constant C_{PTO} for a reactive control law of type $F_{PTO} = B_{PTO}\dot{z} + C_{PTO}z$ as an additional control action [54, 56].

For OWC devices, Q-learning was used to select the optimal parameters a and b for an exponential generator control law $P_{gen} = a\Omega^b$ for different sea states [57, 58] where it was shown to outperform other control techniques such as the cubic power law and latching control in power generation, only being outperformed by MPC.

A notable recent development is the application of the continuous Soft Actor-Critic algorithm to an oscillating body wave energy converter [18], avoiding parameterised control laws, such as the power law, by using the PTO force directly as the algorithm output.

Chapter 3

Reinforcement Learning

Reinforcement learning is, along with supervised and unsupervised learning, one of the three main types of machine learning. While supervised learning's main goal is to generalise and extrapolate new outputs from a previously labeled dataset and unsupervised learning aims to find hidden structure in unlabeled data, reinforcement learning deals with the training process of an intelligent agent that, given a certain observation of an external environment, executes the action that maximises the expected value of a numerical reward function.

In this chapter, the fundamental concepts behind Reinforcement Learning (RL) are introduced, as well as the main state-of-the-art algorithms currently in use, along with the justification for which to choose in the control of the Mutriku OWC.

3.1 Introduction to Reinforcement Learning

Reinforcement Learning as a theoretical concept originated from two main fields of research: the dynamic programming solution to the optimal control problem developed by Bellman [59] and the theory of animal psychology where animals tend to repeat actions that result in a positive consequence and avoid actions that lead to negative outcomes [60].

The well-known book by Sutton and Barto [60] provides a comprehensive description to the mathematical foundation and implementation of solutions for this type of problem, and is used as a source for the introduction below.

As previously mentioned, RL methods are based on an interaction between an agent, who acts as both decision maker and learner, and an environment, which is characterised at a particular time by a set of observations available to the agent. When the agent performs an action on the environment, it changes its state, either deterministically or stochastically, which may change the possible future actions and respective outcomes, and receives a numerical reward.

The objective of the learning process is to maximise not only the individual reward of any given action, but also the expected future rewards of the subsequent action-state combinations. To achieve this goal, the agent must balance the exploitation of actions that have previously achieved a high reward, and the exploration of new, unknown actions. This issue is called in RL literature the exploitation-exploration trade-off, and presents one of the main challenges in RL agent training, since excessive exploitation may lead to convergence to a sub-optimal policy while excessive exploration doesn't allow the agent to leverage previous experience to repeat actions that yielded high expected reward.

The problem structure behind RL may be formulated in the framework of Markov Decision Processes (MDP's), as shown in figure 3.1. At every time step t, the agent is provided with a numerical representation of the environment's state S_t and uses it to select an action A_t to perform on the system. After a time step, at instant $t+1$, the agent receives a reward signal $R_{t+1} = r$ and observes the system's new state S_{t+1}. The state observation must have the Markov property, meaning that it should have as many dimensions and respective values as necessary to fully define all past interactions that will have an effect on future states and the agent actions may also be multidimensional.

As will be further discussed in this chapter, both action and state sets may be either discrete or continuous, depending on the chosen algorithm. The reward, however, should be a scalar real value, either positive or negative, representing a bonus for achieving a favourable outcome or a penalty for an unfavourable one, respectively.

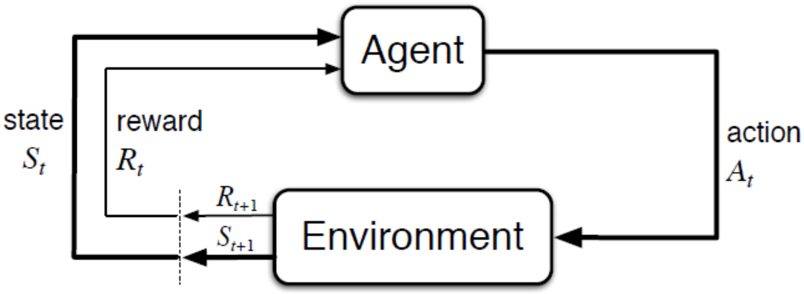


Figure 3.1: Diagram representing the agent-environment interaction in an MDP [60].

In the MDP framework, the RL learning process consists in finding a mapping from every possible state $S_t = s$ to a probability of selecting every possible action $A_t = a$, which will be named policy that will maximise a value function $V_\pi(s)$ that measures the expected cumulative reward, which is called return, represented by G_t. The policy mapping may either be deterministic, which is usually denoted by $\mu(s)$ in the literature, or stochastic, denoted by $\pi(a|s)$. Frequently, the return value is discounted by a factor $\gamma \in [0,1]$, which serves the dual purpose of making the series defining the return, equation 3.1, converge when the MDP is infinite (the final step T tends to infinity) and as a training hyperparameter that will measure the shortsightedness of the agent, where a lower discount factor will make the agent prioritise immediate reward over long term benefit and vice-versa.

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k \tag{3.1}$$

The definition of return allows for the introduction of a formal definition of a value function associated with a given policy π, evaluated at any given state $S_t = s$ as shown in equation 3.2. This form of value function is denominated the state-value function or V-function for policy π. An alternative formulation for the value function is by the definition of the expected value E_π of the return from taking action a in state s under policy π (equation 3.3), called action-value function or Q-function. The letter Q comes from the word quality, as in the quality of action a under state s.

$$V_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|S_t = s\right] \tag{3.2}$$

$$Q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a] = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|S_t = s, A_t = a\right]\text{, for all } s \in S \tag{3.3}$$

Having defined the notion of value in RL, the concept of optimality arises from equations 3.2 and 3.3. Any optimal policy $\pi^\star$ is a policy that maximises the state-value function $V(s)$ and consequently the action-value function $Q(s,a)$ for every state $s \in S$. Multiple optimal policies may exist in the same problem, but all optimal policies share the same optimal state-value function $V^\star(s) = \max_\pi V_\pi(s)$ and action-value function $Q^\star(s,a) = \max_\pi Q_\pi(s,a)$. From the Q-function it is also possible to generate the optimal action $a^\star = \arg\max_a Q^\star(s,a)$ directly, although multiple actions may be optimal for the same case, in which case they could be chosen randomly.

Another key concept in defining the learning process for multiple RL algorithms are Bellman's equations 3.4 which establish a recursive definition for the value functions in which the value of a state is defined as the reward from achieving that state summed to the discounted value of the new state $S_{t+1} = s'$ the environment transitions to under an action chosen from policy π, where the next action $A_{t+1} = a'$ is also sampled from policy π. Value function optimality is also directly defined using Bellman's equations, show in equations 3.5.

$$V_\pi(s) = E\left[r(s,a) + \gamma V_\pi(s')\right] \tag{3.4a}$$

$$Q_\pi(s,a) = E_\pi\left[r(s,a) + \gamma E_\pi[Q_\pi(s',a')]\right] \tag{3.4b}$$

$$V^\star(s) = \max_a E\left[r(s,a) + \gamma V^\star(s')\right] \tag{3.5a}$$

$$Q^\star(s,a) = E\left[r(s,a) + \gamma \max_{a'} Q^\star(s',a')\right] \tag{3.5b}$$

3.2 Taxonomy of Reinforcement Learning Algorithms

There are multiple possible algorithms to approach the problem of determining the optimal policy for a given environment.

Initial research in this field focused on tabular methods, which may be applied in discrete and low dimensional state and action spaces, where every possible action and state may be enumerated into a finite table named Q-table, since it contains information about the Q-value function. Examples of this type of algorithm are SARSA [61] and Q-Learning [62]. Learning in these algorithms occurs through the update of the Q-table using variations of Bellman's equation 3.4b: equations 3.6 and 3.7, respectively.

In both methods, instead of substituting the previous value of $Q(s,a)$ by its estimate from Bellman's equation $r + \gamma Q(s',a')$, which would lead to unstable behaviour of the estimation, a learning rate $\alpha_l \in [0,1]$ is introduced, which weighs previous and new information about $Q(s,a)$.

New observation, action and reward data are usually obtained through the use of an ϵ-greedy policy, which takes the action that leads to the maximum value estimate with probability $1-\epsilon$, promoting exploitation of known actions (greedy behaviour) and a random action with probability ϵ, promoting exploration of previously unknown behaviour. The difference between both these algorithms is that SARSA updates the Q-table through the transitions generated by the followed policy, while Q-Learning uses transitions that are independent of the policy used. Algorithms similar to Q-Learning that may use different policies for value estimation and agent behaviour are called off-policy algorithms while algorithms that share the same policy in both situations are called on-policy algorithms.

$$Q(s,a) \leftarrow Q(s,a) + \alpha_l \left[r + \gamma Q(s',a') - Q(s,a)\right] \tag{3.6}$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha_l \left[r + \gamma \max_a Q(s',a) - Q(s,a)\right] \tag{3.7}$$

For more complex problems, using either continuous or high-dimensional action and state spaces, tabular methods have infeasible memory requirements [60], leading to the need to approximate the value function using a function defined by a set of parameters θ. Multiple approaches have been proposed for this approximation, including linear approximation and Fourier or radial basis functions [60, 63, 64], but the most common approximation method is through the use of neural networks, more specifically Deep Neural Networks (DNN).

Neural networks with non-polynomial activation functions have been demonstrated to be universal non-linear approximators [65] and since the approximation power of a DNN has been shown to grow exponentially with the number of hidden layers [66], they address the ”curse of dimensionality” in RL and dynamic programming as described by Bellman [59], where the computational cost grows exponentially with the number of states describing the environment.

The neural networks used in RL are commonly multilayer perceptrons, which consist in a set of stacked layers of neurons (see figure 3.2).

Each neuron receives an input signal vector $\mathbf{x}$ (either the network input on the first layer or the output from the previous layer in hidden layers) and performs an operation of type $y = f(\mathbf{w}^T\mathbf{x} + b)$, where y is the neuron output, f is the non-linear activation function, $\mathbf{w}$ is a weight vector and b is a bias term.

The most commonly used activation function in DNN is the ReLU (Rectified Linear Unit) function $f(\mathbf{z}) = \max(0,\mathbf{z})$, due to the simplicity in computing its gradient and its property of being scale unbounded, avoiding saturation for large input values). For situations where the output must have other properties such as boundedness or smoothness, other non-linear functions may be used such as the softplus function $f(\mathbf{z}) = \log(1+e^{\mathbf{z}})$, a smooth version of ReLU, the hyperbolic tangent $f(\mathbf{z}) = \tanh(\mathbf{z})$ or the sigmoid $f(\mathbf{z}) = \frac{1}{1+e^{-\mathbf{z}}}$, which squash their outputs to the]-1,1[or]0,1[intervals, respectively. The four activation functions described above are shown in figure 3.3.

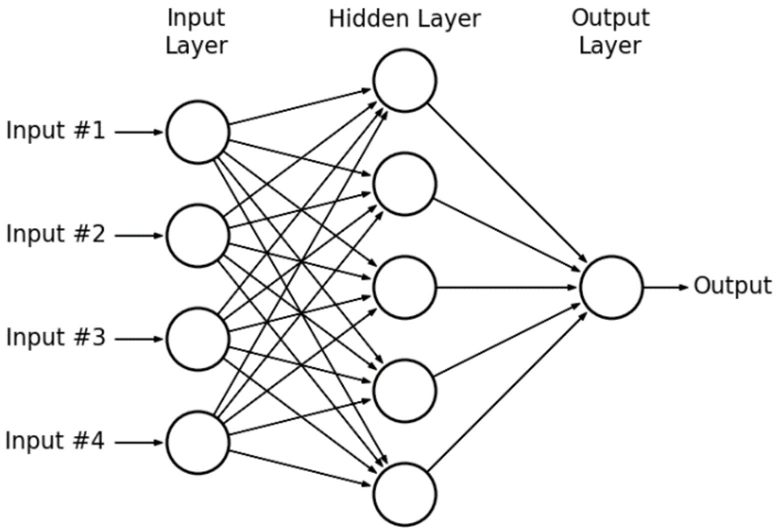


Figure 3.2: Multilayer perceptron network structure [67].

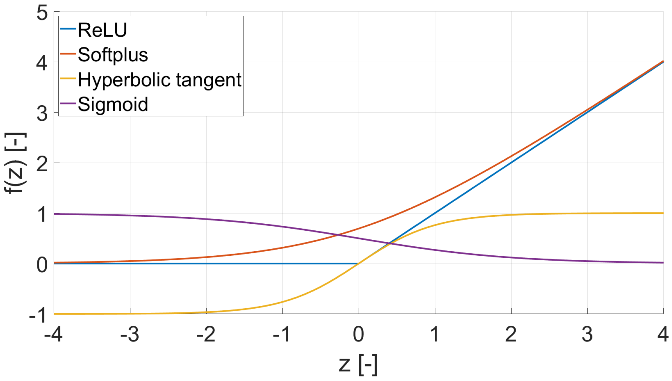


Figure 3.3: ReLU, hyperbolic tangent, softplus and sigmoid activation functions.

Training is done through an algorithm called backpropagation, or a variation of it, by adjusting the network parameters of every neuron $\boldsymbol{\theta}$=[$\mathbf{w}$ b]. For more details on network training, structures and applications, it is recommended to read the book by Goodfellow et al. [68].

Using differentiable functions as a basis for the approximation, a different approach to finding the optimal policy is using gradient methods, by computing the gradient of the expected return with respect to the value function parameters and performing gradient ascent [69], a category of methods called policy optimisation or policy gradient methods.

The development of a set of algorithms that use DNN-based architectures to approximate either the value function, the policy or both led to a new field of RL research, Deep Reinforcement Learning (DRL), combining the objective-driven learning in RL with the approximation and generalisation power in Deep Learning [70].

A commonly used classification system for reinforcement learning algorithms is the one developed by OpenAI [71], shown in figure 3.4, which provides some insight about the main principles and computational strategies used to extend deep learning principles to reinforcement learning.

The first distinction presented in figure 3.4 is between model-based and model-free RL. Model-based

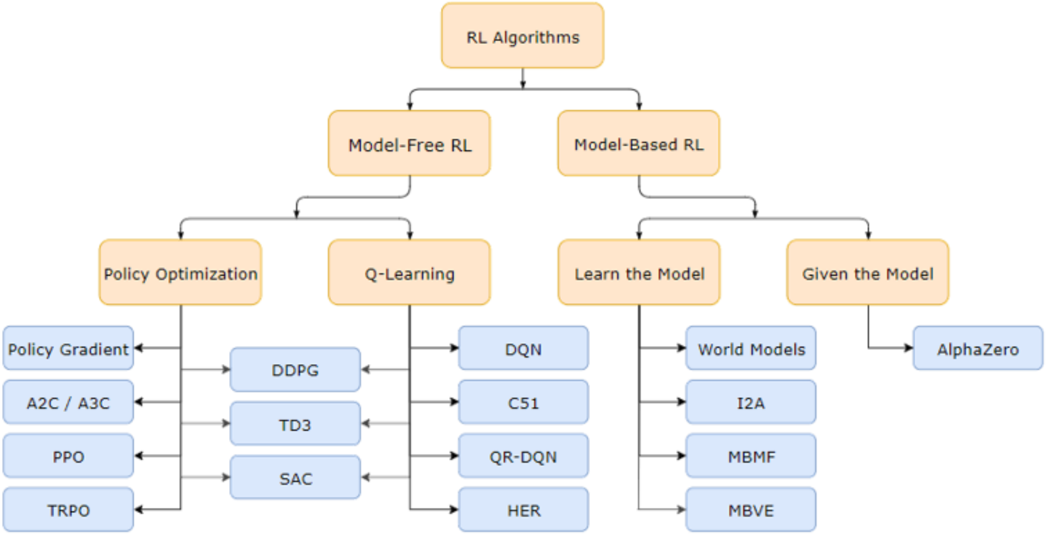


Figure 3.4: Taxonomy of reinforcement learning algorithms [71].

RL requires a complete model for the transition dynamics, both for state and reward signals of the environment. Taking advantage of the agent's knowledge of the model to anticipate future rewards, sample efficiency is increased, but an accurate and explicit model for the environment may not be available and modelling inaccuracies may introduce bias in the agent's behaviour when transitioning to the real environment [72]. In this type of algorithm, the agent must either learn a representation of the model or work with an expert-provided model.

The most significant achievement of model-based RL with a provided model was the development of AlphaZero [73], which after four hours of training was able to outperform both top human players and classical machine learning methods (tree-search algorithms) in board games such as chess, go, and shogi.

In the field of algorithms that learn a model of the environment, algorithms that use DNN's ability to learn a lower dimensional representation of input data to learn from raw pixel data in simulated environments [74, 75] have been successful.

However, in non-virtual and, particularly, engineering applications, model predictive control (MPC) has been preferred to model-based DRL when a model is available [42], due to its stronger control theoretical foundations in guaranteeing stability and avoiding the necessity of training iterations until operation, avoiding time constraints at the expense of more computationally intensive online optimisation algorithms [64].

When a model of the system is not available, model-free RL provides a solution by learning only from exploration and interaction with the environment. The two main principles behind classical RL are also used in model-free DRL: policy based methods (”Policy Optimization” methods in figure 3.4) and value-based methods (”Q-Learning” in figure 3.4). Current state-of-the-art methods also leverage the possibility of combining the two approaches, leading to the development of actor-critic methods, where the actor is an approximator of the policy and the critic an approximator of the value function [70, 72].

Approximating the Q-value function using a parameterisation Q_θ (where θ are the approximator parameters, such as the weights in a DNN) suffers from what Sutton and Barto [60] called the "deadly triad" of RL algorithms: bootstrapping, function approximation and off-policy learning.

Bootstrapping occurs when the value of a state-action pair is estimated using the value of subsequent states as shown in equation 3.7, which is also an estimate. Function approximation means that the DNN used to approximate the value function must generalise from the training samples to previously unseen state-action pairs, which may inappropriately change other states' values, including the ones used for bootstrapping. The final component of the triad, off-policy learning, removes the guarantee that values are updated as soon as they are utilised by the policy, magnifying the effect of the previous two problems [76].

Nevertheless, multiple algorithms using a DNN to approximate Q-value functions have successfully been implemented that include all three elements of the deadly triad [77–80].

In the case of the Mutriku WEC, the transition dynamics of the environment are not fully available due to the unpredictable nature of the wave excitation, so model free DRL is the more natural choice for the controller. Most of the relevant variables in the model are also physical quantities that vary continuously, so to avoid discretisation, which has been a common approach in classical RL applications to WEC devices (see chapter 2), a DRL algorithm that represents continuous states and actions directly should be favoured.

This led to the choice of the three state-of-the-art DRL actor-critic type algorithms shown in figure 3.4: Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3) and Soft Actor-Critic (SAC). In the remainder of this chapter, these algorithms will be described in detail, along with the Deep Q-Network (DQN) algorithm, which represents the first and simplest DRL algorithm, so it is used to introduce some fundamental concepts of the field that will be built upon by the other frameworks.

3.2.1 Deep Q-Network (DQN)

The development of Deep Q-Networks (DQN) in 2015 greatly increased the scientific community's interest in RL algorithms, with the number of yearly published papers in the subject doubling from 2015 to 2019, with a corresponding rise in the number of papers applying RL to energy systems [42]. As such, understanding the fundamental principles behind this method will be necessary to approach all other DRL algorithms that extend this method to more complex problems.

The original DQN implementation uses off-policy exploration with an ϵ-greedy policy, similarly to tabular Q-Learning, but instead of calculating the action-value function $Q(s, a)$ directly from Bellman's equation, the aim is to obtain a parametric representation of the optimal function $Q(s, a; \theta) \approx Q^\star(s, a)$, where θ is the set of weights of the neural network approximator.

To improve training stability, the experiences (s, a, r, s') of the agent at time step t are stored in a replay buffer $\mathcal{D}$, avoiding the sampling of highly correlated, consecutive interactions and allowing for reuse of data, improving sampling efficiency. The algorithm aims to minimise loss function $\mathcal{L}(\theta)$, given by equation 3.8, where $U(\mathcal{D})$ is a minibatch of experiences sampled from the replay buffer. This optimisation process consists in the minimisation of the mean squared error in Bellman's equation, but instead of using the return of the optimal action-value function $y^\star = r + \gamma \max_{a'} Q^\star(s', a')$, a neural network parameterisation $y = r + \gamma \max_{a'} Q(s', a'; \theta_i^-)$ of the target with weights θ_i^- is used. This target network has the same architecture as the original critic network and weights θ_i^- are copied from the critic network at every C time-steps [77] or Polyak averaged [1] between previous target weights and current critic weights $\theta_{i+1}^- = \rho\theta_i^- + (1-\rho)\theta_i$, with weight ρ usually taking a value close to 1 [71].

Minimisation of $\mathcal{L}$ is performed at every training iteration through either gradient descent $\theta_{i+1} = \theta_i - \alpha_l \nabla_{\theta_i} \mathcal{L}(\theta_i)$, where α_l is the learning rate and $\nabla_{\theta_i} \mathcal{L}(\theta_i)$ is the gradient of the loss function with respect to the weights θ_i (given by equation 3.9), or, more commonly, through a variation of stochastic gradient descent such as the Adam optimiser [81].

$$\mathcal{L}_i(\theta_i) = E_{(s,a,r,s') \sim U(\mathcal{D})} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right] \tag{3.8}$$

$$\nabla_{\theta_i} \mathcal{L}(\theta_i) = E_{(s,a,r,s') \sim U(\mathcal{D})} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right) \nabla_{\theta_i} Q(s, a; \theta_i) \right] \tag{3.9}$$

Multiple structures (number of layers, number of neurons per layer and activation functions) are possible for the neural network, as long as the input layer has as many neurons as the dimension of the observation vector and the output layer has as many neurons as the number of possible actions, where the output represents the Q-value of each action. After convergence to the optimal Q-function, the optimal action is chosen by taking the maximum over the output values of the neural network. For continuous problems or problems with a large number of discrete actions the computation of the maximum value may be a computationally expensive process. The algorithms introduced in subsections 3.2.2 to 3.2.4 will present strategies to deal with this issue.

3.2.2 Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) [78] is an actor-critic RL algorithm that serves as an extension of DQN to problems with continuous action spaces. The optimisation problem of choosing the action that leads to the maximum Q-value is avoided through the use of a DNN to approximate not only the Q-value function (the critic network $Q(s, a; \theta^Q)$) but also the behaviour policy (the actor network $\mu(s; \theta^\mu)$). The policy is formulated as a function that deterministically maps a state to the corresponding action.

Similarly to DQN, to improve learning stability and minimize the impact of the deadly triad, experiences are also sampled from a replay buffer for training and target networks are used, in this case both for the critic θ^{Q-} and for the actor $\theta^{\mu-}$, updated using the same Polyak averaging process as described for the DQN algorithm in subsection 3.2.1. Adapting the loss function of the DQN from equation 3.8, the target Q-value is computed from the target actor network instead of taking the maximum Q-value over the possible actions, originating equations 3.10 and 3.11 for the critic loss and its gradient, respectively.

[1] Polyak averaging updates DNN weights by a weighted average of its current θ_t and previous θ_{t-1} weights in the optimisation trajectory.

To train the actor network, Silver et al. [82] proved that the gradient of the expected value of the return (denominated J in DRL notation) from following the policy approximated by actor function μ is given by equation 3.12. In this equation, $\nabla_a Q(s,a;\theta_i^Q)$ is the gradient of the critic network output with respect to the action determined by the actor network and $\nabla_{\theta_i^\mu}\mu(s;\theta_i^\mu)$ is the gradient of the actor output with respect to its parameters θ^μ. Since in this case the objective is to maximise the return of the policy approximated by the actor, parameter updates are performed through gradient ascent $\theta_{i+1}^\mu = \theta_i^\mu + \alpha_l \nabla_{\theta_i^\mu} J(\theta_i^\mu)$.

$$\mathcal{L}_i(\theta_i^Q) = E_{(s,a,r,s')\sim U(\mathcal{D})}\left[\left(r + \gamma Q^-\left(s', \mu^-(s';\theta^{\mu-});\theta_i^{Q-}\right) - Q(s,a;\theta_i^Q)\right)^2\right] \tag{3.10}$$

$$\nabla_{\theta_i^Q}\mathcal{L}(\theta_i^Q) = E_{(s,a,r,s')\sim U(\mathcal{D})}\left[\left(r + \gamma Q^-(s', \mu^-(s';\theta^{\mu-});\theta_i^{Q-}) - Q(s,a;\theta_i^Q)\right)\nabla_{\theta_i^Q} Q(s,a;\theta_i^Q)\right] \tag{3.11}$$

$$\nabla_{\theta_i^\mu} J_i(\theta_i^\mu) = E_{s\sim U(\mathcal{D})}\left[\nabla_a Q(s,a;\theta_i^Q)\nabla_{\theta_i^\mu}\mu(s;\theta_i^\mu)\right] \tag{3.12}$$

Following the algorithm as described above would lead to on-policy training, since the agent would follow the policy determined by μ directly. To train the agent off-policy and promote exploration of the action space, Lillicrap et al. [78] recommend the addition of noise $\mathcal{N}$ to the action computed by the actor network $a = \mu(s) + \mathcal{N}$. The authors of the algorithm recommend the use of Ornstein-Uhlenbeck noise [83] since its temporal auto-correlation properties promote higher exploration efficiency in inertial systems [78]. A less computationally expensive option would be the addition of Gaussian noise, which has been empirically shown to have similar results [71].

A summary of the algorithm is described in algorithm 1. Note that two additional hyperparameters are introduced in this formulation, the number of time steps taken in each episode T and the number of training episodes M.

3.2.3 Twin Delayed DDPG (TD3)

The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm [79] was developed in order to address two problems identified in the DDPG algorithm: overestimation bias and variance in the Q-value estimate. Overestimation bias occurs in the greedy update of the Q-value function (equation 3.7 for Q-Learning and 3.11 for DDPG), since the value maximization procedure defining the target for Bellman's equation will lead to overestimation of the value in the presence of noise and high variance in the value estimates. These issues introduce noise in the policy gradient computation, reducing learning speed [60].

TD3 employs three strategies to reduce the above problems.

The first modification is target policy smoothing, instead of using the target actor network output $\mu(s';\theta^{\mu-})$ to estimate the Q-Learning target, a target action a' is computed by addition of clipped Gaussian noise $n \sim \mathrm{clip}(\mathcal{N}(0,\sigma), -c, c)$, where the noise has mean 0 and standard deviation σ and is clipped

Algorithm 1 Deep Deterministic Policy Gradient algorithm (adapted from Lillicrap et al. [78])

```
Randomly initialize critic network $Q(s,a;\theta^Q)$ and actor network $\mu(s;\theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$.
Initialize target networks $Q^-(s,a;\theta^{Q-})$ and $\mu^-(s;\theta^{\mu-})$ with weights $\theta^{Q-} \leftarrow \theta^Q$ and $\theta^{\mu-} \leftarrow \theta^\mu$.
Initialize replay buffer $\mathcal{D}$.
for $episode = 1, M$ do
    Initialize random Ornstein-Uhlenbeck noise process $\mathcal{N}$ for action exploration.
    Receive initial observation state $s_1$.
    for $t = 1, T$ do
        Select action $a_t = \mu(s_t;\theta^\mu) + \mathcal{N}_t$.
        Execute action $a_t$ and observe reward $r_t$ and the new state $s_{t+1}$.
        Store transition ($s_t$,$a_t$,$r_t$,$s_{t+1}$) in $\mathcal{D}$.
        Sample a random minibatch $U$ of $N$ transitions ($s$,$a$,$r$,$s'$) from $\mathcal{D}$.
        Update critic by minimizing the loss $\mathcal{L}$ using gradient descent (equation 3.11).
        Update actor policy by applying gradient ascent (equation 3.12).
        Update target networks by Polyak averaging with:
            $\theta^{Q-} = \rho\theta^{Q-} + (1-\rho)\theta^Q$
            $\theta^{\mu-} = \rho\theta^{\mu-} + (1-\rho)\theta^\mu$
    end for
end for
```

to the interval $[c, c]$ to keep the values close to the actor output. Adding the constraint that all actions performed by the agent must lie in valid action range $[a_{min}, a_{max}]$, the smoothed target action $a'(s')$ is given by equation 3.13. The added noise prevents policy over-fitting to narrow high value peaks in the Q-function caused by numerical errors or noise in the value estimate, minimizing the effects of high variance in Q.

$$a'(s') = \mathrm{clip}\left(\mu(s';\theta^{\mu-}) + n, a_{min}, a_{max}\right), \; n \sim \mathrm{clip}\left(\mathcal{N}(0,\sigma), -c, c\right) \tag{3.13}$$

The next modification is an adaptation of Double Q-Learning [60] to DDPG called the clipped double-Q trick, where two critic networks $Q_1(s,a;\theta_i^{Q_1})$ and $Q_2(s,a;\theta_i^{Q_2})$ with different architectures or equal architectures but different random weight initialisation are trained simultaneously. The target for Bellman's equation, used for the loss function, is taken as the minimum Q-value between the two networks, reducing the impact of overestimation bias.

Taking into account both modifications, the loss functions for both networks are given by equations 3.14a and 3.14b. Note that the target action a' is computed from equation 3.13 and separate target networks are kept for each of the critic networks $Q(s,a;\theta^{Q_1-})$ and $Q(s,a;\theta^{Q_2-})$

$$\mathcal{L}_i(\theta_i^{Q_1}) = E_{(s,a,r,s')\sim U(\mathcal{D})}\left[\left(r + \gamma \min_{j=1,2} Q_j\left(s', a'; \theta_i^{Q_j-}\right) - Q(s,a;\theta_i^{Q_1})\right)^2\right] \tag{3.14a}$$

$$\mathcal{L}_i(\theta_i^{Q_2}) = E_{(s,a,r,s')\sim U(\mathcal{D})}\left[\left(r + \gamma \min_{j=1,2} Q_j\left(s', a'; \theta_i^{Q_j-}\right) - Q(s,a;\theta_i^{Q_2})\right)^2\right] \tag{3.14b}$$

The final modification is that, to minimize oscillations in the action-value function caused by updates in the policy, the actor network $\mu(s)$ is updated less frequently than both critic networks. The original implementation [79] recommends one policy update for every two value function updates. Every other aspect of implementing the TD3 algorithm is similar to DDPG. The full framework, with the changes to DDPG resulting in TD3, is shown through pseudocode in algorithm 2.

Algorithm 2 Twin Delayed Deep Deterministic Policy Gradient algorithm (based on Fujimoto et al. [79])

```
Randomly initialize critic networks $Q_1(s,a;\theta^{Q_1})$ and $Q_2(s,a;\theta^{Q_2})$ and actor network $\mu(s;\theta^{\mu})$ with weights $\theta^{Q_1}$, $\theta^{Q_2}$ and $\theta^{\mu}$.
Initialize target networks $Q_1^-(s,a;\theta^{Q_1-})$, $Q_2^-(s,a;\theta^{Q_2-})$ and $\mu^-(s;\theta^{\mu-})$ with weights $\theta^{Q_1-} \leftarrow \theta^{Q_1}$, $\theta^{Q_2-} \leftarrow \theta^{Q_2}$ and $\theta^{\mu-} \leftarrow \theta^{\mu}$.
Initialize replay buffer $\mathcal{D}$.
for $episode = 1, M$ do
    Receive initial observation state $s_1$
    for $t = 1, T$ do
        Select action $a_t(s_t) = \mathrm{clip}\,(\mu(s_t;\theta^{\mu}) + n, a_{min}, a_{max})$, where $n \sim \mathrm{clip}\,(\mathcal{N}(0,\sigma), -c, c)$.
        Execute action $a_t$ and observe reward $r_t$ and the new state $s_{t+1}$.
        Store transition ($s_t$,$a_t$,$r_t$,$s_{t+1}$) in $\mathcal{D}$.
        Sample a random minibatch $U$ of $N$ transitions ($s$,$a$,$r$,$s'$) from $\mathcal{D}$.
        Compute target action $a'(s') = \mathrm{clip}\,(\mu(s';\theta^{\mu-}) + n, a_{min}, a_{max})$, where $n \sim \mathrm{clip}\,(\mathcal{N}(0,\sigma), -c, c)$
        Update both critics by minimizing the loss $\mathcal{L}$ using gradient descent (equations 3.14a and 3.14b).
        if $t \bmod policy\_delay = 0$ then
            Update actor policy by applying gradient ascent (equation 3.12).
        end if
        Update target networks by Polyak averaging with:
            $\theta^{Q_1-} = \rho\theta^{Q_1-} + (1-\rho)\theta^{Q_1}$
            $\theta^{Q_2-} = \rho\theta^{Q_2-} + (1-\rho)\theta^{Q_2}$
            $\theta^{\mu-} = \rho\theta^{\mu-} + (1-\rho)\theta^{\mu}$
    end for
end for
```

3.2.4 Soft Actor-Critic (SAC)

Concurrently to the development of TD3, an alternative solution to the problems presented by DDPG was developed in the form of the Soft Actor-Critic (SAC) algorithm [80, 84]. This algorithm introduces a modification to the traditional definition of the Q-value function (equation 3.4b) under a stochastic policy $a \sim \pi(\cdot|s)$, where action a is sampled from policy π when the environment reaches state s, to include a term proportional to the policy's entropy $\mathcal{H} = E_{a \sim \pi}\left[-\log\left(\pi(a'|s)\right)\right]$. Entropy $\mathcal{H}$ is a measure of policy uncertainty or randomness under a certain state. Bellman's equation for the Q-value function including this term takes the form of equation 3.15, where α is a tune-able weight for the entropy term. By increasing the value of α, the agent promotes exploration by making the optimal policy more uncertain and, by decreasing it, the optimal policy becomes closer to deterministic. This behaviour of the algorithm facilitates the fine tuning of the trade-off between exploration and exploitation.

$$Q^{\pi}(s,a) = E_{\pi}[R(s,a,s') + \gamma Q^{\pi}(s',a') + \alpha\mathcal{H}(\pi(\cdot|s))] \tag{3.15}$$

Similarly to TD3, the SAC algorithm takes advantage of the clipped double-Q trick and from the use of target Q-functions in the critic's loss function.

This loss function, with the addition of the entropy term, takes the form of equation 3.16, where $\pi(a'|s';\theta^{\pi})$ is the output of the stochastic actor network and $\tilde{a}' \sim \pi(\cdot|s';\theta_i^{\pi})$ is a new action sampled from the actor network distribution for state s'. The stochasticity in the sampling of $\tilde{a}'$ adds a form of target policy smoothing while avoiding TD3's solution of adding noise to the target action.

$$\mathcal{L}_i(\theta_i^{Q_{1,2}}) = E_{(s,a,r,s')\sim U(\mathcal{D})}\left[\left(r + \gamma \min_{j=1,2} Q_j\left(s', \tilde{a}'; \theta_i^{Q_j -}\right) - \alpha \log(\pi(\tilde{a}'|s';\theta_i^{\pi})) - Q(s,a;\theta_i^{Q_{1,2}})\right)^2\right] \tag{3.16}$$

The definition for a policy's value function is also modified to include the entropy term, yielding equation 3.17, showing that the policy should maximize not only expected future return but also the expected future entropy.

$$V^{\pi}(s) = E_{a\sim\pi}[Q^{\pi}(s,a)] + \alpha\mathcal{H}(\pi(\cdot|s)) \tag{3.17}$$

The actor network must then aim to maximise the objective function J given by equation 3.18, which also includes the value estimate using the clipped double Q trick.

$$J_i(\theta_i^{\pi}) = E_{s\sim U(\mathcal{D}),a\sim\pi}\left[\min_{j=1,2} Q_j\left(s, a; \theta_i^{Q_j -}\right) + \alpha\mathcal{H}(\pi(a|s;\theta_i^{\pi}))\right] \tag{3.18}$$

In order to compute the gradient of this value function, the dependency from the policy on the expected value operator must be removed, which is done through a reparameterisation of the stochastic policy through equation 3.19. This equation computes action $\tilde{a}_{\theta^{\pi}}$ as a function $f_{\theta^{\pi}}$ of state s and independent, Gaussian noise $n \sim \mathcal{N}(0,1)$. Assuming the stochastic actor network outputs a Gaussian distribution with mean $\mu_{\pi}(s)$ and standard deviation $\sigma_{\pi}(s)$, a common choice is the use of a squashed Gaussian policy, where the output is scaled to the interval $[-1,1]$ by an hyperbolic tangent and then adjusted to the desired action range by a bias b and scaling factor k [71].

$$\tilde{a}_{\theta^{\pi}}(s,n) = f_{\theta^{\pi}}(s,n) = k\tanh(\mu_{\pi}(s) + \sigma_{\pi}(s)\cdot n) + b \tag{3.19}$$

The reparameterisation of the actor allows the objective function to be rewritten as equation 3.20, which removes the dependency on policy π from the expected value operator and is differentiable with respect to actor parameters θ^{π}.

$$J_i(\theta_i^{\pi}) = E_{s\sim U(\mathcal{D}),n\sim\mathcal{N}}\left[\min_{j=1,2} Q_j(s, \tilde{a}_{\theta_i^{\pi}}(s,n); \theta_i^{Q_j -}) + \alpha\mathcal{H}(\pi(\tilde{a}_{\theta_i^{\pi}}(s,n)|s;\theta_i^{\pi}))\right] \tag{3.20}$$

The final step of the algorithm is the determination of the entropy weight α. While the original implementation [80] this parameter is set manually, Haarnoja et al. [84] proposed a modification that adjusts α automatically through a minimisation of the expected value of the mean square error between the current entropy of policy π and a target entropy value $\bar{\mathcal{H}}$, resulting in the loss function shown in equation 3.21. This procedure keeps the average entropy of the policy close to the target value while still allowing for higher exploration in states where the optimal policy is uncertain.

$$\mathcal{L}_i(\alpha_i) = E_{a\sim\pi}[\alpha\bar{\mathcal{H}} - \alpha\mathcal{H}(\pi(a|s;\theta_i^{\pi}))] \tag{3.21}$$

Tu summarise, a pseudocode representation of the SAC algorithm is shown in algorithm 3.

Algorithm 3 Soft Actor-Critic algorithm (based on Haarnoja et al. [80])

```
Randomly initialize critic networks $Q_1(s,a;\theta^{Q_1})$ and $Q_2(s,a;\theta^{Q_2})$ and actor network $\pi(s;\theta^{\pi})$ with weights $\theta^{Q_1}$, $\theta^{Q_2}$ and $\theta^{\pi}$.
Initialize target critic networks $Q_1^-(s,a;\theta^{Q_1-})$ and $Q_2^-(s,a;\theta^{Q_2-})$ with weights $\theta^{Q_1-} \leftarrow \theta^{Q_1}$ and $\theta^{Q_2-} \leftarrow \theta^{Q_2}$.
Initialize replay buffer $\mathcal{D}$.
for $episode = 1, M$ do
    Receive initial observation state $s_1$
    for $t = 1, T$ do
        Select action $a_t \sim \pi(\cdot|s;\theta^{\pi})$.
        Execute action $a_t$ and observe reward $r_t$ and the new state $s_{t+1}$.
        Store transition ($s_t$,$a_t$,$r_t$,$s_{t+1}$) in $\mathcal{D}$.
        Sample a random minibatch $U$ of $N$ transitions ($s$,$a$,$r$,$s'$) from $\mathcal{D}$.
        Compute target action $\tilde{a}' \sim \pi(\cdot|s';\theta^{\pi})$
        Update both critics by minimizing the loss $\mathcal{L}$ using gradient descent (equation 3.16).
        Update actor policy by applying gradient ascent and the reparameterisation trick (equation 3.20).
        Update target networks by Polyak averaging with:
            $\theta^{Q_1-} = \rho\theta^{Q_1-} + (1-\rho)\theta^{Q_1}$
            $\theta^{Q_2-} = \rho\theta^{Q_2-} + (1-\rho)\theta^{Q_2}$
    end for
end for
```

Chapter 4

Mutriku OWC model

In order to use RL algorithms to determine a control law for the OWC, an external environment is required for the agent to interact and learn. Due to the number of interactions and training time required, a simulation model for the Mutriku OWC is implemented in *Simulink*. In this case simulation speed is only limited by the available computer hardware, since the plant may be simulated faster than real time. Nevertheless, before application in a real plant, training using a physical prototype is also a possibility, although this would require time and resources that are outside the scope of this work.

The model used in this thesis is based on a *Simulink* model described in a previous case study by Henriques et al. [34] and this chapter is dedicated to the description and derivation of the equations describing this model. In order to achieve this, the model was split in several subsystems: excitation force generation from the wave spectrum, hydrodynamics of the water column, aerodynamics of air compression and expansion, turbine performance curves, generator performance and dynamics and models for valves both in series and parallel with the turbine. This constitutes a wave-to-wire model, since all stages of energy conversion are considered from the mechanical energy in wave oscillation to energy in the form of electricity in the generator.

A diagram describing the subsystems and relevant variables that describe the system's dynamics is shown in figure 4.1.

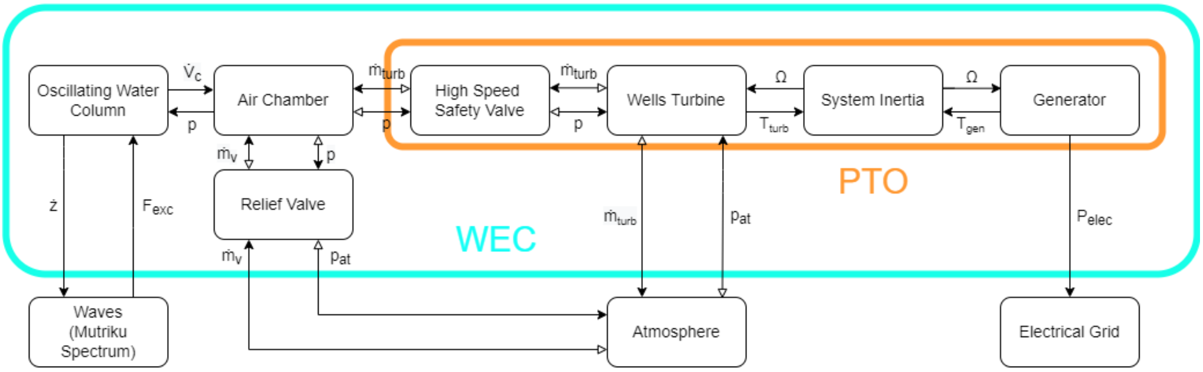


Figure 4.1: Diagram describing the Mutriku system model.

4.1 Plant description

The Mutriku Wave Power Plant is an OWC installation in the town of the same name, located in the province of Guipuzkoa in Spain's Basque Country (see figure 4.3 a)). The plant is operated by Ente Vasco de la Energía (the Basque energy board) and was built in 2007-08 in an already planned breakwater built to protect the city's port [85]. Power generation to provide electricity to the local grid started in July 2011 [86].

The power plant contains 16 OWC chambers, each equipped with a self-rectifying Wells turbine, shown in figure 4.2, although, after a storm in 2009 , only 14 of them are operational [87].

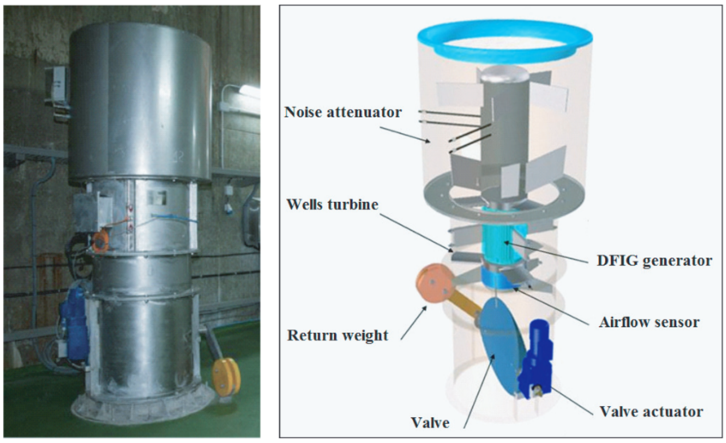
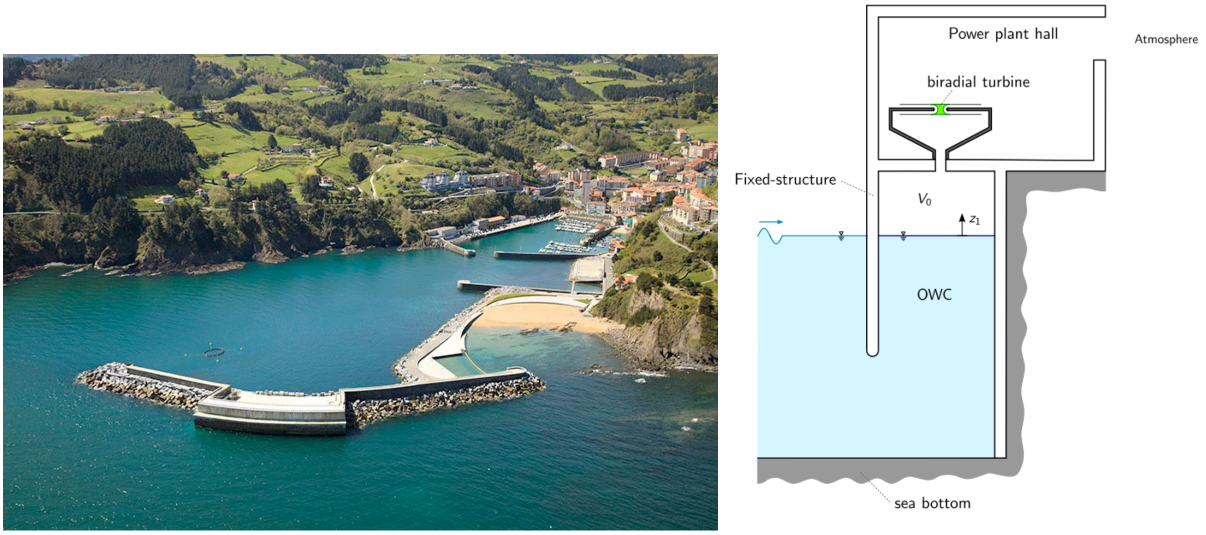


Figure 4.2: Photo and diagram showing one of the Wells turbines and generators installed at Mutriku.

Each turbine air chamber, represented in figure 4.3 b), has a width of 4.5 m, depth of 4.3 m and height of 10 m [34].



(a) Aerial view of the Mutriku breakwater.

(b) Schematic view of the Mutriku OWC chamber.

Figure 4.3: Mutriku Wave Power Plant.

The Mutriku OWC is considered a point absorber type WEC, meaning that its dimensions are small compared to the predominant wavelengths in the surrounding ocean [12, 20]. A common modelling assumption in this type of OWC devices is the modelling of the water surface as a solid piston acting on the compressible air chamber above it, which allows for the application of linear wave theory and previously derived theory for the integration between ocean waves and ships or other floating structures [12]. These two conditions will be fundamental both in the generation of the wave excitation force and modelling the dynamics of the water column and the air chamber.

4.2 Wave Excitation Force Generation

A numerical model for the excitation force exerted by the sea waves on the compressible air chamber is needed to simulate the behaviour of the OWC. For the Mutriku site, previous research [85] suggested the classification of the local wave climate into 14 sea states (SS) using spectral analysis, where each state is defined by their significant wave height, H_s, energy period, T_e, and probability of occurrence p_o, as defined in table 4.1. These sea states represent 63 percent of the total wave energy spectrum, with the remaining coming from smaller waves that are unable to generate significant power and, as such, will not be considered for the training simulation.

Table 4.1: Sea states characterising the wave climate at the Mutriku power plant [85].

Sea state number SS	Significant Height H_s (m)	Energy Period T_e (s)	Probability p_o (%)
1	0.88	5.5	3.23
2	1.03	6.5	3.44
3	1.04	7.5	5.08
4	1.02	8.5	6.11
5	1.08	9.5	10.73
6	1.19	10.5	9.31
7	1.29	11.5	9.52
8	1.48	12.5	7.42
9	1.81	13.5	2.75
10	2.07	14.5	2.96
11	2.59	15.5	1.34
12	2.88	16.5	0.40
13	3.16	11.5	0.27
14	3.20	12.5	0.42
Low Energy Waves	-	-	37.02

The model for the waves may then be generated from the characteristic sea states using equation 4.1, where ω is the wave frequency, S_J are the JONSWAP spectra used to model wind generated waves and $\varphi_{Mutriku}$ is an attenuation function derived from experimental data recovered in the Mutriku site [57], shown in figure 4.4. The JONSWAP spectra are high seas spectra dependent on the characteristic sea state, and the attenuation function adjusts them to the local conditions of Mutriku.

$$S_{Mutriku}(\omega) = S_J(\omega)\varphi_{Mutriku}(\omega) \tag{4.1}$$

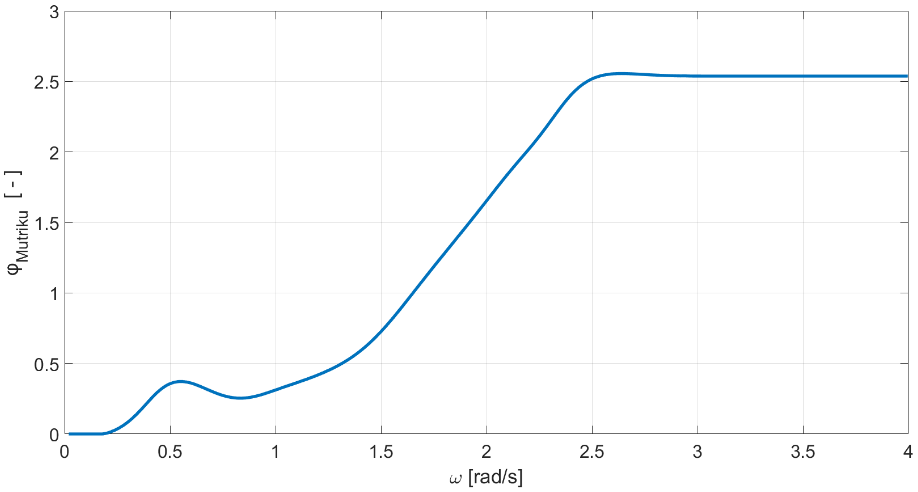


Figure 4.4: Attenuation function between offshore wave data and the Mutriku local wave climate (adapted from Henriques et al. [34]).

The JONSWAP (Joint North Sea Wave Observation Project) spectrum [88] uses experimental data to determine that the wave spectrum from wind waves keeps developing over a large area and period of time. The authors modelled this phenomenon by the addition of a peak enhancement factor γ_s^a to the Pierson-Moskowitz spectrum [89], which assumes that the waves are fully developed and in equilibrium with the wind forces. The spectrum thus takes the form of equations 4.2a to 4.2e, as indicated by Henriques et al. [34]. Note that in this equation A_{γ_s} is a normalising factor, S_{PM} is the Pierson-Moskowitz spectrum, γ_s is the spectrum sharpness parameter, with a value of 2.8, H_s is the significant wave height and ω_p is the peak frequency, defined as $2\pi/T_p$. The peak period T_p is related to the energy period in the JONSWAP spectrum by the relation $T_p = 2\pi\sqrt[4]{\frac{5/4}{1054}}T_e$.

$$S_J(\omega) = A_{\gamma_s}\gamma_s^a S_{PM}(\omega) \tag{4.2a}$$

$$A_{\gamma_s} = 1 - 0.287\ln(\gamma_s) \tag{4.2b}$$

$$a = \exp\left(-\frac{(\omega-\omega_p)^2}{2\omega_p^2\sigma^2}\right) \tag{4.2c}$$

$$\sigma = \begin{cases} 0.07, & \omega \le \omega_p \\ 0.09, & \omega > \omega_p \end{cases} \tag{4.2d}$$

$$S_{PM}(\omega) = \frac{5}{16}\frac{H_s^2\omega_p^4}{\omega^5}\exp\left(-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4\right) \tag{4.2e}$$

The original JONSWAP spectrum, along with the attenuated spectrum defined in equation 4.1 for the highest probability sea state (number 5) is shown in figure 4.5.

From the wave spectrum, and assuming linear water wave theory, the excitation force of the OWC may be computed as a sum of n (here assumed to be 10^3) sinusoidal waves in superposition, as indicated by equation 4.3. In this equation, Γ is imaginary piston heave excitation response, ϕ is the excitation response to the wave component, ϕ_r is a uniform random variable in the $[0, 2\pi]$ interval repre-

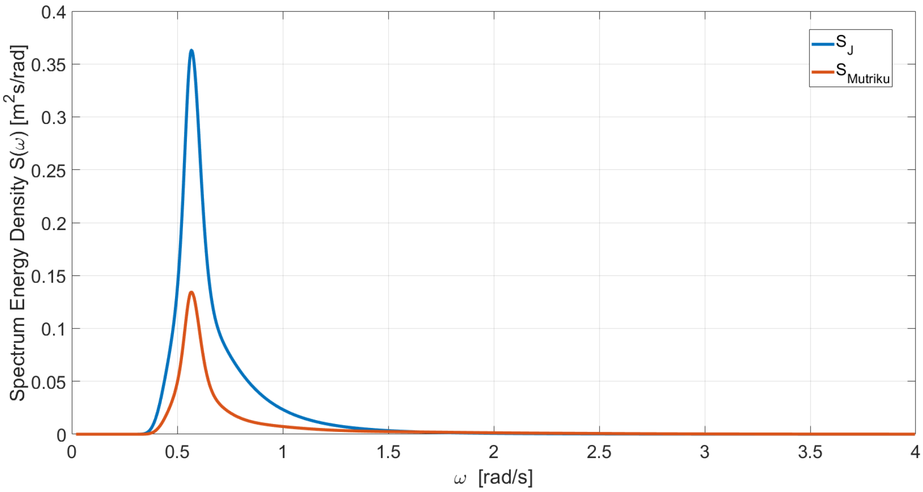


Figure 4.5: JONSWAP and local Mutriku spectra for sea state number 5.

senting a random phase, and A is the amplitude of each frequency component in the wave, as computed by equation 4.4a. The amplitudes are discretised in frequency intervals $\Delta\omega_i$ as defined by Henriques et al. [90] and shown in equations 4.4b and 4.4c, in which $rand$ indicates a random number in the $[0,1]$ interval.

$$F_{exc} = \sum_{i=1}^{n} \Gamma(\omega_i) A(\omega_i) \cos(\omega_i t + \phi_i(\omega) + \phi_{r,i}) \tag{4.3}$$

$$A(\omega_i) = \sqrt{2\Delta\omega_i S_{Mutriku}(\omega_i)} \tag{4.4a}$$

$$\Delta\omega_i = (1 + \pm 0.2 rand)\,\frac{200}{n} \tag{4.4b}$$

$$\omega_i = \omega_{i-1} + \frac{1}{2}(\Delta\omega_i + \Delta\omega_{i-1}), \quad \omega_1 = 0.02\; rad/s \tag{4.4c}$$

Function $\Gamma(\omega)$ and $\phi(\omega)$ were computed by Henriques et al. [34] using the *WAMIT* software for a discrete set of frequencies in the interval $[0,4]$ rad/s and then interpolated to be used in the model. A graphical representation of Γ and ϕ is shown in figure 4.6.

4.3 Water Column Hydrodynamic Model

To model the air chamber compression, an equation in the time domain for the heave motion of the water surface as a function of excitation force is required. Based on the work of Cummins [91] and Ogilvie [92], the differential equation 4.5 describing the variation of the imaginary water piston's surface height z is formulated, where m is the mass of the piston, A^{∞} is the added mass at infinite frequency, ρ_w is the water density, g is the acceleration of gravity, S is the surface area of the OWC, p_{at} is the atmospheric pressure, $p^* = p/p_{at} - 1$ is the dimensionless air pressure in the air chamber, F_{exc} is the wave excitation force and R is the wave radiation memory term.

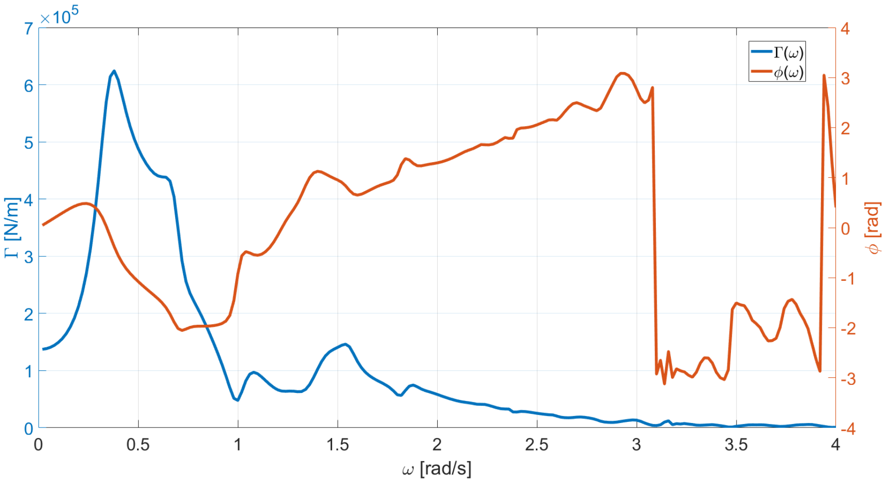


Figure 4.6: Piston heave excitation response Γ and excitation response to the wave component ϕ (adapted from Henriques et al. [34]).

$$(m + A^{\infty})\ddot{z} = -\rho_w g S z - p_{at} S p^* + F_{exc} - R \tag{4.5}$$

The values of the geometric and physical constants used in equation 4.5 are shown in table 4.2.

Table 4.2: OWC model parameters.

Parameter Name	Symbol	Value
Imaginary rigid piston mass	m	72 010 kg
Added mass at infinite frequency	A^{∞}	27 748 kg
Water density	ρ_w	1025 kg m^{-3}
Acceleration of gravity	g	9.81 m s^{-2}
OWC Surface Area	S	19.35 m^2
Absolute atmospheric air pressure	p_{at}	$1.013\,25 \times 10^5$ Pa

In equation 4.5, $R + A^{\infty}\ddot{z}$ is the wave radiation force, which can be decomposed into an instantaneous added mass term $A^{\infty}\ddot{z}$, summed to the real mass inertial term $m\ddot{z}$, and a memory term R, expressed by the convolution integral in equation 4.6a, where the kernel K is the impulse response function of the wave radiation. The calculation of the integral in equation 4.6a presents a computational challenge due to its dependency not only on previous wave data, but also on the prediction of future data [93]. The solution is the approximation of the integral through the Prony method [94].

The Prony method states that K may be approximated as a summation of complex exponential functions, as stated in equation 4.6b, in which coefficients α_k and β_k are either real numbers or pairs of complex conjugates. By applying the approximation to the integral 4.6a, differentiating and applying Leibniz's rule, the expression in 4.6c is obtained. Defining I_k as in equation 4.6d, equation 4.6c may be expressed as differential equation 4.6e. Applying the principle of superposition to this equation and writing it in matrix form, it takes the form of the state-space model in equation 4.6f. This model may be manipulated using the properties of complex conjugates to eliminate the imaginary part, thus avoiding complex integration [34]. Finally, the wave radiation memory term is defined by equation 4.6g. For this work's model, a summation of 16 exponential functions was used to approximate the kernel.

$$R = \int_0^t K(t-\tau)\dot{z}(\tau)d\tau \tag{4.6a}$$

$$K(t) = \sum_{k=1}^{p} \alpha_k e^{\beta_k t} \tag{4.6b}$$

$$\sum_{k=1}^{p} \frac{d}{dt}\left(\int_0^t \alpha_k e^{\beta_k(t-\tau)}\dot{z}(\tau)d\tau\right) = \sum_{k=1}^{p}\left(\int_0^t \alpha_k\beta_k e^{\beta_k(t-\tau)}\dot{z}(\tau)d\tau + \alpha_k\dot{z}(t)\right) \tag{4.6c}$$

$$I_k = \int_0^t \alpha_k e^{\beta_k(t-\tau)}\dot{z}(\tau)d\tau \tag{4.6d}$$

$$\sum_{k=1}^{p} \dot{I}_k = \sum_{k=1}^{p}(\beta_k I_k + \alpha_k\dot{z}(t)) \tag{4.6e}$$

$$\dot{I}_r = \beta_r I_r + \alpha_r \dot{z} \tag{4.6f}$$

$$R = \sum_{k=1}^{p} I_k \tag{4.6g}$$

4.4 Air Chamber Expansion Model

To model the compression of the air chamber by the sea surface oscillation, and the flow passing through the turbine, it is common to assume the air is a perfect gas and that the compression and expansion of the air in the chamber is an isentropic process [37]. Through a mass balance applied to the air chamber, equation 4.7 is derived, where $\dot{m}$ is the mass of air flowing out of the chamber, ρ_c is the air density inside the chamber and V_c is the instantaneous air volume in the chamber, given by $V_c = V_0 - Sz$. At hydrostatic conditions, the air chamber has a height of 7.45 m and area S, yielding a reference volume of $V_0 = 144.1575\ \text{m}^3$.

$$-\dot{m} = \rho_c\dot{V}_c + V_c\dot{\rho}_c \tag{4.7}$$

With the above assumptions, formula 4.8 may be derived for the air density as a function of atmospheric air density ρ_{at}, dimensionless air pressure in the chamber p^* and the specific heat ratio $\gamma = c_p/c_v$ of 1.4.

$$\rho_c = \rho_{at}(p^* + 1)^{1/\gamma} \tag{4.8}$$

By the manipulation of equations 4.7 and 4.8, a differential equation is derived for the dimensionless air pressure p^*, shown in equation 4.9.

$$\dot{p}^* = -\gamma(p^*+1)\frac{\dot{V}_c}{V_c} - \gamma(p^*+1)^{\frac{\gamma-1}{\gamma}}\frac{\dot{m}}{\rho_{at}V_c} \tag{4.9}$$

4.5 Turbine Dynamics Model

The main feature of the self-rectifying Wells turbines used at Mutriku is that rotation and torque direction is independent of the direction of air flow, eliminating the need for the installation of rectifying non-return valves in the turbine duct. A comprehensive description of this type of turbine and its hydrodynamic and geometric properties may be found in the review paper by Falcão and Henriques [12]. For the model used in this work, only a single Wells turbine of the 16 installed at Mutriku will be simulated.

To describe the performance of the turbine, a set of dimensionless numbers [95] is defined, as shown in equations 4.10: dimensionless pressure head Ψ (equation 4.10a), dimensionless flow rate Φ (equation 4.10b), dimensionless power coefficient Π (equation 4.10c), and turbine efficiency η_{turb}, the latter being defined in equation 4.10d as the ratio between the turbine's output mechanical power P_{turb} and the available pneumatic power from the air chamber P_{pneu}. Additional variables that influence the dynamic behaviour of the turbine used to generate the performance curves are the turbine rotor diameter D (0.75 m), the turbine rotational velocity Ω, the stagnation pressure head from the OWC's air chamber to the atmospheric air $\Delta p = p_{at}p^*$, the inlet air density in stagnation conditions ρ_{in} and the mass flow rate into the turbine $\dot{m}_{turb}$.

$$\Psi = \frac{\Delta p}{\rho_{in}\Omega^2 D^2} \tag{4.10a}$$

$$\Phi = \frac{\dot{m}_{turb}}{\rho_{in}\Omega D^3} \tag{4.10b}$$

$$\Pi = \frac{P_{turb}}{\rho_{in}\Omega^3 D^5} \tag{4.10c}$$

$$\eta_{turb} = \frac{P_{turb}}{P_{pneu}} = \frac{\Pi}{\Phi\Psi} \tag{4.10d}$$

The direction of air flow varies depending on air pressure inside the OWC chamber: if pressure is higher than the atmospheric pressure ($p^* > 0$), air is exhaled to the atmosphere and the inlet is on the side of the air chamber, while, if the pressure is lower than the atmospheric pressure ($p^* < 0$), air is inhaled to the air chamber from the surrounding atmosphere and the inlet is on the side of the surrounding air. This changes the definition of inlet stagnation air density ρ_{in}, as shown in equation 4.11. Atmospheric air density is approximated by $\rho_{at} = \frac{p_{at}}{RT_{at}}$ assuming the perfect gas model with a gas constant R of 287 J kg^{-1} K^{-1} and that the atmosphere's temperature is approximately 288.15 K. (15 ℃)

$$\rho_{in} = \begin{cases} \rho_c, & \text{if } p^* > 0 \\ \rho_{at}, & \text{if } p^* \leq 0 \end{cases} \tag{4.11}$$

For air flow inside the turbine with a Reynolds number $Re > 10^6$ and a Mach number $Ma < 0.3$ [95], which were both taken as assumptions for the model, it is an acceptable approximation to consider that these dimensionless groups have no effect on the turbine's performance, meaning each of the previously introduced dimensionless numbers may be presented as functions of the dimensionless pressure head $\Phi = f_\Phi(\Psi)$, $\Pi = f_\Pi(\Psi)$ and $\eta_{turb} = f_\eta(\Psi)$, as shown in figure 4.7.

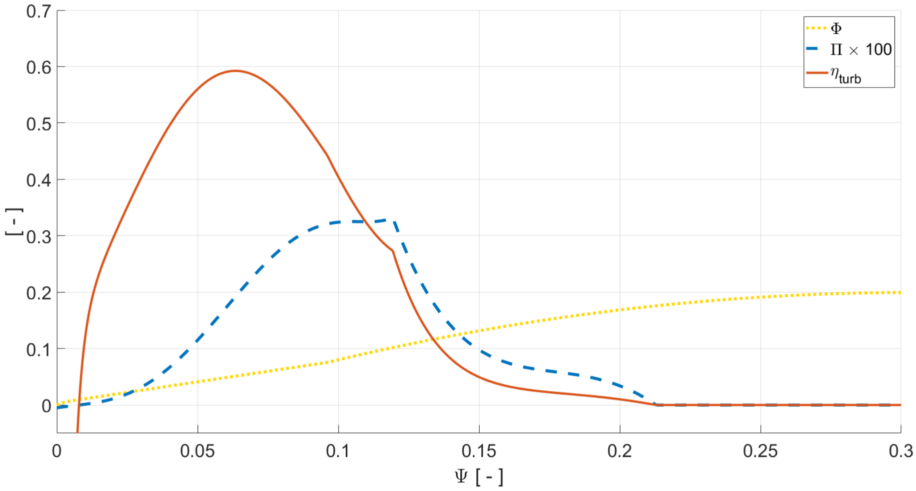


Figure 4.7: Turbine dimensionless flow rate Φ, dimensionless power coefficient Π and turbine efficiency η, as a function of dimensionless pressure head Ψ (adapted from Henriques et al. [34]).

The $f_\Phi(\Psi)$ and $f_\Pi(\Psi)$ functions were implemented in the simulation through a numerical fit for measured experimental points, shown approximately in equations 4.12 and 4.13, while η_{turb} is obtained directly from equation 4.10d [34].

$$f_\Phi(\Psi) = \begin{cases} \text{sign}(\Psi)(-133.811|\Psi|^2 + 2.1516|\Psi|), & \text{if } |\Psi| \leq 5.247 \times 10^{-3} \\ \text{sign}(\Psi)(0.7473|\Psi| + 0.0037), & \text{if } 5.247 \times 10^{-3} < |\Psi| \leq 0.096 \\ \text{sign}(\Psi)(-2.8793|\Psi|^2 + 1.7481|\Psi| - 0.0659), & \text{if } |\Psi| > 0.096 \end{cases} \tag{4.12}$$

$$f_\Pi(\Psi) = \begin{cases} 10856|\Psi|^6 - 2028|\Psi|^5 - 31.208|\Psi|^4 + 18.382|\Psi|^3 - 0.333|\Psi|^2 & \\ +0.008|\Psi| - 5 \times 10^{-5}, & \text{if } |\Psi| \leq 0.119 \\ 46.672|\Psi|^4 - 42.619|\Psi|^3 + 13.862|\Psi|^2 - 1.943|\Psi| + 0.100, & \text{if } 0.12 < |\Psi| \leq 0.21 \\ 0, & \text{if } |\Psi| > 0.21 \end{cases} \tag{4.13}$$

4.6 Generator model

By performing an energy balance on the turbine-generator system, its dynamics may be described by equation 4.14.

$$\frac{dE_{kin}}{dt} = P_{turb} - P_{gen} \tag{4.14}$$

Turbine power P_{turb} is dependent on the available pneumatic power and turbine efficiency for the corresponding operating point, as shown in equation 4.10d. Another way to determine the turbine power is through the manipulation of equation 4.10c, as shown in equation 4.15.

$$P_{turb} = f_{\Pi}(\Psi)\rho_{in}D^5\Omega^3 \tag{4.15}$$

Note that in the above equation the turbine power is proportional to the cube of the rotational speed, Ω^3. To maintain the turbine at its optimal operating point $f_{\Pi}(\Psi_{opt})$ in steady state ($dE_{kin}/dt = 0$) the generator power P_{gen} should balance the turbine power, taking the form $P_{gen} = a_{opt}\Omega^3$, where a_{opt} is approximately constant (considering small variations in inlet air density ρ_{in}) and equal to $f_{\Pi}(\Psi_{opt})\rho_{in}D^5$, where $f_{\Pi}(\Psi_{opt})$ is the dimensionless power at dimensionless pressure head Ψ_{opt}, corresponding to the maximum of the turbine efficiency curve. In practice, the inertia of the turbine-generator set's rotating parts will be large enough to cause the system's kinetic energy ($E_{kin} = \frac{1}{2}I\Omega^2$, where I is the value of the inertia) to vary through time. Other factors like the effects of the OWC's hydrodynamics and variations in the conversion from mechanical to electrical power suggest a more complex control law for the generator power.

Previous work [32, 34, 57] suggests that a control law of the type $P_{gen} = a\Omega^b$ provides an adequate control scheme for the generator, which allows the reformulation of the energy balance into equation 4.16, which, assuming $\Omega \neq 0$ and knowing that $P_{gen,turb} = \Omega T_{gen,turb}$, reduces to the equilibrium of moments along the rotation axis shown in equation 4.17, where T_{gen} and T_{turb} are the torques exerted by the turbine and the generator on the rotating parts, respectively.

$$\dot{\Omega} = \frac{f_{\Pi}(\Psi)\rho_{in}D^5\Omega^3 - a\Omega^b}{I\Omega} \tag{4.16}$$

$$\dot{\Omega} = \frac{T_{turb} - T_{gen}}{I} = \frac{f_{\Pi}(\Psi)\rho_{in}D^5\Omega^2 - a\Omega^{b-1}}{I} \tag{4.17}$$

In this model it is assumed that the control law will be programmed in the plant's PLC (Programmable Logic Controller) and adjusted by the generator's power electronics [32], as well as that the generator dynamics are orders of magnitude faster than the OWC dynamics meaning the generator torque may be freely adjusted, which has been shown to produce low modelling error, particularly when reactive torque is not introduced [96, 97]. This is the case in the model in consideration since, according to the convention defined in equation 4.17, the generator torque T_{gen} will be assumed to be positive, meaning it will always act as a resistive torque counteracting the turbine moment.

Other restrictions that were considered for the generator torque control law are the maximum generator power output P_{gen}^{rated} and the maximum generator torque T_{gen}^{rated}, meaning that for any generic theoretical torque control law T_{gen}^{u}, the model generator torque will be given by equation 4.18.

$$T_{gen} = \max\left(0, \min\left(T_{gen}^{u}, T_{gen}^{rated}, \frac{P_{gen}^{rated}}{\Omega}\right)\right) \tag{4.18}$$

The electrical power P_{elec} output from the generator may be approximated by equation 4.19, where η_{gen} is the generator's efficiency in the conversion from mechanical to electrical power and Λ is the generator's load factor, defined as $\Lambda = P_{gen}/P_{rated}$ (P_{rated} is the rated power of the generator).

The performance curves $\eta_{gen}(\Lambda)$ for generators similar to the one used at Mutriku were determined by Tedeschi et al. [98], resulting in the curve defined in equation 4.20 and shown in figure 4.8. This curve was used in the model along with equation 4.19 to estimate the electricity production by the generator.

$$P_{elec} = \eta_{gen}(\Lambda) P_{gen} \tag{4.19}$$

$$\eta_{gen}(\Lambda) = \frac{7.32 \times 10^6 \Lambda - 1.53 \times 10^6 \Lambda^2}{1.80 \times 10^5 + 7.88 \times 10^6 \Lambda - 1.17 \times 10^6 \Lambda^2 + \Lambda^3} \tag{4.20}$$

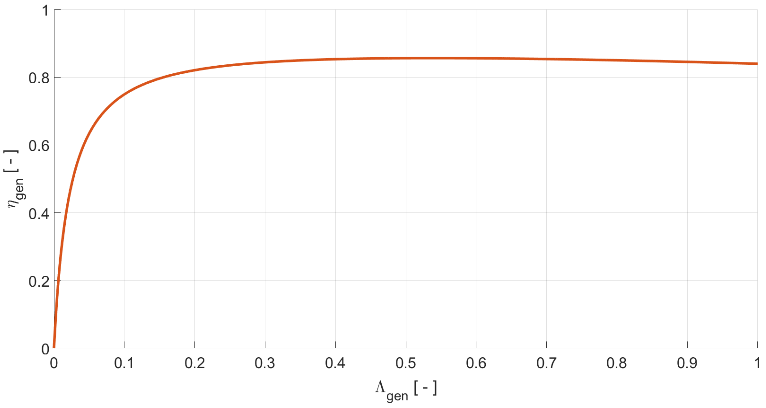


Figure 4.8: Generator efficiency η_{gen} as as function of load factor Λ (adapted from Henriques et al. [34]).

All relevant parameters for the generator, are shown in table 4.3.

Table 4.3: Generator model Parameters.

Parameter Name	Symbol	Value
Turbine-generator set inertia	I	3.06 kg m^2
Generator rated power	P_{gen}^{rated}	18.5 kW
Generator rated torque	T_{gen}^{rated}	90.1875 N m

4.7 Valve Models

As previously mentioned, two types of valves are used in the control of OWC devices: High-Speed Safety Valves (HSSV) installed in series with the turbine to control the air flux, and relief valves installed in parallel with the turbine to control the air chamber pressure [34, 41]. In this thesis, one of the challenges is the analysis of the use of a relief valve in the Mutriku OWC to increase power production in highly energetic sea states, while the HSSV is used only as a fail-safe mechanism to prevent water from reaching the turbine and to prevent excessive rotation velocity.

The model for the air flow through the relief valve $\dot{m}_v$, introduced by Falcão and Justino [37], is represented by equation 4.21, where A_v is the effective valve area, which is the geometric area of the valve duct multiplied by a flow coefficient, ρ_{in} is the same stagnation air density as in equation 4.11 and k_v is the valve aperture state in the interval $[0, 1]$, with 0 and 1 representing a completely closed or opened valve, respectively. The use of this equation requires the simplification of assuming the hydrodynamics of the valve are independent of valve aperture k_v.

$$\dot{m}_v = \text{sign}(p^*) A_v k_v \sqrt{2\rho_{in}|p^* p_{at}|} \tag{4.21}$$

The HSSV valve is modelled assuming that its aperture u is a binary variable, meaning it is either completely open ($u = 1$) or closed ($u = 0$). The turbine power, including this safety mechanism, is thus given by equation 4.22.

$$P_{turb} = f_{\Pi}(u\Psi)\rho_{in} D^5 \Omega^3 \tag{4.22}$$

To ensure the integrity of the turbine and generator, in the simulation the HSSV was programmed to close when the turbine-generator set reaches the maximum rotation velocity that will exceed the maximum centrifugal stress on the turbine, determined to be $\Omega_{max} = 4000$ rpm [34], although this safety mechanism is disabled during controller training to ensure it is exposed to high turbine rotation velocities.

Chapter 5

Problem Definition and Controller Design

In this chapter the architectures for Reinforcement Learning control of the Mutriku OWC will be defined. Three different frameworks for continuous action reinforcement learning were implemented: DDPG, TD3 and SAC. To fully define the state space, a sea state estimator using the algorithm described in section 5.3 will be implemented and tested. Finally, a baseline for plant performance under each of the 14 sea states using the cubic power law is presented, which will be used in further sections to evaluate controller performance.

5.1 MDP Formulation

As mentioned in section 3.1, implementing a Reinforcement Learning controller involves the formulation of the control problem in the MDP framework, which includes the definition of the environment, agent, state and action spaces and reward function.

The environment will be the entire Mutriku simulation model described in section 4, including the sea state estimator. From this environment, a set of observations is taken at every control time step, which will constitute the state space. The variables considered for the state space are the estimates for significant wave height H_s and energy period T_e, the instant dimensionless pressure p^*, the turbine rotation velocity Ω and its derivative $\dot{\Omega}$ and the control actions from the previous time step for the generator torque T_{gen}^{t-1} and valve aperture k_v^{t-1} and their time derivatives $\dot{T}_{gen}^{t-1}$ and $\dot{k}_v^{t-1}$.

The action space a that the agent will output to the environment are the time derivative of the generator torque $\dot{T}_{gen}^{t}$ and the valve aperture $\dot{k}_v^{t}$. It was verified that using the value of the torque and valve aperture instead of their derivative led to large control action oscillations in the initial exploration phases, slowing down learning due to frequent early terminations of the simulations.

The value of the torque and valve aperture are obtained by numerical integration of the imposed derivative at every time step, using as initial conditions on the first step $k_v^0 = 0$ and $T_{gen}^0 = a_{opt}\Omega_0^{b_{opt}-1}$. To summarise, the action space a and state space s, are formulated in equations 5.1 and 5.2, respectively.

$$s = \begin{bmatrix} H_s & T_e & p^* & \Omega & \dot{\Omega} & T_{gen}^{t-1} & k_v^{t-1} & \dot{T}_{gen}^{t-1} & \dot{k}_v^{t-1} \end{bmatrix} \tag{5.1}$$

$$a = \begin{bmatrix} \dot{T}_{gen}^{t} & \dot{k}_v^{t} \end{bmatrix} \tag{5.2}$$

The agent architectures tested were DDPG, TD3 and SAC (see sections 3.2.2 to 3.2.4 for more details). .

While action and state normalisation is not required in Deep Reinforcement Learning (DRL) due to the change in the normalisation range as new state values are observed by the agent through new experiences, it is still recommended in practice to keep measurements in the same order of magnitude [71], as long as a range for the actions and states may be estimated. To improve learning stability, state and action variables that have absolute limits in their possible value, such as the torque or valve apertures were scaled to $[-1, 1]$ interval. For the remaining variables, scaling factors were also applied to keep them in the same order of magnitude, although a perfect re-scaling to the same interval is impossible.

The reward function will be defined according to the control objectives: maximise power production, avoid excessive PTO control effort and preserve the structural integrity of the system. Defining this function is an iterative trial and error process that requires analysis of the agent's behaviour during training.

A generic structure for an appropriate reward function r_t for this problem is shown in equation 5.3. In this equation, a positive reward is obtained which is proportional to the average electrical power $\overline{P_{elec}}$, taken as the moving average of the output power over the controller's sampling interval T_s, which is normalised to the $[0, 1]$ range by the maximum possible electric power output by the generator, defined as the generator's rated power P_{gen}^{rated} multiplied by the maximum generator efficiency η_{gen}^{max}, for which equation 4.20 yields a value of 0.856.

Negative reward penalties are added proportional to the dimensionless generator torque and valve aperture in the previous time step $T_{gen}^{t-1}/T_{gen}^{rated}$ and k_v^{t-1}, in order to avoid excessive control effort.

Excessive variations in the output power are also penalised, by applying a penalty to the time derivative of generated power raised an even power e_{even}. A larger even power will scale the power derivative term non-linearly, giving a harsher penalty for higher derivative values.

Proportionality constants k_1, k_2, k_3 and k_4 allow for separate tuning of the importance of each of the reward terms.

In addition, three binary flags f_1, f_2 and f_3 are included in the function. Flag f_1 takes a value of 1 if the simulation is terminated early due to the turbine rotation velocity reaching zero or due to the dimensionless pressure in the chamber reaching a threshold value of $|p^*| > 0.25$. At every instant when the torque generated by the control action requires a higher generator output power than the rated power ($P_{gen} > P_{gen}^{rated}$), flag f_2 is activated. Finally, if the generator-turbine set reaches its maximum rotation velocity Ω_{max}, flag f_3 will change to 1 and only change back to 0 when Ω is reduced to a safe operating value under Ω_{max}.

$$r_t = k_1 \frac{\overline{P_{elec}}}{\eta_{gen}^{max} P_{gen}^{rated}} - k_2 \left(\frac{d}{dt} \frac{P_{elec}}{\eta_{gen}^{max} P_{gen}^{rated}} \right)^{e_{even}} - k_3 \frac{T_{gen}^{t-1}}{T_{gen}^{rated}} - k_4 k_v^{t-1} - f_1 - f_2 - f_3 \tag{5.3}$$

The values of the proportional constants $k_{1,2,3,4}$ and flags $f_{1,2,3}$ are given in table 5.1.

Table 5.1: Reward constants and flags.

k_1	k_2	k_3	k_4	f_1	f_2	f_3	e_{even}
10	4	0.1	0.05	-20	-5	-0.5	8

5.2 Power Production Baseline

To establish a baseline as a point of comparison through which the performance of the DRL controllers can be evaluated, the previously mentioned cubic power law $P_{gen} = a\Omega^b$ is tested in the simulation model. The parameters used are $b = 3$ and $a = f_{\Pi}(\Psi_{opt})\rho_{in}D^5 \approx 2 \times 10^{-4}$, corresponding to the theoretical optimal power law to maximise turbine efficiency. To analyse the effect of the variability associated with the random nature of the excitation force, 50 simulations of 30 min each were performed for each sea state (identified by their number SS in table 4.1), measuring the average electrical power generation $\overline{P_{elec}}$ during each trial, shown in figure 5.1.

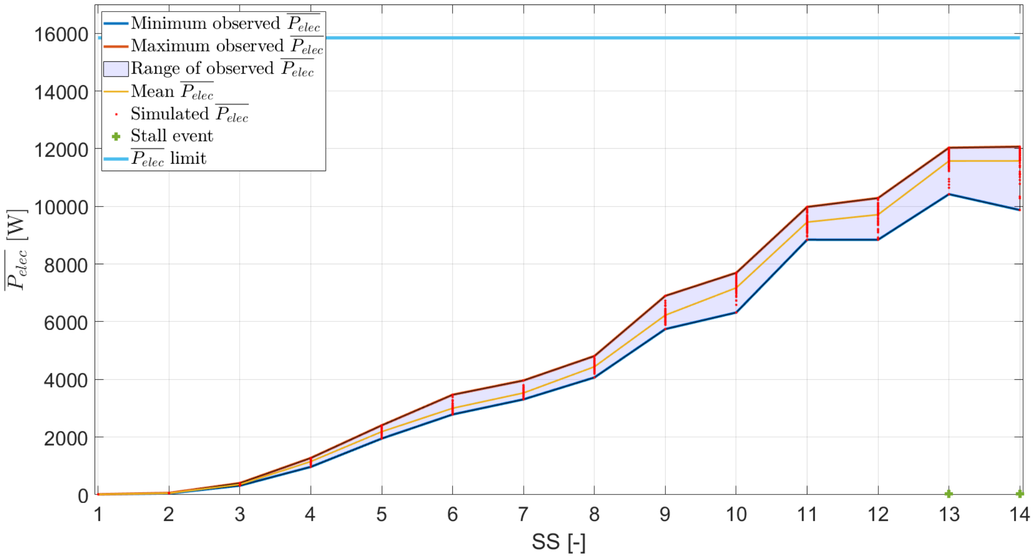


Figure 5.1: Range of observed values for the average electrical power generation $\overline{P_{elec}}$ in each sea state from the Mutriku spectrum.

As expected, since the sea states in table 4.1 were ordered by the amount of energy they contain, power generation on average increases with the sea state number, with generation in the higher sea states approaching the theoretical maximum that the generator may produce, defined as in equation 5.3 as $\eta_{gen}^{max} P_{gen}^{rated}$. Nevertheless, the random nature of the excitation force has a significant effect on the generated power, with the interval between maximum and minimum observed power generation ranging between 12 and 27% of the mean observed value. A notable phenomenon is the occurrence of turbine stalling twice in sea states 13 and 14, where a large initial peak in pressure causes the system to halt.

To better understand the dynamics under this basic control law, a set of key variables describing the system performance under this law is shown in figure 5.2.

The dimensionless pressure data $p^*(t)$ shows that the sea state 13 has an average pressure amplitude that is approximately 2.5 and 10 times the average pressure amplitude of sea states 3 and 8, respectively, showing that this sea state will have higher power generation potential. Note that for the lower energy sea state 3, the rotational velocity and consequently generator torque takes a lower value, leading to a lower generator efficiency η_{gen}. This is to be expected, as the power law optimizes the system to operate in the turbine's best efficiency point but not the generator's, in which efficiency is maximum for load factors Λ over 0.3 (see figure 4.8). Gains in this case may be obtained by balancing the generator and turbine efficiency in the control law.

Meanwhile, for sea state 13, while the generator efficiency reaches a value close to its maximum, the maximum allowed rotation velocity Ω_{max} is reached during most of the operation, meaning the safety valve would need to be open to preserve turbine integrity. In this type of operating mode focus would be on avoiding the usage of the HSSV and aiming to bring the electrical power generation closer to its theoretical maximum.

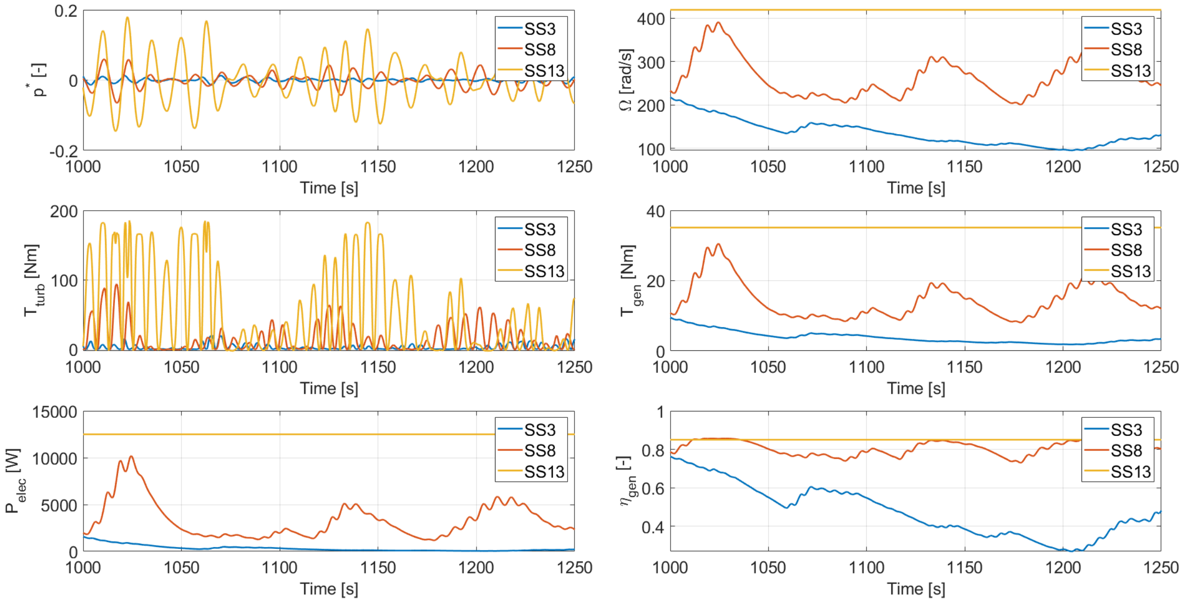


Figure 5.2: Comparison of performance under the baseline control law for sea states 3, 8 and 13.

5.3 Sea State Estimation

The JONSWAP spectrum may be used directly in simulation to generate the excitation force associated with a particular sea state, but in the context of online controller deployment in the Mutriku plant, H_s and T_e must be estimated from time series data that is representative of the sea state. This time series data comes mostly from either the measurement of the motion of a moored buoy that is converted to sea motion using the buoy's hydrodynamic characteristics, which is subject to the accuracy of the buoy model and unexpected buoy motion modes affecting the measurements [99]. Another issue is that high sea buoys may not be representative of the local wave climate around the OWC, which has verified in the Pico OWC [40].

In this implementation, one of the challenges was the control of the Mutriku OWC using time series data for the dimensionless air pressure p^* to approximate the sea spectrum $\hat{S}(\omega)$, which may be collected using sensors installed inside the air chamber.

Since the oscillation of the sea surface is a stochastic process, spectral analysis may be used to characterise the energy contained in its frequency components. Given a power spectral density function $S(\omega)$, wave significant height H_s and energy period T_e are given by equations 5.4 and 5.5, respectively [100].

$$H_s = 4\sqrt{\int_0^\infty S(\omega)d\omega} \tag{5.4}$$

$$T_e = \frac{\int_0^\infty \frac{S(\omega)}{\omega}d\omega}{\int_0^\infty S(\omega)d\omega} \tag{5.5}$$

The power spectral density of signal $y(t)$ sampled at a series of time steps t is given in equation 5.6, which is the Discrete Time Fourier Transform (DTFT) of the signal's autocovariance function $r(k)$. This equation shows that to calculate the power spectral density using the definition complete knowledge of the signal $y(t)$ in the future would be required, since the number of time steps N , or the number of lags used to calculate the autocovariance function tend to infinity [101].

$$S(\omega) = \lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{t=1}^{N} y(t)e^{-i\omega t}\right|^2\right\} = \sum_{k=-\infty}^{\infty} r(k)e^{-i\omega t} \tag{5.6}$$

Multiple methods exist to approximate the power spectral density function of a signal from time series data, but one computationally inexpensive method is the modified periodogram, which is given by equation 5.7. The given equation is already adapted to use a generic sampling frequency $f = \frac{\omega}{2\pi}$ in Hertz and corresponding sampling interval Δ_t instead of the angular frequency ω and includes a window function $w(t)$ that reduces spectral leakage from using a finite time series. To reduce the computational cost of this operation and allow for its performance online, a Fast Fourier Transform (FFT) is used to calculate the periodogram [101]. Estimates for significant height H_s and energy period T_e may then be evaluated from equations 5.4 and 5.5 by numerical integration using the rectangle method.

$$\hat{S}(f) = \frac{\Delta_t}{N}\left|\sum_{t=1}^{N} w(t)y(t)e^{-i2\pi f\Delta_t t}\right|^2 \tag{5.7}$$

This estimation algorithm requires the tuning of a set of parameters to produce accurate results, namely the sampling interval Δ_t of the pressure data, the number of time series points to use N, the window function $w(t)$ and the update time of the periodogram $t_{spectrum}$ (time between periodogram recalculation). Note that while sampling time in a real application will only be limited by the capabilities of the pressure sensor and the PLC, in the simulation model it will influence simulation speed so a balance must be found between accuracy in the spectral density estimate and simulation time.

MATLAB provides a series of predefined window functions [102], which were tested on 10 simulations for each sea state, with a sampling interval Δ_t of 0.05 s, using all 30 minutes of data, in order to reduce estimation errors caused by aliasing due to high sampling time and spectral leakage due the windowing of the periodic signal.

To evaluate the prediction ability of this estimate, a linear correlation analysis is performed between real and estimated significant height (H_s and $\hat{H}_s$) and energy period (T_e and $\hat{T}_e$). Note that while the energy period has similar units in both cases, the estimated significant height will be associated with the oscillation of p^*, which is dimensionless and has a different order of magnitude. Two measures of accuracy for the window were used: the coefficient of determination R^2 and the Mean Squared Error (MSE) of the model. Figure 5.3 shows these two performance parameters for each of the tested window functions in the estimation of H_s and T_e. While using a tapered cosine window leads to a more accurate prediction for the significant height, shown by the higher R^2 and lower MSE, the simpler rectangular window $w(t) = 1$ is the most accurate in the prediction of the energy period, and since this function is less computationally intensive, it was chosen as the window function for the remaining trials.

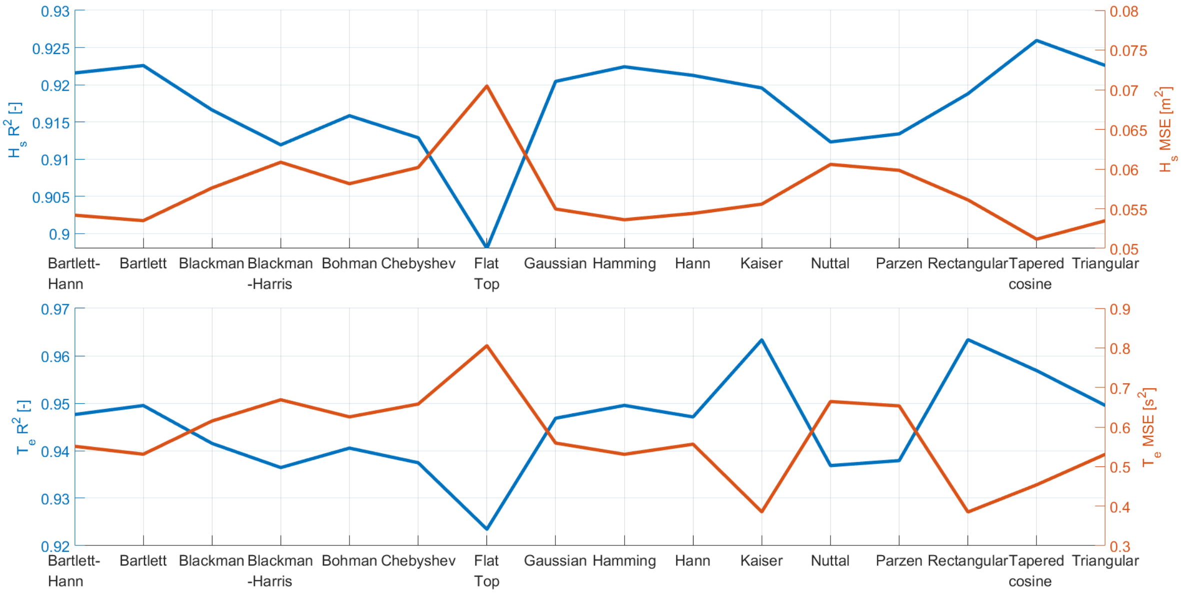


Figure 5.3: MSE and R^2 for the linear regression between actual and estimated significant height H_s and energy period T_e, with different periodogram window functions.

Having chosen the window function, the sample interval Δ_t may be tuned. An indicator for the maximum value that this parameter may take is the Nyquist-Shannon sampling theorem, stating that, to preserve the information in a continuous signal in a discrete domain, the sampling interval must be less than half the period of the highest frequency content in the signal. Table 4.1 shows that the highest frequency sea state has an energy period of 5.5 s, so any sampling time of 2.75 s is enough to describe the signal according to the theorem.

Empirical testing for sea state estimation using sample intervals ranging from 0.05 s to 5 s, shown in figure 5.4 confirms the theorem's results, with the quality of the previously described linear correlation decreasing for sample intervals higher than 2.75 s. Taking into account these results, a sample interval Δ_t of 1 s was chosen, fitting the Nyquist-Shannon theorem's criterion while avoiding high simulation times due to smaller time steps in the differential equation solver of *Simulink*.

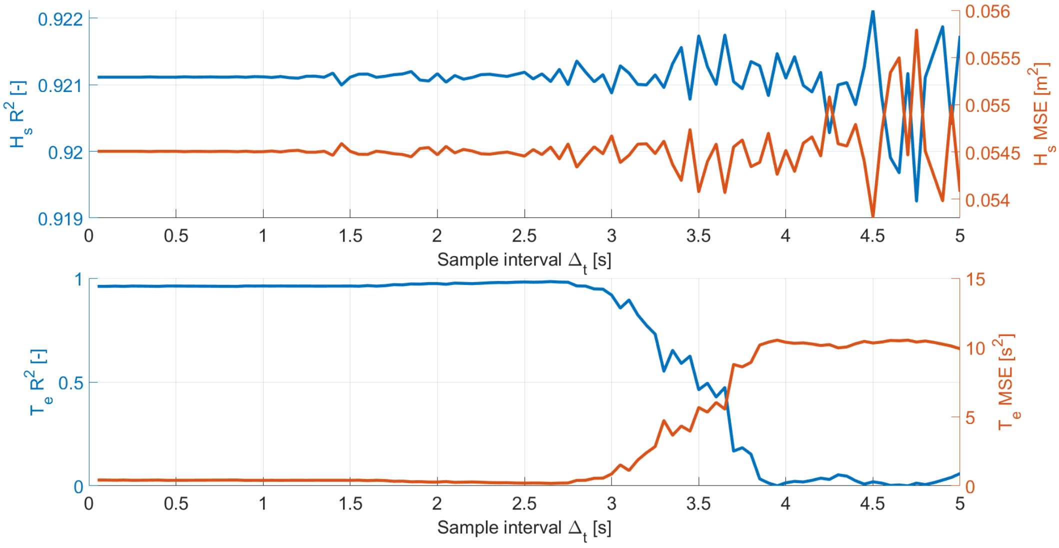


Figure 5.4: MSE and R^2 for the linear regression between actual and estimated significant height H_s and energy period T_e, as a function of sample interval Δ_t.

The size of the window used in the periodogram calculation will also be an important factor in the accuracy of sea state estimation and interval lengths from 1 min to 30 min, Figure 5.5 shows that, according to the previously mentioned parameters, the estimation accuracy suffers a significant increase up to 900 s, with the increase being less pronounced from that point. Once more, when considering the trade off between estimation accuracy and computational intensity, this point, corresponding to the previous 15 min of simulation, is used in the final controller.

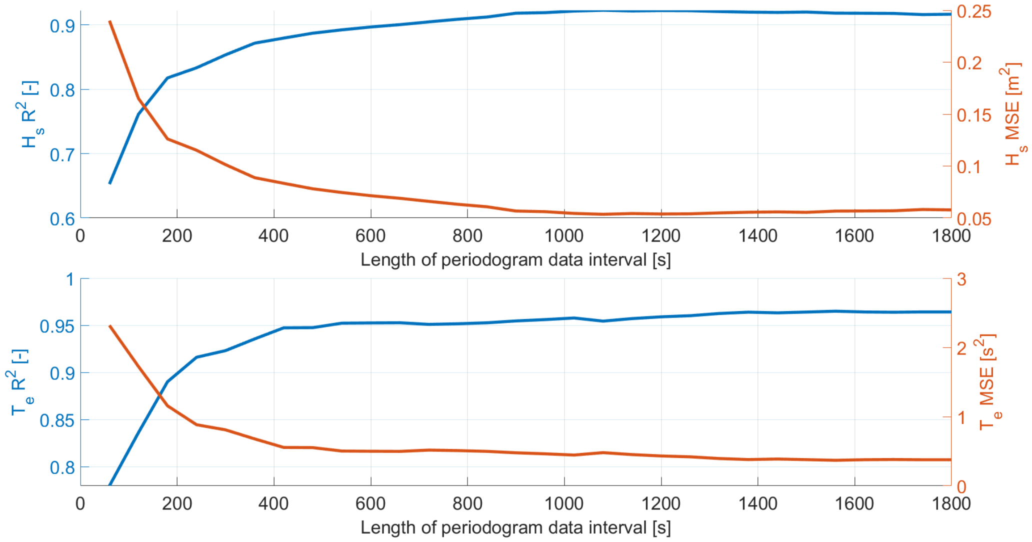


Figure 5.5: MSE and R^2 for the linear regression between actual and estimated significant height H_s and energy period T_e, as a function of interval length used in the periodogram.

Having determined the parameters to use in the periodogram estimation, the resulting algorithm was applied online to a 30 min simulation of the model under each sea state. This simulation was divided into two phases: on the first, pressure data is gathered to provide a first estimate of the periodogram, and on the second, the periodogram estimate is calculated in real time.

Figure 5.6 shows the results of the second phase of this simulation. While the reconstruction of the sea states from pressure data is not perfect, the algorithm correctly sorts the sea state data qualitatively in the correct order according to 4.1. Peaks in the pressure signal may cause a wrong ordering of the sea state parameters, such as when the estimate for the significant height of SS11 temporarily reaches a higher value than the estimate for SS12. The non-linearity of the model used, which includes both the damping effect of the air chamber compression and the frequency dependent attenuation function $\varphi_{Mutriku}$ (figure 4.4) may also be factors that contribute to errors in the estimation of the true sea state. SS13 and SS14 also were estimated to have similar significant heights, which was caused by the activation of the safety valve limiting the amplitude of the pressure signal, leading to an underestimation of H_s.

Another relevant issue are the oscillations in the estimate of the energy period (figure 5.6), which stem from numerical issues in the estimation of the quotient between the two integrals in equation 5.5. For this reason, the spectral density is computed every $t_{spectrum} = 5\,\mathrm{s}$ instead of at the sampling time of the pressure data.

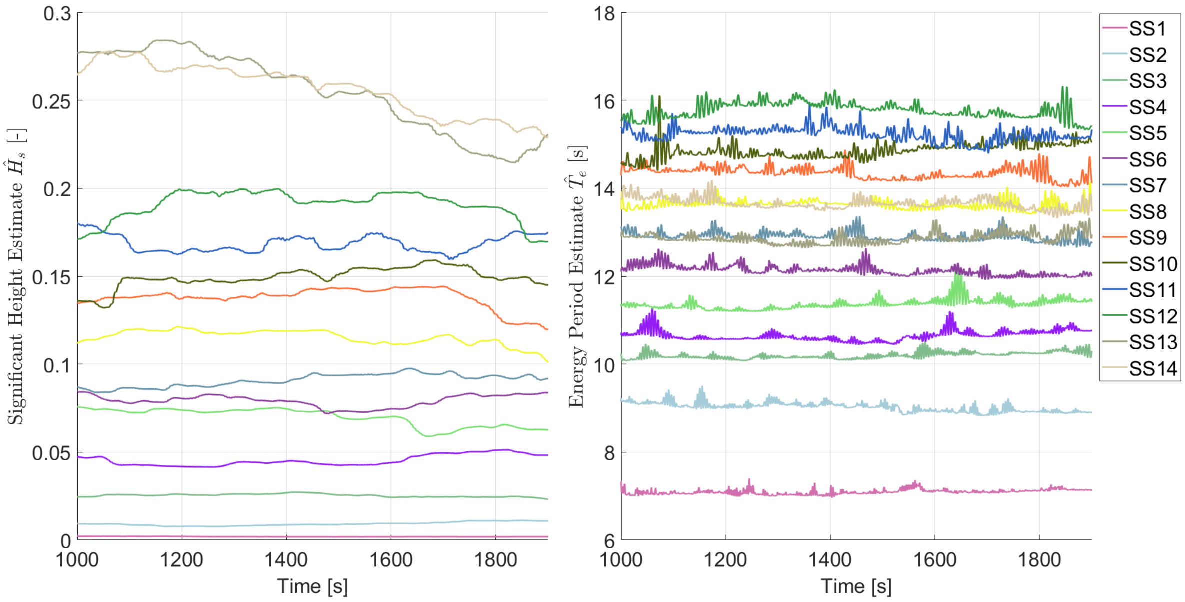


Figure 5.6: Real time estimates for the significant height and energy period for each of the sea states.

Having validated all the parameters using operation data from the simulation, algorithm 4 shows the all the steps of the procedure in pseudocode format, facilitating the implementation in *Simulink*.

Algorithm 4 Sea state Estimation Algorithm

```
Initialise pressure data buffer $p\_buffer$ with size $buffer\_size$.
Define pressure sampling interval $\Delta_t$ and spectrum update time $t_{spectrum}$.
Initialize window function $w(t)$.
while $t < simulation\_end$ do
    if $t \bmod \Delta_t = 0$ then
        Add pressure value $p_t$ to the data buffer $p\_buffer$.
        if $t \bmod t_{spectrum} = 0$ then
            Estimate the pressure signal power spectral density using equation:
                $$\hat{S}(f) = \frac{\Delta_t}{buffer\_size}\left|\sum_{t=1}^{buffer\_size} w(t)p_{buffer}(t)e^{-i2\pi f \Delta_t t}\right|^2$$
            Estimate significant height $\hat{H}_s$ using equation:
                $$H_s = 4\sqrt{\int_0^\infty S(\omega)d\omega}$$
            Estimate energy period $\hat{T}_e$ using equation:
                $$T_e = \frac{\int_0^\infty \frac{S(\omega)}{\omega}d\omega}{\int_0^\infty S(\omega)d\omega}$$
        end if
    end if
end while
```

5.4 Controller Architecture

The chosen DRL methods used in the control of the OWC require a parametric representation using neural networks, with the parameters being the network weights θ. While these parameters are automatically tuned during the training, a set of training hyperparameters must be manually tuned to optimize the learning process. In terms of the network structure, the main hyperparameters are the number of layers, the number of neurons on each layer and the activation function of each neuron. Other hyperparameters are numerical constants or parameters related to the training process, and are, for the most part, specific for each type of algorithm used so, for this reason, will be mentioned separately in the following subsections.

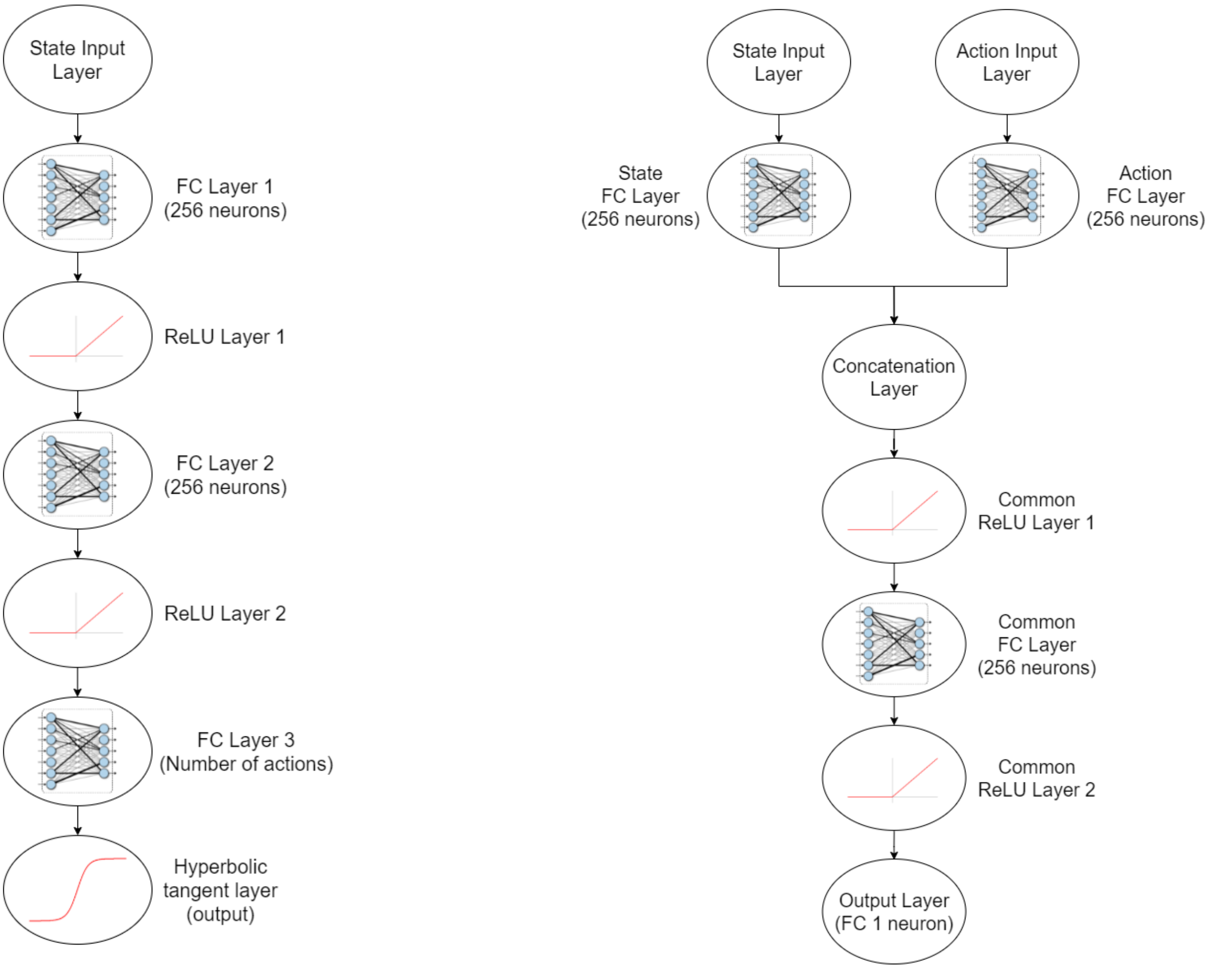
In the absence of theoretical proof for convergence in DRL, an effort was made to follow the scientific community's best practices for the choice of structure and hyperparameters, such as the recommendations from Achiam [71].

Note that in the notation used in *MATLAB*, the multilayer perceptron layers defined in chapter 3 are divided into a layer containing neurons performing the affine part $\mathbf{y} = \mathbf{w}^T\mathbf{x} + b$ of the transformation, called a fully connected (FC) layer, and a part containing the application of the nonlinear activation function f, designated with the name of the activation function in use (such as ReLU layer or Hyperbolic Tangent Layer).

5.4.1 DDPG and TD3 Controller Design

Since TD3 acts as an extension to DDPG that improves training stability, to properly compare these two alternatives it was decided to use the same actor and critic network architectures for both, introducing only a second critic network with similar structure in the TD3 controller.

The chosen structure for the actor (figure 5.7 a)) includes 2 Fully Connected (FC) layers with 256 neurons each, each followed by a layer that applies a ReLU activation function. The deepest layers are an FC layer with as many neurons as the number of control actions followed by an hyperbolic tangent layer that squashes the output to the [-1,1] interval. The control action is then re-scaled to the appropriate range externally to the network before being applied to the plant model. The critic network (figure 5.7 b)) has two parallel input layers, for the state and action inputs separately, followed by two FC layers with 256 neurons each, which are then concatenated into a single, 512-dimensional output on which a ReLU activation function is applied. After another 256 neuron FC layer and ReLU activation function, the output Q-function estimate is given by a single FC neuron, which adds all the outputs of the previous layer.



(a) Actor network structure in the DDPG and TD3 controllers.

(b) Critic network structure in the DDPG and TD3 controllers.

Figure 5.7: DNN architectures used in DDPG and TD3.

Also relevant to the training algorithm are a set of numeric hyperparameters that control the learning process, shown in table 5.2. To increase training stability, the size of the replay buffer and mini-batch were increased from their default implementations, and, to avoid network over-fitting, a gradient threshold was imposed on the gradients used in optimization and an L2 regularisation term $\lambda\frac{1}{2}\mathbf{w}^T\mathbf{w}$ was added to the loss function to penalise large weights, with a tune-able weight λ that may be increase to penalise large DNN weights. The default proposed learning rate of $\alpha_l = 0.01$ was found to converge at an adequate rate for both actor and critic networks in DDPG or TD3.

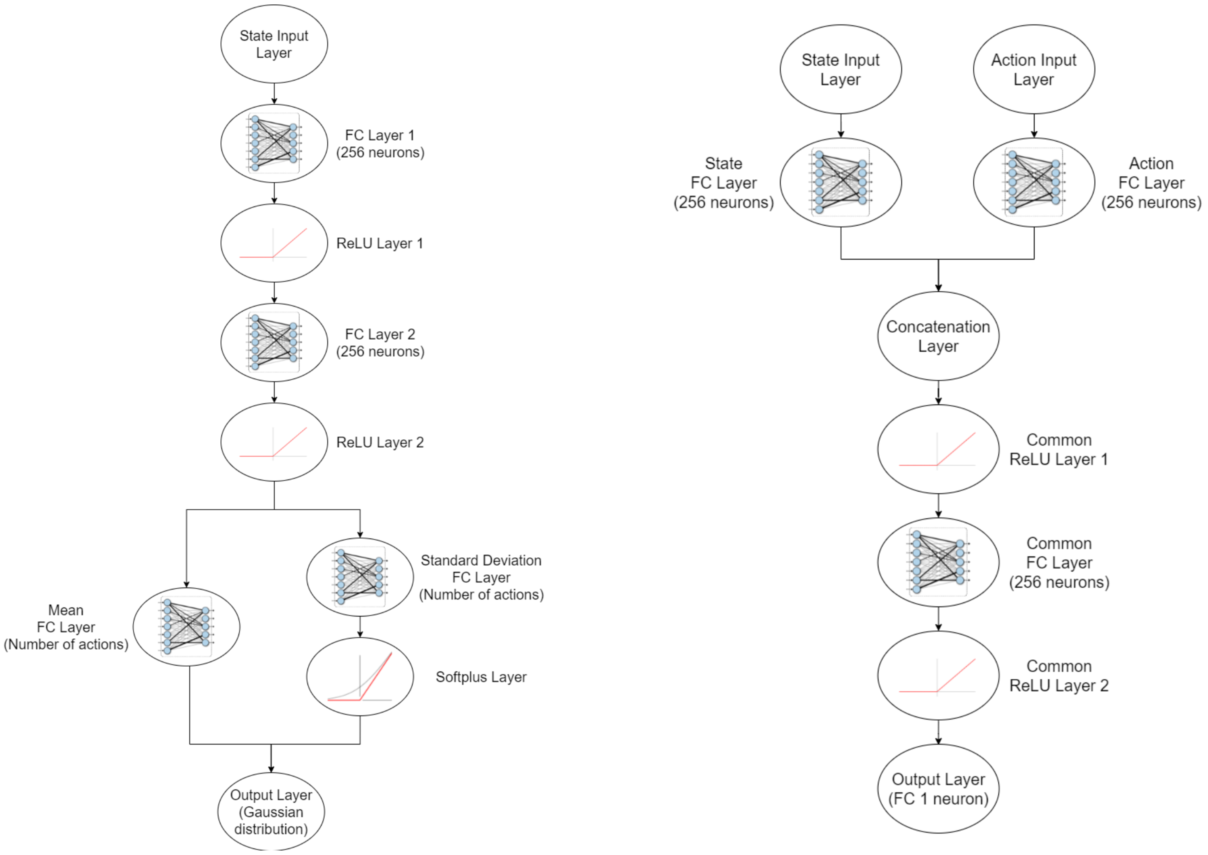
There are three differences in hyperparameters between DDPG and TD3: the exploration noise model, where DDPG uses Ornstein-Uhlenbeck noise with a standard deviation of 0.3 and TD3 uses Gaussian noise with standard deviation of 0.1, the frequency in updating the policy and target networks, done every 2 steps in TD3 and every step in DDPG and the addition of smoothing noise Gaussian to the policy, with a standard deviation of 0.2. Note that these standard deviation values were used because the control actions were all normalised to the [-1,1] range, otherwise they would need to be adjusted to ensure proper exploration.

Table 5.2: DDPG and TD3 algorithm hyperparameters.

Common Hyperparameters	**Value**
Polyak averaging smooth factor ρ	0.001
Actor learning rate	0.01
Critic learning rate	0.01
Gradient threshold	1
Minibatch $U(\mathcal{D})$ size	256
Replay buffer size $\mathcal{D}$	100000
Sample time T_s	2 s
Discount factor	0.99
Optimization algorithm	Adam
L2 Regularization factor λ	0.0001
DDPG specific Hyperparameters	**Value**
Target update Frequency	1 (every step)
Exploration noise model	Ornstein-Uhlenbeck
Exploration noise standard deviation	0.3
TD3 specific Hyperparameters	**Value**
Target update Frequency	2 (every 2 steps)
Policy update Frequency	2 (every 2 steps)
Exploration noise model	Gaussian
Exploration noise standard deviation	0.1
Policy Smoothing Model	Gaussian
Policy Smoothing Standard Deviation	0.2

5.4.2 SAC Controller Design

Unlike the DDPG and TD3 controllers, the SAC controller uses a stochastic actor representation, meaning that while the pair of critic networks may be similar in structure to the one used in the previous two algorithms (see figure 5.8 b)), the actor network must be modified to output a normal distribution. This is done by splitting the network path into two after the first two pairs of 256 neuron FC and ReLU layers, as shown in figure 5.8 a). The branches of the path represent the mean $\mu_{\theta^{\pi}}$ and standard deviation $\sigma_{\theta^{\pi}}$ of the stochastic policy, through an FC layer with as many neurons as the dimensionality of the output actions. The scaled action must then be sampled from this distribution using equation 3.19. The definition of the squashed action means that while the mean may be used directly as the unbounded value coming from the FC layer, the standard deviation must take a smooth positive value, requiring the introduction of a softplus layer to be added before the output.


State Input Layer
FC Layer 1 (256 neurons)
ReLU Layer 1
FC Layer 2 (256 neurons)
ReLU Layer 2
Standard Deviation FC Layer (Number of actions)
Mean FC Layer (Number of actions)
Softplus Layer
Output Layer (Gaussian distribution)
State Input Layer
Action Input Layer
State FC Layer (256 neurons)
Action FC Layer (256 neurons)
Concatenation Layer
Common ReLU Layer 1
Common FC Layer (256 neurons)
Common ReLU Layer 2
Output Layer (FC 1 neuron)

(a) Actor network structure in the SAC controller.

(b) Critic network structure in the SAC controller.

Figure 5.8: DNN architectures used in SAC.

When tuning the numeric hyperparameters, the ones that are common to DDPG were kept with the same values in order to compare the influence of changing algorithms, as displayed in table 5.3. However some parameters are applicable only to SAC, such as the initial entropy weight α, the entropy optimisation learning rate and the target entropy $\bar{H}$, which were kept equal to the original implementation [80]. Furthermore, the choice was made to update the target critics by Polyak averaging at every step, instead of lowering the update frequency, since no training divergence issues were observed that justified the target delay used in TD3.

Table 5.3: SAC algorithm hyperparameters.

Hyperparameter Name	Value
Initial entropy weight α	1
Entropy learning rate	0.003
Target entropy $\bar{H}$	2
Target update Frequency	1 (every step)
Polyak averaging smooth factor ρ	0.001
Minibatch $U(\mathcal{D})$ size	256
Replay buffer size $\mathcal{D}$	100000
Sample time T_s	2 s
Discount factor	0.99
Actor learning rate	0.01
Critic learning rate	0.01
Gradient threshold	1
Optimization algorithm	Adam
L2 Regularisation factor	0.0001

Chapter 6

Results and Discussion

Having defined the architectures for the DDPG, TD3 and SAC controllers, the results from training these controllers are presented with a focus on the trade off between computational resource usage and controller performance. The benefits and drawbacks of applying DRL control compared to the traditional power law are presented along with a comparative analysis between the different types of DRL controller (DDPG, TD3 and SAC).

6.1 Training Process

Using the MDP formulation and training hyperparameters described in the previous chapter, each of the controllers was trained. Since the possible reward from an episode is largely dependent on the sea state, which is sampled randomly, it is inconvenient to set an average reward target as a stopping criterion, so training is stopped at a set number of simulation time steps, with convergence evaluated after training.

In order to expose the agents to multiple different sea states, every episode is initialised by choosing a new sea state and corresponding excitation force F_{exc}, according to equation 4.3. Each episode runs for a total of $30\,\mathrm{min}$ of simulated time in normal operation conditions, but may terminate earlier due to the activation of flag f_1 from equation 5.3, indicating that the controller either caused the turbine to stop or the air chamber pressure to reach an unacceptable value.

The training curve for DDPG and TD3 is shown in figure 6.1. As is common in this type of algorithms, the controller takes a significant number of exploratory initial episodes where the achieved reward is insignificant, after which there is a sudden increase in average reward and, consequently, controller performance. In TD3, this jump occurs around episode number 380 while in DDPG and SAC it happens earlier, near the 300^{th} episode.

Comparing the three controllers, it is clear that the DDPG controller converges to a policy that leads to a lower average reward than TD3 or SAC. This shows the effect of the modifications introduced by these algorithms that aim to avoid local minima and improve convergence speed and stability. The critic output evolution also clearly shows the issue in DDPG of value overestimation, with the value function approximator network outputting a higher average value in DDPG compared with TD3 or SAC on the same number of episodes, even if the true observed average reward is lower. Another significant drawback of DDPG is that, even with an extensive hyperparameter and network structure search, it was found to be prone to over-fitting, where policy output saturates to one of the limits of the action range and to instability, with divergence in the critic output. These phenomena tended to occur at a total number of steps between 1×10^5 and 1.2×10^5 so it was decided to stop training at 1.2×10^5 steps, instead of the 3×10^5 used in the TD3 and SAC controllers.

Even though the controller is learning and performing at an acceptable level, and the average reward tends to a constant value, the critic network still does not fully approximate the episode reward, taking lower values instead. This may be caused by the gradient clipping and L2-regularisation preventing the output from growing to a large value at a quick enough rate, but applying these two techniques was a necessary trade-off to ensure training stability and proper learning convergence of the neural networks.

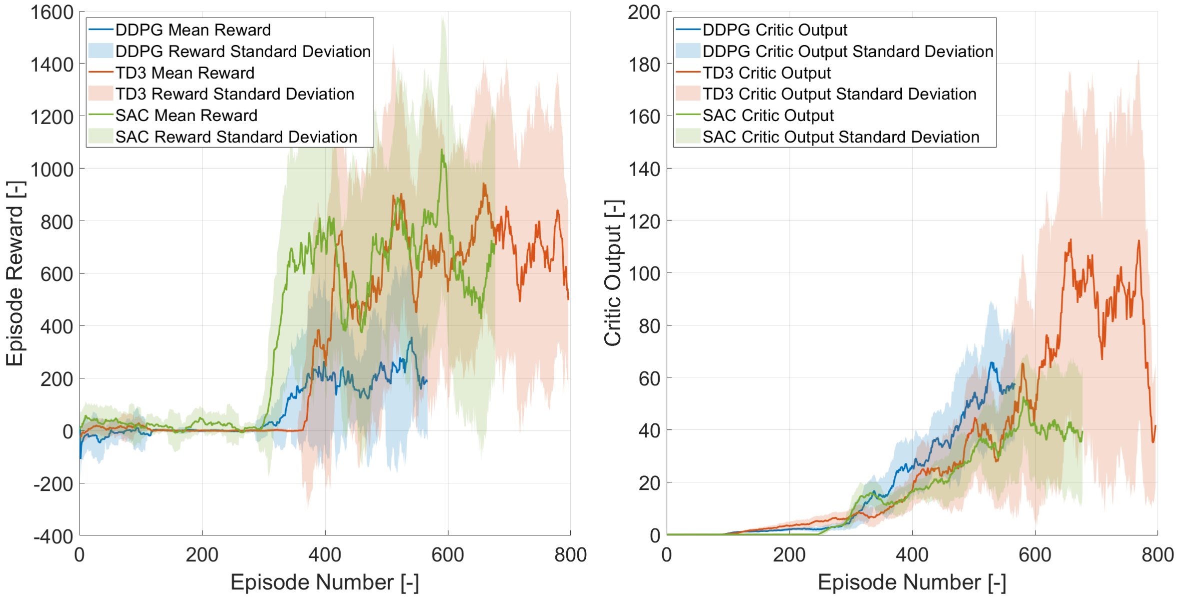


Figure 6.1: DDPG, TD3 and SAC controllers training reward curves.

A relevant factor that will influence the possibility of applying this type of control in real time to a physical system is the training time. It is a necessary requirement that, while training, the available computational resources are enough for *Simulink* and *MATLAB* to simulate the system as fast or even faster than real time. It is also beneficial that the required training time for convergence is as low as possible, since it will lead to less expensive and faster deployment to either a prototype or the real system. The training time, simulated time and ratio between them for each of the trained controllers is shown in table 6.1.

The TD3 and SAC controllers were trained for approximately 6×10^5 s of simulation time (3×10^5 time steps at an agent sample time T_s of 2 s) while the DDPG controller was trained for 2×10^5 s, using an Intel i5 2500k processor and 8 GB of RAM to simulate the environment and a Nvidia GTX 970 graphics card in the optimization process of training the neural networks. The small variations in simulated time are due to the requirement that training ends at the end of a complete episode.

Table 6.1: Training and simulation time for the 3 tested DRL algorithm.

DRL algorithm	Training time		Simulated time		Time ratio
	[s]	[h]	[s]	[h]	
DDPG	25 443	7.06	200 444	55.7	7.87
TD3	76 664	21.3	600 132	166.7	7.82
SAC	117 970	32.8	601 682	167.1	5.10

Simulink automatically adjusts the simulation speed to be as fast as the computing power of the hardware allows, so the results show that, even with a commercial-level computer, the controller would be able to train in real time.

As expected the more complex SAC algorithm is the slowest, simulating the environment 5.1 times faster than real time, followed by TD3 (7.82 times faster) and DDPG (7.87 times faster). DDPG and TD3 show similar ratios between simulated and training time, which is unexpected since TD3 requires the training of an additional critic network when compared to DDPG. Factors that may contribute to TD3 running faster than expected are the delayed policy updates, reducing in half the number of optimization steps required to train the policy, and the fact that DDPG's worse performance causes a higher frequency of early terminations, which require restarting the environment more often.

It was not expected that the DDPG and TD3 controllers had similar ratios between simulated and training time since TD3 requires the training of two critic networks to apply the clipped double-Q trick, but the reduced computation requirements from only training the actor every two steps may be enough to offset this, combined with faster learning of TD3 reducing the number of episodes that end in an early termination, requiring additional restarts of the *Simulink* environment, delaying the training.

Note that the training times in table 6.1 represent an upper limit on the training time in a computer with these specifications, since the simulation of the model also consumes some of the available hardware resources, presenting a further argument in favour of the possibility of implementing this type of control in real time.

6.2 Controller performance

The stated goal of introducing DRL to control an OWC device is the maximisation of power production, meaning that to evaluate the control law learned by the neural networks, the power production must be compared between each of the controllers and against the baseline defined in section 5.2.

Each controller was thus evaluated on 20 different random initialisations of the wave generator using each of the 14 relevant sea states, simulated for 30 min. The results of this evaluation are shown in figure 6.2.

The first conclusion to be drawn is that the DDPG agent is not an adequate solution to this problem, since it fails to outperform the baseline power law on multiple sea states, only showing an improvement on sea states SS3 to SS5, SS9 and the two more energetic states SS13 and SS14. Comparing it to the other two DRL controllers, it only outperforms TD3 on sea state SS6 and SAC on SS3, with increases in energy generation of 6% and 1%, respectively.

TD3 and SAC present similar power generation values for most sea states, but TD3 has a higher mean power generation of the more (SS10 to SS14 and SS8) and less (SS1 to SS4) energetic sea states , while SAC favours electricity production in states with an intermediate significant height (SS5 to SS7 and SS9). These are, however, the most likely states to occur in the Mutriku wave climate.

For the intermediate sea states (SS3 to SS12), the TD3 and SAC controller are able to achieve slightly higher power generation on average than the power law controller, with increases varying from 1% to 16% over the baseline.

The improvement seen in power generation is larger for the most energetic sea states, particularly SS13 and SS14 with improvements of 35.4% and 27.7%, respectively, when using SAC or 35.6% and 31.1% when using TD3.

Low energy sea states also benefit from a significant increase in power generation under SAC and TD3 over the baseline law, which was expected as the power law optimises for turbine efficiency, ignoring generator efficiency which decreases with the load on the generator, which is more likely to be smaller in low energy states. By including the full wave-to-wire model of the plant in the information provided to the controller, the power generation is increased by 112%, 136% and 39% in sea states SS1, SS2 and SS3, respectively, under the action of the TD3 agent, while the SAC agent shows a smaller benefit in these states, with increases of 100%, 43.5% and 16.8% over the baseline.

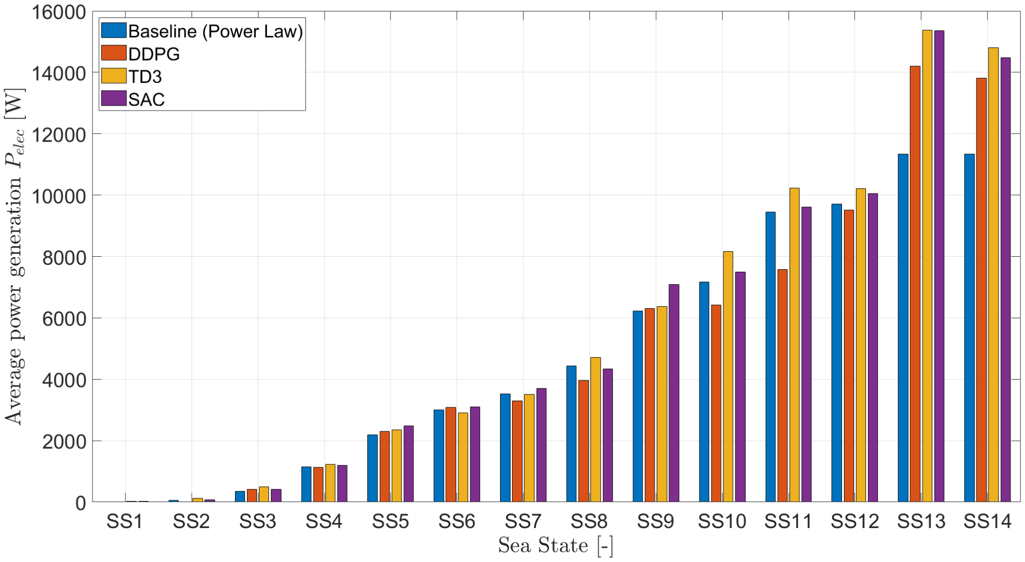


Figure 6.2: Electrical power generation under each control law.

Assuming the probability distribution of the sea states follows table 4.1 over a year, the expected value of the electrical power generation may be calculated by multiplying the probability of occurrence of each sea state p_o with the respective power generation under each control law. Using the average Iberian Electricity Market (MIBEL) price in September 2021 as a reference (160.77 €/MWh) [103], the expected additional earnings from applying the developed control law are shown in table 6.2. These values assume that the control law will be applied on all 14 Wells turbines currently operational at

Mutriku, and that the interaction between the effects of each turbine and OWC's dynamics will have a negligible effect.

Table 6.2 verifies the previous statement that the DDPG does not represent a viable alternative for this problem, since even when comparing with the simple baseline control law, it is not able to generate more power, leading to an expected revenue loss for the plant operator of 1273€ per year. TD3 and SAC, however, present a clear advantage over the baseline law, yielding profits of 2227€ and 2285€ per year, respectively. The SAC expected power generation is able to surpass TD3 over a full operation year due to its better performance on the most likely sea states to occur (SS5 to SS7).

Table 6.2: Yearly energy generation and revenue under each control scheme.

Control algorithm	Energy Generation [MWh/yr]	Total Revenue [€/yr]	Additional Energy [MWh/yr]	Additional Revenue [€/yr]
Power Law	232.8	37 423	0	0
DDPG	224.9	36 150	-7.9	-1 273
TD3	246.6	39 650	13.8	2 227
SAC	247.0	39 708	14.2	2 285

Figure 6.3 shows a set of relevant OWC variables when applying the trained SAC controller on representative sea states SS3, SS8 and SS13.

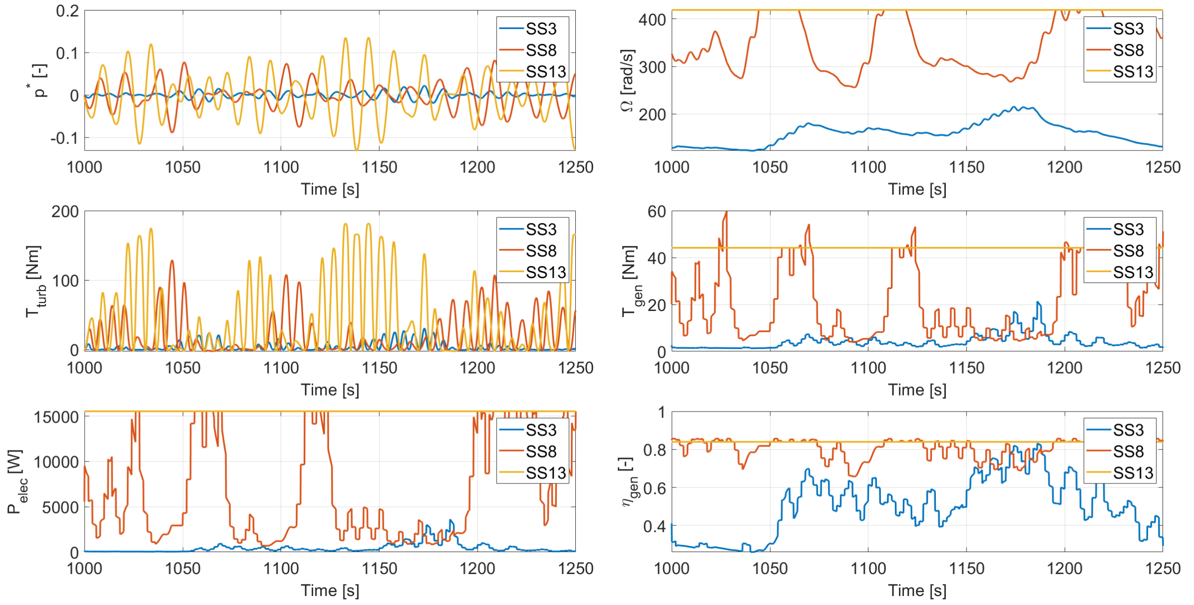


Figure 6.3: Plant performance using the SAC controller on sea states SS3, SS8 and SS13.

Compared with the power law controller (figure 5.2), the most notable change is the presence of sharper peaks in the generator torque and correspondent generated electrical power for sea states 3 and 8. This behaviour may be explained in part by considering the optimality conditions described in the frequency domain control (see section 2.1). The optimal phase and magnitude control law is well-approximated by an on-off type of control law, such as latching control, so the DRL controller learns a similar type of policy to maximise power extraction.

This behaviour is even further encouraged when taking into account the generator conversion efficiency curve shown in figure 4.8, which shows that a higher generator load will lead to higher efficiency. Unlike the power control law that only took turbine efficiency into consideration, the agent took the generator efficiency into account, which in sea states where the available pneumatic power is low, means temporarily lowering the electromagnetic torque to allow rotation velocity to increase, to then use the kinetic energy stored in the turbine-generator rotor at a higher load factor by rapidly increasing the applied torque.

A similar behaviour occurs when using TD3 to control the plant as shown in figure 6.4, where the improvement is seen mainly in SS8, where the controller manages to keep the rotation velocity mostly under the reference value of 4000 rpm ($419\,\mathrm{rad\,s^{-1}}$), unlike the SAC controller where the HSSV had to be actuated in regular intervals.

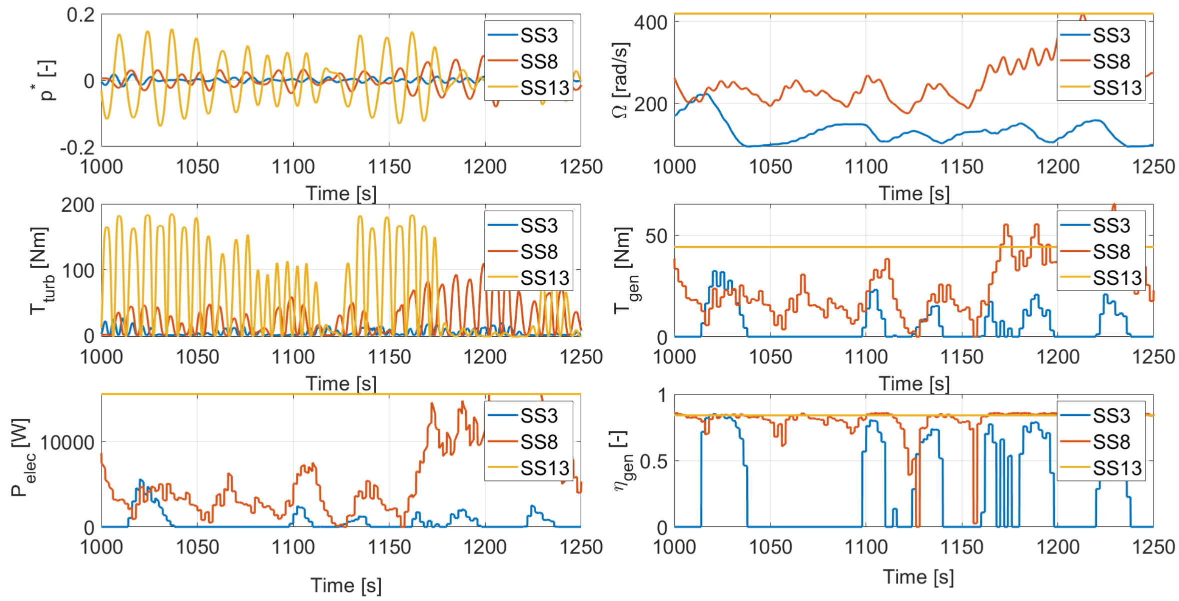


Figure 6.4: Plant performance using the TD3 controller on sea states SS3, SS8 and SS13.

In both TD3 and SAC, the more energetic SS13 required a constant intervention by the HSSV, which artificially limited the rotation velocity to 4000 rpm. The simplification introduced by this limitation to the system's dynamics may be a fault in the model, since the real system would require shutting down power production in those states, compromising the accuracy of results for average power production in the more energetic sea states.

Finally, in order to understand the reason for the lower performance under DDPG, the same testing was done using this type of agent, with results shown in figure 6.5. The on-off behaviour of the generator torque is also observed in this type of agent, but the learned control action is slower to respond to changes, and a significant period of time is spent with no torque applied to the system, contributing to the lower average value.

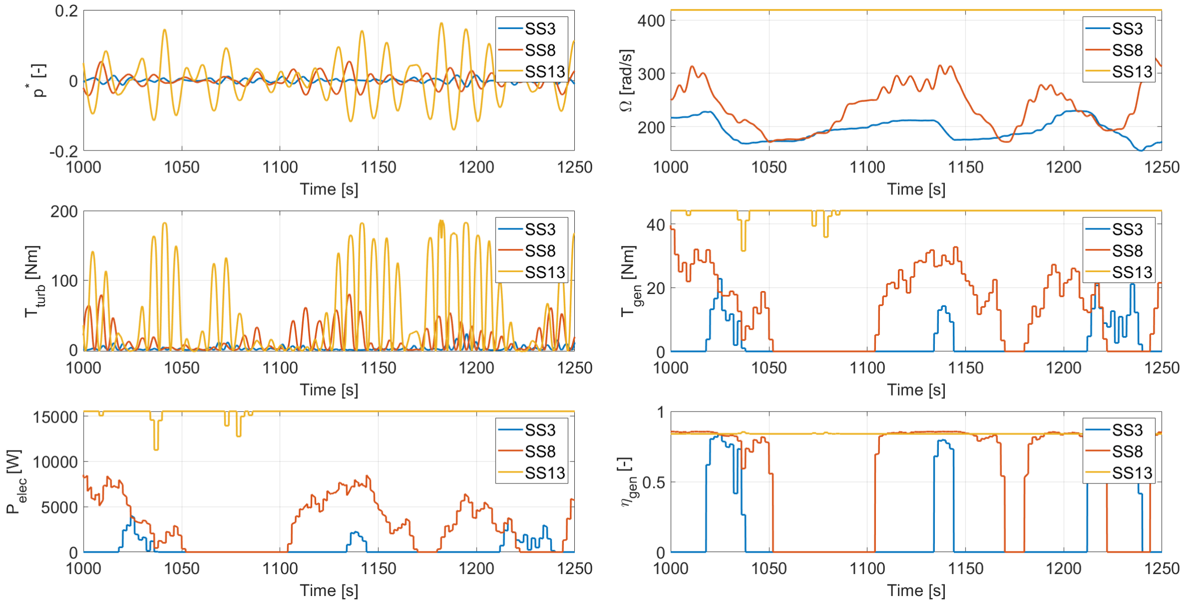


Figure 6.5: Plant performance using the DDPG controller on sea states SS3, SS8 and SS13.

An unexpected result of controller training is the fact that in all three controllers, the valve aperture k_v tended to stay closed as the controller converged to its final behaviour, as shown in figure 6.6.

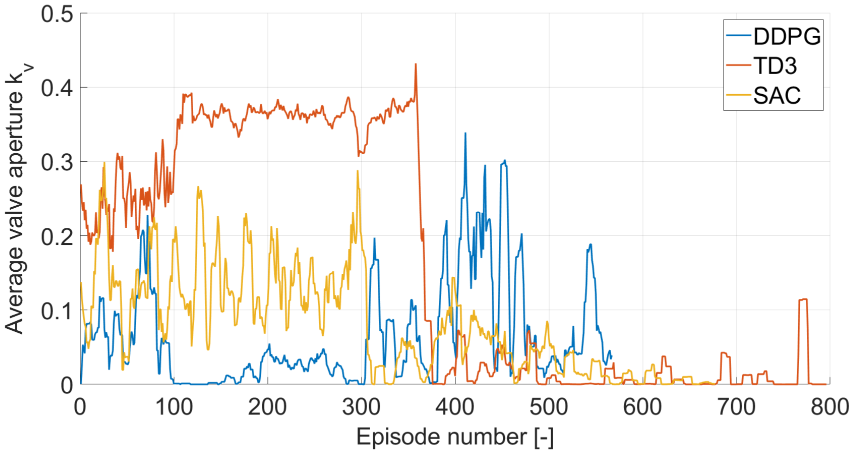


Figure 6.6: Average valve aperture over each training episode.

The reason for this learned behaviour is unclear, since this phenomenon occurred regardless of the magnitude of the penalty applied for valve aperture or for excessive rotation velocity, the area A_v of the valve used (values of A_v ranging from $0.05\,\mathrm{m}^2$ to $1\,\mathrm{m}^2$ were tested) or the controller hyperparameters in use. Analysing figure 6.1, it is also possible to verify that the largest jump in expected reward occurs simultaneously to the drop in usage of the valve in figure 6.6.

Still, by stopping the training of the SAC algorithm early, it is possible to verify the system dynamics when opening the valve, shown in figure 6.7, for a training episode using SS14.

The observed behaviour presents another alternative explanation for the behaviour of avoiding using the valve. Even though this episode occurs under the most energetic sea state, the reduction in dimen-

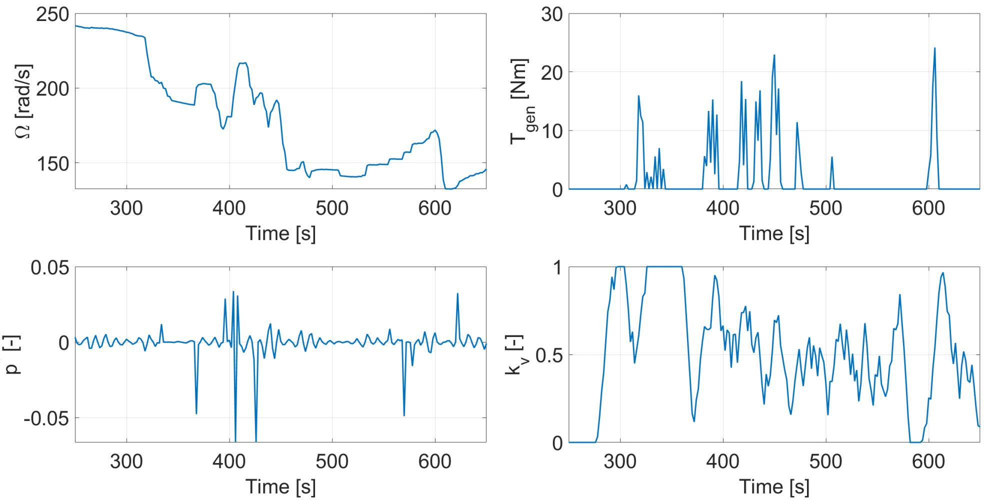


Figure 6.7: Example of training iteration with a partially open valve on SS14.

sionless pressure caused by the valve opening may lead the controller to wrongly identify the sea state as a low energy state. This makes the generator torque behave similarly to what was observed for SS3 in figure 6.4. The lower power generation, and corresponding reward, the agent obtains from applying the wrong control law may be causing it to avoid using the valve altogether. This shows that to apply relief valve control, sea state estimation from outside the air chamber may be a necessity in order to provide a reference for the controller that is independent of its actions on the system.

Chapter 7

Conclusions

In this chapter the main conclusions taken from the work developed throughout this thesis are presented, demonstrating the possibility of applying modern DRL techniques to the control of OWC devices. In addition, a set of proposals for future development of this type of controller, as well as for future applications of DRL to the control of WEC devices.

7.1 Achievements

In this thesis, three different architectures for DRL controllers were presented and applied to the problem of maximising electrical power generation in the Mutriku OWC, by interaction with a simulation environment developed and implemented in *Simulink* and *MATLAB*.

When compared to MPC, the main modern control technique that could be applied to this type of problem, DRL has the advantage of shifting the main computational effort to the training process, that may occur offline, and in simulation, before the deployment of the controller to the testing phase in a prototype. However, this does not exclude the possibility of performing online training as well, which allows the controller to adapt to changing system dynamics, either due to turbine and generator wear, structural changes in the chamber or change in the characteristic local sea states, an increasingly important issue with rising sea levels and climate change.

The chosen DRL algorithms also have the advantage of being model-free control schemes, meaning they do not require a complete model of the plant to function, relying instead on a set of numerical observations describing the system, making it a useful approach when facing systems with high modelling uncertainty, as is the case of the Mutriku OWC. Being model-free, the algorithms used in this thesis should be able to be extended to any OWC system other than Mutriku, with only minimal change in training hyperparameters and reward function weights.

Analysing the training results, the DDPG controller was shown to have a poor performance, due to training instability issues and convergence to sub-optimal local minima, which resulted in an expected yearly power production at the Mutriku plant that is 7.9 MWh (3%) lower than the baseline power law. In contrast, TD3 and SAC have both been shown to be promising alternatives to controlling the Mutriku power plant, with the trained controllers leading do an increase in expected yearly electric power production of 5.9% (13.8 MWh) and 6.0% (14.2 MWh) over the baseline.

The values obtained for the increase in power production are valid for the model used in simulation and may not reflect the real increase in production in an OWC prototype, for the most part due to the simplification introduced in modelling the HSSV.

It was also shown that all three agents that were tested tend to avoid opening the relief valve as they converge. This phenomenon may be caused by the opening of the valve causing a drop in pressure inside the air chamber, which will lead the controller to behave similarly to when it is faced with a low energy sea state. Additional information to add to the state vector is required to study the implementation of this type of valve using a DRL architecture.

The improvement in power generation compared to the baseline power law, combined with adaptability to changing dynamics and low computational requirements of DRL after training, shows that Deep Reinforcement Learning, specifically the TD3 and SAC frameworks, represent a viable alternative to the control strategies that are currently implemented in Mutriku and other OWC installations.

7.2 Future Work

Future research on the application of Deep Reinforcement Learning to the control of OWC devices should focus on testing these types of algorithm on the control of a physical prototype, and, in the eventual success of these tests, the transition to a testing phase in a real installation like the Mutriku OWC or a similar one. Special care should be taken in the training process, since it should either be performed on the physical prototype, which may lead to unsafe operation in the exploration of new actions, or in simulation, where the controller is pre-trained on a simulation model and then transferred to the real system either to be used directly or for further training first, which may reduce the effect of using a model instead of the real system. The first alternative requires a lower level controller that prevents the reinforcement learning algorithm from performing actions that compromise the integrity of the system, while the second one is dependent on the accuracy of the model and the quality of the state measurements. One possible route to solve this issue is transfer learning [68], where a DNN is pre-trained to perform a task and then retrained on a different but related task by removing some of the upper layers and retraining them for the new task, using fewer iterations than in the first task. This strategy presents the possibility of training the controllers on a simulation for a longer time and then retraining on the prototype, improving sample efficiency and reducing the training time needed, as well as reducing the amount of typically unsafe initial exploration actions.

To improve controller performance from what was developed in this thesis, a possible modification is the introduction of additional data measurements to the observation vector. This could include measurements from high-seas buoys or satellites or including prediction data for the chamber pressure or the excitation force. These measurements have the drawback of requiring additional equipment, but they may lead to an improvement in controller performance and subsequent power generation that offsets the implementation cost.

It would also be beneficial for controller robustness to use real pressure data from the Mutriku site as an input to the simulation instead of the wave generator using the modified JONSWAP spectrum. This approach has the drawback of not taking into account the effect of the turbine operation on the chamber pressure, so further studies should be performed to verify if this is a valid approximation.

Finally, an alternate approach to implement the control using a relief valve is taking the control architecture developed in this thesis, which acts mainly on the PTO torque, and use it in a second simulation environment to train a second DRL controller that controls exclusively the relief valve.

Bibliography

[1] A. Díaz, G. A. Marrero, L. A. Puch, and J. Rodríguez. Economic growth, energy intensity and the energy mix. *Energy Economics*, 81:1056–1077, 2019. doi: 10.1016/j.eneco.2019.05.022.

[2] International Energy Agency (IEA). *Electricity Information Overview*. International Energy Agency, 2020.

[3] IRENA. *Innovation Outlook: Ocean Energy Technologies*. International Renewable Energy Agency, Abu Dhabi, 2020.

[4] N. Khan, A. Kalair, N. Abas, and A. Haider. Review of ocean tidal, wave and thermal energy technologies. *Renewable and Sustainable Energy Reviews*, 72:590–604, 2017. doi: 10.1016/j.rser.2017.01.079.

[5] M. Melikoglu. Current status and future of ocean energy sources: A global review. *Ocean Engineering*, 148:563–573, 2018. doi: 10.1016/j.oceaneng.2017.11.045.

[6] IRENA. *Renewable capacity statistics 2019*. International Renewable Energy Agency (IRENA), Abu Dhabi, 2019.

[7] D. Ross. *Power from the Waves*. Oxford University Press, 1995.

[8] A. F. d. O. Falcão. Wave energy utilization: A review of the technologies. *Renewable and Sustainable Energy Reviews*, 14(3):899–918, Apr. 2010. doi: 10.1016/j.rser.2009.11.003.

[9] S. H. Salter. Wave power. *Nature*, 249(5459):720–724, June 1974. doi: 10.1038/249720a0.

[10] K. Budar and J. Falnes. A resonant point absorber of ocean-wave power. *Nature*, 256(5517): 478–479, Aug. 1975. doi: 10.1038/256478a0.

[11] W. Sheng. Wave energy conversion and hydrodynamics modelling technologies: A review. *Renewable and Sustainable Energy Reviews*, 109:482–498, Mar. 2019. doi: 10.1016/j.rser.2019.04.030.

[12] A. F. O. Falcão and J. C. C. Henriques. Oscillating-water-column wave energy converters and air turbines: A review. *Renewable Energy*, 85:1391–1424, Jan. 2016. doi: 10.1016/j.renene.2015.07.086.

[13] C. Perez-Collazo, R. Pemberton, D. Greaves, and G. Iglesias. Monopile-mounted wave energy converter for a hybrid wind-wave system. *Energy Conversion and Management*, 199(111971), June 2019. doi: 10.1016/j.enconman.2019.111971.

[14] H. P. Nguyen, C. M. Wang, Z. Y. Tay, and V. H. Luong. Wave energy converter and large floating platform integration: A review. *Ocean Engineering*, 213(107768), May 2020. doi: 10.1016/j.oceaneng.2020.107768.

[15] K. Gunn and C. Stock-Williams. Quantifying the global wave power resource. *Renewable Energy*, 44:296–304, 2012. doi: 10.1016/j.renene.2012.01.101.

[16] European Commission. *SET Plan - Declaration of Intent on Strategic Targets in the context of an Initiative for Global Leadership in Ocean Energy*. European Commission, Brussels, Sept. 2016.

[17] J. L. Villate, P. Ruiz-Minguela, J. Berque, L. Pirttimaa, D. Cagney, C. Cochrane, and H. Jeffrey. Strategic research and innovation agenda for ocean energy. Technical report, ETIPOCEAN, Brussels, May 2020.

[18] E. Anderlini, S. Husain, G. G. Parker, M. Abusara, and G. Thomas. Towards Real-Time Reinforcement Learning Control of a Wave Energy Converter. *Journal of Marine Science and Engineering*, 8(11), Oct. 2020. doi: 10.3390/jmse8110845.

[19] K. Freeman, M. Dai, and R. Sutton. Control strategies for oscillating water column wave energy converters. *Underwater Technology*, 32(1):3–13, Mar. 2014. doi: 10.3723/ut.32.003.

[20] J. Falnes. Optimum control of oscillation of wave-energy converters. *International Journal of Offshore and Polar Engineering*, 12(2):147–155, 2002.

[21] J. V. Ringwood, G. Bacelli, and F. Fusco. Control, forecasting and optimisation for wave energy conversion. *IFAC Proceedings Volumes*, 47(3):7678–7689, 2014. doi: 10.3182/20140824-6-ZA-1003.00517.

[22] E. Ozkop and I. H. Altas. Control, power and electrical components in wave energy conversion systems: A review of the technologies. *Renewable and Sustainable Energy Reviews*, 67:106–115, 2017. doi: 10.1016/j.rser.2016.09.012.

[23] J. Hals, J. Falnes, and T. Moan. A Comparison of Selected Strategies for Adaptive Control of Wave Energy Converters. *Journal of Offshore Mechanics and Arctic Engineering*, 133(3), Aug. 2011. doi: 10.1115/1.4002735.

[24] Y. Hong, R. Waters, C. Boström, M. Eriksson, J. Engström, and M. Leijon. Review on electrical control strategies for wave energy converting systems. *Renewable and Sustainable Energy Reviews*, 31:329–342, Mar. 2014. doi: 10.1016/j.rser.2013.11.053.

[25] A. F. Falcão, J. C. Henriques, L. M. Gato, and R. P. Gomes. Air turbine choice and optimization for floating oscillating-water-column wave energy converter. *Ocean Engineering*, 75:148–156, 2014. doi: 10.1016/j.oceaneng.2013.10.019.

[26] G. Nunes, D. Valério, P. Beirão, and J. Sá da Costa. Modelling and control of a wave energy converter. *Renewable Energy*, 36(7):1913–1921, 2011. doi: 10.1016/j.renene.2010.12.018.

[27] J. C. C. Henriques, A. F. O. Falcão, R. P. F. Gomes, and L. M. C. Gato. Latching Control of an Oscillating Water Column Spar-Buoy Wave Energy Converter in Regular Waves. *Journal of Offshore Mechanics and Arctic Engineering*, 135(2):1–9, 2013. doi: 10.1115/1.4007595.

[28] J. Henriques, L. Gato, A. Falcão, E. Robles, and F.-X. Faÿ. Latching control of a floating oscillating-water-column wave energy converter. *Renewable Energy*, 90:229–241, 2016. doi: 10.1016/j.renene.2015.12.065.

[29] M. Lopes, J. Hals, R. Gomes, T. Moan, L. Gato, and A. O. Falcão. Experimental and numerical investigation of non-predictive phase-control strategies for a point-absorbing wave energy converter. *Ocean Engineering*, 36(5):386–402, Apr. 2009. doi: 10.1016/j.oceaneng.2009.01.015.

[30] F. Faÿ, M. Marcos, and E. Robles. Novel Predictive Latching Control for an Oscillating Water Column Buoy. In *Proceedings of the 12th European Wave and Tidal Energy Conference*, Cork, Ireland, 2017.

[31] P. A. P. Justino and A. F. d. O. Falcão. Rotational Speed Control of an OWC Wave Power Plant. *Journal of Offshore Mechanics and Arctic Engineering*, 121(2):65–70, May 1999. doi: 10.1115/1.2830079.

[32] A. O. Falcão. Control of an oscillating-water-column wave power plant for maximum energy production. *Applied Ocean Research*, 24(2):73–82, Apr. 2002. doi: 10.1016/S0141-1187(02)00021-4.

[33] J. Henriques, R. Gomes, L. Gato, A. Falcão, E. Robles, and S. Ceballos. Testing and control of a power take-off system for an oscillating-water-column wave energy converter. *Renewable Energy*, 85:714–724, Jan. 2016. doi: 10.1016/j.renene.2015.07.015.

[34] J. Henriques, J. Portillo, W. Sheng, L. Gato, and A. Falcão. Dynamics and control of air turbines in oscillating-water-column wave energy converters: Analyses and case study. *Renewable and Sustainable Energy Reviews*, 112:571–589, Sept. 2019. doi: 10.1016/j.rser.2019.05.010.

[35] M. Alberdi, M. Amundarain, F. J. Maseda, and O. Barambones. Stalling behaviour improvement by appropriately choosing the rotor resistance value in wave power generation plants. In *IEEE International Conference on Clean Electrical Power*, pages 64–67, Capri,Italy, 2009. IEEE. doi: 10.1109/ICCEP.2009.5212082.

[36] M. Amundarain, M. Alberdi, A. J. Garrido, I. Garrido, and J. Maseda. Wave energy plants: Control strategies for avoiding the stalling behaviour in the Wells turbine. *Renewable Energy*, 35(12):2639–2648, 2010. doi: 10.1016/j.renene.2010.04.009.

[37] A. O. Falcão and P. Justino. OWC wave energy devices with air flow control. *Ocean Engineering*, 26(12):1275–1295, Dec. 1999. doi: 10.1016/S0029-8018(98)00075-4.

[38] A. F. d. O. Falcão, L. C. Vieira, P. A. P. Justino, and J. M. C. S. Andre´. By-Pass Air-Valve Control of an OWC Wave Power Plant. *Journal of Offshore Mechanics and Arctic Engineering*, 125(3): 205–210, Aug. 2003. doi: 10.1115/1.1576815.

[39] K. Monk, D. Conley, M. Lopes, and Q. Zou. Pneumatic Power Regulation by Wave Forecasting and Real-Time Relief Valve Control for an OWC. In *Proceedings of the 10th European Wave and Tidal Energy Conference (EWTEC 2013)*, Aalborg, Denmark, Sept. 2013.

[40] K. Monk, D. Conley, V. Winands, M. Lopes, Q. Zou, and D. Greaves. Simulations and Field Tests of Pneumatic Power Regulation by Valve Control Using Short-term Forecasting at the Pico OWC. In *Proceedings of the 11th European Wave and Tidal Energy Conference (EWTEC 2015)*, Nantes, France, Sept. 2015.

[41] A. Scialò, J. C. Henriques, G. Malara, A. F. Falcão, L. M. Gato, and F. Arena. Power take-off selection for a fixed U-OWC wave power plant in the Mediterranean Sea: The case of Roccella Jonica. *Energy*, 215:119085, Jan. 2021. doi: 10.1016/j.energy.2020.119085.

[42] A. T. Perera and P. Kamalaruban. Applications of reinforcement learning in energy systems. *Renewable and Sustainable Energy Reviews*, 137(October 2020):110618, 2021. doi: 10.1016/j.rser.2020.110618.

[43] M. Glavic. (Deep) Reinforcement learning for electric power system control and related problems: A short review and perspectives. *Annual Reviews in Control*, 48:22–35, 2019. doi: 10.1016/j.arcontrol.2019.09.008.

[44] L. Cheng and T. Yu. A new generation of AI: A review and perspective on machine learning technologies applied to smart energy and electric power systems. *International Journal of Energy Research*, 43(6):1928–1973, 2019. doi: 10.1002/er.4333.

[45] M. P. Fernandes, S. M. Vieira, J. C. C. Henriques, D. Valério, and L. M. C. Gato. Short-term prediction in an oscillating water column using artificial neural networks. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2018. doi: 10.1109/IJCNN.2018.8489571.

[46] W. Sheng and A. Lewis. Short-term prediction of an artificial neural network in an oscillating water Column. *International Journal of Offshore and Polar Engineering*, 21(4):248–255, 2011.

[47] F. Fusco and J. V. Ringwood. Short-term wave forecasting with AR models in real-time optimal control of wave energy converters. *IEEE International Symposium on Industrial Electronics*, 1(2): 2475–2480, 2010. doi: 10.1109/ISIE.2010.5637714.

[48] M. Amundarain, M. Alberdi, A. J. Garrido, and I. Garrido. Neural rotational speed control for wave energy converters. *International Journal of Control*, 84(2):293–309, 2011. doi: 10.1080/00207179.2010.551141.

[49] D. Valério, M. J. Mendes, P. Beirão, and J. Sá da Costa. Identification and control of the AWS using neural network models. *Applied Ocean Research*, 30(3):178–188, 2008. doi: 10.1016/j.apor.2008.11.002.

[50] E. Anderlini, D. I. Forehand, E. Bannon, and M. Abusara. Reactive control of a wave energy converter using artificial neural networks. *International Journal of Marine Energy*, 19:207–220, 2017. doi: 10.1016/j.ijome.2017.08.001.

[51] E. Anderlini, D. I. M. Forehand, P. Stansell, Q. Xiao, and M. Abusara. Control of a Point Absorber Using Reinforcement Learning. *IEEE Transactions on Sustainable Energy*, 7(4):1681–1690, Oct. 2016. doi: 10.1109/TSTE.2016.2568754.

[52] E. Anderlini, D. I. Forehand, E. Bannon, and M. Abusara. Constraints implementation in the application of reinforcement learning to the reactive control of a point absorber. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 10, pages 1–10, Trondheim, Norway, June 2017. doi: 10.1115/OMAE2017-61294.

[53] E. Anderlini, D. I. Forehand, E. Bannon, and M. Abusara. Control of a Realistic Wave Energy Converter Model Using Least-Squares Policy Iteration. *IEEE Transactions on Sustainable Energy*, 8(4):1618–1628, 2017. doi: 10.1109/TSTE.2017.2696060.

[54] E. Anderlini, D. I. Forehand, E. Bannon, Q. Xiao, and M. Abusara. Reactive control of a two-body point absorber using reinforcement learning. *Ocean Engineering*, 148:650–658, June 2018. doi: 10.1016/j.oceaneng.2017.08.017.

[55] L. Bruzzone, P. Fanghella, and G. Berselli. Reinforcement Learning control of an onshore oscillating arm Wave Energy Converter. *Ocean Engineering*, 206:107346, Apr. 2020. doi: 10.1016/j.oceaneng.2020.107346.

[56] L. G. Zadeh, D. Glennon, and T. K. Brekken. Non-Linear Control Strategy for a Two-Body Point Absorber Wave Energy Converter Using Q Actor-Critic Learning. In *IEEE Conference on Technologies for Sustainability, SusTech*, 2020. doi: 10.1109/SusTech47890.2020.9150511.

[57] F.-X. Faÿ, J. Kelly, J. Henriques, A. Pujana, M. Abusara, M. Mueller, I. Touzon, and P. Ruiz-Minguela. Numerical Simulation of Control Strategies at Mutriku Wave Power Plant. In *Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering: Ocean Renewable Energy*, volume 10, pages 131–153. ASME, 2018. doi: 10.1115/OMAE2018-78011.

[58] F. X. Faÿ, J. C. Henriques, J. Kelly, M. Mueller, M. Abusara, W. Sheng, and M. Marcos. Comparative assessment of control strategies for the biradial turbine in the Mutriku OWC plant. *Renewable Energy*, 146:2766–2784, 2020. doi: 10.1016/j.renene.2019.08.074.

[59] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[60] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachussets, 2^{nd} edition, 2018.

[61] G. A. Rummery and M. Niranjan. On-Line Q-Learning Using Connectionist Systems. Technical Report 166, Cambridge University Engineering Department, Sept. 1994.

[62] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, May 1989.

[63] R. Nian, J. Liu, and B. Huang. A review On reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139:106886, Aug. 2020. ISSN 00981354. doi: 10.1016/j.compchemeng.2020.106886.

[64] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko. Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control*, 46:8–28, 2018. doi: 10.1016/j.arcontrol.2018.09.005.

[65] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, Jan. 1993. doi: 10.1016/S0893-6080(05)80131-5.

[66] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2847–2854, 06–11 Aug 2017.

[67] H. Mohamed, A. Negm, M. Zahran, and O. C. Saavedra. Assessment of Artificial Neural Network for bathymetry estimation using High Resolution Satellite imagery in Shallow Lakes : Case Study El Burullus Lake . *International Water Technology Conference*, pages 434–444, Mar. 2015.

[68] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[69] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pages 1057–1063. MIT Press, 2000.

[70] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, 34(6):26–38, Nov. 2017. ISSN 1053-5888. doi: 10.1109/MSP.2017.2743240.

[71] J. Achiam. Spinning Up Documentation. OpenAI, 2020. Available at `https://spinningup.openai.com/` (accessed: 25-05-2021).

[72] H. Dong, Z. Ding, and S. Zhang, editors. *Deep Reinforcement Learning: Fundamentals, Research and Applications*. Springer, 1^{st} edition, 2020.

[73] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning

algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, Dec. 2018. doi: 10.1126/science.aar6404.

[74] S. Racanière, T. Weber, D. P. Reichert, L. Buesing, A. Guez, D. Rezende, A. P. Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. Battaglia, D. Hassabis, D. Silver, and D. Wierstra. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 5691–5702, 2017.

[75] D. Ha and J. Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems*, pages 2450–2462, Dec. 2018.

[76] H. van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, and J. Modayil. Deep Reinforcement Learning and the Deadly Triad. arXiv:1812.02648, Dec. 2018.

[77] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015. doi: 10.1038/nature14236.

[78] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, San Juan, Puerto Rico, May 2016.

[79] S. Fujimoto, H. van Hoof, and D. Meger. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 4, pages 2587–2601, Stockholm, Sweden, July 2018.

[80] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 5, pages 2976–2989, Stockholm, Sweden, July 2018.

[81] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, May 2014. URL `http://arxiv.org/abs/1412.6980`.

[82] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *31st International Conference on Machine Learning, ICML 2014*, volume 1, pages 605–619, Beijing, China, June 2014.

[83] G. E. Uhlenbeck and L. S. Ornstein. On the Theory of the Brownian Motion. *Physical Review*, 36 (5):823–841, Sept. 1930. doi: 10.1103/PhysRev.36.823.

[84] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft Actor-Critic Algorithms and Applications. arXiv:1812.05905, Dec. 2018.

[85] Y. Torre-Enciso, I. Ortubia, L. I. López de Aguileta, and J. Marqués. Mutriku Wave Power Plant: from the thinking out to the reality. In *8th European Wave and Tidal Energy Conference (EWTEC 2009)*, pages 319–328, Uppsala, Sweden, 2009.

[86] Ente Vasco de la Energía. Marine Energies. Available at `https://www.eve.eus/Actuaciones/Actuaciones/Marina.aspx` (accessed: 21-04-2021).

[87] E. Medina-Lopez, W. Allsop, A. DImakopoulos, and T. Bruce. Conjectures on the Failure of the OWC Breakwater at Mutriku. In *Coastal Structures and Solutions to Coastal Disasters 2015: Resilient Coastal Communities - Proceedings of the Coastal Structures and Solutions to Coastal Disasters Joint Conference 2015*, pages 592–603, Reston, USA, July 2015. American Society of Civil Engineers. doi: 10.1061/9780784480304.063.

[88] K. Hasselmann, T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Eake, J. A. Euring, A. Gicnapp, D. E. Hasselmann, P. Kruseman, A. Meerburg, P. Mullen, D. J. Olbers, K. Richren, W. Sell, and H. Walden. Measurements of wind-wave growth and swell decay during the joint North Sea wave project (JONSWAP). *Ergnzungsheft zur Deutschen Hydrographischen Zeitschrift*, Reihe A 8(12):95, 1973.

[89] W. J. Pierson and L. Moskowitz. A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. *Journal of Geophysical Research*, 69(24):5181–5190, Dec. 1964. doi: 10.1029/JZ069i024p05181.

[90] J. Henriques, M. Lopes, R. Gomes, L. Gato, and A. Falcão. On the annual wave energy absorption by two-body heaving WECs with latching control. *Renewable Energy*, 45:31–40, Sept. 2012. doi: 10.1016/j.renene.2012.01.102.

[91] W. E. Cummins. The impulse response function and ship motions. In *Symposium on Ship Theory*, volume 9, pages 101–109, Hamburg, Norway, Oct. 1962.

[92] T. F. Ogilvie. Recent progress toward the understanding and prediction of ship motions. In *Fifth Symposium on Naval Hydrodynamics*, pages 3–128, Bergen, Norway, Sept. 1964.

[93] W. Sheng, R. Alcorn, and A. Lewis. A new method for radiation forces for floating platforms in waves. *Ocean Engineering*, 105:43–53, Sept. 2015. doi: 10.1016/j.oceaneng.2015.06.023.

[94] G. Duclos, A. H. Clément, and G. Chatry. Absorption of outgoing waves in a numerical wave tank using a self-adaptive boundary condition. *International Journal of Offshore and Polar Engineering*, 11(3):168–175, 2001.

[95] S. Dixon and C. A. Hall. *Fluid Mechanics and Thermodynamics of Turbomachinery*. Butterworth-Heinemann, Oxford, 7^{th} edition, 2014. doi: 10.1016/C2009-0-20205-4.

[96] M. Penalba and J. V. Ringwood. A reduced wave-to-wire model for controller design and power assessment of wave energy converters. In *Proceedings of the 3rd International Conference on Renewable Energies Offshore, RENEW 2018*, pages 379–386, Lisbon, Portugal, 2018.

[97] J. V. Ringwood. Wave energy control: status and perspectives 2020. *IFAC-PapersOnLine*, 53(2): 12271–12282, 2020. doi: 10.1016/j.ifacol.2020.12.1162.

[98] E. Tedeschi, M. Molinas, M. Carraro, and P. Mattavelli. Analysis of power extraction from irregular waves by all-electric power take off. In *IEEE Energy Conversion Congress and Exposition*, pages 2370–2377, 2010. doi: 10.1109/ECCE.2010.5617893.

[99] F. Ardhuin, J. E. Stopa, B. Chapron, F. Collard, R. Husson, R. E. Jensen, J. Johannessen, A. Mouche, M. Passaro, G. D. Quartly, V. Swail, and I. Young. Observing Sea States. *Frontiers in Marine Science*, 6(124), Apr. 2019. doi: 10.3389/fmars.2019.00124.

[100] W. Sheng and H. Li. A Method for Energy and Resource Assessment of Waves in Finite Water Depths. *Energies*, 10(4):460, Apr. 2017. doi: 10.3390/en10040460.

[101] P. Stoica and R. L. Moses. *Spectral Analysis of Signals*. Pearson Prentice Hall, 1^{st} edition, 2005.

[102] The MathWorks Inc. MATLAB Documentation - Signal Processing Toolbox - Windows, 2021. Available at `https://www.mathworks.com/help/signal/ug/windows.html` (accessed: 15-06-2021).

[103] OMIE. Annual Final Energy, 2021. Available at `https://www.omie.es/en/market-results/annual/average-final-prices/spanish-demand?scope=annual&year=2021` (accessed: 27-10-2021).