

CHALMERS



The development of a Life Cycle Cost model for an offshore wind farm

Master's Thesis in the Solid and Fluid Mechanics programme

MARCUS NORDAHL

Department of Applied Mechanics
Division of Solid and Fluid Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2011
Master's Thesis 2011:08

MASTER'S THESIS 2011:08

The development of a Life Cycle Cost model for an offshore wind farm

Master's Thesis in the Solid and Fluid Mechanics

Marcus Nordahl

Department of Applied Mechanics
Division of Solid and Fluid Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2011

The development of a Life Cycle Cost model for an offshore wind farm
Master's Thesis in the Solid and Fluid Mechanics Master's programme
MARCUS NORDAHL

© MARCUS NORDAHL, 2011

Master's Thesis 2011:08
ISSN 1652-8557
Department of Applied Mechanics
Division of Solid and Fluid Mechanics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
An offshore wind turbine in harsh weather.

Chalmers reproservice / Department of Applied Mechanics
Göteborg, Sweden 2011

The development of a life cycle cost model for an offshore wind farm
Master's Thesis in the Solid and Fluid Mechanics Master's programme
Marcus Nordahl
Department of Applied Mechanics
Division of Solid and Fluid Mechanics
Chalmers University of Technology

ABSTRACT

Not until recently most people have realised the severity of what all emissions are doing to mother earth. Today most countries are eco-friendly and see the pride in working towards a carbon independent community. This has led to an escalating growth of wind power energy and the development in this industry is growing like never before. To get as much out of the wind turbines as possible a lot of wind farms have lately been built offshore. The backside of this is that it is harder to predict the life cycle costs for the wind farm because of the lack of experience with offshore wind farms and the availability due to weather. The aim of this thesis is to develop a model that taking this into count when calculating the life cycle cost for a wind turbine.

The model shows what kinds of costs and revenues that can be expected when building a wind farm. It also calculates the cost of energy per produced energy set over a lifetime. This is an accessible result that easily can be compared and since the results are different for each simulation it is presented in a distribution curve.

Key words: Wind, turbine, offshore, life, cycle, cost, model, LCC, Monte, Carlo, method, Horns, rev, reef, farm, park

Utvecklingen av en livstidskostnadsmodell för en offshore vindkraftspark

Examensarbete inom Solid and Fluid Mechanics

MARCUS NORDAHL

Institutionen för tillämpad mekanik

Avdelningen för Solid and Fluid Mechanics

Chalmers tekniska högskola

Contents

ABSTRACT	I
CONTENTS	III
NOTATIONS	V
1 INTRODUCTION	1
1.1 Problem formulation	1
2 METHOD OF APPROACH	3
3 LIFE CYCLE COST MODEL	5
3.1 Structure of LCC model	5
3.2 Bath tub curve	5
3.3 Analysis of the failure rates	5
3.4 Total costs	8
3.4.1 Total installation costs	8
3.4.2 Operation and Maintenance	8
3.4.3 Capital costs	11
3.5 Energy Production	11
3.6 Decommission	12
3.7 Life Cycle Cost per produced energy	13
4 RESULTS	15
4.1 Starting case	15
4.2 Different input parameters impact on the result	18
4.2.1 Failure rate	18
4.2.2 Interest rate	20
4.2.3 Discount rate	21
4.2.4 Inflation factor	22
4.3 The results compared to data from offshore wind farms in operation	23
5 CONCLUSION AND FURTHER WORK	25
6 REFERENCES	27
APPENDIX A – MATLAB CODE	29

Notations

Roman upper case letters

LCC	Life Cycle Cost
WT	Wind Turbine
O & M	Operation and Maintenance
EP	Energy price, [€/kWh]
TP	Total energy production for a WT
BD	Bath tub distribution
V	Total vessel costs
CC	Component costs
NPV	Net present value
DR	Discount rate
PMC	Preventive Maintenance Costs
NOM	Number of scheduled maintenances annually
Cap	Capital cost
I	Inflation rate
IC	The total installation cost
DC	Decommission cost
W	Warehouse expenses
BC	Basic expenses for an offshore wind park without reparations
Bal	Balancing Cost

1 Introduction

The interest of wind energy has escalated the last years. The aspiration of becoming independent from energy production through fossil fuels has led to a race where more and more countries are expanding their renewable energy sources. This has led to a major increase in new wind farms. To make the wind turbines (WT) more cost efficient they are continuously growing larger. Lately the wind farms are built in more favourable areas such as offshore where size is not an issue and wind is more beneficial.

But there are not only advantages with going offshore, harsh weather and higher loads require more robust WTs. Onshore, access is not an issue, offshore the availability is very dependent on the weather. All stops can cause major problems and even a small cheap component breakdown cost a lot to repair since the transportation to the site by vessel or helicopter is expensive and the weather can lead to a long downtime for the WT.

Those contingencies complicate the prediction of the LCC for the wind turbines. In order to make a trustworthy forecast, of the final cost per produced energy, the model needs to take as many dependent parameters into consideration as possible.

This master thesis is presenting a simplified model that later on can be extended to a full scale LCC model.

1.1 Problem formulation

The aim with this thesis is to make a simplified model that calculates the total cost per produced MWh. Since the model includes variables which are uncertain, the Monte Carlo method is used to show the result in a distribution curve.

The model shall also be tested with changed input parameters to see what impact that got on the result. At last the result shall be compared to statistic data from existing wind farms.

The philosophy of this project is not to make a full scale LCC model with all statistical input data that is necessary to get a reliable result. The model will be a tool that will simplify for further development of the model.

2 Method of approach

The first phase in the project was to gather information, about offshore wind power, to build up a foundation of understanding before starting the modelling. After that the searching for input data to the model started. This was the first big obstacle during the thesis. After several calls to scientists, wind farm owners and WT manufactures, it was clear that my first ideas of what that should be included in the model needed to be reduced. It took quite a long time before the first failure rates were achieved. It was the failure rate for the 9th year on Horns Rev wind farm. That data were the foundation in the model and it was around those values that the model was built around.

In order to make a model where the variables are uncertain and can change from one simulation run to another a Monte Carlo simulation is necessary.

Monte Carlo methods are a class of algorithms that are used to simulate different kinds of mathematical systems. It uses repeated random sampling to compute its result. This method is used when there are many uncertainties such as in the LCC model.

After that the input parameters were change to show what impact each and every one of them had on the result. Finally the results were compared to results achieved in another report.

3 Life Cycle Cost Model

This section describes the structure of and the theoretical background of the LCC model.

3.1 Structure of LCC model

The model depends on several different types of input data as seen in Figure 1. In this chapter these categories of input data will be described and the dependents between them.

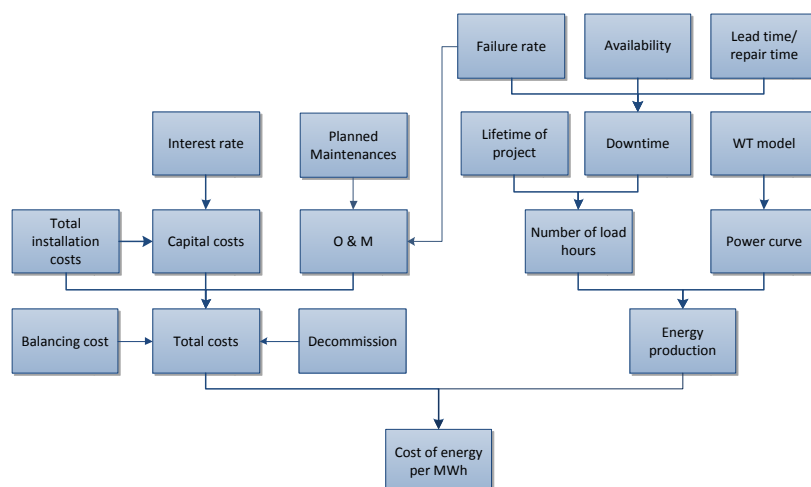


Figure 1: The structure of the LCC model

3.2 Bath tub curve

The bath tub curve is a curve that describes the distribution of failures for WT. In the early years of the WT there are a lot of failures and the probability of failure is relatively high. This is particularly true when the type of wind turbine plant has not accumulated so many years of production. When the initial problems are solved, the failure rate will decrease until the WT will start to fail more frequently due to fatigue problems. This gives a failure rate curve shape that reminds of a bath tub.

3.3 Analysis of the failure rates

The failure rates used in the model is for 43 different components (Lindqvist & Lundin, 2010). Lindqvist and Lundin got the failure rate by checking the spare parts used during the ninth year of operation for Horns Rev, which consists of 80 Vestas V80-2.0MW. This gives the failure rates after 700 800 operating hours. Since it is the ninth year the early problems are gone and the WT are in the bottom of the so called bath tub curve. The bath tub curve is a curve for the failure distribution. In this case the WT's were somewhere in the middle of the bath tub. To compensate for this in the model the failure rate needs to be multiplied with a bath tub factor.

In Figure 2 below one can see how a typical failure rate depends on time. With increased experience in both operation and production the failure rate decreases (S. Faulstich, 2008).

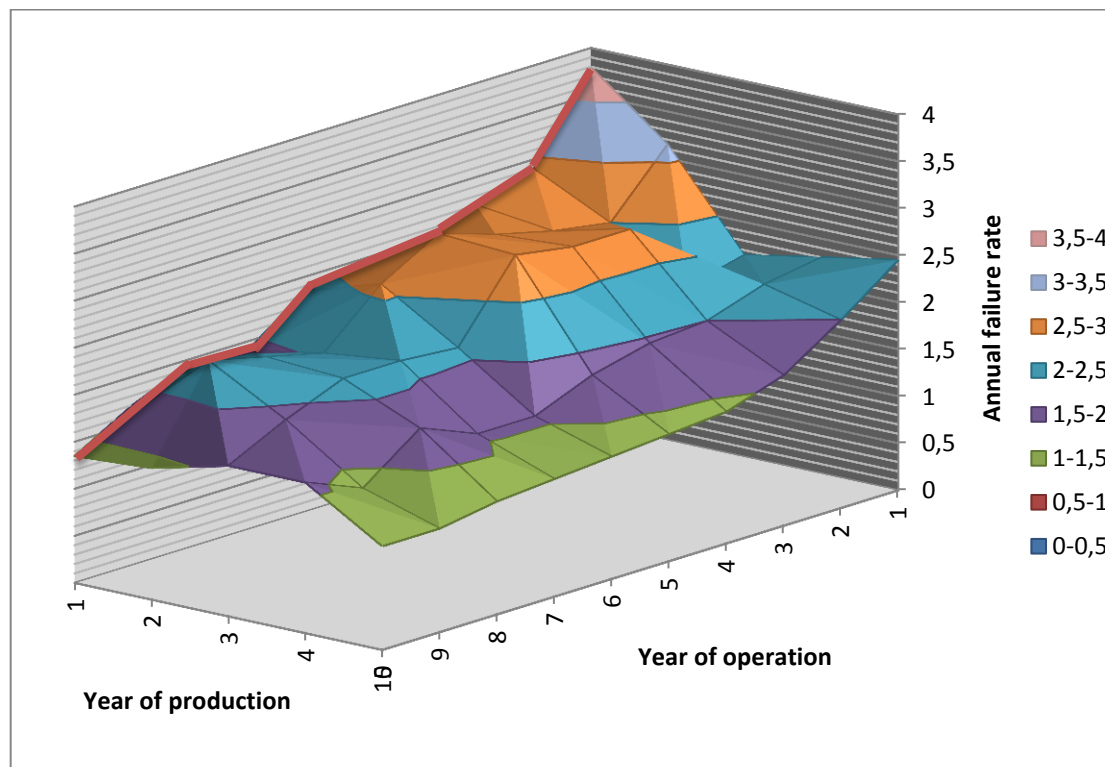


Figure 2: Failure rate as a function of time (S. Faulstich, 2008)

Since Horns Rev was using a new WT model the values for year of production was set to one, see red line in Figure 2. These data were used to make the bath tub curve in Figure 3.

The data shown in Figure 2 are only statistical data for the first ten years. To be able to make a bath tub curve for the full lifetime of 20 years there has been taken an assumption that the bath tub curve for year 11 to 20 is a mirroring of the first ten years.

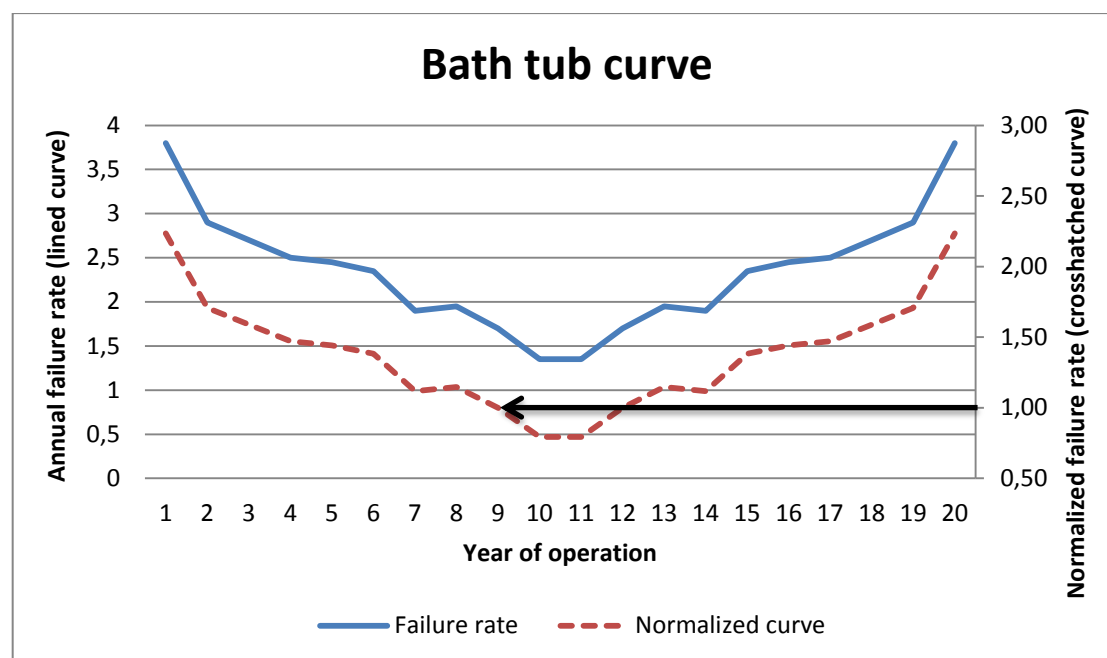


Figure 3: Bath tub curve

By normalizing the curve compared to the failure rate of the ninth year of operation (see crosshatched curve) it can be used in the model. The bath tub factor mentioned before can now be calculated simply by integrating the crosshatched normalized curve and divide it by 20.

$$c_{Bath} = \frac{1}{20} \sum_{year=1}^{20} \{failure\ rate\}_{year}$$

The failure rates taken from (Lindqvist & Lundin, 2010) are failure rates per hours. To calculate the failure rates per year following equation is used:

$$p_{year} = 1 - (1 - p_h)^{365 \cdot 24}$$

The model is using Monte Carlo's method which means that it takes random samplings to compute its result. One sampling could mean that one part either brakes or not. In order to make the Monte Carlo's method to work in the model, the probability for the number of failure needs to be calculated by using:

$$p_{fails} = p^n (1 - p)^{y-n} \binom{y}{n}$$

where p_{fails} is the probability of n failures during y years. By doing this on all the components used in the model the result in Figure 4 can be achieved.

Example: The failure rate for the gearbox is 5.48 ppm per hour. To compensate the failure rate due to the bath tub curve it is multiplied by c_{Bath} . This will give the average failure rate for the full lifetime, before it was the failure rate for the ninth year. The failure rate is then changed from failure rate per hour to failure rate per year. Finally that rate is used to calculate the probability that the gearbox will fail n times (p_{fails}). This gives a distribution vector for the probability of failures and it can look like the table below.

Failures	0	1	2	3	4	5	6	7	8
Fail. rate	26%	36%	24%	10%	3%	0.7%	0.1%	0.02%	0.002%

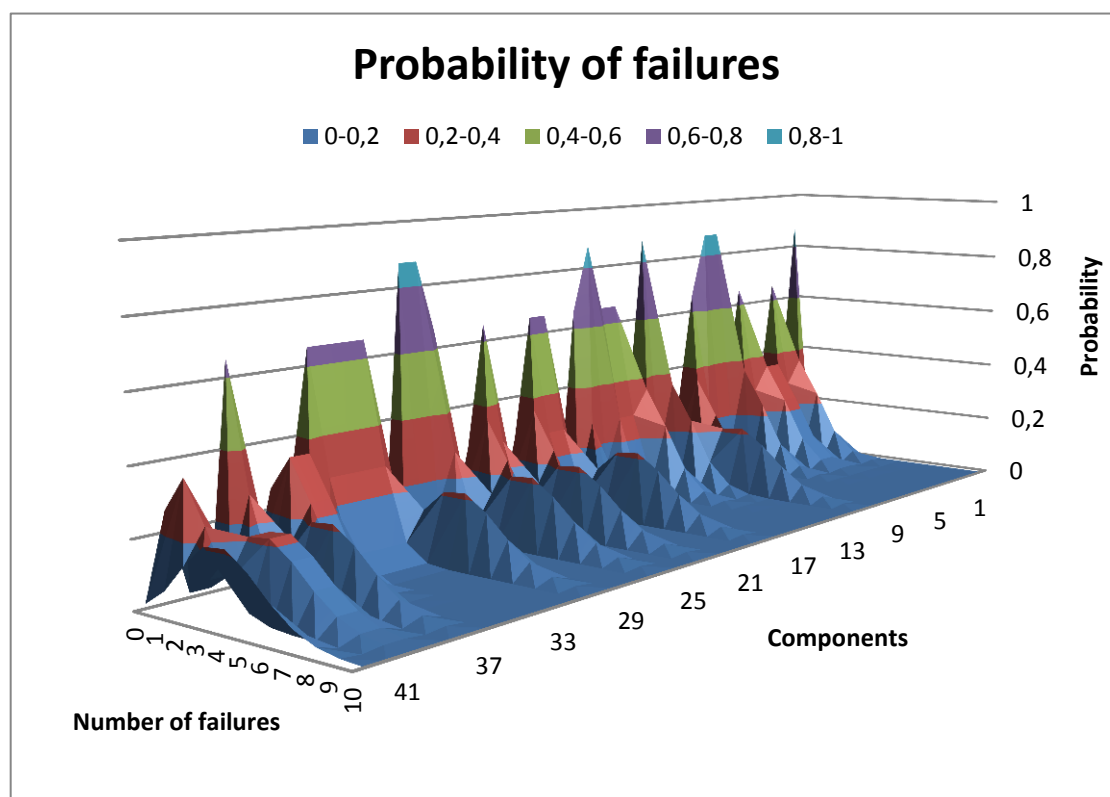


Figure 4: Probability of failure – distributed over time using the mirrored Faulstich data set.

3.4 Total costs

The total costs are the sums of all expenses during a lifetime for the WT. These costs have been divided into three groups: total installation costs, capital costs and O & M.

3.4.1 Total installation costs

These are the total invested money to build a wind park divided by number of WT. This includes all costs up to the start of the WT's.

3.4.2 Operation and Maintenance

The O & M are divided into one operation and one maintenance part. The model is using a fixed operational cost due to (The Crown Estate, 2009) which is approximately €30 million for a wind farm. This is the basic expenses for an offshore wind farm without any reparations.

In the maintenance part there are preventive maintenances twice a year. During summer there are a more detailed service were for example filters and a number of batteries are replaced. During winter a smaller maintenance is done where some tasks are repeated, like filling grease. The average costs for the preventive maintenances are about €5800 for a 2 MW WT according to (McMillan & Ault, 2007).

The reparation costs are dependent of the failure rates, the components prices and the availability of the WT's. The failure probabilities were calculated in the chapter Bath tub curve

The bath tub curve is a curve that describes the distribution of failures for WT. In the early years of the WT there are a lot of failures and the probability of failure is relatively high. This is particularly true when the type of wind turbine plant has not accumulated so many years of production. When the initial problems are solved, the failure rate will decrease until the WT will start to fail more frequently due to fatigue problems. This gives a failure rate curve shape that reminds of a bath tub.

Analysis of the failure rates". The model randomly, considering the probability for each case, chooses for each component how many times it will break down. This means that a component with a failure probability like this:

0 fails	1 fails	2 fails	3 fails	4 fails	5 fails	6 fails	7 fails	8 fails
80%	17%	2.5%	0.42%	0.06%	0.02%	0.003%	260ppm	38ppm

will in about 80% of the simulations fail zero times, fail once in about 17% of the cases and so on. So the discrete sampler weights in the probability when randomly choosing. When one of these is chosen the component price multiplied with number of fails will be added to the total cost.

The downtime for each fail will then be chosen. To do this the weather needs to be analysed. The wind data from Egmond aan Zee offshore wind farm has been analysed. The wind data were given every 10th minute during several years, but it was the wind data from 2006-2009 that were analysed (NordzeeWind, 2006-2009). Since the maximum allowed wind speed for reparation depends on which part that needs to be fixed, the analysis needs to be carried out for all different cases. The wind speeds allowed varies from 7 m/s to 20 m/s (McMillan & Ault, 2007).

The wind data is loaded in a vector, a Matlab program then checks for how long the wind is below analysed wind speed, before it winds up above that speed again, each time. For example see Figure 5 for the case when wind speed 7 m/s is analysed. In this case the wind speed goes below 7 m/s 10 times and the result for this case looks like:

1.7h	20.3h	0.7h	20.5h	1.3h	1.0h	0.7h	0.8h	16.2h	43.0h
------	-------	------	-------	------	------	------	------	-------	-------

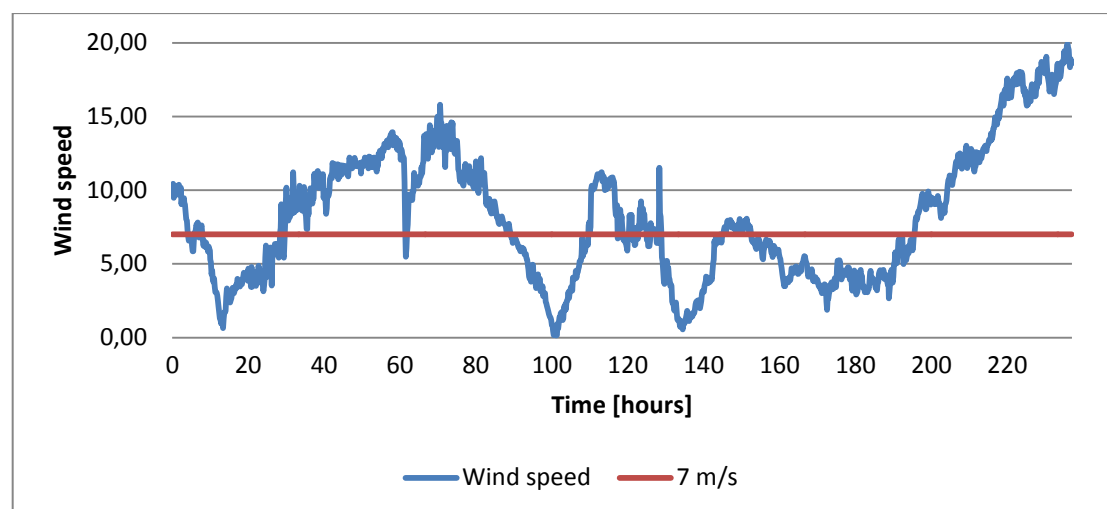


Figure 5: Wind variation with time

The next step is to check, for each component, the probability that the site will be available for the needed reparation time. This means that the winds speed needs to be below the restriction during the entire service. By summing the total time when the site is available for full service and divide it with number of hours during four years, the availability factor is achieved.

The availability factor is used to get the probabilities for waiting time before the weather is good enough. To check the probability that the waiting time will be n days ($prob_n$) following equation is used:

$$prob_n = p_{avail} \cdot (1 - p_{avail})^{(n-1)}$$

where p_{avail} is the availability factor.

The same discrete sampler that was used to get the number of fails is also used to get the waiting time before right weather condition. The vector $prob_n$ is used as an input parameter for the sampler. The downtime for a failure is the sum of lead time, the waiting time (for weather), transportation time to site and reparation time.

Each time there is a failure, on any WT, the model checks if there are any simultaneous failures on any of the other WTs in the park. To do this the failure rate for 80 WTs during one day (instead of one hour) has been calculated:

$$p_{80} = 1 - (1 - p)^{80 \cdot 24}$$

where p is the failure rate for one WT during one hour and p_{80} is the failure rate for 80 WTs during one day.

To get the probability of simultaneous failures the model goes through all components and calculates the waiting time for it. After that it checks through all components again and in those cases when the waiting time for the second component is smaller, it checks if that component will fail before the first is repaired. It will look like Figure 6.

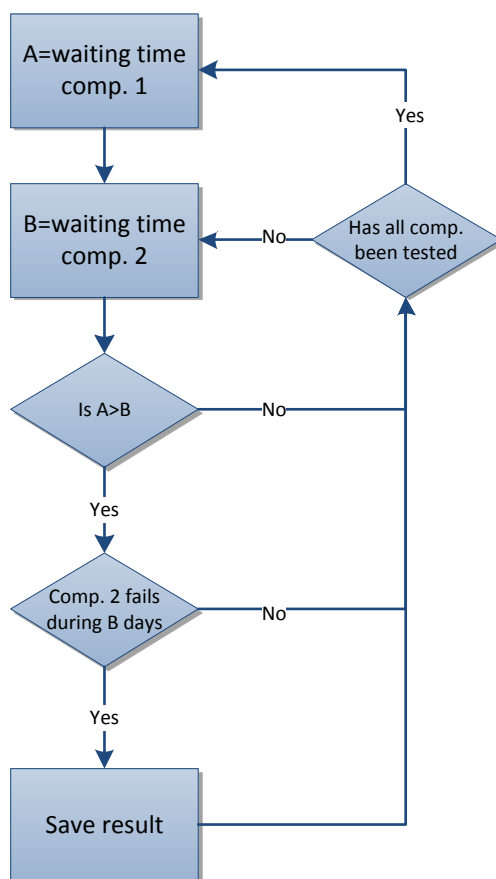


Figure 6: Simultaneous failure model

When this model has been running for the wished number of iterations the result needs to be analysed. The result from the model above has been saved in a matrix with components in the rows and number of simultaneous failures in the columns. By counting the number of simultaneous failures component wise the probability of simultaneous failure can be calculated.

For example, after a simulation with 1000 iterations, component #1 got 912 cases with zero simultaneous failure, 80 cases with one simultaneous failure, 7 cases with two simultaneous failures and 1 case with three simultaneous failures. This gives the following probability vector of simultaneous failures:

Number of fails	0	1	2	3
Probability	91.2%	8%	0.7%	0.1%

Every time the model gets a failure it checks if there are any simultaneous failures using this result and the discrete sampler. An assumption is that there are capacities, in most cases, to do the reparations during the same day. So in the cases when there are simultaneous failures the vessel cost will be divided by number of simultaneous failures.

3.4.3 Capital costs

The capital costs are the interest on the invested money in the wind farm. In the model the interest will start at the same day as the WT's are. The loan is assumed to be paid

down during 20 years with a fix amortisation. The interest and the amortisation are paid once a year. This gives the expression for the interest.

$$\{capital\ costs\}_n = capital_n \cdot interest\ rate$$

Where $\{capital\ costs\}_n$ are the capital cost (year n) and capital is the invested money that reduces every year by the amortisation.

3.5 Energy Production

To calculate the total energy production the power curve, the wind and the downtime needs to be analysed. The power curve is gained for a Vestas V80 WT (Wind Turbine Power Calculator) and it looks like in Figure 7.

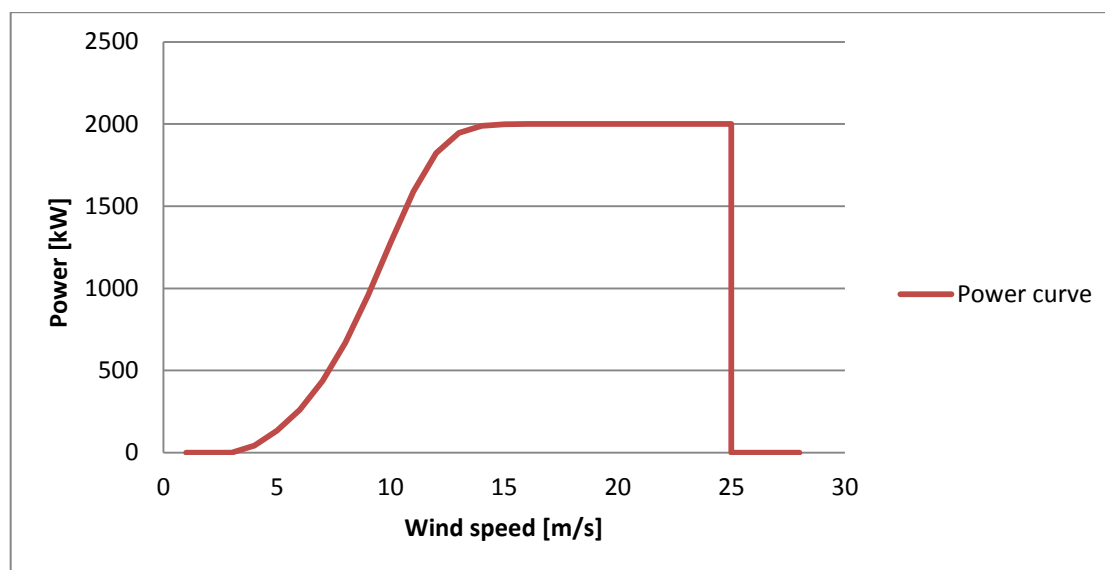


Figure 7: Power curve for Vestas V80 2MW

In table format it looks like:

1-3m/s	4 m/s	5 m/s	6 m/s	7 m/s	8 m/s	9 m/s	10 m/s
0 kW	44,1kW	135 kW	261 kW	437 kW	669 kW	957 kW	1279 kW
11 m/s	12 m/s	13 m/s	14 m/s	15 m/s	16-25 m/s	above	
1590kW	1823 kW	1945 kW	1988 kW	1998 kW	2000 kW	0 kW	

The wind data from Egmond aan Zee offshore wind farm (NordzeeWind, 2006-2009) is used to get the maximum possible energy production during four years. Since the power only is given to integers and the power curve is very steep a linearization between those values are necessary. A Matlab program has been used to calculate the total maximum energy production during 2006-2009. The results are:

2006	2007	2008	2009
8 187 083 kWh	8 746 177 kWh	7 108 901 kWh	8 978 100 kWh

The model randomly picks 20 energy productions from the selection above to get the maximum energy production during a lifetime of 20 years. To get the energy production the model uses that result and den subtracts it with production losses due to

downtime, which is the downtime multiplied with a fix power (default setting for model is 1400 kWh).

3.6 Decommission

The model is using a full scope scenario where the turbine, its foundation and both internal and external cables are commissioned. The actual decommission cost 15 to 20 years from now is hard to predict because it depends off many factors like material prices of steel, stainless steel and copper. According to (Fredrik Ardefors, 2009) the total cost for decommission is approximately €125 000. That amount includes both the sum of all decommission cost and return from the realization of the recycled material.

3.7 Life Cycle Cost per produced energy

The LCC cost is the total cost during the lifetime for a WT. But in order to get the net present value (NPV) a discount rate needs to be considered. The discount rate is the rate of return that could be earned on an investment in the financial market with similar risk. This means that all payments are being discounted back to its present value.

To make the model more accurate a bath tub factor is needed because the failure rates are following a bath tub curve. This means that the WT's fails more often during the first years and the last years of the lifetime than it does halfway through. So by divide every value in the bath tub curve with the sum of all values it will give a Bath tub distribution (BD) with how many per cent of the total number of failures that will occur each year. The result will look like Figure 8 when BD is plotted.

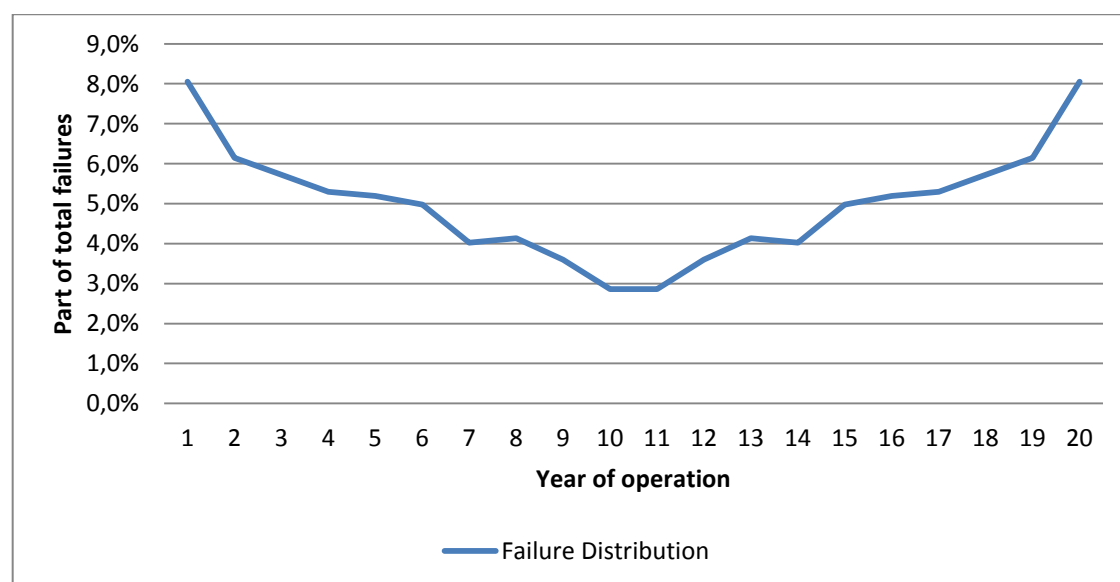


Figure 8: Failure distribution

In some countries, wind farm owners are responsible for balancing the power production from the WTs. In Denmark this balancing cost has been about €3/MWh

according to previous experience in a system where the wind power covers approximately 20% of the national electricity demand (Krohn, Morthorst, & Awerbuch, 2009).

With this in consideration the LCC per produced energy can be calculated by the following formula.

$$LCC/MWh = \frac{IC + \frac{DC}{(1+DR)^{20}} + \sum_{i=1}^{20} \frac{Cap + (1+I)^i \cdot (W + BC + PMC + BD \cdot (V + CC))}{(1+DR)^i}}{TP} + Bal$$

IC	The total installation cost
DC	Decommission cost
Cap	Capital cost (for each year)
I	Inflation rate
W	Cost to have supplies in warehouse (for each year)
BC	Basic expenses for an offshore wind park without reparations (for each year)
PMC	The Preventive Maintenance Costs (for each year)
BD	Bath tub distribution (for each year)
V	The total vessel costs
CC	The total component costs
DR	Discount rate
TP	The total energy production for a WT
Bal	Balancing cost

4 Results

The results will be presented in different cases. At first one starting case will be presented. After that the failure rates, inflation rate, interest rate and the discount rate will be changed one-by-one to show what impact they got on the result.

The most accessible and reliable way to present the result is by presenting the total cost during a WT's life per produced MWh as a distribution curve. This gives a result that most people can relate to.

4.1 Starting case

In this case the input variables are the following:

Interest rate	7%
Inflation factor	2%
Discount rate	7.5%
Number of simulations	10 000

In Figure 10 the result from the simulation of a wind farm with the input parameters shown above is shown. To be able to present the result in a distribution curve as below one need to divide the result into small intervals. If not, the curve would result in a straight line since it is not likely to have multiple identical results. In all simulations in this model the results have been rounded to nearest integer.

Just to show how the interval affects the curve, the same results will also be plotted with the intervals 0.5 MWh and 2 MWh respectively.

The case when the interval was set to 0.5 MWh (Figure 9) the distribution is spikier and the peak is about half the size of the one presented in Figure 10. This is logical since the smaller interval the less results will be included there.

The other case when the interval is 2 MWh (Figure 11) it result in a curve with a peak that is about twice the height compared to Figure 10. The problem with that curve is that it is not as accurate.

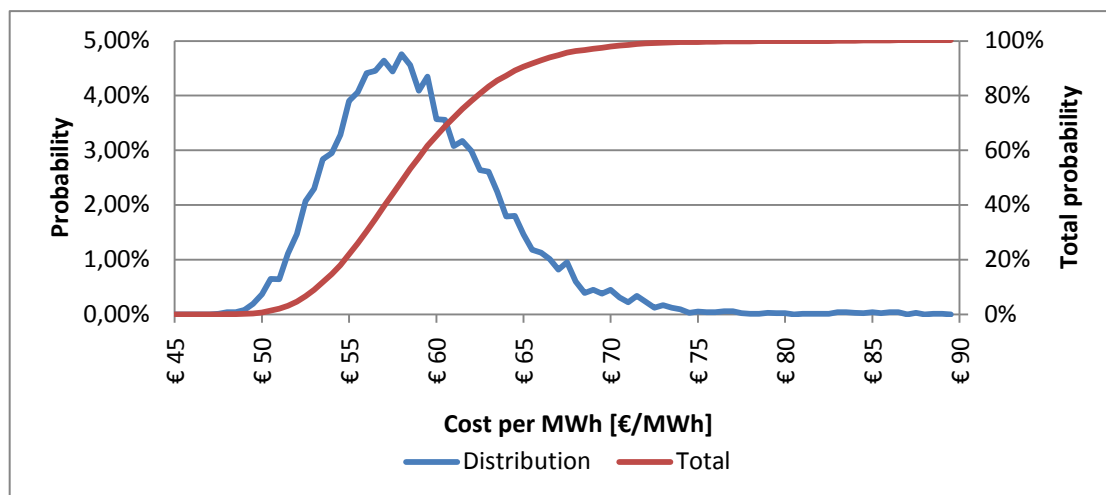


Figure 9: Distribution curve of cost per MWh for the realistic case with interval length of 0.5 €/MWh.

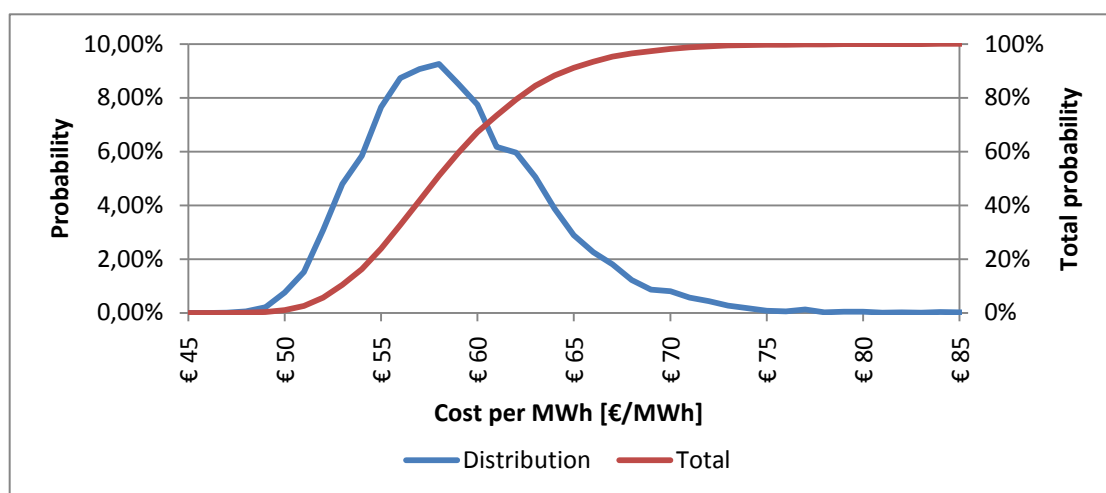


Figure 10: Distribution curve of cost per MWh for the realistic case with interval length of 1 €/MWh.

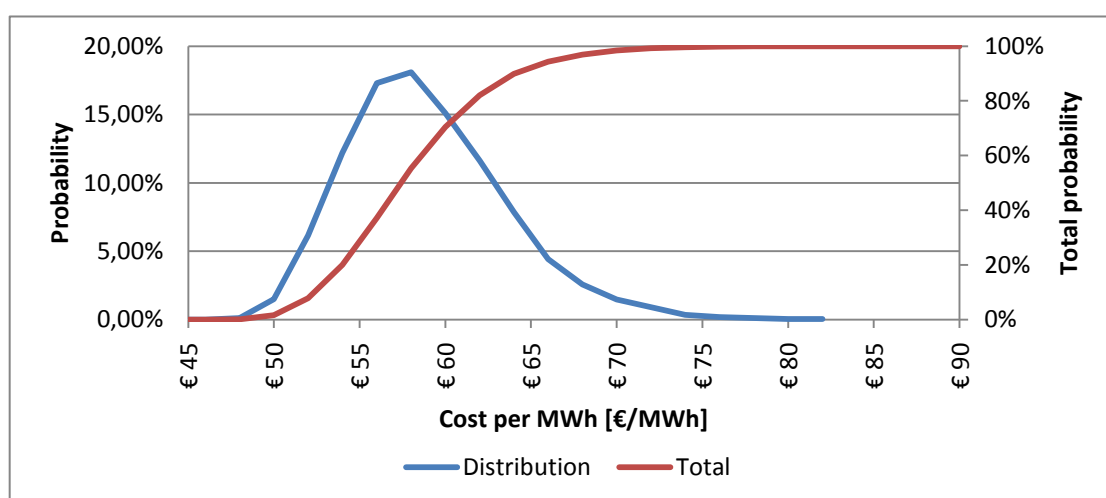


Figure 11: Distribution curve of cost per MWh for the realistic case with interval length of 2 €/MWh.

To be able to make a better comparison between the results later, the average values of the total cost, total production, downtime, vessel cost, interest cost, interest profit and profit will be presented in the table below.

LCC per MWh	€ 58.9/ MWh
Total cost	€ 8.76 millions
Total production	149 GWh
Downtime	11 700 hours
Vessel cost	€ 2.52 millions
Capital cost	€ 1.54 millions

The cost can be divided into three groups, leveled investments, O&M and Balancing cost, where leveled investments are the invested money (installation costs) and the capital cost. A plot of this will look like in Figure 12.

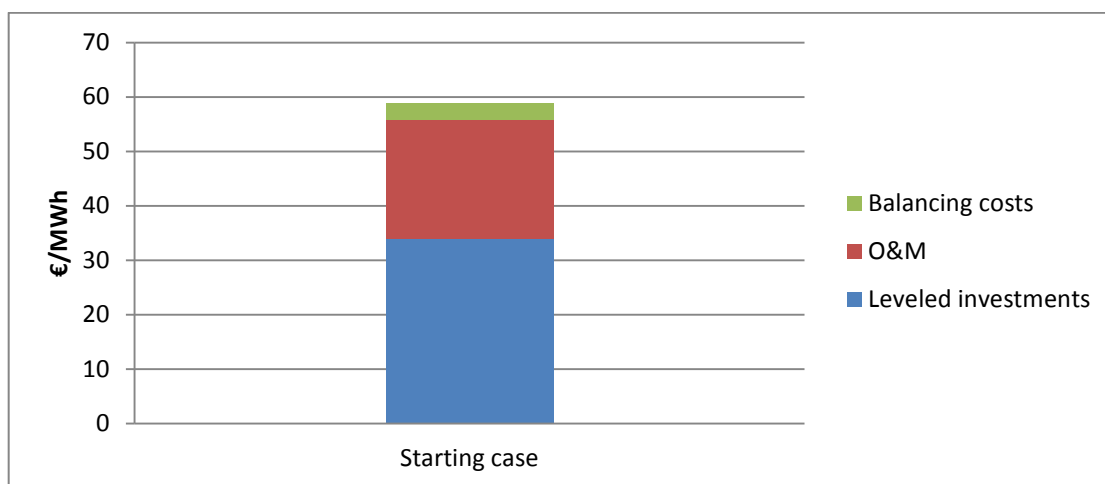


Figure 12: The calculated production cost (starting case)

4.2 Different input parameters impact on the result

To be able to see the impact of the input parameters they need to be changed one-by-one. This will show what it will affect and more interesting how much it will affect the models result. This will be emanating from the optimistic case.

4.2.1 Failure rate

The first test is to see how sensitive the failure rates are. To do this the failure rate has been increased by 20 % on all components.

	Starting case	Starting case +20%	Difference	Difference [%]
LCC per MWh	€ 58.9/ MWh	€ 62.9/ MWh	€ 3.93 /MWh	6.7 %
Total cost	€ 8.76 millions	€ 9.15 millions	€ 0.392 millions	4.5 %
Total production	149 GWh	146 GWh	-3.07 GWh	-2.1 %
Downtime	11 700 hours	13 900 hours	2 290 hours	20 %
Vessel cost	€ 2.52 millions	€ 3.02 millions	€ 0.496 millions	20 %
Capital cost	€ 1.54 millions	€ 1.54 millions	€ 0	0 %

The cost distributions for the both cases are compared in Figure 13.

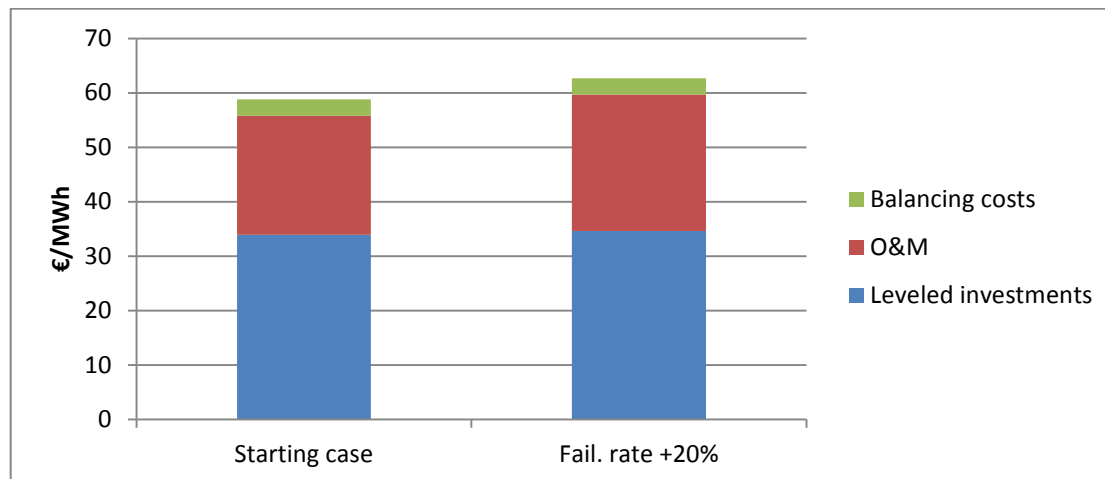


Figure 13: The calculated production cost (starting case and failure rate +20%)

In the next case the failure rate on the gearbox has been raised by 100 %. The reason why the gearbox was chosen is because it is the component that is the most costly (meaning component costs plus vessel cost are the highest). The results for this case are shown in the table below.

	Starting case	Starting case Gearbox +100%	Difference	Difference [%]
LCC per MWh	€ 58.9/ MWh	€ 63.1/ MWh	€ 4.13 /MWh	7.0 %
Total cost	€ 8.76 millions	€ 9.21 millions	€ 0.446 millions	5.1 %
Total production	149 GWh	146 GWh	-2.69 GWh	-1.8 %
Downtime	11 700 hours	13 600 hours	1950 hours	17 %
Vessel cost	€ 2.52 millions	€ 2.72 millions	€ 0.202 millions	8.0 %
Capital cost	€ 1.54 millions	€ 1.54 millions	€ 0	0 %

If instead the second most costly component's, which is the generator, failure rate is increased by 100% the following result is given.

	Starting case	Starting case Generator +100%	Difference	Difference [%]
LCC per MWh	€ 58.9/ MWh	€ 59.8/ MWh	€ 0.83 /MWh	1.4 %
Total cost	€ 8.76 millions	€ 8.89 millions	€ 0.122 millions	1.4 %
Total production	149 GWh	149 GWh	-0.02 GWh	-0.0 %
Downtime	11 700 hours	11 700 hours	71 hours	0.6 %
Vessel cost	€ 2.52 millions	€ 2.57 millions	€ 0.04 millions	1.7 %
Capital cost	€ 1.54 millions	€ 1.54 millions	€ 0	0 %

The cost distribution for the two cases compared to the starting case is shown in Figure 14.

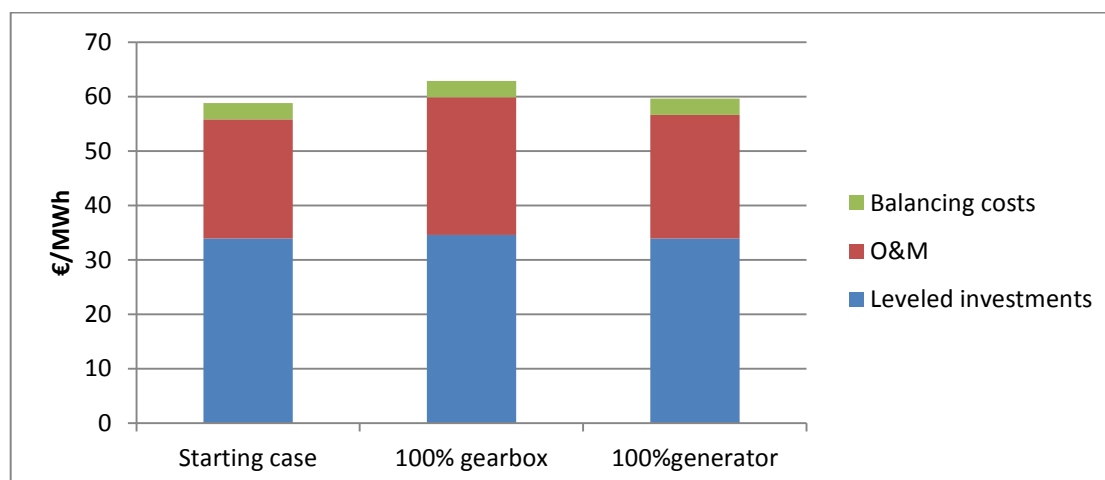


Figure 14: The calculated production cost (starting case, gearbox and generator +100%)

As seen in the first and second case, the results change moderately. A 20 percent change on the failure or a 100% percent change of the gearbox failure rate leads to a 7 percent change in cost per MWh. In the third case the change is relatively low, 1.4 percent.

This means that the most sensitive failure rates needs to be quite accurate if the model are going to give a trustworthy result. If there are single components, apart from the gearbox, that has been given an incorrect failure rate it will, in most cases, not affect the result significantly. If for some reason all components have been given failure rates that all are way off it can affect the result in both directions.

4.2.2 Interest rate

In the starting case the interest rate were set to 7 percent. In this case it will be decreased to 4 percent to see the impact on that change. The result is presented in the table below.

	Starting case Int. rate 7%	Starting case Int. rate 4%	Difference	Difference [%]
LCC per MWh	€ 58.9/ MWh	€ 54.5/ MWh	-€ 7.46/MWh	-7.6 %
Total cost	€ 8.76 millions	€ 7.66 millions	€ 1.11 millions	-13 %
Total production	149 GWh	149 GWh	0.06 GWh	0.0 %
Downtime	11 700 hours	11 700 hours	7 hours	0.1 %
Vessel cost	€ 2.52 millions	€ 2.53 millions	€ 0.003 millions	0.1 %
Capital cost	€ 1.54 millions	€ 0.883 millions	-€ 0.66 millions	43 %

This case is compared to the case where the interest rate was set to 7 percent in the column chart presented in Figure 15.

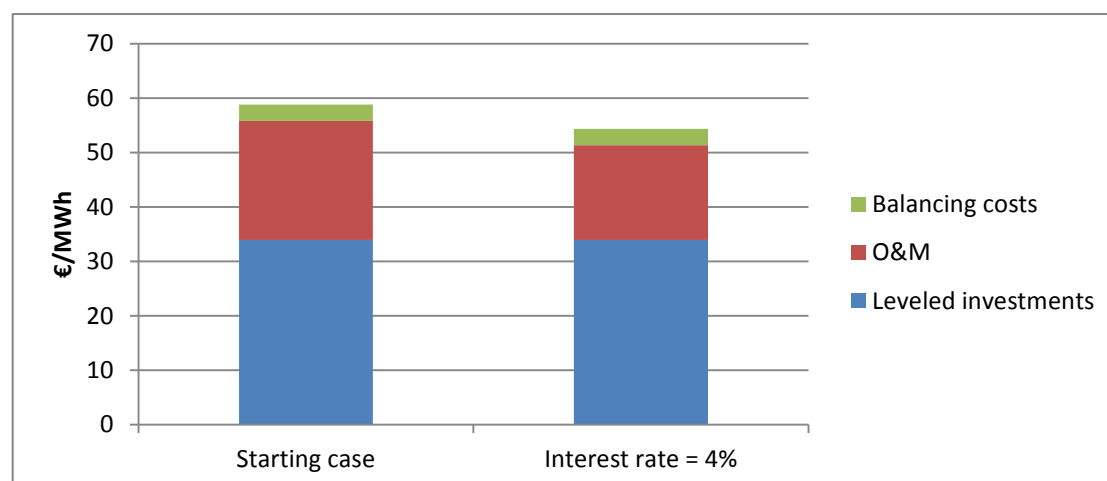


Figure 15: The calculated production cost (interest rate =7% and 4%)

The interest rate certainly affects the final result. An interest of four instead of seven percent changes the cost per MWh by about 8 percent.

4.2.3 Discount rate

In the last test the discount rate will be increased to 10 percent compared to the 7.5 percent in the starting case. The results are presented in the table below.

	Starting case Disc. rate 7.5%	Starting case Disc. rate 10%	Difference	Difference [%]
LCC per MWh	€ 58.9/ MWh	€ 53.8/ MWh	-€ 5.11/MWh	-8.7 %
Total cost	€ 8.76 millions	€ 8.01 millions	€ 0.752 millions	-8.6 %
Total production	149 GWh	149 GWh	0.16 GWh	0.1 %
Downtime	11 700 hours	11 600 hours	-49 hours	-0.4 %
Vessel cost	€ 2.52 millions	€ 2.52 millions	-€ 0.005 millions	-0.2 %
Capital cost	€ 1.54 millions	€ 1.36 millions	-€ 0.187 millions	-12 %

The distribution of the costs is presented in the column chart in Figure 16.

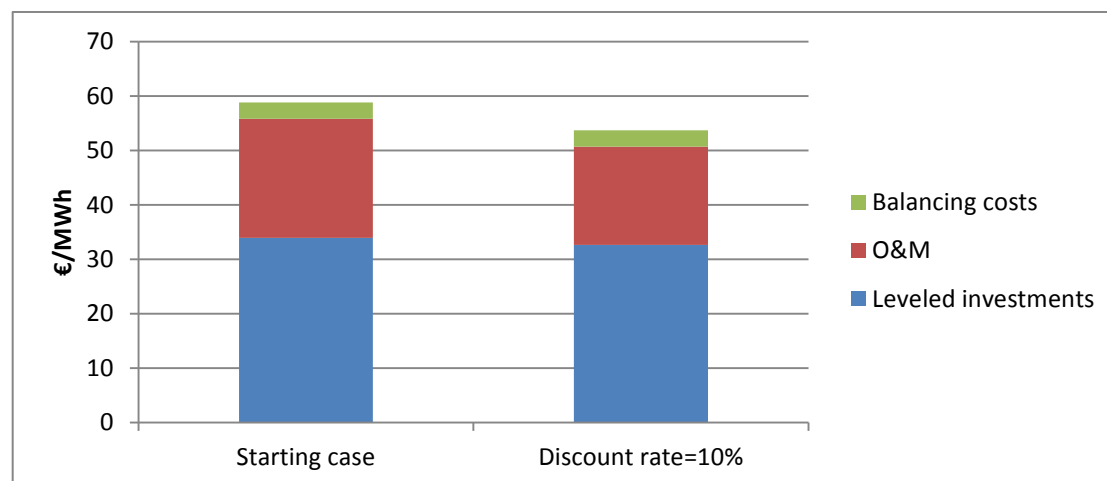


Figure 16: The calculated production cost (discount rate= 7.5% and 10%)

4.2.4 Inflation factor

An inflation factor on two percent is used in the model. By ditching the inflation factor one can see the impact of that parameter. The results are the following.

	Starting case Inflation 2%	Starting case Inflation 0%	Difference	Difference [%]
LCC per MWh	€ 58.9/ MWh	€ 55.6/ MWh	-€ 3.33/MWh	-5.6 %
Total cost	€ 8.76 millions	€ 8.27 millions	€ 0.490 millions	-5.6 %
Total production	149 GWh	149 GWh	0.08 GWh	0.1 %
Downtime	11 700 hours	11 600 hours	-10 hours	-0.1 %
Vessel cost	€ 2.52 millions	€ 2.52 millions	-€ 0.001 millions	-0.0 %
Capital cost	€ 1.54 millions	€ 1.54 millions	€ 0	0 %

The cost distribution is compared to the case where the inflation rate is two percent in the column chart below in Figure 17.

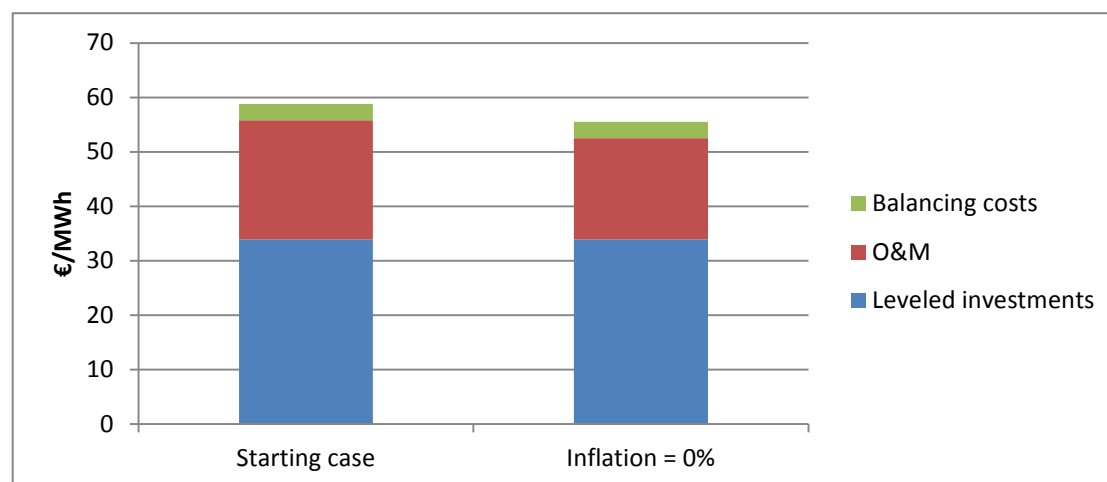


Figure 17: The calculated production cost (inflation rate= 2% and 0%)

As seen in the tables the inflation factor affects the interest which prepossesses the profit and the cost per MWh negatively by 5.6 percent. That seems quite obvious since the inflation rate increases all the costs that are carried out during the lifetime.

4.3 The results compared to data from offshore wind farms in operation

The results gained by the LCC model are presented in the column chart in Figure 18.

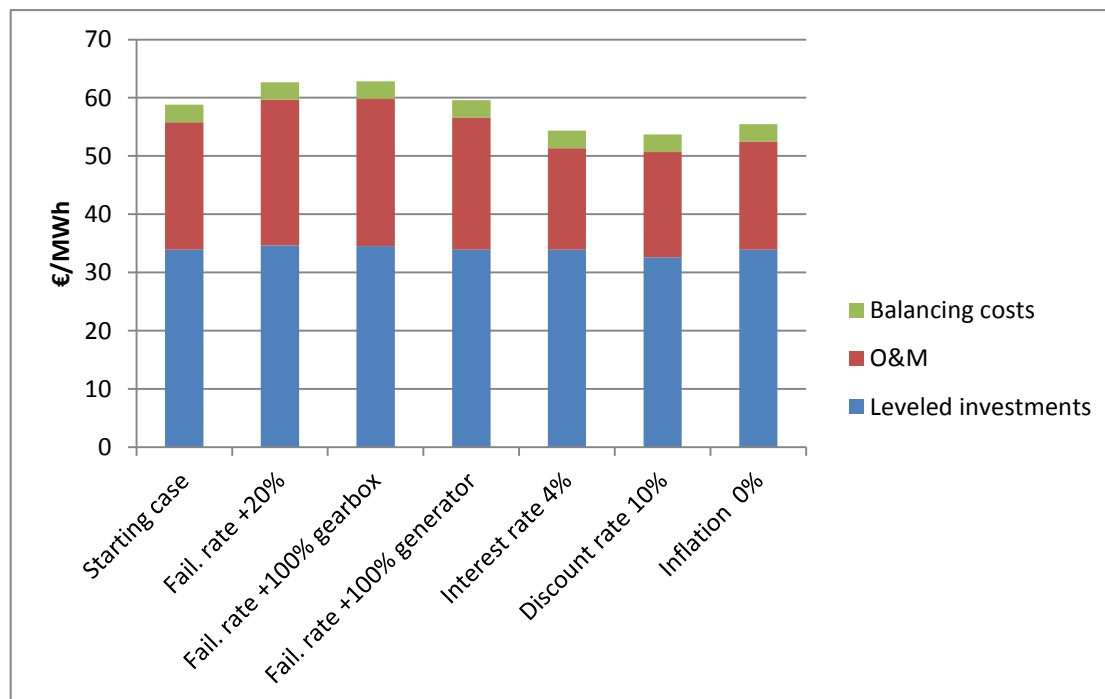


Figure 18: The calculated production cost for the different cases.

In Figure 19 is a re-creating of the calculated production costs for different offshore wind farm done by (Krohn, Morthorst, & Awerbuch, 2009). They have used a discount rate of 7.5 percent per annum, over the assumed lifetime of 20 years, which are equal to all cases, except the one where the discount rate was set to 10 percent.

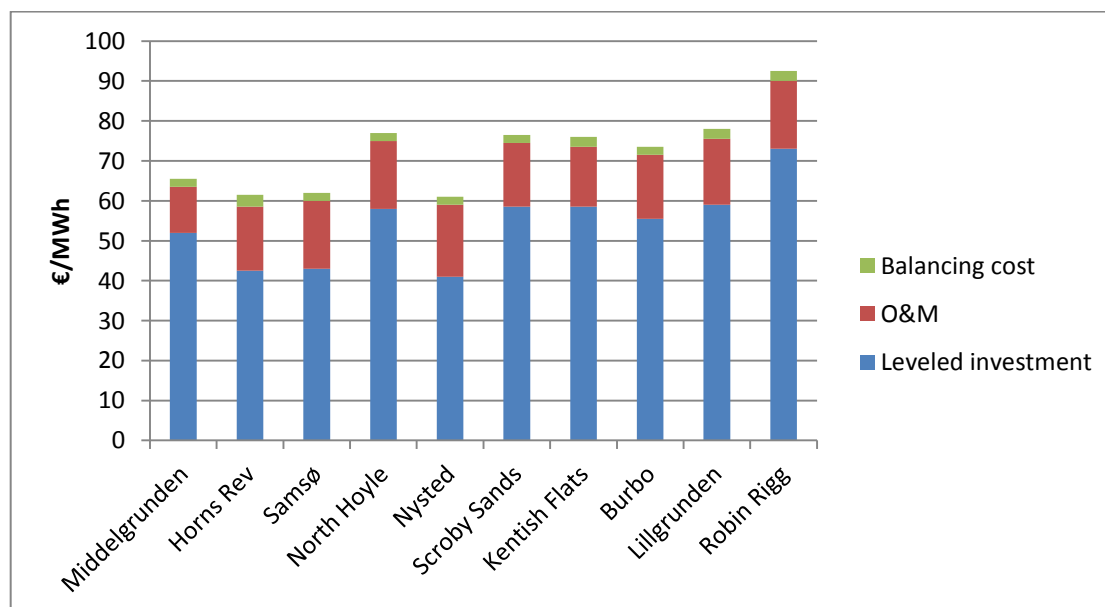


Figure 19: The calculated production cost for the different offshore wind farms.

The LCC model that was developed in this thesis was using failure rate data and balancing cost from Horns Rev and wind data from Egmond aan Zee. This means that it does not need to be comparable to the result from Horns Rev (Figure 19). The distribution between leveled investments and O&M should in some extent be similar. As one can see when comparing the results between Figure 18 and Figure 19 it is obvious that they are not similar. On the other hand the total LCC per MWh are quite similar. This could mean that the group O&M consist of costs that (Krohn, Morthorst, & Awerbuch, 2009) have included in leveled investment.

5 Conclusion and further work

The life cycle cost model developed in this master thesis is what can be called an alpha version, meaning that it is not fully developed yet. The wind turbine manufacturer, wind farm owners and service personnel are careful with their statistical data. This is because they are afraid that it will hit back on them if they are giving it away. For instance a report comparing different WT models can lead to massive losses for the manufacturer.

This issue aggravates the development of a LCC model since the data used in the model are inadequate. In order to make a full scale reliable model the cooperation with WT manufactures or wind farm owner is necessary.

With free access to statistical data for failure rates, reparation costs, downtime statistics and other statistics concerning O&M the model could be simplified and extended and result in more reliable results. As it is now, the model makes compromises and assumptions that might not be accurate.

However the results from the model are realistic and it includes a lot of tools that can be transferred to a new improved model. There are some parts in the model that needs to be improved, such as the installation cost, vessel costs and other fixed costs (warehouses, basic costs). The installation cost is the total investments to complete Horns Rev. If the model are going to be compatible for new wind farms it is important to know what the installation cost are built-up with. To improve the vessel cost, more accurate cost on the vessels, what kind of vessels needed on different kinds of reparations and how many reparations that can be made at the same time is needed. The cost of helicopter also needs to be considered to get an accurate O&M cost.

The econometrics of the model can be seen as the assessment of the total cost, and this is not the same as it should be for private investors. Private investors should need a profit that can be compared to the risk of establishing an offshore farm. This extra cost would be on top of the one calculated in the model.

When all improvements are done a graphic user interface would simplify the usage of the model. As it is today the user needs some Matlab knowledge to be able to use it. Altogether the model is a great tool that gives the user an idea of what costs, and their magnitude, that are included for a WT during its lifetime.

6 References

Reports

- Fredrik Ardefors, M. L. (2009). *Vindkraftverk – kartläggning av aktiviteter och kostnader vid nedmontering, återställande av plats och återvinning*. Technical report. Svensk Vindenergi.. Page 84
- Krohn, S., Morthorst, P.-E., & Awerbuch, S. (2009). *The Economics of wind energy*. A report by the European Wind Energy Association (EWEA). Pages 66-67
- Lindqvist, M., & Lundin, J. (2010). *Spare Part Logistics and Optimization for Wind Turbines - Methods for Cost-Effective Supply and Storage*. Master thesis. Uppsala: University of Uppsala. Pages 53,86
- McMillan, D., & Ault, G. (2007). *Towards Quantification of Condition Monitoring Benefit for Wind Turbine Generators*. Technical report. Institute for Energy & Environment, University of Strathclyde Page 5
- S. Faulstich, B. H. (2008). *Appropriate failure statistics and reliability characteristics*. Technical report. Fraunhofer Institute for Wind Energy and Energy System Technology (IWES), Kassel, Germany. Page 2
- The Crown Estate. (2009). *A guide to an offshore wind farm*. A report published on behalf of the Crown Estate. Page 65-66

Internet

- Wind Turbine Power Calculator*. (n.d.). Retrieved from Danish Wind Industry Association:
<http://guidedtour.windpower.org/en/tour/wres/pow/index.htm>
- NordzeeWind. (2006-2009). *NoordzeeWind.nl*. Retrieved from
<http://www.noordzeewind.nl/> ----[Knowledge / Reports and Data]----

Appendix A – Matlab code

Main program – failurecosts.m

```
clc;clear all;tic
%% Create a matrix that describes the probability of the total cost for a
% wind turbine during a lifetime
%% All constants and variables
load data
load prob
load simfail5000
% Preventive Maintenance: pm [maintenance per year]
pm=2;
years=20; % lifetime
not=80; % number of turbines
power=2000; % P=the power of the WT
tts=1; % tts= Time to get to site in hours

% Planned maintenance costs: pmc
% According to ref: [McMillan, 2007, Towards Quantification of Condition Monitoring
% Benefit for Wind Turbine Generators] the Maintenance cost is
% approximately £5000=€5800 per planned maintenance for a 2MW WT.
pmc=5800;
nom=2; % number of planned maintenance per year

% Total installation and production costs for a wind turbine: cost_inst
% According to Horns Rev, 80 offshore turbines costed €260 millions
cost_inst=270e6/80; % From http://www.vattenfall.dk/da/fakta.htm,

% cost to have supplies in warehouse: warehouse
warehouse=5.8e6*years/not; % ref: [The Crown Estate, A guide to an offshore wind farm
p 66]

% Decommission cost are approx. €125 000
decommission=125000;

% Basic expenses for a offshore park without reparations: basic_cost
basic_cost=30e6/not; % ref: [The Crown Estate, A guide to an offshore wind farm p 66]

% Annual energy production statistics calculated with no downtime: aep
aep=[8187083,8746177,7108901,8978100];

% Vessel rent: V_large, V_small
V_small=40e3; % ref:
[http://www.svenskvindenergi.org/files/Vindkraftverk_nedmontering.pdf]
```

The development of a life cycle cost model for an offshore wind farm

```
%% Interest cost for lifetime

% Interest rate on invested money per year: rate
rate=0.04;
% rate=0.05;

% inflation: infl
infl=0.02;

% discounting factor: DF
DF=1.075;

% Project time to build wind farm: inst_time [years]
inst_time=2;

%% Go from failure rates per hour to failure rates per year
for i=1:length(data)
    data(i,2)=1-(1-data(i,2)/1e6)^(365*24);
end

% Make a vector that compensates for bathtub. The failure data is only for
% the 9th year. Call the vector bath
% bath=[1.5 1.35 1.25 1.18 1.13 1.09 1.06 1.04 1.02 1 1 1.02 1.04 1.06 1.09 1.13 1.18
1.25 1.35 1.5];
bath=[2.24 1.71 1.59 1.47 1.44 1.38 1.12 1.15 1 0.79 0.79 1 1.15 1.12 1.38 1.44 1.47
1.59 1.71 2.24];
% bathpercent: is a factor that describes how many percent of the total
% failures that happens every year
bathpercent=bath/sum(bath);
% integrate bath to get the average probability for failure
p_bath=sum(bath)/length(bath);

% Capital cost: The installation cost is seen as a loan that will be payed
% over 20 years.
capital=cost_inst;
for i=1:years
    interest(i)=capital*rate;
    capital=capital-cost_inst/years;
end

hcc=[7]; % highly cost components

n=10000; % number of simulations
tot_prod=zeros(n,length(hcc)); % The total energy production for the lifetime of the
WT
total_costs=zeros(n,1); % Total cost vector/ matrix
dt=zeros(n,length(hcc)); % The total downtime for the WT, in hours
```

The development of a life cycle cost model for an offshore wind farm

```
data(:,2)=data(:,2)*p_bath;
tot_costs=zeros(n,length(hcc));
vessel=zeros(n,length(hcc));    % The total cost for vessel rent each day
restore=data(:,2);
for h=1:length(hcc)
    data(:,2)=restore;
    data(hcc(h),2)=data(hcc(h),2);
for i=1:length(data)
    for j=0:10
        c(i,j+1)=data(i,2)^j*(1-data(i,2))^(years-j)*nchoosek(years, j);
    end
end
end

for i=1:length(c(:,1))
    for j=1:length(c(1,:))
        p(i,j)=c(i,j)/sum(c(i,:));
    end
end
end
%surf(p)

%% Start the lottery
NoF=zeros(length(data(:,2)),1);

comp_cost=zeros(n,1);

% Number of days the Vessel is needed: Vdays
Vdays=[3 1 1 1 1 1 6 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 1
1 1 1];
dt_power=power*0.7;

for i=1:n
    for j=1:length(data(:,2))
        x=discretesample(p(j,:),1);
        if x>1
            for k=2:x
                z=discretesample(prob(j,:),1); % gets the downtime in days before
weather is good enough to repair the WT
                % down is the sum of lead time (data(j,5)), repair time(data(j,3)),
                % waiting time (for weather) (z) and time to go to site(tts).
                down=data(j,5)*24*7+data(j,3)+24*(z-1)+tts;
                dt(i,h)=dt(i,h)+down;
                sim=discretesample(simfail(j,:),1);
                vessel(i,h)=vessel(i,h)+V_small*Vdays(j)/sim;
            end
        end
        NoF(j)=NoF(j)+x-1;    %number of failure for each part
        comp_cost(i)=comp_cost(i)+(x-1)*data(j,1);
    end
end
```

The development of a life cycle cost model for an offshore wind farm

```
end
for l=1:years
    y=discretesample([.25 .25 .25 .25],1);
    tot_prod(i,h)=tot_prod(i,h)+aep(y);
end
tot_prod(i,h)=tot_prod(i,h)-dt_power*dt(i,h);

% Calculating the total cost
for q=1:years

total_costs(i)=total_costs(i)+interest(q)+(1+infl)^q*(warehouse+basic_cost)/years+pmc
*nom+bathpercent(q)*(vessel(i)+comp_cost(i))/(DF^q);
    capital=capital-cost_inst/years;
end

end

LCC=(total_costs+cost_inst+decommission/(1+DF)^years)./tot_prod;

% int is the step size (in €) when representing total cost/kWh
int=0.001;
curve=round(LCC(:)/int);
for i=1:max(curve)
    cpk(i,h)=length(find(curve==i));
end
cpk(:,h)=cpk(:,h)./sum(cpk(:,h));
cpkwh=LCC*1000;

end
toc
```

The discrete sampler - `disretesample.m`

```
function x = disretesample(p, n)
% Samples from a discrete distribution
%
%   x = disretesample(p, n)
%   independently draws n samples (with replacement) from the
%   distribution specified by p, where p is a probability array
%   whose elements sum to 1.
%
%   Suppose the sample space comprises K distinct objects, then
%   p should be an array with K elements. In the output, x(i) = k
%   means that the k-th object is drawn at the i-th trial.
%
% Remarks
% -----
%   - This function is mainly for efficient sampling in non-uniform
%     distribution, which can be either parametric or non-parametric.
%
%   - The function is implemented based on histc, which has been
%     highly optimized by mathworks. The basic idea is to divide
%     the range [0, 1] into K bins, with the length of each bin
%     proportional to the probability mass. And then, n values are
%     drawn from a uniform distribution in [0, 1], and the bins that
%     these values fall into are picked as results.
%
%   - This function can also be employed for continuous distribution
%     in 1D/2D dimensional space, where the distribution can be
%     effectively discretized.
%
%   - This function can also be useful for sampling from distributions
%     which can be considered as weighted sum of "modes".
%     In this type of applications, you can first randomly choose
%     a mode, and then sample from that mode. The process of choosing
%     a mode according to the weights can be accomplished with this
%     function.
%
% Examples
% -----
%   % sample from a uniform distribution for K objects.
%   p = ones(1, K) / K;
%   x = disretesample(p, n);
%
%   % sample from a non-uniform distribution given by user
%   x = disretesample([0.6 0.3 0.1], n);
%
%   % sample from a parametric discrete distribution with
```

The development of a life cycle cost model for an offshore wind farm

```
%      % probability mass function given by f.
%      p = f(1:K);
%      x = discretesample(p, n);
%
%
%% parse and verify input arguments
assert(isfloat(p), 'discretesample:invalidarg', ...
    'p should be an array with floating-point value type.');
```

```
assert(isnumeric(n) && isscalar(n) && n >= 0 && n == fix(n), ...
    'discretesample:invalidarg', ...
    'n should be a nonnegative integer scalar.');
```

```
%% main
% process p if necessary
K = numel(p);
if ~isequal(size(p), [1, K])
    p = reshape(p, [1, K]);
end
% construct the bins
edges = [0, cumsum(p)];
s = edges(end);
if abs(s - 1) > eps
    edges = edges * (1 / s);
end
% draw bins
rv = rand(1, n);
c = histc(rv, edges);
ce = c(end);
c = c(1:end-1);
c(end) = c(end) + ce;
% extract samples
xv = find(c);
if numel(xv) == n % each value is sampled at most once
    x = xv;
```

The development of a life cycle cost model for an offshore wind farm

```
else % some values are sampled more than once
    xc = c(xv);
    d = zeros(1, n);
    dv = [xv(1), diff(xv)];
    dp = [1, 1 + cumsum(xc(1:end-1))];
    d(dp) = dv;
    x = cumsum(d);
end

% randomly permute the sample's order
x = x(randperm(n));
```

Power calculator – powercurve.m

```
clear all;
load wind2006;
for i=1:length(final_curve(:,1))
    if final_curve(i,1)<4
        power(i,1)=final_curve(3,2);
    elseif final_curve(i,1)<5
        power(i,1)=final_curve(4,2)+(final_curve(i,1)-4)*(final_curve(5,2)-
final_curve(4,2));
    elseif final_curve(i,1)<6
        power(i,1)=final_curve(5,2)+(final_curve(i,1)-5)*(final_curve(6,2)-
final_curve(5,2));
    elseif final_curve(i,1)<7
        power(i,1)=final_curve(6,2)+(final_curve(i,1)-6)*(final_curve(7,2)-
final_curve(6,2));
    elseif final_curve(i,1)<8
        power(i,1)=final_curve(7,2)+(final_curve(i,1)-7)*(final_curve(8,2)-
final_curve(7,2));
    elseif final_curve(i,1)<9
        power(i,1)=final_curve(8,2)+(final_curve(i,1)-8)*(final_curve(9,2)-
final_curve(8,2));
    elseif final_curve(i,1)<10
        power(i,1)=final_curve(9,2)+(final_curve(i,1)-9)*(final_curve(10,2)-
final_curve(9,2));
    elseif final_curve(i,1)<11
        power(i,1)=final_curve(10,2)+(final_curve(i,1)-10)*(final_curve(11,2)-
final_curve(10,2));
    elseif final_curve(i,1)<12
        power(i,1)=final_curve(11,2)+(final_curve(i,1)-11)*(final_curve(12,2)-
final_curve(11,2));
    elseif final_curve(i,1)<13
        power(i,1)=final_curve(12,2)+(final_curve(i,1)-12)*(final_curve(13,2)-
final_curve(12,2));
    elseif final_curve(i,1)<14
        power(i,1)=final_curve(13,2)+(final_curve(i,1)-13)*(final_curve(14,2)-
final_curve(13,2));
    elseif final_curve(i,1)<15
        power(i,1)=final_curve(14,2)+(final_curve(i,1)-14)*(final_curve(15,2)-
final_curve(14,2));
    elseif final_curve(i,1)<16
        power(i,1)=final_curve(15,2)+(final_curve(i,1)-15)*(final_curve(16,2)-
final_curve(15,2));
    elseif final_curve(i,1)<17
        power(i,1)=final_curve(16,2);%+(final_curve(i,1)-16)*(final_curve(17,2)-
final_curve(16,2));
    elseif final_curve(i,1)<18
        power(i,1)=final_curve(17,2);%+(final_curve(i,1)-17)*(final_curve(18,2)-
final_curve(17,2));
    elseif final_curve(i,1)<19
        power(i,1)=final_curve(18,2);%+(final_curve(i,1)-18)*(final_curve(19,2)-
final_curve(18,2));
    elseif final_curve(i,1)<20
```



```
    power(i,1)=final_curve(19,2);%+(final_curve(i,1)-19)*(final_curve(20,2)-
final_curve(19,2));
    elseif final_curve(i,1)<21
        power(i,1)=final_curve(20,2);%+(final_curve(i,1)-20)*(final_curve(21,2)-
final_curve(20,2));
    elseif final_curve(i,1)<22
        power(i,1)=final_curve(21,2);%+(final_curve(i,1)-21)*(final_curve(22,2)-
final_curve(21,2));
    elseif final_curve(i,1)<23
        power(i,1)=final_curve(22,2);%+(final_curve(i,1)-22)*(final_curve(23,2)-
final_curve(22,2));
    elseif final_curve(i,1)<24
        power(i,1)=final_curve(23,2);%+(final_curve(i,1)-23)*(final_curve(24,2)-
final_curve(23,2));
    elseif final_curve(i,1)<25
        power(i,1)=final_curve(24,2);%+(final_curve(i,1)-24)*(final_curve(25,2)-
final_curve(24,2));
    elseif final_curve(i,1)<26
        power(i,1)=final_curve(25,2);%+(final_curve(i,1)-25)*(final_curve(26,2)-
final_curve(25,2));
    else
        power(i,1)=0;
    end
end
end
```

Wind speed analyser – wind.m

```
clear all;clc;
tic
load wind2006;
% load wind2007;
% load wind2008;
% load wind2009;
j=1;
windspeed=7;
k=1;
i=1;
n=10000;
storm7=sparse(n,1);
weather7=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather7(j)=weather7(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm7(k)=storm7(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

1
j=1;
windspeed=10;
k=1;
i=1;

storm10=sparse(n,1);
weather10=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather10(j)=weather10(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm10(k)=storm10(k)+1;
            i=i+1;
        end
        j=j+1;
```

```
        k=k+1;
    end

end

2
j=1;
windspeed=12;
k=1;
i=1;

storm12=sparse(n,1);
weather12=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather12(j)=weather12(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm12(k)=storm12(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

3
j=1;
windspeed=15;
k=1;
i=1;

storm15=sparse(n,1);
weather15=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather15(j)=weather15(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm15(k)=storm15(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end
```

```
end
4
j=1;
windspeed=18;
k=1;
i=1;

storm18=sparse(n,1);
weather18=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather18(j)=weather18(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm18(k)=storm18(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
5
j=1;
windspeed=20;
k=1;
i=1;

storm20=sparse(n,1);
weather20=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather20(j)=weather20(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm20(k)=storm20(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
```

```
6
weather706=weather7/6;
storm706=storm7/6;

weather1006=weather10/6;
storm1006=storm10/6;

weather1206=weather12/6;
storm1206=storm12/6;

weather1506=weather15/6;
storm1506=storm15/6;

weather1806=weather18/6;
storm1806=storm18/6;

weather2006=weather20/6;
storm2006=storm20/6;
%% 2007
load wind2007;
j=1;
windspeed=7;
k=1;
i=1;

storm7=sparse(n,1);
weather7=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather7(j)=weather7(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm7(k)=storm7(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
107
j=1;
windspeed=10;
k=1;
i=1;
```

```
storm10=sparse(n,1);
weather10=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather10(j)=weather10(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm10(k)=storm10(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
2
j=1;
windspeed=12;
k=1;
i=1;

storm12=sparse(n,1);
weather12=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather12(j)=weather12(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm12(k)=storm12(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
3
j=1;
windspeed=15;
k=1;
i=1;

storm15=sparse(n,1);
```

```
weather15=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather15(j)=weather15(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm15(k)=storm15(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
4
j=1;
windspeed=18;
k=1;
i=1;

storm18=sparse(n,1);
weather18=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather18(j)=weather18(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm18(k)=storm18(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
5
j=1;
windspeed=20;
k=1;
i=1;

storm20=sparse(n,1);
weather20=sparse(n,1);
while i<length(final_curve(:,1))
```

```
    if final_curve(i,1)<windspeed
        weather20(j)=weather20(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm20(k)=storm20(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
6
weather707=weather7/6;
storm707=storm7/6;

weather1007=weather10/6;
storm1007=storm10/6;

weather1207=weather12/6;
storm1207=storm12/6;

weather1507=weather15/6;
storm1507=storm15/6;

weather1807=weather18/6;
storm1807=storm18/6;

weather2007=weather20/6;
storm2007=storm20/6;

%% 2008
load wind2008;
j=1;
windspeed=7;
k=1;
i=1;

storm7=sparse(n,1);
weather7=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather7(j)=weather7(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
```



```
        while final_curve(i,1)>=windspeed
            storm7(k)=storm7(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end

end

108
j=1;
windspeed=10;
k=1;
i=1;

storm10=sparse(n,1);
weather10=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather10(j)=weather10(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm10(k)=storm10(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end

2
j=1;
windspeed=12;
k=1;
i=1;

storm12=sparse(n,1);
weather12=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather12(j)=weather12(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm12(k)=storm12(k)+1;

```

```
        i=i+1;
    end
    j=j+1;
    k=k+1;
end

end

3
j=1;
windspeed=15;
k=1;
i=1;

storm15=sparse(n,1);
weather15=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather15(j)=weather15(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm15(k)=storm15(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end

4
j=1;
windspeed=18;
k=1;
i=1;

storm18=sparse(n,1);
weather18=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather18(j)=weather18(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm18(k)=storm18(k)+1;
            i=i+1;
        end
    end
end
```

```
        j=j+1;
        k=k+1;
    end

end

5
j=1;
windspeed=20;
k=1;
i=1;

storm20=sparse(n,1);
weather20=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather20(j)=weather20(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm20(k)=storm20(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end

6
weather708=weather7/6;
storm708=storm7/6;

weather1008=weather10/6;
storm1008=storm10/6;

weather1208=weather12/6;
storm1208=storm12/6;

weather1508=weather15/6;
storm1508=storm15/6;

weather1808=weather18/6;
storm1808=storm18/6;

weather2008=weather20/6;
storm2008=storm20/6;
```

The development of a life cycle cost model for an offshore wind farm

```
%% 2009
load wind2009;
j=1;
windspeed=7;
k=1;
i=1;

storm7=sparse(n,1);
weather7=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather7(j)=weather7(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm7(k)=storm7(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
109
j=1;
windspeed=10;
k=1;
i=1;

storm10=sparse(n,1);
weather10=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather10(j)=weather10(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm10(k)=storm10(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
2
```

```
j=1;
windspeed=12;
k=1;
i=1;

storm12=sparse(n,1);
weather12=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather12(j)=weather12(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm12(k)=storm12(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

3
j=1;
windspeed=15;
k=1;
i=1;

storm15=sparse(n,1);
weather15=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather15(j)=weather15(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm15(k)=storm15(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

4
j=1;
windspeed=18;
```

```
k=1;
i=1;

storm18=sparse(n,1);
weather18=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather18(j)=weather18(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm18(k)=storm18(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
5
j=1;
windspeed=20;
k=1;
i=1;

storm20=sparse(n,1);
weather20=sparse(n,1);
while i<length(final_curve(:,1))
    if final_curve(i,1)<windspeed
        weather20(j)=weather20(j)+1;
        i=i+1;
    elseif final_curve(i,1)>=windspeed
        while final_curve(i,1)>=windspeed
            storm20(k)=storm20(k)+1;
            i=i+1;
        end
        j=j+1;
        k=k+1;
    end
end

end
6
weather709=weather7/6;
storm709=storm7/6;

weather1009=weather10/6;
```

storm1009=storm10/6;

weather1209=weather12/6;

storm1209=storm12/6;

weather1509=weather15/6;

storm1509=storm15/6;

weather1809=weather18/6;

storm1809=storm18/6;

weather2009=weather20/6;

storm2009=storm20/6;

toc


```
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather10(find(weather10>144)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);
sum(weather20(find(weather20>5)))/(4*365*24);

%% These loops makes an vector with the probability for downtime in days
days=365;
for i=1:length(p)
    for j=1:days
        prob1(i,j)=p(i)*(1-p(i))^j;
    end
end

%% These loops normalises the probability vector
for i=1:length(p)
    for j=1:days
        prob(i,j)=prob1(i,j)./sum(prob1(i,:));
    end
end
```

Probability of simultaneous failures – vessels.m and simfails.m

vessels.m:

```
clc;clear all;

load prob
load data
not=80; % number of WT in wind farm
%% Go from failure rates per hour to failure rates per day
for i=1:length(data)
    fail_rate(i,2)=1-(1-data(i,2)/1e6)^(80*24);
    fail_rate(i,1)=1-fail_rate(i,2);
end

%% Tests for each failure if there are any simultaneous failures and if
% how many.
n=100;
z=zeros(length(fail_rate(:,1)),n);
```

```
tic
for m=1:n
    for i=1:length(fail_rate(:,1))
        x(i)=discretedsample(prob(i,:),1);
        for k=1:length(fail_rate(:,1))
            r=discretedsample(prob(k,:),1);
            if r<x(i)
                for j=1:r
                    y(i,j,k)=discretedsample(fail_rate(k,:),1);
                    if y(i,j,k)==2
                        z(i,m)=z(i,m)+1;
                    end
                end
            end
        end
    end
end

end
toc
```

simfails.m

```
clc;clear all;
load simfailure5000

% A matrix with probability for simultaneous failures

for i=1:length(z(:,1))
    for j=1:max(max(z))+1
        simfail(i,j)=length(find(z(i,)==j-1));
    end
end

simfail=simfail/length(z(1,:))
```