

Bjørn Børresen

Multi-objective Optimisation of FPSO Hull Geometry Using Genetic Algorithm Variations

Master's thesis in Marine Technology

Supervisor: Svein Aanond Aanondsen

June 2020



Norwegian University of
Science and Technology

Multi-objective Optimisation of FPSO Hull Geometry Using Genetic Algorithm Variations

Bjørn Børresen

Marine Technology

Submission date: June 2020

Supervisor: Asst. Prof. Svein Aanond Aanondsen, IMT

Co-supervisor: Øyvind Røyseth, Altera Infrastructure

Norwegian University of Science and Technology
Department of Marine Technology

Master Thesis in Marine Systems Design
Stud.Tech. Bjørn Børresen

**Multi-objective optimisation of FPSO Hull Geometry
Using Genetic Algorithm Variations**
Spring 2020

Background

When ordering an FPSO, the customer requirements are often limited to a given value of bbl/day and days of operation. The motions of the vessel will determine whether activities such as helicopter landing, offloading and production will be possible. The vessel motions, based on its seakeeping ability, will depend on turret position, incoming wave amplitude and height and the hull form. It is therefore of interest to evaluate which hull form and what main dimensions are optimal to meet the customer requirements. As part of a multi-objective optimisation problem, it is also of interest to minimise the lightship weight and maximise the deck- area and loading capacity.

Overall aim and focus

The thesis will aim to provide a model that can optimise the main dimensions of an FPSO vessel at an early design phase. The model should consider the deck area and build cost while the vessel's motions comply with offshore regulations. The vessel's motions are to be generated by a simplified model. If necessary, validation will be performed using conventional 2D strip-theory.

1. Describe the general characteristics and the relevant operating context of an FPSO unit
2. Study relevant topics that apply to the optimisation on objective functions
3. Develop geometric model feasible for optimisation and objective functions while giving realistic results
4. Create loading conditions and weight distributions based on optimisation output data
5. Evaluate the most influential input parameters affecting the optimisation algorithm
6. Explore and identify methods that guarantee convergence in the optimisation at the global optimum
7. Develop a framework to fairly evaluate the best individual solution points and complete sets of Pareto optimal solutions
8. Analyse the hulls' motions in regular waves
9. Discuss and conclude the reliability of the optimisation model and its results

Scope and main activities

The candidate should presumably cover the following main points:

1. Provide an overview of any evolutionary algorithms that will be used in the optimisation and how it will be set up.
2. Develop a model that is based on a parametric hull geometry that is possible to optimise and accordingly calculate various loading conditions for the vessel.
3. Analyse the key performance indexes (KPI's) of the vessels to determine which vessels are of interest for further evaluation
4. Discuss and conclude the validity of the optimisation model and the resulting performance indices of the vessel(s)

Collaboration with Altera Infrastructure

Altera infrastructure should provide information of generation III Altera Infrastructure's FPSO's to compare and validate the results from the model. The necessary disclosed information can include, but is not limited to:

- Weigh lists and stability reports from current vessels
- Market and manufacturing data on for current FPSO yards
- Operational criteria to be used for validation

Modus operandi

At NTNU, Assistant Professor Svein Aanond Aanonsen will be the responsible advisor.

At Altera Infrastructure, Senior Naval Architect Øyvind Røyseth will be the responsible advisor.

The work shall follow the guidelines given by NTNU for the MSc Master Thesis work.



Svein Aanond Aanonsen
Department of Marine Engineering, NTNU



Øyvind Røyseth
Altera Infrastructure

Preface

This report is a master thesis within the field of naval architecture and marine engineering. The work has been carried out throughout the spring term of 2020 at the Norwegian University of Science and Technology. The master thesis is equivalent to 30 credits. The topics in the thesis are based on a proposal from the company Altera Infrastructures - previously known as Teekay Offshore - with inputs from myself and my supervisor. Altera Infrastructure contributed with technical drawings, weight list, market insights of FPSO's and assistance whenever needed.

I want to thank my supervisors, Assistant Professor Svein Aanond Aanondsen at NTNU and naval architect Øyvind Røyseth at Altera Infrastructure. I am very grateful for their continuous follow-up throughout a period of a global pandemic and the weekly meetings despite being confined to our living rooms. I also wish to thank Professor Marilena Greco for her comments and inputs on the seakeeping models, Professor Dag Myrhaug for his help in accurately modelling weather conditions, shipbroker Petter Sissener at Lorentzen & Stemoco for his unique insight of the maritime financial market, and Aleksander Vold Kristiansen for making the partnership with Altera Infrastructure possible in the first place.

Lastly, I wish to thank my peers at my office *Blokka Invest* for their input and academic discussion regarding the master thesis subject, and for their support and patience throughout our time in lockdown together.



Bjørn Børresen

Trondheim, 10th of June 2020

Lockdown-day + 91...

Summary

A floating production, storage and offloading unit (FPSO) is often the only viable option for oil and gas production in deep waters. More reservoirs are discovered in deep waters, and the popularity of FPSO's is rising. Key market players seek cost-efficient deliveries with short lead times. Consequently, several entities, including Altera Infrastructure, are invested in standardising solutions for their FPSO's. While different standardised designs are necessary for different location environment, the type of FPSO design evaluated in this thesis must be able to withstand all possible environments and its hull optimised to generate the smallest motions possible. The processing plant on the ship's tank top deck is of great importance, and hence particular focus is given to maximising the space for processing units on deck at a low cost.

The optimisation process was completed by using a genetic algorithm. The thesis objectives of maximising the deck area and minimising the hull build cost represent the objective function of the optimisation process. A parametric hull model with 12 decision variables is generated to shape the vessel's hull accurately. The genetic algorithm's output resulted in a Pareto-front with only non-dominating solutions. Consequently, one could choose any of the optimal points on the front, each with an equal degree of optimality. To further enhance the optimisation process, a local search algorithm was employed. While the genetic algorithm uses an evolutionary method that performs well with many variables and objective functions, the local search is employed once the solution has converged sufficiently, and the search space is considerably limited. Unbiased measures have been used to determine both the performance of individual solutions as well as complete sets of Pareto fronts. Effectively, the genetic algorithm performed best when evaluating a set of optimal solutions, while the local hybrid search algorithm generated solutions closest to the point of maximum deck area and minimum cost, called the Utopian point. Multiple optimisation runs were completed with various input parameters altered for each run. Input parameters such as the Pareto fraction, the maximum number of stall generations and initial population matrix were changed in order to determine which setup rendered the best results. Consequently, 6,255 vessels have been generated from multiple optimisation runs. The deck areas span from 13,525 to 16,824 m² with a hull build cost range of US \$59.78-\$72.2 million. Two vessels were chosen for further evaluation. One vessel has the lowest utility cost, i.e. the lowest price for the largest deck area. The other is closest to the Utopian point in the optimisation. The resulting minimum utility cost of all vessels is US\$4,302/m² and US\$413/m³ of cargo.

Due to the harsh weather conditions on the intended location, a vessel motions evaluation has been completed for both vessels. Particular focus has been given to fulfilling the operational criteria which are set by Altera Infrastructure and other classification societies. Based on a wave frequency spectrum of the given location, the vessels are compliant with most of the criteria during normal conditions. Another vital area of focus is the crew comfort and the measure of the mean sickness dose value measuring the crew's seasickness. Conclusively, in normal conditions, both vessels meet all of the operational requirements. Consequently, one can confirm that the genetic algorithm and the local hybrid search algorithms have efficiently and accurately generated viable hull geometries that should be evaluated in further work.

Sammen drag

En flytende produksjons-, lagrings- og avlastningsenhet (FPSO) er ofte det eneste levedyktige alternativet for olje- og gassproduksjon på dypt vann. Flere reservoarer blir oppdaget på dypt vann, og populariteten til FPSO-er øker. Sentrale aktører i markedet søker kostnadseffektive leveranser med korte ledetider. Følgelig er flere selskaper, inkludert Altera Infrastructure, investert i standardisering av løsninger for FPSO-ene. Mens forskjellige standardiserte konstruksjoner er nødvendige for forskjellige lokasjoner, må typen FPSO-design evaluert i denne oppgaven kunne tåle alle mulige miljøer og skroget er optimalisert for å generere minst bevegelser mulig. Foredlingsanlegget på skipets tankdekk er av stor betydning, og det gis særlig fokus på å maksimere plassen for prosesseringsenheter på dekket til en lav pris.

Optimaliseringsprosessen ble fullført ved bruk av en genetisk algoritme. Oppgavens mål om å maksimere dekkområdet og minimere kostnadene for skrogbygging representerer objektivfunksjonen til optimaliseringsprosessen. En parametrisk skrogmodell med 12 beslutningsvariabler genereres for å forme fartøyets skrog nøyaktig. Den genetiske algoritmen genererte en Pareto-front med bare ikke-dominerende løsninger. Følgelig kunne man velge hvilket som helst av de optimale punktene på fronten, hver med en like grad av optimalitet. For å forbedre optimaliseringsprosessen ble det benyttet en lokal søkealgoritme. Mens den genetiske algoritmen bruker en evolusjonsmetode som fungerer godt med mange variabler og objektive funksjoner, brukes det lokale søket når løsningen har konvertert tilstrekkelig, og søkeområdet er betydelig begrenset. Objektive målinger har blitt brukt for å bestemme både ytelsen til individuelle løsninger så vel som komplette sett med Pareto-fronter. Den genetisk algoritmen presterte best da en evaluerte et sett med optimale løsninger, mens den lokale hybrid-søkealgoritmen genererte løsninger nærmest punktet for maksimalt dekkareal og minimumskostnad, heretter kalt det Utopiske punktet. Flere optimaliseringsløp ble fullført med forskjellige parametere endret for hver kjøring. Parametere som Pareto-fraksjonen, det maksimale antall stallgenerasjoner og den innledende populasjonsmatrisen ble endret for å bestemme hvilket oppsett som ga best resultat. Følgelig er 6,255 individuelle skip blitt generert fra flere optimeringer. Dekkarealene spenner fra 13.525 til 16.824 m² med en skrogkostnad på US\$59,78-72,2 millioner. To fartøy ble valgt for videre evaluering. Ett fartøy har den laveste nyttekostnaden, dvs. den laveste prisen for det største dekkområdet. Den andre er nærmest det utopiske punktet i optimaliseringen. Den resulterende minste nyttekostnad for alle fartøyer er US\$4,302/m² og US\$413/m³ for lasten.

På grunn av de tøffe værforholdene på det tiltenkte stedet, ble begge fartøy sine bevegelser evaluert. Spesielt fokus er gitt på å oppfylle de operasjonelle kriteriene som er satt av Altera Infrastructure og andre klasse-selskaper. Basert på et bølgefrequensspektrum for det gitte stedet, er fartøyene i samsvar med de fleste av kriteriene under normale forhold. Et annet viktig fokusområde er besetningskomforten og målet for den gjennomsnittlige sykedoseverdien som måler besetningens sjøsyke. Dermed, under normale forhold, oppfyller begge fartøyene alle driftskrav. Følgelig kan man bekrefte at den genetiske algoritmen og de lokale hybrid-søkealgorithmsene effektivt og nøyaktig har generert levedyktige skroggeometrier som bør evalueres i videre arbeid.

Acronyms

bbbl	Standard petroleum barrel
BDTI	Baltic Dirty Tanker Index
CAPEX	Capital Expenditure
CCS	Carbon Capturing and storage
COG	Centre of gravity
COT	Crude Oil Tanks
DoF	Degrees of freedom
DWT	Deadweight
FPSO	Floating Production, Storage and Offloading
FSM	Free surface momentum
FSO	Floating Storage and Offloading
GA	Genetic algorithm
HGA	Hybrid genetic algorithm
ISO	Organisation for Standardisation
ITTC	International Towing Tank Conference
KPI	Key performance index
LCB	Longitudinal centre of buoyancy
LCG	Longitudinal centre of gravity
LQ	Living quarters
LR	Long-range (tanker vessel size)
LWT	Lightship weight
MOGA	Multi-objective genetic algorithm
MOLP	Linear multi-objective optimisation problem
MOOP	Multi-objective non-linear optimisation problem
MOP	Multi-objective optimisation
MSDV	Mean sickness dose value
MSG	Maximum stall generations
MT	Metric tonnes
NORSOK	NORsk SOKkels Konkurransesposisjon (Norwegian)
NSGA	Non-dominated sorting genetic algorithm
RAO	Response amplitude operators
TCE	Time-charter Equivalent
TCG	Transverse centre of gravity
VCG	Vertical centre of gravity
VLCC	Very large crude carrier
VOC	Volatile Organic Compounds
WBT	Water Ballast Tanks
WTI	West Texas Intermediate
YTD	Year-to-date

Nomenclature

$\bar{A}(x_i)$	Normalised value of objective function for deck area
$A(x_i)$	Objective function for deck area
B	Breadth
β	Incident wave angle
B_{WL}	Breadth waterline
$\bar{C}(x_i)$	Normalised value of objective function for build cost
$C(x_i)$	Objective function for build cost
C_B	Block coefficient
C_M	Midship coefficient
C_P	Prismatic coefficient
C_W	Waterplane coefficient
$D(X_j)$	Distance to Utopian point for solution X for vessel j
D	Moulded depth
d_j, \bar{d}_j	Spread of Pareto solutions, average spread
E_x	Excursion envelope
η_j	Amplitude of motion in direction j
H_S	Significant wave height
k	Wave number
λ	Wave length
L_{OA}	Length overall
L_{PP}	Length between perpendiculars
L_{WL}	Waterline length
P_F	Pareto fraction
P_c	Painting cost
S	Wet surface area
Sc	Steel cost
σ_C	Constraint tolerance
σ_F	Function tolerance
Sp	Surface preparation cost
T, \bar{T}	Draught, mean draught
ΔT	Longitudinal trim
T_A	Draught at aft perpendicular
T_F	Draught at front perpendicular
T_N	Natural period
Tp	Wave period
U_P	Utopian point
V_P	Percentage of vomiting crew
X_0	Initial population matrix
x_i	i th variable of chromosome X_j

Table of contents

1	Introduction	1
1.1	Background	2
1.2	FPSO and offshore market outlook	3
1.3	Key performance indicators	5
1.4	Objectives	5
1.5	Scope and limitations	6
2	Literature review	7
2.1	FPSO Characteristics	7
2.2	Vessel response in regular waves	9
2.2.1	Sea states at location	9
2.2.2	Response functions	10
2.3	Vessel response criteria	12
2.3.1	Operational criteria	14
2.3.2	Helicopter criteria	14
2.3.3	Excursion envelope	14
2.3.4	Motion sickness criteria	15
2.4	Multi-objective optimisation problem	16
2.4.1	Formulation	16
2.4.2	Pareto front	18
2.4.3	Initial population	18
2.4.4	Events in the genetic algorithm	20
2.4.5	Hybrid approach	23
2.4.6	Hypervolume indicator	25
3	Problem description	26
3.1	Geometry-related optimisation setup	26
3.1.1	Simplified hull model	26
3.1.2	Upper and lower bounds	28
3.1.3	Constraints	29
3.2	Function and algorithm-related optimisation setup	30
3.2.1	Objective functions	30

3.2.2	Algorithm input parameters	32
3.2.3	Hybrid approach setup	34
3.2.4	Stopping criteria and restrictions	35
4	Results	37
4.1	Optimisation output	37
4.1.1	Choice of vessels for further study	38
4.1.2	Variation of algorithm preferences	40
4.2	Optimal hull geometry	44
4.2.1	Numerically optimal vessel	44
4.2.2	Market optimal vessel	49
4.3	Cost breakdown	53
5	Analysis	55
5.1	Algorithm evaluation	55
5.1.1	Goodness of solutions	56
5.1.2	Optimal algorithm parameters	58
5.2	Optimal hull geometries	59
5.2.1	Vessels comparison	59
5.2.2	Trim and heel	60
5.2.3	Turret positioning and cargo tanks	61
5.2.4	Fleet comparison	61
5.3	Vessel response	64
5.3.1	Crew comfort	66
5.3.2	Helicopter operation	68
5.3.3	Heavy weather lockdown	69
5.3.4	Loading and offloading	71
5.3.5	Weather and heading dependent operability	72
5.4	Verification and validation	73
5.4.1	Realistic vessel motions	73
5.4.2	Price estimation	73
5.4.3	Three-dimensional objective space	74
5.5	Sensitivity of results	75
5.5.1	Upper and lower bounds	75
5.5.2	Motion sensitivity	76
6	Conclusion	77
6.1	Further work	79
	Bibliography	81
	Appendices	
	Appendix A Wave frequency table	I

Appendix B Vessel no. 1864	II
B.1 Tank arrangement for fully laden condition	II
B.2 Lines plan	II
Appendix C Vessel no. 4843	IV
C.1 Tank arrangement for fully laden condition	IV
C.2 Lines plan	IV
Appendix D Response amplitude operators	VI
D.1 Vessel no. 1864	VI
D.1.1 Heave motion	VI
D.1.2 Roll motion	VI
D.1.3 Pitch motion	VII
D.2 Vessel no. 4843	VII
D.2.1 Heave motion	VII
D.2.2 Roll motion	VIII
D.2.3 Pitch motion	VIII
Appendix E Simplified vessel motion calculation	IX
Appendix F MATLAB scripts	XI
F.1 longOptimisation.m	XI
F.2 gaSolver.m	XIII
F.3 hybridSolver.m	XV
F.4 getCost.m	XVI
F.5 deckArea.m	XXI
F.6 conditions.m	XXII
F.7 objectiveFunction.m	XXII
F.8 verticalMotions.m	XXIII
F.9 optimisationAnalysis.m	XXVI
F.10 motionResults.m	XXXII

List of Figures

1.0.1 Example of turret-moored FPSO	1
1.2.1 Commodity prices and dirty tanker index	3
2.1.1 Various FPSO mooring systems	8
2.4.1 Example of max-min Pareto front	18
2.4.2 Effect of population size on objective function value	19
2.4.3 The efficiency of hybrid algorithm near global minimum	23
2.4.4 Flow chart for hybrid genetic algorithm	24
2.4.5 Example of hypervolume corresponding to the objective vectors	25
3.1.1 Simplified FPSO hull geometry for optimisation	27
3.1.2 Simplification of aft bilge radius	27
3.2.1 Population size effect on solution quality and computation time	33
3.2.2 Process illustration for changes in optimisation algorithms	34
3.2.3 Flowchart and stopping criteria for vessel optimisation	36
4.1.1 Results of GA and local search algorithm	37
4.1.2 Histogram and distribution of vessel's distance to Utopian point	39
4.1.3 Chosen vessels from resulting data	40
4.1.4 Optimisation results with varying Pareto fraction values	41
4.1.5 Optimisation results with varying limit of maximum stall generations	42
4.1.6 Optimisation result with variations of choosing X_0	43
4.2.1 Visual presentation of vessel no. 4843	45
4.2.2 Body plan of vessel sections	45
4.2.3 GZ curves for various loading conditions for vessel no. 4843	48
4.2.4 Visual presentation of vessel no. 1864	50
4.2.5 Body plan of vessel no. 1864	50
4.2.6 GZ curves for various loading conditions for vessel no. 1864	52
5.1.1 Box-plot results of normalised distance to the Utopian point	55
5.1.2 Hypervolume of the genetic algorithm optimisation	56
5.2.1 Profile outline of chosen vessels	60
5.2.2 Example of turret positioning between cargo tanks	61

5.2.3 DWT and displacement of currently operating vessels	62
5.2.4 Length between perpendiculars and displacement of currently operating vessels	62
5.2.5 Waterline breadth and length between perpendiculars of currently operating vessels	63
5.3.1 RAO for heave displacement and acceleration for vessel no. 4843	65
5.3.2 RAO for pitch displacement and acceleration for vessel no. 4843	65
5.3.3 MSDV at multiple locations	66
5.3.4 Change of MSDV of vessel no. 1864 and variation of H_S	67
5.3.5 Change of MSDV of vessel no. 4843 and variation of H_S	67
5.3.6 MSDV criteria boundary for both vessels	68
5.3.7 Maximum operating limits for helicopter operation	69
5.3.8 Maximum operating limits for partial lock-down on vessel 1864	69
5.3.9 Maximum operating limits for partial lock-down on vessel no. 4843	70
5.3.10 Cumulative probability distribution of H_S and T_p	71
5.3.11 Percentage operability for both vessels	72
5.5.1 Sensitivity of upper and lower bounds	75
5.5.2 Criteria boundary for multiple MSDV for for both vessels	76

List of Tables

1.3.1 Summary of key performance indexes	5
2.2.1 Extreme Metocean Criteria Summary - Omnidirectional, All Year	9
2.3.1 Summary of helideck operations limits	14
2.4.1 Example of scattered crossover with random binary vector.	21
3.1.1 Upper and lower bounds of main dimensions	28
4.1.1 Minimum and maximum of objective function values	38
4.2.1 Main particulars for vessel no. 4843	44
4.2.2 Lightship weight and centres of gravity for vessel no. 4843	46
4.2.3 DWT and COG for vessel no. 4843	47
4.2.4 Stability data for vessel no. 4843	47
4.2.5 Evaluation of intact stability criteria for vessel no. 4843	48
4.2.6 Main particulars for vessel no. 1864	49
4.2.7 Lightship weights and centres of gravity for vessel no. 1864	51
4.2.8 DWT and COG for vessel no. 1864	51
4.2.9 Stability data for vessel no. 1864	52
4.3.1 Costs and surcharges for paint and steel work	53
4.3.2 Main cost breakdown	54
5.1.1 Algorithm-dependent hypervolume results	57
5.1.2 Input parameters of optimisation algorithms	58
5.2.1 Final dimensions of optimal chromosomes	59

Introduction

Floating Production, Storage and Offloading (FPSO) units are vessels employed to process and temporarily store crude oil from wellhead platforms, or directly from a subsea installation. As the cargo space fills up with the extracted petroleum product, shuttle tankers periodically connect to offload the cargo and transport it to refineries. Direct offloading offshore is a clear economic advantage as it does not require high investment costs of fitting a pipeline to shore. FPSO's have the capacity of working in both shallow and deep water but is often employed in the latter due to high investment costs of fixed structures in deep seas. The vessels are highly flexible as they can be shifted from one field to another conveniently and economically [31].



Figure 1.0.1: Example of turret-moored FPSO [36].

In most cases, crude oil is transferred to the vessel's cargo tanks while treated gas is used as fuel for onboard power generation. Excess gas can either be re-injected into the reservoirs, exported to shore via a pipeline or burned off. Given adequate filtration and cleaning, water from the wells may be discharged overboard, or re-injected with any gas [33]. A rendering of an FPSO with an internal turret is illustrated in figure 1.0.1.

1.1 Background

When the oil fields are located in very deep waters and subject to harsh weather conditions, one of a few viable options is the use of an FPSO. The vessels may be purpose-built at a yard as an FPSO or may stem from a converted oil tanker. In calm waters, a lower specification is often sufficient with simpler mooring systems and hull geometry. In the North Sea, other vessel properties are required, and a hull geometry that can withstand the sea states is of much more importance [8, 30]. Consequently, Altera seeks at standardising their FPSO's design so that the vessels can be built at a lower cost and shorter lead time, while also being possible to operate on any field. However, regardless of newbuild or conversion, an FPSO with a cargo capacity of 800,000 to 1,000,000 crude oil barrels (bbls) will culminate to capital expenditure (CAPEX) of US\$1-1.5 bn [30, 38]. Hence, the standardised design should be thoroughly evaluated, and hull optimality ensured.

The main activity and centre of operation for an FPSO is the topside processing plant. The plant separates oil, gas and water before the three are shipped off the vessel again. This vessel module requires plenty of deck area and is located high above the waterline, thus shifting the vessel's centre of gravity further away from the waterline. Consequently, it is of interest to maximise the deck area of an FPSO in order to fit as much as possible on the deck while having the possibility for additional retrofitted modules. Furthermore, the question of intact stability arises when more weight is added high above the waterline, and thus the hull geometry is a vital part of the vessel's operability. In theory, a larger vessel would both generate a massive deck area as well as minimise the vessel's movements in high seas. However, it is often not necessary with a cargo capacity of more than 1,000,000 bbls, and hence a larger ship may increase the vessel's CAPEX unnecessarily [30]. As a result, Altera has decided that their standardised FPSO design should be able to carry one million barrels and their wish is reflected in the problem description of the optimisation.

If one's goal is to increase the deck area, an optimisation will continuously increase the vessel's dimensions. The same will likely happen if one aims to minimise the vessel's motions. However, if the build cost is accounted for, the optimisation is constrained by a secondary objective, and one achieves a convergence resulting with an optimised ship [29]. Vessel optimisation may also have the added benefits of long-term reduction of onboard personnel, eventually producing lean FPSO's with lower manning or even unmanned vessels. Consequently, any entity engineering and manufacturing FPSO's seeks robustness in the optimisation that will help accelerate both the procurement and engineering process. The study of optimising a vessel at an early design phase is therefore particularly important as it seeks to improve the project profitability in Altera in the long run.

1.2 FPSO and offshore market outlook

Due to a highly volatile market with rapid changes in commodity prices and time-charters, one should bear in mind that the forthcoming section was written in May 2020 with approximately 1-2 week old information.

From the onset of the worldwide Covid-19 pandemic, the offshore oil and shipping industry has experienced rapid changes and continuously high volatility. In April of 2020, the West Texas Intermediate (WTI) went negative for the first time in history, closing at -\$37.63 USD/bbl, sending shock-waves into the offshore energy markets. During the six weeks after, the domestic count of oil and gas rigs has fallen in the by 347 rigs in the US itself as the Brent spot oil price continues to plummet, and is currently at -46.55% since January 1st (YTD). The price drop is illustrated in figure 1.2.1. The G20-supported OPEC+ agreement in April to cut oil production was branded as a big success but did not fool the oil market. Despite substantial cuts from member countries and worldwide producers, analysts believe that the market will not be restored to its pre-2020 norm with quashed demand until Q3 2020 [47, 48].

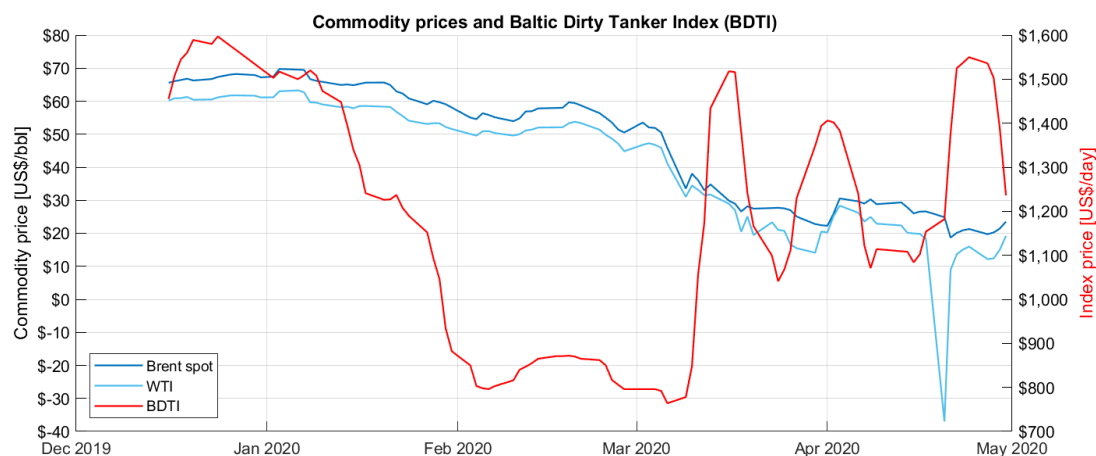


Figure 1.2.1: Commodity prices for Brent spot, WTI and Baltic Dirty Tanker Index (BDTI). The index is based on the weighted average cost of 17 routes carrying unrefined petroleum oil.

Crude oil tanker owners, always masters of volatility, have been able to play the offshore market better than most. Due to increased storage of crude in VLCC's in Asia, the time-charter equivalent on the Middle-East China route (TCE-TD3) has experienced the fourth spike since October 2019, surpassing US\$220,000/day. The surging freight rates have also trickled down to Suezmax tankers due to their lower demurrage costs. Consequently, Aframaxes have followed suit with the same effect being seen on other routes to the UK, the North Sea and the Baltics. Furthermore, the current oil prices create a structure of steep contango, thus allowing traders to charter-in tankers for floating storage at phenomenally high rates. The surge pricing has also forced multiple LR tank-owners to switch from clean cargo to dirty tankers in search of arbitrage opportunities [48]. S&P Global Platts claims that there is currently 300 mill. bbls of crude oil in floating storage, with an additional 100 mill. bbls of clean tonnage [19].

The low crude prices have hurt production and exploration both on land and offshore. While the daily worldwide production is considerably reduced since Q1 2020, the global over-supply has generated a backwardation in the short-term forward production market. Therefore, cheaper production facilities offshore are necessary to meet today's market challenges. For FPSO owners, the cheapest and fastest option to first oil is often through the conversion of either Suezmax' or VLCC's with a minimum delivery time of 24 months [35]. However, the surging rates in the tanker market push the sale prices upward as ship owners jockey for position on the medium-term time-charterers. Consequently, the added benefit of converting a tanker into an FPSO may soon be outweighed by the increased acquisition cost adding to an already staggering CAPEX.

Consequently, future FPSO's on order and in the planning processes need to be able to compete with the current market situation by lowering CAPEX and lead time. Current estimates suggest there are approximately 300 operational vessels worldwide, with another 30 vessels available. Furthermore, it is estimated that over 50 vessels are on order with an additional 224 in the planning stages [7]. However, the current fleet of FPSO's and those in planning stages, 68% are built from conversions from former tank vessels, and thus only 32% being purpose-built at a yard. Additionally, reports from a global FPSO survey suggests that about 75% of the currently operational FPSO's are located in benign waters off South America, south-east Asia and along the west coast of Africa. While a majority of vessels are located in calm waters, increased offshore activity in the North Sea and Barents Sea spurs the development of internal turret moored FPSO's that can withstand the harsh weather conditions. A conversion scheme with standardised vessel modules and topsides is increasing in popularity. While conventional conversions or newbuilds would usually take 24 to 36 months from FPSO contract to first oil, standardised and hull geometry and topside solution can reduce the delivery time by 6 to 12 months. The schedule reduction is applicable for both conversions and newbuilds. Consequently, a generic hull that can withstand these conditions is highly dependent on a thorough optimisation and evaluation.

While the CAPEX of the vessels is seemingly reduced through fast and flexible conversions, other onboard factors have significant influence over the life cycle costs. Better analytical tools and an increased degree of digitalisation for onboard monitoring systems have helped prolong the lifetime of multiple vessels as a result of better fatigue level measures. Several entities are now considering more extensive, innovative steps for the offshore market and FPSO's. As the environmental factors gain attention, solutions regarding carbon capturing and storage (CCS), recovery of volatile organic compounds (VOC Recovery) and onshore power supply must be considered in the vessels' design. Furthermore, innovative solutions such as the unmanned wellhead platform *Oseberg Vestflanken 2* shows that lean FPSO's with lower manning or completely unmanned may be possible in the coming years.

1.3 Key performance indicators

The key takeaways from Altera’s interests as well as the current FPSO market outlook show that there are multiple sets of quantifiable measurements that one can study to evaluate the long-term performance of a vessel, commonly called key performance indicators (KPIs). Consequently, before the optimisation is undertaken, one has to determine which of the available KPIs are of particular interest and which of them can be validated with the current information. Table 1.3.1 summarises the KPIs that are of interest to Altera Infrastructure. These KPIs are also the general focus area involving the *Generation IV* of their newbuild and converted FPSO units.

Table 1.3.1: Summary of key performance indicators for hull design that are of interest from Altera Infrastructure.

KPI	Description
Seakeeping performance	Minimisation of vessel motions to comply with offshore regulations and increase vessel operability
Lightship weight	Minimisation of lightship weight to minimise steel cost
Topside capacity (area and weight)	Increased area for larger processing plant. Maximum weight to be maximised given compliance with intact stability criteria
Weathervaning	Involves optimal placement of turret and minimisation of installed effect for weathervaning propulsion systems
Standardisation	Degree of standardisation and cost of flexibility. Minimisation of lead time with regards to conversions and newbuilds
Vessel CAPEX	Minimisation of build cost

Considering the KPIs in table 1.3.1, one has a broad range of possible optimisation objectives. However, a particular focus area for Altera is the topside capacity and CAPEX. Consequently, ratios involving the lightship weight per barrel produced or deck area per barrel is of interest. Furthermore, the CAPEX per barrel produced is also necessary to consider if a complete operational cost analysis is conducted. While considering all KPIs as an objective function would likely yield stimulating results, insufficient time limits the number of focus areas. Since the majority of the discussion with Altera consisted of matters relating to the topside capacity and seakeeping abilities, these KPIs will be the main focus of the following thesis work.

1.4 Objectives

The research objective of this master thesis is to generate a preliminary hull geometry that is optimised for both maximum deck area and hull build cost through the use of an optimisation model. The process involves understanding how settings and parameters of the optimisation will change the output and its effect on the resulting hull geometry.

Since the the FPSO units are located in harsh weather conditions, their optimality must be measured against their operability and response in various sea states. An evaluation of the vessels' motions will decide whether the optimal ships indeed possess the best possible geometry as well as determining any possible shortcomings. The objectives can, therefore, be broken down into several sub-objectives as listed below and are likely to be necessary steps for the optimisation process and optimal hull selection:

1. Describe the general characteristics and the relevant operating context of an FPSO unit
2. Study relevant topics that apply to the optimisation and objective functions
3. Develop geometric model feasible for optimisation and objective functions while giving realistic results
4. Create loading conditions and weight distributions based on optimisation output data
5. Evaluate the most influential input parameters affecting the optimisation algorithm
6. Explore and identify methods that guarantee convergence in the optimisation at the global optimum
7. Develop a framework to fairly evaluate the best individual solution points and complete sets of Pareto optimal solutions
8. Analyse the hulls' motions in regular waves
9. Discuss and conclude the reliability of the optimisation model and its results

1.5 Scope and limitations

The work in this thesis involves a multi-objective optimisation using two objective functions that aims to maximise the deck area and reduce the build cost. Consequently, the overall aim is to generate one or two optimal hull geometries. The optimisation will be conducted through the use of a multi-objective optimisation algorithm that can guarantee a convergence and generate optimal solutions. The optimisation output must then be evaluated to determine the optimal vessel(s) as well as the optimisation run that generated the best solutions. Included in the optimisation is the general hull geometry as well as placement of cargo tanks and water ballast tanks. The main activity of the work is the optimisation of the hull. If one were to include the position and shape of the accommodation, processing plant, flare tower or helicopter deck a much more comprehensive system build up is required. Consequently, these vessel modules are not included in the scope, nor the model and will therefore not affect the optimisation. Hence, the corresponding build cost only includes the hull's paint and steel cost, and no outfitting is included in the final price estimate. Additionally, the optimisation model considers the vessel's motions as a termination criterion. Hence, if the vessel's motions are too severe, the algorithm must re-iterate. Subsequently, a more thorough analysis is required as part of the post-processing to determine the validity of the results.

The optimisation will only focus on the relationship between the hull geometry and the resulting motions in regular waves. Due to limited time and the complexity of the task, no evaluation of structural integrity, wave resistance, propulsion study or weathervaning study will be completed.

Literature review

2.1 FPSO Characteristics

When one is to optimise the dimensions of an FPSO, it is important to understand the main characteristics of the vessel in terms of operation and safety. If one disregards the market's influence on the hull design, there are still some clear trends that characterise a typical FPSO. If the vessel is not purpose-built at a yard, an FPSO is often a product of a converted oil tanker, usually in size range of a Suezmax or a very large crude carrier (VLCC) that carry typically 150,000 DWT to 320,000 DWT, respectively [9]. In the event of a conversion, one aims at re-using as much as possible of existing vessel systems and equipment to minimise CAPEX for the conversion. The crude oil tanks may be re-used for the FPSO phase, but the entire topside processing plant has to be installed as well. While a purpose-built FPSO may have a completion time of 24-36 months from contract to delivery, a conversion may reduce the lead time to 6-12 months through implementing a more conventional vessel fabrication line [13].

Furthermore, FPSO's are a particular type of vessels that require complex mooring systems. The vessels are stationary most of their lifetime and are secured to the seabed through a variety of possible mooring systems. The mooring-system is determined by the specific weather environment and water depth. With a reliable mooring system, the vessels are suitable for a wide range of environmental conditions and are usually designed for the capability of stationery operation for over 20 years without dry-docking. There are various types of mooring systems and anchor patterns, but the most common are internal or external turrets as well as spread moorings. A visual example of the mooring systems is given in figure 2.1.1.

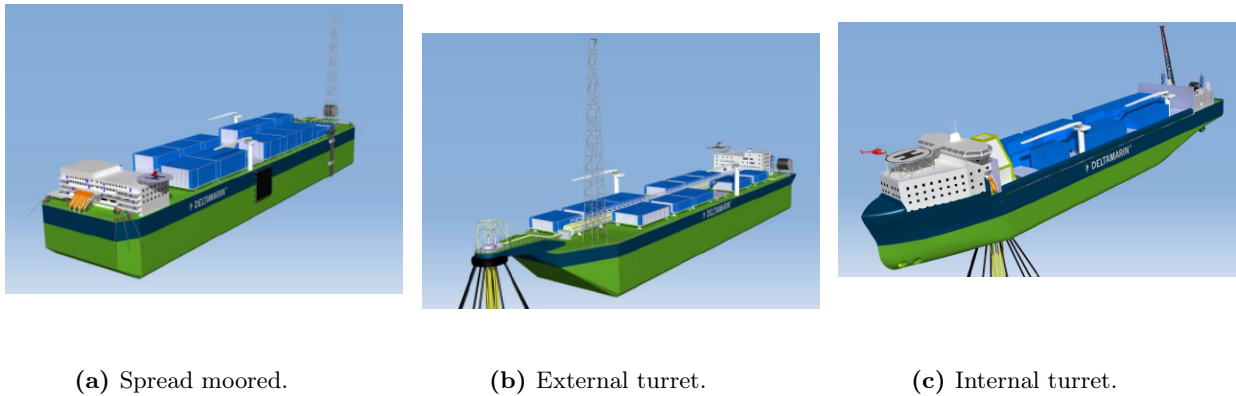


Figure 2.1.1: Various FPSO mooring systems. From left to right: spread mooring, external turret mooring, internal mooring. Photo properties belong to Deltamarin [27].

Figure 2.1.1c represents a vessel with an internal turret mooring. The turret is moored to the seabed with chains and anchors in a radial pattern and has bearings allowing free and unrestricted 360° rotation [33]. The vessel will then adjust its heading according to the weather and will normally lay head to the prevailing environment, thus continuously weathervaning. The vessel with an external turret possesses many of the similar weathervaning capabilities as the internal turret but is likely in need of less energy to adjust the heading at the cost of being less susceptible to harsh weather [5]. The vessel on the left is spread moored. This mooring configuration is often sufficient in calmer waters where weathervaning is unnecessary. Additionally, some FPSO's may have disconnectable mooring systems, allowing the vessel to escape from rough weather like hurricanes or cyclones [27].

While the position of the turret and mooring type is outside the scope of this work, an important factor to consider is the position of the living quarters (LQ) on the vessel. The exact position of the LQ is not of great importance for the optimisation algorithms, but the general position does affect the shape of the hull in practice. However, should the LQ be situated in the bow, the structure has to be protected from green sea. Consequently, green water effects can be mitigated by providing a flare angle in the bow. Conversely, if the LQ is abaft, the flare tower and processing plant must be equally protected. The NORSOK standard state the following; "The installation shall be oriented with respect to the dominant wind direction to minimise the likelihood of a gas release or smoke drifting towards the accommodation and primary evacuation means" [34]. Since the internally turret-moored units are likely to lay head to the wind, most accommodation structures are located in the bow if designed according to NORSOK. Consequently, the volume and geometry of the fore section become a vital part of the optimisation. Furthermore, regardless of the positioning of the LQ, the lifeboats must be positioned above the 10,000-year relative wave height profile.

2.2 Vessel response in regular waves

2.2.1 Sea states at location

The study in this thesis aims at improving a current vessel design that is placed on a location with one of the world's harshest ocean environments. Due to classified information from several corporations, the vessel's location will be called location \mathcal{X} and is situated west of Shetland. Its water depth is for all practical purposes 400 metres. The wave and wind data used for the evaluation of the vessel's response is based on 21 years of continuous hindcast at an undisclosed grid point near location \mathcal{X} . Additionally, the data also comprises of measurements done by other vessel and drilling rigs in the same area. When considering the weather data as an input to calculate the vessel motions, data from both a wave frequency table and estimated data concerning return-periods can be used. Table 2.2.1 represents the latter, showing the significant wave height (H_S) and the peak period of the wave (T_p) given in metres and seconds, respectively. Significant wave height is defined as the average wave height, from trough to crest, of the highest one-third of the waves. The calculated values are based on a JONSWAP wave spectra using a Weibull distribution with γ -range of 2.0 to 3.33 [32].

Table 2.2.1: JONSWAP wave spectra. Central γ : 2.6, σ_a : 0.07 σ_b : 0.09.

	1-Year	10-Year	50-Year	100-Year	1K-year	10K-year
Hs [m]	13.1	15.3	16.9	17.5	19.9	22.2
Tp [sec]	16.5	17.9	18.9	19.2	20.4	21.5

Another important aspect when calculating H_S and T_p for a given return period is through the use of a site-specific wave frequency table. Such a table summarises the number of occurrences of each wave with given wave height and period occurring at a given location. An example of such a table is given in appendix A. This table is used to determine the long-term vessel response in long-crested waves.

Altera has previously adhered to the rules and regulations of DNV GL; hence the same criteria will be considered in the following work. Weather data from table 2.2.1 and appendix A shall be used when considering the vessel's motions on the production site. To test the ultimate hull strength assessment, the design bases for wave load analysis comprises of the following situation for a turret moored unit as stated in DNVGL-OS-C102 [17]:

- Head sea (0°): 60% of operating time
- $\pm 15^\circ$ of head: 30% of operating time
- $\pm 30^\circ$ of head: 10% of operating time

These requirements will be applied for multiple analyses as it represents the vessel's heading probabilities and may affect the vessel's overall operability.

2.2.2 Response functions

As seen from the sea-states presented in table 2.2.1, location \mathcal{X} is often subject to very harsh conditions. The large sea-states will often result in large motions for the FPSO units on location and is likely further to affect the vessel's operation and crew capability. Hence, it is of interest to Altera to evaluate the motions of the vessel at location \mathcal{X} , and thus the frequency response functions must be derived.

The most common practise when calculating the response functions is through the use of 2D strip theory, calculating both the excitation forces and the added mass, damping and restoring forces. However, if this calculation is done for each vessel, for all frequencies, throughout the optimisation, computation time would grow exponentially. Hence, a simpler model derived from empirical methods has been used in the optimisation.

Simplified seakeeping model

The maximum wave-induced motions and accelerations a ship may encounter during its operation lifetime can be undertaken by considering the hull form, its operational profile and mass distribution. While a linear analysis using two- or three-dimensional hydrodynamic procedures based on potential theory is quite straightforward, a lack of detailed vessel data, significant expertise and time may render this procedure futile when optimising. Therefore, a more straightforward method of predicting wave-induced motions have been developed, giving sufficient engineering accuracy in the conceptual design phase [24]. The formulas derived from Jensen et al. are therefore semi-analytical, considering only length, breadth, draught, block coefficient, waterplane area and operational profile.

The simplification assumes that the heave and pitch motions are uncoupled, that the vessel has a constant sectional added mass equal to the displaced water, and that the vessel is homogeneously loaded. Given these assumptions, the equations of motions in a regular wave, with amplitude a can be written as;

$$2\frac{kT}{\omega^2}\ddot{w} + \frac{A^2}{kB\alpha^3\omega}\dot{w} + w = aF \cos(\varpi t) \quad (2.2.1)$$

$$2\frac{kT}{\omega^2}\ddot{\theta} + \frac{A^2}{kB\alpha^3\omega}\dot{\theta} + \theta = aG \sin(\varpi t), \quad (2.2.2)$$

where k is the wave-number, ω is the wave frequency (assuming $\omega^2 = kg$), and where B and T are the breadth and draught of the box respectively. Differentiation with respect to time t is denoted by a dot. A more comprehensive presentation of the method can be found in appendix E. Consequently, the solution to equations (2.2.1) and (2.2.2) gives the frequency response functions;

$$\Phi_w = \eta F \quad (2.2.3)$$

$$\Phi_\theta = \eta G, \quad (2.2.4)$$

where η is given by

$$\eta = \left(\sqrt{(1 - 2kT\alpha^2)^2 + \left(\frac{A^2}{kB\alpha^2} \right)^2} \right)^{-1}. \quad (2.2.5)$$

The frequency response functions for the vertical motion u and acceleration v in the longitudinal position x from the vessel's centre of gravity then becomes:

$$\Phi_u = \sqrt{\Phi_w^2 + x^2 \Phi_\theta^2} \quad (2.2.6)$$

$$\Phi_v = \varpi^2 \Phi_u = \alpha^2 kg \Phi_u. \quad (2.2.7)$$

Φ_w and Φ_θ represent the response amplitude operators (RAO's) for the heave and pitch at position x from the centre of gravity. The model also formulates the equation of motion in equation (2.2.8), and the response frequency equation solved for the roll motion is given in equation (2.2.9).

$$\left(\frac{T_N}{2\pi} \right) C_{44} \ddot{\varphi} + B_{44} \dot{\varphi} + C_{44} \varphi = M \quad (2.2.8)$$

$$\Phi_\varphi = \frac{|M|}{\left(\left[-\varpi^2 \left(\frac{T_N}{2\pi} \right)^2 + 1 \right]^2 C_{44} + \varpi^2 B_{44}^2 \right)^{\frac{1}{2}}} \quad (2.2.9)$$

VERES model

The simplified vessel motions model described in section 2.2.2 assumed a rectangular box with a constant sectional added mass and only considered the uncoupled motions of heave, pitch and roll. Since the vessel may experience incoming sea with β as much as 30° , it may also be important to consider the coupled pitch and roll motions. However, like the simplified model, the VERES program (the calculation tool in ShipX for vessel motions) also have some important assumptions as stated from the ShipX Vessel Response Theory Manual [16]:

- The ship is assumed to oscillate harmonically with a frequency equal to the frequency of encounter. No transient effects due to initial conditions are considered. No hydroelastic effects are considered.
- A linear relation is assumed between the responses and the incident wave amplitude. This assumption will not be correct when considering water on deck and slamming in high seas.
- The superposition principle can be used to derive the loads and motions in a sea-state.

- Potential theory is applied, and the fluid is assumed to be homogeneous, non-viscous, irrotational and incompressible. Viscous roll damping can be accounted through the use of empirical formulae.
- The vessel is assumed to be slender, i.e. the length of the hull is much larger than the breadth and draught.
- The vessel symmetric along the centre line.

Furthermore, the VERES program is based on linear strip theory with the following important assumptions:

- The wave-amplitudes are small compared to some characteristic dimension of the vessel. The resulting motions will then be proportionally small.
- The wave steepness is small, i.e. the waves are far from breaking

In the linear theory used, the wave loads and motions are linearly proportionate to the wave amplitude. If one wishes to evaluate the vessel's behaviour in irregular waves, one can obtain results by adding together results from regular waves of different amplitudes, wavelengths and propagation directions. While this feature shows the model's flexibility, it is assumed that the FPSO will only encounter regular waves.

Since the VERES model assumes coupled motions in all six degrees of freedom (DoF), the model creates individual equations of motions for the j DoF's. Under the assumptions that the responses are linear and harmonic, the six, coupled differential equations of motions can be written as;

$$\sum_{k=1}^6 [(M_{jk} + A_{jk}) \ddot{\eta}_k + B_{jk} \dot{\eta}_k + C_{jk} \eta_k] = F_j e^{i\omega t}, \quad j = 1, \dots, 6. \quad (2.2.10)$$

M_{jk} is the element of the generalised mass matrix, A_{jk} is the element of the added mass matrix, B_{jk} is the element of the linear damping matrix and C_{jk} is the element of the stiffness matrix. Furthermore, ϖ is the angular frequency of encounter, while η_k is the surge, sway, heave, roll, pitch and yaw motion amplitudes. The dots stand for time derivatives so that $\dot{\eta}_k$ and $\ddot{\eta}_k$ are the velocity and acceleration terms, respectively.

F_j is the complex amplitude of the wave exciting forces and moments, with the physical forces and moments given by the real part of $F_j e^{i\omega t}$. Consequently, F_1 , F_2 and F_3 refer to the amplitudes of the surge, sway and heave exciting forces respectively, while F_4 , F_5 and F_6 are the amplitudes of roll, pitch and yaw exciting moments, respectively.

2.3 Vessel response criteria

While DNV GL has a set of regulations to be followed, requirements from the customer or third-party operators have to be considered as well. Such requirements include limitations on horizontal movement (excursion envelope) as well as helicopter operation. Additionally, internal

procedures have been implemented by Altera to safeguard the onboard crew and equipment on deck.

2.3.1 Operational criteria

Part of an internal safety measure is the closing of the personnel elevator when the vessel experiences large displacements. The closure does not necessarily lead to a temporary production halt, rather an inconvenience in terms of operating heavy equipment and limited movement on deck. For some of Altera's FPSO units, the following criteria applies: maximum 8.5 metres of heave, $\pm 10^\circ$ of roll and $\pm 5^\circ$ of pitch.

2.3.2 Helicopter criteria

Crew transport to and from the vessel is primarily done with helicopters. Most of the landing platforms are often situated close to, and above the accommodation structure. Consequently, the vessel must adhere to the regulations set by the Helideck Certification Agency's Helideck Landing Limitation [3]. The limitations depend on the type of helicopter used for the operation, as more agile and advanced helicopters have fewer limitations. The limiting conditions stated in table 2.3.1, therefore represent the strictest criteria (for the least agile aircraft) and will thus be used as the criterion for all helicopter operations.

Table 2.3.1: Helideck limitations list for aircraft in category A and helideck category 1 [3].

Limiting condition	Limiting criteria DAY	Limiting criteria NIGHT
Helideck pitch/roll	$\pm 3.0^\circ$	$\pm 2.0^\circ$
Helideck inclination	$\pm 3.5^\circ$	$\pm 2.5^\circ$
Helideck avg. heave rate	1.3 m/s	1.0 m/s
Helideck heave amplitude	5.0 m	4.0 m

2.3.3 Excursion envelope

An excursion of a vessel entails horizontal movement from its intended position, and thus the excursion envelope is the area of allowable displacement in sway and surge of the vessel. Although not a specific requirement from a given classification society, many operators and customers require a maximum excursion envelope of 10-25%. Hence, on a water depth of 400 metres, the vessel may have a maximum horizontal displacement of 40 metres if the excursion envelope is 10%. If the risers from the bottom are flexible, an excursion envelope (E_x) of 25% is usually required, and a 10% for static steel risers [30].

However, to correctly estimate the vessel's movement, one has to calculate the 2nd-order motions. By solving a second-order problem, one can calculate the mean and oscillating forces of the vessel, i.e. the oscillating surge and sway motions of the vessel that are different from the wave-frequency of encounter [15]. Furthermore, the stiffness and damping of both the mooring and the risers have to be considered to forecast the vessel's movements accurately. Second-order effects are a represents a considerably more complex task and is not in the scope of this thesis. The vessels' motions presented here are only first-order linear motions.

2.3.4 Motion sickness criteria

The British Standards Institution was the first to present a quantitative guideline for estimating the incidence of motion sickness from a parameter called motion sickness dose value. This parameter aims at evaluating human exposure to whole-body vibration at low frequencies. The parameter is defined by the root mean square (RMS) accelerations and accounting for both exposure time and frequencies. This quantitative measure was then adapted by the International Towing Tank Conference (ITTC) to be a measure of both seasickness and fatigue on high-speed marine vehicles. Furthermore, the International Organisation for Standardisation (ISO) presented more specific methods to measure and evaluate the human response to vibrations. Through the use of better specification for vibration instrumentation and introduction of analogue filters in the frequency domain, the MSDV could be calculated more accurately through measured vertical acceleration signals [25]. Consequently, the MSDV is calculated using the British Standard BS 6841/3/, and is defined as follows:

$$MSDV = \sqrt{\int_0^T a^2(t) dt}, \quad (2.3.1)$$

where $a(t)$ is the frequency-weighted acceleration and T is the total period (in seconds) during which the motion should occur. T is set to a maximum of 2 hours. According to the regulations, the percentage of adults (both male and female), likely to vomit (V_p) is approximated by:

$$V_p = \frac{1}{3} \times MSDV. \quad (2.3.2)$$

Hence, an MSDV (measured in $m/s^{1.5}$) value of 15 is equal to 5% of the personnel on board vomits. This number represents the safety limit, i.e. less than 5% of the personnel on board are motion sick 5% of the time per year. Consequently, the following acceptance criteria apply for 95% of the time per year:

$$MSDV < 15. \quad (2.3.3)$$

The acceptance criterion is applicable for all six degrees of freedom [18].

In correspondence with Ø. Røyseth (Naval Architect, Altera Infrastructure), it has been pointed out that DNV GL Advisory has no experience in where $MSDV < 15$ for any commercial tank vessel, nor FPSO. A study conducted by the advisory confirmed the statement while also suggesting that an upper limit of V_p to just 6-7%, would result in total compliance for most vessels.

2.4 Multi-objective optimisation problem

A thorough analysis and evaluation of several optimisation models have been studied in the project thesis written in the autumn of 2019. Here, a genetic algorithm was deemed the most expedient to optimise a hull's geometry. Consequently, experience shows that the genetic algorithm is highly reliable and produces accurate results when considering vessel hulls. Additionally, Altera Infrastructure also uses genetic algorithms for their optimisation of FPSO units showing that the methodology is thoroughly tested and well suited.

When faced with an optimisation problem where one must evaluate interdependent goals, one must carefully choose the optimisation method to achieve accurate results. Most optimisation solving methods can be categorised into either classical methods or evolutionary methods. A genetic algorithm represents the latter, and albeit with many benefits, it is not always the best way to solve every problem. If it is possible to apply, calculus-based methods will outperform the GA and quickly find the solution of a convex analytical function of a few variables. In such cases, the calculus-based algorithm will have found the minimum by the time the GA is still calculating the costs of its initial population.

However, most realistic problems include a large number of variables and large feasible areas. Additionally, a complex optimisation problem would include multiple objective functions, allowing the GA to compute each function in parallel. Furthermore, the GA can optimise a large number of variables that can either be continuous or discrete, as well as operating with extremely complex cost surfaces enabling it to jump out of local minima. Lastly, and importantly for the work to be done in this thesis, the GA provides a list of optimum variables and not just a single solution, allowing the user to select multiple optimal solutions depending on a given selection criterion.

2.4.1 Formulation

A genetic algorithm used for multi-objective optimisation is a search heuristic inspired by Charles Darwin's theory of natural evolution. The algorithm reflects the process of natural selection where the best solutions are selected for reproduction in order to produce better solutions for the next iterations. The genetic algorithm was developed by John Holland in the 1970s and was later popularised by his student David Goldberg, who solved a difficult problem involving the control of a gas pipeline transmission in his dissertation [21, 22]. The genetic algorithm creates a population of individuals represented by chromosomes which are a series of character strings. These chromosomes are then tested against the objective function repeatedly, to see what solutions are optimal. In multi-objective optimisation problems of M objectives and J constraints, the problem can be formulated as:

$$\min_{x \in \mathbf{X}} \{f_1(x), \dots, f_M(x) | g_j(x) \geq 0, j \in [1, J]\}, \quad (2.4.1)$$

where we search for design alternatives x in the design space \mathbf{X} confined within variable the bounds.

The goal is to find such x that will minimise the objective $f(x)$ while satisfying all the constraints $g(x)$. If all the constraints are satisfied, the solution is feasible and belongs in the feasible set Ω , which is denoted by:

$$\Omega = \{x \in \mathbf{X} | g_j(x) \geq 0, j \in [1, J]\}. \quad (2.4.2)$$

The solution to equation (2.4.1) is a Pareto optimal alternative, represented by $x^{*(i)}$. This means that there is no better alternative than $x^{*(i)}$ in the objective space \mathbf{Y} (whose feasible space is denoted with \mathbf{Y}^Ω) [28]. If $x^{*(i)}$ is the vector that optimised the i^{th} objective in the multi-objective optimisation problem with M conflicting objectives:

$$\exists x^{*(i)} \in \Omega, x^{*(i)} = \left(x_1^{*(i)}, x_2^{*(i)}, \dots, x_M^{*(i)}\right)^T \text{ where } f_i(x^{*(i)}) = \min f_i(x) \quad (2.4.3)$$

the vector z^* can then be written as

$$z^* = f^* = (f_1^*, f_2^*, \dots, f_M^*)^T, \quad (2.4.4)$$

where f_M^* is the optimum of the M^{th} objective function. If f^* is closest to the reference point in the feasible area, it will determine this as the ideal objective vector. However, ideal objectives vectors do not necessarily exist for all multi-objective optimisations (MOP). Since the optimal solution for each objective is not necessarily the same solution, the ideal vector, only exists if all the optimal solutions of all objectives are identical.

It is also important to note that if all the objective functions, as well as the constraints of the optimisation, are linear, the problem can be categorised as a linear optimisation problem (MOLP). In contrast, and focus for the following sections, both the objective functions and multiple constraints are non-linear, rendering a non-linear multi-objective optimisation problem, typically referred to as MOOP [4].

Dominant solutions

In MOP's, it is unlikely that there is a single solution that is optimal for all the objectives. Consequently, there is a set of optimal solutions known as Pareto-optimal solutions that are generally equally satisfactory. If an alteration of variables cannot improve one objective function value without worsening another, the solution is Pareto optimal. When several Pareto optimal solutions are generated, they form a Pareto front which is ordered based on dominance.

Given a solution A and B , solution A dominates B if A has a lower cost (better objective function value) than B for at least one of the objective functions and is not worse concerning the remaining objective functions [21]. Dominance can be written in more general terms with the following conditions:

1. The solution $x^{(A)}$ is no worse than $x^{(B)}$ with respect to all objective functions value, or $f_j(x^{(A)}) \not\geq_j (x^{(B)})$ for all $j = 1, 2, \dots, M$.
2. The solution $x^{(A)}$ is strictly better than $x^{(B)}$ in at least one objective function value, or $f_{\bar{j}}(x^{(A)}) < f_{\bar{j}}(x^{(B)})$ for at least one $\bar{j} \in \{1, 2, \dots, M\}$.

Consequently, a solution is Pareto optimal if no other solution dominates that solution with respect to other objective functions. Given that the optimisation has generated a non-dominated set, a Pareto-optimal set exists, and a Pareto-front can be generated from the optimisation.

2.4.2 Pareto front

If the Pareto-optimal set (containing no dominating solutions) is within the feasible search space, the set can be characterised as a global Pareto-optimal set. Figure 2.4.1 illustrates a Pareto-front of two objective functions, when maximising A and minimising B .

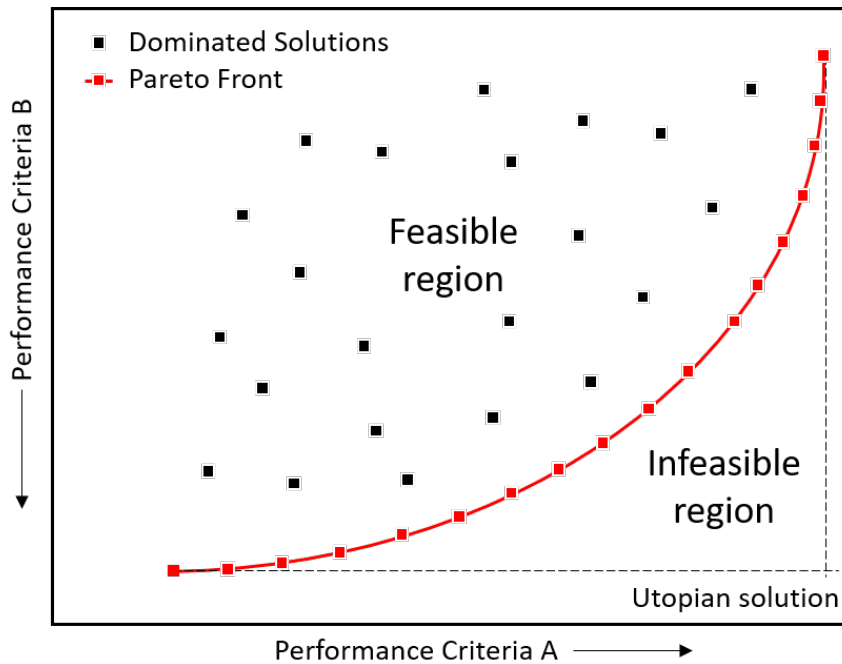


Figure 2.4.1: Example max-min system with two competing performance criteria, A and B.

The Utopian point, U_P , represents the point of optimal value for both objectives. However, since it is within the unfeasible region of the solution space, this is only a theoretical point, the genetic algorithm is attempting to reach. Furthermore, one can observe that the points on the Pareto front are indeed non-dominated, as A or B cannot be improved without detrimental effect on the other [42].

2.4.3 Initial population

Each possible solution in the optimisation problem, generated by the genetic algorithm, is represented by a chromosome. In the mathematical formulation of the optimisation problem, each chromosome is made up of a series of decision variables that represent a possible solution of the optimisation problem at hand, e.g. multiple measures for a vessel's hull. In a N -dimensional problem, a chromosome is an array of size $1 \times N$ and is defined as:

$$X = (x_1, x_2, \dots, x_i, \dots, x_N) \quad (2.4.5)$$

where X represents a feasible solution to the optimisation problem, x_i is the i th decision variable of the feasible solution X , and N is the number of decision variables. The genetic algorithm must then generate the initial population with a random generation of chromosomes. The population size, i.e. the number of possible solutions is denoted as M . The possible solutions based on the population is represented by a matrix of chromosomes of size $M \times N$ [10].

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_j \\ \vdots \\ X_M \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,i} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,i} & \cdots & x_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{j,1} & x_{j,2} & \cdots & x_{j,i} & \cdots & x_{j,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{M,1} & x_{M,2} & \cdots & x_{M,i} & \cdots & x_{M,N} \end{bmatrix} \quad (2.4.6)$$

Here, X_j is the j th solution and where $x_{j,i}$ is the i th decision variable of the j th solution with a population size of M . Each variable represents a floating-point number. While the rows M of the initial population matrix may represent the number of possible solutions, the GA can generate fewer or more non-dominated solutions due to selection and mutation that occur for each iteration. Depending on the algorithm's input settings (such as Pareto Fraction and Generations), a larger initial population size gives the algorithm greater freedom to choose the best rated solutions. Hence, a $1 \times N$ initial matrix X_0 is likely to render few non-dominated solutions. In contrast, huge initial population size increases the computation time as a score has to be given continuously to each individual solution.

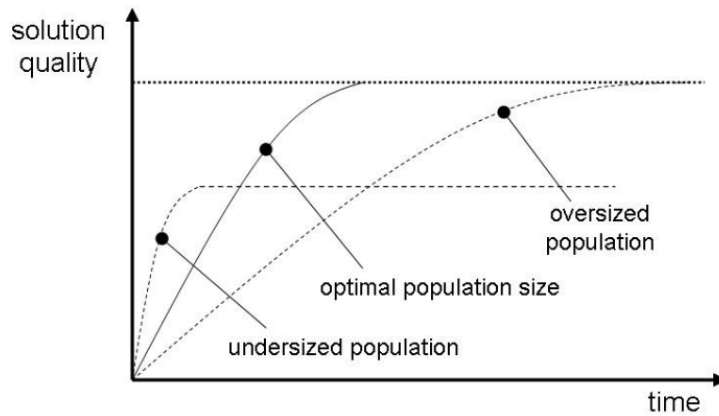


Figure 2.4.2: Effect of solution quality with varying population size, with respect to computation time [44]. Solution quality may be given as collective measure of the goodness of the optimisation.

Figure 2.4.2 shows that an oversized population will eventually reach the best possible solution quality after a long computation time, while the undersized population will never reach the best solution. Thus, the key is to estimate the right population size to minimise the computation time while still reaching the best possible solution quality.

2.4.4 Events in the genetic algorithm

The genetic algorithm is built up of the following five phases:

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

The initial population X_0 is chosen, and their fitness concerning the objective(s) is calculated. The fittest individuals will then be chosen from the population before producing offspring. These new chromosomes inherit the characteristics of the parents and will be added to the next generation. Since the selected parents have a good fit, their offspring will likely be even better and thus the longer chance of surviving the next iteration. There are, however, several methods for selecting the parent chromosome, the most common being proportionate selection and tournament selection.

Selection

Selection is the process where chromosomes are chosen randomly from the initial population X_0 to be further evaluated. The tournament selection chooses each parent by choosing the tournament size players at random and then choosing the best out of that set to be a parent [1]. Determining the most beneficial tournament size can be difficult, as the smallest possible size of 2 will merely choose a random chromosome, while a tournament size equal to the population will choose the entire population for all the following generations. As a rule of thumb, a tournament size of approximately 20% of the population size is favourable [23]. In the MOOP, a tournament size of 20% of the population will, therefore, be chosen. The tournament selection algorithm with k amount of players is described Algorithm 1.

Algorithm 1: Tournament selection with k players

```

for chromosome  $i = 1$  to population size do
    Choose  $k$  individuals from the population at random;
    Calculate cumulative fitness, total fitness ( $P_i$ ) and sum of proportional fitness;
    Choose the best individual from the tournament with probability  $P_1$ ;
    Choose the second best individual with  $P_2 = p \times (1 - p)$ ;
    Choose the third best individual with probability  $P_3 = p \times (1 - p)^2$ ;
    Choose the  $k^{th}$  best individual with probability  $P_k = p \times (1 - p)^k$ ;
end

```

Crossover

In the crossover, the offspring of the initially selected chromosomes are generated and are used in the next generation. There are multiple ways of completing a crossover and choosing crossover points. The simplest method is choosing one or more points in the chromosome to mark as the crossover points. The variables between these individual points are then merely swapped between the two parents. This scheme represents a scattered crossover, as illustrated in table 2.4.1.

Table 2.4.1: Example of scattered crossover with random binary vector.

Parent 1	p_{m1}	p_{m2}	p_{m3}	p_{m4}	p_{m5}	p_{m6}	p_{mN}
Parent 2	p_{f1}	p_{f2}	p_{f3}	p_{f4}	p_{f5}	p_{f6}	p_{fN}
Random vector	1	1	0	1	1	0	0
Child 1	$\uparrow p_{m1}$	$\uparrow p_{m2}$	p_{m3}	$\uparrow p_{m4}$	$\uparrow p_{m5}$	p_{m6}	p_{mN}
Child 2	$\uparrow p_{f1}$	$\uparrow p_{f2}$	p_{f3}	$\uparrow p_{f4}$	$\uparrow p_{f5}$	p_{f6}	p_{fN}

There are other variations of the crossover algorithm, such as uniform crossover where the algorithm selects N number of points and then randomly choose which parent (if any) will contribute its variable at each position. While a new generation of chromosomes has to be created for each iteration, no new information is introduced at this point. Each of the continuous values in the population continues to the next generation, albeit in different combinations. Hence, if only scattered or uniform crossover is used, the algorithm is entirely reliant on the mutation to introduce new genetic material [21].

This problem, can however be solved by combining variable values from the parent solutions into new variable values in their children. Through this blending method, p_{new} derives from a combination of the two parents:

$$p_{new} = \psi p_{mn} + (1 - \psi) p_{fn}, \quad (2.4.7)$$

where ψ is a random number on the interval $[0,1]$, while p_{mn} and p_{fn} is the n th variable of the father and mother chromosome respectively. Consequently, the second offspring is merely the complement of the first (using ψ instead of $1 - \psi$) [39]. Henceforth, to ensure diversity and independence from the mutation, the GA will adopt the blending crossover method in its optimisation.

Mutation

Since the GA can evaluate very complex cost surfaces, local minima are likely to exist. To deter the algorithm from converging too quickly, routines are placed to explore other areas of the cost surface and ensure convergence at the global minimum. This is done by randomly introducing changes or mutations to the variables [21, 45]. Like the crossover routine, there are numerous ways of creating a mutation, but one of the most common and well-tested methods is through the evaluation of the particles' fitness. Considering the fitness of the solution, the mutation adaption can be written as

$$p_m = \frac{k_1 (f_{max} - f)}{f_{max} - \bar{f}}, \quad (2.4.8)$$

where \bar{f} is the average fitness and k_1 is a mutation constant. This means that the mutation rate p_m is zero for the solution with the maximum fitness. Additionally, $p_m = k_1$ for a solution with $f = \bar{f}$. If the solution has a sub-average value, i.e. $f < \bar{f}$, p_m might take a value above 1.0. To prevent this, these additional constraints are added:

$$p_m = k_2, \quad f \leq \bar{f} \quad (2.4.9)$$

The goal of the mutation is to prevent the GA converging into local optima. This prevention is done by employing the solutions with sub-average fitness to search the cost surface that contains the global optimum. Hence, this solution needs to be completely disrupted. Therefore the values of k_1 and k_2 are assigned a value of 0.5. Smaller values could be employed if the values of p_m have little variation, however, for maximum variation and assured deterioration of local optima, a high value is chosen [46]. In conclusion, we have that;

$$\begin{aligned} p_m &= \frac{k_1 (f_{max} - \bar{f})}{f_{max} - \bar{f}}, & f \geq \bar{f} \\ p_m &= k_2, & f < \bar{f} \end{aligned} \quad (2.4.10)$$

$$k_1, k_2 \leq 1.0$$

2.4.5 Hybrid approach

When one is faced with a real-world engineering problem, there is often little or no knowledge of the true global Pareto front. Through the use of evolutionary algorithms (EA) such as the GA, one can set a benchmark but cannot always ensure true global optimum by itself. Hence, hybrid methods have been proposed to ensure the convergence close to the true Pareto-front.

A hybridisation, or a hybrid GA, combines the GA's ability to close in on a global minimum - given a large number of variables - and the computation speed of a local search algorithm. Although the GA ensures that the solution trends towards the global minimum, its efficiency deteriorates once in a smaller solution region. It is therefore desirable at one point, that a local optimiser 'takes over' the process to find the minimum. This takeover can either be performed throughout the generations of the GA (working in parallel, with the local optimiser evaluating numerous local minima) or at the end of the GA to evaluate the region with best fitness [4].

Figure 2.4.3 shows an example of the effect of a hybrid GA (HGA). The dashed GA line and the solid of the HGA are not identical to start with due to the random nature of the algorithms. For the HGA (solid line), one can see that the local optimiser is executed after 458 function evaluations. As an example, a hybrid algorithm will begin its local search when there is little improvement after a given number of generations.

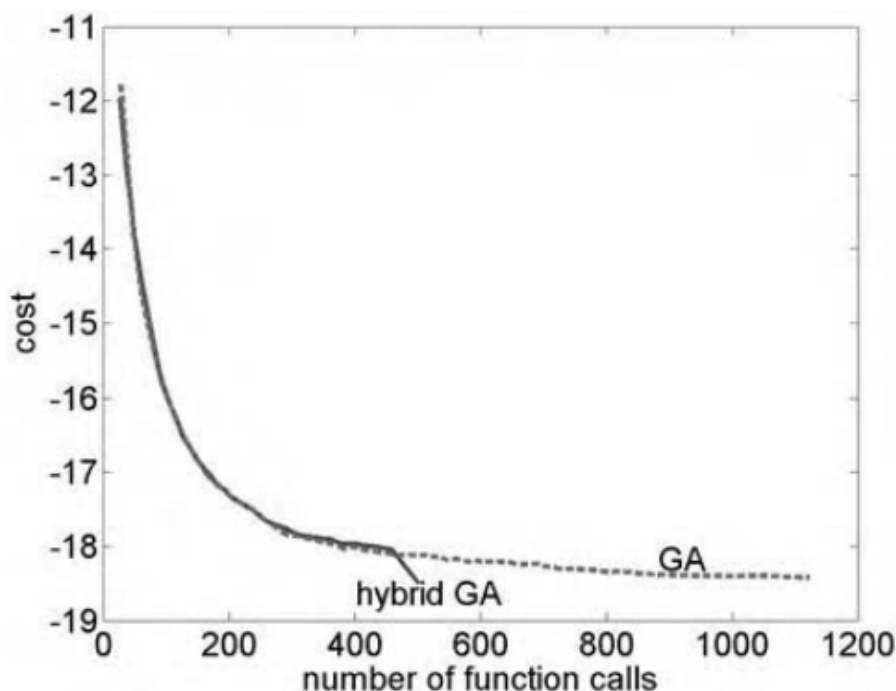


Figure 2.4.3: Example of a hybrid algorithm finding the minimum in fewer function calls than the GA [21].

The HGA will, therefore, use the best chromosome in the current population as the starting point for the local search optimisation. As expected, the HGA finds the global minimum in a fraction of the time taken by the GA.

Figure 2.4.3 shows the possible interaction between local and global search algorithms. The algorithm determines whether a local search should be conducted or if the GA should be rerun on its own. To decide whether the local search is executed or not can be decided by the user or through numeric input.

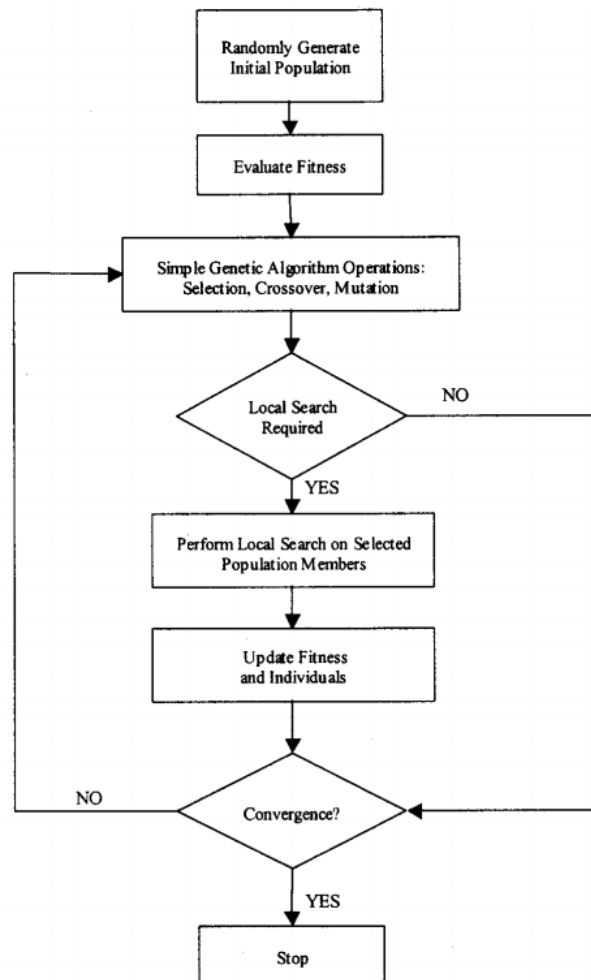


Figure 2.4.4: Flow chart for hybrid genetic algorithm [14].

Like the example in figure 2.4.3, the local search was executed once the gradient of the cost curve was sufficiently close to zero. Alternatively, the user can execute the local search every n th generation given by $\Delta G = n$ on a $P\%$ of the population. By choosing the HGA parameters P and ΔG to be constant, one then chooses a non-adaptive hybrid genetic algorithm (NAHGA).

2.4.6 Hypervolume indicator

When a Pareto front is generated from the optimisation, there are multiple ways of measuring the quality of an optimal set. A popular way to measure the quality, however, is the hypervolume indicator (I_H). The indicator measures the volume of the dominated space bounded from below a reference point [11]. For a two-dimensional optimisation problem like the MOOP in this thesis, the hypervolume is simply an area but can be a volume for a 3-dimensional objective. For an N -dimensional problem, the volume becomes a hypervolume.

By calculating a I_H -value, one is guaranteed strict monotonicity regarding the Pareto dominance, given that a global reference point R is used for all evaluations. Assume that we have a two-dimensional optimisation problem given by $\mathcal{F} : \mathbb{R}^d \rightarrow \mathbb{R}^2$ consisting of the two objective functions $(\mathcal{F}_1(x), \mathcal{F}_2(x)) = \mathcal{F}(x)$. The hypervolume indicator is then given by the objective subspace volume that is dominated by a solution set $A \subset \mathbb{R}^d$. I_H can be defined based on the reference set $R \subset \mathbb{R}^2$;

$$I_H(A) = \lambda(H(A, R)), \quad (2.4.11)$$

where $H(A, R)$ denotes the set of objective vectors that are enclosed on the front $\mathcal{F}(A) = \{\mathcal{F}(x) | x \in A\}$, given by A and the reference set R as illustrated in figure 2.4.5. $\lambda(H(A, R))$ is the characteristic function of $H(A, R)$ [6].

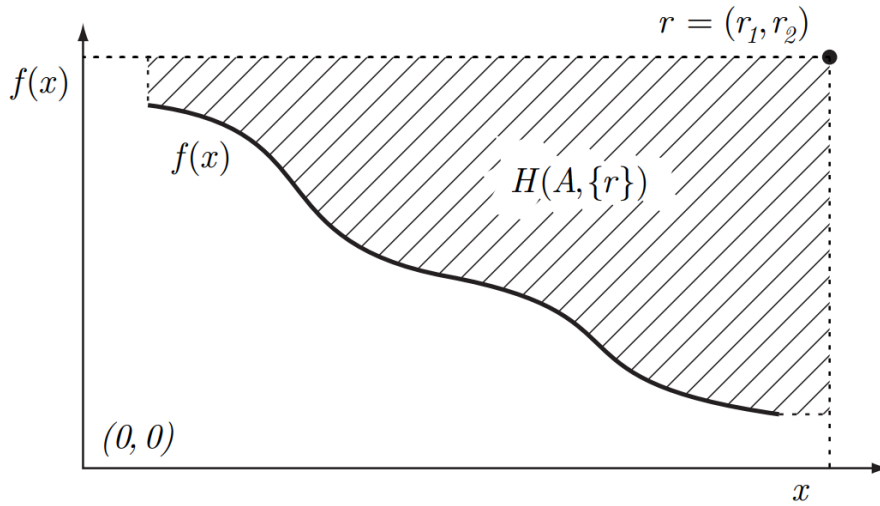


Figure 2.4.5: Example of the hypervolume $H(A, R)$ that corresponds to the set of objective vectors that is dominated by the solution set A , here described by the function $f(x)$, and enclosed by the reference point $R = r$ [6].

Consequently, the hypervolume indicator I_H may be used to evaluate the goodness of the Pareto sets when running multiple optimisations with varying input parameters. Evaluation of individual points on the Pareto set, however, can be measured by their normalised distance to the Utopian point, U_P .

Problem description

The following chapters aim at outlining the problem set-up, the parametric hull generation and the optimisation process so that the optimisation can be performed identically at a later stage.

3.1 Geometry-related optimisation setup

For the MOOP to properly evaluate the deck area of the vessel and the building cost, an accurate hull model must be generated alongside a system set-up that calculates the objective functions value accurately. The simplified hull model, as well as the bounds and constraints for the optimisation problem, were discussed with Altera to ensure that the vessel is adequately realistic for a proper analysis, while also simplified for ease of calculation and reduced computation time.

3.1.1 Simplified hull model

When calculating the deck area and the associated building cost of the vessel, it is desirable with an accurate hull model to get a realistic result in terms of objective function values. However, calculating hydrostatics for double-curved shapes with significant variations in multiple water planes is both time-consuming and a complex task. However, regardless of the ocean's conditions, most FPSO's have simpler shapes that are easier to model parametrically. Hence, almost no double-curved surfaces are used for the model when calculating the vessel's hydrostatic properties. Figure 3.1.1 represents the simplified hull geometry that is used in the optimisation set-up. The hull is divided into three large modules that are further broken down into smaller sub-modules. The main modules consist of the aft of length of x_1 , the midship section with length x_7 , and the bow section with length x_8 .

The numbered x-values ranging from x_1 to x_{12} are used throughout the optimisation and represent that specific length (as described in table 3.1.1). It is important to note that while the figure shows that the aft dimension x_3 is smaller than x_2 , this will not necessarily be the case for the optimal solution, but it allows the stern to be more narrow if desirable. Thus, the relationships between the different variable values are not set to a specific value but chosen as an illustrative example.

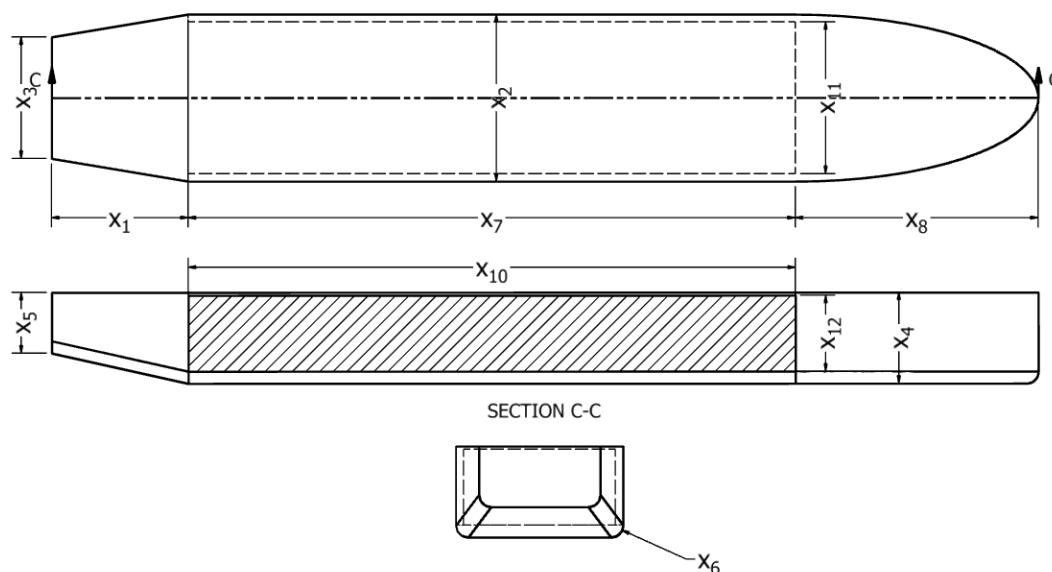


Figure 3.1.1: Simplified FPSO hull geometry for optimisation. The ideal geometry is depicted here as the bilge radius extends all the way to the transom stern. Actual geometry of aft section depicted in figure 3.1.2.

It is desirable to avoid double-curved surfaces on the model for simplicity in the calculations, but there are areas where such surfaces are necessary. The bilge radius (x_6) is assumed to circumvent the entire bottom; thus, an elliptical torus is used at the bottom of the bow. Since the volume and surface calculation of elliptical toruses is a complex numerical calculation, a simpler calculation was used to lower the computation time. Thus, the double-curved surface in the bow is calculated as a circular torus and corrected using the average radius on the ellipse. Since the bilge radius circumvents the entire hull, another double curved corner is found at the forward end of the aft section. A simplification was done here as well to lower the computation time and calculation complexity. The simplification of the aft section is represented in figure 3.1.2.

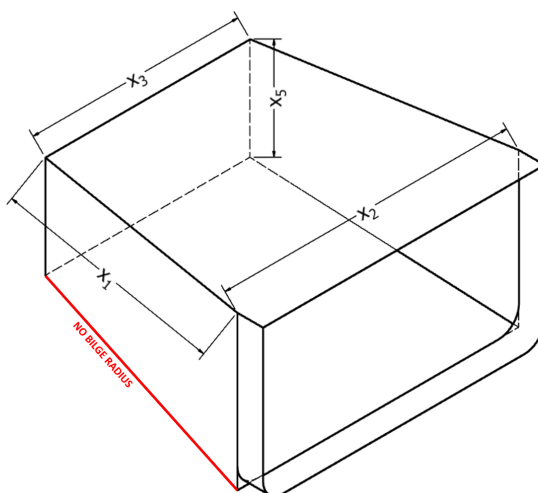


Figure 3.1.2: Simplification of bilge radius in aft section of the simplified hull geometry. Not the change in bilge radius in the corner of the aft-section.

Figure 3.1.2 shows that the bilge radius is not calculated along the edges of the aft section. This simplification will lead to a small increase in wet surface area and but is unlikely to have a significant impact on the optimal solution. Thus, no further calculations are necessary to calculate this volume and area more precisely.

The model also allows the draught of the vessel to exceed the height of the transom stern's edge. When the waterline is above the lower edge of the stern, the accuracy of the waterplane area will only become more accurate, and hence benefiting the optimisation considerably. Since the bow is vertical up to, and above the waterline, there is no change in the waterline area for draughts above the hull's bilge radius (when $T > x_6$).

3.1.2 Upper and lower bounds

The upper and lower bounds of the hull dimensions are represented in table 3.1.1. Large intervals between the upper and lower bounds create large feasible solution areas. Nonetheless, it is also important to keep the dimensions within sensible ranges. Since one of the optimisation objectives is to maximise the deck area, the solver will always opt for the largest possible dimensions in one end of the local Pareto front. Consequently, smaller values are likely to be chosen to minimise the build cost, as long as the constraints are satisfied for both objectives. While the ranges of the bounds are kept as large as possible, it is important to note that the given limits have been selected to develop a slightly similar shape to the FPSO's of Altera for easier comparison.

Table 3.1.1: *Mean draught is calculated based on displacement and waterplane area, but is added to the main dimension vector for results analysis. All dimensions in metres.

Dimension	Description	Lower bound	Upper bound
x_1	Length of aft part	20	45
x_2	Breadth of midship	40	60
x_3	Breadth of transom	30	70
x_4	Moulded depth of midship	15	40
x_5	Moulded depth of aft part	10	25
x_6	Radius of bottom corner	0.5	4
x_7	Length of midship part	100	200
x_8	Length of bow part	40	80
x_9	Mean draught*	10	40
x_{10}	Total length of cargo hold	100	180
x_{11}	Total breadth of cargo hold	20	60
x_{12}	Depth of cargo hold	20	40

While the upper and lower bounds create a feasible area, the constraints of the optimisation problem will further limit the feasible solution space. It is important to note that some of the bounds exceed that of the corresponding constraints. For instance, the upper limit of the vessel's cargo tank length is set to 180 metres, while the constraint limits the total length to $x_{10} \leq x_7$ regardless of the value of x_7 . The limitation is done purposely, rendering a broader solution space when considering the non-linear constraints. Hence, the bounds may either be limited by the bounds or by the linear and non-linear constraints. Additionally, there is some overlap in the bound limits of the variables, especially those concerning the cargo tank dimensions. The breadth of the cargo tank cannot exceed the breadth of the vessel, and thus the depth and length must be altered to meet the cargo volume requirements.

3.1.3 Constraints

While the upper and lower bounds automatically generate one set of constraints for the MOOP, an additional set of linear and non-linear constraints are necessary to ensure the algorithm generates dimensions for the vessel that are sensible. The constraints given in equations (3.1.1) through (3.1.11) have been used throughout the optimisation process for all algorithms. While fewer constraints ensure more freedom in the optimisation and may generate more innovative solutions, a more extensive set will generate more similar ships to the current designs and thus make the results more comparable.

$$\min_{x \in \mathbf{x}} f_1(x_i), f_2(x_i) \quad (3.1.1)$$

$$\text{subject to } x_3, x_4 \leq x_2 \quad (3.1.2)$$

$$x_9 \leq x_4 \quad (3.1.3)$$

$$x_1, x_2, x_9, x_{10} \leq x_7 \quad (3.1.4)$$

$$x_1 \leq x_8 \quad (3.1.5)$$

$$x_{12} \leq x_{11} \quad (3.1.6)$$

$$x_2 \leq 55 \quad (3.1.7)$$

$$x_{11} + \min\left(2, \frac{x_2}{15}\right) \leq x_2 \quad (3.1.8)$$

$$x_{12} + \max\left[0.76, \min\left(2, \frac{x_2}{20}\right)\right] \leq x_4 \quad (3.1.9)$$

$$159,987 \leq x_{10} \times x_{11} \times x_{12} \leq 160,000 \quad (3.1.10)$$

$$x_i \geq 0, i = 1, \dots, 12 \quad (3.1.11)$$

All the dimensions of the vessel are non-negative. Constraints based on more complex calculations such as $GM > 0.5$ and $MSDV < 15$ are implemented in the cost function. If these constraints are broken, the solution is made undesirable by setting the build cost to infinity.

The goal is to minimise the objective functions $f_1(x)$ and $f_2(x)$ that represent the maximum deck area and the building cost of the vessel, respectively. A more detailed outline of the objective functions are given in section 3.2.1. The first is linear constraints that ensure the right relationship between the variables' values. Considerations to length-breadth ratios have not been considered; however, the midship section should be the longest, followed by the bow section and aft section. Furthermore, it is assumed that a vessel does not have adequate intact stability if $B/T \leq 2.5$. However, despite the constraints forcing $B > T$, not B/T ratio is required. Resultingly the search area is kept as large as possible. Additional external considerations come into play, such as the maximum breadth. By limiting the maximum moulded breadth of the vessel to 55 metres, most large yards worldwide have the possibility of constructing such a vessel. However, should the breadth exceed 55 or even 60 metres, only a handful of massive yards have this possibility.

Additionally, the constraints ensure that the cargo tanks are correctly sized for the double bottom and double hull sides, illustrated in equations (3.1.8) and (3.1.9). According to the DNV GL regulations for ships, a maximum of 2 metres, or $B/20$ (with an absolute minimum of 760 mm) is required as a double bottom [18]. $B/15$ is used for the width to ensure space for ballast tanks on the outside of the cargo tanks [49]. The non-linear constraint consists of ensuring enough cargo space is generated. A barrel unit consists of 159.987 litres, and thus the required cargo of 1 million bbls is equal to $159,987 \text{ m}^3$. The required cargo represents the minimum possible amount, and an upper limit of $160,000 \text{ m}^3$ is used as shown in equation (3.1.10).

3.2 Function and algorithm-related optimisation setup

While section 3.1 considered the necessary setup concerning the hull geometry and geometrical considerations and constraints, the following part considers the input parameters and setup of the optimisation algorithm itself. The MOOP setup has been adapted to the possible alterations that are available in the programming software of MATLAB 2019a.

3.2.1 Objective functions

In direct response to the objective of the thesis work, it is desirable to create a large deck area for the vessel, while also considering the build cost of the vessel hull. As mentioned in the scope limitation, the build cost evaluated in the objective function only considers the hull without topside processing plant, nor superstructures on the tank top deck.

Equation (3.2.1) shows the general expression for the optimisation problem where $f_1(x_i)$ represents the deck area function A and $f_2(x_i)$ is the build cost of the vessel's hull, denoted C . The default MOGA setup when using MATLAB 2019a is that of a min-min problem, thus minimising both functions. However, the current problem is a max-min problem, hence solved by minimising the negative value of the deck area, i.e. $-\min_{x \in \mathbf{x}} A$.

$$\min_{x \in \mathbf{x}} f_1(x_i), f_2(x_i) = -\min_{x \in \mathbf{x}} A(x_i), \min_{x \in \mathbf{x}} C(x_i)k_j \quad (3.2.1)$$

By minimising the negative value of the deck area, the algorithm strives to come up with large negative values in the solution and thus maximising the absolute value of the deck area. The method allows all values of $x_i \geq 0$ and does not affect the optimisation in terms of generating an optimal solution. The price breakdown for each cost element is denoted by k_j .

Maximisation of deck area

The maximisation of the deck area is merely a geometric evaluation of the surface on the tank top deck. The function value consists of three parts representing the different modules of the deck; aft, midship and bow deck. The bow deck will most likely be used for the accommodation superstructure and helicopter deck. If the bow area is not included in $A(x_i)$, the algorithm is not likely to enhance the bow's geometry. In such a case, any changes to the bow would not improve the objective function value, and the bow's parameters would likely experience only small alterations from its initial population value. Consequently, the bow area is therefore included in the objective function, despite not altering the effective deck area available for a processing plant.

$$\min_{x \in \mathbf{x}} A(x_i) = - \left(\underbrace{\left(x_1 \frac{x_2 + x_3}{2} \right)}_{\text{Aft deck}} + \underbrace{(x_2 x_7)}_{\text{Midship deck}} + \underbrace{\left(\pi \frac{x_2 x_8}{2} \right)}_{\text{Bow deck}} \right) \quad (3.2.2)$$

$$= - \left(x_2 \left(\frac{1}{2} x_1 + x_7 + \frac{1}{2} \pi x_8 \right) + \frac{1}{2} x_1 x_3 \right) \quad (3.2.3)$$

It is important to note that while maximisation of the deck area only considers the variables x_1, x_2, x_3, x_7, x_8 , all variables x_i are altered simultaneously as the objective function $C(x_i)$ does consider all x_i .

Minimisation of build cost

The minimisation of the build cost is comprised of three main elements; the surface preparation costs (Sp), the paint costs (Pc) and the cost of the steel used in the hull (Sc). Each of these three main elements can be broken down further into smaller subcategories. Different prices k_j applies to different sections of the vessel, such as different paint cost for hull and anti-fouling as well as steel cost and steel-work on the exterior versus inside tanks. The cost function is given in equation (3.2.4).

$$\min_{x \in \mathbf{x}} C(x_i) = Sp(x_i)k_j + Pc(x_i)k_j + Sc(x_i)k_j \quad (3.2.4)$$

The two objective function only focus on the deck area and the build cost. Ideally, the vessel motions should be a third, independent objective function. However, if a response amplitude operator is calculated for a wide range of wave periods for each vessel in the population matrix, the computation time would be astronomic. Initial estimations shows a computation time of approximately nine seconds per vessel. With a population size of 100, the computation time for each iteration would be close to 15 minutes. Consequently, with just 100 population iterations, an optimisation run would take approximately 25 hours. Hence, the vessel motions is only calculated if the vessel is too stiff in terms of stability.

3.2.2 Algorithm input parameters

The genetic algorithm that is used in MATLAB 2019a uses a controlled elitist GA that that is a variant of the Non-dominated Sorting Genetic Algorithm (NSGA-II). The algorithm determines the fitness of each individual by ranking the fitness of non-dominated solutions. To maintain the diversity in the population, the algorithm favours individuals that are relatively far away from the front to be re-iterated. The NSGA-II is slightly more efficient than the original NSGA, as the latter algorithm's diversity mechanism is based on fitness sharing by niching. Niching refers to the formation of groups of individuals in a population, where individuals within a group are similar to each other, while individuals from different groups are very different from each other. The loss in efficiency often lies within choosing the right niching size parameter [26]. Consequently, NSGA-II algorithm will be used, adopting a controlled elitist GA with a set of specific input parameters. The tournament, crossover and selection parameters presented earlier will be used in addition to the following diversification aides; initial population variations and Pareto fraction.

Initial population

There are multiple ways of choosing the initial population matrix X_0 . The most common method is a random variable value vector. This option entails a given population size with the variables' values being randomly assigned within their respective upper and lower bounds. In a $M \times N$ -sized initial population matrix, each value can be written as:

$$x_0(i, j) = L(x_i) + \delta_i (U(x_i) - L(x_i)), \quad (3.2.5)$$

where $L(x_i)$ and $U(x_i)$ represent the upper and lower bound for variable $x_0(i, j)$ respectively. δ_i is a random variable between 0 and 1. Consequently, $x_0(i, j)$ will be a random value between the upper and lower bound of the variable x_i . This calculation is done for all N variables. The user will then determine the population size M , either set randomly or to a specific value. For the optimisation, the analysis has been performed to evaluate what M is most prudent in terms of computation time versus solution quality.

Another method of creating the initial population is a quasi-random initial population selection. The initial population is then based on a random selection of chromosomes from an earlier run. The randomness then only lies within choosing random chromosomes.

This method enables the algorithm to start its search process from a point that is already feasible, and the algorithm will have a feasible starting point, in hopes of further improving the solution. While quasi-random initial population leads to shorter computation time, diversity of the solution is lost early in the process, and the algorithm is even more dependent on diversification in the mutation and crossover stages.

With reference to figure 2.4.2, a similar evaluation has been completed to determine the optimal population size M . Figure 3.2.1 shows the varying population size in regards to its solution quality and computation time. The solution quality is represented by the complement distance from U_P ($1 - \overline{D}(X_j)$), thus the higher the value, the better the solution. Each point on the graph represents a population size that is a multiple of the 12 variables, i.e. $12 \times N$. Here, the graph shows that a population of 240 renders the best solution quality, while 120 is the worst. As expected, a higher population results in longer computation time, and vice versa.

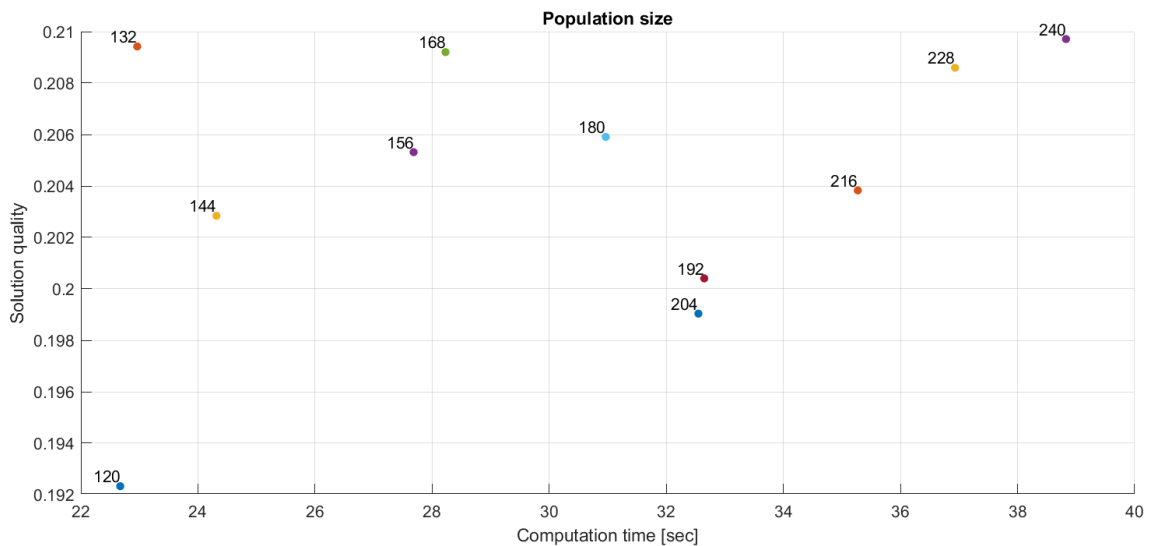


Figure 3.2.1: Varying population size's effect on the solution quality and computation time. Each population is a multiple of the 12 variables. Solution quality is the average of five separate GA runs with a maximum of 20 generations for fast analysis. Longer computation time is expected when there is no limit on the generations.

It is, however, interesting to note that the solution quality does not converge when reaching a population size close to 240. Indeed, population sizes of 132 and 168 have similar solution qualities and were terminated by a fraction of the time of the best solution. Many steps in the GA are based on randomness during selection and mutation. Thus, it may be challenging to achieve convergence in terms of population size. Larger population size is far more likely to cover the search area of global minimum, albeit with a higher likelihood of evaluating more local minimums than a smaller population. Based on the findings in figure 3.2.1, there is no clear choice for the population size in terms of computation time. However, a population of 132 does have the best time versus solution quality, and due to its adequate solution quality, one can therefore safely assume this population size for the remaining optimisation.

Pareto Fraction

An elitist GA will always favour individuals with better fitness, while a controlled elitist GA would favour the individuals who can help increase the diversity of the population. The diversity is enabled through controlling the elite members of the population as the algorithm progresses. The Pareto fraction limits the number of individuals on the Pareto front. If the Pareto fraction, P_F , is 0.35, the algorithm will try to limit the number of individuals in the current population that are on the Pareto front to 35% of the population size. When P_F is smaller, fewer elite solutions are represented in the Pareto front, and a higher degree of diversity is achieved. Conversely, if P_F is high, there is little room for diversity on the front, and the solution may converge too quickly [12].

3.2.3 Hybrid approach setup

Section 2.4.5 outlined the general purpose of using hybrid algorithms that conducted local searches when the rate of convergence of the solution began to diminish. A very similar process is used for the vessel optimisation process where the variation of the NSGA-II starts the process. The GA input parameters such as Pareto fraction and initial value and stall generations are decided beforehand and apply to both the GA and the HGA. For simplicity, the local search is conducted after the stopping criteria have been met. Figure 3.2.2 shows that the local hybrid search can only start when the GA is finished. For the local search to be conducted efficiently, scores from the GA are used to determine where the search should be conducted in the search space.

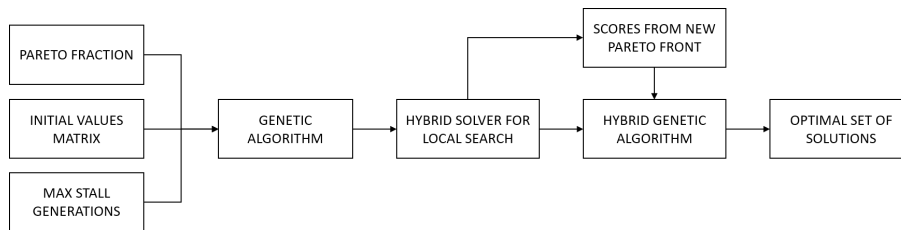


Figure 3.2.2: Process illustration for changes in optimisation algorithms.

When the local search is finished, an optimal set of solutions is likely generated and represent the last output of the flowchart in figure 3.2.2. For comparative purposes, the genetic algorithm without the local search is once again run, only this time, its initial population score is the final score of the local search. While the local search is likely to generate a better solution, a final GA run with the local search scores may diversify the HGA's score and thus find new solutions in the search space the local search could not reach.

3.2.4 Stopping criteria and restrictions

In order for the GA to stop iterating and continuously changing the chromosomes, multiple stopping criteria are used. Such criteria can be determined either by the user, by the algorithm or both. In the optimisation process, the following stopping criteria determines whether the algorithm is terminated:

- **Generations**

The user may define a maximum number of generations that the algorithm may undertake before the optimisation process is automatically terminated. The GA is then terminated regardless of convergence, and the objective function value may be lower than possible if the algorithm reaches the maximum number of generations. Hence, to deter this behaviour, there is no limit, and maximum generations are set to infinity.

- **Maximum stall generations (MSG)**

The algorithm stops if the average relative change in the best fitness function value over stall generations is less than or equal to the function tolerance, σ_F . In more general terms, the algorithm terminates if the fitness function value does not change more than the function tolerance after a given maximum of generations. For the optimisation, the maximum stall generations (MSG) was varied to see the effect on the algorithm's objective function effect. A high MSG-value will result in longer computation time as the almost identical solutions will be regenerated many times, while a small value may result in premature termination.

- **Function tolerance σ_F**

The GA will terminate if the average relative change in the best fitness function value is less than, or equal to the function tolerance. For the optimisation, the function tolerance is set to 10^{-3} , meaning that if the change in the fitness function from one generation to another is less than the tolerance value, the algorithm stops. Termination based on function tolerance is given that the algorithm has not exceeded the number of possible stall generations.

- **Constraint tolerance σ_c**

The constraint tolerance is included as a user input parameter but is not necessarily used as a stopping criterion. The constraint tolerance is used to determine the feasibility concerning the non-linear constraints. In the algorithm's attempt to satisfy its constraints, the GA will try to make the inequality as close to zero as possible, and hence the constraint tolerance decides as to how close is acceptable before the optimisation is terminated. Throughout the optimisation, the constraint tolerance is set to 10^{-4} .

In addition to the before-mentioned stopping criteria, one can also implement restrictions that terminate the current generation and forces the algorithm to start a new iteration or considerably alter the population. Figure 3.2.3 shows the flow of the optimisation algorithm that applies to both the GA, HGA and the local hybrid search. The optimisation process starts by identifying the current setup that includes the user input parameters, stopping criteria and initial population. The algorithm of choice is then chosen before it calculates the fitness, i.e. the deck area and cost of the vessel.

Before another generation is completed, a parameter check is conducted, evaluating both stability and the MSDV value of the vessel. If the vessel does not meet the stability requirement of $GM > 0.5$, the seasickness requirement of $MSDV < 15$, the cost function's value is set to infinity, thus ultimately deterring the algorithm from evaluating this solution further.

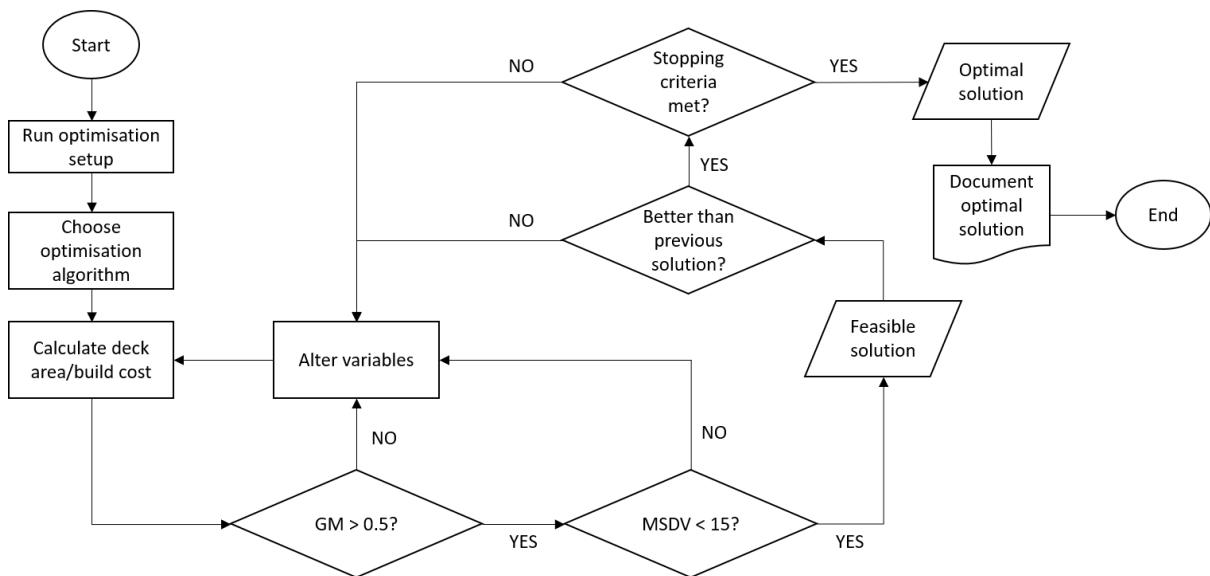


Figure 3.2.3: Flow chart and stopping criteria for optimisation algorithm, both GA and HGA.

If both criteria are met, a feasible solution exists, and its fitness is compared to the earlier fitness values. If the current solution is not better than the previous solution, the algorithm completes another iteration with crossover, selection and mutation before again evaluating the possible solution. This process continues until the stopping criteria are met, and the optimal solution with non-dominating points is documented as the optimal Pareto front.

Results

The results presented in the following chapter are results of the optimisation with both the genetic algorithm (NSGA-II) and the local search hybrid. Meta-data such as stability and righting levers are obtained from DelftShip hull modelling software. The geometry input to DelftShip is a direct result of the optimisation.

4.1 Optimisation output

Figure 4.1.1 shows the results of the optimisation conducted, using both the GA, local search and hybrid GA (GA with local search scores). The final set of result consists of several runs with different Pareto fractions, MSG's and different initial populations. Each result generated an individual Pareto front, as illustrated in figure 4.1.1. Considering the spread in the Pareto fronts on the graph, and the significant variations in the distance to the utopian point, it is clear that both the input parameters and algorithm variations have a substantial effect on the objective function values.

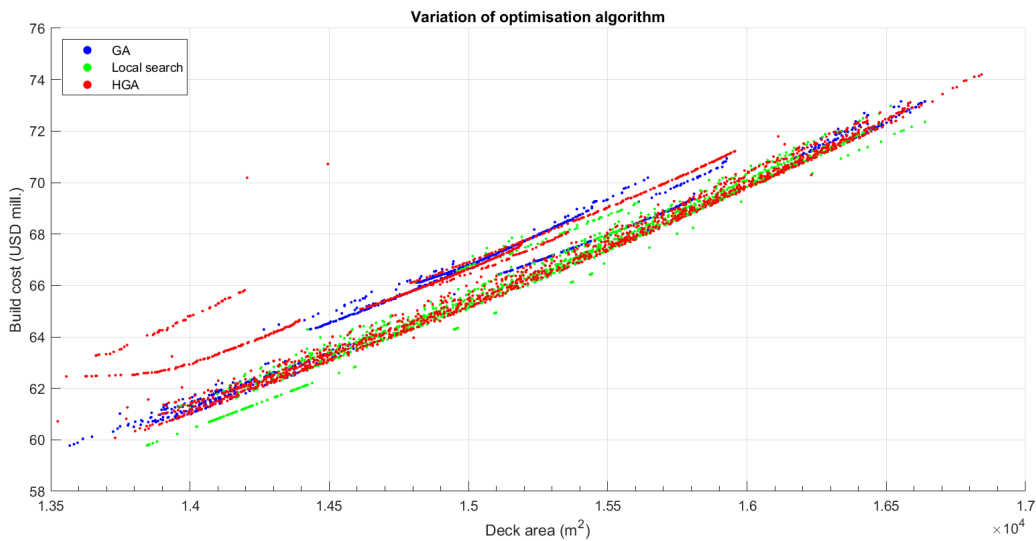


Figure 4.1.1: Result of GA, local search and HGA with multiple Pareto fronts.

Based on the 6,255 individual non-dominated solutions in figure 4.1.1, the following data in table 4.1.1 represent the maximum and minimum values of the two objective functions. The largest difference between the maximum and minimum deck area is about 21.9%, which is also the same difference in the maximum and minimum build cost. Furthermore, one can observe that the largest deck area is a result of the genetic algorithm that uses the scores from the antecedent local search. As expected, the corresponding build cost when the deck area is 16,841 m² is also the highest at US\$74.2 million. Conversely, the cheapest vessel only costs US\$59.78 million with a deck area of 13,525 m².

Table 4.1.1: Objective function values and utility prices for the genetic algorithm, local search and hybrid genetic algorithm. Build cost values given in million USD.

KPI	GA	Local search	HGA
Max. deck area	16,339 m ²	16,639 m ²	16,842 m ²
Mean deck area	15,214 m ²	15,125 m ²	15,128 m ²
Min. deck area	13,568 m ²	13,845 m ²	13,525 m ²
Max. build cost	\$73.16	\$72.98	\$74.20
Mean build cost	\$66.81	\$65.69	\$66.59
Min. build cost	\$59.78	\$59.78	\$60.07
Max. ratio	\$4,507/m ²	\$4,477/m ²	\$4,941/m ²
Mean ratio	\$4,392/m ²	\$4,343/m ²	\$4,403/m ²
Min. ratio	\$4,338/m ²	\$4,302/m ²	\$4,321/m ²

The ratios in table 4.1.1 represent the utility cost, i.e. the dollar cost of each square meter of deck area (US\$/m²). The lowest ratio is at \$4,302/m² generated by the local search algorithm. This ratio also corresponds to the vessel with the highest deck area and highest build cost.

4.1.1 Choice of vessels for further study

The study has generated over 6,000 vessels that can be chosen. As individual runs, each point on the Pareto front are non-dominated solutions; hence any point on the front is optimal. However, when gathering multiple fronts in one figure, one can observe that some runs have resulted in better objective values than others, generating fronts closer to U_P and thus rendering some solutions dominant. The utopian point for this optimisation is the point where the maximum deck area and minimum build cost intersect. Hence, an objective measure, regardless of the runs, is the distance to the utopian point. Since the main objective is to maximise the deck area, while minimising the build cost, one might consider these two vessels as the obvious candidates for further investigation. However, a more prudent performance index is the utility price from the vessel, and thus the vessel with the lowest utility cost is chosen as one of the vessels. The vessel with the lowest build cost might be interesting but has the highest price per square meter of all vessels. Hence, an evaluation of the vessel that performs the best numerically is of interest.

Accordingly, the distance $D(X_j)$ to the Utopian point U_p can be calculated using Pythagoras theorem for each of the points in figure 4.1.1:

$$D(X_j) = \sqrt{(\max\{\bar{A}(x_i)\} - \bar{A}(x_i))^2 - (\bar{C}(x_i) - \min\{\bar{C}(x_i)\})^2}, \quad (4.1.1)$$

where $\bar{A}(x_i)$ and $\bar{C}(x_i)$ are the normalised values (between 0 and 1) of $A(x_i)$ and $C(x_i)$ respectively. Based on the calculated in equation (4.1.1), one can determine the vessel point that is the closest to U_p . As the values are normalised, all distances are between 0 and 1, with $D(X_j) = 1$ being the farthest away, and vice versa. Figure 4.1.2 shows the extent of $D(X_j)$ for all the vessels, where the lowest value, and thus closest to U_p is 0.625. As one might expect, the vessel with the highest deck area (and hence also the highest cost), is farthest away from U_p with $D(X_j)$ equal to 1.

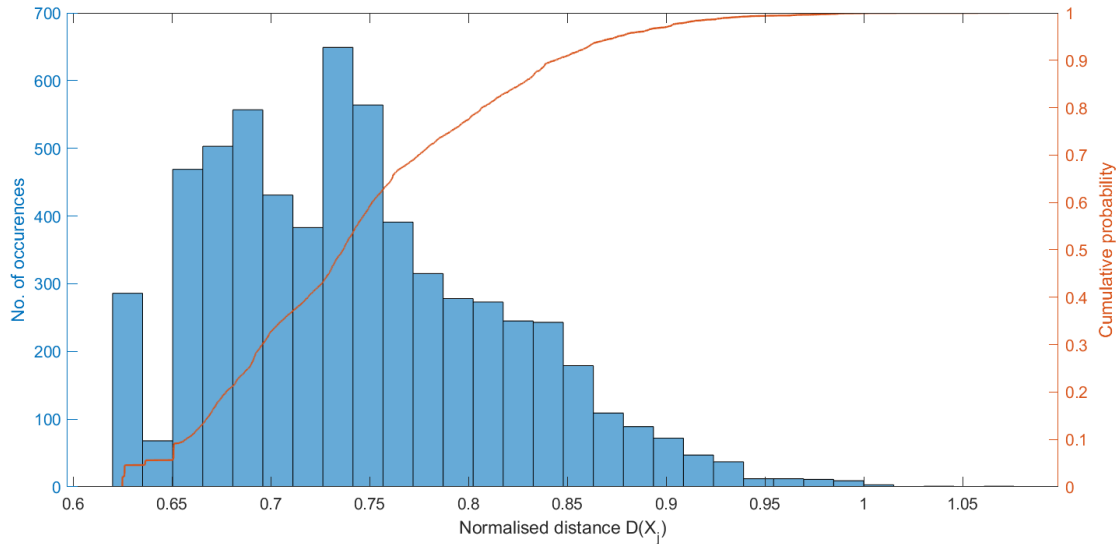


Figure 4.1.2: Histogram of normalised distance, $D(X_j)$ to the Utopian point U_p . Lowest $D(X_j)$ is 0.625.

Figure 4.1.2 also shows the cumulative probability of the normalised distance, with over 60% of the points being below 0.75. The smallest distance 0.625 is the closest of all to U_p which is the case for vessel no. 4843 and will thus be evaluated further in the following sections. The vessel with the lowest utility cost is vessel no. 1864.

Figure 4.1.3 represents the normalised values of the total deck area and the corresponding build cost. Vessel no. 1863 and 4843 are marked with red and green circles, respectively. From the figure, one can see that vessel 4843 is closest to the Utopian reference point marked $+$ in the lower right corner. The distance between the green circle and Utopian point is, therefore, 0.625.

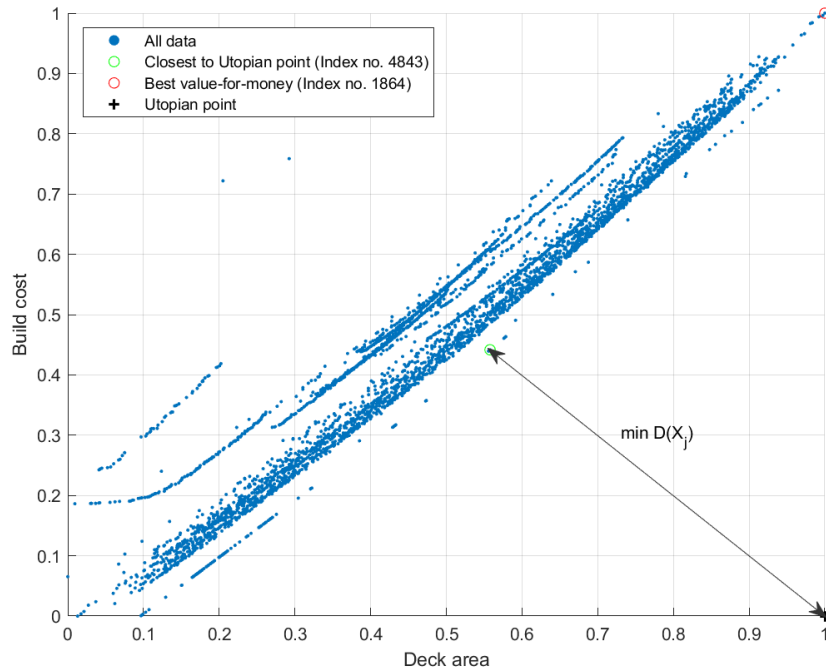


Figure 4.1.3: Normalised deck area and build cost with chosen vessels highlighted. Utopian reference point in the lower right corner.

Based on the observed points that lie outside the central cluster in both figure 4.1.1, the algorithm preferences and choice of the algorithm have a substantial effect on the results. The distances $D(X_j)$ vary from 0.625 to 1, and there is a 13.9% difference between the maximum and minimum utility cost. The input parameters of the optimisation have a significant effect on the output and must be thoroughly evaluated to determine the settings that render the best results.

4.1.2 Variation of algorithm preferences

The 6,255 data points generated in the optimisation have varying input parameters and stopping criteria, while the function tolerance and constraint tolerance is kept constant. Additionally, there is no maximum number of generations. Changes in the parameters were limited to the Pareto fraction, P_F , the maximum stall generations (MSG), and the method of random selection for the initial population.

Figure 4.1.4 shows the effect of the Pareto fraction on the results. As previously mentioned, a higher P_F allows more elitist solutions to gather on the Pareto front, while a smaller value will revisit solutions with worse fitness and iterate them into non-dominating solutions. With the latter P_F , a higher degree of diversification is achieved. However, all the points in the graph are all non-dominated solutions for each of the individual runs. Thus the distance to the Utopian point may not vary much with a variable P_F .

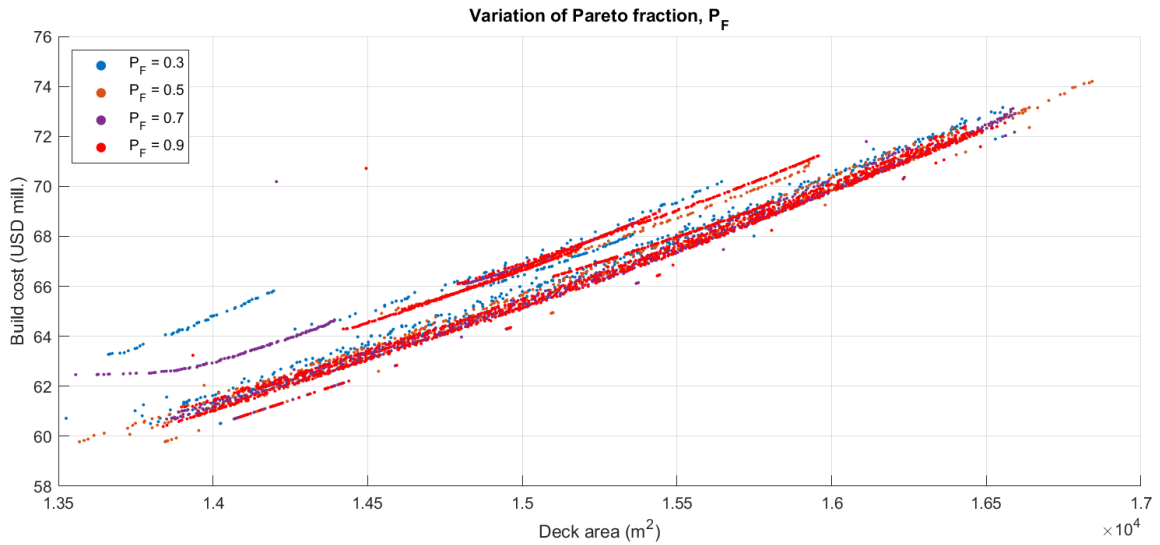


Figure 4.1.4: Optimisation results with varying Pareto fraction values.

Observing the distribution of solutions in figure 4.1.4 and the corresponding P_F -values, there is no clear benefit with either P_F . While it seems that no given Pareto fraction value is more favourable than the other, when evaluating the average $D(X_j)$ for all solutions with a given P_F , $P_F = 0.9$ gives the lowest average distance to the utopian point while $P_F = 0.3$ is on average the farthest away. However, the differences are almost negligible, with $< 5\%$ difference between the average distances $\overline{D}(X_j)$. Therefore the variation in the results based on P_F are not conclusive, and no further focus is given to the evaluation of the P_F -effect. More importantly, one may deduce that the performance of the optimisation is quite independent of elite solutions and that the diversity of the population is adequately generated from other tools in the optimisation.

The variation of the maximum number of stall generations is likely to have a greater effect on the resulting output than varying P_F -values. Optimisation with MSG-values of both 250 and 1,000 may converge at the same point and generate equal quality solutions. A run with $\text{MSG} = 1,000$ will re-iterate a solution that has met the stopping criteria 1,000 times before the algorithm is terminated to verify that no better solutions could be found in that area. A run with MSG-value of only 250 will then stop earlier given the same scenario but have a much shorter computation time. Hence, a higher MSG-value does not guarantee a better solution, or a solution closer to the Utopian point but does give a higher level of confidence of not converging in a local minimum.

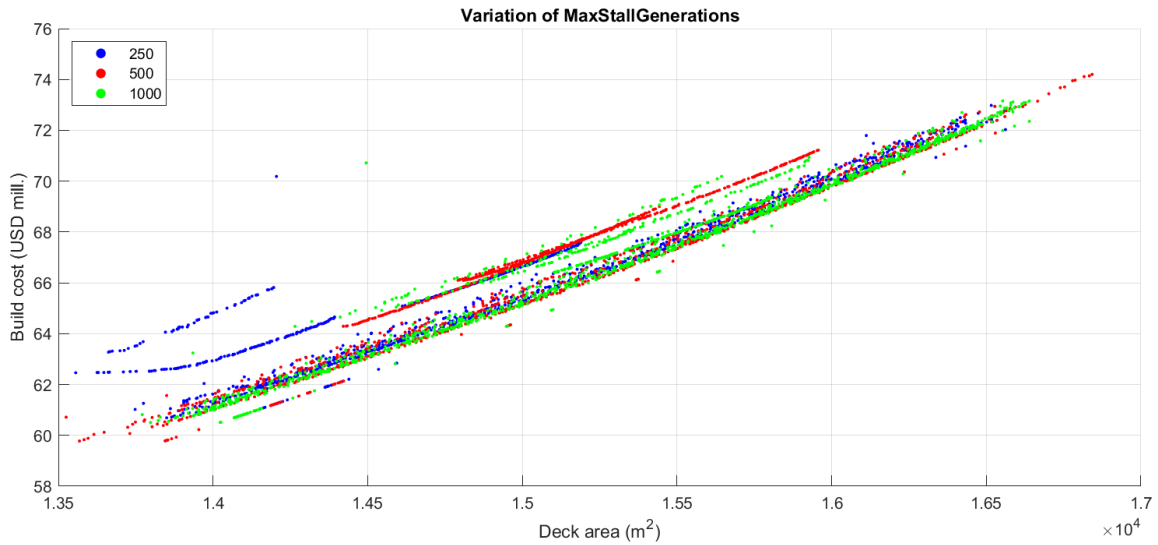


Figure 4.1.5: Optimisation results with varying limit on the maximum stall generations.

Figure 4.1.5 show multiple runs with a varying level of maximum stall generations. The main cluster of solutions have all different MSG-values, and hence there is no clear indication of a favourable MSG-value in terms of the mean $D(X_j)$ for each MSG-value. However, when starting the optimisation with completely random values in the initial population, a higher MSG-value is often required to generate solutions on the Pareto front. The random numbers will often generate an unfeasible solution, and hence the objective functions will become infinitely high. A greater MSG-value allows the algorithm to work through the initial stall before generating viable non-dominating solutions. This problem is indeed visible in figure 4.1.5. The set of solutions that are far away from the main cluster to the left all have the lowest limit of stall generations. These solutions had a completely random X_0 and were terminated prematurely since the algorithm started stalling early in the process.

In setting up the initial population X_0 , it has been discussed whether to use a completely random X_0 or the quasi-random method of choosing random solutions from a previous run. The latter allows the optimisation to iterate the solutions more quickly, as it will not stall at the beginning of the run and always only experience feasible solutions. However, one is then highly dependent on other methods for population diversification. Consequently, the effect on the initial population selection method has also been tested to see the effect on the Pareto fronts.

Figure 4.1.6 shows solutions sorted by initial matrix selection. The iterated (green) points have a $X_0 = X_{j-1}$ while the random (red) solutions are completely random from the onset of the optimisation run. The difference between the two methods has a clear effect on the results, with the iterated results being, on average much closer to the Utopian point. This difference is expected, as the solution is already closer when beginning the optimisation and local search.

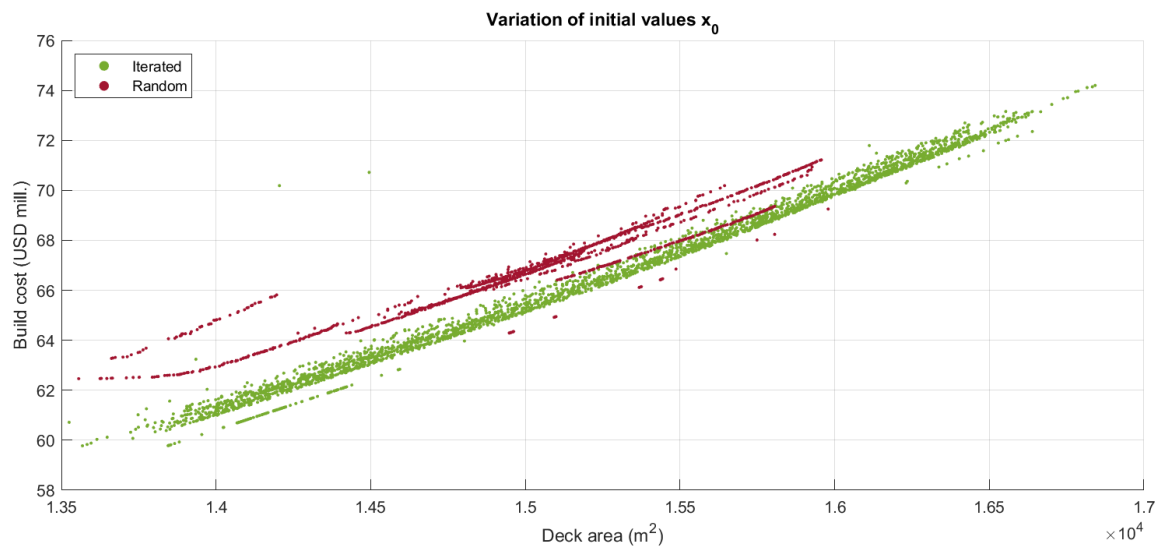


Figure 4.1.6: Optimisation resulting with a variation in choice of initial population matrix X_0 .

However, the loss of diversity is evident as the points outside the main cluster, but closest to the Utopian point, are those with a completely random X_0 . While these points are closest to the U_P , they do suffer from an uneven spread, with only a few points located to the right and below the main cluster. The solutions points in figure 4.1.6 also shows that while generating a few excellent solutions, random X_0 -solutions are much more inconsistent in terms of quality than with the pre-iterated X_0 . Despite a few excellent solutions with random X_0 , one could consider only using the iterated X_0 for better consistency and predictability in the optimisation.

4.2 Optimal hull geometry

In section 4.1.1, vessel no. 1864 and 4843 were chosen. The first vessel was chosen as it has the lowest utility cost, while the latter was chosen because it is the vessel closest to the U_P . The following section outlines the main characteristics and the key performance indexes (KPI's) of these vessels. The loading conditions (weight and centre of gravity) are calculated continuously in the optimisation process. These numbers are scaled with reference to one of Teekay's FPSO's. Hydrostatic data is calculated by the MATLAB model as well, while the stability data is an output from the hull modelling software DelftShip.

4.2.1 Numerically optimal vessel

Vessel no. 4843 is the vessel that is closest to the Utopian point in the optimisation ($D(X_j) = 0.625$). The vessel is designed to carry a maximum of 1,000,000 crude oil barrels before offloaded. The vessel has a build cost of US\$66.15 million and a deck area of 15,375 m².

Main dimensions

The main dimensions are found in table 4.2.1. Following the practice of standard FPSO technical documents, the aft perpendicular is measured at the transom stern of the vessel. Thus the length between perpendiculars L_{PP} is equal to the waterline length L_{WL} .

Table 4.2.1: Main particulars and hydrostatic coefficients for the vessel that is closest to the utopian point (vessel no. 4843).

Description	Symbol	Value
Length between perpendiculars	L_{PP}	289.11 m
Length waterline	L_{WL}	289.11 m
Length overall	L_{OA}	293.58 m
Breadth waterline	B_{WL}	55.00 m
Draught	T	14.84 m
Moulded depth	D	22.00 m
Wet surface area	S	19,684 m ²
Lon. centre of buoyancy from A_P	LCB	150.76 m
Block coefficient	C_B	0.899 -
Prismatic coefficient	C_P	0.899 -
Midship coefficient	C_M	0.999 -
Waterplane coefficient	C_W	0.967 -

The output from the algorithm is used to model the hull geometry in DelftShip. When using 12 variables in the model (instead of only main dimensions), it allows the DelftShip hull geometry to have almost identical hydrostatic values as the MATLAB output. In most cases, there is under 1% difference in the calculation result. Figure 4.2.1 shows the final geometry of the numerically optimal vessel.

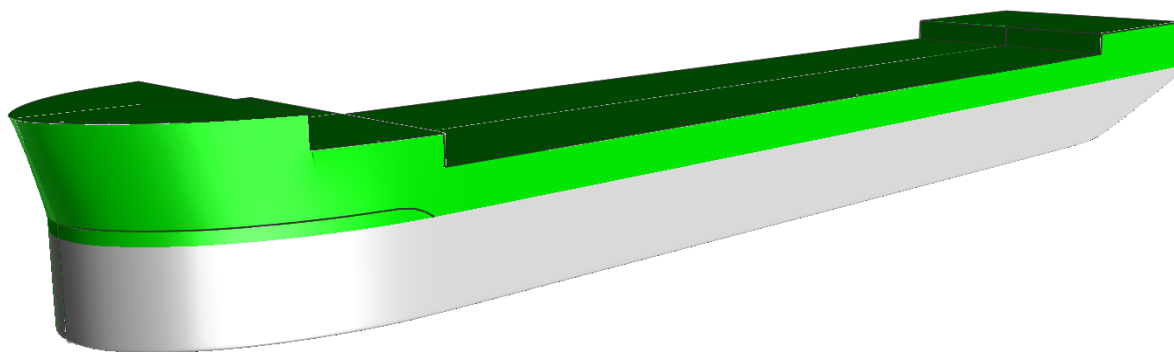


Figure 4.2.1: Visual representation from DelftShip software of vessel no. 4843.

Figure 4.2.2 shows the sectional body plan of vessel no. 4843. As one can see from both this figure and figure 4.2.1, the bilge radius is fairly sharp. The sharp radius is a direct result from the optimisation, where the bilge radius x_6 was iterated to be only 0.5 metres. It is important to note that the additional deck area gained from the bow extension, implemented for generating a wave flare angle, is not considered in the calculated deck area.

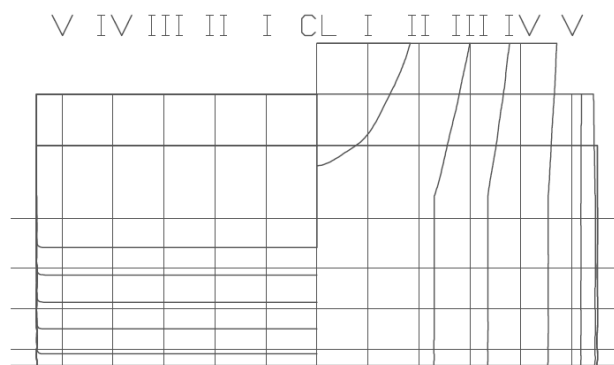


Figure 4.2.2: Body plan of the vessel's sections. Five buttocks are used and highest waterline is the design draught.

Figure 4.2.2 also shows that the underwater hull is vertical in the waterline, and hence the waterplane calculation is accurate with little difference in the MATLAB calculations and DelftShip. There is some difference in wet surface area as the optimisation model over-estimates this measure. It is likely caused by the constant bilge radius that follows the vessel from the bow to the stern. While the MATLAB model does account for less wet surface around the bow's bilge and the centre module, no bilge radius is accounted for abaft. Consequently, DelftShip's estimation is approximately 10% lower.

Loading conditions

Similar to the load cases presented in the technical specifications from Altera, three loading conditions are evaluated for each vessel. The lightweight distribution and centre of gravity stay constant throughout all loading conditions, represented in table 4.2.2.

Table 4.2.2: Lightship weights and centres of gravity for vessel no. 4843.

Description	Weight (MT)	LCG (m)	VCG (m)	TCG (m)
Hull construction	38,714.29	151.533	16.009	-0.150
Topside	19,698.91	134.091	31.268	1.350
Turret	3,838.00	211.456	26.701	0.034
Lightweight	62,251.20	149.708	21.497	0.336

The following conditions apply to the loading conditions LC_0 , LC_1 and LC_2 found in table 4.2.3:

- **LC_0 :**
Ballast loading conditions with almost full ballast tanks. Crude oil cargo tanks completely empty, i.e. no free surface moment (FSM). 100% consumables.
- **LC_1 :**
Laden condition. Almost fully loaded cargo tank with almost empty ballast tanks. Considerable FSM only generated in water ballast tanks. 100% consumables.
- **LC_2 :**
Laden conditions. 50% loaded cargo tanks with FSM from both water ballast tanks and crude oil cargo tanks. 100% consumables.

These loading conditions are used in the stability calculations as well as the seakeeping evaluation in section 5.3. Since loading condition LC_1 has the lowest GM-value, this condition was used for calculating the vessel's response in regular waves.

The centre of gravity for each weight group is scaled according to the current locations onboard Altera's vessels. The longitudinal centre of gravity (LCG), vertical centre of gravity (VCG) and transverse centre of gravity (TCG) apply only for the deadweight in each loading condition's summary.

Table 4.2.3: Deadweight and centre of gravity for all loading conditions for vessel no. 4843.

Description	Weight (MT)	LCG (m)	VCG (m)	TCG (m)
LC₀				
Consumables and liquids	9,706.28	121.531	10.305	-0.700
Water ballast	56,829.12	155.842	7.054	0.000
Crude oil cargo	0.00	0.000	0.000	0.000
Deadweight	66,535.40	150.837	7.528	-0.102
LC₁				
Consumables and liquids	14,171.14	100.200	11.487	-0.480
Water ballast	2,654.60	219.417	2.361	0.000
Crude oil cargo	138,318.69	144.555	11.700	0.000
Deadweight	155,144.43	141.784	11.521	-0.044
LC₂				
Consumables and liquids	14,171.14	100.200	11.487	-0.480
Water ballast	22,614.47	203.685	9.382	0.000
Crude oil cargo	69,158.76	144.555	7.102	0.000
Deadweight	105,944.37	151.244	8.175	-0.064

Stability and righting lever

The stability check and calculation of righting lever are performed in DelftShip according to the International Code on Intact Stability, IMO [40]. Table 4.2.4 shows the resulting stability data. Note that the negative trim represents bow up.

Table 4.2.4: Stability data for vessel 4843. Data extracted from loading analysis in DelftShip.

Description	LC ₀	LC ₁	LC ₂
T _A	9.860	16.131	11.839
\bar{T}	9.103	16.000	11.651
T _F	8.345	13.408	11.463
Trim	-1.515	-2.723	-0.376
KM	33.676	25.349	28.849
VCG	14.280	14.348	13.106
GM	19.396	11.001	15.743
GG'	1.098	1.557	1.917
G'M liquid	18.298	9.444	13.826

Figure 4.2.3 shows the resulting stability graph for vessel no. 4843, for each of the three loading conditions. Some of the most important criteria include that the maximum GZ-value must be at 30° or greater and that the area ratio $A/B \leq 1.0$. The latter criterion implies that the righting lever momentum is greater than the rollback angle righting momentum. Thus, the vessel will have intact stability to straighten the hull if it experiences an external force that results in a rolling motion.

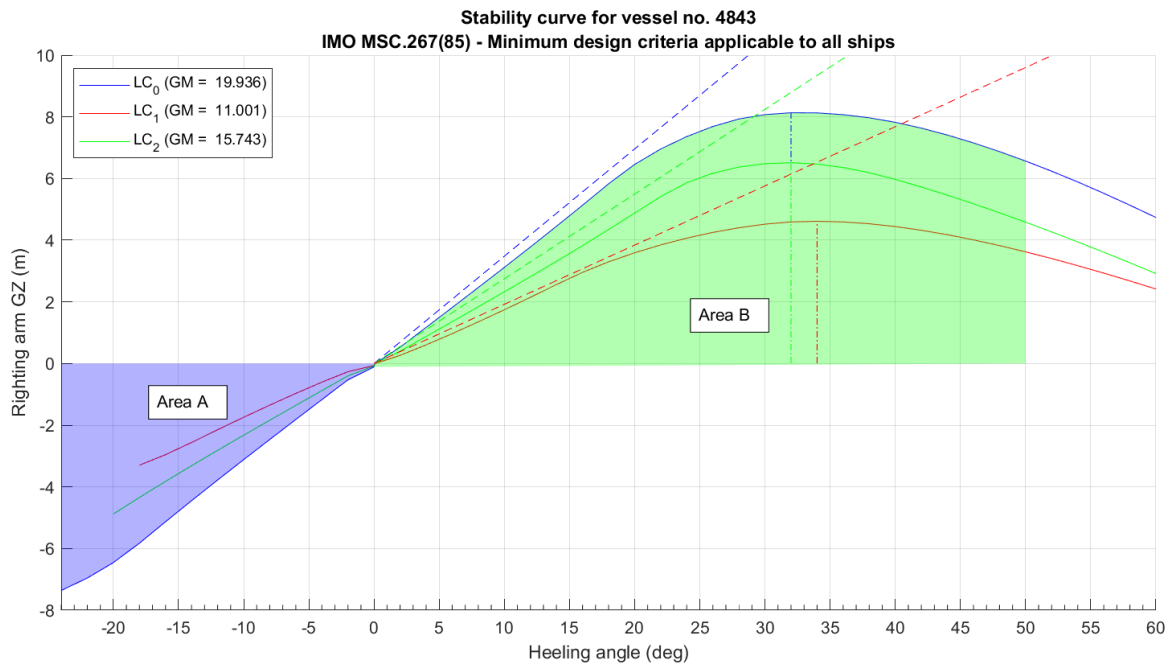


Figure 4.2.3: GZ curves for various loading conditions for vessel no. 4843, including intact stability, GM.

Table 4.2.5 summarises the criteria information from the GZ curve. Area A and area B marked in figure 4.2.3 corresponds to the areas that relate to the loading condition LC₀. The areas on the graph are representative of this loading condition. Hence, the area sizes may differ between the various loading conditions, but all loading conditions meet the intact stability criteria. Further evaluation of the vessel's stability and stiffness is commented on in chapter 5.

Table 4.2.5: Evaluation of intact stability criteria according to IMO MSC.267(85) for vessel no. 4843.

Loading conditions	LC ₀	LC ₁	LC ₂
Rollback angle	-24.6°	-18.1°	-20.7°
Area A	1.832	0.569	0.991
Area B	4.921	2.770	3.780
Max. GZ	8.137 m	4.614 m	6.513 m
Angle of max. GZ	32.8°	33.9°	31.8°

4.2.2 Market optimal vessel

Vessel no. 1864 has the lowest USD/m²-value, rendering it the most cost-efficient in terms of the objective functions. With a deck area of 16,842 m² and a building cost of \$74.20 million USD, the vessel provides the best *value-for-money*.

Main dimensions

The main dimensions are found in table 4.2.6. Similar to vessel 4843, the aft perpendicular is also measured at the start of the transom stern abaft. The differences between the MATLAB and DelftShip outputs are almost negligible (under 1%) for almost all of the dimensions. However, like the calculation for the previous vessel, there is some difference in calculating the wet surface area. For vessel no. 1864, the MATLAB model over-estimated the wet surface area by approximately 5% which is likely due to the loss of accuracy abaft where no bilge radius is present. However, the deviation between DelftShip and MATLAB is as much as 19% different for vessel no. 4843. One might expect that with a smaller bilge radius, the difference would be smaller than vessel 1864. However, the wet surface area of the aft section makes up 15.6% of the total area for vessel 4843. In comparison, the same area represents only 10.4% for the larger vessel no. 1864. The wet surface area does affect the anti-fouling pain cost in the objective function, and the deviation will thus have an effect on the total paint cost. All the other hydrostatic coefficients are, however, less than 1% different from the DelftShip output and hence the model is adequately accurate for optimisation purposes.

Table 4.2.6: Main particulars and hydrostatic coefficients for the vessel with the lowest utility cost.

Description	Symbol	Value
Length between perpendiculars	L_{PP}	323.28 m
Length waterline	L_{WL}	323.28 m
Length overall	L_{OA}	327.67 m
Breadth waterline	B_{WL}	55.00 m
Draught	T	14.13 m
Moulded depth	D	26.54 m
Wet surface area	S	26,554.00 m ²
Lon. centre of buoyancy	LCB	163.67 m
Block coefficient	C_B	0.880
Prismatic coefficient	C_P	0.889
Midship coefficient	C_M	0.990
Waterplane coefficient	C_W	0.974

Figure 4.2.4 shows the final geometry of the vessel from the DelftShip software. Here, the bilge radius is visibly much larger than vessel no. 4843. The hull generation method is identical for both vessels, in terms of input when creating the hull geometry. The setup includes an elliptical bow area and trapezoidal aft. This vessel is approximately 11% longer between the perpendiculars, thus making it equally slimmer than vessel no. 4843 due to their same breadth.

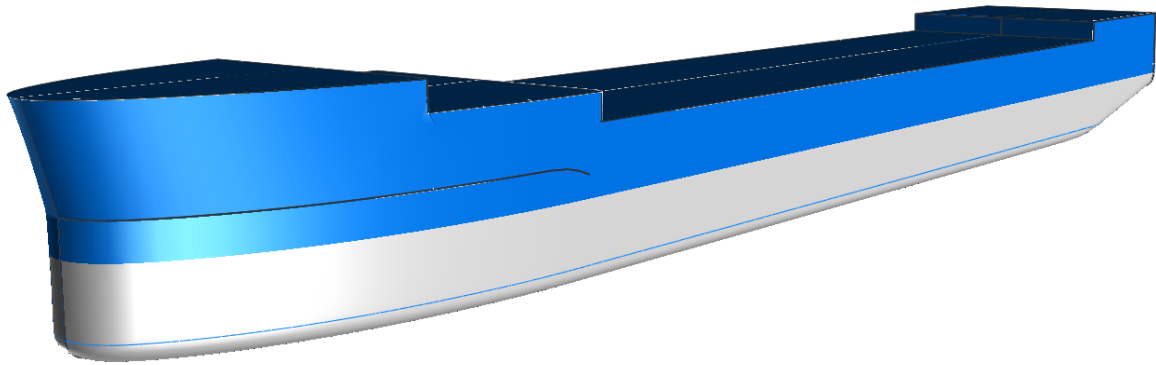


Figure 4.2.4: Visual representation from DelftShip software of vessel no. 1864.

Many of the same characteristics are repeated for both vessels in terms of the bow area. However, the extra length of the bow section of this vessel accounts for much of the added length. The length of the bow section is set by variable x_8 which is almost double for vessel no. 1864. Consequently, a slimmer bow is achieved for this vessel. The sectional body plan is illustrated in figure 4.2.5, showing that while the optimisation allows for a trapezoidal aft section, the algorithm has not done so. Consequently, the breadth of the transom stern is equal to the breadth amidship.

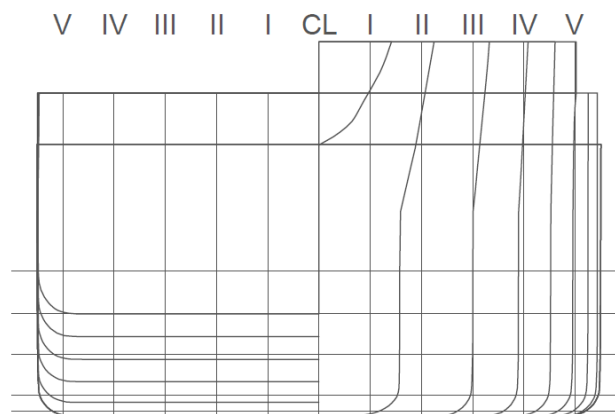


Figure 4.2.5: Body plan of vessel no. 1864's sections. Five buttocks are used and highest waterline is the design draught.

Loading conditions

Similarly to the other vessel, three loading conditions are used in the optimisation. Since LC_1 has the lowest GM-value, the vessel's dimensions have been optimised for this particular loading condition. Furthermore, it is at this load that the vessel has the highest draught and thus, the highest requirement of underwater anti-fouling.

Table 4.2.7: Lightship weights and centres of gravity for vessel no. 1864.

Description	Weight (MT)	LCG (m)	VCG (m)	TCG (m)
Hull construction	43,290.54	169.445	19.311	-0.150
Topside	22,027.43	149.942	37.716	1.350
Turret	4,291.67	236.451	32.207	0.034
Lightweight	69,609.60	167.405	25.930	0.336

Vessel no. 1864 is considerably larger, and thus with a considerably higher lightship weight. This results in a 10.4% difference in lightship weight and 4.2% difference in total displacement in laden condition, LC_1 . Despite the differences in size and displacement, both vessels shall carry the maximum crude oil cargo of 1,000,000 barrels, and thus the difference in deadweight (DWT) is only 1.3%.

Table 4.2.8: Deadweight and centres of gravity for all loading conditions for vessel no. 1864.

Description	Weight (MT)	LCG (m)	VCG (m)	TCG (m)
LC_0				
Consumables and liquids	10,853.62	135.869	12.430	-0.700
Water ballast	63,564.65	166.416	10.266	0.000
Crude oil cargo	0.00	0.000	0.000	0.000
Deadweight	74,418.3	161.961	10.582	-0.102
LC_1				
Consumables and liquids	15,846.25	112.044	13.855	-0.480
Water ballast	2,968.39	69.709	5.646	0.000
Crude oil cargo	138,318.69	161.655	13.983	0.000
Deadweight	157,133.3	154.915	13.813	-0.048
LC_2				
Consumables and liquids	15,846.25	112.044	13.855	-0.480
Water ballast	32,137.79	170.344	2.710	0.000
Crude oil cargo	70,808.44	161.655	8.134	0.000
Deadweight	118,792.48	157.388	7.430	-0.064

Stability and righting lever

Table 4.2.9 and figure 4.2.6 shows the stability data and the right lever for vessel no. 1864. The vessel meets the stability criteria for all loading conditions.

Table 4.2.9: Stability data and evaluation of intact stability criteria according to IMO MSC.267(85) for vessel no. 1864. All units in metres unless otherwise specified.

Description	Symbol	LC ₀	LC ₁	LC ₂
Draught at A _P	T _A	9.422	14.723	12.216
Mean draught	\bar{T}	9.174	13.942	11.737
Draught at F _P	T _F	8.925	13.161	11.258
Longitudinal trim	ΔT	-0.497	-1.562	-0.958
KM		33.059	25.505	28.049
VCG		18.000	17.533	14.265
GM		15.059	7.972	13.784
GG'		0.083	1.429	1.047
G'M liquid		14.976	6.543	12.737
Rollback angle	θ	-27.3 °	-19.9°	-21.7°
Area A	A _A	2.479	0.724	0.1324
Area B	A _B	5.416	3.706	5.132
Max. GZ	GZ _{MAX}	8.685	6.732	8.816
Angle of max. GZ	θ_{MAX}	36.9°	38.8°	39.2°

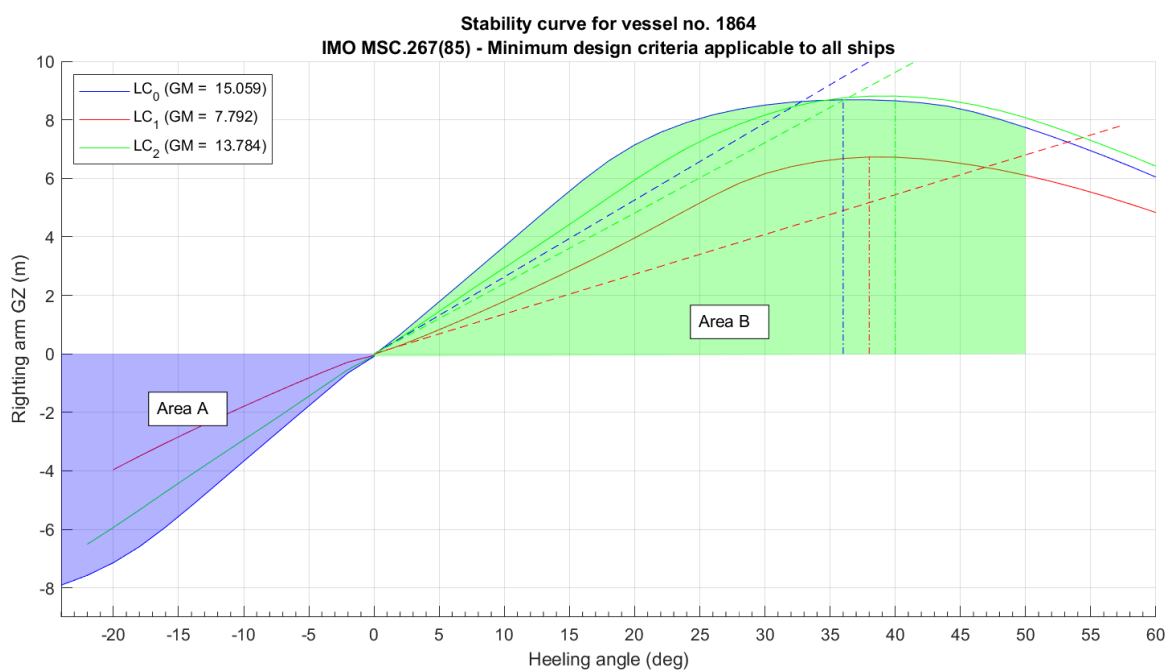


Figure 4.2.6: GZ curves for various loading conditions for vessel no. 1864, including intact stability, GM.

4.3 Cost breakdown

Exact cost estimation of any vessel is difficult without detailed engineering. In addition to material costs, painting, surface preparation and onboard outfitting, external costs such as labour, dry dock rent, scaffolding and outfitting costs must be accounted for as well. However, these additional external costs are not part of the optimisation scope and hence not included in the estimate and will furthermore not affect the output. Additionally, new-build prices are highly volatile and have significant variations depending on the place of production [20]. For the steel costs themselves, much more detail is required, with further weight breakdown, to get a more accurate price estimate.

However, a general estimate of steel prices and surcharges are presented in table 4.3.1. The surcharges are added to the basic rates of steel. The cost of installed construction steel does experience price variations and can range from US\$1,500/MT in the Middle East and China to US\$10,000 in Northern Europe. However, the typical rates vary between US\$2,000-US\$4,000. Hence a base rate of \$3,000 USD per metric tonne is used in the optimisation. The plates that are, e.g. used for tanks cost up to 20% more per tonnes due to its estimated surcharge.

Table 4.3.1: Costs and surcharges for paint, steel work and surface preparations. Currency is in USD. An average of \$22.5/m² was used for the surface preparation cost.

Coating	Rate
Surface preparation	\$20-\$25/m ²
Painting underwater areas incl. anti-fouling	\$40/m ²
Painting decks, topsides, exposed areas	\$20/m ²
Painting water ballast tanks	\$15/m ²
Painting cargo tanks, slop tanks, other tanks	\$10/m ²
Surcharges on basic steel rate	
Shaped plates and internals	15%
Single plates in one tank or compartment	20%

The rates and surcharge rates in table 4.3.1 is not an industry standard or official numbers from any organisation as this is often undisclosed in technical documents. The rates are general estimates from Altera that are used in general estimations [41]. In addition to the listed surcharges, there are many more that involve stiffeners, steel-work to be done in engine rooms and renewals. However, such detailed engineering has not been possible to complete when the scope is limited to the preliminary hull design.

When evaluating the cost estimates for the chosen vessels, it is crucial to keep in mind that the cost image is very simplified. The vessel's steel weight is likely to be somewhat accurate, as it is scaled using a volumetric coefficient, with a similar FPSO. However, the surcharges that accumulate in fitting the internal steel-work cannot all be accounted for, and hence the hull's steel cost is likely to be under-estimated. Additionally, painting and surface preparation costs only include the most extensive areas, such as the tanks, the deck area and the hull's side. The

paint used in the vessel's interior rooms, superstructure and on-deck processing plant, is not included. The resulting costs associated with each of the chosen vessels are shown in table 4.3.2. For both vessels, approximately 94% of the costs are associated with the steel cost.

Table 4.3.2: Main cost breakdown for vessels 1864 and 4843. All costs are in USD. Hull structure costs include the steel cost of internal water ballast tanks (WBT) and crude oil tanks (COT).

Cost area	Vessel 4843	Vessel 1864
Surface preparation	\$3,870,979	\$4,393,721
Painting		
Under water/anti-fouling	\$789,637	\$1,064,928
Horizontal deck	\$307,503	\$336,835
Hullside	\$296,097	\$477,015
Cargo tanks	\$587,477	\$565,373
Water ballast tanks	\$654,510	\$672,005
Total paint cost	\$2,635,224	\$3,116,156
Hull structure	\$59,639,690	\$66,689,443
Total costs	\$66,145,894	\$74,199,320

The differences in total steel costs between the two vessels are about 11%, while the difference between the total paint cost and surface preparation cost is 16.7% and 12.7% respectively. It is, however, expected that the surface preparation cost and paint cost would have a greater difference than the steel costs. Both vessels require the same cargo tank sizes, and thus the difference in their total steel cost is marginal. Conversely, the increased hull surface of vessel no. 1864 increases the painting, and surface preparation cost considerably.

Analysis

5.1 Algorithm evaluation

The FPSO optimisation process has involved the use of three different algorithm variations, mostly based on evolutionary algorithms. It is therefore of interest to determine which of these algorithms have performed best overall. Initially, the results from figure 4.1.1 in section 4.1 shows that no algorithm was definitely superior in generating optimal Pareto-fronts. Hence, it is important to look at the other measures of performance for the algorithms. One of the methods is the measure of normalised distance to U_P . Figure 5.1.1 shows the box-plot results of the distance $D(X_j)$. Initially, one can see that the local hybrid search generates individual solutions with the smallest distance to U_P . While this information may indicate that this algorithm is the best performing, it is important to consider entire Pareto-sets as well, to evaluate the algorithm's performance properly.

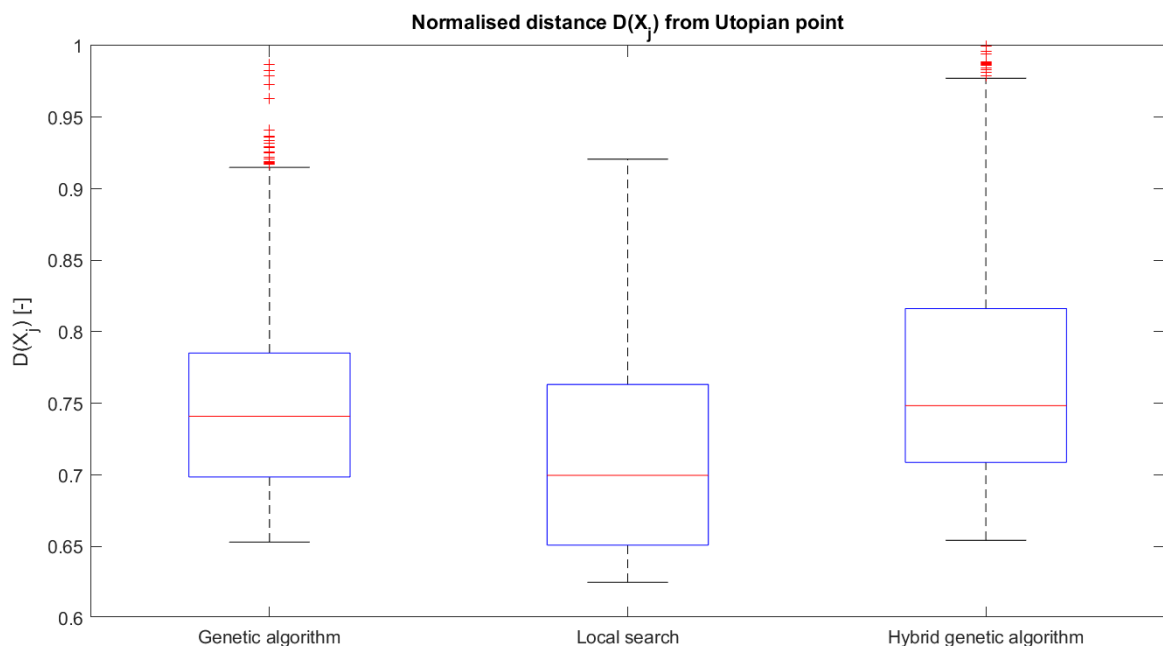


Figure 5.1.1: Box-plot results of the normalised distance $D(X_j)$ to U_P .

5.1.1 Goodness of solutions

When a large number of resulting points are generated from the optimisation, it can be difficult to determine how well the algorithms have performed. When considering individual points in the solution space, the measure of normalised distance to the Utopian point can give a good indication. However, when one wishes to evaluate the optimisation results as a whole, the entire Pareto optimal set should be evaluated. Consequently, one can use the hypervolume calculation to determine the 'goodness' of each optimisation run or a combined set of Pareto fronts. Since three variations of an optimisation algorithm have been used in the preceding process, the hypervolume can help determine which of these performs the best. Figure 5.1.2 shows the hypervolume for the optimisation results of the standard genetic algorithm (no local hybrid search or initial hybrid scores). Here, the reference point $r = (r_1, r_2)$ is chosen arbitrarily to be [13400,76]. While r does not have to be exactly at this point, the reference point's location must stay the same for all hypervolume calculations to correctly compare the results.

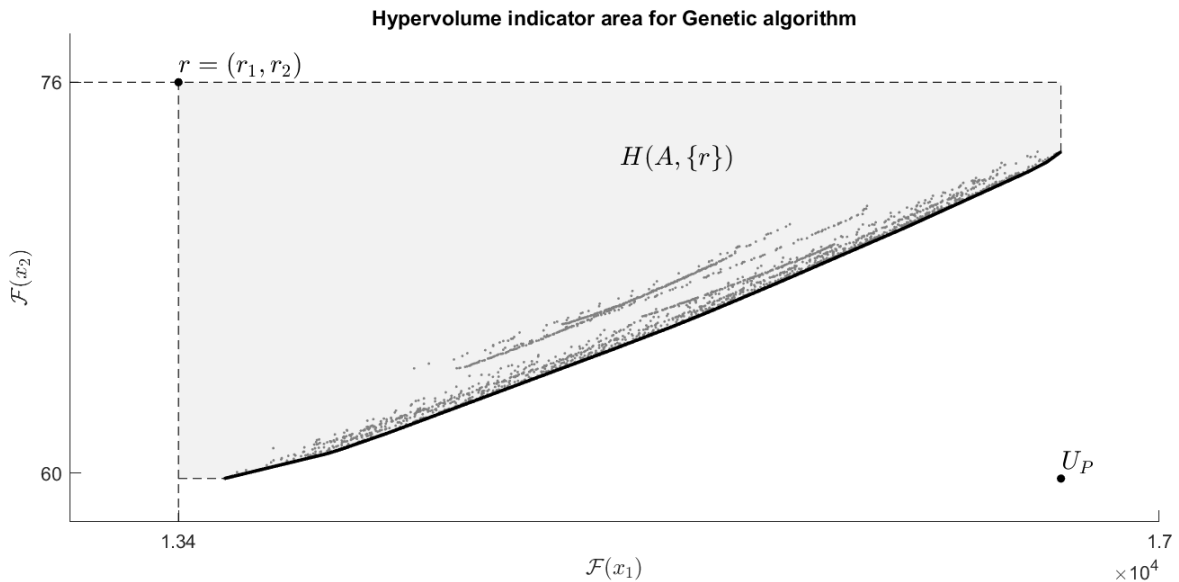


Figure 5.1.2: Hypervolume $H(A, \{r\})$ for the genetic algorithm optimisation, with reference point $r = (r_1, r_2) = [13400, 76]$.

The points inside the hyperspace represent all the individual solutions when this particular algorithm has been used. Each solution is, however, part of a set of Pareto optimal solutions. All the Pareto fronts together, form the overall Pareto front that is represented by the black line in figure 5.1.2. Considering the hyperspace denoted $H(A, \{r\})$, the grey area covers about 64.9% of the current solution space, with the opposite corner of r being the Utopian point U_P . A hypervolume was calculated for each of the algorithm variations, using the same reference point r . The resulting hypervolume indicator, I_H , is shown in table 5.1.1.

Table 5.1.1: Maximum hypervolume of the various optimisation algorithms.

Algorithm variation	Hypervolume indicator, I_H
Genetic algorithm (GA)	64.9%
Local hybrid search	68.1%
Hybrid genetic algorithm (HGA)	64.3%

From table 5.1.1, one may observe that the local hybrid search generates the highest hypervolume ratio at 68.1%, followed by the GA. This result is as expected, as the hybrid algorithm is better at finding a better global minimum once the solution has converged sufficiently. It is not likely that the hybrid search would obtain an equally good result if it did not have the precedent results from the GA. While the local hybrid search does generate reasonable Pareto-optimal solutions, its solutions have an extensive and thin spread. Despite the optimal solutions being viable and close to U_P , the thin spread allows for less flexibility in choosing one solution on the Pareto front.

The spread of the solutions along the optimal Pareto-front is calculated through the average distance measure when using the GA in MATLAB. A smaller average distance measure indicated that the solutions on the Pareto front are evenly distributed. However, the spread is also a measure of movement on the Pareto front. Hence the GA applies the following algorithm;

Algorithm 2: Termination of algorithm from distance measure

```

while stall generations < max. stall generations do
  | if  $\delta \bar{d}_j < \sigma_F$  and  $\sum d_j < \sum \bar{d}_{j-1}$  then
  | | terminate;
  | end
end

```

In short, if the average of the relative change in the spread of the Pareto solutions ($\delta \bar{d}_j$) over stall generations is less than the functional tolerance (σ_F), and the final spread ($\sum d_j$) is smaller than the average spread of over the last stall ($\sum \bar{d}_{j-1}$), then the algorithm stops. Consequently, the algorithm that produces a smaller spread and meets all the stopping criteria is likely to produce better solutions. From the results, the average distance measure between the solutions on the Pareto front is 0.14, 1.23 and 0.17 for the GA, local search and HGA respectively. Based on the average spread, the local search is spread very thinly, which is also evident from figure 4.1.1. Thus, when considering the spread, the local algorithm performs poorly. If one equally weights the performance of the average distance from U_P , the hypervolume ratio, and the total spread $\sum d_j$, the GA and local search perform equally well.

While the two first algorithms have an even performance, one would expect the genetic algorithm that uses the scores from the local search (HGA) to perform better. It is likely that the algorithm is not able to search outside the already converged minimum of the hybrid, and thus not find even better solutions. Consequently, using scores from a previous population, the diversity is severely limited and, as a result, is outperformed by the other algorithms on all measures. Despite not performing above the expectation, this algorithm variation was implemented to compare the solutions with the GA.

5.1.2 Optimal algorithm parameters

Section 3.2.2 presented the input parameters that were considered when setting up the genetic algorithm and local search. While the function and constraint tolerance are set to constant for all the optimisation runs, the choice of the initial population, Pareto fraction and maximum of stall generations were varied. There is no limit to the maximum number of generations, as this is set to infinity for all runs. Figures 4.1.4 and 4.1.5 shows the results sorted by their Pareto fraction and MSG-limit respectively. On neither of the figures, a clear trend indicated that one optimisation setting outperformed the other. Hence, in terms of maximum stall generations, a limit of 250 was adequate. When MSG was equal to 1,000, no better solutions were generated, and the increased number of generations only led to prolonged computation time.

However, a clearer trend is visible in the choice of the initial population. Figure 4.1.6 shows a clear separation between the resulting vessels that were generated through the use of a completely random initial population, or by a random selection of preceding results. If one considers the mean distance from U_P , the results with random X_0 , have a mean distance $\overline{D}(X_j)$ of 0.7837 and a hypervolume ratio of 52.9%. In contrast, the results where $X_0 = X_{j-1}$ have a mean distance of 0.7257 and a hypervolume ratio of 64.81%. Hence, overall the iterated X_0 (selecting previous solutions) gives a better, combined optimisation result. However, when considering individual solutions, some of the random X_0 -results are closer to U_P . Consequently, one of the vessels chosen for further study is generated from the random set of X_0 . The input parameters for the chosen vessels are represented in table 5.1.2.

Table 5.1.2: Input parameters of optimisation algorithms for the chosen vessels. Both vessels also had a constraint tolerance of 10^{-4} and a function tolerance of 10^{-3} .

Input parameter	No. 1864	No. 4843
Algorithm	HGA	Local search
Initial population choice	Quasi-random	Random
Max stall generations	500	500
Pareto Fraction	0.5	0.7

5.2 Optimal hull geometries

Two hull geometries were chosen in order to see the difference in optimisation results across the optimal Pareto front. Naturally, two vessels that are far apart on the front are likely to have different geometries, and hence the vessel diversity is represented. Vessel no. 1864 was chosen as it has the lowest utility cost, i.e. cost per square metre of deck area, while vessel no. 4843 was closest to the Utopian point, U_P . The following section aims at evaluating the difference in the vessels' geometry and characteristic trends of the resulting vessels.

5.2.1 Vessels comparison

From section 4, the main dimensions of the vessels showed that vessel no. 1864 is larger than vessel no. 4843 in most of the vessel's KPI's except for the utility price. Table 5.2.1 represents the individual values of the resulting chromosomes, showing that some of their measures are almost identical. The similarity is especially applicable in terms of the length abaft. The length measures x_1 and x_7 (length of aft sections and holds section respectively) are less than 1% different for the two vessels. However, the added length of vessel no. 1864 is primarily contributed by the increased bow length, represented by x_8 . Figure 5.2.1 illustrates the length difference in the vessels and thus showing that the main length contribution is in the bow. If one were to look at the price estimations of all vessels concerning their length's, the optimisation generates an expected cost of approximately US\$220,000/m length of boat, starting at US\$2.2 million. Consequently, the difference in length of the bow cost represents an 11.5% difference, while only contributing to a 9.1% increase in the deck area.

Table 5.2.1: Final dimensions x_i for the chosen chromosomes X_j . All dimensions of vessels given in metres unless otherwise stated.

Dimension	x_i	Vessel no. 4843	Vessel no. 1864	Difference
Length of aft section	x_1	44.6	44.9	0.7%
Midship width	x_2	55.0	55.0	0.0%
Stern width	x_3	55.0	54.6	0.7%
Moulded depth	x_4	22.0	26.5	18.7%
Height of transom stern	x_5	10.0	20.2	67.4%
Bilge radius	x_6	0.5	3.0	142.9%
Length of middle section	x_7	200.0	199.7	0.2%
Length of bow section	x_8	44.6	78.8	55.5%
Mean draught	x_9	14.8	14.1	4.9%
Length of cargo tank	x_{10}	166.4	173.7	4.3%
Breadth of cargo tank	x_{11}	47.8	39.0	20.3%
Depth of cargo tank	x_{12}	20.0	23.5	16.3%

However, the most noticeable difference in the dimensions in table 5.2.1 is the considerable difference between the vessels' bilge radius. The lower and upper bound for the bilge radius is given by $x_6 \in [0.5, 4]$. From prior studies, the latest designs of similar FPSO's have a bilge radius of around 3 metres. Hence, vessel no 1864's x_6 -value seems most appropriate. However, the bilge radius of vessel no. 4843 is only 0.5 metres. While the small bilge radius does meet the constraints of the optimisation, such a small bilge radius may lead to manufacturing challenges and undesirable stress concentrations. However, the structural integrity of the hull is not accounted for in the optimisation but can be mitigated through increasing the lower bound of x_6 to force a rounder hull geometry.

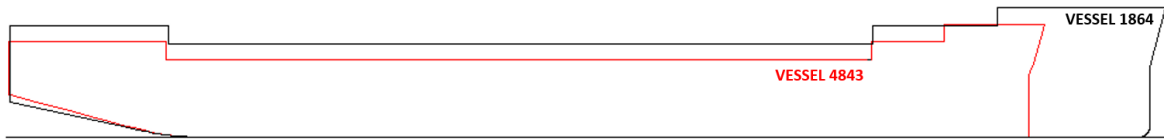


Figure 5.2.1: Centre line buttocks of vessel no. 1864 and no. 4843 in black and red, respectively.

5.2.2 Trim and heel

While the intact stability is a strict constraint of the optimisation, less focus has been given on the trim and heel of the vessel. An approximation of the longitudinal centre of buoyancy is calculated, which serves as a verification of sensible hull shape. However, the longitudinal trim has been excluded both as an objective function and a constraint. The exclusion is done purposely as an optimisation aiming to eliminate any heel or trim would result in a much more complicated task involving the movement of onboard systems and general arrangement setup. However, the centre of gravity and cargo tank weight, that represent 61% and 64% of vessel 1864 and 4843, respectively, is placed amidships to minimise any trim or heel. Additionally, due to the vessels' geometry abaft, less buoyancy is generated, resulting in a longitudinal trim of 1.56 and 2.72 metres for vessels 1864 and 4843, respectively. Both vessels are therefore trimmed bow up in location condition LC_1 .

5.2.3 Turret positioning and cargo tanks

The FPSO units are both assumed to have internal turrets built into the hull and are positioned on the centre line of the vessels' x-axis. While the turret position or its configuration was not part of any objective function, its location has been calculated from the cost function. The centre of gravity and the weight varies with different hull geometries based on scaled data from similar ships. However, there is a significant simplification with regards to the turret. Figure 5.2.2 shows a typical deck view of the location of the cargo tanks, water ballast tanks and the turret. Here, like many other FPSO's, the turret is located amidst the cargo tanks. Consequently, the cargo area is slightly small to accommodate the one million barrels of crude oil cargo and the cargo tanks may need to be longer.

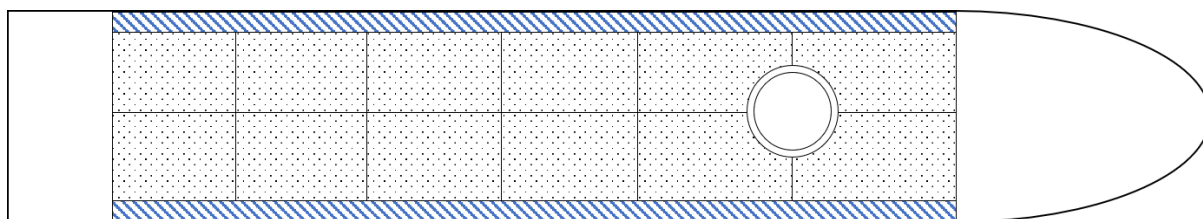


Figure 5.2.2: Example of turret positioning amidst cargo tanks. Water ballast tanks marked in blue with cargo tanks in between. Illustration not to scale nor representative of optimised vessels.

While this simplification may affect the accuracy of the building cost estimate, there are other FPSO units built where the turret is located further in the bow. However, regardless of the location of the turret, additional steel would be used for the turret moonpool and additional strengthening. Consequently, the cost estimate might be slightly low but no particular focus is given on the matter and the moonpool does not affect the optimisation result.

5.2.4 Fleet comparison

When determining the total weight and its distribution, the optimisation model has scaled some of another vessel's measures to get a more accurate representation. This feature is likely to generate accurate weight estimates. Meanwhile, the dimensions of the vessels are determined by the constraints and the bound of the model. Consequently, it would be worthwhile to ascertain whether the main dimensions are within reasonable ranges when comparing to other FPSO's.

Hence, the main particulars for over 170 FPSO's have been found in the maritime database of Sea-Web [43]. The vessels used in the comparison are all FPSO units, with no FSO's or cylindrical hulls (e.g. *Sevan* design) included in the count. Figures 5.2.3 to 5.2.5 shows the chosen vessels marked in red and green alongside the other vessels that are currently operational. A fitting line with R^2 -value of 0.971 is added to show a characteristic trend between the displacement and DWT, and that the chosen vessels lie slightly below this trend. Figure 5.2.3 shows the vessels' deadweight and its displacement showing that both vessels are slightly under the expected trend of the market. The model does account for consumables that are included in the vessel's DWT. Hence, the slightly lower DWT may indicate that the vessel carries a smaller amount of cargo than its displacement would suggest.

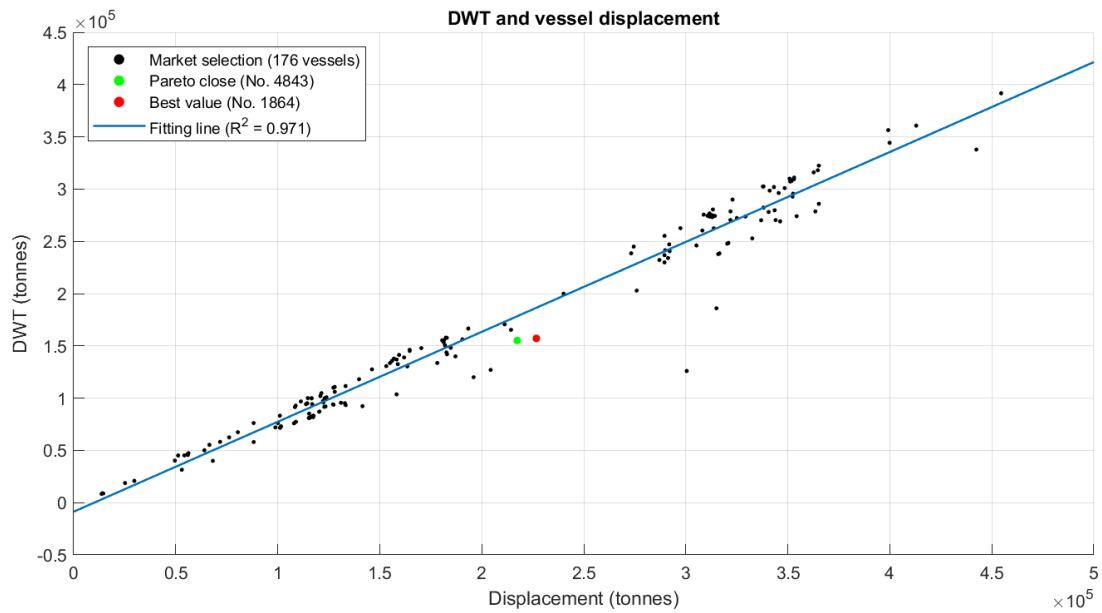


Figure 5.2.3: DWT and displacement of currently operating vessels. Assumed linear relationship for added trendline.

The relationship between the vessels' length and displacement is also of interest, as the length of the vessel is unaffected by any scaling, and entirely determined by the optimisation algorithm. While the relationship between the DWT and displacement was expectantly linear, the best fit for the length and displacement is exponential. The R^2 -value is 0.791, and hence not as strong a trend as in figure 5.2.3 albeit sufficiently strong to evaluate the vessels position. From figure 5.2.4, one can observe that vessel no. 1864 lies slightly over the general trend, while vessel 4843 is slightly under the trend. Both vessels have the same cargo capacity at full load, and hence the larger vessel is expected to be above the trendline.

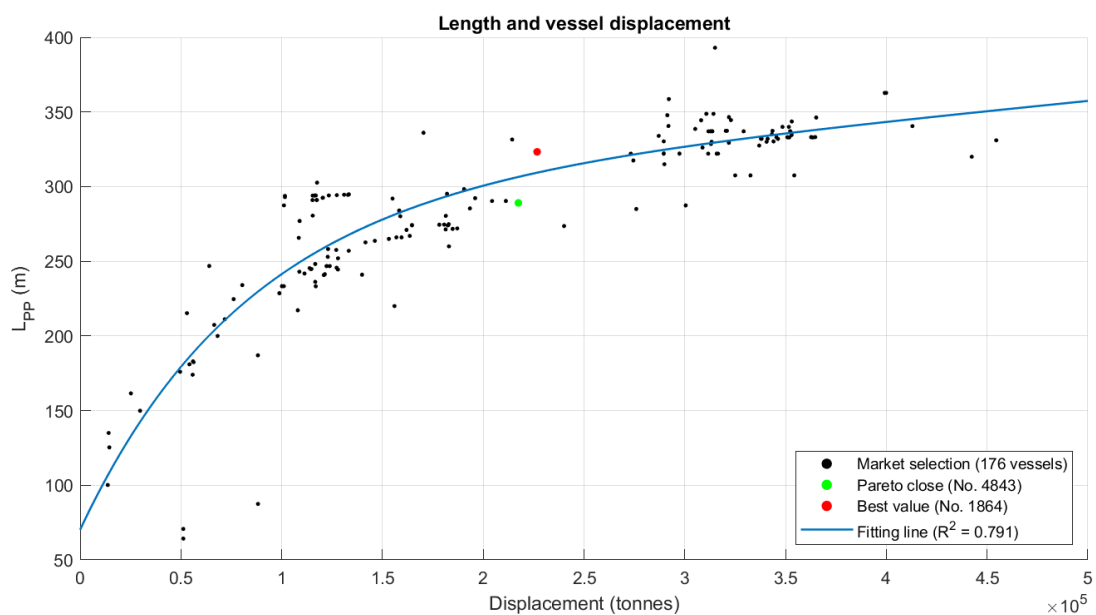


Figure 5.2.4: Length between perpendiculars, L_{PP} and displacement of currently operating vessels. Assumed exponential relationship for added trendline ($f(x) = ae^{bx} + ce^{dx}$).

Lastly, the relationship between the vessels' breadth and length is considered. Although the vessels are seemingly more scattered, a linear trendline with an R^2 -value of 0.829 is the most fitting. From figure 5.2.5, one can observe that vessel no. 1864 is quite close to the general trend, while it would be expected that vessel 4843 would have a breadth of approximately 47 metres, i.e. 15% lower if the trend is followed. However, the high VCG for the topside modules and turret, the algorithm is likely to maximise the breadth to keep $GM > 0.5$. Additionally, the added cost of a vessel's extra length is likely more than added vessel breadth.

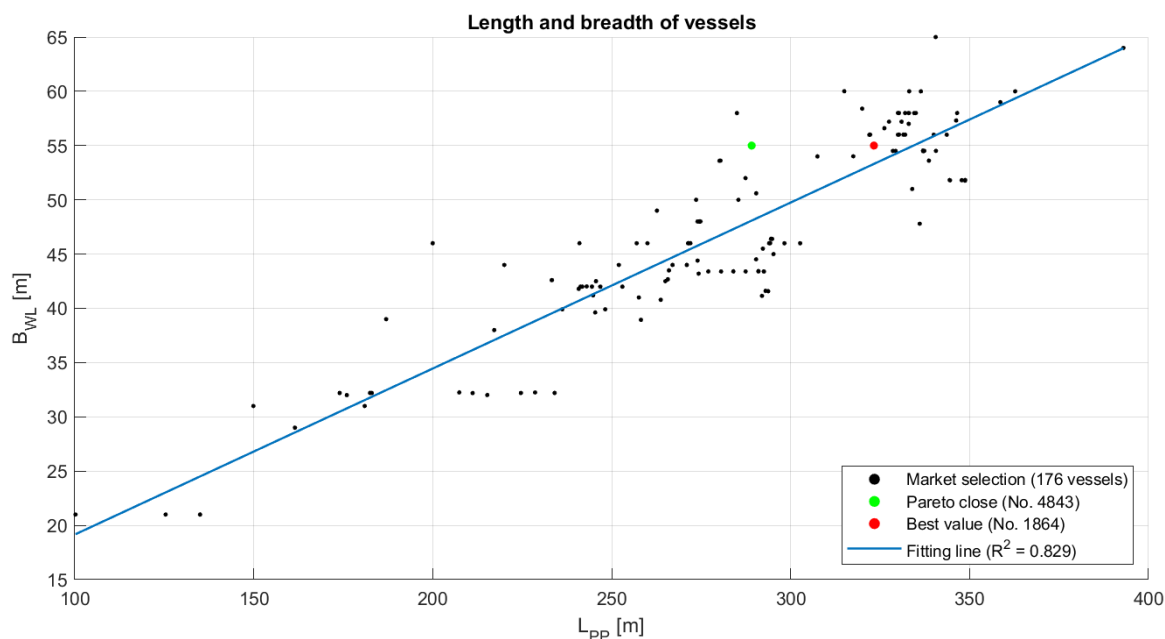


Figure 5.2.5: B_{WL} and L_{PP} for currently operating vessels. Assumed linear relationship for added trendline.

Additionally, one may consider the ratio between the breadth and draught (B/T-ratio) to get an idea of the vessels' loading condition and stability. While the B/T-ratio for vessel 1864 and 4843 is 3.90 and 3.72, respectively, the mean ratio for the currently operating vessels is 2.87. This relationship may indicate that the optimised vessels are slightly stiffer than the operating vessels. Based on the GM-values of Altera's operational FPSO units, the GM-values of vessel 1864 and 4843 are indeed high, but further investigation shows that stiffness is similar to their shuttle tankers. Albeit the intact stability being quite stiff, the ranges of GM-values for the various loading conditions seem reasonable.

5.3 Vessel response

Throughout the optimisation process, the vessel's motions have been evaluated. The stopping criteria illustrated in the flowchart in figure 3.2.3 dictates whether the vessel's motions should be evaluated based on its GM-value. If the minimum intact stability criteria are met, the MSDV-value for the vessel is investigated. Consequently, if $\text{MSDV} > 15$, the solution is not feasible. The heave acceleration is given by $\Phi_v = \varpi^2 \Phi_u$, where $\Phi_U = \sqrt{\Phi_w^2 + x^2 \Phi_\theta^2}$, and hence the longitudinal point x where the evaluation of the MSDV-value must be taken into account. When the distance increases from the centre of gravity, the heave acceleration and pitch acceleration increases as well. Thus, one would expect that the MSDV-value is higher at the accommodation superstructure in the bow than in the vessel's centre of gravity.

All of the vessels that appear on the Pareto front are feasible solutions, and hence passed the criteria regarding the response in regular seas. The testing conditions during the optimisation was the sea state with a return-period of 10,000 years. However, it is not expected that the MSDV-values are below 15 at this point, and the sea state was used to generate the maximum possible motion from the vessel. Since the vessel's motions in a normal sea state are still of interest, a comparative study with the ShipX/VERES vessel motion program was completed. A comparison was therefore completed for the response amplitude operators for heave, pitch and roll as shown in figures 5.3.1 and 5.3.2. The response amplitude operators are given in appendix D).

Figure 5.3.1 shows that the estimation from the simplified model described in section 2.2.2 is quite accurate in terms of difference between the η_j/A -values. The estimate of the model and the ShipX output are very similar up to about when $T_p = 13$ seconds when the RAO's start to diverge more. While it is evident from the graphs that the simplification slightly under-estimates the RAO-values, there is only a 15.8% difference in η_j/A when $T_p > 13.0$ seconds for heave and only 10.9% for pitch.

For both η_3 and η_5 , the simplified model under-estimates the amplitudes of the vessel. Albeit not presented here, the RAO's for vessel no. 1864 suffered from the same under-estimation. When $\beta \neq 0$, the under-estimation is likely caused by the assumptions of uncoupled motions and box-shaped geometry of the vessel. Since the RAO's presented above have incoming waves 30° from the bow, the pitch displacement and acceleration may likely be lower than the actual value when coupled with the roll motion. This limitation is essential to consider when calculating the MSDV, as the accelerations are the basis of this calculation.

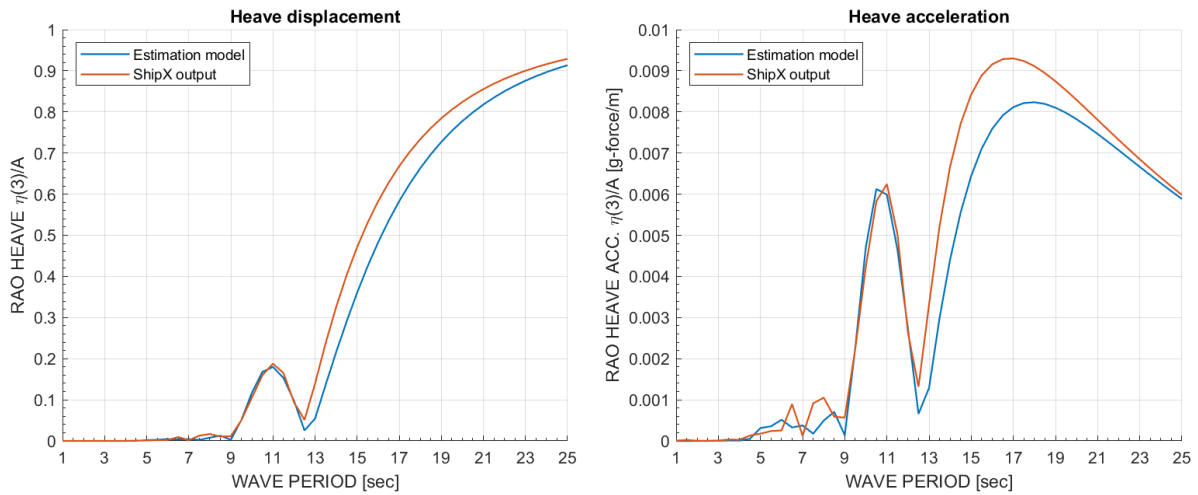


Figure 5.3.1: Response amplitude operators for heave displacement and heave acceleration at centre of gravity for vessel no. 4843. Regular waves, 30° incident wave angle.

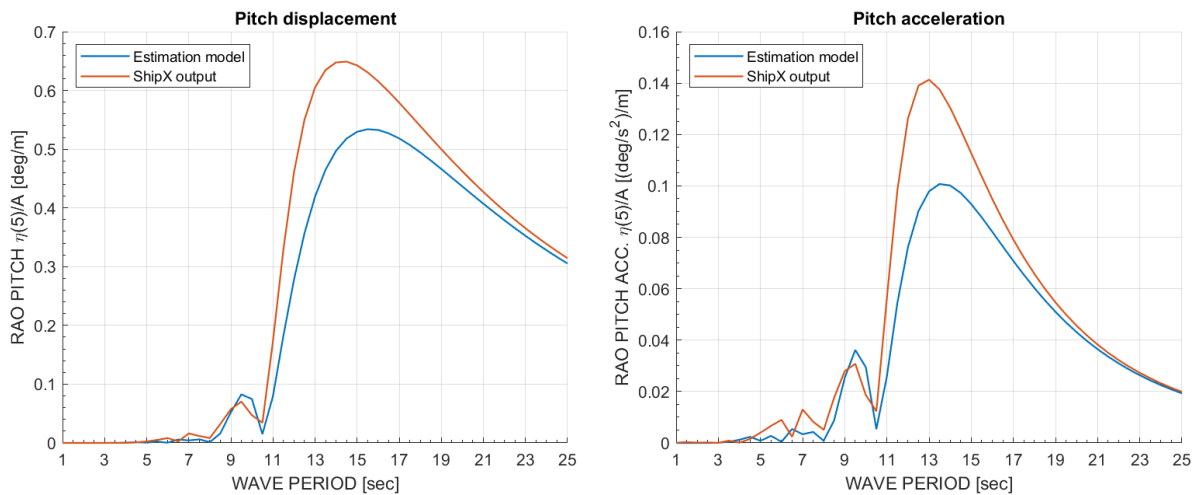


Figure 5.3.2: Response amplitude operators for pitch displacement and pitch acceleration in the centre of gravity for vessel no. 4843. Regular waves, 30° incident wave angle.

Hence, to evaluate the impact of the model's under-estimation, the MSDV's were calculated using both ShipX and the simplified model given the conditions given in the various return-periods. Figure 5.3.3 shows the MSDV at the vessel's centre of gravity and accommodation. It is assumed that the centre of the accommodation superstructure is located at 280 metres in front of A_P . Additionally, the acceleration would likely be even higher when one considers a location on the vessel as high as the accommodation structure or helicopter platform. However, neither ShipX or the simplified model accounts for any gain in acceleration with increased distance from the waterline. This shortcoming in the models will thus likely result in a slight under-estimation of the calculated MSDV.

As expected, the calculation of MSDV using the acceleration data from the model of Jensen et al. under-estimates the values both at the centre of gravity and at the accommodation. While the limit of $15 \text{ m/s}^{1.5}$ is breached for all wave heights when using ShipX, the simplified model never

reaches this limit. Consequently, the optimisation has rendered the vessel a feasible solution in terms of maximum MSDV-value.

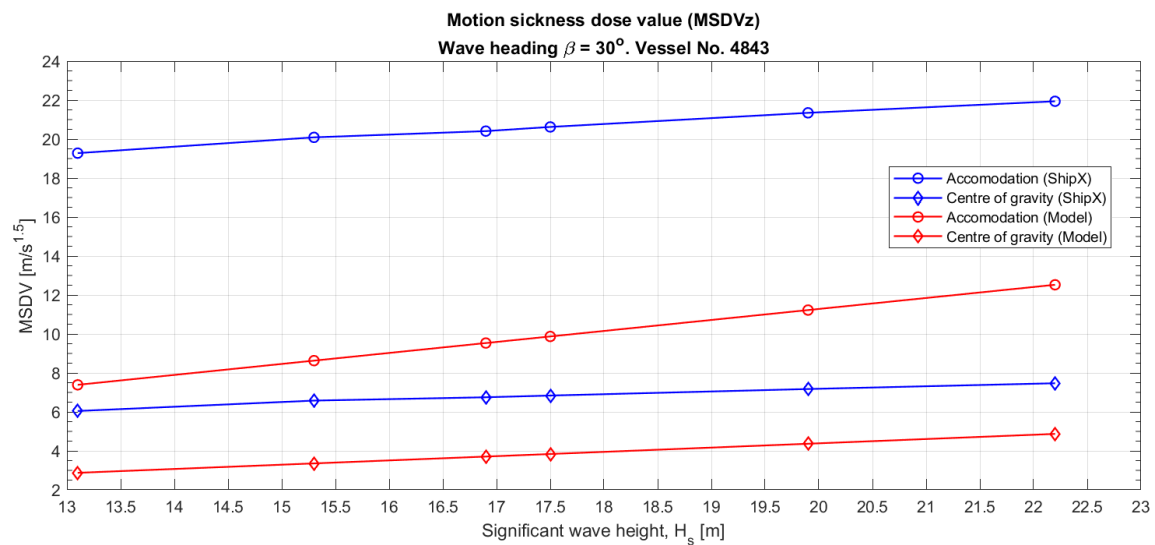


Figure 5.3.3: Motion sickness dose value at centre of gravity (COG) and accommodation area of vessel. Points are equivalent to H_s and T_p for chosen return periods.

From the onset, it was known that the empirical approach to calculating the vessel's motions would have some shortcomings that may affect the result slightly. However, the model is not sufficiently accurate, and hence the chosen vessels do surpass the limit of $MSDV < 15$. While the simplified calculation method allowed a motion evaluation parallel to the optimisation, the method has allowed the chosen vessels that do not necessarily meet the criteria.

Hence, the subsequent analysis of the vessel's motions and operational criteria is evaluated using the ShipX software. Here it is assumed that motions are coupled, and the results are likely to be reliable.

5.3.1 Crew comfort

Figure 5.3.4 and 5.3.5 illustrates the MSDV in relation to the wave peak period, for various H_s -values. Since a wave heading of $\beta = 30^\circ$ generates the most significant values for both vessels, only this heading is used when MSDV for various wave heights. The threshold of $MSDV < 15$ are marked on the figures, showing that the criterion is only met when $H_s < 10$ metres. From initial observation, one can see that vessel no. 1864 has a generally lower MSDV than vessel no. 4843 for all wave periods. Around a wave period of approximately 13-14 seconds, vessel no. 4843 has approximately 18% higher MSDV than the larger vessel no. 1864. Consequently, when the significant wave height is 20 metres, approximately 13% of the crew onboard vessel 4843 will be seasick, in comparison to the 11% on vessel 1864.

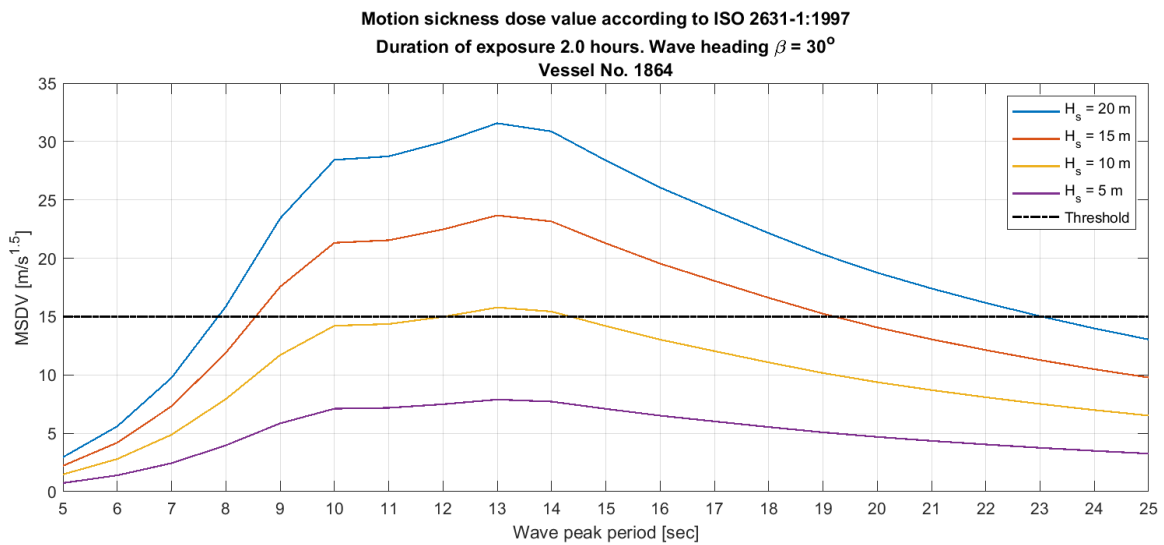


Figure 5.3.4: MSDV with respect to peak period of wave. Maximum value given six degrees of freedom in various H_S .

Vessel no. 1864 is a larger ship, with a length of 323.3 m versus 289.1 m of vessel 4843. Hence, it was expected that the vessel's accelerations would be smaller than the other, and thus have a lower MSDV. Furthermore, assuming deep water and that $k = \frac{2\pi}{\lambda}$ (where λ is the wavelength), one can deduce that the maximum accelerations occur when $\lambda/L \sim 1$. This relation is also expected, as the vessel will achieve the highest pitch and pitch acceleration when $\lambda/L = 1$ as seen in figure 5.3.2. From the response amplitude operators presented in appendix D, the roll acceleration has little effect on the vessels' MSDV, as $\ddot{\eta}_4$ only increases dramatically when the wave peak period, $T_p \rightarrow T_{N,4}$.

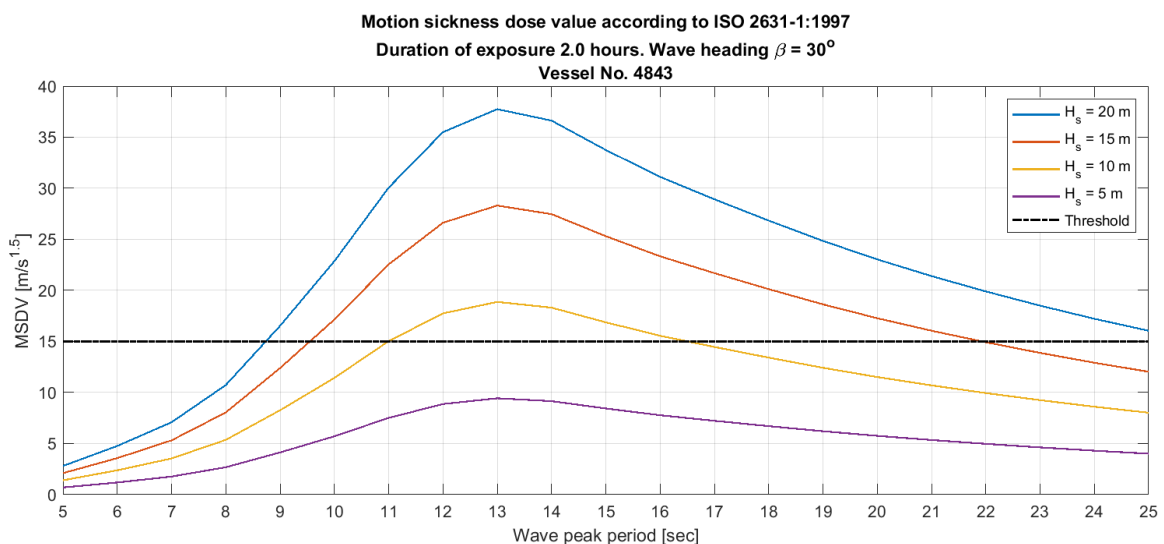


Figure 5.3.5: MSDV and T_p . Maximum value given six degrees of freedom in various H_S .

Figure 5.3.6 shows the corresponding ranges of H_S and T_p where the $MSDV < 15 \text{ m/s}^{1.5}$. The areas below the boundary lines represent the area where the vessel fulfils the MSDV-criteria. As expected, the maximum allowable H_S decreases as $\lambda \rightarrow L$. The minimum point on the criteria boundaries occurs in the same place as the maximum MSDV, which is where $\lambda/L \rightarrow 1$. Additionally, the load cases for the various return-periods are plotted to determine if the criteria are fulfilled in these sea states. The operational limits are not affected when $T_p < 4$ seconds, as excitation forces are insufficient to enact a motion on the vessel. Each wave heading will generate a boundary line, but here the heading probabilities are applied, and hence the boundary lines are weighted according to the heading probabilities.

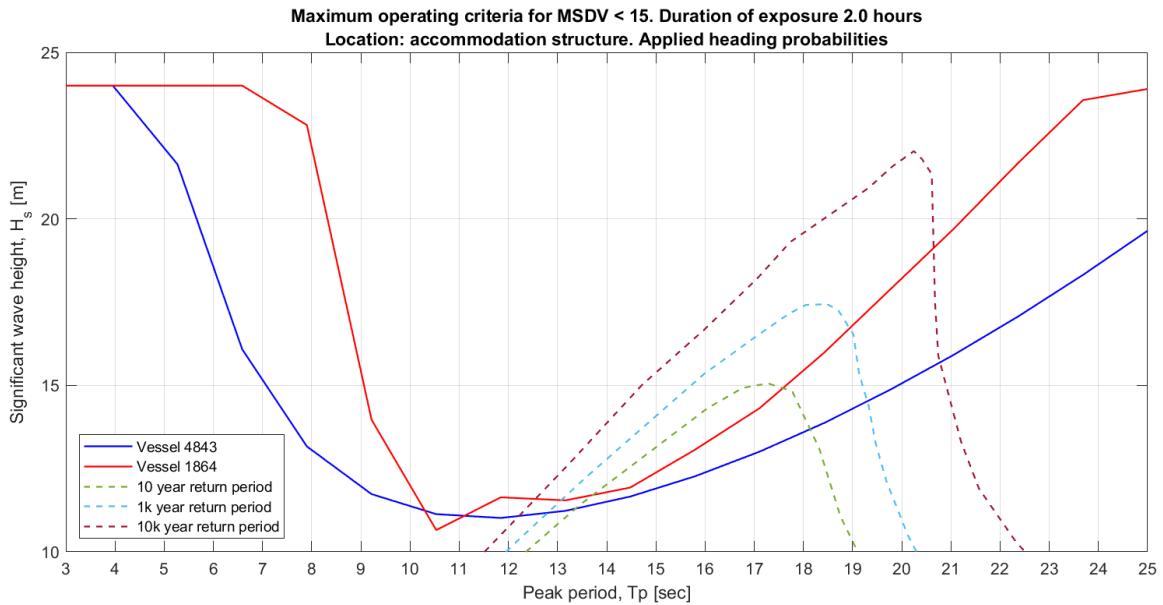


Figure 5.3.6: Criteria boundary for $MSDV < 15$ for both vessels with applied heading probabilities. Return period contours are added. The criteria is met when below the boundary curve.

5.3.2 Helicopter operation

The vessels are highly dependent on helicopter operation to transport crew to and from the vessel when stationed at location \mathcal{X} . Hence, it is important to consider the operability of the helicopter operation. The helicopter operational criteria are not flexible, with strict limitations for day and nighttime flights. Figure 5.3.7 shows the operational boundaries for helicopter operation for vessel no. 1864 and 4843, both day and night. These boundaries are a result of all the criteria listed in table 2.3.1, and it is the minimum value for each criterion that generates the boundary front.

While it may seem that the operational boundaries severely limits the possibility of conducting safe helicopter flights, one must also consider the wave frequencies in appendix A. This wave frequency table applies for all year on location \mathcal{X} . If the wave frequencies are accounted for, as well as the heading probabilities, vessel no. 4843 may perform helicopter operations 55.2% of the year, regardless of assuming as much operation during day and night.

Correspondingly, vessel no. 1864 will have a helicopter operability of 60.2%. For both vessels, the maximum heave rate is the limiting condition for both day and nighttime helicopter operation for all peak wave periods. As with the operational limits depicted in figure 5.3.6, one would expect vessel no. 1864 to have a higher T_p before the maximum H_S decreases. However, the RAO for the heave velocity of the vessels (see appendix D) shows that the heave velocity increases considerably earlier for vessel 1864, and hence the operational boundary's maximum H_S decreases earlier.

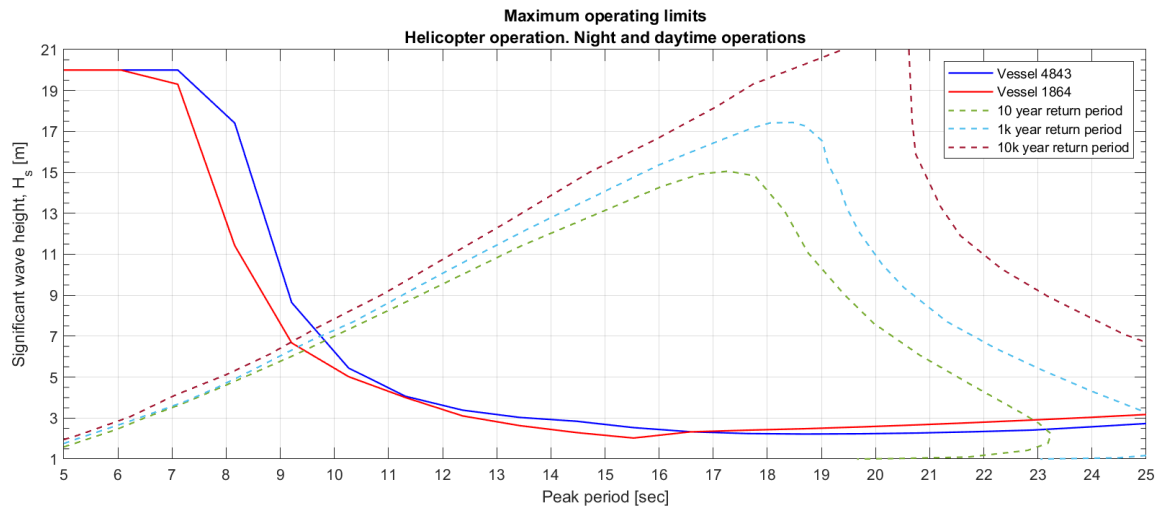


Figure 5.3.7: Maximum operating limits for helicopter operation. Return period contours added.

5.3.3 Heavy weather lockdown

The heavy weather lockdown occurs when the heave displacement is above 8.5 metres, the pitch is higher than 5° and roll is greater than 10° at the centre of gravity. The operation onboard does not necessarily close down completely, but there will be limited movement on deck for the personnel and movement of heavy machinery will be suspended. Figure 5.3.8 and 5.3.9 shows the operational bounds for the given criteria.

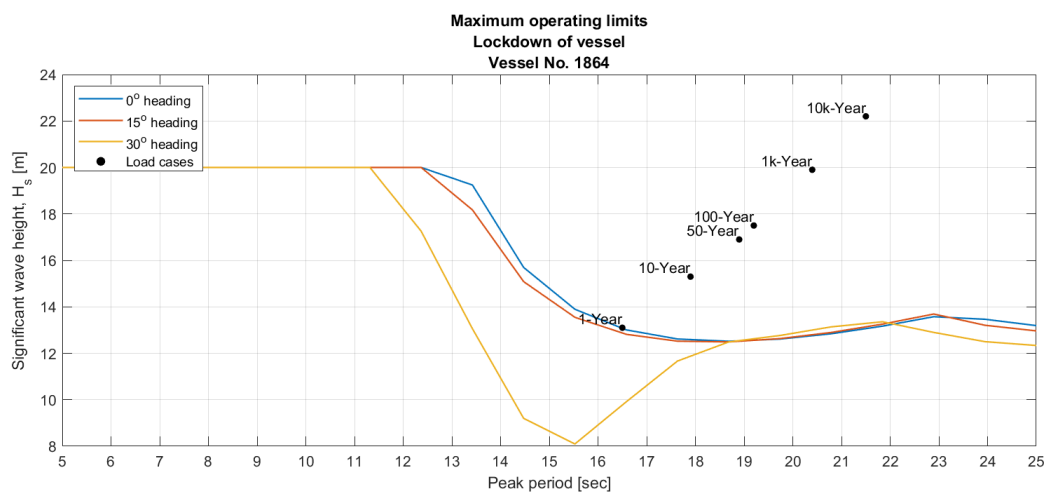


Figure 5.3.8: Maximum operating limits for personnel elevator and limited movement of heavy equipment on vessel 1864. Load cases represent peak of the return period contour line.

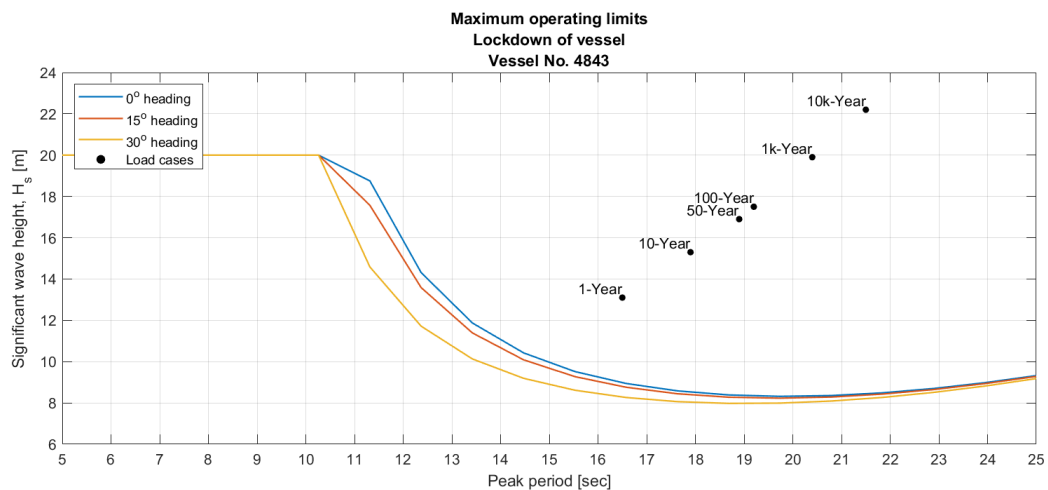


Figure 5.3.9: Maximum operating limits for personnel elevator and limited movement of heavy equipment on vessel no. 4843. Load cases represent peak of the return period counter line.

It is interesting to note that there is a considerable difference between the boundaries of 15° and 30° for vessel no. 1864, while the difference between the corresponding headings for vessel 4843 is not that different. Since the lockdown criteria are dependent on both heave, pitch and roll, one must consider all of the equivalent RAO's. Vessel no. 1864 has a natural period $T_{N,4} \approx 15.5$ seconds, hence the boundary's minimum point at this period. The notable difference in the heading boundaries is likely caused by the relatively significant differences in the RAO for roll. The difference in amplitudes for the various headings in pitch and roll is not very significant, whereas the amplitude difference in roll is twice as high when $\beta = 30^\circ$. Consequently, the effect of the roll limitation is magnified for the lockdown criteria boundary, and hence H_S decreases more rapidly as $\beta \rightarrow 30^\circ$.

Conversely, vessel 4843 experiences less of a difference between the heading angles. It is as expected where a heading of 30° renders the lowest criteria boundary. When the wave period is approximately 10 seconds, the smaller vessel 4843 is excited and the maximum allowable H_S decreases earlier than the larger vessel 1864. While the vessels may need to complete a lockdown procedure when reaching the return-periods illustrated in the figures, the short term statistics (based on the wave-frequency table) allow almost 100% operability throughout the year in normal conditions.

5.3.4 Loading and offloading

The FPSO's are dependent on shuttle tankers to offload the crude oil that is stored in the cargo tanks. The North Sea shuttle tankers are most commonly Aframax or Suezmax tankers with a deadweight capacity of around 80,000 to 160,000 DWT. Hence, least two shuttle tankers are necessary to offload the cargo if the tank is full with its 1,000,000 barrels. However, Altera along with its shuttle tankers have operational limits in terms of a given weather window. The requirements during connection and offloading are different as more precision is required during mooring and connection with the FPSO. During mooring, the maximum significant wave height of 4.5 metres is allowed. When the connection is made with the FPSO, the same limit is moved to $H_S < 5.5$ metres. For both scenarios, a maximum wave period of 15 seconds is allowed. While not part of this analysis, when mooring, the visibility must be at least 500 metres [37].

Figure 5.3.10 shows the cumulative probability of the significant wave height and the cumulative probability of the peak wave period. The probability is based on the wave frequency table given in appendix A. The dashed lines represent the loading limits when mooring and offloading. Looking at the significant wave height and wave period probability, we have that:

$$\begin{aligned}
 P(\text{mooring}) &= P(H_S < 4.5) \times P(Tp < 15) = 87.69\% \\
 P(\text{offloading}) &= P(H_S < 5.5) \times P(Tp < 15) = 90.98\% \\
 P(\text{mooring}|\text{offloading}) &= \frac{P(\text{mooring}) \times P(\text{offloading})}{P(\text{offloading})} = 87.69\%
 \end{aligned}
 \tag{5.3.1}$$

Hence, the FPSO's may complete mooring and offloading procedure 87.7% of the time given the wave probabilities at location \mathcal{X} .

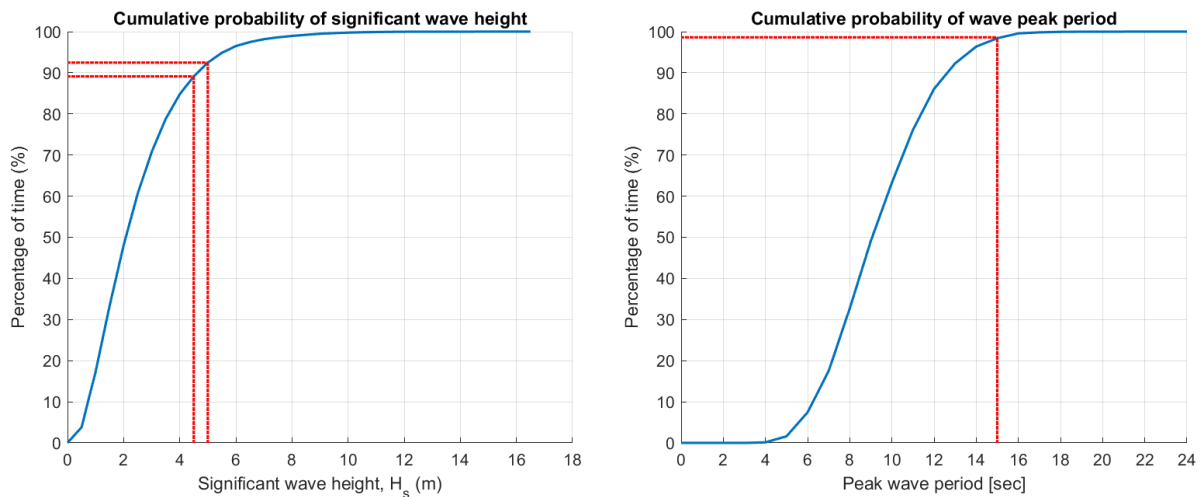


Figure 5.3.10: Cumulative probability distribution of significant wave height from JONSWAP wave spectrum at location \mathcal{X} .

5.3.5 Weather and heading dependent operability

The criteria boundaries for the helicopter operations and lockdown criteria given in figures 5.3.7, 5.3.8 and 5.3.9 give an indication of which sea-states operation is possible. Additionally, one can evaluate the overall operability of each criterion by considering the wave probability from the location's wave frequency table. From figure 5.3.10, one can observe that most of the sea states occur in the middle ranges of H_S and T_p , and hence the actual operational window may be larger than the initial impression from the criteria boundaries.

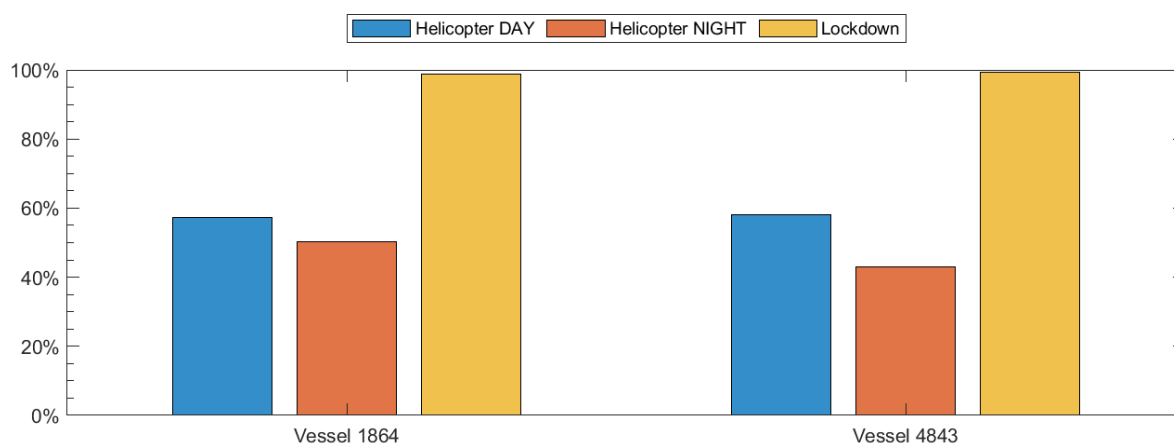


Figure 5.3.11: Percentage operability for both vessels. Heading probabilities applied.

Figure 5.3.11 shows the weighted operability for helicopter operations and lockdown procedures. It is weighted in accordance with the heading probabilities of location \mathcal{X} . Given a constant sea-state of 3 hours, the vessels can perform helicopter operations around 58% of the time during the day, and 47% during the night. The lockdown criteria are less strict, and hence the lockdown only occurs 1% of the time; equivalent to 3.7 days a year. In conclusion, one may argue that of all the requirements and limitations, it is the strict external regulations of helicopter operations that limit both vessels' operability. A customer would normally expect a minimum operability of 70 to 80% of the year for helicopter operation. Hence, this limiting criteria has the greatest need of improvement.

5.4 Verification and validation

The following section aims to evaluate and verify the output of the optimisation, as well as considering the validity of the current optimisation process.

5.4.1 Realistic vessel motions

The RAO generated from ShipX using two-dimensional strip theory, and the simplified empirical model is very similar. As these methods are widely different and have disparate assumptions, one can assume that the data is correct when reaching the same conclusion. Despite the simplified model's shortcomings, both models agree on the areas of maximum amplitude and general shape of the curves. However, to safely validate these motions, model scale testing in a tank facility must be performed. While such activity could further validate the hulls' optimality, such tests are not part of the preliminary design phase. Additionally, such resources are not available for the current scope.

5.4.2 Price estimation

Estimating an accurate build cost in the shipping and offshore industry is extremely difficult, as there are multiple market factors to consider. Production location, commodity prices, regulations and varying customer requirements all contribute to a highly complex pricing scheme. However, an acceptable estimate can be generated if the scope is narrowed, and thus being able to compare the vessel's price to currently known newbuilds. Current estimates within the ship-brokering community suggest that the newbuild price of a Korean-built Suezmax tanker is approximately at US\$55 million, while a VLCC costs approximately US\$88 million [38].

Both the optimised FPSO's carry one million bbl's, which is equivalent to almost 140,000 MT of crude oil cargo, which is slightly less than a conventional Suezmax vessel. Consequently, the expected price range of the FPSO's would be around this price if only the hull is included in the price estimation's scope. However, even small changes or modifications may increase the price dramatically. For instance, a Suezmax shuttle tanker with a fitted external bow turret can cost up to US\$120 million. Thus, the estimated prices of US\$66.15 and US\$74.2 million are slightly high in terms of a conventional tanker, considering the items included in the scope. However, due to the long operational life under harsh conditions, one should also consider the following:

- The lightship steel weight of an FPSO of the chosen size is usually about 35% higher than a conventional Suezmax tank vessel due to hull reinforcement for longer operational life. Hence, a higher steel cost should be expected.
- Offshore regulations are stricter in terms of fatigue and the costs to ensure 25 years of operation without docks increases considerably.
- The LQ is much larger as FPSO's accommodate 60 to 120 crew members in comparison to a tanker's 20. Consequently, more lifeboats and accommodation services are required at an increased cost.

The factors, as mentioned above, can thus verify the high price estimate of the optimised FPSO's. However, it is essential to underline that fact that the estimated price only includes the hull

structure and nothing of the offshore equipment, turret, or deck processing plant. A ready-to-sail FPSO with a cargo capacity of 1 million bbls is likely to cost in the region US\$1.0-1.5 bn [30].

5.4.3 Three-dimensional objective space

A genetic algorithm was used in the precedent optimisation process due to its function of parallel computing of multiple objective functions. Also, since a large number of decision variables were to be optimised, a calculus method may have had difficulties in providing an optimal solution. However, the computational power of an evolutionary algorithm like the GA, is properly exploited when there are several objective functions. Hence, a two-dimensional problem might evolve to larger dimensions given by $\mathcal{F}: \mathbb{R}^2 \rightarrow \mathbb{R}^N$. Consequently, the objective functions are given by $\{\mathcal{F}_1(x), \mathcal{F}_2(x), \dots, \mathcal{F}_N(x)\} = \mathcal{F}(x)$, where $x = (x_1, x_2, \dots, x_n), x \in \mathbb{R}^N$ [2].

While the optimisation of the FPSOs' hull only used two objective functions, there exists the possibility of including additional objective functions. If one were to extend the scope to include structural integrity and hydrodynamics, objective functions evaluating the ultimate strength and resistance could benefit the overall result. However, only two objective functions were used in the optimisation but with some added complexity. The function calculating the vessel's motions based on the empirical model did, to some degree, act as an objective function. Since the function was not part of the GA's objective, the vessel's motions were not minimised. However, the motions had to be sufficiently small in order for the optimisation to generate a viable solution, and hence the vessel motions function has acted as third, binary objective function embedded in the cost function. Additionally, since one would expect exponential growth in computation time by adding an RAO-calculation for each ship, the current setup was likely beneficial. Consequently, the optimisation did not use too much time while considering the vessels' motions only if the algorithm deemed it necessary.

A multi-dimensional optimisation space where $N > 3$ is a possible optimisation and may generate positively unexpected solutions due to the complexity of the algorithm's search space. The current optimisation evaluation considers two-dimensional measures such as the distance to U_P given by $D(X_j)$ and the hypervolume I_H , which is a hyper-area. Consequently, if a third objective function is applied, I_H would become a hyper-volume, and U_P would move to a corner of the feasible search cube. Therefore, while multi-dimensional optimisation where $N \gg 3$ is a better use of the GA, the task of evaluating the results becomes substantially more complex and the solution space abstract.

5.5 Sensitivity of results

5.5.1 Upper and lower bounds

The hull geometry is limited by the linear and non-linear constraints, as well as the lower and upper bounds. Both of these numeric limitations ensures a suitable hull shape. When extending the lower and upper bounds of the optimisation, the bound limits are multiplied by a factor, thus increasing or decreasing the bound for each variable correspondingly. As a result of the easing of the bounds, a larger search space is generated, and better solutions may be found. Thus, for each run, the factor dt changes and is given by $dt = \{0.50, 0.55, 0.60, \dots, 1.00\}$. Hence, the following changes are done before each run;

$$\begin{aligned} \min(x_i) &= \text{LB}_i \cdot dt \\ \max(x_i) &= \text{UB}_i \cdot \frac{1}{dt}. \end{aligned} \tag{5.5.1}$$

When the extension is zero, $dt = 1$ and no change in the bounds occur. Hence, the utility price remains the same as the lowest of the precedent optimisation runs. As dt decreases, the bounds are 'loosened' for both the lower and upper bound, and a broader search area is generated. Ultimately, the bounds are extended by 50% at each end of the bound range. Figure 5.5.1 shows how the mean utility price (USD/m²) decreases as the search area is extended. Consequently, by extending both bounds for each variable by 50%, the decrease of the mean utility rate is only 2.14%. The small change in the mean utility rate is likely to be the cause of the still strict, non-linear constraints and shows that the optimisation is not very sensitive to substantial changes in the upper and lower bounds. In conjunction with the bounds, it is the constraints that are likely limiting the output more than the boundary values.

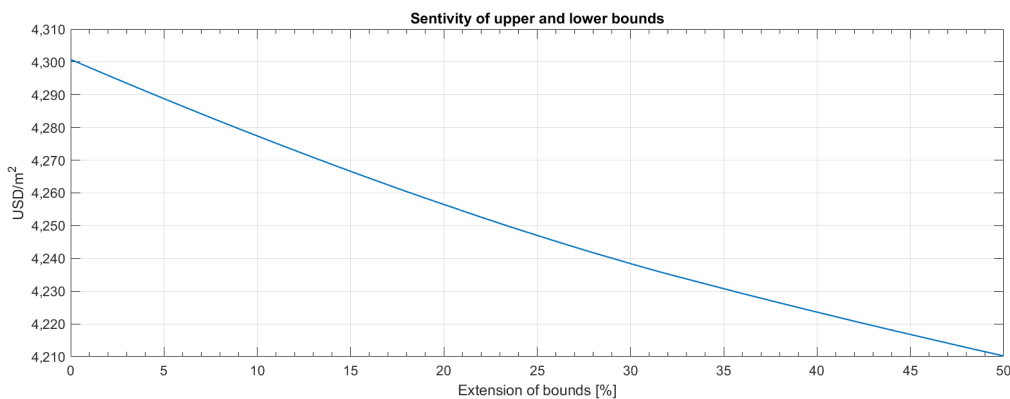


Figure 5.5.1: Change of the average utility price (US\$/m²) as the upper and lower bounds of the algorithm are extended step-wise, up to 50%.

The algorithm's sensitivity when evaluating the bounds may be quite clear as there is a distinct change in the average utility rate. However, the GA, like any other evolutionary algorithm, is dependent on some degree of randomness when the chromosomes evolve to feasible solutions. Consequently, the measure of sensitivity in other areas such as constraint and function tolerance, stall generations and Pareto fractions are less evident.

5.5.2 Motion sensitivity

From figure 5.3.6, one can observe that the vessel does not meet the requirements for most of the return-periods, represented as load cases. However, prior testing on tank vessels and FPSO's indicate that no commercial vessel that is not a cruise ship meets the requirement of $MSDV < 15$. While the onboard personnel boats of the current FPSO's calm motions, the criteria is not met here either. If the V_P -criteria increases from 5% of the personnel being seasick, to 6 or 7%, both vessels meet the requirements for multiple return period's load cases.

Figures 5.5.2 illustrates the current criteria boundary (in blue), in addition to the possible easing of the requirement where the maximum MSDV is 18 and 21, representing 6% and 7% of the crew being seasick, respectively.

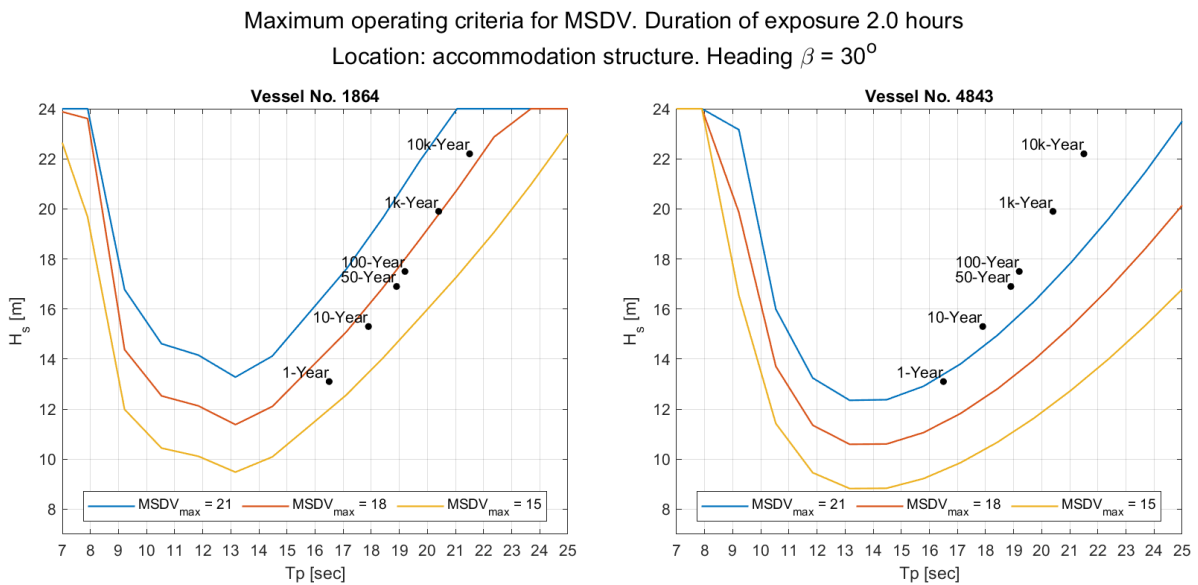


Figure 5.5.2: Criteria boundaries for different limits of MSDV for both vessels. Criteria is met when applied conditions occur beneath the criteria boundary. Load cases represent peak of the return period counter line.

The larger vessel 1864 generates lower accelerations, and thus meets the requirements in more of the load cases than vessel 4843. Although the vessels do not meet the requirement for all the load cases, one should bear in mind that only 6-7% will be seasick, representing only 4-8 persons of the onboard crew. Hence, while it is desirable to create a comfortable work-space, it is not detrimental for the current hull geometry. In terms of sensitivity to the requirements, one can observe that minimal changes in the MSDV-limit are sufficient for vessel 1864 to meet the requirements.

Conclusion

This thesis has studied the optimisation process that was aimed at generating optimal FPSO hulls. A genetic algorithm has been used as the optimisation tool to determine the optimal hull geometries based on the objectives functions that maximise the vessels' deck area and minimises their build cost. With a wide spectre of optimal vessels to choose from, further investigation was completed for two vessels. Vessel no. 1864 was chosen due to its low utility cost of US\$4,302/m² with a deck area of 16,842 m². Meanwhile, vessel 4843 is closest to the Utopian point and is deemed the vessel that performs best overall in the optimisation. To further validate the optimality of these vessels, a vessel response analysis was conducted with the conditions of their intended locations. Based on multiple criteria given by external organisations and internal procedures, the degree of compliance was evaluated.

The genetic algorithm that was used has many input parameters that were carefully chosen in order to generate the best possible results. Thus, considerable focus was given to determine the effect on the optimisation output based on the algorithm's parameters. Consequently, it was determined that many of the parameters have little effect on the results, as much of the algorithm's steps are based on randomness. However, there was a significant effect when changing the properties of the initial population. On average, using an initial population that was the results of a past optimisation generated better solutions in terms of the distance $D(X_j)$ to U_P . However, a completely random initial population did render few, but outstanding solutions. Hence, this showed that this algorithm setting might generate competitive solutions but with much less consistency. Input parameters such as the maximum stall generations, function tolerance, constraint tolerance and Pareto fractions had a negligible effect on the output but were crucial in allowing the algorithm to exit local optima. In conclusion, algorithm parameters that help speed up the optimisation and diversify the population helped the optimisation converge at much better solutions.

Both the genetic algorithm and the local hybrid search were run multiple times with a variety of input parameters to determine the best results. Evaluation of the results was completed with two measures; the normalised distance to the Utopian point given by $D(X_j)$, and the hypervolume indicator I_H . The former calculation method was used to rank individual solutions in the solution space. The closer a solution was to U_P , the better. While $D(X_j)$ evaluated individual solutions, the latter method acts as a measure of goodness for a complete Pareto front, i.e. all the non-dominated solutions on a Pareto front from a single run. Effectively, the genetic algorithm generated the results that rendered the highest hyper-area ratio, while the local hybrid search generated solutions with the lowest average distance, $\overline{D}(X_j)$.

Based on the optimised hull geometries, the optimisation model also generated three loading conditions that represented ballast condition, fully loaded and half-full condition. As a result, cargo tanks and water ballast tanks were modelled in DelftShip. By implementing the optimal hulls in the software, a verification was completed, showing the accuracy of the optimisation model. Furthermore, both vessels were evaluated in the vessel motions software ShipX. While both vessels did comply with the stability criteria and most of the operational criteria, the vessels were not compliant for all regulations regarding helicopter operation or Mean Sickness Dose Value for all return-periods. However, while compliance for all criteria for all return periods would be welcomed, it is never expected nor necessary when one considers sea-states with return-periods of even one year. When designing a vessel to withstand a wave with a return period of 10,000 years, the only criteria is the survival of the crew. Conclusively, both vessels represent feasible solutions, and the calculations generate sensible results that are fitting for a ship of this scale.

In conclusion, a successful model has been generated using a genetic algorithm to generate optimal FPSO hulls. With the given level of detail, all of the most important criteria have been met, thus rendering these hulls feasible and appropriate for further evaluation. However, likely, a higher level of detail in the model's geometry setup and an extension of the scope of what is included in the price estimation will give an even more nuanced image that is likely to generate slightly different solutions than the current ones.

6.1 Further work

The following additions presented as further work can all be implemented as additional objective functions for a more comprehensive multi-objective optimisation.

Motion-based operability

As stated in section 5.4.3, additional objective functions are likely to benefit the optimisation further, rendering the solution more realistic and feasible. Although the method of visualisation and evaluation will become much more complex, the added objectives will generate a more realistic problem and validate the solution further. Hence, an area of further development is to make the vessel's operability part of the objective functions. Currently, the motions determine whether a solution is feasible, acting as a binary termination requirement. However, if the motions are part of the optimisation, an operability profile could be generated. In conjuncture with the wave frequency table and the response amplitude operators, one could, therefore, generate a vessel that has motions of smaller magnitudes. The study could, therefore, lead to a higher degree of helicopter operations, as well as lowering the average MSDV-value of the vessel. The proposal is possible to complete with the simplified model, but for additional validity and robustness, a complete two-dimensional strip theory calculation should be completed.

Increased geometry detailing

The current parametric hull geometry has a few shortcomings and simplifications that affects the optimisation results. The bilge radius which is currently not calculated abaft should, therefore, be included in future modelling. Furthermore, the current model only allows a flat surface in the aft section with no possibility of more detailed curvature on the centre line. Hence, future models should have more variables that dictate the geometry of the aft section underwater. However, if a higher level of detailing is implemented, it would be prudent to use a vessel motions model that assumes a more advanced shape than a box to fully exploit the optimisation potential.

Weathervaning analysis

The main focus of this thesis has been FPSO units with internal turrets. Consequently, an important aspect is the vessel's station-keeping ability and weathervaning. A weathervaning analysis would consist of primarily two essential subjects. The first being an evaluation of the location of the turret that minimises the moment on the turret. Consequently, the second part would involve a propulsion study to calculate and minimise the installed machinery effect required to turn the vessel in various sea-states properly. Hence, if a weathervaning analysis should be included, it would be natural to include both of the problems as objective functions, by minimising the moment on the turret and minimising the machinery effect or cost on board.

General FPSO geometry

This thesis has only focused on FPSO units with conventional hull shapes and internal turrets. However, as an addition to the model, a more comprehensive range of hull geometries can be considered, enabling hull forms such as spread moored units or external turret moored FPSO's to be included in the optimisation output. A step further could also entail the inclusion of cylindrical designs, similar to those of *Sevan Marine's* FPSO units. The extension of the scope would widen the optimisation possibilities, but the widely different hull shapes would entail a clear distinction in calculation methods and hence many more variables need to be employed.

Logistical optimality

The scope of the optimisation was limited to the hull's optimality with no regard to any operational or logistical optimality. The hull's shape will affect the motions of the vessel, which will, in turn, decide the operability of the vessel. Consequently, an area of interest is to evaluate the entire operation, including the logistics of everything from production, offloading and delivery. Furthermore, one can optimise the operation of the vessel by dividing its lifetime into different sections, such as varying the type of operation or location over time. Hence, through the use of an evaluation process like Epoch-Era, one can determine the optimal lifetime operation of a vessel. In turn, the optimisation may focus on the break-even cost of the vessel for its different stages or eras.

Life cycle economy and environmental performance

The build cost has been the only economic consideration throughout the optimisation process. However, it would be of interest in future studies to determine the life cycle costs (LCC) of the vessels. Furthermore, the vessels' environmental impact can also be considered, and thus a life cycle assessment should be included in future works. While these studies would have been of interest, financial matters and specific prices that are accurate are hard to obtain from any company.

Bibliography

- [1] Rosshairy Abd Rahman et al. “Evolutionary Algorithm with Roulette-Tournament Selection for Solving Aquaculture Diet Formulation”. In: *Mathematical Problems in Engineering* 2016 (2016), pp. 1–10.
- [2] Rahib H. Abiyev and Mustafa Tunay. “Optimization of High-Dimensional Functions through Hypercube Evaluation”. In: *Computational Intelligence and Neuroscience* 2015 (2015), pp. 1–11.
- [3] Helideck Certification Agency. “HLL - Part C - Pitch, Roll & Heave”. In: *Helideck Landing Limitations*. HLL Issue 4. Helideck Certification Agency, Aug. 2018.
- [4] Kaveh Amouzgar. “Multi-Objective Optimization using Genetic Algorithms”. PhD thesis. Jönköping: Tekniska Högskolan, Högskolan i Jönköping, May 30, 2010. 72 pp.
- [5] Fahmy Ardiansyah, Aries Sulisetyono, and Wasis Dwi Aryawan. “Heading Analysis of Weathervaning Turret Moored Units”. In: *Applied Mechanics and Materials* 874 (Jan. 2018), pp. 64–70.
- [6] Anne Auger et al. “Theory of the hypervolume indicator: optimal my-distributions and the choice of the reference point”. In: *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms - FOGA '09*. the tenth ACM SIGEVO workshop. Orlando, Florida, USA: ACM Press, 2009, p. 87.
- [7] Crhistopher M. Barton. “FPSO market set for revival”. In: *Offshore Magazine* (Aug. 22, 2019).
- [8] Crhistopher M. Barton. “FPSO market still faces challenges”. In: *Offshore Magazine* 77.8 (Aug. 2017), pp. 58–59.
- [9] Bluewater. *What is an FPSO? – Bluewater*. Library Catalog: www.bluewater.com. URL: <https://www.bluewater.com/fleet-operations/what-is-an-fps/> (visited on 05/15/2020).
- [10] Omid Bozorg-Haddad, Mohammad Solgi, and Hugo A. Loáiciga. “Meta-heuristic and Evolutionary Algorithms for Engineering Optimization”. In: vol. 194. Wiley-Blackwell, Nov. 3, 2017, pp. 53, 67.
- [11] Karl Bringmann and Tobias Friedrich. “Approximation quality of the hypervolume indicator”. In: *Artificial Intelligence* 195 (Feb. 2013), pp. 265–290.

-
- [12] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Paperback edition. OCLC: 269246536. Chichester: Wiley, 2008. 515 pp.
- [13] FLNG FEED Department. *HHI New-building Conversion FPSO HULL*. Feb. 2019.
- [14] Felipe P. Espinoza, Barbara S. Minsker, and David E. Goldberg. “Adaptive Hybrid Genetic Algorithm for Groundwater Remediation Design”. In: *Journal of Water Resources Planning and Management* 131.1 (Jan. 2005), pp. 14–24.
- [15] O. M. Faltinsen. *Sea loads on ships and offshore structures*. Paperback ed., reprinted. Cambridge ocean technology series. OCLC: 247731157. Cambridge: Cambridge Univ. Press, 1998. 328 pp.
- [16] Dariusz Eirik Fathi and Jan Roger Hoff. *ShipX Vessel Responses (VERES) Theory Manual*. Oct. 2017.
- [17] DNV GL. “DNVGL-OS-C102 Chapter 2 Section 2 Design Principles”. In: *Offshore Standard, Structural design of offshore ships (DNVGL-OS-C102)*. July 2015. DNV GL, p. 21.
- [18] DNV GL. “Part 3 Hull Chapter 2 General arrangement design”. In: *DNV GL Rules for classification of ships*. January 2018. DNV GL, p. 14.
- [19] Eklavya Gupte and Charlotte Bucchioni. “Oil on floating storage soars to record highs, but peak still some way off”. In: *S&P Global Platts* (May 14, 2020).
- [20] Arnulf Hagen and Stein Ove Erikstad. *Shipbuilding*. Published: Educational compendium. Nov. 2014.
- [21] Randy L. Haupt and S. E. Haupt. *Practical genetic algorithms*. 2nd ed. Hoboken, N.J: John Wiley, 2004. 253 pp.
- [22] John H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press, 1975. 183 pp.
- [23] Huayang Xie and Mengjie Zhang. “Parent Selection Pressure Auto-Tuning for Tournament Selection in Genetic Programming”. In: *IEEE Transactions on Evolutionary Computation* 17.1 (2013), pp. 1–19.
- [24] Jørgen Juncher Jensen, Alaa E. Mansour, and Anders Smærup Olsen. “Estimation of ship motions using closed-form expressions”. In: *Ocean Engineering* 31.1 (Jan. 2004), pp. 61–85.
- [25] Jae-Han Kim and Yonghwan Kim. “Time-domain analysis of passenger comfort on cruise ships under motion responses in waves”. In: *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 228.4 (Nov. 2014), pp. 331–347.
- [26] Abdullah Konak, David W. Coit, and Alice E. Smith. “Multi-objective optimization using genetic algorithms: A tutorial”. In: *Reliability Engineering & System Safety* 91.9 (Sept. 2006), pp. 992–1007.
- [27] Jaakko Lappi. *DELTAMARIN: Family of parametric FPSO - Combining cost-efficiency with sustainability*.

- [28] Grzegorz Mazerski. “Optimization of FPSO’s Main Dimensions Using Genetic Algorithm”. In: *Volume 1: Offshore Technology*. American Society of Mechanical Engineers, July 2012.
- [29] *Meeting discussing optimisation based on vessel motions*. In collab. with Marilena Professor Greco. Feb. 13, 2020.
- [30] *Meeting with Øyvind Røyseth at Teekay Offshore*. In collab. with Øyvind Røyseth. Oct. 3, 2020.
- [31] Huixing Meng, Leïla Kloul, and Antoine Rauzy. “Production availability analysis of Floating Production Storage and Offloading (FPSO) systems”. In: *Applied Ocean Research* 74 (May 2018), pp. 117–126.
- [32] *Metocean report for West of Shetlands*. Publisher not disclosed. PhysE.
- [33] MODEC. *About an FPSO, Floating Production Solutions*. URL: https://www.modec.com/fps/fps_o_fso/about/index.html (visited on 04/29/2020).
- [34] SN/K 114/EG S NORSOK. *Tehcnical safety. Edition 5, June 2018*. June 20, 2018.
- [35] SBM Offshore. *Fast4Ward Potential - Cost and schedule*. Nov. 18, 2016.
- [36] Teekay Offshore. *Kick-off Next Generation Harsh Weather Hull*. Sept. 10, 2019.
- [37] Teekay Offshore. *Procedure for shuttle tanker operations*. Operations Manual Section 1. Aug. 25, 2018.
- [38] *Phone interview with ship broker Petter Sissener at Lorentzen & Stomoco*. In collab. with Petter Sissener. May 15, 2020.
- [39] Nicholas J. Radcliff. “Forma Analysis and Random Respectful Recombination”. In: *Proceedings of the Fourth International Conference on Genetic Algorithms*. International Conference on Genetic Algorithms. 1991.
- [40] *Resolution MSC.267(85) Adoption of the International Code on Intact Stability*. International Maritime Organisation (IMO). Apr. 12, 2008, pp. 12–17.
- [41] Øyvind Røyseth. *Oppdate on master thesis and sensitivity analysis*. Apr. 16, 2020.
- [42] Artur M. Schweidtmann et al. “Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives”. In: *Chemical Engineering Journal* 352 (Nov. 2018), pp. 277–282.
- [43] *Sea web Maritime Portal*. Published: Online database. Nov. 2019. URL: https://maritime.ihs.com/Areas/Seaweb/authenticated/authenticated_handler.aspx?control=shipsearch&bp=1.
- [44] Akhter Shamin et al. “Self-organizing GA for crop model parameter estimation using multi-resolution satellite images”. In: *International Journal of Geoinformatics* 6 (Dec. 2010), pp. 29–40.
- [45] Uyar Sima, Eryigit Gulsen, and Sariel Sanem. “An Adaptive Mutation Scheme in Genetic Algorithms for Feastening the Convergence to the Optimum”. In: *Proceedings of the 3rd Asia Pacific International Symposium on Information Technology*. Asia Pacific International Symposium on Information Technology. Istanbul, Turkey: Istanbul Tehcnical University, Jan. 2004.

-
- [46] M. Srinivas and L.M. Patnaik. “Adaptive probabilities of crossover and mutation in genetic algorithms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.4 (Apr. 1994), pp. 656–667.
- [47] Lorentzen \& Stemoco. *L&S FIRST TAKE: Oil Price Crashed, Effect on Shipping Markets & Oil Service*. Jan. 4, 2020.
- [48] Lorentzen \& Stemoco. *Weekly Wrap : L&S Comments Tanker and Dry Bulk Market*. Apr. 24, 2020.
- [49] Committee on Tank Vessel Design. *Tanker Spills: Prevention by Design*. Pages: 1621. Washington, D.C.: National Academies Press, Jan. 1, 1991. 226-229 (appendix E).

APPENDIX

Appendix overview

App.	Title	Description
A	Wave Frequency Table	Joint frequency of significant wave height and wave peak period. Representative data for location \mathcal{X} .
B	Vessel no. 1864	Tank arrangement and corresponding fill levels. Line drawing of vessel no. 1864.
C	Vessel no. 4843	Tank arrangement and corresponding fill levels. Line drawing of vessel no. 4843.
D	Response Amplitude Operators	Comparison of ShipX and simplified model RAO's. Headings from 0-30° for heave, roll and pitch.
E	Simplified Vessel Motion Model	Summarised description of simplified vessel motions model by Jensen et. al. Note that not all deductions are included in the appendix summary.
F	MATLAB Scripts	MATLAB scripts describing the setup of the GA and local search. <i>getCost.m</i> represents the objective function for build cost.

Appendix A

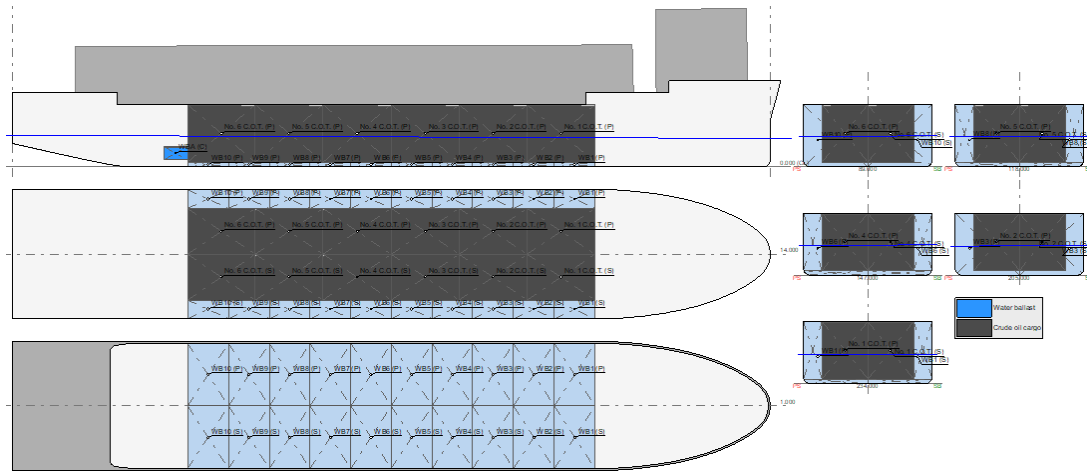
Wave frequency table

Hs (m)	Lower bound	Upper bound	Tp (s)																						Cumulative sum
			4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22				
18.5																							0		
16.0																							0		
16.5																							0		
15.5																							0		
15.0																							0		
14.5																							0		
14.0																							0		
13.5																							0		
13.0																							0		
13.5																							0		
13.0																							0		
12.5																							0		
12.0																							0		
11.5																							0		
11.0																							0		
10.5																							0		
10.0																							0		
9.5																							0		
9.0																							0		
8.5																							0		
8.0																							0		
7.5																							0		
7.0																							0		
6.5																							0		
6.0																							0		
5.5																							0		
5.0																							0		
4.5																							0		
4.0																							0		
3.5																							0		
3.0																							0		
2.5																							0		
2.0																							0		
1.5																							0		
1.0																							0		
0.5																							0		
0.0																							0		
83	906	3576	6173	5317	10065	8099	7887	6103	3808	2503	1252	717	159	75	20	4	4	14	61351						
61,351	61,268	60,362	56,786	50,613	41,296	31,231	22,532	14,645	8,542	4,734	2,231	979	262	103	28	8	4	14	61,351						

Appendix B

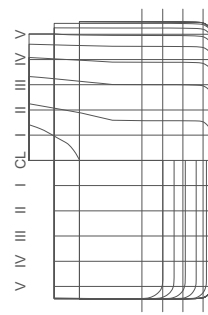
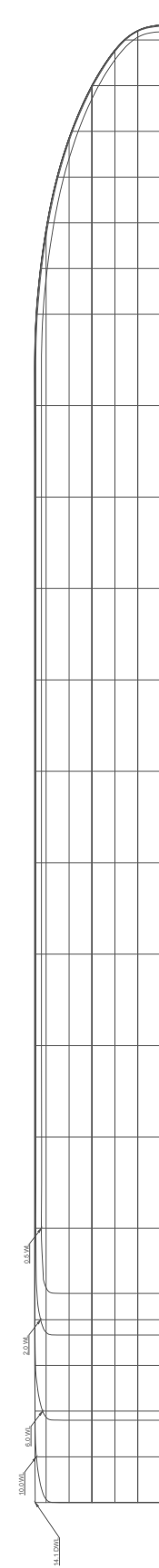
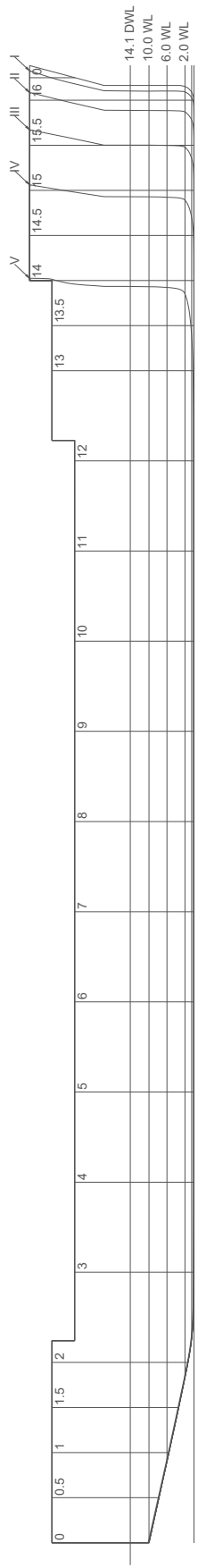
Vessel no. 1864


B.1 Tank arrangement for fully laden condition



Crude oil cargo	Density [t/m ³]	Fill [%]	Weight [MT]	LCG [m]	TCG [m]	VCG [m]	FSM [t*m]
No. 1 C.O.T. (P)	0.87	97.7	11,528.6	234.032	9.742 (PS)	13.983	15,220
No. 2 C.O.T. (P)	0.87	97.7	11,528.6	205.075	9.742 (PS)	13.983	15,220
No. 3 C.O.T. (P)	0.87	97.7	11,528.6	176.118	9.742 (PS)	13.983	15,220
No. 4 C.O.T. (P)	0.87	97.7	11,549.5	147.135	9.742 (PS)	13.983	15,247
No. 5 C.O.T. (P)	0.87	97.7	11,507.7	118.178	9.742 (PS)	13.983	15,192
No. 6 C.O.T. (P)	0.87	97.7	11,516.3	89.263	9.742 (PS)	13.983	15,203
No. 1 C.O.T. (S)	0.87	97.7	11,528.6	234.032	-9.742 (SB)	13.983	15,220
No. 2 C.O.T. (S)	0.87	97.7	11,528.6	205.075	-9.742 (SB)	13.983	15,220
No. 3 C.O.T. (S)	0.87	97.7	11,528.6	176.118	-9.742 (SB)	13.983	15,220
No. 4 C.O.T. (S)	0.87	97.7	11,549.5	147.135	-9.742 (SB)	13.983	15,247
No. 5 C.O.T. (S)	0.87	97.7	11,507.7	118.178	-9.742 (SB)	13.983	15,192
No. 6 C.O.T. (S)	0.87	97.7	11,516.3	89.263	-9.742 (SB)	13.983	15,203
Total			138,318.7	161.655	0.000 (CL)	13.983	182,602

B.2 Lines plan

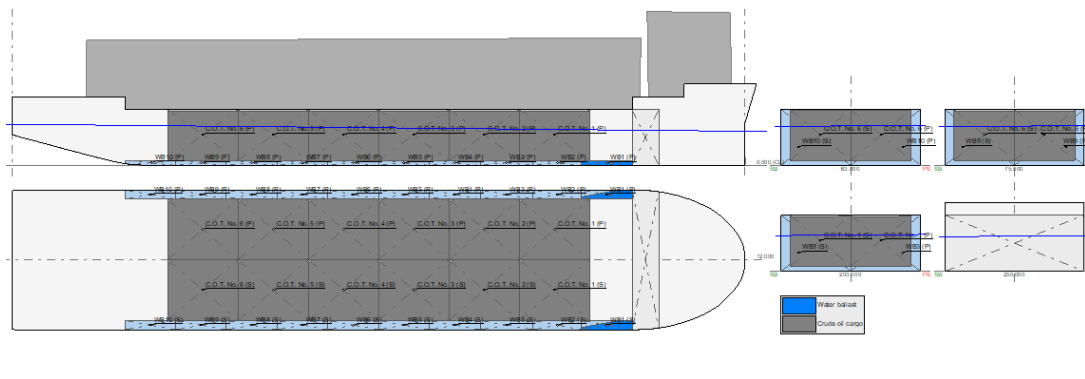


	NAME	FPSO VESSEL NO. 1864		
	TITLE	LINES PLAN OF VESSEL NO. 1864		
DRAWN BY	DWG NO.	SIZE	REV.	
B. BØRRESEN	001	A3	1	
DATE ISSUED	SCALE	SHEET		
10-JUNE 2020	1:1500	1 OF 1		

Appendix C

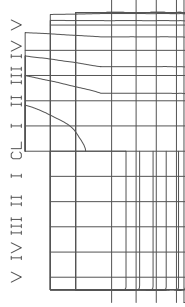
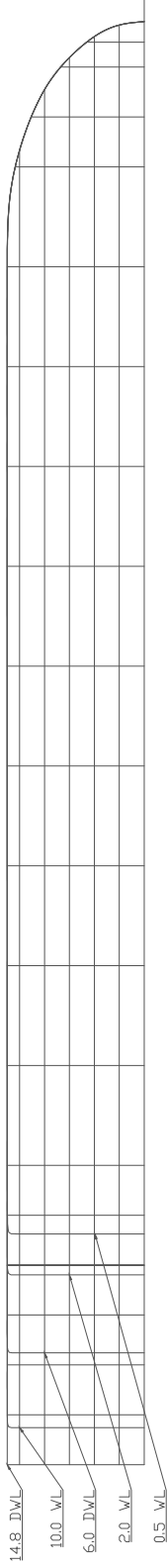
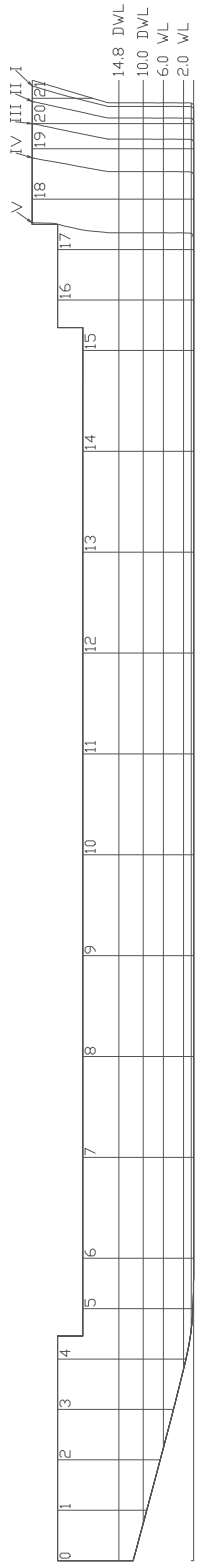
Vessel no. 4843


C.1 Tank arrangement for fully laden condition



Crude oil cargo	Density [t/m ³]	Fill [%]	Weight [MT]	LCG [m]	TCG [m]	VCG [m]	FSM [t*m]
No. 1 C.O.T. (P)	0.87	97	11,526.7	213.871	11.946 (PS)	11.700	28,264
No. 2 C.O.T. (P)	0.87	97	11,526.7	213.871	-11.946 (SB)	11.700	28,264
No. 3 C.O.T. (P)	0.87	97	11,526.3	186.145	11.946 (PS)	11.700	28,263
No. 4 C.O.T. (P)	0.87	97	11,526.3	186.145	-11.946 (SB)	11.700	28,263
No. 5 C.O.T. (P)	0.87	97	11,526.7	158.419	11.946 (PS)	11.700	28,264
No. 6 C.O.T. (P)	0.87	97	11,526.7	158.419	-11.946 (SB)	11.700	28,264
No. 1 C.O.T. (S)	0.87	97	11,528.6	130.689	11.946 (PS)	11.700	28,269
No. 2 C.O.T. (S)	0.87	97	11,528.6	130.689	-11.946 (SB)	11.700	28,269
No. 3 C.O.T. (S)	0.87	97	11,524.4	102.963	11.946 (PS)	11.700	28,257
No. 4 C.O.T. (S)	0.87	97	11,524.4	102.963	-11.946 (SB)	11.700	28,257
No. 5 C.O.T. (S)	0.87	97	11,526.7	75.239	11.946 (PS)	11.700	28,264
No. 6 C.O.T. (S)	0.87	97	11,526.7	75.239	-11.946 (SB)	11.700	28,264
Total			138,318.7	144.555	0.000 (CL)	11.700	339,159

C.2 Lines plan



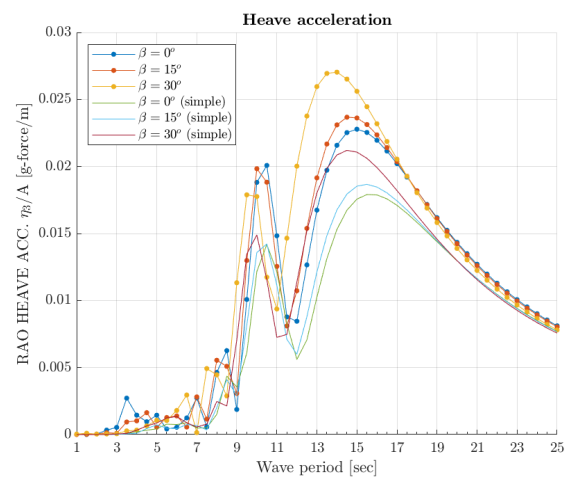
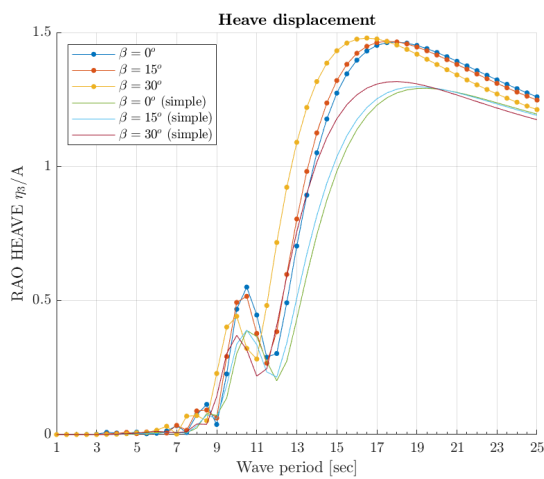
	NAME		FPSD VESSEL NO. 4843	
	TITLE		LINES PLAN DF VESSEL NO. 4843	
DRAWN BY B. BØRRESEN	DWG NO. 001	SIZE A3	REV 1	
DATE ISSUED 10-JUNE 2020	SCALE 1:1500	SHEET 1	DF 1	

Appendix D

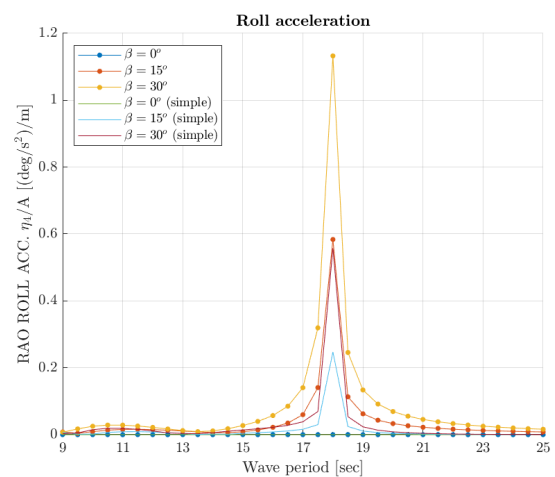
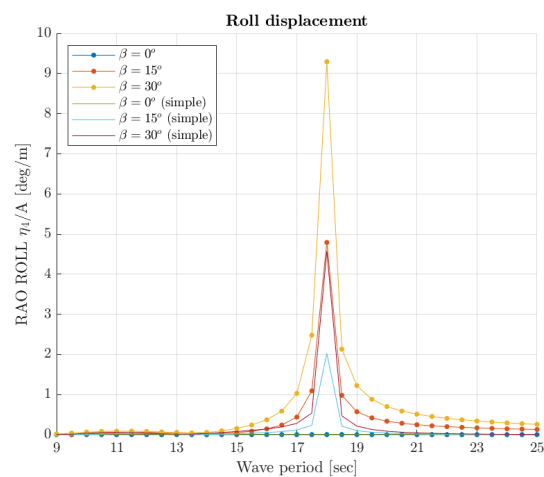
Response amplitude operators

D.1 Vessel no. 1864

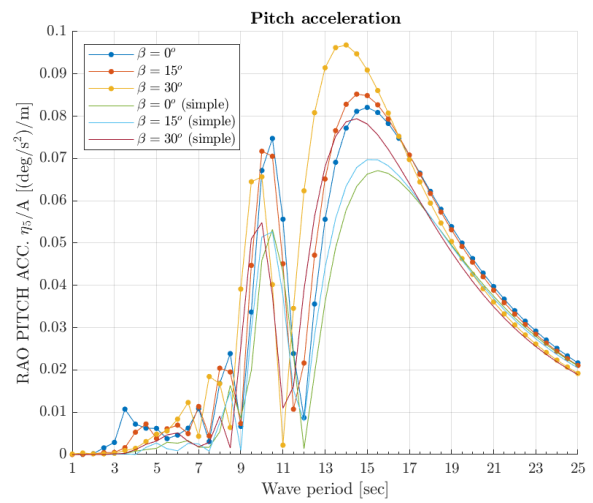
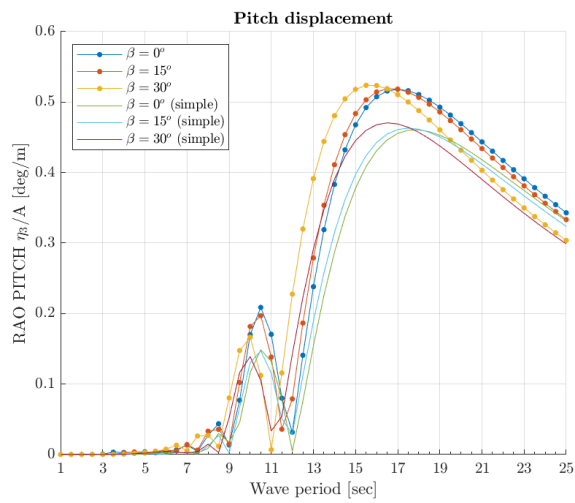
D.1.1 Heave motion



D.1.2 Roll motion

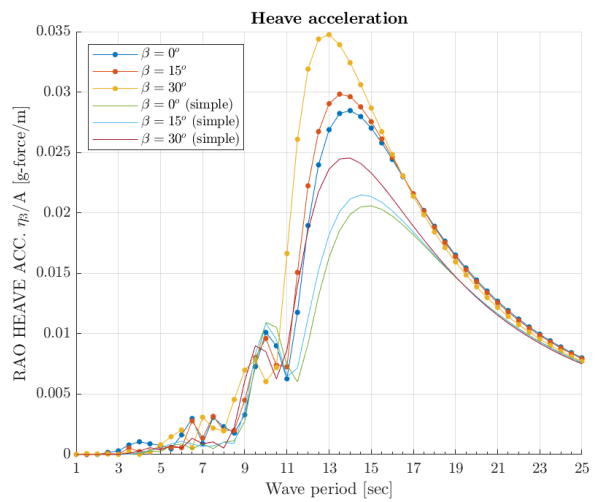
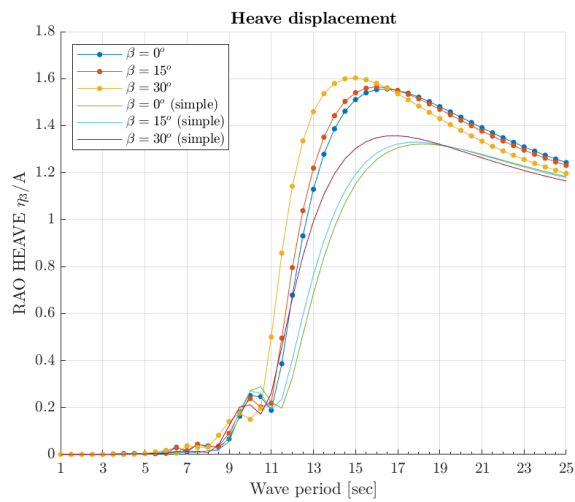


D.1.3 Pitch motion

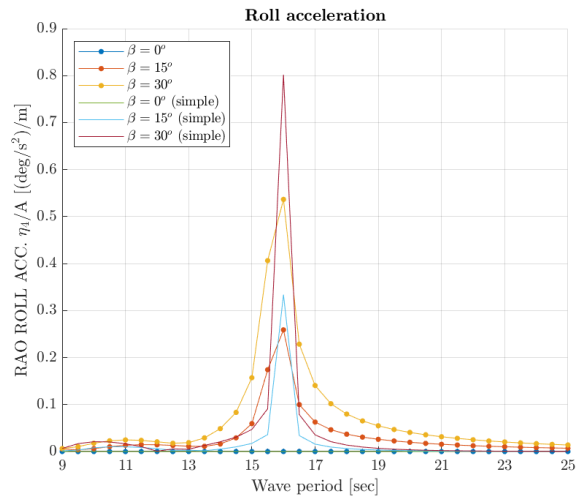
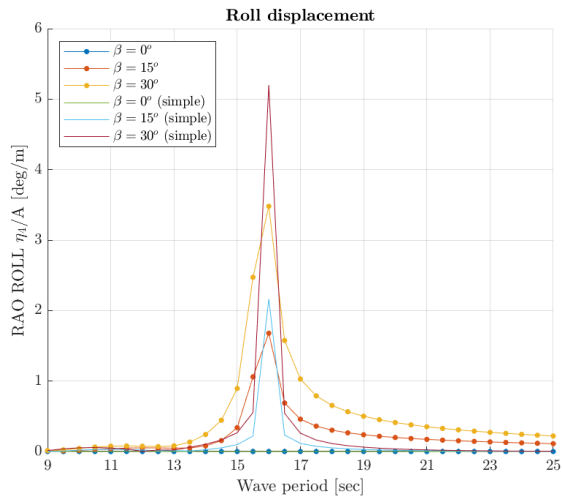


D.2 Vessel no. 4843

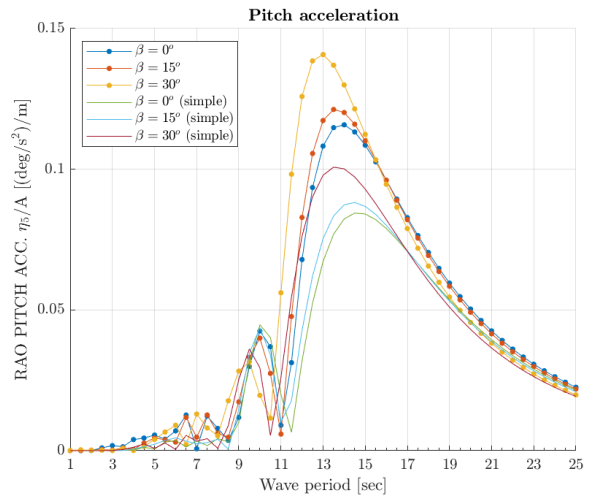
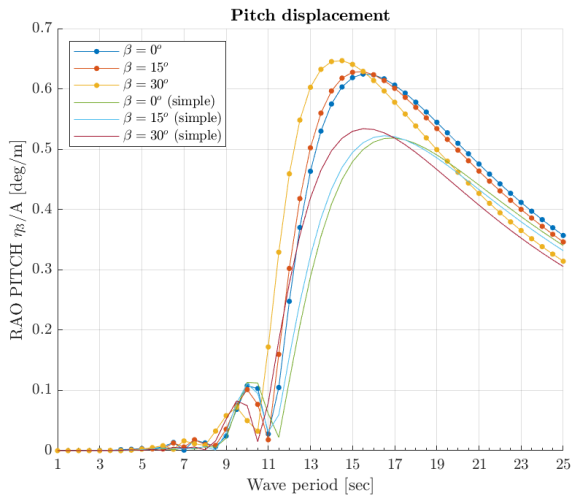
D.2.1 Heave motion



D.2.2 Roll motion



D.2.3 Pitch motion



Simplified vessel motion calculation

The following explanation of the simplified sea-keeping model derives entirely from the article Jensen et al. 'Estimation of ship motions using closed-form expressions [24]. The presented formulae in this appendix are strictly shortened.

The encounter frequency ϖ is given by

$$\varpi = \omega - kV \cos \beta \equiv \alpha\omega, \quad (\text{E.0.1})$$

where V is the forward speed of the vessel and β is the heading angle. α is defined as:

$$\alpha = 1 - Fn\sqrt{kL} \cos \beta, \quad (\text{E.0.2})$$

where the Froude number of the vessel is given by $Fn = V/\sqrt{gL}$. The sectional hydrodynamic damping is modelled by the dimensionless ratio between the incoming and the diffracted wave amplitude through the following approximation:

$$A = 2 \sin\left(\frac{\varpi^2 B}{2g}\right) \exp\left(-\frac{\varpi^2 T}{g}\right). \quad (\text{E.0.3})$$

The forcing functions F and G are then given by;

$$F = \kappa f \frac{2}{k_e} \sin\left(\frac{k_e L}{2}\right) \quad (\text{E.0.4})$$

$$G = \kappa f \frac{24}{(k_e L)^2 L} \left[\sin\left(\frac{k_e L}{2}\right) - \frac{k_e L}{2} \cos\left(\frac{k_e L}{2}\right) \right], \quad (\text{E.0.5})$$

where k_e is the effective wave number, and f are given as;

$$k_e = |k \cos \beta| \quad (\text{E.0.6})$$

$$f = \sqrt{(1 - kT)^2 + \left(\frac{A^2}{kB\alpha^3}\right)^2}. \quad (\text{E.0.7})$$

The Smith correction factor κ is then approximated by

$$\kappa = \exp(-k_e T). \quad (\text{E.0.8})$$

In the simplified model for roll, the ship is assumed to consist of two prismatic beams with the same draught T , but different breadths B_0 and B_1 and cross-sectional areas A_0 and A_1 . The ratio γ between the breadth of the two beams, B_0 and B_1 is found such that the waterplane area coefficient C_{WP} is the same for the real ship and the simplified model of the ship.

$$\begin{aligned} C_{WP} &= \frac{B_0 L (\delta + \gamma (1 - \delta))}{L B_0} \\ \gamma &\equiv \frac{B_1}{B_0} = \frac{C_{WP} - \delta}{1 - \delta} \\ A_0 &= \frac{C_B B_0 T}{\delta + \gamma (1 - \delta)} \\ A_1 &= \gamma A_0 \end{aligned} \tag{E.0.9}$$

The sectional damping coefficient b_{44} is approximated through numerical methods and ultimately renders:

$$\frac{b_{44}}{\rho A B^2} = \sqrt{\frac{B}{2g}} = a(B/T) \exp(b(B/T) \varpi^{-1.3}) \varpi^{d(B/T)}, \tag{E.0.10}$$

where A is the cross-sectional area of the submerged part of the section. The three functions a , b and c are assumed to be linear in B/T . The total hydrodynamic damping coefficient B_{44} is then simply taken as:

$$B_{44} = L b_{44} [\delta + \kappa * 2 (1 - \delta)], \tag{E.0.11}$$

where $\kappa^2 = b_{44,1}/b_{44,0}$, i.e. the ration of the sectional damping for the two beam elements. The viscous roll damping is approximately accounted for by adding a percentage of the critical damping to the inviscid damping. The critical damping is defined as $B_{44}^* = \frac{C_{44} T_N}{\pi}$. Hence, the total damping becomes $B_{44}^{tot} = B_{44} + \mu B_{44}^*$.

The sectional excitation moment m can be expressed in terms of the sectional hydrodynamic damping. The excitation moment for a two-dimensional section in the $y - z$ -plane is given by:

$$m(x) = i \omega \rho e^{i \omega t} \int_{-\infty}^0 \left[\varphi_0 \frac{\partial \varphi_4}{\partial y} - \varphi_4 \frac{\partial \varphi_0}{\partial y} \right]_{y=-\infty}^{y=\infty} dz, \tag{E.0.12}$$

where φ_0 is the potential for the incoming waves with united wave amplitude. Carrying out the integration gives for the sectional excitation moment:

$$m(x) = \sin \beta \sqrt{\frac{\rho g^2}{\omega}} b_{44} e^{-i x k \cos \beta} e^{i \omega t}. \tag{E.0.13}$$

Taking the real part of the moment gives the amplitude $|M|$ for the moment that is used in equations (2.2.8) (2.2.9):

$$\begin{aligned} Re(M) &= |M| \cos(\omega t + \epsilon) = |\sin \beta| \sqrt{\frac{\rho g^2}{\omega}} \frac{2}{k_e} \\ &\times \sqrt{b_{44,0}} \{ \sin^2(0.5 \delta L k_e) + \kappa^2 \sin^2(0.5 (1 - \delta) L k_e) \\ &2 \kappa \sin(0.5 \delta L k_e) \sin(0.5 (1 - \delta) L k_e) \cos(0.5 L k_e) \}^{\frac{1}{2}} \cos(\omega t + \epsilon) \end{aligned} \tag{E.0.14}$$

Appendix F

MATLAB scripts

F.1 longOptimisation.m

```
1 tic;
2
3 getFPSOdata;
4 time = datestr(clock, 'dd/mm/YY');
5 popSize = 500;
6 repeats = 1;
7
8 load('resulting_data.mat')
9 weight_data = load('weight_data.mat');
10 steel_data = load('steel_data.mat');
11 deadweight_data = load('deadweight_data.mat');
12
13 fid = fopen('sensitivityEval3.txt', 'a');
14 dt = 0.5:0.01:1.0;
15
16 for k = 1:length(dt)
17     for u = 1:length(popSize)
18         % ----- GA -----
19         tic;
20         gaSolver;
21         timeGA = toc;
22         for m = 1:size(x,1)
23             fprintf(fid, '%s, %.0f, %f, %f, %f, %f, %f, %f, %f, %f, %f, ', ...
24                 '%f, %f, %f, %.2f, %.0f, %f, %f, %.1f, %.0f, %.0f, %f\n', ...
25                 time, 1, x(m,1), x(m,2), x(m,3), x(m,4), x(m,5), x(m,6), x(m,7), ...
26                 x(m,8), x(m,9), x(m,10), x(m,11), x(m,12), FvalGA(m,1), FvalGA(m,2), ...
27                 Output.averagedistance, Output.spread, ParetoFraction, ...
28                 initialValue, numberOfVariables*popSize(u), dt(k));
29         end
30     end
31
32     load('gaMatrix.mat')
33     if diffAreaGA > gaMatrix(1,1) && diffCostGA > gaMatrix(1,2)
34         gaMatrix(1,1) = diffAreaGA;
35         gaMatrix(1,2) = diffCostGA;
36         save('gaMatrix.mat', 'gaMatrix');
37         fprintf('\n<NEW DATA ADDED IN GA MATRIX>\n');
38     end
39
40     %----- HYBRID -----
41     hybridSolver;
```



```

42 timeHY = toc;
43 for m = 1:size(x,1)
44     fprintf(fid, '%s, %.0f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, ...
45             '%f, %f, %.2f, %.0f, %f, %f, %.1f, %.0f, %.0f, %f\n', ...
46             time, 2, x(m,1), x(m,2), x(m,3), x(m,4), x(m,5), x(m,6), x(m,7), ...
47             x(m,8), x(m,9), x(m,10), x(m,11), x(m,12), FvalHY(m,1), FvalHY(m,2), ...
48             Output.averagedistance, Output.spread, ParetoFraction, ...
49             initialValue, popSize(u), timeHY);
50 end
51
52 load('hybridMatrix.mat')
53 if diffAreaHY > hybridMatrix(1,1) && diffCostHY > hybridMatrix(1,2)
54     hybridMatrix(1,1) = diffAreaHY;
55     hybridMatrix(1,2) = diffCostHY;
56     save('hybridMatrix.mat', 'hybridMatrix');
57     fprintf('\n<NEW DATA ADDED IN HYBRID MATRIX>\n');
58 end
59
60 %----- GA 2 -----
61 options_MOGA = optimoptions(options_MOGA, 'HybridFcn', []);
62 options_MOGA = optimoptions(options_MOGA, 'InitialPopulationMatrix', ...
63     x, 'InitialScoresMatrix', Score);
64 options_MOGA = optimoptions(options_MOGA, 'Display', 'iter');
65 [x, FvalGA, exitFlag, Output, Population, Score] = gamultiobj(FitFcn, ...
66     numberOfVariables, A, b, Aeq, beq, lb, ub, nonclon, options_MOGA);
67
68 timeHG = toc;
69 for m = 1:size(x,1)
70     fprintf(fid, '%s, %.0f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, ...
71             '%f, %f, %.2f, %.0f, %f, %f, %.1f, %.0f, %.0f %f\n', ...
72             time, 3, x(m,1), x(m,2), x(m,3), x(m,4), x(m,5), x(m,6), x(m,7), ...
73             x(m,8), x(m,9), x(m,10), x(m,11), x(m,12), FvalGA(m,1), FvalGA(m,2), ...
74             Output.averagedistance, Output.spread, ParetoFraction, ...
75             initialValue, popSize(u), timeHG);
76 end
77
78 fprintf('\n----- HGA OPTIMISATION OUTPUT-----\n');
79 fprintf('The number of points on the Pareto front was:           ', ...
80     '%d\n', size(x,1));
81 fprintf('The number of generations was:                         ', ...
82     '%d\n', Output.generations);
83 fprintf('The average distance measure of the solutions on the Pareto', ...
84     'front was: %g\n', Output.averagedistance);
85 fprintf('The spread measure of the Pareto front was:           ', ...
86     '%g\n', Output.spread);
87
88 fprintf('\nProcess is at %.2f%% \n', (k*u)/(repeats*length(popSize))*100);
89 fprintf('\nPopulation size: %.0f\n', popSize(u))
90 fprintf('Repeat: %.0f\n', k);
91 end
92
93 resulting_data = zeros(size(x,1), numberOfVariables+2);
94 resulting_data(:,1:numberOfVariables) = x;
95 resulting_data(:,13:14) = FvalGA;
96 save('resulting_data.mat', 'resulting_data');
97
98 fclose(fid);
99
100 elapsed_time = toc;
101
102 fprintf('\nELAPSED TIME: ');
103 fprintf(datestr(datum(0,0,0,0,0,elapsed_time), 'HH:MM:SS '));
104 fprintf(' [hh:mm:ss]\n');

```

F.2 gaSolver.m

```

1 % Non-linear constraints function input
2 nonclon = @conditions;
3
4 % Upper and lower bounds
5 % [1 2 3 4 5 6 7 8 9 10 11 12]
6 lb = [20 40 30 15 10 0.5 100 40 10 100 20 20]*dt(k);
7 ub = [45 60 70 40 25 4 200 80 40 180 60 40]*(1+(1-dt(k)));
8
9 % Number of variables in optimisation
10 numberOfVariables = length(lb);
11
12 initialValue = 1;
13 initialSize = 200;
14
15 if initialValue == 1
16     x0LB = 1;
17     x0UB = size(resulting_data,1);
18     row1 = randi([x0LB x0UB],1,initialSize);
19     x0 = resulting_data(row1,1:12);
20 else
21     for j = 1:initialSize
22         for i = 1:numberOfVariables
23             x0(j,i) = (ub(i)-lb(i))*rand(1,1) + lb(i);
24         end %for
25     end %for
26 end % if
27
28 % linear constraints
29 A = [];
30 b = [];
31 Aeq = [];
32 beq = [];
33
34 % Call for the fitness function
35 FitFcn = @objectiveFunction;
36
37 % Option variables
38 FunctionTolerance = 1e-3;
39 ConstraintTolerance = 1e-4;
40 MaxStallGenerations = 200;
41 ParetoFraction = 0.5;
42 MaxGenerations = 20;
43 tournamentSize = round(0.2*length(x0),0);
44 PopulationSize = 500;
45
46 % Multi-objective optimisation options
47 options_MOGA = optimoptions(@gamultiobj,'PlotFcn',{@gaplotpareto,...
48     @gaplotscorediversity});
49 options_MOGA.DistanceMeasureFcn = {@distancecrowding,'genotype'};
50 %options_MOGA = optimoptions(options_MOGA,'Display','iter');
51 options_MOGA = optimoptions(options_MOGA,'ParetoFraction',ParetoFraction);
52 options_MOGA = optimoptions(options_MOGA,'FunctionTolerance',...
53     FunctionTolerance);
54 options_MOGA = optimoptions(options_MOGA,'MaxStallGenerations',...
55     MaxStallGenerations);
56 options_MOGA = optimoptions(options_MOGA,'MaxGenerations',MaxGenerations);
57 options_MOGA = optimoptions(options_MOGA,'ConstraintTolerance',...
58     ConstraintTolerance);
59 options_MOGA = optimoptions(options_MOGA,'InitialPopulationMatrix',x0);

```

```

60 options_MOGA = optimoptions(options_MOGA, 'PopulationSize', PopulationSize);
61 %%
62 % Runs the optimisation
63 [x, FvalGA, exitFlag, Output, Population, Score] = gamultiobj(FitFcn, ...
64     numberOfVariables, A, b, Aeq, beq, lb, ub, nonclon, options_MOGA);
65
66 %%
67 diffAreaGA = abs(-min(FvalGA(:,1))+max(FvalGA(:,1)))/((-min(FvalGA(:,1)), ...
68     -max(FvalGA(:,1)))/2)*100;
69 diffCostGA = abs(min(FvalGA(:,2))-max(FvalGA(:,2)))/((max(FvalGA(:,2))+ ...
70     min(FvalGA(:,2)))/2)*100;
71
72 fprintf('\n----- GA OPTIMISATION OUTPUT-----', ...
73     '-----\n');
74 fprintf('The number of points on the Pareto front was:           ', ...
75     '%d\n', size(x,1));
76 fprintf('The number of generations was:                         ', ...
77     '%d\n', Output.generations);
78 fprintf('The average distance measure of the solutions on the Pareto', ...
79     'front was: %g\n', Output.averagedistance);
80 fprintf('The spread measure of the Pareto front was:           ', ...
81     '%g\n', Output.spread);
82
83 fprintf('\n----- OBJECTIVE FUNCTION OUTPUT -----\n')
84 fprintf('Max deck area:  %0.f [m^2]           Max CAPEX:  USD %.2f mill.\n', ...
85     -min(FvalGA(:,1)), max(FvalGA(:,2))*1e-6)
86 fprintf('Min deck area:  %.0f [m^2]           Min CAPEX:  USD %.2f mill.\n', ...
87     -max(FvalGA(:,1)), min(FvalGA(:,2))*1e-6)
88 fprintf('Difference:      %.3f %%           Difference: %.3f %%\n', ...
89     diffAreaGA, diffCostGA);

```

F.3 hybridSolver.m

```

1
2 %% Hybrid solution
3 %% Runs the optimisation again with hybrid function from the pareto front
4
5 options_MOGA = optimoptions(options_MOGA, 'HybridFcn', @fgoalattain);
6 %options_MOGA = optimoptions(options_MOGA, 'Display', 'iter');
7 %options_MOGA = optimoptions('fgoalattain', 'Display', 'iter');
8
9 % Reset the random state (to compare with previous run)
10 strm = RandStream.getGlobalStream;
11 strm.State = Output.rngstate.State;
12
13 [x, FvalHY, exitFlag, Output, Population, Score] = gamultiobj(FitFcn, ...
14     numberOfVariables, A, b, Aeq, beq, lb, ub, nonclon, options_MOGA);
15
16 %%
17 diffAreaHY = abs(-min(FvalHY(:,1))+max(FvalHY(:,1)))/, ...
18     ((-min(FvalHY(:,1))-max(FvalHY(:,1)))/2)*100;
19 diffCostHY = abs(min(FvalHY(:,2))-max(FvalHY(:,2)))/, ...
20     ((max(FvalHY(:,2))+min(FvalHY(:,2)))/2)*100;
21
22 fprintf('\n----- HYBRID OPTIMISATION OUTPUT-----\n');
23 fprintf('The number of points on the Pareto front is:           ', ...
24     %d\n', size(x,1));
25 fprintf('The number of generations was:                           ', ...
26     %d\n', Output.generations);
27 fprintf('The average distance measure of the solutions on the Pareto', ...
28     'front was: %g\n', Output.averagedistance);
29 fprintf('The spread measure of the Pareto front was:                 ', ...
30     '%g\n', Output.spread);
31
32 fprintf('\n----- OBJECTIVE FUNCTION OUTPUT -----\n')
33 fprintf('Max deck area:  %0.f [m^2]           Max CAPEX:  USD %.2f mill.\n', ...
34     -min(FvalHY(:,1)), max(FvalHY(:,2))*1e-6)
35 fprintf('Min deck area:  %0.f [m^2]           Min CAPEX:  USD %.2f mill.\n', ...
36     -max(FvalHY(:,1)), min(FvalHY(:,2))*1e-6)
37 fprintf('Difference:    %.3f %%             Difference:  %.3f %%\n', ...
38     diffAreaHY, diffCostHY);

```

F.4 getCost.m

```

1 function totalCost = getCost(x)
2
3 % -----
4 % ----- CONSTANTS -----
5 % -----
6
7 % Natural constants
8 seawater_density = 1.025;
9 required_bbl_volume = 1e6;
10 barrel_fluid = 158.987;
11 crude_oil_density = 0.87;
12
13 % -----
14 % ----- VOLUME | SURFACE | AREA -----
15 % -----
16
17 % Midship
18 midship_area = x(4)*x(2) - 0.5*pi*x(6)^2;
19 midship_deck_area = x(2)*x(7);
20 midship_volume = midship_area * x(7);
21 midship_surface = 2*(x(7)*(x(4)-x(6))) + x(7)*(x(2)-2*x(6)) + pi*x(6)*x(7);
22
23 % Bow section
24 bow_area = ((pi*0.5*x(2)*x(8))/4)*2;
25 bow_volume = bow_area*x(4) - 0.25*((pi*x(6)^2)*(2*pi*((x(2)*0.5+x(8))/2)));
26 bow_circumference = 0.5*pi*( 3*(x(8)+0.5*x(2)) - sqrt( ...
    (3*x(8)+0.5*x(2))*(x(8)+(3/2)*x(2))));
27 bow_surface = bow_circumference*(x(4)-x(6)) + ( 4*pi^2 * ((x(8)+x(2))/2)*x(6));
28
29 % Aft section
30 aft_area = ((x(2)+x(3))/2) * x(1);
31 aft_volume = (aft_area * x(4)) - 0.5*((x(4)-x(5))*x(1))*x(2);
32 aft_surface = 2*x(1)*x(5) + x(3)*x(5) + ((x(2)+x(3))/2)*sqrt( x(1)^2 + ...
    (x(4)-x(5))^2);
33
34 % Summary
35 total_deck_area = bow_area + midship_deck_area + aft_area;
36 total_hull_volume = midship_volume + bow_volume + aft_volume;
37 total_hull_surface = bow_surface + midship_surface + aft_surface;
38
39 % Waterplane area
40 midship_waterplane = x(7)*x(2);
41 bow_waterplane = ((pi*0.5*x(2)*x(8))/4)*2;
42 aft_waterplane = (0.5*(x(2)+x(3)))*x(1);
43 total_waterplane_area = midship_waterplane + bow_waterplane + aft_waterplane;
44
45 L = x(1)+x(7)+x(8);
46
47 % -----
48 % ----- WEIGHT DISTRIBUTION -----
49 % -----
50
51 % ----- LIGHTSHIP -----
52
53 weight_data = xlsread('Weight_summary.xlsx','weight_summary');
54 steel_data = xlsread('Weight_summary.xlsx','hull_construction');
55 deadweight_data = xlsread('Weight_summary.xlsx','deadweight');
56
57 weight_data = load('weight_data.mat');
```

```

58 steel_data = load('steel_data.mat');
59 deadweight_data = load('deadweight_data.mat');
60
61 % Weights
62 hull_weight = weight_data.weight_data(10,3)*L;
63 topside_weight = weight_data.weight_data(15,3)*L;
64 turret_weight = weight_data.weight_data(21,3)*L;
65 LWT = hull_weight + topside_weight + turret_weight;
66
67 % Steel weight
68 hull_net_steel_weight = steel_data.steel_data(1,2)*L;
69 welding_weight = steel_data.steel_data(16,2)*L;
70
71 % LCG
72 hull_LCG = weight_data.weight_data(10,5)*L;
73 topside_LCG = weight_data.weight_data(15,5)*L;
74 turret_LCG = weight_data.weight_data(21,5)*L;
75 LWT_LCG = ...
    ((hull_LCG*hull_weight)+(topside_LCG*topside_weight)+(turret_LCG*turret_weight))/LWT;
76
77 % VCG
78 hull_VCG = weight_data.weight_data(10,7)*x(4);
79 topside_VCG = weight_data.weight_data(15,7)*x(4);
80 turret_VCG = weight_data.weight_data(21,7)*x(4);
81 LWT_VCG = ...
    ((hull_VCG*hull_weight)+(topside_VCG*topside_weight)+(turret_VCG*turret_weight))/LWT;
82
83 % TCG
84 hull_TCG = weight_data.weight_data(10,9)*x(2);
85 topside_TCG = weight_data.weight_data(15,9)*x(2);
86 turret_TCG = weight_data.weight_data(17,9)*x(2);
87 LWT_TCG = ...
    ((hull_TCG*hull_weight)+(topside_TCG*topside_weight)+(turret_TCG*turret_weight))/LWT;
88
89 % ----- DEADWEIGHT -----
90
91 % 1) Ballast condition (100% consumables)
92 % 2) Full load condition (10% consumables)
93 % 3) Full load condition (100% consumables)
94
95 for i = 1:3
96
97     if i == 1
98         deadweight = deadweight_data.deadweight_data(23,2)*L;
99         deadweight_LCG = deadweight_data.deadweight_data(23,4)*L;
100        deadweight_VCG = deadweight_data.deadweight_data(23,8)*x(4);
101        deadweight_TCG = deadweight_data.deadweight_data(23,6)*x(2);
102
103        waterBallast_weight = deadweight_data.deadweight_data(24,2)*L;
104        waterBallast_LCG = deadweight_data.deadweight_data(24,4)*L;
105        waterBallast_VCG = deadweight_data.deadweight_data(24,8)*x(4);
106        waterBallast_TCG = deadweight_data.deadweight_data(24,6)*x(2);
107
108        cargo_weight = 0;
109        cargo_LCG = 0;
110        cargo_VCG = 0;
111        cargo_TCG = 0;
112
113    elseif i == 2
114        deadweight = deadweight_data.deadweight_data(37,2)*L;
115        deadweight_LCG = deadweight_data.deadweight_data(37,4)*L;
116        deadweight_VCG = deadweight_data.deadweight_data(37,8)*x(4);
117        deadweight_TCG = deadweight_data.deadweight_data(37,6)*x(2);

```

```

118
119     waterBallast_weight = deadweight_data.deadweight_data(38,2)*L;
120     waterBallast_LCG = deadweight_data.deadweight_data(38,4)*L;
121     waterBallast_VCG = deadweight_data.deadweight_data(38,8)*x(4);
122     waterBallast_TCG = deadweight_data.deadweight_data(38,6)*x(2);
123
124     cargo_weight = (required_bbl_volume * barrel_fluid * ...
125         crude_oil_density)*1e-3;
126     cargo_LCG = 0.5*(x(1)+x(7)+x(8));
127     cargo_VCG = ((barrel_fluid*1e3)/(x(10)*x(11)))/2 + min(2, (x(2)/20));
128     cargo_TCG = 0;
129
130     elseif i == 3
131         deadweight = deadweight_data.deadweight_data(9,2)*L;
132         deadweight_LCG = deadweight_data.deadweight_data(9,4)*L;
133         deadweight_VCG = deadweight_data.deadweight_data(9,8)*x(4);
134         deadweight_TCG = deadweight_data.deadweight_data(9,6)*x(2);
135
136         waterBallast_weight = deadweight_data.deadweight_data(10,2)*L;
137         waterBallast_LCG = deadweight_data.deadweight_data(10,4)*L;
138         waterBallast_VCG = deadweight_data.deadweight_data(10,8)*x(4);
139         waterBallast_TCG = deadweight_data.deadweight_data(10,6)*x(2);
140
141         cargo_weight = (required_bbl_volume * barrel_fluid * ...
142             crude_oil_density)*1e-3;
143         cargo_LCG = 0.5*(x(1)+x(7)+x(8));
144         cargo_VCG = ((barrel_fluid*1e3)/(x(10)*x(11)))/2 + min(2, (x(2)/20));
145         cargo_TCG = 0;
146     end
147
148     DWT(i) = deadweight + cargo_weight + waterBallast_weight;
149     DWT_LCG(i) = ((deadweight*deadweight_LCG)+(cargo_weight*cargo_LCG)+
150         (waterBallast_weight*waterBallast_LCG))/DWT(i);
151     DWT_VCG(i) = ((deadweight*deadweight_VCG)+(cargo_weight*cargo_VCG)+
152         (waterBallast_weight*waterBallast_VCG))/DWT(i);
153     DWT_TCG(i) = ((deadweight*deadweight_TCG)+(cargo_weight*cargo_TCG)+
154         (waterBallast_weight*waterBallast_TCG))/DWT(i);
155
156     % ----- VESSEL TOTAL -----
157     draught_coeff = 1.07602574;
158
159     displacement(i) = DWT(i) + LWT;
160     LCG(i) = ((DWT_LCG(i)*DWT(i))+(LWT_LCG*LWT))./displacement(i);
161     VCG(i) = ((DWT_VCG(i)*DWT(i))+(LWT_VCG*LWT))./displacement(i);
162     TCG(i) = ((DWT_TCG(i)*DWT(i))+(LWT_TCG*LWT))./displacement(i);
163
164     displaced_volume(i) = displacement(i)/seawater_density;
165     mean_draught(i) = (displaced_volume(i)./total_waterplane_area)*draught_coeff;
166
167     assignin('base','displacement',displacement);
168
169     % ----- STABILITY CHECK -----
170
171     KB(i) = 0.5*mean_draught(i);
172     KG(i) = VCG(i);
173
174     %I_bow = (pi*0.5*x(2)*x(8)^3)/16 + ...
175         ((pi*0.5*x(2)*x(8))/4)*((4*x(2)*0.5)/(3*pi));
176     I_bow = (pi/16)*x(8)*(0.5*x(2))^3;
177     I_mid = (x(7)*x(2)^3)/12;
178     I_aft = (1/48)*(x(3)*x(2))*(x(3)^2+x(2)^2);

```

```

178     I_tot = I_bow + I_mid + I_aft;
179     BM(i) = I_tot/displaced_volume(i);
180
181     GM(i) = KB(i) + BM(i) - KG(i);
182     assignin('base','GM',GM);
183
184     % Effect of free surface
185     GG(i) = ((x(10)*x(11)^3)/12)*crude_oil_density.*(1./displacement(i))*(1/4);
186     GG(1) = 0;
187
188     GM_mark(i) = GM(i) - GG(i);
189     assignin('base','GM_mark',GM_mark);
190
191 end %for
192
193 % ----- WETTED SURFACE AREA -----
194
195 mean_draught = mean_draught(end);
196 assignin('base','mean_draught',mean_draught);
197
198 wetted_mid_surface = x(7)*(x(2)-2*x(6)) + pi*x(6)*x(7) + x(7)*(mean_draught-x(6));
199 wetted_bow_surface = bow_circumference*(mean_draught-x(6)) + ( 4*pi^2 ...
    *( (x(8)+x(2))/2)*x(6) );
200 wetted_aft_surface = ((x(2)+x(3))/2)*sqrt( x(1)^2 + (x(4)-x(5))^2 ) + ...
    x(1)*(x(4)-x(5));
201 wetted_surface = wetted_bow_surface + wetted_mid_surface + wetted_aft_surface;
202
203 bowSubVolume = bow_area*mean_draught - ...
    0.25*( (pi*x(6)^2)*(2*pi*((x(2)*0.5+x(8))/2)));
204 midshipSubVolume = x(2)*mean_draught*x(7) - x(7)*(0.5*pi*x(6)^2);
205 aftSubVolume = 0.5*mean_draught*x(1)*x(2);
206 hullSubVolume = bowSubVolume + midshipSubVolume + aftSubVolume;
207
208 aftLCB = (2/3)*x(1);
209 midLCB = x(1)+(0.5*x(7));
210 bowLCB = x(1)+x(7) + ((4/3)*(x(8)/pi));
211
212 LCB = ((bowSubVolume*bowLCB) + (midshipSubVolume*midLCB) + ...
    (aftSubVolume*aftLCB))/hullSubVolume;
213
214 % Coefficients calculations
215 breadth_waterline = x(2);
216 block_coefficient = displaced_volume/(L * breadth_waterline * mean_draught);
217 midship_coefficient = midship_area/(x(4)*x(2));
218 prismatic_coefficient = block_coefficient/midship_coefficient;
219 waterplane_coefficient = total_waterplane_area/(L*breadth_waterline);
220 assignin('base','block_coefficient',block_coefficient);
221
222 % -----
223 % ----- COST ESTIMATION -----
224 % -----
225
226 % Steel costs
227 baseSteelCost = 1750;
228 shapePlateCharge = 1.15;
229 cargoPlateCharge = 1.20;
230
231 % Painting costs
232 surfacePrepCost = 22.5;
233 underwaterPaintCost = 40;
234 hullDeckPaintCost = 20;
235 tankPaintCost = 10;
236 waterBallastPaintCost = 15;

```



```

237
238 % ----- SURFACE ESTIMATION -----
239
240 % Cargo tanks
241 %numTanks = 12;
242 %tankVolume = (required_bbl_volume*barrel_fluid*1e-3)/numTanks;
243 tankBreadth = x(12)/2;
244 tankLength = x(10)/6;
245 tankDepth = x(11);
246 cargoTankSurface = (2*(tankBreadth*tankLength) + 2*(tankDepth*tankBreadth) + ...
    2*(tankDepth*tankLength))*12 ...
    + 2*x(11)*x(12) + 2*x(11)+x(10) + 2*x(10)*x(12);
248
249 % Waterballast tanks
250 wbHeight = (25.27/26.6)*x(4); % Possible height of tanks with 1.33 metre tanktop
251 wbLength = x(7)/10; % Possible length of tanks
252 wbWidth = 0.5*x(2); % Width of tanks from centre of vessel
253 wbHullSurface = (wbHeight-x(6))*wbLength + (wbWidth-x(6))*wbLength + ...
    (0.5*pi*x(6)*wbLength);
254 wbSideSurface = 2*((wbWidth-x(6))*min(2, (x(2)/20)) + ...
    (wbHeight-x(6))*min(2, (x(2)/15)) + (0.25*pi*x(6)^2));
255 wbTopSurface = min(2, (x(2)/15))*wbLength;
256 wbInnerSurface = min(2, (x(2)/20))*wbLength;
257 wbBigFlatSurface = (wbWidth-x(6))*wbLength + (wbHeight-x(6))*wbLength;
258 wbTankSurface = wbHullSurface + wbSideSurface + wbTopSurface + wbInnerSurface + ...
    wbBigFlatSurface;
259
260 waterBallastSurface = 20*wbTankSurface;
261 waterBallastVolume = 20*(min(2, (x(2)/20))*tankBreadth*tankLength + ...
    min(2, (x(2)/15))*tankLength*tankDepth - 0.25*pi*x(6)^2*tankLength);
262
263 % Other tanks
264 slopTankSurface = 0;
265 otherTanksSurface = 0;
266
267 % ----- COATING COSTS -----
268
269 % Surface preparation cost
270 totalSurfaceAreaShip = cargoTankSurface + 2*total_hull_surface + ...
    total_deck_area + waterBallastSurface + slopTankSurface + otherTanksSurface;
272
273 surfacePrepCost_total = totalSurfaceAreaShip * surfacePrepCost;
274
275 % Painting costs
276 underwaterPaintCost_total = wetted_surface*underwaterPaintCost;
277 deckPaintCost_total = total_deck_area*hullDeckPaintCost;
278 hullsidePaintCost_total = 2*(total_hull_surface - ...
    wetted_surface)*hullDeckPaintCost;
279 cargoPaintCost_total = cargoTankSurface * tankPaintCost;
280 waterBallastPaintCost_total = waterBallastSurface * waterBallastPaintCost;
281
282 paintingCost_total = underwaterPaintCost_total + deckPaintCost_total + ...
    hullsidePaintCost_total + cargoPaintCost_total + waterBallastPaintCost_total;
284
285 % ----- STEEL COSTS -----
286
287 hullsideSteelCost = (hull_net_steel_weight)*baseSteelCost*shapePlateCharge;
288 weldingCost = (welding_weight)*baseSteelCost*shapePlateCharge;
289 steelCost_total = hullsideSteelCost + weldingCost;
290
291 % ----- SUM OF TOTAL COSTS -----
292 totalCost = surfacePrepCost_total + paintingCost_total + steelCost_total;
293

```

```

294 % ----- STABILITY AND MOTION SICKNESS CRITERIA -----
295
296 % Stability criteria for all loading conditions and FSM
297 if (min(GM) < 0.5 || min(GM_mark) < 0.5)
298     totalCost = 1e12;
299 end
300
301 rP = 10000; % chosen return-period (years)
302 wa = 150; % incient wave angle (180 = head sea)
303 MSDV = verticalMotions(rP,x,wa);
304
305 if MSDV > 15
306     totalCost = 1e20;
307 end
308 % % Check of MSDV value for stiff ships
309 % if min(GM(end),GM_mark(end)) > 10
310 %     MSDV = verticalMotions(rP,x,wa);
311 %     assignin('base','MSDV',MSDV);
312 %     if MSDV > 15
313 %         totalCost = 1e12;
314 %     end
315 % end
316
317 if waterBallastVolume*seawater_density < waterBallast_weight(1)
318     totalCost = 1e12;
319 end
320
321 end

```

F.5 deckArea.m

```

1 function total_deck_area = deckArea(x)
2
3 % x(1) = length of aft body
4 % x(2) = breadth at midship at design waterline
5 % x(3) = breadth at aft body
6 % x(4) = hull depth at midship
7 % x(5) = depth of upper aft hull
8 % x(6) = radius of hull bottom
9 % x(7) = length of midship body
10 % x(8) = bow forward radius
11 % x(9) = draught at aft
12 % x(10) = length of cargo tank
13 % x(11) = breadth of cargo tank
14 % x(12) = depth of cargo tank
15
16 % Deck area
17 aft_deck_area = ((x(3)+x(2))/2)*x(1);
18 midship_deck_area = x(7)*x(2);
19 bow_deck_area = ((pi*0.5*x(2)*x(8))/4)*2;
20 total_deck_area = aft_deck_area + midship_deck_area + bow_deck_area;
21 effective_deck_area = total_deck_area - bow_deck_area;
22
23 end %function

```

F.6 conditions.m

```
1 function [c,ceq] = conditions(x)
2
3 % EXAMPLE: ci = A - B -----> A <= B
4
5 % Hull limitations
6 c1 = x(3) - x(2);
7 c2 = x(2) - x(7);
8 c3 = x(1) - x(7);
9 c4 = x(4) - x(2);
10 c5 = x(9) - x(7);
11 c6 = x(9) - x(4);
12 c7 = x(1) - x(8);
13 c14 = x(2) - 55;
14
15 % Cargo limitations
16 c8 = x(12) - x(11);
17 c9 = x(10) - x(7);
18 c10 = (x(11)+min(2,(x(2)/15))) - x(2);
19 c11 = (x(12)+ min(2,(x(2)/20)))- x(4); % DNVGL-RU-SHIP Pt.3 Ch.2.
20 c12 = 1.58987e5 - x(10)*x(11)*x(12);
21 c13 = x(10)*x(11)*x(12) - 1.6e5;
22
23 c = [c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14];
24
25 ceq = [];
26
27 end
```

F.7 objectiveFunction.m

```
1 function y = objectiveFunction(x)
2
3 % Initialise for two objectives
4 y = zeros(2,1);
5
6 % Compute first objective
7 y(1) = -deckArea(x);
8
9 % Compute second objective
10 y(2) = getCost(x);
11
12 end % function
```

F.8 verticalMotions.m

```

1 function MSDV = verticalMotions(rP,x,wa)
2
3 if rP == 10000
4     Hs = 22.2;
5     Tp = 21.5;
6 end
7
8 getParticulars;
9
10 g = 9.81;           % gravitational constant
11 rho = 1025;        % water density
12 v = 0;             % Forward velocity
13 V = v*0.51444444;  % vessel speed in m/s
14 beta = wa*(pi/180); % wave heading angle (head waves is 180 deg)
15 B0 = x(2);         % maximum waterline breadth
16 Cb = block_coefficient; % block coefficient
17 T = mean_draught; % draught
18 L = x(1)+x(7)+x(8); % waterline length
19 Fn = V/(sqrt(g*L)); % Froude number
20 % longitudinal position from the centre of gravity (x=0 means COG)
21 pos = 96;
22 xpos = (L/2)*(pos/100);
23
24 Tn = 17.5;
25 D = displacement*1000; ...
26
27 ;
28 Gmt = 8.299;
29
30 delta = 0.8957;
31 gamma = 0.2234;
32 my = 0.003;
33
34 B1 = gamma*B0;
35 A0 = (Cb*B0*T)/(delta+gamma*(1-delta));
36 A1 = gamma*A0;
37 C44 = g*D*Gmt;
38
39 R = x(2)/mean_draught;
40
41 if R >= 3 && R <= 6
42     a = 0.225*(B0/T)-0.286;
43     b = -0.11*(B0/T)-2.550;
44     d = 0.033*(B0/T)-1.419;
45 else
46     a = -3.94*(B0/T)+13.69;
47     b = -2.12*(B0/T)-1.89;
48     d = 1.16*(B0/T)-7.97;
49 end
50
51 %-----%
52 period = linspace(13.1,22.2,20);
53 omega = (2*pi)./period;
54
55 for i = 1:length(omega)
56     k(i) = omega(i)^2/g;
57     omega_line(i) = omega(i) - k(i)*V*cos(beta);

```

```

58     alpha(i) = 1 - Fn*sqrt(k(i)*L)*cos(beta);
59
60     % Damping ratio between incoming & diffracted wave amplitudes
61     A(i) = 2*sin((omega_line(i)^2*B0)/(2*g))*exp(-(omega_line(i)^2*T)/g);
62
63     % Additional variables
64     k_e(i) = abs(k(i)*cos(beta)); % effective wave number
65     f(i) = (((1-k(i)*T)^2) + (((A(i)^2)/(k(i)*B0*(alpha(i)^3)))^2))^0.5;
66     kappa(i) = exp(-k_e(i)*T); % Smith correction factor
67
68     % Forcing functions
69     F(i) = kappa(i)*f(i) * (2/(k_e(i)*L)) * sin((k_e(i)*L)/2);
70     G(i) = kappa(i)*f(i) * (24/(((k_e(i)*L)^2)*L)) * ( sin((k_e(i)*L)/2)...
71         - ((k_e(i)*L)/2)*cos((k_e(i)*L)/2) );
72
73     eta(i) = (sqrt(((1-2*k(i)*T*alpha(i)^2)^2) + ...
74         ((A(i)^2/(k(i)*B0*alpha(i)^2))^2))^-1;
75
76     % Calculation of B44
77     b0_i = (1/(rho*A0*B0^2))*(sqrt(B0/(2*g)));
78     b1_i = (1/(rho*A1*B1^2))*(sqrt(B1/(2*g)));
79     b440(i) = (a*exp(b*(omega_line(i).^(-1.3)))*(omega_line(i).^d)/b0_i;
80     b441(i) = (a*exp(b*(omega_line(i).^(-1.3)))*(omega_line(i).^d)/b1_i;
81     kappa_square(i) = b441(i)./b440(i);
82
83     % Critical damping due to viscous effects added to B44
84     B44(i) = L * b440(i)*(delta+kappa_square(i)*(1-delta));
85     B44_krit(i) = ( C44 * Tn)/pi;
86     B44_tot(i) = B44(i) + my*B44_krit(i);
87
88
89     % Amplitude of the excitation moment
90     if beta == pi/2
91         M(i) = sqrt(((rho*g^2)/omega_line(i))*b440(i))*(delta+kappa(i)...
92             *(1-delta))*L;
93     else
94         M(i) = abs(sin(beta))*sqrt((rho*g^2)/omega(i))*(2/k_e(i))*...
95             sqrt(b440(i))* ( sin(0.5*delta*L*k_e(i))^2 + kappa_square(i)...
96             *sin(0.5*(1-delta)*L*k_e(i))^2 + 2*kappa(i)*sin(0.5*delta*L*...
97             k_e(i))*sin(0.5*(1-delta)*L*k_e(i))*cos(0.5*delta*L*k_e(i)))^(0.5);
98
99         en(i) = abs(sin(beta))*(sqrt((rho*g^2)/omega(i))*(2/k_e(i))*...
100             sqrt(b440(i)));
101         to(i) = 0.5*delta*L*k_e(i);
102
103         tre(i) = ((sin(to(i))^2) + kappa_square(i))*...
104             (sin(0.5*(1-delta)*L*k_e(i))^2) + 2*kappa(i)*sin(to(i))*...
105             sin(0.5*(1-delta)*L*k_e(i))*cos(to(i)))^0.5;
106
107         M(i) = en(i)*tre(i);
108     end
109
110     % Frequency responsose function in roll
111     Phi_phi(i) = abs(M(i))/((-omega_line(i)^2 * (Tn/(2*pi))^2 + 1 )^2 * ...
112         C44^2 + omega_line(i)^2 * B44_tot(i)^2)^0.5;
113
114     % Frequency response functions
115     Phi_w(i) = eta(i).*F(i); % Heave
116     Phi_theta(i) = eta(i).*G(i); % Pitch
117
118     % Vertical motion at position x
119     Phi_u(i) = sqrt(Phi_w(i).^2 + xpos^2 * Phi_theta(i).^2);
120

```

```
121     % Vertical acceleration at position x
122     Phi_v(i) = -(omega_line(i)^2) * Phi_u(i);
123
124     % Pitch acceleration
125     Phi_p(i) = -(omega_line(i)^2) * Phi_theta(i);
126
127     % Roll acceleration
128     Phi_a(i) = -(omega_line(i)^2) * Phi_phi(i);
129
130
131 end %for
132
133 A = 0.5*Hs;
134 HeaveAmp = abs(Phi_v(end)*A);
135
136 %%Time specifications:
137 Fs = 10;                % samples per second
138 dt = 1/Fs;             % seconds per sample
139 StopTime = 7200;       % seconds
140 t = (0:dt:StopTime-dt)'; % seconds
141 %%Sine wave:
142 Fc = 1/Tp;             % hertz
143
144 % Direct integration of heave acceleration function
145 fun = @(t) (abs(HeaveAmp*sin(2*pi*Fc*t)).^2);
146 q = integral(fun,0,7200);
147 MSDV = sqrt(q);
148
149 end %function
```

F.9 optimisationAnalysis.m

```

1 clear
2 load('allResults11.txt');
3 data = allResults11;
4
5 for i = 1:length(data)
6
7     % Sorting by algorithm
8     if data(i,2) == 1
9         GAarea(i) = data(i,15);
10        GAcost(i) = data(i,16);
11        gaPareto(i) = data(i,17);
12    elseif data(i,2) == 2
13        HYarea(i) = data(i,15);
14        HYcost(i) = data(i,16);
15        hyPareto(i) = data(i,17);
16    elseif data(i,2) == 3
17        HGarea(i) = data(i,15);
18        HGcost(i) = data(i,16);
19        hgPareto(i) = data(i,17);
20    end
21
22    % Sorting by initial value matrix
23    if data(i,20) == 1
24        iteratedx0Area(i) = -data(i,15);
25        iteratedx0Cost(i) = data(i,16);
26
27        % Sorting based on pareto fraction
28        if data(i,19) == 0.3
29            pfArea1(i) = -data(i,15);
30            pfCost1(i) = data(i,16);
31        elseif data(i,19) == 0.5
32            pfArea2(i) = -data(i,15);
33            pfCost2(i) = data(i,16);
34        elseif data(i,19) == 0.7
35            pfArea3(i) = -data(i,15);
36            pfCost3(i) = data(i,16);
37        elseif data(i,19) == 0.9
38            pfArea4(i) = -data(i,15);
39            pfCost4(i) = data(i,16);
40        end
41
42        if data(i,19) > 0.3
43            % Sorting by MSG value
44            if data(i,21) == 250
45                MSG250Area(i) = -data(i,15);
46                MSG250Cost(i) = data(i,16);
47                MSG250Pareto(i) = data(i,17);
48            elseif data(i,21) == 500
49                MSG500Area(i) = -data(i,15);
50                MSG500Cost(i) = data(i,16);
51                MSG500Pareto(i) = data(i,17);
52            elseif data(i,21) == 1000
53                MSG1000Area(i) = -data(i,15);
54                MSG1000Cost(i) = data(i,16);
55                MSG1000Pareto(i) = data(i,17);
56            end
57        end
58    elseif data(i,20) == 2

```

```

60         tilfeldigx0Area(i) = -data(i,15);
61         tilfeldigx0Cost(i) = data(i,16);
62     end
63
64 end
65
66 GAarea = -GAarea(GAarea<0);
67 GAcost = GAcost(GAcost>0)*1e-6;
68 HYarea = -HYarea(HYarea<0);
69 HYcost = HYcost(HYcost>0)*1e-6;
70 HGarea = -HGarea(HGarea<0);
71 HGcost = HGcost(HGcost>0)*1e-6;
72
73 gaPareto = gaPareto(gaPareto>0);
74 hyPareto = hyPareto(hyPareto>0);
75 hgPareto = hgPareto(hgPareto>0);
76
77 iteratedx0Area = iteratedx0Area(iteratedx0Area>0);
78 iteratedx0Cost = iteratedx0Cost(iteratedx0Cost>0);
79 tilfeldigx0Area = tilfeldigx0Area(tilfeldigx0Area>0);
80 tilfeldigx0Cost = tilfeldigx0Cost(tilfeldigx0Cost>0);
81
82 pfArea1 = pfArea1(pfArea1>0);
83 pfCost1 = pfCost1(pfCost1>0);
84 pfArea2 = pfArea2(pfArea2>0);
85 pfCost2 = pfCost2(pfCost2>0);
86 pfArea3 = pfArea3(pfArea3>0);
87 pfCost3 = pfCost3(pfCost3>0);
88 pfArea4 = pfArea4(pfArea4>0);
89 pfCost4 = pfCost4(pfCost4>0);
90
91 MSG250Area = MSG250Area(MSG250Area>0);
92 MSG250Cost = MSG250Cost(MSG250Cost>0);
93 MSG250Pareto = MSG250Pareto(MSG250Pareto>0);
94 MSG500Area = MSG500Area(MSG500Area>0);
95 MSG500Cost = MSG500Cost(MSG500Cost>0);
96 MSG500Pareto = MSG500Pareto(MSG500Pareto>0);
97 MSG1000Area = MSG1000Area(MSG1000Area>0);
98 MSG1000Cost = MSG1000Cost(MSG1000Cost>0);
99 MSG1000Pareto = MSG1000Pareto(MSG1000Pareto>0);
100
101 ratioGA = (mean(GAcost)/mean(GAarea))*1e6;
102 ratioHY = (mean(HYcost)/mean(HYarea))*1e6;
103 ratioHG = (mean(HGcost)/mean(HGarea))*1e6;
104 ratioGA2 = min(GAcost./GAarea)*1e6;
105 ratioHY2 = min(HYcost./HYarea)*1e6;
106 ratioHG2 = min(HGcost./HGarea)*1e6;
107
108 MSGvector(1:length(MSG250Area),1) = MSG250Area;
109 MSGvector(1:length(MSG250Area),2) = MSG250Cost;
110 MSGvector(1:length(MSG250Area),3) = MSG250Pareto;
111 MSGvector(length(MSG250Area)+1:length(MSG250Area)+length(MSG500Area),1) = ...
    MSG500Area;
112 MSGvector(length(MSG250Area)+1:length(MSG250Area)+length(MSG500Area),2) = ...
    MSG500Cost;
113 MSGvector(length(MSG250Area)+1:length(MSG250Area)+length(MSG500Area),3) = ...
    MSG500Pareto;
114 MSGvector(length(MSG250Area)+length(MSG500Area)+1:length(MSG250Area)+length(MSG500Area)
115 +length(MSG1000Area),1) = MSG1000Area;
116 MSGvector(length(MSG250Area)+length(MSG500Area)+1:length(MSG250Area)+length(MSG500Area)
117 +length(MSG1000Area),2) = MSG1000Cost;
118 MSGvector(length(MSG250Area)+length(MSG500Area)+1:length(MSG250Area)+length(MSG500Area)
119 +length(MSG1000Area),3) = MSG1000Pareto;

```



```
120
121 %%
122
123 % Normalize data to find closest point to Pareto
124 normArea = -data(:,15);
125 normCost = data(:,16);
126 zArea = (normArea - min(normArea))./(max(normArea) - min(normArea));
127 zCost = (normCost - min(normCost))./(max(normCost) - min(normCost));
128 rz = [max(zArea) min(zCost)];
129 valueRatio = normCost./normArea;
130
131 for k = 1:length(zArea)
132     d(k) = sqrt((rz(1)-zArea(k))^2 + (zCost(k)-rz(2))^2);
133 end
134
135 % [maxi,index1] = min(data(:,15));
136 % [mini,index2] = min(data(:,16));
137
138 [distMin,index1] = min(d);
139 [valueMax,index2] = min(valueRatio);
140
141 % Determine data range for optimal deck area
142 ar1 = data(index1,2);
143 startIndex1 = index1;
144 endIndex1 = index1;
145 while data(startIndex1,2) == ar1
146     if data(index1-1,2) == ar1
147         startIndex1 = startIndex1-1;
148     end
149 end
150 while data(endIndex1,2) == ar1
151     if data(index1+1,2) == ar1
152         endIndex1 = endIndex1+1;
153     end
154 end
155 rangeArea = startIndex1+1:endIndex1-1;
156
157 % Determine data range for optimal CAPEX
158 ar2 = data(index2,2);
159 startIndex2 = index2;
160 endIndex2 = index2;
161 while data(startIndex2,2) == ar2
162     if data(index2-1,2) == ar2
163         startIndex2 = startIndex2-1;
164     end
165 end
166 while data(endIndex2,2) == ar2
167     if data(index2+1,2) == ar2
168         endIndex2 = endIndex2+1;
169     end
170 end
171 rangeCAPEX = startIndex2+1:endIndex2-1;
172
173 % Output
174 x1 = data(index1,3:14);
175 x2 = data(index2,3:14);
176
177 % Calculation for GZ curves
178 shipData1 = xlsread('stability_curves.xlsx','DeckArea');
179 shipData2 = xlsread('stability_curves.xlsx','CAPEX');
180
181 % Deck area
182 [maxGZLC01,indexLC01] = max(shipData1(:,2));
```

```

183 [maxGZLC11,indexLC11] = max(shipData1(:,3));
184 [maxGZLC21,indexLC21] = max(shipData1(:,4));
185 GMLC01 = 20.60;
186 GMLC11 = 9.54;
187 GMLC21 = 9.69;
188
189 % CAPEX
190 [maxGZLC02,indexLC02] = max(shipData2(:,2));
191 [maxGZLC12,indexLC12] = max(shipData2(:,3));
192 [maxGZLC22,indexLC22] = max(shipData2(:,4));
193 GMLC02 = 24.68;
194 GMLC12 = 10.81;
195 GMLC22 = 11.05;
196 %%
197 close all
198
199 % Large figure scatter plots
200 figure('units','normalized','outerposition',[0 0 1 1])
201 sz = 4;
202
203 subplot(2,2,1)
204 scatter(GAarea,GAcost,sz,'c','filled')
205 hold on
206 grid on
207 scatter(HYarea,HYcost,sz,'m','filled')
208 scatter(HGarea,HGcost,sz,'y','filled')
209 ylabel('CAPEX (USD mill.)')
210 xlabel('Deck area (m^2)')
211 legend('GA','HY','HG','Location','northwest')
212 title('Variation of optimisation algorithm')
213
214 subplot(2,2,2)
215 scatter(iteratedx0Area,iteratedx0Cost,sz,[0.4660 0.6740 0.1880],'filled')
216 hold on
217 grid on
218 scatter(tilfeldigx0Area,tilfeldigx0Cost,sz,[0.6350 0.0780 0.1840],'filled')
219 ylabel('CAPEX (USD mill.)')
220 xlabel('Deck area (m^2)')
221 legend('Iterated','Random','Location','northwest')
222 title('Variation of initial values x_0')
223
224 subplot(2,2,3)
225 scatter(pfArea1,pfCost1*1e-6,sz,[0 0.4470 0.7410],'filled')
226 hold on
227 grid on
228 scatter(pfArea2,pfCost2*1e-6,sz,[0.8500 0.3250 0.0980],'filled')
229 scatter(pfArea3,pfCost3*1e-6,sz,[0.4940 0.1840 0.5560],'filled')
230 scatter(pfArea4,pfCost4*1e-6,sz,[1 0 0],'filled')
231 legend('P_F = 0.3','P_F = 0.5','P_F = 0.7','P_F = 0.9','Location','northwest')
232 xlabel('Deck area [m^2]')
233 ylabel('CAPEX [USD mill.]')
234 title('Variation of Pareto fraction, P_F (x_0 = iterated)')
235
236 subplot(2,2,4)
237 scatter(MSG250Area,MSG250Cost*1e-6,sz,'b','filled')
238 hold on
239 grid on
240 scatter(MSG500Area,MSG500Cost*1e-6,sz,'r','filled')
241 scatter(MSG1000Area,MSG1000Cost*1e-6,sz,'g','filled')
242 ylabel('CAPEX (USD mill.)')
243 xlabel('Deck area (m^2)')
244 legend('250','500','1000','Location','northwest')
245 title('Variation of MaxStallGenerations (x_0 = iterated, P_F > 3)')

```

```

246 %%
247 % GZ curve 1
248 figure % Deck area
249 subplot(2,1,1)
250 hold on
251 grid on
252
253 plot(shipData1(:,1),shipData1(:,2),'b') %LC0
254 plot(shipData1(:,1),shipData1(:,3),'r') %LC1
255 plot(shipData1(:,1),shipData1(:,4),'g') %LC2
256 plot([0 (180/pi)], [0 GMLC01], '--b')
257 plot([0 (180/pi)], [0 GMLC21], '--g')
258 plot([0 (180/pi)], [0 GMLC11], '--r')
259 plot([shipData1(indexLC01,1) shipData1(indexLC01,1)], [0 maxGZLC01], '-.b')
260 plot([shipData1(indexLC21,1) shipData1(indexLC21,1)], [0 maxGZLC21], '-.g')
261 plot([shipData1(indexLC11,1) shipData1(indexLC11,1)], [0 maxGZLC11], '-.r')
262
263 set(gca,'xtick',0:5:60)
264 ylim([0 12])
265 xlabel('Heeling angle (deg)')
266 ylabel('Righting arm GZ (m)')
267 legend(['LC_0 (GM = ' num2str(GMLC01),')'], ['LC_1 (GM = ' num2str(GMLC11),')'],
268 ['LC_2 (GM = ' num2str(GMLC21),')'], 'Location', 'northwest')
269 title({'Stability curve for vessel no. ' num2str(index1)}, ('IMO MSC.267(85) ...
    - Minimum design criteria applicable to all ships'))
270
271 % GZ curve 2
272 subplot(2,1,2) %CAPEX
273 hold on
274 grid on
275
276
277 plot(shipData2(:,1),shipData2(:,2),'b') %LC0
278 plot(shipData2(:,1),shipData2(:,3),'r') %LC1
279 plot(shipData2(:,1),shipData2(:,4),'g') %LC2
280 plot([0 (180/pi)], [0 GMLC02], '--b')
281 plot([0 (180/pi)], [0 GMLC12], '--r')
282 plot([0 (180/pi)], [0 GMLC22], '--g')
283 plot([shipData2(indexLC02,1) shipData2(indexLC02,1)], [0 maxGZLC02], '-.b')
284 plot([shipData2(indexLC12,1) shipData2(indexLC12,1)], [0 maxGZLC12], '-.r')
285 plot([shipData2(indexLC22,1) shipData2(indexLC22,1)], [0 maxGZLC22], '-.g')
286
287 set(gca,'xtick',0:5:60)
288 ylim([0 12])
289 xlabel('Heeling angle (deg)')
290 ylabel('Righting arm GZ (m)')
291 legend(['LC_0 (GM = ' num2str(GMLC02),')'], ['LC_1 (GM = ' ...
    num2str(GMLC12),')'], ['LC_2 (GM = ' ...
    num2str(GMLC22),')'], 'Location', 'northwest')
292 title({'Stability curve for vessel no. ' num2str(index2)}, ('IMO MSC.267(85) ...
    - Minimum design criteria applicable to all ships'))
293
294
295 %%
296
297 % Best pareto value and most value-for-money
298 figure
299 scatter(normArea,normCost*1e-6,4,'filled')
300 hold on
301 grid on
302 scatter(normArea(index1),normCost(index1)*1e-6,'g')
303 scatter(normArea(index2),normCost(index2)*1e-6,'r')
304 scatter(max(normArea),min(normCost)*1e-6,'+k')

```

```
305 xlabel('Deck area (m^2)')
306 ylabel('CAPEX (USD mill)')
307 legend('All data', ['Closest to Pareto front (Index no. ' ...
    num2str(index1), ')'], ['Best value-for-money (Index no. ' ...
    num2str(index2), ')'], ['Pareto ref. point'], 'Location', 'northwest')
308
309 %%
310
311
312 % Save for later use
313 save('x1.mat', 'x1')
314 save('x2.mat', 'x2')
```

F.10 motionResults.m

```

1 clear
2 vesselNo = 4843;
3
4 longCOG = xlsread('Vessel_motions_Pareto.xlsx','Long_COG');
5 longCOG2 = xlsread('Vessel_motions.xlsx','Long_COG');
6 longHel = xlsread('Vessel_motions_Pareto.xlsx','Long_Heli');
7 msData = xlsread('Vessel_motions_Pareto.xlsx','MSDV');
8 msData2 = xlsread('Vessel_motions.xlsx','MSDV');
9 raoCOG = xlsread('Vessel_motions_Pareto.xlsx','RAO_COG');
10
11 heliCrit = xlsread('Vessel_motions_Pareto.xlsx','Heli_crit');
12 opCrit = xlsread('Vessel_motions.xlsx','Operation_crit');
13 specData = xlsread('Vessel_motions_Pareto.xlsx','Spectrum');
14 specData1 = flipud(specData(:,1:2));
15 specData2 = specData(1:25,4:5);
16
17 Tp = heliCrit(:,1);
18 Rp = longCOG(1:7,1);
19 wP = linspace(5,25,21);
20 Rp2 = [1 5 10 20 50 100 1000 10000];
21 heading = [0.6 0.3 0.1];
22
23 loadCases = [16.5 13.1; 17.9 15.3; 18.9 16.9; 19.2 17.5; 20.4 19.9; 21.5 22.2];
24 labels = {'1-Year','10-Year','50-Year','100-Year','1k-Year','10k-Year'};
25
26 hsPoint = zeros(length(specData1),1);
27 tpPoint = zeros(length(specData2),1);
28
29 for i = 1:length(specData1)
30     hsPoint(i) = specData1(i,2)/sum(specData1(:,2));
31 end
32
33 for j = 1:length(specData2)
34     tpPoint(j) = specData2(j,2)/sum(specData2(:,2));
35 end
36
37 totalPointHs = cumsum(hsPoint);
38 totalPointTp = cumsum(tpPoint);
39 %% Statistical data for wave spectrum
40 ax1 = figure;
41
42 ax11 = subplot(1,2,1);
43 grid on
44 hold on
45 plot(specData1(:,1),totalPointHs*100,'LineWidth',1.5)
46 plot([4.5 4.5],[0 totalPointHs(10)*100],':r','LineWidth',1.5)
47 plot([0 4.5],[totalPointHs(10)*100 totalPointHs(10)*100],':r','LineWidth',1.5)
48 plot([5 5],[0 totalPointHs(11)*100],':r','LineWidth',1.5)
49 plot([0 5],[totalPointHs(11)*100 totalPointHs(11)*100],':r','LineWidth',1.5)
50 xlabel('Significant wave height, H_s (m)')
51 ylabel('Percentage of time (%)')
52 title('Cumulative probability of significant wave height')
53 set(gca,'xtick',0:2:18)
54 xlim([0 18])
55 set(gca,'LooseInset',get(gca,'TightInset'));
56 ax11.Position = [0.05 0.1162 0.42 0.8065];
57
58 ax12 = subplot(1,2,2);
59 hold on

```

```

60 plot(specData2(:,1),totalPointTp*100,'LineWidth',1.5)
61 plot([15 15],[0 totalPointTp(16)*100],':r','LineWidth',1.5)
62 plot([0 15],[totalPointHs(16)*100 totalPointHs(16)*100],':r','LineWidth',1.5)
63 grid on
64 xlabel('Peak wave period [sec]')
65 ylabel('Percentage of time (%)')
66 title('Cumulative probability of wave peak period')
67 set(gca,'xtick',0:2:24)
68 xlim([0 24])
69 set(gca,'LooseInset',get(gca,'TightInset'));
70 ax12.Position = [0.56 0.1162 ax11.Position(3) ax11.Position(4)];
71
72 ax1.Position = [622 485 993 420];
73 printName = 'M:\Thesis\Master thesis\IMAGES\waveProbabilities.png';
74 saveas(gcf,printName)
75
76
77 %% MSDV with Hs range [5,20]
78
79 ax2 = figure;
80 plot(wP,msData(:,12),wP,msData(:,11),wP,msData(:,10),wP,msData(:,9),'LineWidth',1.05)
81 grid on
82 hold on
83 plot([5 25],[15 15],'-k','LineWidth',1.5)
84 xlabel('Wave peak period [sec]')
85 ylabel('MSDV [m/s^{1.5}]')
86 set(gca,'xtick',5:1:25)
87 legend('H_s = 20 m','H_s = 15 m','H_s = 10 m','H_s = 5 ...
      m','Threshold','Location','northeast')
88 title({'Motion sickness dose value according to ISO 2631-1:1997',...
      'Duration of exposure 2.0 hours. Wave heading \beta = 30^\circ','Vessel No. 4843'})
89 set(gca,'LooseInset',get(gca,'TightInset'));
90 ax2.Position = [482 353 990 450];
91 saveas(gcf,'M:\Thesis\Master thesis\IMAGES\MSDV_ValueVessel(4843).png')
92
93
94 %% MSDV criteria for both vessels (with applied heading probability)
95
96 MSDVcrit4843 = heading(1)*msData(:,15) + heading(2)*msData(:,16) + ...
      heading(3)*msData(:,17);
97 MSDVcrit1864 = heading(1)*msData2(:,15) + heading(2)*msData2(:,16) + ...
      heading(3)*msData2(:,17);
98
99 ax3 = figure;
100 plot(msData(:,14),MSDVcrit4843,'b','linewidth',1.05)
101 hold on
102 grid on
103 plot(msData2(:,14),MSDVcrit1864,'r','linewidth',1.05)
104 scatter(loadCases(:,1),loadCases(:,2),20,'filled','k')
105 text(loadCases(:,1),loadCases(:,2),labels,'VerticalAlignment','bottom',
106 'HorizontalAlignment','right')
107 xlabel('Peak period, Tp [sec]')
108 ylabel('Significant wave height, H_s [m]')
109 legend('Vessel 4843','Vessel 1864','Load cases','Location','southeast')
110 set(gca,'xtick',3:1:25)
111 xlim([3 25])
112 title({'Maximum operating criteria for MSDV < 15. Duration of exposure 2.0 ...
      hours','Location: accommodation structure','Applied heading probabilities'})
113 set(gca,'LooseInset',get(gca,'TightInset'));
114 ax3.Position = [482 290 990 513];
115 % saveas(gcf,'M:\Thesis\Master thesis\IMAGES\MSDV_criteria_bothVessels.png')
116
117 %% MSDV location dependent
118

```

```

119 COG30 = load('M:\Thesis\Master thesis\ShipX ...
      Results\ParetoVessel\MSDV_COG_30deg.mpl');
120 Acc30 = load('M:\Thesis\Master thesis\ShipX ...
      Results\ParetoVessel\MSDV_Acc_30deg.mpl');
121 ULS = [13.1 15.3 16.9 17.5 19.9 22.2];
122 Acc30Mod = [7.3898 8.6309 9.5334 9.8719 11.2258 12.5232];
123 COG30Mod = [2.8744 3.3571 3.7082 3.8398 4.3665 4.8711];
124
125 ax35 = figure;
126 plot(Acc30(:,1),Acc30(:,2),'-ob',COG30(:,1),COG30(:,2),'-db','LineWidth',1.05)
127 grid on
128 hold on
129 plot(ULS,Acc30Mod,'-or',ULS,COG30Mod,'-dr','LineWidth',1.05)
130 % plot([13 23],[15 15],'-.k','LineWidth',0.9)
131 ylabel('MSDV [m/s^{1.5}]')
132 xlabel('Significant wave height, H_s [m]')
133 legend('Accommodation (ShipX)','Centre of gravity (ShipX)','Accommodation ...
      (Model)','Centre of gravity (Model)','Location','east')
134 set(gca,'xtick',13:0.5:23,'YMinorTick','on','ytick',2:2:24)
135 set(gca,'LooseInset',get(gca,'TightInset'));
136 ylim([2 24])
137 title({'Motion sickness dose value (MSDVz)','Wave heading \beta = 30^\circ. Vessel ...
      No. 4843'})
138 ax35.Position = [482 353 990 450];
139 % saveas(gcf,'M:\Thesis\Master thesis\IMAGES\MSDV_location.png')
140
141 %% Sensitivity with variable MSDV
142
143 loct = [0.5 0.5];
144
145 ax4 = figure;
146
147 ax41 = subplot(1,2,1);
148 plot(msData2(:,14),msData2(:,20),msData2(:,14),msData2(:,21),msData2(:,14),msData2(:,17),
149 'LineWidth',1.05)
150 grid on
151 hold on
152 scatter(loadCases(:,1),loadCases(:,2),20,'filled','k')
153 text(loadCases(:,1),loadCases(:,2),labels,
154 'VerticalAlignment','bottom','HorizontalAlignment','right')
155 xlabel('Tp [sec]')
156 ylabel('H_s [m]')
157 legend({'MSDV_{max} = 21','MSDV_{max} = 18','MSDV_{max} = ...
      15'},'Location','south','NumColumns',3)
158 set(gca,'xtick',0:1:25)
159 xlim([7 25]); ylim([7 24])
160 title('Vessel No. 1864')
161 set(gca,'LooseInset',get(gca,'TightInset'));
162 ax41.Position = [0.05 0.1162 0.42 0.7];
163
164 ax42 = subplot(1,2,2);
165 plot(msData(:,14),msData(:,21),msData(:,14),msData(:,20),msData(:,14),msData(:,19),
166 'LineWidth',1.05)
167 grid on
168 hold on
169 scatter(loadCases(:,1),loadCases(:,2),20,'filled','k')
170 text(loadCases(:,1),loadCases(:,2),labels,'VerticalAlignment','bottom',
171 'HorizontalAlignment','right')
172 xlabel('Tp [sec]')
173 ylabel('H_s [m]')
174 legend({'MSDV_{max} = 21','MSDV_{max} = 18','MSDV_{max} = ...
      15'},'Location','south','NumColumns',3)
175 set(gca,'xtick',0:1:25)

```

```

176 xlim([7 25]); ylim([7 24])
177 title('Vessel No. 4843')
178 ax42.Position = [0.56 0.1162 ax41.Position(3) ax41.Position(4)];
179
180 sgtitle({'Maximum operating criteria for MSDV. Duration of exposure 2.0 hours',
181 'Location: accommodation structure. Heading \beta = 30^\circ'})
182 ax4.Position = [482 353 990 450];
183 saveas(gcf, 'M:\Thesis\Master thesis\IMAGES\MSDV_multiple.png')
184
185 %% Excursion envelope
186
187 % Estimation of 1k and 10k return period data for vessel 4843
188 exp_curve1 = fit(longCOG(1:7,2),Rp, 'exp1');
189 exp_curve2 = fit(longCOG(1:7,3),Rp, 'exp1');
190 surgeCo1 = [exp_curve1.a exp_curve1.b];
191 swayCo1 = [exp_curve2.a exp_curve2.b];
192 surgeData = [(1/surgeCo1(2))*log(1000/(surgeCo1(1))) ...
193             (1/surgeCo1(2))*log(10000/(surgeCo1(1)))];
194 swayData = [(1/swayCo1(2))*log(1000/(swayCo1(1))) ...
195            (1/swayCo1(2))*log(10000/(swayCo1(1)))];
196
197 % Estimation of 1k and 10k return period data for vessel 1864
198 exp_curve3 = fit(longCOG2(1:7,2),Rp, 'exp1');
199 exp_curve4 = fit(longCOG2(1:7,3),Rp, 'exp1');
200 surgeCo12 = [exp_curve3.a exp_curve3.b];
201 swayCo12 = [exp_curve4.a exp_curve4.b];
202 surgeData2 = [(1/surgeCo12(2))*log(1000/(surgeCo12(1))) ...
203             (1/surgeCo12(2))*log(10000/(surgeCo12(1)))];
204 swayData2 = [(1/swayCo12(2))*log(1000/(swayCo12(1))) ...
205            (1/swayCo12(2))*log(10000/(swayCo12(1)))];
206
207 ax5 = figure;
208 hold on
209 plot(longCOG(2:9,2),Rp2, '-sb', 'LineWidth',1.05) % Surge 4843
210 plot(longCOG(2:9,3),Rp2, '-db', 'LineWidth',1.05) % Sway 4843
211 plot(longCOG2(2:9,2),Rp2, '-sr', 'LineWidth',1.05) % Surge 1864
212 plot(longCOG2(2:9,3),Rp2, '-dr', 'LineWidth',1.05) % Sway 1864
213 grid on
214 xlabel('Displacement [m]')
215 ylabel('Return period [years]')
216 set(gca, 'Yscale', 'log', 'xtick', 0:1:12)
217 yticks([1 5 10 20 50 100 1000 10000])
218 yticklabels({'1 year', '5 years', '10 years', '20 years', '50 years', '100 ...
219             years', '1k years', '10k years'});
220 ylim([0 12000])
221 legend('Surge 4843', 'Sway 4843', 'Surge 1864', 'Sway 1864', 'Location', 'northwest')
222 title({'Displacement at centre of gravity', 'JONSWAP (\gamma = 2.6). ...
223         Long-crested seas'})
224 set(gca, 'LooseInset', get(gca, 'TightInset'));
225 ax5.Position = [482 353 990 450];
226 saveas(gcf, 'M:\Thesis\Master thesis\IMAGES\excursion_figure(4843).png')
227
228 %% Helicopter criteria
229
230 ax6 = figure;
231 plot(Tp, 0.5*heliCrit(:,4)+0.5*heliCrit(:,7), 'b', 'LineWidth',1.05)
232 hold on
233 grid on
234 plot(Tp, 0.5*heliCrit(:,10)+0.5*heliCrit(:,13), 'r', 'LineWidth',1.05)

```



```

233 scatter(loadCases(:,1),loadCases(:,2),20,'filled','k')
234 text(loadCases(:,1),loadCases(:,2),labels,
235 'VerticalAlignment','bottom','HorizontalAlignment','right')
236 xlabel('Peak period [sec]')
237 ylabel('Significant wave height, H_s [m]')
238 legend('Vessel 4843','Vessel 1864','Location','northeast')
239 set(gca,'xtick',5:1:25,'YMinortick','on','ytick',1:2:24)
240 ylim([1 21])
241 title({'Maximum operating limits','Helicopter operation. Night and daytime ...
        operations'})
242 set(gca,'LooseInset',get(gca,'TightInset'));
243 ax6.Position = [482 353 990 600];
244 saveas(gcf,'M:\Thesis\Master thesis\IMAGES\helicopter_valueVesselALL.png')
245
246
247 %% Lockdown criteria
248
249 ax7 = figure;
250 plot(Tp,opCrit(:,2),Tp,opCrit(:,3),Tp,opCrit(:,4),'LineWidth',1.05)
251 hold on
252 scatter(loadCases(:,1),loadCases(:,2),20,'filled','k')
253 text(loadCases(:,1),loadCases(:,2),labels,'VerticalAlignment','bottom',
254 'HorizontalAlignment','right')
255 grid on
256 xlabel('Peak period [sec]')
257 ylabel('Significant wave height, H_s [m]')
258 legend('0^o heading','15^o heading','30^o heading','Load ...
        cases','Location','northwest')
259 set(gca,'xtick',5:1:25)
260 title({'Maximum operating limits','Lockdown of vessel','Vessel No. 1864'})
261 set(gca,'LooseInset',get(gca,'TightInset'));
262 ax7.Position = [482 353 990 450];
263 % saveas(gcf,'M:\Thesis\Master thesis\IMAGES\lockdown_valueVessel(1864).png')
264 fileNameax7 = 'lockdown_valueVessel(1864).png';
265 path = 'M:/Master thesis/IMAGES/';
266 figureName = [path,fileNameax7];
267 % sprintf('\includegraphics[width=\textwidth]{%s%s}',path,fileNameax7);
268
269 %% Response amplitude operator for roll
270
271 ax8 = figure;
272 plot(raoCOG(2:31,18),raoCOG(2:31,20),'color',[0.3010 0.7450 ...
        0.9330],'LineWidth',1.05)
273 hold on
274 grid on
275 plot(raoCOG(2:31,18),raoCOG(2:31,21),'color',[0 0.4470 0.7410],'LineWidth',1.05)
276 plot(raoCOG(2:50,22),raoCOG(2:50,24),'color',[0.8500 0.3250 ...
        0.0980],'LineWidth',1.05)
277 plot(raoCOG(2:50,22),raoCOG(2:50,25),'color',[0.6350 0.0780 ...
        0.1840],'LineWidth',1.05)
278 xlabel('Wave period [sec]')
279 ylabel('R.A.O. Roll \eta_4/A [deg/m]')
280 set(gca,'xtick',5:1:25)
281 xlim([5 25])
282 title({'Response amplitude operators'})
283 legend('Vessel 1864: 15^o heading','Vessel 1864: 30^o heading','Vessel 4843: ...
        15^o heading',
284 'Vessel 4843: 30^o heading','Location','northwest')
285 set(gca,'LooseInset',get(gca,'TightInset'));
286 ax8.Position = [482 353 1030 540];
287 saveas(gcf,'M:\Thesis\Master thesis\IMAGES\RAO_figure_ALL.png')
288
289 %% Response amplitude operator for heave velocity

```

```

290
291 parHel = load('M:\Thesis\Master thesis\ShipX ...
      Results\ParetoVessel\RAO_HeavVel.mpl');
292 valHel = load('M:\Thesis\Master thesis\ShipX Results\ValueVessel\RAO_HeavVel.mpl');
293 probs = [0.6 0.3 0.1];
294
295 heaveRatePareto(:,1) = parHel(1:49,1);
296 heaveRateValue(:,1) = valHel(1:31,1);
297 heaveRatePareto(:,2) = parHel(1:49,2)*probs(1) + parHel(50:98,2)*probs(2) + ...
      parHel(99:end,2)*probs(3);
298 heaveRateValue(:,2) = valHel(1:31,2)*probs(1) + valHel(32:62,2)*probs(2) + ...
      valHel(63:end,2)*probs(3);
299
300 close all
301 ax9 = figure;
302 grid on
303 hold on
304 plot(heaveRatePareto(:,1),heaveRatePareto(:,2),'LineWidth',1.05)
305 plot(heaveRateValue(:,1),heaveRateValue(:,2),'LineWidth',1.05);
306 legend('Vessel no. 4843','Vessel no. 1864','Location','northwest')
307 xlabel('WAVE PERIOD [sec]')
308 ylabel('RAO HEAVE VEL. \eta(3)/A [(m/s)/m]')
309 title('Response amplitude operator for heave velocity')
310 set(gca,'xtick',0:1:30,'YMinortick','on')
311 xlim([0 25])
312 set(gca,'LooseInset',get(gca,'TightInset'));
313 ax9.Position = [680 493 921 485];
314 saveas(gcf,'M:\Thesis\Master thesis\IMAGES\RAO_HeaveRate.png')
315
316 %% Operability bars
317
318 opData = xlsread('operability.xlsx');
319
320 %%
321 heliOp = [min(opData(1:4,4)) min(opData(6:9,4)) min(opData(11:13,4));
322 min(opData(1:4,10)) min(opData(6:9,10)) min(opData(11:13,10))];
323 c = categorical({'Vessel 4843','Vessel 1864'});
324
325 ax30 = figure;
326 b = bar(c,heliOp*100,'FaceAlpha',0.8);
327 set(gca,'YMinortick','on')
328 legend({'Helicopter DAY','Helicopter ...
      NIGHT','Lockdown'},'Location','northoutside','NumColumns',3)
329 ytickformat('%g%')
330 ylim([0 100])
331 set(gca,'LooseInset',get(gca,'TightInset'));
332 ax30.Position = [680 671 830 307];
333 saveas(gcf,'M:\Thesis\Master thesis\IMAGES\operabilityBars.png')

```

