



Norwegian University of
Science and Technology

Modelization and Analysis of Offshore Wind Support Vessels in the Scenario of the new Generation of Offshore Wind Farms

Ignacio Scibona

Marine Technology

Submission date: June 2018

Supervisor: Bjørn Egil Asbjørnslett, IMT

Co-supervisor: José Jorge García Agis, IMT

Norwegian University of Science and Technology
Department of Marine Technology



Norwegian University of
Science and Technology

Modelization and Analysis of Offshore Wind Support Vessels in the Scenario of the new Generation of Offshore Wind Farms

Ignacio Scibona

Supervisor: Prof. Bjørn Egil Asbjørnslett

Co-Supervisor: José Jorge García Agis

June 11, 2018

Norwegian University of Science and Technology

Department of Marine Technology

Preface and Acknowledgements

The present thesis has been written as the final work for the degree of Master of Science in Marine Technology. The work was carried out between January and June 2018 under the supervision of Professor Bjørn Egil Asbjørnslett and José Jorge García Agis.

The starting point of the work was the project of the Specialization Project Course taken in the fall of 2017. The development of the tools and methods learned together with the maturation of the knowledge can be noticed when comparing both works.

The author would like to acknowledge the contributors of the work: Professor Bjørn Egil Asbjørnslett for his guidance, especially during the statement definition and model development and José Jorge García Agis, for his guidance and contribution with his expert judgement.

Trondheim, June 2018

(PAGE INTENTIONALLY LEFT BLANK)

Summary

The present thesis studies service operations vessels (SOV) working in operation and maintenance duties in the offshore wind industry.

An analysis of the current industry situation has been carried out and then the vessel's operation has been modeled in the scenario of the forthcoming offshore wind farms, whose characteristics such as size and location differ to a large extent from the traditional ones. Since the development of this type of wind farms is still in an early stage, there is a high degree of uncertainty about the operation characteristics, that, together with the inherent complexity of the marine systems constituted the two biggest challenges of the work.

The main aspects that have been modeled are the influence of the sea state during the operation, maintenance duties scheduling, technical personnel transfers, technical personnel needs, vessel's speed and accommodation capacity and costs.

After running the model simulating different proposed scenarios, the main conclusions are a set of highly influencing variables that lead to changes in the current vessels' design and other set of low influencing variables that the naval architects should be aware of when defining the design hierarchy.

(PAGE INTENTIONALLY LEFT BLANK)

Abbreviations

CTV	Crew Transfer Vessel
SOV	Service Operation Vessel
O&M	Operation and Maintenance
PM	Preventive Maintenance
CM	Corrective Maintenance
LCoE	Levelized Cost of Energy
CAPEX	Capital Expenditure

(PAGE INTENTIONALLY LEFT BLANK)

List of Figures

1.1	SOV During a Transfer Operation	3
1.2	SWATH-type CTV During a Transfer Operation	4
4.1	Markov Chain Transition Diagram [17]	14
4.2	Evolution of the LCoE according to the cumulated offshore wind capacity installed	18
5.1	Operation Flowchart	19
5.2	Simulink Model	20
5.3	SOV Entity Generator	21
5.4	Base Subsystem	22
5.5	Sailing Entity Servers	23
5.6	Wind Farms Subsystem	24
5.7	Maintenance Tasks Generation	25
5.8	Sea State and Wave Forecast Entity Generators	26
5.9	Costs Function	28
5.10	Global Variables and MATLAB Functions Used for Times Calculations	31

5.11	Global Variables for Waiting Times Calculations	32
5.12	Global Variable for Transferred Technical Personnel calculations . . .	33
5.13	Global Variables for Used for SOV Placing	34
5.14	Constants Used to Define Vessels' Characteristics	35
5.15	Constants Used to Define Wind Farms' Characteristics	36
6.1	Maintenance Tasks and Transferred Technical Personnel in Scenario 1	41
6.2	Maintenance Tasks and Transferred Technical Personnel in Scenario 2	42
6.3	Wind Turbines Out of Service in Scenario 2	42
6.4	Maintenance Tasks and Transferred Technical Personnel in Scenario 3	43
6.5	Maintenance Tasks and Transferred Technical Personnel in Scenario 4	44
6.6	Maintenance Tasks and Transferred Technical Personnel in Scenario 5	44
6.7	Maintenance Tasks and Transferred Technical Personnel in Scenario 9	45
6.8	Maintenance Tasks and Transferred Technical Personnel in Scenario 15	46
6.9	Significant Wave Height	46
6.10	Sailing Times	47

List of Tables

1.1	Maintenance actions at an offshore wind farm and their occurrence [1]	2
1.2	Failure Repair Characteristics	3
1.3	Main Transfer Methods Characteristics [2]	4
4.1	Transition Matrix for Figure 4.1	14
4.2	Transition Matrix	15
6.1	Modeled Scenarios Characteristics	40
6.2	Modeled Scenarios Output	41

(PAGE INTENTIONALLY LEFT BLANK)

Table of contents

Preface	i
Summary	iii
Abbreviations	v
1 Introduction	1
1.1 Offshore Wind Farms Operation and Maintenance	1
1.2 Service Operation Vessels	4
2 Problem Description, Boundaries and Scope	7
3 Literature Review	9
3.1 Wind Farms Operation and Maintenance	9
3.2 Ship Design	10
4 Methods and Tools	13
4.1 SimEvents and Simulink	13
4.2 Markov Chain	13

4.3	Sea-State Influence on Operation and Maintenance	15
4.4	Maintenance Tasks Modelling	16
4.4.1	Preventive Maintenance (PM)	16
4.4.2	Corrective Maintenance (CM)	16
4.5	Wind Farms Modelling	17
4.6	Costs	17
5	Model Description	19
5.1	SOV Entity Generator	21
5.2	Base Subsystem	22
5.3	Sailing Entity Servers	22
5.4	Wind Farms Subsystem	23
5.5	Maintenance Tasks	25
5.6	Sea State and Wave Forecast Entity Generators	26
5.6.1	Sea State Entity Generator	27
5.6.2	Wave Forecast	27
5.7	Costs	27
5.8	Global Variables, Constants and other MATLAB Functions	29
5.8.1	Global Variables Used For Maintenance Tasks Generation	30
5.8.2	Global Variables and MATLAB Functions Used for Times Calculations	31
5.8.3	Global Variables for Waiting Times Calculations	32
5.8.4	Global Variable for Transferred Technical Personnel Calculations	33

5.8.5	Global Variables Used for SOV Placing	34
5.8.6	Constants Used to Define Vessels' Characteristics	34
5.8.7	Constants Used to Define Wind Farms' Characteristics	36
6	Results	39
7	Discussion and Conclusions	49
8	Further Work	51
A	Implementation of Markov Chain for Simulating Sea States	55
B	MATLAB Scripts	61
B.1	SOV Entity Generator's Script	61
B.2	Base Subsystem's Scripts	61
B.2.1	Base Entity Server's Script	61
B.2.2	Base Exit Entity Server's Script	62
B.3	Sailing Entity Servers' Scripts	62
B.4	Wind Farms Subsystem's Scripts	62
B.4.1	Waiting Place Entity Server's Script	62
B.4.2	Script for Weather and Sea State Gate MATLAB Function	64
B.4.3	Farm Selector Entity Server's Script	65
B.4.4	Farms Entity Servers' Scripts	66
B.5	Maintenance Tasks Generator's Script	74
B.6	Sea State Entity Generator's Script	76

B.7	Wave Forcast Entity Generator's Script	77
B.8	Costs Function's Script	78
B.9	ShiftTime MATLAB function's script	79
B.10	ShiftDays MATLAB function's script	80
B.11	MinTasks MATLAB functions's Script	80
B.12	MaxTechnicians MATLAB function's Script	80

1 | Introduction

The current and future energy demands for the growing world population, new and more strict environmental regulations, and the availability of new technologies, has triggered an exponential growth of the offshore wind industry worldwide. As the offshore wind farms become larger and are being installed further offshore, with more and larger turbines, new challenges arise for the logistics and service offshore vessels play a key role on it. Under these conditions, it is important for the naval architect to improve the designs, adapting them to the operation demands.

1.1 Offshore Wind Farms Operation and Maintenance

There are many operations that have to be carried out in the offshore wind industry, The main ones are: site investigation surveys; installation and commissioning of foundations/substructures, wind turbines, the offshore substation and subsea cables; operation and maintenance (O&M). In addition, decommissioning or re-powering is performed offshore at the end of the life of the wind farm. However no data is available for this phase due to the newness of the technology. The first offshore wind farms were installed in the 1990s and have still not reached their end of life.

Even though the offshore wind farms are mostly remotely operated from shore-based centers, the regular maintenance works has to be done, which makes necessary to transport maintenance personnel, together with their tools and spares needed, to offshore locations. Depending on the extent of the maintenance required, heavy logistics can be needed.

The average yearly maintenance requirement for a typical wind turbine is between one and two visits for scheduled preventive maintenance tasks, and from two to four corrective maintenance visits due to failures. Therefore, each turbine of a wind farm requires to be serviced between four and six times a year. In addition to that, every five years major overhauls are carried out in which the major components and the underwater parts of the turbine are inspected. This is exemplified in Table 1.1, where it can also be appreciated that most of the maintenance actions involves only small parts and less than 10% require heavy logistics like a jack-up or crane vessel

Table 1.1: Maintenance actions at an offshore wind farm and their occurrence [1]

Maintenance Action	Required Equipment	Occurrence
Replacement of small part (man carried) or inspection	Vessel or Helicopter	69%
Replacement of small part (<1 ton)	Vessel, Permanent internal crane	23%
Replacement of large part	Vessel, Built up internal crane or Crane vessel	7%
Replacement of heavy components	Jack-up vessel with crane	1%

The scheduled maintenance visits to the wind turbines are usually to inspect and to do the regular service, for which no big or heavy equipment is necessary. These tasks and services are usually performed during the summer seasons due to the better access and transfer conditions that the good weather allows and the revenue losses due to the turbine down times that are lower during the less-windy times. Unscheduled maintenance visits take place when failures occur. Before the maintenance crew is allowed to access to the wind turbine, it is stopped remotely from the shore-based operation centre. Table 1.2 shows failure repair characteristics.

The transfer of the maintenance personnel, tools and spares required for the described maintenance is usually performed by service operation vessels (SOV). They act not only as transport but also as accommodation vessel for the days needed to perform all the scheduled tasks. The SOVs are usually equipped with crew transfer

Table 1.2: Failure Repair Characteristics

Repair Type	Repair Time	Technicians Required	Vesse Required	Failure Rate	Repair Cost [EUR]
Manual Reset	3 h	2	CTV/SOV	7.5	0
Minor Repair	7.5 h	2	CTV/SOV	3.0	1,129
Medium Repair	22 h	3	CTV/SOV	0.275	21,000
Major Repair	26 h	4	SOV	0.04	83,000
Major Replacement	53 h	5	Jack Vessel	0.08	377,500
Annual Service	60 h	3	CTV/SOV	1	21.000

vessels (CTV), which are short-range fast transport boats with no accommodation. They are deployed from the SOV to enlarge the transfer capacity. When the sea state makes the access from the water impossible and the wind allows it the transfer can also be carried out by helicopters, but due to the high cost, they operate in wind farms located close to the coastline. Figures 1.1 and 1.2 show a SOV and a CTV during a transfer operation respectively. The main transfer methods and its characteristics are summarized in Table 1.3



Figure 1.1: SOV During a Transfer Operation



Figure 1.2: SWATH-type CTV During a Transfer Operation

Table 1.3: Main Transfer Methods Characteristics [2]

Type	Allowable Weather Condition	Advantages	Disadvantages
SOV	Max. wave height: 3 m	Less sensitive to wave height	More expensive than CTVs
CTV	Max. wave height: 1,5 - 2 m	Simple, fast, economical	Sensitive to wave height and ice. Low cargo capacity
Helicopter	Max. wind speed 15-20 m/s	Not sensitive to wave height	Expensive

1.2 Service Operation Vessels

The fleet of vessels is heterogeneous and each vessel might be fitted with one or two CTVs and/or helicopter [3]

A typical operation of a SOV starts onshore, with the loading of spare parts, tools and consumables, and embarking technical personnel for maintenance duties. Then, the vessel sails to the offshore wind farm/s where acts as accommodation

vessel and logistic center for the days the whole operation lasts (usually between 15 and 30 days) [3]. For safety reasons, the transfer operations take place only during daytime, and when the sea state is favourable, that is up to 3m significant wave height for SOVs and not more than 1,5m for CTVs.

One of the critical steps of the maintenance operation of the wind farms is the transfer of the personnel to and from the turbine because the weather and sea conditions can make it difficult, unsafe or even impossible [2]. Usually the approach of the vessel is from the lee side of the turbine, in order to be sheltered from the wind and waves, and to make the transfer as quiet as possible. Nowadays special designs that make the vessels less sensitive to the waves are showing up, which include heave-compensating systems, catamarans and SWATHS (small waterplane area twin hull) [2].

(PAGE INTENTIONALLY LEFT BLANK)

2 | Problem Description, Boundaries and Scope

The main goal of the present work is to propose changes to the specifications of the current wind support vessels to adapt those to the future needs in larger and further offshore farms. Based on the characteristics of the current fleet, its operational profile and established practices and patterns, the plan is to extract insights to define the requirements of future vessels.

The vessel operation is going to be modeled based on real data taken from the available literature. Different operation scenarios and vessel characteristics will be proposed and tested with the model.

In order to be able to get reliable results from the model, the intention is to make it as real and accurate as possible but at the same time simple without adding unnecessary complexities. That will make the analysis of the results easier and will reduce the likelihood of errors in the output.

The maintenance tasks planning, the routing of the vessel inside the wind farm and the cost analysis will not be covered. They will be included in the model by taking average values from similar cases. The vessel's operation in the shore-base is not to be modeled either, that is loading spare parts, tools and consumables, and embarking the technical personnel is out of the scope.

The maintenance tasks are modeled under the assumption that none will last more than one working day. If longer tasks must be performed, they will be split in several parts that fit into one.

Transfers by helicopter will not modelled because they operate mainly close to

the coast.

For modeling the wind farms, a hypothetical one has been created taking as a base the planned Sørliche Nordsjø (Category A area) Offshore Wind Farm. [4]

3 | Literature Review

3.1 Wind Farms Operation and Maintenance

Tveiten [2] and Stålhane [3] agree to the fact that the current SOV fleet is heterogeneous and to the operation profile consisting of a shore-based vessel that travels to the wind farm depending on the maintenance needs and its cost-efficiency. The main characteristics of these vessels that affects the wind farms' O&M are the accommodation capacity, speed, transfer velocity and wave-height sensitivity, among other minor ones like cargo capacity, comfort, etc.

When it came to the routing and scheduling, the following insights resulted to be relevant for implementing the model: Dai [5] defines accessibility (access to the wind turbines), transport capacity (personnel and cargo) and turbines' maintenance needs as the main variables to be considered. Zhang [6] evaluates the availability of the offshore wind farm and work load on the service vessels to determine or improve the maintenance plan.

Faulstich [7] proposes that modern onshore wind turbines in Europe achieve a high availability of 95–99%. However, despite the wind turbine's technology progress, the reliability has declined with growing turbine size. The failures also appear to be occurring at an approximately constant failure rate over a large period of turbine life, and calculating MTBF as the inverse of failure rate λ appears reasonable. Finally he mentions that major failures account for only 10% of the total.

Ossai [8] and Bi [9] process and analyze data by using a Markovian approach and data comparison respectively.

When it comes to the maintenance needs, the average values of the expected planned and unplanned maintenance is also commonly accepted as 1-2 for preventive and 2-4 for corrective [2] [3] [10] [11] [7] [8] [9].

3.2 Ship Design

Many design approaches can be applied (and have been applied) for dealing with the high complexity that ship design involves. The development of new approaches contribute with new tools for the designers, but they do not completely make the previous ones obsolete. Among the many available approaches and techniques available, the following have been found suitable for applying to the current case.

Martin [12] proposes a modification of the systems engineering “V” diagram for applying to large and complex ships. Which matches with the proposal of Fonseca and Gaspar [13] of the usage of a well defined structured hierarchy to represent the ship as an entity and state model.

use the SFI group system as hierarchy structure to represent the ship as an entity and state model.

Van Breuinessen et al. [14], proposes the classification of the ship design approaches into two main categories: “derivative design” (developed from previous vessels) and “blank-sheet of paper design” (when a new concept is developed). Subsequently the innovation degree is divided into three levels: “incremental innovation” (derived from existing concepts with limited innovations), “differentiating innovation” (aimed to make a substantial difference without changing the concept completely) and “radical innovation” (implement a completely new concept and/or technology for the industry).

Then, to handle both the complexity and uncertainty of the ship design applied to the SOV, it appears to be an agreement between different sources on making use of a structured hierarchy in order to apply an upside-down process (as in systems engineering). When it comes to adapting the current fleet to the new logistic challenges, the derivative design with incremental innovation seems to be the most suitable, since the needs of the new offshore wind farms can be derived from the

current ones.

(PAGE INTENTIONALLY LEFT BLANK)

4 | Methods and Tools

4.1 SimEvents and Simulink

“Simulink” is a block diagram environment for Model-Based Design”. It works together with MATLAB for combining graphical and textual programming [15].

The simulation has been carried out with SimEvents which is a discrete-event simulation engine for Simulink that enables event-driven systems, that is, systems in which events drive changes in states or values. In this type of simulation discrete “entities” are created and simulate items (the SOV in the present case), and are characterized by “attributes”. During the simulation, such entities pass through and are managed by “blocks” like queues, servers, gates, switches, etc., that can be programmed by MATLAB scripts [16].

4.2 Markov Chain

The sea state affects the sea-going capabilities of the SOVs and CTVs to a large extent. The two main effects considered in the model are the sailing time from the base to the farm and vice-versa and the ability of both (SOV and CTVs) to perform transfer operations.

A Markov chain is a model that defines a number of states and the probability of changing from one state to another or staying in the current one. It is a stochastic process in which the probability distribution of the next state does not depend on the previous ones, only on the current. Figure 4.2 shows an example of a 3-states transition diagram that depicts the application to a 3-conditions sea state by showing

them and the transition probabilities to other states and staying in the current one. Table 4.1 is the transition matrix relating to the transition diagram shown in Figure 4.1 and contains the transition probabilities arranged in rows (note that the sum of all the elements of each row must be 1).

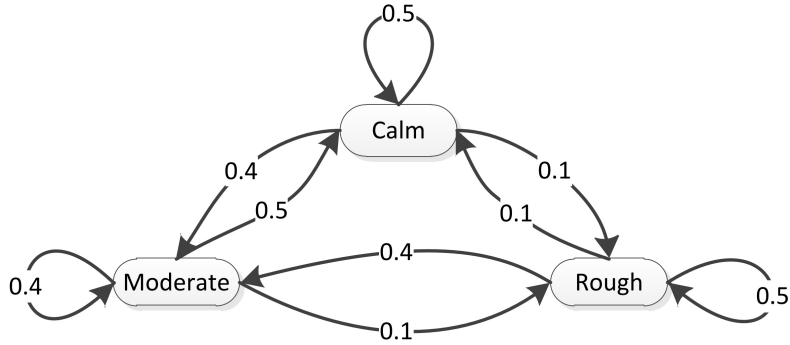


Figure 4.1: Markov Chain Transition Diagram [17]

Table 4.1: Transition Matrix for Figure 4.1

	Calm	Moderate	Rough
Calm	0.5	0.4	0.1
Moderate	0.5	0.4	0.1
Rough	0.1	0.4	0.5

For the application of the Markov chain concept to the current model, wave height data has been obtained from the website waveclimate.com at intervals of 3 hours from January 1st 1992 to December 31st 2016 in a location in the North Sea. 12 states were taken uniformly distributed between the maximum and the minimum wave height given and the transition probability by counting the state changes. This has been done by using MATLAB scripts that are included in the Appendix ???. Then the transition matrix (Table 4.1) was obtained. Then it is used by the “sea state” entity generator (Section 5.1) to account for the impact of the sea state on the sailing time.

The detailed calculation process is included in the Appendix A

An important care had to be taken while applying the Markov chain, and that was avoiding absorbing states, which are the states from where it is impossible to leave, which should make the sea state impact uniform.

Table 4.2: Transition Matrix

state	1	2	3	4	5	6	7	8	9	10	11	12
1	0.8	0.2	0	0	0	0	0	0	0	0	0	0
2	0.032	0.892	0.076	0	0	0	0	0	0	0	0	0
3	0	0.090	0.812	0.095	0.003	0	0	0	0	0	0	0
4	0	0	0.134	0.767	0.099	0	0	0	0	0	0	0
5	0	0	0	0.180	0.662	0.137	0.021	0	0	0	0	0
6	0	0	0	0	0.201	0.651	0.140	0.004	0.004	0	0	0
7	0	0	0	0	0	0.257	0.572	0.145	0.026	0	0	0
8	0	0	0	0	0	0.013	0.355	0.474	0.118	0.026	0.013	0
9	0	0	0	0	0	0	0	0.366	0.512	0.098	0.024	0
10	0	0	0	0	0	0	0	0.125	0.375	0.313	0.188	0
11	0	0	0	0	0	0	0	0	0	0.4	0.1	0.5
12	0	0	0	0	0	0	0	0	0	0.071	0.286	0.643

4.3 Sea-State Influence on Operation and Maintenance

The sea state also determines whether both the SOV and the CTVs can or not transfer personnel safely to the turbines. Nowadays SOVs can perform transfers with significant wave heights up to 3m and the CTVs range from 1,5 to 2m. During a typical operation, a sea state forecast is available and is checked by the crew regularly to not only schedule the work, but also to decide if transfers will be performed a certain day.

For accounting the wave height impact in the transfer operation and, therefore, in the possibility of performing maintenance tasks, the data taken from waveclimate.com was, only for that matter, assumed to be data for creating daily forecasts. Under this assumption, a variable called WaveForecast was created, which is a 2-dimension vector that include the wave height forecast for the daytime window during which the operations are performed. The first component of the vector is the forecast from 06:00 AM to 12:00 PM and the second component the forecast from 12:00 PM to 18:00 PM every day. As explained in Section 5.6.2. After checking the vector, the decision to deliver a certain amount of maintenance crews is taken, knowing that they can be picked up afterwards.

4.4 Maintenance Tasks Modelling

4.4.1 Preventive Maintenance (PM)

As stated in Section 1, the preventive maintenance need of each wind turbine is two visits per year [2] [3] [10] [11]., and since each wind farm is assumed to have 100 wind turbines (as stated in 2), the PM tasks yearly need will be 200. Since they can be set to be performed in the time of the year which is more convenient and safe for the operation, the first maintenance works will be scheduled from March and the second from September. Then, generating 100 PM tasks in March and other 100 in September. This way they will be as equally distributed along the year as possible and with relative good weather and sea conditions for the operation.

4.4.2 Corrective Maintenance (CM)

According to Faulstich et al. [7], failures appear to be occurring at an approximately constant rate over a large period of turbine life, then they can be modelled with a Poisson probability distribution.

For the model an average value of 3 CM tasks per year is taken, as a commonly accepted value is between 2 and 4 [2] [3] [10] [11].

Then the probability of a at least one failure occurring in one wind farm (consisting of 100 turbines) in the period of one hour (the model's time step) (P_f) will be:

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!} \Rightarrow P_f = 1 - P(0) = 1 - \frac{0.03425^0 \times e^{-0.03425}}{0!} = \underline{0.03367}$$

Where:

$$\lambda = \frac{100 \times 3}{8760} = 0.03425. \text{ Expected number of occurrences.}$$

x = Number of failures

In the same way, the probability of occurrence of 1, 2, 3 or 4 failures will be:

$$P(1) = 0.033094, P(2) = 0.00056667, P(3) = 6.4689 \times 10^6, P(4) = 5.5406 \times 10^6$$

These values are used in the model (see 5.5) to generate the CM tasks to be performed by the technical personnel, thus a transfer need to a wind turbine. For such calculations it is considered that up to 3 simultaneous failures can occur in a farm, since the probability of 4 or more simultaneous failures in the same wind farm (as calculated above) is remote, and would add around 25% more simulation time.

The CM maintenance tasks include both, minor failures that requires to take the wind turbine out of service only during the maintenance works (minimum down time), and failures that require the wind turbine being out of service from the moment the failure is detected. According to Faulstich [7] these failures account for 10% of all the CM tasks

4.5 Wind Farms Modelling

As stated in 2, for modeling the wind farms, an hypothetical one has been created taking as a base the planned Sørilige Nordsjø (Category A area) Offshore Wind Farm that is planned to be installed in the region of Rogaland, Norway before the year 2025. [4]

The characteristics of the wind farm to be served are:

- Distance to shore: 149 km (80,45 nm).
- Number of turbines: 100.
- Turbine capacity: 10 MW.

4.6 Costs

The main costs to be accounted are the O&M costs and the wind turbines downtime costs.

The current trend for O&M for the operators is to charter both SOVs and CTVs on a time base. Thus the charter price must be taken as a fixed cost, and

an important factor to improve the cost efficiency is to reduce the waiting times to a minimum and maximize the operating times. An estimation of the vessels' costs that covers both the chartering costs and the voyage expenses could be 10726 EUR/day for the SOV and 1976 EUR/day for each CTV [11].

There is uncertainty in the price of the electricity generated by offshore wind farms, and it is still very expensive under commonly used metrics [18], but the governments have been supporting the offshore wind energy with subsidies, ensuring to the wind farm owner a market price for a certain period. As an example, UK ensured a price of 117,14 £/MWh (165 EUR/MWh) in 2015. [18].

In the other hand, when analyzing the costs, the Levelized Cost of Electricity (LCoE) is used, which incorporates all the power station's lifecycle costs (i.e. capital expenditure (CAPEX), O&M, fuel and decommissioning costs) and divides the discounted sum of such costs by the discounted lifetime output from the station, thus, a levelized lifetime cost per MWh is obtained. From the Figure 4.2 a value of 130 EUR/MWh for the LCoE can be estimated for the year 2015. [18]. Finally, the gross margin for the company could be estimated in $165 - 130 = 35$ EUR/MWh

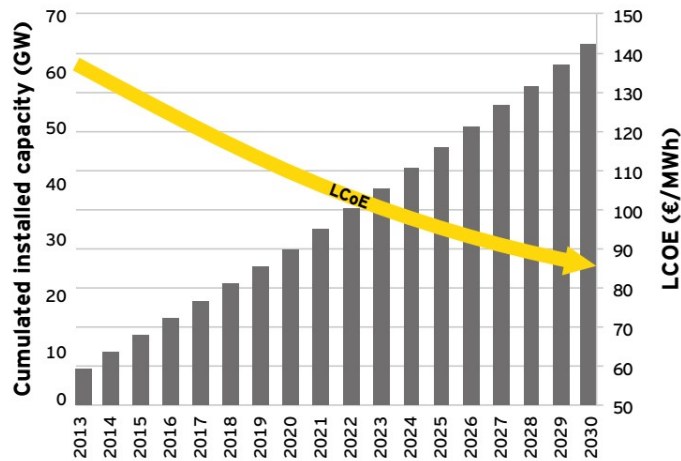


Figure 4.2: Evolution of the LCoE according to the cumulated offshore wind capacity installed

5 | Model Description

Section 1 describes a typical operation of an SOV for maintenance of wind turbines in offshore wind farms. Such operation can be described with the flowchart showed in Figure 5.1. The SOV waits in the shore base until a failure occurs in a turbine and takes it out of service, or enough maintenance tasks (preventive and/or corrective) have been cumulated and makes the voyage to the wind farms and the operation cost-effective. Then the vessel sails to the operation site and will operate when three conditions are given simultaneously: the time elapsed of the shift is less than the maximum, there is daylight and the wave height is less than the maximum allowed for safe operation of the SOV and/or CTVs. Once all the scheduled tasks have been carried out or the planned shift time is over, the vessel sails back to the base.

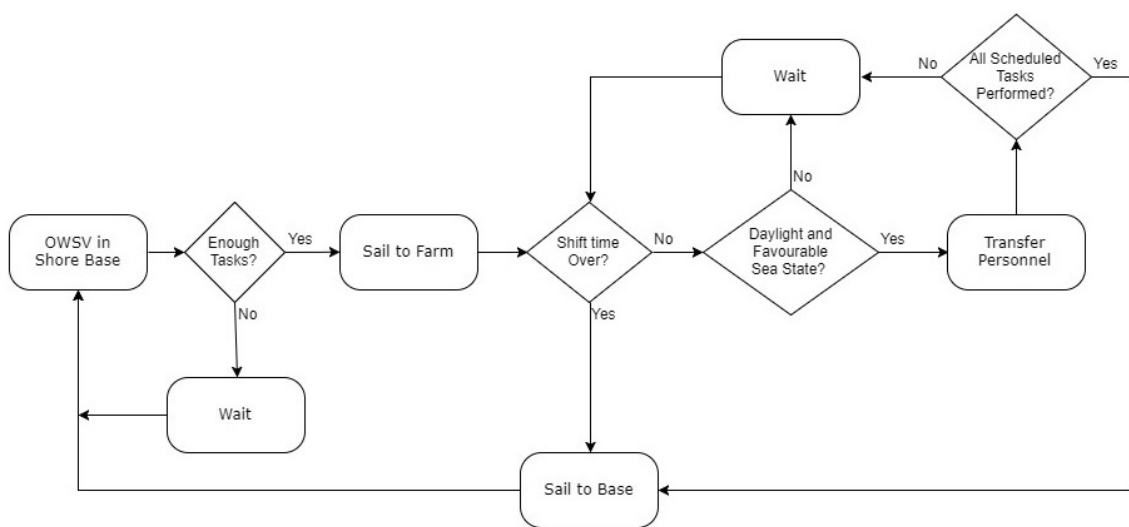


Figure 5.1: Operation Flowchart

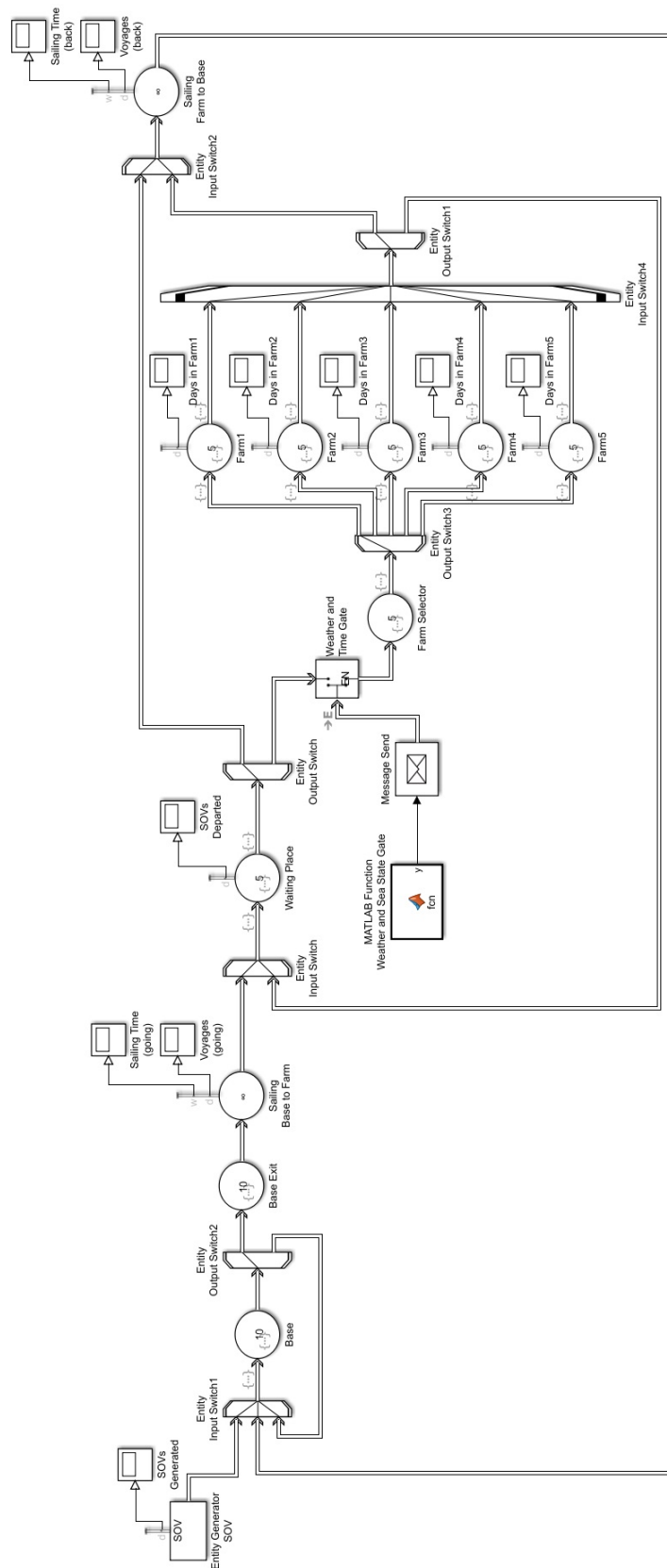


Figure 5.2: Simulink Model

The operation has been modeled in Simulink, and the model is depicted in Figure 5.2. Its main parts are:

- SOV Entity Generator.
- Base Subsystem.
- Sailing Entity Servers.
- Wind Farms Subsystem.
- Maintenance Tasks Generation (not shown in Figure 5.2).
- Sea State and Wave Forecast (not shown in Figure 5.2).
- Costs Calculation (not shown in Figure 5.2).
- Global Variables (not shown in Figure 5.2).
- Constants (not shown in Figure 5.2).

5.1 SOV Entity Generator

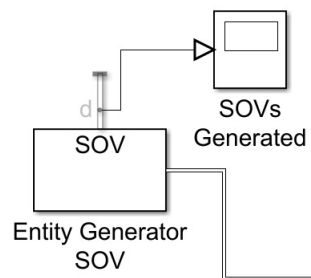


Figure 5.3: SOV Entity Generator

Figure 5.3 shows the Simulink entity generator named “Entity Generator SOV”, its function is only to generate one entity at the simulation’s start which represents the SOV and will flow through the simulation model. The entity is endowed with two attributes: “speed”, that will be used for sailing simulation and “home”, used for routing through the model. The scope “SOVs generated” plots the amount of entities generated (one).

Block’s script is included in Appendix B.1.

5.2 Base Subsystem

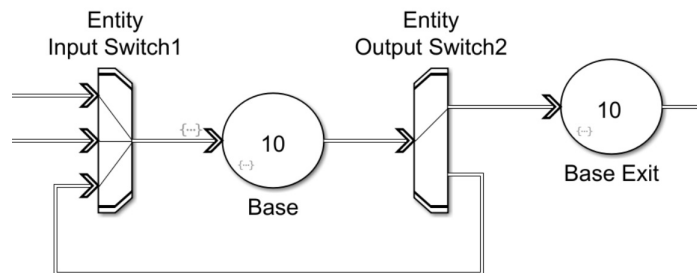


Figure 5.4: Base Subsystem

The shore base where the SOV starts each operation is simulated with the “Base” subsystem, which is depicted in Figure 5.4. The “Entity Input Switch1” receives the entity “SOV” and sends it to the entity server block “Base”. There the waiting times in base are accounted (global variable `WaitingTimeB`) and routes the entity to the farms if the amount of pending maintenance duties makes the operation cost-effective, or makes it stay in the base (through the loop that goes back to the “Entity Input Switch1”) if not. Finally the block “Base Exit” sets the values of the variables “`ShiftStart`” and “`SOVinBase`” that are used to measure the time of every shift and to indicate when the SOV is in the shore-base respectively.

The scripts for the entity generators are included in the Appendix B.2.

As stated in Section 2, in order to not add unnecessary complexities to the model, it does not consider the part of the operation of loading spare parts, tools and consumables, and embarking technical personnel, since it is considered to be under the maintenance duties planning and, thus, out of the scope of the present study.

5.3 Sailing Entity Servers

Figures 5.5a and 5.5b depict the entity servers “Sailing Base to Farm” and “Sailing Farm to Base” respectively and model the sailing voyages of the SOV from

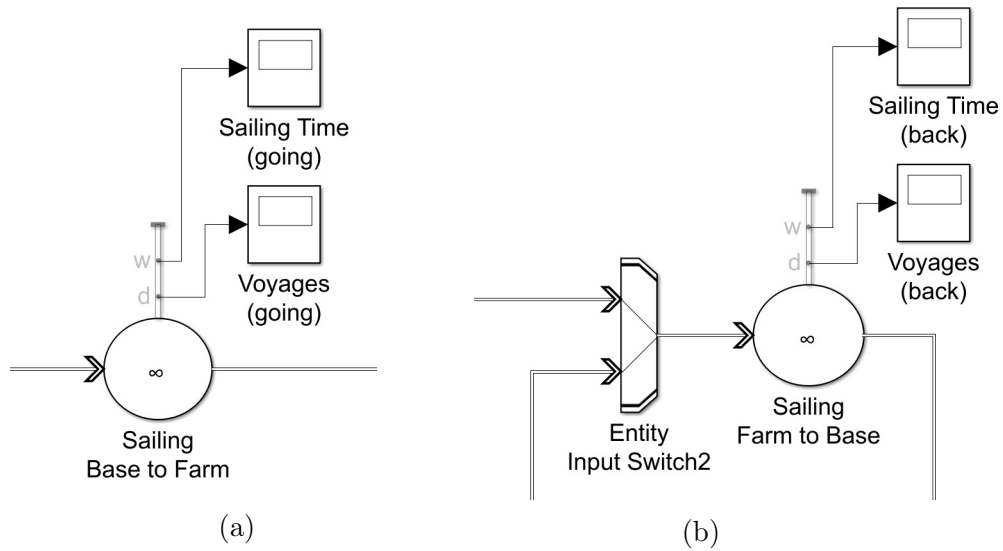


Figure 5.5: Sailing Entity Servers

the shore base to the wind farm and back. They calculate the sailing time by dividing the distance from the base by the speed and multiplying it by a factor that accounts for the sea-state calculated by making use of a Markov Chain, as explained in 4.2. The scopes connected to each block show the sailing times and the number of voyages (thus shifts) performed during the simulation.

The scripts for both sailing entity blocks are the same and are included in the Appendix B.3.

5.4 Wind Farms Subsystem

Wind farms subsystem is depicted in Figure 5.6. After leaving the entity server "Sailing Base to Farm", the entity goes into the next server called "Waiting Place" and remains there until the conditions for safe operation are given, that is: daylight (considered from 06:00 to 18:00) and significant wave height less than 3m (maximum for allowable SOV). That is performed by the "Weather and Time Gate" which is commanded by a MATLAB function. When the maximum time for a shift is passed, the entity goes to the "Sailing Farm to Base" server and will be led to the "Base". After the entity has passed the gate, the "Farm Selector" entity server routes the SOV to the wind farm with more turbines out of service or, if no wind turbines are out of service, to the one with more maintenance duties pending. When the SOV

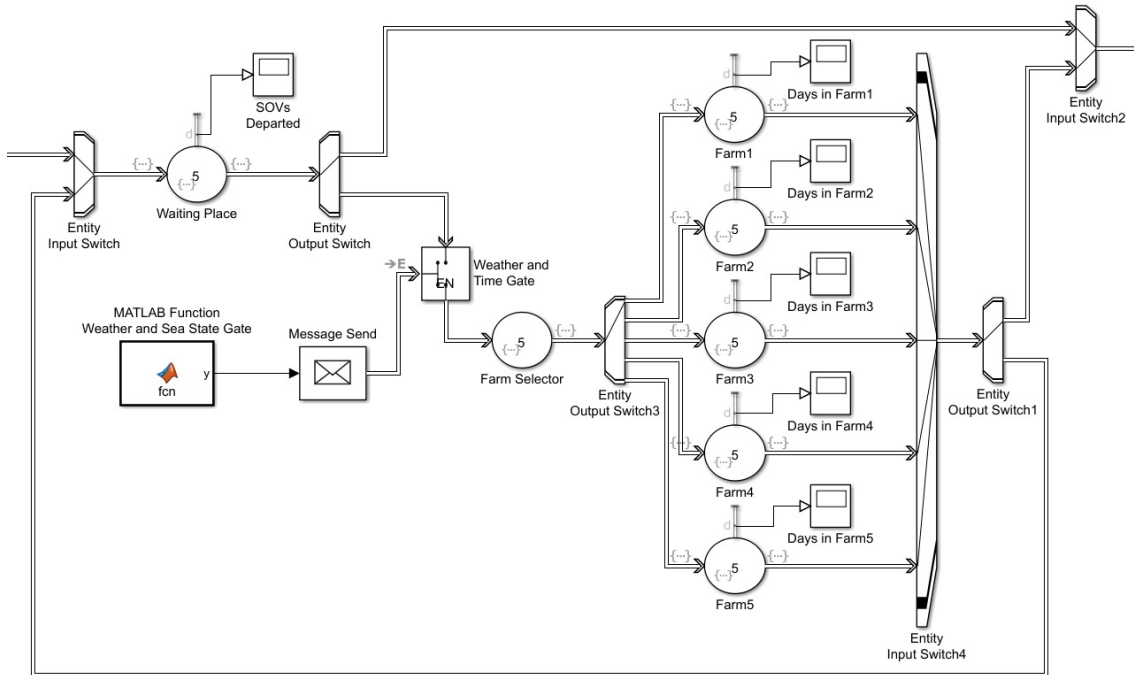


Figure 5.6: Wind Farms Subsystem

is in the farm, it will perform as many transfers of technical personnel to the wind turbines as possible either alone, or by in combination the CTVs. When all the tasks have been carried out, or the working day is over, the entity leaves the server. If there is still maintenance to be done, it goes to the “Waiting Place” block. If not, it starts sailing to the “Base”.

The scopes connected the the entity servers “Waiting Place” and ”Farm” plot the times the SOV departed to operate in a farm and the operation time in each farm respectively.

The scripts for the described blocks are included in Appendix B.4.

Distances between farms to the waiting place and from one farm to other are not considered, since they are supposed to be sailed during the night. Attending two farms during the same day is not considered because of the time needed to reach one farm from another. Doing so the SOV would be sailing during hours that can be used for operations instead.

5.5 Maintenance Tasks

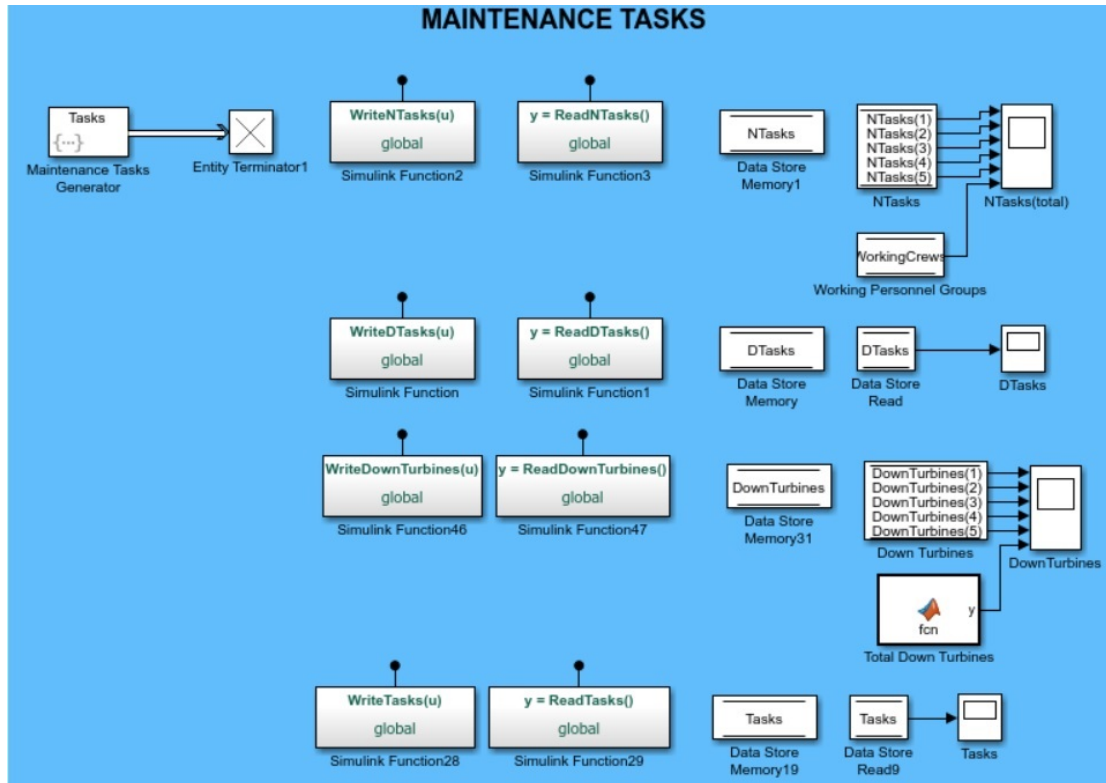


Figure 5.7: Maintenance Tasks Generation

In the model the maintenance tasks that will generate a need for transferring generate technical personnel, are generated in the entity generator block named “Maintenance Tasks Generator” shown in Figure 5.7. It generates:

- **PM tasks:** for each wind farm it generates two PM tasks per year and turbine in the farm, the first at the end of March and the second at the end of September. Thus they are distributed evenly through the year and can be performed in times when the sea state is more favourable for safe operation.
- **CM Tasks:** Are generated randomly following the model described in Section 4.4.
- **Wind Turbines Out of Service:** Are generated randomly and represent the amount of wind turbines out of service. This number is included in the

PM Tasks calculated above, since they generate maintenance actions due to failures

When generating PM and CM tasks, in order to save computational resources, both are assumed equal for all the farms. The variable that is taking independent for each farm is “DownTurbins” (see Section 5.8), thus making the routing to the farms more aleatory. The model has been tested creating independent PM and CM tasks, resulting in almost double running time.

Each variable’s data store read is connected to a scope, which plots the value of the variable against time. The scripts for the block is included in Appendix B.1.

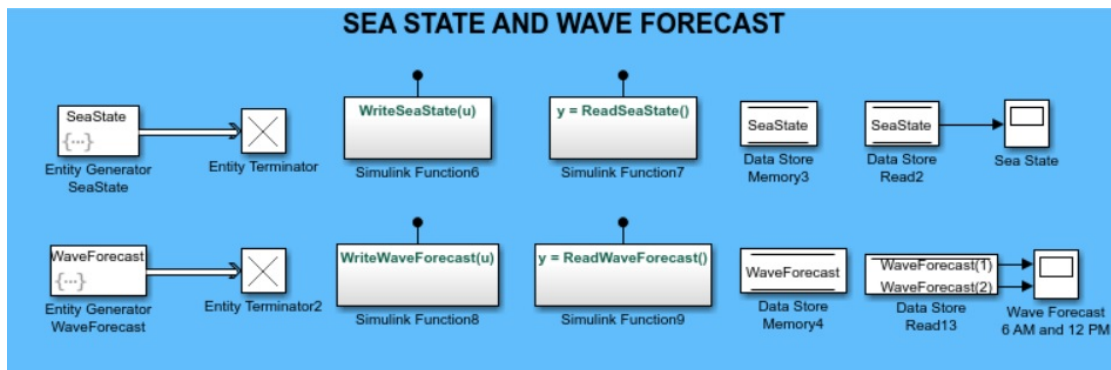


Figure 5.8: Sea State and Wave Forecast Entity Generators

5.6 Sea State and Wave Forecast Entity Generators

The two effects of the sea state in the model are the influence on the speed, therefore in the sailing times and the wave height that will allow or avoid the SOV and CTVs to transfer technical personnel to the wind turbine. The entity generators and global variables used for the implementation are depicted in Figure 5.8. Each variable’s data store read is connected to a scope, which plots the value of the variable against time.

5.6.1 Sea State Entity Generator

As explained in Section 4.2 the sea state that affects the sea-going capabilities of the vessel (speed in the present study) is modeled by using a Markov Chain. The Generator reads the sea states MS Excel file created (see Appendix A) and picks a value randomly, then changes from one state to another also randomly. Finally the sea state value (variable “SeaState”) is saved to be used by the “Sailing” block servers.

The script for the “Sea State” entity generator block is included in Appendix B.6

5.6.2 Wave Forecast

The “Wave Forecast” entity generator creates an array of 2 components. Each component is the expected wave height during a period of 6 hours, from 6:00 AM to 12:00 PM and from 12:00 PM to 6:00 PM. The vessels will transfer technical personnel to the wind farms depending on the the value.

To create such an array, the script reads a .csv file with the sea data taken from waveclimate.com and will extract the data that corresponds to the current date and generate the array the values of the wave heights for the times needed.

Even though the model uses historical data, it is taking as a forecast, since in a real situation, the SOV will have real and reliable informed from actual forecasts.

The script for the “Sea State” Entity enerator block is included in Appendix B.7

5.7 Costs

Under the assumptions stated in Section ??, the costs are calculated by the model by the use of a MATLAB function whose inputs are Simulink constants for the daily costs (SOV chartering, SOV operation, CTV operation and loss due to turbine downtime). The outputs (global variables) are the total cost, the operational

cost and the cost generated by the wind turbines that are out of service (strictly a loss). Each variable's data store read is connected to a scope, which plots the value of the variable against time.

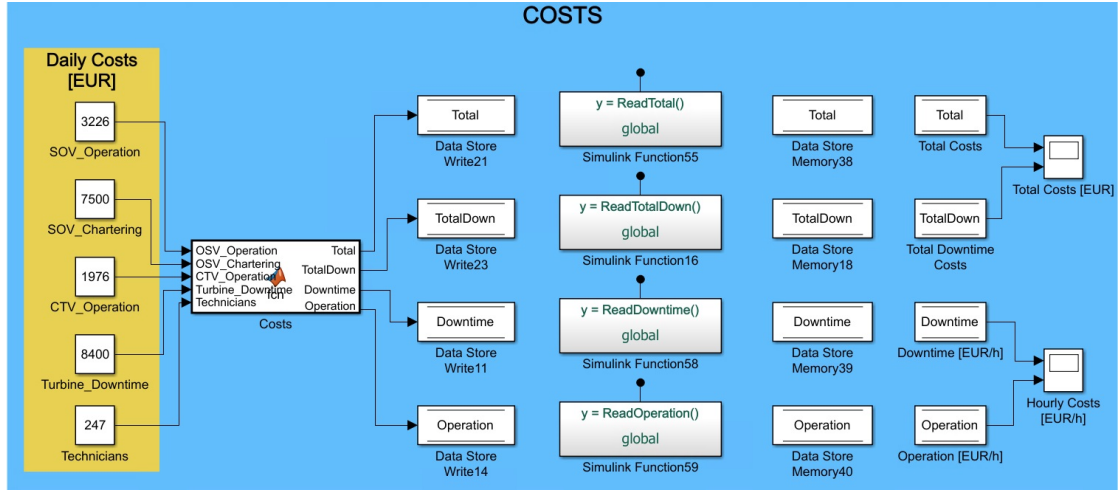


Figure 5.9: Costs Function

When operating, the total operating cost is:

$$Cost_{operation} = Cost_{SOV-ch} + Cost_{SOV-op} + Cost_{CTV} \times N_{CTV} + Cost_{tech}$$

The cost (loss) generated by the wind turbines out of service is:

$$Cost_{down-total} = Cost_{down} \times N_{down}$$

Then, the total cost is:

$$Cost_{total} = Cost_{operation} + Cost_{down-total}$$

Where:

$Cost_{operation}$	=	Total hourly cost of the vessels' operation in [EUR]
$Cost_{SOV-Ch}$	=	Hourly cost of SOV chartering in [EUR]
$Cost_{SOV-Op}$	=	Hourly cost of SOV operation in [EUR]
$Cost_{CTV}$	=	Hourly cost of CTV operation in [EUR]
N_{CTV}	=	Number of CTVs deployed in the operation.
$Cost_{tech}$	=	Cost of the technical personnel on board [EUR]
$Cost_{down-total}$	=	Hourly loss generated by the wind turbines out of service in [EUR]
$Cost_{down}$	=	Hourly loss generated by one wind turbine out of service in [EUR]
N_{down}	=	Number of turbines out of service
$Cost_{total}$	=	Total hourly cost of the operation in [EUR]

Total cumulated cost is the integral of the hourly cost along the elapsed time:

$$Total\ Cumulated\ Cost = \int_0^t Cost_{total}(t) dt$$

The script of the Cost MATLAB function is included in the Appendix B.8

5.8 Global Variables, Constants and other MATLAB Functions

The following global variables and constants were used in the model. Each variable's data store read is connected to a scope, which plots the value of the variable against time. :

5.8.1 Global Variables Used For Maintenance Tasks Generation

As shown in Figure 5.7, for maintenance tasks generation, the following global variables are used:

- **NTasks:** Global variable. Is a 5 components array, each of them is the amount of maintenance tasks (both PM and CM) generated on each wind farm. Every time the vessel calls in the base the value of NTasks increases with the amount of maintenance tasks generated during the last shift. In a similar way, every time the SOV calls in a wind farm, the corresponding component decreases in an amount equal to the maintenance tasks performed (transfers). When a component of NTasks decreases, the global variable DTasks increases in the same value.
- **DTasks:** Global variable. Is the amount of transfers of technical personnel to the wind turbines, and thus, the amount of maintenance tasks performed. It increases its value every time a technicians group is picked from a wind turbine.
- **DownTurbines:** Global variable. 5 components array. Each of them is the amount of wind turbines out of service in the corresponding farm. The variable is handled by the “Maintenance Tasks” entity generator, and its value increases following a poisson distribution. It decreases each time the corresponding maintenance task is performed, that is, technical personnel is transferred (action carried out in the “Farms” entity servers). The variable is also handled in the “Total Down Turbines” MATLAB function. This function makes the sum of all components of the array, giving the total amount of turbines out of service in all the farms being attended by the SOV.
- **Tasks.** Global variable that acts as an intermediate value for assigning preventive maintenance tasks.

5.8.2 Global Variables and MATLAB Functions Used for Times Calculations

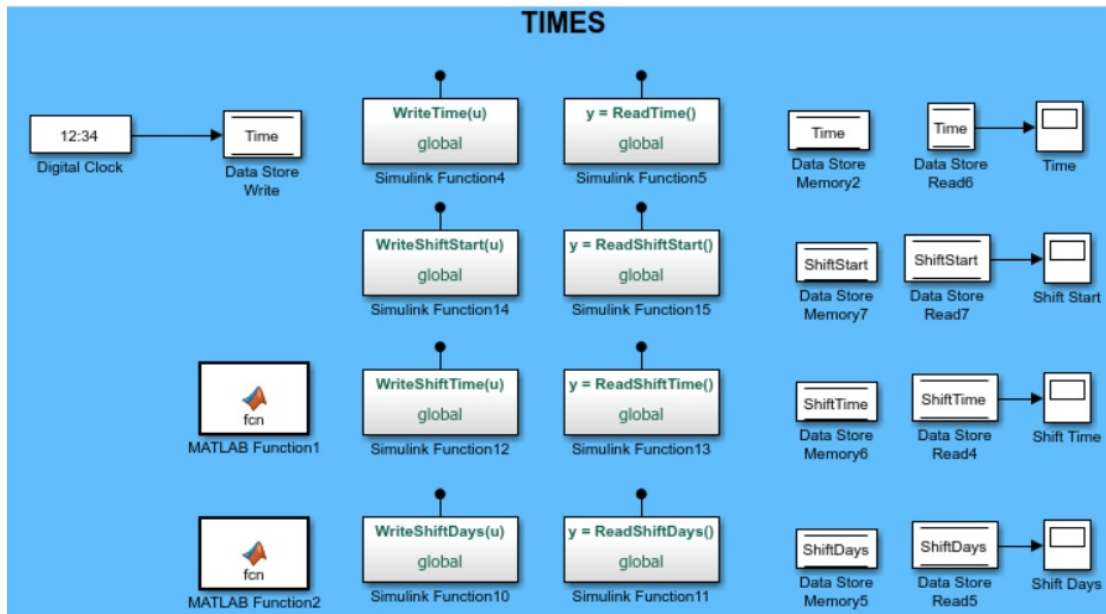


Figure 5.10: Global Variables and MATLAB Functions Used for Times Calculations

Figure 5.10 shows the global variables and MATLAB functions used for times calculations. These are:

- **Time**: Global variable. Elapsed time of the simulation. The time is generated by a digital clock block and written in a data store memory.
- **ShiftStart**: Global variable that sets the starting time of the current shift. This takes place each time the SOV leaves the entity server “Base Exit”.
- **ShiftTime**: Global variable. Elapsed time of the current ship. Used for the CM tasks generation. It is governed by a MATLAB function and performs the operation: $\text{ShiftTime} = \text{Time} - \text{ShiftStart}$. The script for the function is included in Appendix B.9
- **ShiftDays**: Global variable. Gives the days passed since the shift started. It is used in the entity server “Waiting Place” and when its value reaches the maximum planned duration of the shift, the SOV is sent back to the shore base. Is is governed by a MATLAB function, whose script is included in the Appendix B.10.

5.8.3 Global Variables for Waiting Times Calculations

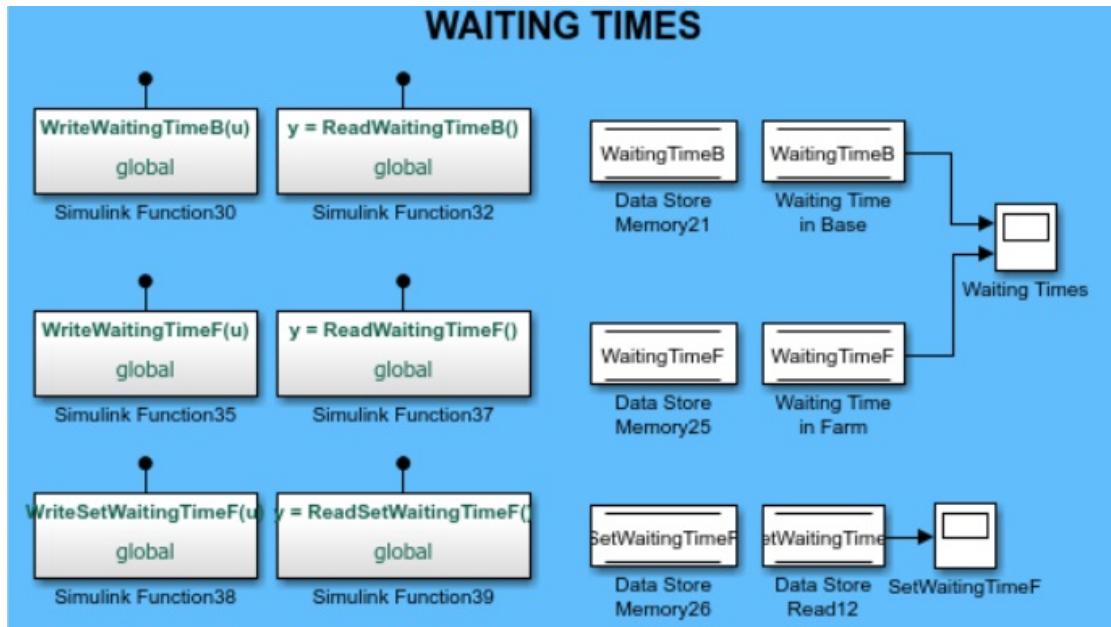


Figure 5.11: Global Variables for Waiting Times Calculations

As stated in Section 5, the SOV will wait in the shore base until the amount and characteristics of the maintenance tasks make the voyage to the wind farms cost effective. In the other hand if the time of the day and/or the weather are not favourable for operation (night and wave height more than 3 m) the vessel will have to wait to operate until the conditions change. These waiting times lead to extra costs.

Figure 5.11 shows the global variables used for the calculation of the waiting times in base and on site. Each variable's data store read is connected to a scope, which plots the value of the variable against time.

The constants are:

- **WaitingTimeB**: Global variable. Is the cumulated time the entity waited in the “Base” subsystem, that is, the SOV waits in the shore base without operating. Its value is increased in 1 each time the vessel enters in the “Base” entity server. The variable is able to account for the waiting time because the entity never waits in the server, if it does not leaves the subsystem, it goes

through the loop back to the server's entrance.

- **SetWaitingTimeF** Global variable. Intermediate variable used to calculate the time the SOV waits in the farm. It takes the value of the variable "Time" when the entity enters the "Waiting Place" entity server.
- **WaitingTimeF** Global variable. Is the cumulated time the entity waited in the "Waiting Place" entity server, that is, the time the SOV is on site without operating. Its value is calculated in the "exit" event action of the "Waiting Place" entity server and is obtained by subtracting to the variable "Time" the values of the variable "SetWaitingTime" and the night times, which do not account as wasted times since the SOV not only can not operate but she can sail to other wind farm.

5.8.4 Global Variable for Transferred Technical Personnel Calculations

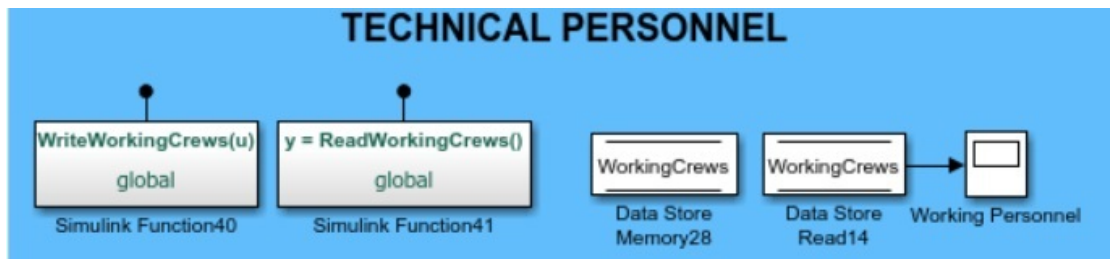


Figure 5.12: Global Variable for Transferred Technical Personnel calculations

Figure 5.12 shows the global variable used for the calculation of the transferred personnel. The variable's data store read is connected to a scope, which plots the value of the variable against time.

- **WorkingCrews**: Is the total the amount of technical personnel groups transferred each day. That means it also accounts for the transfers performed and the maintenance tasks carried out each day. It is calculated in the "Farm" entity servers, increasing its value each time a transfer is performed. Note that the total amount of transfers is calculated by the global variable "DTasks"

5.8.5 Global Variables Used for SOV Placing

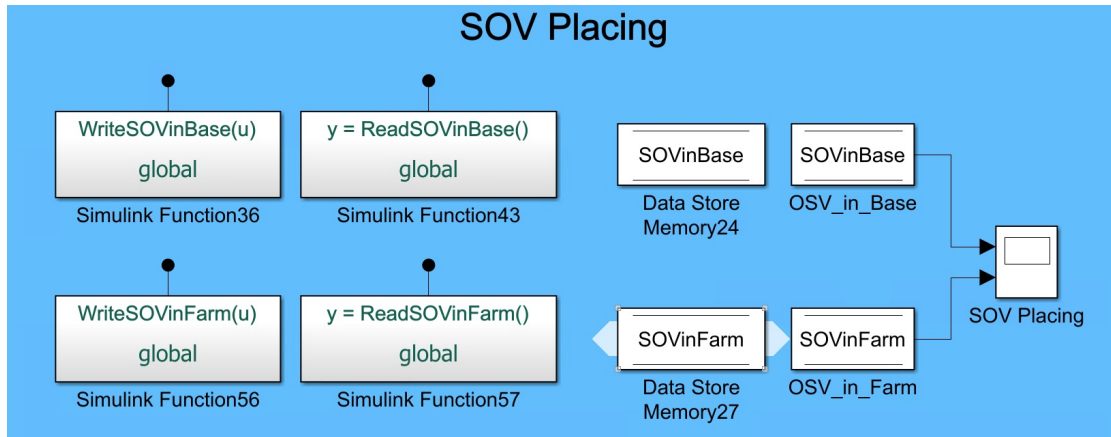


Figure 5.13: Global Variables for Used for SOV Placing

Figure 5.13 shows the global variables used to determine when the SOV entity is in the “Base” or any of the “Farm” entity servers. They will be used in costs calculations. Each variable’s data store read is connected to a scope, which plots the value of the variable against time.

- **SOVinBase:** Global variable. Takes the value 1 when the SOV is in the “Base” entity server and 0 when it is not. It will indicate when the SOV is in the shore base not operating. It is achieved by setting its value either to 1 or 0 each time the entity enters or leaves the server respectively.
- **SOVinFarm:** Global variable. Takes the value 1 when the SOV is in any of the “Farm” entity servers and 0 when is not. It means that it indicates the times in which the SOV is actively operating transferring technical personnel. It is achieved by setting its value either to 1 or 0 each time the entity enters or leaves the server respectively.

5.8.6 Constants Used to Define Vessels’ Characteristics

Figure 5.14 shows the constants that are used in the model to define the vessels’ characteristics. They are:

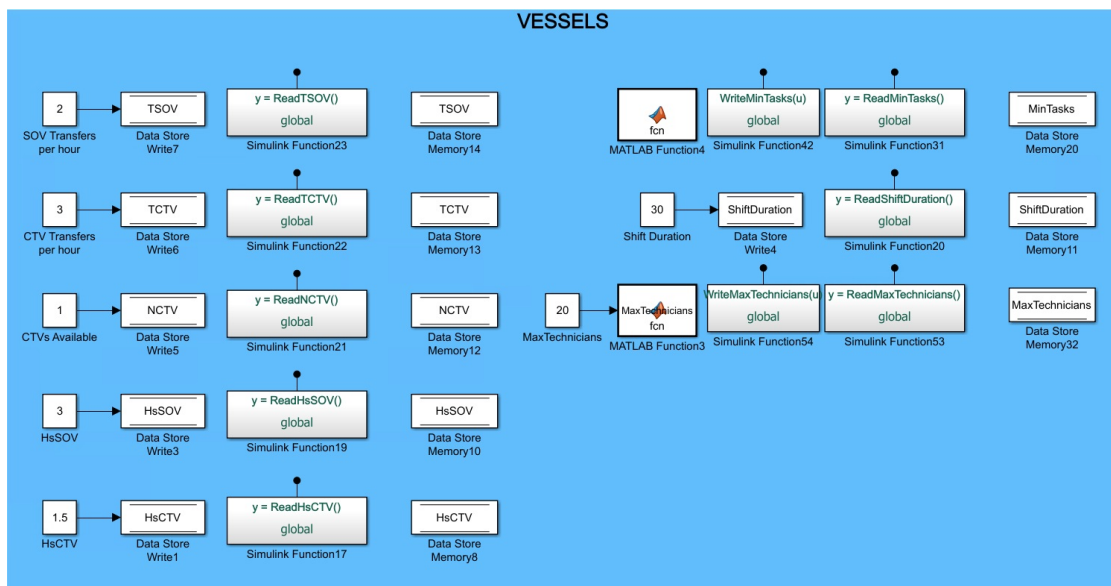


Figure 5.14: Constants Used to Define Vessels' Characteristics

- **TSOV**: Maximum amount of transfers the SOV can perform in an hour. Usually $TSOV=2$.
- **TCTV**: Maximum amount of transfers a CTV can perform in an hour. Usually $TSOV=3$.
- **NCTV**: Amount of CTVs available on the SOV that can be deployed.
- **HsSOV**: Maximum safe operation wave height for the SOV, in meters. Defined by the SOV wave behavior and the transfer methods. Most of the SOV can operate with a significant wave height of up to 3m. Therefore $HsSOV = 3$.
- **HsCTV**: Maximum safe operation wave height for the CTV, in meters. Defined by the CTV wave behavior and the transfer methods. Usually the CTVs operate with wave heights up to 1.5 m, or 2.0 m the newest designs. Therefore $HsCTV$ will range from 1.5 to 2.m.
- **MinTasks**: Is the minimum amount of maintenance tasks that have to be pending to send the SOV to the wind farms to perform them. By use of a MATLAB function (its script is included in Appendix B.11), it can take different values in summer or winter. Its value will be defined depending on the final cost of each transfer.

- **Shift Duration:** Maximum time a shift can last. Usually between 15 and 30 days [3].
- **MaxTechnicians:** Maximum amount of groups of technicians available on-board the SOV to be transferred. It is set manually and will define the SOV accommodation capacity and the amount of maintenance tasks that can be performed per day. Its maximum value is defined by the SOV and CTVs combined transfer capacity and the SOV accommodation. It is used by the “Farm” entity servers (Section 5.4) and handled by a MATLAB function whose script is included in Appendix B.12.

5.8.7 Constants Used to Define Wind Farms’ Characteristics



Figure 5.15: Constants Used to Define Wind Farms’ Characteristics

Figure 5.15 shows the constants that are used in the model to define the wind farms’ characteristics. They are:

- **NTurbines:** Amount of wind turbines in each wind farm. Used by the "Maintenance Tsks" generator to create a realistic amount of maintenance tasks.
- **SailingDistance:** Is the distance from the shore base to each wind farm. Used in the "Sailing" entity servers to calculate the sailing times. It has to be noted that the distance between farms is not considered, since these distances are supposed to be short enough to being able to be sailed during the night time or when the sea state does not allow the SOV to operate.
- **CM:** Is the average amount of CM services a wind turbine needs per year. Usually $CM=3$ [3].
- **PM:** Is the average amount of PM services a wind turbine needs per year. Usually $PM=2$ [3].
- **NFarms:** Is the maximum amount of farms to be attended by one SOV (and its CTVs). It is set manually and the maximum value is 5.

Even though independent variables were defined for each wind farm, when generating the maintenance tasks, their values were taken equal to each other, since generating independently for each wind farm leads to an increase of around 30% of the model running time and does not contribute to enrich the results and conclusions of the present work.

Waiting time during the shifts due to rough sea state:

If the time of the day and/or the weather are not favourable for operation (daylight and wave height less than 3 m) the SOV will have to wait to operate until the conditions change. For accounting for that waiting time, which will lead to an unnecessary (but almost unavoidable) extra cost the following method has been applied:

Variables:

- **SetWaitingTimeF (SWT):** takes the value of the variable 'Time' when entering in the 'Waiting Place' entity server.
- **WaitingTimeF (WT):** Account for the waiting time in or close to the farm/s. WT_1 is the waiting time during the current shift and WT_0 is the waiting time cumulated from previous shifts.

- NightTime (T_{night}): accounts for the night hours during the shift, in which there are no operations, thus not accounting as waiting hours that will lead to extra costs.

Then, the total waiting time will be:

$$WT_1 = WT_0 + (SWT - Time_{exit}) - T_{night}$$

6 | Results

Many different scenarios have to be tested with the model as consequence of the many variables involved. A total of 15 were selected to show the influence of each variable. For all the scenarios the simulation time is 1 year. The first scenario is a simple typical operation of one SOV and then the variables will be modified in order to optimize the operation. The main characteristics of each modeled scenario is shown in Table 6.1. Table 6.2 presents the main outputs of the model.

Scenario's Main Characteristics:

- NFarms: Amount of served wind farms.
- NCTV: Amount of CTVs available to be deployed.
- HsCTV: Maximum significant wave height in which a CTV can operate [m].
- APers: Maximum available number of technical personnel groups.
- MinTasks_{winter}: Minimum amount of pending maintenance tasks during winter to start a shift.
- MinTasks_{summer}: Minimum amount of pending maintenance tasks during summer to start a shift.

Scenarios' Main Outputs:

- N_{Voy}: Number of voyages to the farm and back
- TPers: Maximum amount of transferred technical personnel groups
- DTasks: Amount of maintenance tasks performed
- Cost_{Op}: Total operational cost [EUR]

- $\text{Cost}_{\text{Down}}$: Loss due to wind turbines out of service [EUR]
- Cost_{Tec} : Cost of the technical personnel [EUR]
- $\text{Cost}_{\text{Total}}$: Total cost [EUR]
- WT_{base} : Total waiting time in base [hour]
- WT_{farm} : Total waiting time on site [hour]

Table 6.1: Modeled Scenarios Characteristics

Scenario No.	NFarms	NCTV	HsCTV	APers.	MinTasks _{winter}	MinTasks _{summer}
1	1	0	2	12	10	20
2	1	0	2	12	25	50
3	1	1	2	30	15	25
4	2	0	2	12	15	25
5	2	1	2	30	15	25
6	3	1	2	30	15	25
7	3	1	2	20	15	25
8	3	1	2	16	15	25
9	3	0	2	12	15	25
10	3	0	2	10	15	25
11	5	0	2	12	15	25
12	5	1	2	30	15	25
13	5	1	2	20	15	25
14	5	2	2	48	15	25
15	5	1	1,5	20	15	25

Figures 6.1 and 6.2 show the maintenance tasks generated and performed (NTasks) in the scenarios 1 and 2 respectively. The transferred technical personnel is shown as well. It can be noticed that the less the SOV waits in the shore base (until enough maintenance tasks are cumulated), the more voyages to the wind farms will be needed. Though, it did not affect the cost per transfer to a large extent. In addition to that, Scenario 2 ends up with far more pending duties than Scenario 1. In the other hand, when the variable MinTasks takes a high value,

Table 6.2: Modeled Scenarios Output

No.	N_{Voy}	TPers	DTasks	$Cost_{Op}$	$Cost_{Down}$	$Cost_{Tec}$	$Cost_{Total}$	WT_{base}	WT_{farm}		
1	28	12	482	4,816,000	167,000	796,000	11,991	5,529	1,375		
2	27	12	480	4,879,300	168,700	701,800	11,979	5,914	1,199		
3	25	30	480	4,915,900	162,100	1,725,000	14,173	5,961	1,267		
4	25	12	898	4,372,000	1,291,000	1,507,000	7,984	2,647	2,200		
5	25	30	898	4,476,000	1,294,000	3,658,000	10,499	2,826	2,153		
6	22	30	1449	4,144,000	1,744,000	4,922,000	7,460	776	2,749		
7	22	20	1449	4,119,000	1,745,000	3,339,000	6,351	634	2,786		
8	22	16	1449	4,112,000	1,744,000	2,686,000	5,895	591	2,824		
9	22	12	1449	3,997,000	1,744,000	2,037,000	5,368	500	2,871		
10	22	10	1417	3,982,000	1,744,000	1,715,000	5,251	415	2,880		
11	22	12	Not enough SOV service capacity								
12	12	30	2326	4,053,000	4,208,000	5,280,000	5,822	195	2,914		
13	12	20	2287	4,063,000	4,185,000	3,499,000	5,136	246	2,928		
14	12	48	2326	4,177,000	4,208,000	8,396,000	7,215	247	2,886		
15	12	48	2099	4,006,000	4,185,000	8,191,000	5,580	189	2,944		

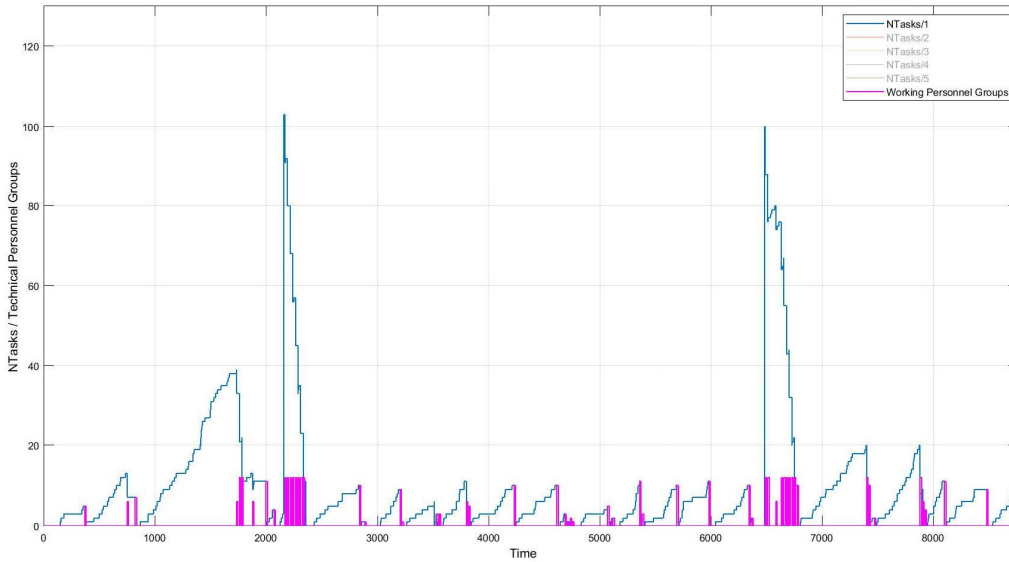


Figure 6.1: Maintenance Tasks and Transferred Technical Personnel in Scenario 1

it is no longer the one that sets the time in which the SOV starts a shift, it will be DownTurbines, that is the one that indicates when a turbine is out of service and then the SOV must set sail. That can be seen by comparing Figures 6.2 and

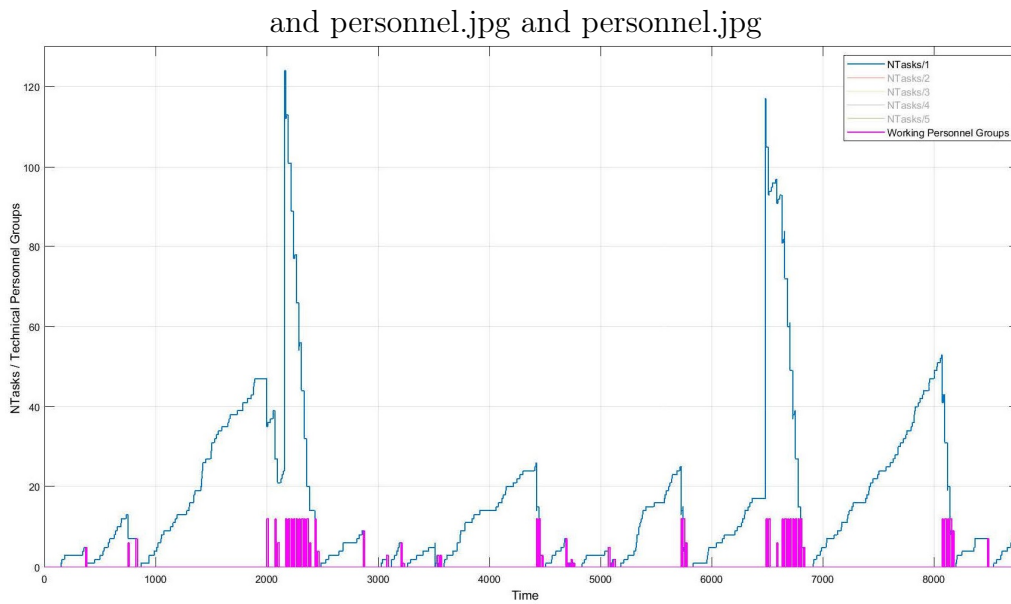


Figure 6.2: Maintenance Tasks and Transferred Technical Personnel in Scenario 2

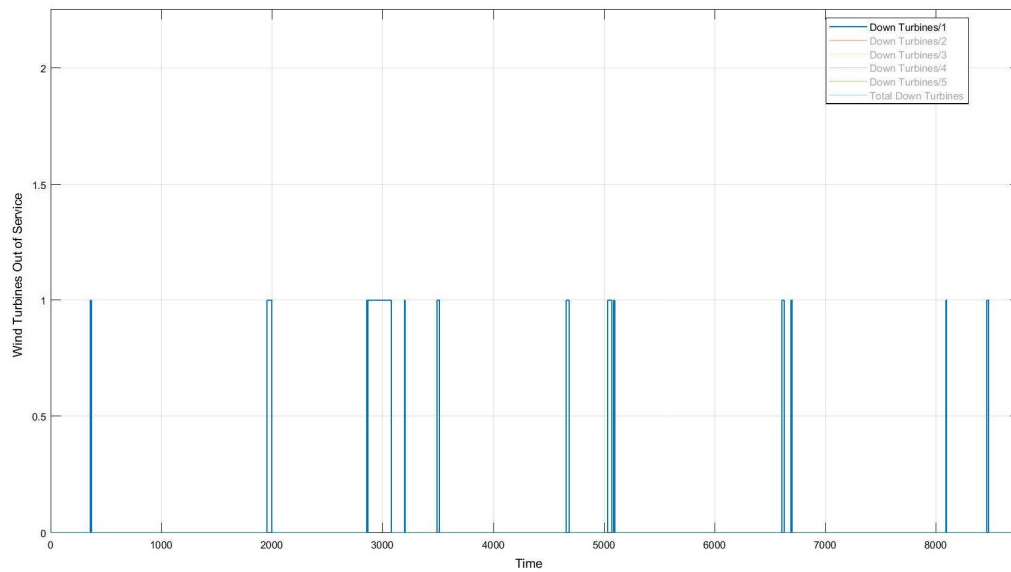


Figure 6.3: Wind Turbines Out of Service in Scenario 2

???. Therefore an intermediate value of the variable MinTasks seems to be more convenient.

In Scenario 3 a CTV is added to the operation in connexion to the SOV (also MinTasks is lower). When comparing the performance of the SOV+CTV configuration (Figure 6.4 against Figures 6.1 and 6.2), it is clear that the maintenance duties are performed faster and more technical personnel can be transferred, but at

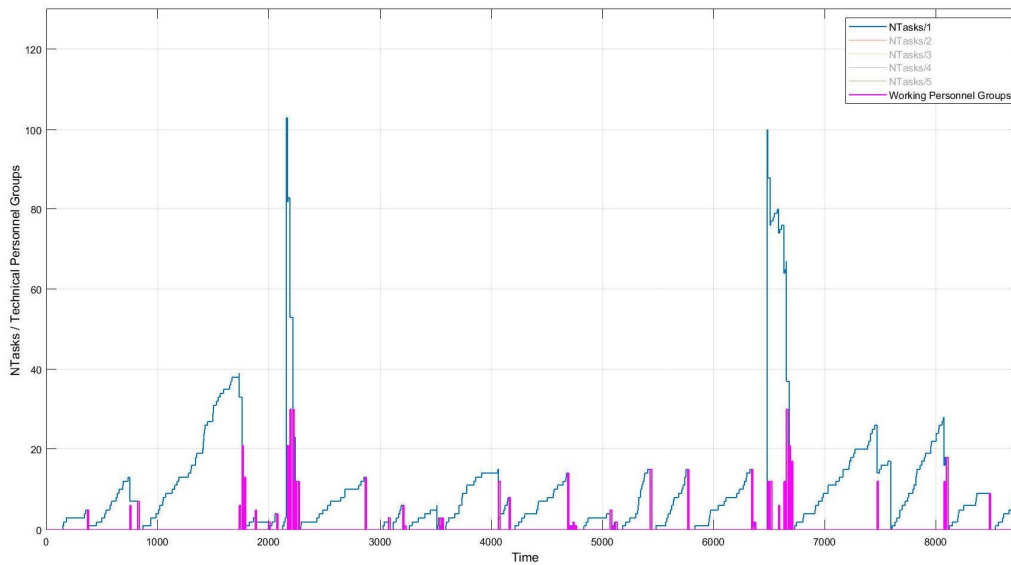


Figure 6.4: Maintenance Tasks and Transferred Technical Personnel in Scenario 3

a much higher cost (around 18%). That is because the amount of maintenance tasks performed is the same but the operational cost rises. Another improvement would be limiting the maximum available technical personnel and distributing it along the year.

Other important fact to be noticed is that more than 60% of the time the SOV is waiting in the base, therefore servicing more wind farms in the same shift would cut down the costs, as they would be distributed among them.

The improvements proposed in the previous paragraph for Scenario 3 were implemented for scenario 4. From Figure 6.5 it can be concluded that only the SOV (without CTV) is able to attend two wind farms. Therefore less technical personnel is able to perform double maintenance tasks, and its work is better distributed along the year. This leads to less waiting times in base (less than 50%), therefore lower costs, to an extent that the cost per transfer drops 34% with respect to Scenarios 1 and 2 and up to 43% compared to Scenario 3.

In scenario 5, in the same way as in Scenario 3, one CTV has been added to the operation, with the same effects, higher transfer rate (as shown in Figure 6.6) but at a much higher cost (32% higher cost per transfer). That shows the importance of the amount of technical personnel to embark and how to distribute the maintenance tasks along the year.

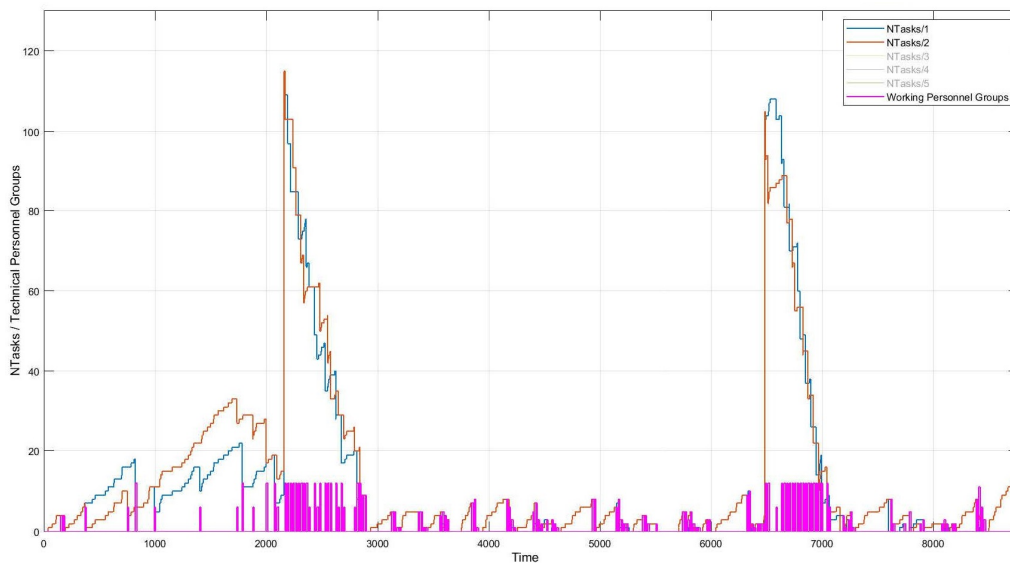


Figure 6.5: Maintenance Tasks and Transferred Technical Personnel in Scenario 4

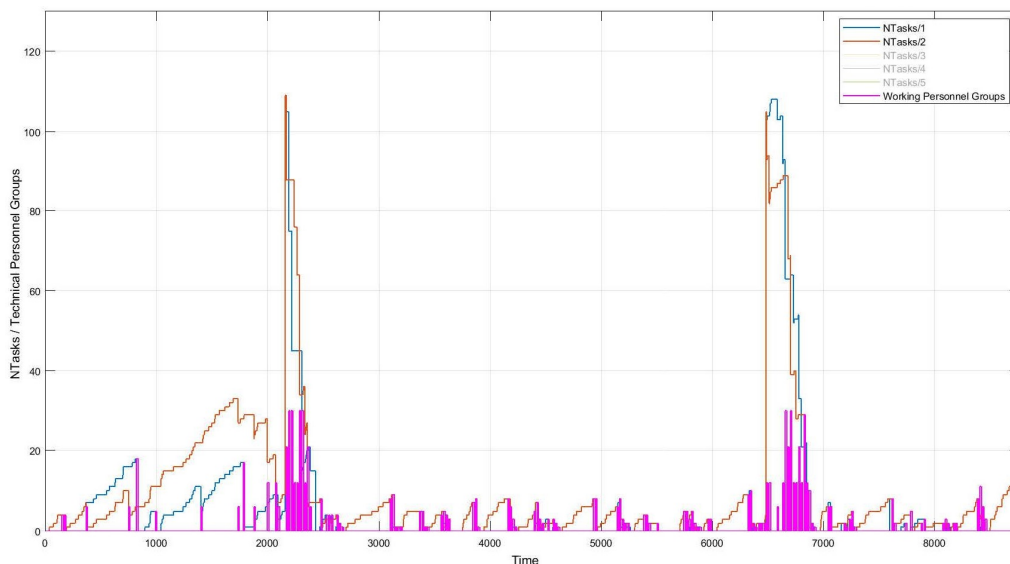


Figure 6.6: Maintenance Tasks and Transferred Technical Personnel in Scenario 5

In Scenarios 6, 7, 8, 9 and 10, a third wind farm is also serviced with different configurations: only SOV operating (scenario 9) and SOV+CTV with different amounts of technical personnel available. When analysing the results in Table 6.2 and Fig 6.7 it is noticed again that as the amount of embarked technical personnel drops and gets evenly distributed along the season, the cost per transfer does so. After a certain limit of available work force (Scenario 10 with 10 technical personnel crews), even though the cost per transfer drops, the efficiency of the operation drops too, since the amount of performed maintenance tasks falls.

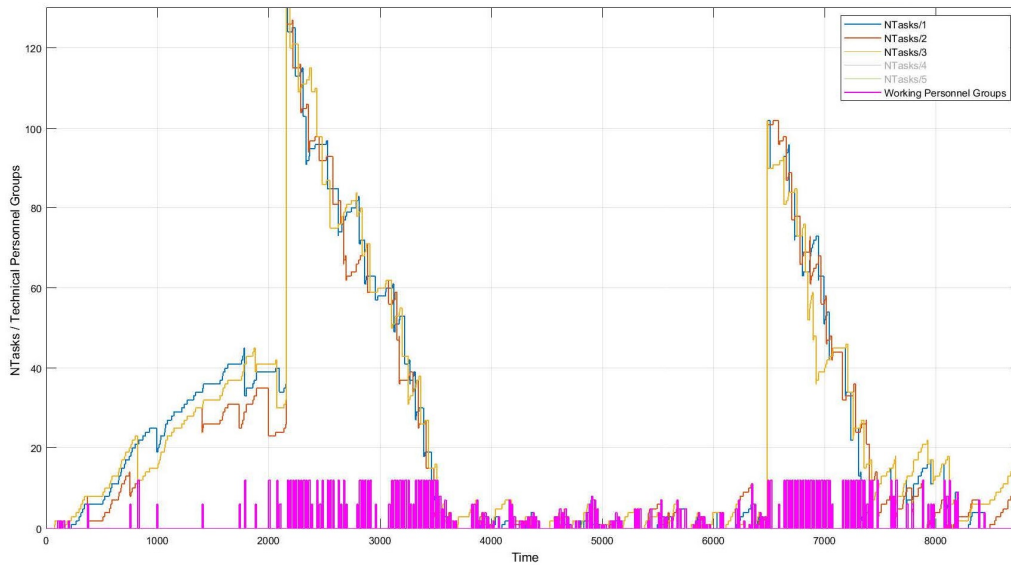


Figure 6.7: Maintenance Tasks and Transferred Technical Personnel in Scenario 9

In Scenarios 11, 12, 13, 14 and 15, 5 wind farms are being serviced, again with different SOV+CTV configurations. When servicing with only the SOV (Scenario 11) she did not have enough capacity to attend all the maintenance tasks and then they ended up cumulating. Then for Scenarios 12 and 13, as seen in Table 6.2, when less technical personnel is embarked the cost of per transfer drops (mainly driven by the save on technical personnel expenses) with no significant change in the operation efficiency (ammount of DTasks).

In Scenario 14 a second CTV is added, which gives a much higher transfer capacity to the operation at a reasonable expense (operational costs rises too little). This overcapacity could be used for attending even more (than 5) wind farms.

Finally in scenario 15, the operation was performed by SOV+CTV, but with a maximum significant wave height allowance of 1,5m for the CTV. It can be noticed the efficiency drop in the transfer rate, due to the fact that the CTV hardly ever can operate due to rough sea state (Figure 6.9). It is shown in Figure 6.8, that not many times, and only in summer time, the amount of technical personnel transferred is more than 12 (SOV capacity).

Another output to be analyzed is the sailing time from shore to the farms and back. As shown in Figure 6.10, the total difference between the maximum and minimum values is not relevant for the scenarios analyzed, since it is less than 2 hours. Nevertheless it is a difference of around 15%, which would be a value to

consider for sea-going ships.

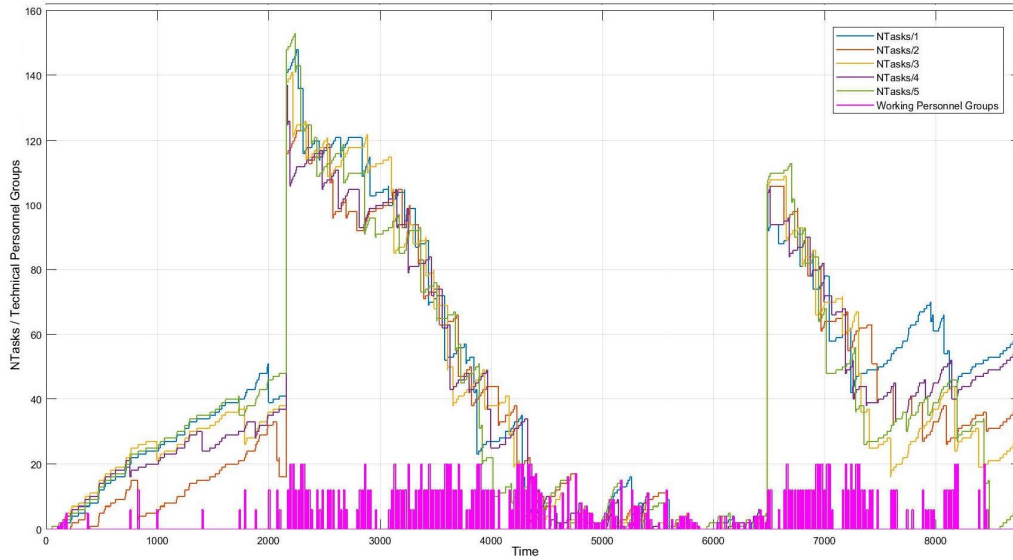


Figure 6.8: Maintenance Tasks and Transferred Technical Personnel in Scenario 15

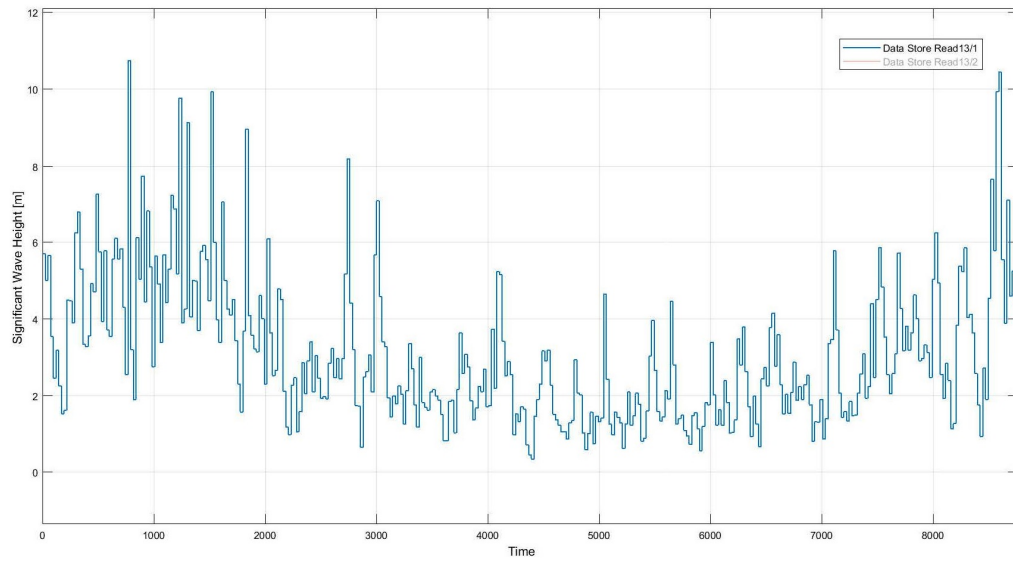


Figure 6.9: Significant Wave Height

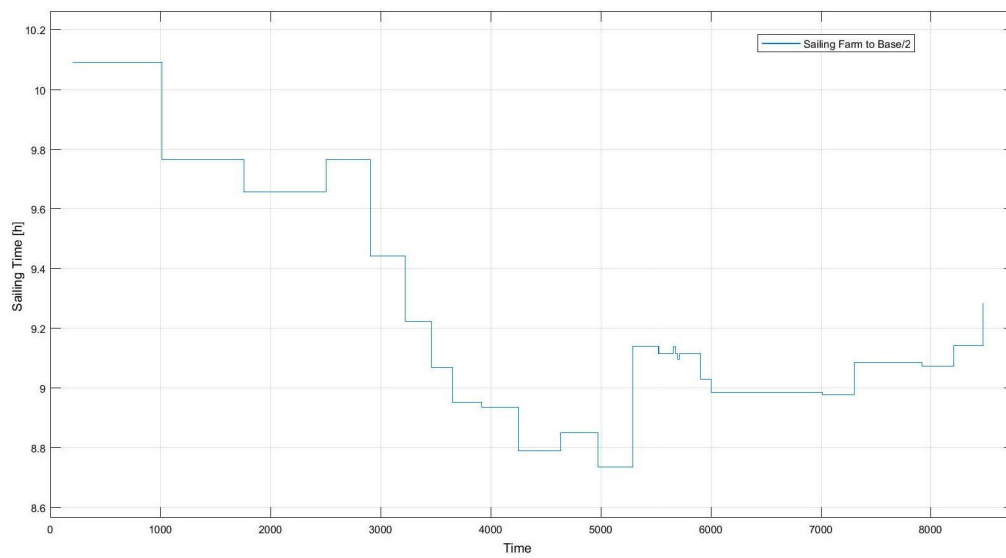


Figure 6.10: Sailing Times

(PAGE INTENTIONALLY LEFT BLANK)

7 | Discussion and Conclusions

The analysis of the results in Chapter 6 made it possible to assess the influence, under certain assumptions, of different variables in the operation of a SOV when servicing wind farms for O&M duties.

The main capability of the vessel is the transfer capacity of technical personnel to and from the wind turbines. The variables that influenced it most were: speed and transfer means (combined they give the hourly transfer rate), available CTVs, accommodation capacity (that means, technical personnel available) and sea-going conditions (maximum significant wave height in which she can operate safely)

The maximum allowed significant wave height was the most influential factor when it came to the number of transfers performed, as it results in the theoretical operability of the vessel. This highlights the importance of coupling the vessel design to the operational area (eg. sea state), and developing vessel designs that are less sensitive to sea state conditions by means of, for example, heave-compensating systems or alternative hull arrangements.

During the winter season, in the selected area, the wave height hardly ever would allow the CTVs to operate due to its maximum allowable significant wave height. That makes the PM tasks planning relevant for the operation outcome, that is, planning to perform PM when the sea state is favourable.

Even though the waiting times in the shore base could be lowered, the situation is different when the SOV is offshore, since she will have to wait for favourable sea conditions (unavoidable, only by rescheduling). The night times can be used for sailing from one wind farm to another in order to make good use of it.

The technical personnel availability is also relevant to define the transfer capac-

ity, these two variables should match as much as possible in order to avoid unused capacity and personnel waiting times on board that would lead to high costs. It could be noticed that when distributing the work force of the available personnel throughout the year, a more cost efficient use of this resource is made. It can also define a SOV with less accomodation capacity (cheaper) but with the same efficiency.

Another point to be noticed is that less wind turbines out of service did not always mean, depending on the scenario, higher total costs, since the daily costs of the SOV is comparable to the loss generated by the down times and they compensate one another.

When the sea state makes it possible to operate with CTVs, a considerable rise in the transfer rate capacity is experienced. In the analysed scenarios it lead to a transfer rate surplus, that could make, in certain areas, to place the SOV close to the wind farms site and attending them by only deploying CTVs.

The vessel's speed did not influence the results to a large extent, since the sailing times between the base and the wind farms is short. It does not influenc the amount of transfers per hour, since the SOV has to sail from one turbine to another at low speed. This last effect has not been studied and might be the objective of a further work.

8 | Further Work

In the present work some aspects of the operation of the SOV were not considered or modelled in detail with the objective of make the model simple and fast. Other point was the lack of available data, as the offshore wind industry is currently in an “early” stage of development.

Possible improvements to make the model more realistic and detailed could be:

- Model the operation of 1 SOV serving several wind farms at the same time by only deploying CTVs from a position nearby.
- Model the routing inside each farm.
- Model the different failure types and occurrence with more detailed data. For example making use of the one given by [8]
- Even though helicopters are used to operate in wind farms located close to the coast, it can be analysed when it is cost effective to embark a helicopter to a SOV when there are many wind turbines out of service and when rough sea conditions are expected for a long time.

(PAGE INTENTIONALLY LEFT BLANK)

Bibliography

- [1] D.C. Salzmann. “Amplemann – The development of an offshore access system”. In: *Presentation at We@Sea Conference* (Dec. 2009).
- [2] C.K. Tveiten, E. Albrechtsen, J. Heggset, M. Hofmann, E. Jersin, B. Leira, and P.K. Norddal. *HSE challenges related to offshore renewable energy*. SINTEF Technology & Society, Trondheim, 2011.
- [3] M. Stålhane, L. Hvattuma, and V. Skaaret. “Optimization of Routing and Scheduling of Vessels to Perform Maintenance at Offshore Wind Farms”. In: *Energy Procedia* 80 (2015), pp. 92–99.
- [4] 4C Offshore Ltd. *Sørlige Nordsjø (Category A area) Offshore Wind Farm*. URL: [https://www.4coffshore.com/windfarms/project-dates-for-s%5C%C3%5C%B8rlige-nordsj%5C%C3%5C%B8-i-\(category-a-area\)-no39.html](https://www.4coffshore.com/windfarms/project-dates-for-s%5C%C3%5C%B8rlige-nordsj%5C%C3%5C%B8-i-(category-a-area)-no39.html). (accessed: 15.03.2018).
- [5] L. Dai, M. Stålhane and I.B Utne. “Routing and Scheduling of Maintenance Fleet for Offshore Wind Farms”. In: *Wind Engineering* 39.1 (2015), pp. 15–30.
- [6] P. Zhang, L. Dai, Y. Liu. “Modelling Operation of Service Vessels in Offshore Wind Farms Using Stochastic Activity Networks”. In: *Ship Technology Research* 61.1 (2014), pp. 48–58.
- [7] S. Faulstich, B. Hahn and P.J. Tavner. “Wind turbine downtime and its importance for offshore deployment”. In: *Wind Energy* 14 (2011), pp. 327–337.
- [8] C.I. Ossai, B. Boswell, I. Davies. “A Markovian approach for modelling the effects of maintenance ondowntime and failure risk of wind turbine components”. In: *Renewable Energy* 96 (2016), pp. 776–783.

-
- [9] R. Bi, K. Qian, C. Zhou, D. Hepburn, J. Rong. “A survey of failures in wind turbine generator systems with focus on a wind farm in China”. In: *International Journal of Smart Grid and Clean Energy* (July 2014).
- [10] F. Besnard, K. Fischer and L. Bertling. “A Model for the Optimization of the Maintenance Support Organization for Offshore Wind Farms”. In: *IEEE Transactions on Sustainable Energy* 4.2 (2013).
- [11] I. Dinwoodie, O.E. Endrerud, M. Hofmann, R. Martin, and I. Bakken Sperstad. “Reference Cases for Verification of Operation and Maintenance Simulation Models for Offshore Wind Farms”. In: *Wind Engineering* 39.1 (2015), pp. 1–14.
- [12] J.L. Martin. *How classic Systems Engineering process must change to be effective for use in Ship Design*. Oct. 2013.
- [13] I. Fonseca, H.M. Gaspar. *An Object-Oriented Approach for Virtual Prototyping in Conceptual Ship Design*. Oct. 2013.
- [14] T. Van Breuinessen, H. Hopman, F. Smulders. *Describing Ship Design, Modeling the Innovative Character(s) of Ship Design Processes*.
- [15] The MathWorks Inc. *Simulink*. URL: <https://se.mathworks.com/products/simulink.html>. (accessed: 15.03.2018).
- [16] The MathWorks Inc. *SimEvents*. URL: <https://se.mathworks.com/products/simevents.html>. (accessed: 15.03.2018).
- [17] *Simulating Ocean Engineering Systems - A Compendium*. 2016-09-28.
- [18] EY. *Offshore Wind in Europe, Walking the Tightrope to Success*. Mar. 2015.

A | Implementation of Markov Chain for Simulating Sea States

Steps:

- i. Select the data from the available. The vector “Hs” is generated with the wave height values.

```
1  % Reads the csv file into a matrix
2  A = csvread('RawDataCSV.csv',23,0);
3  % Loop over all the data
4  for i = 1:size(A,1)
5      % Set first and last month. Can be the same
6      first = 1;
7      last = 12;
8      % Check if the month is in the range wanted
9      if (A(i,2) >= first) && (A(i,2) <= last)
10         if A(i,7) > 0 % Disregard zero and negative
11             values
12                 if (exist('Hs') == 0) % Hs doesn't exist
13                     Hs = A(i,7);
14                 else
15                     % Add the value at the end of Hs
16                     Hs = [ Hs; A(i,7) ];
17                 end
18             end
19         end
20     end
```

-
- ii. Once the Hs vector is generated, we build a transition matrix that show the probability changing from one state to another by simply counting the transitions in the vector HS and normalizing the values so that the sum of all the components on each row of the transition matrix is =1. Then by generating rando

```
1  % Set number of states in the markov chain
2  % Beware of setting this too high. If there are too
   many states, some of the states will be absorbing,
   that is,  $P(j,j) = 1$ , which means it can never
   transition to other states.
3  %
4  numStates = 12;
5
6  % Find upper limit for Hs values and divide the
   values into even bins
7  ul = max(Hs);
8  % Find state ranges - first state [0,stateRange] and
   so on
9  stateRange = ul / numStates;
10 % State values - stateRange, 2xstateRange and so on
   up til ul
11 stateValues = stateRange:stateRange:ul;
12 % Initialize 1D-matrix holding the state of each data
   point
13 HsState = zeros(length(Hs),1);
14
15 % Find each data points state
16 for i = 1:length(Hs)
17     % For each data point
18     for j = 1:numStates
19         % For each state
20         if Hs(i) <= stateValues(j)
21             % Data point is in state j
22             HsState(i) = j;
23             % This data point is categorized, so we
               break and move to the
```

```

24             % next data point
25             break;
26         end
27     end
28 end
29
30 % Find transitions
31 transitions = zeros(numStates);
32 for t = 1:length(HsState)-1
33     % HsState(t) represents the state and HsState(t
34         % +1) represents the state
35     % it transitions to
36     transitions(HsState(t),HsState(t+1)) =
37         transitions(HsState(t),HsState(t+1)) + 1;
38 end
39
40 P = transitions;
41 % Normalize each row in the transition matrix so each
42     % row sums to 1
43 for i = 1:numStates
44     P(i,:) = P(i,:) / sum( P(i,:) );
45 end
46
47 % Check to see if there are any absorbing states
48 % i.e.  $P(i,j) == 1$  where  $i=j$ 
49 absorbstate = zeros(numStates);
50 for i = 1:numStates
51     for j = 1:numStates
52         if P(i,j) == 1
53             absorbstate(i,j) = absorbstate(i,j) + 1;
54         end
55     end
56 end
57 if sum(sum(absorbstate)) >= 1
58     error('Absorbing states. Stopping. Consider
59         reducing number of states or check data.');
```

```

56 end
57
58 xlswrite('MCSS.xlsx',P) %writes the transition matrix
    P into the Excel file MCSS.xlsx

```

- iii. Then an entity generator is created in which each entity created each step of the simulation will generate a change in the sea state based on the probabilities of the transition matrix, and starting from a random seastate

```

1 persistent SeaState MCSeaStates t
2 coder.extrinsic('xlsread');
3 if isempty(SeaState)
4     MCSeaStates=zeros(12,12);
5     MCSeaStates=xlsread('MCSS.xlsx');
6     SeaState=randi(12);
7     t=1;
8     WriteSeaState(SeaState);
9 end
10 if t==3
11     r = rand();
12     for j = 1:length(MCSeaStates(:,1))
13         prob = 0;
14         for k = 1:j
15             prob = prob + MCSeaStates(SeaState,k);
16         end
17         if r <= prob
18             SeaState = j;
19             WriteSeaState(SeaState*1.0508);%
                StateRange
20             break;
21         end
22     end
23     t=0;
24 else
25     t=t+1;
26 end

```

iv. Finally the effect of the sea state is accounted on the speed of the vessel:

```
1 \end{ttfamily}
2 SeaState=ReadSeaState();
3 %How does wave height affect sailing time?
4 dt = (distance/entity.speed)*(1+SeaState/10);
```

(PAGE INTENTIONALLY LEFT BLANK)

B | MATLAB Scripts

B.1 SOV Entity Generator's Script

```
1 persistent count igt
2 if isempty(count)
3     igt=[0 inf];
4     count=1;
5 end
6 dt = igt(count);
7 count=count+1;
```

B.2 Base Subsystem's Scripts

B.2.1 Base Entity Server's Script

Entry

```
1 WriteSOVinBase(1);
```

Service Complete

```
1 %Adds +1 everytime the SOV stays in the base
2 WaitingTimeB=ReadWaitingTimeB();
3 WaitingTimeB=WaitingTimeB+1;
4 WriteWaitingTimeB(WaitingTimeB);
```

```
5
6  %Routing (loop for waiting time or sending entity to farm
   )
7  TotalTasks=sum(ReadNTasks()); %Total pending tasks
8  DownTurbines=sum(ReadDownTurbines); %Total turbines down
9  if TotalTasks>=ReadMinTasks() || DownTurbines>0
10     entity.home=1;
11 else
12     entity.home=2;
13 end
```

B.2.2 Base Exit Entity Server's Script

```
1  WriteShiftStart(ReadTime()); %Set the starting time of
   the shift
2  WriteSOVinBase(0);
```

B.3 Sailing Entity Servers' Scripts

```
1  distance = ReadSailingDistance(); %nm
2  dt = (distance/entity.speed)*(1+ReadSeaState()/10);
```

B.4 Wind Farms Subsystem's Scripts

B.4.1 Waiting Place Entity Server's Script

Entry

```
1  Time=ReadTime();
2  SetWaitingTimeF=Time;
3  WriteSetWaitingTimeF(SetWaitingTimeF);
```

Service Complete

```
1 ShiftDuration=ReadShiftDuration();
2 NTasks=ReadNTasks();
3 NTasks=sum(NTasks);
4 if ReadShiftDays()>ShiftDuration || NTasks==0
5     entity.home=1;
6 else
7     entity.home=2;
8 end
```

Exit

```
1 Time=ReadTime();
2 SetWaitingTimeF=ReadSetWaitingTimeF();
3 WaitingTimeF=ReadWaitingTimeF();
4
5 %Night time, to be subtracted form the waiting time in
6 farms.
7 NightTime1=0 %First day & variable initialization
8 NightTime1=mod(SetWaitingTimeF,24);
9 if NightTime1<6
10     NightTime1=NightTime1-12;
11 elseif NightTime1<18
12     NightTime1=6;
13 else
14     NightTime1=24-NightTime1;
15 end
16 NightTime2=0 %Last day & variable initialization
17 NightTime2=mod((Time-SetWaitingTimeF),24);
18 if (Time-SetWaitingTimeF)<24 %means the first day is not
19 the last
20     NightTime2=0;
21 else
22     if NightTime2<6
23         NightTime2=NightTime2;
24     elseif NightTime2<18
```

```

24         NightTime2=6;
25     else
26         NightTime2=NightTime2 -12;
27     end
28 end
29
30 NightTime3=0    %Days in between and variable
                initialization
31 %days in between
32 NightTime3=((Time -mod(Time ,24)) - (SetWaitingTimeF+(24 -mod(
                SetWaitingTimeF ,24))))*(12/24);
33 NightTime=NightTime1+NightTime2+NightTime3;
34 WaitingTimeF=WaitingTimeF+(Time -SetWaitingTimeF) -
                NightTime;
35 WriteWaitingTimeF(WaitingTimeF);

```

B.4.2 Script for Weather and Sea State Gate MATLAB Function

```

1  function y = fcn
2  t=ReadTime();
3  x = mod(t,24); %b = mod( a , m ) returns the remainder
                after division of a by m, thus, x is the time of the
                current day
4  WaveForecast=ReadWaveForecast();
5  w1=WaveForecast(1);
6  w2=WaveForecast(2);
7  HsSOV=ReadHsSOV;
8  HsCTV=ReadHsCTV;
9  if x >= 6 && (x+6) <=18 && w1 <=HsSOV
10     y=1;
11 elseif x >= 6 && (x+6) <=18 && w2 <=HsSOV
12     y=1;
13 else
14     y=0;
15 end

```

B.4.3 Farm Selector Entity Server's Script

Service Complete

```
1  NFarms=ReadNFarms(); %Amount of farms to be served
2  n=NFarms; %initialize variable for the "for" command
3  DownTurbinesExt=zeros(5,1); %variable initialization
4  NTasksExt=zeros(5,1); %variable initialization
5  DownTurbines=ReadDownTurbines();
6  NTasks=ReadNTasks();
7
8  %Create strings for only serve deisred farms and not the
   others
9  %Makes zero the components of the strings corresponding
   to the farms taht will not be served
10 for x=1:n
11     DownTurbinesExt(x)=DownTurbines(x);
12     NTasksExt(x)=NTasks(x);
13 end
14 DownTurbines=DownTurbinesExt;
15 NTasks=NTasksExt;
16
17 MaxDown=find(DownTurbines==max(DownTurbines)); %Finds the
   farm with maximum amount of turbines out of service
18 MaxNTasks=find(NTasks==max(NTasks)); %Finds the farm with
   maximum amount of turbines out of service
19 MaxDown=MaxDown(1); %If two equal components, picks the
   first
20 MaxNTasks=MaxNTasks(1); %If two equal components, picks
   the first
21 if max(DownTurbines)>0 %at least one farm with one
   turbine out of service, sends SOV to the farm with max
   number of turbines out of service
22     entity.home=MaxDown;
23     DownTurbines(MaxDown)=0; %considers that the turbine/
   s will be on service after maintenance
```

```
24 else    %No turbines down, SOV serves farm with more
          pending tasks
25     entity.home=MaxNTasks;
26 end
27 WriteDownTurbines(DownTurbines)
```

Exit

```
1 WriteSOVinFarm(1);
```

B.4.4 Farms Entity Servers' Scripts

The attached script corresponds to the “Farm1” entity server, the scripts for the other Farm entity servers are the same, with the only difference of the value of the variable “FarmNumber”, which identifies the wind farm number.

Entry

```
1 FarmNumber=1;
2 %Dtasks is the number of tasks performed
3 %Assuming no time to set out and pick up CTV.
4 NTasks = ReadNTasks()
5     NTasks=NTasks(FarmNumber);
6 DTasks = ReadDTasks();
7 WaveForecast=ReadWaveForecast();
8 w1=WaveForecast(1);%wave for time between 6 and 12
9 w2=WaveForecast(2);%wave for time between 12 and 18
10 HsSOV=ReadHsSOV(); %Allowable operation waveheight for
    SOV
11 HsCTV=ReadHsCTV(); %Allowable operation waveheight for
    CTV
12 TSOV=ReadTSOV(); %Amount of transfers one SOV can do in
    one hour
13 TCTV=ReadTCTV(); %Amount of transfers one CTV can do in
    one hour
14 NCTV=ReadNCTV(); %Amount of CTVs available for operation
```

```

15 MaxTechnicians=ReadMaxTechnicians(); %Max amount of
    technical personnel available
16
17 if NTasks==0
18     entity.home=1;
19 else
20 if w1>HsSOV %No work between 6:00 and 12:00
21     if w2<=HsSOV && w2>HsCTV %Only SOV works from 12:00
        to 18:00
22         if NTasks<3*TSOV %Tasks remaining less than the
            maximum transfers that can be performed
23             if NTasks<=MaxTechnicians %Personnel
                available can perform all tasks this day
24                 WriteWorkingCrews(NTasks);
25                 DTasks=DTasks+NTasks;
26                 NTasks=0;
27             else
28                 WriteWorkingCrews(MaxTechnicians);
29                 DTasks=DTasks+MaxTechnicians;
30                 NTasks=NTasks - MaxTechnicians;
31             end
32         else
33             if 3*TSOV<=MaxTechnicians %Personnel
                available can perform all tasks this day
34                 NTasks=NTasks - 3*TSOV; %Amount of tasks
                    that can be done (deliveries).
35                 DTasks=DTasks+3*TSOV;
36                 WriteWorkingCrews(3*TSOV);
37             else
38                 NTasks=NTasks - MaxTechnicians; %Amount of
                    tasks that can be done (deliveries).
39                 DTasks=DTasks+MaxTechnicians;
40                 WriteWorkingCrews(MaxTechnicians);
41             end
42         end
    end
end

```

```

43     elseif w2<=HsCTV %SOV and CTV work from 12:00 to
44         18:00
45         if NTasks<(3*TSOV+3*TCTV*NCTV) %Tasks remaining
46             less than the maximum that can be performed
47             if NTasks<=MaxTechnicians
48                 WriteWorkingCrews(NTasks);
49                 DTasks=DTasks+NTasks;
50                 NTasks=0;
51             else
52                 WriteWorkingCrews(MaxTechnicians);
53                 DTasks=DTasks+MaxTechnicians;
54                 NTasks=NTasks - MaxTechnicians;
55             end
56         else
57             if (3*TSOV+3*TCTV*NCTV)<=MaxTechnicians
58                 NTasks=NTasks - (3*TSOV+3*TCTV*NCTV); %
59                 Amount of tasks that can be done (
60                 deliveries).
61                 DTasks=DTasks+(3*TSOV+3*TCTV*NCTV);
62                 WriteWorkingCrews(3*TSOV+3*TCTV*NCTV);
63             else
64                 NTasks=NTasks - MaxTechnicians; %Amount of
65                 tasks that can be done (deliveries).
66                 DTasks=DTasks+MaxTechnicians;
67                 WriteWorkingCrews(MaxTechnicians);
68             end
69         end
70     end
71 elseif w1<=HsSOV && w1>HsCTV %Only SOV can work between
72     6:00 and 12:00
73     if w2>HsSOV %SOV and CTVs can not work between 12:00
74         and 18:00
75         if NTasks<3*TSOV %Tasks remaining less than the
76             maximum that can be performed
77             if NTasks<=MaxTechnicians
78                 WriteWorkingCrews(NTasks);

```

```

71         DTasks=DTasks+NTasks;
72         NTasks=0;
73     else
74         WriteWorkingCrews(MaxTechnicians);
75         DTasks=DTasks+MaxTechnicians;
76         NTasks=NTasks - MaxTechnicians;
77     end
78 else
79     if (3*TSOV)<=MaxTechnicians
80         NTasks=NTasks - 3*TSOV; %Amount of tasks
81             that can be done (deliveries).
82         DTasks=DTasks+3*TSOV;
83         WriteWorkingCrews(3*TSOV);
84     else
85         NTasks=NTasks - MaxTechnicians; %Amount of
86             tasks that can be done (deliveries).
87         DTasks=DTasks+MaxTechnicians;
88         WriteWorkingCrews(MaxTechnicians);
89     end
90 elseif w2<=HsSOV && w2>HsCTV %Only SOV works between
91     12:00 and 18:00
92     if NTasks<6*TSOV %Tasks remaining less than the
93         maximum that can be performed
94         if NTasks<=MaxTechnicians
95             WriteWorkingCrews(NTasks);
96             DTasks=DTasks+NTasks;
97             NTasks=0;
98         else
99             WriteWorkingCrews(MaxTechnicians);
100            DTasks=DTasks+MaxTechnicians;
101            NTasks=NTasks - MaxTechnicians;
102        end
103    else
104        if (6*TSOV)<=MaxTechnicians

```

```

102         NTasks=NTasks-6*TSOV; %Amount of tasks
           that can be done (deliveries).
103         DTasks=DTasks+6*TSOV;
104         WriteWorkingCrews(6*TSOV);
105     else
106         NTasks=NTasks-MaxTechnicians; %Amount of
           tasks that can be done (deliveries).
107         DTasks=DTasks+MaxTechnicians;
108         WriteWorkingCrews(MaxTechnicians);
109     end
110
111     end
112 elseif w2<=HsCTV %SOV and CTVs work from 12:00 to
           18:00
113     if NTasks<(6*TSOV+3*TCTV*NCTV) %Tasks remaining
           less than the maximum that can be performed
114         if NTasks<=MaxTechnicians
115             DTasks=DTasks+NTasks;
116             NTasks=0;
117             WriteWorkingCrews(NTasks);
118         else
119             DTasks=DTasks+MaxTechnicians;
120             NTasks=NTasks-MaxTechnicians;
121             WriteWorkingCrews(MaxTechnicians);
122         end
123     else
124         if (6*TSOV+3*TCTV*NCTV)<=MaxTechnicians
125             NTasks=NTasks-(6*TSOV+3*TCTV*NCTV); %
           Amount of tasks that can be done (
           deliveries).
126             DTasks=DTasks+(6*TSOV+3*TCTV*NCTV);
127             WriteWorkingCrews(6*TSOV+3*TCTV*NCTV);
128         else
129             NTasks=NTasks-MaxTechnicians; %Amount of
           tasks that can be done (deliveries).
130             DTasks=DTasks+MaxTechnicians;

```

```

131             WriteWorkingCrews(MaxTechnicians);
132         end
133     end
134 end
135 elseif w1<=HsCTV %SOV and CTVs can work from 6:00 to
        12:00
136     if w2>HsSOV %SOV and CTV can not work between 12:00
        and 18:00
137         if NTasks<(3*TSOV+3*TCTV*NCTV) %Tasks remaining
        less than the maximum that can be performed
138             if NTasks<=MaxTechnicians
139                 WriteWorkingCrews(NTasks);
140                 DTasks=DTasks+NTasks;
141                 NTasks=0;
142             else
143                 WriteWorkingCrews(MaxTechnicians);
144                 DTasks=DTasks+MaxTechnicians;
145                 NTasks=NTasks - MaxTechnicians;
146             end
147         else
148             if (3*TSOV+3*TCTV*NCTV)<=MaxTechnicians
149                 NTasks=NTasks - (3*TSOV+3*TCTV*NCTV); %
        Amount of tasks that can be done (
        deliveries).
150                 DTasks=DTasks+(3*TSOV+3*TCTV*NCTV);
151                 WriteWorkingCrews(3*TSOV+3*TCTV*NCTV);
152             else
153                 NTasks=NTasks - (MaxTechnicians); %Amount
        of tasks that can be done (deliveries)
        .
154                 DTasks=DTasks+(MaxTechnicians);
155                 WriteWorkingCrews(MaxTechnicians);
156             end
157         end
158     elseif w2<=HsSOV && w2>HsCTV %Only SOV works between
        12:00 and 18:00

```

```

159         if NTasks < (6*TSOV+3*TCTV*NCTV) %Tasks remaining
160             less than the maximum that can be performed
161             if NTasks <= MaxTechnicians
162                 WriteWorkingCrews(NTasks);
163                 DTasks=DTasks+NTasks;
164                 NTasks=0;
165             else
166                 WriteWorkingCrews(MaxTechnicians);
167                 DTasks=DTasks+MaxTechnicians;
168                 NTasks=NTasks - MaxTechnicians;
169             end
170         else
171             if (6*TSOV+3*TCTV*NCTV) <= MaxTechnicians
172                 NTasks=NTasks - (6*TSOV+3*TCTV*NCTV); %
173                 Amount of tasks that can be done (
174                 deliveries).
175                 DTasks=DTasks+(6*TSOV+3*TCTV*NCTV);
176                 WriteWorkingCrews(6*TSOV+3*TCTV*NCTV);
177             else
178                 NTasks=NTasks - MaxTechnicians; %Amount of
179                 tasks that can be done (deliveries).
180                 DTasks=DTasks+MaxTechnicians;
181                 WriteWorkingCrews(MaxTechnicians);
182             end
183         end
184     elseif w2 <= HsCTV %SOV and CTVs work from 12:00 to
185         18:00
186         if NTasks < (6*TSOV+6*TCTV*NCTV) %Tasks remaining
187             less than the maximum that can be performed
188             if NTasks <= MaxTechnicians
189                 WriteWorkingCrews(NTasks);
190                 DTasks=DTasks+NTasks;
191                 NTasks=0;
192             else
193                 WriteWorkingCrews(MaxTechnicians);
194                 DTasks=DTasks+MaxTechnicians;

```

```

189             NTasks=NTasks - MaxTechnicians;
190         end
191     else
192         if (6*TSOV+6*TCTV*NCTV)<=MaxTechnicians
193             NTasks=NTasks - (6*TSOV+6*TCTV*NCTV); %
                Amount of tasks that can be done (
                deliveries).
194             DTasks=DTasks+(6*TSOV+6*TCTV*NCTV);
195             WriteWorkingCrews(6*TSOV+6*TCTV*NCTV);
196         else
197             NTasks=NTasks - (MaxTechnicians); %Amount
                of tasks that can be done (deliveries)
                .
198             DTasks=DTasks+(MaxTechnicians);
199             WriteWorkingCrews(MaxTechnicians);
200         end
201     end
202 end
203 end
204 end
205
206 %writing NTaks component A: intermediate variable to save
    value
207 A=NTasks;
208 NTasks = ReadNTasks();
209 NTasks(FarmNumber)=A;
210 WriteNTasks(NTasks);
211
212 WriteDTasks(DTasks);
213
214 entity.home=2;

```

Exit

```

1 WriteWorkingCrews(0);
2 WriteSOVinFarm(0);

```

B.5 Maintenance Tasks Generator's Script

```
1 t=ReadTime();
2 NTurbines=ReadNTurbines();
3 PMneed=ReadPM(); %yearly need of preventive maintenance
   per turbine
4 YearTime=t-floor(t/8760)*8760; %sets the time of the
   current year
5
6 persistent PMyear %PMyear is the amount of PM tasks
   generated in the current year
7           %CMShift: number of CM generated since
           the last shift started
8
9 if isempty(PMyear) %initialization of variable PMyear
10     PMyear=0;
11 end
12
13 %First PM are assigned when generated
14 if YearTime<=2160 && PMyear>(NTurbines*PMneed/2) %
   initializes PMyear each new year
15     PMyear=0;
16     PM=0;
17 elseif YearTime>=2160 && PMyear<(NTurbines*PMneed/2) %
   generates half PMs of the year in the first shift from
   the end of march
18     PM=NTurbines*PMneed/2;
19     WriteTasks(ReadTasks()+PM);
20 elseif YearTime>=6480 && PMyear<(NTurbines*PMneed) %
   generates half PMs of the year in the first shift from
   the end of september
21     PM=NTurbines*PMneed/2;
22     WriteTasks(ReadTasks()+PM);
23 else
24     PM=0;
```

```

25 end
26
27 PMyear=PMyear+PM;
28
29 %CM Taks generation
30 NTurbines=ReadNTurbines();
31 CMNeed=ReadCM();
32 Lambda=CMNeed*NTurbines/(365*24);
33 CM=zeros(5,1); %Initialization variable CM: Corrective
   Maintenance tasks generated in one hour
34 R=rand %Variable R that will take the value of a random
   number
35     if R>(1-poisspdf(3,Lambda)) %probability of 3
   failures
36         CM=CM+3;
37     elseif R>(1-poisspdf(2,Lambda)) %probability of 2
   failures
38         CM=CM+2;
39     elseif R>(1-poisspdf(1,Lambda)) %1 failure
40         CM=CM+1;
41     else %no failure
42         CM=CM;
43     end
44 WriteNTasks(ReadNTasks()+ReadTasks()+CM);
45 WriteTasks(0);
46
47 %Down Turbines Generation
48 ADT=0.3; %Average times a year each turbine is expected
   to get to be out of service because of failure
49 Lambda=ADT*NTurbines/(365*24); %Poisson factor for a
   failure leading to turbine down
50 DownTurbines=zeros(5,1); %Variable initialization
51 NFarms=ReadNFarms(); %Maximum farms to be served
52 for n=1:NFarms
53     R=rand %iVariable R that will take the value of a
   random number

```

```

54     if R>poisspdf(0,Lambda)  %Probability of more than 1
55         turbine down
56         if R>(1-poisspdf(2,Lambda)) %probability of 2
57             turbines down
58             DownTurbines(n)=2;
59         else
60             DownTurbines(n)=1;
61         end
62     else
63         DownTurbines(n)=0;
64     end
65 WriteDownTurbines(ReadDownTurbines()+DownTurbines);

```

B.6 Sea State Entity Generator's Script

```

1  persistent SeaState MCSeaStates t
2  coder.extrinsic('xlsread');
3  if isempty(SeaState)
4      MCSeaStates=zeros(12,12);
5      MCSeaStates=xlsread('MCSS.xlsx');
6      SeaState=randi(12);
7      t=1;
8      WriteSeaState(SeaState);
9  end
10 if t==3
11     r = rand();
12     for j = 1:length(MCSeaStates(:,1))
13         prob = 0;
14         for k = 1:j
15             prob = prob + MCSeaStates(SeaState,k);
16         end
17         if r <= prob
18             SeaState = j;
19             WriteSeaState(SeaState*1.0508); %StateRange

```

```
20         break;
21     end
22 end
23 t=0;
24 else
25     t=t+1;
26 end
```

B.7 Wave Forcast Entity Generator's Script

```
1 persistent WaveForecast
2
3 %This action will generate an array of 2 components. Each
4 component is the
5 %expected wave height during a period of 6 hours, at 6:00
6 AM and 12:00 PM (noon)
7 % the vessels to operate depending on the the value
8
9 coder.extrinsic('csvread');
10 % Reads the csv file from waveclimate.com into a matrix
11 % 23 tells csvread to skip the 23 first lines, as the
12 data starts in line 24
13 A = csvread('RawDataCSV.csv',23,0);
14
15 t=ReadTime();
16 %Trasnsforms the global time of the simulation (1 step=1
17 hour) to the
18 %day number and time period of 3 hours (from 1 to 8) of
19 the day in which we are (day 0 at the beginning of the
20 simulation)
21 Day=floor(t/24)+1;
22 t=floor((t-(Day-1)*24)/3)+1;
23 R=(Day-1)*8+1; %identifies row of the hour 0:00 of the
24 current day
25
26
```

```

19 %Define the WaveForecast vector for 6h forecast
20 WaveForecast=zeros(2,1);
21
22 %Assign components of the vector
23 WaveForecast(1)=A(R+2,7);
24 WaveForecast(2)=A(R+4,7);
25
26 WriteWaveForecast(WaveForecast);
27 %WaveForecast vector already created

```

B.8 Costs Function's Script

```

1 function [Total,TotalDown,Downtime,Operation] = fcn(
    SOV_Operation, SOV_Chartering, CTV_Operation,
    Turbine_Downtime, Technicians)
2 persistent Costs CummDown
3 if isempty(Costs) %Variable y initialization
4     Costs=0
5 end
6 if isempty(CummDown) %Variable y initialization
7     CummDown=0
8 end
9
10 A=SOV_Chartering/24;
11
12 Base=ReadSOVinBase();
13 if Base==0 %SOV not in base, then operating
14     B=SOV_Operation/24;
15 else
16     B=0;
17 end
18
19 Farm=ReadSOVinFarm();
20 Wave=ReadWaveForecast();
21 HsCTV=ReadHsCTV();

```

```

22 Time=ReadTime()
23 Day=mod(Time,24) %Time of the day
24 NCTV=ReadNCTV()
25
26 if Farm==1 && Day<12 && Wave(1)<=HsCTV %SOV in farm and
    CTV operating
27     C=(CTV_Operation/24)*NCTV;
28 elseif Farm==1 && Day<18 && Wave(2)<=HsCTV
29     C=(CTV_Operation/24)*NCTV;
30 else
31     C=0 %CTV does not operate,
32 end
33
34 Down=ReadDownTurbines();
35 Down=sum(Down);
36 D=(Turbine_Downtime/24)*Down;
37
38 %Cost technicians
39 E=((Technicians/24)*ReadMaxTechnicians()*2*(8760-
    ReadWaitingTimeB()))/8760;
40
41 Downtime=D
42 Operation=A+B+C+E
43 Costs = Costs+A+B+C+D+E;
44 Total=Costs
45 CummDown=CummDown+D
46 TotalDown=CummDown+D

```

B.9 ShiftTime MATLAB function's script

```

1 function fcn
2 ShiftTime=ReadTime()-ReadShiftStart();
3 WriteShiftTime(ShiftTime);
4 end

```

B.10 ShiftDays MATLAB function's script

```
1 function fcn
2 ShiftTime=ReadShiftTime();
3 ShiftDays=floor(ShiftTime/24)+1;
4 WriteShiftDays(ShiftDays);
5 end
```

B.11 MinTasks MATLAB functions's Script

```
1 function fcn
2 Time=ReadTime();
3 if mod(Time,8760)<2160 || mod(Time,8760)>6480 %Winter
4     MinTasks=10; %Winter
5 else
6     MinTasks=20; %Summer
7 end
8 WriteMinTasks(MinTasks);
9 end
```

B.12 MaxTechnicians MATLAB function's Script

```
1 function fcn(MaxTechnicians)
2 TSOV=ReadTSOV();
3 TCTV=ReadTCTV();
4 NCTV=ReadNCTV();
5 %MaxTechnicians: Intended max amount of technical
   personnel on board
6 if MaxTechnicians==0
7     Max=6*(TSOV+NCTV*TCTV) %Max amount that will be
   possible to transfer in a day
8 elseif MaxTechnicians>6*(TSOV+NCTV*TCTV);
9     Max=6*(TSOV+NCTV*TCTV) %Max amount that will be
   possible to transfer in a day
```

```
10 else
11     Max=MaxTechnicians;
12 end
13 WriteMaxTechnicians(Max);
14 end
```