Advanced Design Tools for Ocean Energy Systems Innovation, Development and Deployment

Deliverable D6.3

Reliability, Availability, Maintainability and Survivability Assessment Tool – Alpha version

Lead Beneficiary	Aalborg University
Delivery Date	30/04/2020
Dissemination Level	Public
Status	Released
Version	1.0
Keywords	Reliability, Availability, Maintainability, Survivability, Fault Tree, Hierarchy, Downtime, Repair time, Ultimate loads, Fatigue loads

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 785921

Disclaimer

This Deliverable reflects only the author's views and the Agency is not responsible for any use that may be made of the information contained therein

Document Information

Grant Agreement Number	785921
Project Acronym	DTOceanPlus
Work Package	WP 6
Related Task(s)	T6.4
Deliverable	D6.3
Title	Reliability, Availability, Maintainability and Survivability Assessment Tool – Alpha version
Author(s)	Yi Yang (AAU), Anup Nambiar (UEDIN), Neil Luxcey (FEM), Francisco Fonseca and Luis Amaral (WavEC)
File Name	DTOceanPlus_D6.3 Systems RAMS Tools_alpha_AAU_20200430_v1.0.docx

Revision History

Revision	Date	Description	Reviewer
0.1	20/04/2020	Draft version for technical review	Donald R Noble, Tianna Bloise Thomaz, Noor van Velzen (UEDIN)
0.2	23/04/2020	First full version for QA review	Donald R Noble (UEDIN)
0.9	29/04/2020	Second version for QA review	Donald R Noble (UEDIN)
1.0	30/04/2020	Final version for the EC	EC

EXECUTIVE SUMMARY

Deliverable D6.3 "Reliability, Availability, Maintainability and Survivability – alpha version" of the DTOceanPlus project include the details of the Assessment Design Tools module: "Reliability, Availability, Maintainability and Survivability" (RAMS), and it presents the result of the work developed during the tasks T6.2 and T6.4 of the project. This document serves as the technical manual of the alpha version of the RAMS module, including all the data requirements, main functions, interfaces and all the pertinent technical details.

This document summarises both the functionalities as well as the more technical aspects of the code implemented for the alpha version of this module. The RAMS module will provide the user with the assessments pertinent to the component-level and system-level reliability, the time-based availability, the maintainability and the survivability (for the perspectives of both ultimate and fatigue limit states). The Business Logic of the code, i.e. the actual functions of the RAMS module, has been implemented in Python 3. Moreover, the code is provided with an Application Programming Interface (API), developed in OpenAPI, in order to interact and communicate with the other modules of the DTOceanPlus platform: A Graphical User Interface (GUI) is being developed, consistently with the other modules, in Vue.js, allowing the user to interact easily with the RAMS module, inputting data and visualising results.

The Business Logic of the code has been partly verified (more than 91%) through the implementation of unit tests, guaranteeing easy maintainability for future developments of the tool. A section of examples completes the present document, showing the capabilities of the tool.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
BL	Business Logic
CL	Complexity Level
CoV	Coefficient of Variance
ED	Energy Delivery
ET	Energy Transformation
FLS	Fatigue Limit State
FORM	First Order Reliability Method
FT	Fault Tree
GUI	Graphic User Interface
LMO	Logistics and Marine Operations
MBL	Maximum Breaking Load
MTTF	Mean time to failure
MTTR	Mean time to repair
PDF	Probability Density Function
PoF	Probabilities of Failure
PTO	Power take-off
RAMS	Reliability Availability Maintainability Survivability
REST	Representational State Transfer
SG	Stage Gate
SI	Structured Innovation
SK	Station Keeping
TEC	Tidal Energy Converter
TTF	Time to Failure
TTR	Time to Repair
ULS	Ultimate Limit State
WEC	Wave Energy Converter

1. INTRODUCTION

1.1 SCOPE AND OUTLINE OF THE REPORT

Deliverable D6.3 "Reliability, Availability, Maintainability and Survivability Tool – alpha version" of the DTOceanPlus project includes the details of the Assessment Design Tool module: "Reliability, Availability, Maintainability and Survivability" (RAMS), and it presents the results of the work developed during the tasks T6.2 and T6.4 in the project. This document serves as the technical manual of the alpha version of the RAMS module, including all the data requirements, main functions, interfaces and all the pertinent technical details. The alpha version of this tool is a fully functional version of the tool in terms of implementation of the calculations covered by the RAMS module (Business Logic).

This document summarises:

1) The **RAMS framework and assumptions** (Section 2),
2) **Use cases and the functionalities** (Section 3) of the RAMS tool, namely providing the user with a set of relevant metrics and assessments on Reliability, Availability, Maintainability and Survivability.
3) The actual **implementation of the tool**, describing the architecture of the tool, the technologies adopted for the implementation and the results of the testing (Section 4).
4) The **testing** of the code: the Business Logic of the code has been tested through the implementation of unit tests, guaranteeing easy maintainability for future developments of the tool (Section 4.2).
5) A set of extensive **examples**, to provide the reader with an overall view of the capabilities of the tools (Section 5).

1.2 SUMMARY OF THE DTOCEANPLUS PROJECT

The RAMS module belongs to the suite of tools "DTOceanPlus" developed within the EU-funded project DTOceanPlus [1]. DTOceanPlus aims to accelerate the commercialisation of the Ocean Energy sector by developing and demonstrating an open source suite of design tools for the selection, development, deployment and assessment of ocean energy systems (including sub-systems, energy capture devices and arrays) and at various levels of complexity.

At a high level, the suite of tools developed in DTOceanPlus will include:

- **Structured Innovation tool (SI)**, for concept creation, selection, and design.
- **Stage Gate tool (SG)**, using metrics to measure, assess and guide technology development.
- **Deployment tools**, supporting optimal device and array deployment:
  - *Site Characterisation (SC):* to characterise the site, including metocean, geotechnical, and environmental conditions.
  - *Machine Characterisation (MC):* to characterise the prime mover.
  - *Energy Capture (EC):* to characterise the device at an array level.
  - *Energy Transformation (ET):* to design PTO and control solutions.

- *Energy Delivery (ED):* to design electrical and grid connection solutions.
- *Station Keeping (SK):* to design moorings and foundations solutions.
- *Logistics and Marine Operations (LMO):* to design logistical solutions and operation plans related to the installation, operation, maintenance, and decommissioning operations.

- **Assessment Tools**, to evaluate projects in terms of key parameters:
  - *System Performance and Energy Yield (SPEY):* to evaluate projects in terms of energy performance.
  - *System Lifetime Costs (SLC):* to evaluate projects from the economic perspective.
  - ***System Reliability, Availability, Maintainability, Survivability*** *(RAMS):* to evaluate the reliability aspects of a marine renewable energy project.
  - *Environmental and Social Acceptance (ESA):* to evaluate the environmental and social impacts of a given wave and tidal energy projects.

These will be supported by underlying common digital models and a global database, as shown graphically in Figure 1.1.

The RAMS module is one of Assessment Tools, and will assess such metrics as reliability, availability, maintainability, and survivability of marine energy conversion systems. The objective of this manual is to document the implementation of theoretical methods assessing these metrics in the RAMS module, and to help the end users understand the technical details of the RAMS module.

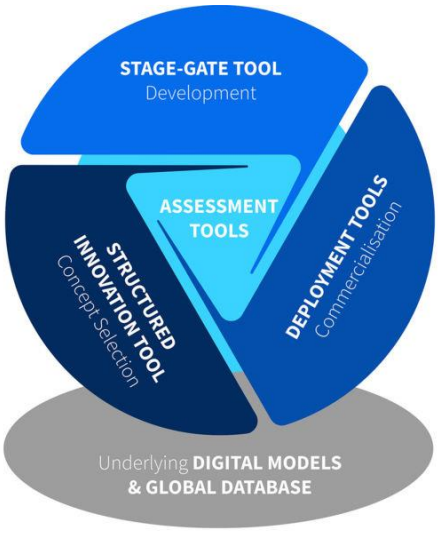


FIGURE 1.1. REPRESENTATION OF DTOCEANPLUS TOOLS

2. RAMS FRAMEWORK AND ASSUMPTIONS

2.1 RAMS FRAMEWORK

2.1.1 OVERVIEW

The definitions of such terminologies as reliability, availability, maintainability, and survivability, are given below.

- **Reliability** – the ability of a structure or structural member to fulfil the specified requirements, during the working life, for which it has been designed [2].
- **Availability** – theoretically refers to the probability that a system or component is performing its required function at a given point in time or over a stated period of time when operated and maintained in a prescribed manner [3]. However, in most of the engineering applications, it is defined as the ratio of the uptime to the design lifetime [4].
- **Maintainability** – the ability of a system to be repaired and restored to service when maintenance is conducted by personnel using specified skill levels and prescribed procedures and resources [5].
- **Survivability** – the probability that the converter will stay on station over the stated operational life [6].

2.1.2 DATA STRUCTURE OF INPUT DATA

2.1.2.1 Hierarchy

A hierarchy of a system/ subsystem is a digital representation illustrating the working philosophy and the interrelationship between the units at different levels in this system/ sub-system. A tree-like data structure, where each unit is denoted as a node, is used to represent such a hierarchy, which can be stored in some typical data formats, e.g. pandas table and dictionary.

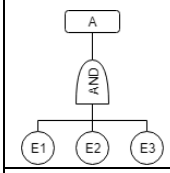
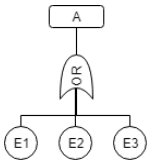
A hierarchy is constructed by means of the bottom-up technique, and the steps for defining such a hierarchy are briefly summarized as follows:

- Identification of nodes representing the basic components.
- Categorization of the basic components into groups; each group is a sub-assembly and stands for an intermediate-level node; according to the rules in the tree-like data structure, the basic components are considered as children of this intermediate-level node; a logic gate is inserted between the basic nodes (representing the basic components) and the intermediate-level node to reflect the logic dependencies between these nodes.
- These identified first-level subassemblies can be considered 'basic' components in order to identify the second-level subassemblies (a higher level) based upon the same method mentioned in the second bullet point; and this procedure can be repeated until the system to be analysed is reached.

The concept of logic gates is borrowed from the classic fault tree analysis [7]. However, the meanings of the logic gates defined in the hierarchy are different from their counterparts in the classic fault tree analysis, because the logic dependencies are defined from the perspective of energy transfer, instead of failures of units. A comparison of logic gates has been made, as given in Table 2.1.

It should be born in mind that the logic gates in a hierarchy will be automatically converted to the correct logic gates in the code in business logic.

TABLE 2.1 COMPARISON OF LOGIC GATES IN HIERARCHICAL STRUCTURES AND FAULT TREES

Logic Gate	Fundamental Definition	Interpretation in Fault Tree	Interpretation in Hierarchy
'AND' 	The 'AND' gate indicates that the output event 'A' occurs only when all the input events (E_i, i=1,2,3) occur at the same time.	If each E_i represents a component and *A* represents a subsystem, the 'AND' gate can be understood in such a way: If all E_is fail, *A* fails.	If each E_i represents a component and *A* represents a subsystem, the 'AND' gate can be understood in such a way: If either of E_is fails, *A* fails.
'OR' 	The 'OR' gate indicates that the output event 'A' occurs if any of the input events (E_i, i=1,2,3) occur.	If each E_i represents a component and *A* represents a subsystem, the 'OR' gate can be understood in such a way: If either of E_is fails, *A* fails.	If each E_i represents a component and *A* represents a subsystem, the 'OR' gate can be understood in such a way: If all E_is fail, *A* fails.

Besides the logic dependencies, the failure modes and the associated failure rates should also be defined in the hierarchy. A failure mode is defined as the manner in which a unit could potentially fail to meet or deliver the intended function(s). The identification of system-level failure modes at the operational stage is out of work scope of the RAMS module. Unless otherwise specified, RAMS only cases about the failure modes of the basic components, including two major types:

- Failure mode 1 – failures which can be restored through repair
- Failure mode 2 – the cases in which the severely failed components can only be replaced

A template of hierarchy is given in Table 2.2. The first column gives the subsystem or system to be analysed. All failure events are considered nodes in the hierarchy. The second column, 'Name of Node', gives the names of these failure events. The third column, 'Design Id', gives the identification labels of the basic components and other units. Design Ids are named according to the rules/ conventions of the Energy Delivery (ED) module. The column, 'Node Type', defines the levels of a hierarchy. The column, 'Node SubType', defines the additional information the design modules use to identify the corresponding node. The column, 'Category', defines which levels the nodes in the 'Name of Node' column belong to in the fault tree. The columns 'Parent' and 'Child' define the dependencies of units at various levels. Each entry in 'Parent' defines the label of the higher-level unit which the current unit in the column 'Name of Node' belongs to. Each entry in 'Child' defines the labels of lower-level units which belong to the current unit. Based upon the aforementioned descriptions, the units in the column 'Child' are connected through a specific logic gate to the higher-level unit. The logic gates are given in the column 'Gate Type'. The logic gate in each entry of this column is used to connect the unit in the column 'Name of Node' and the units in the column 'Child'. The last two columns give the failure rates of basic components for both Type 1 & Type 2 failure modes.

A few examples demonstrating how a hierarchy is defined and used in the system-level reliability assessment will be given in Section 5.

TABLE 2.2 TEMPLATE OF HIERARCHY

System	Name of Node	Design Id	Node Type	Node Subtype	Category	Parent	Child	Gate Type	Failure Rate Repair [1/hour]	Failure Rate Replacement [1/hour]

2.1.2.2 Stresses/ Loads

The survivability will be assessed from the perspectives of both the ultimate and fatigue limit states, based upon the structural reliability method [8] [9] [10]. The ultimate stresses/ loads are used to assess the survival probability from the perspective of the ultimate limit state. The fatigue stress/ load ranges are used to assess the survival probability from the perspective of the fatigue limit state. RAMS requires these inputs from the Energy Transformation (ET) and Station Keeping (SK) modules. A simplified limit state function will be used for the survivability assessment from the perspective of the ultimate limit state. The S-N curve-based approach will be used to define the fatigue limit state [11]. See Section 2.2 for more details regarding the assumptions of the survivability assessment.

2.1.2.3 Downtime

Downtime is a measure of a time duration of a facility which does not work normally (no generation of power). Generally, availability is assessed according to downtime. The downtime provided by the Logistics and Marine Operations (LMO) module is a three-dimensional array. The three dimensions respectively represent the devices, the calendar years, and the hours during the downtime. A schematic illustration of the data structure storing downtime is shown in Figure 2.1.

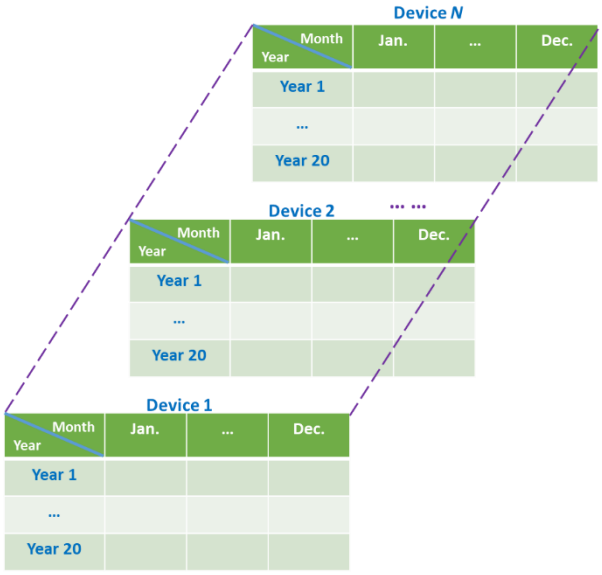


FIGURE 2.1 A SKETCH ILLUSTRATING THE DATA STRUCTURE FOR DOWNTIME

2.1.2.4 Maintenance Plan

A maintenance plan developed by LMO defines the data related to maintenance activities. RAMS only requires the ids of the components (device ids as well) to be repaired and the mean time to repair (MTTR). A few commonly used terminologies regarding maintenance time are shown in Figure 2.2. Time to repair (TTR) includes the waiting time, the repair time and the time for testing and restoring to the normal condition. MTTR can be typically defined as the ratio of the summation of TTRs to the time of repairs.

In nature, TTR is a stochastic variable, which follows a probabilistic distribution. RAMS predefines some default probabilistic distributions for the user to choose from. The MTTR provided by LMO is considered the mean of the probabilistic distribution. Besides MTTR, the user should also input the standard deviation or the coefficient of variance of the TTR.

For the time being, the default probabilistic distributions include Gaussian, LogNormal and Exponential.

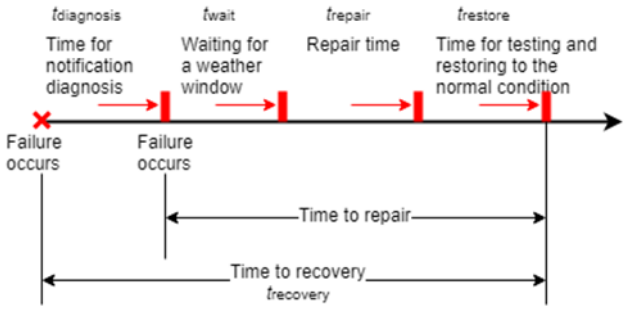


FIGURE 2.2 A SKETCH ILLUSTRATING TIME TO REPAIR (TTR)

2.2 ASSUMPTIONS

It should be noted that the RAMS assessment results make sense, if and only if the assumptions made in the RAMS code hold. These assumptions include the fundamental ones which are embedded in the theoretical basis, and some specific assumptions tailor-made for the DTOceanPlus project.

The fundamental assumptions are:

- Reliability assessment
  - The failures of different basic components are statistically independent.
  - The time-to-failure (TTF) of the same component is statistically independent; so, the exponential probabilistic distribution is used to simulate the TTF for each basic component.
  - The fault tree (FT) approach is used to analyse the failures of units. A unit can be a system, a subsystem or a subassembly.
- Availability assessment
  - The availability of the marine energy farm/ array is the ratio of the total uptime to the sum of the uptime and the downtime, regardless of the causes(s) of the downtime.
- Maintainability assessment

- Due to the lack of historical repair-time data, the parametric method cannot be used to fit the probabilistic distribution of the repair-time; so, three default types of probabilistic distributions, namely, Gaussian, LogNormal, or Exponential, will be the options for the user to choose from.

- Survivability assessment
  - Focus is only put on the probabilities of critical structural/ mechanical components surviving the stresses/ loads during the lifetime.

The specific assumptions for the DTOceanPlus project are:

- Reliability assessment
  - The logic dependencies between units at different levels are defined in a hierarchy from the perspective of energy transfer.
  - No inspection is conducted and the annual of probabilities of failure (PoFs) will not be provided.
- Maintainability assessment
  - According the rule set up by LMO, MTTR is only estimated for the components which need to be repaired.
  - The minimum of the probabilities that the damaged components can be successfully repaired within a stated time interval will be the output.
- Survivability assessment
  - The sea states chosen in the Deployment Design tools and analysed in ET are enough to approximately estimate the ultimate stress/ load probabilistic distribution.
  - The sea states from ET are enough to estimate the probabilistic distribution of the long-term stress ranges.
  - The modelling procedure set by the SK module is refined enough to capture the most critical load cases.
  - The stress range corresponding to the 10000 cycles of the chosen S-N curve is a rough estimation of the yielding strength of the material of the primary structural/ mechanical component.

3. USE CASES AND FUNCTIONALITIES

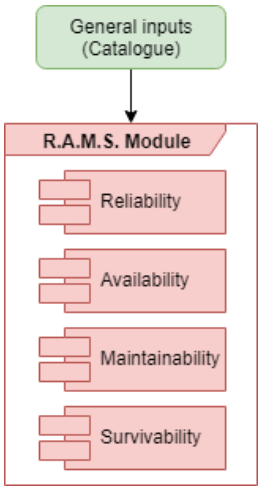
3.1 OVERVIEW

The Reliability, Availability, Maintainability and Survivability (RAMS) module will:

- Perform the component-level and system-level reliability assessment on the units at different levels in a marine energy system (array); the component-level reliability assessment provides the outputs of the mean time to failure (MTTF) of basic components and the simulated time-to-failure (TTF) of these basic components; the system-level reliability assessment obtains the information on the logic dependencies and the interrelationship between units at different levels in the marine energy system (array) defined in the hierarchies, and provides the outputs of the annual probabilities of failure (PoFs) as a function of time for the units at higher levels in the hierarchies.
- Calculate the availability for each device.
- Estimate the probabilities that the damaged components can be successfully repaired within a stated time interval given procedures and resources.
- Estimate the probability of critical structural/ mechanical components surviving the stresses/ loads during the design lifetime.

With the aim to assess the reliability, availability, maintainability, and survivability of units at various levels in an array, the RAMS module requires the inputs from the design tools and exports the outputs to the Stage Gate tool, the Structure Innovation tool and the LMO module, as illustrated in Figure 3.1. Brief descriptions regarding the inputs and outputs are marked above the arrows representing the data flow.

When an analysis is run completely, the output (reliability, availability, maintainability, and survivability) from the RAMS module will be fed to the Stage Gate (SG) and Structured Innovation (SI) tools. The Logistics and Marine Operations (LMO) and RAMS modules will work closely on defining maintenance strategies and develop some shared public functions to provide the optimal maintenance plan.



(a)

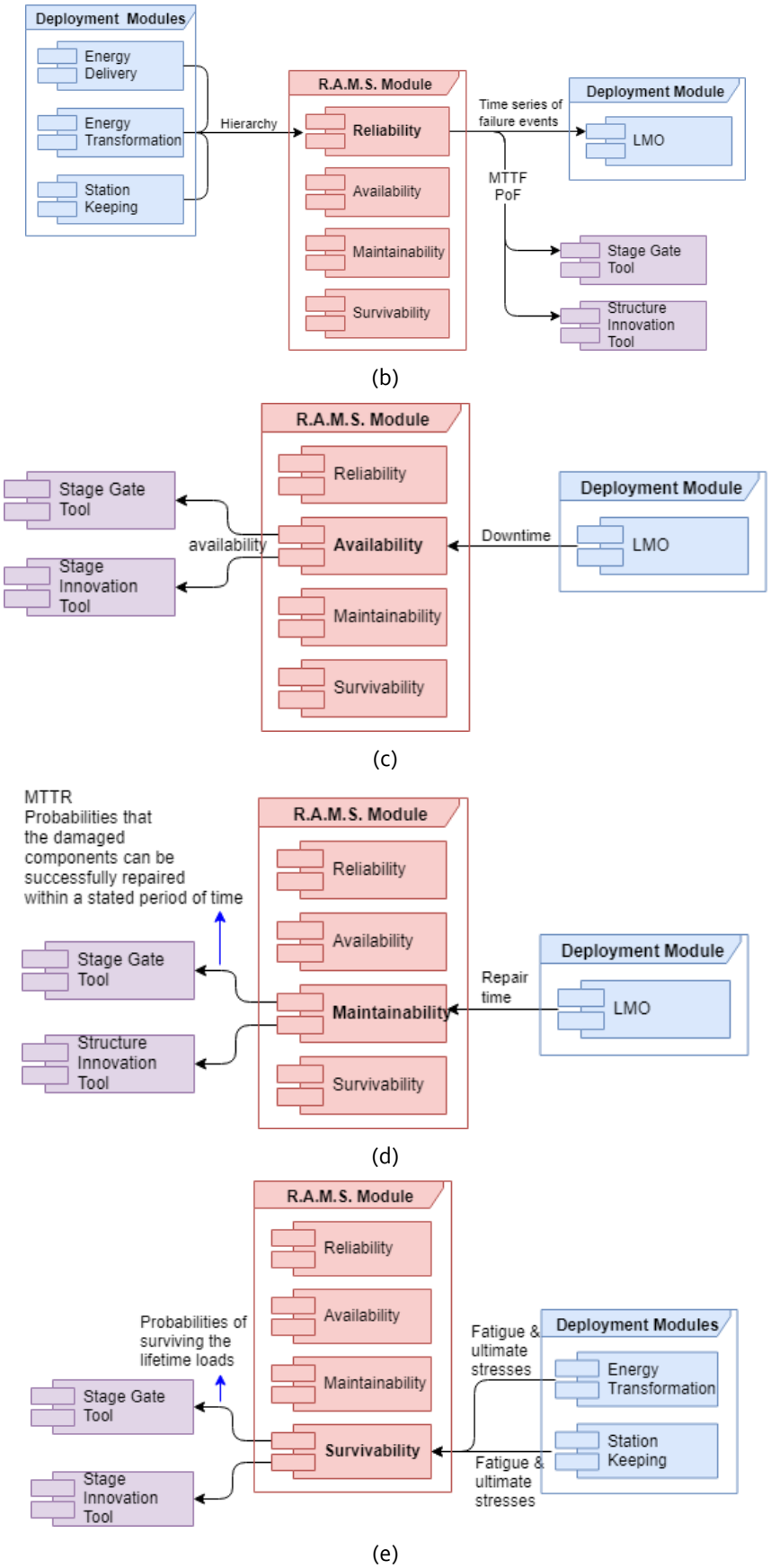


FIGURE 3.1 INTERFACE BETWEEN RAMS AND OTHER MODULES/TOOLS

3.2 THE USE CASES

The Generic User Case can be generally summarised as shown in Figure 3.2.

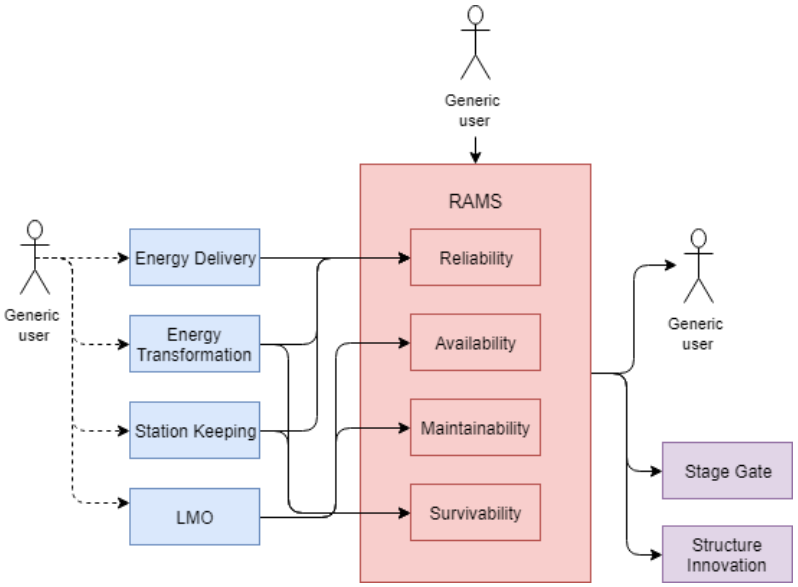


FIGURE 3.2 GENERIC USE CASE FOR USING THE RAMS TOOL

The User can:

1) Run RAMS within the framework of the Stage Gate (SG) or Structured Innovation (SI) tools
2) Run RAMS after running the set of Deployment Design tools of DTOceanPlus
3) Use in standalone mode.

By considering the three Use cases above mentioned, Table 3.1 summarises the dependencies of RAMS from/to other modules in DTOceanPlus.

TABLE 3.1 DEPENDENCIES OF RAMS FROM/TO OTHER MODULES IN DTOCEANPLUS

Modules that provide services that RAMS consumes	Modules that are consuming services from RAMS
Energy Delivery (ED), Energy Transformation (ET), Station Keeping (SK), Logistics & Marine Operations (LMO)	Structured Innovation (SI), Stage Gate (SG)

3.2.1 USE CASE WITHIN THE FRAMEWORK OF SG/SI DESIGN TOOLS

In this case, the RAMS tool will be run within the framework of the Stage Gate or Structured Innovation tools, as shown in Figure 3.3. The following steps are identified for this use case:

1) The user runs the framework of the SI/SG Tools
2) The SI/SG will require eventually some assessment results from the RAMS module
3) The RAMS Module will check if the needed information is available and in case it is not, it will request the user to input the information from the relevant Deployment Design Tools
4) The User will complement the information and run the Deployment Design Tools
5) RAMS will be run and perform the assessments
6) RAMS will provide the assessments to SI/SG Tools to complete their framework

The outcome will be shown to the user.

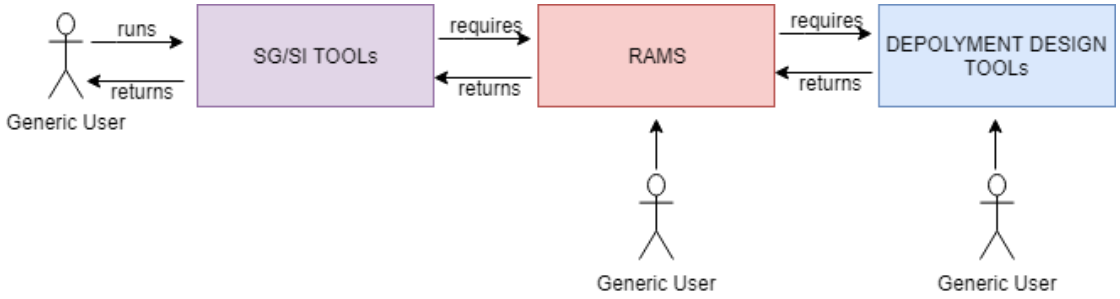


FIGURE 3.3 USE CASE FOR USING THE RAMS TOOL WITHIN THE FRAMEWORK OF SG/SI TOOLS

3.2.2 USE CASE AFTER DEPLOYMENT DESIGN TOOLS

In this case, the User will run one or more Deployment Design Tools and then they will run the RAMS module to carry out the assessments, as shown in Figure 3.4. The numerical results as well as the graphs/diagrams will be exposed to the user.

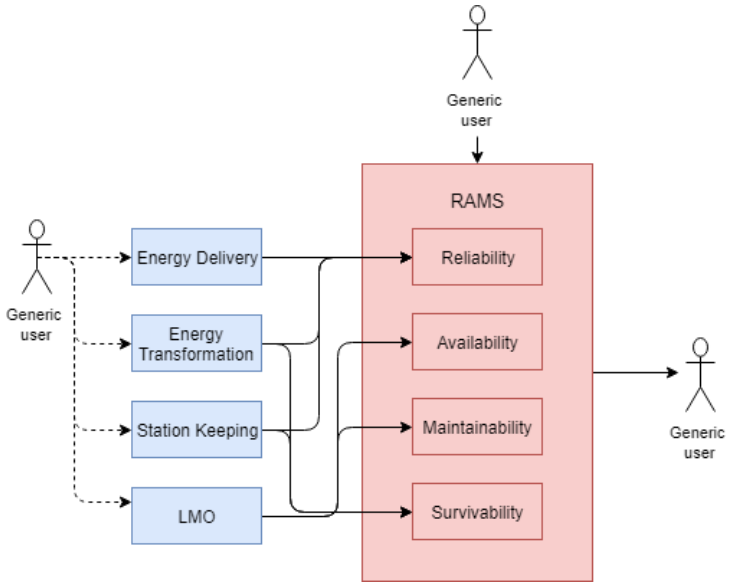


FIGURE 3.4 USE CASE FOR USING THE RAMS TOOL AFTER RUNNING THE DEPLOYMENT DESIGN TOOLS

3.2.3 STANDALONE MODE

In this case, all the input data should be available in advance of running the RAMS tool as highlighted in Figure 3.5.

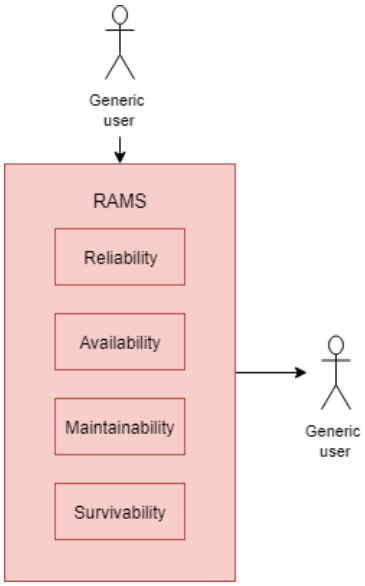


FIGURE 3.5 USE CASE FOR USING THE RAMS TOOL STANDALONE

3.3 THE FUNCTIONALITIES

The RAMS module mainly assesses reliability, availability, maintainability, and survivability, at three complexity levels (CL).

Reliability: it includes two-level assessment, namely

Component-level: gives the time series of failure events of basic components and the mean time to failure (MTTF)

System-level: gives the probabilities of failure (PoFs) of units at various levels based upon the hierarchical structure of this subsystem.

Availability: it is a time-based metric and calculated for every device, based upon the downtime provided by LMO.

Maintainability: it gives the mean time to repair (MTTR) and indicates the probabilities that the damaged components can be successfully repaired within a stated time interval given procedures and resources.

Survivability: it indicates the probability of structural/ mechanical components surviving the lifetime loads/ stresses.

The differences of the functionalities between three CLs are summarised in Table 3.2. It should be noted that: RAMS, as an assessment module, highly depends upon the accuracy of the inputs; most of the functions/ methods between three complexity levels are the same. For example, the hierarchies at complexity level 3 should be more complicated than those at complexity level 1 (probably, more intermediate levels and more basic components). The same code is used; however, the results should be significantly different.

TABLE 3.2 COMPARISON OF THE FUNCTIONALITIES BETWEEN THREE COMPLEXITY LEVELS

Assessment	Complexity Level 1	Complexity Level 2	Complexity Level 3
Reliability	The same functions/methods are used at all levels for both the component-level and system-level reliability assessment		
Availability	The same functions/methods are used at all levels		
Maintainability	The same functions/methods are used at all levels		
Survivability	For ULS, Monte Carlo Simulation is the only available approach; For FLS, Monte Carlo Simulation is the only available approach.	For ULS, both FORM and Monte Carlo Simulation are available; For FLS, Monte Carlo Simulation is the only available approach.	For ULS, both FORM and Monte Carlo Simulation are available; For FLS, Monte Carlo Simulation is the only available approach.

3.3.1 RELIABILITY

3.3.1.1 Objectives

The component-level and system-level reliability assessments are performed for the marine energy conversion farm/array and the ED, ET, and SK subsystems. The component-level assessment aims to simulate the TTF and calculate the MTTFs for the basic components. The system-level assessment aims to estimate the PoFs as a function of time for the units at higher levels in the hierarchy.

3.3.1.2 Inputs, Models and Outputs

Inputs

The input data for assessing reliability are listed in Table 3.3. See the description of the data structure of the hierarchy in Table 2.2 in Section 2.1.2.1.

TABLE 3.3 INPUTS FOR ASSESSING RELIABILITY

ID	Brief Description of Input Quantity	Origin of the Data	Data Model in RAMS	Unit
hierarchy_ed	The hierarchical structure of the ED subsystem	The ED tool	Dictionary	-
hierarchy_et	The hierarchical structure of the ET subsystem	The ET tool	Dictionary	-
hierarchy_sk	The hierarchical structure of the SK subsystem	The SK tool	Dictionary	-

Theoretical basis

See ANNEX 1 for more details of the theoretical basis.

Outputs

The outputs comprise the following items:

- Simulated TTFs of basic components included in the hierarchies
- MTTFs of basic components included in the hierarchies
- The PoFs of units at higher levels in the hierarchies

3.3.2 AVAILABILITY

3.3.2.1 Objectives

The availability assessment aims to estimate the availability of every device during the lifetime, which is defined as the uptime of the concerned device divided by the sum of downtime and uptime.

3.3.2.2 Inputs, Models and Outputs

Inputs

The input data for assessing availability are listed in Section 2.1.2.3.

Table 3.4. See the description of the data structure of the downtime in Figure 2.1 in Section 2.1.2.3.

TABLE 3.4 INPUTS FOR ASSESSING AVAILABILITY

ID	Brief Description of Input Quantity	Origin of the Data	Data Model in RAMS	Unit
t_down	The downtime of devices in the array	The LMO tool	Multi-dimensional array	hour

Theoretical basis

The time-based availability is equal to the uptime divided by the sum of update and downtime, as expressed in Eq. (1) [4].

$$\textbf{Availability} = \frac{t_{\text{uptime}}}{t_{\text{uptime}}+t_{\text{downtime}}} \quad (1)$$

Outputs

The output comprises the following item(s):

- the availability of each device during the lifetime
- the approximate estimation of availability of the array/farm

3.3.3 MAINTAINABILITY

3.3.3.1 Objectives

The maintainability assessment aims to:

- estimate the probability that the damaged components can be successfully repaired within a stated time period (t_specific) [5], based upon the MTTR provided by LMO and the user-defined standard deviation (Std)/ coefficient of variance (CoV) of the repair time.

It should be noted that LMO only estimates the time-to-repair (TTR) for the concerned basic components which need to be repaired or replaced according to the pre-defined decision rules in LMO. The decision rules define which actions should be taken when failures occur.

3.3.3.2 Inputs, Models and Outputs

Inputs

The input data for assessing maintainability are listed in Table 3.5. See the description of the maintenance plan in Section 2.1.2.4.

TABLE 3.5 INPUTS FOR ASSESSING MAINTAINABILITY

ID	Brief Description of Input Quantity	Origin of the Data	Data Model in RAMS	Unit
MTTR	The mean time to repair of basic components	The LMO tool	Dictionary	hour
pd	The probabilistic distribution of theTTR	User-defined	String	-
Std/CoV	The standard deviation or the coefficient of variance of the TTR	User-defined	Float	hour

ID	Brief Description of Input Quantity	Origin of the Data	Data Model in RAMS	Unit
t_specific	The stated time interval during which a repair is done	User-defined	Float	hour

Theoretical basis

The TTR follows a probabilistic distribution, e.g. Gaussian distribution, LogNormal distribution, and Exponential distribution. As mentioned in Section 2.2, MTTR is only estimated for the components which need to be repaired. For one damaged component, MTTR is considered the mean value of the probabilistic distribution for the TTR. The user can choose one of these default probabilistic distributions, input a Std (or CoV) for the chosen probabilistic distribution and input a specific time (*t*). According the definition of Maintainability, it is the probability that a damaged component can be successfully repaired within a stated time interval by given procedures and resources. The specific time (*t*) is the upper limit of the stated time interval.

RAMS should calculate these probabilities for all components in the output from the LMO module and give the minimal probability as the criterion of maintainability for the array/ farm.

The probability density functions (PDFs) for these default probabilistic distributions are as follows:

1. Gaussian distribution

$$f_{\mathrm{T}}(t) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{T}}} e^{-\frac{1}{2\sigma_T^2}(t-\mu_{\mathrm{T}})^2} \tag{2}$$

   where the capitalised T denotes the stochastic variable TTR, μ_{T} and σ_{T} denote the mean and the standard deviation of the TTR, respectively.

2. LogNormal distribution

$$f_{\mathrm{T}}(t) = \frac{1}{\sqrt{2\pi}\sigma_{\ln(\mathrm{T})}} e^{-\frac{1}{2\sigma_{\ln(T)}^2}\left(t-\mu_{\ln(\mathrm{T})}\right)^2} \tag{3}$$

   where the capitalised T denotes the stochastic variable TTR, $\mu_{\ln(\mathrm{T})}$ and $\sigma_{\ln(\mathrm{T})}$ denote the mean and the standard deviation of the logged TTR, respectively.

3. Exponential distribution

$$f_{\mathrm{T}}(t) = \lambda e^{-\lambda t} \tag{4}$$

   where the capitalised T denotes the stochastic variable TTR, λ denotes the failure rate.

Outputs

The outputs comprise the following items:

- MTTR of basic components included in the hierarchies.
- The probabilities of the concerned basic components (included in the hierarchies) that can be successfully repaired within the stated time interval given procedures and resources.

3.3.4 SURVIVABILITY

3.3.4.1 Objectives

The survivability assessment aims to estimate the probability of critical components surviving the stresses/ loads during the design lifetime from the perspectives of the ultimate and the fatigue limit states. Unless otherwise specified, it only focuses on the structural/mechanical units in an array. For example, PTOs in the Energy transformation subsystem (ET), mooring lines in the Station keeping (SK) subsystem.

3.3.4.2 Inputs, Models and Outputs

Inputs

The input data for assessing the survivability are listed in Table 3.6.

TABLE 3.6 INPUTS FOR ASSESSING SURVIVABILITY

ID	Brief Description of Input Quantity	Origin of the Data	Data Model in RAMS	Unit
stress_et_uls	The ultimate stresses exerted upon the components in the ET subsystem	The ET tool	Multi-dimensional array	MPa
stress_et_fls	The fatigue stresses exerted upon the components in the ET subsystem	The ET tool	Multi-dimensional array	MPa
stress_sk_uls	The ultimate stresses exerted upon the components in the SK subsystem	The SK tool	Multi-dimensional array	MPa
stress_sk_fls	The fatigue stresses exerted upon the components in the SK subsystem	The SK tool	Multi-dimensional array	MPa
N	The number of cycles of stress ranges	The SK&ET tool	One-dimensional array	-
a	log(a) is the intercept of log(N)-axis by the linear S-N curve.	The SK&ET tool	One-dimensional array	-
m	The negative inverse slope of the linear S-N curve.	The SK&ET tool	One-dimensional array	-

Theoretical basis

See ANNEX 2 for more details of the theoretical basis.

Outputs

The outputs comprise the following items:

- the probabilities of critical components in the ET subsystem surviving the ultimate/extreme stresses/ loads.
- the probabilities of critical components in the ET subsystem surviving the fatigue stresses.
- the probabilities of critical components in the SK subsystem surviving the ultimate/extreme stresses/ loads.
- the probabilities of critical components in the SK subsystem surviving the fatigue stresses.

4. THE IMPLEMENTATION

4.1 THE ARCHITECTURE OF THE TOOL

Each module of the DTOceanPlus suite is organized in three layers:

- the Business Logic, including a set of modules, classes, libraries implementing all the functionalities of the modules.
- the Application Programming Interface (API) that describes the services requested and provided by RAMS in a human-like language.
- the Graphic User Interface (GUI) which provides the means for interacting with the user, with respect to collecting inputs from the users and displaying results, besides exporting/importing data to/from files.

Brainstorm meetings have been held to confirm the input data provided by ET, ED, SK and LMO modules to the RAMS module. All tool developers have reached an agreement on the data structures of the input and output data in order to make the smooth and automatic data flow. The following two terminologies are defined and used in this manual.

- Provider – refers to the tools providing the input data to other tools.
- Consumer – refers to the tools receiving the input data from other tools.
- Shared – refers to the case where some functions/methods are shared.

The roles of the tools as a provider or a consumer are summarized in Table 4.1. 'P', 'C' and 'S' represent the provider, the consumer, and the shared function, respectively.

TABLE 4.1 LIST OF DATA EXCHANGED BETWEEN RAMS AND OTHER TOOLS

Logic Gate	Provider/ Consumer/Shared Function					
	ED	ET	SK	LMO	RAMS	SG/SI
Hierarchy	P	P	P	C	C	-
Failure modes	P	P	P	C	C	-
Stresses/ Loads	-	P	P	-	C	-
Repair time	-	-	-	P	C	-
Downtime	-	-	-	P	C	-
Failure Events	-	-	-	S	P	-
Reliability	-	-	-	-	P	C
Availability	-	-	-	-	P	C
Maintainability	-	-	-	-	P	C
Survivability	-	-	-	-	P	C

4.1.1 BUSINESS LOGIC

4.1.1.1 Overview

Basically, the RAMS module defines five classes used to estimate the metrics, as summarized below:

- ArrayRams – a class gets the complexity level and calls the corresponding RamsReliabilityCpx#, RamsAvailabilityCpx#, RamsMaintainabilityCpx# and RamsSurvivabilityCpx# to perform assessments.
- RamsReliabilityCpx# – a class performs the component-level and the system-level reliability assessment.
  - The component-level reliability assessment simulates the time to failure (TTF) of basic components and estimates the mean time to failures (MTTFs) of basic components.
  - The system-level reliability assessment estimates the probabilities of failure (PoFs) of units at higher levels in the hierarchies as a function of time.
- RamsAvailabilityCpx# – a class estimates the availability of devices.
- RamsMaintainabilityCpx# – a class estimates the maintainability of devices.
- RamsSurvivabilityCpx# – a class estimates the survivability of devices.

A 'Cpx#' is added to the end of each class name to differentiate the three complexity levels (# is substituted by 1,2 or 3).

4.1.1.2 Class - ArrayRams

Schematic Illustration

See Figure 4.1 for the methods in the class ArrayRams.

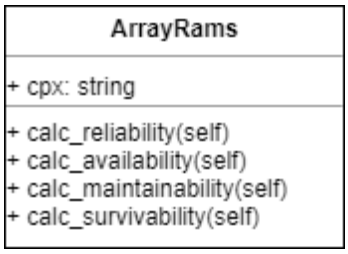


FIGURE 4.1 THE ARRAYRAMS CLASS AND THE METHODS

4.1.1.3 Class - RamsReliabilityCpx#

Schematic Illustration

See Figure 4.2 for the methods in the class RamsReliabilityCpx#.

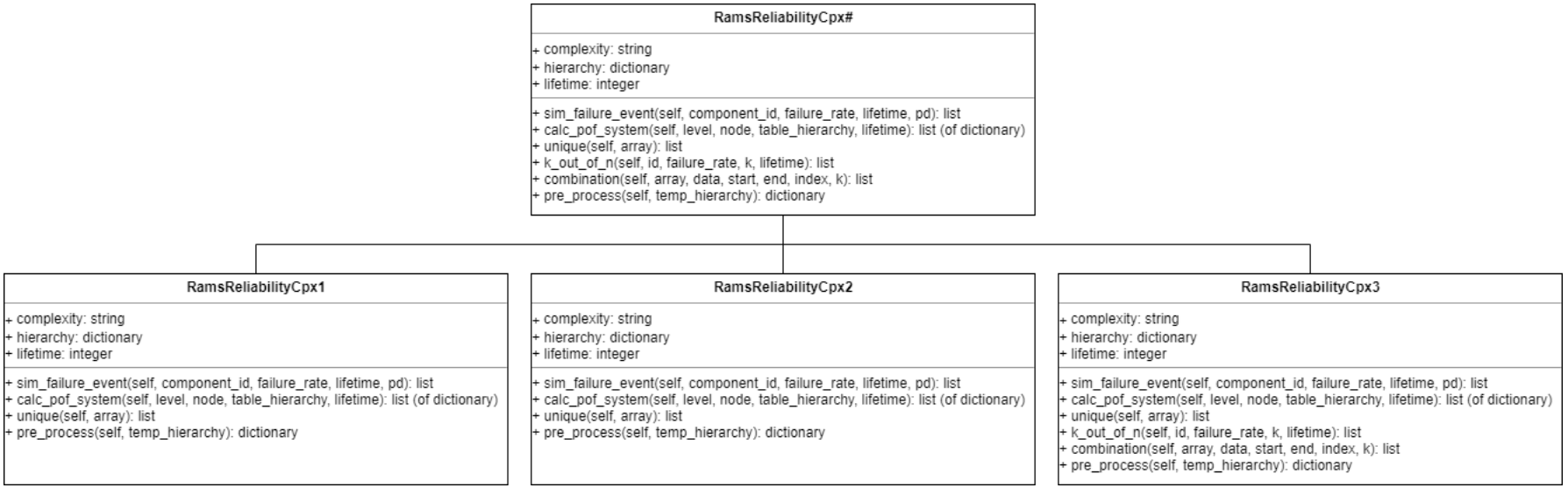


FIGURE 4.2 THE RAMSRELIABILITYCPX# CLASS AND THE METHODS

Inputs

See Table 3.3 for more details.

Methods

sim_failure_event

This method simulates the TTF of basic components in parallel and calculates the MTTFs of the basic components. The outputs are dumped to a dictionary with four keys, namely "component_id", "ts_failure_event", "mttf1" and "mttf2".

The key, "component_id", contains the ids of the basic components for which the component-level reliability is assessed. The values are stored in a one-dimensional list.

The key, "ts_failure_event", contains the simulated TTF of the basic components and the values are stored in a two-dimensional list. Each row contains the TTF of a specific basic component. There is an one-to-one mapping between "ts_failure_event" and "component_id", as shown in Figure 4.3.

The key, mttf1, contains the MTTF of the basic components in "component_id" based upon the theoretical estimation. It is assumed that the TTF follows the exponential distribution. There are two one-to-one mapping, namely between "mttf1" and "component_id' and between "mttf1" and "ts_failure_event", as shown in Figure 4.3.

The key, mttf2, contains the MTTF of the basic components in "component_id" by taking the arithmetic mean of the TTF simulated for the corresponding component in "ts_failure_event". There are two one-to-one mapping, namely between 'mttf2' and "component_id" and between "mttf2" and "ts_failure_event", as shown in Figure 4.3.

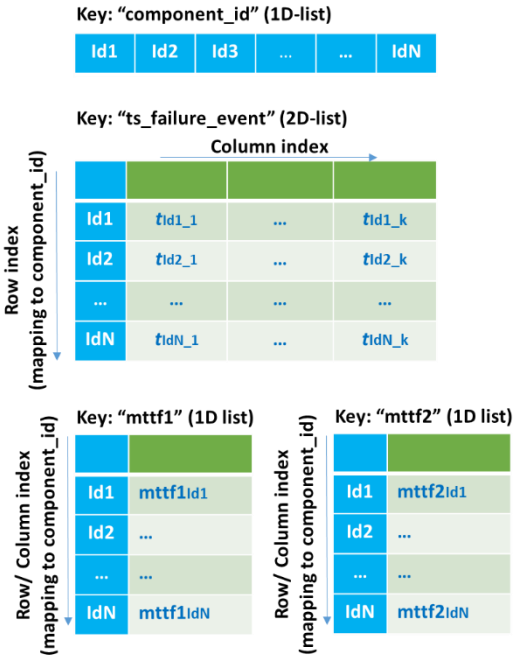


FIGURE 4.3 SKETCH ILLUSTRATING THE ONE-TO-ONE MAPPING – THE OUTPUTS OF THE COMPONENT-LEVEL RELIABILITY ASSESSMENT

calc_pof_system

This method calculates the PoFs of the units at higher levels in the hierarchies based upon the classic fault tree (FT) algorithm. The input of a hierarchical structure must strictly comply with the template, as shown in Table 2.2 in Section 3, agreed by all the module developers. The outputs are dumped to a list of dictionaries. The list has a dynamic number of entries, because of different hierarchies representing different subsystems. The dictionary has three keys, namely "name","pof_accumulate" and "pof_annual".

The key, "name", refers to the name of the item in the corresponding hierarchy, which represents a unit. The value is a string.

The key, "pof_accumulate", contains the accumulate PoFs of the unit with its name defined in "name" as a function of time. These values are stored in a one-dimensional list.

The key, "pof_annuak", contains the annual PoFs of the unit with its name defined in "name" as a function of time. These values are stored in a one-dimensional list.

unique

This method finds the unique entries in a list or array.

k_out_of_n

This method calculates the PoFs for the units which are connected to their children through a 'k-out-of-N' logic gate.

combination

This method finds out all the possible permutations of a list/ array. This method is called by k_out_of_n.

pre_process

This method decodes the raw hierarchies in the json format and identifies / dumps such fields as Design Id, System, Type, Category, Parent, Child, Gate Type, Failure_Rate_Minor and Failure_Rate_Replacement to a numpy array.

4.1.1.4 Class – RamsAvailabilityCpx#

Schematic Illustration

See Figure 4.4 for the methods in the class RamsAvailabilityCpx#.

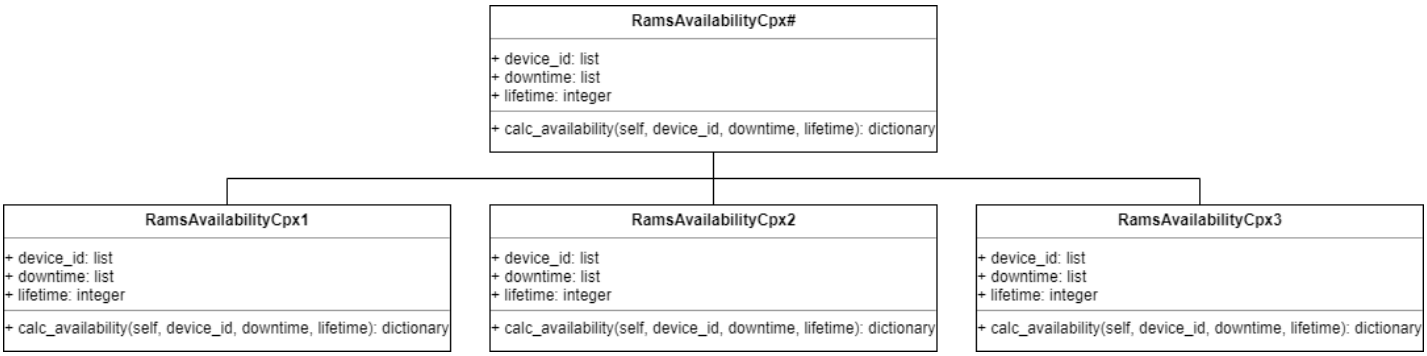


FIGURE 4.4 THE RAMSAVAILABILITYCPX# CLASS AND THE METHODS

Inputs

See Section 2.1.2.3.

Table 3.4 for more details.

Methods

calc_availability

This method estimates the availability on the basis of devices, based upon IEC 61400-25-1. The outputs are dumped to a dictionary, with two keys, namely "device_id" and "availability_tb".

The key, "device_id", contains the ids of the devices for which the availability is assessed. The values are stored in a one-dimensional list.

The key, "availability_tb", contains the availability of these devices and the values are stored in a one-dimensional list. There is a one-to-one mapping between "availability_tb" and "device_id", as shown in Figure 4.5.

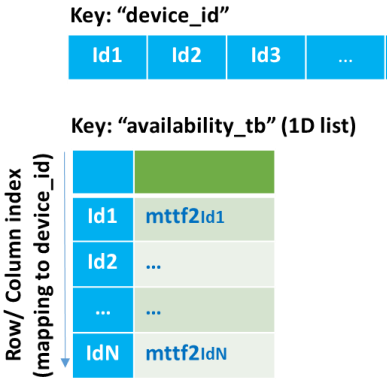


FIGURE 4.5 SKETCH ILLUSTRATING THE ONE-TO-ONE MAPPING– THE OUTPUTS OF THE AVAILABILITY ASSESSMENT

4.1.1.5 Class – RamsMaintainabilityCpx#

Schematic Illustration

See Figure 4.6 for the methods in the class RamsMaintainabilityCpx#.

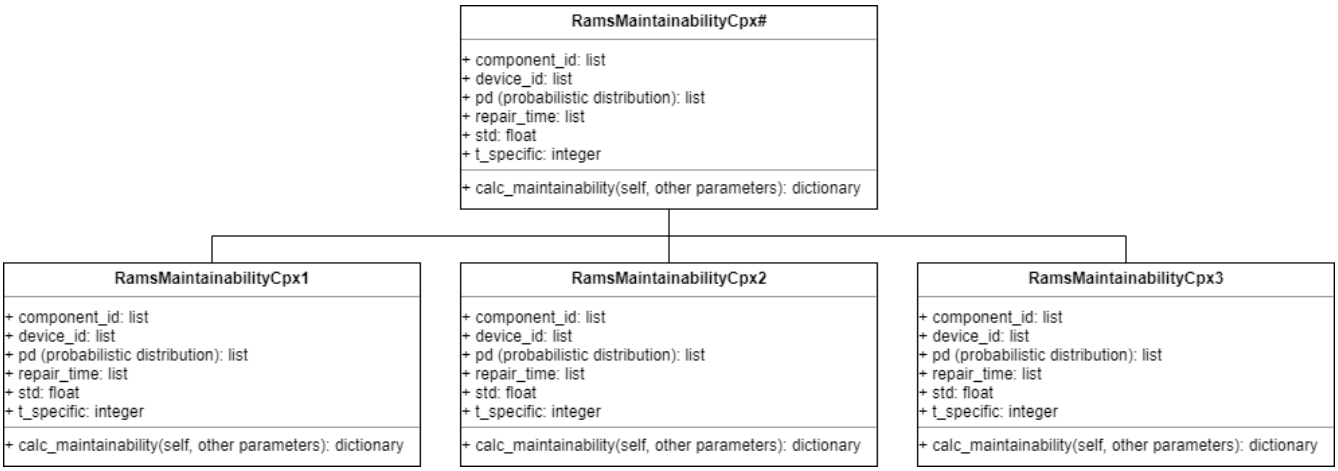


FIGURE 4.6 THE RAMSAVAILABILITYCPX# CLASS AND THE METHODS

Inputs

See Table 3.5 for more details.

Methods

calc_ maintainability

This method estimates the maintainability of devices. There are three typical probabilistic distributions (Gaussian, LogNormal and Exponential) of repair time for the end user to choose from. The outputs are dumped to a dictionary, with two keys, namely "device_id" and "probability_maintenance".

The key, "device_id", contains the ids of the devices for which the maintainability is assessed. The values are stored in a one-dimensional list.

The key, "probability_maintenance", is a 1D list including the probabilities of successfully repairing the damaged components in some devices.

4.1.1.6 Class – RamsSurvivabilityCpx#

Schematic Illustration

See Figure 4.7 for the methods in the class RamsSurvivabilityCpx#.

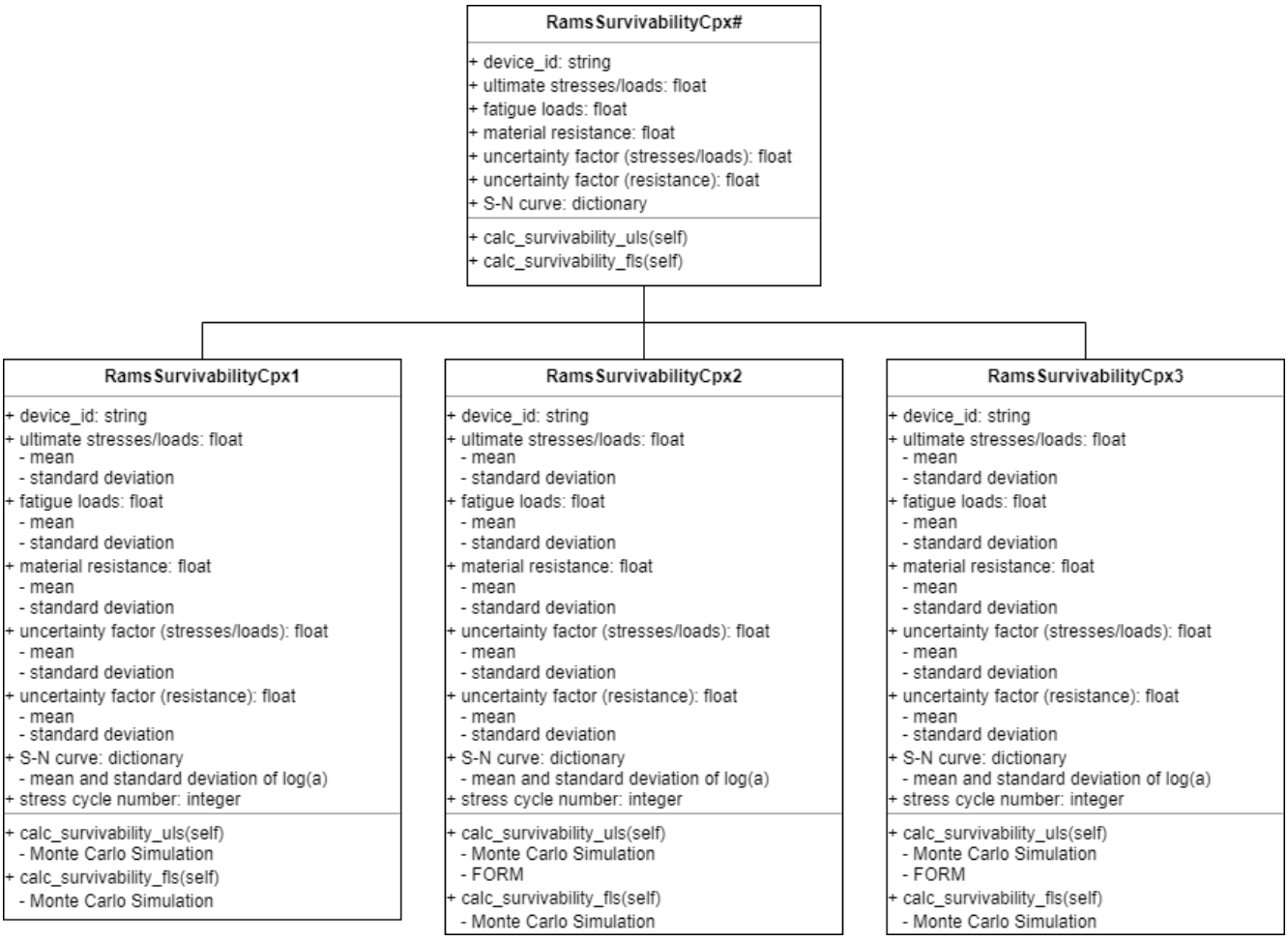


FIGURE 4.7 THE RAMSSURVIVABILITYSTGX CLASS AND THE METHODS

Inputs

See Table 3.6 for more details.

Methods

calc_survivability_uls

This method estimates the probabilities of critical components surviving the stresses/ loads during the design lifetime. The outputs are dumped to a dictionary, with two keys, namely "device_id", and "survival_uls".

The key, "device_id", contains the ids of the devices for which the availability is assessed. The values are stored in a one-dimensional list.

The key, "survival_uls", contains the survival probabilities of devices and the values are stored in a one-dimensional list. Each entry in "survival_uls" gives the survival probability of the device in the same position of the "device_id".

calc_survivability_fls

This method estimates the probabilities of critical components surviving the stresses/ loads during the design lifetime. The outputs are dumped to a dictionary, with two keys, namely "device_id", and "survival_fls".

The key, "device_id", contains the ids of the devices for which the availability is assessed. The values are stored in a one-dimensional list.

The key, "survival_fls", contains the survival probabilities of devices and the values are stored in a one-dimensional list. Each entry in "survival_fls" gives the survival probability of the device in the same position of the "device_id".

4.1.2 API

4.1.2.1 Overview

The API of the DTOceanPlus software follows a representational state transfer (REST) approach and it uses HTTP as the transport protocol. RAMS API describes the services requested and provided by the RAMS modules in a professional manner, based upon the design principles of the OpenAPI specification. The backend of the RAMS module serves to communicate with other modules. The 'GET' requests are sent to other modules to obtain the inputs to execute the assessments.

The framework of the RAMS API will be presented in the following sub-sections.

4.1.2.2 Paths

The main services provided by the RAMS module are accessed through the paths summarized in Table 4.2.

TABLE 4.2 METHODS AND DESCRIPTIONS OF PATHS

Path	Method	Description
/rams	GET	The user can get all the active RAMS applications (projects).
/rams	POST	The user can create a new RAMS application (project).
/rams/{ramsId}/inputs	GET	The user can get a specific RAMS application (project) with the specific ramsId.
/rams/{ramsId}/inputs	PUT	The user can get a specific RAMS application (project) with the specific ramsId and update it.
/rams/{ramsId}/ inputs	DELETE	The user can delete a specific RAMS application (project) with the specific ramsId and update it.
/rams/{ramsId}/reliability_component	GET	The user can get the component-level reliability results for a RAMS application (project) with the specific ramsId.

Path	Method	Description
/rams/{ramsId}/reliability_system	GET	The user can get the system-level reliability results for a RAMS application (project) with the specific ramsId.
/rams/{ramsId}/availability	GET	The user can get the availability results for a RAMS application (project) with the specific ramsId.
/rams/{ramsId}/maintainability	GET	The user can get the maintainability results for a RAMS application (project) with the specific ramsId.
/rams/{ramsId}/survivability_uls	GET	The user can get the survivability results for the ultimate limit state for a RAMS application (project) with the specific ramsId.
/rams/{ramsId}/survivability_fls	GET	The user can get the survivability results for the fatigue limit state for a RAMS application (project) with the specific ramsId.

4.1.3 SCHEMAS

JSON should be briefly reviewed, before schemas are interpreted. JSON stands for JavaScript Object Notation. People can convert any JavaScript object into JSON and send JSON to the server. JSON Schema is a powerful tool for validating the structure of JSON data.

The input and output schemas in the RAMS module are given in Table 4.3.

TABLE 4.3 SCHEMAS INCLUDED IN THE RAMS MODULE

Name of Schema	Description
InputsGeneral.yaml	It defines the data type for the general information of a RAMS application (project), including 'id', 'title', 'desc', 'complexity', 'status' and 'tags'
InputsSpecific.yaml	It defines the data type for the full information of a RAMS application (project), including 'id', 'title', 'desc', 'complexity', 'status', 'tags', 'hierarchy_ed', 'hierarchy_et', 'hierarchy_sk', 'downtime', 'repair-time', 'stress_uls_et', 'stress_fls_et' and 'stress_sk'
Availability.yaml	It defines the data type of the availability assessment results.
ReliabilityCompoment.yaml	It defines the data type of the component-level reliability assessment results.
ReliabilitySystem.yaml	It defines the data type of the system-level reliability assessment results.
Maintainability.yaml	It defines the data type of the maintainability assessment results.
SurvivabiltyUls.yaml	It defines the data type of the survivability assessment results for the ultimate limit state.
SurvivabiltyFls.yaml	It defines the data type of the survivability assessment results for the fatigue limit state.

4.1.4 GUI

The GUI of all DTOceanPlus modules will be based on the same libraries to guarantee a consistent visual look.

The GUI of the RAMS module will be included into the main module and, as it could be seen from the mock-up in Figure 4.8, will generally consist of two parts.

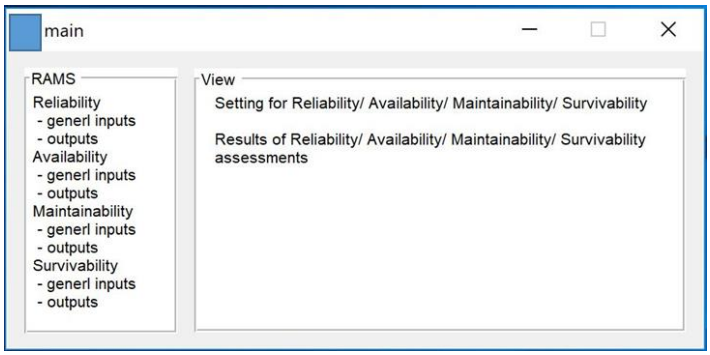


FIGURE 4.8 GUI MOCK-UP OF THE RAMS MODULE

4.1.5 THE TECHNOLOGIES

The classes and functions in Business Logic and the API of RAMS have been developed in Python version 3.7. The installation of the module requires the following packages:

- Numpy
- Flask
- Flask-SQLalchemy
- Flask-Marshmallow
- Flask-cors
- Request
- Pandas
- Scipy
- Math
- Csv

The API will rely on OpenAPI specification v3.0.2. The GUI of the module will be developed in Vue.js, using the library Element-UI.

4.2 TESTING AND VERIFICATION

4.2.1 OVERVIEW

Independent tests of classes and functions in Business Logic, Back End and API should be performed in order to debug the code. For Business Logic and Back End, the Pytest package is used to test the

potential errors and quantify the coverage rates. For API, the Dredd package is used to test the logic of API.

4.2.2 PYTEST

Pytest is a testing framework which allows us to write test codes using python and can be used for software testing at all levels. Ideally, the acceptance criterion of coverage rate is 100%. For the code coverage rates are shown in Figure 4.9.

Name	Stmts	Miss	Cover
src\dtop_rams__init__.py	0	0	100%
src\dtop_rams\business__init__.py	48	12	75%
src\dtop_rams\business\cpx1\RamsAvailabilityCpx1.py	21	0	100%
src\dtop_rams\business\cpx1\RamsMaintainabilityCpx1.py	28	2	93%
src\dtop_rams\business\cpx1\RamsReliabilityCpx1.py	314	38	88%
src\dtop_rams\business\cpx1\RamsSurvivabilityCpx1.py	153	3	98%
src\dtop_rams\business\cpx1__init__.py	4	0	100%
src\dtop_rams\business\cpx2\RamsAvailabilityCpx2.py	21	0	100%
src\dtop_rams\business\cpx2\RamsMaintainabilityCpx2.py	28	2	93%
src\dtop_rams\business\cpx2\RamsReliabilityCpx2.py	301	26	91%
src\dtop_rams\business\cpx2\RamsSurvivabilityCpx2.py	281	3	99%
src\dtop_rams\business\cpx2__init__.py	4	0	100%
src\dtop_rams\business\cpx3\RamsAvailabilityCpx3.py	21	0	100%
src\dtop_rams\business\cpx3\RamsMaintainabilityCpx3.py	28	2	93%
src\dtop_rams\business\cpx3\RamsReliabilityCpx3.py	301	26	91%
src\dtop_rams\business\cpx3\RamsSurvivabilityCpx3.py	280	3	99%
src\dtop_rams\business\cpx3__init__.py	4	0	100%
src\dtop_rams\service__init__.py	27	6	78%
src\dtop_rams\service\api__init__.py	55	4	93%
src\dtop_rams\service\api\assessment__init__.py	150	48	68%
src\dtop_rams\service\api\core__init__.py	6	3	50%
src\dtop_rams\service\config.py	8	0	100%
src\dtop_rams\service\db.py	28	9	68%
src\dtop_rams\service\gui__init__.py	0	0	100%
src\dtop_rams\service\gui\back_end__init__.py	4	4	0%
src\dtop_rams\service\gui\main__init__.py	3	3	0%
src\dtop_rams\service\models.py	39	2	95%
src\dtop_rams\service\schemas.py	9	0	100%
TOTAL	2166	196	91%

FIGURE 4.9 CODE COVERAGE RATES

4.2.3 DREDD TEST

Dredd is a language-agnostic command-line tool for validating API description document against backend implementation of the API. Generally, the procedure for running Dredd tests is as follows:

- take your API description document
- create expectations based on requests and responses documented in the document
- make requests to tested API
- check whether API responses match the documented responses
- report the results

Dredd should be configured through a continuous integration (CI) file, which is named 'dredd-local.yml' in the RAMS module. Besides it, hooks should be defined in a file which is called by Dredd. The file is name 'hooks.py' in the RAMS module. The hooks are summarized in Table 4.4.

TABLE 4.4 SET-UP OF HOOKS

Name of Schema	Description
Hooks for /rams	
@hooks.before("/rams > Returns all the active RAMS projects > 200 > application/json")	To test if the registered function under the blueprint @bp.route('/rams', methods='GET') can return all the RAMS projects, as required in the OpenAPI.
@hooks.before("/rams > Returns all the active RAMS projects > 404")	To test if the registered function under the blueprint @bp.route('/rams', methods='GET') can return an error message due to no created project, as required in the OpenAPI.
@hooks.before("/rams > Creates a new rams project > 201")	To test if the registered function under the blueprint @bp.route('/rams', methods='POST') can create a RAMS project, as required in the OpenAPI.
@hooks.before("/rams > Creates a new rams project > 400")	To test if the registered function under the blueprint @bp.route('/rams', methods='POST') can return an error message due to the wrong required information (e.g. a wrong ramsID) as required in the OpenAPI.
Hooks for /rams/{ramsId}/inputs	
@hooks.before("/rams/{ramsId}/inputs > Returns the specific RAMS project > 200 > application/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>', methods='GET') can return the RAMS project with the specific ramsId as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/inputs > Returns the specific RAMS project > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>', methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/inputs > Modifies a specific rams project > 201 > application/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>', methods='PUT') can modify the information in an existing project as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/inputs > Modifies a specific rams project > 400")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>', methods='PUT') can return an error message due to a wrong ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/inputs > Delete a specific RAMS project > 200")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>', methods='DELETE') can delete the RAMS project with the specific ramsId as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/inputs > Delete a specific RAMS project > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>', methods=' DELETE ') can return an error message due to a wrong ramsId as required in the OpenAPI.
Hooks for /rams/{ramsId}/reliability_component	
@hooks.before("/rams/{ramsId}/reliability_component > Returns the reliability assessment results > 200 #> application/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/reliability_component ', methods='GET') can return the component-level reliability assessment results for the RAMS project with the specific ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/reliability_component > Returns the reliability assessment results > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/reliability_component ', methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.

Name of Schema	Description
Hooks for /rams/{ramsId}/reliability_system	
@hooks.before("/rams/{ramsId}/reliability_system > Returns the system-level reliability assessment #results > 200 > application/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/reliability_system ', methods='GET') can return the system-level reliability assessment results for the RAMS project with the specific ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/reliability_system > Returns the system-level reliability assessment #results > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/reliability_system ', methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.
Hooks for /rams/{ramsId}/availability	
@hooks.before("/rams/{ramsId}/availability > Returns the availability of devices > 200 > application/#json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ availability , methods='GET') can return the availability of devices for the RAMS project with the specific ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/availability > Returns the availability of devices > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ availability , methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.
Hooks for /rams/{ramsId}/maintainability	
@hooks.before("/rams/{ramsId}/maintainability > Returns the maintainability assessment results > 200 > #pplication/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ maintainability , methods='GET') can return the maintainability of devices for the RAMS project with the specific ramsId as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/maintainability > Returns the maintainability assessment results > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ maintainability , methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.
Hooks for /rams/{ramsId}/survivability_uls	
@hooks.before("/rams/{ramsId}/survivability_uls > Returns the probability of surviving the ultimate #loads > 200 > application/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ survivability_uls , methods='GET') can return the survival probabilities of devices for the RAMS project with the specific ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/survivability_uls > Returns the probability of surviving the ultimate #loads > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ survivability_uls , methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.
Hooks for /rams/{ramsId}/survivability_fls	
@hooks.before("/rams/{ramsId}/survivability_fls > Returns the probability of surviving the fatigue loads > 200 > application/json")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ survivability_Fls , methods='GET') can return the survival probabilities of devices for the RAMS project with the specific ramsId, as required in the OpenAPI.
@hooks.before("/rams/{ramsId}/survivability_fls > Returns the probability of surviving the fatigue #loads > 404")	To test if the registered function under the blueprint @bp.route('/rams/<ramsId>}/ survivability_fls , methods='GET') can return an error message due to a wrong ramsId, as required in the OpenAPI.

5. EXAMPLES

In this section, an example for each functionality implemented in RAMS has been performed and the outputs are presented. It is important to stress that specified inputs were generated for illustration purposes only and do not correspond to any specific project or technology. Consequently, the obtained outputs do not hold any meaning and are not necessarily realistic. These were chosen as merely representative values to be used as a demonstration of the computational capabilities of the RAMS module.

5.1 RELIABILITY

A virtual marine energy farm will be used to demonstrate how reliability is assessed will be presented in the following subsections.

Basically, a marine energy conversion system is composed of three major sub-systems, namely

- the energy transformation (ET) subsystem
- the energy delivery (ED) subsystem
- the station keeping (SK) subsystem

It should be noted that the deployment design modules are all at the stage of concept design, and there are no concrete sketches showing the general layout of the corresponding subsystems. Therefore, there are only tables defining the hierarchies of these subsystems in this subsection. The design lifetime is assumed to be 20 years in this subsection.

The target annual probability of failure (PoF) is 10^{-3} recommended in IEC TS 62600-2 (Table 5-2) [12], which is the primary design standard for marine energy conversion systems. The objective of reliability assessment is to:

- check whether or not the estimated annual PoFs of all the subsystems are less than this target during the design lifetime. The maximum annual PoF will be given.

5.1.1 ED SUBSYSTEM

5.1.1.1 Inputs

In this example, the ED subsystem is composed of three independent energy transfer routes, i.e. ED_OEC# (#=1, 2, 3). The ED subsystem fails, if all the energy transfer routes fail. According to the rule defined in Table 2.1, an 'OR' is inserted (an 'AND' gate should be inserted in the fault tree). ED_OEC# contains only one Route#. The symbol # represents _1_1, _2_1 or _3_1. Each Route# contains two basic components, EC# (#=1 ,2, 3) and CON# ((#=1 ,2, 3). Each Route# fails if either EC# or CON# fails. According to the rule defined in Table 2.1, an 'AND' is inserted (actually, an 'OR' gate should be inserted in the fault tree). See the hierarchy defined in

Table 5.1. It should be noted that: the failure rates are only estimated based upon the assumption that the TTF of basic components follows the exponential distribution in this example.

TABLE 5.1 THE HIERARCHY OF THE ED SUBSYSTEM

System	Name of Node	Design Id	Node Type	Node Subtype	Category	Parent	Child	Gate Type	Failure Rate Repair(1) [1/hour]	Failure Rate Replacement(2) [1/hour]
ED	ED Subsystem	"NA"	System	System	Level 3	"NA"	[ED_OEC1, ED_OEC2, ED_OEC3]	OR	"NA"	"NA"
ED	ED_OEC1	"NA"	Subsystem	Subsystem	Level 2	"NA"	[Route1_1]	OR	"NA"	"NA"
ED	ED_OEC2	"NA"	Subsystem	Subsystem	Level 2	"NA"	[Route2_1]	OR	"NA"	"NA"
ED	ED_OEC3	"NA"	Subsystem	Subsystem	Level 2	"NA"	[Route3_1]	OR	"NA"	"NA"
ED	Route1_1	Route1_1	Energy route	Energy route	Level 1	ED_OEC1	[EC1, CON1]	AND	"NA"	"NA"
ED	Route2_1	Route2_1	Energy route	Energy route	Level 1	ED_OEC2	[EC2, CON2]	AND	"NA"	"NA"
ED	Route3_1	Route3_1	Energy route	Energy route	Level 1	ED_OEC3	[EC3, CON3]	AND	"NA"	"NA"
ED	CON1	CON1	Component	Component	Level 0	Route1_1	"NA"	"NA"	5.71E-07(3)	"NA"
ED	CON2	CON2	Component	Component	Level 0	Route2_1	"NA"	"NA"	5.71E-07(3)	"NA"
ED	CON3	CON3	Component	Component	Level 0	Route3_1	"NA"	"NA"	5.71E-07(3)	"NA"
ED	EC1	EC1	Component	Component	Level 0	Route1_1	"NA"	"NA"	5.71E-07(3)	"NA"
ED	EC2	EC2	Component	Component	Level 0	Route2_1	"NA"	"NA"	5.71E-07(3)	"NA"
ED	EC3	EC3	Component	Component	Level 0	Route3_1	"NA"	"NA"	5.71E-07(3)	"NA"

Note:
1) These failure rates refer to failure mode 1 mentioned in Section 2.1.2.1. Failure mode 1 refers to the failures which can be restored through repair.
2) These failure rates refer to failure mode 2 mentioned in Section 2.1.2.1. Failure mode 2 refers to the cases in which the severely failed components can only be replaced.
3) These failure rates correspond to the mean time to failure (MTTF) of 200 years.

5.1.1.2 Results

Component-level

The core algorithm is based on the stochastic sampling, which has uncertainty. Because of uncertainty of the sampling algorithm, TTFs should be different between various realizations. For each basic component, the estimated time to failure (TTF) is dumped to a list. It is possible to obtain an empty list, i.e. '[]', which indicates that no failure occurs during the design lifetime. If failures are estimated, the TTFs are ordered in a list. See one realization for each basic component in Table 5.2. It should be noted that the TTF for each basic component only represent one realization. As mentioned in Section 2.2, TTF is assumed to follow the exponential distribution. So, the theoretical MTTF is just the reciprocal of the failure rate.

TABLE 5.2 RESULTS OF TTF OF BASIC COMPONENTS IN THE ED SUBSYSTEM

Component ID	TTF [hour]
CON_1	[145,738] (about 16.6 years)
CON_2	[44,748] (about 5.1 years)
CON_3	[168,414] (about 19.2 years)
EC_1	[33,242] (about years 3.8)
EC_2	[24,687] (about 2.8 years)
EC_3	[146,814] (about 16.8 years)

System-level

In the code for assessing the system-level reliability, the accumulated PoFs of the units at higher levels in the hierarchy are first calculated, and then the accumulated PoFs are differentiated to obtain the annual PoFs which are compared with the target PoF. The annual PoF as a function of time for the ED subsystem is shown in Figure 5.1. The maximum annual PoF is 7.75×10^{-4}. The annual PoFs indicate that the ED subsystem satisfies the design requirement on the safety, i.e. any PoF is lower than the target PoF recommended in [12].

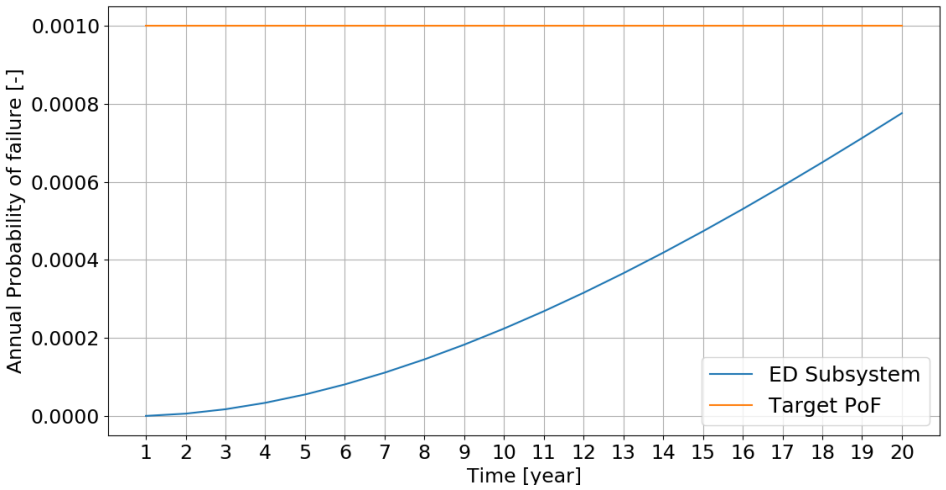


FIGURE 5.1 THE POF OF THE ED SUBSYSTEM AS A FUNCTION OF TIME

5.1.2 ET SUBSYSTEM

5.1.2.1 Inputs

In this example, there is only one device considered in the array. So, the ET subsystem of the array only contains the two PTOs attached to this device. The ET subsystem fails if both of PTOs fail. According to the rule defined in Table 2.1, an 'OR' is inserted (an 'AND' gate should be inserted in the fault tree). Each PTO is composed of three basic components, namely H2M#, M2E# and E2G#. The symbol # represents _1_1 or _1_2. The PTO fails, if either of H2M#, M2E# and E2G# fails. According to the rule defined in Table 2.1, an 'AND' is inserted (an 'OR' gate should be inserted in the fault tree). See the hierarchy in Table 5.3.

5.1.2.2 Results

Component-level

The core algorithm is based on the stochastic sampling, which has uncertainty. Because of uncertainty of the sampling algorithm, TTFs should be different between various realizations. For each basic component, the estimated time to failure (TTF) is dumped to a list. It is possible to obtain an empty list, i.e. '[]', which indicates that no failure occurs during the design lifetime. If failures are estimated, the TTFs are ordered in a list. See one realization for each basic component in Table 5.4. It should be noted that the TTF for each basic component only represent one realization. As mentioned in Section 2.2, TTF is assumed to follow the exponential distribution. So, the theoretical MTTF is just the reciprocal of the failure rate.

TABLE 5.3 THE HIERARCHY OF THE ET SUBSYSTEM

System	Name of Node	Design Id	Node Type	Node Subtype	Category	Parent	Child	Gate Type	Failure Rate Repair(1) [1/hour]	Failure Rate Replacement(2) [1/hour]
ET	Device_01	"NA"	System	System	Level 2	"NA"	[PTO_1_1, PTO_1_2]	OR	"NA"	"NA"
ET	PTO_1_1	"NA"	PTO	PTO	Level 1	Device_1	[H2M_1_1, M2E_1_1, E2G_1_1]	AND	"NA"	"NA"
ET	PTO_1_2	"NA"	PTO	PTO	Level 1	Device_1	[H2M_1_2, M2E_1_2, E2G_1_2]	AND	"NA"	"NA"
ET	H2M_1_1	H2M_1_1	Component	Component	Level 0	PTO_1_1	"NA"	"NA"	1.92E-07(3)	"NA"
ET	H2M_1_2	H2M_1_2	Component	Component	Level 0	PTO_1_2	"NA"	"NA"	1.92E-07(3)	"NA"
ET	M2E_1_1	M2E_1_1	Component	Component	Level 0	PTO_1_1	"NA"	"NA"	1.92E-07(3)	"NA"
ET	M2E_1_2	M2E_1_2	Component	Component	Level 0	PTO_1_2	"NA"	"NA"	1.92E-07(3)	"NA"
ET	E2G_1_1	E2G_1_1	Component	Component	Level 0	PTO_1_1	"NA"	"NA"	1.92E-07(3)	"NA"
ET	E2G_1_2	E2G_1_2	Component	Component	Level 0	PTO_1_2	"NA"	"NA"	1.92E-07(3)	"NA"

Note:
1) These failure rates refer to failure mode 1 mentioned in Section 2.1.2.1. Failure mode 1 refers to the failures which can be restored through repair.
2) These failure rates refer to failure mode 2 mentioned in Section 2.1.2.1. Failure mode 2 refers to the cases in which the severely failed components can only be replaced.
3) These failure rates correspond to the mean time to failure (MTTF) of 595 years.

TABLE 5.4 RESULTS OF TTF OF BASIC COMPONENTS IN THE ET SUBSYSTEM

Component ID	TTF [hour]
H2M_1_1	[108,199] (about 12.4 years)
H2M_1_2	[79,658] (about 9.1 years)
M2E_1_1	[163,592] (about 18.7 years)
M2E_1_2	[98,181] (about 11.2 years)
E2G_1_1	[157,463] (about 18.0 years)
E2G_1_2	[95,996] (about 11.0 years)

System-level

In the code for assessing the system-level reliability, the accumulated PoFs of the units at higher levels in the hierarchy are first calculated, and then the accumulated PoFs are differentiated to obtain the annual PoFs which are compared with the target PoF. The annual PoF as a function of time for the ET subsystem is shown in Figure 5.2. The maximum annual PoF is 8.57×10^{-4}. The annual PoFs indicate that the ED subsystem satisfies the design requirement on the safety, i.e. any PoF is lower than the target PoF recommended in [12].

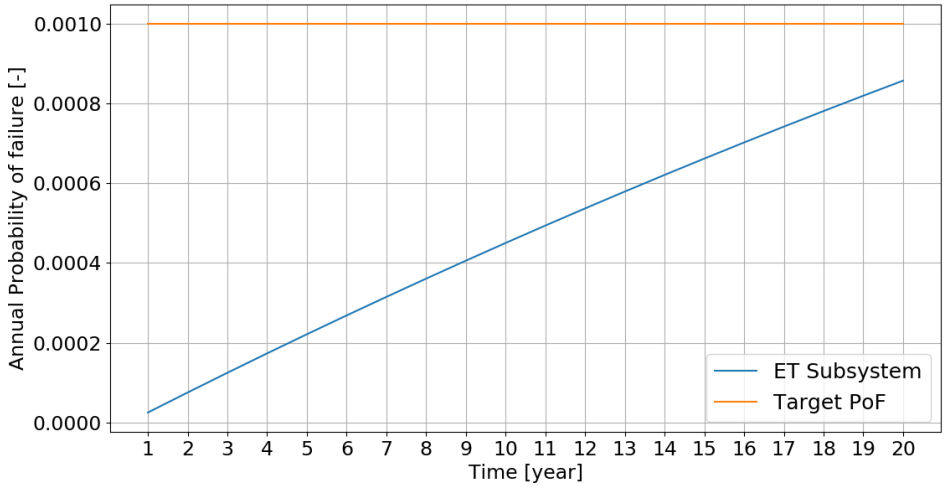


FIGURE 5.2 THE POF OF THE ET SUBSYSTEM AS A FUNCTION OF TIME

5.1.3 SK SUBSYSTEM

5.1.3.1 Inputs

In this example, there is only one device considered in the array. So, the SK subsystem of the array only contains the four mooring lines attached to this device. The SK subsystem fails, if all mooring lines fail. According to the rule defined in Table 2.1, an 'OR' is inserted (an 'AND' gate should be inserted in the fault tree). See the hierarchy in Table 5.5.

5.1.3.2 Results

Component-level

The core algorithm is based on the stochastic sampling, which has uncertainty. Because of uncertainty of the sampling algorithm, TTFs should be different between various realizations. For each basic component, the estimated time to failure (TTF) is dumped to a list. It is possible to obtain an empty list, i.e. '[]', which indicates that no failure occurs during the design lifetime. If failures are estimated, the TTFs are ordered in a list. See one realization for each basic component in Table 5.6. It should be noted that the TTF for each basic component only represent one realization. As mentioned in Section 2.2, TTF is assumed to follow the exponential distribution. So, the theoretical MTTF is just the reciprocal of the failure rate.

TABLE 5.5 THE HIERARCHY OF THE SK SUBSYSTEM

System	Name of Node	Design Id	Node Type	Node Subtype	Category	Parent	Child	Gate Type	Failure Rate Repair(1) [1/hour]	Failure Rate Replacement(2) [1/hour]
SK	SK Subsystem	"NA"	System	System	Level 1	"NA"	[ML1, ML2, ML3, ML4]	OR	"NA"	"NA"
SK	ML1	ML1	Component	Component	Level 0	SK Subsystem	"NA"	"NA"	1.95E-06(3)	"NA"
SK	ML2	ML2	Component	Component	Level 0	SK Subsystem	"NA"	"NA"	1.95E-06(3)	"NA"
SK	ML3	ML3	Component	Component	Level 0	SK Subsystem	"NA"	"NA"	1.95E-06(3)	"NA"
SK	ML4	ML4	Component	Component	Level 0	SK Subsystem	"NA"	"NA"	1.95E-06(3)	"NA"

Note:
1) These failure rates refer to failure mode 1 mentioned in Section 2.1.2.1. Failure mode 1 refers to the failures which can be restored through repair.
2) These failure rates refer to failure mode 2 mentioned in Section 2.1.2.1. Failure mode 2 refers to the cases in which the severely failed components can only be replaced.
3) These failure rates correspond to the mean time to failure (MTTF) of 58 years.

TABLE 5.6 RESULTS OF TTF OF BASIC COMPONENTS IN THE SK SUBSYSTEM

Component ID	TTF [hour]
ML1	[16,085, 67,532, 166,067] (about 1.84, 7.7, 19.0 years)
ML2	[30,297, 111,301] (about 3.5, 12.7 years)
ML3	[24,193, 148,908] (about 2.8, 17.0 years)
ML4	[101,507, 149,302] (about 11.6, 17.0 years)

System-level

In the code for assessing the system-level reliability, the accumulated PoFs of the units at higher levels in the hierarchy are first calculated, and then the accumulated PoFs are differentiated to obtain the annual PoFs which are compared with the target PoF. The annual PoF as a function of time for the SK subsystem is shown in **¡Error! No se encuentra el origen de la referencia.**. The maximum annual PoF is 1.11×10^{-3}. The annual PoFs indicate that the ED subsystem does satisfy the design requirement on the safety after Year 19. The suggestion can be to choose more reliable basic components with lower failure rates than those used in this example.

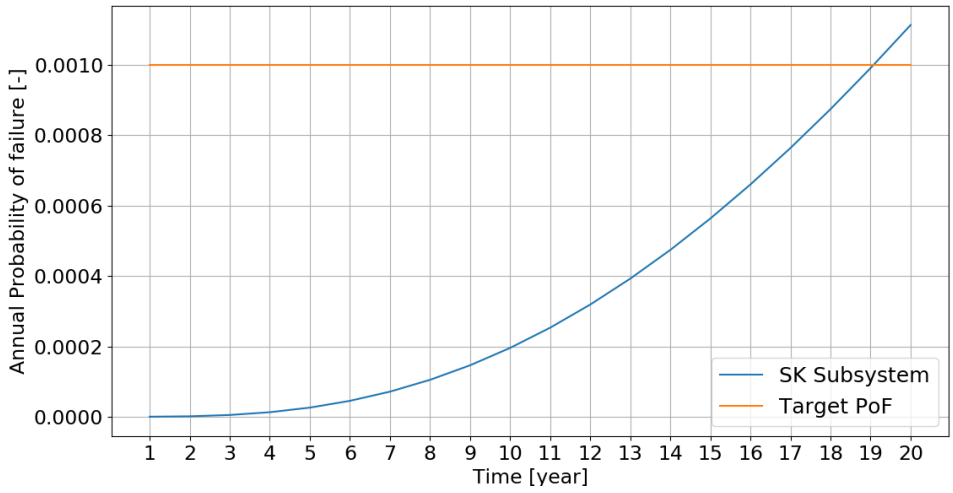


FIGURE 5.3 THE POF OF THE SK SUBSYSTEM AS A FUNCTION OF TIME

5.2 AVAILABILITY

5.2.1 INPUTS

The business logic is the same for three complexity levels. The generic inputs are given in Table 5.7. The number in each entry is the downtime in the corresponding month of the year. It is assumed that the design lifetime is 10 years, because the input of downtime only represents 10 years.

TABLE 5.7 AN EXAMPLE OF DOWNTIME FOR TWO DEVICES

Month	Downtime [hour]									
	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7	Year 8	Year 9	Year 10
Device 11										
Jan.	744	0	0	0	0	0	0	0	0	0
Feb.	0	0	0	0	0	0	0	0	0	0
Mar.	0	43	0	0	0	0	432	15	0	15
April	0	0	0	0	0	564	0	0	0	0
May	0	0	0	0	0	0	0	0	0	0
Jun.	0	0	196	0	124	0	532	0	0	0
Jul.	0	0	0	643	0	324	0	0	0	0
Aug.	56	0	0	0	56	0	56	0	0	0
Sep.	0	145	0	0	0	0	0	0	0	0
Oct.	0	0	0	88	0	0	0	0	0	0
Nov.	0	0	0	0	0	0	0	0	0	0
Dec.	0	0	0	0	0	0	159	0	0	0
Device 12										
Jan.	744	0	0	0	0	0	0	0	0	0
Feb.	0	0	0	0	0	0	0	0	0	0
Mar.	0	43	0	0	0	0	0	15	0	0
April	0	0	0	0	0	532	0	0	0	180
May	0	0	0	0	0	42	0	0	0	0
Jun.	0	0	0	0	124	0	0	0	0	0
Jul.	0	140	0	643	0	324	0	0	0	0
Aug.	56	0	0	0	0	0	0	0	0	720
Sep.	0	0	543	0	0	0	0	0	0	720
Oct.	0	0	0	643	0	0	120	453	0	744
Nov.	0	0	0	0	47	0	0	0	0	720
Dec.	0	0	0	0	96	0	0	0	0	744

5.2.2 RESULTS

All the downtime should be summed for each device and the availability of each device is the ratio of the uptime (design lifetime - downtime) to the design lifetime. The results of availability are given in Table 5.8. The array availability can be approximately taken as the arithmetic average of the device availability.

TABLE 5.8 RESULTS OF AVAILABILITY ASSESSMENT

Unit	Availability [%]
Device11	95.21
Device12	90.42
Array	92.82

5.3 MAINTAINABILITY

5.3.1 INPUTS

The business logic is the same for three complexity levels. The generic inputs extracted from the data set exported by the LMO module are given in Table 5.9. According to the coding rule in the LMO module, the repair time is only estimated for the components which fail during the design lifetime. The TTF is predicted by using the function in the shared library. If a component will not be subject to any failure, the corresponding element in the column MTTR will be 'NA'.

The probabilistic distribution of repair time is assumed to follow the Gaussian distribution. The user may want to know the probability that the damage component can be successfully repaired in t=24 hours. In light of the high uncertainty in offshore on-site repair, a high standard deviation of 18 hours is assumed (in reality, this should be input by the user).

TABLE 5.9 A SAMPLE OF MTTR

Operational Id	Component Id	MTTR [hour]
op13_1	ml11	NA
op12_1	ml12	NA
op13_2	ml13	43
op12_2	ml14	NA

5.3.2 RESULTS

The probability that the component 'mul13' can be successfully repaired within t=24 hours is 14.6%. Since there is only one component, the maintainability of the whole array is 14.6%.

5.4 SURVIVABILITY

Basically, the classic structural reliability analysis methods, e.g. Monte Carlo and First Order Reliability Method (FORM), are used in the code. The critical structural components in the SK module will be used as an example in this subsection.

It is assumed that there is a sample array containing four devices. Each device is kept in place through four mooring lines. Tension is assumed to be the dominating load for both ultimate and fatigue limit states. Unless otherwise specified, the load refers to tension and the stress ranges refer to those

caused by the alternating tensions. Monte Carlo Method is used. The design lifetime is assumed to be 20 years in this subsection.

5.4.1 INPUTS

5.4.1.1 Ultimate Limit State

According to the extreme strength analysis performed in the SK module, the four devices are subject to the identical ultimate loads. The ultimate loads exerted on the four mooring lines of one device are used to estimate the mean load. The standard deviation or coefficient of variance (CoV) is estimated based upon the engineering experience. The maximum breaking load (MBL) is the material resistance of the mooring lines, provided by the SK module. In nature, both load and MBL are stochastic variables. So, two uncertainty factors, which are also stochastic variables, are applied to load and MBL, respectively. See Table 5.10 for these inputs.

Note: Since these variables are assumed to follow the LogNormal distribution, the mean values and standard deviations of the logged variables are given as follows:

$$\mu_{\ln(S)} = 11.2873 \qquad \sigma_{\ln(S)} = 0.2936$$

$$\mu_{\ln(R)} = 12.4866 \qquad \sigma_{\ln(R)} = 0.1980$$

$$\mu_{\ln(X_S)} = -0.004975 \qquad \sigma_{\ln(X_S)} = 0.09975$$

$$\mu_{\ln(X_S)} = -0.004975 \qquad \sigma_{\ln(X_S)} = 0.09975$$

TABLE 5.10 INPUTS FOR THE ULTIMATE SURVIVABILITY ASSESSMENT

Variable	Description	Probabilistic Distribution	Mean μ	CoV
S	The load exerted upon the mooring lines	LogNormal	83,313 N	0.3
R	The material resistance of mooring lines	LogNormal	270,000 N	0.2
X_S	The uncertainty factor for the load	LogNormal	1.0	0.1
X_R	The uncertainty factor for the MBL	LogNormal	1.0	0.1

5.4.1.2 Fatigue Limit State

According to the fatigue analysis performed in the SK module, the four devices are subject to the identical fatigue loads. The fatigue stresses for one device are used in this subsection.

The stress ranges and the corresponding cumulative distribution functions, both of which are discretized to 10 bins, will be used to fit the shape and scale parameters of the 2-parameter Weibull distribution of the long-term stress ranges. The procedure for fitting the parameters is as follows:

The cumulative 2-parameter Weibull distribution function is defined in Eq. (5).

$$F(\Delta s) = 1 - e^{-(\Delta s/B)^A} \tag{5}$$

Take logarithm of both sides in Eq. (5) twice to obtain Eq. (6)

$$\ln\{-\ln[1-F(\Delta s)]\} = A[\ln(\Delta s) - \ln(B)] \tag{6}$$

Let

$$\mathrm{Y} = \ln\{-\ln[1-F(\Delta s)]\} \tag{7}$$

$$\mathrm{X} = \ln(\Delta s) \tag{8}$$

where Δs denotes the stress range. $F(\Delta s)$ denotes the cumulative distribution function of the stress range. *A* and *B* are the shape and scale parameters of the 2-parameter Weibull distribution. A linear regression can be performed through the transformations in Eqs.(7)-(8). The coefficient of determination (R^2) is calculated to evaluate the goodness of fit. The better a linear regression fits the data, the closer the value of R2 is to 1. The equations for calculating R^2 are as follows:

$$R^2 = 1 - \frac{\sum_i (Y_i - \bar{Y})^2}{\sum_i (A[X_i - \ln(B)] - \bar{Y})^2} \tag{9}$$

where Y_i is the i-th observation. $\bar{Y}$ is the mean of all the observations. The term $A[X_i - \ln(B)]$ denotes the fitted *Y* at each X_i.

As mentioned in Section 2.1.2.2, the S-N curve-based approach is used to assess the survivability from the perspective of fatigue limit state. The variables include the number of cycles of stress ranges, the S-N curve parameters *a* and *m*, and the Weibull distribution parameters *A* and *B*. See Table 5.11 for these inputs.

TABLE 5.11 INPUTS FOR THE FATIGUE SURVIVABILITY ASSESSMENT

Variable	Description	Probabilistic Distribution	Mean μ	CoV
a	log(a) is the intercept of log(N)-axis by the linear S-N curve.	LogNormal	27.09(1)	0.25[1)]
m	the negative inverse slope of the linear S-N curve.	Constant	3.0	-
A	The shape parameter of the 2-parameter Weibull distribution of the long-term stress ranges	Constant	See **¡Error! No se encuentra el origen de la referencia.**	-
B	The scale parameter of the 2-parameter Weibull distribution of the long-term stress ranges	LogNormal	See Table 5.13	0.5
N	The number of cycles of stress ranges	Constant	1e8	-

Note: 1) $\mu_{ln(a)} = 27.09$ $\sigma_{ln(a)} = 0.25\mu_{ln(a)}$

5.4.2 RESULTS

5.4.2.1 Ultimate Limit State

The probability that the most critical mooring line can survive the ultimate/ extreme load during the design lifetime, and the corresponding reliability index are given in Table 5.12.

TABLE 5.12 SUMMARY OF SURVIVABILITY ASSESSMENT - ULTIMATE LIMIT STATE

Survival Probability 1-PoF	PoF	Reliability Index
99.92%	7.89×10^{-4}	3.16

5.4.2.2 Fatigue Limit State

Based upon the procedure, the fitted shape and scale parameters, as well as R^2, are given in Table 5.13. The fitted Weibull distribution is shown for four mooring lines in Figure 5.4~Figure 5.7**¡Error! No se encuentra el origen de la referencia.**. Since there are only 10 points used to fit the distribution parameters, all the values of R^2 are quite low, especially for mooring line 2. The goodness of fit can be improved by increasing the number of bins.

TABLE 5.13 SUMMARY OF FITTED SHAPE AND SCALE PARAMETERS

Mooring Line	Shape Parameter *A* [-]	Scale Parameter *B* [MPa]	R^2
ML1	1.845	0.0917	0.219
ML2	1.499	0.0473	0.077
ML3	1.777	0.0867	0.296
ML4	1.786	0.115	0.284

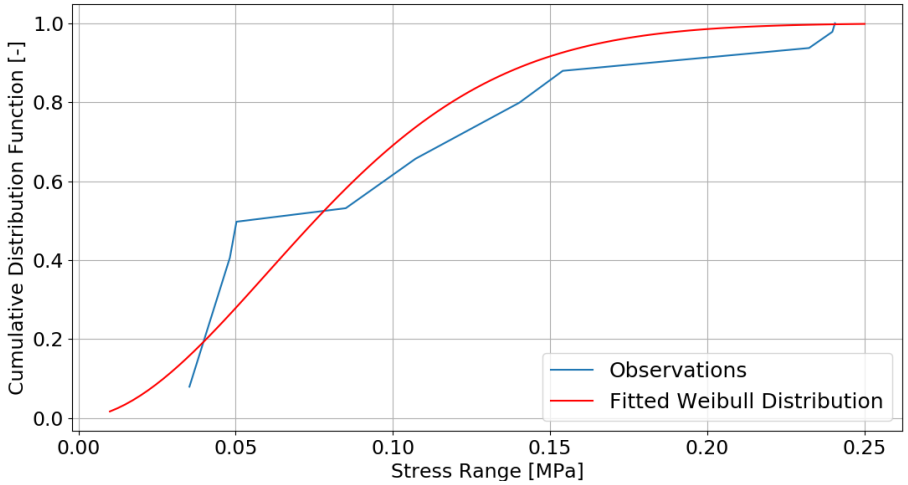


FIGURE 5.4 THE FITTED WEIBULL DISTRIBUTION FOR MOORING LINE 1

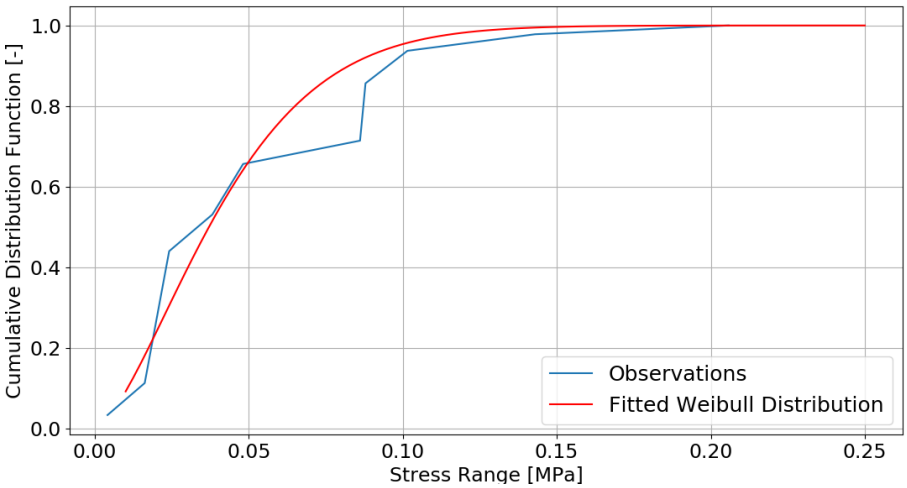


FIGURE 5.5 THE FITTED WEIBULL DISTRIBUTION FOR MOORING LINE 2

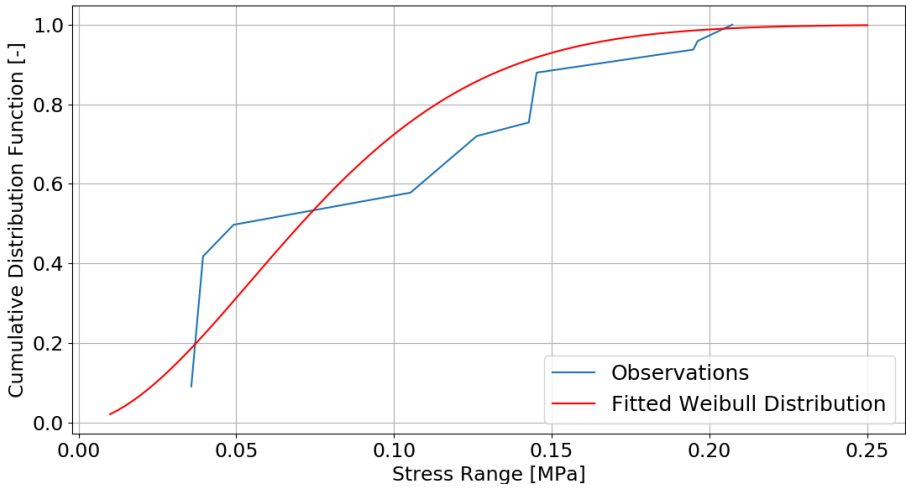


FIGURE 5.6 THE FITTED WEIBULL DISTRIBUTION FOR MOORING LINE 3

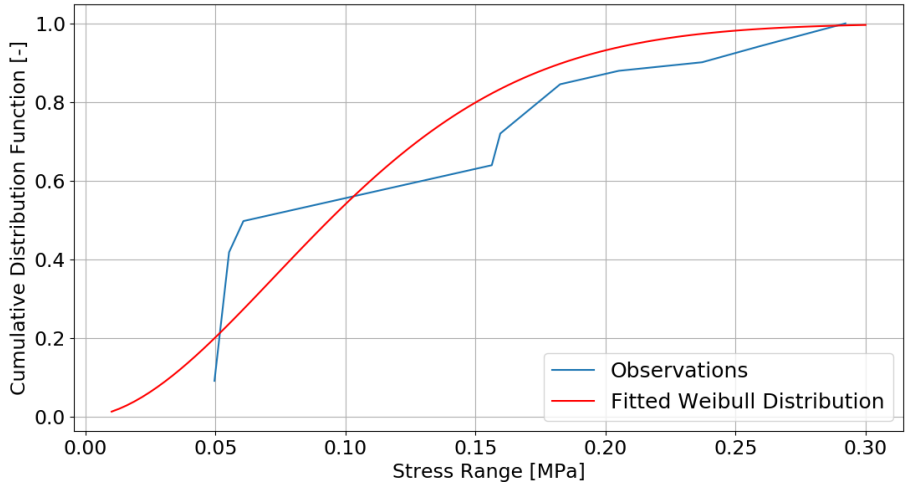


FIGURE 5.7 THE FITTED WEIBULL DISTRIBUTION FOR MOORING LINE 4

The structural reliability analyses have been performed for the four mooring lines, considering the four different cases of the shape and scale parameters. The most critical mooring line is ML4. The survival probability for ML4, and the corresponding reliability index are given in Table 5.14.

TABLE 5.14 SUMMARY OF SURVIVABILITY ASSESSMENT - FATIGUE LIMIT STATE

Survival Probability 1-PoF	PoF	Reliability Index
99.93%	7.44×10^{-4}	3.178

6. FUTURE WORK

The deliverable presents the framework and functionalities of the Reliability, Availability, Maintainability and Survivability (RAMS) module, implemented in T6.4 of the DTOceanPlus project. For the time being, the RAMS module can be run in a standalone mode. However, the additional works summarized as follows should be done in order to integrate it to the DTOceanPlus software platform:

- The OpenAPI file should be "linked" to the other module's equivalent files, in order to guarantee a smooth, robust and consistent data flow among the different pieces of the tool.
- The GUI will be developed to be consistent with the other tools and to provide the user with an easy access to the tool and its functionalities.
- The unit tests, including Pytest, Dredd and PACT, will be improved to fix any potential bugs.
- The verification in T6.7 will be started, when all the modules are fully developed.

The remaining work is part of the continuous development/integration methodology described in Deliverable D7.4 "Handbook of software implementation" [13]. These activities will be developed within T6.7 Verification of the code – beta version in order to extend the functionality of the RAMS module from standalone to fully integrated in the DTOceanPlus toolset.

7. REFERENCES

[1] European Commission, "Advanced Design Tools for Ocean Energy Systems Innovation, Development and Deployment | Projects | H2020 | CORDIS," 2018 January 17. [Online]. Available: https://cordis.europa.eu/project/rcn/214811_en.html. [Accessed 2018 August 6].

[2] ISO, "ISO 2394:2015 General Principles on Reliability for Structures," ISO, 2015.

[3] C. Ebeling, An Introduction to Reliability and Maintainability Engineering, 2nd ed., New York: McGraw-Hill, 1996.

[4] DNVGL, "Definitions of Availability Terms for the Wind Industry," DNV GL, Høvik, Oslo, 2017.

[5] M. Tortorella, Reliability, Maintainability and Supportability: Best Practice for System Engineers, Hoboken, New Jersey: Wiley, 2015.

[6] The European Marine Energy Centre, "Guidelines for Reliability, Maintainability and Survivability of Marine Energy Conversion System," The European Marine Energy Centre, 2009.

[7] M. Rausand and A. Hoyland, System Reliability Theory: Models and Statistical Methods, John Wiley& Sons, INc., 2004.

[8] H. Madsen, S. Krenk and N. C. Lind, Methods of Strucrual Safety, 1986: Prentice-Hall.

[9] DNV, "DNV Classification Note 30.6, Structural Reliability Analysis of Marine Structures," Det Norske Veritas, Høvik, Oslo, 1992.

[10] DNVGL, "DNVGL-RP-C210. Probabilistic methods for planning of inspection for fatigue cracks in offshore structures," DNV GL, Høvik, Oslo, 2015.

[11] DNVGL, "DNV-RP-C203. Fatigue Design of Offshore Steel Structures," DNVGL, Høvik, Oslo, 2011.

[12] IEC, "IEC TS 62600-2 ED2. Marine Energy – Wave, Tidal and Other Water Current Converters – Part 2: Design Requirements for Marine Energy Systems," IEC, 2019.

[13] OpenCASCADE, "Deliverable D7.4 Software Handbook," DTOceanPlus, 2019.

[14] "Military Handbook - Reliability Prediction of Electronic Equipment MIL-HDBK-217F," Department of Denfense, 1991.

ANNEX 1: BRIEF INTRODUCTION ON THE THEORETICAL METHODS FOR RELIABILITY ASSESSMENT

A1.1 COMPONENT-LEVEL RELIABILITY

The objective of component-level reliability assessment is to:

- simulate the time series of failure events of basic components under the fundamental assumption that the failures of these basic components are statistically independent.
- estimate the mean time to failure (MTTF) of these basic components.

The procedure for simulating failure events is summarized as follows:

- obtain the input data, e.g. the design life, the time unit of design life, failure rates of the basic components to be analysed, the probabilistic distribution of time-to-failure (TTF) (by default, it is assumed to follow exponential distribution).
- simulate failure events based upon the algorithm described below.

The Monte Carlo simulation is the basic methodology used to randomly generate discrete failure events. The probabilistic lifetime distribution of all basic components or subassemblies is assumed to follow the exponential distribution. The time to failure has the following probability distribution function.

$$\boldsymbol{F(t) = P(T \leq t) = 1 - e^{-\lambda t}} \quad \text{(A1-1)}$$

The time to failure corresponding to a probability $\hat{F}$ is re-expressed by Eq. (A1-2).

$$\boldsymbol{t = -\frac{1}{\lambda} ln\left[1 - \hat{F}(t)\right]} \quad \text{(A1-2)}$$

where, λ is the failure rate of a basic component. The capitalised *T* denotes the stochastic variable of the time to failure. The variable *t* represents the stated time.

The lifetime simulation of failure events for any component can be presented as follows.

- Step 1: to initiate the input parameters, e.g. the start-up time (to), the lifetime (tL) and the failure rate
- Step 2: to uniformly sample a number from [0, 1] based upon Eq. (A1-1)
- Step 3: to obtain the i-th (i=1,2,3…) failure time t_{fi} based upon Eq. (A1-2)
- Step 4: to check whether or not the termination criterion is satisfied (t_{fi} is greater than t_L); if yes, terminate this lifetime simulation; otherwise, shift to to t_{fi} and go to step 2 to continue sampling.

Based upon the fundamental assumptions in Section 2.2, the TTF follows the exponential distribution. So, the theoretical MTTF of each basic component is just equal to the inverse of its failure rate, as expressed in Eq. (A1-3).

$$MTTF_k = \frac{1}{\lambda_k} \quad \text{(A1-3)}$$

where λ_k denotes the failure rate of the k-th basic component in 1/hour.

Based upon the simulated TTF for each basic component, MTTF can be taken as the average of the TTFs, as expressed in Eq. (A1-4).

$$MTTF = \frac{1}{n}\sum_{i=1}^{n} TTF_i \quad \text{(A1-4)}$$

A1.2 SYSTEM-LEVEL RELIABILITY

Basically, the classic fault tree (FT) is used to calculate the probability of failure (PoF) of the units at higher level in the hierarchy. The detailed review of FT theory is out of the scope of this user manual. Therefore, the application of FT will be elaborated on a case study for the ED subsystem in a marine energy conversion system (array). It should be noted that this case study may be not exactly the same as the examples presented in Section Example, however, the purpose is to demonstrate the procedure to help the readers understand the system-level reliability.

A1.2.1 BACKGROUND INFORMATION OF ENERGY TRANSFER NETWORKS

An energy transfer network, a subsystem in marine energy converter farm, serves to transport the generated power to the onshore terminal. Typical topologies of such an energy transfer network are shown in Figure A1.1. The hierarchy of the system will be detailed in Section 4.2. The white bubbles with dashed boundary lines represent the marine energy converters (MECs) connected to the main cables represented by black bold lines, through the orange circles marked with numbers from 1 to 9 represent the connectors. Due to no tailor-made reliability database for marine energy converters, the failure rates of the basic components can be referred to some generic database for electrical components in other industrial applications [14] and also should be subject to engineering judgement. The failure rates to be used are given in Table A1.1.

TABLE A1.1 INTERPRETATIONS OF NODES IN FT AND BN

Item	Labels in FT	Value [1/hour]
AC1	X1	2.54×10^{-6}
AC2	X2	2.54×10^{-6}
AC3	X3	2.54×10^{-6}
AC4	X4	2.54×10^{-6}
AC5	X5	2.54×10^{-6}
AC6	X6	2.54×10^{-6}
Connector 1	X11	6.24×10^{-7}
Connector 2	X12	6.24×10^{-7}
Connector 3	X13	6.24×10^{-7}
Connector 4	X14	6.24×10^{-7}
Connector 5	X15	6.24×10^{-7}
Connector 6	X16	6.24×10^{-7}
Connector 7	X17	6.24×10^{-7}
Connector 8	X18	6.24×10^{-7}
Connector 9	X19	6.24×10^{-7}
CP1	X20	9.83×10^{-7}

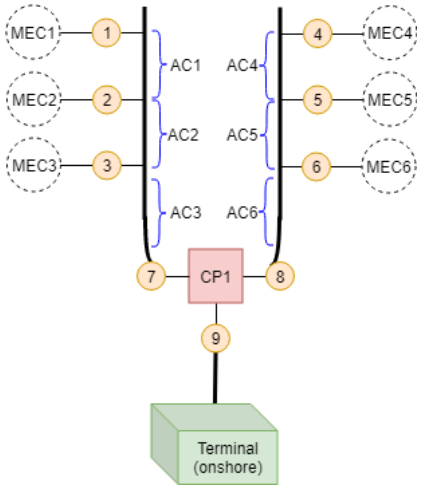


FIGURE A1.1 SKETCH OF ENERGY TRANSFER NETWORK TOPOLOGY

A1.2.2 QUALITATIVE SYSTEM ANALYSIS

The working philosophy of each subsystem and the way these subsystems are aggregated to form the marine energy conversion system can be investigated through qualitative system analysis. The aim of a qualitative system analysis is to obtain the hierarchy of a unit, which could refer to either of system, subsystem, sub-assembly, and to understanding the logic interrelationship between these units. With the knowledge of the hierarchical structures, the traditional FT method will be used to assess the PoFs or reliability of the units at different levels.

Several rounds of brain-storm workshops have been held to understand the working philosophy of the energy transfer system. Based upon the expert's review comments, the hierarchy of the energy transfer system has been clearly defined.

There are two independent energy transfer routes respectively connected to CP1 through the connectors 7 & 8 and two cables AC3 & AC6. The electricity is finally transmitted to the onshore terminal through the connector 9. Suppose that the two energy transfer routes are considered as a virtual unit denoted by T1. T1 is considered as a 1st-level sub-assembly. If either of CP1, X19 and T1 fails, the energy transfer system (T0) will be shut down (no electricity generated).

T1 is composed of two identical energy transfer routes respectively denoted by T2 and T3, which are the 2nd-level sub-assemblies. T1 fails, if both T2 and T3 fail.

T2 comprises X17, X3, a virtual unit T4, which comprises the other connectors directly connected to MECs and the other cables connecting these connectors. If either X17, X3 or T4 fails, this energy transfer route will be shut down. T2 can be considered a series system. T3 comprises X18, X6, a virtual unit T5, which comprises the other connectors directly connected to MECs and the other cables connecting these connectors. If either X18, X6 or T5 fails, this energy transfer route will be shut down. T3 can be considered a series system. Both T4 and T5 constitutes the 3rd-level sub-assembly.

T4 comprises X2, X13 and another virtual unit T6. If either two of X2, X13 and T6 fail, T4 fails. In a similar way, T5 comprises X6, X16 and another virtual unit T7. If either two of X6, X16 and T7 fail, T5 fails. Both T6 and T7 constitutes the 4th-level sub-assembly.

T6 comprises X12 and another virtual unit T8. If either X12 or T8 fails, T6 fails. T7 comprises X15 and another virtual unit T9. If either X15 or T9 fails, T7 fails. Both T8 and T9 constitutes the 5th-level sub-assembly.

T8 comprises X11 and X1. If either X11 or X1 fails, T8 fails. T9 comprises X14 and X4. If either X14 or X4 fails, T9 fails.

Based upon the qualitative system analysis, the hierarchy of the ET subsystem has been clearly defined, with its compact format given in Table A1.2. The data in Table A1.2 are all drawn from the hierarchy presented in Table A1-2, however, are renamed (some fields in Table A1.2 are changed to the technical terminologies in FT) and re-structured.

The first column gives the subsystem to be analysed. All failure events are considered nodes in the hierarchy. The second column, 'Name of Node', gives the names of these failure events. The third column, 'Design Id', gives the identification labels of the basic components and other units. Design Ids are named according to the rules/ conventions of the ED module. The column, 'Type', defines the levels of a fault tree. The column, 'Category', defines which levels the nodes in the 'Name of Node' column belong to in the fault tree. The columns 'Affiliated to' and 'Affiliated by' define the dependencies of units at various levels. Each entry in 'Affiliated to' defines the label of the higher-level unit which the current unit in the column 'Name of Node' belongs to. Each entry in 'Affiliated by' defines the labels of lower-level units which belong to the current unit. Based upon the aforementioned descriptions, the units in the column 'Affiliated by' are connected through a specific logic gate to the higher-level unit. The logic gates are given in the column 'Gate Type'. The logic gate in each entry of this column is used to connect the unit in the column 'Name of Node' and the units in the column 'Affiliated by'. The last two columns give the failure rates of basic components for both Type 1& 2 failure modes. For now, these failure rates are dummy data, which do not influence the demonstration of the system reliability assessment methodology.

Based upon the qualitative system analysis, the fault tree can be constructed accordingly. The top event is denoted 'Failure of ET system (To)'. The intermediate failure events underneath the top event refer to 'Failure of X19 (denoted X19 in the fault tree)', 'Failure of X20 (denoted X20 in the fault tree)' and 'Failure of T1'. An 'OR' gate is inserted according to the working philosophy. For simplicity, the label name is hereafter used to represent the failure event of this unit in the fault tree.

If 'Failure of T1' is considered the top failure event, the intermediate failure events refer to the two 'Failure of T2' or 'Failure of T3' events. An 'AND' gate is inserted according to the working philosophy.

If 'Failure of T2' is considered the top failure event, the intermediate failure events refer to 'Failure of X17 (denoted X17 in the fault tree)', 'Failure of X3 (denoted X3 in the fault tree)' and 'Failure of T4'. An 'OR' gate is inserted according to the working philosophy. If 'Failure of T3' is considered the top failure event, the intermediate failure events refer to 'Failure of X18 (denoted X18 in the fault tree)', 'Failure

of X6 (denoted X6 in the fault tree)' and 'Failure of T5'. An 'OR' gate is inserted according to the working philosophy.

If 'Failure of T4' is considered the top failure event, the intermediate failure events refer to 'Failure of X13 (denoted X13 in the fault tree)', 'Failure of X2 (denoted X2 in the fault tree)' and 'Failure of T6)' A '2/3' gate is inserted according to the working philosophy. If 'Failure of T5' is considered the top failure event, the intermediate failure events refer to 'Failure of X16 (denoted X16 in the fault tree)', 'Failure of X5 (denoted X5 in the fault tree)' and 'Failure of T7'. A '2/3' gate is inserted according to the working philosophy.

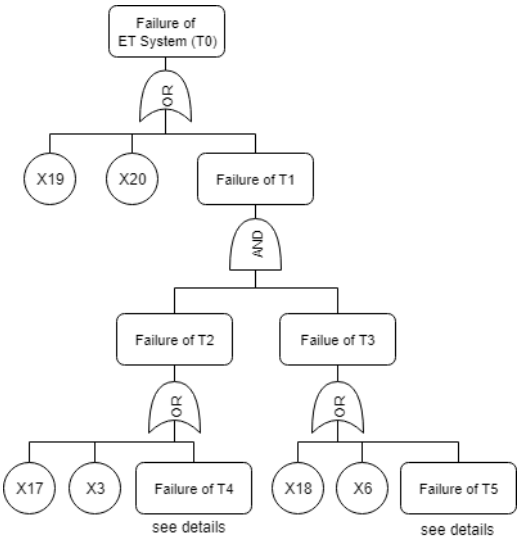
If 'Failure of T6' is considered the top failure event, the intermediate failure events refer to 'Failure of X12 (denoted X12 in the fault tree)' and 'Failure of T8'. An 'OR' gate is inserted according to the working philosophy. If 'Failure of T7' is considered the top failure event, the intermediate failure events refer to 'Failure of X15 (denoted X15 in the fault tree)' and 'Failure of T9'. An 'OR' gate is inserted according to the working philosophy.

If 'Failure of T8' is considered the top failure event, the bottom failure events refer to 'Failure of X11 (denoted X11 in the fault tree)' and 'Failure of X1 (denoted X1 in the fault tree)'. An 'OR' gate is inserted according to the working philosophy. If 'Failure of T9' is considered the top failure event, the bottom failure events refer to 'Failure of X14 (denoted X14 in the fault tree)' and 'Failure of X4 (denoted X4 in the fault tree)'. An 'OR' gate is inserted according to the working philosophy. The fault tree of the ET subsystem is shown in Figure A1.2.

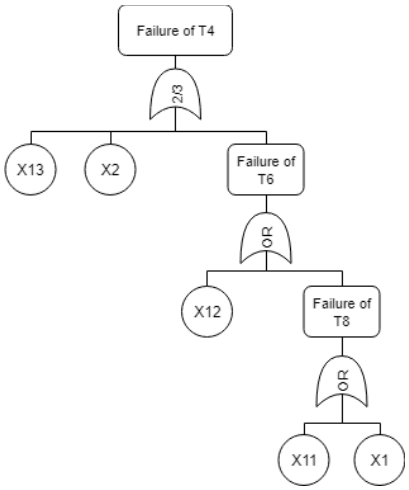
TABLE A1.2 HIERARCHY OF ENERGY TRANSFER SYSTEM

Unit	Type	Category	Affiliated to (parent)	Affiliated by (child)	Gate Type
T0	System	Level 6	N.A.	[X19, X20, T1]	OR
T1	1st-level sub-assembly	Level 5	T0	[T2, T3]	AND
T2	2nd-level sub-assembly	Level 4	T1	[X17, X3, T4]	OR
T3	2nd -level sub-assembly	Level 4	T1	[X18, X6, T5]	OR
T4	3rd-level sub-assembly	Level 3	T2	[X13, X2, T6]	2/3
T5	3rd -level sub-assembly	Level 3	T3	[X16, X5, T7]	2/3
T6	4th -level sub-assembly	Level 2	T4	[X12, T8]	OR
T7	4th -level sub-assembly	Level 2	T5	[X15, T9]	OR
T8	5th -level sub-assembly	Level 1	T6	[X11, X1]	OR
T9	5th -level sub-assembly	Level 1	T6	[X14, X4]	OR
X11	Basic Component	Level 0	T8	N.A.	N.A.
X12	Basic Component	Level 0	T6	N.A.	N.A.
X13	Basic Component	Level 0	T4	N.A.	N.A.
X14	Basic Component	Level 0	T9	N.A.	N.A.
X15	Basic Component	Level 0	T7	N.A.	N.A.
X16	Basic Component	Level 0	T5	N.A.	N.A.
X17	Basic Component	Level 0	T2	N.A.	N.A.
X18	Basic Component	Level 0	T3	N.A.	N.A.
X19	Basic Component	Level 0	T0	N.A.	N.A.
X1	Basic Component	Level 0	T8	N.A.	N.A.
X2	Basic Component	Level 0	T4	N.A.	N.A.
X3	Basic Component	Level 0	T2	N.A.	N.A.
X4	Basic Component	Level 0	T9	N.A.	N.A.

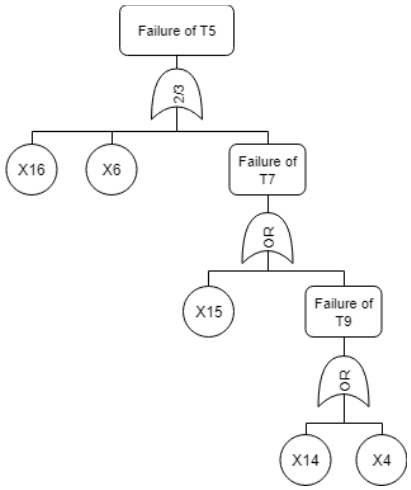
Unit	Type	Category	Affiliated to (parent)	Affiliated by (child)	Gate Type
X5	Basic Component	Level 0	T5	N.A.	N.A.
X6	Basic Component	Level 0	T2	N.A.	N.A.
X20	Basic Component	Level 0	T0	N.A.	N.A.



(a) Main Fault Tree



(b) Detailed Branches underneath T4



(c) Detailed Branches underneath T5

FIGURE A1.2 FAULT TREE OF ENERGY TRANSFER SYSTEM

ANNEX 2: BRIEF INTRODUCTION ON THE THEORETICAL METHODS FOR SURVIVABILITY ASSESSMENT

A2.1 OVERVIEW

This annex briefly presents the methodology of the structural reliability analysis, which serves the theoretical basis for the survivability assessment. The structural reliability analysis will be performed from the perspectives of both ultimate and fatigue limits states.

A2.2 LIMIT STATE FUNCTIONS

A2.2.1 ULTIMATE LIMIT STATE

The ultimate failure modes can be further divided into two categories. Category 1 is the strength failure. Category 2 is the buckling failure (stability). Compared with the progressive failure mode (e.g. fatigue failure), the limit state function for the ultimate limit state is easier formulated. This limit state function is given by Eq. (A2-1).

$$g(X_1, X_2, \dots, X_n) = X_R R(X_1, X_2, \dots) - X_S S(\dots, X_n) \quad \text{(A2-1)}$$

where *R* is a stochastic variable and represents the resistance of the material of a component. *S* is a stochastic variable and represents the stress/ load exerted upon the component. X_R and X_S model uncertainty related to resistance and load models, respectively. $X_1, X_2, \dots, X_n$ refer to the other parameters related to the resistance and stress/load models. The probability $P(g(X_1, X_2, \dots, X_n) > 0)$ is the probability of a component surviving the stresses/ loads during the design lifetime. See [8], [10], [11] for the details regarding how to calculate the probability.

A2.2.2 FATIGUE LIMIT STATE

There are a few ways to define the fatigue limit state functions. In this annex, the S-N-curve-based limit state function will be presented. Unless otherwise specified, the S-N curves for welded steel structures or non-welded structures are only considered.

Linear S-N curve

The limit state function is given by Eq. (A2-2).

$$g(A, B, N, a, m) = -\ln(N) + \ln(a) - m\ln(B) - \ln\left(\Gamma\left(1 + \frac{m}{A}\right)\right) \quad \text{(A2-2)}$$

where A and B are the shape and scale parameters of the 2-parameter Weibull distribution of the long-term stress ranges. $\log(a)$ denotes the intercept of log(N)-axis by the linear S-N curve. m denotes the negative inverse slope of the linear S-N curve. *N* denotes the total number of stress range cycles during the design lifetime. See [8], [10], [11] for the details regarding how to calculate the probability.

Bi-linear S-N curve

The limit state function is given by Eq. (A2-3).

$$g(\bar{X}) = \Delta - \frac{T\upsilon_0}{a_1}\left\{A^{m_1}G_1\left[1+\frac{m_1}{B},\left(\frac{S_{sw}}{A}\right)^B\right] + \frac{a_1}{a_2}A^{m_2}G_2\left[1+\frac{m_2}{B},\left(\frac{S_{sw}}{A}\right)^B\right]\right\} \quad \text{(A2-3)}$$

where $G_1[\cdot,\cdot]$ denotes the complementary incomplete Gamma function, and $G_2[\cdot,\cdot]$ denotes the incomplete Gamma function. $\bar{X} = (A, B, S_{sw}, T, a_1, a_2, m_1, m_2, \Delta, \upsilon_0)$. The stochastic variable, Δ, denotes the fatigue capacity and reflects the resistance of weld material. υ_0 is the mean zero up-crossing frequency. T is the design lifetime (the unit should be compatible with υ_0). $\upsilon_0 T$ represents the total number of stress range cycles during the design lifetime. $\log(a_1)$ and $\log(a_2)$ denote the intercepts of log(N)-axis by the bilinear S-N curve, respectively. m_1 and m_2 denote the negative inverse slopes of the bilinear S-N curve, respectively. A and B are the shape and scale parameters of the 2-parameter Weibull distribution of the long-term stress ranges. S_{sw} denotes the stress range at the intersection of the bi-linear S-N curve. See [8], [10], [11] for the details regarding how to calculate the probability.

CONTACT DETAILS

Mr. Pablo Ruiz-Minguela

Project Coordinator, TECNALIA

www.dtoceanplus.eu

Naval Energies terminated its participation on 31st August 2018 and
EDF terminated its participation on 31st January 2019.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 785921