Oliver Stugard Os
Sebastian Thorsen Øverås

Intelligent Control Design for Power and Energy Management in Zero-Emission Autonomous Vessels

Master's thesis in Engineering and ICT
Supervisor: Mehdi Zadeh
June 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU
Norwegian University of
Science and Technology

Master Thesis

Intelligent Control Design for Power and Energy Management in Zero-Emission Autonomous Vessels

Oliver Os
Sebastian Øverås

June 2020

TMR4930

Marine Technology, Master Thesis

Department of Marine Technology
Faculty of Engineering

NTNU

Abstract

This master thesis investigates different power and energy management system (PEMS) algorithms on a zero-emission hybrid ship, primarily using methods from the field of reinforcement learning, a branch of machine learning.

The International Maritime Organization (IMO) has proposed stringent regulations in order to reduce emissions from shipping. Complying with IMO regulations as a step towards the long term goal of zero-emission shipping has spiked interest in ships powered by fuel cells and batteries in both academia and industry.

Batteries have been a huge success in the automotive industry. However, the insufficient energy density disqualifies it as a standalone energy source for deep-sea shipping. Therefore, hydrogen powered fuel cells are proposed to complement the battery. Fuel cells offer both high efficiency and high energy density, and are well suited for supplying steady power over long periods. On the other hand, batteries are capable of providing excellent power density and responsiveness, ensuring high performance and safety in maritime operations. Despite their promising outlook, fuel cells and batteries still have challenges to overcome. Health-aware control is required as improper usage can lead to a severe reduction in lifetime. Both systems are expensive, and the costs related to degradation and replacement are substantial when compared to the conventional internal combustion engine (ICE). Moreover, the characteristics of fuel cells and batteries change significantly as they undergo degradation. This makes it desirable to design an intelligent PEMS that can update the load sharing policy to ensure optimality despite the changing characteristics.

A health-aware PEMS, that aims to minimize both fuel consumption and component degradation costs, is essential for making zero-emission shipping competitive with ICEs. Traditional methods range from simple, rule-based control strategies, designed using the knowledge of domain experts, to more advanced optimization methods. Reinforcement Learning (RL), a branch of machine learning (ML), has the potential of outperforming traditional methods as it can adapt and learn continuously from changes in the environment. Optimization based methods rely on a predicted load, which is inaccurate due to the random stochastic nature of the ocean. RL is model free, and does not rely on predicting future loads to control the system.

A comprehensive literature review on costs related to fuel cell and battery degradation is conducted. The results are combined in a cost function, which serves as the objective function for learning the optimal power split between fuel cell and battery. Mathematical models for proton exchange membrane fuel cell (PEMFC) stacks and lithium-ion batteries are explored thoroughly. After careful evaluation of the trade-off between accuracy and computational requirement, linearized models for fuel cells and batteries are implemented for online PEMS.

The RL algorithms Q-learning, deep Q-learning and soft actor-critic are implemented. In addition, a rule-based algorithm and dynamic programming are implemented to serve as a benchmark for the RL algorithms. All models and algorithms are programmed in Python by the authors. Simulation of the models was conducted on a load profile from a real ship, and the performances of the algorithms were evaluated and compared. The deep Q-learning algorithm was able to decrease the cost of fuel cell degradation with 53 %, compared with the best performing benchmark algorithm. The soft actor-critic algorithm managed to reduce the fuel cost by 31 % and the battery degradation cost by 0.1 %, when compared to the rule-based algorithm.

The simulation results indicate that learning algorithms can reduce the total operating costs of ship power systems. Nonetheless, the learning based PEMS has room for improvement, as the field is still immature. Challenges such as complexity in reward function, continuous action and state space, overfitting training data and reliability issues have to be addressed to make it a viable competitor to the existing methods. All these issues are subject to further work.

Sammendrag

Denne masteroppgaven undersøker ulike algoritmer for kraft- og energistyringssystemer (PEMS, eng: power and energy management system) på et nullutslipp hybridskip. Det er hovedsakelig lagt vekt på metoder fra forsterkende læring, en gren av maskinlæring.

Den internasjonale sjøfartsorganisasjonen (IMO, eng: International Maritime Organisation) setter stadig strengere reguleringer for å redusere utslippene fra shippingindustrien. Med et mål om å tilfredsstille retningslinjene til IMO, samt overholde de langsiktige målene om nullutslipp shipping, har miljøvennlige skip med brenselceller og batterier som fremdriftssystemer, tiltrukket seg stor forskningsinteresse fra industrien og akademia de senere årene.

Batterier har vært en stor suksess i bilindustrien. Til tross for dette mangler dagens batteriteknologi energitettheten som kreves for å benyttes alene som fremdriftssystem til langdistanseskipsfart. Derfor har brenselsceller, med hydrogen som drivstoff, fått økt oppmerksomhet for bruk sammen med batteri i skip. Brenselsceller har både høy virkningsgrad og energitetthet, og kan tilføre gjevn kraft over lengre perioder. Batterier har derimot høy krafttetthet og kan håndtere store umiddelbare kraftendringer, noe som kreves for å gjennomføre trygge, maritime operasjoner med høy presisjon. Til tross for deres lovende utsikter er det flere utfordringer knyttet til bruken av brenselceller og batterier. Kontrollsystemer som inkluderer slitasje i beregningene er essensielt, da uforsiktig bruk kan resultere i drastisk kortere levetid både for brenselsceller og batterier. De er også dyre, og kostnadene knyttet til slitasje og utskiftning er betraktelig høyere enn for tradisjonelle forbrenningsmotorer. I tillegg endres karakteristikken til batterier og brenselsceller når betydelig slitasje påføres. Dermed er det nødvendig med et intelligent kraft- og energistyringssystem, som kan oppdatere kraftdelingsplanen til kontrolleren kontinuerlig for å sikre optimalitet uavhengig av karakterendringer.

Et slitasjebevisst PEMS med mål å minimere både drivstoff- og slitasjekostnader er vitalt for å gjøre skipsfart med nullutslipp konkurransedyktig med forbrenningsmotorer. Enkle regelbaserte algoritmer og optimeringsmetoder er tpyiske strategier for PEMS kontroll. Forsterkende læring (RL, eng: Reinforcement learning) er en undergren av maskinlæring (ML, eng: Machine learning) som potensielt kan utfordre tradisjonelle kontrollmetoder, da slike algoritmer kan tilpasse seg og lære fra endringer i omgivelsene. De optimeringsbaserte metodene tar utgangspunkt i en predikert last, som vil være unøyaktig grunnet tilfeldige lastpåkjenninger som bølger og vind. RL benytter seg ikke av en modell, og trenger heller ikke å predikere fremtidige laster for å kontrollere PEMS.

En omfattende litteraturstudie på kostnader relatert til slitasje av batterier og brenselceller som følge av bruksmønstre er utført. Resultatene er samlet i en kostnadsfunksjon for å finne den optimale kraftfordelingen mellom batteri og brenselscelle. Matematiske modeller for begge komponentene er også grundig undersøkt. Etter kritiske evalueringer av fordeler og ulemper knyttet til nøyaktighet og beregningshastighet, ble to lineariserte modeller for brenselsceller og batteri implementert for simuleringer av en online PEMS.

RL algoritmene Q-læring, dyp Q-læring og soft actor-critic algoritme er implementert for PEMS kontroll. I tillegg har dynamisk programmering og en regelbasert algoritme blitt implementert for sammenligningsgrunnlag for prestasjonen til RL algoritmene. Alle modeller og algoritmer har blitt implementert i Python av forfatterne. Modellsimuleringer ble gjennomført på lastprofilen fra et ekte skip, og prestasjonen til algoritmene ble evaluert og sammenlignet. Dyp Q-læringsalgoritmen klarte å minke slitasjekostnader på brenselscellen med 53 % og soft actor-critic algoritmen reduserte drivstoffkostnader med 31 % og batterislitasjekostnader med 0.1 % sammenlignet med den regelbaserte algoritmen.

Simuleringsresultatene indikerer at læringsalgoritmene kan redusere de operasjonelle kostnadene knyttet til kraftsystemet på skip. Til tross for dette har læringsbasert PEMS stort forbedringspotensial, da forskningsfeltet er nytt. Det er flere utfordringer knyttet til både belønningsfunksjon, kontinuerlige handlings- og tilstandsverdier, overtilpasning av treningsdata og pålitelighet som må adresseres før de kan bli en reell konkurrent til de eksisterende metodene for PEMS kontroll. Disse utfordringene er anbefalt som videre arbeid.

Preface

This paper is the result of a master thesis at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU) in Trondheim. The work is a continuation of our project thesis from the fall of 2019, written on the same subject. It marks the end of our Master of Science (MSc) degrees, both with a specialization in Marine Cybernetics.

The thesis is motivated by the demand for adopting advanced computational tools and utilize them for marine applications. The marine industry, although conservative, has lately picked up the pace in terms of digitalization. At the same time, we have eagerly pursued computer technology as well as cybernetics with the hope to participate in the ongoing transformation towards an increasingly digitalized industry. We aim to combine the domains of marine control systems with computer science. The main focus of this of this thesis has been to explore the use of reinforcement learning to optimize marine control systems.

Acknowledgment

We would like to express our gratitude to our supervisor Associate Professor Mehdi Zadeh for guidance and counseling during the work on the master thesis. A huge thanks is also directed to our co-supervisor PhD Fellow Namireddy Praveen Reddy for his consistent collaboration, support and advice. Our rewarding discussions have been a major encouragement throughout the process.

Trondheim, June, 2020

Oliver Stugard Os

Sebastian Thorsen Øverås

Table of Contents

List of Figures

List of Tables

Abbreviations

AC	Alternating Current
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BESS	Battery Energy Storage System
DC	Direct Current
DDPG	Deep Deterministic Policy Gradient
DOD	Depth of Discharge
DP	Dynamic Programming
DQL	Deep Q-Learning
DQN	Deep Q-Network
ECMS	Equivalent Cost Minimization Strategy
EEMS	Energy and Emission Management System
EMS	Energy management system
EOL	End-of-Life
ESS	Energy Storage System
FC	Fuel Cell
FCS	Fuel Cell System
FL	Fuzzy Logic
GDL	Gas Diffusion Layer
HESS	Hybrid Energy Storage System
HEV	Hybrid Electric Vehicle
ICE	Internal Combustion Engine
IMO	International Maritime Organization
LHV	Low-Heat Value
MASS	Maritime Autonomous Surface Ship
MDP	Markov Decision Process
ML	Machine Learning
MSE	Mean Squared Error
PMS	Power Management System
PEMFC	Proton-Exchange Membrane Fuel Cell
PEMS	Power and Energy Management System
RL	Reinforcement Learning
SAC	Soft Actor-Critic
SEI	Solid Electrolyte Interphase
SOC	State of Charge
SOH	State Of Health
TD	Temporal Difference

1 Introduction

1.1 Background and motivation

During the last couple of decades the attention towards the environmental impact from the maritime industry has increased. Although shipping is considered a conservative sector, most actors have by now set clear goals on how to cut greenhouse gas emissions and other pollutants. Governments and academia are mobilizing to prepare both regulations and technology to reduce emissions, further pushing corporations in the same direction.

Together with the rise of an increasing environmental conscience, autonomy is gaining traction within the global maritime sector. The International Maritime Organization (IMO) is trying to keep pace with the accelerated momentum the area has gained by instituting regulations on "Maritime Autonomous Surface Ships" (MASS) [1]. As the field matures, not only legitimate code of conduct and decrees are of importance. The technical aspect is equally eminent. Before autonomous vessels can be viewed as a viable alternative to ordinary ships, high standards in safety, security and emissions need to be in place. Strict requirements on the reliability and durability of ship systems, together with lower maintenance needs, are paramount in order to make autonomy economically feasible.

Conventional diesel-electric propulsion systems operate together with generators to deliver the required power load. Dynamic loads causes power fluctuations that increase the peak demand the engine has to provide for, which results in the need of additional generators. The power fluctuations reduce the overall efficiency along with an increased maintenance need [2]. Electrifying the ship propulsion system is a way of bypassing these obstacles. Besides, it's by now clear that fossil fuels cannot account for the future energy demand in the maritime sector, electric alternatives must be examined.

The automotive industry has had tremendous commercial success in its introduction of electric vehicles. Although the appetite for hybrid electric vehicles (HEV) has been more subtle, the technical maturity of hybridization greatly exceeds the maturity of such matters in the maritime sector. In spite of the many similarities, the industries differ on several important aspects. The electrification of the automotive industry is heavily based on the advancements in battery technology. Nevertheless, the inadequate energy density compared to liquid fuels like diesel, the low gravimetric density (high weight), and the enormous power demand of marine vessels makes batteries not suitable as a main energy source. The sheer size of on-board vessel propulsion systems for deep-sea shipping vessels renders the exclusive use of batteries pointless. Another supplemental energy source need to be considered.

Enter the fuel cell (FC). This promising technology can run on hydrogen and produce only water and heat as by-products. Fuel cells generally have a higher efficiency than combustion engines, are reliable and silent as there are few moving parts, and are not polluting. Their higher energy density compliments the lack of such in batteries. Fuel cells are a suitable alternative to the conventional generator set in marine vessels, as they can deliver the slowly varying power to meet the demand. They do, however, suffer from limitations such as high system price as and short life span. Despite these challenges, the additional untapped potential of fuel cells have secured monumental funds for research and development in the hopes of establishing it as the energy source of choice in the near future.

Undeterred by the current high cost levels, all-electric marine vessels brings forth several advantages compared to conventional vessels. A FC-battery system is an example of a hybrid power system, which is explored throughout this thesis. Batteries compensate for transient loads that are too fast for the fuel cell's dynamic response. Furthermore, excess power produced in the FC can be re-captured and used to charge the battery. Proper load management decreases the fuel consumption and can help curb component degradation by ensuring health-aware load demands from the components. A more sophisticated energy management system can achieve even better results by utilizing each power source at, or close to, their maximum efficiency. Ultimately, hybridization adds flexibility across the operational spectrum of a marine vessel as the system can meet demands from more diverse loads.

Two pilot projects, Yara Birkeland and the NTNU Autoferry, highlights many of the aforementioned topics. Yara Birkeland, launched in 2020 and fully autonomous by 2022, will be the world's first zero-emission, autonomous container ship. It will replace 40 000 truck trips along the Norwegian coast every year, contributing to the reduction of greenhouse gas emissions and improving road safety [3]. The Autoferry at NTNU is a concept that introduce a more flexible and environmentally-friendly passenger ferry for urban water transport. It is located in Trondheim, is all-electric, and will operate autonomously as an on-demand ferry. Norway is a pioneer in the "autonomatization" at sea, with many ongoing government-backed projects. Hybridization of ships will play an important role in the development towards full autonomy at sea. All-electric ships can either be fully battery-powered or combined with another all-electric energy source like fuel cells.

The utilization of autonomy in shipping can also lead to economic advantages. As regulations on pollutants continue to increase, it is easy to envision a near future where emission taxes make hybrid alternatives economically viable. Fully autonomous ships can plan with only the mission in mind, not restrained by the needs of on-board crew. For autonomy to be successful in the maritime industry, the control system must outperform human operators. The ability to optimize for all system variables is therefore essential. The power flow should be distributed among the hybrid power sources such that each source is optimally used, resulting in lower costs related to fuel and degradation of components [4].

1.2 Objectives

The work done in this thesis aim to find the costs related to the use of various control strategies on a FC-battery hybrid system on a ship. Zero-emission technology is not yet competitive with ICEs mainly due to cost. One of the dominant contributions to the high cost of fuel cells is the short life span. Ill-conceived use of FC-battery hybrid systems accelerate the aging process of components, leading to a higher replacement rate. Therefore, to narrow the price gap between FC-battery hybrid systems and ICEs, component degradation should be included in the cost optimization.

Thus, fuel consumption and several degradation processes of the fuel cell and battery is considered. In order to model this, different aging mechanism needs to be mapped. The literature that considers both the fuel cost and wear and tear of components is limited as the control problem is highly nonlinear. Degradation rates vary across the system's lifetime and ought to be managed in such a way that the demanded power yields minimal strain on the components.

Several control strategies are explored and their performance compared. To get an accurate representation of the power dynamics of marine vessels, data on the required power from real-life ships are essential. Hence, the proposed algorithms use load profiles from the industry to manage the distribution of power in order to minimize the running costs.

The final objective is to investigate whether the more sophisticated control strategies yield a lower running costs than conventional rule-based methods used in the industry. The trade-off between complexity and computational efficiency is also of interest in order to enable intelligent energy management on autonomous ships.

The main objectives if the thesis can be summarized as follows:

1. Modeling of the fuel cell and battery components and a power and energy management system. The models account for the costs of hydrogen fuel and the internal aging processes related to the use of each component.

2. Formulate a cost function that translates the degradation into U.S. dollars.
3. Implement various intelligent algorithms to control the PEMS.
4. Run simulations with the control strategies on load profiles from a real vessel to compare the costs of the respective algorithms.
5. Discuss the validity of the results rooted in the assumptions and limitations of the model.

1.3 Scope and limitations

This paper researches control strategies for a power and energy management system that includes a proton-exchange membrane fuel cell (PEMFC) and battery. Other zero-emission energy sources, like the supercapacitor, are principally omitted in the model due to the additional complexity they would add. Supercapacitors have an even quicker dynamic response than batteries and can be used in combination for hybrid FC-batter-supercapacitor power systems. As the model presented in the paper does not include such a hybrid arrangement, it is assumed that the battery acts instantly, and can thus efficiently deal transient loads.

The whole electric power system is based on a DC grid with a constant bus voltage. With the running costs of a ship as the main focus, component sizing is deemed out of scope. The results presented are solely a study derived from offline simulations of ship load profiles and are thus not based on real ship experiments.

Several aging mechanism behave nonlinear and are unfeasible to implement in a power and energy management system. Multiple approaches, including linearization of characteristics and limiting the operational range of the components are explored to overcome this challenge. Additionally, some parameters like temperature does indeed affect the aging of FCs and batteries, but are not considered.

1.4 Thesis structure

The thesis expands the work of our project thesis submitted in the fall of 2019. Especially the first chapters are based on theory accumulated in the literature review. The organization of the ensuing report is divided into the following chapters:

Chapter 2 presents the working principles and characteristics of fuel cells and batteries. In addition, it provides the proposed topology of the shipboard power system.

Chapter 3 explains the importance of a PEMS and investigates how different control methods increases the performance of hybrid power systems.

Chapter 4 discusses costs related to usage of zero-emission energy sources. Both fuel and component degradation are considered, and a complete cost function for a FC-battery energy system is proposed.

Chapter 5 describes the FC and battery models, in addition to the power and energy management system. It also discusses and elaborates on the underlying assumptions and simplifications.

Chapter 6 gives a thorough introduction and discussion on the theory of the control strategies used in the simulations.

Chapter 7 explains how the algorithms described in the previous chapter are implemented and how the simulations are carried out, before presenting and discussing the corresponding results.

Chapter 8 conclude the results of the study. In addition, suggestions for further work is presented.

2 Zero Emission Energy Sources for Marine Applications

In this chapter the arrangement of an all-electric hybrid power system and its components are presented. A fuel cell-battery hybrid propulsion solution is outlined. The following sections identify and describe the fundamental working principles and characteristics of a PEM fuel cell and a lithium-ion battery. Strengths and weaknesses of the technologies, mostly explored in the project thesis this paper is based on, are recapitulated and discussed. Especially mechanisms related to the aging and degradation of components are comprehensively reviewed as these greatly alters lifetime of components and ultimately the vessels operation costs.

Furthermore, a short recap of key component features is presented. Table 2.1 summarize the most important characteristics and the current status of fuel cells and lithium-ion batteries, including their costs.

The chapter is rounded off by a short discussion on how and why the aforementioned components can be used together to take advantage of their strengths. A system architecture for a marine vessel is outlined to give an overview of the energy flow of a vessel.

2.1 Fuel cell

Fuel cells (FC) generate electrical power through a chemical reaction. Whereas combustion engines release heat energy, FCs produce electrical energy. The fuel cells studied in this paper operate on hydrogen gas, which is the most common fuel used in fuel cells. We will exclusively investigate the proton exchange membrane fuel cell (PEMFC), the most common fuel cell for transport applications [5]. In general, they are efficient compared to combustion engines, with a practical efficiency in the range of 50-60 %.

The main loss factors are activation losses, Ohmic losses and mass transport losses, also known as concentration losses [6]. The only by-products are heat and water, which makes fuel cells inherently clean as they do not emit any environmentally dangerous pollutants.

PEM fuel cells require a constant flow of oxygen (O_2) and pure hydrogen (H_2) to operate. All fuel cells are made up of an anode, a cathode, and an electrolyte membrane. The membrane has important functions in the fuel cell such as proton exchange between electrodes and separating the cathode and anode environments. In a PEMFC, a stream of hydrogen passes through the anode where it splits into electrons and protons (hydrogen ions) by a catalyst, usually made of platinum. The protons permeate through the electrolyte membrane while the electrons are forced through a circuit, generating electricity and heat. It is critical that the membrane only permits hydrogen ions as the contrary would lead to a short-circuit. At the cathode side, water is formed as oxygen molecules are reacting to the protons that have permeated through the membrane and the electrons arriving from the external circuit. Figure 2.1 illustrates the working principle of a PEMFC.

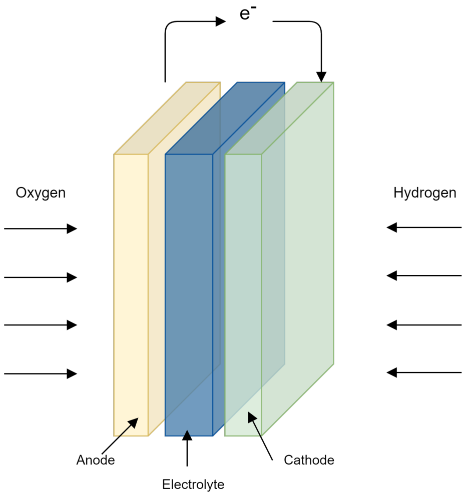


Figure 2.1: Fundamental PEMFC operation [7]

To deliver a larger amount of energy, fuel cells can either be placed in series to yield higher voltage or in parallel to allow higher current to be supplied. Such a design is called a fuel cell stack and shown in Figure 2.2.

2.1.1 Characteristics

Unlike internal combustion engines (ICE) that convert chemical energy into heat by combustion, the efficiency of fuel cells are not related to the maximum operating temperature. Hence, they are not restricted by the Carnot efficiency limit [9]. The efficiency depends on the chemical reaction inside the fuel cell, which results in

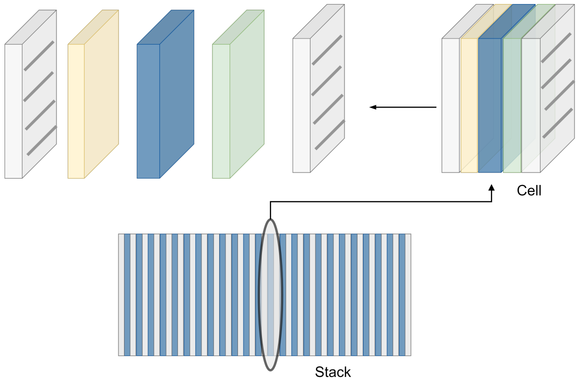


Figure 2.2: Overview of a fuel cell stack [8]

a significantly higher theoretical value than traditional ICEs [10]. The efficiency of each cell is calculated by dividing the actual voltage, V_c, with the open circuit voltage with reference to the low-heat value (LHV) [5]:

$$\text{efficiency} = \frac{V_c}{\text{LHV}} \cdot 100\ \% = \frac{V_c}{1.25} \cdot 100\ \% \tag{2.1.1}$$

In any case, it should be noted that the practical efficiency is around 60 % over a wide range of the power spectrum [11].

PEMFCs use a humidified polymer-based membrane as an electric insulator. The operating temperature ranges from 50 to 100 °C. Higher temperatures are not feasible as the membrane needs to be humid under operation [12]. Given the low operation temperature, little excess heat is generated and therefore heat recovery is not an option. However, the low temperature allows for a swifter startup time compared to other fuel cell types.

The dynamic response of fuel cells is inadequate when handling rapid load changes which marine vessels are subject to. This is a result of their relatively low specific power density. It is recognized as a major weakness of fuel cell systems [6]. If the fuel cell is unable to provide the required instantaneous power output demand from accelerations, it will deteriorate the electric dynamic responses [13]. To counter this time-delayed response and limited power output, auxiliary power devices such as batteries and supercapacitors should be combined to make a hybrid propulsion system.

PEMFCs are suitable as a main source of power in marine vessels due to their high efficiency, low operation noise, low temperature, vibration levels, and their low environmental impact. There are, however, various obstacles the fuel cell technology has to overcome for it to be the go-to power source choice in the maritime industry. One of the main issues today is the cost. The platinum catalyst needed in a PEMFC is expensive, leading to high unit costs [12]. According to the U.S.

Department of Energy, the PEMFC cost was in 2015 at \$53/kW$_{net}$. The target price set for 2020 is \$40/kW$_{net}$ [14]. In the last 10 years, the price has dropped substantially from \$69/kW$_{net}$ in 2009 [15]. The reduction in fuel cell prices is displayed in Figure 2.3. Please note that it looks at the fabrication of 100–500 thousand units manufactured each year.

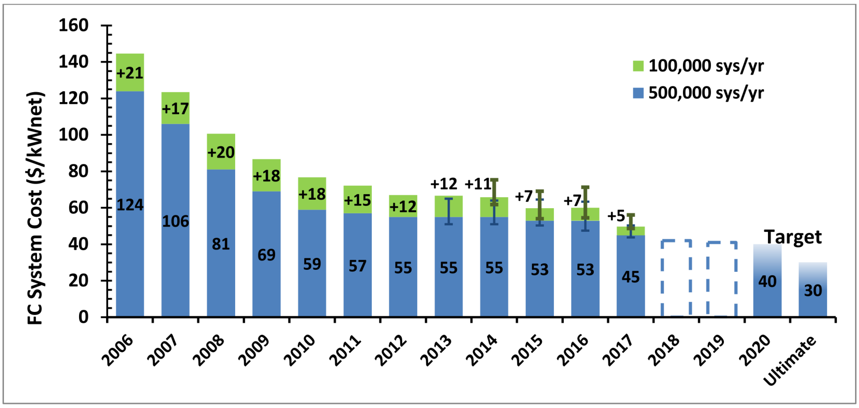


Figure 2.3: Price development for fuel cells [14]

Improvement in fuel cell technology is the main driver behind the growth in the hydrogen economy [16]. Hydrogen has great energy density characteristics with respect to mass, which makes it a lightweight option compared to other fuel alternatives, and fuel cells are thus considered high-energy systems. However, it has a poor energy density on a volumetric basis, resulting in an increased space demand.

Storing hydrogen is also a key obstacle for the commercialization of fuel cell technology. For hydrogen to become a feasible power source for marine vessels, there are requirements of safety, compactness and cost-efficient storage solutions that need to be addressed. As hydrogen transport is an expensive process, the arguments for distributed production are well founded [17]. Hydrogen can be stored as compressed gas, liquid or in solid phase [18]. Transporting hydrogen as a compressed gas is a viable option for short distances. To reduce the distribution costs and make it easier to transport, an option is to cool the hydrogen down through a cryogenic liquefaction process.

2.1.2 Degradation and lifetime

One of the main challenges that hinder fuel cell technology to enter the industry is their short life span. FC lifetimes, currently in the range of 2000–4000 hours, are not yet within the U.S. Department of Energy's durability targets of 5000 running hours [14]. Measures to boost fuel cell lifetimes include among others material composition, reduction of degradation causes and enhancing the stack design [19]. Reducing the costs related to wear and tear is one of the main objectives of this thesis, and some of the most significant FC degradation mechanism will be elaborated below.

Fuel starvation is one of the main contributors to PEMFC degradation and aging. The starvation takes place when reactants are used faster than they are supplied to the cell, resulting in a reversing of fuel cell voltage which further leads to corrosion. Fuel starvation causes permanent damage to the cell as well as a reduction in its performance [6]. Thus, fuel starvation should be avoided, even for brief moments. FCs are especially prone to starvation during transients as the fuel delivery system has slow dynamics due to the mechanical equipment such as valves, which are slow in adjusting their setting based on the reference value. Fuel starvation is more likely to happen in the oxygen supply system due to the time delay of mechanical valves and the compressor motor that supplies the air. To help avoid fuel starvation, the oxygen excess ratio can be adjusted by changing the mass flow into the cathode such that minimum fuel cell stress is inflicted. In practice, this is achieved by setting constraints on the fuel cell's power slope. Experiments have proven that this improves overall fuel cell performance and lifetime [20]. Furthermore, effects such as high transient loading, start/stop cycles, and high/low power contribute to starvation.

The list below summarizes some of the most common FC degradation methods [21]:

1. **Catalyst degradation** is one of the most well-known causes of decay in FC performance. Platinum on the surface of the catalyst is initially spread evenly over the surface, but over time the molecules have a tendency to agglomerate, decreasing the surface area covered by platinum. This leads to a reduced cell voltage. Fuel starvation and running at low current densities are some of the major contributors to this phenomenon.

2. **Membrane degradation** causes degradation in the form of thermal, mechanical or chemical stress on the membrane, and reduces the membrane quality. Avoiding high temperature in the engine will help prevent this.

3. **Gas diffusion layer degradation** (GDL) possesses many of the same degradation methods as catalyst degradation. The same materials are often used in both, and the result can be a lack of sufficient reactant supply locally in the fuel cell. GDL degradation is caused by fuel starvation, high transient loading or start/stop cycles.

To represent aging effects it is beneficial to define the end-of-life (EOL) of the fuel cell. The term indicates when the FC is at the end of its life-cycle and can be used to estimate the remaining useful life of the system. EOL thresholds can be set based on mission conformity or as a definitive limit that renders the fuel cell not fit for further use [19]. The U.S. Department of Energy defines fuel cell end-of-life when it reaches a 10 % voltage drop [22]. However, this threshold may not provide a conclusive representation of FC durability when the loads are varying. Alternative EOL definitions includes using the cumulative energy of the FC, but introduces complexity to the overall approach. Chen et al. [23] accounts for hydrogen consumption, PEMFC stack price and system efficiency to propose an economical lifetime threshold. This results in an estimated fuel cell lifetime that is cost-effective, but assumes the degradation rates are known. The focus of this thesis is on the costs related to the usage of the hybrid power system. Less emphasis has been put on the replacement of components, as this is just a cost obtained from the manufacturer.

Calculating the cost of fuel cell degradation is a complex process, involving multiple features. A fuel cell stack consists of multiple fuel cells in series, as displayed earlier in Figure 2.2. Each cell consists of different components, including the membrane, the electrodes, the gas diffusion layers and the bipolar plates. Different degradation processes occur on each component. In addition, for each cell in the stack, degradation transpire at different rates. For instance, the cells on the edges of the stack tend to degrade faster than cells in the middle of the stack [19]. These effects, however, are difficult to model and is outside the scope of this thesis.

Multiple chemical effects contribute to fuel cell degradation. Usage of the FC stack substantially determines how much, and where, degradation occurs. The following list summarizes important measures that can significantly reduce fuel cell degradation:

- Avoid running the FC at *high power* as it causes reactant starvation [24], [6].
- Prevent running the fuel cell in an idle state, i.e. *low power*, as it will cause electrochemical active surface area reduction [21].
- Avoid needless *transient loading* to preserve humidity and temperature as well as preventing local fuel starvation [25]
- Prevent *start/stop cycles* as it contributes drastically to degradation as a result of carbon corrosion in the cell [26].
- Avoid *high power load cycles* in order to prevent humidity changes that causes holes in the membrane [27].
- Reduce the fuel cell load if the *temperature* is too high [21].

2.2 Battery

In contrast to the previously mentioned main energy sources, ICE and FC, energy storage devices serve as auxiliaries to give the energy system desirable characteristics. A battery is an example of a energy storage device used in zero-emission ships. Supercapacitors are frequently used as part of an all-electric marine power system, but are not considered here. Auxiliary energy sources are added in order to provide higher responsiveness and power density, in addition to increased reliability and safety. This section covers an introduction of batteries and their properties.

A battery is an electrochemical device that can store, charge and discharge energy.

Batteries consist of one or more electrochemical cells that are built on three components; an anode, a cathode, and the electrolyte. Figure 2.4 illustrates a basic lithium-ion battery layout. The purpose of a cell is to convert chemical energy into electrical energy. This happens through two different reactions; one at the cathode and one the anode. At the anode, electrons are released through an oxidation reaction between the metal atoms of the anode and the electrolyte. At the cathode, electrons are released to the electrolyte through a reduction reaction. The anode and the cathode are coupled together through an electrical leading material. Due to the difference in charge, negative at the anode and positive at the cathode, the electrons travel from the anode to the cathode, generating electric voltage. When charging the battery, a current is used to reverse the process. This way the battery can efficiently convert electric energy to chemical energy, store it, and then discharge it back as electric energy.

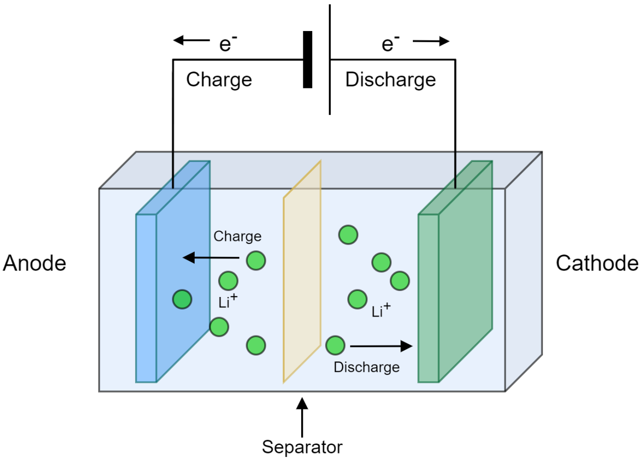


Figure 2.4: Fundamental lithium-ion battery operation

2.2.1 Characteristics

Batteries have several characteristics that are important for ships. The energy density is a function of the voltage and capacity of the cells, which depends on the chemical properties of the system like anode and cathode materials. Typically, the energy density of batteries are too low for marine application due to the linear increase in cost with battery size. For applications such as deep-sea shipping, a huge amount of energy is required. A battery's insufficient energy density makes them unfit as a primary energy source for such operations.

The power density is a function of the voltage and is mainly determined by the surface area of the anode and cathode, which is important for the speed of the redox reactions in the cell. In general, batteries have a high power density, as they can deliver high amounts of power in a short time [28]. This is important for ships to provide the required maneuvering and acceleration abilities.

The state of charge (SOC) is the percentage of energy remaining for use in the battery. As an example, a SOC of 100 % indicates that the battery is fully charged, whereas the battery is depleted when the SOC is 0 %. Monitoring the SOC of a battery is a complex task that includes measuring voltage, current flow, and temperature of the battery. Monitoring and adapting battery usage to the SOC is important, as the state of health (SOH) is heavily influenced by the SOC.

The SOH describes the general condition of the battery. It is a measurement of how well the battery performs, compared to a similar, brand-new battery. Over time, the voltage delivered, energy density and general performance, all related to the SOH, decrease. For lithium-ion batteries, the battery is considered to fail when the SOH is less than 80 % of its initial value [29]. The SOH is determined by the age of the battery and how it's used. Since batteries for transport applications are expensive, they should not be used carelessly to prevent avoidable economic losses related to the replacement of the battery [30]–[32].

2.2.2 Degradation and lifetime

Battery degradation refers to the process where battery performance decreases with time and usage. The process accounts for most of the cost related to battery usage and is therefore important to take into consideration when using the battery. It is a very complex process, which varies with different battery parameters. Some of the most relevant degradation mechanisms are discussed in this section.

Aging factors that contribute to battery degradation in lithium-ion batteries are discussed thoroughly by Harting et al. [33]. Figure 2.5 summarizes some of the most prominent aging factors in lithium-ion batteries.

According to Xu et al. [28], degradation of lithium-ion batteries can be split into two main components; a linear and nonlinear effect. The degradation rate is reliant on the battery's current state of life, which can accelerate the degradation from other processes. The linear process can be divided into two separate effects. Calendar aging is related to the battery's inherent degradation. This happens over time, regardless of how the battery is used, and is a function of time only. Cycle aging depends on the operational temperature and SOC of the battery and describes the life lost between one cycle of charging and discharging. In addition to the average temperature and SOC of the cycle, the depth of discharge (DOD), which denotes how much energy is cycled in and out of the battery in the given cycle, also contributes to loss of battery life [34].

Experiments with lithium-ion batteries have shown that the battery degradation rate is significantly higher in the early stages of battery life. Then the degradation rate is low for most of the battery lifetime, before it increases rapidly as the battery approaches its EOL. Therefore, the degradation process is highly nonlinear with respect to the battery lifetime and the number of charge-discharge cycles. Several

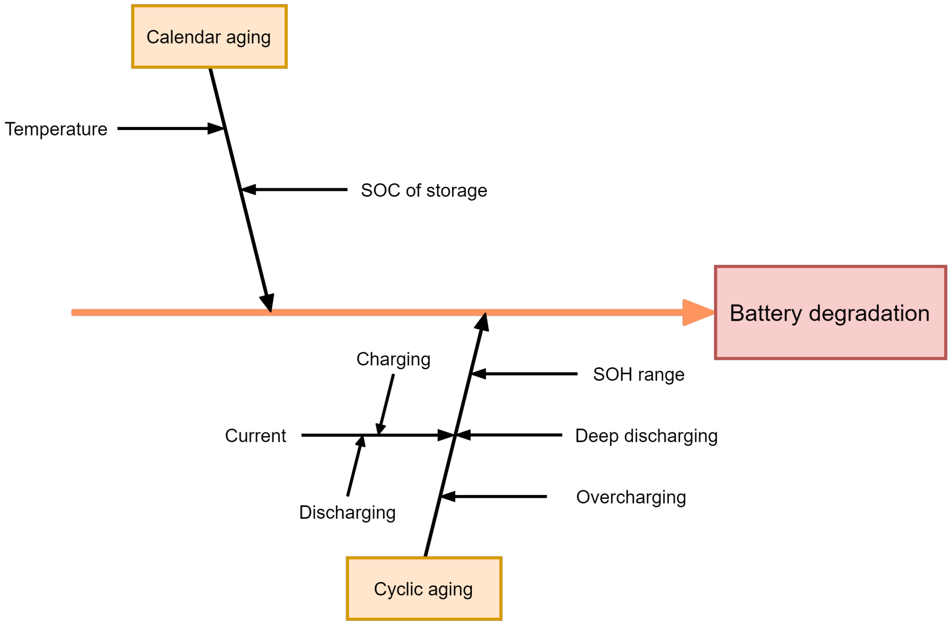


Figure 2.5: Fishbone diagram of battery aging processes due to operational factors. Excerpt from Harting et al. [33]

effects cause this, but one of the most prominent is the formation of solid electrolyte interphase (SEI) film. The SEI film typically forms during charging and causes loss of lithium on the negative electrode, which negatively influences the capacity of the battery cell [35].

Battery aging can be modeled as an internal resistance model, as the internal resistance of the battery tends to increase with battery aging. This makes the internal resistance a good indicator of the remaining expected battery life. A capacity degradation model is also commonly applied. When the capacity of a battery is reduced by 20 % of its original value, batteries have reached EOL and should be replaced. As a result, it is the most common indicator of battery SOH. There have been many attempts to calculate the battery capacity in the literature, experimentally and theoretically, which have proved to be a challenge [8].

Koller et al. [34] suggests a way of incorporating battery degradation in a battery energy storage system (BESS) based on model predictive control. The battery was modeled as a linear time-invariant system to make the optimization computationally feasible. Degradation of the battery, however, is a highly nonlinear process, and the resulting optimization is not convex. The paper lists four major causes of battery degradation, of which the last three are considered in the resulting cost function:

1. High operation temperature.

2. High and low SOC.
3. High DOD.
4. High current-rate/high power-rate.

The use of batteries in combination with FCs has several benefits. Batteries have successfully been deployed in a wide range of different transport segments, and have been through extensive research and testing. Auxiliary power sources for marine applications such as batteries increase system performance and fuel efficiency. It also supports the FC by providing high currents during rapid load changes that otherwise would induce stress on the FC [36].

2.3 Fuel cell and battery comparison

Figure 2.6 and Figure 2.7 outlines the energy and power densities as well as the dynamic response time of fuel cells and batteries. The Ragone plot is redrawn from Kötz et al. [37] with a logarithmic scale and the dynamic response of the FC and battery is redrawn from Thounthong et al. [38]. Note that the power unit in the latter figure is normalized and given *per unit*.

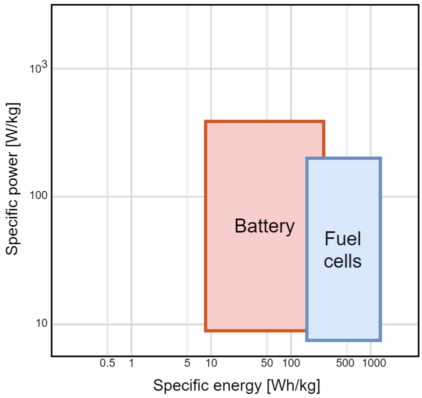


Figure 2.6: Energy density Ragone plot redrawn from Kötz et al. [37]

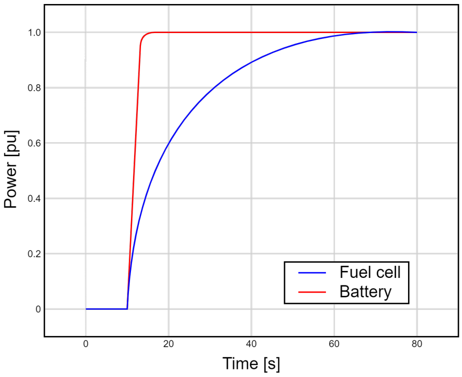


Figure 2.7: Dynamic response time redrawn from Thounthong et al. [38]

The specific power density of an energy source indicates the power output it can provide per unit of mass. A high power density means that the system can release energy abruptly. Batteries generally enjoy a high specific power density, and are capable of managing large fluctuations in energy output. PEMFCs does not have the same dynamic capabilities, and can thus not control large power transients. On the other hand, the specific energy (gravimetric) density is a measure of capacity. It indicates how much energy a system contains per unit of mass. The high energy density of hydrogen justifies the use of fuel cells as a primary energy source.

The high power density of batteries together with the high energy density of fuel cells provide a desirable foundation for a power system. With PEMFCs which are an order of magnitude slower than batteries, the combination of the two are progressively becoming a more feasible alternative for all-electric ship power systems. Table 2.1 summarize the key characteristics of PEMFCs and lithium-ion batteries.

Table 2.1: Key characteristics of FC and battery [14], [39]–[42]

Fuel cell and battery characteristics

Feature	PEMFC	Li-ion battery
Cost	45 $/kW	176 $/kWh
Lifetime	2000–4000 hours	5–10 years / 5000 cycles
Energy density	800–10 000 Wh/kg	120–240 Wh/kg
Power density	1–10 W/kg	50–2500 W/kg

2.4 Hybrid power systems

Due to shortcomings in fuel cell dynamics and their limited resilience to voltage fluctuations, fuel cells are not suitable as a single energy source in marine applications. Hybrid power systems consisting of both FC and battery provides a solution to this problem [43]. Hybridization with FCs as the primary power source works by connecting secondary energy storage units, like a battery, to the complete system. This allows the system to split the power between the components to achieve a greater system efficiency. The process of determining the share of power to each component is governed by the control strategy of the system level controller. Chapter 3 explains how this power balance is chosen and what benefits the system gains from this procedure.

2.4.1 Power system architecture

The power system architecture highlights how the components of the power system are connected. The energy sources and energy storage devices for marine applications are usually delivering current through a grid system. Traditionally, AC (alternating current) grid systems have been used for ships. In this setting, the frequency of current generators connected to the grid needs to match both the system voltage and frequency. With the introduction of multiple energy sources in hybrid energy systems, this is a huge drawback. As a result, a DC (direct current) grid provides several advantages, and is becoming increasingly popular. First of all, it enables variable engine speeds. This means that the speed of the engines and generators can be optimized to the system load situation, which is a huge benefit when controlling it with a PEMS as it can significantly increase the operating efficiency. Moreover, the main engines can operate at their optimal efficiency. It also enables the integration of an energy storage, which gives dynamic flexibility, better safety and increased efficiency to the energy system. Thirdly, a DC bus makes it easier to integrate multiple energy sources, as they don't need synchronization, and removes the need for multiple conversion and transformation stages. These components also lead to efficiency loss and increased fuel consumption. Additionally, DC grids are simpler than AC grids, which has the benefit of increased safety and better fault prediction utilities. An example of a DC grid system with the proposed FC-battery power system is shown as a single line diagram in Figure 2.8.

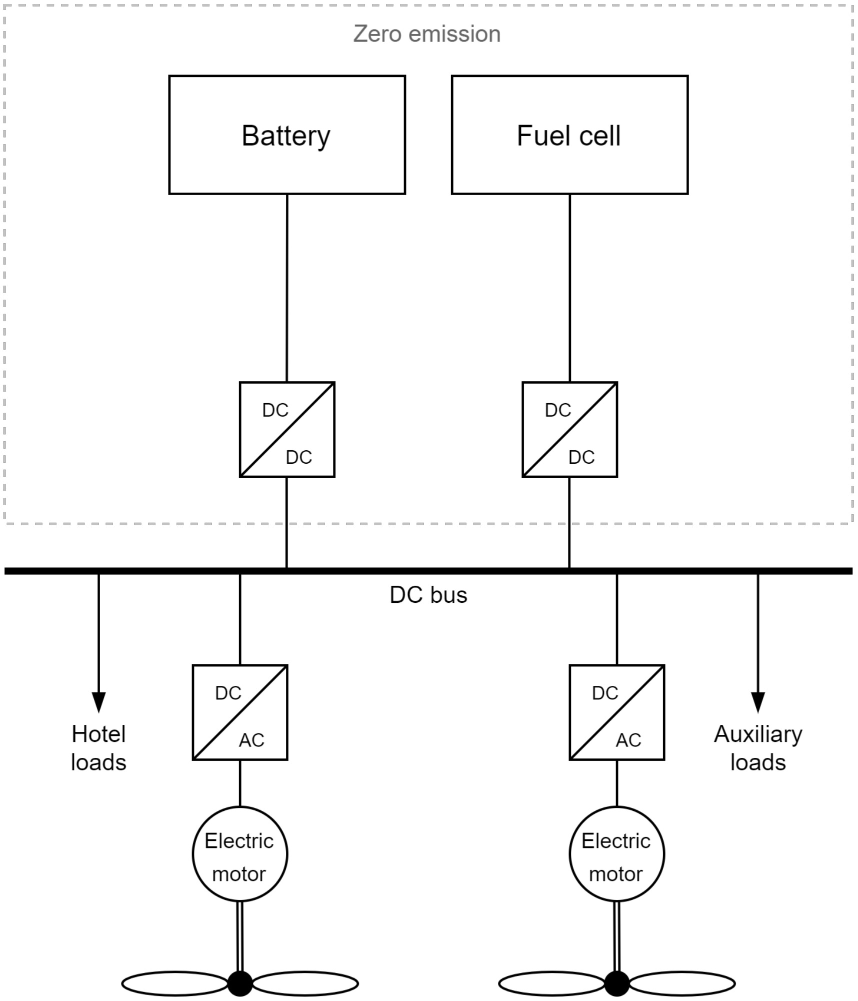


Figure 2.8: Single line diagram displaying a FC-battery hybrid propulsion system [44]

3 Power and Energy Management System

Power and energy management systems (PEMS) are computer-aided tools that are used to control and optimize the performance of power systems. The overall objective of the system is to regulate the power on the DC bus to cover the demand for energy at any given time. The energy management system (EMS) governs the high-level system control, determining the amount of energy to use from each power source to meet the energy demand. The EMS controls the flow of energy from the FC and from/to the energy storage systems (ESS). Additionally, the EMS manage the load sharing between energy sources and is thus controlling the charging and discharging of the battery. An advanced EMS is able to learn from historical data to predict future usage.

The power management system (PMS) ensures that the calculated electrical power from the EMS is properly transmitted to the energy sources. Another crucial task of the PMS is to override decisions from the EMS if the demand is outside the energy source's safety boundaries. Figure 3.1 illustrates the modeled control level topology on a marine vessel with a fuel cell and battery. For the rest of the report, both the EMS and PMS are considered as one integrated unit called PEMS as both terms are used interchangeably in the industry.

For a system with more than one energy source, energy management is essential. By utilizing the strengths of each source, the PEMS can have a positive influence on fuel consumption, lifetime of the energy sources, and overall performance. As discussed in Chapter 2, the energy sources presented in this paper have different strengths and weaknesses. These characteristics play a major role in how the PEMS should operate to reduce costs, which will be discussed further in this chapter.

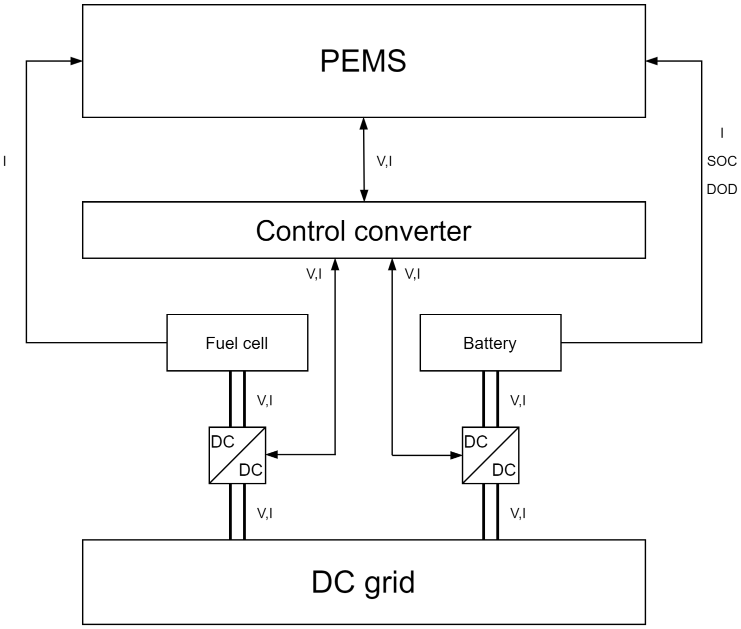


Figure 3.1: Block diagram of a complete FC-battery control system

3.1 Control objectives

The general aim of a PEMS is to utilize a minimum of energy to operate the system at the lowest cost possible, while staying well within regulations and safety constraints. According to A. Sørensen [45], the primary objectives of a PEMS consist of three main functions; the generation and management of power, load management, and power distribution.

- **Power generation and management** encompass monitoring the energy flow of the vessel and it's frequencies. Supervision of load sharing functions and control logic is employed to coordinate energy sources as needed.
- **Load management** addresses monitoring of the required load as well as the limitations of the power manage
- **Distribution management** manage the sequence of power configuration. This includes allocating loads to always meet the energy demand.

An energy and emission management system (EEMS) is an extension of a PEMS as it is a high-level controller that also includes the emissions from the system [4]. The main objectives of an EEMS are shown in Figure 3.2.

Additional PEMS functions include maneuvering capabilities, dynamic positioning and blackout restoration. The primary objective, however, is to deliver the required power to the engine in a stable manner.

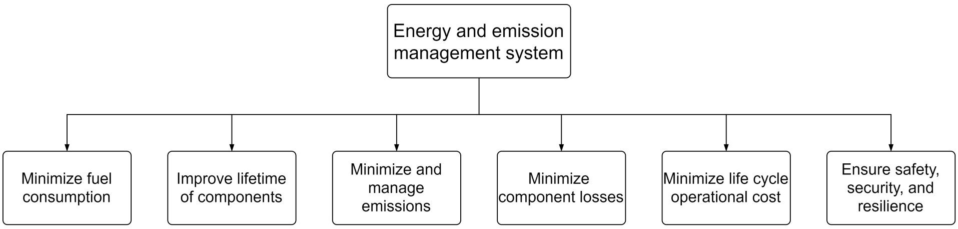

Energy and emission management system
Minimize fuel consumption
Improve lifetime of components
Minimize and manage emissions
Minimize component losses
Minimize life cycle operational cost
Ensure safety, security, and resilience

Figure 3.2: Overview of energy and emission management system objectives [4]

3.1.1 Load management

Load management is the process of adjusting the electricity supply while maintaining the same power output. Balancing the supply of electricity helps reduce the need for electricity at peak demand by clever load management. The process usually involves utilizing stored power from the ESS units when the demand is high, and use any excess power when the demand is low to recharge these units. Chapaloglou et al. [46] integrated load forecasting by an artificial neural network into the EMS and achieved an optimal operating level for the diesel generators by handling peak demands with a battery storage system.

Peak shaving is a load management method that aims to reduce the peak demand for highly variable loads. On vessels that use FCs as the main energy source, peak shaving can be accomplished by supporting the FC with other energy sources such as batteries. ESSs have faster dynamic responses and can thus reduce the peak power the FC needs to generate. To support peak shaving for multiple instances, the energy storage systems must be recharged. During low power demand, the FC can be run at close to optimal efficiency and use excess electricity to charge the ESSs. Figure 3.3 illustrates how peak shaving can reduce the peak load provided by the main power source, i.e. the FC, during an operation.

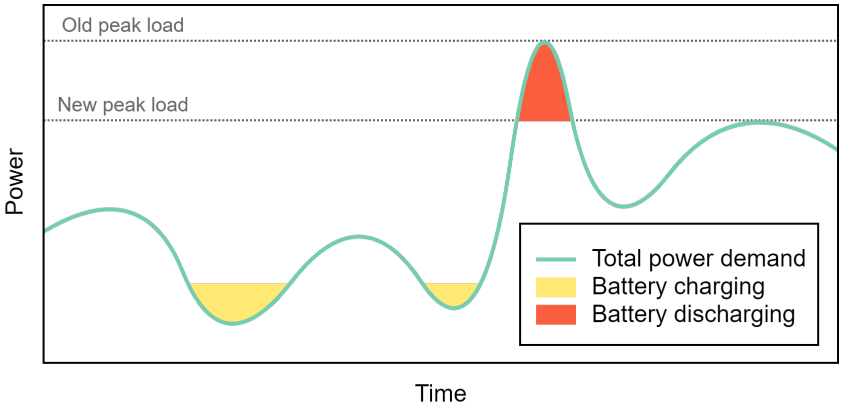


Figure 3.3: Peak shaving on a generic load profile

Another load management method, often used in combination with peak shaving methods, is *load leveling*. By load leveling, the ESS delivers the fluctuating loads, while the main power source produces slowly varying power to meet the average power demand. Bø et al. [2] suggests a control hierarchy where the battery dynamically removes power variations depending on variations and battery temperature to achieve a more stable load for the main power source. Figure 3.4 shows the concept of load smoothing.

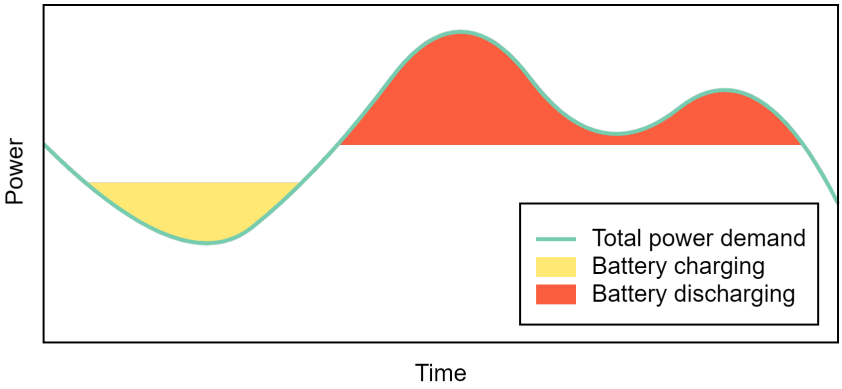


Figure 3.4: Load smoothing on a generic load profile

Excessive fluctuations leads to a series of negative consequences. For mechanical systems, torque and power fluctuations increase wear and tear from mechanical stress. Electrical systems are also negatively influenced as fluctuations reduce efficiency and power quality. Supercapacitors are proposed as a short-term energy source to supply pulse power loads in Y. Tang and A. Khaligh [47]. The combination of battery and supercapacitor is called a hybrid energy storage system (HESS) and is beneficial when controlling ships to mitigate thrust and torque fluctuation effects in the propulsion system. The benefit of the dynamic response from a FC-battery-supercapacitor hybrid system is crucial as neither the FC nor the battery can in real life handle the most abrupt load changes. However, as a simplification, this advantage is incorporated in our model by assuming the battery can react to sudden load fluctuations.

J. Hou and J. Sun [48] describes how land-based hybrid electric vehicles (HEV) deviate from ship propulsion systems by underlining three challenges unique to marine vessels:

- **Larger scale power systems.** Marine propulsion systems generally have a higher power rating, leading to differences in the optimal control configurations.
- **Multi-frequency.** Ships experience both slowly varying loads from wave-induced motion and rapidly varying loads and motions corresponding to the 1^{st}-order wave loads which induce oscillatory motions.

- **Environmental influence.** Environmental disturbances such as waves, currents, and wind have a much greater effect on ships compared to land-based vehicles. These disturbances also vary considerably, requiring ships to scale their systems thereafter.

When designing a shipboard power system, the effects of proper load management must be considered to determine the size of the power sources. As the power system assessed in this thesis consists of components with parameters from the industry, such sizing considerations are not investigated further.

4 Control problem formulation

The objective of the PEMS is to distribute the power demand from the operator to the different power sources. To optimize the performance, it is important to create a measure of how well the PEMS is performing, which is the purpose of the cost function. In this section, we will give an overview of suitable cost functions for the PEMS, which are used to formulate the objective function for the learning algorithms.

The cost function serves as an objective to minimize for the PEMS optimization. The costs are economically motivated. Environmental emissions can be included, but since a zero-emission ship with fuel cells and battery is studied, they are non-existent and will not be considered when formulating cost functions. The cost associated with a FC-battery system can be split into two groups; fuel consumption, which is the direct and immediate cost, and degradation cost, which shortens component lifetimes. The cost related to fuel consumption is relatively straight forward to calculate as it mainly depend on the fuel cell current.

The costs related to degradation are complex and requires significant computational power. The degradation process, which is a dominating cost factor for both FCs and batteries, consists of multiple intricate processes related to various components within the fuel cell and battery. As a result, an exact solution to the degradation impact is yet to be found.

Most of the research in this field only considers fuel consumption due to its simplicity compared to degradation issues. Fletcher et al. [49] argues that degradation significantly contributes to the operational running costs and should hence be included in the cost function. Other studies, by Li et al. [50] and Martel et al. [51], have taken degradation of energy sources into account by setting boundaries to the battery's SOC and the operational dynamics of the FC. This is not optimal, as it excludes real-time degradation effects [8].

The main factors of operational costs of hybrid energy systems boil down to how much energy it consumes and how properly each component is run. To minimize running costs the aim is therefore to use each system component close to its maximal efficiency, but within the range that does not accelerate degradation. The following sections describes a cost function that takes fuel consumption and component degradation into account and discusses each of the influential factors.

4.1 General cost function

A general cost function for the hybrid FC-battery system is proposed. The total cost function encapsulates costs related to fuel consumption and degradation of the battery and fuel cell, and is on the following form:

$$C_{total} = -(C_{fuel} + C_{FC,deg} + C_{bat,deg}) \tag{4.1.1}$$

4.2 Fuel cost

The cost related to fuel consumption is split between the hydrogen used by the FC and amount of hydrogen the FC uses to cover the internal power loss in the battery. The cost can be defined as:

$$C_{fuel} = C_{FC,fuel} + C_{bat,loss} \tag{4.2.1}$$

FC fuel consumption

Fuel cost is the most immediate cost and is relatively simple to calculate. The fuel cell uses hydrogen to directly power the propulsion system and to charge the battery. The total cost of hydrogen consumption can be calculated by multiplying the cost of hydrogen with the amount of hydrogen used. [52]:

$$C_{FC,fuel} = C_{H_2} \cdot H2_{cons} = C_{H_2} \cdot \frac{N}{F} \cdot I_{FC} \; \mathrm{d}t \tag{4.2.2}$$

where C_{H_2} is the price of fuel per kg, with unit \$/kg, $H2_{cons}$ is the total consumed hydrogen mass, N is the number of cells in the stack, F is the Faraday constant, I_{FC} is the FC current and dt is the time step used for the simulation. The total hydrogen consumption can be calculated by integrating the FC current over the entire driving cycle.

Today, several production methods are able to produce hydrogen to a cost of less than \$2/kg [53]. Furthermore, this production cost is expected to decrease in the years to come due to development in zero-emission technology. Nonetheless, the price of hydrogen, C_{H_2}, is assumed to be \$2/kg in further calculations.

Internal battery power loss

The battery's internal power loss can be translated to a cost by calculating how much fuel the FC uses to cover the loss. When the battery is running, some of the energy is lost in the process of charging/discharging. To account for this power loss, it is included in the cost function. This can be done using the following steps. First the the voltage drop in the battery is found. This is calculated as a product of the current running through the battery and the internal resistance. The loss of power in the battery can then be calculated as:

$$V_{bat,drop} = R_{bat} \cdot I_{bat} \rightarrow \Delta P_{bat} = R_{bat} \cdot I_{bat}^2$$

In order to calculate the cost of this loss, we have to calculate how much H_2 the fuel cell would have used in order to generate the power. Therefore, we consider this power loss as if it was generated by the fuel cell.

$$P_{bat} \cong P_{FC}$$

The FC current is needed to calculate the fuel consumption. This is easily found by dividing the FC power with the nominal FC voltage. The voltage of the FC varies, but in the long run, the nominal voltage should be close to the average operating power. This will be somewhat inaccurate, but is considered the most suitable way of translating the battery power into fuel consumption.

$$I_{FC} = \frac{P_{FC}}{V_{FC,nom}}$$

The equivalent fuel consumption is finally derived with the same logic as in Equation (4.2.2). Hence, the cost corresponding to the battery loss is:

$$C_{bat,loss} = C_{H_2} \cdot \frac{N}{F} \cdot \frac{R_{bat} \cdot I_{bat}^2}{V_{FC,nom}} \, \mathrm{d}t \tag{4.2.3}$$

4.3 Fuel cell cost

PEMS behavior influences the degradation of the fuel cell significantly, and the policy should therefore consider both fuel consumption and degradation costs. The PEMFC temperature is assumed to be within its operation limits, hence the effects from temperature variations are disregarded. According to Fletcher et al. [25], the FC operating conditions lead to considerable degradation effects include; *low power operations (idling)*, *high power operations*, *high power transients* and *start/stop cycles*. By assuming independent degradation mechanisms, the total fuel cell degradation cost can be summarized and included in the total fuel cell cost:

$$C_{FC,deg} = C_{FC}(D_{power,low} + D_{power,high} + D_{transients} + D_{cycle}) \tag{4.3.1}$$

C_{FC} is the fuel cell price, $D_{power,low}$ is the degradation due to low power, $D_{power,high}$ is the degradation due to high power, $D_{transients}$ is the degradation from change in FC power and D_{cycle} is the degradation from shutting down the FC. By assuming that the FC is always on during operation, and resorting to the FC running idle when the power demand is low, cycle degradation can be omitted. Seeing that the FC starts and stops once each operation, this degradation can be considered inevitable and is therefore excluded in the optimization. It should, however, be included for longer operations where shutting the engine on and off is an alternative, as the impact of each start/stop cycle heavily shortens the FC lifespan [54].

The terms $D_{power,low}$ and $D_{power,high}$ are the degradation costs as a result of idling and high power operation, respectively. If the power is below 10 % or above 80 % of the max FC power, P_{max}, the corresponding degradation is [25], [49]:

$$D_{power,low} = \begin{cases} \alpha_{low} \cdot \frac{0.1P_{max} - P_{FC}}{0.1P_{max}} \text{ d}t, & \text{if } P_{FC} < 0.1 \cdot P_{max} \\ 0, & \text{otherwise} \end{cases} \tag{4.3.2}$$

$$D_{power,high} = \begin{cases} \alpha_{high} \cdot \frac{P_{FC} - 0.8P_{max}}{0.2P_{max}} \text{ d}t, & \text{if } P_{FC} > 0.8 \cdot P_{max} \\ 0, & \text{otherwise} \end{cases} \tag{4.3.3}$$

α_{low} and α_{high} are the degradation rates for low and high operation condition, summarized in Table 4.1. dt is the time step used in the simulations. Equation (4.3.2) and (4.3.3) shows that the degradation cost is applied when above or below the set thresholds and increase linearly. α_{low} and α_{high} are in Fletcher et al. [25] applied as a constant value for the entire high and low power area. To make the penalty more realistic, we use a linear penalty. The average penalty of the linearized penalty is desired to equal the constant penalty, and therefore α_{low} and α_{high} is multiplied by a factor of 2.

$D_{transients}$ covers the FC degradation caused by transient loading. The degradation is modeled as a function of the power delivered by the FC:

$$D_{transients} = f\left(\frac{\text{d}P_{FC}}{\text{d}t}\right) = \beta \cdot \left|\frac{\text{d}P_{FC}}{\text{d}t}\right| \text{ d}t = \beta \cdot |\text{d}P_{FC}| \tag{4.3.4}$$

β is the degradation rate from Table 4.1. The transient degradation is modeled as proportional to the rate of change in the power provided by the fuel cell. Since the penalty is applied every time step, it has to be weighted by the time increment, and the penalty thus becomes only a function of the change in power. Avoiding high transients is important for maintaining a stable temperature and humidity in the cell, which prevents local fuel starvation.

Table 4.1: Scaled PEMFC degradation rates [25]

PEM fuel cell degradation rates (per cell)

Symbol	Operating condition	Degradation rate
α_{low}	Low power operation	20.34 μV/h
α_{high}	High power operation	23.48 μV/h
β	Transient loading	0.0441 μV/Δkw

The degradation factors are summarized in Table 4.1. The values are given per cell, and the fuel cell considered has 900 cells. Therefore the values are scaled up with a factor of 900. The nominal voltage of our fuel cell is defined at 50 % of maximum power, where the voltage is 629 V. EOL for the fuel cell is defined as when the voltage drops by 10 %. Thus, the voltage causing the fuel cell to reach EOL is:

$$629\ V \cdot 10\ \% = 62.9\ V \tag{4.3.5}$$

This means that the cost of a voltage drop of 62.9 V imply the same operating cost as the entire fuel cell.

In order to translate the degradation rate values into actual costs in dollar, the total cost of the FC is needed. The FC used in our experiments has a rated power of 120 kW. Using the fuel cell cost of \$45 /kW, seen in Table 2.1, the total cost can be calculated.

$$C_{FC} = \$45/\text{kW} \cdot 120\ \text{kW} = \$5400 \tag{4.3.6}$$

The degradation rates in dollar values can now be calculated. This is done by first scaling up the cell degradation rates by 900, the number of cells, before dividing all the numbers by 62.9 V. This gives the fraction of total degradation they contributes to. Then, the result is multiplied by the fuel cell cost of \$5400 in order to get the actual costs. The resulting degradation cost rates are as follows:

Table 4.2: PEMFC degradation rates scaled to U.S. dollars [25]

PEM fuel cell degradation cost rates (per cell)

Symbol	Operating condition	Degradation rate
$\alpha_{low,\$}$	Low power operation	1.57 \$/h
$\alpha_{high,\$}$	High power operation	1.81 \$/h
$\beta_{\$}$	Transient loading	0.0034 \$/ΔkW

4.4 Battery cost

Battery costs primarily consist of the price of degradation and the energy it consumes. By constraining state variables, accelerated aging can be limited. Maintenance costs are not included and considered to be outside the scope of this work. Both time degradation and usage contributes to battery degradation, both of which are comprehensively described below.

Degradation cost of batteries are studied in depth, and there has been conducted a vast amount of research in the field. Generally, the rate of deterioration for a battery can be divided into two components; cycle-life and calendar-life [55]. Cycle-life is reduction in battery performance due to cycling processes, whereas calendar-life deterioration is the inherent battery aging and a function of time only. Hence, the cost related to calendar aging is not included in the model, as it adds unnecessary complexity.

The total cost of running the battery can be modeled as:

$$C_{bat,deg} = C_{bat}(D_{SOC} + D_{DOD}) \tag{4.4.1}$$

where C_{bat} is the price of the lithium-ion battery, D_{SOC} is the degradation from SOC wear and D_{DOD} is the degradation related to the depth of discharge stress which will be defined later.

4.4.1 State of charge

There have been multiple efforts of implementing a health-aware PEMS that accounts for the cost of battery degradation, without the usage of a chemical model. Note that these methods will indirectly affect the chemical degradation reactions in the cell. A common strategy is to put bounds on the SOC in both ends, as operating above or below these levels are a major cause of battery degradation. However, this is very simplified, and will not yield any optimal behavior concerning battery lifetime. Another common, more sophisticated way of considering battery lifetime, is to incorporate a cost proportional to the quadratic deviation in SOC from SOC$_{ref}$, where SOC$_{ref}$ is the operating SOC level that minimizes degradation. Thus, deviation from the SOC reference point is penalized for both low and high values, with a penalty that increase with the distance from the reference SOC. The SOC related degradation $D_{SOC,quadratic}$ is commonly written as a penalty proportional to the following expression:

$$D_{SOC,quadratic} = \gamma \cdot (SOC_{ref} - SOC(t))^2 \ \mathrm{d}t \tag{4.4.2}$$

where γ is the degradation rate given in $\$/s$. The issue with this penalty is that, to the authors knowledge, γ has yet to be estimated in a precise way. Most likely, this is because battery degradation is a lot more complex than this equation suggest. There are a lot of research on battery degradation in the literature. Most of the used models are complex, incorporating degradation factors such as cycle depth, temperature, current rate and average state of charge [28]. Complex battery degradation models are considered outside the scope of this work. However, there is a clear consensus in the literature that high and low SOC is damaging the battery health. As a result, we have applied a significant penalty if the SOC goes above 0.7 or below 0.3. This leads to the battery avoiding high and low SOC whenever possible, thus acting as a soft constraint on the SOC. The implementation of this penalty does not significantly hinder the functionality of the battery.

$$D_{SOC} = \begin{cases} 1 \cdot \mathrm{d}t, & \text{if } SOC < 0.3 \\ 1 \cdot \mathrm{d}t, & \text{if } SOC > 0.7 \\ 0, & \text{otherwise} \end{cases} \tag{4.4.3}$$

The number 1 is arbitrarily, but its magnitude is sufficient to serve its purpose when applied at each time step.

Depth of discharge

Depth of discharge is defined as how deep a continuous cycle where the battery charges/discharges is. It consists of two parameters, DOD$_{charge}$ and DOD$_{discharge}$, that define how much the battery has charged or discharged without interruption. The evolution of DOD$_{charge}$ and DOD$_{discharge}$ under an example battery load is visualized in Figure 4.1. Calculating the battery cost related to depth of discharge is non-trivial. Koller et al. [34] suggested a model for battery degradation from DOD stress. The stress models are obtained either empirically or derived theoretically. The degradation has a nonlinear impact on the battery, which can be challenging to estimate. Xu et al. [28] argues that the models used in the literature do not give an adequate representation of the battery degradation.

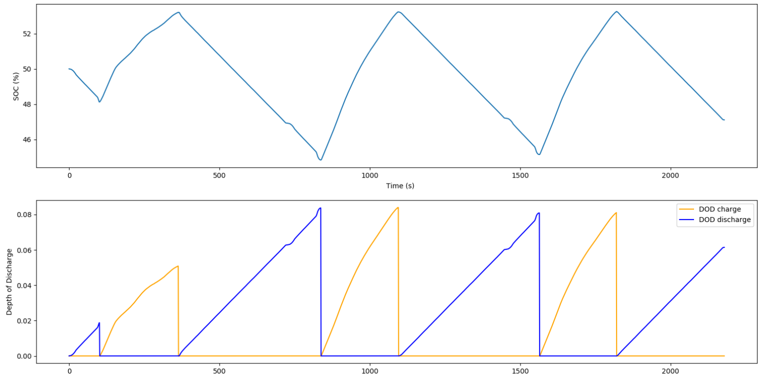


Figure 4.1: Evolution of depth of discharge for an arbitrarily load profile [34]

In recent years, however, there have been more focus on developing good battery degradation models for PEMS optimization. Wang et al. [56] performs test on $LiFePO_4$, a popular type of lithium-ion battery, under different temperatures, DOD and fixed C-rates in order find a function that estimates the battery degradation from such parameters. The authors experimented with cyclic behavior on the batteries until EOL, while testing the capacity of the batteries at fixed time intervals to measure the SOC. The study, in contrast to many other studies, concluded that the battery DOD had only a minor impact on battery degradation for low C-rates. In addition, instead of using time as a parameter, they switched it to Ah-throughput Ah_{th}. Ah_{th} is the total amount of current that has left or entered the battery, and is therefore directly proportional to the time at a fixed C-rate. C-rate is a metric for how quick a battery is charged or discharged, relative to it's maximum capacity. As an example, a C-rate of 1 indicates that the battery will fully discharge in 1 hour given that the battery current stays the same. After several experiments and curve fitting for the parameters, the following capacity loss estimate was found [56]:

$$Q_{loss} = A \cdot \exp\left[\frac{-E_a + B \cdot C_{rate}}{RT}\right] \cdot (Ah_{th})^z \tag{4.4.4}$$

where A is the pre-exponential factor, E_a is the activation energy of the $LiFePO_4$ battery examined, B is the exponential factor weighting C-rate properly, and z is a factor for how much to emphasis the effect from the Ah-throughput.

In an attempt to quantify the effect from DOD and C-rate on battery degradation, Chen et al. [57] uses the result in Equation (4.4.4) to model the capacity loss in the battery as a function of DOD and C-rate. First, they utilize the fact that Ah_{th} can be calculated in the following way:

$$Ah_{th} = Q_{full} \cdot DOD \cdot N \tag{4.4.5}$$

where N is the amount of cycles. This way, the number of cycles the battery can sustain before EOL, with a given DOD and C-rate, is quantified by rearranging and combining Equation (4.4.4) and Equation (4.4.5).

$$N(DOD, C_{rate}) = \left[\frac{Q_{loss}}{A \cdot e^{\left(\frac{-E_a + B \cdot C_{rate}}{RT}\right)}}\right]^{\frac{1}{z}} \cdot \frac{1}{Q_{full} \cdot DOD} \tag{4.4.6}$$

Q_{loss} is the amount of capacity allowed before the battery is considered to have reached EOL. It should be commented that two simplifications have been made to use this equation. The original formula from Chen et al. [57] assumes a constant C_{rate}. This will rarely be the case for an online PEMS where a varying power demand will cause the C-rate to fluctuate. As a result, we have applied the average C-rate in a given charge/discharge cycle in the equation above. Furthermore, the pre-exponential factor A varies to some degree with different C-rates. A is set equal to the C-rate corresponding to 2 C, which is considered a typical C-rate for the battery in our application. The relevant constants in the equation are given in Table 4.3, and the log of Equation (4.4.6) is plotted in Figure 4.2.

Table 4.3: DOD degradation parameters

Lithium-ion battery DOD degradation parameters

Symbol	Description	Value
Q_{loss}	Maximum allowed capacity loss	2.355 kWh
Q_{full}	Maximum capacity	17.748 kWh
E_a	Activation energy	31.500 J/mol
A	Pre-exponential factor	19.300 kWh
B	Exponential effect of C_{rate}	370.3 J/(A · mol)
z	Power law factor	0.55
R	The gas constant	8.314 J/(K · mol)
T	Battery cell temperature	298.15 K

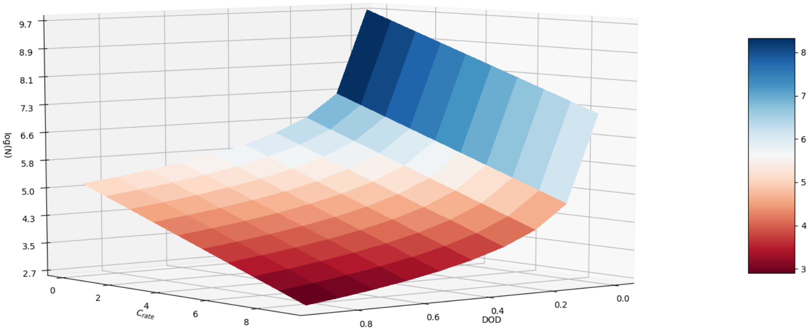


Figure 4.2: DOD and C-rate's effect on total remaining battery cycles

As Figure 4.2 displays, the remaining number of cycles of the battery decreases significantly with higher C-rates and DOD. Also, both the C-rate and DOD have an distinct impact on the battery life. Therefore, the intuition is that the battery lifetime is extended when it is operated at lower DOD and C-rate levels.

To calculate the actual cost of using the battery, we have to know the total battery cost. When at 50 % SOC, the battery has a voltage of 493 V, and the battery capacity is set to 36 Ah for our simulations. Thus, the maximum amount of kWh delivered can be estimated to:

$$Q_{full} = \frac{493 \text{ V} \cdot 36 \text{ Ah}}{1000} = 17.748 \text{ kWh} \tag{4.4.7}$$

The price for lithium-ion batteries was in 2018 estimated to be \$176/kWh by Bloomberg, a research firm that covers the clean energy industry [41]. As a result, the total battery cost can be calculated as follows:

$$C_{bat} = 17.748 \text{ kWh} \cdot \frac{\$176}{\text{kWh}} = \$3124 \tag{4.4.8}$$

The cost of half a charge/discharge cycle, given an average C-rate is then calculated by dividing C_{bat} with the total number of cycles the battery could perform before EOL. The battery degradation cost related to DOD and C-rate becomes:

$$D_{DOD} = \frac{1}{2 \cdot N(DOD, C_{rate})} \tag{4.4.9}$$

This cost is applied whenever the depth of discharge resets. The C-rate is the average discharge/charge current for the given cycle, and DOD is the depth of the cycle. The reason for dividing with 2 is that the penalty is applied for every half cycle. As a result, is has to be divided by two to give the cost as if it was an entire cycle.

4.5 Cost Optimization

Control strategies are compared on total the calculated cost during testing. A complete cost function containing the factors introduced above can be derived from the general cost function. Equation (4.5.1) again presents the form of the total cost function.

$$C_{total} = -(C_{fuel} + C_{FC,deg} + C_{bat,deg}) \tag{4.5.1}$$

Equations (4.5.2) to (4.5.4) shows the finalized cost terms.

$$C_{fuel} = C_{FC,fuel} + C_{bat,loss} \tag{4.5.2}$$

$$C_{FC,deg} = C_{FC}(D_{power,low} + D_{power,high} + D_{transients}) \tag{4.5.3}$$

$$C_{bat,deg} = C_{bat}(D_{SOC} + D_{DOD}) \tag{4.5.4}$$

Table 4.4 presents each addend of the finalized cost function applied in the simulations:

Table 4.4: Cost function

Complete cost function breakdown

Notation		**Equations**	
$C_{FC,fuel}$	$=$	$C_{H_2} \cdot \frac{N}{F} \cdot I_{FC}\ \mathrm{d}t$	
$C_{bat,loss}$	$=$	$C_{H_2} \cdot \frac{N}{F} \cdot \frac{R_{bat} \cdot I_{bat}^2}{V_{FC,nom}}\ \mathrm{d}t$	
$D_{power,low}$	$=$	$C_{FC}(\alpha_{low,\$} \cdot \frac{0.1P_{max} - P_{FC}}{0.1P_{max}}\ \mathrm{d}t),$	if $P_{FC} < 0.1 \cdot P_{max}$
$D_{power,high}$	$=$	$C_{FC}(\alpha_{high,\$} \cdot \frac{P_{FC} - 0.8P_{max}}{0.2P_{max}}\ \mathrm{d}t),$	if $P_{FC} > 0.8 \cdot P_{max}$
$D_{transients}$	$=$	$C_{FC} \cdot \beta_{\$}\ \lvert \mathrm{d}P_{FC} \rvert$	
D_{SOC}	$=$	$C_{bat} \cdot \frac{1}{C_{bat}} \cdot \mathrm{d}t,$	if $SOC < 0.3 \vee SOC > 0.7$
D_{DOD}	$=$	$C_{bat} \cdot \Big(2 \cdot N(DOD, C_{rate})\Big)^{-1}$	

5 Ship Power System Model

There are multiple ways of representing both the fuel cell, battery and PEMS mathematically. A substantial number of models have been proposed in the literature, varying from simple to extremely complex. The latter is obviously more precise. For our purposes, generating massive amount of data is crucial for the algorithms to learn patterns. Therefore, decreasing computation time has been a priority. Furthermore, in-depth knowledge in electrochemical modeling is required in order to investigate the more complex models. On the other hand, an inaccurate model is essentially useless. Thus, modeling the system components is a trade-off that has to be considered carefully by the control designer.

In the following sections the implemented fuel cell and battery model are presented, including the equations and relevant graphs describing how energy is drained and how power is produced. The aim is also to describe the basic internal characteristics of both the fuel cell and battery. Subsequently the PEMS including its system architecture is described. The models for battery, fuel cell and PEMS are all implemented from scratch by the authors in Python.

5.1 Fuel cell model

The PEMFC model is based on the generic model created by Motapon et al. [58]. A simplified version of the model is shown in Figure 5.1 and depicts a fuel cell stack as a controlled voltage source, E, in series with an internal FC resistance, R_{ohm}.

The output FC power is calculated by modifying Ohm's law:

$$P_{FC} = i_{FC} \cdot V_{FC} \tag{5.1.1}$$

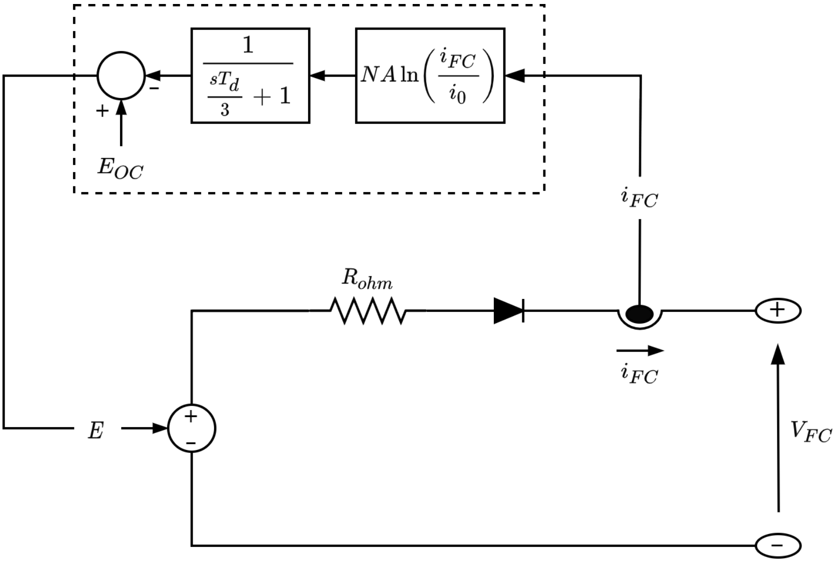


Figure 5.1: A simplified fuel cell model [58]

In order to formulate a fuel cell model, it is useful to make some assumptions to simplify. In the model to be described, the following is assumed to be true [11]:

- The stack temperature and humidity are constant during operation.
- Hydrogen and air supplied to the stack are considered ideal.
- The only losses are activation losses, which are linear.
- Pressure drops inside the stack are not considered.

A polarization curve is a common tool to describe fuel cell characteristics. It is a graph showing the relation between current density and voltage output of the FC. In order to find the relevant parameters for our model, the fuel cell's polarization curve is used.

Figure 5.2 displays a generic polarization curve. The plot shows three distinct areas based on the FC current density. In the activation and mass transport region the cell voltage drops nonlinearly, whereas the Ohmic region experiences a roughly linear drop. The regional differences come from internal losses that originates from activation losses, Ohmic losses, and concentration losses, respectively. Modeling each of the losses separately is considered irrelevant for this work, as it would add too much complexity to the model. The fuel cell parameters determine the size of each region, and the polarization curve can be obtained from the manufacturer.

The controlled voltage source of the fuel cell is described as follows [11], [58]:

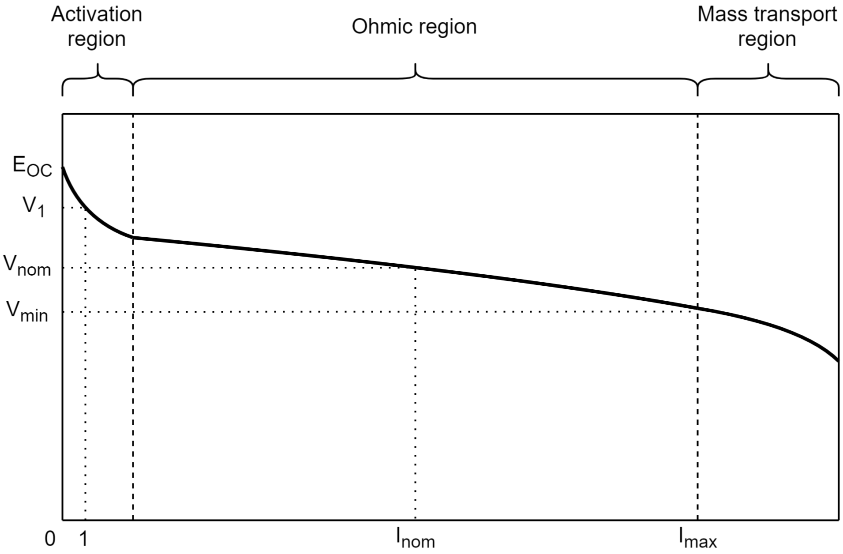


Figure 5.2: A generic PEMFC polarization curve [59]

$$E = E_{OC} - \overbrace{NA\ln\left(\frac{i_{FC}}{i_0}\right)}^{\text{Activation loss}} \tag{5.1.2}$$

where E_{OC} is the open circuit voltage, N is the number of cells, A is the Tafel slope, i_{FC} is the FC current and i_0 is the exchange current.

The open circuit voltage is obtained by the Nernst equation, and is affected by the temperature, partial pressures of hydrogen and air, as well as their concentrations. Furthermore, the remaining values are found [58]:

$$\begin{aligned}
NA =& \frac{(V_1 - V_{nom})(i_{max} - 1) - (V_1 - V_{min})(i_{nom} - 1)}{\ln(i_{nom})(i_{max} - 1) - \ln(i_{max})(i_{nom} - 1)} \\
R_{ohm} =& \frac{V_1 - V_{nom} - NA\ln(i_{nom})}{i_{nom} - 1} \\
i_0 =& \exp\left(\frac{V_1 - E_{OC} + R_{ohm}}{NA}\right)
\end{aligned} \tag{5.1.3}$$

R_{ohm} is the internal fuel cell resistance. E_{OC} is the voltage at 0 A, while V_1 equals the voltage at 1 A. V_{nom} and i_{nom} are the voltage and current at nominal operation point. V_{min} and i_{max} are the voltage and current at maximum operation.

The values can be found by examining four points on the polarization curve in Figure 5.2 as described by Motapon et al. [58]. The FC output voltage, V_{FC}, can then be calculated as:

$$V_{FC} = E - \overbrace{R_{ohm} \cdot i_{FC}}^{\text{Ohmic loss}} \tag{5.1.4}$$

A simplified version of the generic polarization curve can be created by linearizing it. Both the activation and Ohmic region are piecewise linearized. The mass transport area is neglected by constraining the max FC current density at I_{max}. This is considered a valid constraint, as the FC efficiency drops drastically in the region and degradation rates are high. Thus, an intelligent controller would rarely command the fuel cell to operate at a such high current. The resulting linearized polarization curve is displayed in Figure 5.3.

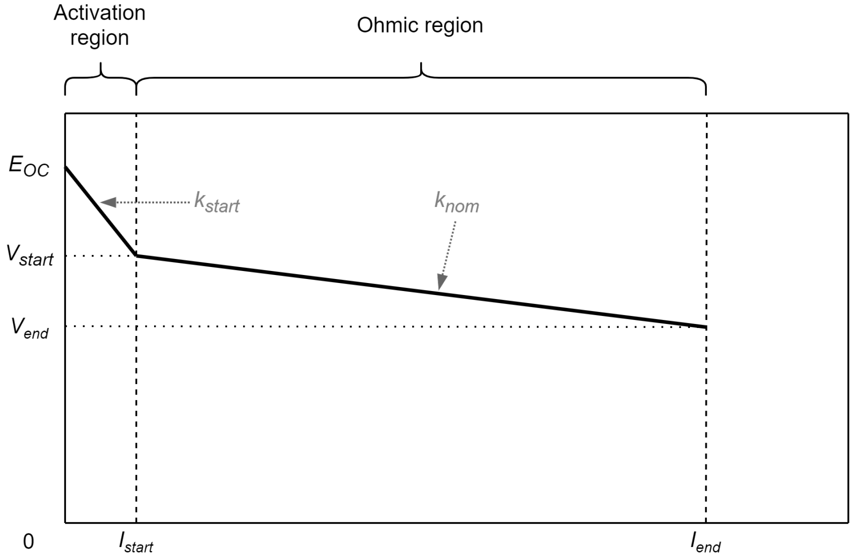


Figure 5.3: Linearized PEMFC polarization curve

From the polarization curve of a the PEMFC given in the Simulink documentation, the following parameters are collected in Table 5.1 [59]:

An alternative to using Nernst equation to calculate the controlled cell potential by using Equation (5.1.2), is graphically using the slope from Figure 5.3, k_{start} and k_{nom}.

Table 5.1: FC parameters from polarization curve [59]

PEMFC

Parameter	Value	Description
E_{OC}	900 V	Open circuit voltage
V_{start}	800 V	Voltage at start of Ohmic region
V_{end}	430 V	Voltage at end of Ohmic region
I_{start}	20 A	Current at start of Ohmic region
I_{end}	280 A	Current at end of Ohmic region

$$k_{start} = \frac{E_{OC} - V_{end}}{I_{start}} \tag{5.1.5}$$

$$k_{nom} = \frac{V_{start} - V_{end}}{I_{max} - I_{start}} \tag{5.1.6}$$

The current cell voltage, V_{FC}, is then determined by the current, I, the following way:

$$V_{FC} = \begin{cases} V_{start} - k_{nom} \cdot (I - I_{start}), & \text{if} \quad I > I_{start} \\ E_{OC} - k_{start} \cdot I, & \text{otherwise} \end{cases} \tag{5.1.7}$$

The output FC power is finally calculated the same way as in Equation (5.1.1):

$$P_{FC} = I_{FC} \cdot V_{FC} \tag{5.1.8}$$

Thus, the power produced from the fuel cell can be plotted against the current, as seen in Figure 5.4.

It is important to consider the efficiency when operating an FC. Higher efficiency means that more energy is converted from hydrogen to power. Therefore, operating at higher efficiency implies less fuel consumption. Fuel cells have an overall efficiency superior to conventional engines due to their direct energy conversion without combustion [11], [60]. However, the efficiency varies a lot with respect to the operating power of the fuel cell.

In the FC model, the fuel consumption is calculated using Equation (4.2.2). Because fuel consumption is proportional to the FC current, the efficiency of the model is highest at lower currents, since more power is produced per unit of current. This, however, is not an entirely correct representation of a real fuel cell. Typically, the FC efficiency is low at low current, before it quickly increases with increased current. As a result, the model we use is not entirely accurate for low

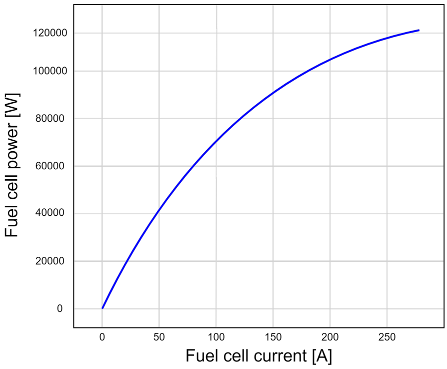


Figure 5.4: FC current vs. power

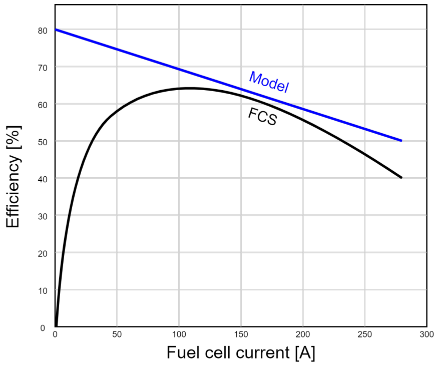


Figure 5.5: FC current vs. efficiency [11]

current levels. The modeled efficiency (Model) vs. the actual efficiency of a typical PEM fuel cell system (FCS) is plotted in Figure 5.5 [11].

To model the internal FC delay, the rate of change in current is constrained to 10 % of I_{end} per second. Essentially this constraint encapsulates the dynamic capabilities of the FC. The hydrogen flow cannot instantaneously increase more than a given amount. By limiting the rate of current, the model is able to account for the rather slow FC dynamics.

$$|\Delta I_{FC}| \leq 0.1\ I_{FC,max} \cdot \mathrm{d}t \tag{5.1.9}$$

5.2 Battery model

The battery model we have implemented is a simplified version of the one proposed by Tremblay and Dessaint [61], which is also applied by MATLAB in its example battery model [62].

In Figure 5.6 the battery characteristics from Tremblay and Dessaint are visualized by plotting the battery voltage as a function of capacity. In the exponential zone, the battery has high state of charge, and the battery voltage grows exponentially as the battery is charged. In the nominal zone, the voltage is decreasing slowly as the battery is discharged. This is where the battery thrives, and is the best range to operate the battery in order to prolong lifetime. When the SOC falls below Q_{nom} the battery voltage decreases exponentially as the charge approaches 0, where the battery is completely discharged.

The model includes several assumptions [62]:

- The internal resistance is constant for both charge and discharge cycles. It is not affected by current variations.
- Charge and discharge characteristics are assumed to be identical.

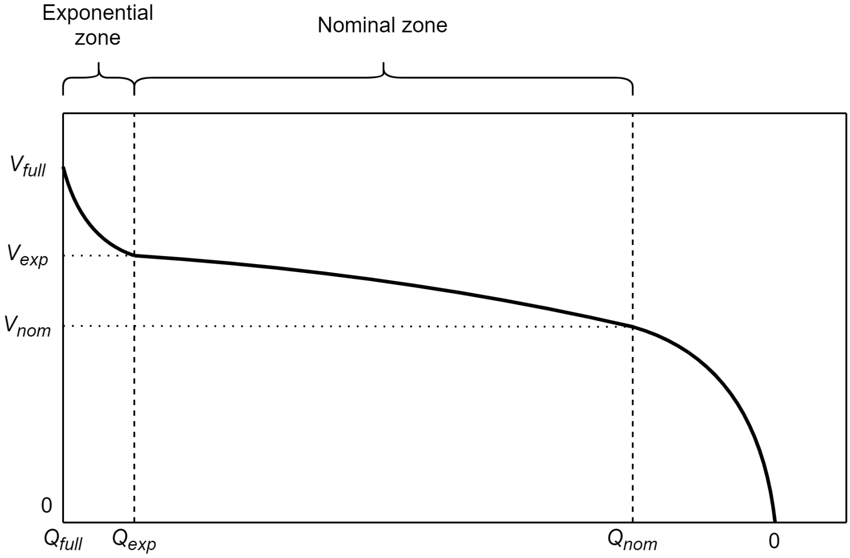


Figure 5.6: Battery characteristics

- The battery capacity is not dependent on the current amplitude (Peukert effect is neglected).
- There is no internal memory effect in the battery.
- The behavior of the battery is independent of internal temperature.
- Battery self-discharge is neglected.

All assumptions aim to simplify the model computationally. Two of the limitations that follows, in addition to reduced model accuracy, are that the battery is unable to overcharge and the minimum no-load voltage is 0 V. In tests performed on real batteries, the proposed model was almost identical for steady state validation for several C-rates. Accuracy tests for dynamic models, where both the SOC and the current varies, the model proves to give the SOC with an error margin of ±5 % of the actual SOC for relevant values (SOC $\geq$ 20 %) in all the test conducted [62].

As discussed, computational efficiency is key in order to generate sufficient data for training the machine learning algorithms. Therefore, the model we have applied for training the algorithm is a simplified version of the one described above. The battery is assumed to exclusively operate in the nominal zone, and thus linearize voltage as a function of the state of charge. The linearized battery characteristics is shown in Figure 5.7.

The linearization introduces new limitations in the battery model. Namely that the SOC is unable to fall below Q_{min} and rise above Q_{max}. This is considered fair, as batteries should not be operated at very high or very low SOC, due to the significant

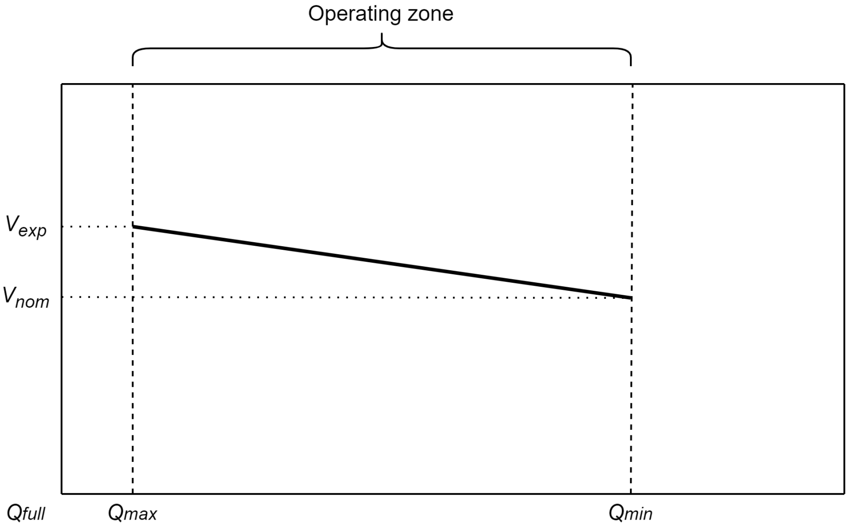


Figure 5.7: Linearized battery characteristics

battery degradation it causes. The relation between SOC and open circuit voltage, E_0, is found from the linearized battery characteristics curve. Figure 5.7.

$$E_0 = V_{exp} - \frac{V_{exp} - V_{nom}}{Q_{max} - Q_{min}} \cdot (Q - Q_{min}) \tag{5.2.1}$$

The battery output voltage is calculated by subtracting the battery current with the internal resistance. Thus, the output battery voltage is modeled in the following way:

$$V_{bat} = E_0 - R_{internal} \cdot i_{bat} \tag{5.2.2}$$

Then, the output power from the battery is calculated as follows:

$$P_{bat} = V_{bat} \cdot i_{bat} \tag{5.2.3}$$

The variation in SOC is based on simple calculation on how much current enters or leaves the battery. The resulting model for change in capacity Q is as follows:

$$Q(t) = Q(0) - \int_0^t i_{bat} \mathrm{d}t \tag{5.2.4}$$

The capacity Q of the battery is given in Ah, and as a result, the SOC is updated in the following way at each time step of model simulation:

$$SOC(t + \mathrm{d}t) = SOC(t) - \frac{i(t) \cdot \mathrm{d}t}{3600 \cdot Q_{full}} \tag{5.2.5}$$

The relevant parameter values for the battery model used for training is given in Table 5.2.

Table 5.2: Battery model parameters

Lithium-ion battery

Parameter	Value	Description
Q_{min}	6.6 Ah	Minimal battery capacity
Q_{max}	32 Ah	Maximal battery capacity
V_{exp}	545 V	Voltage at end of exponential zone
V_{nom}	430 V	Voltage at end of nominal zone
$SOC(0)$	0.5	Initial SOC level

5.3 Power and energy management system

The energy management system serves the task of splitting the required power between the battery and the fuel cell. This can be done using a huge variety of algorithms, some of which are implemented and described in depth in the following chapter. The general purpose of the algorithms is to reduce the combined cost of the fuel cell, battery and fuel cost, as described in the previous subsections.

There are multiple ways of designing the PEMS, all having their advantages and disadvantages. The PEMS can take in different state variables in order to decide the action, and the actions made to control the system can also vary. As an example, the action can either be to control the current of the FC or the power of the FC. There is no right way of designing the controller output/input. What matters is the result in terms of minimizing the operating cost.

Figure 5.8 shows how the energy management system for our machine learning based algorithms work. The input to the energy management system is the Battery SOC, the FC current and the load currently required from the ship. One of the algorithms (SAC, discussed in Chapter 6), also has DOD variables as input. After calculations executed depending on the algorithm, the PEMS outputs the change in FC current demanded. The power and loss of the FC is then calculated, before it outputs the updated current and the power it produces. The current is used as input for the PEMS and the power signal from the FC is subtracted from the total power required. The remaining power requirement is served by the battery, which is assumed to be able to instantaneously serve the remaining load required. When the demanded battery power is negative, the surplus power is used to charge the battery. In the end, the battery SOC, and DOD variables are sent back to the PEMS for new calculations, and the process repeats until termination.

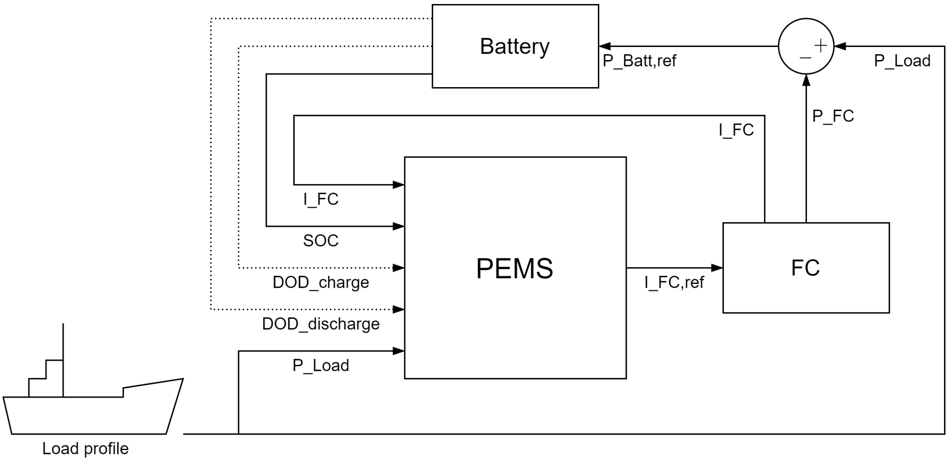


Figure 5.8: Power and energy management system

6 Control strategies

This section presents the relevant theory and discusses the PEMS algorithms implemented in Python by the authors.

6.1 Benchmark methods

In this section, the benchmark control algorithms implemented in code by the authors will be covered in depth. A literature review on both rule based control algorithm and dynamic programming is conducted, together with the relevant theory for both algorithms is presented along the way. The expected performance, as well as the strengths and weaknesses of both algorithms will be assessed.

6.1.1 Rule-based

Rule-based control of the PEMS is widely applied in the industry due to its robustness and stability. The strategy is based on a group of `if-then` statements considering PEMS input parameters, forming a decision tree. If the `if`-statement holds, then that part of the rule refers to a corresponding output policy of the controller. This applies to all `if`-statements in the algorithm. The output of the tree is usually the power set-point of the different energy sources. The implementation of such rules is straightforward. It is easy to understand and can be expressed in natural language. Furthermore, it doesn't need any model to perform well, and is very efficient computationally. However, to design a system that performs well, human domain expertise is required. The knowledge of effects such as the engine components operating efficiency and degradation factors should be utilized to design a PEMS with satisfying performance. It is also possible to combine it with other sophisticated control strategies for better functionality. Rule-based energy management is also referred to as fuzzy logic (FL) control EMS in the literature.

Table 6.1 shows a simple example of how a rule-based PEMS algorithm works. The HEV proposed in Zhu et al. [63] consists of an ICE and a battery. Based on the SOC of the battery and the power demand from the operator, the distribution in power between the ICE and battery is determined. The report splits the decision into five different operating modes on the battery; positive large, positive small, zero, negative small and negative large. Negative battery power means that the battery is charging. The remaining load is provided by the ICE.

Table 6.1: Rule-based control strategy [63]

Rule-based control strategy for a HEV

Premise		**Consequence**
Requested power	Battery SOC	Battery power
Large	High	Positive large
Large	Medium	Positive small
Large	Low	Zero
Medium	High	Positive small
Medium	Medium	Zero
Medium	Low	Negative small
Small	High	Zero
Small	Medium	Negative small
Small	Low	Negative large

Despite the advantages of rule-based control, it has several limitations. Firstly, there is no optimization or intelligence built into the system. If the system component's performance is changed due to factors as wear and tear, the optimal policy might change significantly. For fuzzy logic systems, the logic is programmed into the PEMS and remains unaltered until changed by a human operator. Thus it is unable to adapt to changes in the performance of system components. Furthermore, the discretization of action space implies that the actions will be suboptimal for major parts of the continuous state space. As a result, there are often other control techniques that yield better results in terms of optimal cost-efficiency.

6.1.2 Dynamic programming

Finite Markov decision process

The basis of the dynamic programming (DP) is that the environment you operate in is modeled as a Markov decision process (MDP). A MDP consists of functions that models the dynamics of how an agent interacts with an environment. The agent is the decision maker and performs actions in the environment. The environment is what the agent interacts with, and consist of everything except for the agent. MDP also contributes with rewards to the agent; special numeric values that the agent aims to maximize in the long run by clever action choices. The agent is always in a state of the environment. A state can be thought of as a signal the agent has access to, which influences the next action to be taken by the agent. When the agent performs an action A_t, the state S_t of the agent changes to a new state S_{t+1} and the agent receives a reward R_{t+1}. Then, being in the next state, the process is repeated, until the agent ends up in a terminal state. Thus, the agent and MDP gives a sequence of states, actions, and rewards in the following way:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ..., S_{t-1}, A_{t-1}, R_t, S_t \quad [64] \tag{6.1.1}$$

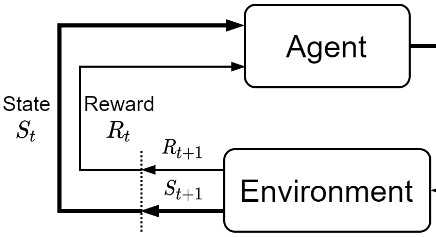


Figure 6.1: The agent-environment interaction in a Markov decision process [64]

Figure 6.1 illustrates the described agent-environment interaction. When performing action A_t in state S_t, the next state S_{t+1} is determined by a function p, which describes the environment and the dynamics of the MDP. The function p defines the probability that being in state s and performing a will result in next state s' and reward r. This probability is defined for $\forall\ s, s' \in \mathcal{S}$, $\forall\ r \in \mathcal{R}$ and $\forall\ a \in \mathcal{A}(s)$, where $\mathcal{S}$ is the set of all states in the MDP, $\mathcal{R}$ is the set of all rewards and $\mathcal{A}(s)$ is the set of all possible actions in state s. Formally, p can be written in the following way:

$$p(s', r \mid s, a) = Pr(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a) \tag{6.1.2}$$

where Pr is the conditional probability of going to state s' and receiving reward r when being in state s and performing action a. Thus p describes the probability distribution of the outcome of performing action a in state s, which means that the total probability of all possible outcomes equals one:

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s) \tag{6.1.3}$$

Optimal policy and optimal policy value function

As previously stated, the agent aims to maximize the total accumulated reward. If you have access to the finite MDP of a process, the optimal policy can be determined.

A policy π is defined as a mapping from state to action, meaning that if you are in state s, you will follow action a with probability $\pi(a \mid s)$. An optimal policy

π^* is a policy which maximizes the expected reward the agent receives over time. Note that there might be multiple optimal policies, as different optimal policies may yield the same expected accumulated reward over time. A value function v_π is a mapping from state to the expected value of all future rewards, when following a policy π. The value function can thus be interpreted as a performance metric for a corresponding policy, and is defined as follows:

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n R_{t+n+1} \mid S_t = s \right] \tag{6.1.4}$$

In the equation above R_{t+n+1} are the instantaneous rewards at step n when following policy π. The optimal policy π^* will have a value function that is larger or equal to the value functions of all other policies. The value function of the optimal policy is given as [64]:

$$v^*(s) = \max_\pi v_\pi(s), \forall s \in \mathcal{S} \tag{6.1.5}$$

Dynamic programming is a group of algorithms that can be applied to find the optimal policy, given an MDP. The core equation of the algorithm is the Bellman equation [65]. It states that the value of a state, given an optimal policy, must be equal to the expected return for best action from that state, and can be formalized in the following way.

$$v^*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v^*(s')] \tag{6.1.6}$$

This means that the optimal value of a state is the value corresponding to the action that yields the highest reward and future values, weighted with the transition probability p. The emphasis put on future rewards is determined by the discount factor γ, where $\gamma = 1$ means that future rewards are weighted equally as immediate rewards. DP builds on the principle of optimality: An optimal policy has the property that whatever the initial state and initial decisions (actions) are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions. This means that the optimal policy in a state s is built on other optimal policies as well, which implies that solution to the problem consist of solutions to subproblems and can be solved recursively. This is the fundamental principle of dynamic programming [65].

Dynamic programming for PEMS control

There are several examples of dynamic programming as a PEMS algorithm. Given a known driving cycle and a predefined reward function that reflects the actual cost of fuel and other cost related to degradation, the dynamic programming PEMS is guaranteed to obtain optimal control for the system. Because of this guaranteed optimal control policy, dynamic programming is often used as benchmark for other PEMS algorithms to evaluate their performance [66].

However, DP PEMS has several limitations. First of all, the requirement of the known driving cycle in advance is very limiting for real-time control. For marine applications, this is never the case due to the stochastic nature of the ocean. Even though predefined routes could give an estimate of the driving cycle, knowing the exact power demand in advance is simply impossible. On the contrary, dynamic programming on a tramway PEMS have proved to yield good results as the driving cycle is somewhat known in advance [67]. The second primary limitation that comes with DP PEMS is computational time. For longer driving cycles and big state spaces, it can be computationally infeasible in some cases. Dynamic programming suffers from the curse of dimensionality, which states that the computational time that is required to estimate a function grows exponentially with the number of states [68]. As a result, the number of parameters included in such an optimization is limited, which can affect the accuracy.

In Kalikatzarakis et al. [69], dynamic programming is used as a benchmark for the real-time PEMS strategies on a ship. It has access to the a priori knowledge of the operating profile of the ship, and thus the global optimum solution is obtained. Two equivalent cost minimization strategies (ECMS) and a rule-based control strategy are compared with the dynamic programming approach. The report concludes that DP is great for analyzing the dynamic performance and determining whether hybrid propulsion and power generation can reduce the wear on engine due to thermal loading and improve acceleration times.

Moura et al. [70] suggests a stochastic dynamic programming approach for a PEMS on a hybrid electrical vehicle with a lithium-ion battery pack. It considers both the SOH of the battery through electrochemical modeling and energy consumption cost with the aim of obtaining the optimal trade-off between the two. A Markov chain with a terminal state is identified from real world data to model the distribution of daily trip lengths. As a result, the need for the predetermined driving cycle is removed, as the probability distribution of previous driving cycles is applied instead. The input to the controller is the state of the HEV, and it is mapped to the engine torque input. They test the algorithm for two different degradation models, where the results contradict each other; one depletes the battery quickly, whereas the other is conservative in the rationing. The complexity of battery degradation makes it hard to create a simple enough model that is able to overcome the curse of dimensionality. The concluding remark, which is highly relevant for this paper, is that an accurate model of engine component degradation is crucial in order to achieve good performance of the trade-offs between fuel consumption and lifetime of components.

6.2 Learning based methods

Machine learning (ML) is a branch within artificial intelligence (AI) that has received a lot of attention the last decade. Simply put, ML aims to program a computer to learn from experience and be able to perform tasks humans would classify as intelligent, without explicitly being programmed. Machine learning is generally split into three main categories; supervised learning, unsupervised learning and reinforcement learning. Supervised learning relies on labeled data to train the algorithm. After training, the algorithm aims to predict correct values for new, unlabeled data. Unsupervised learning aims to find hidden structure or patterns in data that is unlabeled. This is typically applied on huge amounts of data as prepossessing, while it's not common as a direct control technique. Reinforcement learning does not require any data for training, where instead an agent interacts with an environment while continuously learning how to behave to maximize a reward signal in the long run. It is closely related to how we as humans learn, and many algorithms within the field are inspired by brain research on living creatures. Reinforcement learning is the branch of machine learning that is best suited for optimal control problems. It will be covered extensively in the following subsections, as the majority of our implemented algorithms falls within the category.

The three main branches of machine learning include a huge amount of literature. An overview of the three methods is visualized in Figure 6.2.

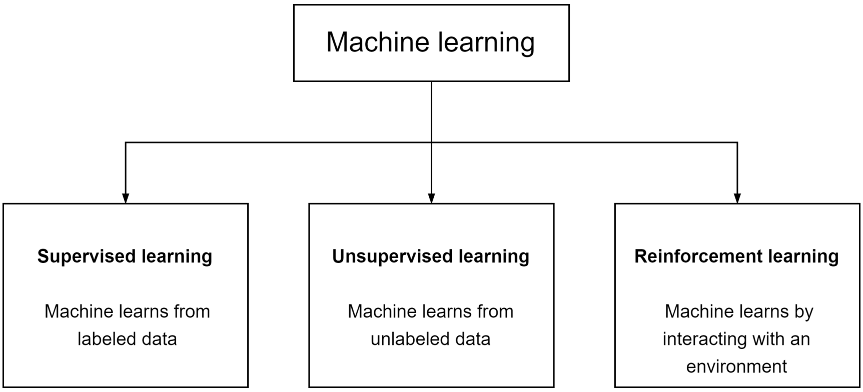

Machine learning
Supervised learning
Machine learns from labeled data
Unsupervised learning
Machine learns from unlabeled data
Reinforcement learning
Machine learns by interacting with an environment

Figure 6.2: Branches of machine learning

In this section, we will explain the theory behind several learning algorithms that we have implemented in Python. Central concepts that are both relevant for the algorithms and reinforcement learning in general will be presented along the way.

6.2.1 Tabular Q-learning

Tabular Q-learning is one of the first algorithms developed in the field of reinforcement learning. Despite being a learning algorithm, it is closely related to the dynamic programming algorithm explained in the previous section. The goal of the algorithm is to learn the Q-function of a policy, which gives the expected total discounted reward for being in state s and performing action a, and then following the policy until a terminal state. It can be formalized in the following way:

$$q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} \gamma^n R_{t+n+1} \mid S_t = s, A_t = a \right] \tag{6.2.1}$$

It is very similar to the value function described in the dynamic programming section, but there are some significant distinctions between the two. First off all, it uses the Q-function instead of the value function to predict the best action. This has the advantage that it doesn't need a model to estimate the value of each action. Given the value function, you have the optimal value of each state if you follow the optimal policy. However, if the transition function is unknown (which is one of the main reasons for using reinforcement learning on optimal control problems), the value function itself is useless for choosing an action as you can't know which action gives the best value in the next state. Therefore, the Q-function is usually applied to estimate the value of state action pairs in reinforcement learning.

Furthermore, while dynamic programming iteratively solves the entire state space in sweeps, Q-learning learns online through episodes from a starting point to a terminal state. This makes Q-learning able to learn online, and continuously improve its actions through experience from the environment, while dynamic programming has to be calculated in advance. However, while dynamic programming yields the optimal global solution given a correct model, there is no guarantee that Q-learning is able to find the best policy.

In it's basic form, Q-learning uses a table that links state action pairs to values. Therefore, like dynamic programming, it suffers from the curse of dimensionality when the state or action space grow large. During training, Q-learning uses the following update rule to learn:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \tag{6.2.2}$$

There are some interesting points to note here. First off all, temporal difference (TD), one of the most central concepts in reinforcement learning, is used. It is based on the idea that the optimal Q-value in a state should be equal to the reward from the optimal action, plus the Q-value of the resulting state. This is true when an optimal policy is found, but not necessarily before the Q-table has converged. Despite being related to the principles of DP, Q-learning is able to learn from it's own experience, without knowledge about the underlying dynamics or model. The learning rate, denoted α, describes how much emphasis new information gets when the model is learning. Typically, with a higher learning rate, you will learn faster in the initial stages of learning, but convergence will be slower or impossible. Therefore, you have to tune the learning rate in order to achieve both efficient training and convergence of the algorithm. A properly tuned learning rate is sufficiently low to ensure that the algorithm converges to reveal valuable information without overlooking important patterns. The full Q-learning algorithm is described in Algorithm 1.

Algorithm 1 The Q-learning algorithm [64]

```
Initialize $Q(s,a)$ arbitrarily
repeat
     (for each episode):
    Initialize $s$
    repeat (for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q$
        Take action $a$, observe $r, s'$
        Update
          $Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
        $s \leftarrow s'$;
    until $s$ is terminal
```

Explore-exploit dilemma

Another central theme in reinforcement learning that Q-learning highlights is the trade-off between exploration and exploitation. When the algorithm is exploiting, it performs the action that has the highest estimated return, and follows the policy that is best given the current information. However, there might be parts of the state space that is not yet visited, and thus might yield a better total reward. As a result, there is a conflict between exploring and exploiting, which has been the basis for massive research effort. For Q-learning, the most applied technique to address the issues is the epsilon-greedy function. A random action is selected with probability ϵ, or the highest value action is selected with probability $(1-\epsilon)$. By tuning epsilon properly, this can achieve satisfactory results for the Q-learning algorithm.

On-policy vs. off-policy

This leads us to one more very relevant feature of all reinforcement learning algorithms. They are described as either on-policy or off-policy. On-policy means that the algorithm uses the same policy it tries to learn while exploring. In contrast, off-policy algorithms uses two different policies: One to generate behavior and gather data with, and one policy that is continuously improved using the generated data. Q-learning is an off-policy algorithm, as it uses epsilon greedy strategy to generate data, while the policy that is learned is based on a purely greedy policy which value is based on the maximum Q-value of the next state. This is in contrast SARSA, which is a similar algorithm but learns the optimal policy given that epsilon-greedy is used both for generating behavior and policy learning. The differences in learned behavior by the two algorithms are summarized in Figure 6.3.

The agent start in state S and G is the only terminal state. For each step, the

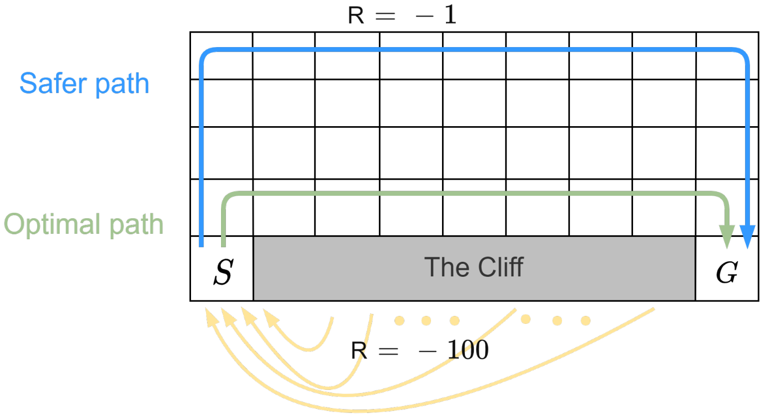


Figure 6.3: On-policy vs. off-policy [64]

agents receives a reward of -1, but if the agent walks off the cliff, the agent gets a reward of -100, and continues the episode in state S. As the figure shows, the optimal path is along the cliff, as the reward from that path is the highest. Q-learning is off-policy and tries to learn the greedy policy, it will choose the path along the cliff. SARSA, however, is learning the epsilon-greedy policy, and will thus risk taking a random move off the cliff in the next step if walking along the cliff. It will therefore prefer the safer path. One of the most significant implications of this is that off-policy algorithms can learn from previous data gathered, while on-policy algorithms can only learn from new experience.

Limitations

Despite its popularity, the Q-learning algorithm has several drawbacks. First of all, both Q-learning and dynamic programming suffers badly from the curse of dimensionality. As the computational burden grows exponentially as the number of states increases, it is rarely possible to apply the techniques to environments with more than 6–7 states. For real world applications with high dimensional action space this is a heavier burden for Q-learning, as each state has many corresponding actions, which leads to an exponential growth in the table size. This makes the agent unable to choose the optimal actions without immense amount of training, as huge parts of the state action space will not be visited. As an example, imagine a robot arm with 8 joints, and a course discretization of 3 possible actions per joint $(-1, 0, 1)$. That results in a huge state space of $3^8 = 6561$ actions, which in many cases is computationally infeasible.

Q-learning also has the issue that it requires both a discrete state and action space. This is problematic for real world applications because most robots has both continuous actions and states, not discrete. As an example, driving a car with 5 different speeds as possible actions is completely unimaginable. In addition, with discrete state and action space, there will be rounding errors as a result of the system being in a slightly different state than the algorithm thinks. This leads to inaccuracies that reduce performance.

6.2.2 Deep Q-learning

In order to overcome the issue of discrete state space in Q-learning, a function approximation is needed. Solving for each possible state and action for continuous problems will require an infinite amount of both time and storage. Instead of estimating the value function or the Q-function as a table, it can be estimated as a function with weight parameters $\mathbf{w}$. The function approximation aims to find a generalized pattern that maps state or state action pairs to value. Thus, the estimated Q-function for some given weights $\mathbf{w}$ can be written as follows:

$$\hat{q}(\mathbf{w}, s, a) \approx q_\pi(s, a) \tag{6.2.3}$$

There are several ways of designing the function approximation. One of the simpler ways is in the form of a linear function, where $\mathbf{w}$ is the weights of the features represented. However, in recent years, artificial neural networks (ANN) has become increasingly popular for approximating functions. And for reinforcement learning purposes, it has become the go-to method for solving the issue of discrete states and actions.

Artificial neural network

Artificial neural networks (ANNs) consist of several weights, symbolized by $\mathbf{w}$. The number of weights in the network will typically be way less than the actual number of possible states. This leads to the fact that changing one weight changes the value of many state action pairs, and thus the function approximation aims to find a general pattern that can map state action pairs to values. ANNs are in many ways designed the same way some parts of the brain works. It is an interconnected network of neurons, that usually requires a given signal strength in order to send signals to the neurons it is connected to. Artificial neural networks can be used in very sophisticated functions and has been central in recent breakthroughs in the image and speech recognition. Their success in these fields derive from their ability to finding patterns in huge sets of data. Furthermore, ANNs have been used for interesting reinforcement learning tasks such as learning to play the games of chess or Go.

A simple artificial neural network is visualized in Figure 6.4. It has four inputs, one hidden layer with four neurons and one output neuron that returns an output value. Therefore, this can be used as a value function, by mapping the states to a value calculated by the ANN. The number of hidden layers and neurons in each layer can vary significantly, as more complex data sets might need more neurons to model patterns accurately. The connections between the neurons has different weights $\mathbf{w}$ that amplifies or decreases the signals that is sent between them. During learning, algorithms using neural networks aim to tune **w** in order to give more desirable output.

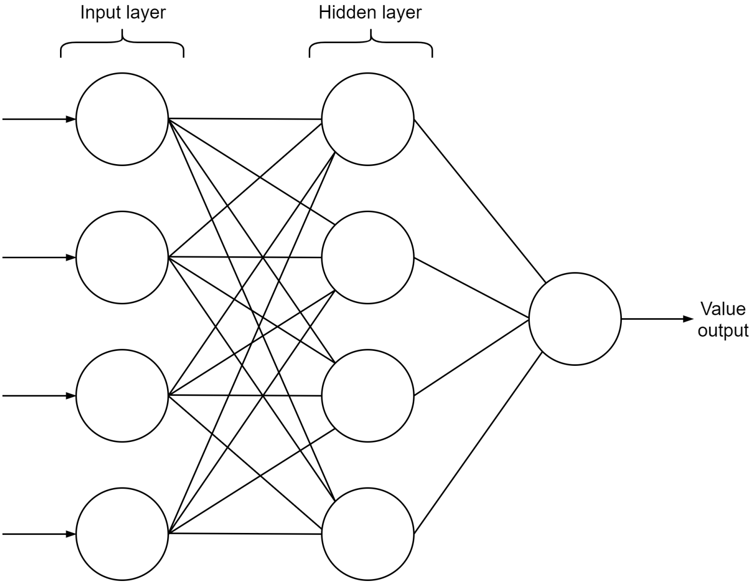


Figure 6.4: Artificial Neural Network

Usually, the neurons in the ANN are semi-linear units, meaning they compute a weighted sum of the input signals before they apply an activation function to produce the output. This is typically a nonlinear function, this enables the network to learn nonlinear patterns. Typically, the activation function is continuous and has a derivative, as this simplifies the learning process significantly. Commonly applied activation functions are the rectified linear unit (relu) function and the sigmoid function given in Equation (6.2.4), respectively.

$$\begin{aligned} f(x) &= \max(0, x) \\ f(x) &= \frac{1}{1+e^{-x}} \end{aligned} \tag{6.2.4}$$

To learn mapping input to desired outputs, the network uses a loss function, $L(x)$. It evaluates the quality of a predicted output versus the actual output. The aim is to minimize the loss for new, unseen examples, where the network is able to serve as a general mapping from state to correct output values. An example of loss functions typically used for linear output function is the mean squared error (MSE):

$$L(x) = \frac{1}{n}\sum_{i=1}^{n}(x_{actual} - x)^2 \tag{6.2.5}$$

To minimize the loss during training, a technique called backpropagation is applied to adjust the weights in the network. Briefly explained, backpropagation calculates the contribution to the loss from each weight in the network by derivation and using the chain rule, and then adjusts the weights slightly in a direction that decreases the loss most. The process of taking an incremental step in the direction that decreases the loss mostly for all weights is called *gradient descent*. This process is performed for every data point in the batch of training data.

Estimating the Q-value

The aim of Deep Q-learning (DQL) is to train a deep artificial neural network to map state action pair to Q-values. The neural network, called the deep Q-network (DQN), takes in the action and state as input, and gives the corresponding Q-value. Figure 6.5 illustrates a deep Q-network.

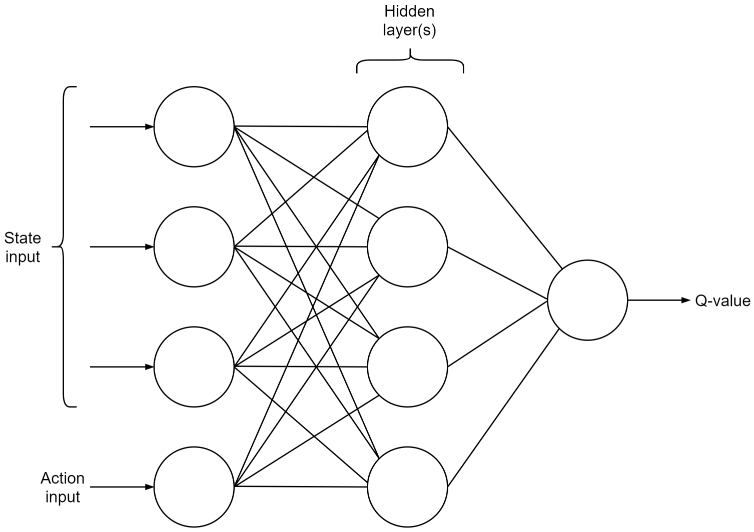


Figure 6.5: Deep Q-network

The only significant difference between tabular Q-learning and DQL is the function that approximates the Q-value. As the aim of the DQN is to predict the Q-value, the bellman temporal difference error should be minimized. The agent trains using (s, a, r, s') pairs generated from exploration in the environment. Then, it aims to minimize the TD-error using data from environment interaction.

$$TD_{error} = \left| r + \gamma \cdot \max_{a'} Q(\mathbf{w}, s', a') - Q(\mathbf{w}, s, a) \right| \tag{6.2.6}$$

The loss function that the network aims to minimize is the TD-error squared.

$$L_i(\mathbf{w}, s, a) = \Big(r + \gamma \cdot \max_{a'} Q(\mathbf{w}, s', a') - Q(\mathbf{w}, s, a)\Big)^2 \tag{6.2.7}$$

Minimizing this loss function is done by performing gradient descent on the DQN. The weights are changed incrementally, such that the loss on the training example is reduced. The gradients of the loss, with respect to the weight, can simply be computed by the chain rule and is denoted as $\Delta_{\mathbf{w}} Q(\mathbf{w}, s, a)$. The gradients symbolize how much a small change in each of the weights will affect the loss of the Q-network. Using this, the update rule of the network can be formulated.

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha\Big(r + \gamma \cdot \max_{a'} Q(\mathbf{w}, s', a') - Q(\mathbf{w}, s, a)\Big) \cdot \Delta_{\mathbf{w}} Q(\mathbf{w}, s, a) \tag{6.2.8}$$

α is the learning rate, which is an important parameter for ensuring convergence. Too small learning rate results in slow learning that might lead to local optimums, whereas a too big learning rate will cause the algorithm to never converge. The learning rate is typically the most influential hyperparameter, and tuning it is both time-consuming and important in order to achieve desired performance.

Issues and limitations

There are some issues with using a neural network for predicting the Q-value that has to be overcome. Firstly, the weight update on one training sample might affect the predicted Q-value on other training samples negatively. As a result, huge amount of data is needed to give precise predictions. Furthermore, this effect becomes distinct in reinforcement learning, as the state distribution of consecutive samples is correlated. This might lead to the Q-network training entirely on one specific part of the state space, and not being able to predict good Q-values for the rest of the state space. To address this, all transitions, (s, a, r, s'), during training are stored in an experience buffer. Then, when training the network, samples from this buffer are chosen randomly, in order to ensure a more uniform distribution of states. Another advantage is that it provides greater data efficiency, as data points during training can be used multiple times, which can speed up the training process. This process is called *experience replay* and is commonly applied in the field of reinforcement learning [71].

Another issue is related to the same network calculating both the target value and the predicted value. This causes both values to update simultaneously each time the network trains, and have proved to cause divergence and instability during training [72]. To address this, two networks is used, one for making the predictions of the Q-value, and one for producing the target. The prediction network is continuously trained, while the target network is only updated every n iteration, by copying the weights of the prediction network. This leads to a generally more stable training and increases the chance of converging. This process is visualized in Figure 6.6. $\mathbf{w}_i^-$ is the weights of the target network at time i.

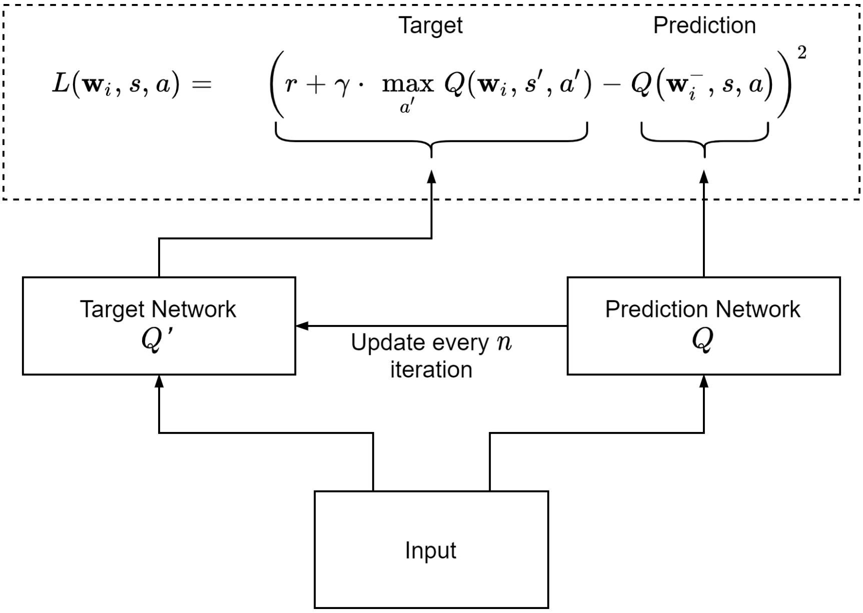

Target
Prediction
$L(\mathbf{w}_i, s, a) = \Big(\underbrace{r + \gamma \cdot \max_{a'} Q(\mathbf{w}_i, s', a')} - \underbrace{Q(\mathbf{w}_i^-, s, a)}\Big)^2$
Target Network
Q'
Update every n
iteration
Prediction Network
Q
Input

Figure 6.6: Prediction and target network in DQL

$$L(\mathbf{w}_i, s, a) = \Big(r + \gamma \cdot \max_{a'} Q(\mathbf{w}_i^-, s', a') - Q(\mathbf{w}_i, s, a)\Big)^2 \tag{6.2.9}$$

Furthermore, in order to evaluate the best possible action to make, the network has to perform a prediction of a state and action value(s). As a result, it is only possible to evaluate a limited amount of actions, which means that the action space has to be discrete. The disadvantages that comes with discrete actions are highlighted in Section 6.2.1.

6.2.3 Soft actor-critic

Soft actor-critic is one of the later breakthroughs within the field of reinforcement learning. After it was published in 2018, it has gained reputation as one of the most stable off-policy methods for continuous control problems [73]. For our application, this is crucial, as using continuous action and state representation are by far favorable when controlling a real ship.

There are several interesting aspects of this algorithm. First of all, it uses the actor-critic framework, which is shared among many RL algorithms. It is based on the general policy iteration algorithm, which incrementally improves the value function and policy in a known environment, by alternating between policy evaluation and policy improvement. This algorithm is not RL as it requires a known environment. However, several actor-critic algorithms are based on the same idea. Generally, this implies that it aims to learn at least two functions. One is a value function for estimating the value of a state or a state action pair. This is called the *critic*. The *actor* however, is a policy function, that aims to learn the optimal policy. The former is used to update the latter, and both functions improve when gaining new experience. The learning process is visualized in Figure 6.7.

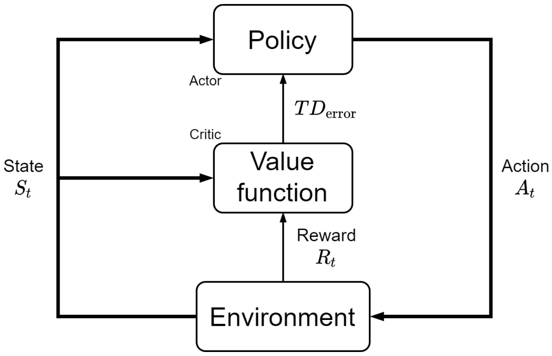


Figure 6.7: Actor-critic architecture

In the soft actor-critic algorithm, an artificial neural net is used both for the actor and the critics. The actor typically gives the probability of taking the different possible actions.

$$\pi(a \mid s, \mathbf{w}) = Pr\left[a_t = a | s_t = s, \mathbf{w}_t = \mathbf{w}\right] \tag{6.2.10}$$

The equation above states that the policy gives the probability of taking action a_t, given that you are in state s_t with network weights $\mathbf{w}_t$. There is one significant difference that separates soft actor-critic from standard RL methods. Commonly, RL algorithms aim find the policy that maximizes the reward accumulated.

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \sum_{s',r} (s', r \mid s, a) \left[r + \gamma v^*(s')\right] \tag{6.2.11}$$

This however, is convenient for algorithms that learns a deterministic policy. This means that a given state will always result in the same action until the policy is updated. This enforces the algorithm to have some kind of exploration built in order to gather more diversified data points. Typically, it is done through some noise added to the policy, with strategies such as epsilon-greedy. However, soft actor-critic learns a stochastic policy. For stochastic policies, the aim is to maximize the objective function $J(\pi)$, which for a given policy π is the sum of the probability of visiting all states times the reward it yields.

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s,a)\sim\rho_\pi} \left[\gamma^t \cdot r(s, a)\right] \tag{6.2.12}$$

In the objective function in Equation (6.2.12), the of expected discounted rewards given a probability distribution for visiting each state, ρ_π, should be maximized in the long term. However, this has the potential drawback that it does not reward exploration, and therefore it might end up with suboptimal policies as a result of biasing towards exploitation. In order to deal with this, soft actor-critic maximizes both the reward and the entropy of the policy. The entropy indicates how random the policy is, and therefore, more deterministic policy will yield lower entropy. This ensures that the policy trained has sufficient exploration in the stochastic policy it learns. The objective function to be maximized is as follows:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s,a)\sim\rho_\pi} \left[r(s,a) + \alpha \mathbb{H}(\pi(\cdot \mid s)) \right] \tag{6.2.13}$$

where $\mathbb{H}(\pi(\cdot \mid s))$ denotes the entropy of the policy π in state s is [74].

Characteristics

Maximizing the objective function using the actor-critic framework ensures a stable algorithm that is able to handle both continuous states and actions. It gives better convergence properties than similar popular algorithms such as deep deterministic policy gradient (DDPG), which is typically brittle and often requires extensive work of tuning parameters. Furthermore, as an off-policy algorithm, it has the ability to learn from previous gathered data, which makes it more data-efficient than popular continuous space on-policy algorithms such as proximal policy optimization (PPO) and SQL. In addition, it is effective in high dimensional spaces, which makes it able to learn good policies in complex environments. Despite this, it has the possible disadvantage that there is no guarantee on converging to the optimal policy, as it might be stuck in an local optimum. This is also the case for most RL algorithms that perform well in continuous environments.

7 Simulation and Discussion

In this chapter, the simulation setup is described and discussed. The load profiles applied in the simulations will be presented and the performances of several algorithms are given. The simulation results will be covered in detail, before the results are discussed. The chapter is rounded off by a quantitative and qualitative discussion, comparing the fuel consumption, distribution of costs and aging effects of the different control strategies.

7.1 Load profile

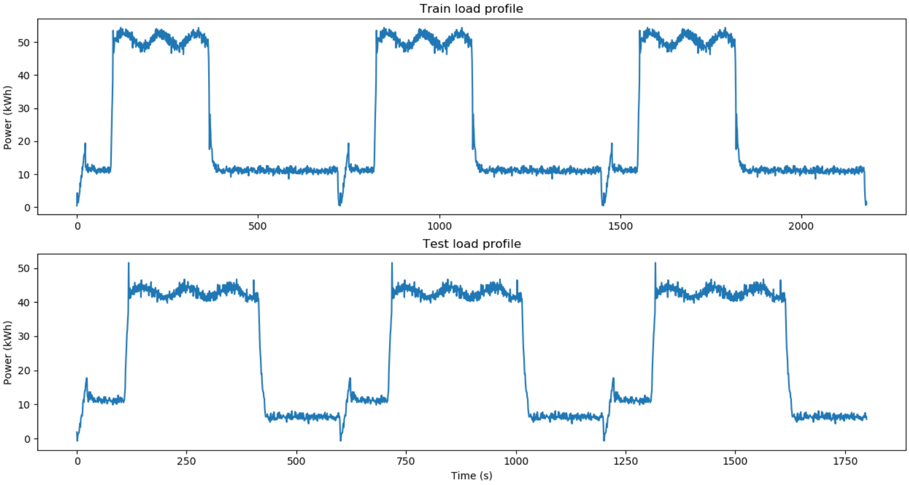
For the machine learning algorithms to learn, huge amounts of data is required. To learn how to operate and split the power between battery and fuel cell properly, training on this data is required for satisfying results. The training data used in the simulations consist of real ship data from a harbor tugboat operating in Singapore. The vessel is equipped with two gensets and batteries. Note that despite the fact that the ship does not have the same power system as the one considered in this paper. This does not matter as only a real load profile is needed for our simulations. It should be mentioned that the ship's modes of operation are *transit*, *idle* and *ship assist* [75]. Table 7.1 shows the tugboat's system parameters [76].

The data represents the load demanded by the ship operator, in transit between two locations. The PEMS has the task of split the power between the fuel cell and the battery in the most cost efficient manner. One thing to note is that the optimal PEMS policy might be significantly different in separate operation modes. For example, it is far easier for the PEMS to predict the future load during deep sea shipping, as compared to loads during maintenance work for offshore wind farms. As a result, the controller can in one case keep the FC at a steady level, reducing wear and tear related to transient loading in one case, while it may be challenging to maintain a steady load in the other situation. It should, however,

Table 7.1: Harbor tugboat parameters [76]

Symbol	Value	Description
$P_{load,max}$	3800 kW	Maximum propulsion power
$P_{gen,1max}, P_{gen,2max}$	1200 kW	Generator rated power
V_{OC}	1000 V	Open circuit voltage
Q	520 Ah	Battery capacity
$P_{bat,min}$	2 C	Maximum charging
$P_{bat,max}$	3 C	Maximum discharging
$n_{gen,1}, n_{gen,2}$	96.5 %	Generator efficiency
n_{DC}	94.5 %	Drivetrain efficiency

be possible to train the PEMS to perform well for several different operations if sufficient amounts of data becomes available. This means that the results for the different PEMS strategies presented in the following section are trained for this particular load profile. Figure 7.1 displays the two load profiles used in the simulations. It consist of two slightly different plots; a training load profile and a test load profile.


Train load profile

Test load profile

Figure 7.1: Load profiles for training and testing

The training load profile is used to train the algorithms. It consists of one load cycle of approximately 700 seconds, which is repeated three times. This is to allow more variations in the battery SOC, as it takes several hundred seconds for it to change significantly during heavy battery charging/discharging.

The testing load profile is used for the soft actor-critic algorithm, in order to test it's performance on another load profile. In practice, it is important that the PEMS doesn't overfit, meaning that it is able to perform well on new data, not just the training data. This is especially true for marine applications, where the stochastic nature of the ocean causes randomness in the load demand. The testing load profile is not, however, used for the tabular methods. Since the tabular methods are not able to generalize, they can only indicate what to do in exact states that it has seen before. In other words, they are not able to predict actions that are optimal unless the state has previously been explored. To combat this, huge amount of real ship data is required to ensure that all or most states are visited by the tabular algorithm. This a feature that makes tabular algorithms a weak choice for both marine and general purpose energy management systems.

7.2 Results

Simulations for all the algorithms introduced in Chapter 6 were performed. In the following subsections, a basic description of the implementation of the algorithms will be presented. This includes discussion of the algorithm design choices, the pros and cons of the algorithm as well as the relevant parameters used. Then, the results from each simulation are described and presented. Simulations were performed on both the load profiles described in the previous section. The power delivered from the battery and FC will be presented for all algorithms, as well as the SOC and all the costs. The sections also includes a discussion how the results reflects what was expected, and how to improve the results. All the algorithms and models are developed by the authors, and the relevant code is attached.

7.2.1 Rule-based

Implementation

The rule-based control algorithm is implemented as a benchmark for evaluating the performance of the learning based algorithms. The set points are based on the algorithm proposed in Han et al. [11]. It is simplistic, and considers only two variables: The battery's state of charge and the load required. All operation states for the rule-based algorithm is given in Table 7.2 while Table 7.3 explains the used terms.

$P_{FC,min}$ is set to 10 % of the maximum FC power. Reducing the FC power further will result in terrible efficiency on real fuel cells. $P_{FC,max}$ is set to 90 % of the maximum FC power, as exceeding this will result in bad efficiency as well as significant FC degradation. $P_{FC,opt}$ is set to 50 % of the maximum fuel cell power, as the FC efficiency is good and degradation is generally low at this operating point. $P_{bat,opt}$, P_{optdis} and $P_{optchar}$ is set to the charge/discharge rate of 1C. This is considered sensible, as the battery we use in the simulations has relatively small capacity compared with the operating voltage. Note that these values are by no means the best possible (which obviously will be different from different battery/fuel cells), but they are in line with the general principles in operating FC and battery, and are therefore considered sufficient as a benchmark.

Table 7.2: Rule-based control algorithm [11]

Rule-based operating states

SOC	State	Load power	Reference power of FC
SOC > 70 %	1	$P_{load} \leq P_{FC,min}$	$P_{FC,min}$
	2	$P_{load} \leq P_{FC,min} + P_{optdis}$	$P_{FC,min}$
	3	$P_{load} \leq P_{FC,max} + P_{optdis}$	$P_{FC} = P_{load} - P_{optdis}$
	4	$P_{FC,max} + P_{optdis} < P_{load}$	$P_{FC,max}$
30 % ≤ SOC ≤ 70 %	5	$P_{load} \leq P_{FC,min}$	$P_{FC,min}$
	6	$P_{load} \leq P_{FC,opt} - P_{bat,opt}$	P_{load}
	7	$P_{load} \leq P_{FC,opt} + P_{bat,opt}$	$P_{FC,opt}$
	8	$P_{load} \leq P_{FC,max}$	P_{load}
SOC < 30 %	9	$P_{load} > P_{FC,max}$	$P_{FC,max}$
	10	$P_{load} \leq P_{FC,max} - P_{optchar}$	$P_{load} + P_{optchar}$
	11	$P_{load} > P_{FC,max} - P_{optchar}$	$P_{FC,max}$

Table 7.3: Terms and values from the rule-based algorithm

Name	Value	Description
P_{load}	Varying	Load required
P_{FC}	Varying	FC power
$P_{FC,min}$	12 kW	Minimum operating FC power
$P_{FC,max}$	108 kW	Maximum operating FC power
$P_{FC,opt}$	60 kW	Optimal Fuel Cell power
$P_{bat,opt}$	19.62 kW	Optimal battery power
P_{optdis}	19.62 kW	Optimal battery charging power
$P_{optchar}$	19.62 kW	Optimal battery discharge power

The algorithm mainly have three purposes. The first is keeping the state of charge of the battery in a desired region, between 30 % and 70 % of the maximum SOC. This is considered the operating region that is best for the battery SOH. Second, it aims to keep the battery as close to the optimal battery charge and discharge rates. Third, it aims to avoid the low power region of the fuel cell, where it typically has very low efficiency. The result is a robust controller, that is designed both for reducing emission and fuel costs by avoiding inefficient operating powers for the battery and FC. However, it is by no means an optimal controller, and the degradation effects it aims to limit is just limiting the battery SOC and the power of FC and battery. Thus, the performance of the controller can be expected to be stable and reliable, but far from optimal.

Simulation Results

The algorithm was simulated on the train load profile. There is no reason for using two load profiles, as the rule-based algorithm does not learn from experience, and will not cause it to overfit. This is obviously unless the parameters are tuned to give exceptional results on one specific load profile, which is not the case in this study. The resulting power split between the fuel cell and battery is given in the Figure 7.2.

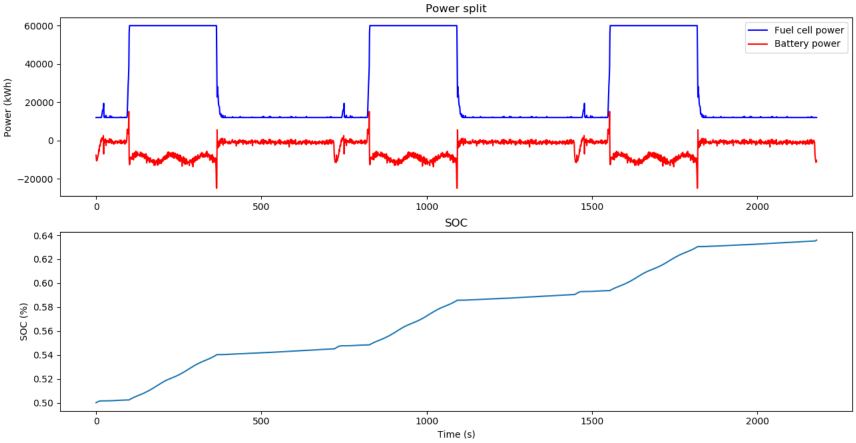


Figure 7.2: Power split and SOC for RB control

In the simulation, the FC primarily operates in two power regions, at 12 kW, 10 % of max FC power, and at 60 kW, which is the optimal power set point for the FC. The FC provides a little more power than what is demanded, which causes the battery to charge slowly for the entire simulation, despite the change in load required.

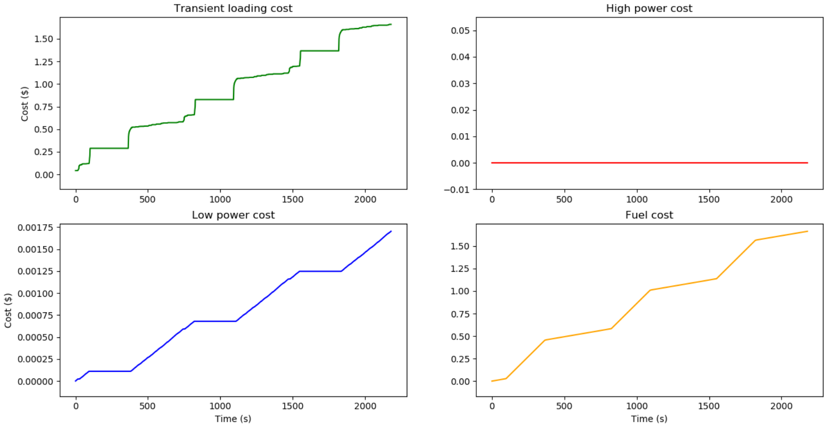


Figure 7.3: FC cost for RB control

The FC costs from the simulation is given in Figure 7.3. The FC costs are primarily related to transients in the load, and fuel costs. The power split shows that the FC power is generally stable, except for when the algorithm switches between $P_{FC,min}$ and $P_{FC,opt}$. At these points, the fuel cell increases at maximum rate. This causes significant degradation due to transient loading at these instances. The fuel cost is increasing at a significantly lower rate where the fuel cell is providing low power, which is expected. The costs due to low and high power is negligible and zero respectively.

The battery related costs are displayed in Figure 7.4. The SOC is always in the non-penalty zone, and the penalty is thus zero. DOD costs are increasing steadily, which means that between the battery charging, there are minor intervals of discharging. This is good for the battery health, as from experience, the DOD related cost tend to increase more severely when the cycles are deeper. The cost of power loss, which is proportional to i_{batt}^2, is low for the entire cycle. This is expected, as the charge and discharge rates are small for the entire simulation.

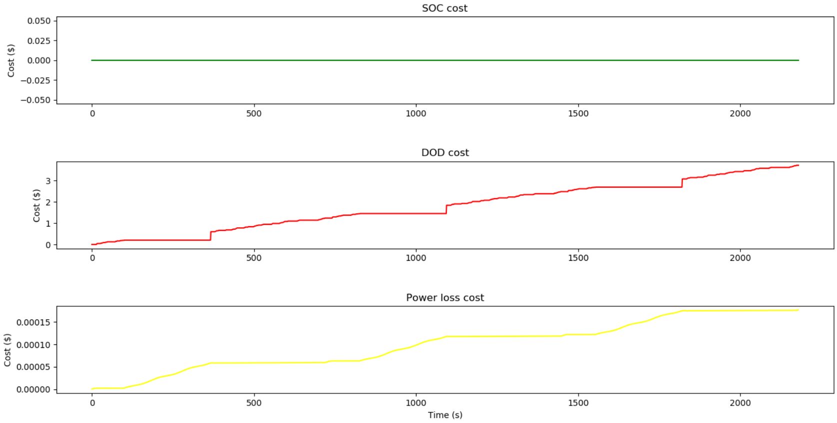


Figure 7.4: Battery cost for RB control

The total costs of the rule-based control strategy are given in the Figure 7.5.

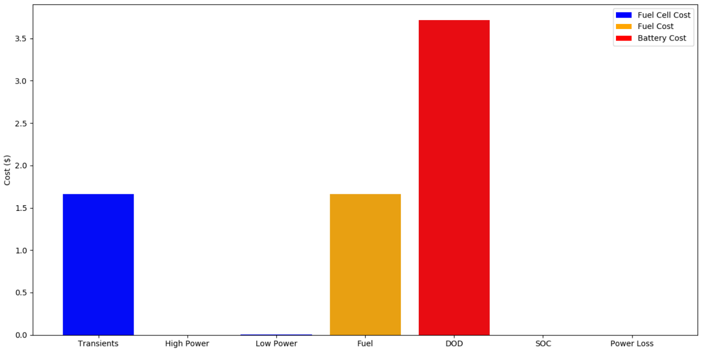


Figure 7.5: Operating costs for RB control

For the total time of 2181 s of simulations, the total cost of the vessel is estimated to be \$7.04. This is considered to be a fairly good result, which will be clear when comparing to the other algorithms. Of the degradation costs, the two dominating factors is battery DOD and fuel cell transients. The other factors are considered negligible. The fuel cost accounts for approximately a quarter of the total costs. Despite the inaccuracies that certainly exist in the models and the cost functions, this proves the point that considering degradation is essential in order to design a PEMS that is cost efficient.

In the first simulation for RB control, the performance seems good. The controller only operates in the region where 30 % $\leq$ SOC $\leq$ 70 %. However, in order to know how the controller performs in all settings, it has to be tested. Therefore, a simulation with an initial SOC of 65 % was conducted. The results for the power sharing, battery SOC and costs are given in the figures below.

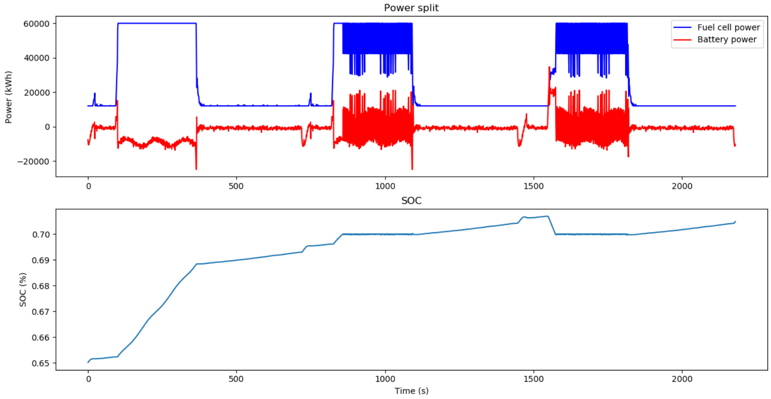


Figure 7.6: Power split and SOC for RB control. High initial SOC

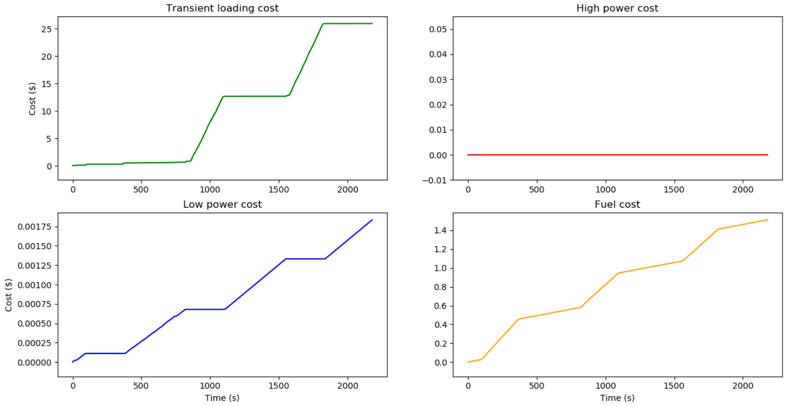


Figure 7.7: Fuel Cell costs for RB control. High initial SOC

We can see that it occur significant power oscillations when the battery reaches 70 % of maximum SOC. This causes the algorithm to fluctuate between two operating modes, causing massive load fluctuations in both the battery and the FC.

The massive power fluctuations leads to significant cost due to transient loading. Other costs for fuel cell remain low.

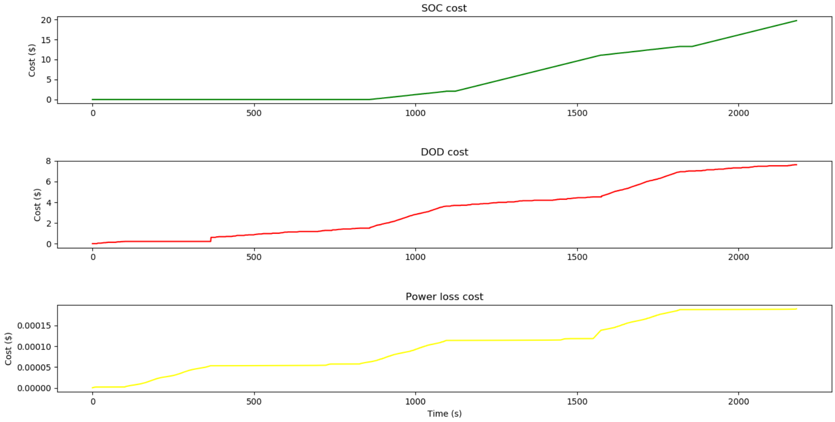


Figure 7.8: Battery costs for RB control. High initial SOC

When the SOC increases past 70 %, the penalty for SOC starts ticking, and the SOC cost is becomes a significant cost factor. Also, due to the heavy load fluctuations, the DOD cost increases in magnitude.

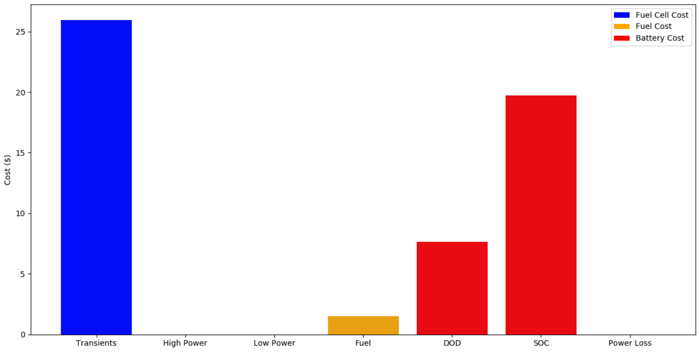


Figure 7.9: Total costs for RB control. High initial SOC

The result is a total cost of $54.8. This is 778 % higher than the case where initial SOC was equal to 0.5, meaning that the controller performs immensely worse when operating out of the defined desirable SOC range. This is the controller implemented and discussed in a well cited paper, though there might be some differences in the parameters used. Rule-based control strategies might work well in some cases, but this proves a significant weakness that they generally possess. They are not adaptive and their performance vary to a great extent in some cases compared to other.

7.2.2 Dynamic Programming

The dynamic programming algorithm was implemented, but not used in simulations. As the algorithm is very time intensive, the use of DP was dismissed based on the time scope of this thesis. When running dynamic programming, every possible state is explored, making it vulnerable to the curse of dimensionality.

To achieve satisfactory results with dynamic programming, the state space needs a sufficiently fine grid distribution. As an example, the battery's SOC should at least be divided into 100 different states. Ideally, this number should be an order of magnitude greater to represent the real change in SOC. Nevertheless, this is not feasible as the training time would be far too long. Curbed by the timeline of this master thesis, only a lower state space could be considered.

A low state space does, however, contribute to a major drawback of dynamic programming. Several actions will get mapped to the same state, greatly reducing the usefulness. Let's say we have a course SOC grid size of 11, meaning that the SOC can have the values from 0 to 10. Actions, in the form of a change in fuel cell power, gives a corresponding battery power and a resulting SOC. When the grid is small, several action map to the same SOC state. The algorithm can not distinguish between those actions as their impact on the system is ambiguous.

The reason the control problem needs a fine grid is to properly differentiate between actions that are beneficial to the cost optimization. As we have seen, this requires such a high state space which renders DP too time-consuming. Thus, it is concluded that this strategy is out of scope.

7.2.3 Tabular Q-Learning

Implementation

The tabular Q-learning algorithm uses three discrete states, the battery SOC, the fuel cell current and the load demanded. During training simulations, the values are rounded to a grid with a finite set of states. For our simulations, we used a grid discretization of [101, 41, 51] states for [SOC, I_{FC}, P_{load}] respectively, resulting in a total of 211191 states. This is a huge amount of states that has to be visited before the algorithm is able to yield a good result. Generally, a finer grid leads to a more exact result, but it increases the training time. Therefore, this is a trade-off that has to be considered carefully. We tested with different grid sizes, and landed on a good compromise between speed and performance.

One interesting thing to note is that the SOC grid needs to be quite fine. The reason for this is that the SOC changes more slowly than the other states. If the grid is coarse, cases where regardless of what action you perform, the SOC doesn't change might occur. This leads to even more training time needed before the algorithm converges, and some times it is even not possible. As an example, if the SOC grid only consists of 21 points, with the continuous state being rounded to the nearest 5 %. Here, depending on the battery capacity,it might take several hundred time steps of aggressive actions before the the battery SOC changes. For the algorithm, it will then be hard to know exactly what actions contributed to the change in SOC, and what actions contributed in the other direction.

The action choices are 9 different discrete values, which represents the change in the value of I_{FC}. The rate of change of the FC is limited to 28 A/s, which is 10 % of the maximum FC current. This rate constraint goes for both increasing and decreasing the current. With a time step of 1 second, the resulting action possibilities **a** are as follows:

$$\mathbf{a} = [-28,\ -21,\ -14,\ -7,\ 0,\ 7,\ 14,\ 21,\ 28] \tag{7.2.1}$$

The size of the resulting Q-table is the number of actions times the number of states. The relevant size parameters are given in Table 7.4.

Table 7.4: Q-table parameters

Parameter	**Value**
State	211191
Action	9
Q-table size	1900719

In order to balance exploration and exploitation, a decaying epsilon greedy strategy have been applied during training. This has the effect that during early stages of training, when the agent has little knowledge of what actions to prefer, it explores. Then, during later stages of training, it will tend to explore the promising strategies instead as the probability of taking the greedy action increases. For the simulations epsilon started at 0.5, and decayed exponentially by a factor of -0.00002 for each episode. The lower limit of epsilon during training was set to be 0.05, in order to ensure some exploration. If there is zero exploration, the agent will experience minimal learning. The development of epsilon during the episodes simulated is visualized in Figure 7.10.

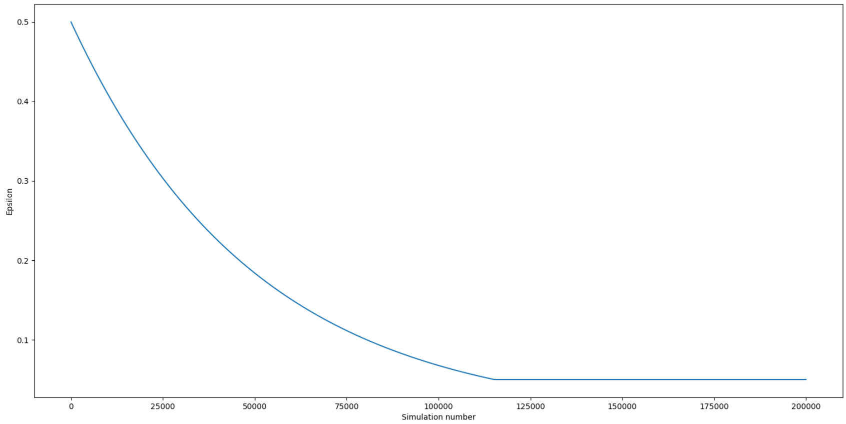


Figure 7.10: Evolution of epsilon during training of Q-learning

The reward function for the tabular Q-learning algorithm during training was defined as follows:

$$R_{QL} = -(C_{fuel} + C_{FC,deg} + C_{bat} \cdot D_{SOC}) \tag{7.2.2}$$

Where equations and values for the coefficients are given in Table 4.4. The algorithm aims to minimize the cost of operation. The degradation related to DOD was not included in the reward function, as the DOD was not included in the state space. The reason this cost is excluded is that it requires at least one more state to represent the depth of discharge. This will cause the simulation time to increase by a factor of number of DOD possible DOD states, which was considered too heavy computationally for our purposes. However, the authors strongly believe that the algorithm also could give good results with DOD included in the state space, given sufficient time.

The learning rate, describing how much you emphasize new observations vs old observations was set to 0.5 during the simulations. This is quite high compared to what is usual, but considering that the only random factor in the underlying Markov decision process is the load of the next state, it should be a feasible learning rate.

The discount factor, which essentially determines how much you weight future rewards, is an interesting topic of discussion. In the reward function, some of the rewards are instantaneous, whereas some are delayed. As an example, the penalty for high transient loading and fuel usage is instant, whereas the penalty for SOC is delayed. The SOC might be close to the area where it enters the penalty zone, defined as above 70 % or below 30 % of maximum SOC. But it doesn't get a negative reward until you actually enter the penalty zone. However, the actions that took the battery very close to the penalty zone should also be given some responsibility for the penalties that occur once it actually enter the penalty zone for SOC. As a result, high discount factor is needed to highly emphasize the future rewards that you can expect to get from taking an action in a given state. The discount factor for our simulations is set to 0.999.

Simulation Results

The Q-learning algorithm was both trained and tested on the training load profile. This had to be done, as many of the required loads in the testing load profile was not present in the training load profile. This highlights one of the biggest weaknesses that comes with tabular methods. They do not generalize, and can only give a hint on what action to take in the states that are visited before. In order to obtain a tabular Q-learning algorithm that is able to perform on any load profile, vast amount of data is needed, and thus even more training. However, testing the performance on the training load profile is considered sufficient in order to get an overall picture of how the tabular Q-learning algorithm learns, and in addition to it's strengths and weaknesses.

The algorithm simulated for 200 000 episodes, where each episode is a complete simulation through the entire load profile. In Figure 7.11, the evolution of the rewards achieved is given. It is represented as a moving average of 100 rewards in order to avoid a noise in the display. It can be seen that the algorithm learns steadily, decreasing the cost over the simulations, until convergence at approximately 150 000 simulations. It should be noted that there is a close relation between this graph and the epsilon decay in Figure 7.10. As epsilon decays, the performance will improve as a result of the agent choosing the action representing the highest Q-value with a higher probability. However, it can be seen that the rewards improves at a steeper rate prior to 100 000 simulations, which is a clear indication that the agent learns during training.

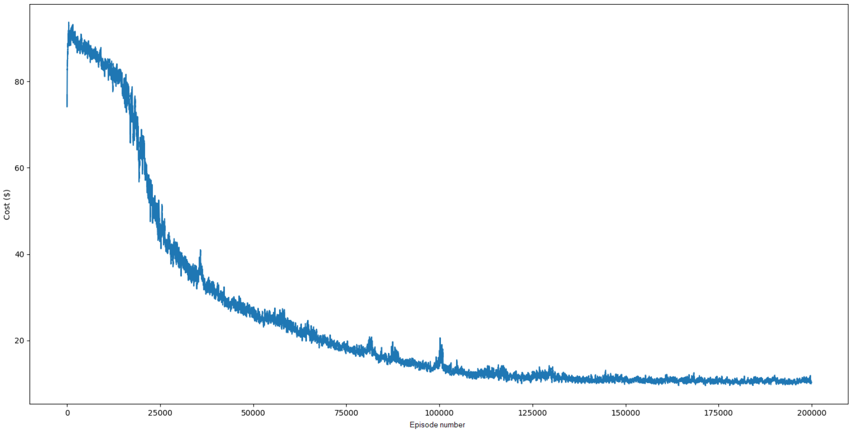


Figure 7.11: Rewards during training of Q-learning

The power split between battery and fuel cell, as well as the evolution of the battery SOC is given in Figure 7.12.

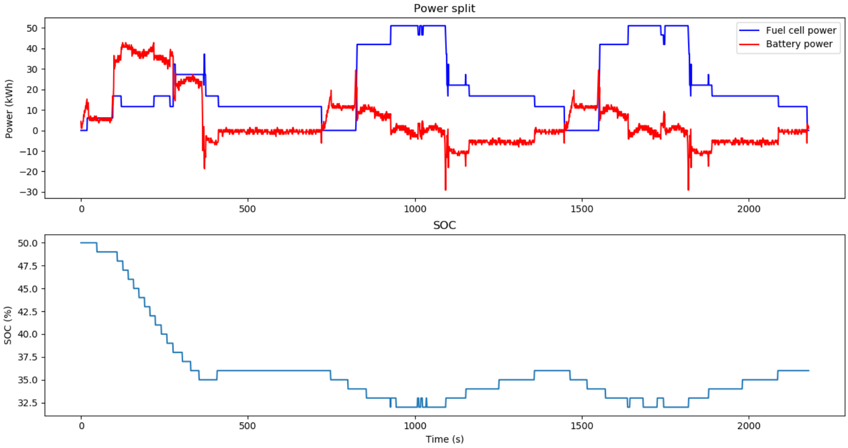


Figure 7.12: Power split and SOC for tabular Q-learning control

It can be seen that the FC usually keeps a steady profile, which avoids extensive transient loading. For most of the simulation time, it also avoids both high and low power, which also is good for reducing FC cost. The algorithm is able to contain the battery SOC in the desired region, between 30 % and 70 % for the entire simulation, which avoids cost due to low SOC.

The costs due to operating the FC is given in Figure 7.13. It can be seen that costs due to transient loading and fuel dominate, whereas the low power cost is negligible and the high power cost is equal to zero. This is as expected, considering the FC power profile, and reflects that the algorithm is able to operate the fuel cell in an economic sensible way.

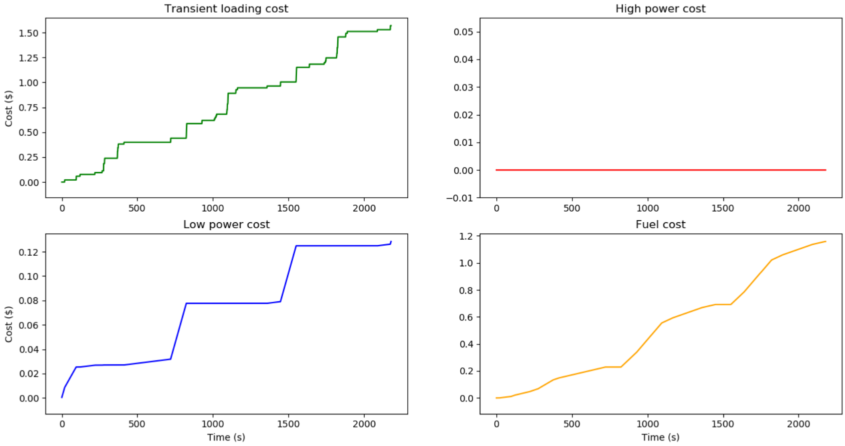


Figure 7.13: Fuel cell costs for tabular Q-learning control

The battery costs related to SOC and power loss are zero and negligible, respectively. The cost due to depth of discharge, however, is significant. This is because it is not included in the reward function for tabular Q-learning. Therefore, the performance of the algorithm can not be evaluated based on the DOD. It is included for the purposes of providing a complete picture of all the costs.

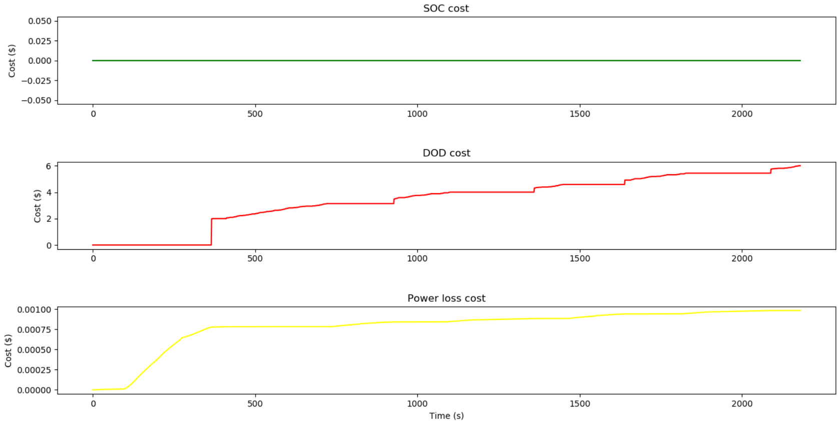


Figure 7.14: Battery costs for tabular Q-learning control

The total costs, DOD excluded are given in Figure 7.15. The Q-learning algorithm successfully achieves low cost on the metrics it is evaluated on, and yields a good overall performance. However, the lack of continuous states and actions makes it very hard to generalize the algorithm for online applications without vast amount of data.

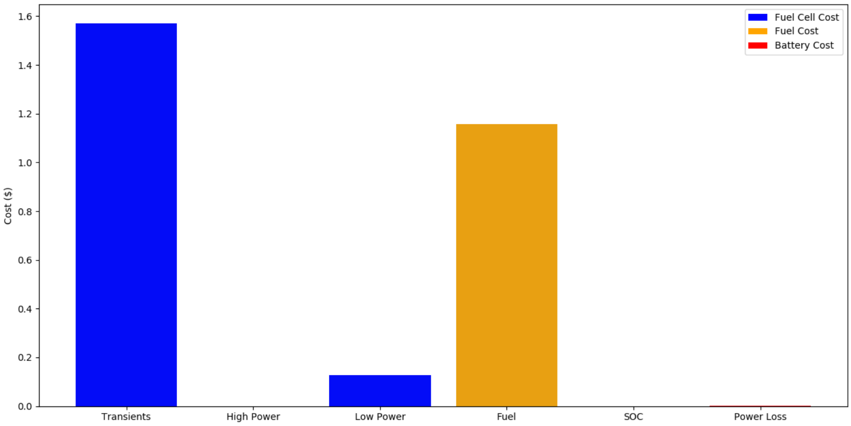


Figure 7.15: Total costs of tabular Q-learning control

7.2.4 Deep Q-learning

Implementation

The deep Q-learning algorithm aims to predict the Q-value of a state, and uses this to choose the best possible action among a discrete set of actions. The state of the system consist of 3 parameters: The battery SOC, the fuel cell current and the load demanded. These are, in contrast to tabular Q-learning, continuous values, which helps DQL overcome the curse of dimensionality. In order to stabilize and make the training process more efficient, the state **s** is normalized, such that it always is in the interval $[0, 1]$. The input state can thus be written in the following way:

$$\mathbf{s} = \left[\frac{SOC}{SOC_{max}}, \frac{I_{FC}}{I_{FC,max}}, \frac{P_{load}}{P_{load,max}}\right] \tag{7.2.3}$$

where $SOC_{max} = 100$, $I_{FC,max} = 280$ A and $P_{load,max}$ is the maximum load of the training load profile. It should be commented that for testing load profile, the maximum load might exceed the maximum load during testing, and thus the value of $\frac{P_{load}}{P_{load,max}}$ might exceed 1. However, if the algorithm generalizes well, it should not have a huge effect on the performance. This is obviously unless P_{load} in the testing data is significantly higher than $P_{load,max}$ from the testing data. However, in this case, the issue is insufficient training data, which would be problematic if the values were not normalized as well. DOD was not considered, as the algorithms performance when considering only three states was unsatisfying. Therefore, adding more complexity to the state space and reward function was determined to be waste of time.

The action discretization are the same as for Q-table, given in Equation (7.2.1). The actions are represented as input to the neural network as a one-hot encoded vector, which is a common way of representing categorical features in neural networks. The first index of the one-hot encoded actions represents a decrease in FC current by 28, whereas the last index represents an increase in FC current by 28. Representing the action as the correct numeric value is also a possibility, but the result should in theory be the same. The one-hot representation of taking action 0 is given in Equation (7.2.4).

$$\mathbf{a}_{0,one-hot} = [0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0] \tag{7.2.4}$$

The normalized state vector and the action vector is concatenated and fed into the neural network as a single vector of size 12. The ANN architecture for the DQL implemented is visualized in Figure 7.16.

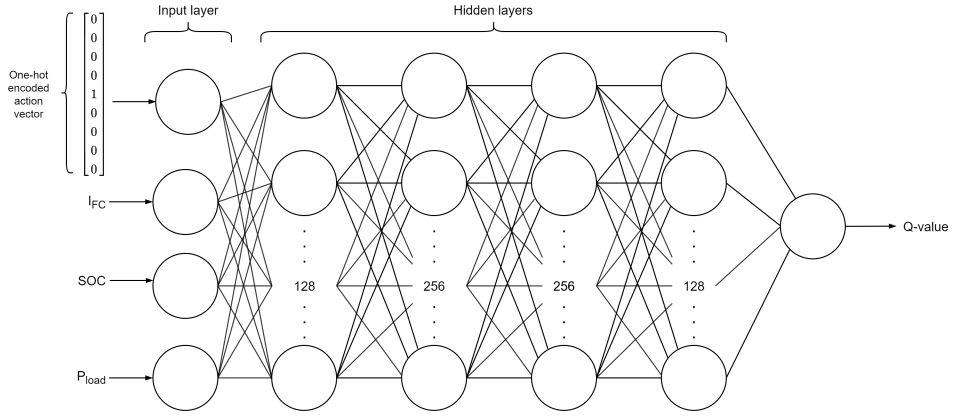


Figure 7.16: The ANN architecture of the DQL algorithm

The reward function for the DQL training was given as follows:

$$R_{DQL} = -(C_{fuel} + C_{FC,deg} + C_{bat} \cdot D_{SOC}) \tag{7.2.5}$$

where equations and values for the coefficients are given in Table 4.4. The algorithm aims to minimize the cost of operation. The degradation related to DOD was not included in the reward function as the DOD was not included in the state space.

Like the tabular Q-learning algorithm, DQL uses a decaying epsilon greedy strategy to combine the concepts of exploration and exploitation. Epsilon starts at 0.5, and decays to 0.1 over the span of the simulations, with a more aggressive decay than the one proposed in tabular Q-learning. This is justified by the long simulation times for each period due to calculations using the neural network instead of a lookup table. The learning rate of the neural network was tested with different values, and ended up as $5 \cdot 10^{-4}$. The network consists of 4 hidden layers, where the number of neurons are 128, 256, 256 and 128 in the respective layer. There is no exact science in setting the hyperparameters of a neural network. There are some guidelines, such that the learning rate commonly is between 0 and 1, and is typically way closer to 0 for functions that are more complex in nature. Furthermore, the amount of hidden layers, and the number of neurons in each layer is also a matter of trial and error. A neural network with a single hidden layer could potentially represent any function. However, selecting a feasible network dimension might significantly increase the probability of successfully training the network to achieve the desired performance.

The discount factor is set to be 0.999, which is considered relatively high. This is essential for good performance of the algorithms, as many of the rewards are significantly delayed, while some are instant.

Simulation results

The algorithm was trained on the training load profile, and tested on the testing load profile. This is already an improvement compared to the tabular Q-learning. Contrary to the tabular algorithm, it is able to generalize, and potentially give good predictions on states that have not been explored.

The algorithm takes a significantly longer time to simulate than the tabular Q-learning version, as a result of the vast amount of computation needed to calculate the predictions from the neural network. Also, training the network is very time-consuming, as it was trained for 5 epochs on a mini-batch of 1000 (s, a, r, s') samples after each episode. There were thus performed 1000 episodes of simulation on the training load profile before testing the algorithm. However, the observed change in behavior after about 100 episodes is very limited.

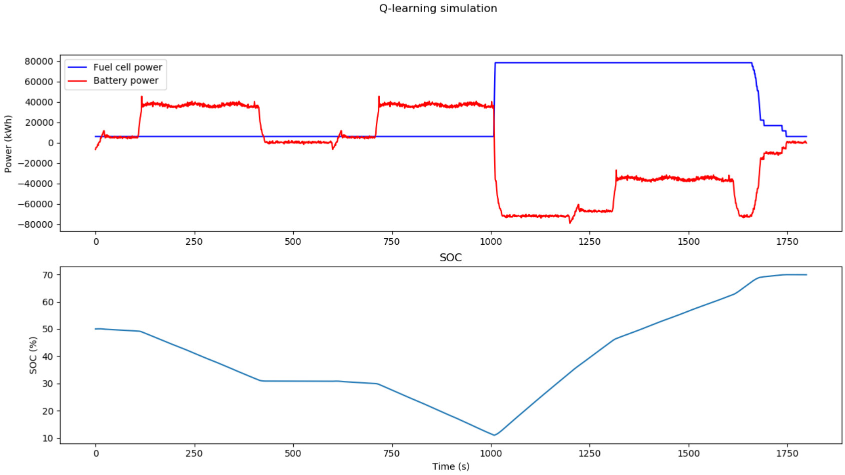


Figure 7.17: Power split and SOC for DQL control

Figure 7.17 shows the simulation results for the FC and battery power, as well as the battery SOC. The fuel cell power is very stable during the simulation, which should result in low transient costs. The power seems stable, and increases when the SOC of the battery goes low. However, the algorithm is not able to keep the battery SOC above 30 %, which leads to big additional costs. This was the case for all simulations performed using DQL, and is the main reason that it's performance is suboptimal. Still, it seems like that the algorithm learns to avoid charging the battery over 70 %, bypassing costs related to overcharging. The reason might be reflected in one of the weaknesses of using a Q-function. Even minuscule differences in Q-values will result in the agent picking the higher one. Therefore, in some cases, there might be very small differences causing the agent to pick a suboptimal action over an optimal one. Nonetheless, the reason might also be an unknown factor. This is one of the significant drawbacks using neural networks. They serve as function approximators that can approximate almost any function. They are complex in nature, and it is impossible to know how they learned what results to give. Despite the huge advances in machine learning the last couple of years, the field is still immature when it comes to understanding how the algorithms arrive at their results.

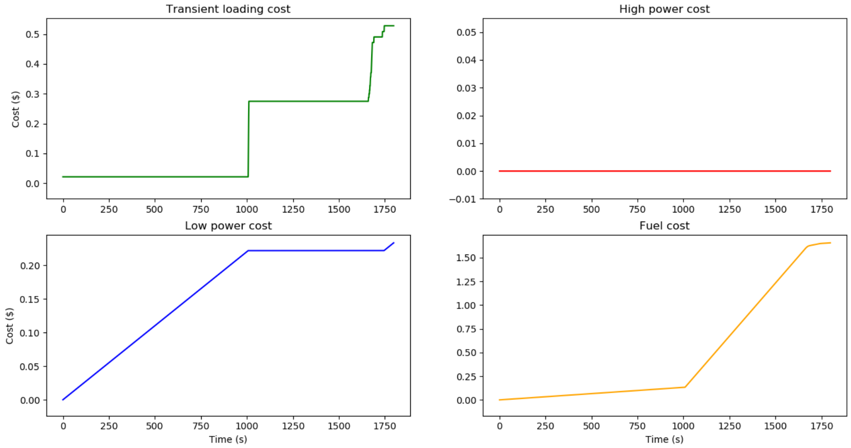


Figure 7.18: Fuel cell costs for DQL control

Figure 7.18 shows the fuel cell costs during the testing of the DQL algorithm. The overall costs are low compared to other methods. Low power costs are quite large, compared to what other algorithms yield, however the transient loading costs are the best among the algorithms tested. This is well reflected in the power chart.

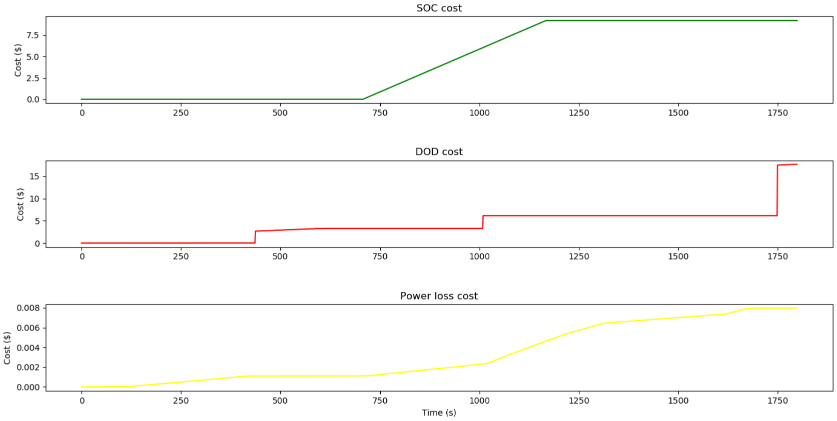


Figure 7.19: Battery costs for DQL control

The costs of operating the battery are given in Figure 7.19. The SOC costs increase significantly in the interval where the SOC is below 30 %. The power loss is negligible, which is generally the case for all algorithms, as a result of the low battery internal resistance. The battery has few cycles, which makes the DOD costs seldom occur. Nonetheless, the figure shows that when it occurs, it has a massive effect on the cost. This indicate that operating the battery using smaller cycles might be beneficial in terms of cost. However, the DOD costs is not considered by the algorithm due to it's limited performance without DOD as discussed previously in this section. Therefore, the DOD cost performance is random and does not give any indication on the actual algorithmic performance.

The total cost of operating the system with DQL, DOD excluded, is given in Figure 7.20. The costs sum to \$11.582, where the huge majority is the SOC costs. It should be noted that the SOC cost was intended as a soft constrain on the SOC. The fact that the DQL algorithm was not able to learn this makes it's performance seem worse than it actually is, since the SOC cost is the only penalty with an arbitrary value. However, there is a significant weakness of the algorithm that it is not able to learn such simple soft constraints. It might be that the algorithm is able to learn it if it is the only penalty, but in combination with other penalties it struggles with learning a good behavior that limits the SOC change. Tuning of hyperparameters or changing the numbers of hidden units or layers in the NN might improve the performance, but the effect of these changes are hard to predict and multiple simulations of trial and error is required to success.

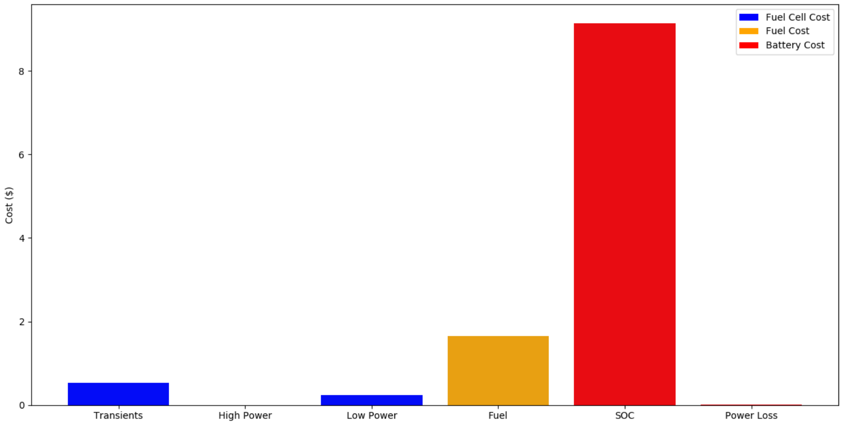


Figure 7.20: Total costs of DQL control

7.2.5 Soft actor-critic

Implementation

The soft actor-critic algorithm, in contrast to the Q-value algorithms, trains a policy network that is used for selecting actions. The code is based on the spinning up open AI code [73], [74]. It takes in the state of the system, and outputs an action in the range of $[-28\ A, 28\ A]$, which represent the change in current over a time step of 1 second. One of the main advantages of using this algorithm is that it is able, unlike the other algorithms discussed, to learn both continuous action and state values. SAC performed well using the three same input states as DQL. Because of the promising results, DOD was included in the state variables and the reward function. The state variables were normalized, while the DOD is by definition between 0 and 1. The state for the actor-critic model is thus as follows:

$$\mathbf{s} = \left[\frac{SOC}{SOC_{max}}, \ \frac{I_{FC}}{I_{FC,max}}, \ \frac{P_{load}}{P_{load,max}}, \ DOD_{charge}, \ DOD_{discharge} \right] \tag{7.2.6}$$

The resulting ANN architecture for the SAC implemented is visualized in Figure 7.21.

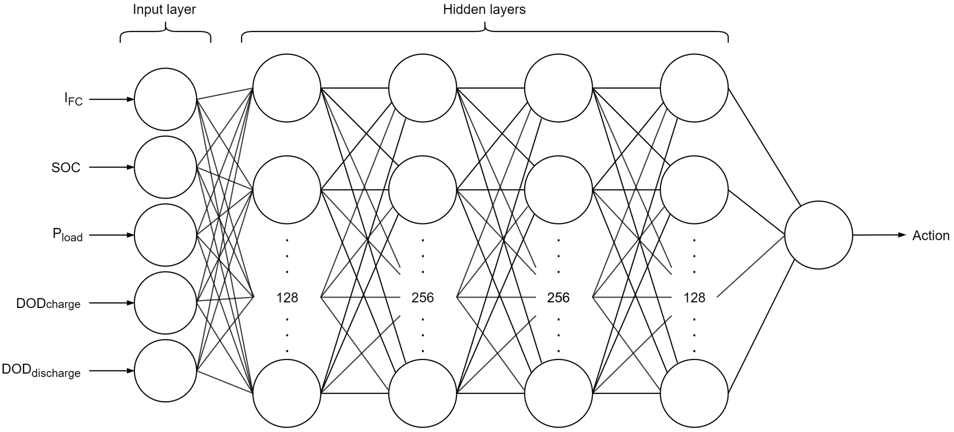


Figure 7.21: The ANN architecture of the SAC algorithm

The soft actor-critic algorithm is the only algorithm of those tested that was able to perform well using five states and including costs in the reward function. One of the advantages using SAC is that it is not very dependent on hyperparameter tuning. This is a huge advantage as this typically is very time-consuming for many RL applications. As an example, we implemented DDPG, another actor-critic algorithm with continuous state and action space, but after a days of hyperparameter tuning without good result, we ended up implementing SAC instead. Despite the comparatively low influence of hyperparameters, the reward function needs the correct tuning for the network to learn properly. It is inverse correlated with how stochastic the optimal policy is. Therefore, at lower rewards, the probability of different action choices becomes uniform, but for too high rewards, the policy is nearly deterministic [74]. As a result, we had to tune the reward function, and ended up scaling it up by a factor of 50. The total reward function used for SAC can thus be written in the following way:

$$R_{SAC} = -50 \cdot (C_{fuel} + C_{bat,deg} + C_{FC,deg}) \tag{7.2.7}$$

The network dimensions was the same as for the DQL, with 4 hidden layers of 128, 256, 256 and 128 neurons respectively. The learning rate used for the algorithm was set to $1 \cdot 10^{-4}$. As the SAC algorithm includes DOD related costs, the issue of delayed rewards should be addressed. The DOD penalty is received by the algorithm the moment the battery goes from charging to discharging or vice versa. For high C-rates, when DOD grows, you get a massive penalty as Figure 4.2 shows. The actions leading up to the point where you go from one battery mode to another is the actions that essentially creates this penalty, whereas the action that makes the shift from charging to discharging is just what triggers the inevitable. Therefore, the discount factor was set to 0.9995 in order to put high emphasis on future rewards.

SAC was trained on the training load profile, and tested on both the training and testing load profile to validate the performance. The algorithm ran for 2000 episodes, but it converged successfully after approximately 300 episodes. The power split between the fuel cell and battery, as well as battery SOC from the simulation on the test load profile is given in Figure 7.22.

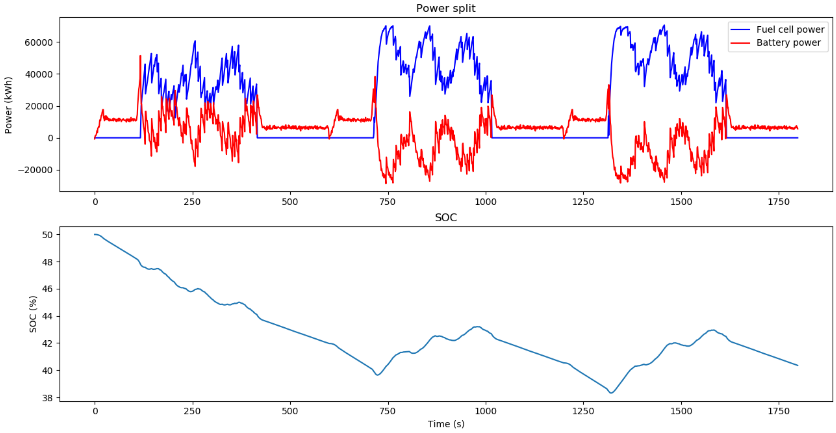


Figure 7.22: Power split and SOC of SAC control on test load profile

The algorithm delegates more FC power in the areas where the required power is high, while the FC delivers no power in low power demand areas. The control strategy gives substantial FC power oscillations for large parts of the simulation. This might be a result of overfitting on the training data. We can see in Figure 7.1 that there are some differences in the load profiles. The training load profile's high power demand regions lies at approximately 50 kW whereas the testing load profile's high power demand region lies slightly above 40 kW.

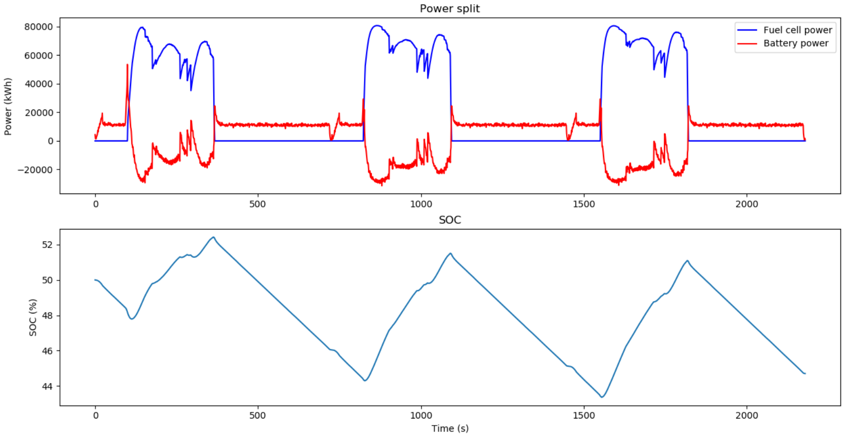


Figure 7.23: Power split and SOC of SAC control on training load profile

In the training data, a load demand of 40 kW always means that the imminent demand is either growing or decaying at a high rate. This means that the optimal next action (change in FC power) is always either the maximum if the required load is growing, or the minimum if the demand is decaying. Since this is what the algorithm trains on, rapid changes in FC power gives good results. As a result, when the test load profile is about 40 kW, the algorithm expects that the load demand goes up or down, since this was the case for the training data. This might have misled the algorithm and be a reason behind the oscillations seen in the fuel cell power delivered. This is also reflected in the simulation on the training data, given in Figure 7.23, where the oscillations in FC power are significantly lower. In order to improve this, a more varied training load profile is required.

For both the training load profile and the the testing load profile, the SOC is kept within the desired range, which proves that the algorithm is able to learn important patterns from the reward function.

The costs of operating the fuel cell with on the testing and training load profile is given in Figure 7.24 and Figure 7.25, respectively. It can be seen that the transient costs are relatively large for both load profiles. They are, however, significantly lower for the training load profile. The testing load profile uses less fuel, as the magnitude of the load peaks are lower than in the training load profile. High power is zero for both, but the low power costs are comparatively large to the other algorithms as the FC power is zero for large parts of the simulation.

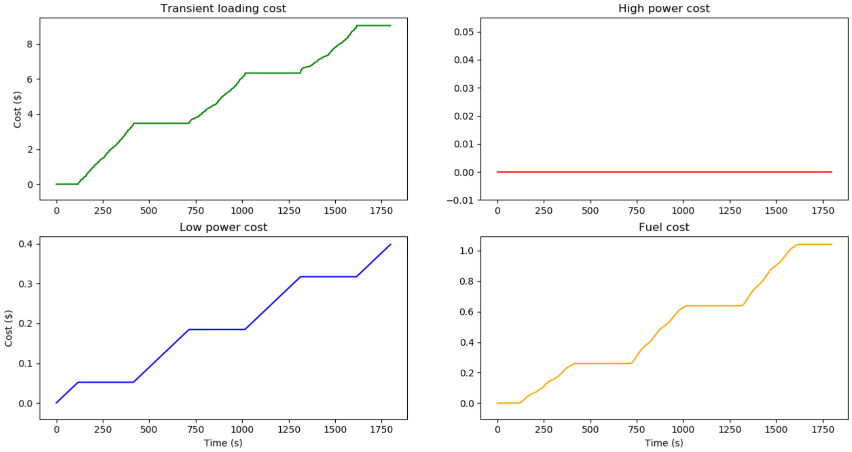


Figure 7.24: FC costs of SAC control on testing load profile

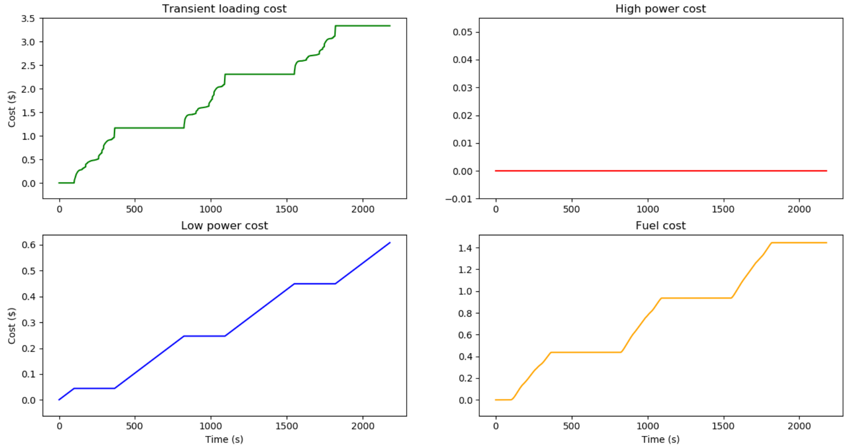


Figure 7.25: FC costs of SAC control on training load profile

The battery related costs from simulation on the test load profile are given in Figure 7.26. The SOC costs and the costs due to power loss are negligible. The DOD related cost, however, is the lowest among all the algorithms. This is expected, as SAC is the only algorithm that includes DOD in the cost calculations. It is able to reduce the battery degradation costs by 0.1 % compared with the best result from rule-based algorithm. Finding an optimal policy for reducing the DOD costs is nontrivial. The cost is calculated using a complex function, involving the C-rate and DOD, where both factors are important. High cycle depth is penalized heavily, as seen in Figure 7.19, but small DOD cycles can also lead to significant costs, which Figure 7.8 proves. The SAC algorithm manages to balance the DOD cycles at a suitable level while keeping the C-rates at a reasonable level. The DOD cycles for the simulation on the testing load profile is visualized in Figure 7.27. It can be seen that the cycle depths are low, at a magnitude of maximum 3 % of SOC, and the battery is discharging more than it is charging. This is reflected in the SOC at the end of the simulation, which is significantly lower than at the beginning.

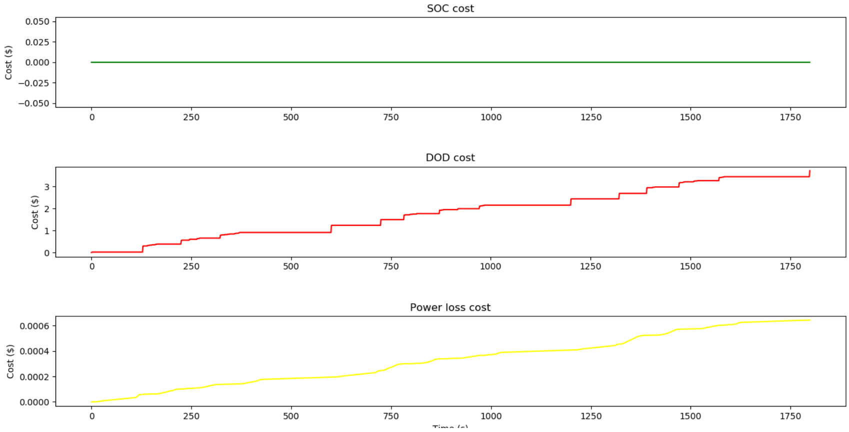


Figure 7.26: Battery costs of SAC control on testing load profile

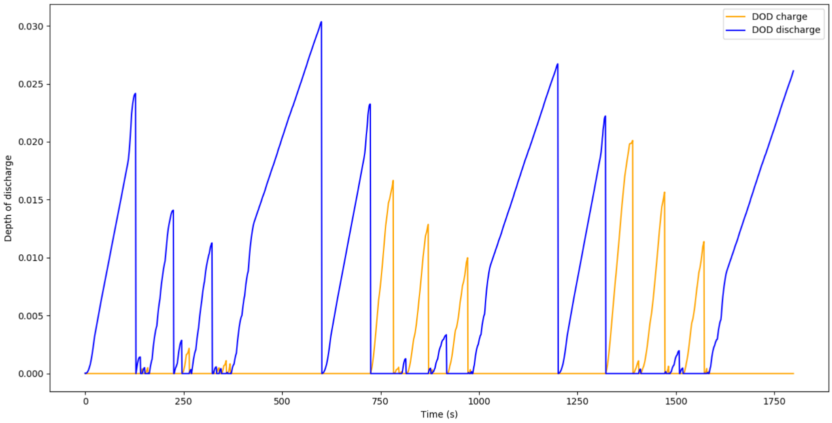


Figure 7.27: Battery DOD during SAC control on testing load profile

The total costs of the SAC algorithm on the testing load profile is given in Figure 7.28. Like discussed, the fuel cell transients costs are dominating because of the lack of more varied training data. Other than that, the performance is on par with or better than the other algorithms. The costs of both fuel and DOD is the best for all algorithms tested. The total cost for operating the system with SAC is $14.199.

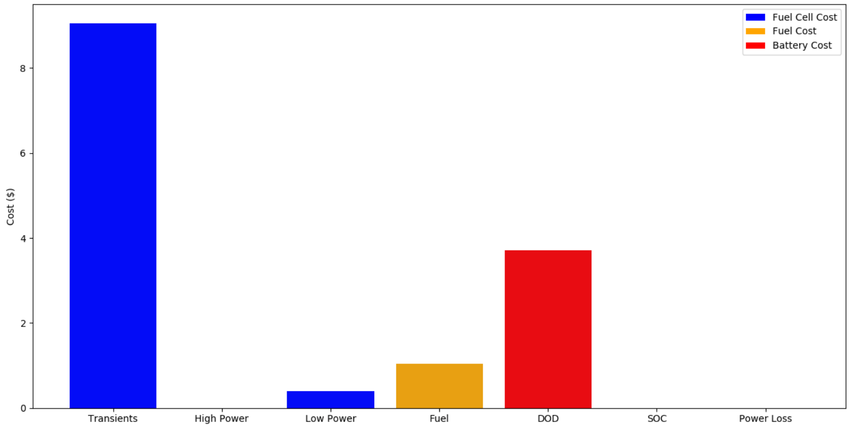


Figure 7.28: Total costs for SAC control on testing load profile

7.3 Performance and cost comparison

7.3.1 Quantitative discussion

A summary of the operating cost with the different algorithms is given in Table 7.5.

Table 7.5: Qualitative cost table

Cost comparison of control strategies

Algorithm	**Fuel cost**	**FC cost**	**Battery cost**	**Total cost without DOD**	**Total cost with DOD**
Rule-based 50 % initial SOC	\$1.663	\$1.661	\$3.715	\$3.325	\$7.040
Rule-based 65 % initial SOC	\$1.510	\$25.944	\$27.364	\$47.195	\$54.819
Q-learning	\$1.158	\$1.698	\$6.002	\$2.858	\$8.858
Deep Q-learning	\$1.653	\$0.782	\$26.503	\$11.582	\$28.938
Soft actor-critic	\$1.040	\$9.448	\$3.710	\$10.489	\$14.199

The Q-learning and SAC algorithms perform best in terms of fuel costs. They manage to reduce fuel costs by 23.3 % and 31.1 %, respectively. This indicates that they operate the fuel cell and battery at an efficient level. However, there are two things to be noted here. Firstly, SAC and DQL were evaluated based on the testing load profile, whereas the rest of the algorithms were tested on the training load profile. Despite the fact that these load profiles are quite similar, there are some differences that can result in some percentages difference in the fuel consumption. Secondly, the costs of using energy stored in the battery is not included. The reason for this is that the FC charges the battery, and that cost is taken into consideration. That way, the cost will not be counted for twice. However, in the cases where SOC$_{final}$ is below SOC$_{init}$, the battery has used some power that the FC did not provide. This cost is not accounted for in our simulations. In order to account for this, a cost could be included on the deviation between SOC$_{init}$ and SOC$_{final}$. In the cases where SOC$_{final}$ > SOC$_{init}$ this should be given as a reward to the system, as the net power from the fuel cell is higher then the power consumed by the system.

The fuel cell costs are for all algorithms dominated by the cost of transient loading and the algorithms that manage to keep transients low are performing best in this area. The high power cost is never applied for any of the algorithms. This is because the magnitude of the load is significantly less than the maximum power of the fuel cell, causing it to operate in lower power regions. The cost of low power is present, but is being dominated by the huge transient costs. The DQL is the most FC cautious of the algorithms evaluated, with a operating cost related to degradation of mere \$0.782, which is 53 % lower than the performance of the rule-based benchmark algorithm. This is reflected well in Figure 7.17, where the FC power is constant for the majority of the simulation. SAC and rule-based control with 65 % SOC$_{init}$ comes out worst in terms of FC costs. The rule-based control algorithm switches between two control strategies when the SOC reaches 70 %, causing huge fluctuations in the FC power. For the SAC, the overfitting on the training data causes large fluctuations in the high power regions of the load profile.

Battery costs are dominated by DOD and SOC costs, while the costs due to power loss is negligible as a result of low internal resistance in the battery model. It should be commented that the SOC cost is not necessarily as big as the value indicates. It was intended as a soft constraint on the SOC value. However, DQL was not able to keep SOC in the desired range, and as a result, the battery cost grew very large. SAC is the only algorithm that includes DOD costs in the reward function. Accordingly, the quality of the DQL and Q-learning algorithms can not be judged based on that cost. Anyhow, the fact that it was considered not feasible to include the DOD cost in the reward functions is definitely a drawback. For DQL, the performance with three states was not good enough that adding more complexity was a legitimate option. As for Q-learning, the state space grows such that computation times of simulations that are more than one week is deemed out of the scope.

Looking at the total costs, we see that Q-learning has the best performance when looking at costs without DOD. There is one factor that has a significantly influence on this result. Q-learning is the only algorithm that is tested on the training load profile. The lack of training data gives a sparse Q-table that cannot predict the action values from several states as a result of them never being visited during training. With DOD included, the rule-based when starting at 50 % SOC performs best, but as discussed previously, it's brittle since the initial SOC vastly affects it's performance. Among DQL and SAC, which both were tested on the testing load profile, SAC yields the best performance. SAC is bad at keeping transient fuel costs low, whereas DQL results in huge costs related to both SOC and DOD.

7.3.2 Qualitative discussion

There are several aspects of the algorithms and the training process that needs more attention. First of all, despite its mediocre performance, the SAC algorithms is evaluated to have the best potential of the RL algorithms. It is the only algorithm that was considered capable of including DOD in the state space. Furthermore, it is the only algorithm that operates in both continuous action and state space. For real applications, this is a huge advantage as it increases both the flexibility and the potential performance of the algorithms — imagine a car only able to operate at four different speeds!

There is also a huge difference in performing tests on the same load profile used for training, versus performing on an independent load profile. Real ships are not able to predict the future load, as a result of the stochastic nature of the environment it operates in Currents, waves, wind and other factors constantly influences the ships motion in unpredictable ways. The ability to learn a policy that generalizes well for unseen load profiles is therefore crucial for satisfying performance. Even so, a sufficient quantity of data is a requirement for machine learning algorithms to learn complex patterns. With more training data, the results would most likely have improved significantly.

Using RL to control PEMS of real ships is an immature field. The methods used in this paper, with using the estimated actual cost of energy system components to learn the optimal policy, is to the authors knowledge not tested before. Subsequently, there is a huge potential of improvement. The reward function that the agent tries to maximize over time is a complex RL task. It consists of seven separate functions applying costs to actions, some of which are conflicting in nature. As an example, the agent aims to minimize transients and fuel costs, while keeping the SOC above 30 %. The two former would suggest that the agent should just keep the fuel costs at zero. But this would lead to the SOC decreasing below quickly, resulting in huge SOC costs. The balance between these costs is hard to learn, and the agent is prone to being stuck in a local minimum.

The challenge of delayed rewards is also something that has to be addressed. Costs of DOD are applied only when the agent goes from charging to discharging, or vice versa, while the actions taken prior to this determines the magnitude of the cost given. It is important that the agent learns that the actions prior to the reward is essential to generating that reward. We have handled this problem by using high discount factors. However, this is probably not the best option to deal with this problem as it is time consuming, and actual value of two different states become very similar. Thus, small inaccuracies might lead to different policies. RL methods that might be applied to deal with this is Monte Carlo simulations, where you simulate for an entire episode before you update all states visited with the accumulated rewards. Another solution is to include eligibility traces, which is a technique that combines the principles of Monte Carlo simulations and temporal-difference, the technique for learning applied in Q-learning and DQL.

The algorithms implemented and tested proved promising results. All algorithms are able to learn sensible control policies that work well for some parts of the PEMS. However, the results indicate that none of the algorithms are able to yield a global optimal policy for the entire optimization. Therefore, the authors conclude that there is definitely a potential for improvement on the results presented in this chapter.

8 Conclusion

This thesis has investigated the viability of different reinforcement learning algorithms for controlling the power and energy management system of a zero-emission hybrid ship. The algorithms aim to estimate and minimize the ship's operating costs when fuel consumption and degradation of both the battery and fuel cell are taken into consideration.

The increased attention towards harmful emissions has accelerated the pursuit for zero-emission solutions in the shipping industry. Fuel cell and battery technologies are promising environmentally friendly energy sources, complimenting each other well for maritime applications. The battery has great power density and response time attributes, making it an excellent power source for load fluctuations. These qualities are inadequate in fuel cells, but their excellent energy density makes them a suitable candidate as main power source. The insufficient energy density in batteries is the main reason why they are not feasible as a primary energy source in deep-sea shipping.

To distribute the demanded load between the fuel cell and battery, a PEMS is required. Several considerations must be assessed when implementing a PEMS. In addition to the reliability and safety required, the cost of operating the system should be optimized. This includes the fuel and degradation costs, caused by operating the fuel cell and battery. Fuel costs are simple to calculate, whereas degradation costs are complex and time-consuming to estimate. When evaluating the performance of different PEMS strategies, a reward function for estimating all operational costs was established. Models for fuel cells and batteries, along with different PEMS strategies, were implemented in Python to evaluate their performance.

The thesis presents several reinforcement learning algorithms for PEMS control, aiming to minimize the operating costs. Research on learning based PEMS is still immature. Despite this, the learning algorithms are able to outperform the benchmark rule-based control method on several metrics. The deep Q-learning algorithm was able to decrease the cost of fuel cell degradation with 53 %, compared with the best performing benchmark algorithm. Soft actor-critic managed to reduce fuel cost by 31 % and the battery degradation cost by 1 %, when compared to the rule-based algorithm. The field of learning based algorithms for PEMS control definitely has a huge potential for improvement and can become the go-to method for power distribution in hybrid power systems in the years to come.

8.1 Further work

The work in this thesis revolve around the models developed by the authors, and the strategies used to control these models. As discussed in Chapter 5, both the fuel cell and battery model are based on several assumptions. Thoughts on how to improve the quality of the work is briefly discussed below.

The first suggestion is to expand the existing models. The fuel cell model only considers the voltage drop from the Ohmic region. Including varying pressures, temperatures and flow rates would result in a more sophisticated model, accurately representing a real-life fuel cell. Likewise, the battery model can be extended by including temperature effects, differentiating between charge and discharge characteristics, and introducing nonlinear effects. An improved, more precise reward function should accordingly be developed. Effects incorporated into the model should be translated to rewards that the PEMS can effectively include in its optimization.

The simulation results enlightened the drawbacks of having insufficient or uniformly distributed training data. This contributes to the problem of finding a generalized policy, providing close to optimal control when tested on new load data. It is proposed to train on several diversified load profiles, and increase the training simulations to enhance the PEMS' performance. The amount and diversity of training data is paramount for machine learning methods, in order to generalize to an approximate global optimal policy.

An alternative approach is to perform online simulations on a real ship. Although complex and resource intensive, this would yield important results. Simulation results are highly uncertain due to model inaccuracies. Testing on a real ship can validate the actual performance of the algorithms, and whether the chosen actions cohere with offline simulations. Accessing the actual power system of a specific marine vessel would also eliminate any guesswork related to component characteristics, and states could be monitored continuously.

To improve the performance of the PEMS, further work on the learning based algorithms should be conducted. The algorithms implemented show promising results, but they are not optimal. There is limited research in the field of learning based PEMS, and the potential of improvement is significant. Algorithmic challenges that should be addressed include delayed rewards, improved reliability and, testing of on-policy algorithms such as proximal policy optimization.

Recommended further work also includes extending the horizon of operations. In order to run simulations spanning several years, the model needs to be time-varying. The characteristics of all power components change during their lifetime. Such effects are negligible when looking at short time series. However, if the objective is to control the power sources for their entire lifetime, the change in characteristics should be included. By considering state of health in the optimization, optimal performance for the entire life-cycle could be achieved. Degradation effects proposed in this thesis is only included in the reward function. A time-variant model should incorporate these effects in the model itself.

These ideas can further be enlarged by envisioning a future ship with the proposed fuel cell-battery hybrid power system. Sensors enable the algorithms access to huge amounts of system data. The effects from every action and the corresponding degradation is available instantaneously, allowing the model to learn continuously. As degradation effects are somewhat uncertain, substantial amounts of data are needed to accurately model their impact on the system.

Although learning based methods in marine control systems are still in its infancy, there are many indicators that advertise its potential. Some of the untapped potential can be further explored by acquiring vast amounts of data from the power system. Data based learning methods can contribute to increase the demand for zero-autonomous vessels in the shipping industry.

Bibliography

[1] H. Ringbom, "Regulating autonomous ships-concepts, challenges and precedents", *Ocean Development & International Law*, vol. 50, no. 2, pp. 141–169, 2019.

[2] T. I. Bo and T. A. Johansen, "Battery power smoothing control in a marine electric power plant using nonlinear model predictive control", *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1449–1456, 2017.

[3] Yara birkeland — the first zero emission, autonomous ship, Yara Birkeland, [Online]. Available: `https://www.yara.com/knowledge-grows/game-changer-for-the-environment/` (visited on 12/14/2019).

[4] N. P. Reddy, M. K. Zadeh, C. A. Thieme, R. Skjetne, A. J. Sorensen, S. A. Aanondsen, M. Breivik, and E. Eide, "Zero-emission autonomous ferries for urban water transport: Cheaper, cleaner alternative to bridges and manned vessels", *IEEE Electrification Magazine*, vol. 7, no. 4, pp. 32–45, Dec. 2019.

[5] A. L. Dicks and D. A. J. Rand, "Fuel cell systems explained", in *Fuel Cell Systems Explained*, Second Edition, Chichester, UK: John Wiley & Sons, Ltd, 2018, pp. 32–35.

[6] P. Thounthong and P. Sethakul, "Analysis of a fuel starvation phenomenon of a PEM fuel cell", 2007, pp. 731–738.

[7] *Proton-exchange membrane fuel cell*, in *Wikipedia*, Dec. 11, 2019.

[8] M. Yue, S. Jemei, R. Gouriveau, and N. Zerhouni, "Review on health-conscious energy management strategies for fuel cell hybrid electric vehicles: Degradation models and strategies", *International Journal of Hydrogen Energy*, vol. Vol.44(13), pp. 6844–6861, Mar. 8, 2019.

[9] X. Li, "Thermodynamic performance of fuel cells and comparison with heat engines", in *Advances in Fuel Cells*, vol. 1, Elsevier, 2007, pp. 1–46.

[10] Hydrogen fuel cell engines and related technologies course manual, Energy.gov. Module 4: Fuel Cell Technology, [Online]. Available: `https://www.energy.gov/eere/fuelcells/downloads/hydrogen-fuel-cell-engines-and-related-technologies-course-manual` (visited on 12/06/2019).

[11] J. Han, J.-F. Charpentier, and T. Tang, "An energy management system of a fuel cell/battery hybrid boat", *Energies*, vol. 7, no. 5, pp. 2799–2820, 2014.

[12] EMSA and DNV GL, "EMSA study on the use of fuel cells in shipping", Jan. 2017.

[13] Jun Gou, Pengcheng Li, Xing Yuan, and Pucheng Pei, "Dynamic response during PEM fuel cell loading-up", *Materials*, vol. 2, no. 3, pp. 734–748, 2009.

[14] DoE. (2019). Fuel cells, Energy.gov, [Online]. Available: `https://www.energy.gov/eere/fuelcells/fuel-cells` (visited on 12/14/2019).

[15] Y. Wang, K. S. Chen, J. Mishler, S. C. Cho, and X. C. Adroher, "A review of polymer electrolyte membrane fuel cells: Technology, applications, and needs on fundamental research", *Applied Energy*, vol. 88, no. 4, pp. 981–1007, 2011.

[16] M. Momirlan and T. N. Veziroglu, "The properties of hydrogen as fuel tomorrow in sustainable energy system for a cleaner planet", *International Journal of Hydrogen Energy*, vol. 30, no. 7, pp. 795–802, Jul. 1, 2005.

[17] J. D. Holladay, J. Hu, D. L. King, and Y. Wang, "An overview of hydrogen production technologies", *Catalysis Today*, vol. 139, no. 4, pp. 244–260, 2009.

[18] I. Staffell, D. Scamman, A. Velazquez Abad, P. Balcombe, P. E. Dodds, P. Ekins, N. Shah, and K. R. Ward, "The role of hydrogen and fuel cells in the global energy system", *Energy & Environmental Science*, 2019.

[19] M. Jouin, M. Bressel, S. Morando, R. Gouriveau, D. Hissel, M.-C. Péra, N. Zerhouni, S. Jemei, M. Hilairet, and B. Ould Bouamama, "Estimating the end-of-life of PEM fuel cells: Guidelines and metrics", *Applied Energy*, vol. 177, pp. 87–97, 2016.

[20] M. A. Danzer, S. J. Wittmann, and E. P. Hofer, "Prevention of fuel cell starvation by model predictive control of pressure, excess ratio, and current", *Journal of Power Sources*, vol. 190, no. 1, pp. 86–91, 2009.

[21] J. Wu, X. Z. Yuan, J. J. Martin, H. Wang, J. Zhang, J. Shen, S. Wu, and W. Merida, "A review of PEM fuel cell durability: Degradation mechanisms and mitigation strategies", *Journal of Power Sources*, vol. 184, no. 1, pp. 104–119, 2008.

[22] J. Kurtz, C. Ainscough, and G. Saur, "V.f.10 fuel cell technology status - degradation", p. 5, 2015.

[23] H. Chen, P. Pei, and M. Song, "Lifetime prediction and the economic lifetime of proton exchange membrane fuel cells", *Applied Energy*, vol. 142, pp. 154–163, Mar. 15, 2015.

[24] C. Hähnel, V. Aul, and J. Horn, "Power control for efficient operation of a PEM fuel cell system by nonlinear model predictive control", *IFAC PapersOnLine*, vol. 48, no. 11, pp. 174–179, 2015.

[25] T. Fletcher, R. Thring, and M. Watkinson, "An energy management strategy to concurrently optimise fuel consumption & PEM fuel cell lifetime in a

hybrid vehicle", *International Journal of Hydrogen Energy*, vol. 41, no. 46, pp. 21 503–21 515, 2016.

[26] S. Zhang, X. Yuan, H. Wang, W. Mérida, H. Zhu, J. Shen, S. Wu, and J. Zhang, "A review of accelerated stress tests of MEA durability in PEM fuel cells", *International Journal of Hydrogen Energy*, vol. 34, no. 1, pp. 388–404, 2009.

[27] D. Seo, S. Park, Y. Jeon, S.-W. Choi, and Y.-G. Shul, "Physical degradation of MEA in PEM fuel cell by on/off operation under nitrogen atmosphere", *Korean Journal of Chemical Engineering*, vol. 27, no. 1, pp. 104–109, 2010.

[28] B. Xu, A. Oudalov, A. Ulbig, G. Andersson, and D. S. Kirschen, "Modeling of lithium-ion battery degradation for cell life assessment", *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1131–1140, 2018.

[29] Jianbo Yu, "State-of-health monitoring and prediction of lithium-ion battery using probabilistic indication and state-space model", *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 11, pp. 2937–2949, 2015.

[30] M. Armand and J. -M. Tarascon, "Building better batteries", *Nature*, vol. 451, no. 7179, pp. 652–657, 2008.

[31] M. Winter and R. J. Brodd, "What are batteries, fuel cells, and supercapacitors?", *Chemical reviews*, vol. 104, no. 10, in collab. with M. Winter and M. Winter, pp. 4245–4269, 2004.

[32] DNV-GL, *Handbook for maritime and offshore battery systems*, 2016.

[33] N. Harting, R. Schenkendorf, N. Wolff, and U. Krewer, "State-of-health identification of lithium-ion batteries based on nonlinear frequency response analysis: First steps with machine learning", *Applied Sciences*, vol. 8, no. 5, 2018.

[34] M. Koller, T. Borsche, A. Ulbig, and G. Andersson, "Defining a degradation cost function for optimal control of a battery energy storage system", in *2013 IEEE Grenoble Conference*, Jun. 2013, pp. 1–6.

[35] X. Ma, Y. Zhang, C. Yin, and S. Yuan, "Multi-objective optimization considering battery degradation for a multi-mode power-split electric vehicle", *Energies*, vol. 10, no. 7, 2017.

[36] P. Thounthong, V. Chunkag, P. Sethakul, B. Davat, and M. Hinaje, "Comparative study of fuel-cell vehicle hybridization with battery or supercapacitor storage device", *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 3892–3904, 2009.

[37] R. Kötz and M. Carlen, "Principles and applications of electrochemical capacitors", *Electrochimica Acta*, vol. 45, no. 15, pp. 2483–2498, May 3, 2000.

[38] P. Thounthong, S. Raël, and B. Davat, "Energy management of fuel cell-battery-supercapacitor hybrid power source for vehicle applications", *Journal of Power Sources*, vol. 193, no. 1, pp. 376–385, 2009.

[39] DOE technical targets for fuel cell systems and stacks for transportation applications, Energy.gov. Library Catalog: www.energy.gov, [Online]. Available: `https://www.energy.gov/eere/fuelcells/doe-technical-targets-fuel-cell-systems-and-stacks-transportation-applications` (visited on 04/02/2020).

[40] M. Mutarraf, Y. Terriche, K. Niazi, J. Vasquez, and J. Guerrero, "Energy storage systems for shipboard microgrids - a review", *Energies*, vol. 11, no. 12, 2018.

[41] A behind the scenes take on lithium-ion battery prices, BloombergNEF, [Online]. Available: `https://about.bnef.com/blog/behind-scenes-take-lithium-ion-battery-prices/` (visited on 05/25/2020).

[42] Y. Shao, M. F. El-kady, J. Sun, Y. Li, Q.-H. Zhang, M. Zhu, H. Wang, B. Dunn, and R. B. Kaner, "Design and mechanisms of asymmetric supercapacitors.", *Chemical reviews*, 2018.

[43] J. Thangavelautham, "Degradation in PEM fuel cells and mitigation strategies using system design and control", in *Proton Exchange Membrane Fuel Cell*. InTechOpen, 2018, pp. 67–71.

[44] Shipping 4.0 - onboard DC grid - a system platform at the heart of shipping 4.0 - generations — ABB marine, Shipping 4.0 - Onboard DC Grid, [Online]. Available: `https://new.abb.com/marine/generations/generations-2017/business-articles/onboard-dc-grid-a-system-platform-at-the-heart-of-shipping` (visited on 12/14/2019).

[45] A. J. Sørensen, *Marine cybernetics: modelling and control : lecture notes*, [5th ed.]., ser. Kompendium (Norges teknisk-naturvitenskapelige universitet. Institutt for marin teknikk). Trondheim: Marinteknisk senter, Institutt for marin teknikk, 2006, vol. UK-2006-76, viii+433.

[46] S. Chapaloglou, A. Nesiadis, P. Iliadis, K. Atsonios, N. Nikolopoulos, P. Grammelis, C. Yiakopoulos, I. Antoniadis, and E. Kakaras, "Smart energy management algorithm for load smoothing and peak shaving based on load forecasting of an island's power system", *Applied Energy*, vol. 238, pp. 627–642, 2019.

[47] Y. Tang and A. Khaligh, "On the feasibility of hybrid battery/ultracapacitor energy storage systems for next generation shipboard power systems", IEEE, 2010, pp. 1–6.

[48] J. Hou, J. Sun, and H. Hofmann, "Mitigating power fluctuations in electrical ship propulsion using model predictive control with hybrid energy storage system", in *Proceedings of the American Control Conference*, Institute of Electrical and Electronics Engineers Inc, 2014, pp. 4366–4371.

[49] T. Fletcher, R. H. Thring, M. Watkinson, and I. Staffell, "Comparison of fuel consumption and fuel cell degradation using an optimised controller", *ECS Transactions*, vol. 71, no. 1, pp. 85–97, Feb. 12, 2016.

[50] H. Li, A. Ravey, A. N'Diaye, and A. Djerdir, "Equivalent consumption minimization strategy for fuel cell hybrid electric vehicle considering fuel cell degradation", in *2017 IEEE Transportation Electrification Conference and Expo (ITEC)*, Jun. 2017, pp. 540–544.

[51] F. Martel, Y. Dube, L. Boulon, and K. Agbossou, "Hybrid electric vehicle power management strategy including battery lifecycle and degradation model", IEEE, 2011, pp. 1–8.

[52] A. M. Bassam, A. B. Phillips, S. R. Turnock, and P. A. Wilson, "Development of a multi-scheme energy management strategy for a hybrid fuel cell driven

passenger ship", *International Journal of Hydrogen Energy*, vol. 42, no. 1, pp. 623–635, 2017.

[53] Y. Wang, S. J. Moura, S. G. Advani, and A. K. Prasad, "Power management system for a fuel cell/battery hybrid vehicle incorporating fuel cell and battery degradation", *International Journal of Hydrogen Energy*, vol. 44, no. 16, pp. 8479–8492, 2019.

[54] P. Pei, Q. Chang, and T. Tang, "A quick evaluating method for automotive fuel cell lifetime", *International Journal of Hydrogen Energy*, vol. 33, no. 14, pp. 3829–3836, 2008.

[55] W. Jing, C. Lai, W. S. Wong, and M. D. Wong, "Cost analysis of battery-supercapacitor hybrid energy storage system for standalone PV systems", in *4th IET Clean Energy and Technology Conference (CEAT 2016)*, Nov. 2016, pp. 1–6.

[56] J. Wang, P. Liu, J. Hicks-Garner, E. Sherman, S. Soukiazian, M. Verbrugge, H. Tataria, J. Musser, and P. Finamore, "Cycle-life model for graphite-LiFePO 4 cells", *Journal of Power Sources*, vol. 196, no. 8, pp. 3942–3948, 2011.

[57] L. Chen, Y. Tong, and Z. Dong, "Li-ion battery performance degradation modeling for the optimal design and energy management of electrified propulsion systems", *Energies*, vol. 13, no. 7, p. 1629, 2020.

[58] S. N. Motapon, O. Tremblay, and L.-A. Dessaint, "A generic fuel cell model for the simulation of fuel cell vehicles", in *2009 IEEE Vehicle Power and Propulsion Conference*, Sep. 2009, pp. 1722–1729.

[59] MathWorks. Implement generic hydrogen fuel cell stack model - simulink, Simulink Documentation, [Online]. Available: `https://www.mathworks.com/help/physmod/sps/powersys/ref/fuelcellstack.html` (visited on 01/21/2020).

[60] D. Feroldi, M. Serra, and J. Riera, "Energy management strategies based on efficiency map for fuel cell hybrid vehicles", *Journal of Power Sources*, vol. 190, no. 2, pp. 387–401, 2009.

[61] O. Tremblay and L.-A. Dessaint, "Experimental validation of a battery dynamic model for EV applications", *World Electric Vehicle Journal*, vol. 3, no. 2, pp. 289–298, 2009.

[62] MathWorks. Generic battery model - simulink, Simulink Documentation, [Online]. Available: `https://www.mathworks.com/help/physmod/sps/powersys/ref/battery.html` (visited on 01/27/2020).

[63] Y. Zhu, Y. Chen, G. Tian, H. Wu, and Q. Chen, "A four-step method to design an energy management strategy for hybrid vehicles", in *Proceedings of the American Control Conference*, vol. 1, 2004, pp. 156–161.

[64] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: an introduction.* Cambridge, Massachusetts, London, England: The MIT Press, 2018.

[65] R. Bellman, *THE THEORY OF DYNAMIC PROGRAMMING*, in collab. with Rand Corp Santa Monica Ca, 1954.

[66] J. T. B. A. Kessels, M. W. T. Koot, P. P. J. van Den Bosch, and D. B. Kok, “Online energy management for hybrid electric vehicles”, *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3428–3440, 2008.

[67] L. Johannesson and B. Egardt, “Approximate dynamic programming applied to parallel hybrid powertrains”, *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3374–3379, 2008.

[68] E. Keogh and A. Mueen, “Curse of dimensionality”, in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds., Boston, MA: Springer US, 2017, pp. 314–315.

[69] M. Kalikatzarakis, R. D. Geertsma, E. J. Boonen, K. Visser, and R. R. Negenborn, “Ship energy management for hybrid propulsion and power supply with shore charging”, *Control Engineering Practice*, vol. 76, pp. 133–154, 2018.

[70] S. J. Moura, J. L. Stein, and H. K. Fathy, “Battery-health conscious power management in plug-in hybrid electric vehicles via electrochemical modeling and stochastic control”, *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 679–694, 2013.

[71] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning”,

[72] S. Kim, K. Asadi, M. Littman, and G. Konidaris, “Removing the target network from deep q-networks with the mellowmax operator”, in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, vol. 4, International Foundation for Autonomous Agents and Multiagent Systems IFAAMAS, 2019, pp. 2060–2062.

[73] 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 [cs, stat]*, Jan. 29, 2019.

[74] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”, in collab. with S. Levine, 2018.

[75] L. W. Y. Chua, T. Tjahjowidodo, G. G. L. Seet, and R. Chan, “Implementation of optimization-based power management for all-electric hybrid vessels”, *IEEE Access*, vol. 6, pp. 74 339–74 354, 2018, Publisher: IEEE.

[76] R. R. Chan, L. Chua, and T. Tjahjowidodo, “Enabling technologies for sustainable all — electric hybrid vessels (invited paper)”, in *2016 IEEE International Conference on Sustainable Energy Technologies (ICSET)*, Nov. 2016, pp. 401–406.

Oliver Stugard Os, Sebastian Thorsen Øverås

Intelligent Control Design for Power and Energy Management in Zero-Emission Autonomous Vessels

NTNU
Norwegian University of
Science and Technology