EE3065TU:
Reliability of Sustainable Power Systems
(Python Version)

Practicum Manual

By: dr.ir. Bart W. Tuinema, dr.ir. José L. Rueda Torres, Eduardo Jerez

March, 2022

Contents

Introduction

This manual contains the questions for the practicum of EE3065TU: Reliability of Sustainable Power Systems. During the practicum, you will apply the theory of the lectures in practice. The main objective of this practicum are:

1. To improve your understanding of the theory from the lectures and course reader
2. To develop practical experience in Python with reliability modelling of power systems
3. To apply reliability models for small and large systems as described in the course

This practicum thereby mainly supports the learning objectives 2-4 of this course: collecting and modelling of failure statistics (LO2), evaluation of the reliability of small and large systems (LO3) and definition of countermeasures to mitigate reliability threats (LO4). The practicum consists of six sessions plus a Python tutorial:

0. Python tutorial
1. Failure statistics
2. Reliability networks
3. Markov models
4. Generation system adequacy
5. State enumeration
6. Monte Carlo simulation

In figure 1, it is shown how the practicum session are related to the chapters of the course reader. In each session, some models as described in the reader will be applied in practice to study the reliability of the power system. In each session, the possibilities to improve the reliability and to reduce the risk will be discussed as well.

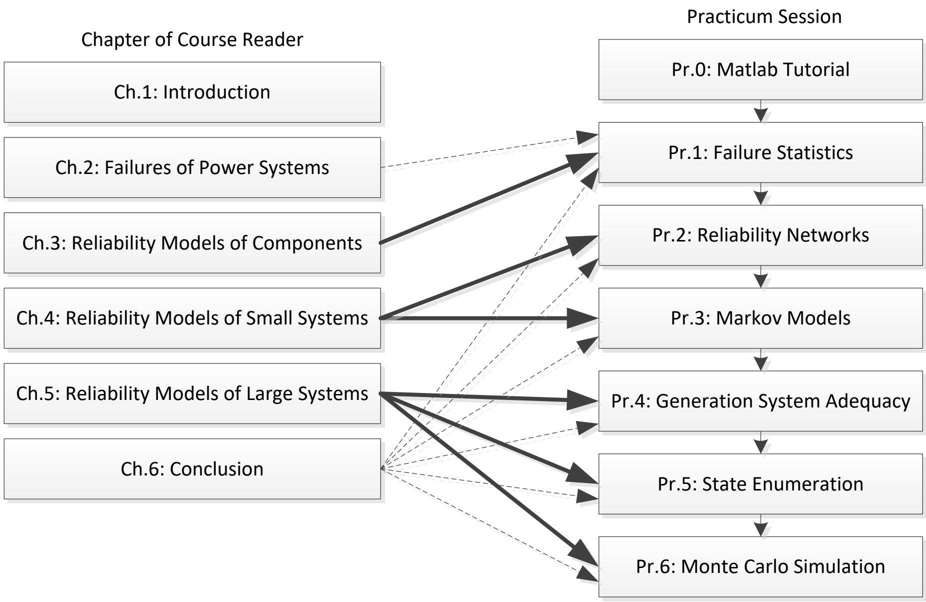


Figure 1: Overview of course reader chapters and practicum sessions.

For the practicum, the .csv files are available to be downloaded from the Practicum Section (i.e. go to *Content* > *Practicum*) of the Brightspace site of the course. These files are mainly a PDF document for the "Python Tutorial" and a ZIP file – "Practicum Files (zip)"- containing all the necessary data files (.csv) and Python script files (.py) for the remaining practicums. In the description of the practicum sessions, it is referred to specific files of these practicum files.

During the practicum, you can work in a group with up to 4 students on the assignments. Students must register in Brightspace (i.e. go to *Collaboration* > *Groups*) in one group before 31st Aug, 2021. After each of the practicum sessions 1-6, you will write short reports of your findings in groups of up to 4 students and submit it via the Assignments section of the course on Brightspace before the previous day of the next practicum session (i.e. Wednesday 5:00 pm). The grades along with the solutions to the practicum will be published a week later. These reports contain the calculations, results and figures you made, together with your conclusions and a short reflection on your work. You don't need to write a report about the Python tutorial. The reports will be graded on a scale of 1-10 and the average grade will count for 30% of the final grade for this course. The grading points for each practicum assignment is provided in the manual for each assignment question.

We hope you will enjoy this practicum. In case of any further questions, don't hesitate to ask the responsible instructors:

José L. Rueda Torres (j.l.ruedatorres@tudelft.nl)
Simon Tindemans (S.H.Tindemans@tudelft.nl)

Teaching Assistants:
Sattama Datta (S.Datta-7@student.tudelft.nl)
Shivesh Choudhary (S.choudhary@student.tudelft.nl)
Eduardo Jerez (E.E.Jerez@student.tudelft.nl)

Session 0: Python tutorial

For the purposes of introducing you to the Python Programming Language, we provide you with a Python reader developed by Prof. Jacco Hoekstra for the Faculty of Aerospace Engineering, which the professors over at AE have kindly allowed us to share with you and use for this course. You can also order this reader in print form on https://onlinereaders.tudelft.nl/index.php?orderableObject=22514484

Alternatively, you can go to the TU Delft Readers' Catalogue, select the Luchtvaart en Ruimtevaart faculty, BSc 1st Year, and download "AE1205 Programming and Scientific Computing in PYTHON - 2022".

We highlighted the chapters and sections of the reader that you might need or find useful.

- Chapter 1 has instructions on how to set up Python along with its package installer (PIP) and an IDE that you can use to make, edit and run code. Please for this course install Anaconda to set up your environments and Spyder IDE, see the page below.
- Chapters 2 through 6 will show you the fundamentals that you need to know in order to use Python to write functional code: types and variables, syntax, how to use modules and how to define your own functions.
- Chapter 9 instructs you on how to use the matplotlib module to make graphs and charts.
- Chapter 11 shows you how to use SciPy and Numpy, which you'll find very handy for making powerful code in Python.
- Chapter 12 brings a quick look into some elements of Object Oriented Programming that you might find useful.
- For information on how to use Pandas, you can look up online or Python Data Science Handbook by Jake VanderPlas. If you look it up on Google Scholar while connected to the TU Delft WiFi or while using the TU Delft VPN, you should be able to download it.

Session 1: Failure statistics

In this practicum session, you will analyse some failure statistics (i.e. failure frequencies and repair times). Statistical averages will be calculated, but also confidence intervals will be studied. For the repair times, boxplot modelling and approximation by the (negative) exponential distribution will be studied as well.

This practicum session is mainly related to sections 2.2, 3.2 and 3.3 of the course reader (i.e. failure statistics, failure frequency, repair time, component life cycle). Appendix A can be used to refresh your knowledge of probability and statistics (e.g. statistical mean, boxplots, probability). You'll at least need the math module and the curvefitting tool from SciPy for this practicum.

Assignments:

In the Netherlands, failure statistics of components of the power grid are collected in the NESTOR (NEderlandse STORingsregistratie) database. Suppose that failure statistics of overhead lines (OHLs) are collected for 5 years. 25 failures in EHV (380/220 kV) OHLs and 51 failures in HV (150/110 kV) overhead lines were reported. The average total OHL circuit length (averaged over the 5 studied years) is 2310 km for the EHV network and 3329 km for the HV network.

1. Calculate the failure frequency (*f*, in /cctkm·yr, per circuit-kilometre-year) for EHV OHLs and HV OHLs. Which one is the largest? Calculate the failure frequency for EHV/HV OHLs in general (i.e. combined). Also find the MTBF. [1 point]

According to [1], the confidence interval of a failure rate can be calculated by:

$$\frac{\chi^2_{1-\alpha/2}(2F)}{2T} \leq \lambda \leq \frac{\chi^2_{\alpha/2}(2F+2)}{2T} \qquad (1)$$

Where:
λ = average failure rate [/cctkm·yr] or [/comp·yr]
T = total considered time length (component-years) [cctkm·yr] or [comp·yr]
F = statistical number of failures within T [-]
α = significance level [-]
χ^2 = Chi-square distribution. *SEE BELOW FOR INSTRUCTIONS ON HOW TO USE THIS IN YOUR CODE*
1-α/2 or α/2 = probability
2F or 2F+2 = degrees of freedom

Figure 2 shows the χ^2-distribution. For equation (1), the inverse cdf of the χ^2-distribution is used to calculate for what value the area of the tail equals the value $\alpha/2$ (or $1-\alpha/2$), using the given degrees of freedom ($2F$ or $2F+2$).

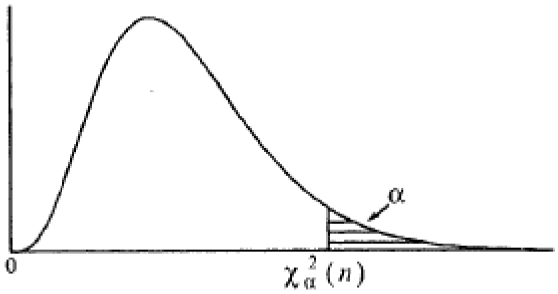


Figure 2: χ^2 distribution [1].

You'll need the chi2.ppf function for this. In order to use it, you need to include "*from scipy.stats.distributions import chi2*" (without the quotation marks) at the top of your file. The function is used as follows: $X^2_{1-\alpha/2}(2F)$ is written: *chi2.ppf(alpha/2, 2*F)*

2. Assuming that $f \approx \lambda$ (because MTTR<<MTTF), calculate the 95% confidence intervals (α=0.05) for the failure frequencies of EHV, HV, and EHV/HV OHLs. Plot the results as a boxplot (using matplotlib) in the practicum files (or MS Excel). What can you say about the difference between EHV and HV OHLs? [1 point]

Three years later, there are more failure statistics available. Suppose we have now 42 EHV OHL failures and 167 HV OHL failures. The average total OHL circuit lengths (averaged over the 8 studied years) are now 2471 km for the EHV network and 4078 km for the HV network.

3. Calculate the failure frequencies of EHV, HV, and EHV/HV OHLs again, together with their 95% confidence intervals. Also plot the results as a boxplot (or MS Excel). What can you say about the difference between EHV and HV OHLs? Can you think of possible reasons for this difference? [1 point]

In the practicum files, you can find a list of repair times for EHV OHLs (repair_times_OHL.csv).

4. Open the file with repair times. What is the average repair time? What is the minimum/maximum repair time? [1 point]
   NOTE: To load the data from the .csv file into Python, you need to "import pandas as pd" and then use the following chunk of code:
   df = pd.read_csv('repair_times_OHL.csv')
   data = df['Repair Times'].values.tolist()

5. Draw a boxplot of the data in Python. What are the 1st-quartile, median and 3rd-quartile values? What can you say about the average repair time in comparison with the boxplot values? [1 point]

We will now try to fit an exponential distribution to the repair time data. In Python, this can be done with the function curve_fit included in SciPy.

In order to use curve_fit, you have to include "*from scipy.optimize import curve_fit*" at the top of your file. The curve_fit function needs you to first define a mathematical function. You'll likely need Numpy for this, include it by adding "*import numpy as np*" at the top of your code too. For an exponential function, for example, this would be:

def exponential(t, a, b, c):

*return a*np.exp(-b*np.asarray(t))+c*

$$F(t) = e^{-t/T_1} \quad (2)$$

Where:
$F(t)$ = fraction of unrepaired components (1=no repaired components, 0=all components repaired) [-]
t = time [h]
T_1 *(-b)* = repair time (according to the exponential distribution) [h]

a = coefficient of the exponential term

c = independent term

You'll then need to call the curve_fit function in the following manner:

data.reverse() #the data as extracted from the .csv will likely need to be reversed

Ft = np.arange(0,1.00001,1/(len(data)-1)) #this sets up a range of values for the repair time axis

x_axis = np.arange(0,data[0]+0.00001,0.1) #sets up an x_axis

params, covariances = curve_fit(exponential, data, Ft) #you only really care about the params, which are the variables you set in the mathematical function you defined

plt.scatter(data, Ft)

*plt.plot(x_axis, exponential_linear_comb(x_axis, *params), '--') #this will compare the data and the curve produced by the curve_fit function so you can judge how well it fits*

6. With the exponential curve fit, we are trying to find T_1 *(-b)*. What should we do with the repair time data to create vectors for the time t and $F(t)$ *(-b)* in equation (2)? [1 point]

7. Try other functions to fit the curve. You can try a 3rd or 4th degree polynomial, for example. [1 point]

It is also possible to make a linear combination of two functions. Try making a linear combination of two exponential functions:

$$F(t) = c_1 \cdot e^{-t/T_1} + c_2 \cdot e^{-t/T_2} \quad (3)$$

8. Fit this linear combination to the repair time data. What are the values of c_1, c_2, T_1, and T_2? What can you conclude from T_1 and T_2? What does is mean if c_1 and c_2 do not add up to 1? Does this function fit the data better? What would be the consequences for reliability analysis of larger networks if this model is used? [2 points]

9. Which is, in your opinion, the best representation/model for the repair times? The average, the boxplot, the single exponential, the two exponentials, something different? What are the consequences for reliability analysis of large networks for these models? [1 point]

Write a small report (in a group of up to 4 students) about this practicum session.

Session 2: Reliability networks

In this practicum session, reliability networks will be applied to study the reliability of small systems. This practicum session is mainly related to section 4.2 of the course reader. Python will be used for simple mathematical calculations in this practicum session.

Assignments:

In this assignment, the reliability of EHV underground cables (UGCs) and overhead lines (OHLs) will be compared. For the UGC circuit, the Randstad380 configuration is used. Normally, overhead transmission lines are installed in double circuits. One circuit consists of three phases. As will be discussed in Introduction to Electrical Power Engineering, the voltage of each phase follows a sinusoidal, but there is a (time) lag of 120 degrees between the voltages of the three phases. The photo on the left shows a double circuit overhead line. On each side of the tower, there is one circuit consisting of three phases. Failure frequencies of overhead lines are normally given in /cctkmy, meaning: per (3-phase) circuit-kilometre-year.

Whereas underground cable circuits up to 220 kV consist of single three-core cables, an underground cable circuit of 380 kV consists of three separate cables. This provides better isolation and cooling of the cables. Because in the Randstad380 configuration a higher transmission capacity was desired, it was decided to use two separate cables per circuit phase. In total, this gives thus 6 separate cables per circuit. An underground cable circuit consists of cable parts connected by joints, and with terminations at the beginning and end. The photo on the right shows a single underground cable circuit consisting of 6 separate cables, together with a set of joints.

The failure frequencies of the cable parts are normally given in /cctkmy, meaning per (3-phase or 3 separate cables) circuit-kilometre-year. Because in Randstad380 two separate cables are used per circuit phase, this value must be doubled. Failure frequencies of joints and terminations are normally given in /compy, meaning per (single) component-year. In the Randstad380 configuration, joints and terminations are installed in sets of 6. The failure frequencies as shown in the table below can be used for the calculations [2]. The repair time of OHLs is 8 h and the repair time of UGCs 730 h (1 month).

Component	Failure frequency	Unit	Repair time	Unit
EHV OHL	0.00220	/cctkmy	8	h
EHV cable	0.00120	/cctkmy	730	h
EHV joint	0.00035	/compy	730	h
EHV termination	0.00168	/compy	730	h

1. Calculate the failure frequency and unavailability of an EHV OHL circuit of 10 km. In this assignment, we only consider independent failures. If we have a double circuit, what are then the probabilities of having 0, 1 or 2 circuit(s) available? How many hours/year is this? [2 points]

2. Draw the layout of a 10 km UGC circuit with Randstad380 configuration. The cable part length is 800 m. In this assignment, we only consider independent failures. What are the failure frequency and unavailability of this circuit? If we have a 10 km double circuit UGC, what are then the probabilities of having 0, 1 or 2 circuit(s) available? How many hours/year is this? If you compare the results for OHLs and UGCs, what can you conclude? [3 points]

In reality, double circuits can fail independently and dependently. For independent double circuit failures, first one circuit must fail, then the other circuits must fail during the repair time of the first circuit. For the frequency of independent double circuit failures, we can write:

$$f_{2ind} = 2 \cdot U_1 \cdot f_1$$

Where U_1 is the unavailability and f_1 the failure frequency of a single circuit. For dependent double circuit failures, we can define a dependent failure factor (c_{cc}) and write:

$$U_{2dep} = c_{cc} \cdot U_1$$

$$f_{2dep} = c_{cc} \cdot f_1$$

3. We can assume that the dependent failure factor is $c_{cc} = 0.1$. If you perform these calculations for UGCs and OHLs and compare the results, what can you conclude then? Do you think the dependent failure factor for EHV underground cables would be higher or lower than the dependent failure factor for overhead lines in reality? [3 points]

4. If one wishes to improve the reliability of the UGC connection, what are then possible solutions? Study the effect of one such solution by changing the input parameters of your calculation. What are your recommendations for UGC design? [2 points]

Write a small report (in a group of up to 4 students) about this practicum session.

Session 3: Markov models

In this practicum session, Markov models will be applied to study the reliability of small systems. This practicum session is mainly related to section 4.3 of the course reader. Python will be used for simple mathematical calculations and the solution of a matrix equation in this practicum session.

Assignments:

In the previous practicum session, we studied and compared the reliability of overhead line (OHL) and underground cable (UGC) connections with reliability networks. In this practicum session, we will study and compare the reliability of OHL and UGC connections again with Markov models. It will be analysed whether the reliability of an UGC connection can be improved by installing a spare cable.

Note that to solve a system in matrix form, you'll need to use include import numpy (as np) to your file and use the function *C = np.linalg.solve(A_mat, B_vect)*

1. In this assignment only independent failures are considered. Draw the (3-state) Markov model of a 10 km double circuit OHL. Clearly indicate in the figure the names of the states, failure rates, repair rates, and which states lead to 0/1/2 available circuits. Create the state transition matrix in Python (please remember that all λs and μs must have the same dimension). Solve the state transition matrix equation and calculate the probabilities of having 0, 1, or both circuits available (also in h/y). Also, calculate the state transition frequencies. Compare the results with your results from the previous practicum session. [2 points]

2. Similar for the 10 km UGC double circuit: Draw the (3-state) Markov model, create and solve the state transition matrix in Python. Again, only independent failures are considered. Based on example 4.8 in the course reader, you can consider one UGC circuit as one component. Compare the results with your results from the previous practicum session. [2 points]

3. To increase the reliability of the UGC double circuit, we decide to install a spare UGC circuit which can be used if one of the circuits fails. Draw a Markov model for the UGC double circuit with spare circuit (clearly indicate which states lead to 0/1/2 available circuits). Assume a switching time of 24 h for the spare circuit. For this assignment, it is best to consider the system as consisting of three components: circuit 1, circuit 2 and the spare circuit. [2 points]

4. Create the state transition matrix in Python. Solve the state transition matrix equation and calculate the probabilities of having 0, 1, or both circuits available (also in h/y). Compare your results with the results from questions 1&2. Does the spare cable increase the reliability of the double circuit UGC much? [2 points]

5. Would it make a difference if dependent failures were also considered in the analysis? Can you think of any practical complications of using an UGC configuration with a spare cable? [2 points]

Write a small report (in a group of up to 4 students) about this practicum session.

Session 4: Generation system adequacy

In this practicum session, the reliability of a generation system will be studied in a generation adequacy analysis. This practicum session is mainly related to section 5.3 of the course reader. Appendix C of the reader gives an overview of some reliability indicators. In this assignment, Python (or MS Excel) will be used to create a Capacity Outage Probability Table (COPT) and to calculate several reliability indices.

Assignments:

1. In examples 5.6 and 5.7 in the course reader, a COPT of a generation system consisting of four generators is created. In example 5.8 in the course reader, this COPT is used to calculate several reliability indicators. In the practicum files, you can find the file ‘COPT_4gens.py’. This script calculates the COPT of the course reader example. You can also find the COPT in the Excel ‘COPT_4gens.xlsx’. Perform the calculation as described in example 5.8 to see whether you arrive at the same results. [1 point]

2. Add a fifth generator (with capacity 400 MW and unavailability 0.05) to the generation system. Create the new COPT (in MS Excel or Python) by following the algorithm as described in section 5.3.2 of the course reader. Plot the second column and compare this with the graph shown in example 5.6 in the course reader. What are the differences? Calculate the reliability indicators of example 5.8 again. Has the reliability of the generation system improved much? [2 points]

3. In the practicum files, there is also a function in the ‘COPT_function.py’ file, which calculates the COPT using the algorithm as described in the course reader. The inputs of this function are the generator capacities (Cgens), the generator unavailabilities (Ugens), and a rounding value to which the generator capacities are rounded (P_round). This rounding value also determines the step size of the first column of the COPT. Verify whether this function works for the generation system consisting of 5 generators by comparing the results of this question with the results of the previous question. [1 point]

4. We will now perform a generation adequacy analysis of a generation system with about the size of the Dutch generation system. In the practicum files, you can find the data files “Cgens_Ugens.csv” and “load_Pwind.csv”, containing the capacities and unavailabilities of the generators, an hourly load scenario and three offshore wind scenarios (see below how to load them). What is the size of the smallest and largest generator? What are the smallest and largest unavailabilities? What is the total installed capacity of the generation system? What is the peak load? Create the COPT of this generation system (you can round the capacities to 50 MW (e.g. P_round = 50)) and plot the second column. What do you see in this graph? What can you conclude from this graph? [1 point]

5. We will now calculate the reliability indicators LOLP/LOLE for this generation system. Write a Python script that calculates the LOLP/LOLE for each hour of the year using the COPT of the previous question and the load scenario (*NLh_load* see below for instructions). What are the values of LOLP and LOLE? Plot a graph of the load scenario and the LOLP/LOLE per hour of the year. If you compare the graph of the LOLP/LOLE with the graph of the load, what can you conclude? [2 points]

6. If offshore wind is added to the system, this will have an impact on the calculated reliability indices. The load_Pwind.csv data file contains three offshore wind scenarios (see below for instructions). The first is the base scenario. The other scenarios are basically the same, but the second is shifted 24 h in time, while the third scenario is shifted 48 h in time. Consider the first scenario. What is the total installed capacity of the offshore wind? What is the capacity factor of the offshore wind? Subtract the wind production from the system load and calculate the LOLP/LOLE again. What are the values of the LOLP/LOLE now? [1 point]

7. It is interesting to calculate the capacity credit of wind energy. As described in the course reader, one definition of capacity credit of wind is: *“The amount of load that can be added to the system if wind capacity is installed, while keeping the same level of reliability.”* This is also known as the ‘load carrying capability’ of wind energy. Based on this definition, how can we calculate the capacity credit of wind energy in this case? Perform the calculation. What is the capacity credit (in MW and as % of the peak load)? Does it make a difference if the other wind scenarios (i.e. the scenarios that are 24 h and 48 h shifted in time) are used, and why? And how can we avoid this? [2 points]

In order to read the data for the dutch generation system, you’ll need to “*import pandas as pd*” at the top of your file, and then use the following chunk of code:

```
df = pd.read_csv('Cgens_Ugens.csv')
Cgens = df['Cgens'].values.tolist()
Ugens = df['Ugens'].values.tolist()

df = pd.read_csv('load_Pwind.csv')
NLh_load = df['NLh_load'].values.tolist()
P_off_wind1 = df['P_off_wind1'].values.tolist()
```

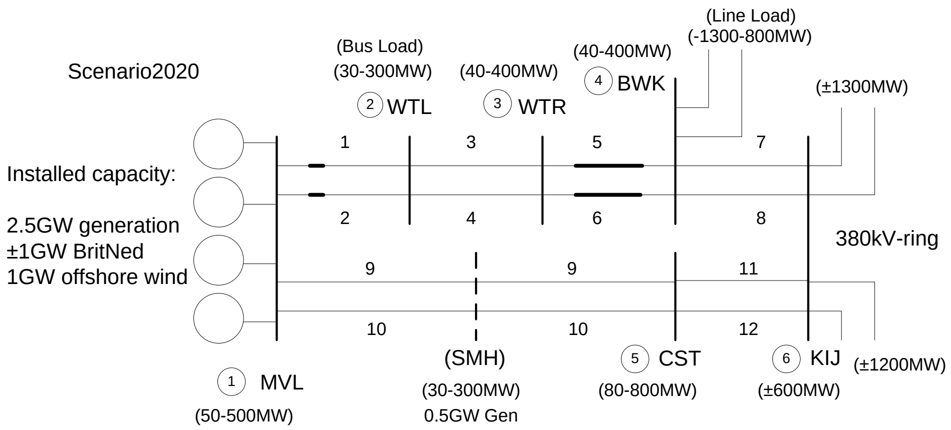
Write a small report (in a group of up to 4 students) about this practicum session.

Session 5: State enumeration

In this practicum session, state enumeration will be applied to analyse the reliability of large transmission networks. This practicum session is mainly related to section 5.2 of the course reader. Appendix B of the reader can be read to refresh your knowledge of load flow calculations. Appendix C gives an overview of reliability indicators. Python will be used to perform the state enumeration.

Assignments:

In this practicum session, we will study the reliability of a part of the Dutch EHV transmission network: the network around Maasvlakte (MVL) substation. In the figure below, this part of the network is illustrated. On the left, the large-scale generation centre MVL can be seen, which is connected to the main 380kV-ring of the Netherlands on the right. In the figure, the connections (circuits) and substations (buses) are numbered. The installed generation, ranges of the bus loads and line loadings according to a load flow scenario for 2020 are indicated as well. The circuits connected to SMH (i.e. 9-10) have the same number from MVL to CST. As SMH is a tapping station, a circuit failure between MVL and SMH will lead to the disconnection of the complete MVL-SMH-CST circuit. The same holds for a failure between SMH and CST. For simplicity, we will not consider SMH substation in this study and add the generation/load at SMH to the generation/load at MVL substation. The underground cables of Randstad380 Zuid are indicated with bold lines in the diagram. 11 km UGC double circuit is located between WTR and BWK, while 2 km UGC double circuit is located between MVL and WTL.



In the practicum files, you can find the Python files: **'get_B_matrix_NL_func2020_MVL_.py', 'get_P_lines_MVL_func.py' and 'deterministic_contingency_analysis_MVL.py'**. These files together describe a deterministic contingency analysis of this network. 'get_B_matrix…' provides the B-matrix of the network, which is needed for the DC load flow calculation. In 'get_P_lines…', the DC load flow is calculated, which results in the line loadings (in % of the lines capacities). The data files **'scenario2020_line_flows_MVL.csv' and 'scenario2020_MVL.csv'** contain the bus loads/generation and line loadings of the connections to the rest of the 380 kV network, according to scenario2020.

1. Perform the deterministic contingency analysis by running 'deterministic_contingency_analysis_ MVL.py'. The resulting graph shows boxplots of the line loadings in normal operation (when all lines are in operation), and the maximum line loadings under 1st-order contingencies (indicated by 'o' in the graph). Which lines are loaded the most in normal operation, and which the least? Can

you explain this using the network diagram as shown above? What is the maximum loading of these lines under 1st-order contingencies? Is this network n-1 redundant? [2 points]

2. Carefully study the Python file ‘deterministic_contingency_analysis_MVL.py’, as this will be the basis of your state enumeration. Where in the program is the DC load flow calculated and what are the in- and outputs? Why is it checked whether a bus (i.e. substation) is islanded (i.e. isolated from the rest of the network? Why is it checked whether the slackbus is smaller than 100 MW? When is a line overloaded? [2 points]

As described in the course reader, the main difference between (deterministic) contingency analysis and (probabilistic) state enumeration is that in the latter, the probabilities of the possible system failure states are included in the calculation. These state probabilities are based on the unavailabilities of the circuits in the network. The lengths of the circuits in this network are as shown below.

Circuit nr.	1-2	3-4	5-6	7-8	9-10	11-12
Length [km]	17.6	10	22.2	17.8	66	14.6

In circuits 1-2, 2 km underground cable with Randstad380 configuration is embedded and in circuits 5-6, 11 km UGC with Randstad380 configuration is installed. The failure frequencies and repair times of the connection components are as shown below.

Component	**Failure frequency**	**Repair time**
EHV OHL	0.00220 /cctkmy	8 h
EHV cable	0.00120 /cctkmy	730 h
EHV joint	0.00035 /compy	730 h
EHV termination	0.00168 /compy	730 h

From the failure frequencies and repair times and the circuit lengths, the unavailabilities of the circuits can be calculated. These are shown in the table below.

Circuit number	**Unavailability [-]**
1-2	2.46e-3
3-4	2.01e-5
5-6	6.18e-3
7-8	3.58e-5
9-10	1.33e-4
11-12	2.93e-5

3. We will now develop a 1st-order state enumeration, which only studies independent single circuit failures (and the normal operation state of course). Make a copy of ‘deterministic_ contingency_analysis_MVL.py’ and call this ‘state_enumeration_MVL.py’. Add a vector with the unavailabilities of the circuits at the beginning of this state enumeration. [2 points]

4. Modify the state enumeration program such that the failure state probability is calculated from the circuit unavailabilities each time a contingency is considered. Also, change the program such that the probability of an islanded substation, the probability of an overloaded circuit and the total probability of all considered system states are calculated. Perform the state enumeration. What is the probability of an islanded substation? What is the probability of an overloaded circuit? Is the result as expected? What is the total probability of not-considered (higher-order) states? Is this state enumeration accurate? [2 points]

5. We will now develop the state enumeration into a 2nd-order state enumeration. Modify the state enumeration program such that combinations of 2 independent circuit failures are studied as well. Calculate the probabilities of these states and the probability of an islanded bus or an overloaded line. What are the values of these indicators now? Is the network n-2 redundant? What is the total probability of not-considered (higher-order) states? Did the accuracy improve much? [1 point]

6. In the future, more offshore wind will be connected to MVL substation. In the state enumeration, 1 GW offshore wind capacity is currently connected to MVL (bus 1). We will now study what will happen if this capacity is increased. You can do this by changing the section in the program where the bus loads are defined. As the excess of wind power must be absorbed somewhere, subtract the extra wind energy from bus 6 (KIJ), to which rest of the 380 kV network is connected. Perform the state enumeration for 1 to 5 GW (in steps of 0.5 GW) at MVL and draw a graph of the results. What are the probabilities of an islanded substation and an overloaded network now? Is the network still n-1 redundant? And is it n-2 redundant? [1 point]

7. (optional) In reality, dependent double circuit failures can occur in the network as well. You can assume that the probability of a dependent double circuit failure is 0.1 times the unavailability of a single circuit. Modify the state enumeration such that dependent double circuit failures are considered as well. Also, study combinations of one dependent double circuit failure and one (independent) single circuit failure in the state enumeration. Moreover, include combinations of two dependent double circuit failures in the analysis. What are the probabilities of an islanded substation and an overloaded network now? What is the total probability of not-considered (higher-order) states?

Write a small report (in a group of up to 4 students) about this practicum session.

Session 6: Monte Carlo simulation

In this practicum session, Monte Carlo simulation is used to analyse the reliability of large transmission networks. This practicum session is mainly related to section 5.4 of the course reader. Appendix B of the reader can be used to refresh the basic knowledge of load flow calculations. Appendix C gives an overview of some reliability indicators. Python will be used to perform the Monte Carlo simulation.

Assignments:

In this practicum session, we will analyse the reliability of the network around Maasvlakte again, but now using Monte Carlo simulation. The network configuration and details were already discussed in the assignments of the previous practicum session.

1. Start again with the Python file ‘deterministic_contingency_analysis_MVL.py’. Make a copy of this file and call it ‘Monte_Carlo_simulation_MVL.py’. We will develop a non-sequential Monte Carlo simulation that only considers independent failures. Add the vector with the unavailabilities of the circuits to the program. [3 points]

2. Modify the program such that it determines the statuses of the circuits in the network by generating random (unit uniform) samples and comparing these to the circuit unavailabilities. For the bus loads, take random samples (i.e. snapshots) from the load flow scenario defined in the matrix ‘bus_loads’. Start with a small simulation time, which you can increase later if your program works. To speed up the Monte Carlo simulation, you can decide to skip the calculations if there aren’t any failures in the network. [3 points]

3. We will first study the (extreme) case in which the offshore wind at MVL is multiplied by 5 (the extra wind power is absorbed at KIJ: the main 380kV-ring). In this scenario, the network is not n-1 redundant anymore. Perform the Monte Carlo simulation. What are the probabilities of an islanded substation and an overloaded circuit in the network? Calculate the 95%-confidence intervals as described in example 5.12 in the course reader. Is your Monte Carlo simulation accurate? Are the results comparable to the results found in the previous practicum session? Which approach, state enumeration or Monte Carlo simulation, do you prefer for this study? [4 points]

4. (optional) In reality, dependent double circuit failures can occur as well. Modify your Monte Carlo simulation such that dependent double circuit failures are also considered. It is recommended to work with contingency lists here (e.g. [circuit# unavailability] and [circuit# circuit# probability]). Perform the Monte Carlo simulation. What are the probabilities of an islanded substation and an overloaded circuit in the network?

Write a small report (in a group of up to 4 students) about this practicum session.

References

[1] W. Li, *Risk Assessment of Power Systems – Models, Methods, and Applications.* Canada: Wiley Interscience – IEEE Press, 2005.

[2] B.W. Tuinema, J.L. Rueda, L. van der Sluis, M.A.M.M. van der Meijden, "Reliability of Transmission Links consisting of Overhead Lines and Underground Cables," *IEEE Trans. Power Del.*, Vol. 31(3), 2016.