Low Carbon Installation and O&M Simulator Design Report

Document Title		Low Carbon Installation and O&M Simulator Design Report	
Document Reference		D5.2	
Date of Issue		10/01/2022	
Author		Guest, Felicity	
Revision History	**Date**	**Amended by**	**Reviewed By**
First Draft	05/01/2023	Olosunde, Justin	
Second Draft	06/01/2023	Chen, Barton	
First Issue	09/01/2023		Johanning, Lars
Rev 1			
Rev 2			
Rev 3			
Rev 4			

Contents

1 Executive Summary 2
2 Introduction 2
3 Development Approach 2
4 Alpha Version of E^c Simulator 3
  4.1 FLOW Simulator 4
  4.2 FLOW Modules 5
    4.2.1 Design Module 5
    4.2.2 Assembly & Installation Module 7
    4.2.3 Operation & Maintenance Module 10
  4.3 FLOW Calculator 12
5 Further Development 13
  5.1 Modules 13
  5.2 Graphical User Interface 13
6 Bibliography 16
7 Appendix - Introduction of Discrete Event Simulation 16

Figure 1 - A diagrammatic representation of the alpha version of the E^c Simulator 4
Figure 2 - Exemplar minimum spanning tree design module output for 34 x 15 MW turbines with a single substation. 6
Figure 3 - The structure of the assembly and installation module. 7
Figure 4 - Exemplar of simulation results of the installation timeline of a wind turbine. 9
Figure 5 - The working principle of the University of Exeter O&M Tool 11
Figure 6 - Simple wireframe for module selection. 14
Figure 7 - Simple wireframe for data capture or selection. 15
Figure 8 - Simple wireframe for strategy comparison. 15
Figure 9 - Example of Petri net diagram (a) before and (b) after the transition is fired 16
Figure 10 - A Petri-net model for offshore wind operation considering delay time, limited resources, and weather limits. 17

Table 1 - Required inputs of the Design Module. 6
Table 2 - Summary of typical tasks and their simulation parameters. 8
Table 3 - Required inputs of the Assembly & Installation Module. 10
Table 4 - Required inputs of the Operation & Maintenance (O&M) Module 12

1 Executive Summary

Commented [JL1]: Can this be made executive summary and moved to the top.

This report presents a high-level overview of the alpha version of the E^c Simulator. It provides insight into the design of the core components, namely the FLOW Simulator, FLOW Modules and FLOW Calculator, along with explanations as to why certain decisions were made. It details brief summaries of the functionality and overarching design of the three modules that have been integrated, specifically the Design Module, Assembly & Installation Module and Operations & Maintenance Module. In addition to this, it explains how development is being undertaken, including various software development best practices that have been employed, and sets out future development plans. The research priority is to initially focus on establishing links between the FLOW Calculator and the rest of the E^c Simulator, to ensure delivery of the alpha version as of the 31st of January 2023. The focus will then “switch” to the extending the three built simulation modules (Design, Assembly & Installation and Operations & Maintenance, and integrating two additional simulation modules (Power-to-X and Decommissioning) and producing a GUI with a software wizard.

2 Introduction

The E^c Simulator is currently being developed by the University of Exeter on behalf of the ERDF part-funded Cornwall FLOW Accelerator (CFA) Project to assist Floating Offshore Wind (FLOW) developers, supply chain operators and policymakers in decision-making processes. It is a software tool which will enable its users to assess the life cycle of floating offshore wind farms against certain key performance indicators (KPIs), which consider economic and environmental impact. The E^c Simulator employs optimisation, building on the E^c Calculator (a prior CFA Project interim output), which performed the same function but with static calculations alone. The E^c simulator will be deployed at project completion to generate a Celtic Sea FLOW Farm report on behalf of the CFA Project which models the life cycle for three pre-identified CFA development sites within the Celtic Sea.

This report provides an overview of the alpha version of the E^c Simulator. It will also provide insight into how it has been designed and is being developed, along with the University of Exeter plans to extend its functionality and improve its usability. It should be noted that the E^c Simulator is still in the early stages of development, and the alpha release of the simulator (31st January 2023) requires expert input to run simulations and is not a stand-alone user device. The beta version of the E^c simulator currently in development will have independent user access. The beta version has an anticipated second-quarter 2023 release date.

3 Development Approach

To date, the development team has included researchers with backgrounds in wind power and related fields and dedicated support from the in-house University of Exeter Research Software Engineering Group (RSE Group) (https://www.exeter.ac.uk/research/idsai/team/researchsoftwareengineers/).

Each researcher has been responsible for developing their own “*plug & play*” module which simulates a step in the life cycle of an offshore wind farm, such as design, assembly, and installation or

European Union
European Regional
Development Fund

operations & maintenance. The RSE Group has been responsible for the architecture and piecewise construction of the E^c Simulator. Employing an iterative approach to development and to enable further development in the future, the RSE team has developed an extensible programming architecture to support future beta version development and beyond.

Techniques deployed to date during development by the RSE Group have observed software development best practices, ensuring the readability, reliability, and robustness of the software. A common collaborative code repository had been utilized by all researchers; managed and operated by the RSE team on behalf of the CFA project.

Notably, a version control system has been used to track and manage changes, unit tests have been written to ensure the various components are working as expected, and documentation has been produced to improve clarity. A virtual environment has been deployed to minimize reproducibility issues and the software has been engineered to ensure that it is platform agnostic with respect to the operating system used to execute the software.

4 Alpha Version of E^c Simulator

The alpha version of the simulator is coded in Python, a particularly popular programming language, but it also makes use of code written in MATLAB, a language typically used for numerical computing. Ultimately, this version enables the user to employ the first variants of the Design Module, Assembly & Installation Module and Operation & Maintenance (O&M) Module. It also includes code to calculate the KPIs, namely the levelized cost of energy (LCOE), carbon intensity and energy returned on energy invested (ERoEI), along with code to perform the required calculations for manufacturing, power-to-X and decommissioning.

As indicated in Figure 1, this code forms the FLOW Calculator component of the E^c Simulator, aptly named as it is akin to the E^c Calculator, whilst the modules reside in the FLOW Modules unit. Each simulation is controlled by the FLOW Simulator which, in reality, encapsulates the FLOW Modules and FLOW Calculator entities, but for the sake of simplicity these key components of the E^c Simulator are shown separately in Figure 1. Two vitally important subcomponents also reside in the FLOW Simulator, one controls the simulation (Simulator) while the other ensures the simulation has all the information it needs (Data Manager).

The E^c Simulator has been segmented into components and subcomponents for a number of reasons but, most importantly, this structure enables parallel development while simplifying what is a complex system, as each part is focused on a single aspect of the simulator's functionality. Consequently, it should also be extendable. The following subsections explain each of the three key components of the E^c Simulator in more detail.

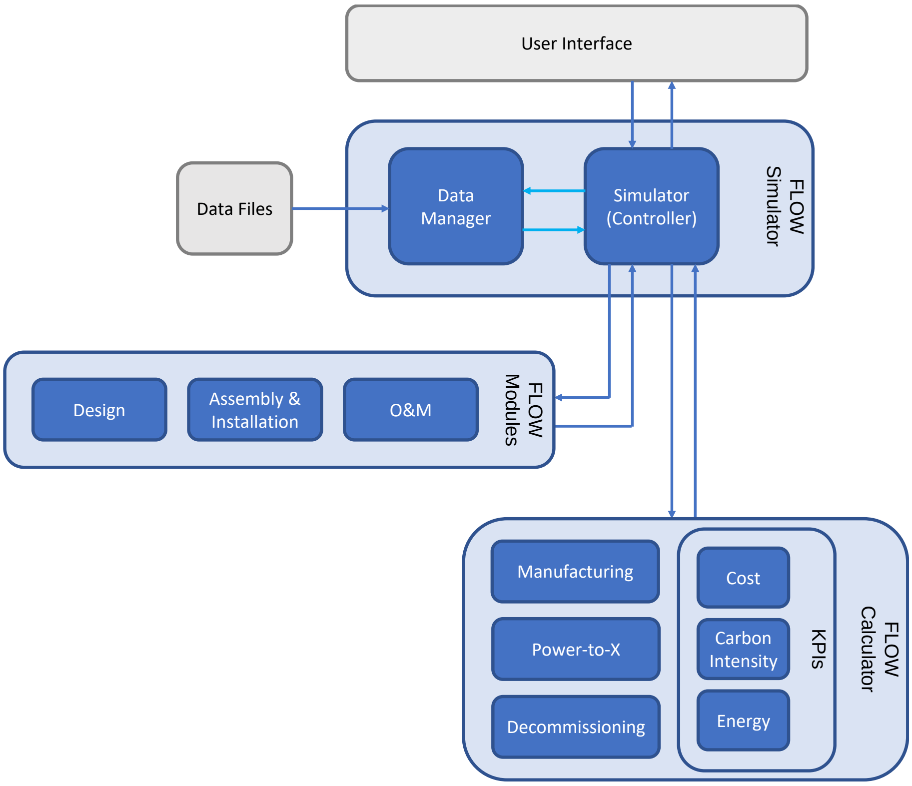


Figure 1 - A diagrammatic representation of the alpha version of the E^c Simulator.

4.1 FLOW Simulator

Functionality is achieved within the FLOW Simulator by the combination of two subsystems, one controls the simulation (Simulator) while the other ensures the simulation has all the information it needs (Data Manager). Fundamentally, the Simulator acquires data via a user interface which, at this stage of development, takes the form of a system configuration file, where the user can simply specify the required data for their chosen modules. This data is passed to the Data Manager to be stored and subsequently processed. Once data has been acquired, a simulation can then be initiated, during which the Simulator invokes each of the chosen modules in turn after triggering data processing procedures within the Data Manager that ensure the relevant data is provided to said modules in the appropriate format. Each module returns data which is also stored in the Data Manager; this data can be utilised by other modules and will enable the KPIs to be calculated.

A research decision was made to keep the data processing aspect separate to avoid overcomplicating the code which controls and runs the simulation. It is also accepted good practice to separate distinct functionality into different objects (or classes) when writing object-orientated code, which one could argue is the recommended approach when modelling real entities and systems. In time, the intent is to extend the data processing capabilities to enable the user to select and make use of data provided by the CFA Partners (The Offshore Renewable Energy (ORE) Catapult, Celtic Sea Power & The University of Plymouth, specifically by means of software wizard. It should be noted that we already have a database (co-convened from The ORE Catapult datasets and literature sources), which currently takes the form of numerous CSV (comma-separated values) and JSON (JavaScript Object Notation) files, and this functionality has been implemented for the Design Module, providing proof of concept. However, this functionality will be developed further alongside the graphical user interface (GUI), as these elements are closely coupled. Incidentally, our database comprises a myriad of data, including data pertaining to the wind farm, such as the number of turbines and their boundary points, along with data for the turbines themselves, such as their diameter and hub height.

4.2 FLOW Modules

For the initial alpha version of the E^c Simulator, three modules reside in FLOW Modules, specifically the Design Module, Assembly & Installation Module, and Operation & Maintenance (O&M) Module. As discussed, these modules, which have been developed by the researchers and integrated by the RSE coordinator simulate each of the three to-date defined steps in the life cycle of an offshore wind farm and are invoked by the Simulator. Each researcher has generated inputs and output data files of their modules, along with their design and current functionality, and an overview of the current research methodology deployed and planned future development methodology which is summarised within the remainder of this report.

4.2.1 Design Module

4.2.1.1 Design Module Methodology

The design module facilitates a user-defined array for turbine distribution at a user-specified longitude and latitude and is designed to allow spacing at intervals of 8D, 10D, and 12D where D represents the wind turbine’s total rotor diameter. The design module can also currently “*auto-design”* the layout of an offshore wind farm, deciding on the positioning of turbines and an offshore substation, as well as inter-array cable routing. To enable auto-design the module utilizes optimization to determine the layout, specifically employing an optimization framework for the simultaneous design of wind turbines (WTs) and cable layout for a collection system of offshore wind farms (OWFs) the approach adopted is sequential, with an initial annual energy production (AEP) maximization, followed then by the collection system design. (Cutululis & Antonio, 2022) to create a minimum spanning tree, commonly used to solve cable routing problems. Notably, this version of the module only considers a single substation, as it is assumed the farm has a maximum capacity of 500MW, and cable redundancy is yet to be considered.

European Union
European Regional
Development Fund

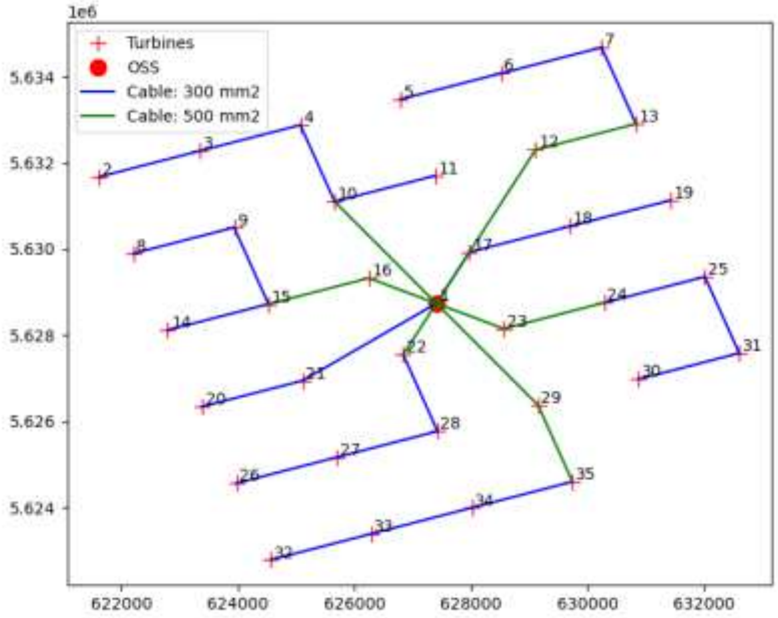


Figure 2 - Exemplar minimum spanning tree design module output for 34 x 15 MW turbines with a single substation.

4.2.1.2 Design Module Simulation inputs

The Design Module requires the inputs detailed in Table 1 and provides a configuration of the farm as output, including its substation, turbines, and inter-array cables. Additional information as to the properties of the selected cables is also provided, such as length and cost, along with the total system cost. Embedded carbon is derived from static data files retrieved by the data manager.

Table 1 - Required inputs of the Design Module.

Name	**Description**
Site Name	Name of the wind farm site.
Site Boundary Points	Lat/Long coordinates of the four corners of the site boundary.
Metocean Data File	File containing Metocean data for the site.
Number of Turbines	Number of turbines in the wind farm.
Turbine Properties:	*(for all wind turbines)*
Name	Name of the wind turbine.
Diameter	Diameter of the wind turbine.

European Union
European Regional
Development Fund

Hub Height	Hub height of the wind turbine.
Wind Speeds	Wind speeds for the power and power coefficient (Cp) curves.
Power Values	Power values for the wind turbine's power curve.
Power Coefficient Values	Power coefficient values for the wind turbine's Cp curve.
Carbon Intensity	KgC02$_{e}$ per unit Mass
Cross-sectional Area	Cross-sectional area of the cable.
Capacity	Maximum number of turbines the cable can sustain.
Unit Cost	Unit cost of the cable.
Carbon Intensity	KgC02$_{e}$ per unit Mass

4.2.2 Assembly & Installation Module

4.2.2.1 Assembly and Installation Module Methodology

Assembly & Installation Module is developed to simulate the construction of floating offshore wind farms based on discrete event simulation method. The model can simulate the logistics of upstream (i.e. manufacturing, transport to installation sites) and midstream (i.e. assembly and installation of offshore wind farms). Figure 3 shows the structure of the module. More information about discrete event simulation can be found in Appendix - Introduction of Discrete Event Simulation.

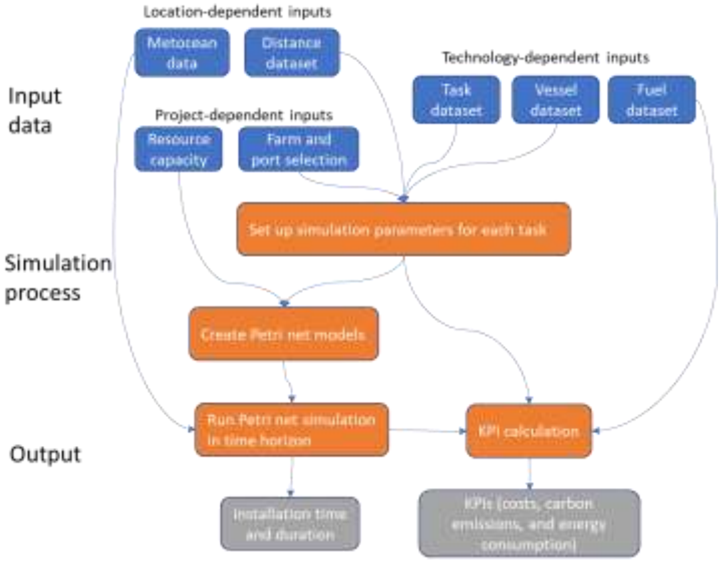


Figure 3 - The structure of the assembly and installation module.

European Union
European Regional
Development Fund

The model has three types of input data: technology-dependent, project-dependent, and location dependent. Project-dependent inputs are the parameters related to the development plan of a wind farm. Location-dependent inputs are the data related to the location of wind farm and ports. Technology-dependent inputs are the data that is related to the selected technology for the design and construction of the wind farm. In general, those data are independent from the locations (e.g. which port is used for installation and storage) and individual wind farm development. Table 2 summarises the simulation parameters of typical tasks for installation of offshore wind farms.

Type of task	Manufacturing	Transport (manufacture to installation)	Assembly	Offshore installation
Location	Manufacturing site	x	Assembly/ installation port	Installation port; offshore site
Task duration	Manufacturing time	Transit time; load in/out time	Assembly time	Transit time; installation time
Resource and capacity	Manufacturing capacity; storage space	Transport vessel; vessel transport capacity; storage space at port	Crane; labour; space at the port/shipyard	Vessel; crew
Inventory: input	(raw materials)	x	Components for assembly	Components for installation
Inventory: output	Components for assembly	x	Components for installation	x
Tasks with weather limits	x	Load in/out; transit	Assembly task (e.g. lifting)	Offshore operation

Table 2 - Summary of typical tasks and their simulation parameters.

The model has four main simulation processes. Firstly, the simulation parameters of each task (load from Task dataset) are set up based on the inputs from datasets, such as transit time is calculated from the distance and vessel speed from the datasets.

Secondly, three types of Petri net model are created, i.e. task, resources, and inventory. To link the Petri net model with time-series data, synthetic places are created to represent the status of weather window for tasks. The number of tokens in the place is synchronised with the value of a time-series data. The transitions in different models can be synchronised to simulate the corresponding action.

Thirdly, the Petri net simulation is run per time step (e.g. every hour). At each time step, the token number in synthetic places is updated first, and then all transition will be checked if it is enabled. The enabled transition will be fired based on the order of creation. The simulation will run at the next time step once all transitions are checked. The simulation will be terminated if all tasks are completed, or the simulation time is out of the range of time-series data.

Lastly, the KPIs (i.e. costs, carbon emissions, and energy consumption) are calculated based on the results from the simulation. The costs associated with the simulation are vessel costs (e.g. day rate, mobilisation/demobilisation costs, standby cost), fuel costs, labour costs, hiring costs for port/shipyard and cranage. The carbon emissions and energy consumptions come from the energy use (e.g. fuels for vessels and cranes) of each task.

Figure 4 - Exemplar of simulation results of the installation timeline of a wind turbine.

4.2.2.2 Assembly & Installation Module Simulation inputs

The Assembly & Installation Module simulates the execution of numerous assembly and installation tasks for the offshore wind farm. These tasks could include the transportation of the steel tower from its manufacturing to installation port, the assembly of the wind turbine generator (WTG) in the harbor, and the installation of for example a drag embedment anchor. At this stage of development, the module ascertains the time required for each task on a vessel-by-vessel basis, along with the cost, taking into consideration various vessel costs, fuel consumption, and carbon emissions. This information is provided as output, whilst the information outlined in Table 3 is required as input.

Name	Description
Site Name	Name of the wind farm site.
Manufacturing Port	Name of manufacturing port.
Installation Port	Name of installation port.
Export Cable Length	Length of export cable.
Port Pairing Properties:	*(for each port pairing)*
Departure Port	Name of departure port.
Destination Port	Name of destination port.
Distance	Distance between departure and destination ports.
Fuel Properties:	*(for each fuel type)*
Name	Name of fuel.
Type	Type of fuel.
Price	Price of fuel (per litre).
Carbon Intensity	KgC02$_e$ (per litre).
Installation Vessel Properties:	*(for each installation vessel type)*
Name	Name of vessel.
Transit Speed	Transit speed of vessel.
Installation Rate	Installation rate of vessel.
Mobilisation Cost	Mobilisation cost of vessel.
Demobilisation Cost	Demobilisation cost of vessel.

Day Rate	Day rate of vessel.
Fuel	Fuel of vessel.
Fuel Consumption Rate (Transit)	Fuel consumption rate of vessel during transit.
Fuel Consumption Rate (Operation)	Fuel consumption rate of vessel during operation.
Fuel Consumption Rate (Device)	Fuel consumption rate of vessel with device.
Installation Task Properties:	*(for each installation task)*
Name	Name of task.
Subsystem Type	Subsystem associated with task.
Operation Time	Operation time of task.
Operation Length	Operation length (or distance) of the task.
Departure Location	Name of departure location.
Destination Location	Name of the destination location.
Vessel Resource	Name and number of each available vessel type.

Table 3 - Required inputs of the Assembly & Installation Module.

4.2.3 Operation & Maintenance Module

4.2.3.1 Operation & Maintenance Methodology

The Operation & Maintenance (O&M) Module, originally developed by Giovanni Rinaldi and adapted for the E^c Simulator by Chenyu Zhao, simulates the execution of operation and maintenance tasks for specific components of the offshore wind farm. Rinaldi, Garcia-Teruel, Jeffrey, Thies, & Johanning (2021) provide a detailed explanation of the design and functionality of the module, referred to as the UNEXE O&M tool. Notably, it employs an approach based on a Markov Chain Monte Carlo method, which is often used for these types of problems.

The working principle of the University of Exeter tool is illustrated in Figure 3. The idea of this tool is that by exploiting the metocean data (hindcast or synthetic) of the location where the offshore farm is or will be located, together with all the specifications of the projects in terms of devices, vessels, sub-system failure predictions and maintenance strategies, it is possible to obtain a series of results that can be analyzed in an iterative procedure to characterize the dynamics of the farm and optimize the maintenance strategy.

Starting from the met-ocean data of the location selected for the farm deployment, the project specifications are added in terms of the installed devices (including details about their power curves and constituent components) and operational strategy (including corrective and preventive maintenance as well as maintenance assets). In accordance with the requirements of the Monte Carlo method, the same simulation is run enough times (according to the variance of the outputs or based on previous experiences with similar scenarios). Each of these runs simulates the operational lifetime of the Floating Offshore Wind Farm considering all the mechanisms and constraints defined by the user. Once the simulations are completed, a series of results describing the farm performance (i.e., interim KPIs) are obtained. These include energy production accounting for downtime, availability, revenue, and overall O&M costs and carbon emissions. More detailed information lists the number of failures per component or the hours of operation of each vessel. The results contain the full statistical distribution of each parameter, including mean value, standard deviations, and confidence bounds. These results permit the identification of underlying problems in the operation of the FLOW farm, and,

European Union
European Regional
Development Fund

if needed, the proposal of corrective measures. This model provides support in the decision-making process required for the successful management of a project. Statistical indicators, such as exceedance probabilities and progressive average values over the simulations, can also be analysed to evaluate the level of confidence in the results obtained. This allows the user to consider the elements of stochasticity related to both the met-ocean environment and the reliability of the components of the device.

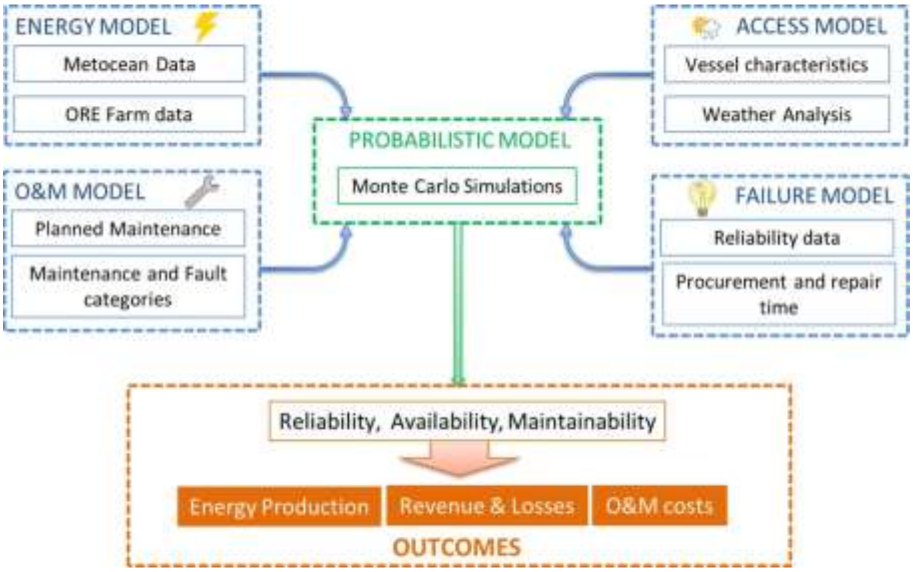


Figure 5 - The working principle of the University of Exeter O&M Tool.

4.2.3.2 Operation & Maintenance Module Simulation Inputs

This code was originally developed using MATLAB. As it would have been too time-consuming to translate the code into Python, Python code was written to initiate the MATLAB code with a minimal set of inputs, which are detailed in Table 4. This Python code also extracts outputs, namely the number of failures and downtime for components, as well as fuel burn, associated carbon emissions and usage for vessels. Steps were taken to ensure the code would run on any operating system and the plan is to extend the set of inputs.

Name	**Description**
Number of Simulations	Number of O&M simulations.
Number of Turbines	Number of turbines in the wind farm.
Number of Export Cables	Number of export cables in the wind farm.
Number of Affected Turbines (Export)	Number of turbines affected by export cable failure.
Number of Inter-array Cables	Number of inter-array cables in the wind farm.

European Union
European Regional
Development Fund

Number of Affected Turbines (Inter-array)	Number of turbines affected by inter-array cable failure.
Port-to-Farm Distance	Distance from port to wind farm site.
Metocean Time Step	Time step of weather data.
O&M Vessel Properties:	*(for each O&M vessel type)*
Name	Name of vessel.
Number	Number of vessel type.
Speed	Speed of vessel.
Day Rate	Day rate of vessel.
Standby Rate	Standby rate of vessel.
Mobilisation Cost	Mobilisation cost of vessel.
Fuel Burn (Transit)	Fuel burn of vessel during transit (for a one-way trip).
Fuel Burn (Operation)	Fuel burn of vessel during operation.
Carbon Intensity	KgC02$_{e}$ (per litre).
Wind Limit	Wind limit of vessel.
Current Limit	Current limit of vessel.
Component Properties:	*(for each O&M task / component)*
Name	Name of component.
Repair Time	Repair time of component.
Failure Rate	Failure rate of component (per year).
Vessel Selection	Vessel selection for component.
Leading Time	Leading time of component (inc. time to find a vessel).
Onshore Requirement	Whether the component needs onshore maintenance.
Downtime	Whether the failure can cause turbine downtime.

Table 4 - Required inputs of the Operation & Maintenance (O&M) Module.

4.3 FLOW Calculator

As previously explained, the FLOW Calculator includes code to calculate the KPIs, namely the LCOE, carbon intensity and ERoEI, along with code to perform the required calculations for manufacturing, power-to-X, and decommissioning. This code utilizes the outputs of the three integrated modules, facilitated by the Data Manager and Simulator. These links enable the assessment of the life cycle of the floating offshore wind farm against the KPIs. The resultant KPIs are presented to the user via the Simulator, eventually by means of the GUI but, currently as the interface is still within the developmental phase, outputs are produced in the terminal window, where the E^{c} Simulator is currently executed.

As indicated in 4.2.2, the Assembly & Installation Module is being extended to consider manufacturing; this would enable us to retire the manufacturing calculation in the FLOW Calculator. It is envisaged the calculations for power-to-X and decommissioning will also be retired in the future in favour of modules that simulate these aspects of the life cycle.

5 Further Development

Taking an iterative approach to development is vital when working on a piece of software of this nature and complexity, as ensuring it has strong foundations is key. As we have been working to produce an alpha version of the E^c Simulator (i.e., the first iteration), there is more work to do and this report has highlighted several areas where further development is planned. Initially, the developer teams will focus on establishing links between the FLOW Calculator and the rest of the E^c Simulator, to ensure delivery of the alpha version by the 31st of January 2023. The team will then focus on extending the three modules discussed and producing a graphical user interface (GUI) with a software wizard. The following subsections provide an overview of the work planned for the modules, which is being directed by the researchers, except for the Assembly & Installation Module as was discussed in 4.2.2. There is also a brief discussion on the planned work for the delivery of the GUI.

5.1 Modules

Work has already begun to extend the functionality of the Design Module, enabling it to provide site-specific mooring system designs according to dynamic or quasi-static analysis. Steps are also being taken to improve the cable length calculation so it can consider the lazy wave shape. Time permitting, it is hoped the farm layout designs the module produces will also be improved by considering cable loss and redundancy, along with multiple offshore substations.

As stated in 4.2.3, we plan to extend the set of inputs to the O&M Module, enabling the user to have more control over the O&M strategy being simulated. In conjunction, we will update and integrate the accompanying optimization code, developed by Giovanni Rinaldi in an earlier version of Python, into the E^c Simulator. This code looks to optimize some of the current inputs to the O&M Module, such as the number of each type of O&M vessel and the distance from the port to the wind farm site.

Two additional modules, developed by Barton Chen, are yet to be integrated into the E^c Simulator, namely the Energy System Simulation and Energy Generation Simulation modules. The integration of these modules has not been prioritized for the alpha version as they investigate external factors, such as energy prices, which aren't typically considered during wind farm development. These modules will, however, be integrated in the coming months.

5.2 Graphical User Interface

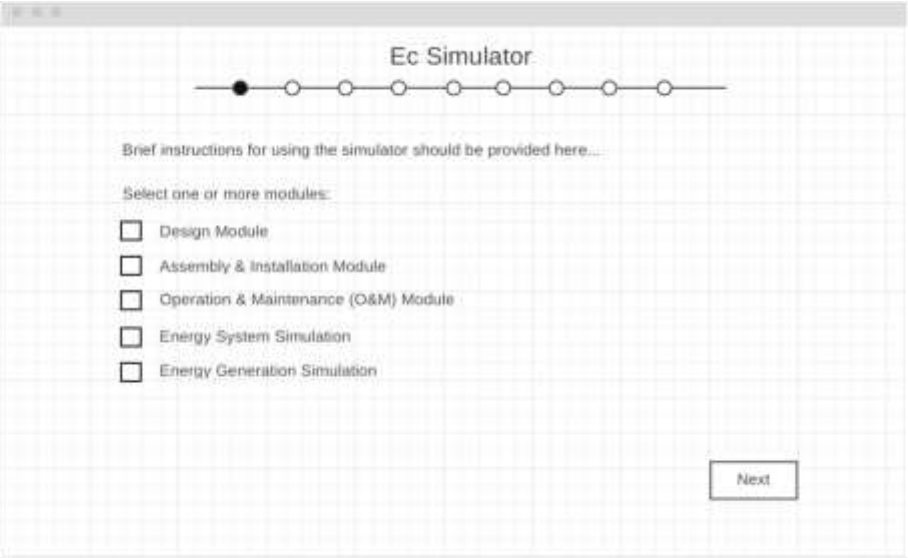
To improve the usability of the E^c Simulator, a graphical user interface (GUI) with a software wizard will be developed. As explained in 4.1, the software wizard will enable the user to select and make use of data provided by us rather than having to provide their own. To ease implementation, our database will be upgraded from a series of CSV and JSON files to a genuine SQL (Structured Query Language) database as the GUI is developed. Notably, this development will be undertaken in an iterative manner by a fellow Research Software Engineer, who has recently joined the project. It is envisaged they will create one or two prototypes before producing a fully functional, visually appealing, user-friendly GUI, which will be resizable.

European Union
European Regional
Development Fund

Three simple mock-ups or wireframes for the GUI are produced for demonstration (see Figure 6, Figure 7 and Figure 8). The first shows what could be deemed a simple start-up page that enables the user to select their chosen modules. Selection could be achieved by means of checkboxes and the page should feature brief instructions, along with a “Next” button. A progress indicator which would permit the user to see how far they are through set up is included.

The second wireframe provides a simple example of an input page. There would be a number of these pages, ordered to ensure ease of use, featuring input boxes and dropdown menus to allow users to supply or select data. This example shows how the GUI could enable the user to supply or select several cables for the Design Module, specifically by means of an “Add Cable” button. If the user were to click on this button, the set of input boxes would be replicated below those already present, enabling the user to supply or select another cable. This process could be repeated for the desired number of cables. It should be noted that the attributes would be automatically populated if a cable was selected from the “Name” dropdown. In addition to the progress indicator and “Next” button, which were present on the start-up page, I have included a “Back” button to permit the user to amend any previous inputs and have indicated a brief explanation of the inputs that should be provided. I believe it would also be beneficial for a description of each input, as well as its data type and unit (if applicable), to be shown when the user hovers over their name.

The third wireframe shows how the KPIs for multiple strategies, resulting from multiple runs of the E^c Simulator, could be displayed to the user at the prototype stage. As numerous inputs are required to run a single simulation, the inputs of any subsequent runs should be prepopulated based on the previous run to save the user time; the user would be able to make modifications to the inputs. Another strategy could be simulated and assessed by clicking the button at the bottom of the page, which would take the user back to the start of setup. In addition to the KPIs, the GUI should display any warnings or interim outputs from the modules.



Ec Simulator

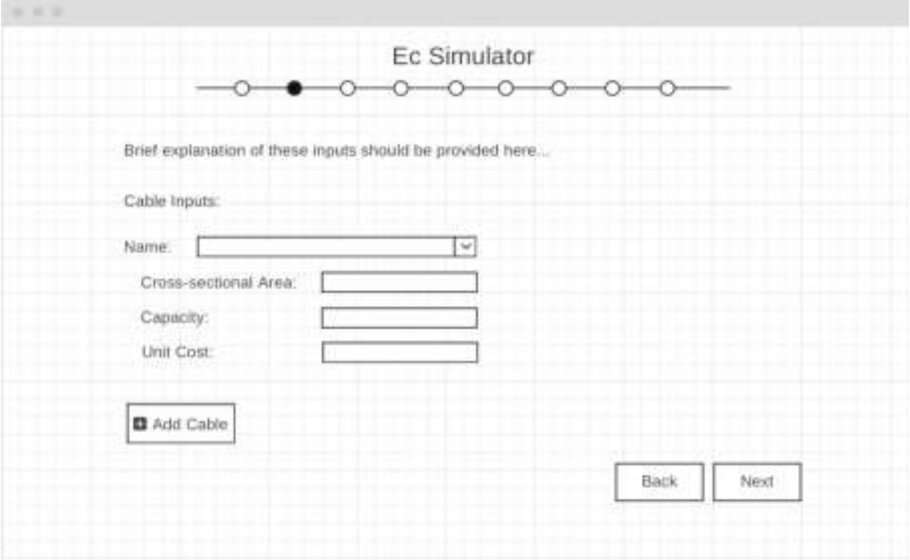
Brief instructions for using the simulator should be provided here...

Select one or more modules:

☐ Design Module
☐ Assembly & Installation Module
☐ Operation & Maintenance (O&M) Module
☐ Energy System Simulation
☐ Energy Generation Simulation

Figure 6 - Simple wireframe for module selection.

European Union
European Regional
Development Fund



Ec Simulator

○ ● ○ ○ ○ ○ ○ ○ ○

Brief explanation of these inputs should be provided here...

Cable Inputs:

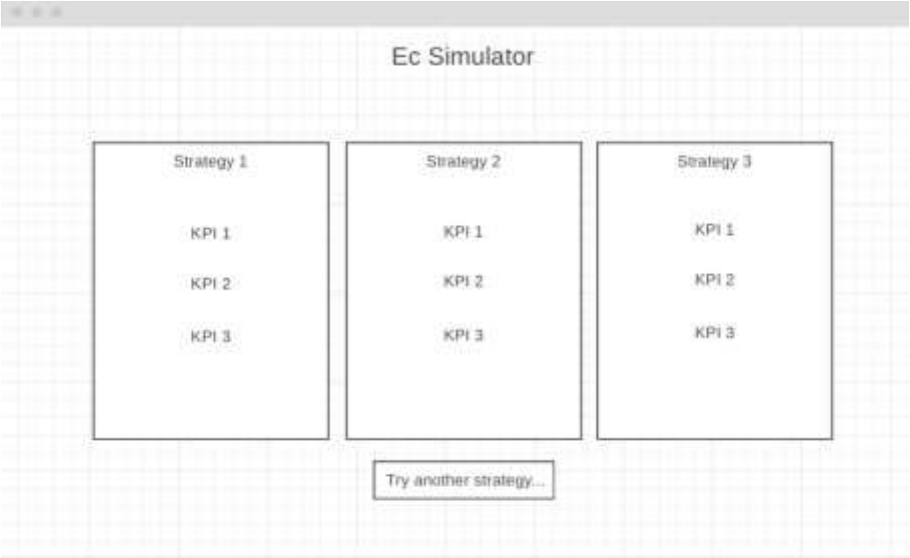
Name:

Cross-sectional Area:

Capacity:

Unit Cost:

Figure 7 - Simple wireframe for data capture or selection.



Ec Simulator

Strategy 1	Strategy 2	Strategy 3
KPI 1	KPI 1	KPI 1
KPI 2	KPI 2	KPI 2
KPI 3	KPI 3	KPI 3

Figure 8 - Simple wireframe for strategy comparison.

6 Bibliography

Cutululis, J., & Antonio, N. (2022). A framework for simultaneous design of wind turbines and cable layout in offshore wind. *Wind Energ. Sci., 7*, 925–942.

Rinaldi, G., Garcia-Teruel, A., Jeffrey, H., Thies, P., & Johanning, L. (2021). Incorporating stochastic operation and maintenance models into techo-economic analysis of floating offshore wind farms. *Applied Energy*.

7 Appendix - Introduction of Discrete Event Simulation

Discrete event simulation is a type of method to simulate batch processes which can be described in a series of non-continuous states. Discrete event simulation has been used in logistics and supply chain management. Petri net is one of the discrete event simulation models and has been used in the Assembly and Installation Module due to its capability to simulate various conditions and its extensibility to advanced features (e.g. timed Petri net, stochastic Petri net, and coloured Petri net).

Basic Petri net model

Petri net is one of the models for discrete event simulation which is based on a state-transition system. Petri net consists of three basic elements (i.e. place, transition, and arc) and can be presented in diagrams.

- Places represent the state of a process and contain discrete number of marks (also called tokens).
- Transitions represent the change of state from; a delay time can be set
- Arc links between places and transitions

A transition may fire if it all enabling conditions are satisfied (e.g. all input places have sufficient tokens). If a transition is fired, the tokens in input places are consumed and create tokens in output places. If multiple transitions are enabled at the same time, the transition will fire in any order unless an execution policy is defined.

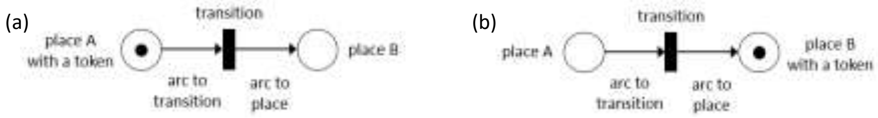


Figure 9 - Example of Petri net diagram (a) before and (b) after the transition is fired.

The above diagram shows an example of Petri net diagram consisting of two places (i.e. place A with a token and place B without token), a transition between place A and B, and two arcs (i.e. place A to transition; transition to place B).

In this example, place A is the input place of the transition, and place B is the output place of the transition. When the transition is fired, the token in place A is consumed and another token is produced.

Example of Petri net for offshore wind operation

In the Assembly and Installation Module, timed Petri nets are used to simulate the logistics and supply chain of offshore wind farms by simulating the following features:

- The sequence of a task (e.g. operation at the port -> transit to offshore site -> offshore operation -> transit to the port)
- Operation duration (e.g. 10 hours for transit to the offshore site, 20 days for floating platform manufacturing)
- Precedence task (e.g. anchors need to be installed before connecting WTG to the mooring system)
- Requiring resources for operation (e.g. vessels, crews, storage space at the port)
- Weather limits (e.g. wave height and/or wind speed need to be lower than a certain value)
- Input and output components of installation/assembly processes (e.g. WTG assembly task has a wind turbine and a floating platform as inputs, and a WTG as the output)

Figure 10 shows a Petri-net model for a basic offshore vessel operation, which has six stages (i.e. not started, transit to offshore site, wait for operation, under operation, return to the port, and task completed). A token is added in the first place to represent the status of the task. Transitions are added to simulate the status change from one to another. Additional input places and delay time are added to the transition to simulate that certain conditions need to be satisfied to trigger the transition. The enabling conditions of each transition are:

- Starting task: required vessels and crews are available; precedence tasks are completed
- Arriving offshore site: a delay time to represent the vessel transit time from port to the offshore site (t_{tosite})
- Starting operation: weather window for the operation
- Completing operation: a delay time to represent the operation time (t_{oper})
- Arriving port: a delay time to represent the transit time from offshore time to the port (t_{toport})

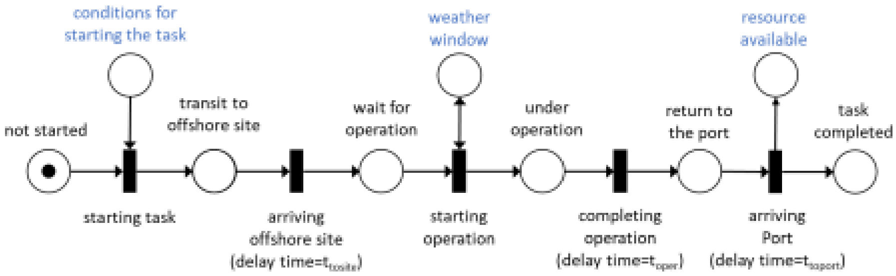


Figure 10 - A Petri-net model for offshore wind operation considering delay time, limited resources, and weather limits.