Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

Shamsi Musayev

The Cost Model of Subsea Production Systems and Flowlines including Automatic Flowline Routing

Master's thesis in Petroleum Engineering
Supervisor: Tor Berge Gjersvik
June 2020

NTNU
Norwegian University of
Science and Technology

Shamsi Musayev

The Cost Model of Subsea Production Systems and Flowlines including Automatic Flowline Routing

Master's thesis in Petroleum Engineering
Supervisor: Tor Berge Gjersvik
June 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

NTNU
Norwegian University of
Science and Technology

Summary

Nowadays, optimization of field development projects in a cost-efficient way is one of the crucial topics in the petroleum industry. The current economic conditions and unpredictable fluctuations of oil price force all members of the petroleum industry to work on optimization projects to minimize total cost of field development projects.

Optimization of wellhead locations in subsea fields is an important part of the overall optimization process in subsea field development projects. The reason behind this is that the process of selecting wellhead locations has direct effect on two elements of total subsea field development costs. The first element is well costs and the second element is costs of subsea production systems (SPS) and flowlines. When locations of wellheads are changed but target points for the wells are kept same, trajectories of the wells changes, hence, the well costs are changing too. Also, when locations of wellheads are changed, lengths and routes of the flowlines and umbilicals in SPS changes, therefore, costs of SPS and flowlines are also changing. The objective of wellhead placement optimization is finding the wellhead locations which minimizes sum of the mentioned two cost elements.

In order to achieve optimization of wellhead locations, two cost models are required to be used in the optimization process. One cost model for well costs and another one for costs of SPS and flowlines. This work has focused on creation of the latterly mentioned cost model, which is for costs of SPS and flowlines. The aim of the thesis was creating a cost model, which can be used to calculate total cost of SPS and flowlines in a certain subsea field by using seabed topography, wellhead locations and subsea equipment cost data as an input.

In the creation process of the desired cost model, firstly main components of SPS and flowlines, which have been selected for adding to the cost model, have been discussed. Then, the most common types of SPS layouts (template, clustered satellite wells, satellite wells and daisy-chain) and life cycle cost (LCC) of subsea field development project have been analysed. Moreover, different elements of LCC have been discussed and it was decided to focus on only capital expenditures (CAPEX) in the thesis. The reason behind this decision is that CAPEX is the biggest part of the LCC and more precise information is available for CAPEX during wellhead location process.

After discussing all necessary preliminary information for the cost model, it was decided to divide the cost model into two elements. As seabed topography and wellhead locations are considered as input data, the first element of the cost model has been dedicated for automatically finding the best routes for flowlines and umbilicals between given points in the certain subsea field and calculating lengths of the determined routes. For this purpose, MATLAB code has been written. In the written MATLAB code, Dijkstra's algorithm has been used for determining the shortest routes between given points, cubic spline interpolation has been used for smoothing the determined routes and the Gaussian-Kronrod quadrature method of numerical integration has been used for calculating lengths of the determined routes.

The second element of the cost model has been dedicated for gathering all available data and all obtained results from the first element and for calculating total CAPEX of SPS and flowlines. For this purpose, a spreadsheet has been created. So, the final version of the cost model, which can be used in optimization of wellhead locations, contains the written MATLAB code and the created spreadsheet.

After creating the desired cost model for SPS and flowlines, applicability of the model has been checked by doing a case study in the artificially created imaginary subsea field. In the case study, CAPEX values of SPS and flowlines with four different subsea layouts have been determined. Moreover, locations of wellheads have been changed in the case study and different CAPEX values have been obtained for different arrangements of wellhead locations. So, it has proven that the created model can be used for comparing different subsea layouts in a certain subsea field and it can be used as a part of the optimization of wellhead locations. All above mentioned steps and results are widely discussed in this thesis.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisor Professor Tor Berge Gjersvik and to my co-supervisor Professor Audun Faanes for their priceless supervisions, consultations and constructive feedbacks which increased the quality of my thesis significantly. It was a great pleasure for me to be supervised by them and I hope they have enjoyed in our discussions as much as I have.

Moreover, I am grateful to Rashad Nazaraliyev for his friendship and support throughout my master's degree education.

Finally, I want to express the biggest thankfulness to my family, whose continuous support and encouragement allowed me to be where I am. I dedicate this work to my family and to my friends that I see as a part of my family.

Shamsi Musayev

Trondheim, June 2020

Contents

List of Figures

List of Tables

Nomenclature

ABBREVIATIONS

API	American Petroleum Institute
BOP	Blowout Preventer
CAPEX	Capital Expenditure
DCV	Directional Control Valve
DHSV	Downhole Safety Valve
FBS	Flow Base Structure
GEBCO	General Bathymetric Chart of the Oceans
HOST	Hinge Over Subsea Template
HP	High Pressure
HT	High Temperature
HXT	Horizontal X-mas Tree
IPU	Integrated Production Umbilicals
ITS	Integrated Template Structure
LCC	Life Cycle Cost
LCP	Least Cost Path
LP	Low Pressure
OPEX	Operational Expenditures
PLEM	Pipeline End Manifold
PNU	Polymer Nanotube Umbilicals
RAMEX	Reliability, Availability, Maintainability Expenditures
RISEX	Expenditures Related to Blowout Risk
ROV	Remotely Operated Vehicle
SCM	Subsea Control Module
SCS	Subsea Control System
SDU	Subsea Distribution Unit
SEM	Subsea Electronic Module
SPS	Subsea Production System
SUTA	Subsea Umbilical Termination Unit
TCP	Thermoplastic Composite Pipes
TH	Tubing Hanger
VXT	Vertical X-mas Tree
WH	Wellhead

SYMBOLS

C_b	Blowout cost
C_l	Cost of the lost production
C_m	Expenditure for the maintenance work
L_{total}	Total length of the determined route
P_b	Blowout probability
R_{min}	Required minimum radius of curvature
T_H	Residual tension/horizontal bottom tension
W_s	Submerged weight of the flowline
a_i, b_i, c_i, d_i	Coefficients for the third order polynomial of i -th interval

d_e	Euclidean distance between two points
l_i	Sum of the Euclidean distances till i -th point
l_{total}	Total sum of the Euclidean distances in the determined path
$\vec{p}$	Vector function of the determined route
s_i	Third order polynomial for the i -th interval
$\tilde{x}$	Gauss-Kronrod quadrature points
$[a, b]$	Interval for integration
F	Safety factor
S	Set of third order polynomials
x, y, z	Cartesian coordinates
μ	Lateral soil-flowline friction factor
ω	Gauss-Kronrod quadrature weights

1. Introduction

The reports of leading energy analysis firms show that the world's energy consumption is continuously increasing and this tendency will certainly continue into the following decades. Nowadays, huge amount of money is invested in renewable energy sources and renewable energy sources are beginning to play crucial role while meeting world's energy demand. However, it is forecasted that fossil fuels such as oil and gas will still constitute approximately 50 percent of the primary energy consumption in the world by 2040 (Figure 1. 1).

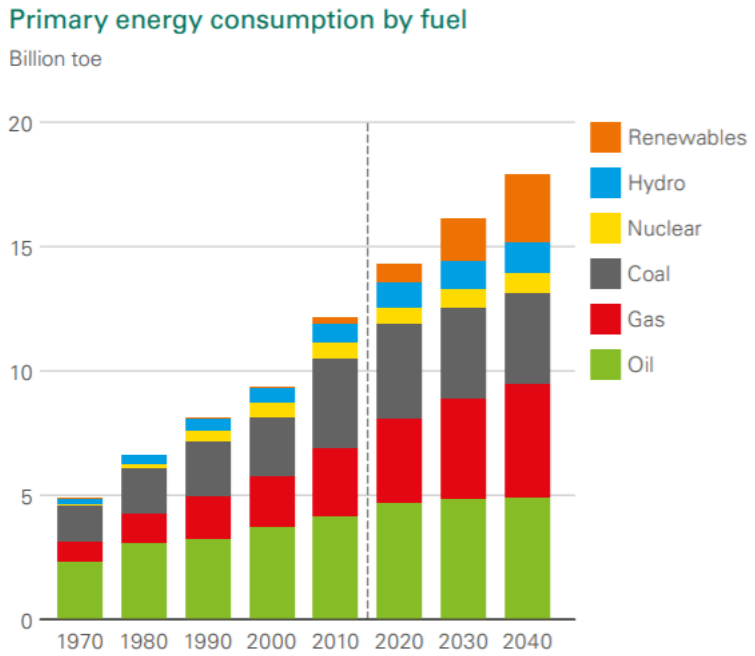


Figure 1. 1. Prediction for primary energy consumption (BP Energy Outlook 2019)

As petroleum energy recourses are non-renewable sources, it will be too difficult to meet required demand for petroleum resources in the future. One of the main reasons behind this problem is that it is becoming more difficult to find conventional petroleum fields in the world. Hence, petroleum companies are forced to focus on possible developments of unconventional petroleum recourses. However, it is not easy to develop unconventional fields in the current economic environment with unpredictable oil price fluctuation. Therefore, in today's life, it is a popular topic among industry leaders, engineers and researchers to work on optimization projects for finding cost-efficient solutions for different problems of hydrocarbon field development. It is worth to mention that, cost optimization gained crucial importance in the petroleum industry especially after the global turndown in the petroleum industry (Labbé et al. 2019).

The optimization of wellhead locations is one branch of overall optimization of subsea field development projects. Decisions about locations of wellheads have a direct influence on two cost elements of total subsea field development cost. The first cost element is well costs because if completion targets are not changed, well trajectories are changing while changing location of wellheads. The second cost element is the costs of subsea production system (SPS) and flowlines because it is obvious that routes and lengths of flowlines and umbilicals are changing while changing positions of wellheads. Therefore, the main target during the wellhead placement optimization is minimizing sum of well costs and costs of SPS and flowlines. However, in order to analyse effect of wellhead placement on total cost of subsea field development project and to find the best locations for wellheads, it is required to have certain cost models for well costs and for costs of SPS and flowlines.

This work focuses on latterly mentioned cost modelling – cost modelling of SPS and flowlines. The objective of the work is creating a model which can be used to calculate cost of SPS and flowlines by using seabed topography data, coordinates of wellheads and riser base as input parameters. For this purpose, firstly, all main cost components (i.e. subsea equipment) for the section from wellhead to riser base will be discussed. Then, the most common types of SPS layouts (template, clustered satellite wells, satellite wells and daisy-chain) will be mentioned. Life cycle cost (LCC) of subsea field development projects and its elements will also be analysed in the thesis.

The desired cost model will be created as a combination of two elements. The first element will be a tool, which can be used to automatically find route of flowlines by using seabed topography, wellhead coordinates and riser base coordinates as input data. For this purpose MATLAB programming language will be used. After that, as a second element of the cost model, the spreadsheet will be created to gather all available information for calculating total cost of SPS and flowlines. In the end, a case study will be done to check applicability of the created model and to analyse effect of the wellhead placement optimization. For the case study, imaginary subsea field will be created and the created cost model will be used to calculate total cost of SPS and flowlines in the imaginary subsea field.

1.1. Outline of the Thesis

The short summaries of the following chapters are described below:

Chapter 2: Subsea Equipment Costs. In this chapter, several subsea equipment, which covers the biggest part of subsea field development cost and is the most common, will be discussed. Moreover, the most common types of each mentioned equipment will be described in the chapter. Then, chapter will continue with brief information about different cost types (e.g. CAPEX, OPEX, etc.).

Chapter 3: Types of Subsea Production System Configurations. This chapter will include short information about four common types of subsea production system layouts (template systems, clustered satellite wells systems, satellite wells systems and daisy chain systems) will be given. Several advantages and disadvantages of each configuration will also be mentioned in the chapter.

Chapter 4: Automatic Flowline Routing. In this chapter, the tool, which can be used to automatically determine the most optimum flowline routes and to calculate their length, will be created. The desired tool will be created in MATLAB programming language and the chapter will include explanation of all used theories while creating MATLAB code. The written code will be explained step-by-step and any problem that can be encountered while using the code will be discussed. In the end, possible improvements to the existing code will be recommended.

Chapter 5: Case Study – Effect of the Wellhead Placement Optimization. In this chapter, information about the second element of the cost model, which is a spreadsheet, will be given. Then, the case study will be done for checking applicability of the created cost model. For the case study, artificial subsea field will be created. In the case study, CAPEX of SPS and flowlines in the imaginary subsea field with four different subsea field configurations (template systems, clustered well systems, satellite well systems and daisy chain systems) will be calculated. Additionally, importance of the wellhead placement optimization will be analysed.

Chapter 6: Conclusion. This chapter will include end-summary of the work and concluding remarks. Some recommendations for further work on this topic will also be mentioned in the chapter.

2. Subsea Equipment Costs

Depending on field development strategy, subsea production systems (SPS) can contain different subsea equipment. The cost model, which is targeted to be created in this work, will include only certain subsea equipment, which is frequently used in subsea field development projects and covers the biggest part of the total cost of SPS and flowlines. List of the selected subsea equipment, which will be included to the cost model in this thesis, is shown below:

1) Foundations
2) Templates
3) Protection structures
4) Wellheads
5) Tubing hangers
6) Christmas trees
7) Manifolds
8) Subsea control modules
9) Subsea distribution units
10) Umbilicals
11) Flowlines

The chapter is dedicated to give a brief explanation about each selected equipment and to discuss the most common types of selected equipment. Life cycle cost (LCC) of subsea field development projects and its all elements will also be discussed and analysed in the chapter.

2.1. Foundations

Foundation is a part of subsea production system which is used to transfer loads of subsea equipment (e.g. templates, manifolds, etc.) to the soil. Foundations are mainly categorized as shallow foundations and deep foundations. However, there is also another category of foundations which is a combination of the mentioned two categories. This category foundations are named as hybrid foundations (Dimmock et al. 2013).

Depending on properties of the soil, loads of subsea equipment can be carried by three different types of foundation structures. These types are foundation structures that supported by seabed, foundation structures that supported by piles (Figure 2. 1) and foundation structures that supported by mudmats with connected skirts. However, it

Figure 2. 1. The picture of suction pile (Faulk 2008)

is also possible to use a mixture of the three mentioned options.

Skirt Supported and Pile Supported Foundations – While using these foundations, the foundation structure should be properly combined with skirt or pile (Figure 2. 2). For the connection process, mechanical device can be used or annular gap between sleeve and the pile can be grouted. Moreover, during design process of the mentioned foundation types, all possible shear stresses, tensional stresses, compressional stresses and lateral loads should be considered.

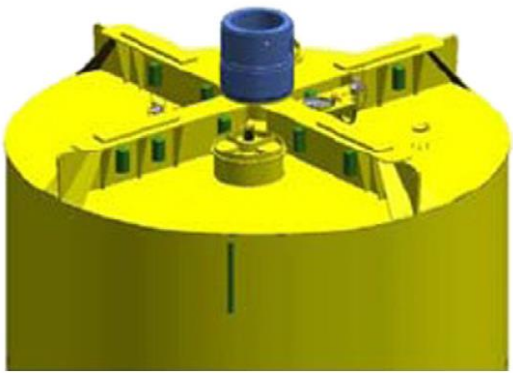


Figure 2. 2. Connected foundation structure and pile (Faulk 2008)

Seabed Supported Foundations – Two main requirements should be met during design process of these foundation structures. Firstly, in order to carry all expected loads, the designed foundation structure should have enough bearing capacities in horizontal and vertical directions. Secondly, all possible contact stresses, which can occur during life of the field, must be considered. It is an option to use under base grouting to accomplish the needed load allocation and stability on the seabed.

As it was mentioned before, selection process of foundation structures depends on the soil properties of the area. The soil in the area can be loose sand, can be consolidated and hard or can be something between these two. If the area has the soil with enough softness, pile supported or skirt supported foundations can be chosen because in this type of location it will be possible to penetrate these two structures to enough depth and foundation structures will be able to carry the weight of the subsea equipment. However, it is also worth to mention that each location has its own requirements and properties, hence, it is not easy to make general statements about selection process and each selection should be done according to the detailed analysis of the location.

2.2. Templates

Template is the part of subsea production systems which is used to group numerous subsea wells at a certain location on the seabed. Template is also used to support manifolds, completion equipment, drilling equipment, wellheads, risers, etc. Additionally, they are used as a guidance while drilling processes (Brinkmann et al. 1987).

During production phase of the field, produced hydrocarbons are sent from templates to shore, to platforms or to floating vessels. Templates should be designed in a way that they will have sufficient load capacity to meet requirements for future drilling and maintenance operations. Furthermore, the chosen template should have sufficient load capacity to handle pipeline installation forces and loads from thermal expansion of the pipelines and the wellheads. If load capacity requirements cannot be met during design stage of the template, breakaway apparatus can be used for protection purposes.

There are several types of templates and typical template types are discussed below:

1) **Modular Template** – Modular templates are placed around a particular structure (e.g. well). It is possible to install modular template as a single module or as a combination of numerous units. Modular templates offer a chance to make last minute adjustments in the production program and allow companies to "build as you go".
2) **Manifold Template or Multiwell Template** – This type of template contains numerous slots for wells and special place for the manifold (Figure 2. 3).

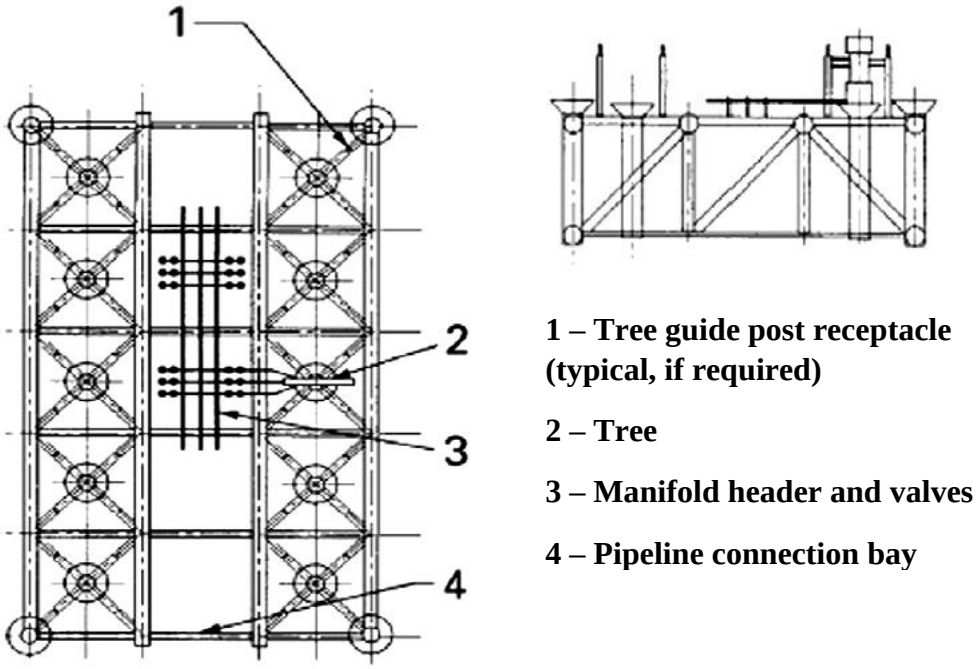


Figure 2. 3. The schematic of manifold or multiwell template (API 2002)

3) **Separate Manifold Template** – Separate manifold template is used only for the manifold and it is not possible to drill any well through the template. This type of template is installed around numerous satellite subsea wells.
4) **Well Spacer Template or Tie-back Template** – Well spacer template is similar to multiwell template. The only difference between these two types is that well spacer

templates can only be used for wells, and manifolds cannot be installed on well spacer templates. These templates mainly play guidance role to predrill subsea wells before installation of surface facility. Then, during completion stage, the predrilled wells can be tied back to the surface facility. Separate risers can be used to tie back the predrilled wells to the surface facility (Figure 2. 4).

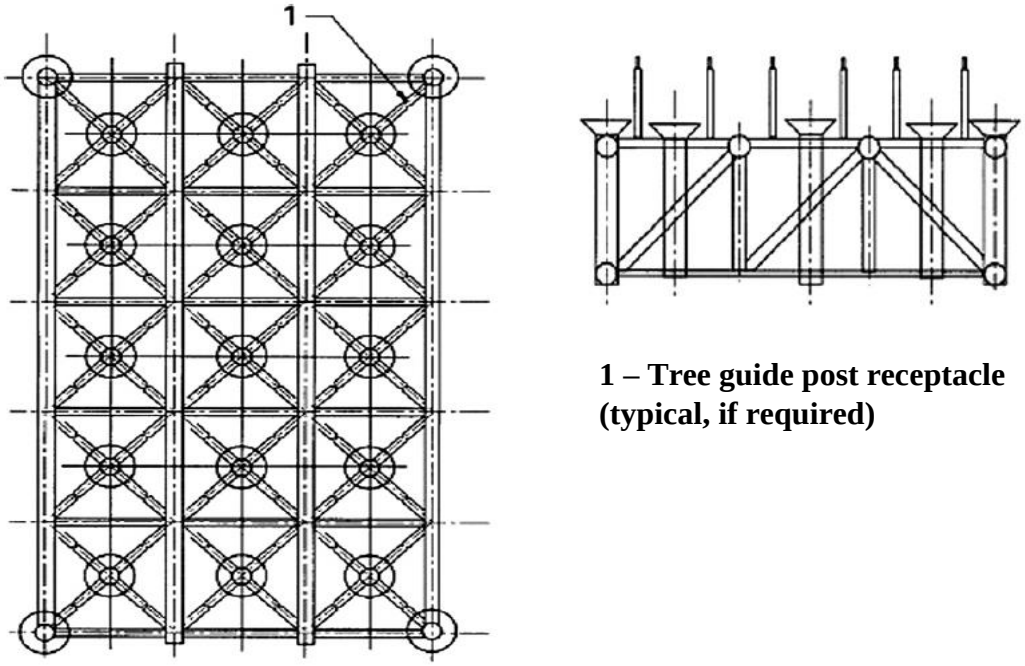


Figure 2. 4. The schematic of well spacer template (API 2002)

5) **Riser-Support Template** – Riser-support templates are designed for supporting production risers and loading terminals. Pipeline connection capability can also be added to the design of these templates.

Frequently used templates, such as hinge over subsea templates (HOST), integrated template structures (ITS), flow base structures (FBS), can merge features of numerous template types in one design.

In today's operations, installation of templates is very expensive, especially in deep water locations. The reason behind this is that currently used templates are too heavy and hence huge vessels are used to install these templates. Therefore, there are several ongoing researches with objective of adjusting the material of current templates to decrease the weight of templates. For example, Lunde & Nesheim 2017 suggests that aluminium can be used for this purpose and aluminium can replace steel in the manufacturing process of templates. However, this suggestion still needs further analysis before its implementation.

2.3. Protection Structures

In subsea fields, it is crucial requirement to protect installed subsea equipment from potential damages or to minimize consequences of possible damages. In the subsea field, damages can usually be from dropped objects, fishing gear or anchor snags (Copsey & Johnson 1993). In order to protect subsea equipment, protection structures are designed and installed. These structures are categorized as local structures and global structures. Global structures are designed to safeguard subsea systems. However, sometimes smaller parts of subsea systems (e.g. swab valves or master valves of X-mas trees, etc.) entails higher level of protection. In these cases, local protection structures are installed.

As global protection structures have bigger effect on the total cost of the subsea field development, these structures will be discussed more widely in the thesis.

Global protection structures are divided into two categories. These categories are deflection structures and snagging structures (Towers-Perkins 1987).

1) **Snagging Global Protection Structures –** Snagging structures are used in the areas where fishing activities are forbidden or limited fishing activities are expected (Figure 2. 5). Several characteristics of these structures are mentioned below:

- Roof installation for trawl gear deflection is not mandatory while using snagging structures
- Structure has V shape, hence the net, which is snagged, will not ride up the legs and the snagged net will drop into the structure

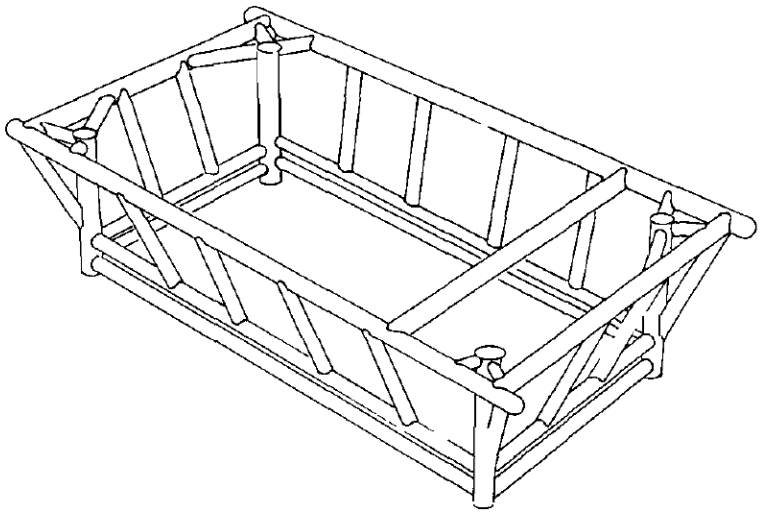


Figure 2. 5. The schematic of snagged global protection structure (Towers-Perkins 1987)

- If there is any smaller equipment which needs higher level of protection, a local protection structure should be installed to prevent it from dropped objects.

2) **Deflection Global Protection Structures** – Deflection structures are used in locations where frequent fishing activities are expected (Figure 2. 6). Several characteristics of these structures are mentioned below:

- Roof installation is mandatory while using deflection type global protection structures because subsea equipment, which is protected by protection structure, should not be tangled with fishing nets. Frequently used roof types are hinged roofs, integrated module roofs and retrievable roofs.
- While intervention work is done, installed roof should be removed or to be opened.
- These structures have raking tubulars which force wires of anchor and gears of trawl to pass over the deflection type protection structure. As a result, the protected subsea equipment and the gears of trawl are not damaged.
- Installed roofing system usually does not protect subsea equipment from falling objects. Therefore, installation of local protection structures may still be requirement.

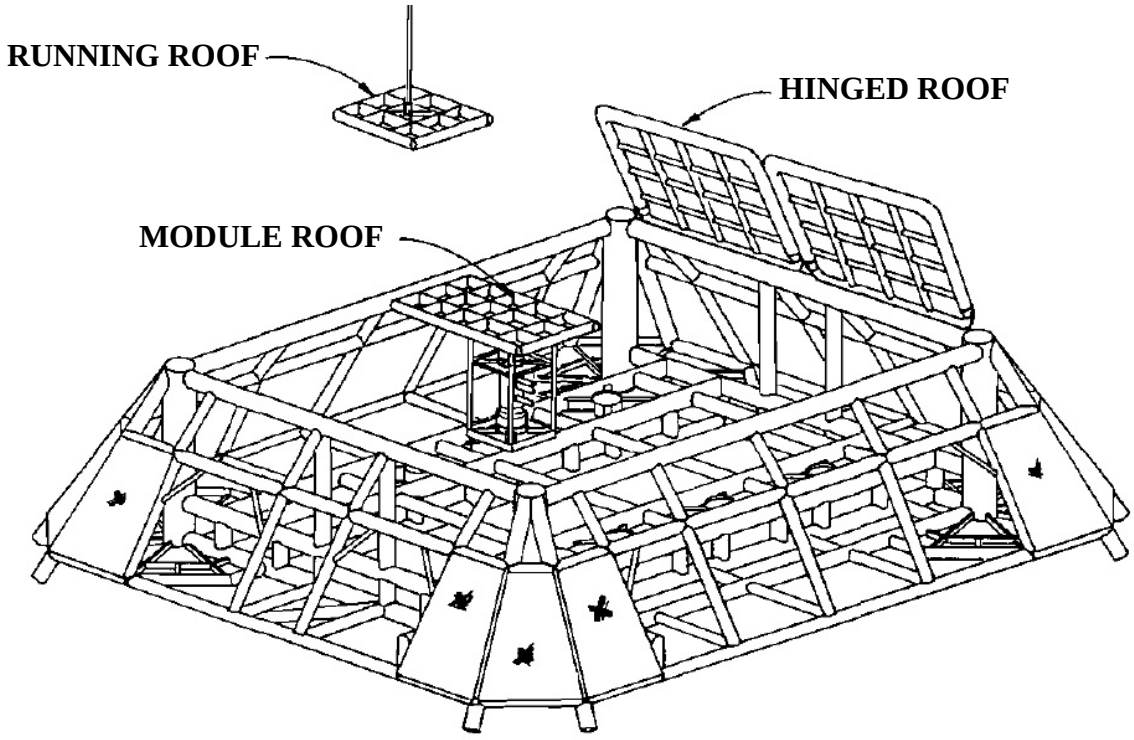


Figure 2. 6. The schematic of deflection global protection structure (Towers-Perkins 1987)

Steel or concrete are usually used as a raw material for manufacturing of protection structures. However, numerous ongoing researches are done to analyse possible adjustments of the raw material, as it is done for templates.

2.4. Wellheads

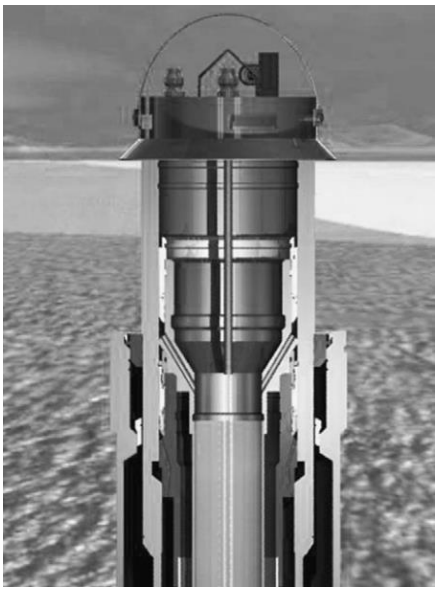


Figure 2. 7. Subsea Wellhead (Bai and Bai 2018)

The term "wellhead" is used for describing the well component which is designed to provide structural resistance for handling external loads and to provide pressure containing interface for different stages of production and drilling operations (Kaculi 2015). Subsea wellhead systems should contain various interfacing components and sub-systems, therefore, these systems have fairly complex designs. The components in the subsea wellhead system are wellhead housing, hangers for intermediate casing and production casing, lockdown bushing, annulus seal assemblies, BOP test tool, isolation test tool, etc. The internal profile of the wellhead is designed in a way that it can isolate annulus and support casing strings. It is crucial for wellheads to have a safe and robust design because there are some parts in the wellhead system which are the single barrier element between the environment and the wellbore fluid (Kaculi and Witwer 2014). Subsea wellheads are permanently mounted equipment; hence it is not possible to recover wellheads for repairment or inspection purposes after installation.

The main functions of the wellhead are mentioned below:

- To support blowout preventer (BOP) and Christmas tree system and to interface with them
- To ensure verticality, alignment and concentricity of the wellhead housing and the conductor housing
- To accept all possible loads during production, completion and drilling operations including thermal expansion

Nowadays, there are numerous subsea wellhead suppliers in the petroleum industry and they offer several different types of wellhead systems. The wellhead selection depends on load capacity, pressure rating, size, etc. requirements and these requirements are changing depending on configuration of the well, reservoir characteristics and so on (Evans and McGrail 2011).

2.5. Tubing Hangers

Tubing hangers are used to support and to seal off production tubing in the well. Depending on type of the Christmas tree, the tubing hanger can be installed in the wellhead or in the Christmas tree. Tubing hangers are generally divided into two categories:

1) **Non-orienting or concentric bore tubing hangers** – Concentric bore tubing hanger systems have only one central bore with threaded box for making up to one tubing string.
2) **Orienting or multibore tubing hangers** – Multibore tubing hanger systems contain two or more pockets which can be used for several stab receptacles and tubing strings. This design also gives an opportunity to operators to enter to the annular space directly from the top.

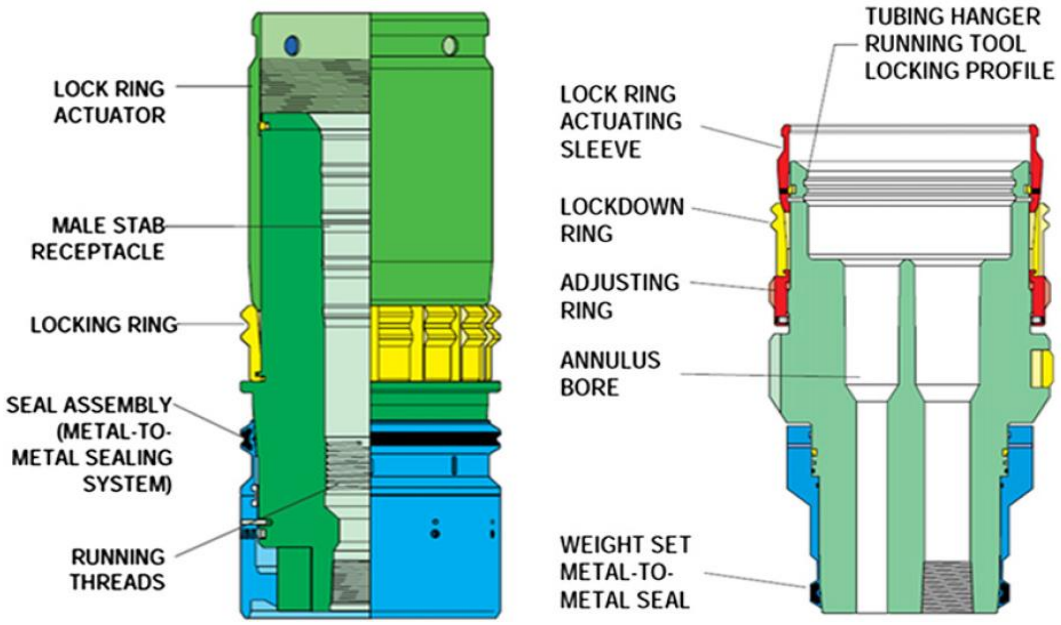


Figure 2. 8. (a) Concentric bore tubing hanger system; (b) Multibore tubing hanger system (Bai and Bai 2018)

The main functions of the tubing hanger systems are mentioned below:

- To support tubing strings in the well
- To seal off the annular space between the casing and the tubing string
- To provide required access to the annular space for operators
- To provide several conduits for chemical injection, monitoring and downhole safety valve control
- To supply an interface for the Christmas tree

2.6. Christmas Trees

Another important element of subsea production system is Christmas trees. Christmas trees have following main functions:

- To regulate oil and gas flows from the well by using choke valves
- To canalize oil and gas flows from the well to the flowlines in production wells and to direct the injection fluid (gas or water) to the target formation in injection wells
- To observe numerous well parameters (e.g. pressure, temperature, etc.)
- To safely stop production or injection during shutdowns
- To inject protection fluids (e.g. corrosion inhibitors, hydrate inhibitors, etc.) to the subsea well

Subsea Christmas trees have two types. The first one is horizontal Christmas trees (HXT) and the second one is vertical Christmas trees (VXT).

1) **Vertical Christmas Trees** – In VXTs, the master valve is vertically stacked and is installed above the tubing hanger. Production bores and annulus bores vertically pass through the tree body in VXTs. In VXTs, the well completion is finished prior to the tree installation. Therefore, the tubing hanger (TH) is stacked in the wellhead first and then installation of the VXT is done. This means VXTs can be recovered without the need for removing the well downhole completion. VXTs are widely used in the petroleum industry because of their high level of flexibility for installations and operations (Bai and Bai 2018).

Figure 2. 9. Vertical subsea Christmas tree (FMC)

2) **Horizontal Christmas Trees** – HXTs do not have barrier valves (e.g. swab valves) in the vertical section of the tree and valves are stacked in the horizontal sides. The most important difference between HXTs and VXTs is that in HXTs, the TH is not stacked in the wellhead. The TH is stacked in the HXT body. Hence, differently from VXTs, HXTs are mounted before the TH installation. Because of this feature of HXTs, it is

possible to replace the well downhole completion without the need of removing the HXT. Therefore, HXTs are preferred for the wells in which interventions (e.g. recompletion, etc.) will frequently be done. So, there will not any additional time lost to remove the Christmas tree before interventions. However, it is also worth to mention that HXTs are more expensive than VXTs. Hence, HXTs are chosen only for the wells in which intervention work frequency will be high enough to justify the additional investment.

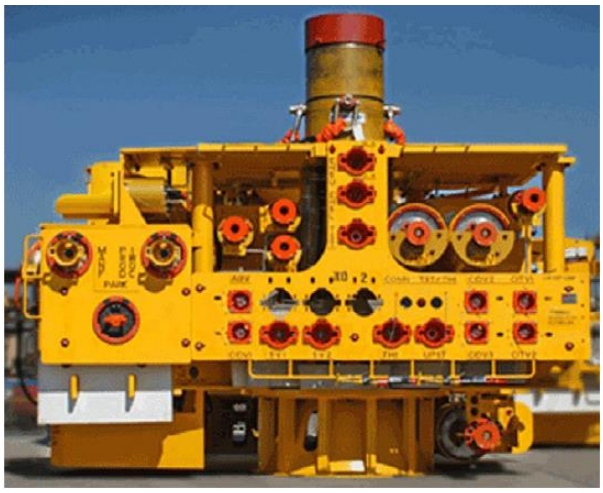


Figure 2. 10. Horizontal subsea Christmas tree (FMC)

2.7. Manifolds

A manifold is a composition of piping or/and numerous valves which is used to distribute, to combine, to monitor and to control fluid flow (Nmegbu & Ohazuruike 2014). Manifolds regulate distribution of gas or water injection into wells and combines produced fluid from numerous subsea wells for transferring produced hydrocarbons to flowlines. Usage of subsea manifolds optimizes flow of produced fluids and simplifies arrangement of subsea production systems by decreasing number of risers and flowlines (Paula et al. 2001). Because of the mentioned factors, manifolds seem as an attractive option for minimizing capital expenditures.

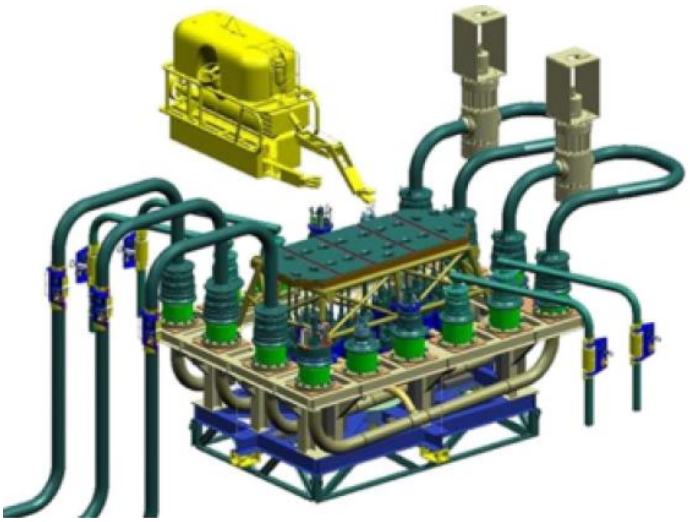


Figure 2. 11. The picture of subsea manifold (Nmegbu & Ohazuruike 2014)

In the petroleum industry, numerous kinds of manifolds are used. Depending on requirements and purposes, a basic pipeline end manifold (PLEM) can be chosen or large and complicated

structures may be preferred. However, the PLEM is the one which is mostly used in the industry. Manifolds have following main functions:

- To play an interface role between the production pipeline or the injection pipeline and the subsea wells
- To merge production from numerous wells and to allocate injection fluids into numerous wells
- To safeguard and to support all piping and all valves
- To play an interface role for sea-fastening
- To be used as a platform to support remotely operated vehicles (ROV)
- To support pipeline hubs, umbilical hubs and wing hubs
- To contain lifting points which can be used while installation and recovery of the manifold system

The cost of the manifold has undeniable influence on the total cost of the subsea production system. Hence, it is crucial to select the most optimum manifold type for the subsea production system. The manifold cost is a function of its location, its type and number of wells which are connected to the manifold (Grimmett and Startzman 1987). Latest researches propose that modularization and standardization of manifolds should be done for decreasing the manifold cost (McWilliams et al. 2018). It is also worth to mention that the compact modular type manifold is thought as the future for designs of manifold (Sundt and Ali 2019).

2.8. Subsea Control Systems

The subsea control system (SCS) is an important part of any subsea field development. SCSs are used to operate chokes and valves on pipelines, on manifolds and on subsea Christmas trees. SCSs also allow engineers to continuously monitor the production status by transferring the data (e.g. sand detection data, pressures, temperatures, etc.) among the surface facility and the subsea production system. SCSs contain numerous control elements and these elements are categorized as topside elements and subsea elements. Topside control elements are master control station, hydraulic power station, electrical power unit, etc. Subsea control elements are subsea control modules (SCM), subsea distribution units (SDU), umbilicals, etc. As this work focuses on the part from the wellhead to the riser base, only subsea control elements, which are commonly used and have a big influence on the total cost, will be discussed in the thesis.

2.8.1. Subsea Control Modules

Subsea control modules (SCM) are independently recoverable units. SCMs are installed on the special base which is designed for SCMs. Depending on how deep is the location, installation

Figure 2. 12. A typical base of SCM (FMC)

process can be performed by a ROV or by a diver (Broadbent 2010). Differently from SCM, its base is designed as a welded structure which is earth bonded and bolted to the subsea tree frame or to the manifold frame (Figure 2. 12) (Bai and Bai 2018). The base of the SCM plays interface role between valves of the subsea Christmas tree or valves of the manifold and the SCM. The base has hydraulic couplers, which is used for high pressure (HP) and low pressure (LP) supplies, and electrical couplers, which is used for signals and power.

SCM is known as the brain of the SCS (Kolios et al. 2017). Throughout the production stage of the subsea field, the SCM provides well control and monitoring functions. It is designed to interpret all coming signals and to allocate hydraulic and electrical power. The SCM is used to actuate subsea valves (e.g. shutoff valves, choke valves, manifold diverter valves, chemical injection valves, etc.) and downhole safety valve (DHSV). It is also used to monitor flow rates, sand detection data, temperatures and pressures.

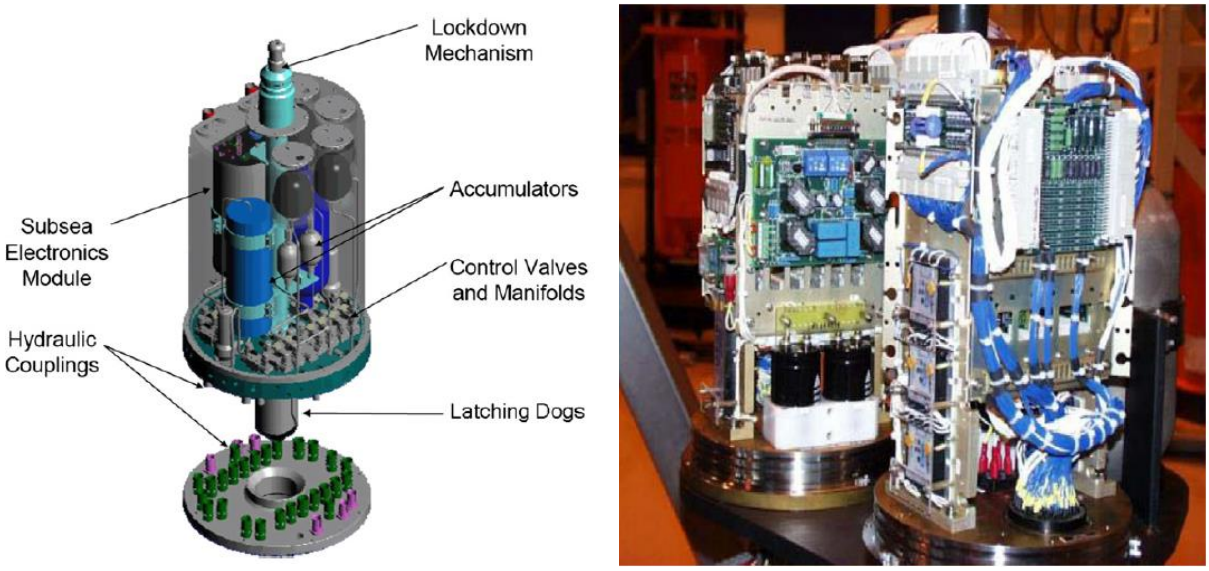
Figure 2. 13. Picture of SCM (FMC)

The working principle of the SCM is briefly explained below:

- Hydraulic supply, electrical power and electrical signal is sent from the surface facility to the SCM by umbilicals

- The subsea electronic module (SEM) decodes sent electrical signals and operates the appropriate solenoid directional control valve (DCV)
- Then the fluid is directed to the certain valve by DCV
- The SEM also encodes signals which is coming from subsea sensors and transmit them to the host facility.

Usually, two totally independent SEMs are installed in the SCM. If any of them fails, it is required to switch the control link from the failed one to the other one. Typically, this shifting operation is manually done by the control operator in the surface facility.



(a) **(b)**

Figure 2. 14. (a) Components of SCM (SUT 2008); (b) The picture of SEM (Bai and Bai 2018)

2.8.2. Subsea Distribution Units

Subsea distribution units (SDU) are designed to allocate supplied electrical power, electrical signals, supplied hydraulics and chemical injections to the appropriate subsea equipment (e.g. Christmas trees, manifolds, etc.). SDUs are installed on SDU frame which is typically made from carbon steel. However, it is also possible to install the SDU on a basic protective frame, on a monopile or on a mudmat. SDUs are connected with umbilicals through subsea umbilical termination assemblies (SUTA) and then distributed lines are connected to bases of SCMs via ROVs.

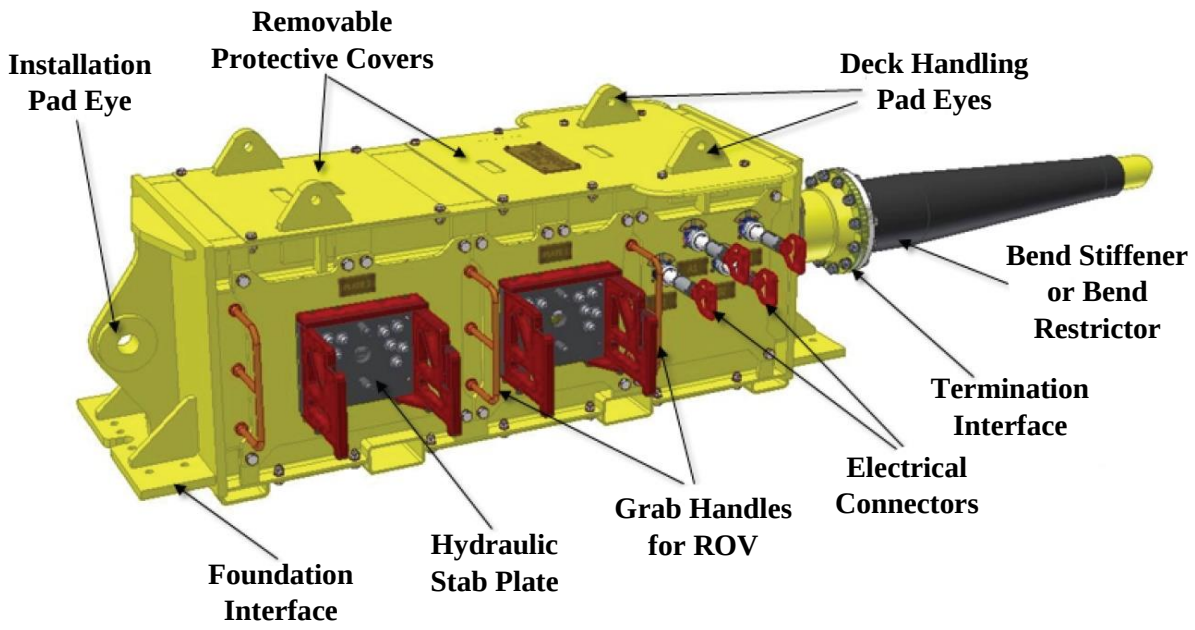


Figure 2. 15. A typical SDU (Beedle 2010)

In subsea production systems, SDUs are used for two main purposes. The first purpose is allocate umbilical functionality to numerous subsea equipment and the second one is to transmit axial load from the umbilical during installation (Beedle 2010). Depending on installation method, distribution requirements and functionality, SDU designs can be different. However, there are numerous elements which are common for all designs. These elements are described below:

1) **Primary and secondary lifting points** – Primary lifting point is used to transmit loads during final installation of the structure. Secondary lifting points are placed around the frame of the SDU and they are used to simplify manoeuvring and packing while deck handling processes.
2) **Termination interface** – Termination interface is used to attach the umbilical to the structural framework of the SDU.
3) **Bend restrictor or bend stiffener** – It is possible to encounter with excessive bending at the termination interface because the SDU structure is rigid and the umbilical is flexible. Therefore, bend restrictors are used to remove excessive bending of the umbilical.
4) **Grab handles for ROVs** – In deep water locations, divers cannot be used during installations. Therefore, ROVs are used while numerous installation processes. During these installation processes, grab handles play an anchor role to keep the ROV in the required position.

5) **Stab plates, individual connectors and functional interfaces** – Mentioned elements are used to transmit electrical power, electrical signals, supplied hydraulics and chemical injections to the secondary arrangements of the distribution (e.g. flying leads, umbilicals).

2.8.3. Umbilicals

Umbilical has bundled combination of electrical conductors, piping and tubing which is placed in armoured cover. Umbilicals connect the surface facility and the subsea production system. Umbilicals may contain steel tubes, thermoplastic hoses, fiber optic lines and electrical cables. Steel tubes are used for injection of certain fluids (e.g. methanol) to the certain subsea equipment and for monitoring certain pressures. Electrical cables are used to transfer power which is required for installed electronic devices (Nmegbu & Ohazuruike 2014). The number of tubes and conductors can change depending on umbilical complexity.

Figure 2. 16. Subsea umbilical (Collins 2008)

Integrated Production Umbilical (IPU) is a newer type of umbilical design which merges flowlines and umbilicals into single line (Figure 2. 17) (Heggdal 2005). IPU is designed to minimize the cost of flowlines and umbilicals. IPUs may contain following features:

- Flowlines (from 4” to 10”)
- Thermal insulation
- Thermal monitoring
- Electrical heating
- Power cables
- Parts for stress relieving to allow installations of electrical cables in the locations with high sea depth

- All control umbilical functions

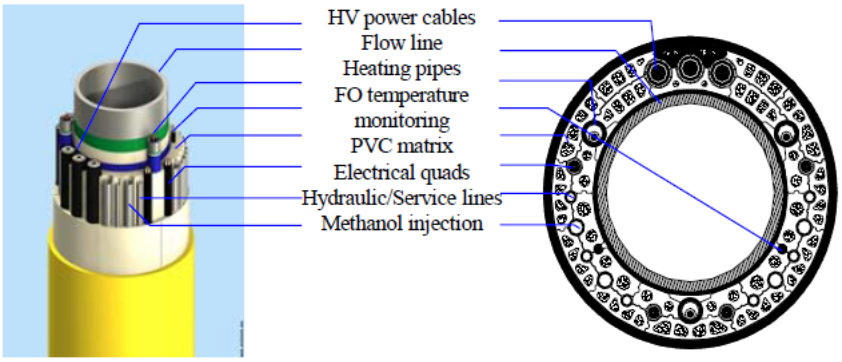


Figure 2. 17. IPU (Heggdal 2005)

Furthermore, several researches are ongoing which aim to optimize material of umbilicals for minimizing the umbilical cost. Copper is widely used as a raw material of conductors for conventional umbilicals. However, copper has some disadvantages, such as lower strength, corrosion, etc. Polymer Nanotube Umbilical (PNU) is suggested to be used for preventing mentioned problems. In PNUs, polymer nanotubes with ultrahigh conductivity are used and these nanotubes are made from polymer jacketed carbon (Dyke et al. 2015). However, this suggestion still needs further analysis for implementation.

2.9. Flowlines

Typically, flowlines are defined as following: "Flowlines are pipelines which are used to transmit production flow or injected gas/water between Christmas tree and the riser base" (Bai and Bai 2018). Although, there are several other definitions for flowlines, the mentioned definition will be used in this work and "flowlines" term will cover all production and injection lines between Christmas trees and the riser base.

Flowlines are made from flexible or rigid pipes. If pigging operations are expected, flowlines should be designed with crossover spools and crossover valves to make its configuration suitable for the pig circulations. Furthermore, it is possible to design flowlines with insulations for avoiding several flow assurance problems, which happen because of low temperatures (e.g. wax formation, hydrate formation, etc.). Nowadays, it is frequently required to design flowlines in a way that they can be used under high pressure and temperature (HP/HT) conditions. Hence, usage of flowlines that made of higher grade material (e.g. HP/HT grade) is rising.

Flowline routing is a crucial part of the subsea field development project because it can significantly influence success of the project (Kang & Lee 2017). It is important to design flowline routes in a way that it will be safe, economical and eco-friendly. Because of poorly

selected flowline route, costly surprises and unexpected interruptions in the operation can occur (Palmer and King 2004). Thus, it is worth to spend several days and a few thousands of dollars on a sensitive and thoughtful flowline routing in the planning stage of the project to save months and millions in the later stages of the project (Tharigopula 2019).

The first stage in the flowline routing is analysing survey data (e.g. geotechnical data, topographical data, etc.). As a second stage, a corridor is selected by engineers for the flowline routing. Then, the most suitable route is chosen in the corridor by considering following requirements:

- Requirements of authorities and third parties must be met. For example, other licenses, future platforms and wells, fishing areas, etc. should be considered during flowline routing
- The route with the shortest length should be chosen from all possible routes for saving hardware and installation costs
- Costs related to seabed intervention must be optimized. For this purpose, the number of free spans must be minimized, crossings with boulders must be minimized and so on
- Hazards and risks, such as environmental forces, sand waves, geohazards, etc. must be minimized and avoided
- The route should be designed eco-friendly. Threatened specimen must be protected, coral reefs should be safeguarded, etc.
- Safe distance from other pipelines and installations must be obtained
- In order to achieve the desired route while installation, radius of curvature on horizontal plane throughout the flowline should be more than the required minimum radius of curvature. The required minimum radius of curvature for achieving the desired route during installation are calculated by using the following equation (Lee 2009):

$$R_{min} = \frac{F * T_H}{W_s * \mu} \tag{2.1}$$

Where,

R_{min} is required minimum radius of curvature (m or ft)

F is safety factor (~ 2.0)

T_H is residual tension/horizontal bottom tension (N or lb)

W_s is submerged weight of the flowline (N/m or lb/ft)

μ is lateral soil-flowline friction factor (~0.5)

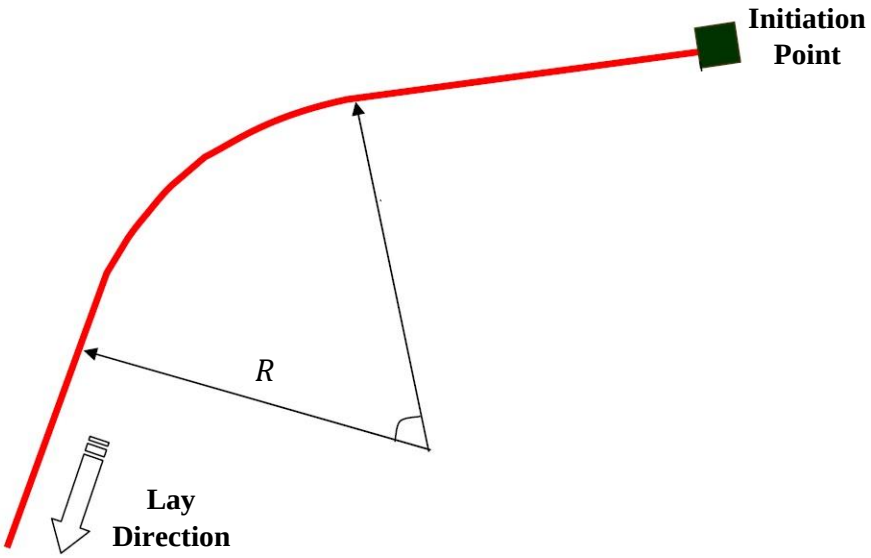


Figure 2. 18. The schematic of changing direction during flowline installation (top view) (Lee 2009)

Figure 2. 18 illustrates schematic of changing direction during flowline installation. Flowline route should be designed in a way that in each turn, R should be more than R_{min}. If this requirement is not met while changing direction of the flowline, the flowline will slide and it will not be possible to obtain desired curvature (Shatilov 2019). If it is not possible to meet this requirement, the route of the flowline must be changed or counteracts may be used as a support to obtain desired curvature (Meisingset et al. 2004). Also, there should be sufficient lengths of the straight lines before curve sections. However, flowline routes are usually designed with minimum number of turns and it is not common to see frequent turns in the flowline route. Hence, the straight-line length requirement mainly does not become a limitation in the design stage.

Flowline routing is an important topic for this thesis because one of the objectives in the thesis is achieving automatic flowline routing by using MATLAB programming language. Therefore, in the Chapter 4, flowline routing will be touched again and the tool will be created for automatic flowline routing by using MATLAB programming language.

Moreover, there are several ongoing researches with objective of optimizing the material of flowlines. For example, Steuten & Onna 2016 suggests that Thermoplastic Composite Pipes (TCP) can be used for eliminating drawbacks of using steel (e.g. heavy weight, corrosion, etc.) and decreasing the total cost of flowlines. Current trends show that TCP flowlines will frequently be used in the future.

2.10. Cost Types

The term "project cost" contains all costs which can occur during the project from the beginning of the project to the end of the project. It is required to make life cycle cost (LCC) study for determining the project cost. The LCC analysis gives an opportunity to choose the most appropriate option among all options and to make decisions which are the most profitable for the analysed project (Ribeiro et al. 1995).

LCC for subsea production systems and flowlines can be split into four types of the cost. These costs are CAPEX, OPEX, RAMEX and RISEX. So, it is possible to define LCC as following:

$$LCC = CAPEX + OPEX + RAMEX + RISEX \tag{2.2}$$

All mentioned types of the cost will be briefly explained below:

1) **CAPEX** – CAPEX means capital expenditures. CAPEX covers installation and hardware costs of the subsea production system and flowlines. Furthermore, expenditures for commissioning and testing of subsea equipment are also taken as CAPEX.
2) **OPEX** – OPEX means operational expenditures. OPEX includes costs of scheduled maintenance operations and costs of planned intervention works for recompletions.
3) **RAMEX** – RAMEX means reliability, availability and maintainability expenditures. RAMEX contains costs which occur because of equipment failure during life cycle of the project. When subsea equipment fails, two different types of the cost may appear. The first one is expenditures for the maintenance of the equipment and the second one is the cost of the lost production. Thus, RAMEX can be defined as following:

$$RAMEX = C_m + C_l \tag{2.3}$$

Where,

C_m is expenditure for the maintenance work

C_l is cost of the lost production

4) **RISEX** – RISEX includes costs which are linked to risk of the blowout during the life of the project. RISEX can be defined as multiplication of two components: blowout probability and consequence cost of any blowout accident. Therefore, RISEX can be defined by using following equation:

$$RISEX = P_b * C_b \tag{2.4}$$

Where,

P_b is blowout probability

C_b is blowout cost

Current trends in the petroleum industry show that the biggest and the most important element of the LCC is the CAPEX in subsea field development projects. Furthermore, it is difficult to obtain general numbers for OPEX, RAMEX and RISEX because they are changing depending on designs of well system, reservoir characteristics and operating procedures (Goldsmith et al. 2001). Also, in the stage of selecting wellhead locations, not so much information becomes available for OPEX, RAMEX and RISEX. Because of these factors, this work will only focus on CAPEX and the created cost model will only contain CAPEX of the subsea production systems and flowlines.

3. Types of Subsea Production System Configurations

There are several types of configurations that can be used for subsea production systems. Depending on operator company's approach and field characteristics, different configurations can be selected. Subsea layout selection has an important effect on CAPEX, OPEX, flexibility of the field and risk management (Kelly and Strauss 2009). In this chapter, four types of SPS arrangements will be discussed. These arrangements are template systems, clustered satellite wells systems, satellite wells systems and daisy chain systems. Pros and cons of each SPS configuration will also be discussed in the chapter. Figure 3. 1 illustrates mentioned four configurations.

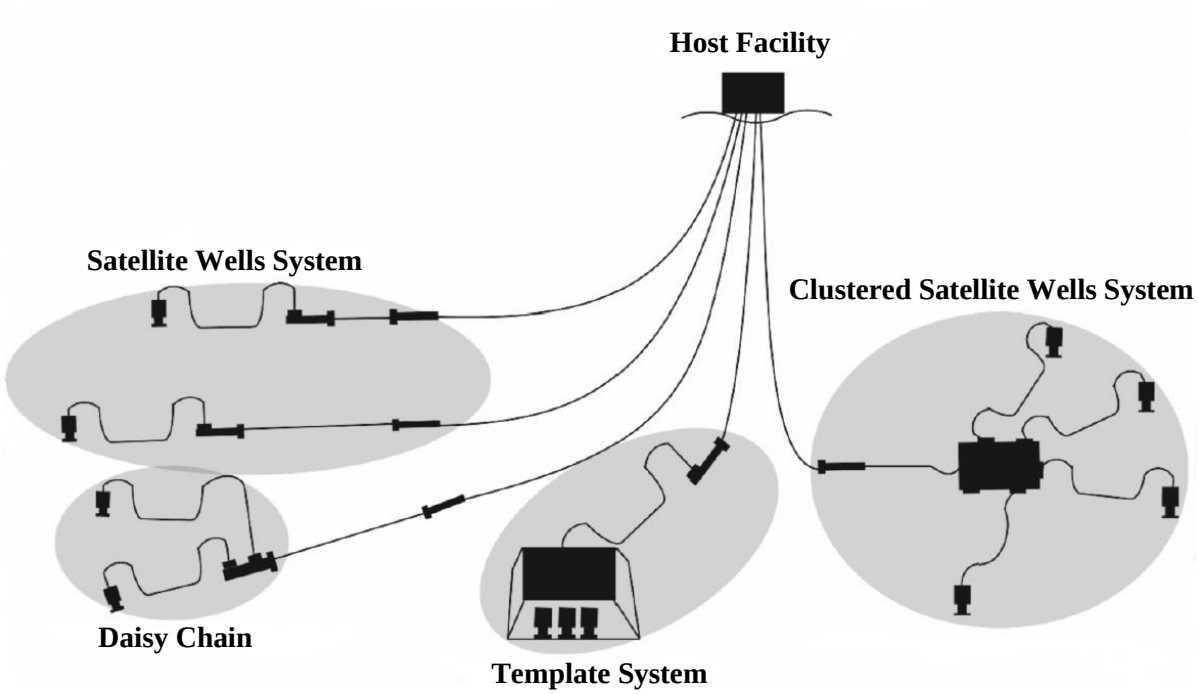


Figure 3. 1. Four different types of SPS configurations (Silva and Soares 2019)

3.1. Template System

In template configuration, subsea layout includes several subsea templates, which are installed in a few places of the field area. These subsea templates typically contain a manifold and slots for the wells. This configuration is the most common subsea layout in certain locations around the world. Having template system configuration brings some advantages to the subsea production system. However, there are also some disadvantages of selecting template system arrangement. Some advantages and disadvantages are mentioned below.

Advantages:

- Length of the flowlines is less than other configurations. Therefore, all flowline related costs are decreased.
- Because of concentrated subsea equipment, the configuration has less footprint
- Less riser connections are required in the host facility. Thus, it is possible to use riser balcony with simpler design
- Because of modularized equipment, less time is needed for installation and it is not required to frequently change location of the drilling vessel for drilling wells
- Wells are spaced precisely
- Having manifold offers flexible operability. Therefore, if any problem happens in one well, this does not affect other wells in the template

Disadvantages:

- The configuration has less flexibility for locations of wells. Because of this, it is required to drill more directional wells, hence drilling costs are increased
- Because of space limitations, ROV access is restricted
- It is risky to drill a new well while other wells are producing. Therefore, sometimes it is required to stop the production and this means the production lost.
- There is higher probability to encounter with subsurface instability problems while using heavy templates
- Manifolds typically are costly, bulky, and complex equipment and they can require specialized installation resources and construction

3.2. Clustered Satellite Wells System

In clustered satellite wells systems, the planned satellite wells are divided into groups that each group typically contains two or more than two wells and one manifold is installed for each group of the wells. This configuration has some similarities with template system because template system is also another type of clustered system. In spite of those similarities, the clustered satellite wells system has its own advantages and disadvantages. Some advantages and disadvantages of clustered satellite well systems are mentioned below.

Advantages:

- Because of clustered design, length of flowlines is less than satellite wells system
- This configuration has higher flexibility for locations of wells than template system

- Less riser connections are required in the host facility. Thus, it is possible to use riser balcony with simpler design
- Having manifold offers flexible operability. Therefore, if any problem happens in one well, this does not affect the other wells in the cluster
- Because of concentrated subsea equipment, the configuration has less footprint
- Easily accessible by ROVs

Disadvantages:

- Manifolds typically are costly, bulky, and complex equipment and they can require specialized installation resources and construction
- If subsea wells are not placed closely, it will be required to frequently change location of the drilling vessel

3.3. Satellite Wells System

In satellite wells system, subsea wells are individually tied-back to the host facility. It is typical to place wells far off from each other in this configuration. Satellite wells system is mostly used for fields which has smaller area and requires drilling of a few wells. In Brazil, this subsea architecture is commonly used for different fields, especially, in the pre-salt area (Buckley and Uehara 2017). The configuration has several advantages and disadvantages. They are mentioned below:

Advantages:

- The configuration has higher flexibility for locations of the wells. Hence, it is possible to drill more vertical wells and decrease drilling costs
- Because of individual flowlines and risers, it is possible to independently control each well from the topside. So, metering and flow control can be done on the surface facility
- Easily accessible by ROVs

Disadvantages:

- High number of risers requires riser balcony with complex design.
- Wells are placed far off from each other. Therefore, it is required to frequently change location of the drilling vessel
- Length of flowlines is very high. Therefore, all flowline related costs are increased. Also, individual flowlines make seabed congested, especially area around risers. It is

more difficult to prevent crossing of flowlines. Avoiding crossings results even longer flowlines

3.4. Daisy Chain System

In daisy chain systems, numerous satellite wells are connected to the common trunk flowline. The trunk line can be connected to the wells by jumpers or it can be connected directly to the wells' flow bases. Depending on flowrates of the wells and size of the trunk flowline, number of trunk lines and number of wells on each trunk line are changing. Each line from different subsea wells are connected to the trunk line by using in-line tees. Choke valves are installed in each subsea well to control pressures of flowlines from each well and to avoid possible pressure imbalances among the wells. It is common technique to design the trunk flowline in a way that it begins from the riser base, collects production from several wells and then ends at the riser base again. This technique allows to create a loop for pigging operations. Daisy chain configuration is mainly used in small fields or in medium-sized fields (Wang et al. 2014). Some advantages and disadvantages of daisy chain system are mentioned below:

Advantages:

- The configuration has higher flexibility for locations of the wells. Hence, it is possible to drill more vertical wells and decrease drilling costs
- Length of flowlines is less compared to satellite wells system. Hence, all flowline related costs are less than satellite wells system
- Flowlines are combined. Thus, less riser connections are required in the host facility and it is possible to use riser balcony with simpler design
- Easily accessible by ROVs

Disadvantages:

- Wells are placed far off from each other. Therefore, it is required to frequently change location of the drilling vessel
- Wells on the same trunk line are not independent from each other. This means that if any problem happens in one well, this problem can affect other wells on the same trunk line. This issue decreases system availability

In Chapter 5, mentioned subsea production system configurations will be touched again and all mentioned configurations will be implemented in the created imaginary subsea field.

4. Automatic Flowline Routing

The main objective of this work is creating a cost model for subsea production systems and flowlines which can be used in optimization of wellhead locations. All necessary preliminary information for the creation process of the desired cost model has been discussed in the previous chapters.

Some cost data (e.g. hardware and installation costs of included subsea equipment), seabed topography and coordinates of wellheads and riser base will be only inputs in the created cost model. It is obvious that routes and lengths of flowlines and umbilicals are changing while changing wellhead locations, therefore one requirement in creation process of the desired cost model is having an available tool which can be used to automatically find the most optimum routes between the wellheads and the riser base and to calculate lengths of the determined routes. So, as it was mentioned in Chapter 1, it has been decided to divide the cost model in to two elements. The first element will be a tool which can be used for determining the most optimum routes between wellheads and the riser base and for calculating lengths of the determined routes. The second element will be a spreadsheet which can be used to gather all obtained results from the first element and all available cost data for calculating total CAPEX of SPS and flowlines.

This chapter is dedicated to the creation process of the first element. For this purpose, it has been decided to use the MATLAB programming language for creating the desired tool. In the chapter, all steps, which were taken while creating the desired tool in MATLAB, will be explained separately. Also, all theories, which were used during this creation process, will be discussed.

4.1. Generation of the Seabed Topography in MATLAB

As it was mentioned in the earlier part of the chapter, one of the inputs for the cost model is the seabed topography of the certain area. In the thesis, the certain area from the North Sea, which covers 70 km x 70 km area, has been chosen for using in the case study. The same area will be used in this chapter while explaining working principle of the written MATLAB code. The selected area is shown in Figure 4. 1.

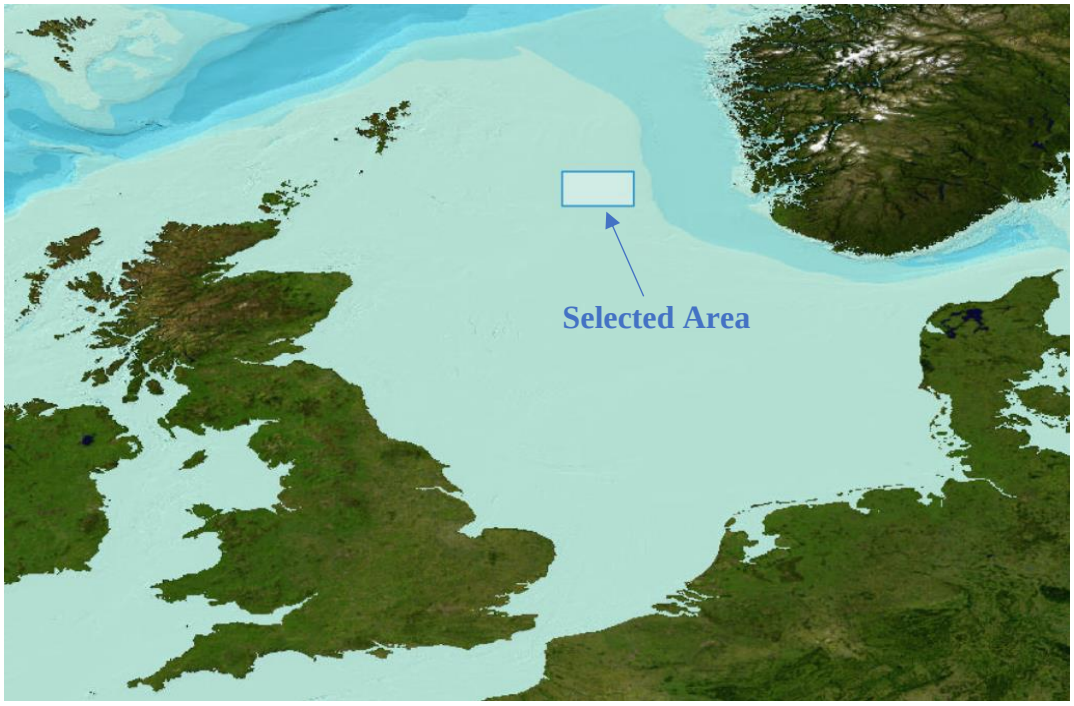
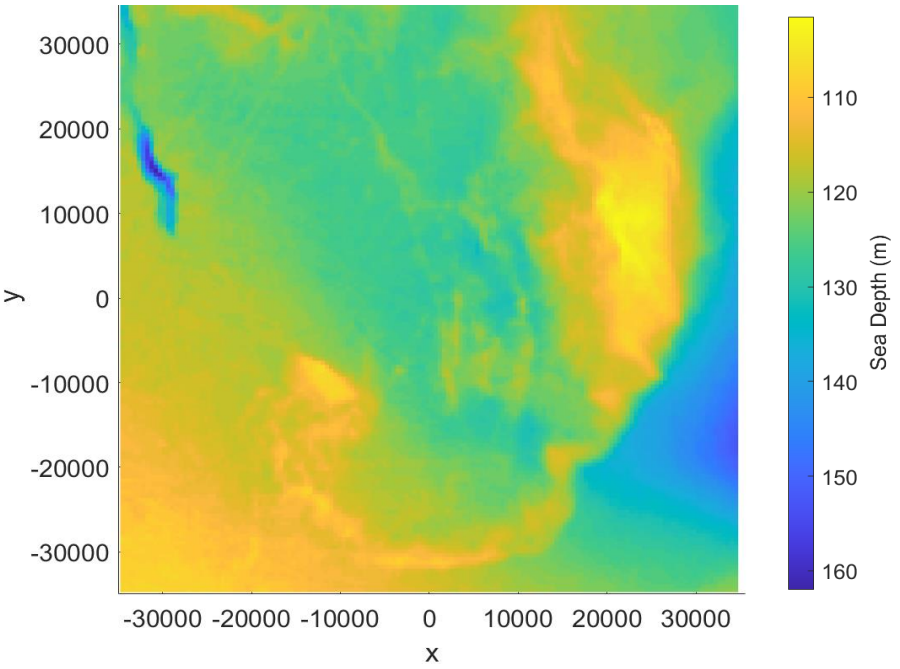


Figure 4. 1. Position of the selected area in the North Sea

The seabed topography of the selected area should be added to MATLAB in a grid format. In order to create a grid system for the seabed topography, it is required to have bathymetry data of the selected area which contains information about the water depths in the selected area. For this purpose, bathymetry data of the selected area has been taken from the General Bathymetric Chart of the Oceans (GEBCO), which is available online. The taken data has been divided into grids in a way that each grid has approximately 450 m x 450 m size and single set of co-ordinate data (x,y,z) has been assigned to each grid. In the end, generated data was plotted in MATLAB and the obtained seabed surface is shown in Figure 4. 2.

(a)



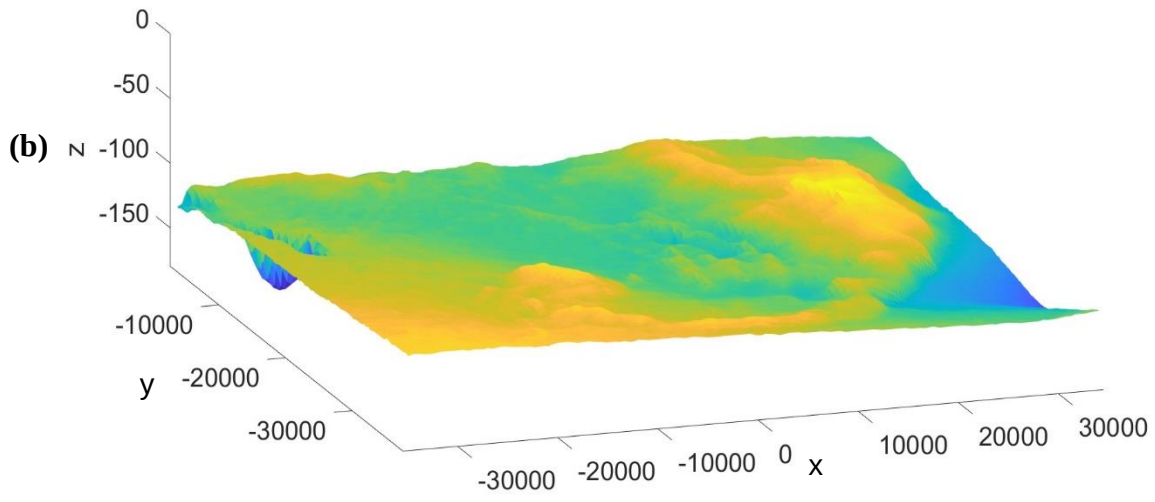


Figure 4. 2. (a) Top view of the selected area; (b) 3D view of the selected area (The graph has been zoomed in z-axis for clearly seeing sea depth change throughout the area)

As it seems from Figure 4. 2, central point of the selected area has (0, 0) x and y coordinates and each axis changes from -35000 to 35000. As the selected area covers 70 km x 70 km area, unit change of each axis is equal to 1 meter.

4.2. Determination of the Shortest Route by Using Dijkstra's Algorithm

In Chapter 2.9, importance of the flowline routing was shown and the main requirements, which should be considered while flowline routing process, were discussed. The created MATLAB codes have been designed in a way that it assumes that the inputted seabed area is already meeting most of the requirements. For example, requirements of authorities and third parties have been considered, there is not any other pipeline in the area, there is not any risk to the life of sea creatures, etc. So, the route can pass through any point of the inputted area. Therefore, the objective of the written MATLAB codes is finding the shortest route between given points (e.g. wellheads and riser base) in the selected area by taking into account minimum allowable radius of curvature requirement.

In order to achieve the mentioned objective, a certain optimisation form should be used and for this project it was decided to use least cost path (LCP) algorithm. This algorithm is used to determine the least cost paths between two points in different systems. Depending on preference of the user, the determined least cost path can be the path which requires minimum time, the path which has the shortest length, etc. The LCP algorithm is a widely used method in different industries for different purposes. For instance, it is implemented in planning of roads (Yu et al. 2003, Berglund et al 2003), it is implemented in designing of autonomous self-

driving cars (Fraichard and Ahuactzin 2001), it is implemented in creating video games, etc. In our case, it has been used for determining the shortest route between two points on the seabed.

There are several LCP algorithms, however, in this thesis the Dijkstra's algorithm (Dijkstra 1959), which is one the most known LCP algorithms, has been used for determining the shortest route. The Dijkstra's algorithm is a classic algorithm which is used for a graph search. Graphs are mathematical structures, which contain numerous nodes. These nodes are connected with each other by weighted edges (Figure 4. 3).

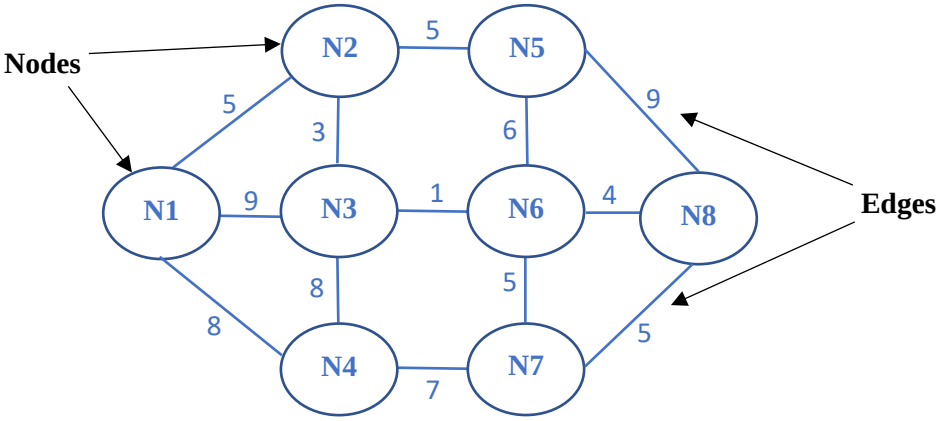


Figure 4. 3. Graph example

In Figure 4. 3. numbers on edges illustrates weights (costs) of edges. Depending on the user's purpose, these weights can be distances between nodes or required times for visiting from one node to another one or something else. The Dijkstra's algorithm determines the LCP between any selected two nodes from the graph by using weights (costs) of edges (Ando and Kimura 2012). The working principle of the Dijkstra's algorithm, which is used to find the LCP between the beginning node and the final node, is divided into 5 steps:

1) Set beginning node as a currently visited node and set all other nodes as unvisited nodes.
2) Give tentative cost values to nodes. For the beginning node set the tentative cost value to zero and set tentative cost values of all other nodes to infinity.
3) Determine total cost of the path (this path should include the current node) between the beginning node and unvisited neighbours of the current node. If the new cost value is less than the previously assigned cost value, change the tentative value of the neighbour node. Otherwise, keep the previously given tentative cost value.
4) After finishing all calculations for the current node, remove this node from unvisited nodes list and set it as a visited node. Visited nodes will not be used in future calculations.

5) If final node has been visited, then finish the algorithm. Otherwise, visit to the one of the unvisited nodes with the lowest cost value and set it as a current node. Then, repeat steps from 3 to 5.

For example, if the Dijkstra's algorithm is applied in Figure 4. 3 for finding the LCP between the nodes N1 and N8, the determined route will be N1>N2>N3>N6>N8 and the cost (e.g. distance, required time for travelling, etc.) of the path will be 5+3+1+4=13.

The pseudocode for the Dijkstra's algorithm is shown below:

Algorithm 4.1. The pseudocode for the Dijkstra's algorithm (Huang and Gartner 2012)

```
1.  function Dijkstra (Graph, beginning_node, final_node):
2.  %% Initialization of the function
3.  cost[beginning_node] = 0    %% Tentative value of the beginning node
4.
5.  for each vertex (v) of the Graph:
6.      if v ≠ beginning_node
7.         cost[v] = infinity    %% Tentative values of unvisited nodes
8.         previous[v] = undefined                %% Predecessor of v
9.
10. Unvis_nodes = all vertices of the Graph %% Set all nodes as unvisited
11.
12. while Unvis_nodes is not empty:                %% The main loop
13.     a = vertex in Unopt_nodes with the smallest tentative cost value
14.     if cost[a] = infinity break    %% It is not possible to find route
15.                                from the beginning node to the final node
16.     elseif a = final_node break    %% Final node has been visited
17.     else remove a from Unvis_nodes
18.          for each neighbour vertex (v) of a %% v must be in Unvis_nodes
19.              alternative_value = cost[a] + cost between a and v
20.              if alternative_value < cost[v] %% comparing new cost value
21.                                             and previous cost value
22.                 cost[v] = alternative_value
23.                 previous[v] = a
24.
25. %% Reading results
26. return cost[a] %% it is the total cost of the path between the
27.               beginning node and final node. If it gives infinity,
28.               there is not any possible route between these two nodes.
29. Seq = empty sequence
30. while previous[a] is defined
31.       insert a to the beginning of the Seq
32.       a = previous[a]
33. return Seq %% it is the nodes in the determined path from the beginning
34.           node to the final node
```

In our case, grids, which were obtained in the previous part of the report, are used as nodes and Euclidean distances between grids are used as edge weights. The Euclidean distance between two points is the length of a segment which connects the two points (Figure 4. 4).

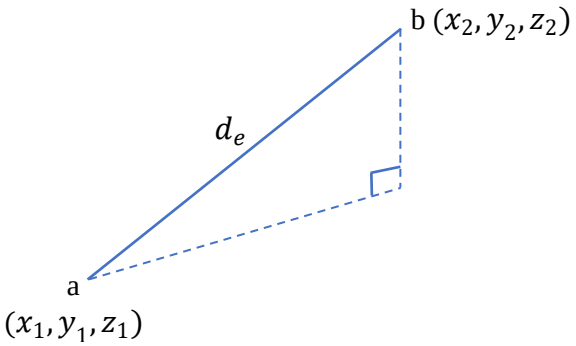


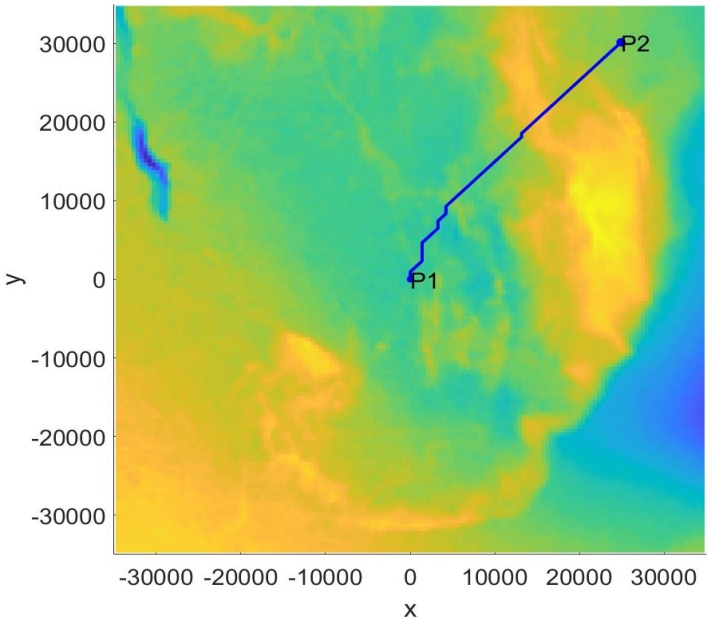
Figure 4. 4. An example of the Euclidean distance

For example, in Figure 4. 4, the Euclidean distance (d_e) between a and b is equal to:

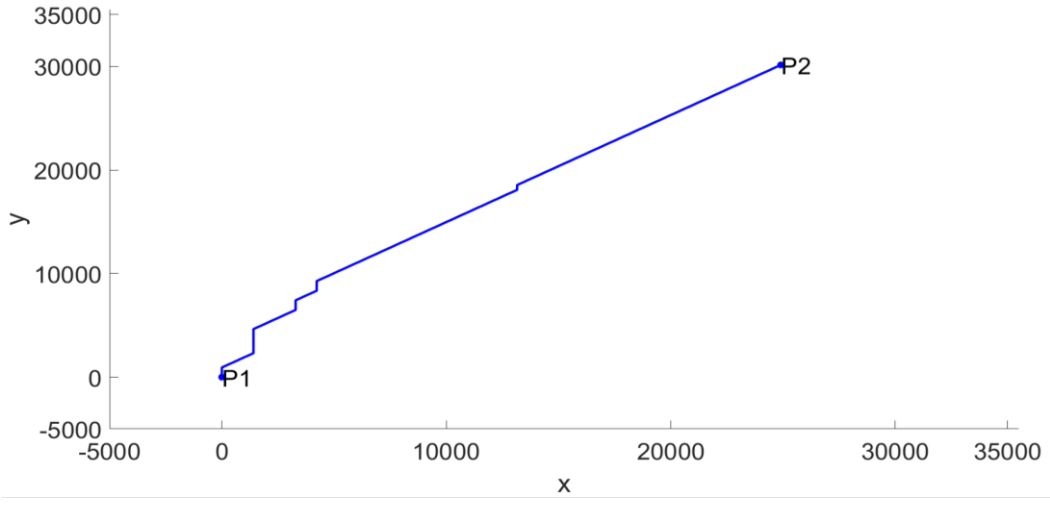
$$d_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{4.1}$$

The Dijkstra's algorithm has been applied for determining the shortest path between two points in the Figure 4. 2 with coordinates (0,0) and (25000,30000) and the determined path is shown in the Figure 4.5.

(a)



(b)



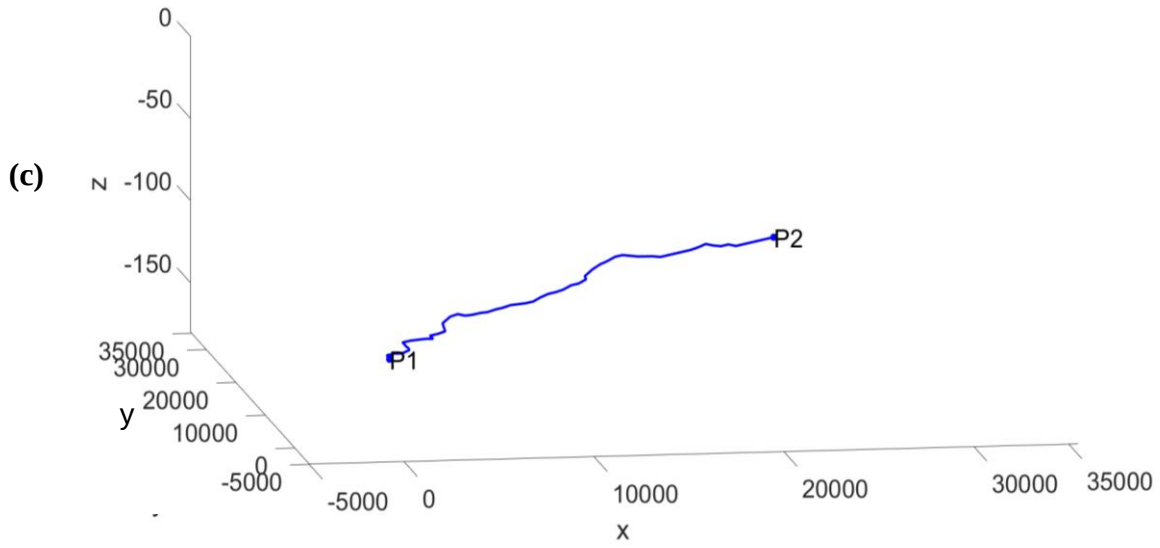


Figure 4. 5. (a) The determined shortest path with seabed map; (b) The determined shortest path w/o map; (c) 3D view of the determined shortest path (a and b are zoomed versions)

Subsea production systems usually contain more than one flowline. Therefore, the MATLAB code has been written in a way that the nodes, which are included in the determined path, are removed from the graph after determining the path and those nodes are not used while determining the next routes for other flowlines. This technique prevents possible crossings of the flowlines.

4.3. Smoothing the Determined Shortest Route

The shortest route has been found in the previous part. However, it seems from Figure 4. 5 that there are several sharp turns and bends in the determined route. As it was mentioned before, one of the most important requirements in subsea flowline routing is that radius of curvature on horizontal plane throughout the flowline should be more than the minimum allowable radius of curvature. It is important to meet this requirement for preventing certain problems while installation. Therefore, the route, which was determined by using Dijkstra's algorithm, should be smoothed by considering minimum allowable radius of curvature requirement.

In this work, the cubic spline interpolation has been chosen for the smoothing process. The objective of the cubic spline interpolation is to determine cubic splines which are the third order piecewise polynomials and pass through each of the given points (Wang 2013). This technique has an advantage that in this method, piecewise third order polynomials are determined for the given data points instead of determining one polynomial for all given data points. This feature prevents "polynomial wiggle" phenomenon and allow the user to get more precise results.

Therefore, the cubic spline interpolation is used widely in path smoothing and it has been chosen for this work too.

The fundamental working principle of the cubic spline interpolation comes from the tool of engineers which is used for drawing smooth curves that pass through numerous points. This tool contains several weights that are mounted to the flat surface at the selected points, which must be connected. Then, in order to achieve a satisfyingly smooth curve, a certain flexible strip is bent through the mounted weights. The same principle is applied in the cubic spline interpolation. In this case, the given numerical data is used instead of the points, the determined coefficients of the third order polynomials are used instead of the weights. These determined coefficients ensure that the line is bent through the given numerical points without continuity breaks or irregular behaviour (McKinley and Levine 1998).

While applying the cubic spline interpolation, if $n+1$ numerical points are given with co-ordinates $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n), (x_{n+1}, y_{n+1})$, then n splines should be determined in the following form:

$$S(x) = \begin{cases} s_1(x) & if \ \ x \in [x_1, x_2] \\ s_2(x) & if \ \ x \in (x_2, x_3] \\ & \vdots \\ s_n(x) & if \ x \in (x_n, x_{n+1}] \end{cases} \tag{4.2}$$

Where $s_i(x)$ is a third order polynomial for $i = 1, 2, 3, \dots, n$ and it is defined as following:

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{4.3}$$

As it seems from Equation 4.3, there are $4n$ coefficients that should be determined. This means that it is required to have $4n$ equations to be able to determine all coefficients.

The required equations are determined by using four properties of the cubic spline interpolation. These four properties are:

1) $S(x)$ interpolates all given numerical points
2) $S(x)$ is continuous in the $[x_1, x_n]$ interval
3) The first derivative of $S(x)$ is continuous in the $[x_1, x_n]$ interval
4) The second derivative of $S(x)$ is continuous in the $[x_1, x_n]$ interval

Firstly, it is known that each of the polynomials should pass through two of the given points:

$$
\begin{gathered}
s_1(x_1) = d_1 = y_1 \\
s_1(x_2) = a_1(x_2 - x_1)^3 + b_1(x_2 - x_1)^2 + c_1(x_2 - x_1) + d_1 = y_2 \\
s_2(x_2) = d_2 = y_2 \\
s_2(x_3) = a_2(x_3 - x_2)^3 + b_2(x_3 - x_2)^2 + c_2(x_3 - x_2) + d_2 = y_3 \\
\dots \\
s_n(x_n) = d_n = y_n \\
s_n(x_{n+1}) = a_n(x_{n+1} - x_n)^3 + b_n(x_{n+1} - x_n)^2 + c_n(x_{n+1} - x_n) + d_n = y_{n+1}
\end{gathered}
\tag{4.4}
$$

So, $2n$ equations are determined from Equation 4.4. Additional $2n$ equations should still be determined. For this purpose, the first derivatives and the second derivatives of the polynomials are used. The first derivative and the second derivative of the polynomials are:

$$s'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \tag{4.5}$$

$$s''_i(x) = 6a_i(x - x_i) + 2b_i \tag{4.6}$$

Where, $i = 1, 2, 3, \dots, n$. The first derivatives and the second derivatives of polynomials should be identical at the points in which polynomials are touching with each other. These conditions can be written as following:

$$
\begin{gathered}
s'_1(x_2) = s'_2(x_2) \\
s'_2(x_3) = s'_3(x_3) \\
\dots \\
s'_{n-1}(x_n) = s'_n(x_n)
\end{gathered}
\tag{4.7}
$$

Equation 4.7 can also be written as:

$$
\begin{gathered}
3a_1(x_2 - x_1)^2 + 2b_1(x_2 - x_1) + c_1 = c_2 \\
3a_2(x_3 - x_2)^2 + 2b_2(x_3 - x_2) + c_2 = c_3 \\
\dots \\
3a_{n-1}(x_n - x_{n-1})^2 + 2b_{n-1}(x_n - x_{n-1}) + c_{n-1} = c_n
\end{gathered}
\tag{4.8}
$$

And for the second derivative:

$$
\begin{gathered}
s''_1(x_2) = s''_2(x_2) \\
s''_2(x_3) = s''_3(x_3) \\
\dots \\
s''_{n-1}(x_n) = s''_n(x_n)
\end{gathered}
\tag{4.9}
$$

Equation 4.9 can also be written as:

$$\begin{gathered} 6a_1(x_2 - x_1) + 2b_1 = 2b_2 \\ 6a_2(x_3 - x_2) + 2b_2 = 2b_3 \\ \dots \\ 6a_{n-1}(x_n - x_{n-1}) + 2b_{n-1} = 2b_n \end{gathered} \tag{4.10}$$

Additional $2n - 2$ equations are defined from Equation 4.8 and from Equation 4.10. So, now totally $4n - 2$ equations have been defined. However, 2 more equations are still required for determining $4n$ coefficients. These 2 equations are defined by using boundary conditions. In this work, not-a-knot spline boundary conditions have been used. While using not-a-knot spline boundary conditions, it is assumed that the third derivatives of the first two polynomials are identical at the point in which they are touching with each other and the same condition is applied for the last two polynomials. This condition can be shown as following:

$$\begin{gathered} s'''_1(x_2) = s'''_2(x_2) \\ s'''_{n-1}(x_n) = s'''_n(x_n) \end{gathered} \tag{4.11}$$

Equation 4.11 can also be written as:

$$\begin{gathered} a_1 = a_2 \\ a_{n-1} = a_n \end{gathered} \tag{4.12}$$

So, $4n$ equations have been defined by using Equations 4.4, 4.8, 4.10 and 4.12. Now, $4n$ coefficients can easily be determined by solving linear system of $4n$ equations in a matrix form and n polynomials can be obtained for the intervals between $n + 1$ points.

However, there is one issue in the implementation of the cubic spline interpolation for our case. As it seems from above explanation, this method is for 2D data points with (x, y) coordinates. However, in our case, points in the path are 3D points with (x, y, z) coordinates. In order to implement the cubic spline interpolation for 3D points, one of the most common techniques has been used. The given points from the path have been correlated with the Euclidean distances between points of the path and three different cubic spline interpolation have been done for each dimension. In order to apply the mentioned technique, the determined path from Dijkstra's algorithm should be expressed by a vector function as following:

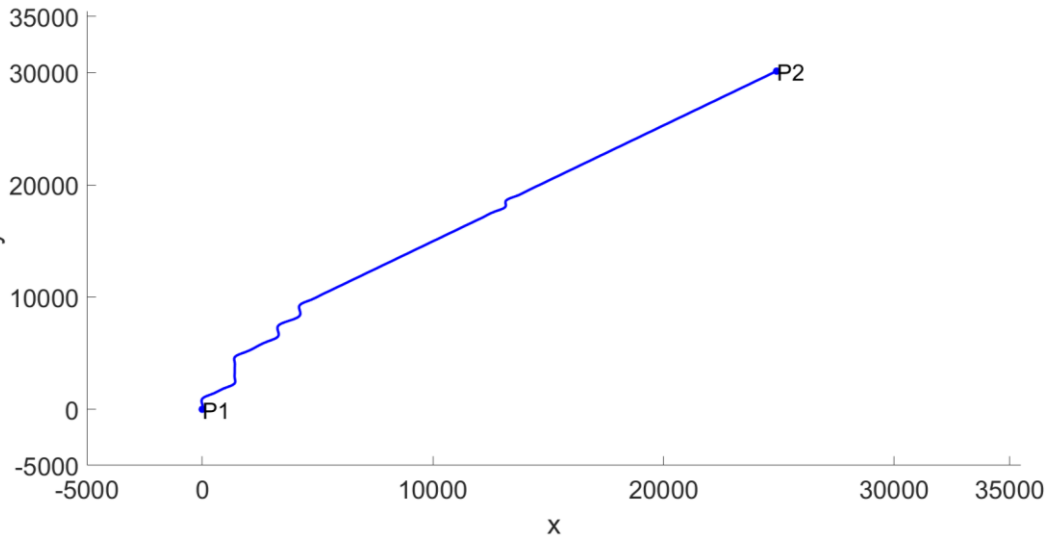
$$\vec{p}(l) = (x(l), y(l), z(l)) \tag{4.13}$$

Where, l is the sum of the Euclidean distances in the path from the first point to the certain point (the path should pass through all previous points) and it is in the interval $[0, l_{total}]$. l_{total} is the total sum of Euclidean distances between all given points.

After defining vector function for the determined path, $x(l), y(l)$ and $z(l)$ should be interpolated separately. For example, if 4 points are given from the determined path $(\vec{p}_1, \vec{p}_2, \vec{p}_3, \vec{p}_4)$ with coordinates (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4) and if Euclidean distances between $\vec{p}_1$ and $\vec{p}_2$, $\vec{p}_2$ and $\vec{p}_3$, $\vec{p}_3$ and $\vec{p}_4$ are equal to d_{12}, d_{23}, d_{34} respectively, then l_1 will be 0, l_2 will be $l_1 + d_{12}$, l_3 will be $l_2 + d_{23}$ and l_4 will be $l_3 + d_{34}$. After defining l values for the given points from the path, three sets of data can easily be created with points (l_i, x_i), (l_i, y_i), (l_i, z_i) where $i = 1, 2, 3, 4$ and 2D cubic spline interpolation can be used for interpolating each set of data for determining desired number of interpolated x, y and z values for smoothing process. Then, determined x, y and z values should be combined again as data points with (x, y, z) coordinates to express smoothed version of $\vec{p}(l)$. The only thing that should not be forgotten while using this technique is that all interpolations should be done for the same l values.

The cubic spline interpolation with the mentioned technique has been used for the determined path, which is shown in the Figure 4. 5. The new version of the path after smoothing process is shown in Figure 4. 6.

(a)



(b)

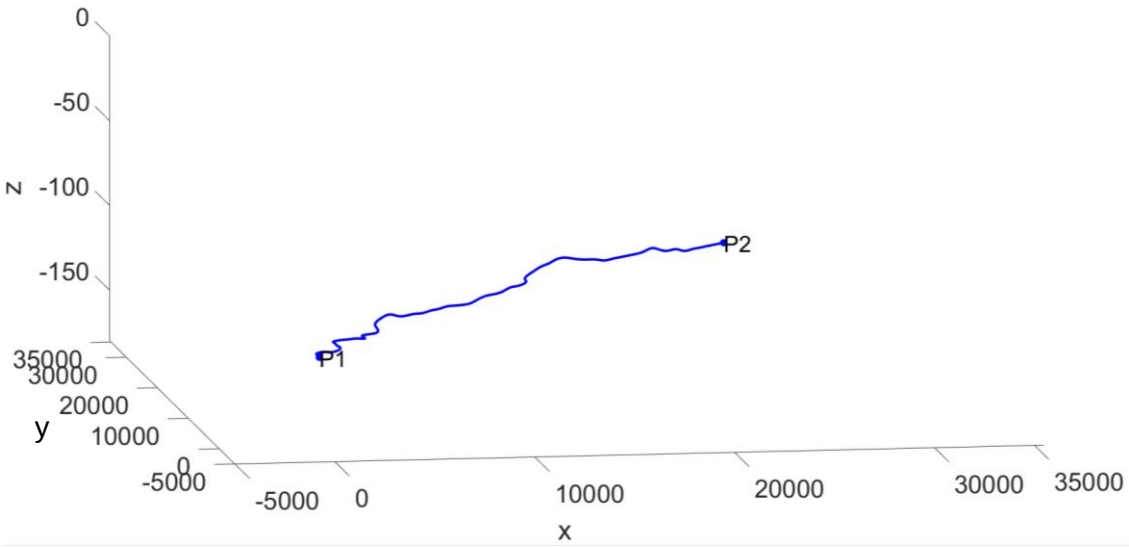


Figure 4. 6. (a) The smoothed version of the determined route; (b) 3D view of the smoothed version of the determined route

Now the route, which was determined by using Dijkstra's algorithm has been smoothed by using the cubic spline interpolation. However, as it was mentioned before, minimum allowable radius of curvature (R_{min}) requirement on horizontal plane should also be added to the smoothing process for preventing previously mentioned installation problems. For this purpose, it was decided to create an iterative process. In this iterative process, firstly, the path, which was determined by using Dijkstra's algorithm, is smoothed by using cubic spline interpolation and after smoothing process, radius of curvature is calculated for each three neighbouring points in the smoothed path. If there are some parts in which calculated radius of curvature is less than R_{min}, then midpoints are removed from those parts for increasing radius of curvature in those parts. Then, the remaining points are smoothed by using cubic spline interpolation. After that, again radius of curvature is calculated for each three neighbouring points in the new smoothed path and midpoints are removed from the parts where radius of curvature is less than R_{min}. This iterative process continues till the step when minimum radius of curvature requirement is met in all parts of the determined smoothed path and there is no need to remove any points from the determined path. The mentioned iterative process has been applied for the route which is shown in Figure 4. 5 with $R_{min} = 2000\ m$ requirement and the new smoothed version of the path is shown in Figure 4. 7.

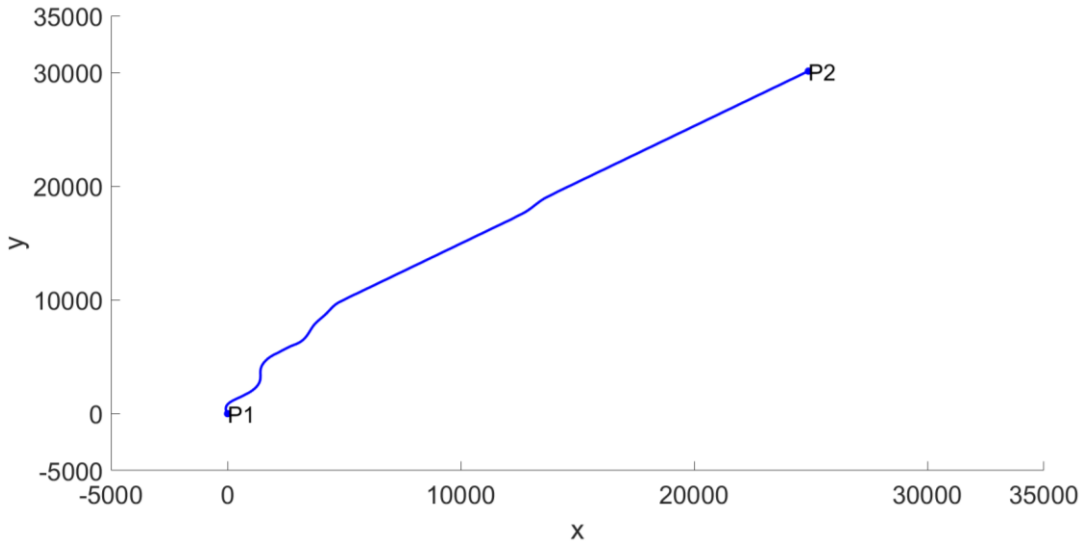


Figure 4. 7. The smoothed version of the determined route with $R_{min} = 2000\ m$

While comparing Figure 4. 7 and Figure 4. 6, it seems that there are considerable differences between the route which is obtained from the first smoothing process and the finally determined route from the iteration process.

4.4. Calculating Length of the Determined Final Route

The shortest flowline route has been determined and the determined shortest route has been smoothed by taking into account minimum allowable radius of curvature requirement. So, automatic flowline routing process has finished. However, as it was mentioned before, another output of the written code should be the length of the determined flowline path. In this part, calculation of the length of the determined route will be discussed.

As it is known from Calculus, if a 3D line is given as a vector function $\vec{r}(t) = (x(t), y(t), z(t))$, where t is in the interval $[a,\ b]$, then the length of the line (L) is equal to (Dawkins, 2007):

$$L = \int_a^b \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}\, dt \tag{4.14}$$

In our case, hundreds of points from the determined path have been obtained from the previous steps. Therefore, as it was done in the smoothing process the determined path can be expressed as $\vec{p}(l) = (x(l), y(l), z(l))$ and $x(l)$, $y(l)$, $z(l)$ polynomials can be obtained for all intervals between given points. In this case, in order to calculate the total length of the determined path (L_{total}) with n splines, following equation should be used:

$$L_{total} = \sum_{i=1}^{n} \int_{l_i}^{l_{i+1}} \sqrt{\left(\frac{dx}{dl}\right)_i^2 + \left(\frac{dy}{dl}\right)_i^2 + \left(\frac{dz}{dl}\right)_i^2}\, dl \tag{4.15}$$

$x(l)$, $y(l)$ and $z(l)$ are third order polynomials, therefore, it is better option to implement numerical integration for calculating length of each cubic spline. Gauss – Kronrod version of the Gaussian quadrature method has been selected for the numerical integration because Gaussian quadrature method is the best method for numerical integration, when functions are known and available analytically (Weisstein 2003). Gaussian quadrature method is known as an approximation method for definite integrations of functions. In order to achieve this approximation, values of the function at certain specified points are summed by using specified weights for each point (Golub and Welsch 1969). It is shown below:

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} \omega_i f(x_i) \tag{4.16}$$

Where, ω_i are weights and x_i are specified points at which $f(x)$ should be evaluated. However, Equation 4.16 is for the interval $[-1,\ 1]$. Therefore, following equation is used for changing interval of the integration to any $[a,\ b]$ interval:

$$\begin{aligned} \int_a^b f(x)dx &= \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b-a}{2}\tilde{x} + \frac{b+a}{2}\right) d\tilde{x} \\ \int_a^b f(x)dx &= \frac{b-a}{2} \sum_{i=1}^{n} \omega_i f\left(\frac{b-a}{2}\tilde{x}_i + \frac{b+a}{2}\right) \end{aligned} \tag{4.17}$$

In Gaussian quadrature method, number of weights and points (n) depends on order of the function. In order to use Gaussian quadrature method for the polynomials with $2n-1$ or less order, n number of weights and points should be selected. Gauss – Kronrod version adds additional weights and points between Gaussian weights and points for increasing precision of the numerical integration (Laurie 1997). Therefore, in this thesis, Gauss – Kronrod version of the Gaussian quadrature method has been selected for getting more precise results. As it is default for the Gauss – Kronrod integration function of the MATLAB, in the written MATLAB code 15 Gauss – Kronrod weight-point pairs, which are more precise version of the 7 Gaussian weight-point pairs, are used for the numerical integration of each step in Equation 4.15. The used weights and points are shown in Table 4. 1.

Table 4. 1. The used 15 Gauss-Kronrod weight-point pairs (Kronrod 1965)

$\tilde{x}_i$	ω_i
0	0.2094821410847278
±0.2077849550078985	0.2044329400752989
±0.4058451513773972	0.1903505780647854
±0.5860872354676911	0.1690047266392679
±0.7415311855993944	0.1406532597155259
±0.8648644233597691	0.1047900103222502
±0.9491079123427585	0.0630920926299785 5
±0.9914553711208126	0.0229353220105292 2

The mentioned techniques have been used for calculating length of the determined path between P1 (0,0) and P2 (25000, 30000) which is shown in Figure 4. 7 and the obtained length is equal to *39936 m*.

So, the required MATLAB code has been written for determining the best route between given points and for calculating length of the determined route (all scripts are shown in the Appendix). However, there is one issue in the created tool which should be mentioned. As it was mentioned before, the seabed topography data is added to the code as a grid format and it is known fact that each grid has only one x, one y and one z value. Therefore, while finding the shortest route between two points by using Dijkstra's theorem, it is possible to encounter with some unnecessary turns in the path even if the seabed is flat. This problem is illustrated in Figure 4. 8.

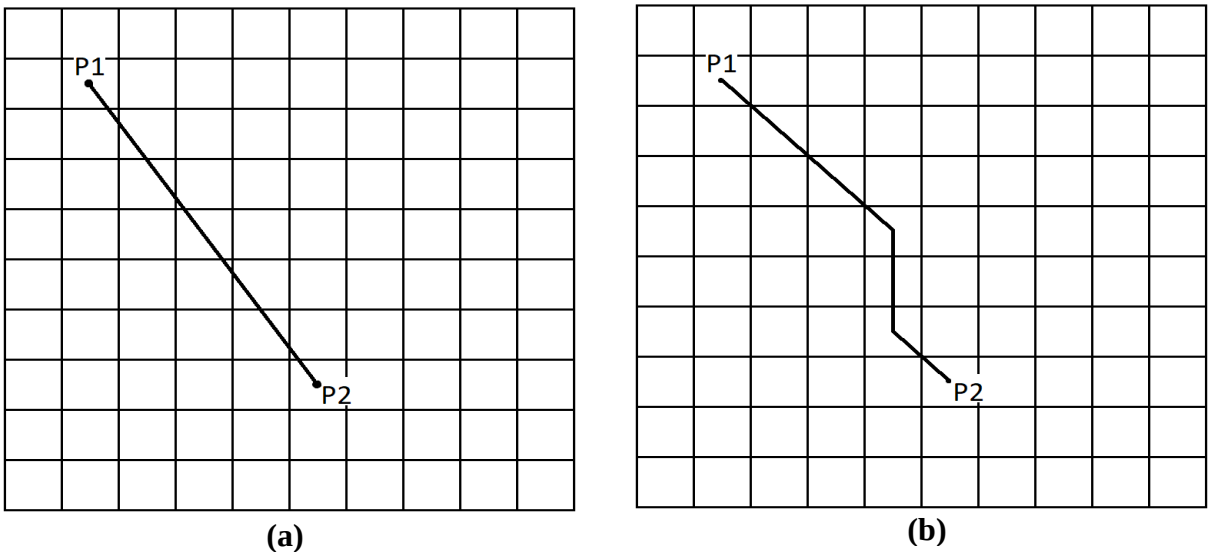


Figure 4. 8. (a) The shortest route between P1 and P2; (b) The possible route which can be determined by the written code

There are two options for solving this problem. The first and the best option is decreasing size of the grids by increasing number of grids. However, it is not possible to apply this solution in this thesis because all scripts are run in the personal computer of the author and properties of the computer limit maximum number of grids for calculations. The second option is increasing value of the minimum allowable radius of curvature (R_{min}) in the smoothing process. As it was explained before, in the smoothing process some points are removed which causes smaller radius of curvature than R_{min}. Therefore, unnecessary turns can be removed by increasing required R_{min} and if it is done, the real R_{min} requirement will still be met in the determined path. If this option is applied in the Figure 4. 7 and R_{min} is increased from $2000\ m$ to $5000\ m$, then the new path without unnecessary turns, which is shown in Figure 4. 9, will be obtained.

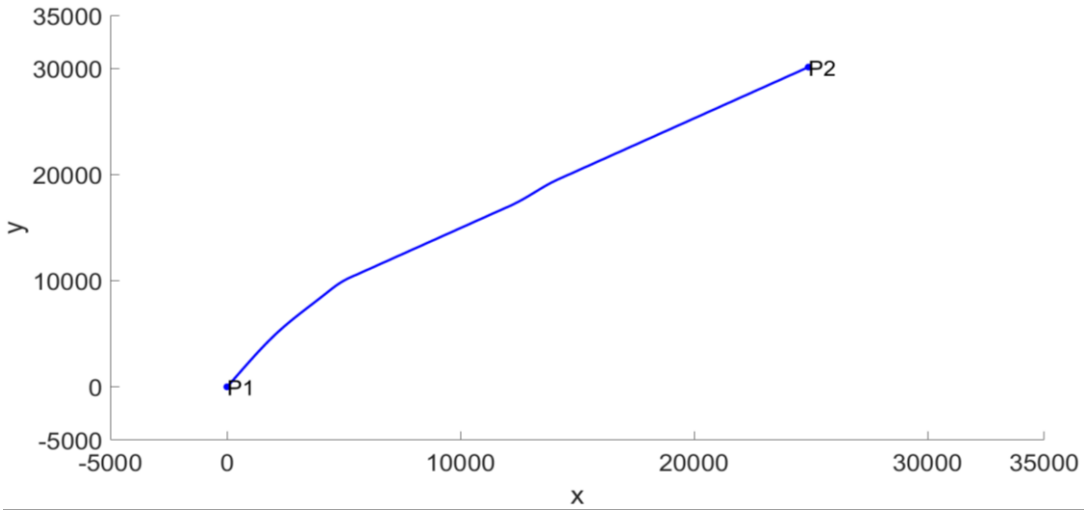


Figure 4. 9. The smoothed version of the determined route with $R_{min} = 5000\ m$

While comparing Figure 4. 9 and Figure 4. 7, it seems that unnecessary turns near P1 are removed in Figure 4. 9 and the length of the path in Figure 4.9 is *39536 m* which is 400 m less than the length of the path in Figure 4. 7. However, while implementing this option for removing unnecessary turns, the user should be sure that the turns in the path are really unnecessary turns and they are not there for avoiding certain barriers (e.g. excessive uphill etc.). In our case, there is not any excessive uphill in the selected area and as it was mentioned before, it is assumed that the flowline can pass through any point in the selected area. Hence, this option will be used in the case studies in the Chapter 5 while determining the best routes between given points. However, for the users who has access to computers, which are designed for simulation purposes, it is recommended to implement the first option and to solve this problem by decreasing size of the grids and increasing the number of the grids in the selected area.

5. Case Study – Effect of the Wellhead Placement Optimization

In the previous chapters, all parts of subsea production systems (SPS) and flowlines, which are included in the cost model, were discussed and the tool, which is the written MATLAB code, was created for determining the best routes between given points and for calculating lengths of the determined routes. So, now there is only one component is missing, which is the second element of the cost model to combine all these data and to be used for determining total CAPEX of SPS and flowlines. For creating the second element of the cost model, it has been decided to create a spreadsheet, where the desired subsea layout can be selected and all required input data and results from the MATLAB tool can be entered for calculating total CAPEX of SPS and flowlines. So, the final product, which can be used for wellhead placement optimization, contains two parts as it was mentioned before. The first part of the final product is the written MATLAB codes for four different subsea configurations, where seabed topography and locations of riser base, wellheads, manifolds or templates should be added as input data and the written script will determine the best routes between given points and the total length of the determined routes (all written codes have been added to Appendix). The second part of the final product is the spreadsheet, where type of subsea layout, amount of subsea equipment, cost values of subsea equipment and the results from MATLAB code should be added as input data and the spreadsheet gives total hardware cost, total installation cost and total CAPEX of SPS and flowlines (Some screenshots from the created spreadsheet have been added to Appendix). The created spreadsheet contains all subsea equipment which was mentioned in Chapter 2.

In this chapter, the case study will be done for the artificially created imaginary subsea field by using the created cost model. In the case study, CAPEX values will be determined for the SPS and flowlines with four different subsea layouts, which were mentioned in the Chapter 3. Moreover, importance of the wellhead placement optimization will also be analysed by changing wellhead locations in the four different subsea layouts. As there will be a lot of assumptions about costs of subsea equipment, it is not aimed to get exact results and to compare them in the case study. The main objectives of the case study are proving applicability of the created cost model, showing implementation of the created cost model in subsea fields with different subsea architectures and showing possible effect of the wellhead placement optimization on CAPEX of SPS and flowlines.

5.1. Case Data

As it was mentioned before, 70 km x 70 km area from the North Sea, which is shown in the Figure 4. 1, has been selected for the imaginary subsea field. In the created cost model, one of the required data for the written MATLAB code is the seabed topography, hence, bathymetry data has been taken from the General Bathymetric Chart of the Oceans (GEBCO) for creating seabed surface in the MATLAB as it was shown in the Chapter 4 (Figure 4. 2).

Second input data is the locations of the wellheads. For this input it has been assumed that 18 target coordinates for the reservoir are known. These target locations mean that if wells are located in these coordinates, it is possible to reach the desired targets by drilling vertical wells. The target coordinates are shown in Table 5. 1.

Table 5. 1. Target coordinates

Target №	*x*	*y*	Target №	*x*	*y*
1	-16500	-22000	10	16500	24500
2	-22000	-22000	11	22000	22000
3	-24500	-17500	12	21500	17500
4	-29000	-4000	13	27000	3000
5	-29500	0	14	30000	0
6	-26000	4500	15	29500	-4000
7	-23500	18500	16	26500	-20000
8	-22500	22500	17	23000	-23000
9	-18000	24500	18	18500	-22500

It is assumed that coordinate of the riser base is (0,0). The target points and the riser base location are shown on the seabed of the selected area in the Figure 5. 1.

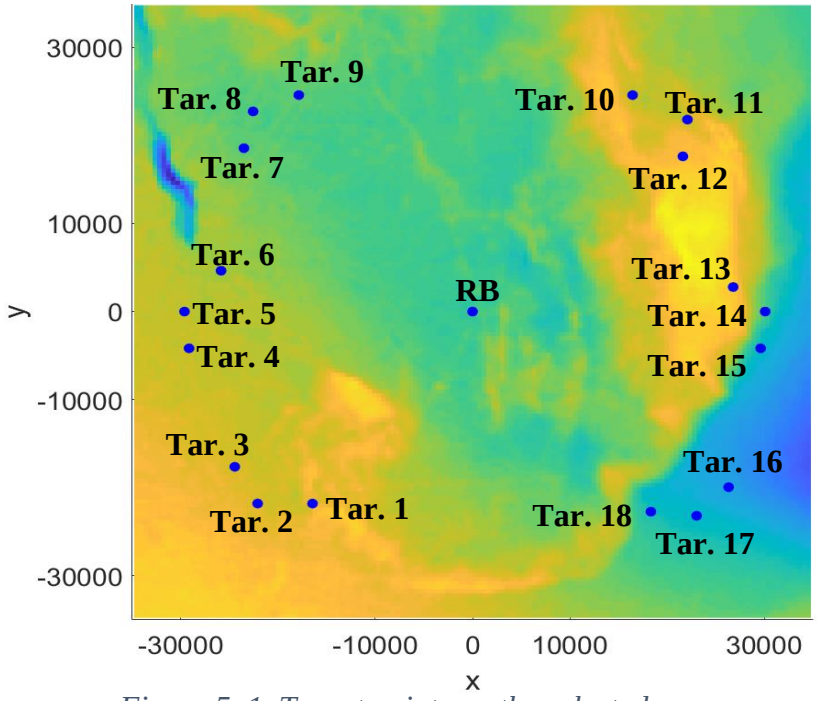


Figure 5. 1. Target points on the selected area

So, now two required inputs have been achieved and only one more input data is required for making case study calculations by using the created cost model. This input is data for the hardware and installation costs of subsea equipment. For this thesis the numbers, which are shown in the Table 5. 2 and in the Table 5. 3, are assumed as costs of subsea equipment. It is believed that these numbers can be used as representative numbers for today's operations.

Table 5. 2. Hardware costs of subsea equipment

Hardware Costs				
Properties	Value			Unit
	Min	Ave	Max	
Price of each template (4 slot ITS) including protection structure and suction anchors	21,000,000	30,000,000	39,000,000	NOK
Price of each suction anchor for manifold	2,100,000	3,000,000	3,900,000	NOK
Price of each manifold	49,000,000	70,000,000	91,000,000	NOK
Price of each protection structure for satellite wells and manifolds	2,100,000	3,000,000	3,900,000	NOK
Price of each X-mas tree with choke and multiphase meter	31,500,000	45,000,000	58,500,000	NOK
Price of each wellhead system	2,100,000	3,000,000	3,900,000	NOK
Price of each tubing hanger system	2,100,000	3,000,000	3,900,000	NOK
Price of each subsea control module	2,100,000	3,000,000	3,900,000	NOK
Price of each subsea distribution unit	3,500,000	5,000,000	6,500,000	NOK
Price of flowline per meter (rigid 6")	685	1,102	1,601	NOK/m
Price of flowline per meter (rigid 12")	1,897	2,864	3,957	NOK/m
Price of flowline per meter (rigid 16")	2,529	3,966	5,652	NOK/m
Price of umbilical per meter	9,800	14,000	18,200	NOK/m

Table 5. 3. Installation costs of subsea equipment

Installation Costs				
Properties	Value			Unit
	Min	Ave	Max	
Installation duration of each template	2	3	4	day
Installation duration of each manifold	2	3	4	day
Installation duration of each protection structure for wells and manifolds	1	1.5	2	day
Installation duration of each suction anchor	1	1.5	2	day
Installation duration of each X-mas tree	1	2	3	day
Installation duration of each SDU	1	1.5	2	day
Installation duration of each wellhead system	1	1.5	2	day
Installation duration of each tubing hanger system	1	1.5	2	day
Installation duration of flowlines	0.9	1.1	1.3	day/km
Installation duration of umbilicals	0.9	1.1	1.3	day/km
Cost of equipment installation vessel	2,250,000	2,750,000	3,250,000	NOK/day
Cost of umbilical and flowline installation vessel	2,800,000	4,000,000	5,200,000	NOK/day

Now all required input data has been obtained. So, CAPEX calculations can be done for four different scenarios in the imaginary subsea field. In each scenario, one type of mentioned subsea layouts will be used as a configuration of the subsea production system.

5.2. CAPEX of SPS and flowlines with four different layouts

5.2.1. CAPEX of SPS and flowlines with template layout

In the first scenario, it is assumed that template subsea layout has been selected for the subsea production system and flowlines. Therefore, the target points are divided into 6 groups in a way that each group contains 3 target points. Then, central points have been determined for each three-point group for placing the templates. This means that it is planned to drill 3 directional wells from the template to the 3 nearby target points. The determined locations for the templates are shown in the Figure 5. 2.

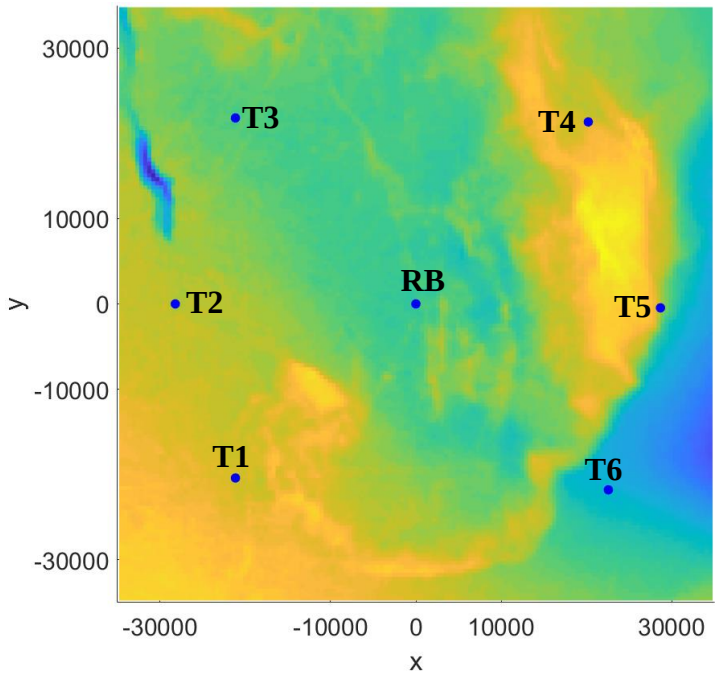
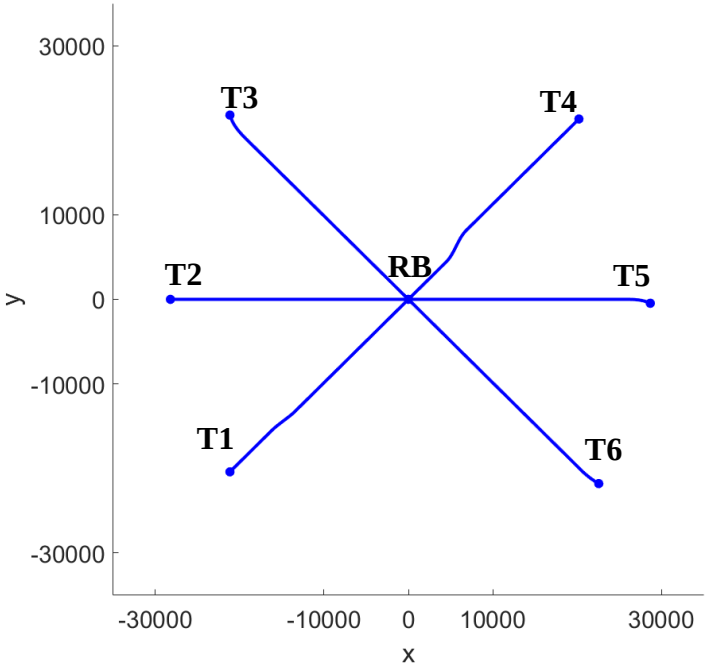


Figure 5. 2. Locations of templates

Then, the written MATLAB code has been used for determining the best flowline routes between the templates and the riser base and for calculating the total length of the determined flowline paths. The determined routes are shown in the Figure 5. 3.

(a)



(b)

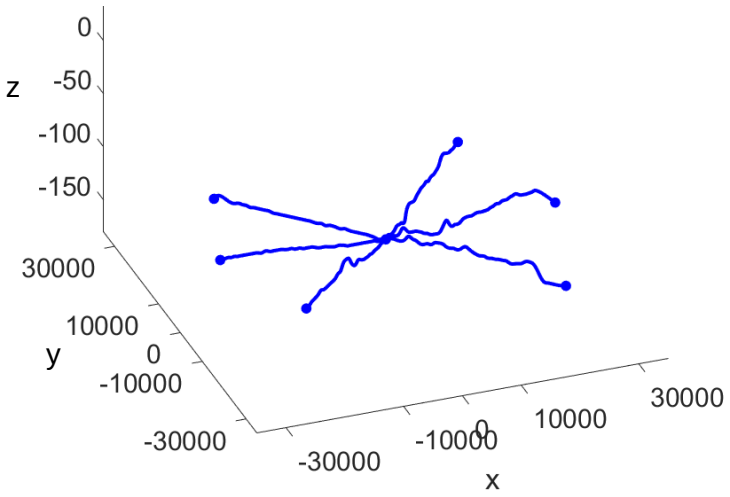


Figure 5. 3. (a) Top view of the determined flowline routes for the template layout; (b) 3D view of the determined flowline routes (Zoomed in z direction)

It has also obtained that the total length of the determined flowline paths is equal to **177631 m.** However, as it is common in today's operations, it has been assumed that two flowlines are used between each template and the riser base for pigging and testing operations. Therefore, the total length of the flowlines in the subsea system is equal to **355262 m**. Other assumptions for this scenario are shown below:

- All wells are production wells
- Foundation systems are only used for the templates and there is not any other foundation system in the SPS
- The number of subsea control modules (SCM) is equal to the number of manifolds plus the number of X-mas trees and SCMs are installed while installing manifolds and X-mas trees
- Integrated Template Structures (ITS) are used in the SPS. Hence, protection structures and foundations are installed while installing templates
- Although 4 slot ITS templates are used, 3 wells are planned to be drilled from each template and one empty slot is kept for possible future wells
- The number of Subsea Distribution Units (SDU) is equal to the number of templates
- 12" rigid pipes are used for connecting templates and the riser base
- The same paths, which were determined for the flowlines, is used for the umbilicals. So, the length of the umbilicals is equal to **177631 m**
- Same types of subsea equipment are used in the whole field

All these assumptions, the results from the written MATLAB code and the data from Table 5. 2 and Table 5. 3 have been added to the created spreadsheet for calculating the hardware cost, the installation cost and the total cost of the SPS and flowlines with template system layout in the imaginary subsea field (some screenshots from the spreadsheet have been added to the Appendix). The following results have been obtained:

Table 5. 4. Results for SPS and flowlines with template layout

	Minimum	Average	Maximum	Unit
Hardware Cost	3.55	5.12	6.75	Billion NOK
Installation Cost	1.53	2.72	4.21	Billion NOK
Total Cost	**5.08**	**7.84**	**10.96**	**Billion NOK**

5.2.2. CAPEX of SPS and flowlines with clustered satellite wells layout

In this scenario, it is assumed that manifolds are installed in the locations between each three target points (the template locations in the previous scenario). So, satellite wells are connected to the manifolds and the manifolds are connected to the riser base. In this scenario, the satellite wells are placed at the target points which means that it is required to drill only vertical wells to reach the targets in the reservoir. The well locations, the manifold locations and the riser base location is shown in the Figure 5. 4.

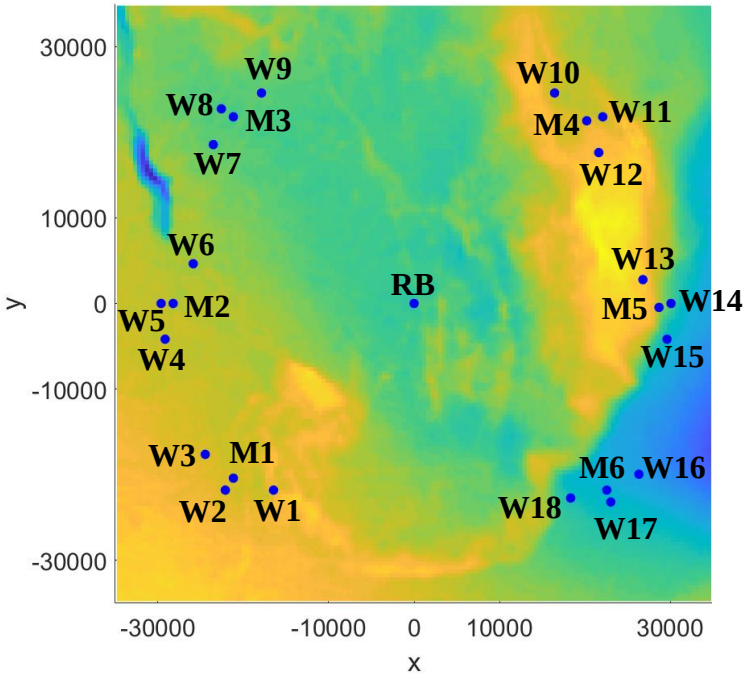


Figure 5. 4. The locations of the wells and the manifolds

Then, the written MATLAB code has been used for determining the best flowline routes between the wells and the manifolds and between the manifolds and the riser base. The determined routes are shown in the Figure 5. 5.

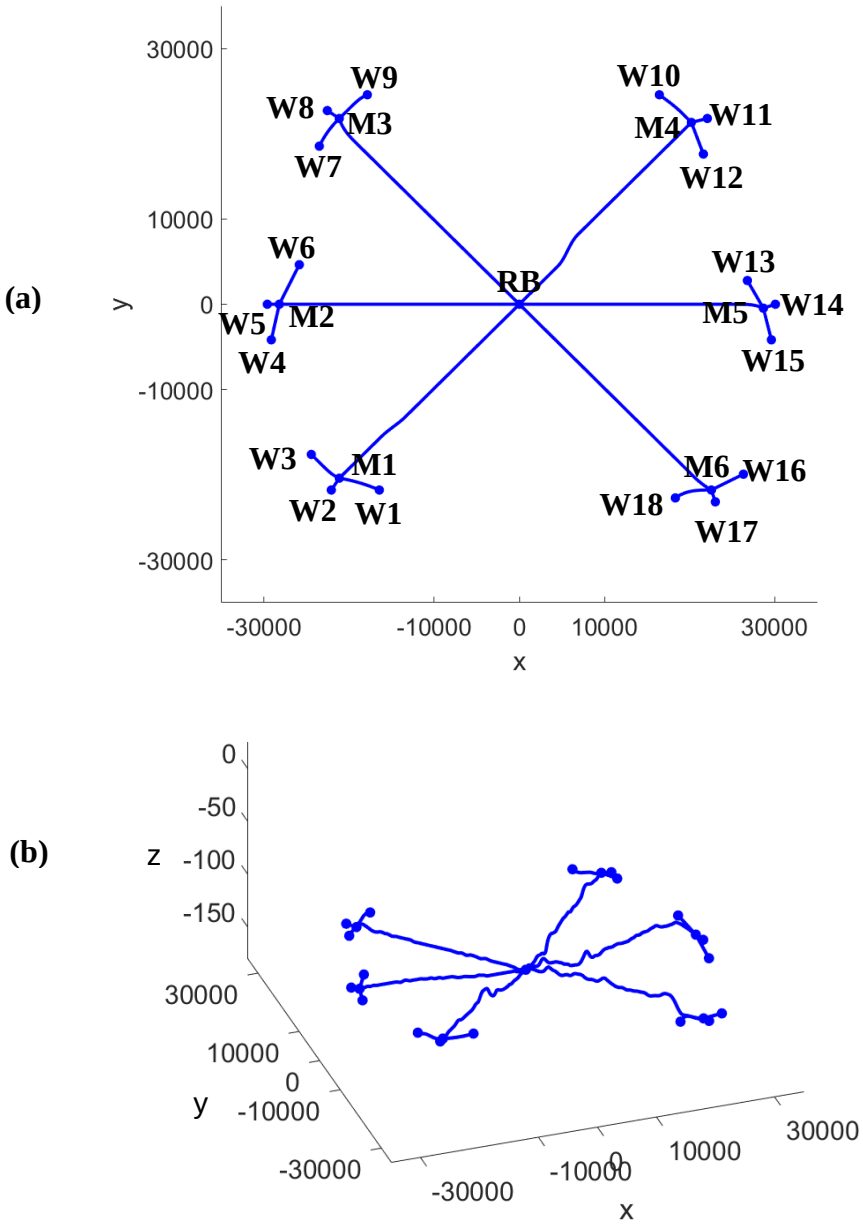


Figure 5. 5. (a) Top view of the determined flowline routes for clustered satellite wells layouts; (b) 3D view of the determined flowline routes (Zoomed in z direction)

The total length of the determined flowline paths has also been determined by the written MATLAB code. The total length of the smaller ID flowlines between the wells and the manifolds is equal to **61862 m**. The total length of the bigger ID flowlines between the manifolds and the riser base is equal to **177631 m.** However, as it is common in today's operations, it has been assumed that two flowlines are used between each manifold and the

riser base. Therefore, total length of the flowlines with bigger ID is equal to **355262 m**. Other assumptions for this scenario are shown below:

- All wells are production wells
- Foundation systems are only used for the manifolds and there is not any other foundation in SPS
- The number of subsea control modules is equal to the number of manifolds plus the number of X-mas trees. The SCMs are installed while installing manifolds and X-mas trees
- It is required to have protection structures for the manifolds and the wells. Same type of protection structure is used for the manifolds and the wells
- The number of subsea distribution units is equal to the number of the manifolds
- 12" rigid pipes are used for the bigger ID flowlines and 6" rigid pipes are used for the smaller ID flowlines
- The same paths, which were determined for the flowlines, are used for the umbilicals. So, the total length of the umbilicals is equal to **239492 m**
- Same types of subsea equipment are used in the whole field

All these assumptions, the results from the written MATLAB code and the data from Table 5. 2 and Table 5. 3 have been added to the created spreadsheet for calculating the hardware cost, the installation cost and the total cost of the SPS and flowlines with clustered satellite wells layout in the imaginary subsea field (some screenshots from the spreadsheet have been added to the Appendix). The following results have been obtained:

Table 5. 5. Results for SPS and flowlines with clustered satellite wells layout

	Minimum	Average	Maximum	Unit
Hardware Cost	4.13	5.96	7.85	Billion NOK
Installation Cost	1.88	3.34	5.16	Billion NOK
Total Cost	**6.02**	**9.30**	**13.01**	**Billion NOK**

5.2.3. CAPEX of SPS and flowlines with satellite wells layout

In this scenario, satellite wells are placed at the target points which means that it is required to drill only vertical wells to reach the targets in the reservoir and these satellite wells are connected with the riser base separately. The well locations and the riser base location is shown in the Figure 5. 6.

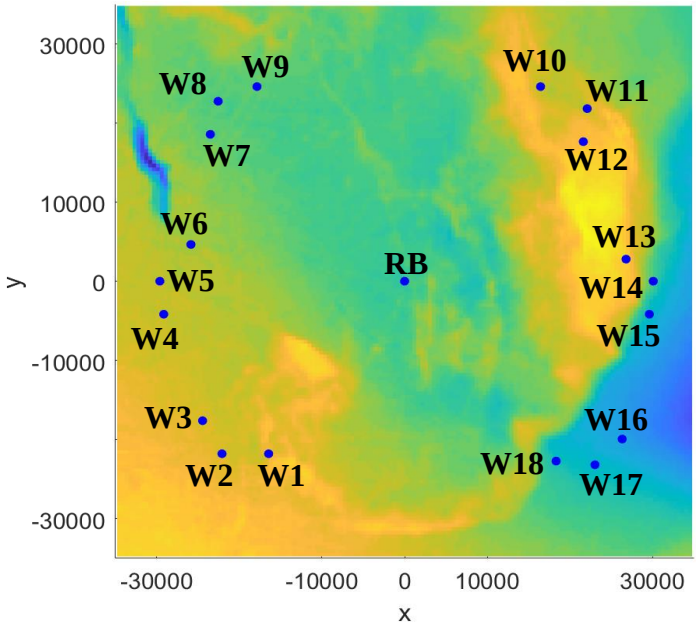
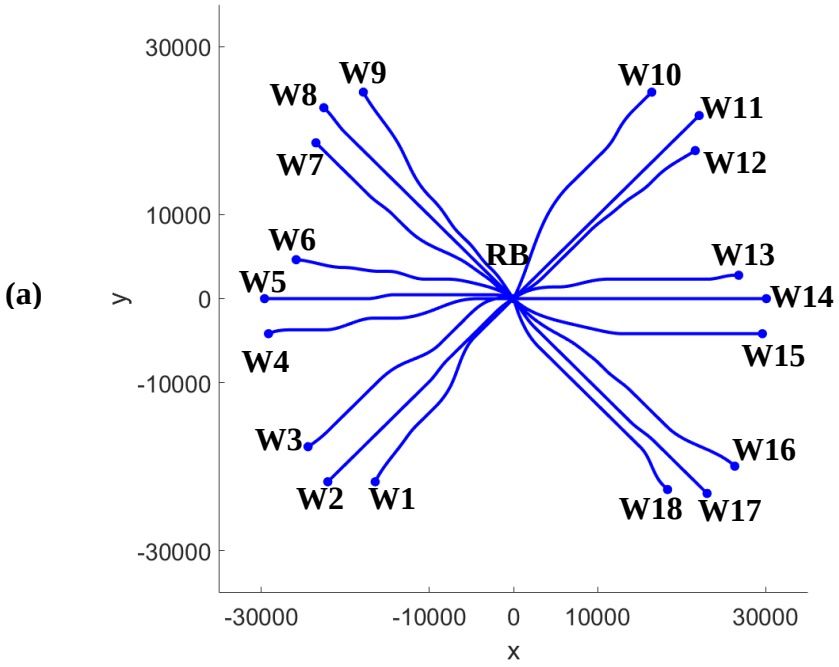


Figure 5. 6. The locations of the wells and the riser base

Then, the written MATLAB code has been used for determining the best flowline routes between the wells and the riser base. The determined routes are shown in Figure 5. 7.



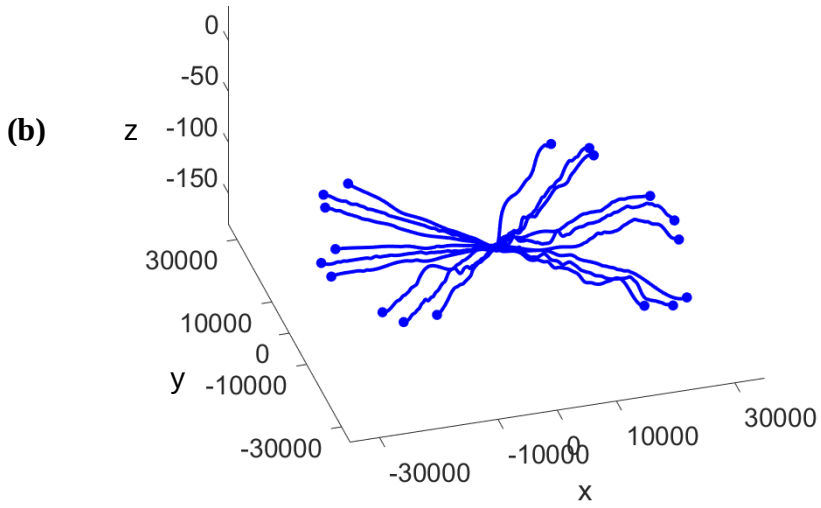


Figure 5. 7. (a) Top view of the determined flowline routes for satellite wells layout; (b) 3D view of the determined flowline routes (Zoomed in z direction)

The total length of the determined flowline paths has also been determined by the written MATLAB code. The total length of the flowlines between the wells and the riser base is equal to **540492 m**. Other assumptions for this scenario are shown below:

- All wells are production wells
- Only one flowline is used between each well and the riser base
- There is not any foundation system
- The number of subsea control modules is equal to the number of subsea X-mas trees and subsea control modules are installed while installing subsea X-mas trees
- It is required to have protection structures for the wells
- The number of subsea distribution units has been kept same with the clustered satellite wells scenario. So, the number of SDUs is equal to six
- The routes of the umbilicals have also been kept same with the clustered satellite wells scenario. Hence, the total length of the umbilicals is equal to **239492 m**
- 6” rigid pipes are used between the wells and the riser base
- Same types of subsea equipment are used in the whole field

All these assumptions, the results from the written MATLAB code and the data from Table 5. 2 and Table 5. 3 have been added to the created spreadsheet for calculating the hardware cost, the installation cost and the total cost of the SPS and flowlines with satellite wells layout in the imaginary subsea field (some screenshots from the spreadsheet have been added to the Appendix). The following results have been obtained:

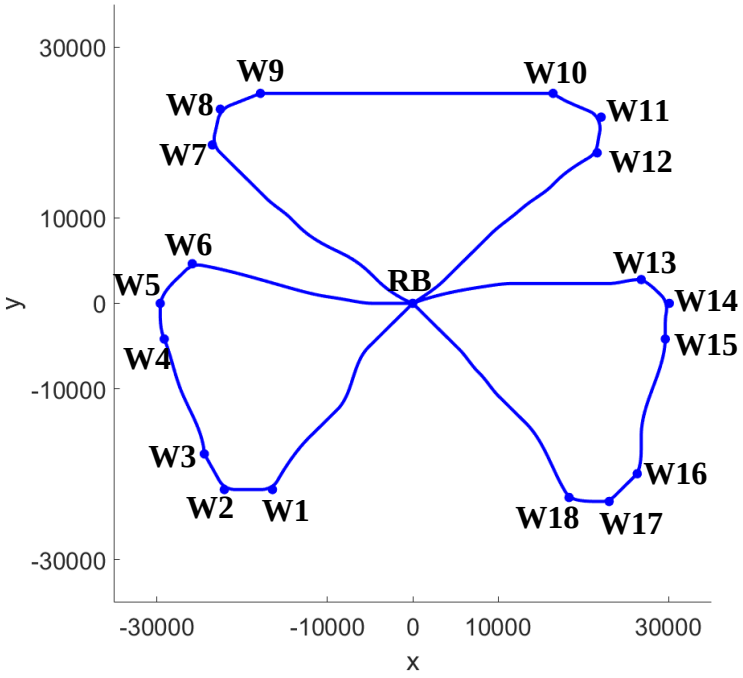
Table 5. 6. Results for SPS and flowlines with satellite wells layout

	Minimum	Average	Maximum	Unit
Hardware Cost	3.46	5.01	6.60	Billion NOK
Installation Cost	2.14	3.78	5.84	Billion NOK
Total Cost	**5.60**	**8.79**	**12.44**	**Billion NOK**

5.2.4. CAPEX of SPS and flowlines with daisy-chain layout

In this scenario, satellite wells are placed at target points again, hence, the locations of the wells are same with the previous case and has already been shown in the Figure 5. 6. However, in this case, wells are not connected with the riser base separately. In this scenario, the wells are divided into 3 groups and each group contains 6 wells. One trunk line is used for each group of the wells. These trunk lines create a loop which means that they begin from the riser base, collect production from all six wells in the certain group and end at the riser base. The written MATLAB code has been used for determining the best flowline routes for each trunk line and the determined routes are shown in Figure 5. 8.

(a)



(b)

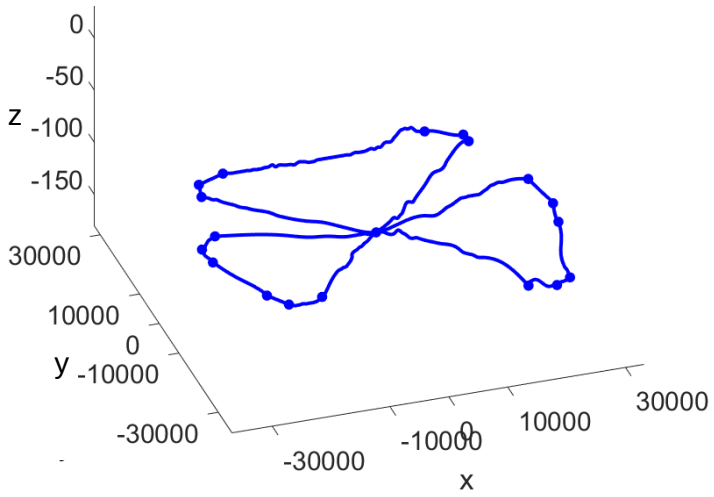


Figure 5. 8. (a) Top view of the determined flowline routes for daisy-chain layout; (b) 3D view of the determined flowline routes (Zoomed in z direction)

The total length of the determined flowline paths has also been determined by the written MATLAB code. The total length of the flowlines is equal to **289789 m**. Other assumptions for this scenario are shown below:

- All wells are production wells
- There is not any foundation system
- The number of subsea control modules is equal to the number of X-mas trees and subsea control modules are installed while installing X-mas trees
- It is required to have protection structures for the wells
- The number of subsea distribution units has been kept same with the clustered satellite wells scenario and the satellite wells scenario. So, the number of SDUs is equal to six
- The routes of the umbilicals have also been kept same with the clustered satellite wells and the satellite wells scenarios. Hence, the total length of the umbilicals is equal to **239492 m**
- One trunk flowline is used for each loop
- 16” rigid pipes are used for each trunk line
- Same types of subsea equipment are used in the whole field

All these assumptions, the results from the written MATLAB code and the data from Table 5. 2 and Table 5. 3 have been added to the created spreadsheet for calculating the hardware cost, the installation cost and the total cost of the SPS and flowlines with daisy-chain layout in the imaginary subsea field (some screenshots from the spreadsheet have been added to the Appendix). The following results have been obtained:

Table 5. 7. Results for SPS and flowlines with daisy-chain layout

	Minimum	Average	Maximum	Unit
Hardware Cost	3.82	5.56	7.37	Billion NOK
Installation Cost	1.51	2.68	4.14	Billion NOK
Total Cost	**5.33**	**8.23**	**11.51**	**Billion NOK**

So, four different subsea layouts have been used in the imaginary subsea field and CAPEX values have been determined for each scenario. These results are summarized in Table 5. 8.

Table 5. 8. CAPEX values of all scenarios

Scenario	**Total CAPEX of SPS and flowlines**			**Unit**
	Minimum	Average	Maximum	
Template	5.08	7.84	10.96	Billion NOK
Clustered Wells	6.02	9.30	13.01	Billion NOK
Satellite Well	5.60	8.79	12.44	Billion NOK
Daisy-chain	5.33	8.23	11.51	Billion NOK

Table 5. 8 shows that SPS and flowlines with template layout has the minimum CAPEX for the imaginary subsea field. However, it should not be forgotten that in other three scenarios the wells are vertical wells but in template layout scenario the wells are directional wells. This means that drilling costs are higher in the template layout scenario than other three scenarios. So, in order to select the best scenario for the imaginary subsea field drilling costs should also be taken into account.

Moreover, it is also worth to mention that these results have been obtained by using assumed cost numbers, therefore, the results cannot be representative for all situations. As it was mentioned before, the main objective of this case study is not to compare the different layouts. The main objective of this case study is showing usage of the cost model and proving that the created cost model, which is a combination of the written MATLAB code and the created spreadsheet, can be used for calculating total CAPEX of SPS and flowlines with different subsea production system architectures.

5.3. Effect of the wellhead placement optimization

As it was mentioned several times in the previous parts of the thesis, the main objective of this thesis is creating a cost model for SPS and flowlines which can be used in the wellhead

placement optimization. The desired cost model has been created in the previous parts of the thesis and the created cost model has been used to calculate total CAPEX in the imaginary field by using different subsea layouts. In this part of the thesis, locations of the wells will be changed to show influence of wellhead location selection process on CAPEX of SPS and flowlines.

In order to show effect of wellhead placement optimization, the wellhead locations, the manifold locations and the template locations have been changed in a way that they become 2000 m closer to the riser base. The new locations of the wells, the manifolds and the templates are shown in Figure 5. 9.

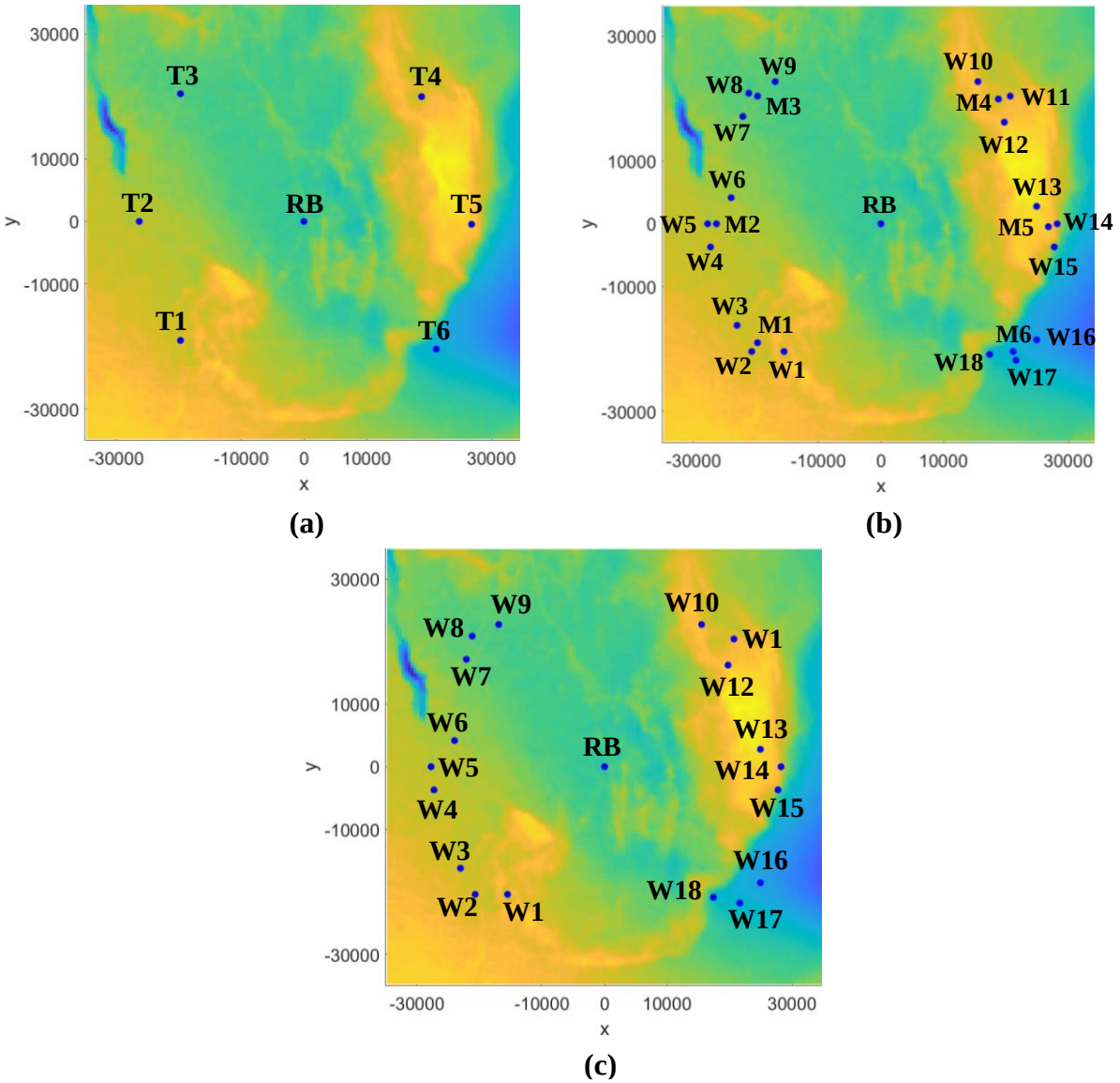


Figure 5. 9. (a) The new template locations for template layout; (b) The new locations of the wells and the manifolds for clustered wells layout; (c) The new locations of the satellite wells for satellite wells layout and daisy-chain layout

All other parameters have been kept same as they were in the previously mentioned scenarios. Then, same sequence of calculations has been done for determining new average CAPEX values for each subsea layout. The new results and the previous results are shown in Table 5. 9.

Table 5. 9. Comparison of CAPEX values

Scenario	Ave. total CAPEX of SPS and flowlines		Difference between results	Unit
	Previous results	New results		
Template	7.84	7.45	0.39	Billion NOK
Clustered Wells	9.30	8.82	0.48	Billion NOK
Satellite Well	8.79	8.30	0.49	Billion NOK
Daisy-chain	8.23	7.77	0.46	Billion NOK

Table 5.9 shows that CAPEX of SPS and flowlines can be decreased 400 – 500 million NOK, by decreasing distance 2000 m between the riser base and the wells, the manifolds or the templates. These numbers are relevant for the created imaginary subsea field and for the subsea equipment with previously assumed cost numbers. Hence, effect of the wellhead placement optimization can be bigger or smaller for different situations. Especially, if more expensive flowlines (e.g. flexible flowlines), umbilicals or pipe installation vessels are used, then well placement optimization will have bigger effect on the CAPEX of the SPS and flowlines.

However, it should not be forgotten that drilling costs are increasing while placing wells in the locations which are closer to the riser base. The reason behind this is that completions targets are kept same while making this change, hence trajectories of the wells become more complex. Therefore, as it was mentioned before, wellhead placement optimization is a two-part process. While implementing the cost model, which has been created in the thesis, for the SPS and flowlines, another one should also be used for well costs and only combination of these two cost models can give the most optimum places for the wellheads.

6. Conclusion

The objective of the thesis was creating a cost model of subsea production systems and flowlines which can be used in optimization of selecting wellhead locations. It was aimed to design the cost model in a way that it can give a total cost result for SPS and flowlines by adding seabed topography, wellhead locations and costs of subsea equipment.

In order to create the desired cost model, firstly main components of SPS and flowlines, which are the most common and have big influence on total cost of the system, were determined in the thesis. Brief information about each component was also given.

Then, four parts of life cycle cost (CAPEX, OPEX, RISEX, RAMEX) were discussed in the thesis. As it is the biggest part of LCC in subsea field development projects and it is easier to define general numbers for it at the beginning of the field development project, it was decided to focus on only CAPEX in the created cost model. So, other three elements were kept out of scope.

The most common types of subsea layouts (template, clustered satellite wells, satellite wells and daisy-chain) were also analysed in the thesis. Advantages and disadvantages of each subsea production system architecture were discussed. It was shown that depending on operator company's approach and field characteristics, the best option for the subsea layout can change from field to field.

After giving all necessary preliminary information for the cost model, the process of creating the cost model for SPS and flowlines was started. It was decided that to divide the desired cost model into two elements. As seabed topography and wellhead locations are considered as input data, the first element was dedicated for identifying the most optimum routes between any given points from the subsea field. For the first element of the cost model, MATLAB programming language was used to create a tool which can be used to determine the optimum routes between any given points and to calculate lengths of the determined routes. In MATLAB code, Dijkstra's algorithm, which is one the most known algorithms for determining shortest paths, was used to determine shortest routes between given points. Then, 3D implementation of cubic spline interpolation was added to the code for smoothing the determined routes by considering minimum radius of curvature requirement for the flowlines. As a final part of the first element, the code was written to determine lengths of determined routes by using Gaussian-Kronrod quadrature method of numerical integration.

For the second element of the cost model, which was planned to be used for gathering all available data and for calculating total CAPEX of SPS and flowlines, it was decided to create a spreadsheet. The created spreadsheet is used to gather all cost data for subsea equipment, information about the planned amount of subsea equipment in the subsea field and the results from the first element of the cost model (the written MATLAB code). It also allows the user to select one of the mentioned four subsea layouts for his/her calculations. After adding all available data, the spreadsheet calculates total CAPEX of SPS and flowlines.

In order to check applicability of the created cost model, the imaginary subsea field was created in the thesis and case study was done for this field. The certain area was selected from the North Sea and seabed topography of the selected area was used as an input. The created cost model was applied in the imaginary subsea field for calculating total CAPEX of SPS and flowlines with four different subsea layouts and results were shown. The four different CAPEX values were compared for the imaginary field and it was proven that the created model can be used for determining the best subsea layout for the subsea fields. Moreover, effect of the wellhead placement optimization also was shown by changing wellhead locations in the imaginary field and the result showed that wellhead placement optimization can have important effect on total CAPEX. This analysis also proved that the created cost model can be used in wellhead placement optimization.

In conclusion, the objective of the thesis has been achieved throughout the work and applicability of the created cost model has been proven. So, the created cost model can be used for calculating total CAPEX of SPS and flowlines in different subsea fields and also for making some sensitivity analyses to check effect of different cost parameters.

Recommendations for further work. As it was mentioned in the previous parts of the thesis, it is required to have two cost models for wellhead placement optimization. One for SPS and flowline costs and one for well costs. In this thesis, the cost model has been created for the SPS and flowline costs. In order to finalize wellhead placement optimization, it is recommended to create another cost model for the well costs and to use combination of these two cost models.

Moreover, as a future work, the created cost model can be used for making several sensitivity analyses in different locations around the world for determining effect of different cost parameters on total CAPEX of SPS and flowlines.

7. References

American Petroleum Institute, Recommended Practice for Design and Operation of Subsea Production Systems, API-RP- 17A (2002)

Ando, Y., & Kimura, H. (2012). An automatic piping algorithm including elbows and bends. Journal of the Japan Society of Naval Architects and Ocean Engineers, 15, 219-226.

Bai, Y., & Bai, Q. (2018). Subsea engineering handbook. Gulf Professional Publishing.

Beedle, A. (2010, January). Current trends and design limitations of subsea control hardware. In Offshore Technology Conference. Offshore Technology Conference.

Berglund, T., Jonsson, H., & Soderkvist, I. (2003, July). An obstacle-avoiding minimum variation b-spline problem. In 2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings (pp. 156-161). IEEE.

Brinkmann, P. E., Barbler, G. J., & Rodriguez, W. (1987). Design and Installation of a 20-Slot Template in the Gulf of Mexico in 760 ft of Water. SPE Drilling Engineering, 2(02), 99-103.

Broadbent, P. A. (2010, January). Controls reliability and early life of field failure of subsea control modules. In Subsea Control and Data Acquisition (SCADA) Conference. Society of Underwater Technology.

Buckley, K., & Uehara, R. (2017, October). Subsea Concept Alternatives for Brazilian Pre-Salt Fields. In OTC Brasil. Offshore Technology Conference.

Collins, P. (2008) Subsea Production Control and Umbilicals, SUT, Subsea Awareness Course, Houston.

Copsey, N. R., & Johnson, S. E. (1993, January). Protection measures for subsea structures against external hazards. In Offshore Europe. Society of Petroleum Engineers.

Dawkins, P. (2007). Calculus III.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische mathematik, 1(1), 269-271.

Dimmock, P., Clukey, E., Randolph, M. F., Murff, D., & Gaudin, C. (2013). Hybrid subsea foundations for subsea equipment. Journal of Geotechnical and Geoenvironmental Engineering, 139(12), 2182-2192.

Dyke, C., Ecklund, M., & Horton, H. (2015, May). Ultra-High Conductivity Umbilicals: Polymer Nanotube Umbilicals. In Offshore Technology Conference. Offshore Technology Conference.

Evans, J., & McGrail, J. (2011, January). An evaluation of the fatigue performance of subsea wellhead systems and recommendations for fatigue enhancements. In Offshore technology conference. Offshore Technology Conference.

Faulk, M. (2008). FMC ManTIS (Manifolds & Tie-in Systems), SUT Subsea Awareness Course, Houston.

Fraichard, T., & Ahuactzin, J. M. (2001, May). Smooth path planning for cars. In Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164) (Vol. 4, pp. 3722-3727). IEEE.

GEBCO Compilation Group (2020) GEBCO 2020 Grid (doi:10.5285/a29c5465-b138-234d-e053-6c86abc040b9)

Goldsmith, R., Eriksen, R., Childs, M., Saucier, B., & Deegan, F. J. (2001, January). Lifecycle cost of deepwater production systems. In Offshore Technology Conference. Offshore Technology Conference.

Golub, G. H., & Welsch, J. H. (1969). Calculation of Gauss quadrature rules. Mathematics of computation, 23(106), 221-230.

Grimmett, T. T., & Startzman, R. A. (1987, January). Optimization of offshore field development to minimize investment. In SPE Hydrocarbon Economics and Evaluation Symposium. Society of Petroleum Engineers.

Heggdal, O. (2005, January). The Integrated Production Umbilical (IPU «) and Design Tools for Deep Water. In Offshore Technology Conference. Offshore Technology Conference.

Huang, H., & Gartner, G. (2012). Collective intelligence-based route recommendation for assisting pedestrian wayfinding in the era of Web 2.0. Journal of location based services, 6(1), 1-21.

Kaculi, J. (2015, May). Next generation HPHT subsea wellhead systems design challenges and opportunities. In Offshore Technology Conference. Offshore Technology Conference.

Kaculi, J. T., & Witwer, B. J. (2014, May). Subsea wellhead system verification analysis and validation testing. In Offshore Technology Conference. Offshore Technology Conference.

Kang, J. Y., & Lee, B. S. (2017). Optimisation of pipeline route in the presence of obstacles based on a least cost path algorithm and laplacian smoothing. International Journal of Naval Architecture and Ocean Engineering, 9(5), 492-498.

Kelly, T. P., & Strauss, R. H. (2009, January). Agbami Field Development; Subsea Equipment Systems, Trees, Manifolds and Controls. In Offshore Technology Conference. Offshore Technology Conference.

Kolios, A. J., Umofia, A., & Shafiee, M. (2017). Failure mode and effects analysis using a fuzzy-TOPSIS method: a case study of subsea control module.

Kronrod, A. S. (1965). Nodes and weights of quadrature formulas: Sixteen-place tables. Not Avail.

Labbé, D. F., Jan, K., Xavier, M., Tadeu, C., & Vilchez, M. (2019, October). The Remarkable Growth of Integrated Subsea Projects from 2016 to 2019. In Offshore Technology Conference Brasil. Offshore Technology Conference.

Laurie, D. (1997). Calculation of Gauss-Kronrod quadrature rules. Mathematics of Computation, 66(219), 1133-1145.

Lee, J. (2009). Introduction to offshore pipelines and risers. N/P.

Lunde, K. P. L., & Nesheim, H. W. (2017). Possibilities and Implications by Designing an Aluminium Integrated Template Structure (Master's thesis, NTNU).

McKinley, S., & Levine, M. (1998). Cubic spline interpolation. College of the Redwoods, 45(1), 1049-1060.

McWilliams, R. S., Phipps, P. J., Myres, D. J., Erickson, M. R., & Morrison, D. S. (2018, April). Compact Modular Manifold Design for Reduced Cost and Added Value Subsea Production Systems. In Offshore Technology Conference. Offshore Technology Conference.

Meisingset, H., Hove, J., & Olsen, G. (2004, January). Optimization of pipeline routes. In the Fourteenth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers.

Nmegbu, C. G., & Ohazuruike, L. V. (2014). SUBSEA TECHNOLOGY: A WHOLISTIC VIEW ON EXISTING TECHNOLOGIES AND OPERATIONS. International Journal of Application or Innovation in Engineering & Management (IJAIEM).

Palmer, A. C., & King, R. A. (2004). Subsea pipeline engineering. PennWell Books.

Paula, M. T. R., Labanca, E. L., & Childs, P. (2001, January). Subsea manifolds design based on life cycle cost. In Offshore Technology Conference. Offshore Technology Conference.

Ribeiro, E. M., Farias, M. A., & Dreyer, S. R. B. (1995, January). Equipment cost optimization. In Offshore Technology Conference. Offshore Technology Conference.

Shatilov, D. (2019). Design features of offshore facilities for South-Western Kara Sea conditions (Master's thesis, University of Stavanger, Norway).

Silva, L. M. R., & Soares, C. G. (2019). An integrated optimization of the floating and subsea layouts. Ocean Engineering, 191, 106557.

Society for Underwater Technology, Subsea Production Control, SUT, Subsea Awareness Course, 2008.

Steuten, B., & Onna, M. V. (2016, March). Reduce project and life cycle cost with TCP flowline. In Offshore technology conference Asia. Offshore Technology Conference.

Sundt, H. K., & Ali, H. (2019, April). Compact Modular Manifold–Standardization and Modularization for Low Cost in a Low Volume Market. In Offshore Technology Conference. Offshore Technology Conference.

Tharigopula, V. (2019). Survey and Routing (Presentation, Equinor, Norway)

Towers-Perkins, P. H. (1987, January 1). Protection of Subsea Installations. Society of Underwater Technology.

Wang, K. (2013). A STUDY OF CUBIC SPLINE INTERPOLATION. InSight: Rivier Academic Journal, 9(2).

Wang, Y., Wang, D., Duan, M., Xu, M., Cao, J., & Estefen, S. F. (2014). A mathematical solution of optimal partition of production loops for subsea wells in the layout of daisy chains. Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment, 228(3), 211-219.

Weisstein, E. W. (2003). Numerical Integration. https://mathworld. wolfram. com/.

Yu, C., Lee, J. A. Y., & Munro-Stasiuk, M. J. (2003). Extensions to least-cost path algorithms for roadway planning. International Journal of Geographical Information Science, 17(4), 361-376.

8. Appendix

8.1. The written MATLAB codes

Main Scripts:

The code, which is shown below, is for creating grid system of the seabed topography which will be used in later stages:

```
%Creating Grid System for Seabed Topogrophy
%Note: x, y and z tables for grids should contain same amount of data
%Also, names of imported tables should be as following:
%x table >>> xyzmapz, y table >>> xyzmapy, z table >>> xyzmapz
x = xyzmapx{:,:};
y = xyzmapy{:,:};
z = xyzmapz{:,:};
%Plotting seabed map
figure(1)
h=surf(x,y,z)
set(h,'LineStyle','none')
zlim([-180 100])
xlim([-35000 35500])
ylim([-35000 35500])
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
c = colorbar;
c.Label.String = 'Sea Depth (m)';
grid off
%Creating nodes for the Graph Object
[m,n]=size(x);
n_values=numel(z);
Nodes=zeros(n_values,3);
xt=transpose(x);
yt=transpose(y);
zt=transpose(z);
Nodes(:,1)=xt(:);
Nodes(:,2)=yt(:);
Nodes(:,3)=zt(:);
clear xt yt zt
% Calculating Euclidean distances between each nodes
dist = pdist(Nodes); %This is available MATLAB function to calculate
                     %Euclidean distances

% Converting determined data to adjacency matrix
adj_m = squareform(dist);
clear dist
adj_m=forcetoneighbour(adj_m,m,n); %This is written function for
              %removing edges between the nodes which are not neighbours
% Generating Graph object
names = cellstr(string(1:n_values));
R = graph(adj_m);
R.Nodes = array2table(Nodes,'VariableNames',{'X','Y','Z'});
```

```
R.Nodes.Name = names';
```

The code, which is shown below, is for determining the best routes between templates and riser base in template layout and calculating total length of the determined routes:

```
%Input data:
%Adding co-ordinate information for riserbase and templates
RBC = [0 0];
T1= [-21000 -20500]; T2= [-28167 166.67]; T3= [-21333 21833];
T4= [20000 21333]; T5= [28833 -333.33]; T6= [22667 -21833];
%If additional templates have been added, they should also be added to
%the below cell array
T={T1 T2 T3 T4 T5 T6};
Tn=numel(T);
%Adding minimum allowable radius of curvature
%Also, dx and dy values for the created grid system
Rmin=10000; %Adding minimum radius of curvature
dx=0.469798658*1000; %dx of one grid in meter
dy=0.463576159*1000; %dy of one grid in meter
%End of Input data
%Determining node number of given co-ordinates
x_RBC= round(abs(x(1,1)-RBC(1,1))/dx+1);
y_RBC= round(abs(y(1,1)-RBC(1,2))/dy+1);
n_RBC= round((y_RBC-1)*n+x_RBC);
x_T=cell(1,Tn);
y_T=cell(1,Tn);
n_T=cell(1,Tn);
for i=1:Tn
x_T{i}= round(abs(x(1,1)-T{i}(1,1))/dx+1);
y_T{i}= round(abs(y(1,1)-T{i}(1,2))/dy+1);
n_T{i}= round((y_T{i}-1)*n+x_T{i});
end
% Calculating shortest unsmoothed paths between templates and riserbase
% by using Dijkstra's algorithm. Also, plotting unsmoothed paths
G=R;
length_unsmoothed=0;
L_unsmoothed=cell(1,Tn);
UnsmoothedPaths=cell(1,Tn);
for i=1:Tn
    caption1 = sprintf('%d',n_T{i});
    ni_T = findnode(G,caption1);
    caption2 = sprintf('%d',n_RBC);
    ni_RBC = findnode(G,caption2);
    [P,L_unsmoothed{i}] = shortestpath(G,ni_T,ni_RBC); %available MATLAB
                                    %function for the Dijkstra's algorithm
    UnsmoothedPaths{i}(:,1)=G.Nodes{P(1,:),1};
    UnsmoothedPaths{i}(:,2)=G.Nodes{P(1,:),2};
    UnsmoothedPaths{i}(:,3)=G.Nodes{P(1,:),3};
    figure(2)
    plot3(G.Nodes{P(1,:),1},G.Nodes{P(1,:),2},G.Nodes{P(1,:),3}, 'color', 'b',
'linewidth', 2)
    hold on
    plot3(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},'b.',
'MarkerSize',  20)
    caption3 = sprintf('T%d', i);
    text(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},caption3,
'fontsize', 18)
```

```
    length_unsmoothed=length_unsmoothed+L_unsmoothed{i};
    a=numel(P);
    P(a)=[];
    P(a-1)=[];
    G = rmnode(G,P(:));
end
caption2 = sprintf('%d',n_RBC);
ni_RBC = findnode(G,caption2);
plot3(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'b.', 'MarkerSize',
20)
text(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'RB', 'fontsize', 18)
zlim([-180 180])%This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000])
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Smoothing lines to prevent slipping of flowlines while installation
for i=1:Tn
    SmoothPaths{i}=pathsmoothing(UnsmoothedPaths{i},Rmin);
                    %Written MATLAB function for smoothing process
end
%Plotting Smoothed Lines
for i=1:Tn
figure(3)
plot3(SmoothPaths{i}(:,1),SmoothPaths{i}(:,2),SmoothPaths{i}(:,3), 'color', 'b',
'linewidth', 2) ;
zlim([-180 180]) %This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000])
hold on
plot3(SmoothPaths{i}(1,1),SmoothPaths{i}(1,2),SmoothPaths{i}(1,3),'b.',
'MarkerSize',  20)
caption4 = sprintf('T%d', i);
text(SmoothPaths{i}(1,1),SmoothPaths{i}(1,2),SmoothPaths{i}(1,3),caption4,
'fontsize', 18)
end
plot3(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'b.', 'MarkerSize',
20)
text(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'RB', 'fontsize', 18)
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Calculating total length of smoothed lines
length_smoothed=0;
L_smoothed=cell(1,Tn);
for i=1:Tn    %Written function for calculating length
L_smoothed{i}=smoothedpathlength(SmoothPaths{i}(:,1),SmoothPaths{i}(:,2),SmoothP
aths{i}(:,3));
length_smoothed=length_smoothed+L_smoothed{i};
end
X = sprintf('Total length of the flowlines is equal to %d', length_smoothed);
```

```
disp(X);
```

The code, which is shown below, is for determining the best routes between wells and manifolds and manifolds and riser base in clustered satellite wells layout and calculating total length of the determined routes:

```
%Input data:
%Adding co-ordinate information for riserbase, manifolds and wells
RBC = [0 0];
M1= [-21000 -20500]; M2= [-28167 166.67]; M3= [-21333 21833];
M4= [20000 21333]; M5= [28833 -333.33]; M6= [22667 -21833];

W1= [-16500 -22000]; W2= [-22000 -22000]; W3= [-24500 -17500];
W4= [-29000 -4000]; W5= [-29500 0]; W6= [-26000 4500];
W7= [-23500 18500]; W8= [-22500 22500]; W9= [-18000 24500];
W10= [16500 24500]; W11= [22000 22000]; W12= [21500 17500];
W13= [27000 3000]; W14= [30000 0]; W15= [29500 -4000];
W16= [26500 -20000]; W17= [23000 -23000]; W18= [18500 -22500];
%If additional manifolds or wells have been added, they should also
%be added to the below cell arrays
M={M1 M2 M3 M4 M5 M6};
Mn=numel(M);
W={W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16 W17 W18};
Wn=numel(W);
%Adding minimum allowable radius of curvature
%Also, dx and dy values for the created grid system
Rmin=10000; %Adding minimum radius of curvature
dx=0.469798658*1000; %dx of one grid in meter
dy=0.463576159*1000; %dy of one grid in meter
%End of Input data
%Determining node number of given co-ordinates
x_RBC= round(abs(x(1,1)-RBC(1,1))/dx+1);
y_RBC= round(abs(y(1,1)-RBC(1,2))/dy+1);
n_RBC= round((y_RBC-1)*n+x_RBC);
x_M=cell(1,Mn);
y_M=cell(1,Mn);
n_M=cell(1,Mn);
for i=1:Mn
x_M{i}= round(abs(x(1,1)-M{i}(1,1))/dx+1);
y_M{i}= round(abs(y(1,1)-M{i}(1,2))/dy+1);
n_M{i}= round((y_M{i}-1)*n+x_M{i});
end
x_W=cell(1,Wn);
y_W=cell(1,Wn);
n_W=cell(1,Wn);
for i=1:Wn
x_W{i}= round(abs(x(1,1)-W{i}(1,1))/dx+1);
y_W{i}= round(abs(y(1,1)-W{i}(1,2))/dy+1);
n_W{i}= round((y_W{i}-1)*n+x_W{i});
end
k=0;
Wn_perM = Wn/Mn;
for i=1:Mn
    for j=1:Wn_perM
        n_crossingpoints(j+k) = n_M{i};
    end
    k=k+Wn_perM;
end
% Calculating shortest unsmoothed paths between wells and manifolds and
% between manifolds and riserbase by using Dijkstra's algorithm.
% Also, plotting unsmoothed paths
G=R;
length_unsmoothed=0;
L_unsmoothed_M=cell(1,Mn);
```

```
UnsmoothedPaths_M=cell(1,Mn);
for i=1:Mn
    caption1 = sprintf('%d',n_M{i});
    ni_M = findnode(G,caption1);
    caption2 = sprintf('%d',n_RBC);
    ni_RBC = findnode(G,caption2);
    [P,L_unsmoothed_M{i}] = shortestpath(G,ni_M,ni_RBC); %available MATLAB
                                    function for the Dijkstra's algorithm
    UnsmoothedPaths_M{i}(:,1)=G.Nodes{P(1,:),1};
    UnsmoothedPaths_M{i}(:,2)=G.Nodes{P(1,:),2};
    UnsmoothedPaths_M{i}(:,3)=G.Nodes{P(1,:),3};
    figure(2)
    plot3(G.Nodes{P(1,:),1},G.Nodes{P(1,:),2},G.Nodes{P(1,:),3}, 'color', 'b',
'linewidth', 2)
    hold on
    plot3(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},'b.',
'MarkerSize',  20)
    caption3 = sprintf('M%d', i);
    text(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},caption3,
'fontsize', 18)
    length_unsmoothed=length_unsmoothed+L_unsmoothed_M{i};
    a=numel(P);
    P(a)=[];
    P(1)=[];
    G = rmnode(G,P(:));
end
L_unsmoothed_W=cell(1,Wn);
UnsmoothedPaths_W=cell(1,Wn);
for i=1:Wn
    caption1 = sprintf('%d',n_W{i});
    ni_W = findnode(G,caption1);
    caption2 = sprintf('%d',n_crossingpoints(i));
    ni_M = findnode(G,caption2);
    [P,L_unsmoothed_W{i}] = shortestpath(G,ni_W,ni_M);
    UnsmoothedPaths_W{i}(:,1)=G.Nodes{P(1,:),1};
    UnsmoothedPaths_W{i}(:,2)=G.Nodes{P(1,:),2};
    UnsmoothedPaths_W{i}(:,3)=G.Nodes{P(1,:),3};
    plot3(G.Nodes{P(1,:),1},G.Nodes{P(1,:),2},G.Nodes{P(1,:),3}, 'color', 'b',
'linewidth', 2)
    plot3(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},'b.',
'MarkerSize',  20)
    caption3 = sprintf('W%d', i);
    text(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},caption3,
'fontsize', 18)
    length_unsmoothed=length_unsmoothed+L_unsmoothed_W{i};
    a=numel(P);
    P(a)=[];
    G = rmnode(G,P(:));
end
caption2 = sprintf('%d',n_RBC);
ni_M = findnode(G,caption2);
plot3(G.Nodes{ni_M,1},G.Nodes{ni_M,2},G.Nodes{ni_M,3},'b.', 'MarkerSize',  20)
text(G.Nodes{ni_M,1},G.Nodes{ni_M,2},G.Nodes{ni_M,3},'RB', 'fontsize', 18)
zlim([-180 180])    %This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000])
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Smoothing lines to prevent slipping of flowlines while installation
for i=1:Mn
    SmoothPaths_M{i}=pathsmoothing(UnsmoothedPaths_M{i},Rmin);
                        %Written MATLAB function for smoothing process
end
for i=1:Wn
```

```
    SmoothPaths_W{i}=pathsmoothing(UnsmoothedPaths_W{i},Rmin);
end
%Plotting Smoothed Lines
for i=1:Mn
figure(3)
plot3(SmoothPaths_M{i}(:,1),SmoothPaths_M{i}(:,2),SmoothPaths_M{i}(:,3),
'color', 'b', 'linewidth', 2) ;
zlim([-180 180]) %This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000])
hold on
plot3(SmoothPaths_M{i}(1,1),SmoothPaths_M{i}(1,2),SmoothPaths_M{i}(1,3),'b.',
'MarkerSize',  20)
caption4 = sprintf('M%d', i);
text(SmoothPaths_M{i}(1,1),SmoothPaths_M{i}(1,2),SmoothPaths_M{i}(1,3),caption4,
'fontsize', 18)
end
for i=1:Wn
plot3(SmoothPaths_W{i}(:,1),SmoothPaths_W{i}(:,2),SmoothPaths_W{i}(:,3),
'color', 'b', 'linewidth', 2) ;
plot3(SmoothPaths_W{i}(1,1),SmoothPaths_W{i}(1,2),SmoothPaths_W{i}(1,3),'b.',
'MarkerSize',  20)
caption4 = sprintf('W%d', i);
text(SmoothPaths_W{i}(1,1),SmoothPaths_W{i}(1,2),SmoothPaths_W{i}(1,3),caption4,
'fontsize', 18)
end
plot3(G.Nodes{ni_M,1},G.Nodes{ni_M,2},G.Nodes{ni_M,3},'b.', 'MarkerSize',  20)
text(G.Nodes{ni_M,1},G.Nodes{ni_M,2},G.Nodes{ni_M,3},'RB', 'fontsize', 18)
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Calculating total length of smoothed lines
length_smoothed_biggerID=0;
L_smoothed_M=cell(1,Mn);
for i=1:Mn
                %Written function for calculating length
L_smoothed_M{i}=smoothedpathlength(SmoothPaths_M{i}(:,1),SmoothPaths_M{i}(:,2),S
moothPaths_M{i}(:,3));
length_smoothed_biggerID=length_smoothed_biggerID+L_smoothed_M{i};
end
length_smoothed_smallerID=0;
L_smoothed_W=cell(1,Wn);
for i=1:Wn
L_smoothed_W{i}=smoothedpathlength(SmoothPaths_W{i}(:,1),SmoothPaths_W{i}(:,2),S
moothPaths_W{i}(:,3));
    length_smoothed_smallerID=length_smoothed_smallerID+L_smoothed_W{i};
end
X = sprintf('Total length of the smaller ID flowlines is equal to %d',
length_smoothed_smallerID);
Y = sprintf('Total length of the bigger ID flowlines is equal to %d',
length_smoothed_biggerID);
disp(X);
disp(Y);
```

The code, which is shown below, is for determining the best routes between wells and riser base in satellite wells layout and calculating total length of the determined routes:

```
%Input data:
%Adding co-ordinate information for riserbase and wells
RBC = [0 0];

```

```
W1= [-16500 -22000]; W2= [-22000 -22000]; W3= [-24500 -17500];
W4= [-29000 -4000]; W5= [-29500 0]; W6= [-26000 4500];
W7= [-23500 18500]; W8= [-22500 22500]; W9= [-18000 24500];
W10= [16500 24500]; W11= [22000 22000]; W12= [21500 17500];
W13= [27000 3000]; W14= [30000 0]; W15= [29500 -4000];
W16= [26500 -20000]; W17= [23000 -23000]; W18= [18500 -22500];

%If additional wells have been added, they should also be added to the
%below cell array
W={W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16 W17 W18};
Wn=numel(W);
%Adding minimum allowable radius of curvature
%Also, dx and dy values for the created grid system
Rmin=10000; %Adding minimum radius of curvature
dx=0.469798658*1000; %dx of one grid in meter
dy=0.463576159*1000; %dy of one grid in meter
%End of Input data
%Determining node number of given co-ordinates
x_RBC= round(abs(x(1,1)-RBC(1,1))/dx+1);
y_RBC= round(abs(y(1,1)-RBC(1,2))/dy+1);
n_RBC= round((y_RBC-1)*n+x_RBC);
x_W=cell(1,Wn);
y_W=cell(1,Wn);
n_W=cell(1,Wn);
for i=1:Wn
x_W{i}= round(abs(x(1,1)-W{i}(1,1))/dx+1);
y_W{i}= round(abs(y(1,1)-W{i}(1,2))/dy+1);
n_W{i}= round((y_W{i}-1)*n+x_W{i});
end
% Calculating shortest unsmoothed paths between wells and riserbase by
% using Dijkstra's algorithm. Also, plotting unsmoothed paths
G=R;
length_unsmoothed=0;
L_unsmoothed=cell(1,Wn);
UnsmoothedPaths=cell(1,Wn);
for i=1:Wn
    caption1 = sprintf('%d',n_W{i});
    ni_W = findnode(G,caption1);
    caption2 = sprintf('%d',n_RBC);
    ni_RBC = findnode(G,caption2);
    [P,L_unsmoothed{i}] = shortestpath(G,ni_W,ni_RBC); %available MATLAB
                                            function for the Dijkstra's algorithm
    UnsmoothedPaths{i}(:,1)=G.Nodes{P(1,:),1};
    UnsmoothedPaths{i}(:,2)=G.Nodes{P(1,:),2};
    UnsmoothedPaths{i}(:,3)=G.Nodes{P(1,:),3};
    figure(2)
    plot3(G.Nodes{P(1,:),1},G.Nodes{P(1,:),2},G.Nodes{P(1,:),3}, 'color', 'b',
'linewidth', 2)
    hold on
    plot3(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},'b.',
'MarkerSize',  20)
    caption3 = sprintf('W%d', i);
    text(G.Nodes{P(1,1),1},G.Nodes{P(1,1),2},G.Nodes{P(1,1),3},caption3,
'fontsize', 18)
    length_unsmoothed=length_unsmoothed+L_unsmoothed{i};
    a=numel(P);
    P(a)=[];
    P(a-1)=[];
    P(a-2)=[];
```

```
    P(a-3)=[];
    G = rmnode(G,P(:));
end
caption2 = sprintf('%d',n_RBC);
ni_RBC = findnode(G,caption2);
plot3(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'b.', 'MarkerSize',
20)
text(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'RB', 'fontsize', 18)
zlim([-180 180]);   %This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000]);
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Smoothing lines to prevent slipping of flowlines while installation
for i=1:Wn
    SmoothPaths{i}=pathsmoothing(UnsmoothedPaths{i},Rmin);
                    %Written MATLAB function for smoothing process
end
%Plotting Smoothed Lines
for i=1:Wn
figure(3)
plot3(SmoothPaths{i}(:,1),SmoothPaths{i}(:,2),SmoothPaths{i}(:,3), 'color', 'b',
'linewidth', 2) ;
zlim([-180 180]);   %This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000]);
hold on
plot3(SmoothPaths{i}(1,1),SmoothPaths{i}(1,2),SmoothPaths{i}(1,3),'b.',
'MarkerSize',  20)
caption4 = sprintf('W%d', i);
text(SmoothPaths{i}(1,1),SmoothPaths{i}(1,2),SmoothPaths{i}(1,3),caption4,
'fontsize', 18)
end
plot3(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'b.', 'MarkerSize',
20)
text(G.Nodes{ni_RBC,1},G.Nodes{ni_RBC,2},G.Nodes{ni_RBC,3},'RB', 'fontsize', 18)
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Calculating total length of smoothed lines
length_smoothed=0;
L_smoothed=cell(1,Wn);
for i=1:Wn
                %Written function for calculating length
L_smoothed{i}=smoothedpathlength(SmoothPaths{i}(:,1),SmoothPaths{i}(:,2),SmoothP
aths{i}(:,3));
length_smoothed=length_smoothed+L_smoothed{i};
end
X = sprintf('Total length of the flowlines is equal to %d', length_smoothed);
disp(X);
```

The code, which is shown below, is for determining the best routes between wells and riser base in daisy-chain layout and calculating total length of the determined routes:

```
%Input data:
%Adding co-ordinate information for riserbase and wells
RBC = [0 0];
W1= [-16500 -22000]; W2= [-22000 -22000]; W3= [-24500 -17500];
W4= [-29000 -4000]; W5= [-29500 0]; W6= [-26000 4500];
W7= [-23500 18500]; W8= [-22500 22500]; W9= [-18000 24500];
W10= [16500 24500]; W11= [22000 22000]; W12= [21500 17500];
W13= [27000 3000]; W14= [30000 0]; W15= [29500 -4000];
W16= [26500 -20000]; W17= [23000 -23000]; W18= [18500 -22500];

%If additional wells have been added, they should also be added to the
%below cell array
W={W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16 W17 W18};
Wn=numel(W);
%Adding minimum allowable radius of curvature and number of well in each
%loop. Also, dx and dy values for the created grid system
Rmin=2000; %Adding minimum radius of curvature
dx=0.469798658*1000; %dx of one grid in meter
dy=0.463576159*1000; %dy of one grid in meter
Wells_in_loop = 6; %It is assumed that each loop contains same number of wells
Num_loops=Wn/Wells_in_loop;
%End of Input data
%Determining node number of given co-ordinates
x_RBC= round(abs(x(1,1)-RBC(1,1))/dx+1);
y_RBC= round(abs(y(1,1)-RBC(1,2))/dy+1);
n_RBC= round((y_RBC-1)*n+x_RBC);
x_W=cell(1,Wn);
y_W=cell(1,Wn);
n_W=cell(1,Wn);
for i=1:Wn
x_W{i}= round(abs(x(1,1)-W{i}(1,1))/dx+1);
y_W{i}= round(abs(y(1,1)-W{i}(1,2))/dy+1);
n_W{i}= round((y_W{i}-1)*n+x_W{i});
end
%Dividing wells into several loops (Assuming each loop contains same number
%of wells).
Points_in_loops=cell(1,Num_loops);
for i=1:Num_loops
    Points_in_loops{i}=zeros(1,Wells_in_loop+1);
    Points_in_loops{i}(1)=n_RBC;
    for j=1:Wells_in_loop
        Points_in_loops{i}(j+1)=n_W{j+(i-1)*Wells_in_loop};
    end
    Points_in_loops{i}=[Points_in_loops{i}, n_RBC];
end
% Calculating shortest unsmoothed paths between wells and riserbase
% by using Dijkstra's algorithm. Also, plotting unsmoothed paths.
G=R;
length_unsmoothed=0;
L_unsmooth=cell(1,(Wells_in_loop+1)*Num_loops);
L_unsmoothed=zeros(1,Num_loops);
UnsmoothPaths=cell(1,(Wells_in_loop+1)*Num_loops);
UnsmoothedPaths=cell(1,Num_loops);
for i=1:Num_loops
```

```
    for j=1:Wells_in_loop+1;
    caption1 = sprintf('%d',Points_in_loops{i}(j));
    ni_1 = findnode(G,caption1);
    caption2 = sprintf('%d',Points_in_loops{i}(j+1));
    ni_2 = findnode(G,caption2);
    [P,L_unsmooth{j+(i-1)*(Wells_in_loop+1)}] = shortestpath(G,ni_1,ni_2);
                        %available MATLAB function for the Dijkstra's algorithm
    UnsmoothPaths{j+(i-1)*(Wells_in_loop+1)}(:,1)=G.Nodes{P(1,:),1};
    UnsmoothPaths{j+(i-1)*(Wells_in_loop+1)}(:,2)=G.Nodes{P(1,:),2};
    UnsmoothPaths{j+(i-1)*(Wells_in_loop+1)}(:,3)=G.Nodes{P(1,:),3};
    L_unsmoothed(i)=L_unsmoothed(i)+L_unsmooth{j+(i-1)*(Wells_in_loop+1)};
    a=numel(P);
    figure(2)
    plot3(G.Nodes{P(1,:),1},G.Nodes{P(1,:),2},G.Nodes{P(1,:),3}, 'color', 'b',
'linewidth', 2)
    hold on
    if j<Wells_in_loop+1
    plot3(G.Nodes{P(1,a),1},G.Nodes{P(1,a),2},G.Nodes{P(1,a),3},'b.',
'MarkerSize',  20)
    caption3 = sprintf('W%d', j+(i-1)*Wells_in_loop);
    text(G.Nodes{P(1,a),1},G.Nodes{P(1,a),2},G.Nodes{P(1,a),3},caption3,
'fontsize', 18)
    end
    length_unsmoothed=length_unsmoothed+L_unsmooth{i};
    P(a)=[];
    P(a-1)=[];
    P(1)=[];
    P(2)=[];
    G = rmnode(G,P(:));
    end
end
caption2 = sprintf('%d',n_RBC);
ni_2 = findnode(G,caption2);
plot3(G.Nodes{ni_2,1},G.Nodes{ni_2,2},G.Nodes{ni_2,3},'b.', 'MarkerSize',  20)
text(G.Nodes{ni_2,1},G.Nodes{ni_2,2},G.Nodes{ni_2,3},'RB', 'fontsize', 18)
zlim([-180 180]);   %This figure settings have been made for the used data
xlim([-35000 35000]); %for other data these can be changed
ylim([-35000 35000]);
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off

%Smoothing lines to prevent slipping of flowlines while installation
a=numel(UnsmoothPaths);
R=12500; %This value can be kept same with Rmin, if smaller grids are used
for i=1:a
    SmoothPaths{i}=pathsmoothing(UnsmoothPaths{i},R);
end
for i=1:Num_loops
    for j=1:Wells_in_loop+1
       if j>1
 SmoothPaths{j+(i-1)*(Wells_in_loop+1)}(1,:)=[];
 UnsmoothedPaths{i}=[UnsmoothedPaths{i};SmoothPaths{j+(i-1)*(Wells_in_loop+1)}];
       else
 UnsmoothedPaths{i}=[UnsmoothedPaths{i};SmoothPaths{j+(i-1)*(Wells_in_loop+1)}];
       end
```

```
    end
end
for i=1:Num_loops
    SmoothPathsFinal{i}=pathsmoothing(UnsmoothedPaths{i},Rmin);
                        %Written MATLAB function for smoothing process
end
%Plotting Smoothed Lines
for i=1:Num_loops
figure(3)
plot3(SmoothPathsFinal{i}(:,1),SmoothPathsFinal{i}(:,2),SmoothPathsFinal{i}(:,3)
, 'color', 'b', 'linewidth', 2) ;
zlim([-180 180]);   %This figure settings have been made for the used data
xlim([-35000 35000]) %for other data these can be changed
ylim([-35000 35000]);
hold on
end
%Plotting Points
for i=1:Wn
  caption1 = sprintf('%d',n_W{i});
  ni_W = findnode(G,caption1);
  plot3(G.Nodes{ni_W,1},G.Nodes{ni_W,2},G.Nodes{ni_W,3},'b.', 'MarkerSize',  20)
  caption3 = sprintf('W%d', i);
  text(G.Nodes{ni_W,1},G.Nodes{ni_W,2},G.Nodes{ni_W,3},caption3, 'fontsize', 18)
end
plot3(G.Nodes{ni_2,1},G.Nodes{ni_2,2},G.Nodes{ni_2,3},'b.', 'MarkerSize',  20)
text(G.Nodes{ni_2,1},G.Nodes{ni_2,2},G.Nodes{ni_2,3},'RB', 'fontsize', 18)
xlabel('x');
ylabel('y');
zlabel('z');
pbaspect([1 1 1]);
grid off
hold off
%Calculating total length of smoothed lines
length_smoothed=0;
L_smoothed=cell(1,Num_loops);
for i=1:Num_loops
                %Written function for calculating length
L_smoothed{i}=smoothedpathlength(SmoothPathsFinal{i}(:,1),SmoothPathsFinal{i}(:,
2),SmoothPathsFinal{i}(:,3));
length_smoothed=length_smoothed+L_smoothed{i};
end
X = sprintf('Total length of the flowlines is equal to %d', length_smoothed);
disp(X);
```

Created functions:

The function, which is shown below, has been created for removing edges between nodes which are not neighbours:

```
function [new_matrix] = forcetoneighbour(pre_matrix,row_num,column_num)
%This function removes all edges between nodes which are not neighbours
%Adjacency matrix with edge values between nodes, the number of its rows
%and colums should be added as an input data
[x,y]=size(pre_matrix);
n_values=x;
J=cell(1,n_values);
j=zeros(1,n_values);
```

```
i=1;
j=[i+2:1+column_num-1,i+column_num+2:n_values];
J{i} = j; %First Point
for i=2:column_num-1
    j=[1:i-2,i+2:i+column_num-2,i+column_num+2:n_values];
    J{i} = j;
end %Mid of First Row
i=column_num;
j=[1:i-2,i+1:i+column_num-2,i+column_num+1:n_values];
J{i} = j; %Last point of first row
for mul=1:row_num-2
for k=1:column_num
    i=column_num*mul+k;
    if k==1
    j=[1:i-column_num-1,i-column_num+2:i-1,i+2:i+column_num-
1,i+column_num+2:n_values];
    J{i} = j;
    elseif k==column_num
    j=[1:i-column_num-2,i-column_num+1:i-2,i+1:i+column_num-
2,i+column_num+1:n_values];
    J{i} = j;
    else
    j=[1:i-column_num-2,i-column_num+2:i-2,i+2:i+column_num-
2,i+column_num+2:n_values];
    J{i} = j;
    end
end
end %Points of mid-rows
i=n_values-column_num+1;
j=[1:i-column_num-1,i-column_num+2:i-1,i+2:n_values];
J{i} = j; %First point of last row
for i=n_values-column_num+2:n_values-1
    j=[1:i-column_num-2,i-column_num+2:i-2,i+2:n_values];
    J{i} = j;
end %Mid points of last row
i=n_values;
j=[1:i-column_num-2,i-column_num+1:i-2];
J{i} = j; %Last point of last row
for i=1:n_values
   pre_matrix(i,J{i})=0;
end
[new_matrix]=pre_matrix;
end
```

The function, which is shown below, has been created for smoothing the determined route by taking into account minimum allowable radius of curvature requirement:

```
function [newpath] = pathsmoothing(input_path,Rmin)
%This function smoothes unsmoothed lines by taking into account required
%minimum radius of curvature to prevent slipping of flowlines during
%installation. Path data and Rmin should be added as an input
x=input_path(:,1);
y=input_path(:,2);
z=input_path(:,3);
% The first interpolation by using cubic spline method
num = numel(x) ;
L = zeros(num,1) ;
```

```
for i=2:num
    arc_length = sqrt((x(i)-x(i-1))^2+(y(i)-y(i-1))^2+(z(i)-z(i-1))^2);
    L(i) = L(i-1) + arc_length;
end
L=L./L(num);
x_t = spline(L,x) ;
y_t = spline(L,y) ;
z_t = spline(L,z) ;
tt = linspace(0,1,500) ;
xi = ppval(x_t,tt) ;
yi = ppval(y_t,tt) ;
zi = ppval(z_t,tt) ;
LinePath1=[transpose(xi), transpose(yi), transpose(zi)];
LinePathCal1=[transpose(xi), transpose(yi)];
LinePath2=LinePath1;
LinePathCal2=LinePathCal1;
num_removingpoints=1;
check=1;
%Iterative process for determining smoothed route with required minimum
%allowable Rmin
while check>0
    check=0;
while num_removingpoints>0
    [m,c]=size(LinePathCal1);
    num_pathpoints=m;
    removingpoints=[];
for i=1:m-2

R=radiusofcurvature(LinePathCal1(i,:),LinePathCal1(i+1,:),LinePathCal1(i+2,:));
        %Writtenn function for calculating radius of curvature for three
        %points
    if R<Rmin
        removingpoints=[removingpoints, i+1];
    end
end
LinePathCal1(removingpoints,:)=[];
LinePath1(removingpoints,:)=[];
num_removingpoints=numel(removingpoints);
end
[m1 n1]=size(LinePath1);
[m2 n2]=size(LinePath2);
if m1~=m2
    check=1;
    num = m1 ;
    L = zeros(num,1) ;
for i=2:num
    arc_length = sqrt((LinePath1(i,1)-LinePath1(i-1,1))^2+(LinePath1(i,2)-
LinePath1(i-1,2))^2+(LinePath1(i,3)-LinePath1(i-1,3))^2);
    L(i) = L(i-1) + arc_length;
end
L=L./L(num);
x_t = spline(L,LinePath1(:,1)) ;
y_t = spline(L,LinePath1(:,2)) ;
z_t = spline(L,LinePath1(:,3)) ;
tt = linspace(0,1,500) ;
xinew = ppval(x_t,tt) ;
yinew = ppval(y_t,tt) ;
zinew = ppval(z_t,tt) ;
LinePath1=[transpose(xinew), transpose(yinew), transpose(zinew)];
```

```
LinePathCal1=[transpose(xinew), transpose(yinew)];
LinePath2=LinePath1;
LinePathCal2=LinePathCal1;
end
end
[newpath]=LinePath1;
end
```

The function, which is shown below, has been created for calculating radius of curvature of three given points:

```
function [Radius] = radiusofcurvature(v1,v2,v3)
% Calculation of radius of curvature
%  v1,v2,v3 are 2D coordinates of the three neighbouring points in the
%  route
MPv1v2= (v2+v1)/2;
gradv1v2=(v1(2)-v2(2))/(v1(1)-v2(1));
perpgradv1v2=-1/gradv1v2;
MPv2v3= (v3+v2)/2;
gradv2v3=(v2(2)-v3(2))/(v2(1)-v3(1));
perpgradv2v3=-1/gradv2v3;
MPv1v3= (v3+v1)/2;
gradv1v3=(v1(2)-v3(2))/(v1(1)-v3(1));
perpgradv1v3=-1/gradv1v3;
CP=zeros(1,2);
if perpgradv1v2==perpgradv2v3
    R=1/0;
    elseif gradv1v2==0
        CP(1)=(perpgradv1v3*MPv1v3(1)-MPv1v3(2)-
perpgradv2v3*MPv2v3(1)+MPv2v3(2))/(perpgradv1v3-perpgradv2v3);
        CP(2)=perpgradv1v3*CP(1)-perpgradv1v3*MPv1v3(1)+MPv1v3(2);
        R=sqrt((v3(1)-CP(1))^2+(v3(2)-CP(2))^2);
    elseif gradv2v3==0
        CP(1)=(perpgradv1v2*MPv1v2(1)-MPv1v2(2)-
perpgradv1v3*MPv1v3(1)+MPv1v3(2))/(perpgradv1v2-perpgradv1v3);
        CP(2)=perpgradv1v2*CP(1)-perpgradv1v2*MPv1v2(1)+MPv1v2(2);
        R=sqrt((v3(1)-CP(1))^2+(v3(2)-CP(2))^2);
    else
        CP(1)=(perpgradv1v2*MPv1v2(1)-MPv1v2(2)-
perpgradv2v3*MPv2v3(1)+MPv2v3(2))/(perpgradv1v2-perpgradv2v3);
        CP(2)=perpgradv1v2*CP(1)-perpgradv1v2*MPv1v2(1)+MPv1v2(2);
        R=sqrt((v3(1)-CP(1))^2+(v3(2)-CP(2))^2);
end
[Radius]=R;
end
```

The function, which is shown below, has been created for calculating length of the determined routes:

```
function [total_length] = smoothedpathlength(x_points,y_points,z_points)
%This function calculates length of smoothed lines
Points = [x_points(:),y_points(:),z_points(:)];
num_dim = size(Points,2);
% calculating total length of euclidean distances
each_eudist = sqrt(sum(diff(Points,[],1).^2,2));
length_eudist = each_eudist;
```

```
% determining 3rd order polynomials for the sections
splines = cell(1,num_dim);
spline_d = splines;
for i = 1:num_dim
      splines{i} = spline([0;cumsum(length_eudist)],Points(:,i));
      num_coef = numel(splines{i}.coefs);
      if num_coef < 4
        splines{i}.coefs = [zeros(1,4-num_coef),splines{i}.coefs];
        splines{i}.order = 4;
      end
  % Differentiation
  differentiation_array = [3 0 0;0 2 0;0 0 1;0 0 0];
  x_p = splines{i};
  x_p.coefs = x_p.coefs*differentiation_array;
  x_p.order = 3;
  spline_d{i} = x_p;
end
% Numerical integration
polynomial_array = zeros(num_dim,3);
for i = 1:splines{1}.pieces
  % extract polynomials for the derivatives
  for j = 1:num_dim
    polynomial_array(j,:) = spline_d{j}.coefs(i,:);
  end
  each_eudist(i) = quadgk(@(t) integ_equation(t),0,length_eudist(i));
                %Available MATLAB function for Gaussian-Kronrod quadrature
                %method of numerical integration
end
total_length = sum(each_eudist);

%Equation for integration
  function [equation] = integ_equation(t)
    % sqrt((dx/dl)^2 + (dy/dl)^2 + (dz/dl)^2)
    equation = zeros(size(t));
    for k = 1:num_dim
      equation = equation + polyval(polynomial_array(k,:),t).^2;
    end
    [equation] = sqrt(equation);
  end
end
```

8.2. Some screenshots from the created spreadsheet

Several screenshots from the created spreadsheet are shown below. Screenshots illustrate CAPEX calculations, which were used in the case study, with average cost values.

TEMPLATE LAYOUT:

	Cells with this colour are for input data and can be changed.

	Cells with this colour are dependent cells. Therefore, these values should not be changed.

Select Subsea Production System (SPS) Configuration	Template

Assumptions:	Foundations are only used for templates. Number of Subsea Control Modules (SCM) is equal to number of manifolds plus number of X-mas trees and SCMs are installed while installing manifolds and X-mas trees. Number of Subsea Distribution Units (SDU) is equal to number of templates. Same umbilical is used from riser to SDU and from SDU to template. Same type of equipment is used in the whole field.

General Data				
Number of templates	6	Total length of flowlines	355261	m
Number of protection structures	6	Total length of umbilicals	177631	m
Number of wells	18			

Hardware Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Price of each template	30,000,000	NOK	Total price of SPS (excluding umbilicals and flowlines)	1,620,000,000	NOK
Price of each foundation system	-	NOK			
Price of each manifold	70,000,000	NOK			
Price of each protection structure	-	NOK	Total price of flowlines	1,017,468,143	NOK
Price of each X-mas tree with choke and MPM	45,000,000	NOK			
Price of each wellhead system	3,000,000	NOK			
Price of each tubing hanger system	3,000,000	NOK	Total price of umbilicals	2,486,828,562	NOK
Price of each subsea control module	3,000,000	NOK			
Price of each subsea distribution unit	5,000,000	NOK			
Price of flowline per meter	2,864	NOK/m	**Total hardware cost**	**5,124,296,705**	**NOK**
Price of umbilical per meter	14,000	NOK/m			

Installation Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Installation duration of each template	3	day	Total installation cost of SPS (excluding umbilicals and flowlines)	371,250,000	NOK
Installation duration of each manifold	3	day			
Installation duration of each X-mas tree	2	day	Total installation cost of flowlines	1,563,149,382	NOK
Installation duration of each SDU	1.5	day			

Installation duration of each WH system	1.5	day	Total installation cost of umbilicals	781,574,691	NOK
Installation duration of each TH system	1.5	day			
Installation duration of flowlines	1.1	day/km	**Total Installation Cost**	**2,715,974,072**	**NOK**
Installation duration of umbilicals	1.1	day/km			
Daily rate of equipment installation vessel	2,750,000	NOK/day			
Daily rate of umbilical and flowline installation vessel	4,000,000	NOK/day			

Total Cost					
Total Hardware Cost		Total Installation Cost		**TOTAL COST**	
5,124,296,705	NOK	2,715,974,072	NOK	**7,840,270,777**	**NOK**

CLUSTERED SATELLITE WELLS LAYOUT:

Select Subsea Production System (SPS) Configuration	Cluster

Assumptions:	Foundations are only used for manifolds. Number of Subsea Control Modules (SCM) is equal to number of manifolds plus number of X-mas trees and SCMs are installed while installing manifolds and X-mas trees. Same protection structures are used for X-mas trees and manifolds. Number of Subsea Distribution Units (SDU) is equal to number of manifolds. Same umbilical is used from riser to SDU and from SDU to manifold. Same Installation vessel is used to install smaller ID and bigger ID flowlines. Same type of equipment is used in the whole field.

General Data				
Number of manifolds	6	Total length of flowlines from X-mas trees to manifolds (smaller ID)	61861.6	m
Number of protection structures	24	Total length of flowlines from manifolds to riser (bigger ID)	355261	m
Number of wells	18	Total length of umbilicals	239492	m

Hardware Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Price of each foundation	3,000,000	NOK	Total price of SPS (excluding umbilicals and flowlines)	1,530,000,000	NOK
Price of each manifold	70,000,000	NOK			
Price of each protection structure	3,000,000	NOK			
Price of each Christmas tree with choke and MPM	45,000,000	NOK	Total price of flowlines	1,085,639,598	NOK
Price of each wellhead system	3,000,000	NOK			
Price of each tubing hanger system	3,000,000	NOK			
Price of each subsea control module	3,000,000	NOK	Total price of umbilicals	3,352,890,602	NOK
Price of each subsea distribution unit	5,000,000	NOK			
Price of smaller ID flowline per meter	1,102	NOK/m			
Price of bigger ID flowline per meter	2,864	NOK/m	**Total hardware cost**	**5,968,530,200**	**NOK**
Price of umbilical per meter	14,000	NOK/m			

Installation Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Installation duration of each manifold	3	day	Total installation cost of SPS (excluding umbilicals and flowlines)	445,500,000	NOK
Installation duration of each protection structure	1.5	day			
Installation duration of foundation system	1.5	day	Total installation cost of flowlines	1,835,340,309	NOK
Installation duration of each X-mas tree	2	day			
Installation duration of each SDU	1.5	day	Total installation cost of umbilicals	1,053,765,618	NOK
Installation duration of each WH system	1.5	day			
Installation duration of each TH system	1.5	day			
Installation duration of flowlines	1.1	day/km			
Installation duration of umbilicals	1.1	day/km	**Total Installation Cost**	**3,334,605,927**	**NOK**
Cost of equipment installation vessel	2,750,000	NOK/day			
Cost of umbilical and flowline installation vessel	4,000,000	NOK/day			

Total Cost					
Total Hardware Cost		Total Installation Cost		**TOTAL COST**	
5,968,530,200	NOK	3,334,605,927	NOK	**9,303,136,127**	**NOK**

SATELLITE WELLS LAYOUT:

Select Subsea Production System (SPS) Configuration	Satellite

Assumptions:	There is not any foundation. Number of Subsea Control Modules (SCM) is equal to number of X-mas trees and SCMs are installed while installing X-mas trees. Same umbilical is used between Subsea Distribution Units (SDU) and riser base and between SDUs and wells. Same type of equipment is used in the whole field.

General Data				
Number of subsea distribution unit (SDU)	6	Total length of flowlines	540492	m
Number of protection structures	18			
Number of wells	18	Total length of umbilicals	239492	m

Hardware Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Price of each protection structure	3,000,000	NOK	Total price of SPS (excluding umbilicals and flowlines)	1,056,000,000	NOK
Price of each Christmas tree inc. MPM and choke	45,000,000	NOK			
Price of each wellhead system	3,000,000	NOK	Total price of flowlines	595,622,093	NOK
Price of each tubing hanger system	3,000,000	NOK			
Price of each subsea control module	3,000,000	NOK	Total price of umbilicals	3,352,890,602	NOK
Price of each subsea distribution unit	5,000,000	NOK			
Price of flowline per meter	1,102	NOK/m	**Total hardware cost**	**5,004,512,695**	**NOK**
Price of umbilical per meter	14,000	NOK/m			

Installation Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Installation duration of each protection structure	1.5	day	Total installation cost of SPS (excluding umbilicals and flowlines)	346,500,000	NOK
Installation duration of each X-mas tree	2	day			
Installation duration of each SDU	1.5	day	Total installation cost of flowlines	2,378,164,436	NOK
Installation duration of each WH system	1.5	day			
Installation duration of each TH system	1.5	day	Total installation cost of umbilicals	1,053,765,618	NOK
Installation duration of flowline	1.1	day/km			
Installation duration of umbilical	1.1	day/km	**Total Installation Cost**	**3,778,430,054**	**NOK**
Cost of equipment installation vessel	2,750,000	NOK/day			
Cost of umbilical and flowline installation vessel	4,000,000	NOK/day			

Total Cost					
Total Hardware Cost		Total Installation Cost		**TOTAL COST**	
5,004,512,695	NOK	3,778,430,054	NOK	**8,782,942,749**	**NOK**

DAISY-CHAIN LAYOUT:

Select Subsea Production System (SPS) Configuration	Daisy-Chain

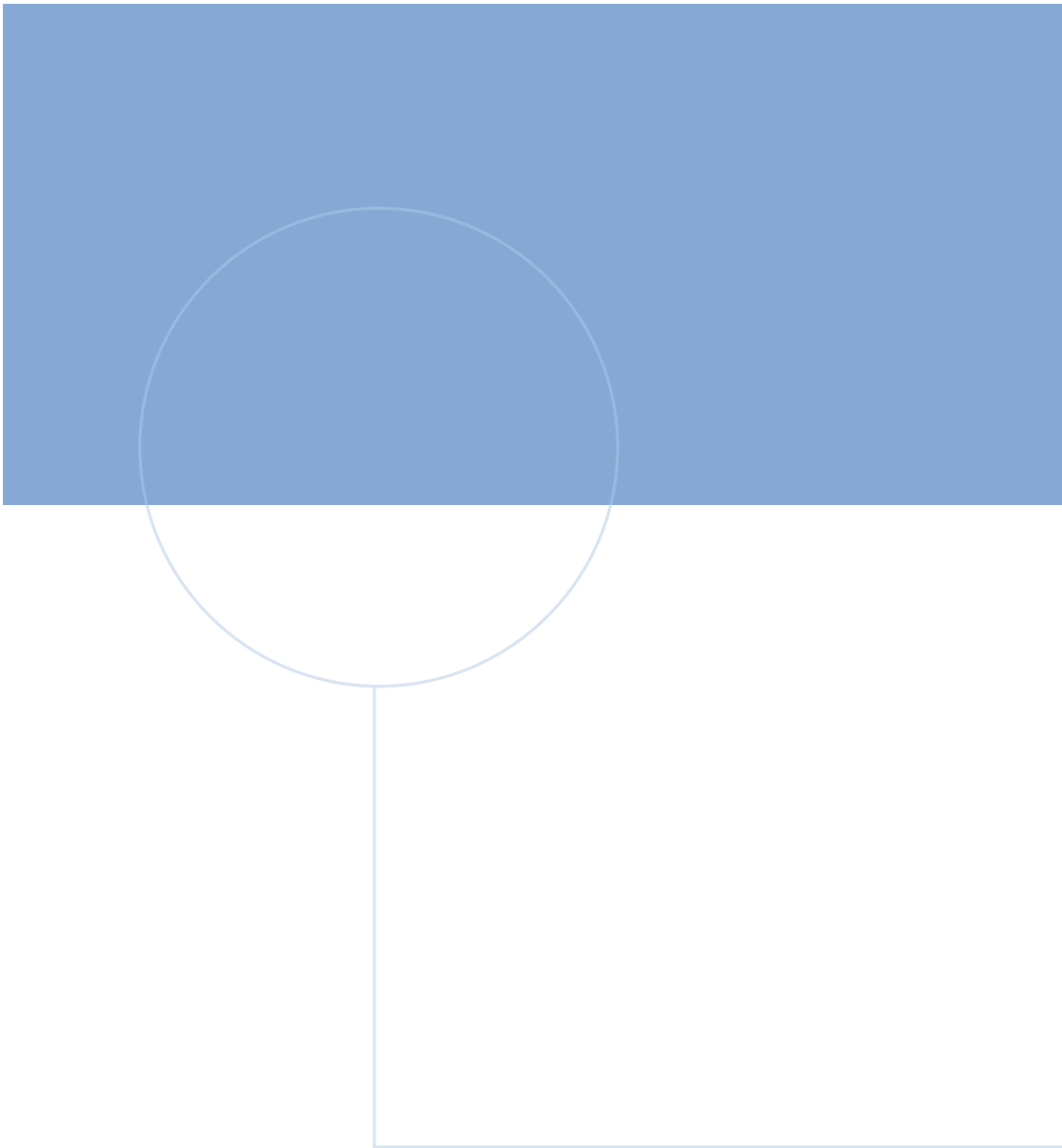
Assumptions:	There is not any foundation. Number of Subsea Control Modules (SCM) is equal to number of X-mas trees and SCMs are installed while installing X-mas trees. Same umbilical is used between Subsea Distribution Units (SDU) and riser base and between SDUs and wells. Same type of equipment is used in the whole field.

General Data				
Number of subsea distribution unit (SDU)	6	Total length of flowlines	289789	m
Number of protection structures	18			
Number of wells	18	Total length of umbilicals	239492	m

Hardware Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Price of each protection structure	3,000,000	NOK	Total price of SPS (excluding umbilicals and flowlines)	1,056,000,000	NOK
Price of each Christmas tree inc. MPM and choke	45,000,000	NOK			
Price of each wellhead system	3,000,000	NOK	Total price of flowlines	1,149,304,071	NOK
Price of each tubing hanger system	3,000,000	NOK			
Price of each subsea control module	3,000,000	NOK	Total price of umbilicals	3,352,890,602	NOK
Price of each subsea distribution unit	5,000,000	NOK			
Price of flowline per meter	3,966	NOK/m	**Total hardware cost**	**5,558,194,673**	**NOK**
Price of umbilical per meter	14,000	NOK/m			

Installation Cost					
Input Data			Results		
Properties	Value	Unit	Properties	Value	Unit
Installation duration of each X-mas tree	2	day	Total installation cost of SPS (excluding umbilicals and flowlines)	346,500,000	NOK
Installation duration of each protection structure	1.5	day			
Installation duration of each SDU	1.5	day	Total installation cost of flowlines	1,275,072,595	NOK
Installation duration of each WH system	1.5	day			
Installation duration of each TH system	1.5	day	Total installation cost of umbilicals	1,053,765,618	NOK
Installation duration of flowline	1.1	day/km			
Installation duration of umbilical	1.1	day/km	**Total Installation Cost**	**2,675,338,213**	**NOK**
Cost of equipment installation vessel	2,750,000	NOK/day			
Cost of umbilical and flowline installation vessel	4,000,000	NOK/day			

Total Cost					
Total Hardware Cost		Total Installation Cost		**TOTAL COST**	
5,558,194,673	NOK	2,675,338,213	NOK	**8,233,532,886**	**NOK**



NTNU

Norwegian University of
Science and Technology