Document type	**Deliverable**
Document number	**D2.2**
Primary Author(s)	George Kostalas | Rhoé
Document Version | Status	2.0 | Final
Distribution level	PUB - PUBLIC

Project Acronym	**HYPOBATT**
Project Title	**Hyper Powered vessels battery charging system**
Project website	WWW.HYPOBATT.COM
Project Coordinator	Endika Bilbao | IKE | ebilbao@ikerlan.es
Grant agreement number	101056853

Date of deliverable: [28.07.2023] Date of submission: [31.07.2023]

©HYPOBATT has received funding from the European Horizon Europe Framework Programme (HORIZON) under grant agreement No. 101056853. This reflects only the author's views. The Agency is not liable for any use that may be made of the information contained therein.

CONTRIBUTIOR AND FORMAL REVIEWERS

	Name | Organisation	Date
Document Manager	George Kostalas (RHOE)	11/07/2023
Contributor 1	Olmos Josu (IKERLAN)	29/05/2023
Contributor 2	Paulo Cardoso (BRING)	19/06/2023
Contributor 3	Hugo Joudrier (CEA)	06/07/2023
Contributor 4	Stefanos Sekis (RHOE)	11/07/2023
Internal reviewer 1	Álvaro Reina (BRING)	17/07/2023
Internal reviewer 2	Endika Bilbao (IKERLAN)	16/07/2023
Internal reviewer 3	Paulo Cardoso (BRING)	17/07/2023
External reviewer 1	Maraike Pommer (FRISIA)	28/07/2023
External reviewer 2	Foad Gandoman (RSTER)	28/07/2023

DOCUMENT HISTORY

Document Version	Date	Author | Organization	Description
0.1	2023-04-06	George Kostalas (RHOE)	Initial version
1.0	2023-07-11	George Kostalas (RHOE)	First internal released version
1.1	2023-07-26	George Kostalas (RHOE)	Internal Reviewed version
1.2	2023-07-31	George Kostalas (RHOE)	External Reviewed version
1.5	2023-07-31	George Kostalas (RHOE)	Edited final version
2.0	2023-07-31	George Kostalas (RHOE)	Final version – Submitted to EC

Approval Status			
	Organisation Name	Checked By	Approved
WP Leader Approval	Rhoé Urban Technologies	Georgios Kostalas	Date: 31.07.2023 Signature:

Table of contents

LIST OF ABBREVIATIONS, ACRONYMS AND DEFINITIONS

Abbreviation	Word
API	Application Programming Interface
CC	Constant Current
CV	Constant Voltage
DB	Database
DT	Digital Twin
EMS	Energy Management Strategy
HIL	Hardware-In-the-Loop
ID	Identification
KPI	Key Performance Indicator
OSM	OpenStreetMap
PI	Performance Indicator
PV	Photovoltaic
SOC	State-of-Charge
SOH	State-of-Health

LIST OF TABLES

LIST OF EQUATIONS

LIST OF FIGURES

1. EXECUTIVE SUMMARY

In this report the developed control and energy management architecture is presented. The WP2 Digital Twin's general architecture is described. The overall framework is divided into three sections: the Database, the Upper Control Layer and the Lower Control Layer. The control objective of the upper control layer includes the generation of vessel charging scenario based on big data. In this context, big data refers to vast quantities of data that are too complex, fast-changing, or large to be effectively processed using traditional data-processing applications. In the case of the WP2 Digital Twin architecture, it's referring to a multitude of data sources relevant to vessel charging scenarios. This data can come from a variety of sources, including but not limited to: Historical charging data: Records of when and how much vessels needed charging in the past can help predict future needs. Vessel operational data: This might include details like vessel size, energy consumption rates, voyage duration, speed, and route, which would impact how and when the vessel needs charging. Environmental data: Things like weather patterns, sea conditions, and seasons can affect vessel operations and, consequently, their charging needs. Infrastructure data: This includes information about the charging stations themselves, such as their locations, capacity, availability, and technical specifications. Concerning the lower control layer will be the generation of optimal planning for charging vessel based on the algorithm developed in the upper control layer. The database will include inputs and outputs from the two layers. Two algorithms, one for each layer, were created and they are presented in the deliverable. For the scope of the project, the solution will be built for the Frisia port's case (Norddeich-Norderney ports). This deliverable uses the case scenarios produced in earlier deliverables to present some test runs that were performed with approximations of the parameters in the models of the real system. This initial iteration lays the groundwork for future enhancements in the months to come, including the proper model parametrization and any potential additional features that would be appropriate over the long term.

This report provides a comprehensive overview of the developed control and energy management architecture. The primary focus is on the description of the general architecture of the WP2 Digital Twin.

To support the functionalities of the upper and lower control layers, a comprehensive Database is incorporated into the architecture. It stores relevant information required for decision-making and facilitates the exchange of data between the layers as well as accepting physical system inputs from the port charging system.

The specific case scenario chosen for this project is the Frisia port's case, focusing on the Norddeich-Norderney ports. This particular setting serves as the context for implementing and testing the developed solution. To demonstrate the efficacy of the proposed solution, a series of test runs were conducted using approximations of the parameters in the models of the real system. These initial iterations provided valuable insights and formed the foundation for future enhancements.

As the project progresses, further developments and refinements will be made. This includes proper model parametrization, ensuring accurate representation of the real system, and the addition of any relevant features that can enhance the overall functionality of the architecture. These enhancements will be crucial in the later stages of the project when integrating the solution into the cloud platform in Work Package 2, specifically Task 2.3 that is called Operational Optimisation.

Keywords: Digital Twin, Control Management, Database, Upper Control Layer, Lower Control Layer, Cloud Platform

2. OBJECTIVES

Main objective: The developed control and energy management architecture has been designed with several objectives in mind. The main goal is to facilitate efficient vessel charging specifically at Frisia port. Utilizing big data analysis to optimize the charging process, this architecture promotes robust energy management and diminishes vessel charging costs. These steps contribute to an extended battery lifetime and lesser degradation.

Initially, a comprehensive review of the entire architecture will be provided. This will be followed by an in-depth analysis concentrating on the database, as well as the upper and lower control layers.

A key aspect of the architecture is the integration of big data. The upper control layer utilizes a variety of data sources to make informed decisions in vessel charging scenarios. By considering factors such as energy demand, vessel schedules, and environmental conditions, the architecture generates charging strategies that are optimized for each specific situation.

Looking ahead, the architecture lays the groundwork for future advancements in the field. It allows for proper model parametrization and the incorporation of additional features that can further improve the efficiency and effectiveness of the vessel charging process. Furthermore, the architecture is designed to be seamlessly integrated into a cloud platform, enhancing accessibility, scalability, and overall functionality of the solution.

This will be integrated online to a cloud solution to do online simulations, optimize the overall performance and daily operation plans of the system according to the defined requirements and provide flexibility to any specific EMS needed.

Additional objectives:

- Showcase the initial version of the model integrating the algorithms from both the Upper and Lower Control layers.
- Modify the model to suit the operational environment and enhance scalability.

3. INTRODUCTION

The maritime industry plays a crucial role in global trade and transportation, with vessels serving as the backbone of these operations. However, the growing demand for energy-efficient and sustainable solutions has prompted the need for innovative approaches to vessel charging and energy management. To address these challenges, the development of advanced control and energy management architectures has gained significant attention in recent years.

This report presents a comprehensive overview of a developed control and energy management architecture designed to optimize vessel charging in the maritime industry. The architecture incorporates elements of digital twin technology, big data analysis and layered control structures to enhance energy efficiency and minimize the cost of charging for vessels. The primary objective of this architecture is to ensure efficient charging of vessels in the Frisia port's case, focusing on the Norddeich-Norderney ports.

Building upon the insights from scientific research, the developed control and energy management architecture encompasses three main sections: the Database, the Upper control layer, and the Lower control layer.

To validate the architecture's performance and effectiveness, a series of test runs were conducted using approximations of the parameters in the models of the real system. These initial iterations provided valuable insights and formed the foundation for future enhancements and refinements.

Looking ahead, the developed control and energy management architecture paves the way for future advancements in the field. It allows for proper model parametrization, accurate representation of the real system, and the integration of additional features to further improve efficiency and effectiveness. Furthermore, the architecture is designed to be seamlessly integrated into a cloud platform, enabling online simulations, optimized performance, and flexibility in implementing specific energy management systems.

In conclusion, the developed control and energy management architecture presented in this report offers a promising solution for optimizing vessel charging in the maritime industry. By leveraging digital twin technology, big data analysis, and layered control structures, the architecture contributes to enhanced energy efficiency, sustainability, and overall operational effectiveness. The integration of this architecture with cloud platforms opens up new avenues for online simulations, real-time optimization, and the implementation of customized energy management systems.

4. WP2 OVERALL FRAMEWORK

The Work Package 2 overall framework is presented above in Figure 1 :

This system comprises of two primary divisions: a virtual component, which includes an internal cloud backoffice, and a physical real-world component. The virtual system can be subdivided into four main entities: (1) Digital Twins Triple Ecosystem, (2) Database, (3) Upper Control Layer, and (4) Lower Control Layer. Digital Twins Triple Ecosystem is a digital replication of the real-world system, used to model and predict system behaviour. Database is where the output of the digital twin system gets stored, including data from the internal cloud platform. Upper Control Layer is the processing center that receives data from the database and runs optimization algorithms to generate estimated control parameters for the lower control layer. Lower Control Layer is responsible for implementing the energy management strategy based on the control parameters received from the upper control layer. The Lower Control Layer is also tested through a hardware-in-the-loop system, a form of real-time simulation that combines software and physical systems. The real-world system includes: (1) The Port Charging System, and (2) The Communication Cloud (represented by the yellow dotted line). The Port Charging System communicates with the Digital Twins Triple Ecosystem to exchange standard data such as voltage, temperature, intensity, and grid limitations. It also facilitates fast communication during emergencies. The Communication Cloud forms the communication interface between all parts of the system, managed by the port in the HYPOBATT project. The port charging system also communicates directly with the energy management strategy to exchange standard operations data and modes of operation. After the Lower Control Layer processes all the data, it adjusts the Digital Twin and validates the internal strategy. The Upper Control Layer is tested through a hardware-in-the-loop system and communicates directly with the Communication Cloud platform. This, in turn, directly communicates with the Port Charging System, closing the communication loop among all parts of the system.

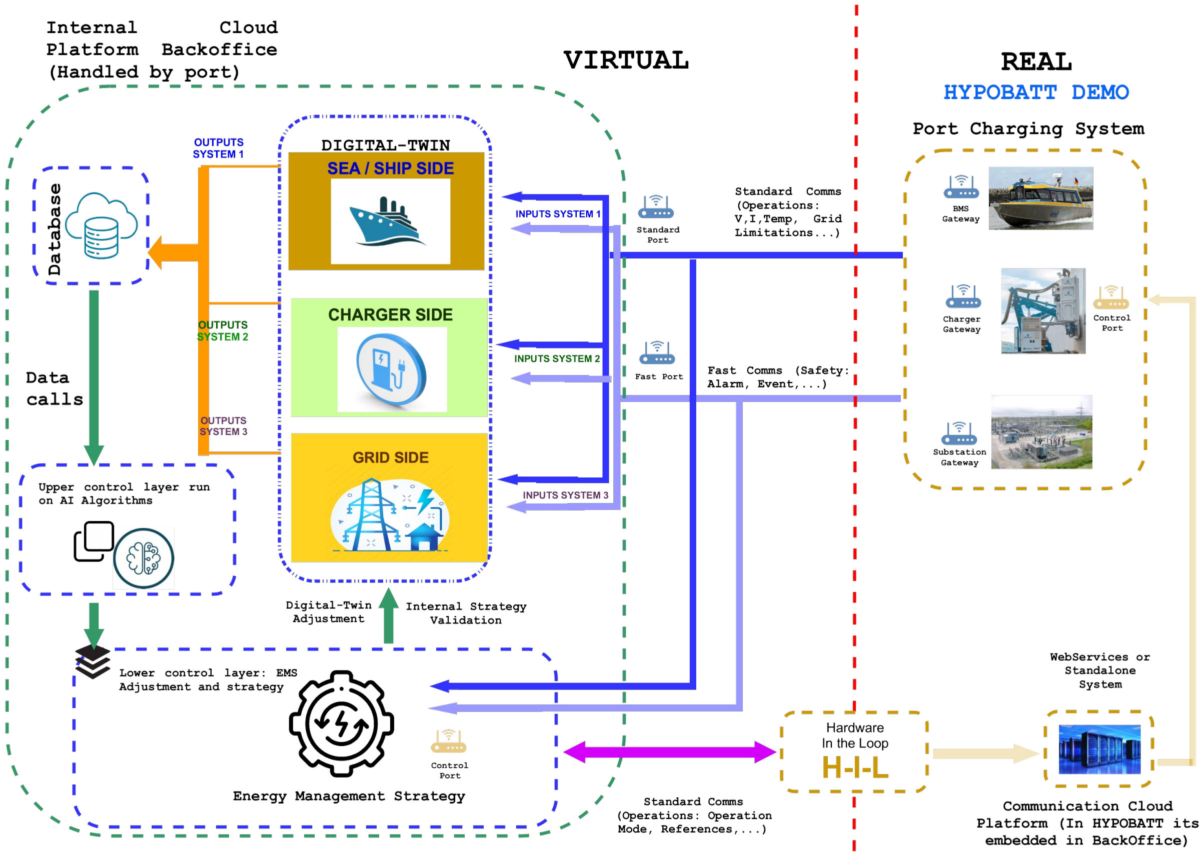

Figure 1 WP2 System Representation

4.1 Virtual system

4.1.1 Digital Twin-Triple ecosystem

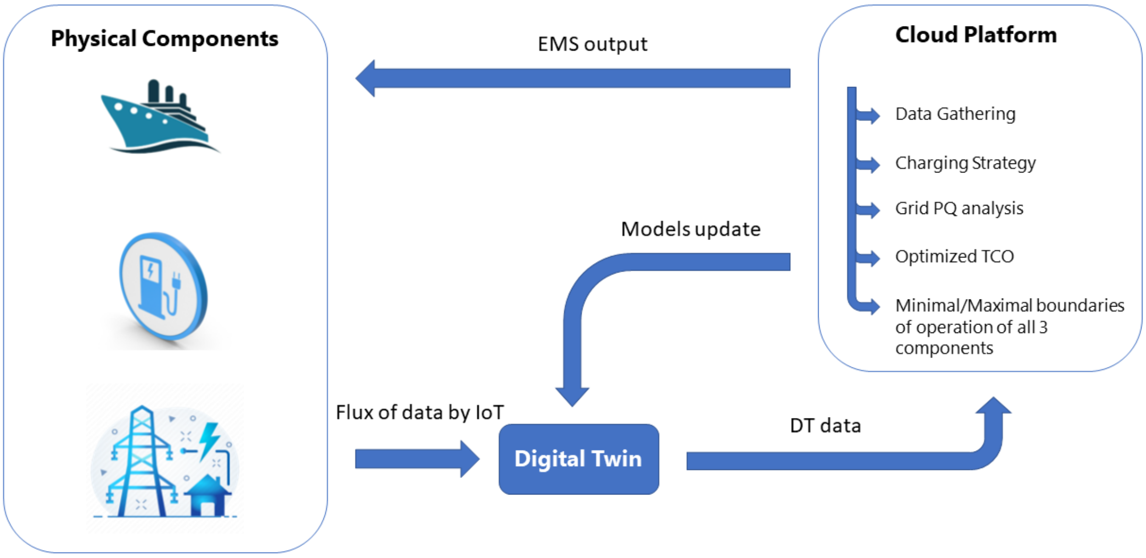

Figure 2 Overall architecture of the Digital Twin.

The system architecture that's been developed integrates three sub models representing the battery, the charger, and the grid. These are encompassed within a Digital Twin (DT) framework which facilitates real-time data transmission and processing.

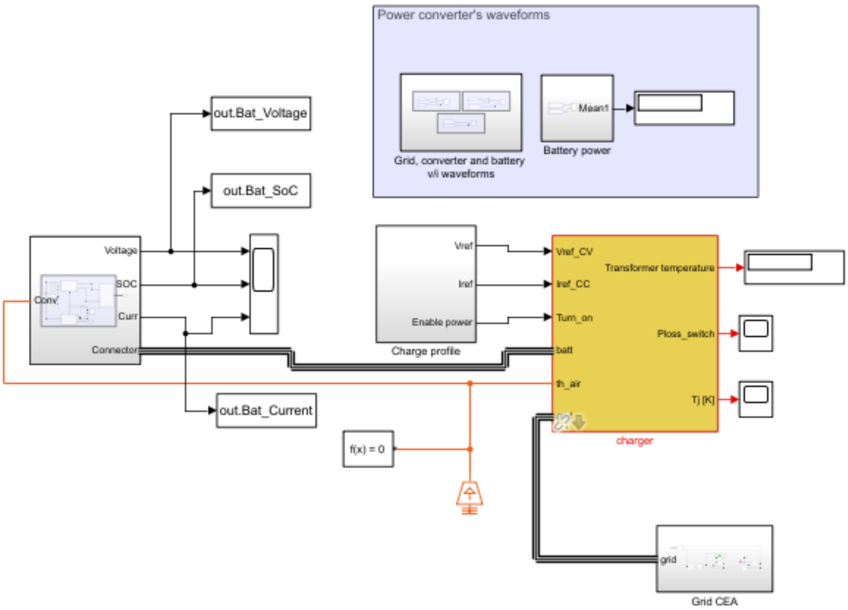

Figure 3 Digital twin model developed in Simulink.

The DT is essentially a virtual replica of these three physical entities. It receives real-world data, combines it with its own output, and presents an insightful overview of the entire system as depicted in Figure 2Figure 2.

By exploiting this framework, we can precisely estimate the power requirement from the grid, including the timing for power deployment. This allows us to effectively optimize energy usage based on financial considerations and also preserve the health of the battery.

The DT's algorithm is designed for continual learning. It improves its predictions over time by consistently learning from the real system's inputs, thereby leading to substantial cost savings and streamlined port operations.

The DT uses input variables from the battery, charger, and grid to provide corresponding output variables. These become the basis for making well-informed decisions regarding power usage and system performance optimization.

In essence, the integration of the sub-models into the DT framework fosters precision, optimization, and predictive capabilities. These advantages contribute to overall cost savings, energy efficiency, enhanced battery health, and smooth port operations. A broader view of DT foundations can be found in D2.1 Report on the digital twin of the whole system of the HYPOBATT project.

4.1.2 Database Introduction

A Mongo DB structure was developed to create a database capable of storing the required inputs from the Digital Twin. MongoDB stands out due to its schema flexibility, scalability, strong performance, and robust geospatial support. It effectively handles evolving business requirements, large data volumes, and geographical data, ensuring efficient and speedy operations. Given these advantages, MongoDB aligns well with the project's needs, making it an optimal choice. The database architecture is designed to efficiently store and manage the data needed for the system.

The database serves as a direct communication channel between the upper control layer and the Digital Twin. It facilitates the seamless exchange of data, ensuring that all the necessary information is readily available for analysis and decision-making.

It is important to note that the management and administration of the database will be handled by the port as an integral part of their internal cloud platform Backoffice. This ensures that the database is securely maintained and accessible within the port's infrastructure.

By utilizing the Mongo DB structure, the database provides a reliable and scalable solution for storing and retrieving the inputs required by the Digital Twin. This seamless integration enhances the overall functionality and efficiency of the control and energy management architecture.

4.1.3 Upper Control Layer Introduction

The upper control layer plays a crucial role in the control and energy management architecture by providing an optimization algorithm for optimal vessel scheduling. It collects data from various sources, including battery logs, charger logs, and cost information. This layer communicates with the database to gather relevant data, in addition to the inputs provided by the user.

Once the data collection is complete, the upper control layer employs a simulation function to generate feasible solutions. These solutions are evaluated to identify the optimal scheduling for each vessel, taking into account factors such as energy availability, charging requirements, and cost considerations.

It is important to note that in the case of Frisia's ports, where electricity prices are standard, the cost function is unnecessary. However, for ports like Valencia and others utilizing the system, the electricity log containing power availability and electricity prices needs to be inserted into the database as an input from the internal cloud platform managed by the port.

By incorporating the optimization algorithm and simulation function, the upper control layer facilitates efficient vessel scheduling and energy management. It ensures that vessels are charged optimally, considering available resources, operational requirements, and in applicable cases, cost factors. The integration of the database and the internal cloud platform further enhances the functionality and adaptability of the system across different port environments. Section 6 is dedicated to analyzing the objective of the upper control layer.

4.1.4 Lower Control Layer Introduction

The Lower Control Layer is responsible for generating the optimal planning for charging vessels using the developed Energy Management System (EMS) system. To accomplish this, a rule-based strategy has been chosen as the approach to provide an optimal charging protocol.

The EMS system, integrated within the Lower Control Layer, analyzes various factors and parameters related to vessel charging. By considering inputs such as energy demand, vessel schedules, and battery conditions, the system applies a set of predefined rules to determine the most efficient charging plan for each vessel.

The rule-based strategy employed by the Lower Control Layer ensures that vessels are charged in an optimized manner, taking into account factors such as energy availability, charging requirements, and operational constraints. By utilizing the EMS system and rule-based approach, the architecture enhances the overall efficiency and effectiveness of the charging process, leading to improved energy management and operational outcomes.

Section 7 is dedicated to analyzing the objective of the lower control layer.

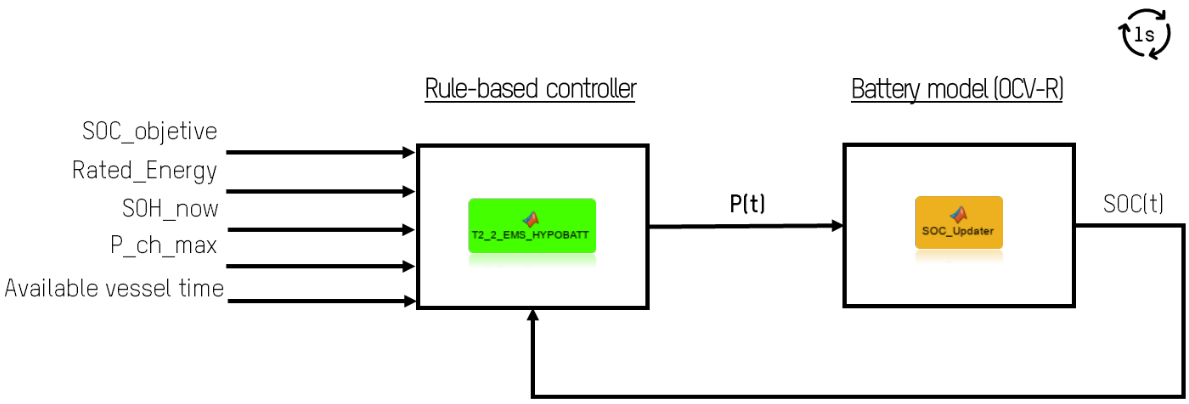

Figure 4 Lower Control layer EMS Results.

4.2 Real system; Hardware In the Loop, validation and real operation.

To verify the proper functioning of the system, a hardware-in-the-loop (HIL) setup will be installed in Brussels. The HIL setup serves as a testing environment to validate the system's performance and ensure that all components operate as intended.

The primary purpose of the HIL setup is to assess the functionality of the internal cloud platform backoffice. This involves testing the communication channels between all parties involved and ensuring that data is transmitted accurately and efficiently.

By simulating real-world scenarios within the HIL setup, the system can be thoroughly tested in a controlled environment. It allows for the validation of data transfer, data processing, and overall system integration. The HIL setup acts as a bridge between the physical components and the virtual elements, providing a comprehensive evaluation of the system's performance.

The installation of the HIL setup in Brussels demonstrates the commitment to quality assurance and thorough testing. It will be used to assess models within the system by replicating real-world conditions. It does this by generating physical signals that mimic the behaviour of the grid, charger, and vessel. Alongside these emulations, HIL will also simulate the final outputs of the lower control layer. It ensures that the system is robust, reliable, and capable of handling the data transport and communication requirements seamlessly. By validating the internal cloud platform backoffice and the communication infrastructure, any potential issues or bottlenecks can be identified and addressed, ensuring the overall functionality and efficiency of the system.

Following the examination of the HIL solution, the final depiction of the system is presented in Figure 5.

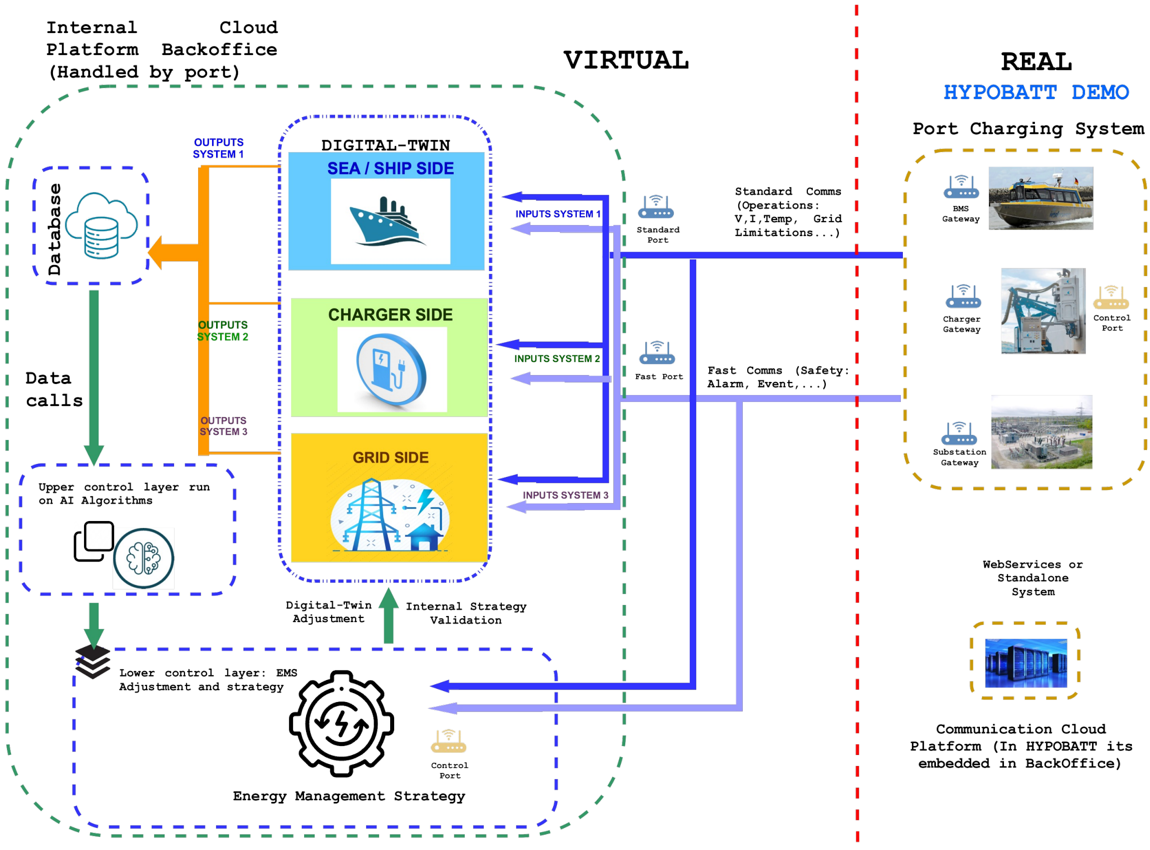

Figure 5 WP2 System Representation after HIL testing

4.3 Requirements

In this section, a detailed exploration of the operational requirements and the nature of data necessary for the effective operation of the system is undertaken. This discussion includes system requirements along with crucial data characteristics. It delves into aspects like database installation and access, user authentication, data protection, and factors such as waterways, routes, grid power, and electricity prices. Furthermore, the Key Performance Indicators (KPIs) of the project, aimed at enhancing battery life through smart charging profiles and bolstering EU's competitiveness in fast-charging technology, are explored. Later aspects cover the port's internal cloud back-office installation and its associated cybersecurity considerations. A thorough comprehension of these elements is vital for the successful operation and implementation of the architectured system. The table below furnishes a detailed list of these requirements and serves as a guiding pathway for the development process to ensure alignment with project goals. It should be noted that the table will continue to be updated regularly until HYPOBATT project conclusion.

Table 1 WP2 Requirements

Req. ID	*Scope*	Requirement Description	Class
Req_D2.1-001	*Sys*	**Standard communications from Port Charging System installed** Data includes voltage, intensity, temperature, grid limitations	*SH*
Req_D2.1-002	*Sys*	**Fast communications from Port Charging System installed** Data includes safety notifications	*SH*
Req_D2.2-001	*Sys*	**Port Database installation** Access The database can be accessed using generated credentials through MongoDB Compass or via a programming language.	*SH*
Req_D2.2-002	*Sys*	**Database Login Access Permission** Protect the database from unauthorized changes. For the particular system information may only be changed by the system's administrator.	*SH*
Req_D2.2-003	*Sys*	**Database Authentication** Authentication involves accepting credentials from the entity and validating them against an authority.	*SH*
Req_D2.2-004	*Sys*	**Database Data Protection** Data protection ensures that request and response have not been tampered.	*SH*

Req. ID	*Scope*	Requirement Description	Class
Req_D2.2-005	***Sys***	**Data inputs needed for Digital-twin´s Triple ecosystem** **1. Ship side digital twin** [Operations] SoC (initial state/status) Cells/Modules temperature (initial state/status) Cells/modules enclosure temperature (initial state/status) Nº of cycles [Safety] Overvoltage Undervoltage Excess internal temperature **2. Charger-side digital twin** [Operations] Grid Current/Voltage [Safety] Overvoltage Undervoltage Overcurrent **3. Grid-side digital twin** [Operations] PV power Loads Power Main Grid Power at PCC Grid voltage range available	***SH***

Req. ID	Scope	Requirement Description	Class
Req_D2.2-006	*Sys*	**Data inputs needed for Upper Control Layer** **1. Vessel Route Info** [Operations] Vessel Timetable (routes from port-to-port, pier to pier) Average Speed Input per waterway/Route (shallow/deep part/arrival/departure) Weather margin Move to charger time Current towards sea **2. Vessel Characteristics** [Operations] Initial Soc Total Battery Capacity Vessel hotel load Manoeuvring power Manoeuvring time System efficiency safety margin Charging target SoC **3. Charger Characteristics** [Operations] Grid power Availability **4. Waterways and Routes** Waterways length and tags (shallow/deep part, arrival/departure) from port to port and pier to pier **5. Electricity Price** Daily database updated electricity price per hour	*SH*
Req_D2.2-007	*Sys*	**KPI: Improving Battery Lifetime through intelligent optimal charging profile** Increase TRL from 4 to 7.	*SH*
Req_D2.2-008	*Sys*	**KPI: Increasing EU technological competitiveness on fast charging technology** Increase TRL from 4 to 7.	*SH*
Req_D2.3-001	*Sys*	**Port Internal Cloud Backoffice installation** TBD later in the project	*SH*
Req_D2.3-002	*Sys*	**Communication Cloud Platform connection** TBD later in the project	*SH*
Req_D2.3-003	*Sys*	**Port Internal Cloud Backoffice Cybersecurity** The Internal Cloud backoffice needs to be accompanies with a cyber security report. The report should describe the security tests related to data transfers between DT, Upper and Lower Control layer and the cloud storage and test characteristic to web portal security (i.e. OWASP).	*SH*
Req_D2.3-004	*SyS*	**Port Internal Cloud Backoffice Warning** The port internal cloud backoffice should notify the system administrator of an emergency in each of the systems by sending a warning or an alarm.	*SH*

5. DATABASE

5.1 Description of Work

As previously stated, a database has been established to house data crucial for the upper control layer. This database also interacts with the digital twin model developed in the preceding deliverable, storing variables related to relationships, routes, nodes, waterways, electricity/clearing prices, speed value profiles, and port collections.

This database functions as a direct link between the upper control layer and the Digital Twin, facilitating a smooth data transfer and ensuring that all the required information is instantly accessible for analysis and decision-making purposes.

By enhancing data accessibility, accuracy, and real-time adjustments while optimizing resources and fostering collaboration, this integration significantly improves the effectiveness and efficiency of the control and energy management structure. In creating this comprehensive system, a MongoDB database structure has been employed. It's worth noting that for task-specific needs, the database will need to be managed by the port as an internal cloud backoffice, to ensure the necessary variable updates.

5.1.1 Network Mapping

The network mapping process for the Frisia's network involved using the osmapi library to interact with the OpenStreetMap (OSM) API and fetch network data. The main task was to retrieve the specific relation ID for the waterway relation between the Norddeich and Norderney ports. By querying this relation, it was possible to extract detailed information about its members, including the waterways, piers, and nodes along the route.

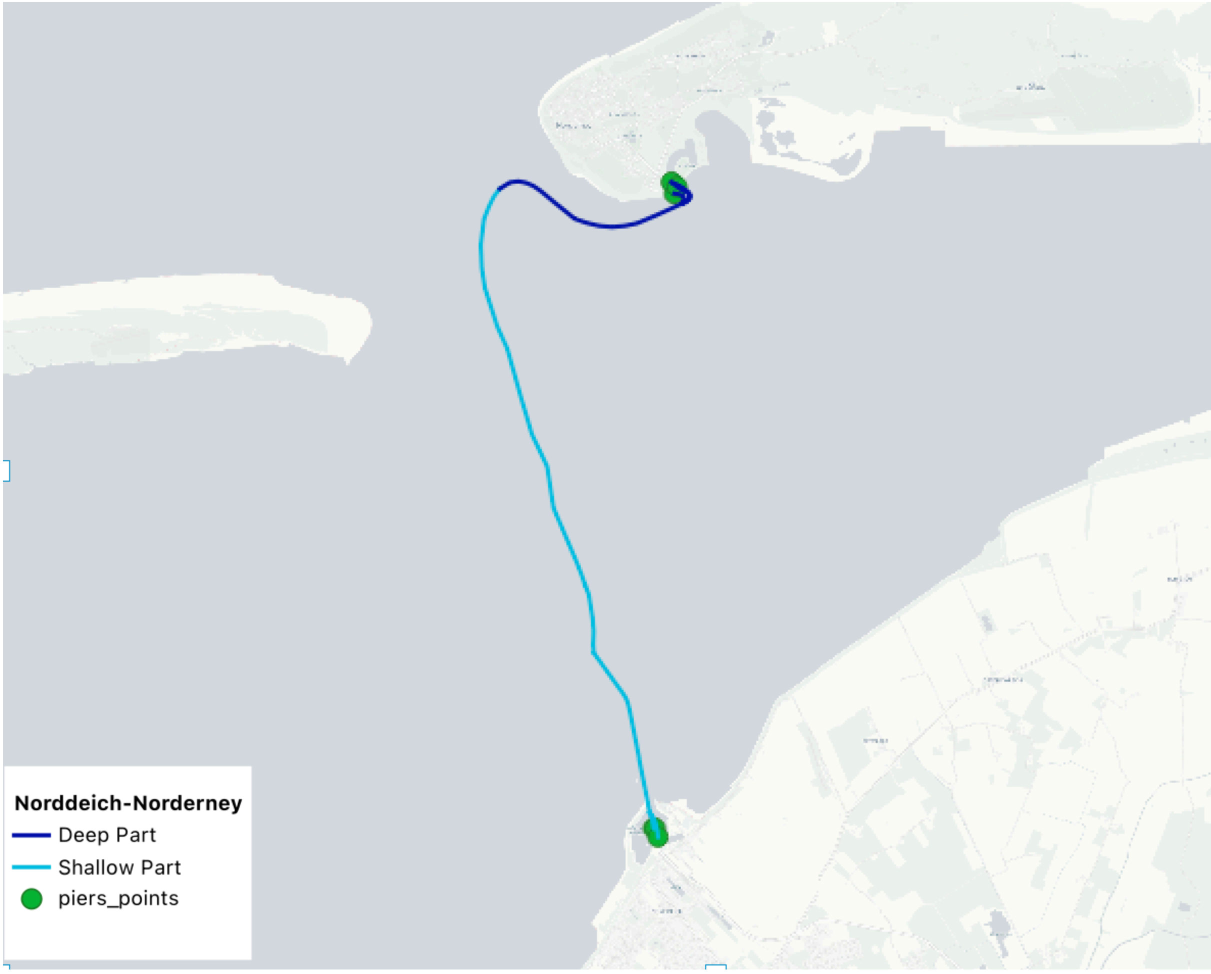

Figure 6 Norddeich-Norderney Network.

For every waterway, the associated way information, which is comprised of the nodes defining the waterway, was extracted and organized. This data was stored in a suitable data structure for easy access. Additionally, the depth of each waterway was identified using information from the Navionics website.[1] Figure 6 describes the network between Norddeich and Norderney port.

Each pier and node had relevant details such as the pier name, associated port, and coordinates extracted and stored appropriately. Figure 7 shows the available piers at Norddeich port while Figure 8 shows the ones in Norderney port.

[1] https://webapp.navionics.com/

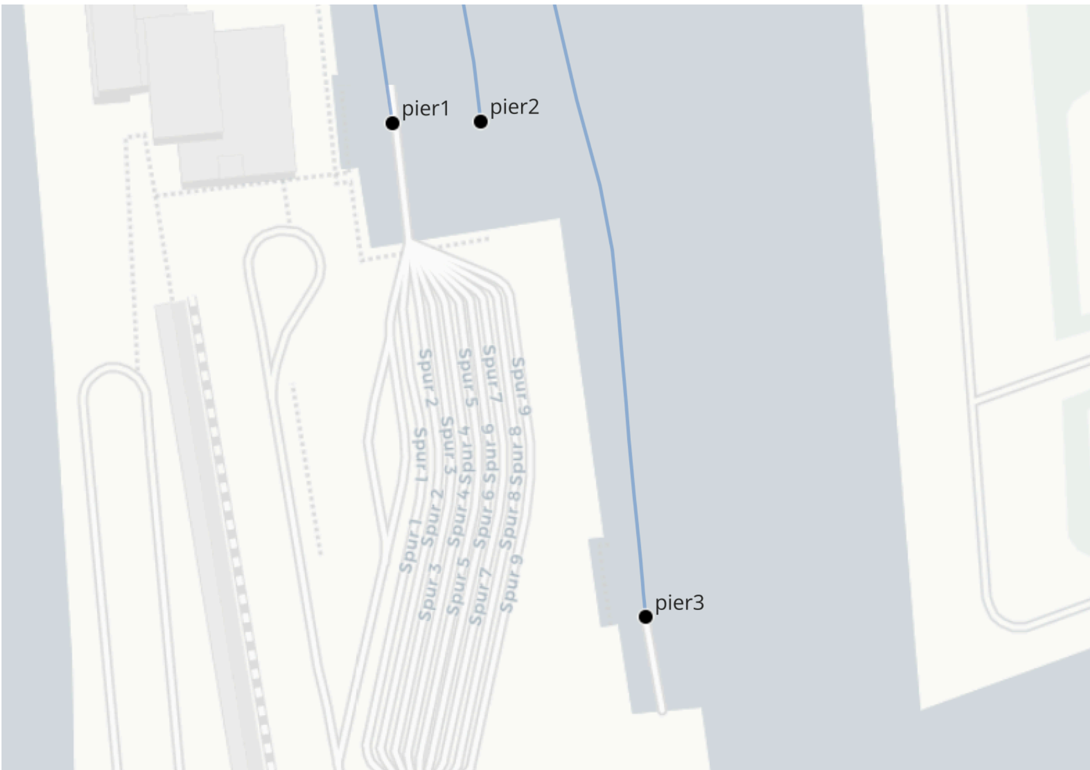

Figure 7 Piers in Norddeich port.

To manage this data efficiently, MongoDB was used to create collections for the waterways, piers, and nodes. The waterways were stored using their unique OSM IDs, and the corresponding nodes forming the waterways were saved as object IDs. Moreover, the length of each waterway was calculated by determining the distance between consecutive coordinates using the geopy library. This length information was stored alongside the waterway data in the MongoDB collection.

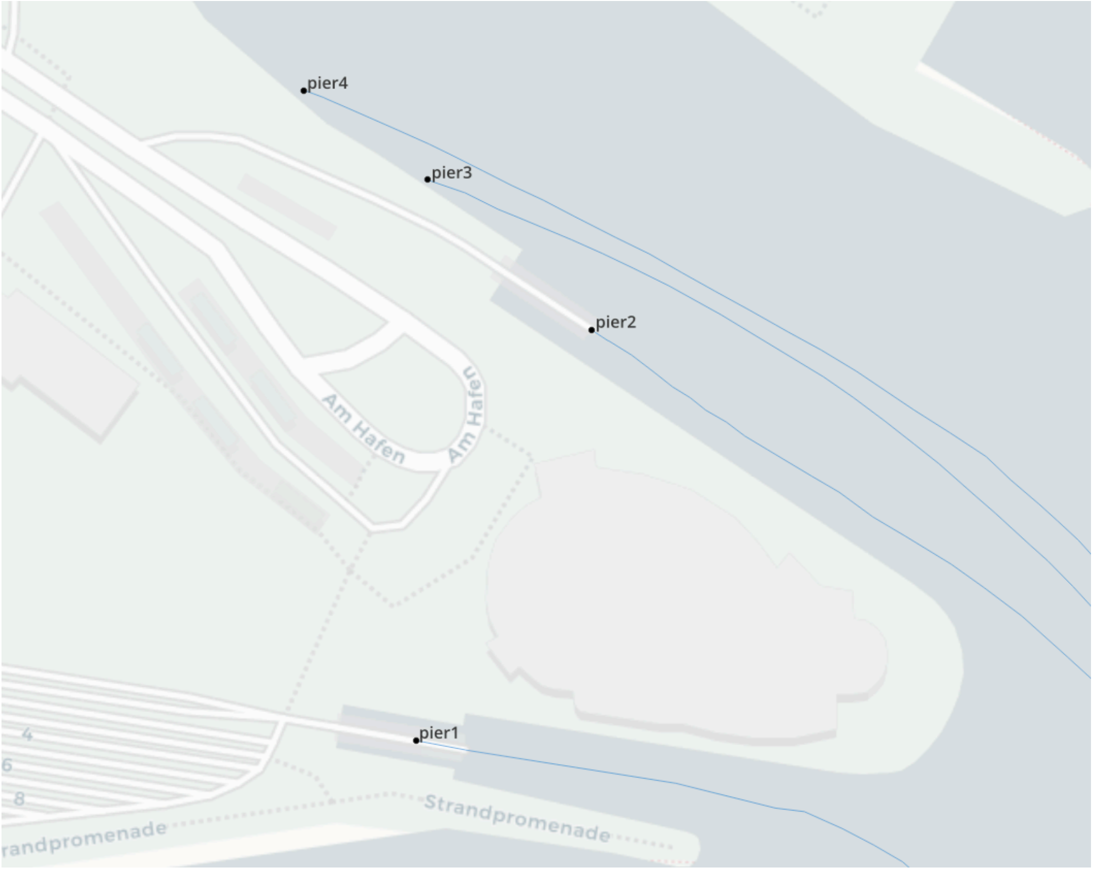

Figure 8 Piers in Norderney port.

It's important to note that this process also accounted for future infrastructure developments. Some piers did not exist at the time of data collection but were planned for construction in the future. These planned piers were manually added to the data structure, ensuring the model would be able to adapt to changes in the infrastructure, enhancing its long-term utility and applicability.

5.2 Implementation

The database was implemented using MongoDB, a NoSQL database renowned for its flexibility and scalability. MongoDB is designed to store and process large amounts of data, making it an ideal choice for handling a wide variety of data types and structures. Data in MongoDB is organized into collections, where each collection is used to hold related data. For instance, one collection might store data on vessel speed and power, while another might hold information about waterways.

Figure 9, Figure 10 display the HYPOBATT Database.

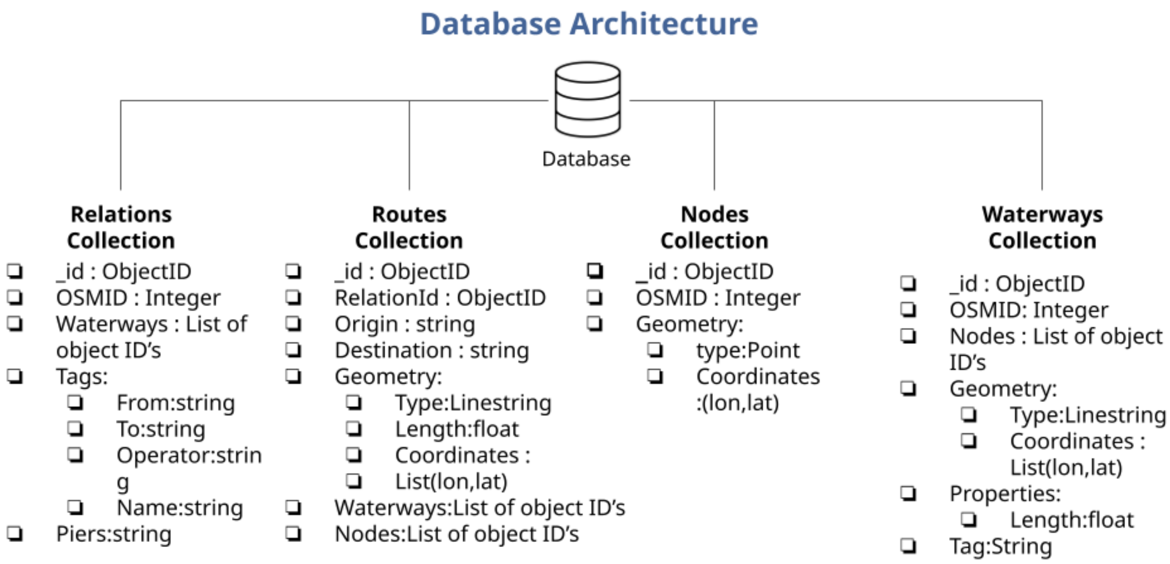

Figure 9 HYPOBATT Database Part One.

Data is stored in the form of documents within these collections. A document is a set of key-value pairs, akin to a JSON object. This allows for each document in a collection to have a unique set of fields, adding to the flexibility of the database to store diverse data types and structures.

Python scripts are used to read data from CSV files and other sources, process the data, and then insert it into the appropriate collection as new documents. These Python scripts employ data manipulation libraries like pandas for tasks like cleaning and interpolation, ensuring the integrity and accuracy of the data stored in the database.

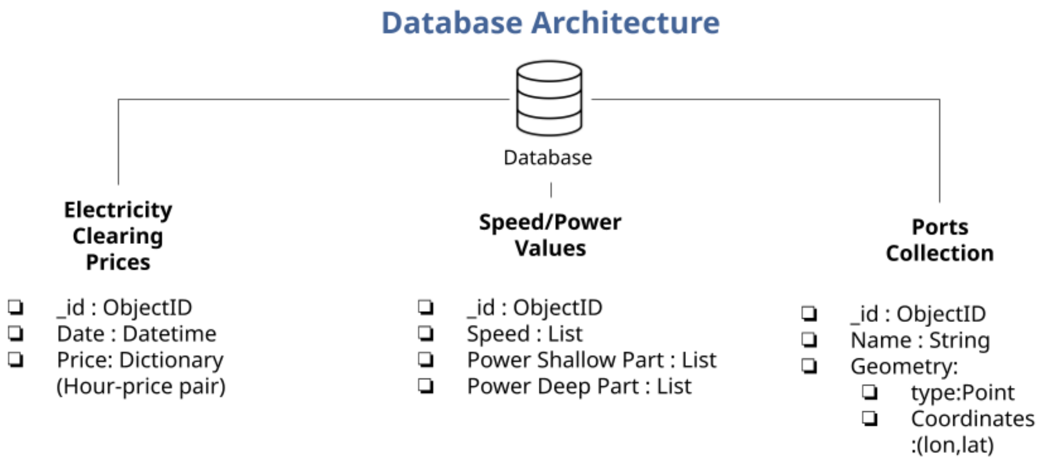

Figure 10 HYPOBATT Database Part Two.

Certain data types have tags or identifiers added to them to enhance the utility of the data. For example, waterways might be tagged based on their depth or if they serve as designated arrival or departure points. These tags aid in efficient data retrieval and processing during subsequent stages.

Moreover, MongoDB provides robust querying functionality, allowing data to be accessed, filtered, and sorted based on various parameters. This functionality enables detailed analysis and efficient data retrieval, facilitating informed decision-making.

Currently, the clearing prices for electricity were manually extracted for one day and then inputted into the database. Given the dynamic nature of the energy market where prices can fluctuate multiple times within a day, there is considerable scope for enhancing the efficiency and effectiveness of this process. Therefore, future work will aim to develop an automated solution for daily price forecast extraction.

Lastly, the database is designed with scalability in mind. As the volume of data increases, MongoDB's scalability features allow for seamless database expansion, ensuring that the system remains efficient and reliable even under increased data load.

6. UPPER CONTROL LAYER

6.1 Definition and Description of Work

The upper control layer is pivotal within the control and energy management architecture as it supplies an optimization algorithm for the best vessel scheduling and cost computation. It compiles data from a variety of sources, such as battery and charger logs, cost data, and user-inputted information. This layer interfaces with the database to accumulate pertinent data.

Once the data has been collected, the upper control layer applies a simulation function to produce potential solutions. These solutions are appraised to pinpoint the best scheduling for each vessel, factoring in elements like energy supply, charging needs, and cost considerations.

It's crucial to highlight that for ports like Frisia, where electricity prices are fixed, the cost function isn't required.

The integration of the optimization algorithm into the upper control layer streamlines vessel scheduling and energy management. It guarantees that vessels are charged in the most financial and energetically efficient manner, taking into account available resources, operational needs, and, where applicable, cost factors. The fusion of the database and the internal cloud platform bolsters the system's functionality and adaptability across diverse port environments. It's important to note that the upper control layer currently provides estimation points that must be entered through the lower control layer before the actual outcome of the charging process is inputted.

6.2 Optimisation Algorithm

6.2.1 Inputs

1. **Vessels**: This section includes all the following vessel input in the system:
    - **Timetable**: Array representing the vessel's schedule. Each list includes destination, date and time of arrival, and the pier where the vessel will stop.
    - **Speed**: This attribute represents the input speeds for the vessel at different stages of the journey, corresponding to different parts of the waterway - "Departure", "Shallow_Part", "Deep_Part", and "Arrival". Each stage has an array of speeds, mapping to each trip specified in the timetable.
    - **Others**: Various additional factors affecting the operation of the vessel.
        - **Initial_Soc**: The initial state of charge of the vessel's battery.
        - **Battery_Capacity**: Total capacity of the vessel's battery.
        - **Hotel_Load**: Power required for non-propulsive, onboard services like lighting, air conditioning, and appliances. Independent of vessel speed.
        - **Manoeuvring_Power**: Power required to manoeuvre the vessel.
        - **Manoeuvring_Time**: Time taken to manoeuvre the vessel.
        - **System_Efficiency**: A factor denoting the vessel's energy utilization efficiency. It influences the total power required for both non-propulsive services and vessel propulsion.

- **Safety_Margin**: The minimum energy level that the vessel's battery should never fall below.

2. **Chargers**: This section lists all the available chargers at different locations. Each charger has a location identifier and carries the following information:
    - **Availability**: Indicates the availability status of the charger (True or False).
    - **Gross_Charging_Power**: Indicating available power from the charger at different hours of the day.
3. **System**: Global parameters affecting the entire system.
    - **Weather_margin**: A factor representing the weather impact on the system ().
    - **Charge_Threshold**: The docked time (in seconds) that decides whether the vessel will charge or not. If docked time is above this threshold, the vessel should charge.is above this threshold, the vessel should engage in charging.
    - **Move_to_charger_time**: Time taken for the vessel to reach the charger.
    - **Current_towards_sea**: An array that takes positive and negative values, representing the movement of water from one port to another, measured in knots.
4. **Genetic Algorithm Parameters**: This section defines parameters that control the behaviour of the genetic algorithm used in the system.
    - **No_of_Generations**: The number of iterations or generations that the genetic algorithm should go through.
    - **Mutation_Rate**: The probability of mutating an individual in the population.
    - **Population_Size**: The initial number of potential solutions (individuals) when starting the genetic algorithm.

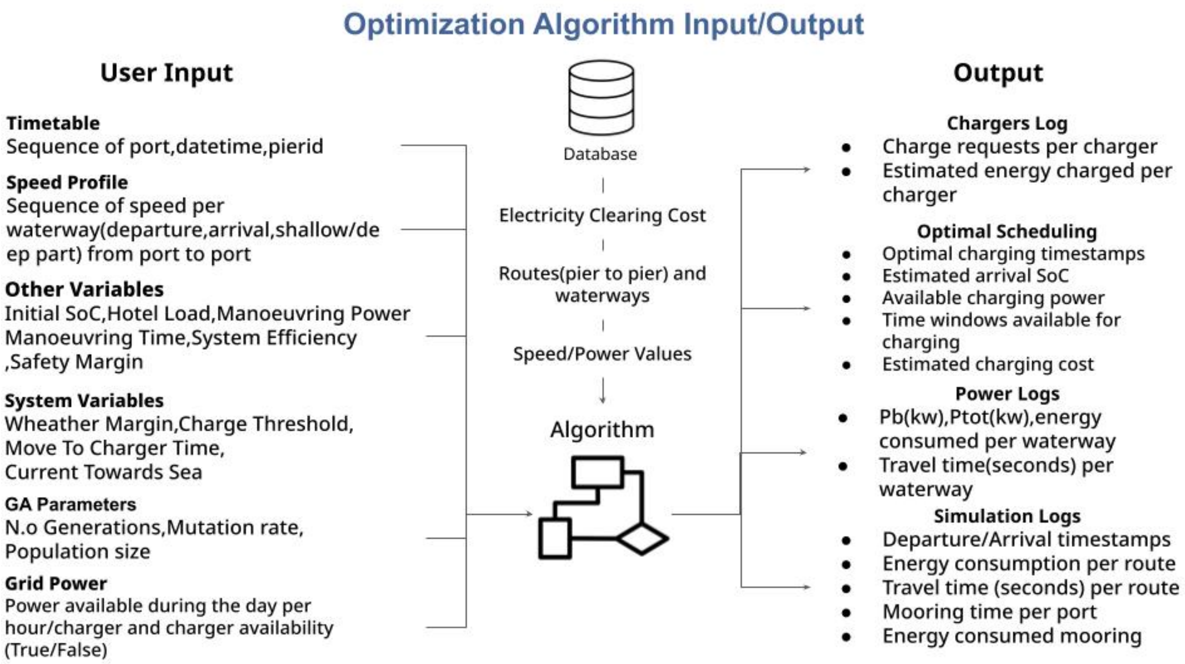

Figure 11 Upper Control layer Optimization Algorithm Overview.

6.2.2 Outputs

6.2.2.1 Charger Logs

The "Charger Logs" output offers an organized view of the charging activities that occurred at each port during the simulated period. It indicates the total count of charging requests ("Requests") initiated by the vessel at each port, and the cumulative amount of energy in kilowatt-hours ("Total_Energy_Charged") that has been supplied in response to these requests.

6.2.2.2 Optimal Scheduling

The Optimal Scheduling output offers an overview of the proposed charging plan for the vessel at various ports during its simulated journey. It presents information about the expected amount of energy that will be charged into the battery, and the cost associated with this charging process.

1. "Port": The location where the battery charging is anticipated to take place.
2. "Arrival" and "Departure": The simulated times when the vessel arrives and leaves the port.
3. "Start_Charging" and "End_Charging": The estimated times when the charging of the vessel's battery will commence and conclude.
4. "Charger Power": Power used, in average, to charge a vessel a certain deltaSOC amount, using previous charging events as baseline knowledge.
5. "Time_Charged_Seconds" and "Time_Available_Charging_Seconds": The anticipated duration in seconds for the charging process and the total time available for charging at the port, respectively.
6. "Energy_Charged": The forecasted amount of energy that will be charged into the ferry's battery.
7. "Start_SoC" and "End_SoC": The simulated state of charge of the battery at the beginning and end of the charging process, respectively.
8. "Start_SoC_Percentage" and "End_SoC_Percentage": The percentage of the battery's total capacity at the start and end of charging.
9. "Clearing_Cost": The expected cost of charging at each charging event

6.2.2.3 Power Logs

The **Power Logs** provide a record of the power and energy consumption details during specific segments of the ferry's journey. These segments include:

1. **Manoeuvring Departure**: This phase covers the period when the vessel is preparing to leave the port. It involves power and energy use as the ferry maneuvers out of its pier and navigates the port waters.
2. **Route Sections**: These are the different sections of the route that the ferry takes from the departure port to the arrival port. This can include parts of the journey where the vessel is sailing in shallow or deep waters, and the power and energy usage can vary based on water depth.

3. **Manoeuvring Arrival**: This phase represents the time when the vessel is nearing its destination and is preparing to dock at the arrival port.
4. **Mooring**: This is the time when the ferry is stationary at the pier, either at the departure port before setting off, or at the arrival port after completing its journey. Energy is still consumed during this time for systems that remain operational even when the vessel is not in motion.

In each of these stages, the power logs document the following details:

- Power in kilowatts (Pb_Kw and Ptot_Kw): Pb_Kw represents the base power, while Ptot_Kw is the total power utilized.
- Energy consumed in kilowatt-hours (Energy_Consumed_Kwh): This is the total energy used during the specific stage, which directly affects the remaining battery capacity.
- Time in seconds (Time_Seconds): This is the duration of each stage of the journey, which, when multiplied by the power used, gives the total energy consumed during that phase.

6.2.2.4 Simulation Logs

The simulation logs essentially contain detailed information about each route made by the vessel. For each route the logs provide the following information:

1. **Departure and Arrival Details**: The departure and arrival timestamps, the departure and arrival ports, as well as the specific departure and arrival piers.
2. **Speeds**: Different speeds for various parts of the journey, including the departure speed, the speed at the shallow part of the journey, the speed at the deep part of the journey, and the arrival speed.
3. **Travel Time and Energy**: The total time taken for the journey (in seconds) and the total energy consumed during the route (in kilowatt-hours).
4. **Battery Details**: The remaining battery capacity at the end of the route, whether or not the vessel was charged during its route, the battery capacity after charging (if applicable), and the charging time in seconds (if applicable).
5. **Mooring Time**: The time spent at mooring (in seconds), the energy consumed during mooring, and the battery capacity after the mooring period.

6.2.3 Implementation

The algorithmic process initiates by assembling data, which includes user inputs and pre-stored information like specific routes from port to port and pier to pier, waterway lengths, and arrays converting speed into power for shallow and deep parts of the routes.

Next, based on the input timetable, charging combinations are generated in a boolean format. These combinations serve as potential solutions to be evaluated by the algorithm.

Following the generation of these combinations, they are fed into the genetic algorithm alongside the assembled data. The algorithm operates with the aim of identifying the most optimal or near-optimal solution while respecting specific constraints within its fitness function. These constraints ensure that the State of Charge (SoC) does not surpass 0.8 and the battery energy never falls below the defined safety margin.

In the subsequent sections, we will unpack the intricacies of this process, focusing on the generation of charging combinations, the simulation function, and the mechanics of the genetic algorithm.

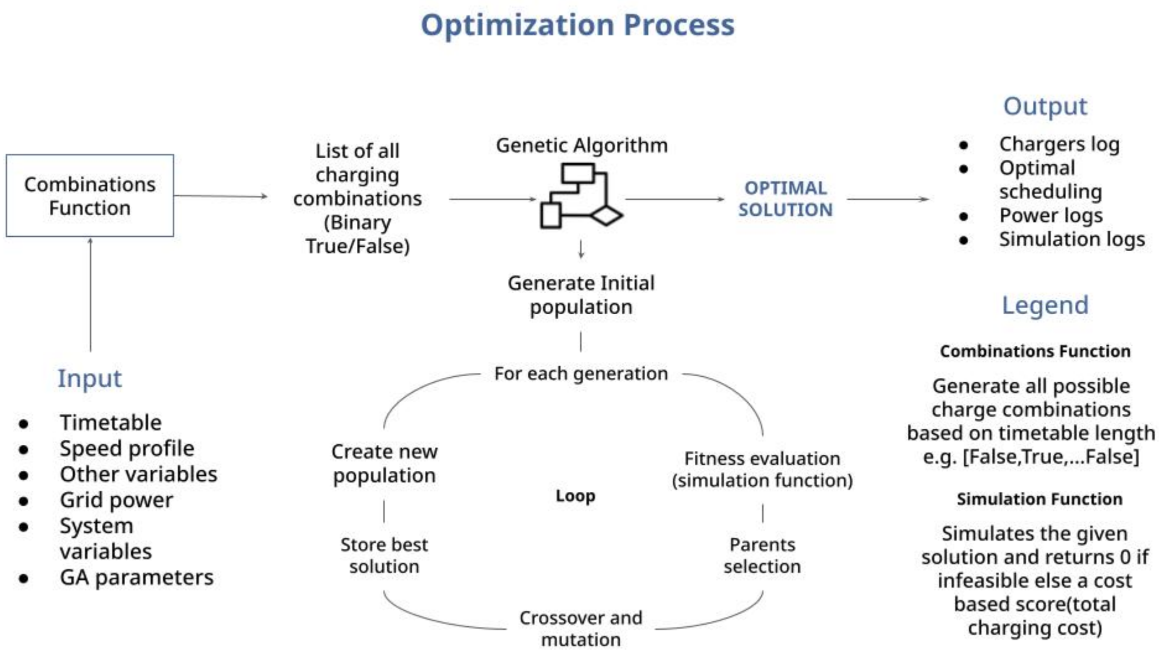

Figure 12 Upper Control layer Optimization Algorithm Process.

6.2.3.1 Combinations Process

First, the process starts with a timetable, which is a list of scheduled vessel departures. Each departure is a list that includes the departure port (either Norddeich or Norderney), the time of departure, and the specific pier from which the vessel departs. Alongside this timetable, there is a dictionary containing information about the availability and gross charging power of chargers at both Norddeich and Norderney. These inputs form the basis of the combination generation process.

The process then generates combinations of possible charging stops based on the availability of the charging stations at each port. It excludes the first and last index from the timetable as no charging is considered to occur at these times (representing the start and end of the simulation). Combinations are generated for each intermediate stop, allowing for the possibility of having both chargers available or only one of the chargers available at each port.

The combinations are then grouped by their lengths. This means that all combinations with one stop are grouped together, all combinations with two stops are grouped together, and so on.

Next, the combinations of indices are converted into a binary representation. A new list of lists is generated, where each list is as long as the timetable. If a certain index is included in the combination, the corresponding element in this list is set to True, otherwise, it is set to False. This step is critical because it reformats our data into a format that can be efficiently processed by the genetic algorithm.

Finally, the process flattens the list of lists into a single list, which is now ready to be fed into the genetic algorithm. Each list in this single list represents a possible charging plan, in binary form, based on the availability of the charging stations, allowing for combinations with both chargers available or only one charger available at each stop.

This expanded approach provides more flexibility in the charging plan generation, considering scenarios where both chargers can be utilized or when only one charger is available at specific port. It enhances the diversity of solutions and allows the genetic algorithm to explore a broader range of charging strategies.
The result is a binary encoded list of all possible charging plans. This format is compatible with genetic algorithms, which are often used to find optimal solutions in such scenarios.

6.2.3.2 Simulation Function

The simulation function models the operation the vessel between the two ports (Norddeich and Norderney). It takes as input the parameters and data mentioned in the paragraphs above. At each iteration of the timetable, the function uses the specified origin port, origin pier, destination port, and destination pier to retrieve the corresponding waterway paths from the database. These paths include sequences of waterways classified by whether they are part of a shallow or deep section and their respective lengths.

The function uses these inputs to simulate the operation of the ferry for one day, as follows:

1. For each route in the timetable, the function calculates the energy consumed per waterway based on the input speed per waterway given. It also calculates the arrival datetime at the next port, time available till the next departure, and the distance travelled.
2. If the value in the charge combination list corresponding to that destination port is True and a charger is available at the destination port, the function simulates charging the battery. It always attempts to charge as much as possible up to 80% of the State of Charge (SOC). During this process, it calculates the estimated amount of energy charged, the estimated charging time, and the estimated cost of charging.
3. Regardless of whether or not the battery was charged, the function also calculates the energy consumed while the ferry is moored at the port before its next departure.
4. When the vessel departs from the corresponding port, the function updates the remaining battery capacity, taking into account the energy consumed during the voyage, the energy charged (if any), and the energy consumed during mooring.
5. The function keeps track of the total travel time, total energy consumed, total charging time, total mooring time, and total energy output of each charger, among other

parameters. It also creates a detailed timestamped record of all these events and parameters.

At the end of the simulation, if the simulated combination is feasible, the function returns a cost-based score, which serves as the objective for a genetic algorithm to minimize, along with other outputs. By doing so, it aims to find the most cost-effective charging strategy for the vessel. If the simulated combination is infeasible, it returns 0(zero).

6.2.3.3 Genetic Algorithm

The genetic algorithm operates through the following sequence of steps:

1. **Initialization**: An initial population of all the combinations is chosen randomly. Each solution, also known as a chromosome, is a specific combination of charging stops. The population size is a parameter of the algorithm, user defined.
2. **Fitness Evaluation**: For each individual in the population, calculate the fitness score. This is done by using the simulation function which simulates the route using the specific charging decisions represented by the individual. The simulation function returns a score value based on the cost of the solution. The lower the cost, the better the fitness of the individual.
3. **Selection**: Based on the calculated fitness, the selection of parents for the next generation is performed using the Elitism selection method. In this method, the individuals are sorted by their fitness scores, and the ones with the best fitness (in this case, the lowest costs) are chosen to become parents for the next generation. This selection method ensures the best individuals are always preserved for the next generation.
4. **Crossover**: Random pairs of selected parents undergo a single-point crossover operation to produce offspring for the next generation. In this method, a random point in the parent chromosomes is selected and all data beyond that point in the first parent chromosome is combined with data before that point in the second parent chromosome, creating one new offspring. This process is reversed for the second offspring.
5. **Mutation**: To maintain diversity within the population and prevent premature convergence on a solution, a bit-flip mutation operation is applied to the new offspring. This operation randomly flips the state of a gene (a specific charging stop) within a chromosome, altering its value between true and false based on a certain mutation rate. The true/false binary representation of the solution corresponds to whether a charging action is performed at a particular stop or not.
6. **New Generation and Iteration** : The new offspring replace the current population, and the process from step 2(two) is repeated for a fixed number of iterations.
7. **Solution**: The best solution from the final population is then returned as the optimal scheduling. The best solution is the one with the lowest cost according to the fitness evaluation.

6.2.4 Demonstration

In this demonstration, the effectiveness of a Genetic Algorithm in optimizing charging plans is showcased, under the constraints that each charging event can only charge the battery up to 80% of its capacity or until the next departure timestamp of the vessel. The efficiency and speed of the Genetic Algorithm in finding high-quality solutions that prioritize cost optimization within a reasonable timeframe are emphasized. The speed of the Genetic Algorithm is further illustrated through a comparison of runtimes for different charging combinations compared to an exhaustive search algorithm.

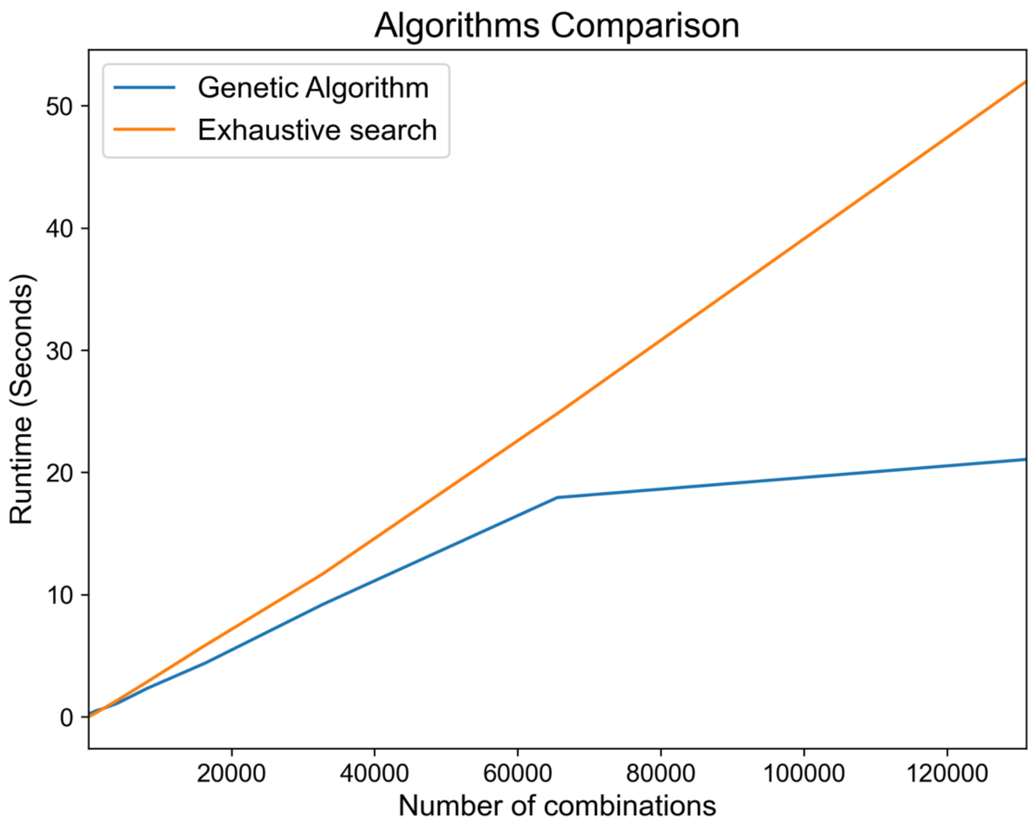

Figure 13 Runtimes per number of combinations.

Continuing from the previous section, this demonstration now turns to illustrate the effectiveness of the Genetic Algorithm in finding the best solution. The best solution, in this case, represents the most cost-effective charging plan. To ensure that the Genetic Algorithm is indeed reaching a global optimum under the constrains mentioned, or a solution close enough to it, a Brute Force algorithm is developed for comparison. This algorithm iterates through all possible charging combinations, providing a benchmark against which the Genetic Algorithm's solutions can be measured. Despite the computational intensity of the Brute Force algorithm, it offers a means to verify that the Genetic Algorithm is functioning as expected, optimizing the charging plans to their fullest extent. Figure 13 above presents the graph of the runtimes per number of combinations.

For the subsequent graphs and visualizations included in this demonstration, a real timetable from Frisia is used. The speeds inputted for from port to port within this timetable are randomly assigned for each test. Moreover, these tests consider both chargers available, each with power levels and electricity costs that vary at different hours of the day.

Timetable eCat	
from *Norddeich*	from *Norderney*
06:00	06:50
08:00	08:50
10:00	10:50
(13:00) if needed	(13:50) if needed
15:00	15:50
17:00	17:50
19:00	19:50

Figure 14 Timetable of Frisia's vessel for journey Norddeich-Norderney.

This real-world example helps to ground the demonstration and shows the practical applicability of these algorithms in optimizing real-world problems.

In the following sections of this demonstration, three distinct graphs are presented. Each of these graphs plots the test number on the x-axis against the optimization score (the total cost in these simulations) on the y-axis. The test number represents each individual run of the algorithm, with different, randomly assigned speeds from port to port influencing the optimization score in every test.

The three graphs correspond to three distinct points in the progression of the Genetic Algorithm: after three generations as seen in Figure 15, five generations in Figure 16, and ten generations in Figure 17.

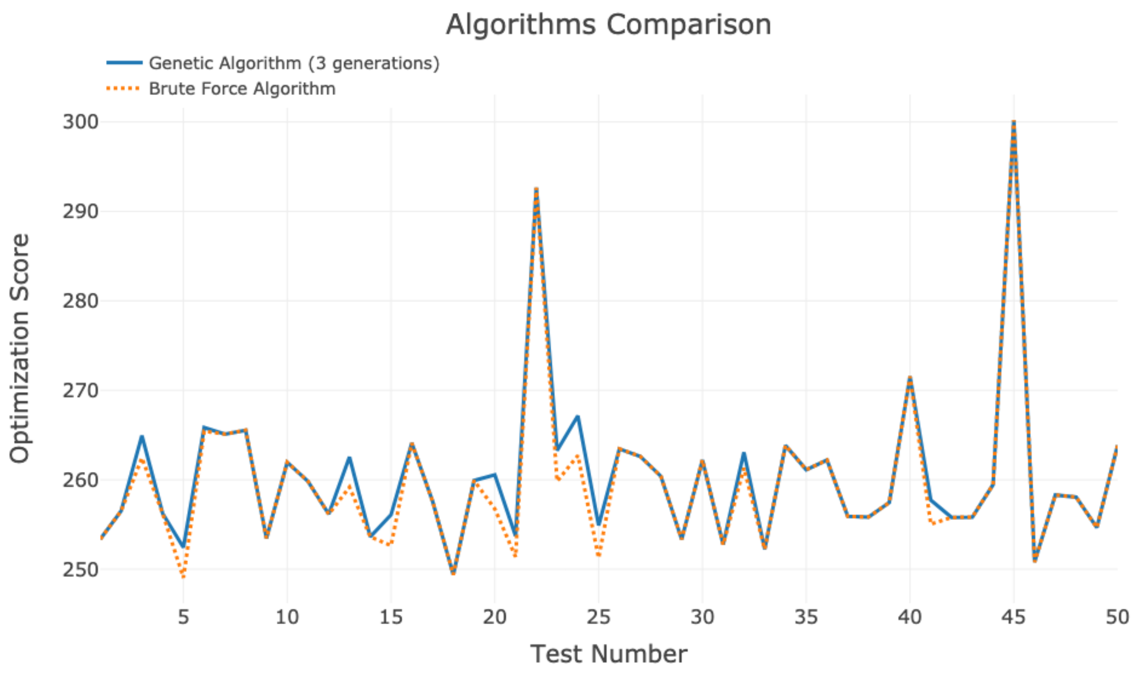

Figure 15 Performance of three generations.

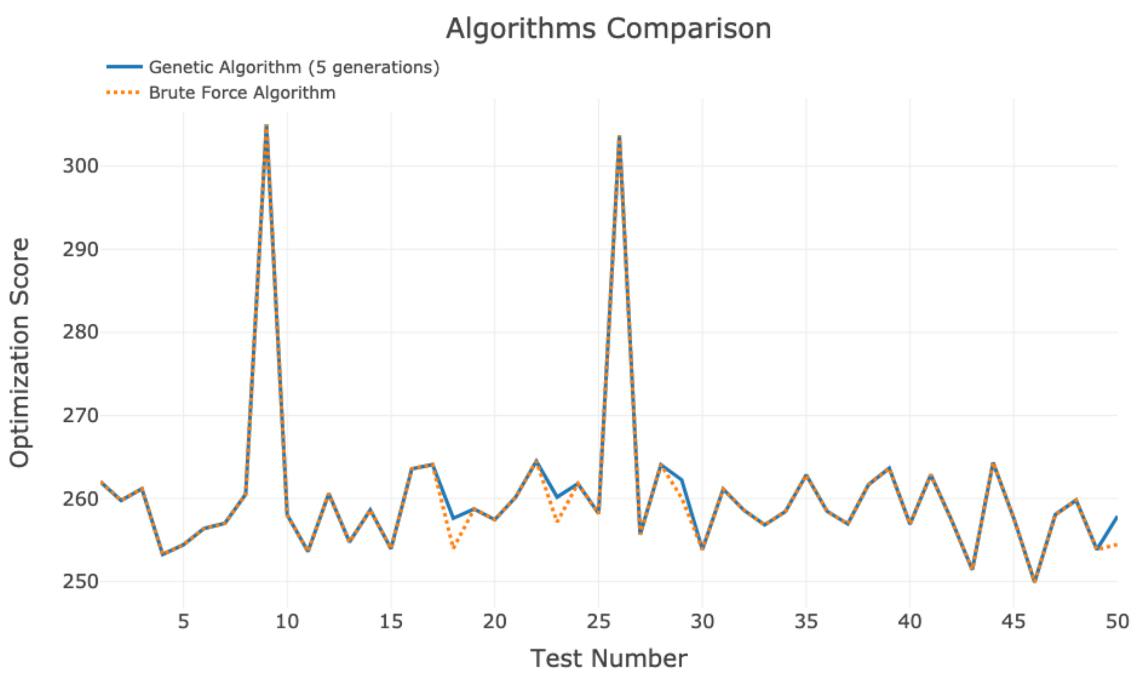

Figure 16 Performance of five generations.

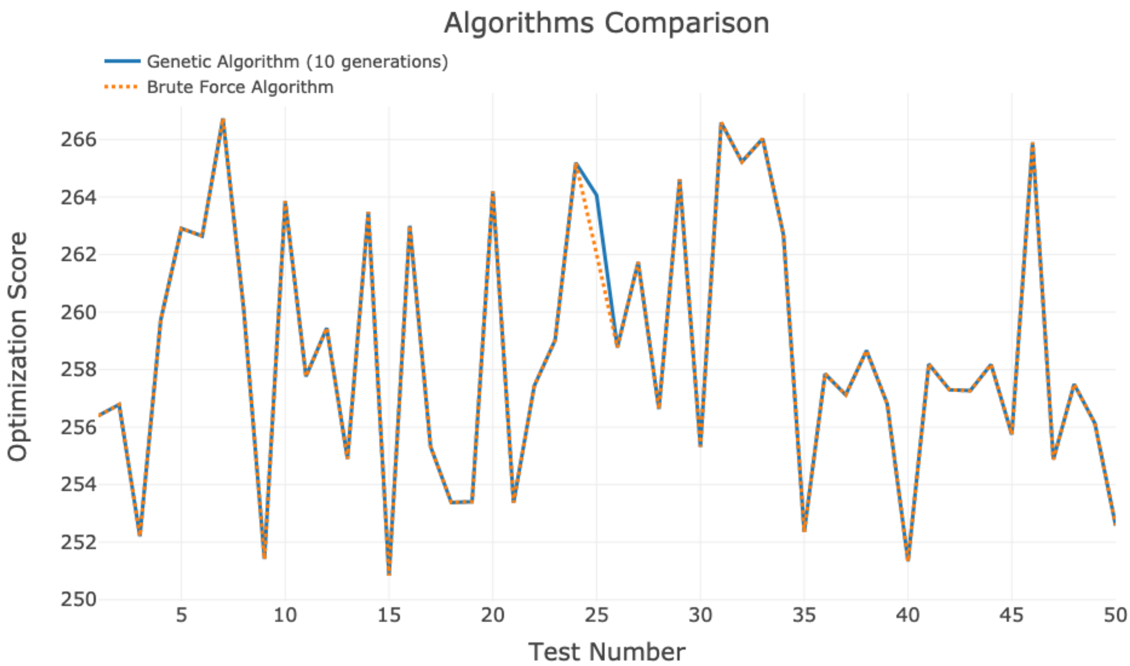

Figure 17 Performance of ten generations.

Each graph provides a visual representation of the Genetic Algorithm's improvement from generation to generation in locating the most cost-effective charging plan.

6.2.5 Validation

The validation step that is discussed in this section focuses on the algorithm that have been implemented. The validation consists in the analysis of the output of the algorithm depending on the input that describe each instance. Note that the implementation of the algorithm has not been analysed, only the results it provides.

Three instances have been provided and used to validate the algorithm. Each one of the instance is made of several files:

- **Chargers_params.xlsx** : it contains the maximal power that each charger can inject into battery for each time step (in the examples twenty-four time steps of one hour).
- **Pricing_params.xlsx** : it contains the clearing price of the energy/power for each time steps (in the examples twenty-four time steps of one hour).
- **System_params.xlsx** : it contains generic parameters that are required to compute the energy consumption of vessels during travels, margins, thresholds and specific durations.
- **Vessel_1_params.xlsx** : it contains all the parameters that are relatives to vessel #1. Several vessels would require several files. For each file, it details the timetable of each vessel. For each occurrence in the timetable that correspond to a departure, the speed for departure, shallow water, deep water and arrival are specified. Also, specific

parameters define the initial State of Charge (SoC), capacity, hotel load, manoeuvring time and power, efficiencies and margins.

The output files are:

- **Chargers_logs.xlsx**: it contains output global data of chargers, the number of charge requests and the energy charged for each charger.
- **Charging_logs.xlsx**: it details all charges for each vessel with starting and ending time, the power and the duration of charge, the energy that is associated, the initial SoC, the final SoC and the clearing cost of the charge.
- **Power_logs.xlsx**: this file details the power and the energy consumption all the steps for each vessel. Steps are manoeuvrings, moorings, departures, arrivals, travels on shallow water parts and travels on deep water parts.
- **Simulation_logs.xlsx**: it details the travels for all vessels and the charge at the arrival ports.

The validation protocol is made of two steps. The first step consists in the analysis of the feasibility of the solutions that are provided by the algorithm, while in a second step optimality considerations will be discussed. A third section presents some scalability aspects and perspectives.

6.2.5.1 Feasibility

Several feasibility constraints have been analysed.

- The state of energy of the vessel should be greater than the safety margin and lower than hundred percent at the end of any time step k
$$SafetyMargin \;<=\; SoC(k) \;<=\; 1$$
- The average power of charge on any time step k must be positive (the energy goes from the charger to the vessel, not the opposite) and lower than the gross charging power.
$$0 \;<=\; PowerCharge(k) \;<=\; GrossChargingPower(k)$$
- The duration of the charge cannot exceed the time the vessel stays at port reduced by the time it required to move to the charger
$$DurationCharge \;\leq\; Time\ at\ port - MoveToChargerTime$$
- The energy that is received by the vessel when charging must be equal to the product of the power of the charge, the duration of the charge and the efficiency in charge of the system
$$Energy\ charge(k) = \;Power\ charge(k) * \;DurationCharge(k) * \;efficiency$$

Equation 1 Energy Charge Equation.

- The state of energy (SoE) that is the energy stored in the vessel at the end of the time step k must be coherent with the energy that is received from the charge and the one that is consumed :
$$SoE(k+1) = \;SoE(k) + \;energyCharge(k) - \;energyConsumption(k)$$

Equation 2 State of Energy Equation.

Several constraints have not been checked in this validation process such that:

- The consistency of time/speed/power/energy for each travel because this information does not come from the algorithm but is computed at upper level with the IA/Database layer and this information creates a scenario input for the algorithm.

- A charge that covers several time steps, for example (12h-13h + 13h-14h) with variations on clearing prices and gross power limits have not been studied. This test can be relevant because charger power limits are described for each hour but in the three simulations that have been analysed, all charges have been performed on a single time step.

Regarding all those constraints, the feasibility of the solution that the algorithm provides is validated. Nevertheless, it is not conceivable to test all variations of inputs and to fully validate the feasibility criteria of the solutions created by the algorithm.

6.2.5.2 Optimality

In this validation step, two different objectives have been analysed:

1. The first objective that have been analysed is the minimization of the number of charge and then the minimization of the total charging time of the solution.
2. The second objective that have been analysed is the minimization of the global cost of all charges (using a forecast of the clearing price variation in the input files).

Regarding these two objectives, one by one, it appears that for now, the solutions that the algorithm provides are not optimal and can be improved to (1) minimizing the number of charge and the total time in charge when this objective is required, or (2) minimizing the total charging cost when this objective is required.

The optimality deficiency is large, and these results can have several explanations. For example, in the case of the first instance that have been analysed, the solution that have been generated by the algorithm is made of

- 6(Six) charges / a total charging time of 7294 (Seven thousand two hundred ninety-four) seconds
- a global cost around 334 (Three hundred thirty-four) £

This solution is dominated by two feasible solutions that are more optimal for the two objectives:

- If the first objective is required, a better solution made of 4(Four) charges and 6329(Six thousand three hundred twenty-nine seconds) of total charging time can be computed.
- If the second objective is required, a better solution that cost 287(Two hundred eighty-seven) £ can be computed.

6.2.5.3 Scalability & Perspectives

From the validation step and the study of input/output files of the algorithm, some critical points should be noted. They could represent some limits on the scalability of the methods.

Single/Multiple vessels :

The three tests that have been analysed are made of a single vessel problem. When multiple vessels are considered, the chargers could represent competitive resources and the charge of one vessel could make this critical resource unavailable to other vessels. This problem is increased by the fact that time steps seem to be dependant of vessel schedules.

Single/Several chargers :

The three tests that have been analysed are made of a single charger problem on each port. The gross charging power parameters that are defined in Chargers_params.xlsx are sufficient to treat this specific use case. Situation could be more complex when multiple chargers are installed in a single port. That is because the power available at any time could be limited by other constraints than the gross charging power parameter of the charger in case of simultaneous charge of multiple vessels at the same time: the grid connection point (power limits), the PV production and the topology of the electrical network of the chargers (clusters of chargers and power limits on lines).

Areas of improvement :

- The algorithm should control the volume of energy to be delivered during each charge to optimize charging costs and to give more flexibility to the lower control layer.

- The algorithm should consider time steps with constant conditions only. When the gross charging power differ, when the clearing cost changes or when any other parameters vary, the algorithm should create a new time step for the new condition. This improvement is necessary to deal with charging when vessels are moored in port for a long period of time. It could charge when clearing price is low, when gross charging power is high or any condition that will improve the quality of the solution.

- Increase the number of time steps when vessels are in port, for example by considering time steps of 60 seconds when vessels are moored. This enhancement could allow a vessel to charge for a short period, then to manoeuvre to another pier and allow other vessels to move to the charger. This improvement will also increase the complexity of the problem and then it will decrease the number of solutions scanned by the algorithm for a similar computation time.

- The genetic algorithm could be couple with a local search optimization step to improve the solutions locally and converge toward more optimal solutions. This specific improvement should considerably improve the quality of the results provided by the algorithm.

- Add a new parameter “maximal charging power” for the vessels. This parameter could be used to manage vessel fleet with various maximal power of charge that may be more limiting than the gross charging power.

7. LOWER CONTROL LAYER

7.1EMS Definition

The Energy Management Strategy (EMS) developed in HYPOBATT project aims for smart, fast vessel charging. Its main goal is optimising the fast-charging protocol to reduce both the vessel charging time and the ageing mechanisms of the battery. It is worth to mention that these two objectives are typically reported to be conflictive, as reducing the charging time means increasing the charging power, what typically accelerates battery degradation. As output, the EMS defines the power setpoint for the battery charging. Each time the EMS is executed, this setpoint is updated. In order to solve the mentioned conflict, the EMS is designed based on a real-time optimisation.

With the aim of designing the EMS, a problem has been proposed, which is solved by the proposed optimization. The problem is the charging activity itself: starting from the state at the current time step, the battery has to be charged until the maximum SOC, reducing both the charging time and the degradation. In order to reduce the complexity of the EMS, each time the problem is proposed (i.e., each time the EMS is executed), the returned solution is static. That is to say, each solution considers that the proposed charging power will be constant until the problem is solved (i.e., until the battery is completely charged).

Considering this framework, each time the EMS is executed, there are potentially infinite solutions (infinite charging powers) to solve the problem. However, these infinite possibilities are constrained by two limiting possibilities:

A. Maximum charging power allowed by the battery or the charger. This maximum power is linked to the minimum charging time.
B. Maximum time that the vessel can stay in the port. This maximum time is linked to the minimum charging power.

Figure 18 shows (in yellow) the constrained area for the optimization problem: starting from an initial SOC, and in order to reach a maximum SOC, the two limiting paths consist on charging at the maximum allowable power (leading to the minimum charging time) and charging at the minimum allowable power (which leads to the maximum charging time, constrained by the maximum time the vessel can stay in the port).

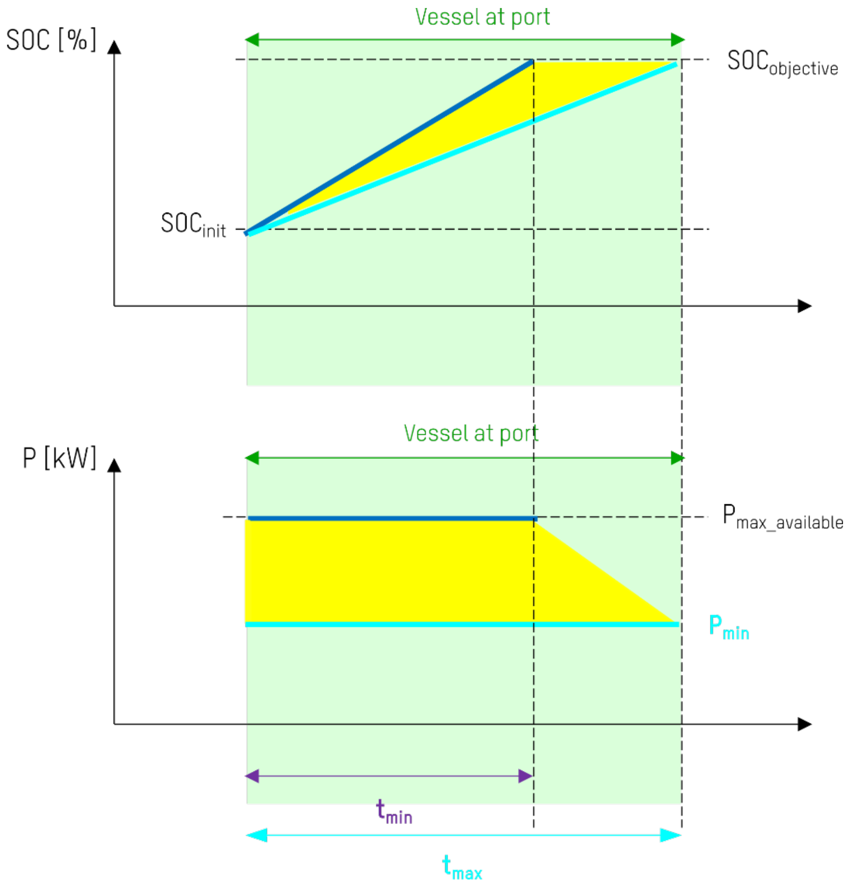

Figure 18 Power setpoint action framework (in yellow).

In order to obtain an optimal solution in the constrained area, a relation has to be set between the charging power and the degradation suffered by the battery.

In this context, it is worth to mention that the literature typically reports that increasing the charging power increases the degradation suffered by the battery. Therefore, the solution of the EMS has to obtain a balance between the two conflictive objectives: reducing charging time and reducing battery degradation.

In order to numerically solve the problem, a relation between the charging power and the degradation is proposed, which is presented in Equation 3. The equation is a conversion of the cycle life (CL) model presented in Deliverable 2.1 of the HYPOBATT project[2], assuming that the degradation is the inverse of the cycle life.

$$degr = (0.0778 \cdot (C - rate)^2 - 0.6478 \cdot (C - rate) + 1.5722)^{-1}$$

Equation 3 Equation for the relation between charging power and degradation.

Figure 19 below shows graphically the boundaries of the problem, which consists of the asset's minimum (blue) and maximum (purple) degradation limit. Once the degradation is quantified, the minimization function can be set, as presented in Equation 4. For this purpose, a relative

[2] "Report on digital twin of whole system", Deriverable 2.1 of HYPOBATT project.

weight between the minimum degradation and the criterion of load time minimisation must be defined. This relative weight is defined by the variable α.

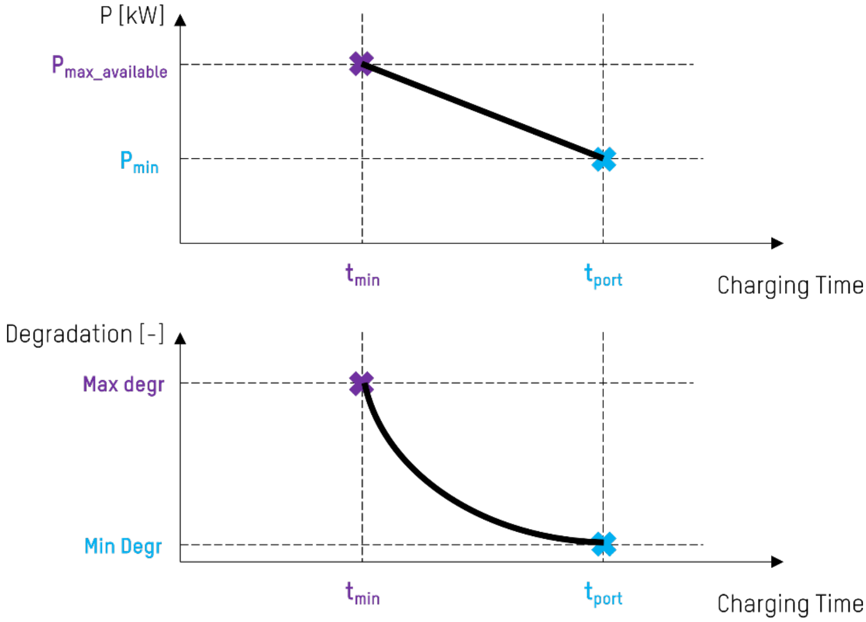

Figure 19 Relationship between available charging time range, charging power and degradation factor.

$$P_{ch} = minimization(\alpha \cdot degr + (1 - \alpha) \cdot time) \qquad [1]$$

Equation 4 Equation for the relation between minimum degradation and charging time.

By solving Equation 4, therefore, the EMS can give a solution to the proposed charging problem. As initially explained, the solution given is understood to be 'static'. However, as potential disturbances may happen during the charging activity, the given solution should be periodically updated. That is to say, each time the EMS is executed, the problem conditions may vary, leading to an updated charging problem that always aims at reducing charging time and battery degradation.

In the same way, variable α may be dynamically updated in order to give more importance to the reduction of the charging time or to the reduction of the battery degradation. This allows reducing the charging power at 'stress' situations, such as high temperatures or a low/high battery SOC.

7.1.1 Charging Algorithm

Multiple solutions for charging were approached. The literature presents multiple charging strategies, each with its benefits and costs. The usual passive methods include the known constant current-constant voltage (CC-CV), constant power-constant voltage (CP-CV), multi-CC stages controlled by the internal resistance and SOC, boost charging and all forms of pulse charging. While these present themselves as viable options, they all lack adaptability to the multitude of scenarios to which a charging event can be presented. These also do not consider physical manifestations of the cell, simply sticking to an established routine with fixed setpoints. The ability to mold the charging profile to different cell behaviors and time and power availability restrictions is fundamental in maximizing the system's efficiency [3].

To do so, a variable current profile was established as the go to charging method, with a final CV stage to top off the capacity. This type of charging gives leverage to manage the heat generation intense stage. To achieve such detailing, a fuzzy logic controller was introduced. This was a two-input-one-output fuzzy logic unit, consisting of the state of charge (SOC) and absolute temperature per iteration. These two inputs come directly from the Digital Twin of the battery system developed in a previous task. The output after defuzzification is the alpha coefficient established in section 6.2. Controlling this parameter will allow the EMS to centralize its focus on the minimum degradation or charging time.

7.1.2 Implementation

The EMS was implemented directly into the digital twin architecture as an extra subsystem. The CV stage was implemented as a control unit already within the charger architecture.

7.1.3 Validation

To validate, 2(Two) charging events were compared, using 2(Two) different charging profiles: the normal CC-CV, and the optimized by the EMS and the PI control unit. The initial conditions are established in Table 2.

Table 2 Simulation parameters.

Simulation parameters	Values (units)
Charging Time	1800 s
Initial SOC	50%
Initial Cell Temperature	298.15 K
Ambient Temperature	298.15 K
Initial C-rate	1.3 C
Max. charging power	2.5 MW

From the charging events, a comparison between SOC, current and temperature curves was done and are presented in Figure 20. It is clear that the overall temperature behaviour is more controlled when the EMS is operating, as it does not exceed 315(Three hundred fifteen) K. The current profile does not stand on the maximum for long as the EMS adjusts to the amount of time left and SOC left.

The SOH after the first charging cycle was also measured for both charging events. At the end, an improvement of 0.01% was achieved for a single iteration from 50 to 90% SOC. Naturally it does not seem like a big number but given the number of cycles the vessel will perform, the compound effect of this improvement will be felt in the long run. Further testing and optimization to the actual use case will be done in the future.

The end result is a smart current profile adjusted to the needs of the vessel. With this baseline, it can be pursued a more optimized version of the EMS, in order to further enhance its capabilities, with the KPIs of interest in watch.

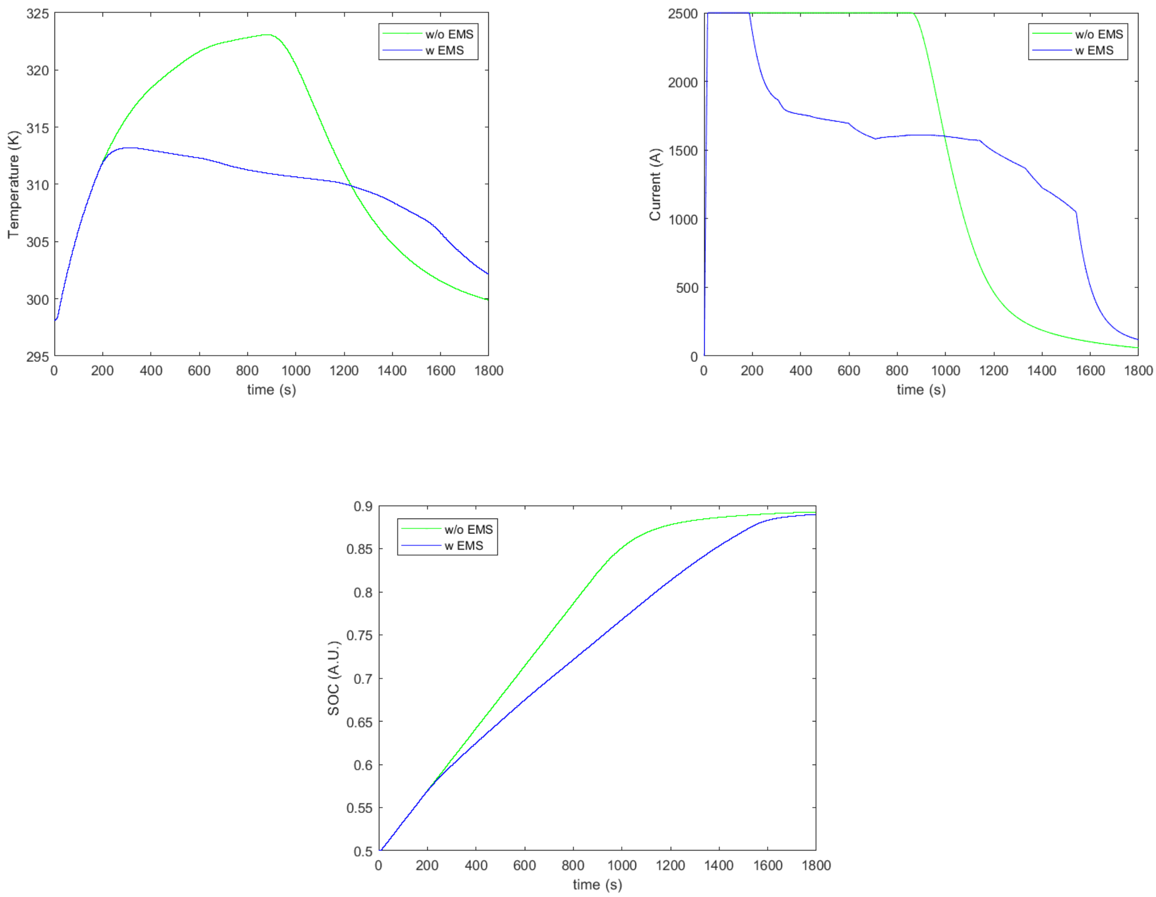

Figure 20 Results for the charging from 50 to 90% SOC with and without the EMS architecture.

8. CONCLUSIONS

In conclusion, the upper control layer and lower control layer work together to optimize vessel scheduling, energy management, and battery charging in the context of a port environment.

The upper control layer plays a pivotal role in the control and energy management architecture by providing an optimization algorithm for vessel scheduling and cost computation. It collects data from various sources, such as battery and charger logs, cost data, and user inputs, and interfaces with the database to accumulate relevant information. The layer applies a simulation function to generate potential solutions, considering factors like energy supply, charging needs, and cost considerations. The integration of the optimization algorithm into the upper control layer streamlines vessel scheduling and energy management, ensuring efficient charging while considering available resources and operational needs.

The lower control layer, specifically the Energy Management Strategy (EMS), focuses on smart, fast vessel charging. The EMS aims to optimize the charging protocol to minimize both the charging time and battery degradation. It addresses the conflict between reducing charging time and minimizing battery aging by utilizing a real-time optimization approach. The EMS defines the power setpoint for battery charging, considering constraints such as the maximum charging power allowed by the battery or charger and the maximum time the vessel can stay in the port. The EMS incorporates a relation between charging power and battery degradation, allowing for a balance between charging time reduction and degradation reduction. The charging algorithm within the lower control layer employs a variable current profile and a fuzzy logic controller to adapt the charging process based on the battery's state of charge and temperature, further optimizing the trade-off between charging time and battery degradation.

Overall, the combination of the upper control layer and lower control layer enables efficient vessel scheduling, energy management, and battery charging in port environments. However, there are areas for improvement, such as considering time steps with varying conditions, increasing the number of time steps during vessel mooring, and incorporating a local search optimization step to enhance solution quality. With further enhancements, the control layers can provide even more optimal and scalable solutions for vessel charging and energy management in port operations. Additionally, it is important to note that while the optimization algorithm has been primarily focused on the Frisia cases in its current implementation, there are plans to further develop and enhance its scalability for larger-scale applications. This includes considering a larger number of vessels, reducing the charging times, and expanding the routes beyond the current scope. The algorithm's adaptability and effectiveness will be tested and extended to accommodate the specific requirements of the port of Valencia in Spain. By expanding its use to different ports and adapting it to various scenarios, the system aims to provide optimized vessel scheduling, energy management, and charging solutions across diverse port environments.

9. NEXT STEPS

On a more comprehensive scale, the imminent task for the entire system involves laying down the foundation for effective communication and connection between the upper and lower control layers. This crucial initiative is earmarked for implementation during the impending project phase, thereby paving the way for a more integrated and cohesive control system.

Going forward, the project will also delve into the examination of an alternative algorithm for the system, utilizing linear programming methodologies. This represents an exciting new frontier for the project, promising to potentially enrich the system's capabilities and effectiveness.

In the months ahead, an evaluative process will be carried out. This will involve a comparative study of the currently employed genetic algorithm and the newly proposed linear programming algorithm. The aim of this exercise is to discern the practicability of merging the linear programming solution into the system, a move that could significantly enhance the system's adaptability and problem-solving prowess.

With regard to Work Package 2 (WP2), in the coming months, efforts will be directed towards finalizing vertical actions, such as updating the WP2 Data Requirements table. Additionally, connections will be made between WP2 and other work packages (WP3, WP4, WP5) to ensure that the system designed within this work package communicates effectively with the developed infrastructure and the charger.

Lastly, but certainly not least, the issue of the system's scalability has been slotted into Work Package 6 (WP6) within the framework of the HYPOBATT Project. This critical phase of the project will not only seek to amplify the system's capacity to handle larger, more complex operations, but it will also feature case study testing of the system. Such practical, real-world tests are invaluable in fine-tuning the system and ensuring that it is fully optimized to meet the demands it will face in its operational environment.

10. REFERENCES

[1] Zhang, Y., Huang, T., & Bompard, E. F. (2018). Big data analytics in smart grids: A review. *Energy Informatics*, *1*(1), 8.

[2] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091–8126.

[3] Gao, Y., Zhang, X., Cheng, Q., Guo, B., & Yang, J. (2 (Gao Y, 2019)019). Classification and review of the charging strategies for commercial lithium-ion batteries. *Ieee Access*, *7*, 43511-43524.