



Norwegian University
of Life Sciences

Master's Thesis 2024 30 ECTS
Faculty of Science and Technology

Optimizing Subsea Operations Through Digital Twin Based Monitoring of Flexible Products

Tuva Carey Selsaas
Mechanical, Process, and Product Development

Optimizing Subsea Operations Through Digital Twin Based Monitoring of Flexible Products

May 15, 2024

Abstract

Monitoring of flexible products, such as flexible risers (FRs) and inter-array cables (IACs), is crucial to ensure the structural integrity of the product to be sure it is not damaged during the installation operation. Particularly important is the monitoring of the bend radius of the product, which should never exceed the minimum bend radius (MBR).

Traditional methods for monitoring the flexible products involve remotely operated vehicles (ROVs) for spot checks and continuous monitoring of the touch down point (TDP). Calculations are carried out through OrcaFlex to set requirements and limitations for the weather condition for the operation. In this way, they have some control over the operation. However, the ROV has a very limited view, and the images can often be poor due to murky waters. Today's solution for monitoring has potential for improvement. With real-time measurements of the product's condition, the operation will be considerably made more efficient and optimized considerably.

In collaboration with DeepOcean, this thesis develops a digital twin (DT) framework designed to enhance the monitoring of flexible products. Utilizing OrcaFlex for dynamic analysis and Unreal Engine 5 (UE5) for advanced visualization, the system provides real-time data collection and processing that significantly improve operational accuracy and awareness. The proposed DT framework also includes an integrated alarm system to immediately address critical situations, such as bending radius beyond the MBR.

This research demonstrates the potential of DT technology in subsea operations, ensuring higher safety standards and operational efficiency, and suggests future pathways for enhancing digital monitoring systems. The study focuses on step 2 (Concept Development) of the Integrated Product Development (IPD) process. In this phase, the prototype serves as a proof of concept, demonstrating the data flow and viability of the proposed solution. The DT prototype is divided into four key modules – real-world data collection, analysis program, live view, and storage system – each of which are responsible for their own tasks within the DT system.

Acknowledgments

This thesis represents the end of my master's degree in mechanical engineering at the Norwegian University of Life Science (NMBU), within the Faculty of Science and Technology. I would like to use the opportunity to say a big thank you to my fellow students and teachers over the years at Campus Ås, making these years truly unforgettable.

The thesis was conducted in collaboration with DeepOcean, a leader in subsea services. I extend my deepest gratitude to the entire Visuals Department at DeepOcean, whose team has been instrumental in my research and thesis work. Their generosity in sharing valuable knowledge and insight has greatly enhanced both my project and my personal expertise. I am profoundly thankful for their comprehensive support, technical and emotional, that helped me navigate the challenges of this project while making me feel like a valued part of the team. A special thank you to Ørjan Gloppen, Geir Magne Kvinnesland, Harry R. Berg, Ole Gunnar Johnsen, and Maria Leirvik for guiding and assisting me whenever I needed it.

Additionally, I would like to extend a thanks to the 9 a.m. coffee meetings which have been a source of daily joy and camaraderie. These gatherings provided not only a delightful start to each day but also fostered a supportive community that contributed significantly to my thesis journey.

Haugesund, May 2024
Tuva Carey Selsaas

List of Figures

Figure 1: Project Plan, starting January 8 th and ending May 15 th .	3
Figure 2: Edda Freya, one of DeepOcean's larger vessels [4].	4
Figure 3: The six main configurations – Free Hanging Catenary, Lazy Wave, Lazy-S, Steep Wave, Steep-S, and Pilant Wave – figure inspired by [9].	5
Figure 4: Buoyancy clamp on flexible cable, the figure is accessed from CRP Subsea [10].	6
Figure 5: Midwater arch by Cortland [13].	6
Figure 6: Typical subsea configuration of a floating WTG, consisting of floater, bend stiffener, ballast, BMs, tethers and anchors, touch down protection, bend restrictor, and connector, figure from BVG Associates [16].	7
Figure 7: Critical parameters during cable lay operation with catenary cable.	8
Figure 8: OrcaFlex line model from OrcaFlex documentation [46].	12
Figure 9: Catenary of the flexible product. Point 1 and 2 are the start- and endpoint of the catenary of the blue line, point 2 and 3 are start- and endpoint for the catenary of the light blue line, etc. 2, 3, and 4 represents BMs, while 5 represent a clump weight.	13
Figure 10: Blueprint example. This code runs a function that clears the spline points, loops through an array (where the newest data is stored) to add new spline points from the X, Y, and Z element in the array. This example converts the values of the elements from m to cm by multiplying with 100.	15
Figure 11: Example of prompt and answer using ChatGPT.	22
Figure 12: IPD plan, where step 0 and 1 was performed in advance of the project, while step 3 to 5 remains. Step 2 is the phase of the product development during this project.	25
Figure 13: House of Quality presenting the customer requirements vertically and the technical requirements horizontal.	26
Figure 14: Requirement prioritisation chart based on HOQ.	26
Figure 15: Function analysis system technique diagram where features are made into functions.	27
Figure 16: Pugh's model comparing analysis software, showing that OrcaFlex scored the highest.	28
Figure 17: Pugh's model for comparing 3D-modeling software, where Unreal Engine 5 scores the highest.	28
Figure 18: Available data during operation today. A and B represents end A of the product (where it leaves the vessel) and end B of the product (the end that leaves the vessel first) respectively, where the x-, y-, and z-coordinates are known. The length from B to each BM is known.	31
Figure 19: Pugh's method used to compare data collection concepts.	32
Figure 20: Visual schematic of the digital twin system.	35
Figure 21: Product schematic of the DT system. Light blue boxes represent the modules, which are divided into different elements sorted in coloured boxes.	35
Figure 22: System architecture presenting the data (presented in Table 7) flow throughout the DT system.	37
Figure 23: OrcaFlex model of the test scenario. Red object is the vessel, yellow object is the line representing the flexible product, and the pink objects attached to the line is the attachments representing the BMs. Blue line represents the water line, and the orange.	42
Figure 24: Object data of the line, where 'end A', 'end B', and 'length' is shown in the yellow boxes.	42
Figure 25: Part of the output data from unit test of unit 2 of the Analysis Program, stored in a JSON structure. X, Y, and Z value was set as float, while MBR (is the Bend Radius exceeding the MBR?) value was set as Boolean.	43
Figure 26: Error messages when static calculation won't execute.	44

Figure 27: Live View in play mode. Red part of the cable illustrates the parts where the MBR is exceeded, while green shows the parts that is not exceeding the MBR. Message box in the top right corner with warning message telling the user to take immediate action..... 45

Figure 28: OrcaFlex model after the static calculation was performed. 46

Figure 29: The prototype showing red for the nodes with Boolean set to "True", and green for the nodes with Boolean set to "False"...... 46

List of Tables

Table 1: List of abbreviations used in this thesis.	vi
Table 2: Level of Digital Twin based on Jones [30].	10
Table 3: Types of interview [81]	19
Table 4: Mission Statement of the Live View Digital Twin for Subsea Operations.	24
Table 5: Customer needs and responding customer requirements.	25
Table 6: Analysis Program components with input, output, and description.....	33
Table 7: <i>Data categorisation</i>	36
Table 8: Properties of each object in OrcaFlex for unit testing of the Analysis Program.....	40
Table 9: Simulated input data for unit test of unit 1 of the Analysis Program, set as key and value in JSON file, for testing control question 1-4.....	41
Table 10: Simulated input data for unit test of unit 1 of the Analysis Program, set as key and value in JSON file, for testing control question 5.....	41
Table 11: Control questions and result from unit test of unit 1 of the Analysis Program.	41
Table 12: Simulated input data for unit test of unit 2 of the Analysis Program, set as key and value in JSON file, for testing control question 1-6.....	42
Table 13: Simulated input data for unit test of unit 2 of the Analysis Program, set as key and value in JSON file, for testing control question 7.....	43
Table 14: Control questions and result from unit test of unit 2 of the Analysis Program.	43
Table 15: Input data for generating JSON file containing input data for the live view. Value 1 and value two generated two distinct scenarios.	44
Table 16: Control questions and results with comment from the unit test of the Live View blueprint code.	44
Table 17: Properties of each object in OrcaFlex for prototype testing.	45
Table 18: Digital twin system update time results.	47
Table 19: Digital twin total storage results.	47
Table 20: Control questions, results, and comments for the prototype testing.	47
Table 21: Overview of interview with the work position of the interviewee, the questions, and answers.	48

Abbreviations

Table 1: List of abbreviations used in this thesis.

<i>Abbreviation</i>	<i>Description</i>
<i>API</i>	Application Programming Interface
<i>BM</i>	Buoyancy Module
<i>CLV</i>	Cable Lay Vessel
<i>DNV</i>	Den Norske Veritas
<i>DOF</i>	Degree of Freedom
<i>DT</i>	Digital Twin
<i>DTSS</i>	Distributed Temperature and Strain Sensing
<i>FEM</i>	Finite Element Model
<i>FR</i>	Flexible Riser
<i>GBP</i>	Great British Pound
<i>HSE</i>	Health, Safety, and Environment
<i>HSEQ</i>	Health, Safety, Environment, and Quality
<i>HOQ</i>	House of Quality
<i>IAC</i>	Inter-Array Cable
<i>IDE</i>	Integrated Development Environment
<i>IPD</i>	Integrated Product Development
<i>IoT</i>	Internet of Things
<i>LCOE</i>	Levelized Cost of Energy
<i>MBR</i>	Minimum Bending Radius
<i>MVP</i>	Minimum Viable Product
<i>MQTT</i>	Message Queuing Telemetry Transport
<i>RAO</i>	Response Amplitude Operators
<i>RF</i>	Radio Frequency
<i>ROV</i>	Remotely Operated Vehicle
<i>SCR</i>	Steel Catenary Riser
<i>SLWR</i>	Steel Lazy Wave Riser
<i>TDP</i>	Touch Down Point
<i>UE5</i>	Unreal Engine 5
<i>VPN</i>	virtual private network
<i>WTG</i>	Wind Turbine Generator

Contents

1	Introduction.....	1
1.1	Problem Statement.....	1
1.2	Objective.....	1
1.2.1	Primary objective.....	2
1.2.2	Research Questions.....	2
1.2.3	Milestones.....	2
1.3	Limitations.....	2
1.4	Project Plan.....	3
1.5	Structure.....	3
2	Theory.....	4
2.1	Cable Laying.....	4
2.1.1	Cable Laying Vessel.....	4
2.1.2	Laying Methods.....	5
2.1.3	Buoyancy Modules.....	6
2.2	Flexible Riser Replacement.....	7
2.2.1	Inter-Array Cable Installation.....	7
2.2.2	Monitoring.....	8
2.2.3	Cable Lay Parameters.....	8
2.3	Digital Twin.....	9
2.3.1	Digital Twin Introduction.....	9
2.3.2	Digital Twin Technology.....	10
2.3.3	Sensor Technology and Underwater Communication.....	10
2.4	Line Theory.....	11
2.4.1	OrcaFlex Line Object.....	12
2.4.2	Catenary Line.....	13
2.4.3	Line in Unreal Engine 5.....	13
2.5	Programming.....	13
2.5.1	OrcFxAPI.....	14
2.5.2	Python.....	14
2.5.3	Blueprinting.....	15
2.6	OrcaFlex Analysis.....	15
2.7	System Architecture.....	16

2.8	Related Work.....	16
3	Methodology.....	18
3.1	Literature Review.....	18
3.2	Qualitative Research.....	18
3.2.1	Interviews.....	18
3.3	Prototype Development.....	19
3.3.1	Concept Development.....	19
3.3.2	Software.....	19
3.3.3	Programming Languages.....	20
3.3.4	Hardware.....	20
3.3.5	Analysis.....	20
3.3.6	Visual Representation.....	20
3.3.7	Alarm System.....	21
3.4	Testing.....	21
3.4.1	Unit Testing.....	21
3.4.2	Prototype Testing.....	21
3.5	Use of Artificial Intelligence.....	22
4	Design and Development.....	23
4.1	Pre-Planning.....	23
4.1.1	Company Vision.....	23
4.1.2	Product Vision.....	23
4.1.3	Product Purpose.....	23
4.1.4	Customer.....	23
4.1.5	Mission Statement.....	24
4.1.6	Integrated Product Development Plan.....	24
4.1.7	Product Requirements and Specifications.....	25
4.2	Concept Design.....	27
4.2.1	Analysis Concept.....	27
4.2.2	3D-Replication Concept.....	28
4.2.3	Alarm System Concept.....	28
4.3	Real-World Data Collecting Concepts.....	29
4.3.1	Temporary Transponders.....	29
4.3.2	Smart Buoyancy Clamps.....	29
4.3.3	Utilizing Current Available Data.....	30
4.3.4	Pugh's Method.....	31
4.3.5	Summary.....	32
4.4	Product Architecture.....	32
4.4.1	Product Schematic.....	34
4.4.2	Data Management.....	35
4.5	System Architecture.....	37

4.6	Prototype.....	37
4.6.1	Real-World Data Simulation.....	38
4.6.2	OrcaFlex Model.....	38
4.6.3	Analysis.....	38
4.6.4	Building the live 3D-Model in Unreal Engine 5.....	38
4.6.5	Alarm System.....	38
4.6.6	Start Menu.....	39
5	Results.....	40
5.1	Unit Test.....	40
5.1.1	Unit Test of Analysis Program Results.....	40
5.1.2	Live View.....	44
5.2	Prototype Testing.....	45
5.3	Feedback.....	48
6	Discussion and Conclusion.....	50
6.1	Summary.....	50
6.1.1	Objective.....	50
6.1.2	Proposed Solution.....	50
6.1.3	What have been done?.....	50
6.1.4	Prototype Development.....	52
6.1.5	Assessment of the Solution.....	52
6.1.6	Use of OrcaFlex.....	52
6.1.7	The Research Journey.....	52
6.1.8	Unit Testing.....	52
6.1.9	Prototype Testing.....	53
6.2	Conclusion.....	54
6.2.1	Key Components and Architecture of a Digital Twin System for Subsea Operations.....	54
6.2.2	Integrity of Digital Twin Technology into Existing Operations.....	55
6.2.3	Advantage of Digital Twin Technology Over traditional Methods.....	55
6.2.4	Challenges and Limitations.....	55
6.2.5	Contribution to Environmental Sustainability.....	55
6.3	Further Work Recommendations.....	56
6.3.1	Further Product Development.....	56
6.3.2	Optimizing the Solution.....	56
6.3.3	Smart Buoyancy.....	56
7	References.....	58
8	Appendix.....	62
8.1	AnalysisProgram.py.....	63
8.2	Spline_BP.....	66
8.3	Unit Test.....	68
8.3.1	Unit Test 2.....	68

8.3.2	Error: Clump Colliding with Vessel	68
8.3.3	Static Results.....	71
8.3.4	Compare Function.....	72
8.4	Unit Test Live View	73
8.5	Prototype Testing.....	74
8.5.1	Digital Twin.....	74
8.5.2	AnalysisResult.json.....	77

1 Introduction

DeepOcean is a subsea services provider founded in 1999 in Haugesund, Norway [1]. They are now located several places in Norway, UK, USA, and Mexico offering services worldwide within oil and gas, offshore renewables, deep sea minerals, and other non-energy niches [2].

DeepOcean specializes in operations involving flexible products such as flexible risers (FR) and inter-array cables (IACs). These operations typically include installation or replacement of the products. The flexible products are frequently utilized between floating offshore platforms and seabed facilities. They offer the advantageous characteristics of being able to bend, making them ideal for various applications. The flexibility makes them particularly well suited to absorb the forces resulting from movement in the floating structures.

The catenary shape of these cables must be meticulously analysed to ensure the integrity of the product, particularly focusing on the bending, which should never exceed the minimum allowable bending radius. The current methodology lacks documentation of bending moments or other critical parameters during the operation. However, pre-operational analyses are typically conducted to verify and establish operational limit conditions.

To enhance the monitoring process, a proposed solution involves the implementation of a DT concept. This approach aims to collect real-time data on the product's position, enabling more precise and comprehensive monitoring to ensure the Minimum Bending Radius (MBR) is not exceeded. The DT framework is designed to offer a more detailed and continuous observation of the operation, surpassing the limitations of current Remotely Operated Vehicle (ROV)-based methods.

Working in the sea presents challenges due to its unpredictable and demanding nature. One significant challenge is the limited visibility of subsea operations. Often, real-time monitoring is minimal, and images provided by the ROVs are degraded by murky water conditions.

Improved visual overview of the subsea operation can be of great advantage and will provide increased efficiency of the operations. The visual part of the DT, proposed in this thesis, is a good contribution to improving this problem.

1.1 Problem Statement

The primary challenge related to installation of flexible products in marine environment is to ensure the structural integrity of the product throughout the entire operation. This involves monitoring the bending moment or strain to prevent damage to the product.

1.2 Objective

Specific objectives have been established to address the challenges identified and presented in this research. These objectives, presented as primary objective and research questions, serves as guiding benchmarks

throughout the study to ensure a focused and systematic approach to answering the problem statement. They offer insights into the potential direction of the research.

1.2.1 Primary objective

The primary objective is to design and develop a method to monitor and control the cable catenary shape, ensuring the integrity of the product through the installation operation.

1.2.2 Research Questions

- What key components and architecture are required in the development of a digital twin framework for real-time monitoring of subsea operations?
- How can digital twin technology be integrated into existing operational framework to optimize monitoring and decision-making processes?
- How does digital twin technology improve subsea operations compared to traditional monitoring methods?
- What are potential challenges and limitations in implementing digital twin technology in offshore operations, and how can they be addressed?
- To what extent can digital twin technology contribute to the environmental sustainability of subsea operations?

1.2.3 Milestones

To guarantee the success of this thesis project, key milestones have been established. These serve as minor objectives and checkpoints, ensuring that the research remains on track and progresses systematically towards the primary objective. By adhering to these milestones, the project aims to maintain focus, meet deadlines effectively, and achieve the desired outcomes with a structured approach.

- Set up a project plan.
- Background study.
- Identify key challenges.
- Develop a concept and prototype for the solution.
- Testing and validation.

1.3 Limitations

To ensure the feasibility of the thesis project, certain limitations were set. These are particularly linked to a limited timeline, which was less than 5 months. The project deals with the development of a digital twin framework as a solution to the problem, which normally requires more than 5 months of work. The thesis primarily focuses on the concept development and delivery of a proof of concept, which is important work in this phase of the product development process.

An important factor for the solution is real-time data, which was not available during the project timeline. This resulted in the use of simulated data, representing real-time data from the offshore field. As no data was collected from offshore operations, Message Queuing Telemetry Transport (MQTT) communication was not of priority during the development of the prototype. MQTT is used in other projects today, this knowledge can therefore be transferred to the product at a later stage.

1.4 Project Plan

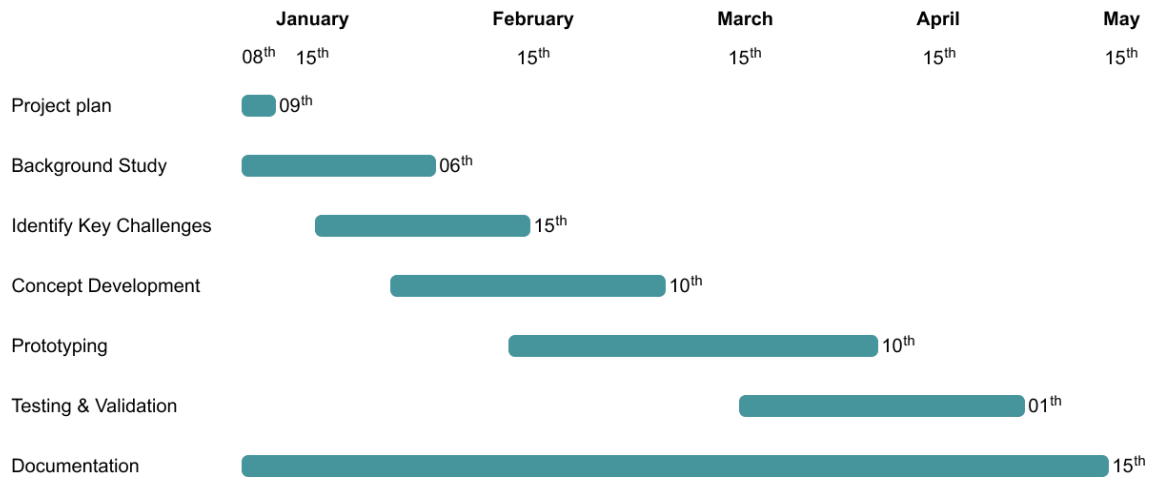


Figure 1: Project Plan, starting January 8th and ending May 15th.

1.5 Structure

The thesis is systematically organized into six chapters to facilitate a coherent flow from the theoretical framework to the practical application and findings.

- Chapter 1 outlines the research background, objectives, and scope.
- Chapter 2 reviews the theoretical foundations and literature.
- Chapter 3 describes the research methodology employed.
- Chapter 4 details the design and development of the digital twin prototype.
- Chapter 5 presents the results.
- Chapter 6 concludes the study and suggests directions for future research.

Each chapter logically segues into the next, methodically advising the thesis from concept to conclusion.

2 Theory

This chapter presents relevant theories to provide a solid understanding of subsea operations related to installation of flexible products, mainly FRs and IACs, DT technology, other relevant elements for the product, and related studies.

2.1 Cable Laying

Cable laying is a typical operation executed by offshore and subsea service providers, such as DeepOcean. This type of operation is typical for riser replacement and installation of IACs for offshore wind farms. During this operation, the potential risk of exceeding critical limit values for tension and bending moments becomes especially significant. This requires meticulous monitoring of the integrity of these critical parameters.

Offshore operations are typically customized to meet specific requirements, as structures are often uniquely designed and custom-built for particular tasks and locations. This customization extends to cable laying as well. This thesis aims to explore the most common methods of cable laying, with a focus on identifying and addressing challenges encountered in these operations.

2.1.1 Cable Laying Vessel

Cable laying operations require specialized vessel, known as CLVs (cable lay vessels), equipped with necessary tools [3]. These vessels exist in various sizes and is equipped with different tools. Each operation necessitates careful consideration to determine the appropriate CLV required. Typically, they carry large reels of cables. One of DeepOcean's vessels is depicted in Figure 2 below.



Figure 2: Edda Freya, one of DeepOcean's larger vessels [4].

2.1.2 Laying Methods

There are three main methods – S-Lay, J-Lay, and Reel-Lay – for laying operations. Each of these methods has its unique advantages and is chosen based on the specific requirements of the subsea operation, including water depth, pipeline diameter, and proximity to the fabrication site.

S-Lay Method

The S-lay method is characterized by obtaining a “S” curve during the laying operation and leaves the vessel from a stinger [5]. The stinger is located at a ramp, at the end of the vessel [5], equipped with workstations for tasks like welding and tensioning [6]. Typically, the cable is continuously welded and fed into the ocean while the vessel moves at a specific speed in the lay direction [7]. This method is suitable for all pipe diameters and lengths, it is faster than other methods, and works well in shallow to intermediate waters [6]. However, for deepwater installations, significant modifications to the stinger are required [6].

J-Lay Method

In J-Lay, the cable configuration resembles the letter 'J' during installation [6] as it leaves the vessel in a vertical position [5]. This method involves welding cables in a vertical tower, on the vessel, which has better dynamic motion characteristics than the S-Lay vessel but offers limited productivity as it operates at a slower lay rate [6]. This method is only suitable for deep water as it would cause damage to the cable in shallow water due to sharp bend at the seabed [5].

Reel-Lay Method

The reel-lay method is suitable for cables with small diameter up to 16 in [8]. This method involves spooling the cable onto large reels on the vessel, and laying it directly from the reel to the seabed [6], with a configuration similar to the S-lay method [8]. A major advantage is that welding and inspection are performed onshore, often at a spool base, allowing for controlled fabrication before being spooled onto the reel [6]. This method is cost-effective and particularly suitable for installing Steel Catenary Risers (SCRs) and Steel Lazy Wave Risers (SLWRs) close to the spool base [6]. However, the cable can be exposed for plastic deformation when it is spooled [8].

Flexible Cable Configuration

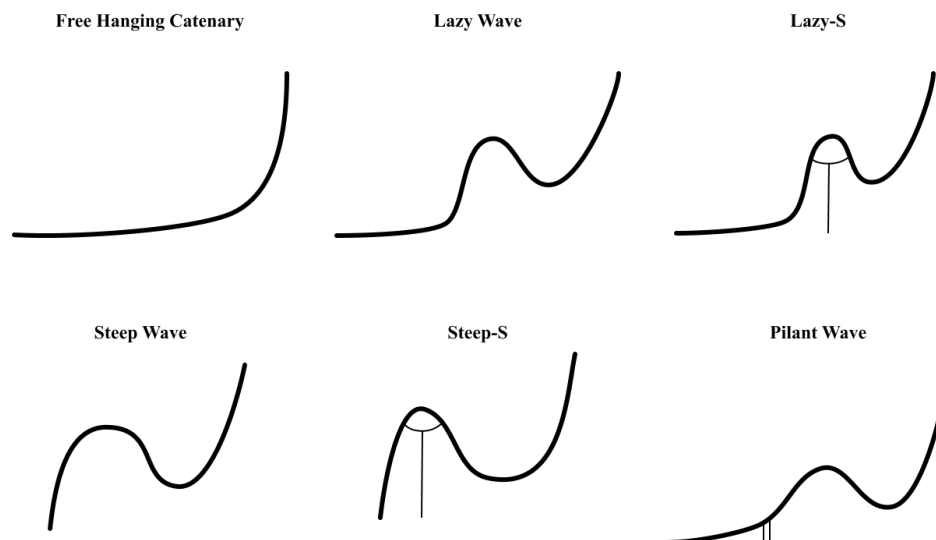


Figure 3: The six main configurations – Free Hanging Catenary, Lazy Wave, Lazy-S, Steep Wave, Steep-S, and Pilant Wave – figure inspired by [9].

2.1.3 Buoyancy Modules

Buoyancy Clamps

Buoyancy clamps are a type of buoyancy module (BM) that are directly attached to the product, typically installed during installation, as depicted in Figure 4. Some cable configurations require BMs to create the specific shape, e.g. the lazy wave or S-shape configuration. Their primary purpose is therefore to secure the position of the BM assembly along the cable [10]. The modules themselves are designed to exert an upward force, preventing the cable from sinking or shifting, thereby ensuring the cable's stability and integrity [11].

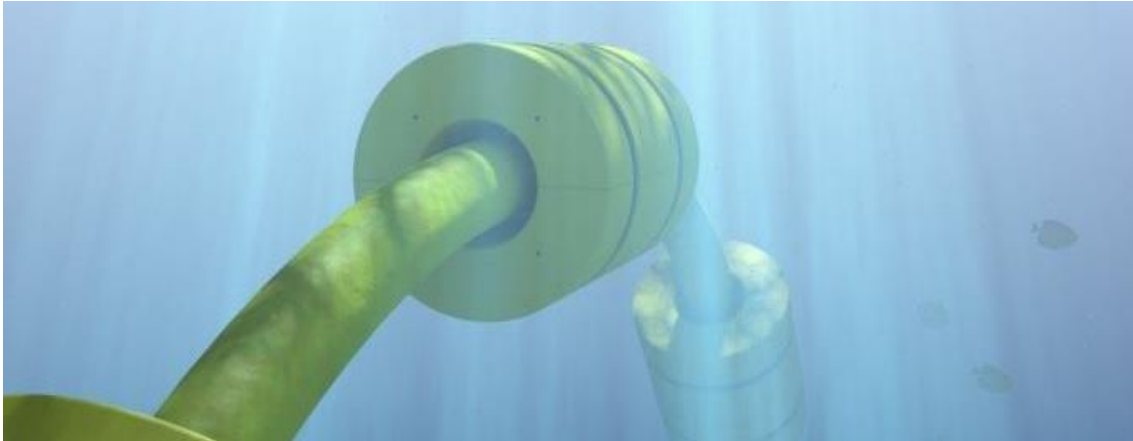


Figure 4: Buoyancy clamp on flexible cable, the figure is accessed from CRP Subsea [10].

According to a marked overview by Raket Research [11] there is a significant growing demand for this type of BM, due to the increasing oil and gas industry and exploration of deep-water regions. In response to increased demand, manufacturers are enhancing their products by designing clamps with improved buoyancy and stability, enhanced corrosion resistance, and easier installation and maintenance [11].

Mid Water Arch

A mid water arch, referred to as MWA, is a buoyancy structure typically used in FR and umbilical systems [12]. As shown in Figure 5, the MWA creates the wave-shape of the product. This structure is tethered to the seabed and is typically placed mid-way through the water column [12], hence the name “mid water arch”.

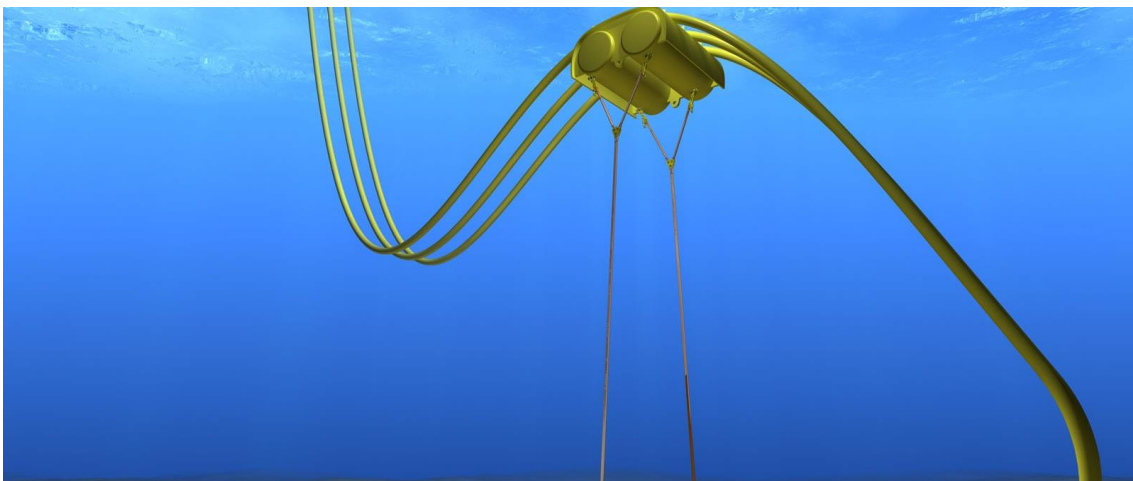


Figure 5: Midwater arch by Cortland [13].

2.2 Flexible Riser Replacement

What is a flexible riser (FR)?

A marine riser is a cylindrical conduit, which functions to transport crude oil from the subsea wellhead to the offshore facility [14]. FRs are usually installed as one of the six configurations illustrated in Figure 3. The FR needs to be designed with flexibility to endure fatigue loads, while simultaneously possessing the strength to withstand axial tension and bending moment.

Offshore risers are categorized into three groups – TTR, Compliant Riser, and Hybrid Riser – in DNV-SE-0476 Offshore Rise System [15]. These groups are further elaborated in [6].

Riser Replacement

FR replacement involves both the recovery of the existing FR and installing a new one. Such operations require meticulously planning and monitoring to prevent potential failure and complications. In certain fields, where multiple FRs are positioned in proximity, there is a risk of entanglement during this operation [3], underscoring the importance of visual monitoring of the assets. Furthermore, these replacement operations carry an increased risk of oil spills [3], making thorough planning and comprehensive documentation of the entire operation essential.

2.2.1 Inter-Array Cable Installation

What is an Inter-Array Cable?

IACs are responsible for transporting the high voltage generated by the wind turbine generators (WTGs) to a substation, from where it is then conveyed to shore through an export cable [16]. The IAC are uniquely designed for each project [17] and are tailored to achieve mechanical properties based on varying environmental conditions. Typical components for an IAC are depicted in Figure 6.

They are designed with flexible properties to withstand loads and stresses due to dynamic motion of the floating WTG. Various configurations for the IACs are currently being explored. A study by Rentschler et al. [18] looked at both free hanging catenary and lazy wave configurations. The study concluded that, in most instances, the lazy wave configuration is preferred over the free hanging catenary shape. However, a downside with this configuration is the cost related to the buoyancy modules [18].

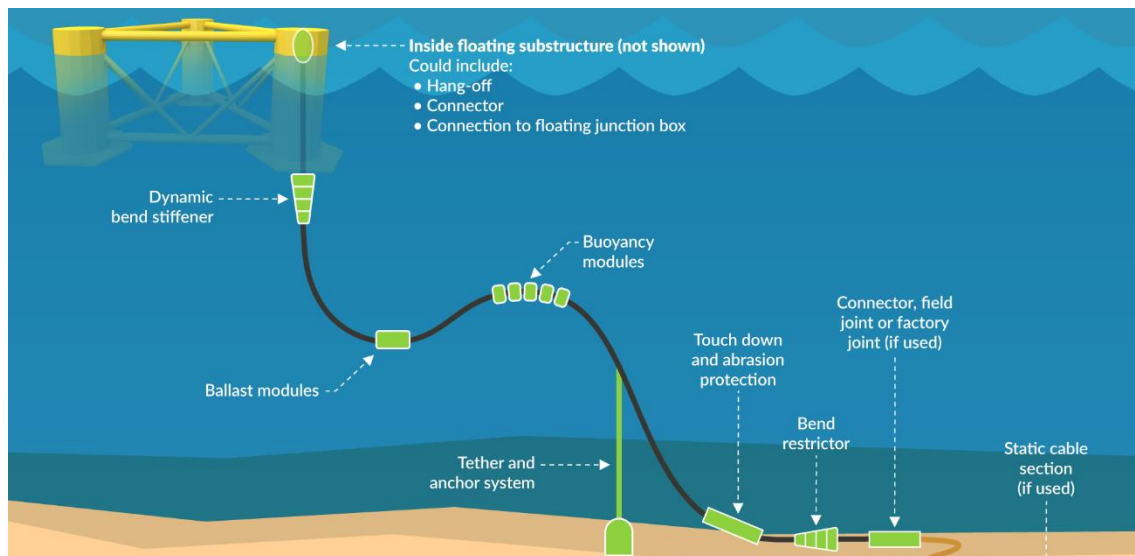


Figure 6: Typical subsea configuration of a floating WTG, consisting of floater, bend stiffener, ballast, BMs, tethers and anchors, touch down protection, bend restrictor, and connector, figure from BVG Associates [16].

Cost Related to Offshore Wind

According to NVE [19], the levelized cost of energy (LCOE) related to floating offshore wind power production is significantly higher compared to other wind power production methods. As a result, floating offshore wind emerges as the most expensive option for wind power generation, necessitating cost reduction across all stages of its life cycle [20]. The need to reduce costs can pose challenges, especially when it comes to developing new technology.

IAC Installation

When installing an offshore wind farm, numerous IACs must be installed as each WTG is connected to the network through an individual IAC. Indicates that this type of assignment is time consuming due to the number of products to be installed, thus time reduction per installation would be a natural requirement to make the operation more efficient.

The report by BVG Associates [16] indicates that the estimated cost of the installation of offshore cables for a 450 MW floating wind farm is £63 million. This expenditure covers the cost of CVL, the cable lay and burial, as well as the pull-in operations and electrical testing and termination. Additionally, it includes the cost related to survey works, route clearance, and the installation of the cable protection systems.

2.2.2 Monitoring

Confirmed by a structured interview (details can be found in Table 21) with a DeepOcean employee [21], working with cable lay operations, the configuration is monitored by performing spot checks with ROVs of hog and sag, and other strategic points along the product, to verify that these points are at the expected positions according to a step table and analysis models, which usually are quite accurate. Prior to the operation there is performed analysis providing information and requirements for certain configurations, this is usually performed using the analysis software OrcaFlex.

When it comes to IACs, they have internal fibreoptic cables [22], which is a cable based communication system [23], allowing for DTSS (distributed temperature and strain sensing) instrumentation technology along the cable. This allows for continuous monitoring of the cable integrity throughout the lifespan of the IAC.

2.2.3 Cable Lay Parameters

Submarine cables are fragile and easily damaged [24], underscoring the importance of careful monitoring of the cable condition. This involves observing the shape of the cable and ensuring that critical parameters (some of them illustrated in Figure 7) never exceed their limits.

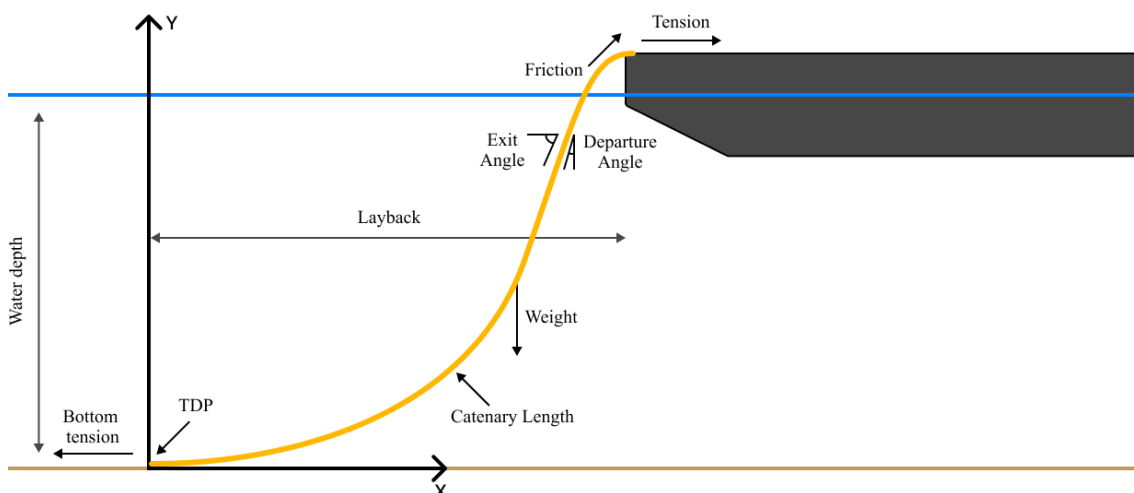


Figure 7: Critical parameters during cable lay operation with catenary cable.

Wave Spectra / Vessel Motion

Described by Bai [8] vessel motion is determined through a global performance analysis that consider wave, wind, and current loads using either time-domain or frequency-domain analysis. The resulting motion data are presented as time trace or response amplitude operators (RAOs), specifically defined at the centre of gravity (CoG) of the floating structure (vessel or platform) under specific loading conditions.

Vessel motion significantly impacts the tension and bending experienced by the cable [24]. A vessel can move in 6 DOFs, these are Pitch, Roll, Yaw, Heave, Sway, and Surge [25]. These movements are largely influenced by ocean waves.

Ocean waves are difficult to describe as they can appear in random lengths and periods. Through an application of certain simplifications, it is possible to approximate the characteristics of the sea surface, this approach is known as the spectrum of ocean waves [26]. It offers a more manageable understanding of the sea surface's dynamic nature by providing a distribution of wave energy across various frequencies and wavelengths [26].

Critical Cable Limits

Flexible products typically adopt a wave-shaped or S-shaped configuration, which make the product bend. Thus, a critical limit during installation is the MBR, which must never be exceeded to prevent damage to the product [24]. Ensuring compliance with the limit require meticulous monitoring during the operation.

The layback length and the catenary length, illustrated in Figure 7, influences the MBR. Layback length refers to the distance between the TDP, where cables rests on the seabed, and the deployment point, where the cable leaves the vessel.

It is essential to withstand the maximum compression and tension limits to maintain the integrity of the cable and prevent permanent damage [17].

A study by Mamatsopoulos et al. [9] provides an analysis of the relationship among various parameters – Bottom Tension, Layback Distance, Catenary Length, Exit Angle, MBR, and Tensioner – which are presented as the most important for cable installations. According to the study, an increase in bottom tension results in a corresponding rise in layback length, catenary length, and MBR, while the exit angle decreases.

Another critical aspect is the bottom tension, which not only contributes to an increased tensional force at the laying equipment, but also significantly influences the catenary shape [27]. Both excessively low and excessively high bottom tension can be problematic. Insufficient tension may result in the formation of loops and induce twisting in the cable [28]. However, high tension can create high residual stresses within the cable, potentially complicating subsequent burial processes [27].

2.3 Digital Twin

2.3.1 Digital Twin Introduction

Digital twins, as a concept, are subject to various interpretations and are employed differently across organizations and companies. For the purposes of this study the definition provided by SINTEF [29] is used to describe DT. This decision is grounded in the stature and recognition of SINTEF as a leading research institution, whose comprehensive understanding and articulation of digital twins are well-respected and align with the objectives of this research.

“A Digital Twin (DT) is defined as a virtual representation of a physical asset enabled through data and simulators for real-time prediction, optimization, monitoring, controlling, and improved decision making.” [29]

DTs can be divided into five different levels: Descriptive Twin (level 1), Informative Twin (level 2), Predictive Twin (level 3), Comprehensive Twin (level 4), and Autonomous Twin (level 5) [1]. These levels are further described in Table 2.

Table 2: Level of Digital Twin based on Jones [30].

<i>Level of sophistication</i>	<i>Description</i>
<i>1: Descriptive Twin</i>	This is the most basic form of a digital twin, acting as a live, editable version of design and construction data. It serves as a visual replica of a physical asset, allowing users to customize the inclusion of information and extraction of specific data.
<i>2: Informative Twin</i>	This level integrates operational and sensory data. It not only captures and aggregates specific data but also ensures the verification of this data for cohesive system functionality.
<i>3: Predictive Twin</i>	At this level, the digital twin utilizes operational data to provide insights and predictions.
<i>4: Comprehensive Twin</i>	This level goes beyond current or past data and starts simulating future scenarios. It is useful for exploring “what-if” questions/ scenarios, allowing for advanced planning and decision-making based on these simulations.
<i>5: Autonomous Twin</i>	This is the most advanced level, which is capable of learning from data and making decisions or taking actions on behalf of users. It represents a significant leap in digital twin technology, enabling more automated and intelligent responses to changing conditions or requirements.

A digital twin consists of the following three components [31]:

- The physical asset
- The digital representation of the asset
- The communication between the physical and digital asset

2.3.2 Digital Twin Technology

Digital Twins have found applications across numerous sectors, yet their usage poses several challenges as presented by Rasheed et al. [32], particularly in the realm of human decision-making. It's crucial that decisions involving humans are not only made with care but are also comprehensible to those affected, this is further elaborated in [33]. This principle is vital in the implementation of Digital Twins. In this specific study, while the DT will not directly make decisions impacting humans, it will facilitate decision-making processes for humans to execute, underscoring the need for clear and understandable decision-making pathways.

2.3.3 Sensor Technology and Underwater Communication

Collecting data from the real-world assets of the digital twin is the core of a DT system. This can be done through IoT technology, connecting assets to a virtual private network (VPN) through the internet [34]. The real-world assets of the DT developed in this thesis are placed underwater.

Underwater Communication Technology

As of today, there are three available technologies – RF, Optical Transmission, and Acoustic Transmission – for wireless communication in subsea environments. Enhancing these technologies is a critical objective, given the unique challenges posed by seawater as a medium. Factors influencing communication is salt concentration, pressure, temperature, and amount of light. Addressing these elements is essential for advancing the reliability and efficiency of subsea communication systems [35].

A noteworthy alternative technology has been proposed in a study by H. Zhao et al. [36], where the application of Maxwell's equations is central. This study has yielded promising results, bolstering the belief that this method could become a promising approach for enhancing wireless communication in subsea environments.

Another significant research effort [37] has verified ongoing advancements in the methods and systems for transmitting broadband optical signals over extended distances. This wireless communication approach is distinguished by its low installation and operational costs, high data transmission rates, and secure communication links [37].

M. Kong et al. [38] introduces *AquaE-net*, the first prototype of an underwater optical wireless sensor network. The paper delves into the challenges inherent in underwater communication. A key issue highlighted, in the paper, is the limited power availability for sensor nodes in underwater networks, compounded by the challenges in charging or replacing batteries in such environments. Consequently, the design of these sensor networks demands a focus on low-power consumption ensuring operational efficiency and longevity [38].

Subsea Transponders

A subsea transponder is used to access the position of an asset. It is defined by Meriam Webster [39] as a transceiver emitting a radio signal when a designated signal is received, which is especially used for detection, identification, and location of objects.

Among the available technologies, long-lasting transponders such as the Fetch Aza by Sonardyne, which boasts a lifespan of up to 10 years [40], exemplify the advancements in persistent data collection devices. These durable solutions are invaluable for long-term monitoring and analysis, offering sustained insights into the product's performance and condition over an extended period.

Conversely, for operations primarily focused on the installation phase, more cost-effective alternatives may be preferred. Kongsberg, a Norwegian company, offers a selection of such transponders, including the cNode series, characterized by a lifespan of approximately 30 days [41]. These devices present a pragmatic option for projects where the primary goal is to ensure accuracy during the laying process, thereby minimizing upfront costs without compromising the quality of data acquisition.

MQTT

Message queuing telemetry transport (MQTT) is a publish/subscribe messaging protocol for the IoT providing communication between a client and broker [42]. The MQTT client can either publish messages or subscribe to receive messages through the MQTT broker, the intermediary between the publishing and the subscribing client [43].

2.4 Line Theory

A practical approach to replicate a cable or cable-like products is through the use of a line representation. A line is defined as a straight one-dimensional geometrical figure that has no thickness and extends infinitely in both directions [44]. It is uniquely determined by two distinct points [44]; it is the shortest path that connects these points, which is also referred to as a straight line.

To accurately model a flexible cable, or similar product, using the concept of a line, the cable's structure is segmented into a multitude of points connected by lines. By minimizing the distance between these points, each segment approaches the form of a straight line, allowing for a closer approximation of the cable's true, flexible shape. This methodology is akin to the principle used in describing a circle, where the curvature is approximated by a series of short, straight segments [45]. As the segments become shorter, the representation becomes increasingly precise, enabling a realistic simulation of the cable's behaviour and configuration under various conditions.

2.4.1 OrcaFlex Line Object

The theory of the line object in OrcaFlex is described in their web documentation [46]. OrcaFlex [46] represents the line through a Finite Element Model (FEM), illustrated in Figure 8. “The line is divided into a series of line segments which are then modelled by straight massless model segments with a node at each end.” [46]. As depicted in Figure 8, the nodes act as straight rods, representing the two adjoining half-segments, except for terminal nodes, which are adjacent to only one half-segment and thus represent a single half-segment [46]. “The following figure gives greater detail of the line model, showing a single mid-line node and the segments either side of it. The figure includes the various spring-dampers that model the structural properties of the line.” OrcaFlex [46]. Each model segment simulates the axial and torsional characteristics of the line [46]. Conceptually, a segment can be envisioned as comprising two co-axial, telescoping rods connected by springs and dampers that account for axial and torsional forces, facilitating a detailed and nuanced analysis of the line's behaviour under various conditions [46].

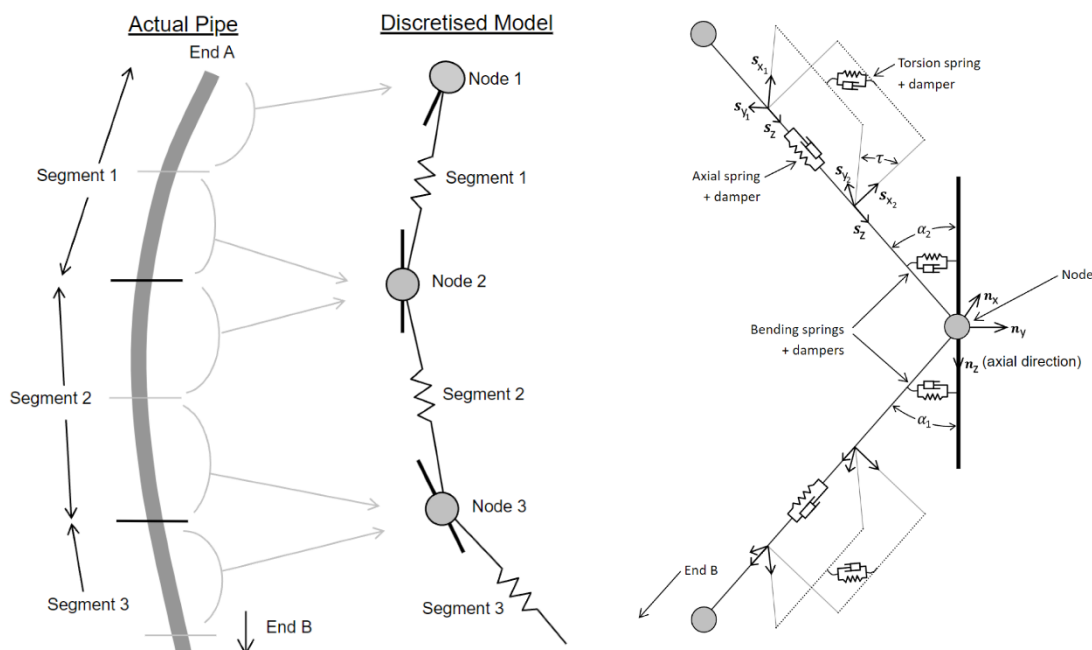


Figure 8: OrcaFlex line model from OrcaFlex documentation [46].

The model utilizes spring-dampers to simulate the line's mechanical properties across three areas: axial, bending, and torsional behaviours, as depicted in Figure 8.

Line Calculation

OrcaFlex divide the calculation of the line into five stages where the following is calculated – step 1: tension forces, step 2: bend moments, step 3: shear forces, step 4: torsion moments, and step 5: total load – which is further described in the OrcaFlex documentation [46].

The line statics calculation in OrcaFlex consists of two steps [47]:

- Step 1: offering six different methods, further described in [47], where catenary is one of them. This method calculates the equilibrium position of the line object, but it does not include the effects of bending and torsional stiffness [48].
- Step 2: offering two methods, which is none or full statics [47]. Full statics calculates an equilibrium position for the model, including bend stiffness, based on the results from step 1.

2.4.2 Catenary Line

A catenary line can be described as a hanging flexible line supported at its ends and acted upon by a uniform gravitational force [49]. The wave-shaped model is built up by several catenaries, as depicted in Figure 9.

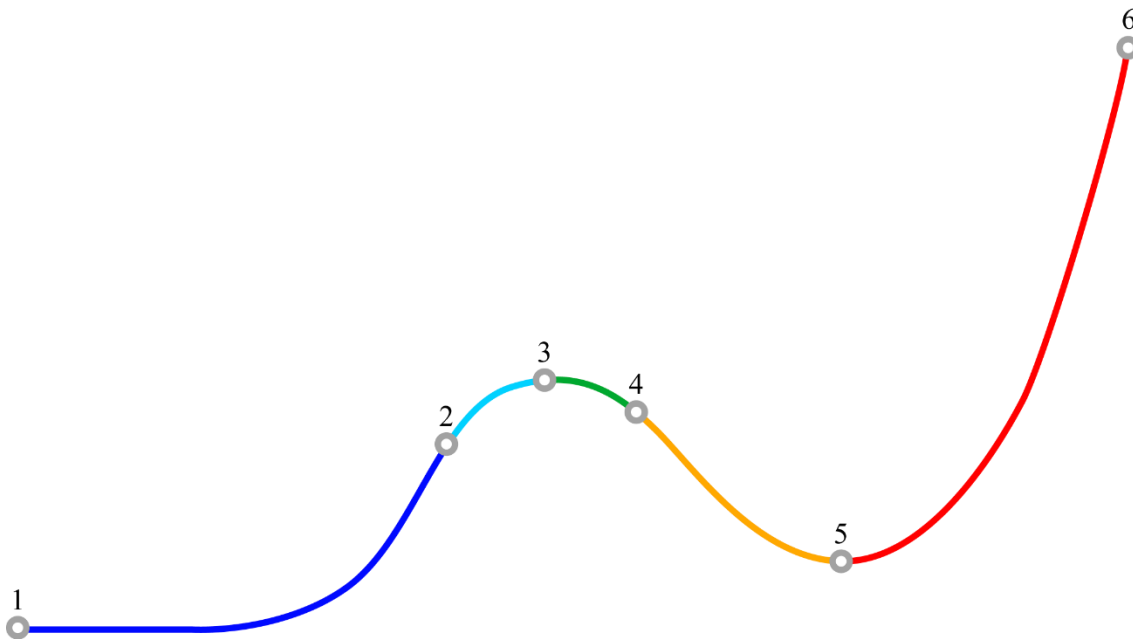


Figure 9: Catenary of the flexible product. Point 1 and 2 are the start- and endpoint of the catenary of the blue line, point 2 and 3 are start- and endpoint for the catenary of the light blue line, etc. 2, 3, and 4 represents BMs, while 5 represent a clump weight.

2.4.3 Line in Unreal Engine 5

The line was also created in the DT which was created in Unreal Engine 5 (UE5). As of today, there are two different spline types available in the spline node, these are BSpline and Hermite. The BSpline is a method where the spline is created as smoothly as possible passing through the first and last control points [50]. The other option is the Hermite Spline, where the curve will pass through all the control points [50].

2.5 Programming

Automating and tailoring the OrcaFlex calculation can be achieved through Python programming by leveraging the software's API. Programming was also utilized to create the live view in UE5 using blueprints.

2.5.1 OrcFxAPI

An application programming interface (API) is a machine-readable interface, which is a collection of programming codes that enables exchange of data between different software applications [51]. OrcaFlex has a Python API, called OrcFxAPI, making it possible to communicate with the software and accessing a range of OrcaFlex's facilities [52]. This allows the user to create Python programs to automate the analyses, which can run in the background of the DT.

2.5.2 Python

Python [53] is a widely used programming language and is considered a high-level language. Known for its readability and user-friendliness, Python serves as a wide range of applications, excelling particularly in automation and data analysis [54].

Just like other languages, Python employs a specific set of syntax and semantics rules to structure communication [55]. This ensures that the instructions provided to the computer have a clear meaning. To establish a solid programming flow and structure, several key concepts are instrumental. Understanding and applying these principles can significantly enhance the readability, maintainability, and efficiency. Here are some of the foundational concepts in programming:

Module

In the context of Python programming, a module is identified as a code consisting of Python definitions and statements, intended for utilization within other scripts [56]. Modules can be described as folders, full of a variety of valuable components such as data, functions, and classes, all organized for optimal reusability within Python codes [57].

Class

Similar to modules, classes serve as a mechanism for grouping data and functionality [58]. While a module exists as a single instance within a program, multiple instances of a class can be created [57].

Within the OrcFxAPI, an example of a class is "Model", which is identified as a class by the keyword "class" placed before the class name. This class comprises a suite of functions created for managing a model in OrcaFlex. It exemplifies the utility of classes in structuring and streamlining specific tasks.

Another class within the OrcFxAPI is "ObjectType". Unlike the "Model" class, "ObjectType" does not contain functions; instead, it houses a collection of variables that can be accessed through the class. This distinction showcases the flexibility of classes in structuring data and operations in software development. By providing a framework for bundling data with related functionalities, classes facilitate the management of large and complex codes, enhancing the clarity, maintainability, and scalability of software projects [58].

Function

In Python, functions consist of a series of statements engineered to execute specific tasks and deliver a return value to the caller, as described by the Python documentation [59]. This capability allows developers to define a function once and then invoke it with various inputs, significantly streamlining the codebase by eliminating the need to write the same code multiple times for different inputs [60]. Python functions are essential constructs that enable programmers to organize code efficiently and avoid repetition.

In Python, a function is defined using the keyword "def" followed by the function name and its parameters. This distinctive keyword marks the start of a function, making it identifiable within the OrcFxAPI environment. Functions can be defined within classes or modules; however, these classes or modules must be invoked or imported to access the functions contained within them. For instance, when creating an object in OrcaFlex using Python, the OrcFxAPI needs to be imported first. Following this, the "Model" class is invoked. Only then can the "CreateObject" function be used. This sequence ensures that the necessary context and dependencies are established for the function to operate correctly within the OrcaFlex simulation environment.

2.5.3 Blueprinting

Blueprint Visual Scripting, referred to as blueprinting, is a node-based programming language system in UE5 [61]. This system translates Blueprint into C++ making it possible for non-programmers to use C++ utilities [62]. C++ is another widely used high-level programming language, suited for game and software development [63]. This is the mainly used language in UE5 [64]. An example of how blueprinting looks like is shown in Figure 10, below.

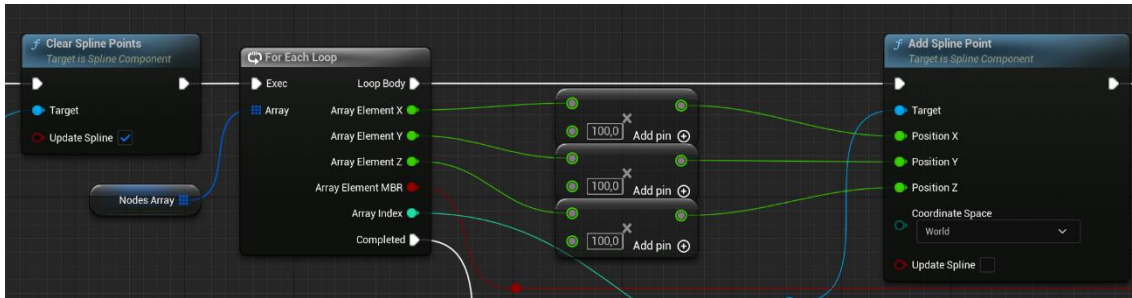


Figure 10: Blueprint example. This code runs a function that clears the spline points, loops through an array (where the newest data is stored) to add new spline points from the X, Y, and Z element in the array. This example converts the values of the elements from m to cm by multiplying with 100.

2.6 OrcaFlex Analysis

Building the model

Before performing OrcaFlex calculations, an initial model setup is required. A model consists of a marine environment and objects replicating the real-world system. The objects within the model replicate the structures to be analysed, while the environment defines the forces to which these objects are subjected [46].

Environment

To initiate the processing stage of the digital twin, an OrcaFlex data file is created with default environmental settings, which then need to be customized to replicate the specific real-world conditions accurately.

Objects

To simulate the real-world scenario the model requires a system of objects representing the real-world assets. OrcaFlex provides a suite of objects pivotal for the simulation, which are described in OrcaFlex's documentation [46].

OrcaFlex [46] imposes no inherent restrictions on the number of connections between objects, enabling the modelling of highly complex systems. This flexibility allows users to construct detailed simulations that can capture intricate interactions within marine systems. However, it's important to note that as the complexity of the model increases, so does the required time for the analysis.

Analysis

OrcaFlex offers two types of calculations, which is Static and Dynamic Simulations. The primary objective of conducting static calculations in OrcaFlex is to determine positions and orientations of each element within the model to achieve a state of equilibrium for all forces and moments [46]. Typically, this is both quick and reliable which is crucial for real-time monitoring and decision making.

There are two approaches to dynamic analysis, which are frequent domain, which is linear, and time domain, which is fully nonlinear [46].

Executing the OrcaFlex calculations through a Python script is simply done by calling the function by name within the model class. To perform a static analysis, the function “CalculateStatics” is used, while dynamic analysis is initiated with the function called “RunSimulation”. When the static or dynamic calculation is executed, the line results are available.

2.7 System Architecture

System architecture describes how the system is built and organized [65]. Some system architecture models are described below.

Client-Server Architecture

Client-Server architecture is a model where clients request and receive services and resources from a centralised server [66]. This structural design facilitates distributed processing, enhances scalability, and ensures a clear division of responsibilities between the client-side operations and the server-side processes [65].

Microservices architecture

This model designs an application as a collection of distinct services, characterised by their ability to be deployed independently and their loose coupling [67].

Each service is designed around a unique business function and interacts with other services through simple, lightweight protocols [65]. This approach facilitates scaling individual components as needed, simplifies the deployment process, and enhances fault isolation, leading to more resilient and flexible applications [65].

Layered Architecture

Layered architecture is a model where the system is organised by multiple layers [65]. Layers consists of modules or components with similar functionalities, and the layer performs a specific role within the system [68]. This architectural approach promotes modularity and adaptability, streamlining the system for easier testing, maintenance, and scalability [69].

2.8 Related Work

Digital twin technology is constantly developing and is used in an increasing number of industries at various levels.

TechnipFMC collaborated closely with DNV developing an industry standard for digital twins, DNV-RP-A204 Assurance of Digital Twins [70], which was released in October 2020 [71]. The standard outlines a recommended practice for systematically developing reliable digital twins, aiming to ensure their trustworthiness while minimizing safety and financial risks from concepts through to operation [70].

Digital Twin for Subsea Operations / Subsea Structures

A research performed by Wanasinghe et al. [72] in 2020, suggest that digital twins in the oil and gas industry is at an early stage. They emphasize the importance of HSE within the oil and gas industry, which has forced the industry to be highly regulated. Together with lower oil prices, the strict regulations have pushed companies within the industry to research and develop innovative solutions to enhance efficiency [72]. This

is an important confirmation to be able to start investing in the development of digital twin technology in this industry.

In October 2023, the National Subsea Centre [73], in collaboration with industry partner SeaFlo [74], announced the establishment of an Asset Digital Twin Laboratory [75]. This new facility is dedicated to researching, developing, and implementing digital twin solutions aimed at predicting maintenance needs to enhance the longevity of subsea structures, equipment, and systems [75].

4Subsea delivers a range of services, products, and support systems leveraging digital twin technology, enabling operators to anticipate events before they occur [76]. Their solutions enable real-time data collection via IoT, offering benefits such as an easy-to-use digital service for decision support. This includes a straightforward, easy-to-install instrumentation package for continuous integrity monitoring of offshore wind turbines, autonomous retrofittable sensors enhanced with advanced machine learning algorithms, and domain expertise that assists operators in significantly reducing LCOE [76].

Dynamic Analysis for Cable Structure

Wang et al. [77] conducted research titled “Continuous Dynamic Analysis Method and Case Verification of Cable Structure Based on Digital Twin”. This study highlights the necessity for new construction analysis model for cable structures. The researchers compared analysis results using Midas, OrcaFlex, and Ansys, noting that the modelling process for both OrcaFlex and Ansys were more complex. Their findings suggest that continuous dynamic analysis during the construction phase of cable structures is theoretically feasible.

In the “Handbook on Design and Operation of Flexible Pipes” [78], the authors emphasize the need for industry’s need for monitoring, detection, and prediction solutions to ensure the integrity of flexible products. Additionally, they affirm the necessity of monitoring the product throughout its entire lifespan, concluding that this practice is increasingly acknowledged and universally accepted across the industry.

3 Methodology

In this chapter, the methodology used during the project of the master's thesis is presented in detail. This study involves several methods – Literature Study, Concept Development, Prototyping, and Evaluating Results – to create a proof of concept of the proposed solution.

3.1 Literature Review

The project requires well understanding of the today's method for monitoring operations with flexible products. To obtain this an in-depth literature study was performed in the initial phase of the project. The literature research focused on cable lay operations and the concept of digital twins, as they form a key component of the proposed solution presented in this master's theses.

The literature study encompassed a wide range of materials, including scientific articles, books, reports, previous master's thesis, and other relevant publications. This study was approached with a critical eye, carefully evaluating the reliability and creditability of each source. This thorough investigation was instrumental in laying a solid foundation for the thesis work.

3.2 Qualitative Research

Qualitative research is a method where the data collected are non-numerical, often in the form of text [79]. This method is typically used to gather in depth understanding of concepts, opinions, and experiences to better describe a problem or discover and discuss new ideas for research [80].

3.2.1 Interviews

An interview, as a method of qualitative research, employs questioning to gather information [81]. Interviews can be categorized as presented in Table 3.

Table 3: Types of interview [81]

<i>Type of Interview</i>	<i>Description</i>
<i>Structured Interview</i>	Involving a fixed set of questions delivered in a specific sequence. This type of interview is well suited for comparing responses between participants. This format is conducive to conducting interviews rapid and in large quantities, as it streamlines the process and ensures consistency in the information collected.
<i>Unstructured Interview</i>	Allowing for a more spontaneous and flexible dialogue. The method allows for gathering detailed information and follow-up questions. During these interviews it is important to not ask leading questions. This type of interview is suitable for delving into and contextualization initial hypotheses.
<i>Semi-structured</i>	Offering a balance, featuring some preplanned questions while also permitting exploratory conversation. This method ensures the dialogue to follow the desired direction while also allowing follow-up questions.

Conducting interviews with employees across different departments and roles is a crucial qualitative research method aimed to exploring and understanding their experiences, perceptions, and attitudes. This approach is particularly valuable in ensuring that the project outcomes are well-rounded and reflective of the diverse perspectives within an organization.

Unstructured interviews, primarily conducted as dialogues, did occur continuously throughout the project to validate the work, and gather feedback, ensuring a steady progress.

3.3 Prototype Development

Throughout the project, a prototype will be developed to serve as a proof of concept. This product is a software application that utilizes digital twin technology to monitor subsea operations, specifically focusing on cable lay operations.

3.3.1 Concept Development

Pugh's method was employed to evaluate various concepts against established requirements and identify the optimal solution. This method uses one reference concept to benchmark the others. The concepts are rated relative to this reference, with higher scores indicating better alignment with the requirements.

3.3.2 Software

Orcaflex

Orcaflex is a software for performing dynamic analysis of offshore systems [82]. This software is widely used and well recognized in the subsea engineering industry and is frequently used by the Dynamics Department of DeepOcean.

This software facilitates the analysis of the cable, automatically assessing whether it is subjected to acceptable levels of stress via a programming script that enables tailored analysis. Additionally, it allows for the exportation of analysis results for subsequent utilization.

Unreal Engine

UE5 is a game engine for real-time 3D content [83]. This tool is daily used by the Visuals Department of DeepOcean, which is the department responsible for the development of digital twins. This game engine is one of the most productive in the industry [84].

This software was used for creating the visual part of the digital twin, including the live view and the alarm system.

Figma

Figma, an interface design tool, was used to design visual components, such as backgrounds and symbols used in the live view module.

3.3.3 Programming Languages

The development of the prototype requires two different programming languages, Python and Blueprinting. It is possible to use C++ instead of or in combination with Blueprinting in UE5. The prototype will be built by Blueprints, but the finished product might as well be built up of both.

Python is used to access OrcaFlex and to automate the analysis part of the system. While Blueprinting is used to build the live view software in UE5.

3.3.4 Hardware

Real-time data collection necessitates the strategic placement of sensors and transponders on the product during installation. While developing the prototype does not require the actual use of this hardware, it is essential to consider the optimal locations for these devices and the specific types of transponders that should be used.

3.3.5 Analysis

Providing insight into the cable integrity during the installation requires calculations and analysis of the data gathered from the real-world. This can be done in the dynamic analysis software Orcina OrcaFlex. Using the Python API, OrcFxAPI, the calculation can be performed in near real-time. OrcFxAPI makes it possible to access OrcaFlex functionality within a Python script.

The Python script will generate an estimated model of the cable, collecting position data that will be exported to UE5 to create a new visual representation. The OrcaFlex model will provide more data points than those initially collected from the real world.

In addition to creating a model, the Python script will run a rapid simulation of the model. It will calculate the bending radius and tensions in the cable. The script also saves the results in a required file format, typically JSON, making the data usable for the visual representation and the response system.

3.3.6 Visual Representation

The visual representation of the flexible product will be performed in UE5, by constructing a 3D-model of the product. The model is created by sweeping a geometry, resembling the product, along a path. This method divides the configuration of the product and the cross-section area into two distinct entities, allowing for independent modification. The data shaping these elements is derived from two discrete sources: the cross-sectional specifications are sourced from the product's data sheet, and the path configuration is informed by real-time data collected during the operation.

This is done by Blueprint programming, as this allows for near real-time simulation by updating the model continuously during the operation. The program takes the x, y, and z positions from the OrcaFlex results as input to create a set of spline points in the digital world. Then a spline is created through the spline points. The spline type that is used is called "Hermite".

3.3.7 Alarm System

Elevating the digital twin involves integrating an alarm system, thereby transforming it into an informative twin. This enhancement is achieved by alerting the user when limit values are on the verge of being exceeded. Essentially, this process entails comparing the line results from the analysis to the limit values specified in the product data sheet. If the result is exceeding the limit value (assigned to a safety factor) the alarm system will be activated.

This system is created in a Python script through an if-statement returning a value telling if the limit is exceeded or not. The next step in the alarm system is created in UE5 by a Blueprint program. If the alarm system is activated a message box will appear within the live view software and the colour of the cable will change to a colour indicating an alarm, typically red or orange [85]. The development of this system is further described in 4.6.5.

3.4 Testing

Testing is a fundamental part of product development [86], as it gives feedback during the development ensuring the success of the product. Feedback is important for optimizing the product and to discover any unforeseen issues.

3.4.1 Unit Testing

During the prototype development, unit tests were performed for the code parts of the product. These tests served the purpose of validating if the code worked as intended and met the requirements [87].

Unit Test of Analysis Program

The Analysis Program module consists of two units. Unit 1 creates the model and prepares it for analysis, while unit 2 runs the calculation and saves the desired results. During the development of these two units, there were performed unit tests. Table 11 and Table 14 in 5.1.1 presents the control questions of the unit test for representatively unit 1 and unit 2.

Unit Test of Live View

A unit test of the code in the Live View module was also performed. Control questions for the unit test can be found in Table 16 in 5.1.2.

3.4.2 Prototype Testing

When the prototype was developed, it was tested to verify if it aligned with the product requirements. The control questions for the prototype test are detailed in Table 20.

Testing the system required:

- Pre-defined OrcaFlex specifications, presented in Table 17.
- A test scenario to replicate the real-world data.
- A script (AnalysisProgram.py in 8.1) running automatically taking real-world data as input and the live view input data as output.
- A live view program updating the model every time the input data is updated.

3.5 Use of Artificial Intelligence

NMBU allows the use of AI in master's theses [88], adhering to specific regulations. Using AI requires thorough security of the credibility and sources used by the AI model. In this thesis, ChatGPT (GPT-4) was employed primarily to overcome writing blocks and to assist in identifying relevant literature and sources, not as a direct citation source. AI was not directly used to generate content for this assignment but served as a tool to refine search terms for literature searches and to provide links to articles and websites that could be beneficial for the project, an example of this is presented in Figure 11. It was also used to assist the writing by proposing better synonyms for words and to help construct certain sentences to obtain better readability.

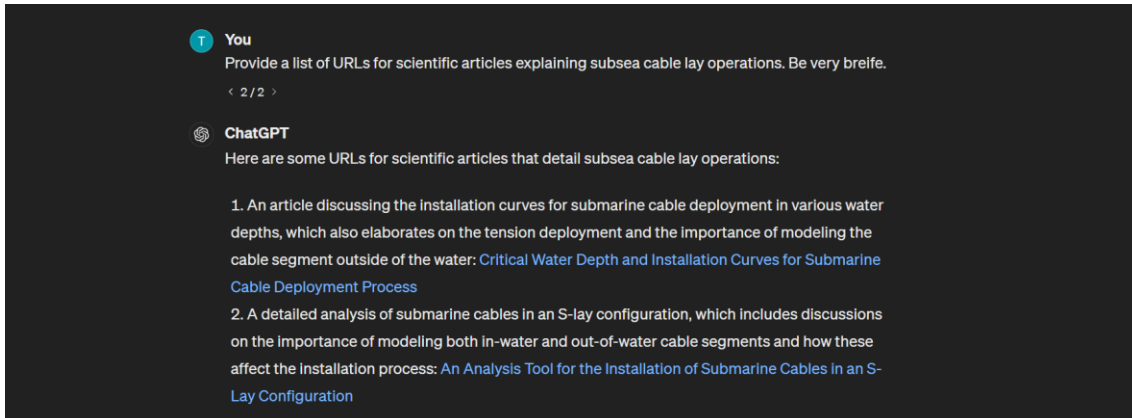


Figure 11: Example of prompt and answer using ChatGPT.

4 Design and Development

This chapter presents the design and development of a DT-based solution to improve monitoring of flexible subsea products during installation.

4.1 Pre-Planning

The first stages of the product development included systematically determining the mission statement for the product.

4.1.1 Company Vision

DeepOcean have a vision of being “a world-leading ocean services provider, enabling the energy transition and sustainable use of ocean resources” [2].

4.1.2 Product Vision

The vision of the product is to optimize subsea operations, related to flexible products, by real-time monitoring and alarm system. Ensure product integrity during installation / lay operations by analysing the catenary shape to ensure the bending moment / strain within acceptable range [89]. Reduce number of ROVs (from two to one) [90].

4.1.3 Product Purpose

The purpose of the product is to provide an effective solution for monitoring flexible products during installation operations, by utilizing the concept of DT technology.

By offering real-time tracking of the position of the flexible product, a DT could act as an additional observational tool for ROV operators, effectively serving as their “third eye”. Beyond its role in operational oversight, the DT facilitates the creation of additional documentation, serving as a comprehensive record of the operation. This technology is capable of providing more accurate and detailed information about both the operation and the subsea environment. Its capability for round-the-clock monitoring ensures continuous oversight. Additionally, the DT is invaluable for documenting the cable’s condition and for planning maintenance tasks, ensuring the longevity and reliability of subsea infrastructure.

4.1.4 Customer

The customer of the product was initially set as the management of the company (DeepOcean), who plan to use it to optimize the company’s services. Meanwhile, the intended users of the product are the employees, as they are the ones performing the services. More specifically, the intended users are the ship supervisor, ROV pilots, and in some cases engineers of different departments.

4.1.5 Mission Statement

The mission statement, presented in Table 4, was established to obtain clarity of the intension of the product to ensure that the development was headed in the right direction. The objective of the product was, at this stage, as:

- Obtain awareness
- Better documentation
- Increased efficiency

The product line is “Digital Twin”, which there was not fully developed a structured product line specification for at this stage of the project. The company was at an early stage of their DT journey. However, the Visuals Department described DT as real-time visualisation.

Table 4: Mission Statement of the Live View Digital Twin for Subsea Operations.

Mission Statement: Cable lay Monitor Digital Twin

<i>Product Description</i>	<ul style="list-style-type: none"> • Real-time DT-based monitoring of cable lay/installation operation in subsea environment.
<i>Benefit Proposition</i>	<ul style="list-style-type: none"> • Better documentation – know the state of the cable, better lifetime estimation, planning future operations, predict IMR intervals. • Monitor bending radius to prevent damage. • Third eye of operation.
<i>Key Business Goals</i>	<ul style="list-style-type: none"> • Reduce risk. • Better documentation. • Operational awareness. • Reduce time.
<i>Primary Market</i>	<ul style="list-style-type: none"> • Subsea Engineering and Services Providers.
<i>Secondary Markets</i>	<ul style="list-style-type: none"> • Marine Research and Exploration. • Oil and Gas Companies. • Offshore Wind Energy Developers.
<i>Assumption and Constraints</i>	<ul style="list-style-type: none"> • Meet industry standards and regulations. • Position data of vessel and TDP. • Product data of flexible product and buoyancy modules.
<i>Stakeholders</i>	<ul style="list-style-type: none"> • Organizations in the offshore energy industry and marine research sector. • Subsea Engineering and Service Providers. • Project Managers. • Competitors. • Sales and Marketing Teams.

4.1.6 Integrated Product Development Plan

The project (this thesis) was started at Step 2 Concept Development of the IPD plan detailed in Figure 12. This step included setting technical specifications and requirements for the product. At this stage, concepts were developed to purpose solutions of the initial problem statement. This stage was responsible for ensuring that the product would be developed as a viable solution.

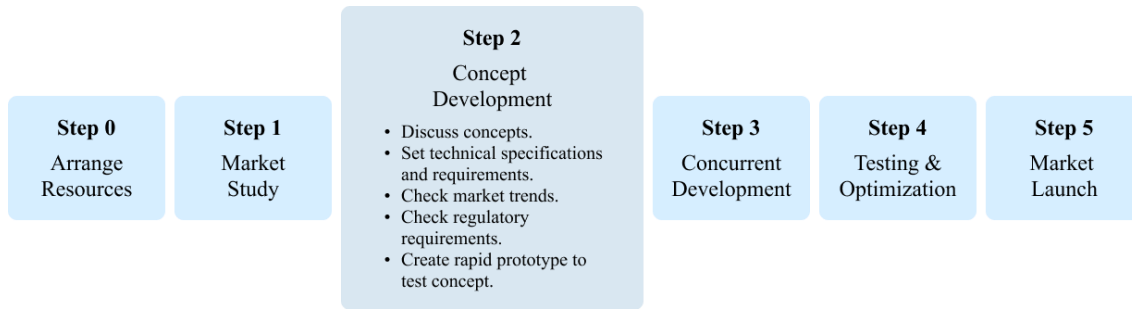


Figure 12: IPD plan, where step 0 and 1 was performed in advance of the project, while step 3 to 5 remains. Step 2 is the phase of the product development during this project.

4.1.7 Product Requirements and Specifications

Customer Needs

Customer needs were identified through interviews, in the form of dialogues, with employees of the customer. These conversations provided the critical insights necessary to transform expressed needs into definitive customer requirements, presented in Table 5. These requirements served as the foundation for constructing the House of Quality (HOQ) chart, as depicted in Figure 13. By assigning weights to each customer requirements, a hierarchy of the importance was established, ensuring that the product’s development is aligned with customer priorities to guarantee satisfaction.

Table 5: Customer needs and responding customer requirements.

Needs	Requirements
Ensuring product integrity	Reliability; Safe Operation; Increased Documentation
Reduce number of ROVs	Efficiency; Cost-efficiency
Utilized by several	Ease of use; Modularity; Scalability
New product development	Cost-efficiency; Lower price; Ease of integration; Scalability; Modularity
Increased service offer	Reliability; Efficiency; Cost-efficiency; Ease of use; Scalability; Ease of integration

Customer Requirements

Key customer requirements such as reliability, efficiency, safe operation, cost-effectiveness, and increased documentation are paramount, each receiving the maximum importance rating of 5 out of 5. These interrelated requirements are integral to the company’s HSEQ commitments [91]. Environmental impact is also a significant HSEQ concern, but within the context of this product, it has been assigned a lower importance score of 1 out of 5. This lower prioritization is due to the company’s specific environmental goals, which are primarily focused on reducing CO₂ emissions [92].

Additional customer requirements, while considered less critical, still play a role in the product's success. These include ease of use, lower price, ease of integration, modularity, and scalability. Each of these factors contributes to the overall user experience and long-term viability of the product, though they do not bear the same weight as the primary requirements in the decision-making process.

Product specifications

Based on the identified customer needs technical requirements (also referred to as product specifications) were established. These specifications clearly outline the agreed-upon requirements necessary for achieving customer satisfaction [93]. They are detailed in the HOQ, with their respective prioritization presented in Figure 14.

House of Quality

The product specifications and their significance were determined using the House of Quality method, presented in Figure 13. The importance of each specification is depicted in a chart in Figure 14, highlighting the relative priority of these requirements within the HOQ framework.

Establishing product specifications and determine their importance was done through a HOQ method, presented in. The importance of each requirement, presented as specifications in the HOQ, is represented in the chart in.

Row #	Max Relationship Value Row	Relative Weight	Weight / Importance	Direction of Improvement	Column #														
					1	2	3	4	5	6	7	8	9	10	11	12	13	14	
				1 = weak correlation 3 = medium correlation 9 = strong correlation	Maximum data update frequency	Maximum system update time	Minimum accuracy of position data	Minimum accuracy of calculations	Minimum accuracy of digital model	Maximum alarm system safety factor	Minimum storage capacity	Maximum documentation update time	Minimum documentation coverage	Maximum installation time	Mean time before failure	Minimum efficiency increase	Maximum onboarding time	Minimum user satisfaction score	
1	9	13%	5	Reliability	9	9	9	9	9	9	3	1	1			9	3		9
2	9	13%	5	Efficiency	9	9	9	9	9	3	3	1	1	9	3	9	3	9	
3	9	3%	1	Low Price		9	9	9	9	1	3					1			
4	9	13%	5	Safe Operation	9	9	9	9	9	9				1	1	3		9	
5	9	13%	5	Increased Documentation	3		9	9	9	1	9				1	9		9	
6	9	8%	3	Ease of Use		9				9		9	9	9	9	1	9	9	
7	9	13%	5	Cost-Effectiveness		3	1	1	1	9				9	1	9	9	9	
8	3	3%	1	Environmental Impact						3				3	1	1	1	1	
9	9	8%	3	Ease of Integration									3	9		9	9	9	
10	9	5%	2	Modularity		9						3							
11	9	10%	4	Scalability	3	9	3	3	3		3	9	9	1	1			1	
				Unit	seconds	seconds	%	%	%	NA	TB	days	%	minutes	hours	%	hours	%	
				Target or Limit Value	1	3	95	95	95	3	1.00	7	80	NA	1000	10	NA	80	
				Difficulty	3	3	4	4	4	2	2	1	1	2	3	3	2	1	
				Max Relationship Value in Column	9	9	9	9	9	9	9	9	9	9	9	9	9	9	
				Weight / Importance	162	240	206	206	206	186	90	79	82	156	107	197	114	284	
				Relative Weight	7%	10%	9%	9%	9%	8%	4%	3%	4%	7%	5%	9%	5%	12%	

Figure 13: House of Quality presenting the customer requirements vertically and the technical requirements horizontal.

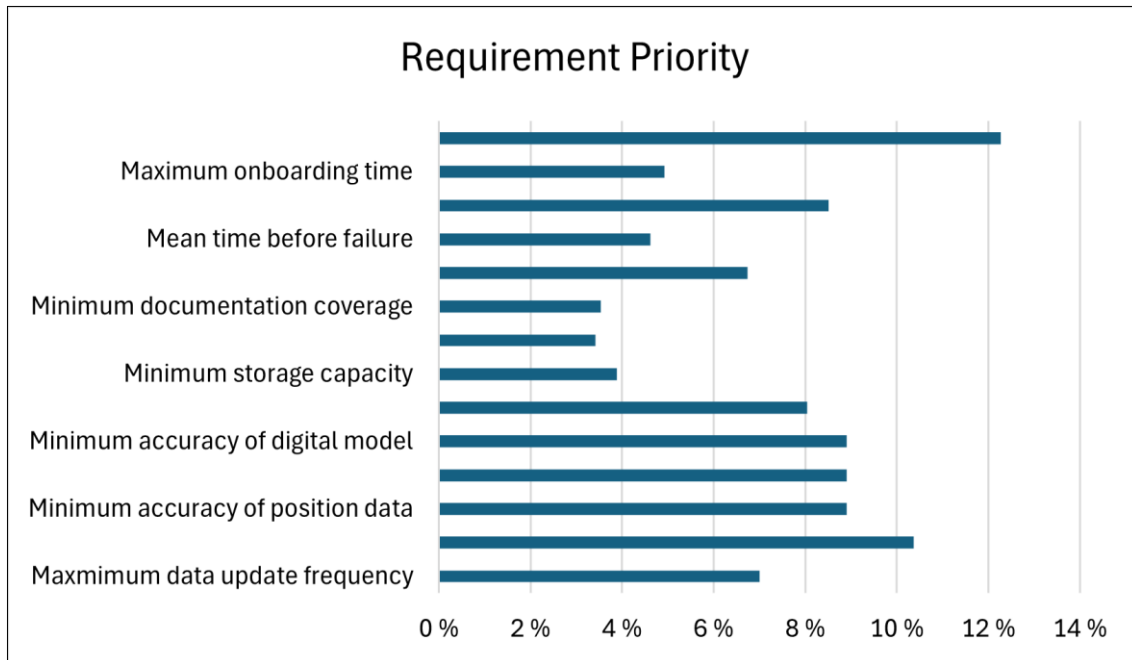


Figure 14: Requirement prioritisation chart based on HOQ.

4.2 Concept Design

A function analysis system technique (FAST) diagram was developed to determine the specific functions the product delivers to its user. FAST is a method used to dissect and comprehend complex systems by transforming the activities carried out within the system into functions that serve its users [94]. Activities are the “how” features, while functions are the “why” functions. The importance of this is presented by J. E. Bartolomei and T. Miller [94], where they describe the functions as the value for the user, while the activities are the actions that results in the value.

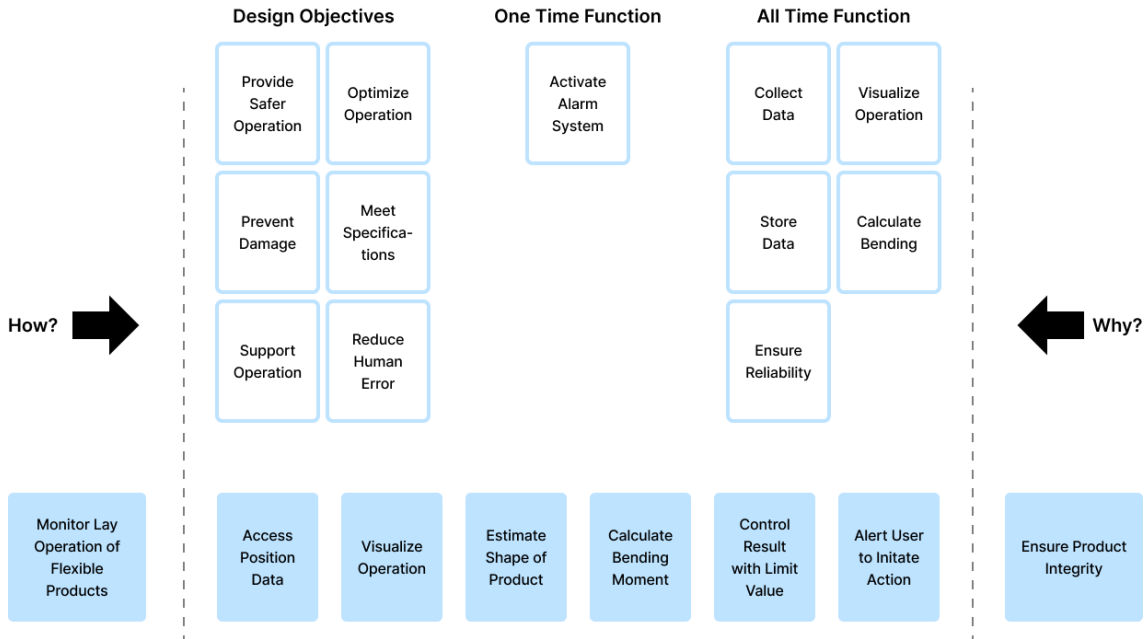


Figure 15: Function analysis system technique diagram where features are made into functions.

The digital twin process consists of different phases:

1. Real-world data collection
2. Creation of digital model of real-world asset
3. Performing analysis on the real-world data
4. Presenting the data to the user
5. Storing data

Each of these phases has their separate concept development.

Real-world data collection is an important part of the digital twin, as this sets the limitations of the following steps. Due to its importance, different concepts were developed in this phase, which is presented in 4.3.

4.2.1 Analysis Concept

Several analysis software – OrcaFlex, Ansys Mechanical, Ansys Workbench, and SolidWorks – was considered for calculating the integrity of the product. The software should be able to provide following functionality:

- Estimate Shape of Product
- Calculate Integrity of Product
- Compare Results with Limit Value

Requirements were set and weighted to compare the four software against each other, this was done through Pugh’s model in Figure 16.

Requirements		Weights	OrcaFlex Reference	Ansys Mechanical	Ansys Workbench	SolidWorks
1	Analysis Capabilities	5	0	-1	-1	-2
2	Ease of Use	2	0	0	0	0
3	Integration	2	0	0	0	2
4	Material and Property Libraries	2	0	1	1	0
5	Industry Fit	5	0	-1	-1	-1
7	Automation	5	0	-1	0	0
8	Scalability	4	0	1	1	-1
9	Solver Efficiency	4	0	1	1	0
10	Post-Processing	4	0	1	1	0
11	Price	3	0	-1	-1	0
12	Learning Resources	2	0	-1	-1	-2
Score			0	-6	-1	-19

Figure 16: Pugh's model comparing analysis software, showing that OrcaFlex scored the highest.

4.2.2 3D-Replication Concept

The concept for visualising the operation was established to be through a 3D-model replicating the real-world assets. The following software were considered:

- Autodesk Fusion 360
- Autodesk 3ds Max
- Blender
- Unreal Engine 5
- SolidWorks

These were compared by setting weighted requirements and scoring them for each software using Pugh's model, presented in Figure 17.

Requirements		Weights	Autodesk Fusion 360 Reference	Autodesk 3ds Max	Blender	Unreal Engine 5	SolidWorks
1	Price	3	0	-2	2	-1	-2
2	Prototype Cost	5	0	0	1	1	0
3	Compatibility	4	0	0	-1	-2	1
4	Ease of Use	4	0	-1	-1	0	0
5	Industry Fit	3	0	1	-1	-2	1
7	Integration	3	0	1	-1	0	1
8	Modeling Tools	2	0	-1	0	-1	1
9	Simulation Tools	2	0	-1	0	-1	1
10	Rendering	5	0	1	1	2	0
11	Automation	5	0	0	1	1	0
12	Learning Resources	2	0	0	1	1	0
13	Scalability	5	0	0	0	2	0
Score			0	-3	9	11	8

Figure 17: Pugh's model for comparing 3D-modeling software, where Unreal Engine 5 scores the highest.

4.2.3 Alarm System Concept

A few concepts were developed for what components the alarm system should include.

Vision

The purpose of the alarm system is to make the user to take immediate action when a critical situation has occurred. It should be activated immediately when critical parameters are near to being exceeded, within a safety factor allowing for time to correct the situation before any damage has occurred.

Components

1. Documentation (if the limit values are exceeded, this must be documented)
2. Alerting the user within short time
 - a. Visual alert
 - i. Message box
 - ii. Change colour of 3D-model to red

4.3 Real-World Data Collecting Concepts

Three different concepts for the real-world data collection were developed and discussed. Among these, two strategies involved the integration of additional transponders to enhance data collection capabilities. Conversely, another approach posited that the currently available data sufficed for operational needs, thus not necessitating further transponder deployment.

4.3.1 Temporary Transponders

One of the methods considered for data collection involves affixing temporary transponders to the product for the duration of the lay operation. This process entails both the installation and subsequent removal of the transponders, within the same operational timeframe. Consequently, data would be accessible solely throughout the actual laying process. The removal of these transponders post-operation would necessitate the deployment of a ROV and tooling, which would carry out the uninstallation task.

Usage

This approach would be best suited to FRs and umbilical installation, as they are related to the oil and gas industry, with higher budgets, and there is usually one or a few products being installed. This means that even if it takes some time to install the transponders and to uninstall them again, it will not be much of a problem.

Benefits

The utilization of temporary transponders presents several benefits, including cost savings as these devices are generally less expensive than the durable versions designed for "smart buoyancy clamps." Additionally, deploying multiple transponders can result in a richer data set, enhancing the granularity and accuracy of the information collected. This abundance of precise data can be invaluable in ensuring the integrity of the cable-laying operation and subsequent monitoring.

Disadvantages

Among the disadvantages, the use of temporary transponders for data collection extends the duration of cable-laying operations due to the additional time needed to install the transponders onto the product as it is deployed. Furthermore, the uninstallation process not only demands extra time but also the deployment of an ROV and its crew, which could result in increased operational costs. These factors collectively could lead to a more protracted and expensive installation phase.

4.3.2 Smart Buoyancy Clamps

An innovative approach involves engineering buoyancy clamps equipped with long-lasting, integrated transponders. This design requires a strategic framework for data utilization that extends beyond installation, ensuring that the gathered information continues to deliver value throughout the operational lifecycle. In the context of a floating wind farm, such a system could play a pivotal role in real-time monitoring of cable integrity during active phases. Additionally, this data-rich approach provides a robust

foundation for forecasting and planning Inspection, Maintenance, and Repair (IMR) activities, thereby enhancing the operational efficiency and longevity of the wind farm's infrastructure.

Usage

This solution is primarily designed for IACs utilized in offshore wind farms. With the burgeoning growth in the renewable energy sector, floating offshore wind farms are poised for a significant expansion. Projections from DNV's 2023 Energy Transition Outlook suggest a surge in capacity, with forecasts anticipating an increase to nearly 270 GW by 2050 [95]. This expansion underscores the potential of the solution to play an important role in supporting the scalable and sustainable development of global wind energy infrastructure.

Market

The development of floating wind farms necessitates the deployment of numerous buoyancy modules, typically fashioned as clamps, essential for achieving the desired wave-shaped cable configuration that is critical for maintaining cable integrity. According to estimates from BVG Associates [16], the expenditure associated with buoyancy solutions for a 450 MW wind farm is approximately 2.5 million GBP, highlighting the significant scale of buoyancy module utilization. Given this extensive use, the innovation of a "smart buoyancy clamp" represents a promising opportunity in a market poised for substantial growth.

Presented by BVG Associates [16], some buoyancy module suppliers are Balmoral, CRP Subsea, DeepWater Buoyancy, SBT Energy, and Tekmar. If the solution is to develop the "smart buoyancy clamp", these suppliers would be direct competitors.

Benefits

This solution offers considerable time savings related to the lay operation by integrating permanent transponders into the buoyancy clamps, eliminating the need for their removal post-installation, as is the case with temporary transponders. Additionally, this integration facilitates ongoing monitoring of cable integrity, enhancing the functionality of the digital twin. This scalable approach not only supports real-time operational oversight but also enables predictive maintenance forecasting, thereby contributing to the extended lifespan and reliability of subsea infrastructures.

Consequently, this solution requires convincing the industry that the transponders not only improve the installation effectiveness but also facilitates cost savings in later life cycle stages, such as maintenance. By leveraging the DT to accurately predict maintenance schedules, it becomes possible to minimize inefficiently allocated time and resources on unnecessary inspections, thereby offering a potential counterbalance to the initial higher investment.

Disadvantages

According to NVE [19], the LCOE related to floating offshore wind power production is significantly higher compared to other wind power production methods. As a result, floating offshore wind emerges as the most expensive option for wind power generation, necessitating cost reduction across all stages of its life cycle.

The introduction of durable transponders to develop a smart buoyancy module, contributes to increase costs compared to currently available BMs. Such financial implications may lead to adversity from the industry, including operators or owners of wind farms, due to the elevated expenses.

4.3.3 Utilizing Current Available Data

The data available today might be enough for the digital twin to be successful. This would require detailed data about the product and the environment to obtain the most accurate simulation as possible.

When building the model in OrcaFlex, the line is resembling the flexible product. This object can be specified by editing and adding line data, such as setting data for the line type (e.g. bending stiffness, Poisson ratio, and diameter)

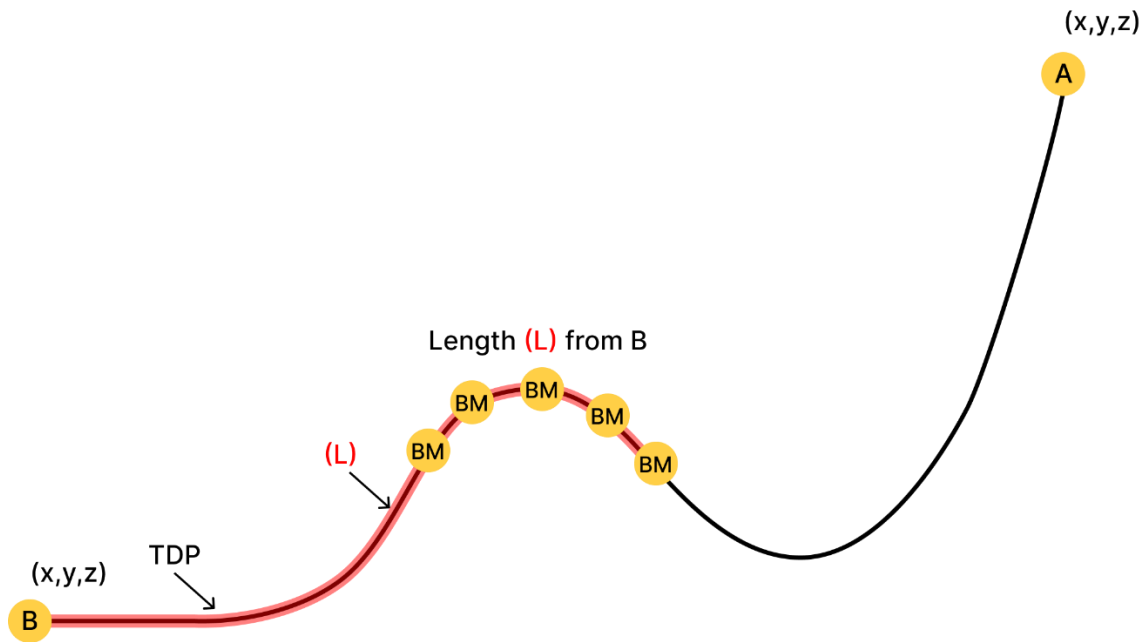


Figure 18: Available data during operation today. A and B represents end A of the product (where it leaves the vessel) and end B of the product (the end that leaves the vessel first) respectively, where the x -, y -, and z -coordinates are known. The length from B to each BM is known.

Available Data

As of today, there are relatively few data points of the real-world system, these are depicted in Figure 18. The existing method monitors the TDP, but at one point, the 'end B' becomes the TDP, which represents the placement of the product when it reaches touch down for the first time. Therefore, this is a known data point. However, this point will be calculated in relation to the vessel, which serves as the reference point for all other data points.

Usage

This solution is suitable for all products described in this thesis, including FRs, umbilicals, and IACs.

Benefits

The advantages of this approach include the elimination of the need for resource-intensive development efforts typically associated with data collection initiatives. This results in a more cost-effective solution compared to other concepts, providing significant savings without compromising the quality of data collection.

Disadvantages

A potential drawback of this solution is the risk of reduced data accuracy, which could compromise the reliability of operational documentation. In the worst-case scenario, this might result in the failure to detect critical conditions, potentially leading to undetected damage to the product.

4.3.4 Pugh's Method

Comparing the concepts has been done through Pugh's method. When requirements related to development are prioritized, which it is in an early phase, the concept that should be prioritized is the Available Data Utilization as it scores the highest in Figure 19.

Requirements		Weights	Temporary Transponder Reference	Smart Buoyancy Clamp	Available Data Utilization
1	Reliability	5	0	0	-1
2	Installation Efficiency	5	0	2	2
3	Ease of Integration	4	0	1	2
4	Scalability	5	0	2	-1
5	Accuracy	5	0	0	-2
7	Resource Use	2	0	-2	2
8	Development Cost	5	0	-2	2
9	Development Time	4	0	-2	2
10	Amount of Data	5	0	0	-2
11	Maintenance	2	0	-2	0
12	Compatibility	3	0	-1	0
Score			0	-5	10

Figure 19: Pugh's method used to compare data collection concepts.

4.3.5 Summary

Three distinct concepts were developed for data collection within the real-world part of the digital twin system. Each concept presents its own set of advantages and challenges that was thoroughly evaluated before any decision of further development was established. The discussed concepts are as follows:

1. Installing temporary transponders along the flexible product.
2. Developing smart buoyancy, which integrates durable transponders within buoyancy modules.
3. Utilizing currently available data to conduct the most accurate simulations possible.

It was decided that the third concept (utilizing available data) should be tested first, as this concept did not require any additional hardware for data collection, making it the least time-intensive of the three concepts. However, a goal of developing the prototype to be scalable, allowing for further development suited for the two other proposed solutions, was set for this solution.

4.4 Product Architecture

The product was divided into four key modules – Real-World, Analysis Program, Live View, and Storage System – all responsible for separate stages of the DT process. Each module was then built up of components offering unique functional elements.

Real-World Module

The Real-World module functions as the data collection of the real-world assets, which is the initial stage of the DT process. This module was built up of two components – Vessel Data and Product Sheet – both providing essential data to the DT model.

The function of the “Vessel Data” component is to provide real-world accurate data to the OrcaFlex model representing the real-world scenario. This data is accessed via MQTT subscription giving the RAO of the vessel, environmental data, TDP of the product, and position of ROV (or other assets used during the operation).

The “Product Sheet” serves the function of providing limitational data for the calculation of the integrity of the product. Monitoring the bending radius requires accessing the MBR from the data sheet of the product.

Analysis Program Module

The purpose of the Analysis Program module is to calculate the integrity of the product. This module consists of several components – Analysis Preparation, OrcaFlex Analysis, Comparing, and Result Storing – serving a specific role in the DT process, which are further described in Table 6.

The module consists of one python script that includes all the components performing its own task. However, the components must be placed in a specific order in the script for them to be successful, as these are all related to each other and uses the outcome of the previous component.

Example:

A model must be created in OrcaFlex before the analysis can be performed, and the analysis must be performed in order to access the results and so on.

Table 6: Analysis Program components with input, output, and description.

Component	Input	Output	Description
Analysis Preparation	Vessel Data, Data Sheet	OrcaFlex model	Creates a model in OrcaFlex simulating the real-world, using OrcFxAPI.
OrcaFlex Analysis	OrcaFlex model	OrcaFlex simulation, Static results	Analyses the model and accesses the static results of the line object, using OrcFxAPI.
Comparing	Data Sheet, Static results	Boolean value	Compares the result value to the limit value from the data sheet and creates a Boolean based on an if-statement.
Result Storing	Static results, Boolean value	JSON file	Stores the results in a JSON structure and saves it to the desired location.

The Analysis Program was developed as a Python script including two units. The first unit builds the OrcaFlex Model and makes it ready for analysis, while the second unit runs a static calculation and accesses the static results of the line object before it is compared to limit values and saved to a JSON file.

Requirements for the units:

Unit 1:

- Creates a OrcaFlex model – objects, properties
- Accurate model of the real-world (ideal is identical, but this is not feasible)
- Accurate configuration of the line object (flexible product)
- Make sure the model is ready for analysis

Unit 2:

- Runs a static calculation
- Access static result (any parameter)
- Compare result to limit
- Store results in desired format

Live View Module

Live View is the module interacting with the user, which is responsible for the visual representation of the real-world scenario. This is the final stage of the DT process and is the module lifting the DT to the next level from descriptive to informative. The Live View module consists of following components:

- 3D-model
- Alarm System
- User Interface

One of the components in the Live View module is the “3D-Model”, which is a Blueprint program creating a 3D-model of the operation. In the prototype, the sole asset created is the product being installed, as it is

the focus of monitoring. This model takes the node positions from the OrcaFlex results to create spline points, which results in a path representing the configuration of the flexible product. Another element of the “3D-Model” component is a mesh, representing the cross area of the product, which is swept along the path.

The “Alarm System” is a component consisting of several elements (Boolean value, message box, and colour change), which together has the objective of alerting the user of a potential danger and make the user take action to prevent any damage to the product. The Boolean value works as a switch for activating the two other elements, which are both visual elements of the alarm system. When the alarm system is activated (Boolean value = True), a message box will appear on the screen, created as a widget in UE5, informing the user of the situation. Simultaneously, the colour of the 3D-model of the affected points is set to red through a Blueprint program, working as an additional alerting tool.

As the module is interacting with the user, the “User Interface” element is an essential part of the module. It has the purpose of ensuring an intuitive interface of the software. This component includes widgets in UE5 creating buttons and frames.

The Live View module consists of the following elements:

- Blueprint class that creates the 3D-model of the flexible product being installed.
- Accesses OrcaFlex static results, stored in a JSON file called “AnalysisResutls.json”. Sets these points into an empty array in the UE5 project.
- Loops through the array that was created:
 - Scales the node positions of the line to meet the default unit in UE5 (m -> cm).
 - Adds line-node positions, referred to as x, y, and z, as spline points.
 - Sets static mesh to each point and sets as array elements in “Spline Mesh”.
 - Sets colour based on Boolean value determined by the comparison of result and limit value.
 - If True sets colour to red
 - If False sets colour to green
 - Creates warning message when Boolean value = True.
- Updates the spline:
 - Loops through the “Spline Mesh” array.
 - Gets the location and tangent of each spline point.
 - Sets the start and end of each spline element.

Storage System Module

The Storage System is where the data is stored and hosted. The other modules are interacting with this module through MQTT. The modules can subscribe or publish information via this module.

4.4.1 Product Schematic

A product schematic was established, describing how each element interacts within the product. Each element was placed within a module, where it serves its function. A simple illustration of the system is provided in Figure 20, while a more detailed description is depicted in Figure 21.

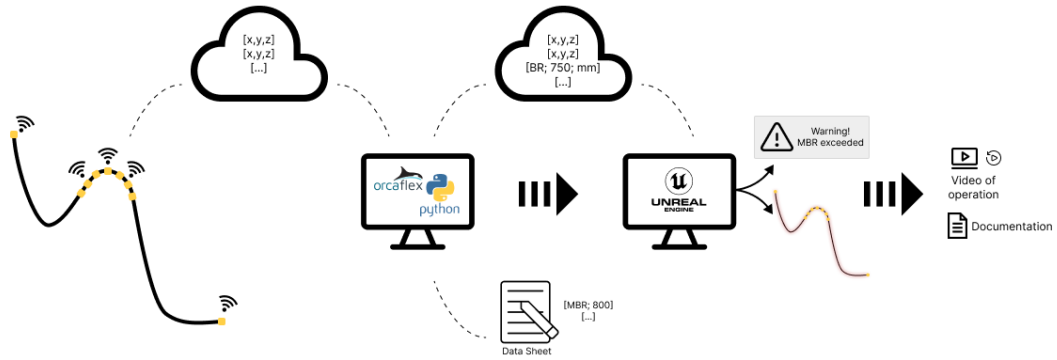


Figure 20: Visual schematic of the digital twin system.

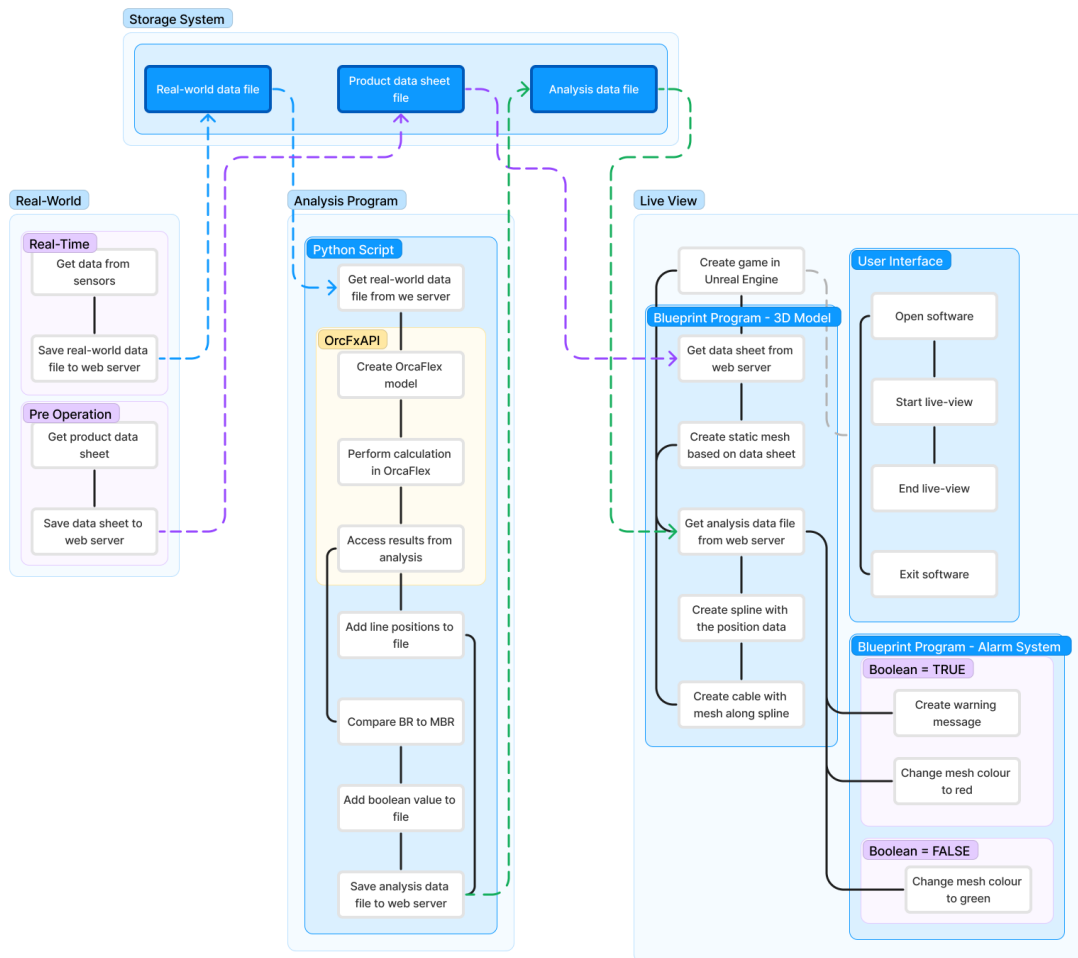


Figure 21: Product schematic of the DT system. Light blue boxes represent the modules, which are divided into different elements sorted in coloured boxes.

4.4.2 Data Management

Data management is the process of collecting, storing, and organizing data [96].

Data Storage

The data gathered and generated by the DT model must be stored in readable formats.

Data Management

The DT model consists of different sets of data, which are presented in Table 7, gathered, generated, and used at different stages of the process.

Firstly, the real-world data is collected by sensors and transmitted to the storage system. This data is later processed to gather new data to determine the integrity of the product and to create the digital model replicating the real-world scenario.

Data collection

- The real-world data is generated by a transponder, returning an identifying signal [97] (x-, y-, and z-coordinates).
- The analysis results data are generated by OrcaFlex, which is processed for further usage through a Python script.

The journey of the data:

- Position data A collected by transponders and stored for later use.
- Data A is used to create a model in OrcaFlex.
- Data B and C is collected from OrcaFlex analysis results and stored for later use.
- Data B is used to create a model in UE5 to replicate the real-world scenario.
- Data C is compared to data D and generating data E.
- Data E is used to activate the alarm system.

Table 7: Data categorisation

<i>Data set</i>	<i>Description</i>
<i>Data A</i>	Real-world x-, y-, z-position, surge, sway, heave, roll, pitch, and yaw
<i>Data B</i>	OrcaFlex line x-, y-, z-position
<i>Data C</i>	OrcaFlex node bending radius
<i>Data D</i>	Boolean value
<i>Data E</i>	Data sheet maximum bending radius

Data Collection

Data A is collected by transponders sited on the real-world assets (vessel, ROV, and BM). Data B, C, and D is collected from the OrcaFlex results. Data E is collected from the product's data sheet.

Data Transmission

Data A is transmitted from the operational site to the storage system through MQTT communication. It is then transmitted from the storage system via the Analysis Program. Data B, C, and D is transmitted from OrcaFlex to the storage system through the Python script. This data is then transmitted from the server to UE5 through a blueprint. Data E is transmitted from the data sheet to the Analysis Program manually by the user in advance of the operation.

Data Processing

The data collected throughout the DT process is archived on the server in readable formats. Data A, B, and D is all saved in JSON files, where data B, and D are located in the same file.

Data A is used as input parameter to create the OrcaFlex model, which is fundamental in prior to the analysis. Data B and C is directly generated by the calculation performed by OrcaFlex. Data C is compared to data E to generate data D. Data B and D is then used in UE5, where data B is used as input to create the 3D-model of the flexible product and data D is used to activate the alarm system.

Data Visualisation

Data B and D is visualised in UE5 respectively as a 3D-model and its colour. Data D is also visualised in the form of a message box appearing on the screen.

The objective is to visualise data A in UE5 as well, this will serve the user more detailed information about the operation.

4.5 System Architecture

The DT's system architecture has elements from different models – client-server architecture, microservices architecture, and layered architecture – making it a hybrid architecture. It is presented in Figure 22, below.

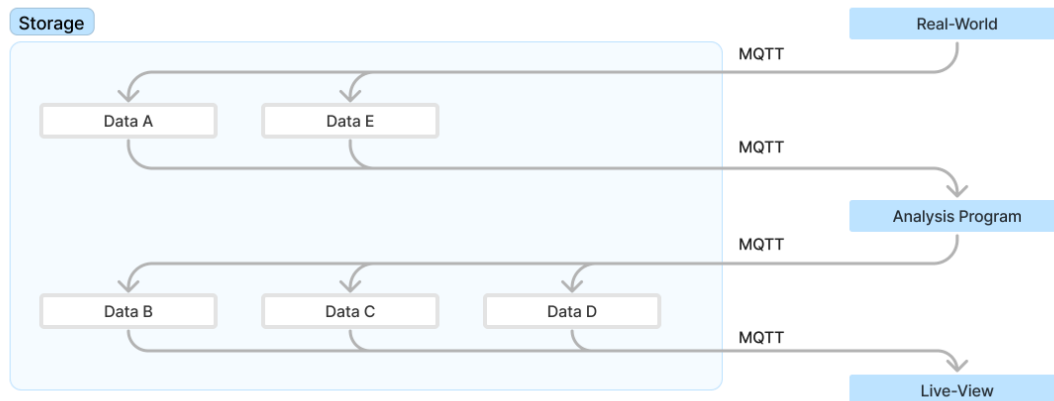


Figure 22: System architecture presenting the data (presented in Table 7) flow throughout the DT system.

The DT system was developed with a hybrid architecture, as it takes elements from different architecture models. It is structured into four distinct layers – Data, Application/Service, Presentation, and Storage – resembling a layered architecture model. Each layer fulfils a specific function within the DT system and interact with each other through simple communication, which reminds of a microservices architecture. The structure is also similar to a client-server architecture due to the way the layers are communicating with each other.

The “Data” layer is dedicated to the real-time collection of data, achieved through the use of transponders, with data retrieval facilitated via MQTT.

The “Application/Service” layer utilizes this data to evaluate the integrity of the flexible product. This evaluation is conducted by calculations within an analysis software, OrcaFlex, using a Python script.

Additionally, this layer is responsible for generating new data that is essential for the “Presentation” layer. Here, the data is transformed into a real-time visual simulation of the operation within a 3D environment, achieved using UE5, thereby serving as an effective monitoring tool during operations.

The final layer, “Storage”, acts as a repository, storing files generated and accessed during the digital twin's lifecycle.

4.6 Prototype

A prototype, acting as a proof of concept, was developed. It was used to decide whether the concept should be further invested in and continue the development. It was designed as the baseline of the end-product.

4.6.1 Real-World Data Simulation

The prototype was created by using a simple example of a tentative real-world scenario (inspired by [98]) during the testing and development. This was done by setting a list of data points representing a specific time of the operation. Which was developed as a JSON file including the 'end A' position of the flexible product, 'end B' position of the flexible product and the length of the flexible product.

4.6.2 OrcaFlex Model

OrcaFlex was used to replicate the real-world scenario. This was done by, firstly, creating a OrcaFlex model, which is an empty dat-file with default settings. The next stage included creating the two objects, which is a vessel and a line representing the CLV and the flexible product. When this was done, the object data was set for each of them. This included setting the x-, y-, and z- position accessed from the real-world data. Then clump attachments, representing the BMs, were added to the line object set with a specified distance from 'end B'.

The OrcaFlex model was created through a Python program by using OrcFxAPI. The program can be found in the AnalysisProgram.py script in 8.1.

4.6.3 Analysis

Performing analysis of the OrcaFlex model was done through a Python program. The program was set to perform a static calculation, which would give the required results. When the calculation was executed, the results, which includes x-, y-, and z- positions and bend radius for each node along the line, was accessed. The bend radius values for each node were compared to the MBR by an if-statement returning a Boolean value. The position values and the Boolean value was structured in a JSON format and saved to the storage system, making it available for further use.

4.6.4 Building the live 3D-Model in Unreal Engine 5

The 3D-model representing the flexible product was created by sweeping a cylindrical static mech along a spline reflecting the configuration of the product, for example a wave-shape.

A static mesh was created by adding a cylinder shape (an actor) to the UE5 world and editing the size of its diameter. Then the material (colour) of the mesh was set in the blueprint program. The default (Boolean value = False) material was set to a green colour.

In UE5, a blueprint class was created with the purpose of creating the 3D replication of the flexible product. This was done through a spline component, which was already existing within UE5.

Create spline:

1. Add spline component (from UE5)
2. Add spline points (x,y,z)
3. Add spline mesh component (cylindrical shape - static mesh)
4. Set material (yellow or red colour)
5. Update spline

4.6.5 Alarm System

The alarm system was created in UE5, built up by a widget blueprint and a material. In the "Spline_BP" (presented in 8.2) blueprint, a branch node was added allowing for actions to be executed only when the Boolean was set to "True", which was generated to activate the alarm system. Then two separate actions

were created, one executed only once, and one executed for each node where the limit value was exceeded (Boolean value = True).

Widget Blueprint

The message box was created as a widget, which will be added to the UE5 level once the alarm system is activated. It was created as a border filled with a yellow background making it look like a box. Then icons and text were added to present the message. And a button was added to the right corner of the box to make it possible to hide the widget when the button is clicked on. To make the button visually aesthetic, the button itself was made invisible and put above an image of the desired symbol, which made an illusion that the “X” symbol is the clickable component.

After the widget was designed, it was added to the “Spline_BP” (presented in 8.2) blueprint. The “Do Once” function, followed by a node creating the widget and a node adding it to the viewport, was used to make the widget appear only once, when the alarm system was activated (Boolean value = True).

Change of Colour

The “Set Material” function was integrated to the “Spline_BP” (presented in 8.2) blueprint class and would be executed after the creation of the warning message widget. This was orchestrated by a “Sequence” function allowing two (or more) separate actions to be executed one by one in a desired sequence.

The alarm material was created with its base colour set to red (Hex sRGB = FF00BFF), and an emissive colour configured as the base colour multiplied by 10. This material was then applied to the mesh at each node where the Boolean value was “True”.

4.6.6 Start Menu

A start menu was created as a widget added to the viewport once the UE5 level begins playing. The background design of the start menu was created as an image in Figma.

Two buttons were added to the widget, each serving a unique purpose when clicked on. The first button had the purpose of taking the user to the live view, the other one served as a way to exit the software.

5 Results

This chapter presents the results of the purposed solution, including unit tests of the software, scenario testing of the prototype, and feedback from users.

5.1 Unit Test

The purpose of the unit tests was to test different parts of the software separately during development to make sure each part was working as intended. The results of the tests are presented as tables and figures in the following subchapters.

5.1.1 Unit Test of Analysis Program Results

Unit testing, described in 3.4.1, was executed for the two code units of the Analysis Program module. Before the testing began, properties, (specified in Table 8) were manually added into the script.

Table 8: Properties of each object in OrcaFlex for unit testing of the Analysis Program.

<i>Object</i>	<i>Property</i>	<i>Value</i>	
<i>Vessel Line</i>	All	Default	
	End A connection	'Vessel1'	
	End A x-, y-, and z-position	See Table 9, Table 10, Table 12, and Table 13	
	End B x-, y-, and z-position	See Table 9, Table 10, Table 12, and Table 13	
	Length	See Table 9, Table 10, Table 12, and Table 13	
	Number of sections	1	
	Target segment length	[1.0]	
	Attachment type	Attachments	
	Number of attachments	8	
	Attachment z	[110, 105, 100, 95, 90, 85, 80, 75]	
	Attachment z relative to	['End B', 'End B', 'End B', 'End B', 'End B', 'End B', 'End B', 'End B']	
	Other	Default	
	<i>Clump</i>	Name	"Clump type 1"
		Mass	0.48
Volume		1.2	
Height		2	
Offset		0	
Align with		"Line axes"	
Other	Default		

Unit Test of Unit 1:

In addition to the properties established in Table 8, the end A, end B, and length parameters varied according to different test cases. The input parameters for control question 1 to 4 are presented in Table 9, while those for control question 5 are detailed in Table 10. The results of the tests are documented in Table 11, accompanied with comments.

Table 9: Simulated input data for unit test of unit 1 of the Analysis Program, set as key and value in JSON file, for testing control question 1-4.

<i>Key</i>	<i>Value</i>
<i>endA</i>	[35, 0, -7]
<i>endB</i>	[190, 5, -100]
<i>length</i>	200

Table 10: Simulated input data for unit test of unit 1 of the Analysis Program, set as key and value in JSON file, for testing control question 5.

<i>Key</i>	<i>Value</i>
<i>endA</i>	[35, 0, -7]
<i>endB</i>	[35, 5, -17]
<i>length</i>	0

Table 11: Control questions and result from unit test of unit 1 of the Analysis Program.

<i>Control question</i>	<i>Result</i>	<i>Comment</i>	
1	Is there created a new OrcaFlex model?	Yes	See Figure 23
2	Is the configuration set up correctly?	NA	The expected shape was created, but there was no comparison to the real world.
3	Is the object created in OrcaFlex?	Yes	See Figure 23
4	Are the properties set correctly in OrcaFlex?	Yes	See Figure 24
5	What happens if invalid data is provided to OrcaFlex?	Error	The python script generates an error message and won't proceed the calculation, see AnalysisProgram.py in the appendix 8.1.

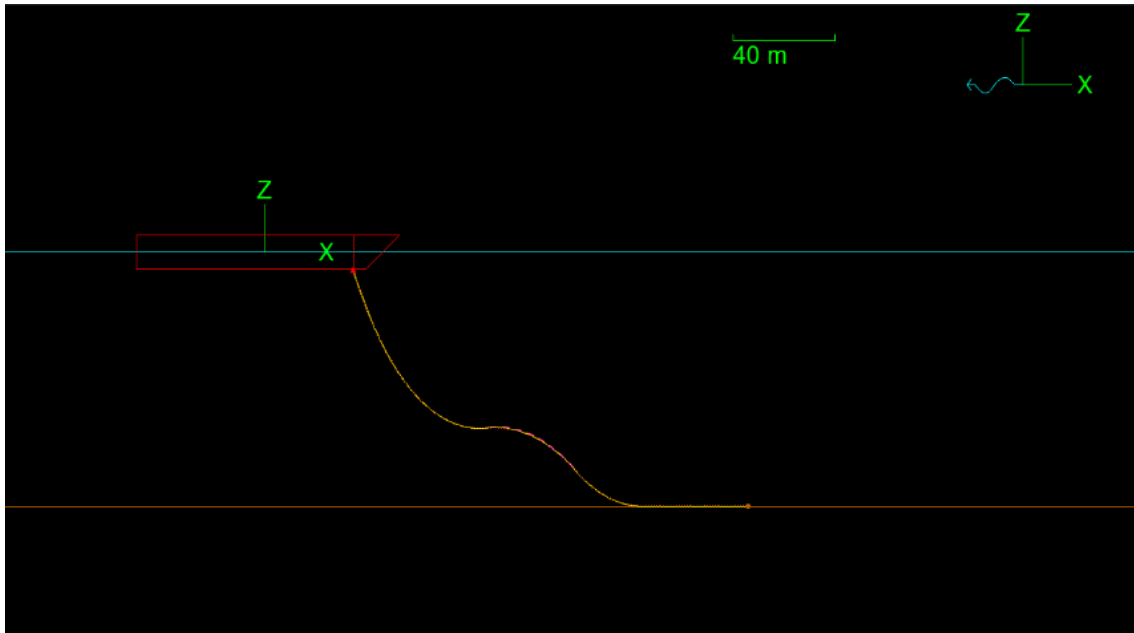


Figure 23: OrcaFlex model of the test scenario. Red object is the vessel, yellow object is the line representing the flexible product, and the pink objects attached to the line is the attachments representing the BMs. Blue line represents the water line, and the orange.

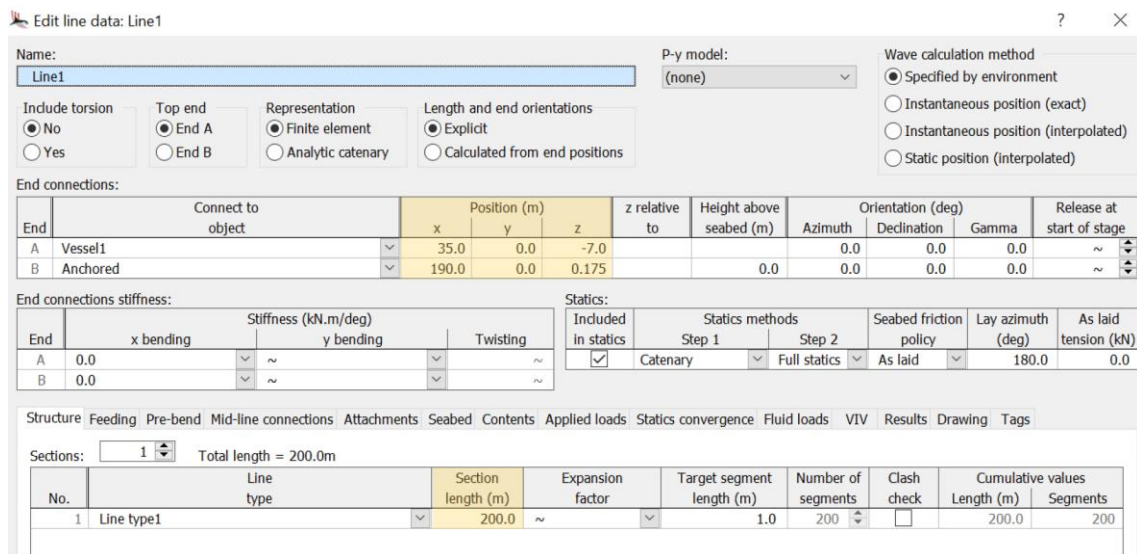


Figure 24: Object data of the line, where ‘end A’, ‘end B’, and ‘length’ is shown in the yellow boxes.

Unit Test of Unit 2:

Similar to the unit test for unit 1, the end A, end B, and length parameters varied for different test cases of the unit test of unit 2, these are detailed in Table 12 and Table 13. The results are presented in Table 14, also with comments.

Table 12: Simulated input data for unit test of unit 2 of the Analysis Program, set as key and value in JSON file, for testing control question 1-6.

Key	Value
endA	[35, 0, -7]
endB	[190, 5, -100]
length	200

Table 13: Simulated input data for unit test of unit 2 of the Analysis Program, set as key and value in JSON file, for testing control question 7.

Key	Value
endA	[35, 0, -7]
endB	[35, 5, -17]
length	200

Table 14: Control questions and result from unit test of unit 2 of the Analysis Program.

Control question	Result	Comment
1 Is the static analysis executed in OrcaFlex?	Yes	
2 Are the results accessed?	Yes	
3 Can the script access different types of results?	Yes	Effective tension, see Fig. 7 and Fig. 8 in appendix
4 Does the compare function work?	Yes	Printed OrcaFlex result and the compare result, see Compare Function in appendix 0
5 Are the results saved as a JSON?	Yes	See Figure 25
6 Is the JSON file usable in UE5?	Yes	
7 What happens if defected data from real-world is inputted?	Error in OrcaFlex	OrcaFlex provides a warning message prior to the calculation, see Error: Clump Colliding with Vessel in appendix 8.3.2.

```

1  {
2    "Array": [
3      {
4        "X": 35.0,
5        "Y": 0.0,
6        "Z": -7.0,
7        "MBR": false
8      },
9      {
10       "X": 35.29857605500895,
11       "Y": 0.013074275993232817,
12       "Z": -7.954401125803944,
13       "MBR": false
14     },
15     {
16       "X": 35.598900358400286,
17       "Y": 0.026225105619861938,
18       "Z": -8.908251375154496,
19       "MBR": false
20     },
21     {
22       "X": 35.901834408137724,
23       "Y": 0.03949021280970748,
24       "Z": -9.86127328119852,
25       "MBR": true
26     },
27     {
28       "X": 36.20782396777706,
29       "Y": 0.05288911699732206,
30       "Z": -10.813315528823171,
31       "MBR": true
32     }
33   ]
34 }

```

Figure 25: Part of the output data from unit test of unit 2 of the Analysis Program, stored in a JSON structure. X, Y, and Z value was set as float, while MBR (is the Bend Radius exceeding the MBR?) value was set as Boolean.

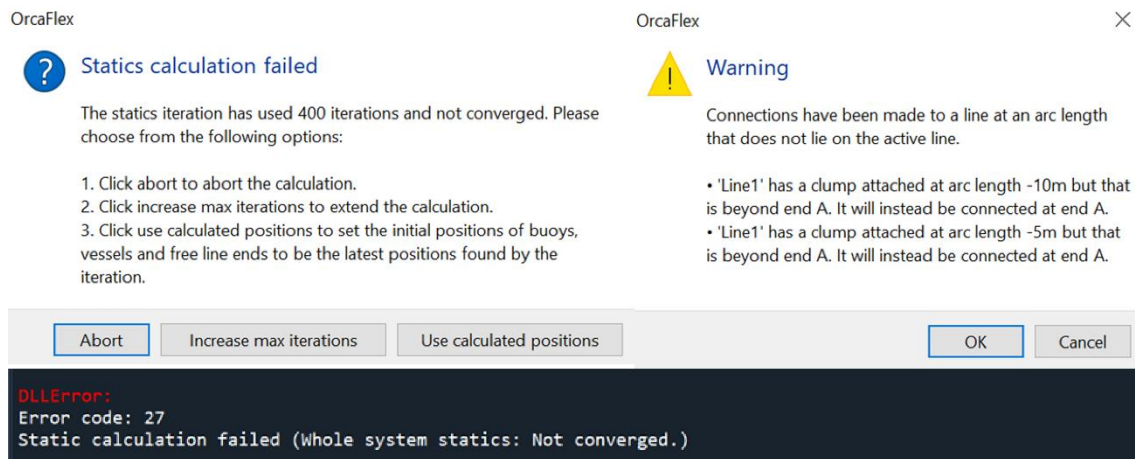


Figure 26: Error messages when static calculation won't execute.

5.1.2 Live View

A unit test of the code in the Live View module was performed. In advance of the testing, the desired file path of the input data file, the output of the Analysisprogram.py in 8.1, was added to the code. The was executed two times, one with input value 1 and input value 2 from Table 15, which generated two different JSON input data files for the Live View module.

Table 15: Input data for generating JSON file containing input data for the live view. Value 1 and value two generated two distinct scenarios.

Key	Value 1	Value 2
endA	[35, 0, -7]	[35, 0, -7]
endB	[35, 0, -57]	[190, 0, -100]
length	50	200

The results from the test are documented in Table 16. And a screenshot from the visual part of the software during the test is shown in Figure 27.

Table 16: Control questions and results with comment from the unit test of the Live View blueprint code.

Control question	Result	Comment
1 Are the results added to the array?	Yes	See Unit Test Live View in appendix 8.4
2 Is the spline created?	Yes	See Figure 27
3 Does the spline mesh have the right shape?	Yes	Cylindric shape, see Figure 27 Correct configuration according to input data file
4 Does the spline mesh have the right colour?	Yes	Green where limits are not exceeded, and red where limits are exceeded, see Figure 27
5 Does the warning message appear correctly?	Yes	Appears only once
6 Is the model updated when the input data file is updated?	Yes	Automatic update when new data in input file
7 Is the previous actor (spline) removed from the live view when a new one is created?	Yes	

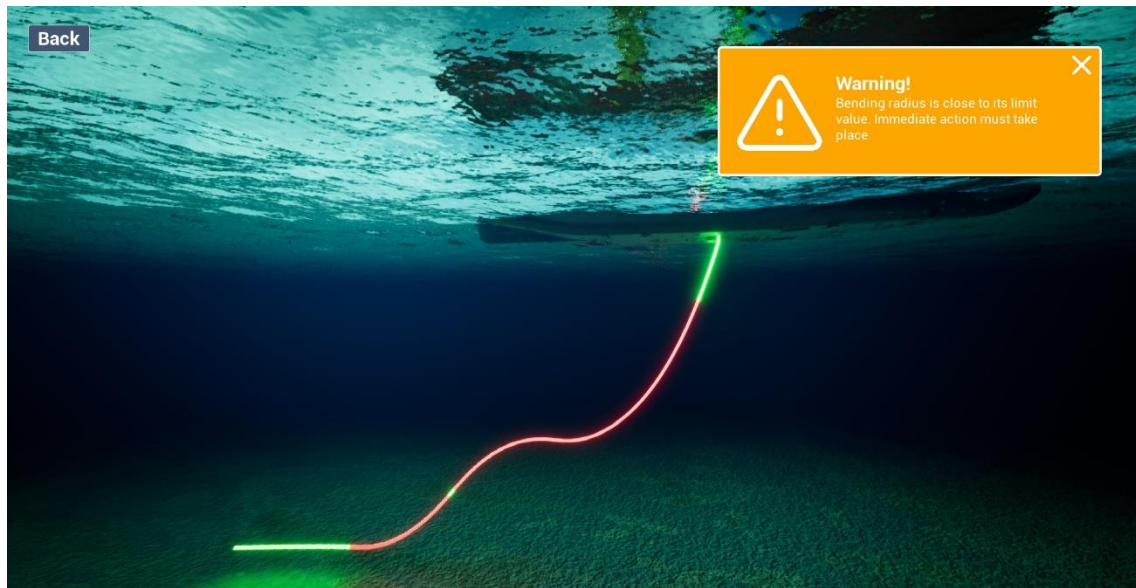


Figure 27: Live View in play mode. Red part of the cable illustrates the parts where the MBR is exceeded, while green shows the parts that is not exceeding the MBR. Message box in the top right corner with warning message telling the user to take immediate action.

5.2 Prototype Testing

During this study, a prototype of the digital twin was developed. The prototype is a software created in the game engine UE5. This software offers live view monitoring through a 3D-model updated every time there is new input data.

In advance of the prototype testing, details about the planned operation (in this case, a simulated operation) were set, which are presented in Table 17. These properties were set in the python script manually. The values for “end A”, “end B”, and “length” was accessed through a JSON file that was updated throughout the test, to represent the live real-world data.

Table 17: Properties of each object in OrcaFlex for prototype testing.

Object	Property	Value
Vessel	All	Default
	Line	
	End A connection	'Vessel1'
	End A x-, y-, and z-position	Value from key “endA” in inData.json
	End B x-, y-, and z-position	Value from key “endB” in inData.json
	Length	Value from key “length” in inData.json
	Number of sections	1
	Target segment length	[1.0]
	Attachment type	Attachments
	Number of attachments	8
	Attachment z	[110, 105, 100, 95, 90, 85, 80, 75]
	Attachment z relative to	['End B', 'End B', 'End B', 'End B', 'End B', 'End B', 'End B', 'End B']
Clump	Other	Default
	Name	“Clump type 1”
	Mass	0.48
	Volume	1.2
	Height	2
	Offset	0
	Align with	“Line axes”
	Other	Default

The prototype used the real-world data as input and created a model in OrcaFlex, which is depicted in Figure 28. The model represented the real-world scenario, which in this case was a simulated scenario where the configuration was a wave-shape.

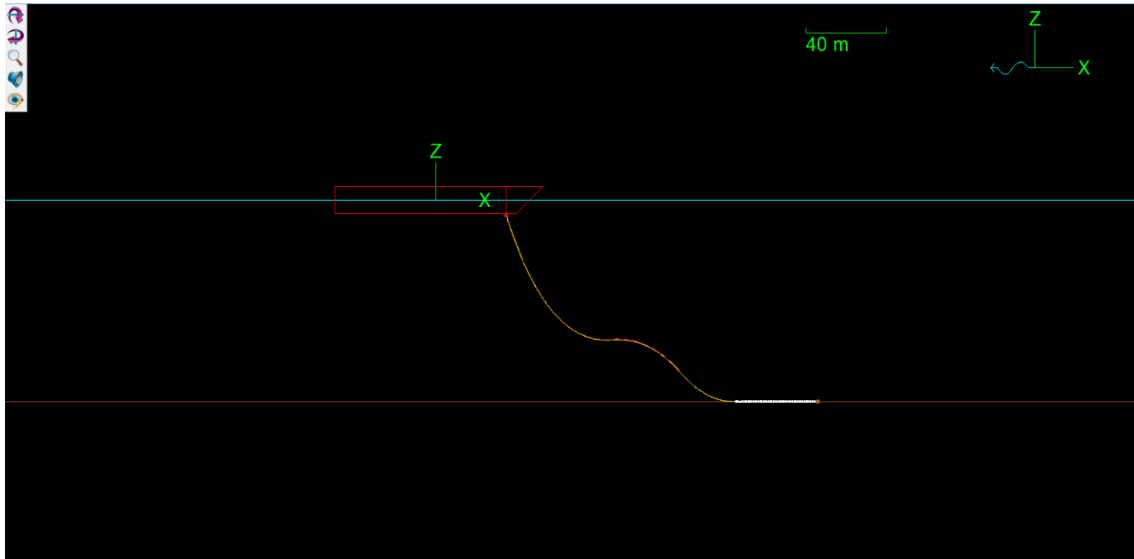


Figure 28: OrcaFlex model after the static calculation was performed.

The desired parameters to be calculated were manually entered into the script in advance of the testing. The parameters specified in the script included static results for the line object, with 'X', 'Y', 'Z', and 'Bend radius' for each node. These were set using a for-loop in the script. It was also predetermined which results should be compared with their limit values. For the prototype, the only result compared was the 'Bend radius' against the MBR, which was predefined in the script. This comparison was facilitated by a function that took the 'Bend radius' and the MBR as inputs. Additionally, the inputs for the JSON structure were manually set before the test was performed, by assigning the name of each result as the key and the value of the result as the value of the key.

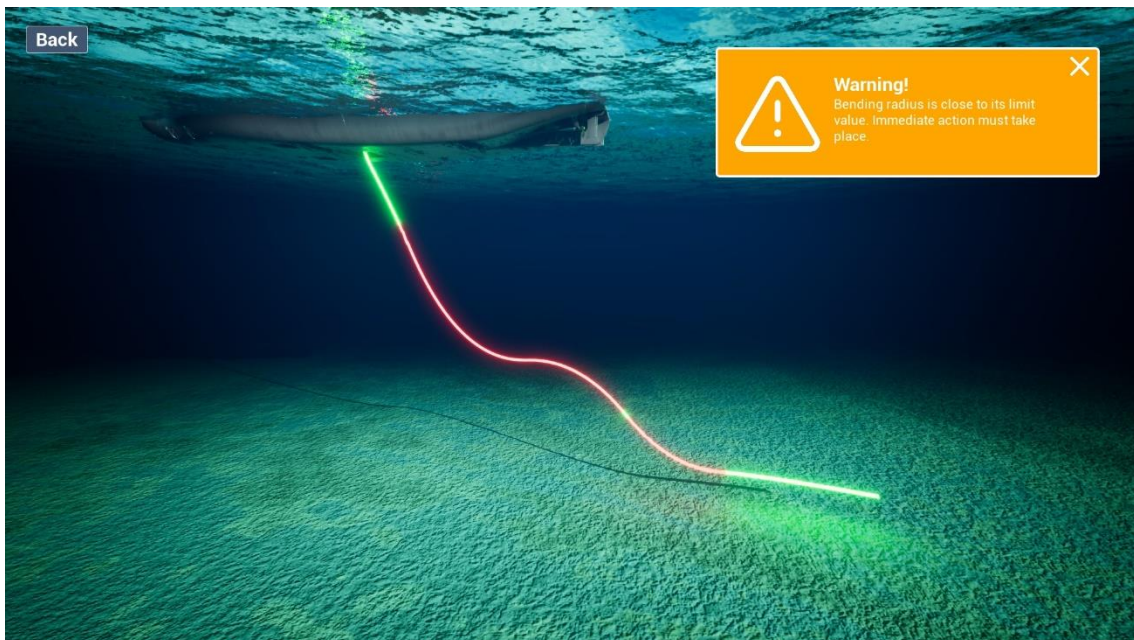


Figure 29: The prototype showing red for the nodes with Boolean set to "True", and green for the nodes with Boolean set to "False".

When the Boolean value was set to “True” the mesh of the affected nodes did change colour to red and the warning message box did appear, as shown in Figure 29. Once the message box was removed (by clicking on the “X”-button), it would not re-appear when new data was sent through the system.

The total update time of the DT system from when the real-time data is accessed was calculated to be on average 0.66 seconds, which is a very good result according to the HOQ where the target maximum update time was set to 3 seconds. However, it must be noted that this does not include the time it takes for the real-world data to be transmitted from the vessel to the DT system. The time was measured through 8 updates in the prototype, the results are presented in Table 18.

Table 18: Digital twin system update time results.

	<i>Analysis Run Time [s]</i>	<i>Live View Update Time [s]</i>	<i>Total Update Time [s]</i>
1	0.3145	0.3333	0.6478
2	0.3348	0.3334	0.6682
3	0.3085	0.3333	0.6418
4	0.3085	0.3333	0.6418
5	0.3382	0.3341	0.6723
6	0.3420	0.3333	0.6753
7	0.3407	0.3340	0.6747
8	0.3088	0.3334	0.6422
Average	0.3245	0.3335	0.6580

Total storage volume of the DT system was calculated by summing all files and components. An estimation of the real-world data volume was calculated by analysing the volume of data archived annually from the vessel today. This analysis involved accessing data via MQTT over a 10-minute period and then projecting this data consumption over a full year to estimate the total annual data storage volume. This were calculated to be 1.076 TB, making the other components irrelevant to the storage volume of the DT. The data volume of each component is presented in Table 19.

Table 19: Digital twin total storage results.

<i>Component</i>	<i>Storage Volume [GB]</i>
<i>Analysis Program</i>	3.8E-06
<i>Live View</i>	3.5
<i>Real World</i>	1101.824
<i>OrcaFlex Model</i>	0.000154
<i>Analysis Result</i>	3.15E-05
Total	1105.324

Table 20: Control questions, results, and comments for the prototype testing.

<i>Control question</i>	<i>Result</i>	<i>Comment</i>
1 Does the Analysis Program take the simulated real-world data as input?	Yes	Simulated data representing real-world data in JSON.
2 Does it set up the OrcaFlex model correctly?	Yes	Vessel, line, and clump attachments.
3 How long time did the calculation take place?	0.32 s	From real-world data is accessed to analysis results are stored in JSON.
4 How long time did the spline take place?	0.33 s	From calculation data is accessed to the live view model is viewed.
5 How much time does the entire DT system require to update?	0.66 s	From real-world data is accessed to live view is updated. Excluding real-world data gathering.
6 Are the desired result parameters extracted from the analysis?	Yes	X, Y, Z, and Bend radius for each node.
7 Are the results saved as a JSON file in the right placement?	Yes	

8	Does the Live View take the result JSON file as input?	Yes	
9	Does it create a spline based on the input file?	Yes	
10	Is the spline set to the right colour?	Yes	Red when MBR = True Green when MBR = False
11	Does the alarm system work as intended?	Yes	However, it was only tested with one decisive parameter, and message box only appears once.
12	Does the DT work autonomously?	Yes	
13	What is the total storage of the DT?	1105 GB	Including an estimation of the real-world data (annually)

5.3 Feedback

Interviews were conducted throughout the development phase to gather feedback on the project. These were in a semi-structured format, as they included prepared questions, as well as allowing for follow-up questions and comments for both parts of the interview. Table 21, below, presents the interview object, the questions that were asked, and the answers.

Table 21: Overview of interview with the work position of the interviewee, the questions, and answers.

<i>Interviewee</i>	<i>Question</i>	<i>Answer</i>
<i>Senior Project Engineer Methods DeepOcean</i>	How much time goes to the ROV movement when it is controlling the bend on the flexible product?	ROVs are used for spot checks of hog and sag, as well as a few specified points, controlling that the product is where it is supposed to according to step tables and analysis models, which are usually exact.
	Do you believe there will be time savings if the DT would replace the ROV?	I absolutely believe that using transponders could be a good solution for gather real-world data about the products geometry during installation, however, these must be installed to the product and there would be need for many transponders, which would be time assuming. This might not give much better data than the OrcaFlex models created for the different phases of the operation.
	What parameters are typically desired to monitor in OrcaFlex?	Not answered
	Do you have any other thoughts about the project/product?	Sounds like an interesting project.
	Other comments	Usually there is a static configuration going through extreme analysis of waves and currents to verify the operational weather condition limits. I believe this might be an overkill for installation of the product. However, real-time monitoring would be interesting for “in-place”, for monitoring the integrity during the product’s lifetime or during specific scenarios (such as a storm).

<i>ROV Supervisor DeepOcean</i>	How is monitoring of flexible product during installation performed today?	The product is marked with 2 and 3 lines. The lay spread shows the meters.
	How many ROVs are typically used during this operation?	Typically, one at the TDP.
	Have you experienced any problems with the method used today?	Not answered
<i>Project Engineer Inspection DeepOcean</i>	Do you believe a digital model of the operation would be beneficial?	I believe a digital model would be beneficial.
	Do you have any thoughts about the project?	A critical problem is the fact that the buoyancy modules are moving and are not entirely fixed to the prescribed position of the cable.

6 Discussion and Conclusion

This chapter discusses the outcomes of the thesis by analysing the findings derived from the project timeline. It discusses the implications of the research, evaluates the prototype developed, and formulates recommendations for future research and advancements based on the work completed.

6.1 Summary

6.1.1 Objective

The project's primary objective was to design and develop a solution for DT based monitoring to optimize subsea operations, specifically targeting the monitoring of the cable catenary shape during installation processes.

6.1.2 Proposed Solution

The proposed solution was a software tool designed to augment the installation process through real-time 3D visualization and an integrated alarm system. This system alerts users when limit values approach thresholds, thereby preventing damage to the flexible product. The design comprised four main modules:

- Real-World: Includes assets like the CLV, flexible product, data sheets, BM, ROVs, transponders, and various sensors.
- Analysis Program: Utilizes a Python script interfacing with OrcaFlex via the OrcFxAPI for data analysis.
- Live View: Developed using UE5, this module offers a real-time 3D visualization of subsea operations.
- Storage System: Accessing real-time data and storing it at the server via MQTT. Accessing stored data requires an API communication between the DT and the server.

6.1.3 What have been done?

Establishing requirements

Firstly, customer requirements were established through pre-planning, which included setting up the company's vision, defining the product vision, describing the problem to be solved, and establishing a mission statement for the product. Customer interviews were conducted to identify customer needs, which were then used to define the customer requirements. These were subsequently translated into technical requirements. The customer requirements were rated based on their importance, and the correlation between the customer requirements and the technical requirements was mapped in a HOQ matrix, as shown in Figure 13. This process prioritized the technical requirements, as presented in Figure 14. This method was chosen because it effectively creates technical requirements and assesses their relative importance. Keeping this in mind during both concept development and product development would be beneficial for creating a successful product.

Concept Development

The next step was to develop concepts. Firstly, a FAST diagram (Figure 15) was developed to set the product's functions. Then concepts for each of these functions were developed.

A concept for the analysis functions were developed, where several software were considered. Through a Pugh's model, OrcaFlex was chosen as best suited for solving the following functions:

- Estimate Shape of Product
- Calculate Integrity of Product
- Compare Results with Limit Value

OrcaFlex offered a Python API, which allowed to automate the model creation, calculation, and result accessing. This also allowed for tailored analysis and result handling, such as the compare functionality.

This concept included taking real-world data as input to create a model replicating the real-world scenario before calculations of the integrity was executed. Then the results were compared to limit values and exported to a desired file format for further usage.

Subsequently, a concept for the visualization functionality was developed. Several software were considered and compared up against each other through Pugh's model presented in Figure 16. UE5 was chosen as the preferred software. This was mainly determined by low costs linked to the development of the prototype, how easy it was to get started (this software was already widely used by the team collaborating on this thesis project, and other DT projects), high rendering qualities, and the possibility to scale the project for further development of the DT.

The concept involved using the calculation results as input to replicate the model created in the analysis software. The rationale for using the model from the analysis software to recreate it in the visualization software is that the analysis software provides the most accurate real-world simulation, while the visualization software offers superior rendering and visual aesthetic.

A concept to create an alarm system was developed to provide the "Alert User to Initiate Action" functionality. A vision for the alarm system was established before required components was described. The alarm system concepts could be further discussed and developed by user interface (UI) and user experience (UX) designers.

Three concepts were developed for the real-world data collecting functionality. Two of the concepts included adding transponders to the product to get several data points, while one concept suggested that the amount of available data from the vessel today was enough to replicate the situation. All these concepts should be further discussed and evaluated.

Accessing real-world data should be done through MQTT communication protocols, which is how it is already accessed today.

Establishing Product Architecture

The product architecture was developed by putting the concepts together from data gathering to visual representation to form a full DT model. The product was divided into four modules each serving unique functionalities to the DT model.

- The Real-World module included all real-world assets and corresponding datasheets, as well as real-time environmental data giving the immediate (within acceptable delay) state of the real-world scenario.
- The Analysis Program module provides the functionalities related to calculating the integrity of the product. This involves using the real-world data to generate result data for the live view module.
- The Live View module is responsible for providing the desired information to the user. This is done through a real-time updated 3D-model of the real-world scenario. It will also alert the user if specified limits are about to be exceeded.
- The Storage System module involves a system for storing the data collected, generated, and used throughout the DT system.

6.1.4 Prototype Development

A prototype was developed to serve as a proof of concept, incorporating:

1. Simulated Real-Time Data: Generating baseline data for analysis.
2. Analysis Program:
 - i. Model creation (vessel, line, clump).
 - ii. Data configuration and static calculation.
 - iii. Evaluation of results against predefined limits, outputting JSON formatted data.
3. Live View Implementation: Visualizing the flexible product's status and alerting through dynamic widgets based on analysis results.

6.1.5 Assessment of the Solution

The prototype effectively demonstrated the feasibility of monitoring the bending radius and overall configuration of the flexible product, with immediate alerts when critical limits were approached. This not only validates the DT approach but also enhances operational safety by providing real-time data for immediate decision making.

6.1.6 Use of OrcaFlex

OrcaFlex was instrumental in simulating offshore conditions and providing valuable data insights, underscoring its efficiency for subsea applications. The accuracy of the simulation improves with increased real-world data input, raising questions about the acceptable levels of deviation in practical scenarios. This is something that should be further evaluated through testing of the accuracy of the DT model.

Other analysis software can be used to calculate the integrity of the flexible product; however, the chosen software is industry standard.

6.1.7 The Research Journey

This thesis embarked on a mission to elevate the accuracy of monitoring flexible subsea products. Employing a DT framework, this research not only represents a technological advance but also proposes a shift in how subsea operations are monitored. The proposed solution provides real-time documentation throughout the operation, enabling continuous overview which can prevent critical situations.

6.1.8 Unit Testing

The unit test result was presented in 5.1 where all showed promising results. These results sat the baseline for the evaluation of the further development of the code parts of the product, which are now discussed.

Analysis Program

Unit 1, the analysis preparation in OrcaFlex, did create a model in OrcaFlex consisting of a vessel and a line. The position of the vessel matched with the position that was set in the script. All properties of the line object were also matching what was set in the script. Because there were no available ongoing real-world operations, it was not possible to test the accuracy of the configuration of the OrcaFlex model and the real-world. However, the setup did resemble a wave shape as intended (depicted in Figure 23), but this does not validate the actual accuracy of the configuration of the product. Therefore, the product should be tested during operation, where it would be used in addition to the traditional method, this is further described in 6.3.1.

Unit 2, the OrcaFlex analysis, performed the static analysis and accessed the desired results. When the results are accessed through the Analysis Program, they are unitless meaning that no additional preparation

are required before usage. This means that the script of the prototype can be used in the future when other result parameters are required if they are offered by OrcaFlex.

The comparison function did work as intended, as this was built as a simple if-statement returning a Boolean value based on the condition. The prototype only used this function for one of the result parameters, the bend radius, which was compared to the MBR. The system was not tested for a scenario where several parameters were set in this function, meaning that there is uncertainty related to how the alarm system should work for these scenarios. For example, should the alarm system always be activated when a Boolean is set to “True”, or should this only apply to some specified parameters? Or should the warning message also tell the user what parameter is about to be exceeded, also provided with a suggested action to prevent it from being exceeded? These are some of the questions that should be further discussed at a later stage of the product development.

When invalid data was provided to the OrcaFlex model, the calculation would not be executed. This was shown as an error message in the IDE and in OrcaFlex as shown in Figure 26 in 5.1.1. When this happened the script that was running stopped immediately, which made the DT stop updating as well. To make sure that the live update, which is the core of the solution, won't stop when this type of error occurs, there should be added some error handling system to the script. This would ensure that the script keeps running even when the calculation is not.

Live View

All control questions in the unit test of the live view were marked as “Yes” meaning that everything was working as expected. The code added the OrcaFlex results into an array in UE5 which was used to set the spline points, then there was created a spline through these points. As shown in Figure 27 in 5.1.2, the mesh was set to each spline point, which was shaped as a cylinder creating the flexible product. The mesh had both green and red colour depending on the bend radius of the node. When the input file was updated, the model did update as well, and the previous model was deleted. The warning message box did appear when the Boolean was set as “True” in the input file, however, it only appeared once. If the message box was removed by clicking on the X-button it would not reappear even if the input file did include a Boolean set as “True”.

6.1.9 Prototype Testing

The prototype testing included preparing the test scenario, starting the execution of the analysis script in the IDE, and starting the live view in UE5.

Preparation

The preparation of the test scenario was done by creating a JSON file consisting of the input data for each time interval of the simulated operation. This could have been a file containing historical data played over again. The script takes in the value of the first key of the requested name, meaning that if that key is deleted the next key would be the first one when the script is executed again. This method made the JSON file represent real-time data as it was continuously updated.

Another part of the preparation included adding objects and properties to the script, as detailed in Table 17 in 5.2. This step must be manually done for the end-product, as the model will vary for every operation. However, once scripts are made for different operations, they can be reused with just few changes. The objective should be to have a predefined list of operation to choose between, and a set of objects (where all the properties are set) that can be added to the script. This would make the prework much more effective and less time-consuming. This is something that should be considered for future development of the product.

Analysis

The prototype used simulated data as input to create a OrcaFlex model of that real-time scenario. Then it executed a static simulation on the OrcaFlex model before the desired results were accessed. The bend radius was accessed from the static line result for each node. Then it was compared to the MBR through a

comparison function that returned a Boolean value giving the state of that specific node. The Boolean was set as “True” when the bend radius was less than the MBR, while False was set when the bend radius was bigger than the MBR. The desired results were then put into a JSON file, stored at the desired location where it could be accessed for further usage.

The MBR should include a safety factor ensuring that the operators have time to act and correct the situation before any damage to the product. Ideally, the colour of the product in the DT should gradually change to red when the parameter approaches the limit value.

The accuracy of the calculation was not tested as there were no real-world data for it to be compared to. However, as mentioned in the interviews, OrcaFlex is used today to verify the operational conditions. This indicates that the data provided by OrcaFlex is accepted as reliable.

Visualization and Alarm System

Once the JSON file containing the analysis results were created or updated it was accessed by the UE5 “Spline_BP” (presented in 8.2) blueprint.

The node positions (x, y, z) were used to create the spline points, which was successful as there was a high number of nodes making the spline create the desired shape. The spline points were controlled by checking a selected number of points and control it up against the corresponding node, the result was that these were indeed set to the same number. This means that the shape of the spline is the same as the line object in OrcaFlex.

When the Boolean was set to “True” the warning message box did appear as intended. However, it would only appear once due to the “Do once” function. But if it would have been set every time the Boolean was “True” a widget would be created for each of the nodes having this Boolean value. This means that there could be hundreds of widgets on top of each other which would require more capacity and it would take a long time to click them all away. A new solution for this problem should be created.

The alarm system could be further improved by adding a blinking effect to the screen strengthening the signal. Another improvement would be to add an audio effect with an alarming sound.

Summary

The prototype did work autonomously when the properties for the operations were properly set to the script and when there were no errors occurring in the OrcaFlex calculation. When the real-world data was updated, the analysis program generated a results file used in UE5 to replicate the cable and activate the alarm system when limits were about to be exceeded. This confirmed that there is autonomous communication between the real-world, OrcaFlex, and UE5, making the DT model complete.

6.2 Conclusion

In this subchapter, the research questions are answered while summarizing the project.

6.2.1 Key Components and Architecture of a Digital Twin System for Subsea Operations

The development of a DT system for real-time monitoring of subsea operations involves several crucial components and architectural considerations. These are detailed in 4.

The system was developed with a modular architecture consisting of four key modules:

- Real-World: Capturing and transmitting data from the subsea operation.

- Analysis Program: Utilizes OrcaFlex static line calculations, accessed through Python programming, to assess the structural health of the flexible product based on real-time data.
- Live View: Creating a live view digital model of the operation.
- Storage System: Archives all data.

Each module is designed for reusability, employing Python functions that can easily be integrated into similar scripts. The objective of the end-product is to have a storage system that communicates via MQTT, a standard protocol ensuring broad compatibility. The prototype, however, was developed with a storage system which was managed locally on the same computer to simplify the file access.

6.2.2 Integrity of Digital Twin Technology into Existing Operations

DT technology can be seamlessly integrated into existing operational framework to enhance monitoring and decision-making processes. This integration, as outlined in 4 and demonstrated by the prototype results, involves:

- Backend Components:
 1. Data Acquisition: Collects real-world data.
 2. Simulation and Analysis: Performs calculations using OrcaFlex.
 3. Data Management: Stores results in a structured format for easy access and analysis.
- Frontend Components:
 1. 3D Live View: Offers real-time visualization of subsea operations.
 2. Alarm System: Alerts operators when critical limits are near, prompting immediate action.

To facilitate this integration, operational sites would require dedicated display for the live view, an OrcaFlex license, a capable analysis computer, access to vessel data, and training for personnel on software usage.

6.2.3 Advantage of Digital Twin Technology Over traditional Methods

DT technology offers substantial improvements over traditional monitoring methods by providing:

- Enhanced Insights: Deliver deeper, actionable insights that were previously unavailable, enhancing operational efficiency and safety.
- Operational Documentation: Acts as a comprehensive record of the operation, useful for planning future operations and providing verifiable documentation to the client.

6.2.4 Challenges and Limitations

While promising, the implementations of DT technology in offshore operations faces several challenges:

- Data Accuracy: A perfect replication of real-world conditions is unattainable due to the need for absolutely accurate data on every influencing factor, including environmental conditions and unforeseen activities.
- Integration Complexity: Incorporating DT technology into existing systems can be complex and requires significant changes to operational protocols and training.

These challenges can be addressed by setting realistic expectations about data accuracy and incrementally integrating DT technologies to allow for adaptation and learning.

6.2.5 Contribution to Environmental Sustainability

The DT can significantly contribute to the environmental sustainability of subsea operations by:

- Reducing Operation Time: Faster operations mean less time vessels spend at sea, thereby reducing CO2 emissions.
- Minimizing Material Waste: Improved operational accuracy reduces the likelihood of damage to the flexible product, thus decreasing material waste.

By enhancing operational efficiency and decision-making, the DT not only supports safer and more effective subsea operations but also promotes a reduced environmental footprint.

6.3 Further Work Recommendations

As concluding remarks from this study, recommendations for further work have been outlined. These recommendations outline the next steps for the development and optimization of the DT model based on the findings of this study.

6.3.1 Further Product Development

The next step in the IPD plan is Step 3, Concurrent Development, where the concept should be undergoing quality assurance before it is being properly tested in the field, this involves:

- Low-level Testing and Validation: Different engineering departments should validate specific aspects of the product to ensure it meets predefined requirements.
- Continuous Refinement: Based on feedback from initial testing, specifications and requirements should be regularly reviewed and refined to enhance product functionality and success.

Once preliminary assessments are satisfactory, the product should undergo real-world testing during lay operations, which necessitates additional resources. These tests should include:

- Comparative Testing: Testing the DT model alongside traditional methods to benchmark and evaluate the accuracy of the DT model and identify challenges and limitations with the traditional approaches.
- Extended Data Collection: Implement temporary transponders to gather more data points along the flexible product. This will facilitate a comprehensive comparison of different model configurations, enhancing the understanding of the DT's accuracy across various scenarios. When implementing transponders, it is worth mentioning that the coverage area of the transponder signals must be considered meticulously.

The outcomes of these test will be crucial for validating the DT's accuracy and readiness for operational deployment.

6.3.2 Optimizing the Solution

The effectiveness of the DT model hinges on the acquisition of real-world data. Recommendations include:

- Advanced Data Collection Methods: Develop more sophisticated monitoring techniques, particularly for critical metrics such as the TDP. Enhancing the accuracy and the reliability of this data directly improve the performance and reliability of the DT model.
- Underwater Communication Technologies: Utilize state-of-the-art communication technologies to ensure rapid and efficient data transmission.

6.3.3 Smart Buoyancy

The "Smart Buoyancy" concept, aimed primarily at maintenance operations for IACs in the floating wind industry, requires:

- In-depth Industry Research: Collaborate with the renewables department to gain a thorough understanding of industry needs and operational requirements.
- Resource Allocation: Given the higher costs and the emerging stage of the floating wind sector, substantial resources and strategic planning are needed to determine if this concept should proceed.
- Risk and Opportunity Assessment: As the floating wind sector is still emerging, developing technologies in this area involves significant risk. However, successful development could position the company as a leader in this future market.

These steps will ensure that further development of the DT technology not only meets current operational needs but also adapts to future advancements in subsea operations.

7 References

- [1] 'DeepOcean celebrates 25-year anniversary'. Accessed: May 14, 2024. [Online]. Available: <https://www.deeпоceangroup.com/articles/deeпоcean-celebrates-25-year-anniversary>
- [2] 'DeepOcean'. Accessed: Mar. 11, 2024. [Online]. Available: <https://www.deeпоceangroup.com/about/about-us>
- [3] S. Yuan, P. Major, and H. Zhang, 'Flexible riser replacement operation based on advanced virtual prototyping', *Ocean Engineering*, vol. 210, p. 107502, Aug. 2020, doi: 10.1016/j.oceaneng.2020.107502.
- [4] 'Edda Freya'. Accessed: May 14, 2024. [Online]. Available: <https://www.deeпоceangroup.com/vessels/edda-freya>
- [5] J. Herdiyanti, 'Comparisons Study of S-Lay and J-Lay Methods for Pipeline Installation in Ultra Deep Water', Universitetet i Stavanger, Thesis, 2013. [Online]. Available: <http://hdl.handle.net/11250/183162>
- [6] L. T. Jayathilake, 'Parametrization of Steel Lazy Wave Riser Configuration for Fatigue Cosiderations at the Touch Down Point', Universitetet i Stavanger, Thesis, 2022. Accessed: Jan. 09, 2024. [Online]. Available: <https://hdl.handle.net/11250/3032568>
- [7] Y. Bai, *Pipelines and risers*, 1st ed. in Elsevier ocean engineering book series, no. 3. Amsterdam: Elsevier, 2001.
- [8] Y. Bai and Q. Bai, *Subsea engineering handbook*, Second edition. Cambride, MA: Gulf Professional Publishing, 2019.
- [9] V. A. Mamatsopoulos, C. Michailides, and E. E. Theotokoglou, 'An Analysis Tool for the Installation of Submarine Cables in an S-Lay Configuration Including "In and Out of Water" Cable Segments', *JMSE*, vol. 8, no. 1, p. 48, Jan. 2020, doi: 10.3390/jmse8010048.
- [10] CRP Subsea, 'Products: Buoyancy Clamps', Products. Accessed: Jan. 31, 2024. [Online]. Available: <https://www.crpsubsea.com/products/product-families/buoyancy-floats/distributed-buoyancy/buoyancy-clamps/>
- [11] Raket Research, 'Buoyancy Clamp Market Research Report Unlocks Analysis on the Market Financial Status, Market Size, and Market Revenue upto 2030', Oct. 2023, Accessed: Jan. 31, 2024. [Online]. Available: <https://www.linkedin.com/pulse/buoyancy-clamp-market-research-report-unlocks-analysis/>
- [12] J. Hill, S. Laycock, S. Chai, C. Balash, and H. Morand, 'Hydrodynamic loads and response of a Mid Water Arch structure', *Ocean Engineering*, vol. 83, pp. 76–86, Jun. 2014, doi: 10.1016/j.oceaneng.2014.02.032.
- [13] Cortland, 'Tethers', Cortland. Accessed: Mar. 31, 2024. [Online]. Available: <https://www.cortlandcompany.com/tethers/>
- [14] *Design of Marine Risers with Functionally Graded Materials*. Elsevier, 2021. doi: 10.1016/C2020-0-00360-9.
- [15] Den Norske Veritas, 'DNV-SE-0476 Offshore rise systems'. [Online]. Available: <https://www.dnv.com/oilgas/download/dnv-se-0476-offshore-riser-systems.html>
- [16] BVG Associates, 'Guide to a Floating Offshore Wind Farm', BVG Associates, 2023. [Online]. Available: <https://guidetofloatingoffshorewind.com>
- [17] 'Utsira Nord Floating Wind Farm - Optimalisation of marine operatijons related to inter-array cable installation', Universitetet i Stavanger, Thesis, 2023. [Online]. Available: <https://hdl.handle.net/11250/3089217>
- [18] M. U. T. Rentschler, F. Adam, P. Chainho, K. Krügel, and P. C. Vicente, 'Parametric study of dynamic inter-array cable systems for floating offshore wind turbines', *Mar Syst Ocean Technol*, vol. 15, no. 1, pp. 16–25, Mar. 2020, doi: 10.1007/s40868-020-00071-7.
- [19] 'Kostnader for kraftproduksjon - NVE'. Accessed: Apr. 08, 2024. [Online]. Available: <https://www.nve.no/energi/analyser-og-statistikk/kostnader-for-kraftproduksjon/>

- [20] J. O. G. Tande, 'Six factors that will determine the cost of offshore wind power', Six factors that will determine the cost of offshore wind power. Accessed: Feb. 02, 2024. [Online]. Available: <https://www.sintef.no/en/latest-news/2021/six-factors-that-will-determine-the-cost-of-offshore-wind-power/>
- [21] DeepOcean Employee, 'Utilizing ROVs to Monitor Integrity', Feb. 13, 2024.
- [22] '04-cables-accessories.pdf'. Accessed: Apr. 16, 2024. [Online]. Available: <https://www.offshorewindscotland.org.uk/media/g3mbquou/04-cables-accessories.pdf>
- [23] 'A Complete Guide to Fibre Optic Cables | RS'. Accessed: Apr. 16, 2024. [Online]. Available: <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/fibre-optic-cables-guide>
- [24] M. Paalvast, 'Cable Lay Engineering', Cable Lay Engineering. Accessed: Jan. 30, 2024. [Online]. Available: <https://www.linkedin.com/pulse/cable-lay-engineering-mark-paalvast/>
- [25] HydroCharting NG, 'Six Types of Ship motion Explained!', Nov. 2022, Accessed: Jan. 30, 2024. [Online]. Available: <https://www.linkedin.com/pulse/six-types-ship-motion-explained-hydrocharting-ng/>
- [26] R. Stewart, 'Waves and the Concept of a Wave Spectrum', Waves and the Concept of a Wave Spectrum. Accessed: Jan. 19, 2024. [Online]. Available: https://wikiwaves.org/Waves_and_the_Concept_of_a_Wave_Spectrum
- [27] T. Worzyk, *Submarine Power Cables: Design, Installation, Repair, Environmental Aspects*. in Power Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-01270-9.
- [28] M. F. Opgåtd, 'Torsion Instability of Dynamic Cables During Installation', Norwegian University of Science and Technology, Thesis, 2017. Accessed: Jan. 19, 2024. [Online]. Available: <http://hdl.handle.net/11250/2452268>
- [29] K. A. Johannesen, 'Digital Twin', Accessed: Jan. 19, 2024. [Online]. Available: <https://sintef.no/en/expertise/digital/applied-mathematics/digital-twin/>
- [30] S. Jones, 'What Is a Digital Twin? How Intelligent Data Models Can Shape the Build World', What Is a Digital Twin? How Intelligent Data Models Can Shape the Build World. Accessed: Jan. 31, 2024. [Online]. Available: <https://www.autodesk.com/design-make/articles/what-is-a-digital-twin>
- [31] B.-Q. Chen, P. M. Videiro, and C. Guedes Soares, 'Opportunities and Challenges to Develop Digital Twins for Subsea Pipelines', *JMSE*, vol. 10, no. 6, p. 739, May 2022, doi: 10.3390/jmse10060739.
- [32] A. Rasheed, O. San, and T. Kvamsdal, 'Digital Twin: Values, Challenges and Enablers From a Modeling Perspective', *IEEE Access*, vol. 8, pp. 21980–22012, 2020, doi: 10.1109/ACCESS.2020.2970143.
- [33] European Union, 'Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons With Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing'. 2016. [Online]. Available: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>
- [34] 'Tingenes internett (IoT) – teknologi for et bedre samfunn', SINTEF. Accessed: Apr. 12, 2024. [Online]. Available: <https://www.sintef.no/fagomrader/tingenes-internett/>
- [35] C. M. G. Gussen, P. S. R. Diniz, M. L. R. Campos, W. A. Martins, F. M. Costa, and J. N. Gois, 'A Survey of Underwater Wireless Communication Technologies', *JCIS*, vol. 31, no. 1, pp. 242–255, 2016, doi: 10.14209/jcis.2016.22.
- [36] H. Zhao *et al.*, 'Underwater wireless communication via TENG-generated Maxwell's displacement current', *Nat Commun*, vol. 13, no. 1, p. 3325, Jun. 2022, doi: 10.1038/s41467-022-31042-8.
- [37] G. Schirripa Spagnolo, L. Cozzella, and F. Leccese, 'Underwater Optical Wireless Communications: Overview', *Sensors*, vol. 20, no. 8, p. 2261, Apr. 2020, doi: 10.3390/s20082261.
- [38] M. Kong *et al.*, 'Toward Automatic Subsea Operations Using Real-Time Underwater Optical Wireless Sensor Networks', *IEEE Photonics J.*, vol. 14, no. 1, pp. 1–8, Feb. 2022, doi: 10.1109/JPHOT.2021.3136922.
- [39] Meriam Webster, 'Transponder', Transponder. Accessed: Jan. 31, 2024. [Online]. Available: <https://www.merriam-webster.com/dictionary/transponder>
- [40] 'Fetch AZA', Sonardyne. Accessed: Mar. 31, 2024. [Online]. Available: <https://www.sonardyne.com/products/fetch-aza/>
- [41] '372212_cnode_maxi_battery_lifetime_estimation_tech_note.pdf'. Accessed: Mar. 31, 2024. [Online]. Available: https://www.kongsberg.com/globalassets/maritime/km-products/product-documents/372212_cnode_maxi_battery_lifetime_estimation_tech_note.pdf
- [42] 'MQTT - The Standard for IoT Messaging'. Accessed: Apr. 16, 2024. [Online]. Available: <https://mqtt.org/>
- [43] 'What is MQTT and How Does it Work?', IoT Agenda. Accessed: Apr. 16, 2024. [Online]. Available: <https://www.techtarget.com/iotagenda/definition/MQTT-MQ-Telemetry-Transport>

- [44] C. Stover and E. W. Weisstein, 'Line'. Accessed: Mar. 07, 2024. [Online]. Available: <https://mathworld.wolfram.com/>
- [45] E. W. Weisstein, 'Circle'. Accessed: Mar. 07, 2024. [Online]. Available: <https://mathworld.wolfram.com/>
- [46] 'OrcaFlex Help'. Accessed: Feb. 27, 2024. [Online]. Available: <https://www.orcina.com/webhelp/OrcaFlex/>
- [47] 'Line data: Statics'. Accessed: May 07, 2024. [Online]. Available: <https://www.orcina.com/webhelp/OrcaFlex/Content/html/Linedata,Statics.htm#LineStaticsStep1Method>
- [48] 'Line statics: Catenary'. Accessed: May 07, 2024. [Online]. Available: <https://www.orcina.com/webhelp/OrcaFlex/Content/html/Linestatics,Catenary.htm>
- [49] E. W. Weisstein, 'Catenary'. Accessed: May 03, 2024. [Online]. Available: <https://mathworld.wolfram.com/>
- [50] 'ESplineType'. Accessed: Mar. 13, 2024. [Online]. Available: <https://docs.unrealengine.com/5.3/en-US/API/Plugins/ControlRigSpline/ESplineType/>
- [51] 'What is API: Meaning, Types, Examples', AltexSoft. Accessed: Feb. 21, 2024. [Online]. Available: <https://www.altexsoft.com/blog/what-is-api-definition-types-specifications-documentation/>
- [52] 'OrcFxAPI Help'. Accessed: Feb. 21, 2024. [Online]. Available: <https://www.orcina.com/webhelp/OrcFxAPI/>
- [53] 'Python', About. Accessed: Feb. 13, 2024. [Online]. Available: <https://www.python.org/about/>
- [54] R. Scarlett, 'Why Python keeps growing, explained', Github blog. Accessed: Feb. 13, 2024. [Online]. Available: <https://github.blog/2023-03-02-why-python-keeps-growing-explained/>
- [55] 'The Python Tutorial', Python documentation. Accessed: Mar. 07, 2024. [Online]. Available: <https://docs.python.org/3/tutorial/index.html>
- [56] '6. Modules', Python documentation. Accessed: Mar. 05, 2024. [Online]. Available: <https://docs.python.org/3/tutorial/modules.html>
- [57] J. Sundnes, *Introduction to Scientific Programming with Python*. Cham: Springer International Publishing, 2020. Accessed: Mar. 04, 2024. [Online]. Available: <http://link.springer.com/10.1007/978-3-030-50356-7>
- [58] '9. Classes', Python documentation. Accessed: Mar. 06, 2024. [Online]. Available: <https://docs.python.org/3/tutorial/classes.html>
- [59] 'Glossary', Python documentation. Accessed: Mar. 06, 2024. [Online]. Available: <https://docs.python.org/3/glossary.html>
- [60] 'Python Functions', GeeksforGeeks. Accessed: Mar. 06, 2024. [Online]. Available: <https://www.geeksforgeeks.org/python-functions/>
- [61] Unreal Engine, 'Introduction to Blueprints'. Accessed: Feb. 13, 2024. [Online]. Available: <https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/>
- [62] Epic Online Learning, 'Comparing Blueprints and C++ Use Cases'. Accessed: Feb. 13, 2024. [Online]. Available: <https://dev.epicgames.com/community/learning/tutorials/qM2K/unreal-engine-comparing-blueprints-and-c-use-cases>
- [63] Wikipedia, 'C++', Wkipedia. [Online]. Available: <https://en.wikipedia.org/wiki/C%2B%2B>
- [64] Unreal Engine, 'Programming and Scripting'. Accessed: Feb. 13, 2024. [Online]. Available: <https://docs.unrealengine.com/5.0/en-US/unreal-engine-programming-and-scripting/>
- [65] Lentreo, 'Types of System Architectures', Medium. Accessed: Mar. 12, 2024. [Online]. Available: <https://medium.com/@abhijitgunjal1648/types-of-system-architectures-eb37dd496971>
- [66] 'Client-server architecture | Definition, Characteristics, & Advantages | Britannica'. Accessed: Mar. 12, 2024. [Online]. Available: <https://www.britannica.com/technology/client-server-architecture>
- [67] 'What are microservices?', microservices.io. Accessed: Mar. 12, 2024. [Online]. Available: <http://microservices.io/index.html>
- [68] Y. Duan *et al.*, 'Layered Architecture'. [Online]. Available: https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/Layered.pdf
- [69] 'Layered Architecture Used in Software Development', DEV Community. Accessed: Apr. 02, 2024. [Online]. Available: <https://dev.to/sardarmudassaralikhani/layered-architecture-used-in-software-development-8jd>
- [70] 'DNV-RP-A204 Assurance of digital twins'. Accessed: Apr. 12, 2024. [Online]. Available: <https://www.dnv.com/oilgas/download/dnv-rp-a204-assurance-of-digital-twins/>
- [71] 'TechnipFMCs subsea digital twin - TechnipFMC plc', Technip FMC. Accessed: Apr. 12, 2024. [Online]. Available: <https://www.technipfmc.com/en/media/news/2021/07/how-technipfmc-subsea-digital-twin-empowers-a-robust-digital-ecosystem/>

- [72] T. R. Wanasinghe *et al.*, ‘Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges’, *IEEE Access*, vol. 8, pp. 104175–104197, 2020, doi: 10.1109/ACCESS.2020.2998723.
- [73] ‘Spotlight: Digital Twin Technology for Subsea Maintenance’, National Subsea Centre. Accessed: Apr. 10, 2024. [Online]. Available: <https://www.nationalsubseacentre.com/news-events/news/2023/november/spotlight-digital-twin-technology-for-subsea-maintenance/>
- [74] ‘Seaflo Consultancy|Subsea Services & Solutions’. Accessed: Apr. 12, 2024. [Online]. Available: <https://seafloconsultancy.com/>
- [75] ‘National Subsea Centre and SeaFlo Partner to Deliver Asset Digital Twin Lab’, National Subsea Centre. Accessed: Apr. 12, 2024. [Online]. Available: <https://www.nationalsubseacentre.com/news-events/news/2023/october/national-subsea-centre-seaflo-partner-asset-digital-twin-lab/>
- [76] ‘Digital Twin Technology – Cost-efficient Solutions’, 4subsea. Accessed: Apr. 12, 2024. [Online]. Available: <https://www.4subsea.com/digital-twin-technology-article/>
- [77] Z. Wang, M. Li, Z. Liu, M. Dezhkam, Y. Zhao, and Y. Hu, ‘Continuous Dynamic Analysis Method and Case Verification of Cable Structure Based on Digital Twin’, *Sustainability*, vol. 15, no. 22, Art. no. 22, Jan. 2023, doi: 10.3390/su152216125.
- [78] D. Fergestad and S. A. Løtveit, *Handbook on Design and Operation of Flexible Pipes*. 2017. Accessed: Jan. 30, 2024. [Online]. Available: https://www.4subsea.com/wp-content/uploads/2017/07/Handbook-2017_Flexible-pipes_4Subsea-SINTEF-NTNU_lo-res.pdf
- [79] S. Grønmo, ‘kvalitativ metode’, *Store norske leksikon*. Aug. 22, 2023. Accessed: Mar. 25, 2024. [Online]. Available: https://snl.no/kvalitativ_metode
- [80] P. Bhandari, ‘What Is Qualitative Research? | Methods & Examples’, Scribbr. Accessed: Mar. 25, 2024. [Online]. Available: <https://www.scribbr.com/methodology/qualitative-research/>
- [81] T. George, ‘Types of Interviews in Research’. Accessed: Feb. 13, 2024. [Online]. Available: <https://www.scribbr.com/methodology/interviews-research/>
- [82] OrcaFlex, ‘OrcaFlex’. Accessed: Jan. 31, 2024. [Online]. Available: <https://www.orcina.com/orcaflex/>
- [83] Unreal Engine, ‘Unreal Engine 5’, Unreal Engine 5. Accessed: Jan. 31, 2024. [Online]. Available: <https://www.unrealengine.com/en-US/unreal-engine-5>
- [84] U. of S. Valley, ‘What is a Game Engine?’, University of Silicon Valley. Accessed: Mar. 13, 2024. [Online]. Available: <https://usv.edu/blog/what-is-a-game-engine/>
- [85] bizmarquee, ‘The Psychology of Color.’, Web Design Company. Accessed: Mar. 25, 2024. [Online]. Available: <https://www.bizmarquee.com/the-psychology-of-color/>
- [86] Campos, ‘The Importance of Product Testing in Product Development’, The Importance of Product Testing in Product Development. Accessed: Apr. 05, 2024. [Online]. Available: <https://www.campos.com/resources/product-development-product-testing>
- [87] ‘Unit Testing - Software Testing’, GeeksforGeeks. Accessed: May 01, 2024. [Online]. Available: <https://www.geeksforgeeks.org/unit-testing-software-testing/>
- [88] ‘Kunstig intelligens ved REALTEK | NMBU’. Accessed: Apr. 19, 2024. [Online]. Available: <https://www.nmbu.no/fakulteter/fakultet-realfag-og-teknologi/kunstig-intelligens-ved-realtek>
- [89] DeepOcean, ‘Touch Down Monitoring’, 2022.
- [90] *DigitalHose*, (2022).
- [91] ‘HSEQS’. Accessed: Apr. 11, 2024. [Online]. Available: <https://www.deeipoceangroup.com//hseqs>
- [92] ‘ESG’. Accessed: Apr. 11, 2024. [Online]. Available: <https://www.deeipoceangroup.com//about/esg>
- [93] K. T. Ulrich and S. D. Eppinger, *Product design and development*, Sixth edition. New York, NY: McGraw-Hill Education, 2016.
- [94] J. E. Bartolomei and T. Miller, ‘Functional Analysis Systems Technique (F.A.S.T.) as a Group Knowledge Elicitation Method for Model Building’.
- [95] ‘Floating Offshore Wind’. Accessed: Mar. 31, 2024. [Online]. Available: <https://www.dnv.com/focus-areas/floating-offshore-wind/>
- [96] ‘What Is Data Management and Why Is It Important?’, Data Management. Accessed: Apr. 06, 2024. [Online]. Available: <https://www.techtarget.com/searchdatamanagement/definition/data-management>
- [97] ‘What is a Transponder? | Definition from TechTarget’, IoT Agenda. Accessed: Apr. 04, 2024. [Online]. Available: <https://www.techtarget.com/iotagenda/definition/transponder>
- [98] *Introduction to OrcaFlex 10: adding line attachments*, (Oct. 14, 2022). Accessed: May 13, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=sjwIg_nPnBM

8 Appendix

Fig. 1: The first part out of 3 of the blueprint ‘Spline_BP’	66
Fig. 2: The second part out of 3 of the blueprint ‘Spline_BP’	66
Fig. 3: The third part out of 3 of the blueprint ‘Spline_BP’	67
Fig. 4: Warning message box in OrcaFlex from unit test 2 of control question7.	68
Fig. 5: Warning message from OrcaFlex shown in the IDE (Spyder).	68
Fig. 6: The clump attachment (pink line) collides with the vessel (red object) in OrcaFlex.	69
Fig. 7: The initial warning message in OrcaFlex providing information about the colliding objects.	69
Fig. 8: <i>The message in OrcaFlex informing about the failed calculation.</i>	70
Fig. 9: <i>OrcaFlex informing about the failed calculation through the IDE by providing the error message and stopping the execution of the script.</i>	70
Fig. 10: Static results in OrcaFlex.....	71
Fig. 11: The node array in UE5 when the spline points are created.	73
Fig. 12: The line representing the flexible product during installation visualised in UE5.	74
Fig. 13: The line representing the flexible product during installation visualised in UE5.	74
Fig. 14: The line representing the flexible product during installation visualised in UE5. The product reached touch down.	75
Fig. 15: The line representing the flexible product during installation visualised in UE5. The flexible product is starting to create a wave-shaped configuration.....	75
Fig. 16: The line representing the flexible product during installation visualised in UE5. The flexible product with a wave-shaped configuration.....	76

8.1 AnalysisProgram.py

The following presents the Python script performing the OrcaFlex analysis.

```

import sys
sys.path.append(r'C:\Program Files (x86)\Orcina\OrcaFlex\11.3\OrcFxAPI\Python')
import OrcFxAPI
import JSON
import time

start = time.time()

model = OrcFxAPI.Model()
vessel = model.CreateObject(OrcFxAPI.ObjectType.Vessel)
vesselPos = []
TDP = []      # This is the first TDP (or end B of the line)
length = []

filename = 'C:\\Users\\TuvaCareySelsaas\\OneDrive - DeepOcean
Group\\Documents\\DigitalTwin\\inData.json'

depth = 100

with open(filename, 'r') as file:
    data = JSON.load(file)
    Array = data["Array"]
    data = Array[0]
    vesselPos = data['endA']
    TDP = data['endB']
    length = data['length']

    if length[0] <= depth:
        TDP[2] = -float(length[0]+7)
    else:
        TDP[2] = -depth

line = model.CreateObject(OrcFxAPI.ObjectType.Line)
line.EndAConnection = 'Vessel1'
line.EndAX, line.EndAY, line.EndAZ = vesselPos
line.EndBX, line.EndBY, line.EndBZ = TDP

lineType = model['Line type1']

if length[0] >= depth:
    line.EndBConnection = 'Anchored'
    line.EndBHeightAboveSeabed = 0
else:
    line.EndBConnection = 'Free'

line.NumberOfSections = 1
line.Length = length
line.TargetSegmentLength = [1.0]

MBR = 50

clump = model.CreateObject(OrcFxAPI.ObjectType.ClumpType)
clump.Name = "Clump type 1"
clump.Mass = 0.48

```

```

clump.Volume = 1.2
clump.Height = 2
clump.Offset = 0
clump.AlignWith = "Line axes"

Attachments = list(line.AttachmentType)
Attachments.append(f"{clump}")
line.AttachmentType = Attachments

line.NumberOfAttachments = 8
line.Attachmentz = [110,105,100,95,90,85,80,75]
line.AttachmentzRelativeTo = ['End B','End B','End B','End B','End B','End B','End B','End B']

file = 'C:\Code\Python\OrcFxAPI\Test06052024.dat'
model.SaveData(file)

model.CalculateStatics()
#model.RunSimulation()

for i, j in zip(line.Length, line.TargetSegmentLength):
    number = i * j

a = int(number+1)

nodeList = [i for i in range(1,a,1)]

nodeData = []

def compare(limit, result):
    if result <= limit:
        result = True
    else:
        result = False
    return result

for nodes in nodeList:
    nodes = OrcFxAPI.oeNodeNum(nodes)
    x = line.StaticResult('X', nodes)
    y = line.StaticResult('Y', nodes)
    z = line.StaticResult('Z', nodes)
    br = line.StaticResult('Bend radius', nodes)

    oneClamp = { }
    oneClamp['X'] = x
    oneClamp['Y'] = y
    oneClamp['Z'] = z

    br = compare(MBR,br)

    oneClamp['MBR'] = br

    nodeData.append(oneClamp)

array = { }
array['Array'] = nodeData

path = 'C:\\Users\\TuvaCareySelsaas\\OneDrive - DeepOcean
Group\\Documents\\DigitalTwin\\AnalysisResultsTEST06052024.json'

```

```
nodePos = JSON.dumps(array, indent=4)
with open(path, 'w') as file:
    file.write(nodePos)

#print("\AnalysisRestuls.json is updated\n\n")

print("\nUpdated\n\n")

end = time.time()

duration = end-start

print(duration)
time.sleep(2)
exec(open("AnalysisProgramV2.py").read())
```

8.2 Spline_BP

The following figures presents the ‘Spline_BP’ which is the blueprint in UE5 that creates the live view of the flexible product. The resolution of these figures are poor, but they shows the concept of how a blueprint is built up.

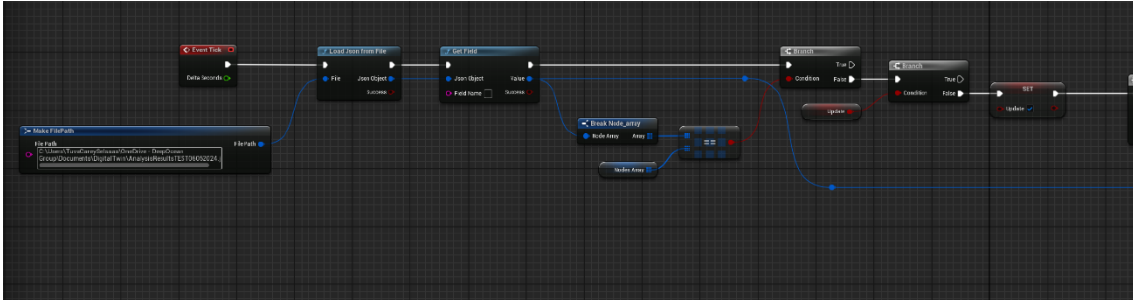


Fig. 1: The first part out of 3 of the blueprint 'Spline_BP'.

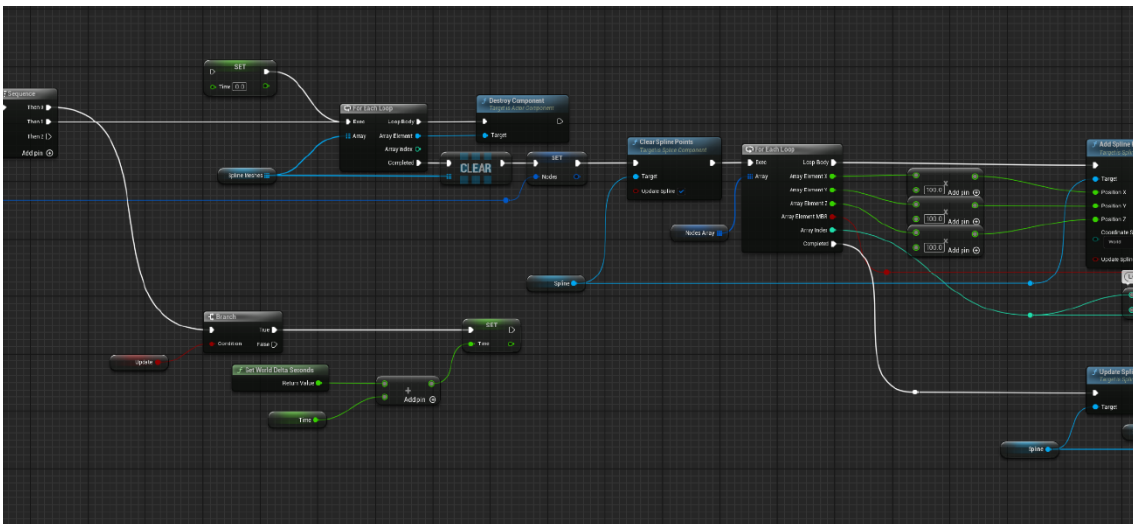


Fig. 2: The second part out of 3 of the blueprint 'Spline_BP'.

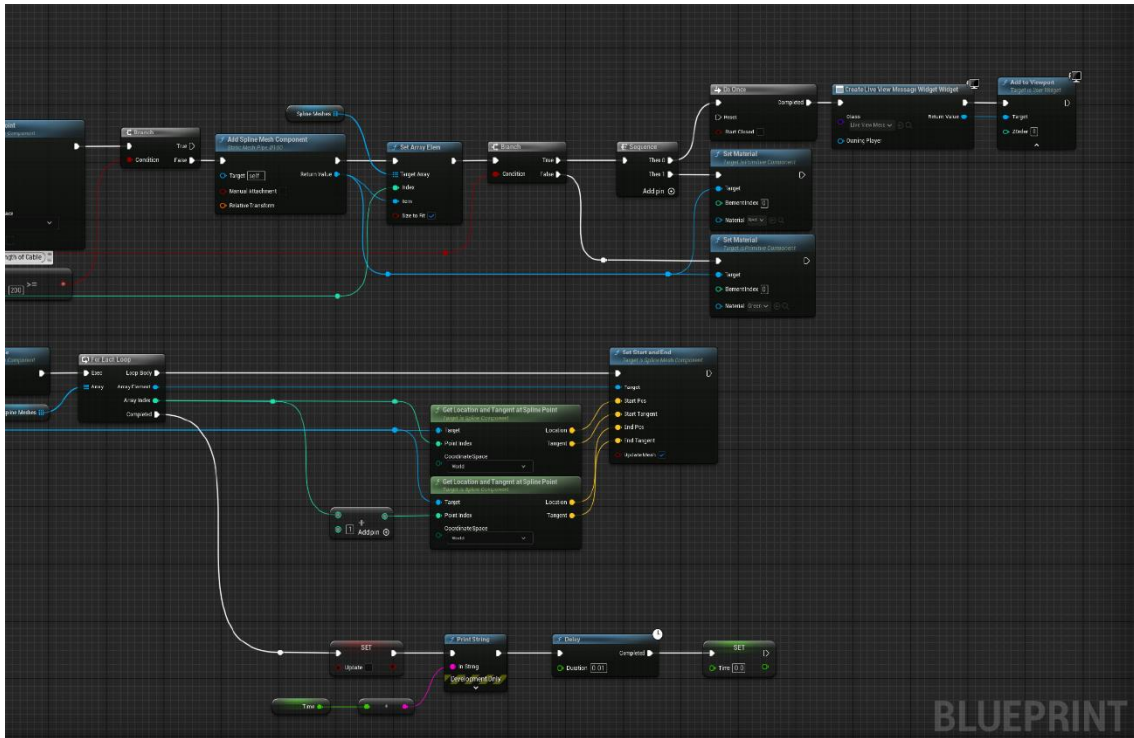


Fig. 3: The third part out of 3 of the blueprint 'Spline_BP'.

8.3 Unit Test

8.3.1 Unit Test 2

Fig. 4 and Fig. 5, demonstrate the warning message as a result of unit test 2 of control question 7.

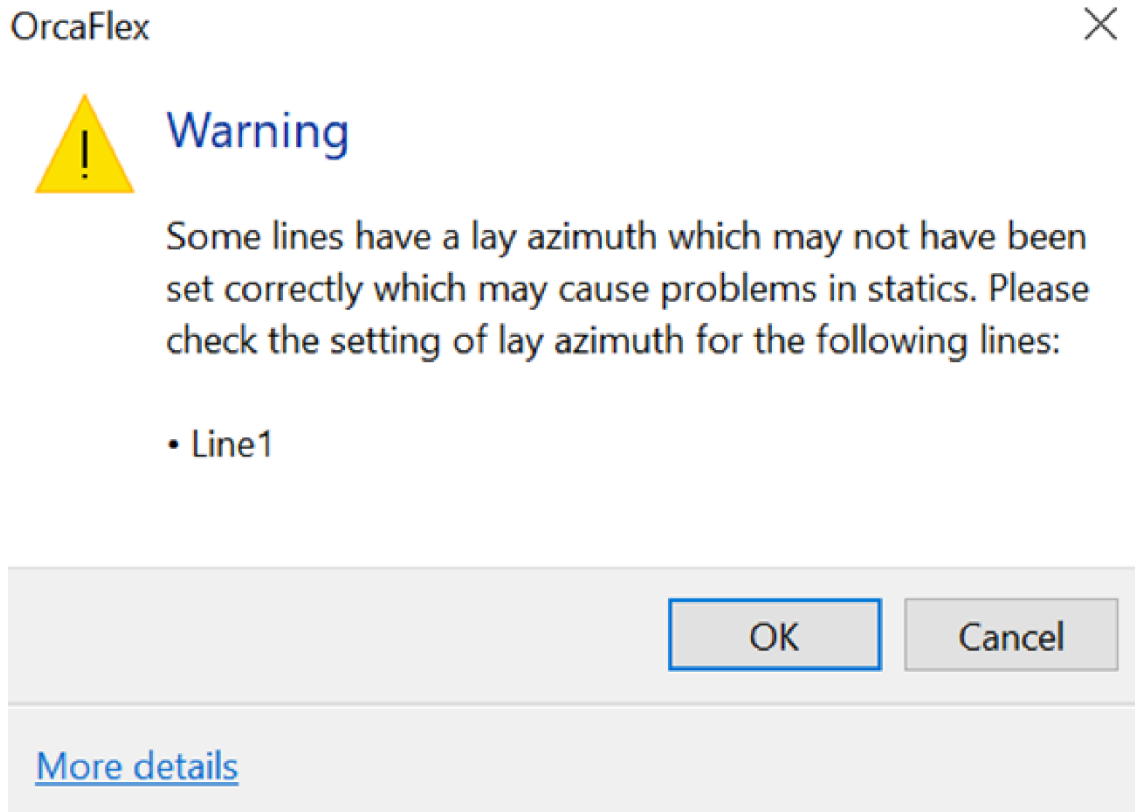


Fig. 4: Warning message box in OrcaFlex from unit test 2 of control question 7.

```
File C:\Program Files (x86)\Orcina\OrcaFlex\11.3\OrcFxAPI\Python\OrcFxAPI.py:1226 in _checkStatus
raise DLLError(StatusCode(status.value))
DLLError:
Error code: 33
Invalid data access (Length: Must be positive)
```

Fig. 5: Warning message from OrcaFlex shown in the IDE (Spyder).

8.3.2 Error: Clump Colliding with Vessel

The clump attachment collides with the vessel object (depicted in Fig. 6) which makes the calculation fail. This makes the Analysis Program script stop running as shown in Fig. 9. In OrcaFlex this is first alerted by the message shown in Fig. 7, then when the static calculation is tried to run a new message box appear which is depicted in Fig. 8.

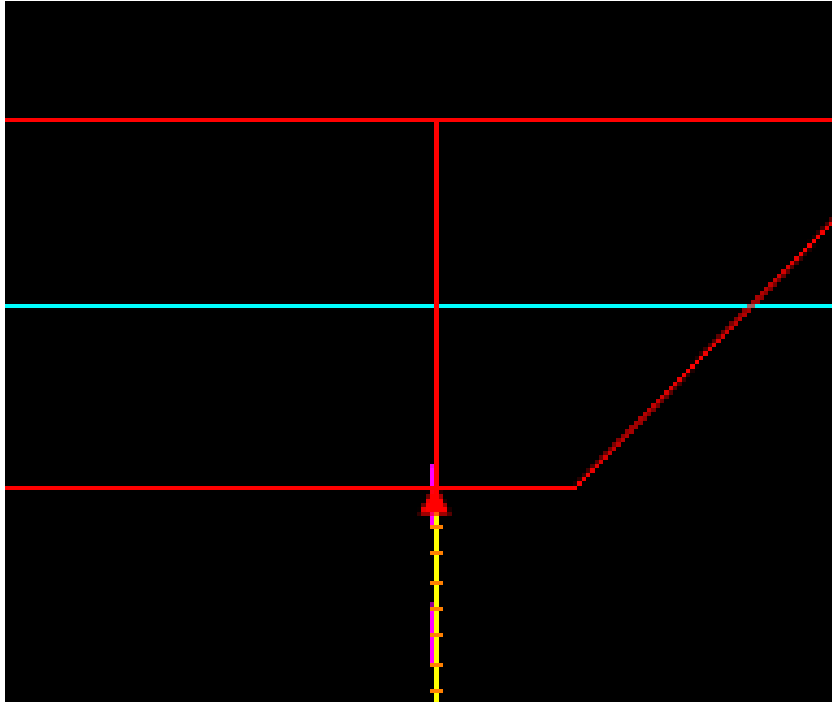


Fig. 6: The clump attachment (pink line) collides with the vessel (red object) in OrcaFlex.

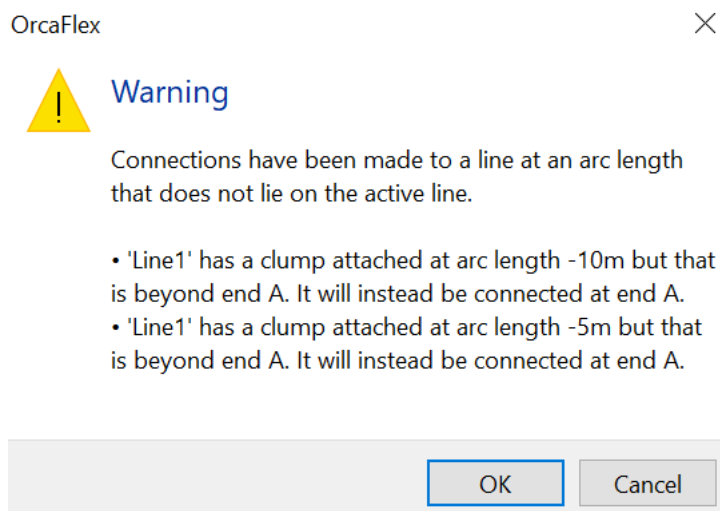


Fig. 7: The initial warning message in OrcaFlex providing information about the colliding objects.

OrcaFlex

? Statics calculation failed

The statics iteration has used 400 iterations and not converged. Please choose from the following options:

1. Click abort to abort the calculation.
2. Click increase max iterations to extend the calculation.
3. Click use calculated positions to set the initial positions of buoys, vessels and free line ends to be the latest positions found by the iteration.

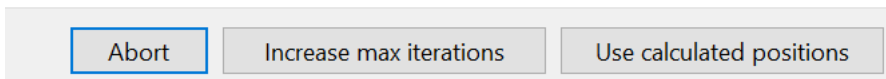


Fig. 8: The message in OrcaFlex informing about the failed calculation.

```
In [5]: runfile('C:/Code/Python/OrcFxAPI/Test/AnalysisProgramV2.py', wdir='C:/Code/Python/OrcFxAPI/Test')
Reloaded modules: OrcFxAPIConfig, OrcFxAPI, UpdateInData
Traceback (most recent call last):
  File ~\AppData\Local\Programs\Python\Python38-64\python.exe in compat_exec
    exec(code, globals, locals)
  File c:\code\python\orcfxapi\test\analysisprogramv2.py:76
    model.CalculateStatics()
  File C:\Program Files (x86)\Orcina\OrcaFlex\11.3\OrcFxAPI\Python\OrcFxAPI.py:6426 in CalculateStatics
    _CheckStatus(status)
  File C:\Program Files (x86)\Orcina\OrcaFlex\11.3\OrcFxAPI\Python\OrcFxAPI.py:1226 in _CheckStatus
    raise DLLError(StatusCode(status.value))
DLLError:
Error code: 27
Static calculation failed (Whole system statics: Not converged.)
```

Fig. 9: OrcaFlex informing about the failed calculation through the IDE by providing the error message and stopping the execution of the script.

8.3.3 Static Results

There are several different variables calculated in the static calculation. The desired results can be accessed by selecting the desired variables, as shown in Fig. 10.

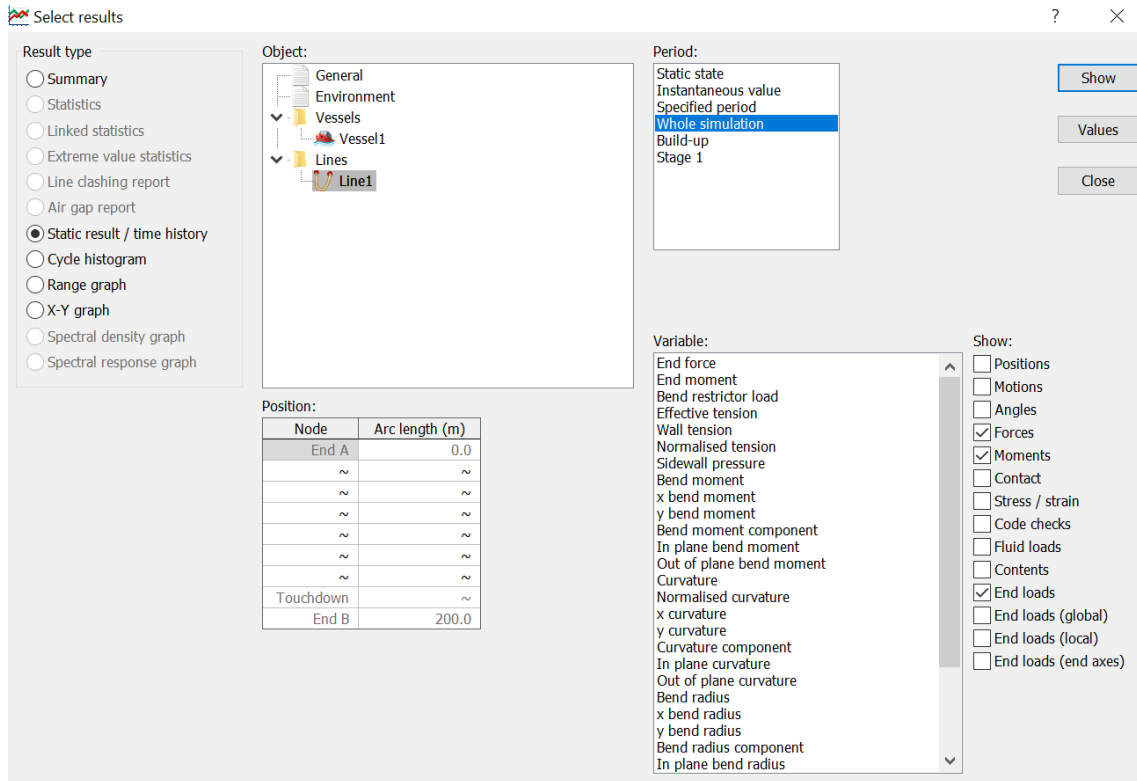


Fig. 10: Static results in OrcaFlex.

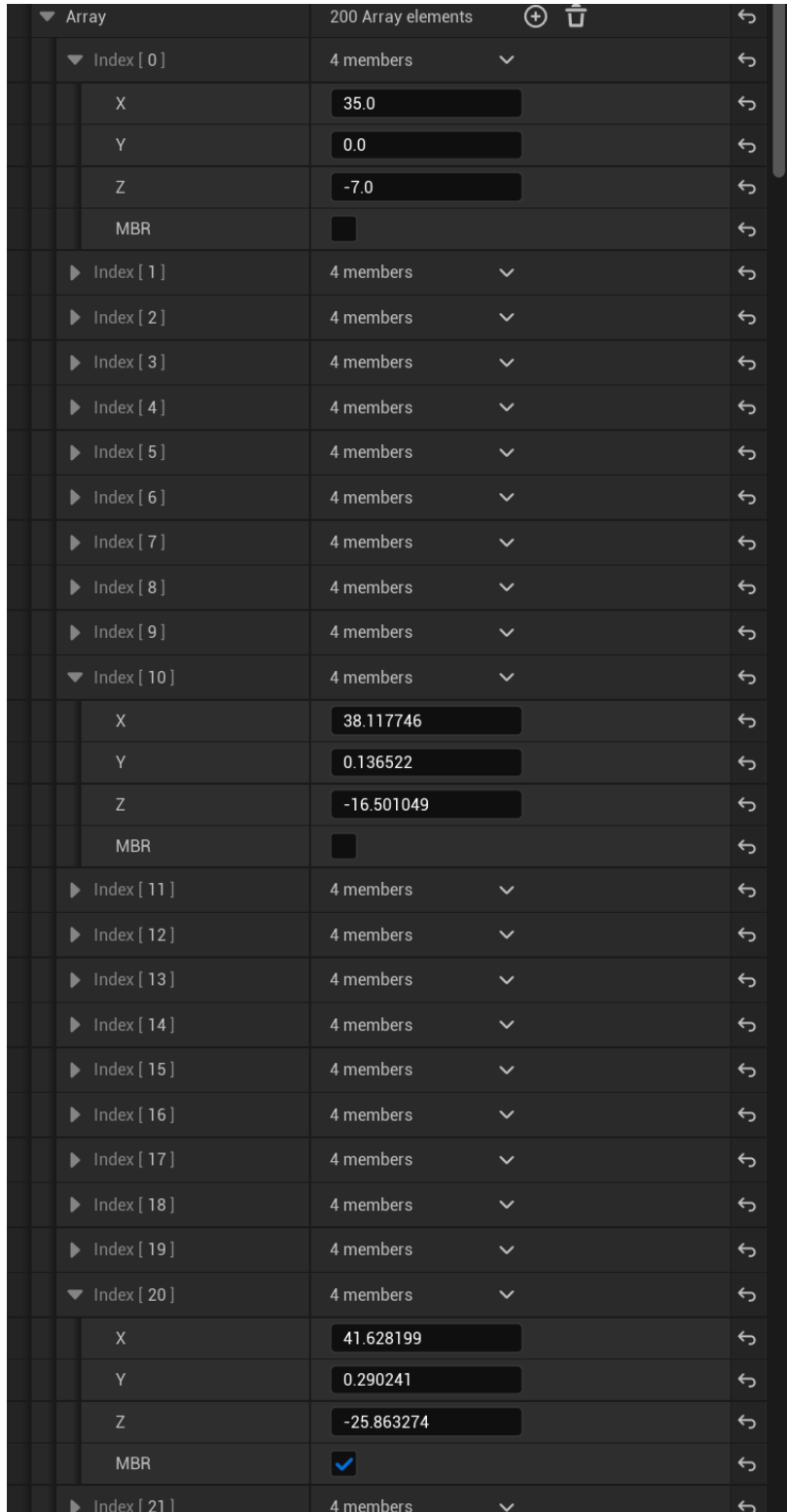
8.3.4 Compare Function

The compare function was tested by printing the bend radius (the input parameter to the function) and the Boolean from the function controlling that False was set when bend radius was more than 50 (MBR) and True was set when bend radius was less than 50 (MBR).

137.10690806939192	False
False	66.75631119491653
132.9769070255798	False
False	64.269660178893
128.92461508435488	False
False	61.86226525304476
124.95006017387625	False
False	59.53425037553889
121.0532731087579	False
False	57.28573752237278
117.2342878601855	False
False	55.116844062734316
113.49314184501569	False
False	53.027679793128485
109.82987623312557	False
False	51.01834372656755
106.2445362699374	False
False	49.08892080070277
102.73717161171716	True
False	47.23947875813829
99.30783666794167	True
False	45.47006557056294
95.95659094338606	True
False	43.780707933654384
92.6834993716165	True
False	42.171411567598675
89.48863262725328	True
False	40.64216433922321
86.37206740049378	True
False	39.192943612344756
83.33388661474534	True
False	37.82372978862946
80.37417956281192	True
False	36.534528810144536
77.49304193324424	True
False	35.32540759997485
74.69057569199973	True
False	34.19654824342776
71.9668887819528	True
False	33.14832952907693
69.32209459830226	True

8.4 Unit Test Live View

The analysis results were added to the node array as depicted in Fig. 11.



The screenshot shows a node array in UE5 with 200 elements. The array is organized into indices, each containing 4 members (X, Y, Z, MBR). The values for X, Y, and Z are displayed in input fields, and the MBR member has a checkmark.

Index	Members	X	Y	Z	MBR
Index [0]	4 members	35.0	0.0	-7.0	
Index [1]	4 members				
Index [2]	4 members				
Index [3]	4 members				
Index [4]	4 members				
Index [5]	4 members				
Index [6]	4 members				
Index [7]	4 members				
Index [8]	4 members				
Index [9]	4 members				
Index [10]	4 members	38.117746	0.136522	-16.501049	
Index [11]	4 members				
Index [12]	4 members				
Index [13]	4 members				
Index [14]	4 members				
Index [15]	4 members				
Index [16]	4 members				
Index [17]	4 members				
Index [18]	4 members				
Index [19]	4 members				
Index [20]	4 members	41.628199	0.290241	-25.863274	✓
Index [21]	4 members				

Fig. 11: The node array in UE5 when the spline points are created.

8.5 Prototype Testing

8.5.1 Digital Twin

The following figures, Fig. 12, Fig. 13, Fig. 14, Fig. 15, and Fig. 16, illustrates the digital twin during the prototype testing showing different stages of the operation.

A visual error occurred during the testing of the prototype. The 3D-model of the flexible product was glitching as illustrated by Fig. 12 and Fig. 13. This only happened when the line was going straight down.

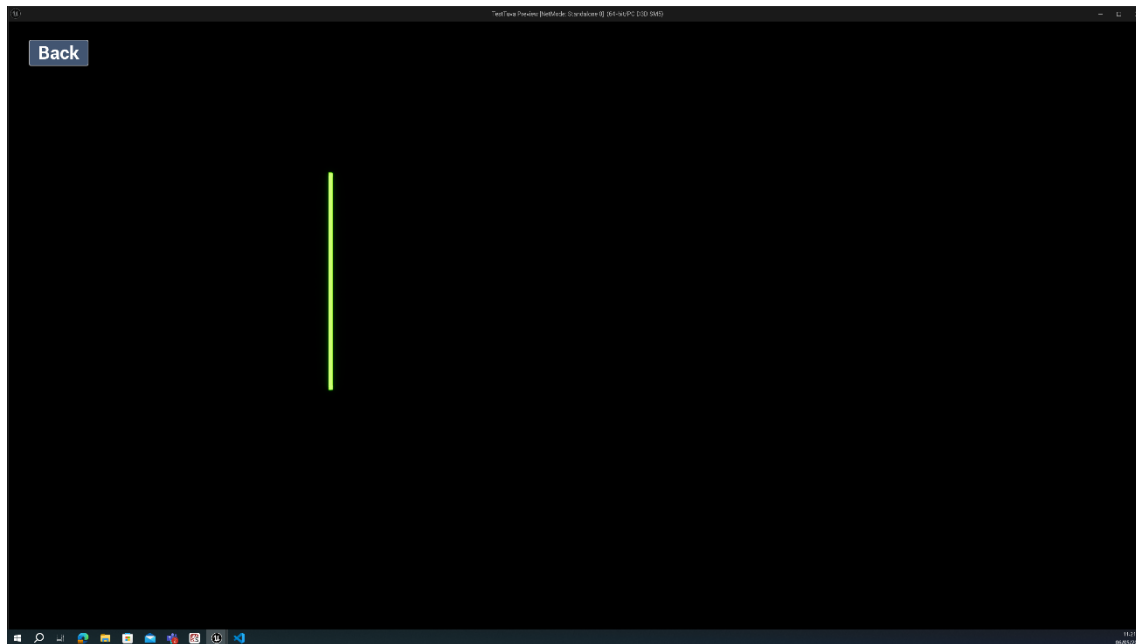


Fig. 12: The line representing the flexible product during installation visualised in UE5.

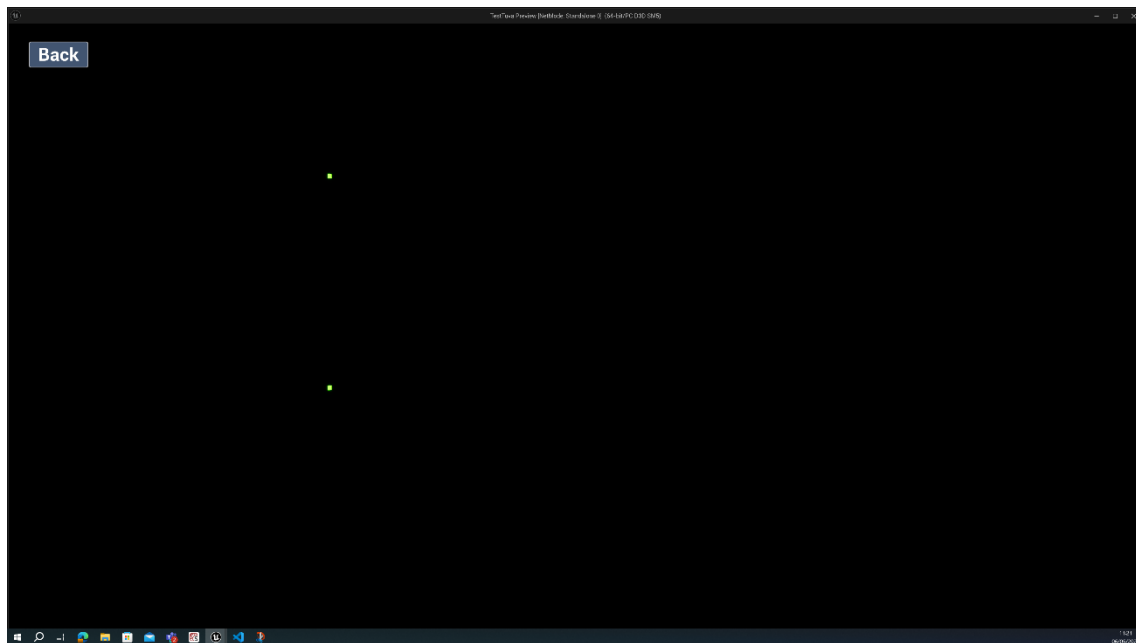


Fig. 13: The line representing the flexible product during installation visualised in UE5.

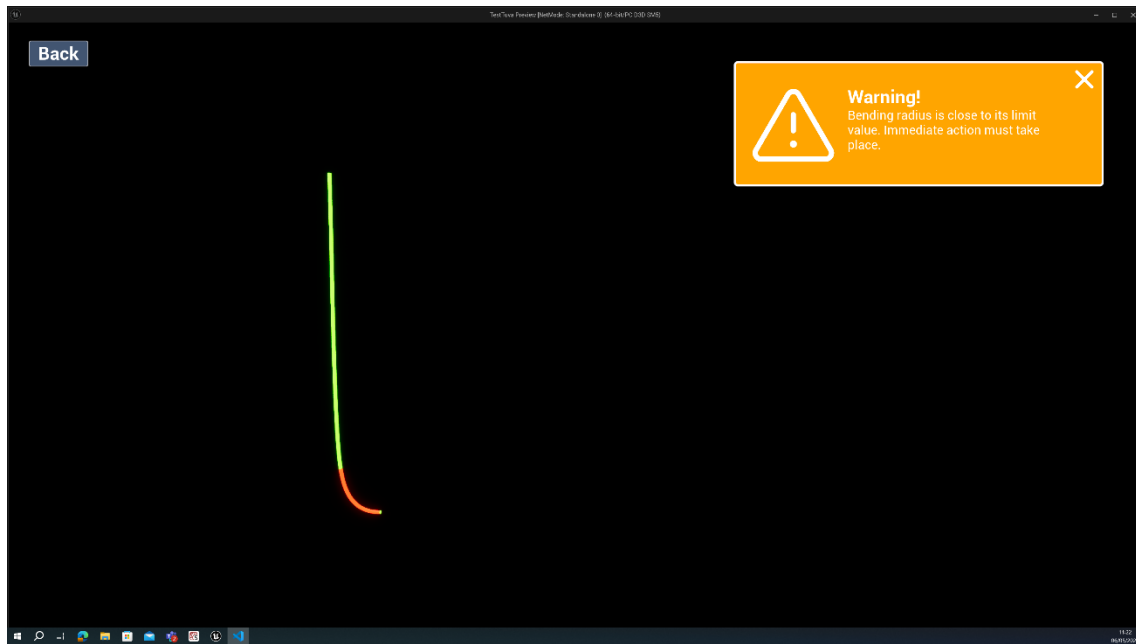


Fig. 14: The line representing the flexible product during installation visualised in UE5. The product reached touch down.

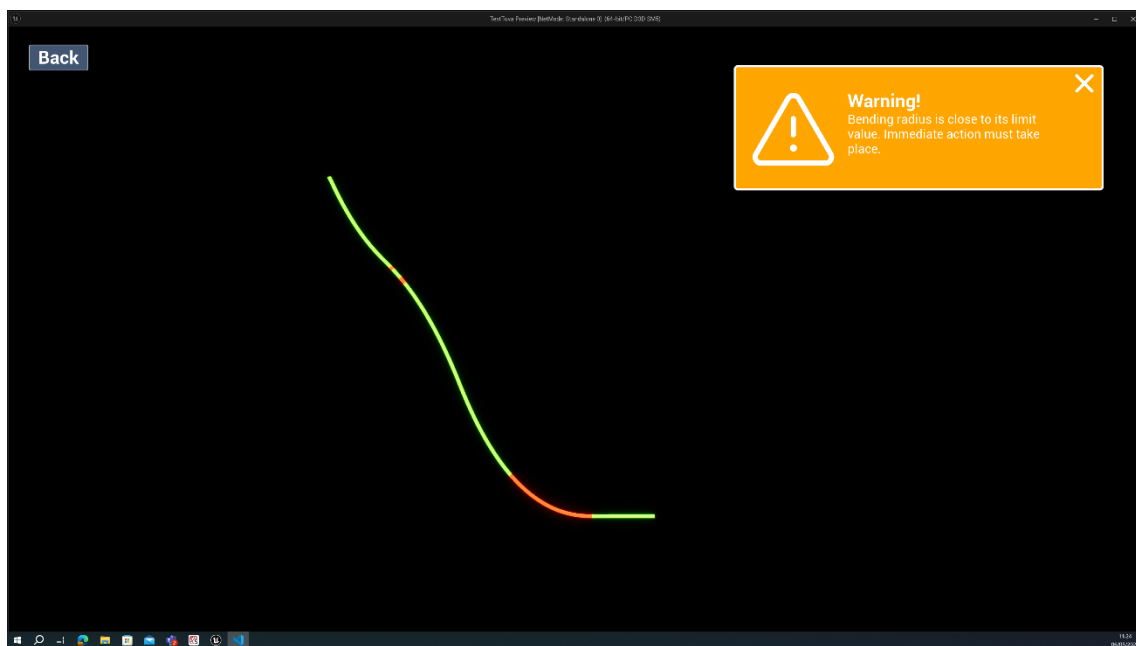


Fig. 15: The line representing the flexible product during installation visualised in UE5. The flexible product is starting to create a wave-shaped configuration.

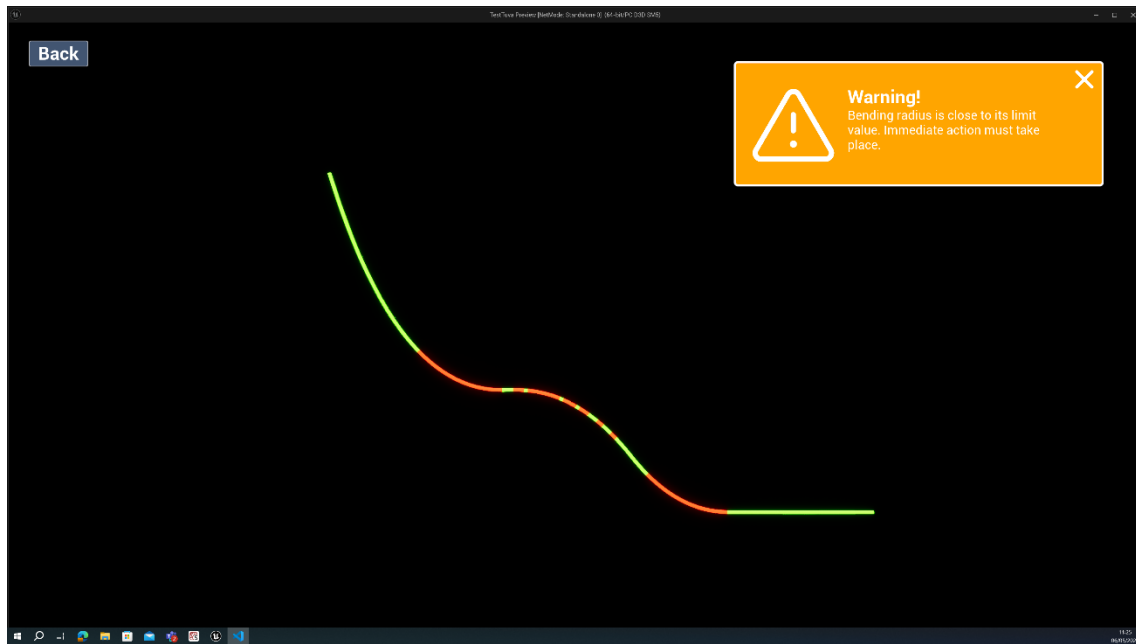


Fig. 16: The line representing the flexible product during installation visualised in UE5. The flexible product with a wave-shaped configuration.

8.5.2 AnalysisResult.json

The following shows part of the output file from the analysis program.

```

{
  "Array": [
    {
      "X": 35.0,
      "Y": 0.0,
      "Z": -7.0,
      "MBR": false
    },
    {
      "X": 35.20042644258914,
      "Y": 1.7350584739534458e-17,
      "Z": -7.979802636660364,
      "MBR": false
    },
    {
      "X": 35.40213319909997,
      "Y": 3.481200413194503e-17,
      "Z": -8.959341362645212,
      "MBR": true
    },
    {
      "X": 35.60577302820516,
      "Y": 5.2440766412961677e-17,
      "Z": -9.938478900384919,
      "MBR": true
    },
    {
      "X": 35.81169649406967,
      "Y": 7.02672193409556e-17,
      "Z": -10.91713757000365,
      "MBR": true
    },
    {
      "X": 36.02010882105579,
      "Y": 8.830912884861837e-17,
      "Z": -11.895268101272734,
      "MBR": true
    },
    {
      "X": 36.23114574915951,
      "Y": 1.0657824572228391e-16,
      "Z": -12.872834579486982,
      "MBR": true
    }
  ],
  {
    "X": 36.44491038853043,
    "Y": 1.2508349603660355e-16,
    "Z": -13.849807073857114,
    "MBR": true
  },
  {
    "X": 36.66149123090244,
    "Y": 1.4383254037421961e-16,
    "Z": -14.826157977662055,
    "MBR": true
  }
  ... ] }

```




Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway