Modularity to Support the Design of a Super Yacht

The implementation of a modular method in the early stage design process of a custom luxury yacht.

De Voogt Naval Architects

H.J.J. Marcus

Delft University of Technology

Modularity to Support the Design of a Super Yacht

The implementation of a modular method in the early stage design process of a custom luxury yacht.

by

H.J.J. Marcus

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday November 25, 2024 at 14:00.

Student number:	4654846	
Project duration:	March 11, 2024 – November 25, 2024	
Thesis committee:	Dr. Ir. P. de Vos,	TU Delft, Chair
	Ing. B. C. Jongepier,	DVNA
	Ir. J. L. Gelling,	TU Delft
	Ir. E. L. Scheffers,	TU Delft
	Dr. Ir. A. A. Kana,	TU Delft

This thesis is confidential and cannot be made public until December 31, 2024.

Cover: Feadship concept Dunes

TU Delft

Thesis for the degree of MSc in Marine Technology in the specialization of
Ship Design

Modularity to Support the Design of a Super Yacht

By

Hendrik Jan Jaap Marcus

Performed at

De Voogt Naval Architecture

This thesis **MT.24/45.016.M** is classified as confidential in accordance with the general conditions for projects performed by the TUDelft.

25-11-2024

Company supervisors
Responsible supervisor: Ing. T. Weeda

Daily Supervisor: Ing. B. C. Jongepier

Thesis exam committee
Chair/Responsible Professor: Dr. Ir. P. de Vos
Staff Member: Ir. J. L. Gelling
Staff Member: Ir. E. Scheffers
Staff Member: Dr. Ir. A. A. Kana
Company Member: Ing. B. C. Jongepier

Author Details
Studynumber: 4654846

Preface

Before you is my master's thesis: "Modularity to Support the Design of a Super Yacht." This research was conducted in fulfillment of the requirements for the degree of Master of Science in Marine Technology. Throughout my life, I have had a profound passion for everything that floats, but I could never have imagined that this passion would eventually lead me to graduate in this field. The fact that I not only had the opportunity to complete my studies in a field I love but also contribute to the yacht design industry makes this achievement even more meaningful.

I would like to begin by thanking Jaap Gelling, who first introduced me to Feadship. Jaap's support throughout my research did not end there. As my supervisor from TU, he was always ready to help put "unsolvable problems" into perspective and to work with me to find solutions. And of course i would like to thank Bram Jongepier for his daily guidance of my work within the Design Studio. I am grateful for your humor, creativity, and, of course, expertise during our many meetings.

My thanks go to Tanno Weeda for giving me the incredible opportunity to conduct my research within the Design Studio on our challenging and ambitious topic, "Modularity within Feadship." Alongside Tanno and Bram, I would like to thank the rest of the Design Studio team for their support and "gezelligheid" throughout my research period.

I would also like to express my gratitude to Evelien Scheffers for all her support during the project. Without your scientific insights and wisdom, my research would not have reached its current academic standard. On that note, I would like to thank the chair of my thesis committee, Peter de Vos, for his contribution to this research. At key moments, you knew exactly how to challenge and motivate me, which helped bring this final report to completion.

A nine-month research project is not something you can endure alone, so I would also like to thank my friends for all the essential moments of enjoyment during this period. Special thanks to my fellow thesis companions, Willem Toet and Sietse Duister, for sharing all the highs and lows of the graduation journey.

Finally, I want to extend a special thanks to my parents for their endless support throughout my entire study period.

I hope you enjoy reading this research.

Jaap Marcus
Amsterdam, November 2024

Abstract

This thesis investigates the implementation of modularity in the early-stage design process of custom luxury yachts, specifically targeting the Feadship fleet between 75 and 110 meters. The research aims to determine whether modular design methods can optimize efficiency and creativity while maintaining the high degree of customization demanded by clients. To address this, the thesis introduces Modular Function Deployment (MFD) as a structured framework for identifying and evaluating yacht systems suitable for modularization.

Key insights from the study demonstrate that MFD, coupled with innovative tools such as the Area Prediction tool and the Arrangement Generator tool, can enhance both design creativity and efficiency. Contrary to concerns that modularity might restrict creativity, these tools offer designers a structured yet flexible platform for exploring numerous configurations. This encourages the exploration of innovative design arrangements that push the boundaries of conventional yacht architecture.

The study also focuses on evaluating how designers can benefit from modular principles. The Area Prediction tool, based on the Random Forest regression model, predicts the surface areas of different yacht modules. The Arrangement Generator tool allows designers to visualize potential layouts, iterating through various combinations rapidly. These tools support designers in generating new, optimized arrangements that maintain high levels of customization. A case study with Feadship designers highlights the fact that modularity offers substantial benefits, although challenges remain in terms of integrating these tools fully into the creative process.

Future research is suggested to explore whether yacht clients will accept modularity without perceiving the designs as less bespoke, how designers can shift from traditional bespoke methods to modular approaches, and how regulatory challenges may be navigated. But more important, the impact of the ongoing energy transition on future yacht designs is considered significant, necessitating future updates to the prediction tools as yacht specifications evolve.

Contents

1
Introduction

The luxury yacht industry is characterized by a demand for highly customized and innovative designs that respond to the unique preferences of demanding clients. However, the traditional design process often involves long timelines and complex iterations, which can hinder the ability to quickly adapt to customer feedback and emerging trends. In this context, the concept of modularity offers a promising solution by enabling more efficient, flexible and innovative design processes.
Modularity in the early stage design of luxury yachts has the potential to revolutionize how these vessels are conceptualized and developed. By dividing the yacht into distinct, self-contained modules, designers can more rapidly iterate on designs, incorporate client feedback, and explore a wider variety of configurations. This approach not only enhances the creative possibilities but also streamlines the overall design workflow, reducing lead times and allowing for quicker adaptation to new technologies and evolving client needs. Furthermore, modularity can significantly improve operational efficiency and maintenance throughout the yacht's lifecycle. Modular components and building facilitate easier updates, repairs, and upgrades, which are crucial for maintaining the vessel's performance. This adaptability is particularly valuable in an industry where innovation and technological advancements are constant.
This research will focus specifically on the feasibility and benefits of implementing modularity during the early stage design process of luxury yachts. It will examine the potential for modularity to enhance design efficiency, creativity, and client satisfaction while identifying the limitations and challenges inherent in adopting this approach within the context of fully custom luxury yacht design. The study will not cover the detailed engineering phase or the broader operational and economic impacts of modularity beyond the design stage. Additionally, the investigation will be limited to the design practices and workflows at Feadship, a leading shipyard in the luxury yacht industry, ensuring that the findings are directly applicable to real-world scenarios within this specific context.

Feadship - De Voogt Naval Architects

This research is conducted in collaboration with De Voogt Naval Architects and Delft University of Technology. The project will be carried out within the design studio of De Voogt Naval Architects, which is a part of Feadship.
Feadship is short for the First Export Association of Dutch SHIPbuilders. This organization was established in 1949 following the aftermath of World War II, which severely affected the yacht building industry in the Netherlands. With Europe facing financial constraints that limited the demand for pleasure boats, Dutch yacht builders sought opportunities abroad. Six shipyards and De Voogt Naval Architects decided to promote their top-notch products in the United States under the brand name Feadship. Today there are three parties left in Feadship: Two of the original yards – Van Lent and De Vries – and De Voogt Naval Architects as shown in figure 1.1.

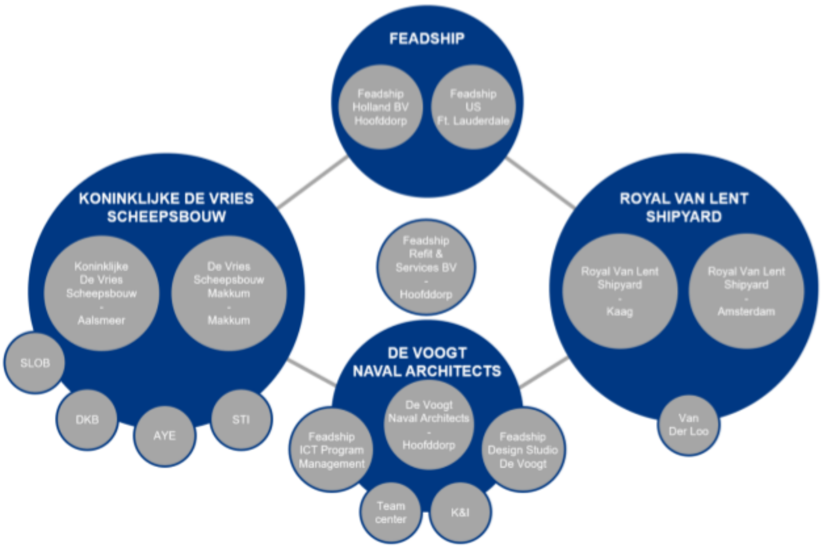


Figure 1.1: Company Structure [Feadship]

"Feadship designs more than just yachts. It crafts wants and desires, quirks and eccentricities. It creates the individual and unique, catering to the uncompromising and exacting. Transforming blank pages and open minds into dreams realized in wood and steel. That task of building just yachts is left to the others. Feadship builds Feadships"[Feadship,]. As described by the marketing department of Feadship, the building process of a Feadship differs from an ordinary yacht. In the design and production process of Feadship, the client plays an active role at every step. It begins with a designer who listens to the client's ideas, visions, and dreams, and then transforms them into a Feadship yacht. De building process of a Feadship starts each project again with a blank sheet of paper assuming that everything is possible. Because each Feadship's design is entirely customized according to the preferences of its new owner, no two Feadships are alike, making each project innovative. Alongside the philosophy of carte blanche design, craftsmanship is highly valued at Feadship. Feadship yachts are renowned globally for their exceptional quality, with comfort and safety being the two main pillars of focus.

Relevance

This research addresses a gap in the scientific literature by exploring how modularity can be applied to the early stage design of luxury yachts, a field traditionally dominated by bespoke approaches. By adapting Modular Function Deployment (MFD) to this context, the study contributes new insights into how modular principles can enhance creativity, efficiency, and adaptability in custom yacht design. It offers a fresh perspective on balancing customization with standardization, advancing scientific understanding of modular design in complex, high-end products. Something that hasn't been done within existing literature, therefore giving a scientific research gap.
The study provides practical value for the luxury yacht industry, particularly for companies like Feadship, by proposing methods to integrate modularity into the design process. This can reduce lead times, streamline workflows, and improve design flexibility, addressing key industry challenges. The use of modularity also facilitates easier maintenance, upgrades, and compliance with new regulations, enhancing the operational and long-term value of luxury yachts. The research offers actionable guidelines that could lead to more efficient and adaptable design practices across the industry, while focussing on the Feadship design process.

Problem statement, objective and research questions

The luxury yacht design process is highly customized and complex, often leading to lengthy timelines, high costs, and limited flexibility to adapt to client feedback and technological advancements. Traditional design methods do not adequately address the need for efficiency and adaptability in the early stages of yacht design, resulting in missed opportunities for innovation and client satisfaction. The potential of modularity, a well-established approach in other industries, remains underutilized in luxury

yacht design, leaving a gap in both the literature and industry practice.
The objective of this research is to investigate the feasibility and benefits of implementing modularity in the early stage design process of luxury yachts. Specifically, it aims to develop a modular design framework that enhances design efficiency, reduces lead times, and supports creative design while maintaining the high level of customization required by clients. The study seeks to provide practical guidelines and tools for integrating modular principles into yacht design, offering a pathway to more innovative and efficient design practices in the luxury yacht industry.
To investigate this research objective, the following **research question** has been formulated:

How can a custom yacht company implement modularity to optimize design efficiency, and creativity in the early stage design process while maintaining the high level of customization required for a custom yacht?

To develop a well-substantiated answer to this research question, the following sub-questions have been formulated:

1. How can a modular design method be implemented in the design process to identify the required systems for luxury yachts, and evaluate which systems are suitable as a module?
2. What tool can predict the surface areas of modules based on design brief parameters, and how do owner-specific requirements influence these predictions?
3. How can designers be supported in exploring innovative arrangements of modules during the early stage design of luxury yachts?

Outline

This study explores how modularity can enhance the early stage design process of luxury yachts by combining theoretical insights and practical applications. Beginning with a literature review on Modularity, it examines the benefits and challenges of modular design. The research then focuses on Modular Function Deployment (MFD) and the implementing of MFD at Feadship, identifying key module drivers and evaluating systems for modularization to balance efficiency with customization. A System Area Prediction model is developed to estimate module sizes based on design brief parameters, distinguishing between standard influences and owner-specific needs. Additionally, a packing algorithm is introduced to help designers explore innovative module arrangements, supporting creativity within the modular framework. Ultimately, the investigated and developed methods will be applied to existing design numbers and corresponding input parameters from their design briefs. The generated results will be compared with the initial designs created by the De Voogt designer. Subsequently, the results will be analyzed and discussed in collaboration with the Senior Designer and Senior Specialist.

2
Modularity

2.1. Understanding Modularity

Definition and evolution

Modularity is a key architectural strategy which involves dividing systems into distinct, self-contained units known as modules. Historically, this concept can be traced back to ancient civilizations with examples such as the pyramids of Egypt and the aqueducts of Rome. The modern era of modularity in architecture began with the Bauhaus movement, which emphasized the combination of standardization and functional design.

Technical and Biological Perspectives

In the biological context, modularity refers to the arrangement of interconnected elements within organisms, highlighting patterns of diversity and similarity across species. Technically, modularity has evolved from tangible products to intangible software systems, defining modules as self-contained units that function independently and interface with other modules through standardized connections. A module, in a technical design context, can be seen as a physical realization of a function [Pahl and Beitz, 1988]. This perspective emphasizes that each module not only serves a structural role but also carries out specific functions within the overall system.

Key terms

Modularity offers several benefits including resource efficiency, product variety, flexibility, and the ability to work on components in parallel. Key terms include:

- **Module:** A self-contained functional unit with standarised interfaces, often considered a physical realization of a function.
- **Modularity:** The cahracteristic of a system being composed of modules.
- **Modularization:** The process of dividing a system into modules.

2.2. Modularity in technical product design

Types of product design modularity

As shown in figure 2.1, [Ulrich, 1994] identifies five types of modularity in product design:

- **Component sharing:** Using common components across different products, such as motors in various power tools.
- **Component swapping:** Exchanging specific components to customize the product while keeping the base unchanged, like interchangeable lenses in sunglasses.
- **Fabricate to fit:** Tailoring products to specific dimensions, for example custom-made clothing.
- **Bus modularity:** Connecting different components via a standarized interface, such as electronic components on a computer motherboard.

- **Sectional modularity:** Directly connecting modules without a separate linking system, like LEGO bricks.

As shown in figure 2.2, [Salvador et al., 2002] identifies six types of modularity by adding combinational modularity , which involves diverse interfaces without a central body.

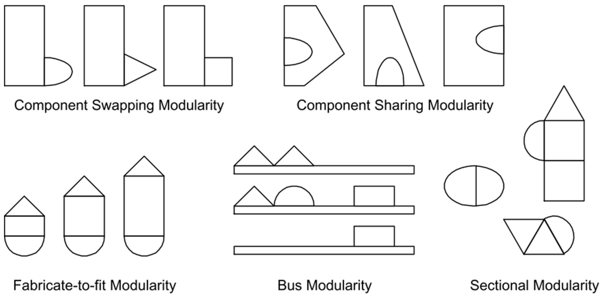


Figure 2.1: Types of Modularity [Ulrich, 1994]

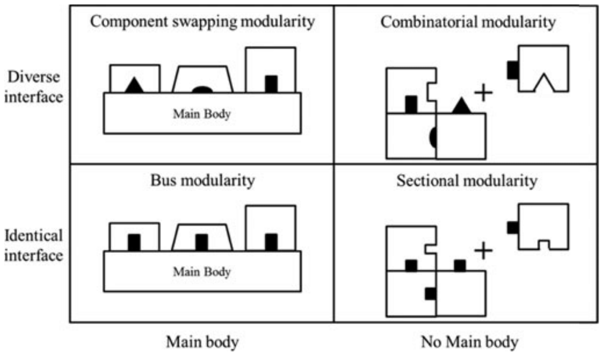


Figure 2.2: Types of Modularity [Salvador et al., 2002]

2.3. Application in ship design.

Modularity enhances different stages of a vessel's life cycle: design, construction, and operation. It allows for rapid design iterations, increased creativity, and efficiency during construction. Modular systems also facilitate easier maintenance and upgrades, enabling ships to adapt to new regulations and technologies with minimal disruption[Erikstad, 2019].

Design phase contributions

Modularity can reduce lead time during the design phase, enabling quick reassessment and new solutions. This leads to a wider variety of designs, increased creativity, and greater flexibility. However, it can also restrict creativity by confining designers to specific modules and interfaces.

Construction phase contribution

Modularization significantly impacts the construction phase, increasing efficiency in shipyard facilities. Standardizing modules and interfaces reduces engineering and installation time, further benefitting the construction process.

Operation phase contributions

Modularization facilitates modifications driven by new regulations or technological advancements. It simplifies component replacement, minimizing downtime and operational disruptions. The approach also supports maintenance strategies based on component rotation, similar to aviation practices, enhancing reliability and lifespan.

2.4. Methodical support for modular design

To use modularity as tool to support early stage of the design, the modules first have to be defined. Research done by [Smit, 2019] shows different approaches for methodical support for modular design.

2.4.1. Methods to support modular design

According to [Stjepandić et al., 2015], it is important to articulate on physical assembled aspects, to make modularity manageable and understandable. Therefore various methods to support modular design are developed. The methods are stated by [Stjepandić et al., 2015], [Fuchs and Golenhofen, 2019] and [Jiao et al., 2007]. These methods are listed by [Smit, 2019] as seen below:

- **Axiomatic design** [Suh and Suh, 2001]: Systematically analyzes the transformation of customer needs into functional requirements.
- **Function modeling or heuristic approach** [Stone et al., 2000]: Defines modules within a product's architecture by clustering components or functions.
- **Design Structure Matrix (DSM)** [Malmström and Malmqvist, 1998]: Defines modules within a product's architecture by clustering components or functions.
- **Modular Function Deployment (MFD)** [Erixon and Ostgren, 1993][Erlandsson et al., 1992]: Identifies modules within a modular platform development project.
- **Variant mode and effects analysis** [Bergman et al., 2009]: Illustrates the effects of product variants and identifies opportunities for cost reduction.
- **Five-step algorithm**[Hölttä et al., 2003]: Groups and creates a dendrogram to identify common modules across products.

To apply methodical support for modular design on fully custom luxury yachts, Research conducted by [Smit, 2019] shows Modular Function Deployment to be the most suitable. Although other methods could be benefitial for super yachts as well. The assumption that MFD is the most suitable is based on the work of [Smit, 2019] and a limited research time frame. MFD is designed for modular platform development projects. Although luxury yachts are fully custum they can still benefit from a modular approach, expecially if there are certain components and features that can be seen as common elements across different yacht models. These custom luxury yachts can be seen as platforms with fabricate to fit modularity applyed to their modules. With the use of MFD the modular components can be identified, giving the opportunity to gain efficiency from modularity while keeping fully customization.

2.4.2. Modular Function Deployment (MFD)

Modular Function Deployment (MFD) is a strategic methodology aimed at enhancing product development by breaking down systems into modular components that meet customer demands. This approach focuses on improving flexibility, scalability, and alignment with business objectives, making it a valuable tool for companies looking to boost product development, enhance customer satisfaction, and gain a competitive market advantage[Lange and Imsdahl, 2013].

Key stakeholders

MFD considers several stakeholders throughout different lifecycle phases, addressing various areas of responsibility and expertise. Following [Jiao et al., 2007] the three primary stakeholders are:

- **Voice of Customer:** Represents the sales and marketing departments, focusing on the quality and cost of the module from the customer's perspective.
- **Voice of Engineer:** Involves engineering, manufacturing, quality, supply chain, and aftermarket departments to ensure that modules fit together cocrrectly and function as intended.
- **Voice of business:** Reflects the strategic intent of the business, involving shareholders or corporate officers who decide on the crucial value disciplines for the product and overall business success.

Modular Function Deployment promotes a definition of modularity that reflects all three voices by stating: *A module is a functional building block with specified interfaces, driven by company-specific* *reasons.[Erixon, 1998]*

Product Management Map (PMM)

To organise the Modular Function Deployment for a fully custom modular design method; information, data and knowledge has to be gathered involving the design phase of a luxury yacht. This will be illustrated in a collection of matrices known as the Product Management Map (PMM) shown in figure 2.3.

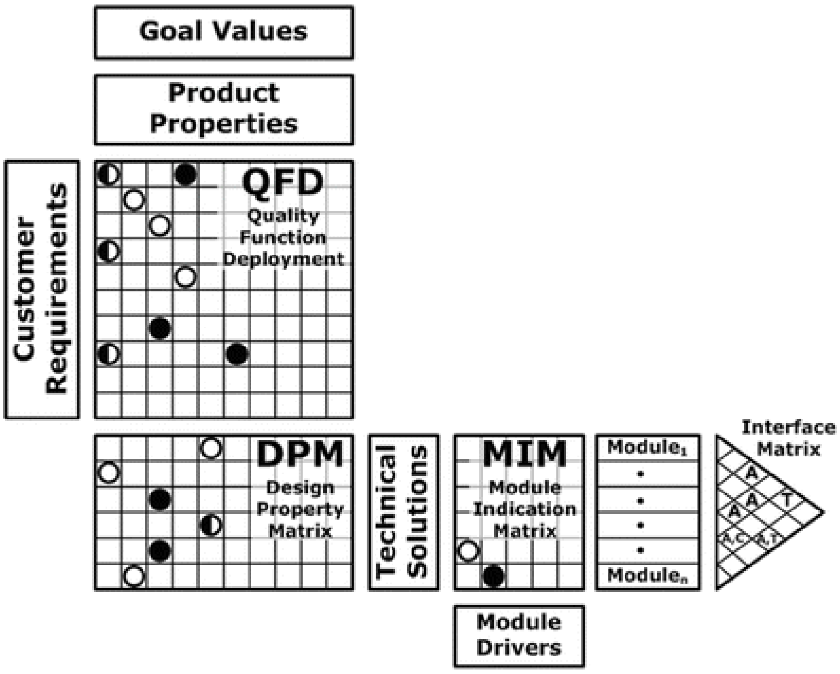


Figure 2.3: Product Management Map [Lange and Imsdahl, 2013]

The Product Management Map consist of the Quality Function Deployment (QFD), the Design Property matrix (DPM), Module Indication Matrix (MIM) and at last the Interface Matrix. Figure 2.4 shows a road map following the Modular Function Deployment consisting the stakeholders and the matrices of the Product Management Map.

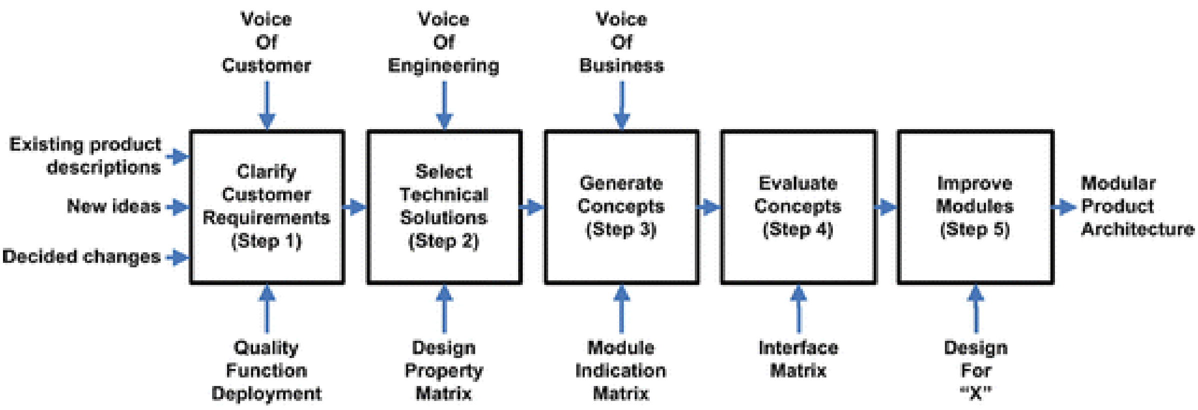


Figure 2.4: Modular Function Deployment roadmap [Lange and Imsdahl, 2013]

The Product Management Map starts with the Quality Function Deployment matrix. This matrix shows the requirements of the stakeholders against the functions. The Design Property Matrix subsequently shows these functions mapped to the systems. Therefore the DPM shows the solution domain with its system requirements. Next matrix is the Module Indication Matrix that shows the system and its functions against the modularity drivers. The MIM maps each function carrier against a modularity driver, it discuses and ranks these with a to be decided rating scale. Finally the module concepts obtained from the MIM matrix are evaluated based on how the modules can be physically joined using module interfaces. This happens in the Interface matrix using the following physical interfaces [Dobberfuhl and Lange, 2009]:

- **Attachment interface**: The physical connection that puts the peaces together and connects them.
- **Transfer interface**: Provides the ability to transfer power and media between modules.
- **Command & control interface**: The communication between different modules.
- **Spatial interface**: Gives boundary's between modules based on spatial location and the volume a module can occupy.

Module drivers

In the early development of Modular Function Deployment, a research into companies who promoted their product as modular was done by [Östgren, 1994]. The aim of this research was to find the reasons of the choices made by product designers creating their modules. This resulted in twelve heuristics named "Module Drivers". The aim of the module drivers is to cover the entire product lifecycle from introduction to growth, maturing and decline[Lange and Imsdahl, 2013]. The module drivers give every stakeholder a voice, also known as "voices of X" and therefore gives them a way to show their particular strategic intent. Figure 2.5 shows a example of module drivers and their voices, but because module drivers are generic heuristics , a project team can introduce new and modified heuristics that apply to their specific company.

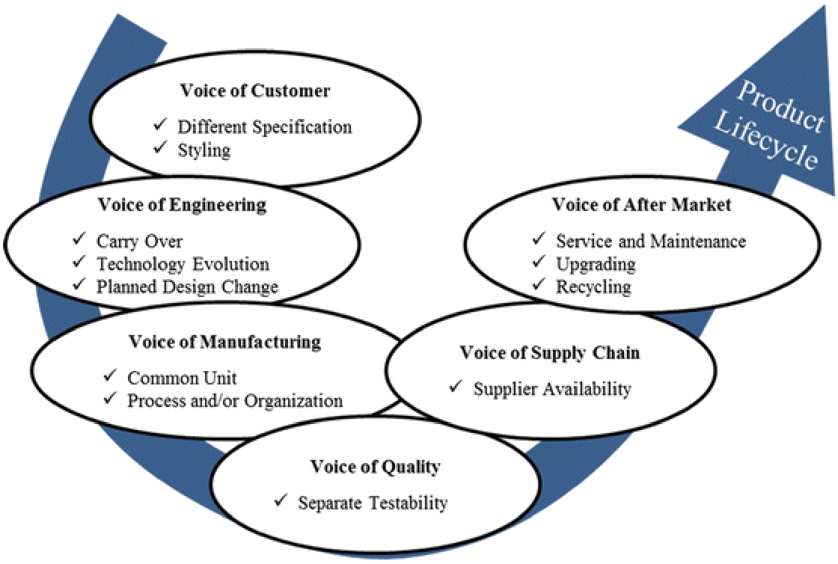


Figure 2.5: Module drivers along a product lifecycle stream. [Lange and Imsdahl, 2013]

3

Feadship Design Process

The aim of this chapter is to map out the design process that takes place at Studio de Voogt. By clearly mapping the current design process, it can be investigated where the opportunities for enhancing the design process can be found. The current Feadship design process is explored through a combination of literature research and interviews in collaboration with Sr. Designer Ruud Bakker and Head of Design Tanno Weeda.

3.1. Design workflow analysis

3.1.1. Interview Ruud Bakker (Sr. Designer)

Collaboration with Ruud Bakker (Feadship Sr. Designer) has been conducted to get a good understanding of the detailed workflow of the current design process. Aim of the interview is to get a clear view of the design process, understand the workflow, and identify the specific steps throughout the desing process.
To clarify the workflow of the design process, it will be followed chronologically. Subsequently, with the information gathered from the interview, the various phases within the design process will be substantiated and clarified. in figure 3.1 a flowchart is shown that visualises the design workflow.

Initial client interaction

The design process begins with the initial contact between the owner team and the shipyard. An account manager is assigned to the intended new owner, and ideally, the designer is present at this first meeting. The primary goal of this interaction is to gather all necessary information for the design brief as shown in Appendix B. The initial conversation aims to gain a comprehensive understanding of the client's unique wishes and desires for their yacht, encapsulated by the main question: "Why do you want us to build a yacht for you?" The design brief captures these desires as accurately as possible.
During these initial conversations, the client's family situation, business, culture, interests, lifestyle, hobbies, and important values are considered to understand the intended use of the yacht. It is also important to know if the yacht will be used privately, for business, or charters. Preferences for the style and feeling of the yacht are discussed, and any previous experiences with yachts are considered. An estimation of the yacht's dimensions is made based on the client's wishes and budget. Ensuring the client feels comfortable and can convey their wishes is crucial, and these meetings often take place at the client's home or on their current yacht.

Initial sketch and layout creation

Following the initial contact, the designer begins by creating the first sketches of the layout, producing a 2D block layout based on the client's preferences. When the designer initiates these first sketches, they start from a self-determined point. From this point, the designer, drawing on their experience, will make numerous decisions regarding various design choices. The placement of each space within the arrangement is the result of numerous considerations, with each decision being carefully deliberated. Sketches of the exterior and interior are also created to convey the overall aesthetic and atmosphere of the yacht to the client. It is crucial to present these initial sketches to the client promptly to sustain their interest, particularly given that many design firms may be competing for their attention.

Collaboration with naval architect and specialist

Together with the initial sketches and layouts, collaboration with the Naval Architect begins. The Naval Architect ensures the yacht's dimensions and draft are correct, along with its resistance, propulsion, and stability. To verify that the designer's plans meet all technical requirements and regulations, a Specialist reviews and evaluates these designs. This collaboration ensures the technical feasibility and regulatory compliance of the initial designs.

Feedback and iteration

After the initial designs are ready, they are presented to the owner by the account manager, and feedback is received. The designer adjusts the design based on this feedback, resulting in several iterations. Throughout these iterations, the Naval Architect and the Specialist remain involved to ensure the technical feasibility of the design. This iterative process continues until the design and layout meet the expectations of both the account manager and the owner.

3D design and detailed rendering

During the design process, the 2D layout drawings are converted into 3D designs, and detailed renderings are produced. These renderings are shown to the client for further feedback. If the client is satisfied with the 3D designs, the Layout Designer uses CAD programs to develop the design layout in detail. This detailed development ultimately forms the contract general arrangement.

Final design development and contract signing

The detailed development culminates in the creation of the contract general arrangement, which is signed during contract signing. Once the contract is signed, the design is handed over to the shipyard, and detailed engineering can begin. This marks the transition from the design phase to the construction phase, ensuring that the client's vision is accurately realized in the final product.

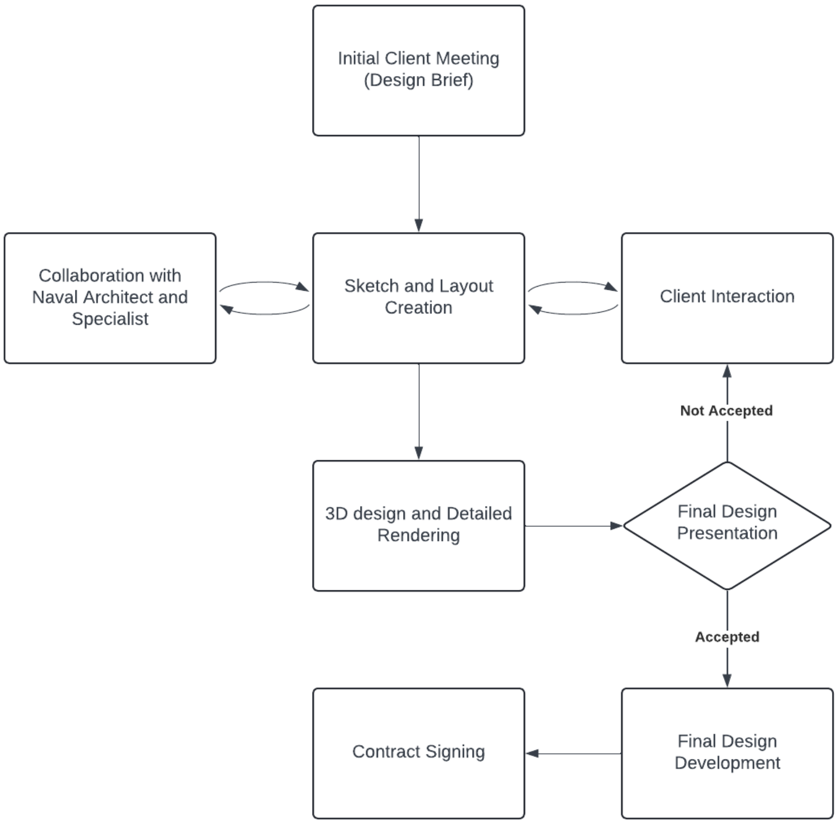


Figure 3.1: Design WorkFlow

3.1.2. Interview Tanno Weeda (Head of Design)

Collaboration with Tanno Weeda (Feadship Head of Design) has been conducted to understand the aim of innovation within the Feadship design process. As Head of Design, Tanno Weeda is responsible for the design studio and its development.

As clearly visible in the studio process from Figure 3.2, the first phase of yacht design in the design studio, consists of the concept part and the technical concept design part. The execution of the concept part is also known as the design process, which is shown in Figure 3.3. This figure shows that the design process consists of a part called "investigate" followed by a part called "create," and ultimately a part called "inspire." In the first part, all the owner's requirements are captured in the design brief. The create phase is the creative phase that goes through multiple iterations. The goal is to perfectly translate all the owner's requirements into a design, minimizing the need for iterations. In the inspire phase, the final design created in the create phase is visualized for the client. Within the scope of the research, it will be investigated what gains can be achieved within the create part. Here, both time savings and gains in creativity are considered very important. The entire process from the first conversations with the client to the final design package is visualized in Figure 3.4. The timeline of the various processes is also shown. From Figure 3.4, it is clear that within the studio, most gains can be achieved within the concept design visualization part. However, this part lies outside the scope and is already under construction within the design studio itself. For this research, the focus will be on achieving gains within the first design and concept design sketch phases. Since these processes already take up a small part of the design time, the focus will not be on saving time within these processes but on enhancing creativity in the design process. The time savings that will still be made by applying modularity can then be invested in the creative part of the design process. This will ensure that De Voogt can come up with more creative and therefore more successful yacht designs.

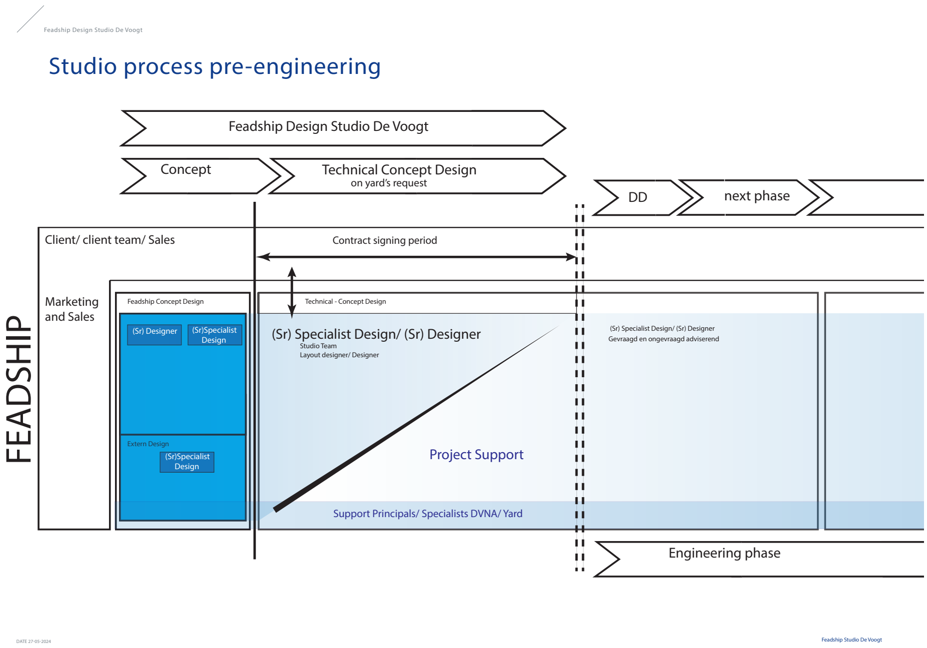


Figure 3.2: Studio process pre-engineering [Feadship]

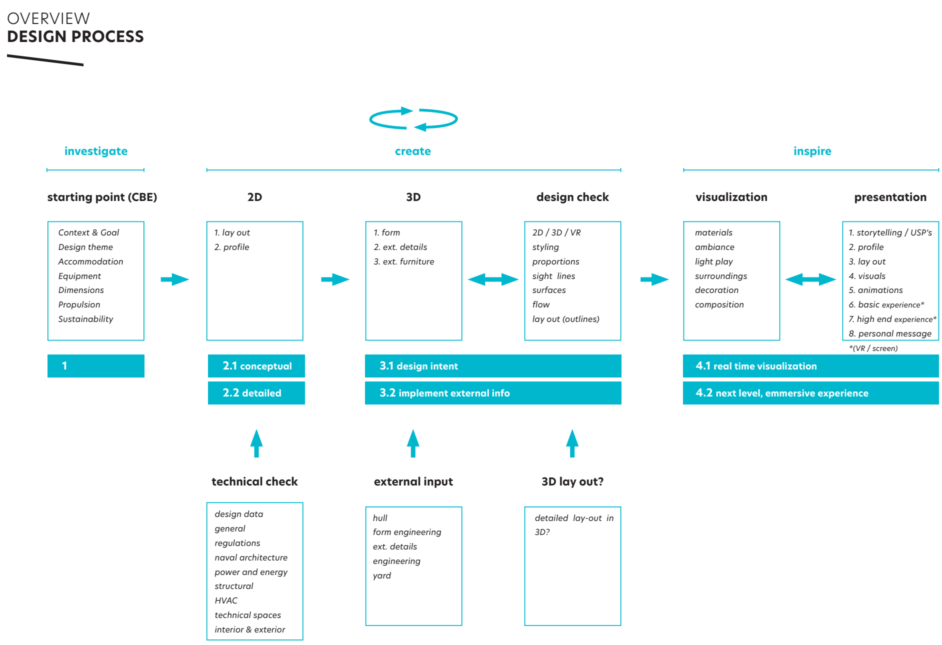


Figure 3.3: Overview Design Process [Feadship]

Figure 3.4: Studio De Voogt Sales Package [Feadship]

4

Implementation of Modular Design method

This chapter discusses the implementation of Modular Function Deployment (MFD) in the design process of a custom yacht. The primary objective of implementing MFD is to achieve higher design efficiency. Additionally, the implementation aims to create more creative freedom within the design process by avoiding a fixed starting point from the designer.

The implementation of MFD is divided into three distinct studies. The first study focuses on determining the theoretical implementation of modularity. The goal is to identify, evaluate, and assess the modules based on the standard requirements for a super yacht and MFD principles.

The second study aims to determine the surface areas of the different modules. The objective of this study is to predict the surface area of each module using input parameters derived from the design brief. Additionally, this study examines which modules exhibit significant variations in surface area predictions, indicating that these modules are not determined by input parameters and thus should be specified as owner-specific inputs in the design brief.

The third study investigates the feasibility of using a design tool, based on the implementation of modules, during the early-stage design phase of a super yacht. This tool supports designers by generating potential yacht arrangements based on inputs from the design brief.

4.1. Theoretical MFD implementation

The implementation of Modular Function Deployment (MFD) in the design process at Feadship could lead to improvements in efficiency, innovation, and client satisfaction. This section provides a detailed step-by-step approach to integrating MFD into Feadship's yacht design workflow, highlighting the benefits at each stage. Subsequently, this section presents a gap analysis comparing the current design method with the MFD-implemented design method. This comparison substantiates the advantages that MFD implementation can offer to the design process. Additionally, a SWOT analysis is conducted to illustrate the strengths, weaknesses, opportunities, and threats associated with the implementation of MFD.

4.1.1. MFD implemented design workflow

Initial Client Interaction

The process begins with comprehensive client interactions to gather detailed requirements. Initial meetings aim to understand the client's needs, preferences, lifestyle, and intended use of the yacht. These meetings often take place at the client's home or on their current yacht to ensure comfort and clarity. The outcome is a well-documented design brief that categorizes and prioritizes requirements into standard and owner-specific. This initial step ensures a precise understanding of client needs, reducing miscommunication and rework, while also building strong client relationships by demonstrating a thorough understanding of their desires.

QFD Matrix Creation
Next, the client requirements together with the specific yacht functions are gathered and mapped using the QFD (Quality Function Deployment) matrix. This matrix establishes relationships between requirements and functions, providing a structured approach to managing requirements. Organizing requirements and functions systematically ensures that all client needs are addressed, while also helping to prioritize features based on their importance to the client, focusing on high-impact areas first.

DPM Matrix Development
The gathered information from the QFD matrix is used within the DPM (Design Property Matrix) matrix to create the technical solutions. By identifying the technical solutions early, this stage saves time and resources. Additionally, it facilitates the possibility to a modular design approach, allowing for easy updates and modifications.

MIM Matrix Utilization
The MIM (Module Indication Matrix) is used to assess the added value of implementing systems as modules. This matrix evaluates technical solutions against module drivers such as customizaion, adaptability, and innovation potential. This helps identify which modules offer the highest value. Focusing resources on modules that provide the greatest benefit optimizes the overall design and encourages exploration of new modules that can offer competitive advantages.

Interface Matrix
With the interface matrix, the module concepts are evaluated based on the way they are joint together. This way the connections between different modules can be determined and used for the module arrangement.

Initial Sketch and Layout Creation
Using the modules from the MFD method, initial 2D block layouts and sketches are created. These layouts integrate all standard and owner-specific requirements and include exterior and interior sketches to visualize the design. The ability to quickly generate initial layouts speeds up the generation of new arrangements. This can help clients by visualizing the yacht arrangement early, improving feedback quality and client engagement.

Feedback and Iteration
The initial sketches and layouts are presented to the client for feedback. Collaborating with the Naval Architect and Specialist ensures technical feasibility and regulatory compliance. The design is revised iteratively based on feedback from all stakeholders. The modular approach ensures that all requirements are considered from the start, minimizing the number of iterations needed. Faster and more precise iterations increase client satisfaction and confidence.

3D Design and Detailed Rendering
Once the 2D layouts are refined, they are converted into detailed 3D designs using CAD software. High-quality renderings are created to present the final design to the client, ensuring the design adheres to the modular structure. Detailed 3D renderings provide a clear and realistic representation of the final product, allowing for early detection of potential issues and reducing costly changes later in the process.

Final Design Development
The detailed CAD drawings are finalized, ensuring all modules are clearly defined and integrated. The contract general arrangement drawings are prepared, forming the basis for the construction phase. This step ensures that all design details are thoroughly documented, facilitating smooth construction. The modular design approach guarantees consistency and accuracy in the final design documents.

Contract Signing and Handover to Shipyard
The final design is presented to the client for approval and contract signing. Detailed design documents are handed over to the shipyard for detailed engineering, marking the transition from design to construction. Detailed and well-documented design increases client confidence in the project, while clear and detailed handover documents ensure a smooth transition from design to construction.

4.1.2. Gap analysis

Initial Client Interaction

In Feadship's current design process, the initial client interaction focuses on gathering comprehensive information about the client's wishes, lifestyle, and preferences to create a detailed design brief. This meeting involves the client, an account manager, and ideally the designer. The MFD approach enhances this step by categorizing client requirements into standard and owner-specific needs from the outset. The gap between the two processes lies in the structured categorization of requirements in the MFD approach, which reduces miscommunication and rework. By adopting this systematic categorization, Feadship could achieve a more precise understanding of client needs, ensuring all aspects are considered early in the process.

Initial Sketch and Layout Creation

Feadship's current process involves creating 2D block layouts and sketches quickly to maintain client interest, based on the gathered requirements. The MFD process, on the other hand, uses modules identified in the Design Property Matrix (DPM) to create these initial sketches, incorporating both standard and owner-specific requirements. The gap here is the use of modular design principles in MFD, which streamlines the sketch creation process and enhances visualization for the client. Feadship could benefit from integrating modular design elements to speed up the sketch and layout creation phase, thus improving client engagement and feedback quality.

Collaboration with Naval Architect and Specialist

Currently, Feadship collaborates with a Naval Architect to ensure the yacht's technical feasibility, and a Specialist to review designs for regulatory compliance. In MFD, this collaboration is enhanced by using the Module Interface Matrix (MIM) to evaluate technical solutions against module drivers like cost and feasibility. The gap between the processes is the structured approach to assessing technical solutions in MFD, optimizing resources by focusing on high-value modules. Feadship could improve their collaboration process by adopting the MIM, ensuring a more efficient and innovative design evaluation.

Feedback and Iteration

The current process involves iterative design presentations to the client, with adjustments based on feedback, ensuring technical feasibility throughout. The MFD approach streamlines this by ensuring all requirements are considered from the start through modular design, reducing the number of iterations needed. However, the current process excels in its flexibility to adapt to unique client demands through repeated feedback loops, which can sometimes better capture the nuanced preferences of the client. The gap lies in balancing efficiency with the ability to refine and adapt designs thoroughly. Feadship should look to maintain their high flexibility while integrating the efficiency gains from the MFD process, potentially by establishing a hybrid feedback mechanism that leverages both detailed initial modular planning and iterative refinement.

3D design and Detailed Rendering

Once 2D layouts are finalized, Feadship converts them into detailed 3D designs for client approval, helping to visualize the final product and identify potential issues. The MFD approach maintains a modular structure in 3D designs, ensuring consistency and early detection of issues. The gap is the adherence to modular design in MFD, which reduces costly changes later. By adopting a modular approach, Feadship could ensure a more streamlined and precise 3D design phase, reducing potential issues and costs.

Final Design Development and Contract Signing

Feadship's current process concludes with the creation of the contract general arrangement, leading to detailed engineering and construction. In MFD, final CAD drawings are clearly defined and integrated into the modular structure, ensuring thorough documentation. The gap is in the detailed and modular documentation in MFD, facilitating a smoother transition to construction. Feadship could improve their final design development by incorporating modular documentation, enhancing accuracy and reducing potential issues during construction. However, Feadship's current process may offer more detailed bespoke solutions tailored precisely to individual client demands, which can sometimes be more difficult to achieve in a strictly modular framework. Ensuring that bespoke client preferences are not lost in the modular approach is essential, and a balanced integration of detailed customization within modular documentation could offer the best of both worlds.

4.1.3. SWOT analysis

Strengths:

- **Efficiency in Design Process:** MFD streamlines the design process by enabling quicker generation of initial sketches and layouts through pre-defined modules. This leads to faster project initiation and reduced time-to-market compared to the traditional design process.
- **Reduced Rework and Miscommunication:** By categorizing client requirements into standard and owner-specific requirements in the early design process, MFD minimizes miscommunication an reduces the need for rework.
- **Optimized Recource Allocation:** The structured approach of MFD, using tools like the Module Interface Matrix (MIM), ensures that resources are focused on the most impactful areas. This optimization helps in effectively managing time and budget constraints.
- **Enhanced Consistency and Accuracy:** Modular designs ensure uniformity and precision across different projects. This consistency reduces the likelihood of errors and discrepancies, leading to a smoother transition from design to construction.
- **Improved Client Engagement:** By providing clear and realistic visualizations early in the process, MFD enhances client satisfaction. Clients can see how their requirements are systematically addressed, leading to more informed decision-making and higher confidence in the final product.
- **Streamlined Feedback Process:** The modular approach facilitates easier and quicker iterations, as changes can be made to individual modules without affecting the entire design. This leads to a more efficient feedback and iteration cycle, reducing the overall design timeline.

Weaknesses:

- **Potential Loss in Customization:** The structured nature of MFD might limit the ability to create highly bespoke designs tailored to individual client preferences, which is a key strength of Feadship's current process.
- **Initial Implementation Costs:** Integrating MFD requires significant investment in training, new software, and possibly restructuring parts of the design workflow, resulting in higher initial costs and resource allocation.
- **Complexity in Transition:** Shifting from the traditional design process to MFD can be complex and time-consuming. This transition may disrupt existing workflows and require significant changes in how design projects are managed.
- **Learning Curve:** Designers and engineers will need time to adapt to the new MFD approach, which could temporarily affect productivity and efficiency as they become used to the new processes and tools.

Opportunities

- **Improved Design Flexibility:** MFD allows for more flexible design iterations by enabling easy adjustments to specific modules without overhauling the entire design, this can be an improvement over the current process that might require more extensive changes for adjustments.
- **Enhanced Collaboration:** The structured nature of MFD facilitates better collaboration among different teams (designers, engineers, specialists) as modules provide clear interfaces and responsibilities, leading to smoother workflows and integration of various subsystems.
- **Client Transparency:** The modular approach provides a clear, organized framework that can be more easily communicated to clients, helping them understand the design process and see how their requirements are being systematically addressed. By applying modularity, multiple options can be presented to the client more easily. This approach leads to the client becoming more quickly and deeply involved in decision-making, rather than being confronted with a singular vision from the designer, which would otherwise be justified solely through accompanying explanations.
- **Future-proofing Designs:** Modular designs can be more easily updated in the future, providing clients with yachts that can adapt to new technologies or changing needs over time. This contrasts with the current process where significant redesigns might be necessary for upgrades.

Threats:

- **Resistance to Change:** Designers and stakeholders accustomed to the traditional design process might resist the transition to MFD, affecting morale and productivity.
- **Market Perception:** Clients who value the highly bespoke and personalized designs offered by Feadship's current process may perceive modular designs as less unique, which could impact Feadship's reputation for luxury and customization.
- **Technical Challenges:** Ensuring that modular designs meet all technical and regulatory requirements can be challenging, particularly in the highly specialized field of yacht design.
- **Competitor Response:** Competitors may quickly adopt similar modular approaches, diminishing the competitive advantage gained from MFD integration.
- **Implementation Risks:** Any issues or failures during the implementation phase could disrupt ongoing projects, leading to delays and potential client dissatisfaction.
- **Previous failures:** Within De Voogt Naval Architects, several studies have been conducted prior to this research regarding the application of modularity within Feadship. Thus far, there has been little actual implementation following these various research efforts. The fact that none of these attempts have resulted in a successful application of modularity indicates a risk of failure for new research endeavors.

4.2. Module indication and evaluation

This section describes how the practical implementation of Modular Function Deployment (MFD) is carried out through the Product Management Map (PMM) and the Interface Matrix. The primary objective of applying MFD is to identify and evaluate potential modules during the early stage design phase of a yacht. This involves analytically determining the relative weighting for modularity across all systems and assessing the relative weighting of various modularity drivers. Additionally, the relationships between different systems are identified, which are then used to establish arrangement conditions for the layout of a yacht.

4.2.1. Product Management Map

To visualize how the MFD method operates, the Product Management Map shown in figure 2.3 is utilized. The PMM maps out the relationships between requirements, functions, systems, and module drivers and analyzes the connections between the different systems. Using this method, it becomes possible to illustrate how standard and owner requirements determine suitable modules for a custom yacht. Relationships between various requirements, functions, systems, and module drivers are depicted. These relationships, together with the findings from the interface matrix, can be used to identify suitable modules, their added value, and the interrelations between different modules within the layout of a yacht. The PMM applied for the design process of Feadship is shown in Appendix C but will be further discussed and clarified.

Quality Function Deployment

The QFD matrix applyed for the standard requirements of a luxury yacht shows the relations between requirements and functions. To effectively translate the problem domain into the solution domain, the standard requirements are systematically allocated to specific functions. Each technical solution is exclusively aligned with a particular function to adhere to the principles of modularization[Smit, 2019]. The requirements were based on research into the De Voogt design brief as seen in Appendix B and the examination of existing general arrangements. Based on this research, the standard requirements for a superyacht are assumed with the help of known Feadship requirements. This is done by implementing a literature study and subsequently substantiated by conducting a study of existing GAs. The standard requirements are formulated as follows:

- **Estended Stays Onboard:** Provide facilities to ensure comfortable living for several consecutive days.
- **Convenient Meal Preparation:** Offer the capability to prepare and enjoy high-quality meals onboard.

- **Comfortable Sleeping Environment:** Ensure a restful and private sleeping experience during extended stays.
- **Continuous Hygiene Maintenance:** Provide onboard laundry capabilities for cleanliness during long trips.
- **Comfortable Living Conditions:** Ensure comfortable living conditions by providing among others climate control and air quality control.
- **Safe Living Conditions at Sea:** Provide stability and buoyancy to ensure safety in varying sea conditions.
- **Self-Reliance During Emergencies:** Equip the yacht with systems for emergency protection and response.
- **Structural Safety:** Ensure construction integrity for safety and long-term durability.
- **Safe Navigation:** Ensure safe and precise navigation for uninterrupted travel.
- **Independence in Movement:** Provide efficient propulsion and manoeuvring for autonomy at sea.
- **Secure Mooring:** Equip the yacht for secure docking and anchoring in various conditions.
- **Luxurious Accommodation:** Offer luxurious, private accommodations for guests.
- **Entertainment and Leisure:** Provide recreational facilities for enjoyment and relaxation onboard.
- **Crew Comfort and Efficiency:** Ensure operational efficiency and living comfort for the crew.
- **Long-Term Self-Sufficiency:** Provide capabilities for extended operations without external support.
- **Safety, Security and Compliance:** Ensure the yacht meets safety standards and regulatory requirements.
- **Luxury Aesthetics, Craftsmanship & tailored Design:** Offer high-quality, luxurious interior and exterior design, and Allow for custom design to meet the owner's personal preferences and needs.

Subsequently, the requirements can be translated into functions [Smit, 2019]. The functions determined for the QFD matrix are:

- **Meal Preparation and Dining**
- **Comfortable Sleeping**
- **Provide Laundry Facilities**
- **Provide Sanitary Facilities**
- **Climate Control and Air Quality**
- **Stability and Buoyancy**
- **Emergency Self-Protection**
- **Construction Integrity**
- **Safe Navigation**
- **Efficient Propulsion and Maneuvering**
- **Secure Docking and Anchoring**
- **Luxurious Accommodation**
- **Recreational Facilities**
- **Operational Efficiency for Crew**
- **Extended Operation Capabilities**
- **Safety and Regulatory Compliance**

By presenting the requirements and functions against each other, as shown in the QFD in figure 4.1, and assigning scores, the connection between these different requirements and functions can be investigated. Based on strong connections, it can then be determined whether these connections together can form a system.

QFD	Meal Preparation and Dining	Comfortable Sleeping	Provide Laundry Facilities	Provide Sanitary Facilities	Climate Control and Air Quality	Stability and Buoyancy	Emergency Self-Protection	Construction Integrity	Safe Navigation	Efficient Propulsion and Maneuvering	Secure Docking and Anchoring	Luxurious Accommodation	Recreational Facilities	Operational Efficiency for Crew	Extended Operation Capabilities	Safety and Regulatory Compliance
Extended Stays Onboard	●	●	◉	○	○							○	○	○	●	○
Convenient Meal Preparation	●		○												●	
Comfortable Sleeping Environment		●	●	●	●	○						●				
Continuous Hygiene Maintenance			●	●								○		●	○	○
Comfortable Living Conditions	○	○	◉	○	●	◉						◉				
Safe Living Conditions on Sea						●	●	●	◉							●
Self-reliance During Emergencies							●							◉		◉
Structural Safety								●								●
Safe Navigation						◉			●	●				◉		
Independence in Movement						◉			●	●				◉		
Secure Mooring						○				●	●			●		
Luxurious Guest Accommodation		●	◉	◉	●							●				
Entertainmend and Leisure	○		○										●	○		
Crew Comfort and Efficiency		●	◉	○										●	○	◉
Long-Term Self-Sufficiency	◉		○											○	●	
Safety, Security and Compliance								●	◉					◉	○	●
Luxury Aesthetics, Craftsmanship & Tailored Design	◉	●										●				

Figure 4.1: QFD Matrix

The assessment of the connections is done through a point driven system. No circle indicates no connection, a white circle indicates a weak connection, gray indicates an average connection, and black indicates a strong connection. Notably, the QFD matrix logically shows that requirements translated into functions have a strong connection with each other. Additionally, various clusters of points are visible in the matrix, which were investigated to see if they could lead to potential systems. From the QFD, it can be concluded that all requirements can be linked to existing functions. This is important to ensure that, through the DPM, all requirements are captured by the systems. With help of the QFD matrix and gathered knowledge from literature research assuptions can be made on suitable systems for yacht design. These assumptions have been made together with knowledge from [Smit, 2019] and Feadship specialists.

Design Property Matrix

The design property matrix, as shown in figure 4.2, is a framework in which functions are mapped against the systems. This is done to ensure that the designated systems meet the specified requirements and effectively contribute to the functionality and performance of the final product. This allows engineers to gain insight into which characteristics have a significant impact on specific requirements and functions.

Upon studying the detailed DPM, it becomes apparent that all functions have strong relationships with at least one system. This means that all functions, and thereby all standard requirements of the yacht, are covered by the designated systems. It is important to conclude this because meeting all requirements is crucial for a complete yacht design. Since all functions are covered by the systems, further work can proceed with the designated systems, and the added modular value can be determined.

Meal Preparation and Dining	Comfortable Sleeping	Provide Laundry Facilities	Provide Sanitary Facilities	Climate Control and Air Quality	Stability and Buoyancy	Emergency Self-Protection	Construction Integrity	Safe Navigation	Efficient Propulsion and Maneuvering	Secure Docking and Anchoring	Luxurious Guest Accommodation	Recreational Facilities	Operational Efficiency for Crew	Extended Operation Capabilities	Safety and Regulatory Compliance	DPM
				◉					●	◉			◉	○	○	Power System
								●	●	◉						Propulsion System
					○					●		○	◉	○		Mooring & Anchoring System
◉	●		○	●							●	◉	○		○	HVAC System
◉		◉	○						◉	○			◉	●	○	Extended Operations System
				◉	○	○		●	●	●			◉		◉	Wheelhouse System
												●	◉			Tender Garage & Toy Storage System
○		○	◉									●	○			Beachclub system
●		○	◉	◉							◉		◉	◉		Dining & Galley System
○	●	◉	●	◉							●					Owner Accommodation System
	●	◉	●	◉							●					Guest Accommodation system
	●	◉	●	◉								◉	○		○	Crew Accommodation System
◉		●				◉				●	○		●	○	◉	Crew Operations System
●	◉		○	○	●	●	●	○	◉				○	○	●	Safety and Compliance System

Figure 4.2: Design Property Matrix

Systems

Based on the gathered information from the QFD matrix and the DPM matrix, the systems have been identified. These systems are required to meet all the requirements that have been established as standard requirements for a superyacht. The follwing systems are established:

- **Power System**
  - Contains the complete engine room and associated technical rooms such as the switchboard room.
- **Propulsion System**
  - Propulsion room contains only the bow thruster. This is because within the scope only electric thruster pod propulsion is considered which takes up negligible space within the ship.
- **Mooring & Anchoring System**
  - System includes the mooring platform which is located in the bow of the yacht.
- **HVAC System**
  - Contains all HVAC spaces within the ship.
- **Extended Operations System**
  - Extended operations system consists of all stores for consumables as well as the provisioning bay. Dry, cold, freezer stores are included in this as well as stores for other supplies that may run out over time. Tanks are not included in this as they are located in the tank deck and therefore outside the scope.
- **Wheelhouse System**
  - The wheelhouse system contains the bridge, captain's office and captain's cabin.
- **Tender Garage & Toy Storage System**
  - The location which is equipped with hatches in which tenders and other water toys are stored in combination with further storage spaces for toys.
- Beachclub System
  - The entire open area located at the rear of the ship on the lower deck with opening hatches which is designated as a beach club.
- **Dining & Galley System**

- The combined system in which the main dining room and the adjacent galley are joined together.

- **Owner Accommodation System**
  - The enclosed area within the ship reserved for the owner. This includes the owner cabin, bathrooms, offices and personal lounges.
- **Guest Accomodation System**
  - guest accommodation includes all guest cabins and associated bathrooms as well as VIP cabins with associated bathrooms and possible offices.
- **Crew Accomodation System**
  - Spaces reserved as sleeping quarters for crew, officers, security and maids.
- **Crew Operations System**
  - Spaces that are closed off from guests in which the crew can perform their duties as well as spaces where the crew can be in their free time. This includes mess, pantry, crew hymn, linen store, laundry and deck lockers.
- **Safety & Compliance System**
  - Contains the spaces on board that are assigned to mandatory safety systems. These include within the scope of the regulations the hi-fog system and the emergency generator.

Module drivers and MIM

In the Module Indication Matrix, the systems are mapped against the modularity drivers. The MIM aims to identify and evaluate potential modules within a product design. The MIM helps in creating a modular product structure that promotes flexibility, scalability, and efficiency throughout the entire lifecycle of the yacht. For the situation of a custom luxury yacht, the standard modularity drivers [Ericsson and Erixon, 1999] have been adjusted to appropriately assign stakeholders that cover the entire product lifecycle of a luxury yacht. To quantify the impact of the modularity drivers on the technical solutions, a questionnaire was developed to support the decisions made in the MIM matrix. This questionnaire was distributed as a survey and completed by specialists in the field within ”De Voogt Naval Architects.” Despite the fact that the survey was conducted among a small number of specialists, it is assumed that their expertise is convincing enough to determine the influence of these module drivers on the systems. This assumption is made due to the lack of access to multiple specialists within the possibilities of the research. The module drivers specifically developed for application to custom luxury yachts are shown below. These module drivers outline the reasons for a system to be modular. All of these reasons have been formulated based on the stakeholders who influence a yacht throughout its entire lifecycle. Additionally, the module drivers are accompanied by an explanation and the corresponding question from the questionnaire that was presented to the specialists.

Voice of Customer Experience

1. Customization and Personalization
   - **Explanation:** Prioritizes bespoke design and features tailored to the owner’s preferences and lifestyle, ensuring a unique and personalized yacht experience.
   - **Question:** To what extent is this part influenced by the need for custom design options? <strongly, fairly, to some extent, not>
2. Luxurious Aesthetics
   - **Explanation:** Focuses on high-end, brand-driven appearance and unique styling in both interior and exterior design, reflecting the essence of luxury.
   - **Question:** Is this part significantly influenced by trends and fashion in terms of form and/or color? <strongly, fairly, to some extent, not>

Voice of Engineering Excellence

3. Advanced Technology Integration

- **Explanation:** Incorporates the latest marine and luxury technologies to enhance performance, safety, and convenience on board.
- **Question:** To what extent does this part need to integrate with cutting-edge technology?<strongly, fairly, to some extent, not>

4. Modular Adaptability
   - **Explanation:** Ensures the yacht can easily accommodate planned design changes and upgrades, facilitating modifications and enhancements throughout its lifecycle.
   - **Question:** How important is it for this part to be easily upgradable? <all, most, some, none>

Voice of Operational Efficiency

5. Efficient Production and Assembly
   - **Explanation:** Streamlines the manufacturing process to maintain high standards of quality and precision, ensuring timely delivery.
   - **Question:** Are there significant advantages to separating this part into a module to streamline production? <strong, medium, some, no>
6. Crew Efficiency and Welfare
   - **Explanation:** Optimizes crew facilities and operations for efficiency and the well-being of the crew, contributing to overall operational excellence.
   - **Question:** To what extent does modularizing this part enhance crew operational efficiency? <greatly, moderately, slightly, not at all>

Voice of Quality Assurance

7. Independent Testing and Quality Assurance
   - **Explanation:** Ensures separate testing of critical systems and components to guarantee reliability and safety before integration into the yacht.
   - **Question:** Does separate testing of this part improve overall quality assurance? <greatly, moderately, slightly, not at all>
8. Safety and Regulatory Compliance
   - **Explanation:** Complies with all relevant maritime safety and environmental regulations, ensuring safe operation and minimizing environmental impact.
   - **Question:** Does this part need to be modular to comply with safety regulations?<strongly, fairly, to some extent, not>

Voice of After-Market Support

9. Serviceability and Maintenance
   - **Explanation:** Facilitates easy access and straightforward maintenance for crew and service teams to maintain operational efficiency and luxury standards.
   - **Question:** Will making this part modular significantly ease service and maintenance tasks? <all, most, some, none>
10. Upgradability and Flexibility
    - **Explanation:** Allows for easy upgrades and customization over time to meet evolving owner preferences and technological advancements.
    - **Question:** Can future upgrades be simplified if this part is modular? <all, most, some, none>

Voice of Sustainability

11. Sustainability and Environmental Compliance
    - **Explanation:** Ensures the yacht meets or exceeds environmental regulations and standards, including waste management and emissions controls, promoting eco-friendly practices.

- **Question:** Does modularity in this part contribute to meet environmental standards and sustainability goals? <strongly, fairly, to some extent, not>

12. Long-Term Durability
    - **Explanation:** Focuses on the use of high-quality, durable materials and construction techniques to ensure the yacht remains in excellent condition over its lifespan.
    - **Question:** How much does modularity contribute to the long-term durability of this part? <greatly, moderately, slightly, not at all>

MIM	Customization and Personalization	Luxurious Aesthetics	Advanced Technology Integration	Modular Adaptability	Efficient Production and Assembly	Crew Efficiency and Welfare	Independent Testing and Quality Assurance	Safety and Regulatory Compliance	Servicability and Mainenance	Upgradability and Flexibility	Sustainability and Environmental Compliance	Long-Term Durability
Power System	○		●	○	◉	○	◉	○	◉	◉	○	○
Propulsion System	○		●		◉	○	○	○	○	●	○	○
Mooring & Anchoring System					○	○	○	◉		○		
HVAC System	○		◉		○		◉		○	○	◉	○
Extended Operations System	◉		○						○		◉	
Wheelhouse System	◉	○	○	○	○	◉	◉	◉	○	○	○	◉
Tender Garage & Toy Storage System	●	○	◉	○	●	○	○	○	○	◉		◉
Beachclub system	●	●	○		○					○		
Dining & Galley System	◉	○				○			○		○	○
Owner Accommodation System	◉	◉	○	○	○		○					
Guest Accommodation system	◉	◉	○	○								
Crew Accommodation System						○		○	○			
Crew Operations System	○		○		○	○	○	○	○	○	○	○
Safety and Compliance System				◉		○	◉	●	○	○		

Figure 4.3: Module Indication Matrix

The results of the questionnaire are gathered in the MIM matrix and shown in the table in figure 4.3. To give a better visualisation of the results from the MIM, the outcomes are plotted in a table against the percentual relative weight. The relative weight is a way to express the contribution of every individual component as a percentage of the total. This will provide a clear overview of the answers from the questionnaire survey collaborated with the specialist. The table shows the beneficial relevance of every system for being modular. Table 4.4 shows the relative weighing of systems, therefore showing the added value of being modular for every system. Table 4.5 shows the relative weighing of modularity drivers, therefore showing the impact of every module driver for a yacht design.

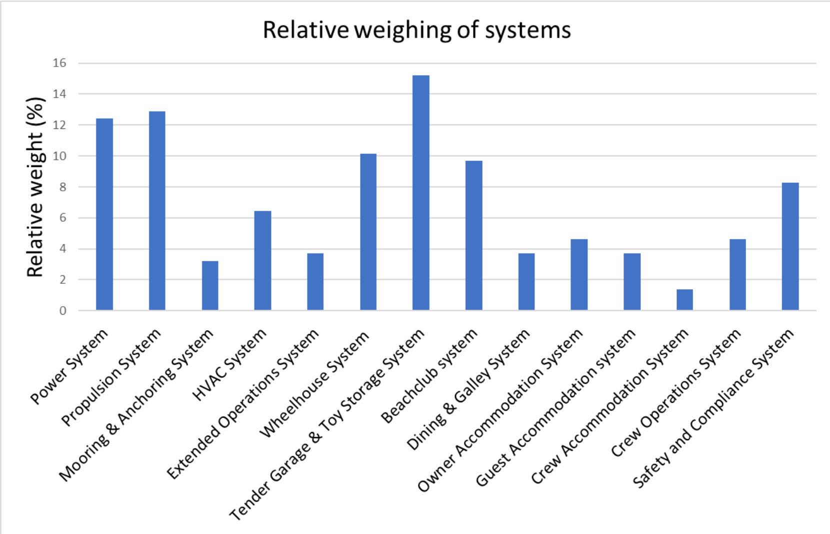


Figure 4.4: Relative weighing of systems

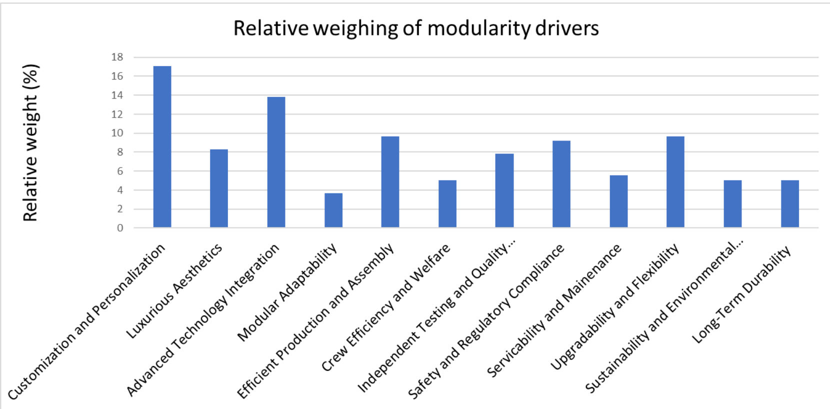


Figure 4.5: Relative weighing of modularity drivers

The results plotted in figure 4.4 illustrate the extent to which the systems benefit from modularity advantages. The data from figure 4.5 further indicates which of the module drivers have the greatest impact on a yacht, in other words, which form of modularity provides the most benefit. Based on the conducted research and the expertise of the specialists, following the relative weighing graphs, it can be concluded that modularity adds value to a superyacht. This conclusion is supported by the finding that all systems derive added value from modularity principles, as shown in figure 4.4 by the fact that all systems are assigned a weight percentage. The Tender Garage system scores exceptionally high, achieving points for every module driver except for sustainability and environmental compliance. This indicates that the Tender Garage system would benefit from being approached as a module throughout the entire lifecycle of the yacht. Another noteworthy result the crew accommodation system which achieved the lowest score. Initially, it was expected that, due to its repetitive nature, simplicity, and continuity across different yachts, the crew accommodation system would be highly suitable as a module. However, according to MFD principles, this turns out to be less true. MFD reveals that modularity is not just about building in blocks; it encompasses much more throughout the entire lifecycle of a yacht. From this, it can be stated that the visualized results of the MFD study provide a clear picture of the added value of modularity over the yacht's lifecycle. Subsequently, the relative weighting of modularity drivers from Figure 4.5 is analyzed. This analysis identifies which module drivers have the greatest influence on the systems within a yacht. The figure shows that every module driver has some degree of influence on the systems, meaning that modular principles can offer benefits for each driver throughout the yacht's lifecycle.

In addition to the expected high scores for advanced technology integration, efficient production and assembly, and upgradability, there is also a high score for customization and personalization. This indicates that, according to MFD principles, customization and personalization can benefit from modular principles, despite the initial assumption that modularity might limit creativity in the design process. Further research conducted in chapter 5 Determines whether applying modularity can indeed enhance creativity within the yacht design process.

4.2.2. Interface matrix

The interface matrix shows the connections and interactions between different systems. Here, we distinguish Attachment (A), which represents a physical connection that puts the pieces together or connects them physically. Transfer (T) indicates when a transfer of energy occurs between systems. Command & Control (C) determines how the state of a component is communicated or controlled by other components. Spatial (S) determines the boundary between modules, spatial location, and volume of a component. Using the information from the interface matrix, the relationships between the systems can be visualized. This information can be used to support the determination of the topological relations of the modules within the layout of the yacht.

INTERFACE	Power System	Propulsion System	Mooring & Anchoring System	HVAC System	Extended Operations System	Wheelhouse System	Tender Garage & Toy Storage System	Beachclub System	Dining & Galley System	Owner Accommodation System	Guest Accommodation System	Crew Operations System	Crew Accommodation System	Safety and Compliance System
Power System		T	T	T, A	T, C	T, C	T	T	T	T,S	T,S	T,C	T,C,S	T, C
Propulsion System			C			C						C		C,T
Mooring & Anchoring System						C						C		
HVAC System						A,C		A	A	A	A	A, C	A	
Extended Operations System						C			C, T			C		C
Wheelhouse System							C	C				C		C
Tender Garage & Toy Storage System												C		C
Beachclub system									C, T,			C		C
Dining & Galley System										T, C		C		
Owner Accommodation System											S	C		
Guest Accommodation system												C		
Crew Operations System													C	
Crew Accommodation System														C
Safety and Compliance														

Figure 4.6: Interface Matrix

4.3. System Area Prediction tool

The modules identified through MFD research in section 4.2 are further analyzed in this section. This analysis focuses on the minimum space requirements of these various modules within a yacht. The study investigates whether the module areas are determined by standard input parameters from the design brief or if they are influenced by the specific preferences of the owner. Input parameters are defined based on the key characteristics of a yacht, which are established during the initial discussions for the design brief (Appendix B). The research aims to determine whether the various modules are primarily depended by these input parameters or if the owner's preferences play a more significant role.

4.3.1. Module area calculation

To determine the minimum required surface area for each module, known data from the "De Voogt design database" is utilized. A dataset relevant to the scope of the research is compiled using information from the database. All Feadship yard numbers within the 75-110 meter range are included in this dataset. The 75-110 meter range was chosen for several reasons. Within Feadship, this range is identified as the company's future-proof sweet spot, meaning that it is expected that most future designs will fall within this range. Additionally, this range has been maintained because it typically consists of four decks and a tank deck with standing height. For the vessels within this range, the following input information has been collected:

- Length
- Width
- Draught
- Nr. of Crew
- Nr. of Guests
- Design Speed
- Range

This data corresponds to the information typically known after the initial discussions in the design brief.

For all these ships, the surface areas are extracted from the Feadship design database and compiled into an Excel file. Because of the confidential information from the Feadship database, the file isn't included in the Appendix. Within this file all areas corresponding to each of the 14 identified module systems are gathered from the database. The Excel file processes this information, resulting in a dataset that includes not only the input parameters for each yard number (yacht) but also the corresponding surface area for each module within these ships.

From this point on, all calculations and figures will be based on a case study yacht. This case study yacht is a far developed never build Feadship prospect with Design Number 3408 - Bulldog. The input parameters for DN3408 are gathered from the De Voogt Design Database and are:

Length	88	[m]
Width	14,5	[m]
Draught	4	[m]
Nr. of Crew	17	[-]
Nr. of Guests	14	[-]
Design Speed	16	[kts]
Range	5000	[NM]

With the help of supporting AI tool, a Python script is generated that utilizes machine learning techniques to predict the surface area of each module based on the given design parameters. Two different models, Linear Regression and Random Forest, are employed to create as robust as possible predictions. By employing the two different algorithms they can be compared to see which model provides more accurate and reliable predictions for the areas of the modules. The dual approach helps to allow for a comprehensive evaluation of the models strengths and weaknesses.
Linear Regression models the relationship between the input features and the target variables by fitting a linear equation to the observed data. It assumes a linear relationship between the predictors and the response. The simplicity of Linear Regression makes it easy to interpret and implement, and it often serves as a baseline model in predictive modeling tasks [Myers et al., 2012].
Random Forest is an esemble learning method that builds multiple decision trees and merges their results to produce a accurate and stable prediction. Each tree is trained on a random subset of the data, and the final prediction is obtained by averaging the predictions of all trees. Random Forest handles non-linear relationships and interations between variables effectively, making it suitable for complex datasets [Breiman, 2001].

4.3.2. Script Explanation

The script applies machine learning techniques to predict various ship system areas based on the design parameters by using multi-output regression models. The process involves several key steps, including data preprocessing, model training, evaluation, and sensitivity analysis. All steps are chosen to optimize the accuracy and interpretability of the area predictions. The complete script with indeep explanation can be found in Appendix D.

Data loading and preprocessing

The Ship design data from the Feadship database is imported from the confidential Excel file and preprocessed. Zeros are replaced by NaNs to handle missing values appropriately[Little and Rubin, 2019], features and targets are converted to numeric types to ensure data integrity [James et al., 2013], and missing values are imputed using the mean, maintaining dataset size and preventing bias [Schafer and Graham, 2002].

Model training and evaluation

The dataset is split into training and testing sets using an 80/20 ratio. A random state is set to ensure the split is reproducible, which is critical for scientific experiments to allow for verification of results [Kuhn, 2013]. Two models are trained: a multi-output linear regression model and a multi-output random forest regression model. Because of its simplicity and interpretability linear regression is chosen. The linear regression is by training a MultiOutputRegressor on the data. For its robustness and ability to handle non-linear relationships a RandomForestRegressor is trained on the data as well [Breiman, 2001]. The models are evaluated using multiple metrics. MSE, MAE, MAPE, MSPE and R^2 are used, providing a comprehensive performance assessment [Willmott and Matsuura, 2005].

Sensitivity analysis and visualization

Sensitivity analysis is conducted to assess the impact of disturbances in input features on the model predictions. This analysis identifies which features most significantly affect the target outputs, providing valuable insights into the relative importance of different design parameters[Saltelli, 2008]. Results are visualized to compare feature importance across different ship system areas.

4.3.3. Optimal evaluation metric for high variation identification

By using linear regression or bootstrap sampling, predictions can be made regarding the areas of the various modules. However, little can be concluded about these predictions initially, as their accuracy is unknown. To assess this, evaluation metrics are employed. In this research case, predictions are evaluated for their realism by scoring for high explanatory power. Additionally, substantial variability in predictions is examined to identify which systems show significant deviations. Such deviations may indicate the presence of owner-specific requirements, as these areas cannot be predicted solely based on input parameters.

To identify the evaluation metric that best reflects the relative highest variation in yacht area predictions and, therefore, non realistic predictive performance, it is crucial to assess the characteristics and suitability of each metric in the context of regression analysis. The selection of an appropriate metric is critical for identifying areas where input parameters fail to sufficiently predict target values. Below is a detailed evaluation of the metrics used in the model performance assessment:

Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average magnitude of errors in predictions, without considering their direction, presenting the errors in the same units as the target variable. MAE is valued for its interpretability as it represents the average absolute deviation from actual values. However, it does not emphasize larger errors, thus limiting its sensitivity to variability in predictions. As a result, while MAE is effective for assessing general prediction accuracy, it may not be the best metric for highlighting areas with significant prediction variability[Hastie et al., 2009].

Mean Squared Error (MSE)

Mean Squared Error (MSE) calculates the average of the squared differences between predicted and actual values, placing greater emphasis on larger errors. This property makes MSE particularly sensitive to variability in predictions, as it magnifies larger deviations. However, MSE's sensitivity to outliers can sometimes skew results, potentially exaggerating variability if extreme values are not indicative of typical model performance. Despite this, MSE remains a fundamental metric in regression analysis, especially when the goal is to capture significant prediction errors[Bishop, 2006].

Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) expresses prediction errors as percentages of the actual values, making it unitless and facilitating comparisons across different scales. MAPE is popular due to its straightforward interpretation. However, it can become unstable when actual values are close to zero, leading to disproportionately high error percentages. This limitation can reduce MAPE's reliability when variability in predictions needs to be assessed, particularly when small prediction errors are overemphasized[Hyndman, 2018].

Mean Squared Percentage Error (MSPE)
Mean Squared Percentage Error (MSPE) extends the concept of MAPE by squaring the percentage errors, thus enhancing sensitivity to larger proportional deviations. This metric is particularly advantageous for identifying predictions with significant variability relative to the actual values, as the squaring operation disproportionately penalizes larger percentage errors. MSPE's focus on proportional deviations aligns well with scenarios where the relative accuracy of predictions is critical, making it a robust tool for detecting areas where the predictive power of input parameters is inadequate [Makridakis et al., 2008].

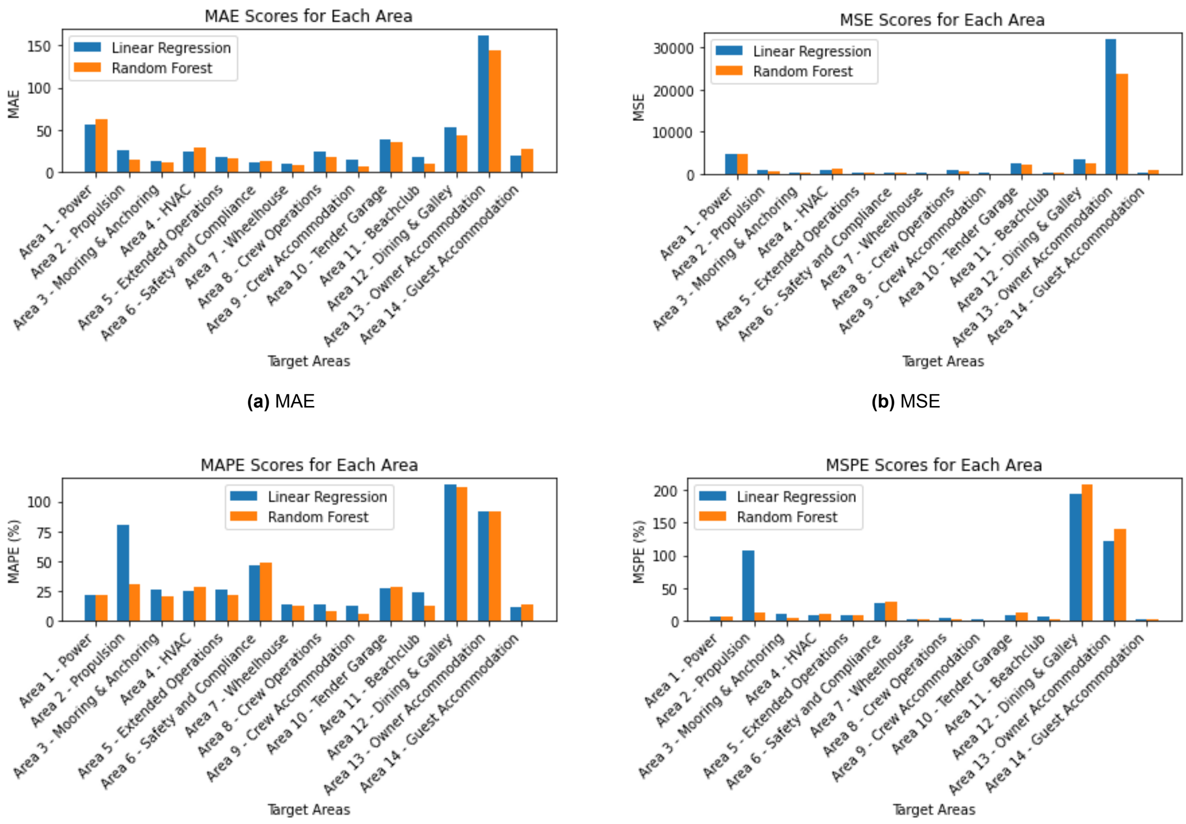
R-squared (R^2)
R-squared (R^2) measures the proportion of variance in the dependent variable that can be explained by the independent variables, offering an indicator of the model's overall explanatory power. However, negative R^2 values, as observed in some areas, suggest that the model performs worse than a simple mean prediction. While R^2 provides insight into the overall model fit, it may be less effective for comparing prediction variability across different areas, particularly in cases where the metric yields negative values[James et al., 2013].

Optimal metric
To effectively identify areas with the highest prediction variability, Mean Squared Percentage Error (MSPE) is considered as the most suitable metric. MSPE's sensitivity to relative deviations and its propensity to penalize larger errors through squaring makes it especially adept at capturing significant variations in predictive performance. Unlike MAPE, MSPE accounts for both the magnitude and proportionality of errors, providing a comprehensive view of prediction reliability.

Empirical evaluation of the model indicates that areas with high MSPE values suggest substantial variability in predictions, highlighting a imperfection in the predictive capability of the input parameters for these areas. Using MSPE enables a targeted approach to refining model accuracy by focusing efforts on areas where prediction improvements are most needed. While metrics like MAE and MSE provide insights into average error magnitudes and R^2 reflects overall explanatory power, MSPE excels in pinpointing areas with high relative prediction variability, making it the most suitable metric for identifying the predictive limitations of input parameters in the current regression framework.

4.3.4. Optimal Regression method
To determine whether linear regression or random forest is the better regression method for the application, it's important to consider the performance metrics and the underlying characteristics of each algorithm. In figure 4.7 and Appendix G the performance metrics of both regression methods are shown.
In the comparison of linear regression and random forest for predicting ship system areas, random forest proved to be the superior method due to its ability to handle complex, non-linear relationships between input features and target variables. This prove can be concluded by observing and comparing the performance metrics results from Appendix G. The fact that random forrest proves to be better comes from the fact that linear regression relies on the assumption of linearity, which often does not hold true in real-world datasets with intricate interactions, as seen in the ship system data. This limitation was evident in the performance metrics, where linear regression frequently showed poor fits, indicated by negative R^2 values and higher errors.

Random forest, on the other hand, outperformed linear regression across various metrics, including mean squared error (MSE), mean absolute error (MAE), and R^2 scores. Its ensemble approach effectively models non-linear interactions and adapts to the complexities of the data, capturing patterns that linear regression cannot. Furthermore, random forest's robustness to overfitting, ability to handle missing values, and adaptive response to feature perturbations further reinforce its suitability. These characteristics make random forest the preferred regression method for modeling ship system areas, where complex and non-linear dependencies are prevalent, providing more accurate and reliable predictions than linear regression.



(a) MAE

(b) MSE

(c) MAPE

(d) MSPE

Figure 4.7: Evaluation Metrics

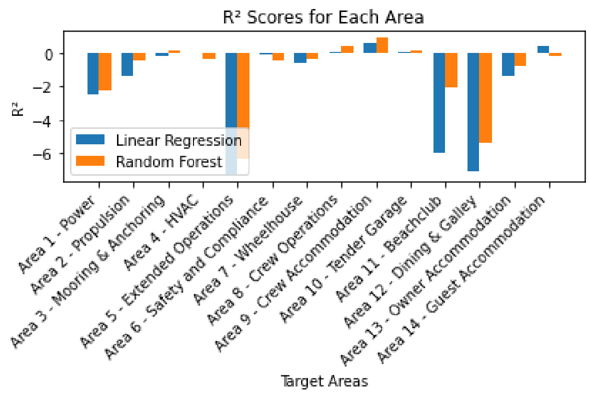


Figure 4.8: R^2 Evaluation Metric

4.3.5. Prediction Tool Evaluation

Upon loading the Prediction tool, both regression methods are executed and evaluated using specific evaluation metrics. The tool generates the predicted areas alongside the bar charts from figure 4.7. Analysis of these bar charts reveals significant outliers across all evaluation metrics. To assess the overall explanatory power of both regression models, the R^2 values are examined in Figure 4.8. This figure shows many negative values, indicating that the models perform worse than a simple mean prediction.

The poor performance can be attributed to two main factors: errors in the prediction script or limitations in the dataset. The first step is to explore potential improvements to the script. A more complex predictive model could potentially yield better results if the issue lies within the prediction model itself. Research suggests that hyperparameter tuning through cross-validation is a viable approach to enhancing the script. Furthermore, bootstrap sampling is a method that may achieve better results in cases involving smaller datasets. A detailed explanation of the functioning of both methods can be found in Appendix F.

Bootstrap Sampling

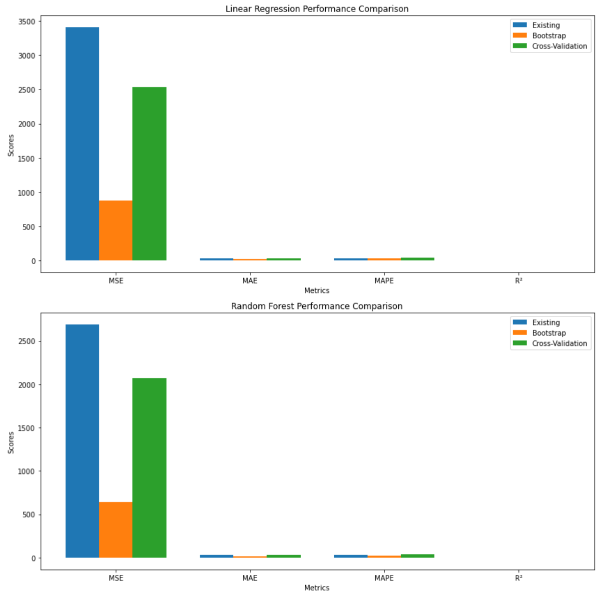
Bootstrap sampling is a statistical method used to estimate the accuracy of predictive models by resampling with replacement from the original dataset. This technique generates multiple simulated samples (bootstrap samples), each the same size as the original dataset but containing repeated instances without generating new input. By fitting the model on these samples and evaluating its performance on the remaining data, the variability and reliability of the model's predictions can be accessed. This approach helps in reducing variance and bias in model estimates, ultimately improving prediction accuracy[Efron and Tibshirani, 1994].

Cross-Validation

Cross-validation is a robust technique for assessing the predictive performance of statistical models by partitioning the data into complementary subsets. One commonly used form is k-fold cross-validation, where the data is split into k subsets, and the model is trained k times, each time using k-1 subsets for training and the remaining subset for validation. This process is repeated such that each subset serves as the validation set once. Cross-validation helps in detecting overfitting, enhancing the generalizability of the model, and ensuring that the performance metrics are a reliable reflection of the model's predictive power[Stone, 1974].

Performance improvement

To evaluate the potential added value of a more complex system, the prediction model was enhanced with both bootstrap sampling and cross-validation, and the performance was compared with the current model. As shown in Figure 4.9, both enhancements lead to an improvement in prediction accuracy according to the evaluation metrics. These improvements were tested independently before investigating their combined application.




```
Linear Regression Scores:
Existing - MSE: 3408.3735, MAE: 34.6998, MAPE: 37.9426%, R²: -1.8087
Bootstrap - MSE: 881.9702, MAE: 19.3588, MAPE: 35.1953%, R²: 0.0037
Cross-Validation - MSE: 2534.2809, MAE: 32.1856, MAPE: 44.7866%, R²: -1.1263

Random Forest Scores:
Existing - MSE: 2690.6685, MAE: 31.3224, MAPE: 32.5263%, R²: -1.2039
Bootstrap - MSE: 641.9909, MAE: 13.0189, MAPE: 23.9382%, R²: 0.5395
Cross-Validation - MSE: 2067.2415, MAE: 28.8264, MAPE: 38.8897%, R²: -0.6047
```

Figure 4.9: Performance Comparison

To integrate both cross-validation and bootstrap sampling into the script, an advanced evaluation framework based on the work of [Efron and Tibshirani, 1997] was developed where bootstrap sampling is conducted within each fold of the cross-validation, Appendix F shows a detailed explanation of the method. This approach involves resampling the training data within each fold, training the models on these bootstrap samples, and then averaging the predictions. This method enhances the robustness and stability of the models' performance evaluations. A flowchart shown in figure 4.10 shows the visualisation of the conduction of bootstrap sampling within each fold of the cross-validation. This flowchart shows the method applied to the area prediction script.

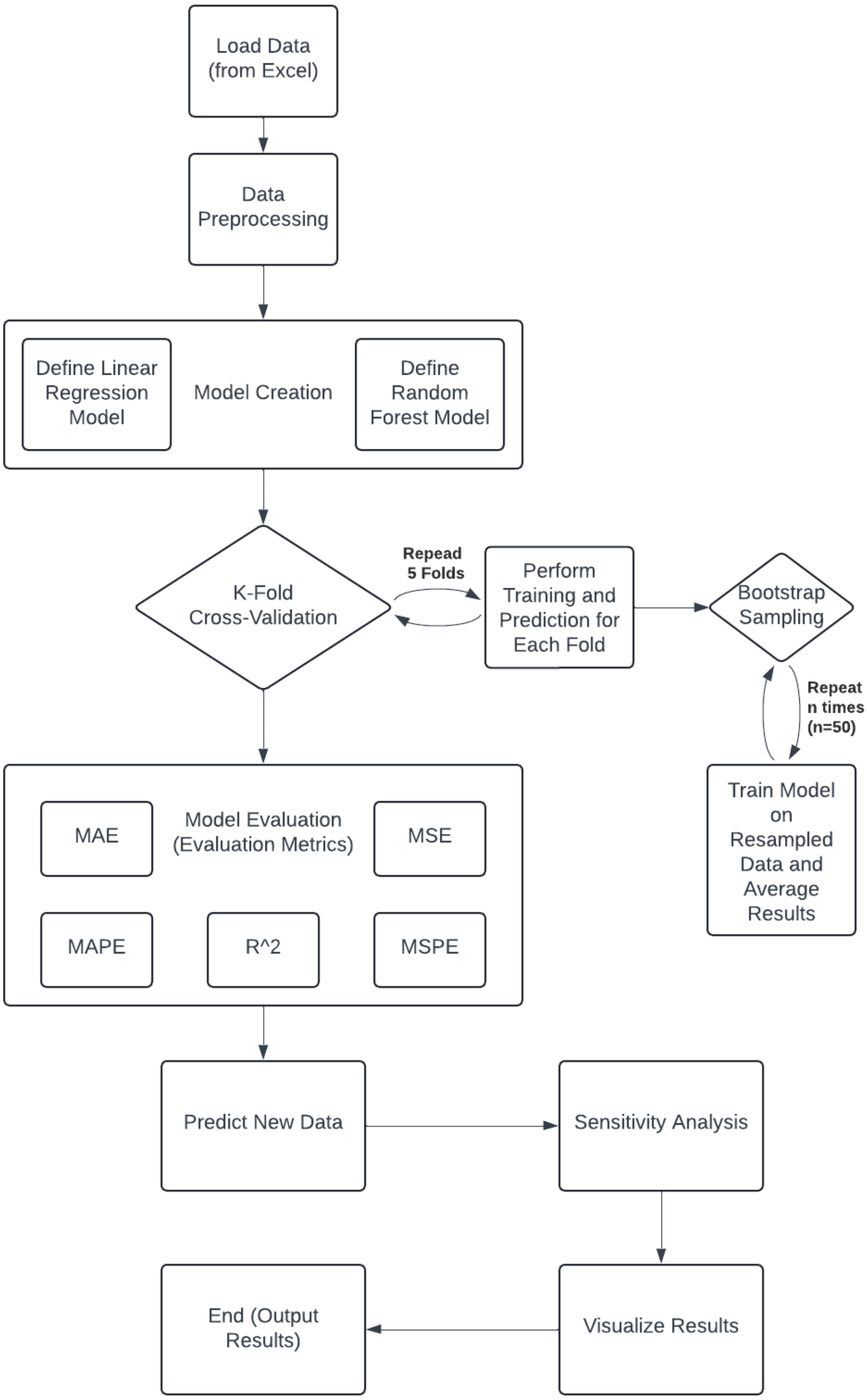


Figure 4.10: Flowchart Optimized Regression Model

The original script used a single train-test split. In the updated script, cross-validation (KFold) is introduced to provide a more reliable estimate of model performance. Cross-validation splits the data into multiple folds and evaluates the model across these folds, reducing the variance in performance metrics and better simulating how the models will perform on unseen data. Bootstrap sampling was added to further improve the robustness of the model evaluations. For each fold in the cross-validation, bootstrap resampling was performed to create multiple training datasets, and the models were trained on these resampled datasets. The predictions from each bootstrap sample were averaged to produce more stable and reliable performance metrics. For combining Cross-Validation and Bootstrap Sampling the bootstrap_cross_val_predict function was introduced, which combines cross-validation and bootstrap sampling by resampling the training data within each fold of the cross-validation. This approach leverages the strengths of both techniques, ensuring that the performance metrics are not overly influenced by any single split or specific sample of the data. A detailed explanation of the script improvements can be found in Appendix E.

4.3.6. Prediction tool results

To determine whether the updated script, which utilizes cross-validation and bootstrap sampling, produces more reliable results, the R^2 evaluation metrics of the original method and the complex updated method are compared. These R^2 evaluation metrics can be found in Figure 4.8 and Figure 4.11. A comparison of both bar charts reveals that the improvements in the updated script are clearly visible. In the original script, only four systems display a positive R^2 value, indicating a more reliable prediction than a simple mean prediction. In contrast, the updated script shows nine systems with positive R^2 values. This demonstrates that the incorporation of cross-validation and bootstrap sampling leads to a script with more reliable predictions. However, five systems still have negative R^2 values, indicating unreliable results. To further investigate this, additional evaluation metrics are examined and compared.

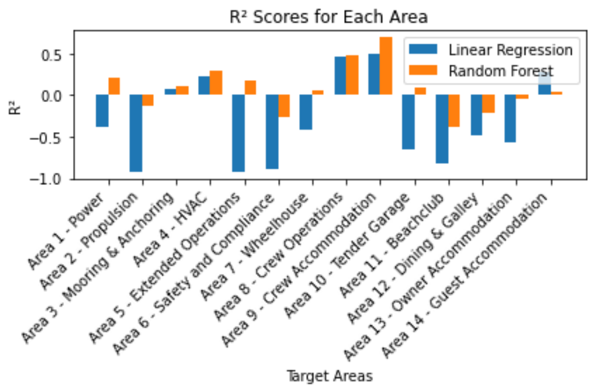
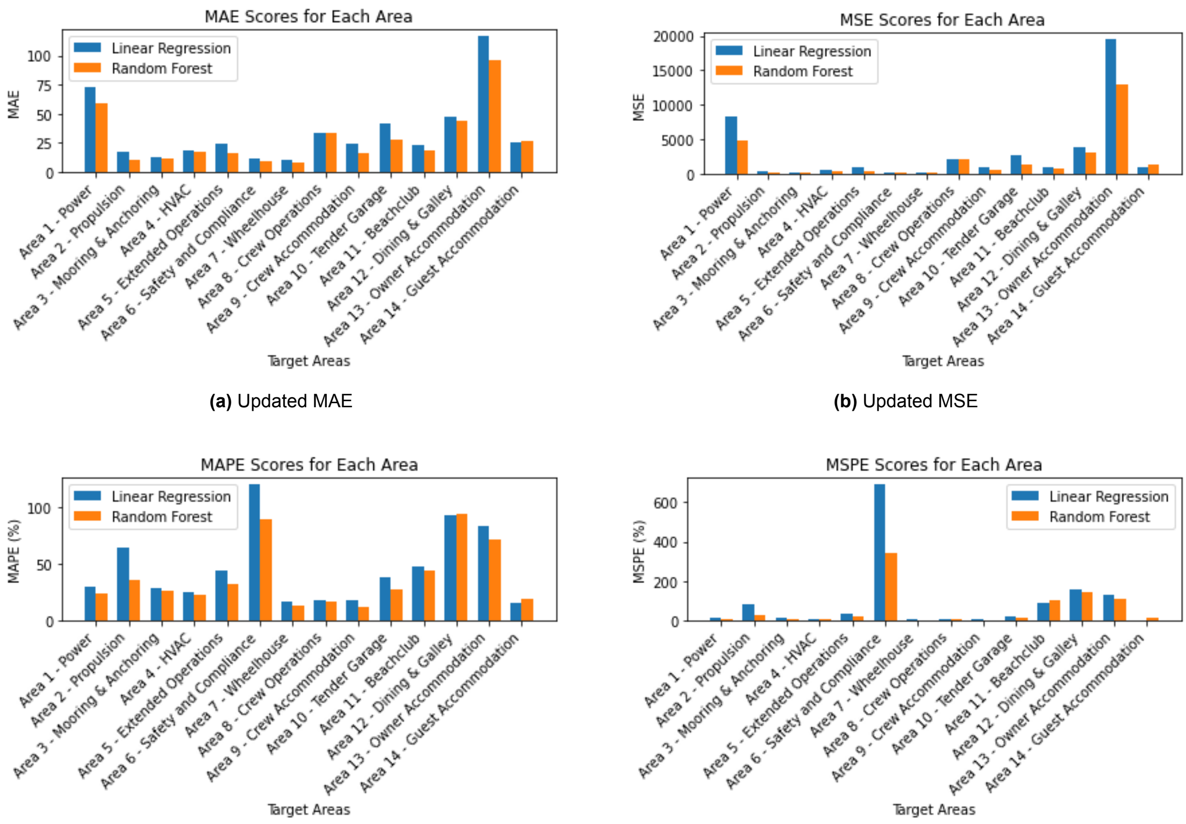


Figure 4.11: R^2 Updated Evaluation Metric

When comparing the other evaluation metrics, it becomes apparent that several systems still exhibit deviations. This is expected, as the overall explanatory power has increased due to improved R^2 values, but this does not necessarily mean that the prediction reliability has become more accurate. As substantiated earlier in Section 4.3.4, the random forest prediction model is used as the primary predictive model. A comparison of the prediction models for random forest reveals no significant changes between the MAE and MAPE scores. However, when comparing the MSE and MSPE scores, the MSPE for Safety & Compliance and Beachclub in the new system show higher MSPE scores. Since it has been established in Section 4.3.3 that MSPE is the most suitable metric for evaluating the predictions, this will be further investigated.



(a) Updated MAE

(b) Updated MSE

(c) Updated MAPE

(d) Updated MSPE

Figure 4.12: Updated Evaluation Metrics

When examining the R^2 and MSPE evaluation metrics, it is noticeable that in Figure 4.11, where the R^2 score is negative and thus indicates an unreliable prediction, the corresponding MSPE bar chart in Figure 4.12 also shows large percentage errors. This is observed in the systems Safety & Compliance, Beachclub, Dining & Galley, and Owner Accommodation. This implies that for these systems, a realistic and valid prediction of the area cannot be made. The Propulsion system has a positive R^2 value but also shows a significant MSPE deviation; here, a deviation is considered significant if the MSPE is greater than 30%. Section 4.2 explores whether this assumption is realistic. From these results, it can be concluded that the model cannot provide valid area predictions for the systems Safety & Compliance, Beachclub, Dining & Galley, Owner Accommodation, and Propulsion. All other systems show a positive R^2 value and an MSPE value below 30%, suggesting that the areas of these systems can be predicted using the prediction model.

Area Results

The predicted areas obtained using the random forest prediction model are shown in Table 4.1. These predictions were made based on the input parameters of the case study yacht DN3408. The predicted areas will be used in the subsequent research.

System	Valid prediction [Y/N]	Area [m^2]
Power	Y	285.35
Propulsion	N	44.69
Mooring & Anchoring	Y	61.98
HVAC	Y	113.48
Extended Operations	Y	67.52
Safety & Compliance	N	24.41
Wheelhouse	Y	73.78
Crew Operations	Y	191.34
Crew Accomodation	Y	113.94
Tender Garage	Y	127.49
Beachclub	N	86.21
Dining & Galley	N	61.15
Owner Accomodation	N	178.18
Guest Accomodation	Y	183.21

Table 4.1: Predicted system area's

4.3.7. Defining the owner specific requirements input areas

To investigate whether systems that, according to the prediction tool, cannot be realistically predicted and are therefore considered owner-specific requirements, are also perceived as owner-specific requirements in reality, the expertise of Sr. Designer Ruud Bakker was utilized. The results of this inquiry will help substantiate whether the assumption that a prediction with a MSPE deviation of up to 30% is realistic can be considered valid. For this purpose, the 14 systems were presented and explained to the Sr. Designer, together with the input parameters. The question of which systems, in his experience, are predominantly determined by the input parameters and which systems are significantly influenced by owner-specific requirements is asked. This question is asked in a interview conducted with Sr. Designer Ruud Bakker. In the interview, the designer evaluates the different systems to determine whether they depend on input parameters or are owner-specific. According to the designer, the Tender Garage, Beach Club, Dining & Galley, Owner Accommodation, and Guest Accommodation systems are considered owner-specific. These systems can then be compared with the systems classified as owner-specific based on the assumption that all systems with an MSPE (Mean Squared Percentage Error) > 30% reflect owner-specific requirements. These results are presented in Table 4.2.

System:	MSPE:	Owner specific MSPE >30%	Owner specific MSPE >10%	Owner Specific following Designer:
Power	9.7	N	N	N
Propulsion	27.2	N	Y	N
Mooring & Anchoring	9.31	N	N	N
HVAC	8.3	N	N	N
Extended Operations	22.7	N	Y	N
Safety & Compliance	360.2	Y	Y	N
Wheelhouse	3.1	N	N	N
Crew Operations	5.6	N	N	N
Crew Accomodation	2.8	N	N	N
Tender Garage	16.1	N	Y	Y
Beachclub	100.5	Y	Y	Y
Dining & Galley	141.3	Y	Y	Y
Owner Accommodation	108.3	Y	Y	Y
Guest Accommodation	11.2	N	Y	Y

Table 4.2: Systems owner specific requirements by MSPE and designer

When comparing the systems with an MSPE > 30% to those identified as owner-specific by the designer, it becomes evident that there is alignment for the Beach Club, Dining & Galley, and Owner Accommodation systems. However, the Tender Garage and Guest Accommodation, which the designer also classifies as owner-specific, score relatively high but do not exceed the 30% threshold. Interestingly, the Safety & Compliance system shows the highest MSPE score, though the designer does not consider it owner-specific. Senior Designer B. Jongepier offers an explanation for this discrepancy. Because the propulsion system, which also has a high MSPE score of 27% and a negative R^2 score, is closely related to the Safety & Compliance system. This can be explained by the fact that, according to the database, the propulsion system only accounts for the bow thruster area, while the main propulsion is located at the stern and does not occupy internal space. The Safety & Compliance system, on the

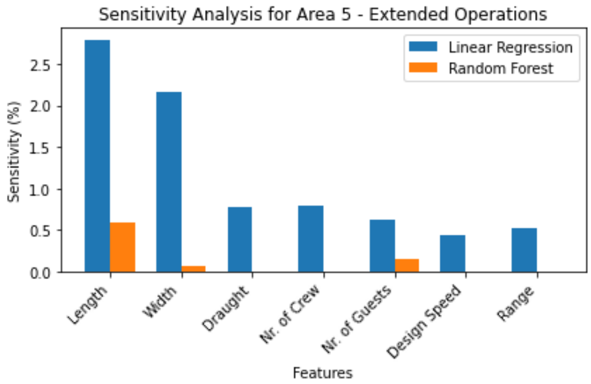


Figure 4.13: Sensitivity Analysis Extended Operations

other hand, includes the emergency generator and the Hi-Fog equipment. According to the specialist, the Hi-Fog room is often located in the bow thruster room, which reduces the space allocated to Safety & Compliance and increases the space for the propulsion system. As a result, both systems deviate significantly, leading to negative R^2 values and high MSPE scores. Based on this conclusion, it can be assumed that the high MSPE and negative R^2 values are not due to owner-specific requirements but can be explained by a data deficiency. Predicted values without added margins will be used, with the caveat that these may deviate from reality.

When examining the systems classified as owner-specific by the designer but with an MSPE < 30%, it is notable that both score above 10%, which is relatively high compared to other systems. Only the Extended Operations System also scores above 10%, with a score of 22.7%, even though the designer did not label it as owner-specific. Upon further investigation, it becomes clear that the Sensitivity Analysis (see Figure 4.13) for Extended Operations shows that this system, in the case of the random forest model, depends on length, width, and the number of guests. Range is not considered a determining factor for the Extended Operations system, which may explain the relatively high MSPE score for this system. Upon reviewing the database in Appendix **??**, it is evident that nearly all ships exhibit a range of 5000-6000 NM. In reality, these ships are likely classified based on specific ranges and periods of autonomous operation rather than purely on range. This explains why this system does not depend on standard parameters but could still be classified as owner-specific. Based on this assumption, it can be concluded that systems with an MSPE > 10% can be identified as reflecting owner-specific requirements. This conclusion can be further validated using the expertise of the Senior Designer and Senior Specialist, as well as the Sensitivity Analysis. In the arrangement generation tool, the predicted surface areas will be used to determine the space allocated to each module within the arrangement. By adding margins within the arrangement generation tool, it will be possible to assess whether systems with owner-specific requirements occupy more or less space than the predicted values, based on the owner's requirements for the ship.

4.4. Modular Arrangement Generator Tool

To investigate whether the modules can lead to innovative arrangements under certain input parameter conditions, a final study is conducted. In this study, a tool is developed to support the designer in the initial creative phase of the design process. The purpose of the design tool is to generate innovative yacht layouts based on the area output from the system area prediction tool from section 4.3. In addition to the areas, the tool utilizes known input parameters and arrangement conditions for the systems. A packing algorithm is employed, which uses the input and variables to generate various yacht arrangements. These yacht arrangements are simplified 2D representations of the side view of the yacht. The generated arrangements will ultimately be compared with the general arrangement of the prospect used for the input parameters (Appendix J). With the assistance of Sr. Designer R. Bakker and Sr. Specialist B. Jongepier, these results will be analyzed and discussed.

System	Height preference:	Height score:	Length preference:	Length score:	Close to System 1:	Close to Score 1:	Reasoning:	Far from System 1:	Far from Score 1:	Reasoning:
Power System	Low	9	-	0	-	0	T connection transfers via electricity cables so no need for physical connection to be close to a certain system.	Owner Accommodation	9	Because of noise generation from power system
Propulsion System	Low	9	Bow	9	-	0	Bowthruster has to be under waterline at the bow of the ship.	-	0	
Mooring & Anchoring System	Low	6	Bow	9	-	0	Mooring has to be on the main deck in the bow of the ship.	-	0	
HVAC System	-	0	-	0	Owner Accommodation	9	Research done in General arrangements and due to interview with Sr. Specialist B. Jongepier.	-	0	
Extended Operations System	-	0	-	0	Dining & Galley System	0	To get resources to the galley	-	0	
Safety and Compliance System	High	3	-	0	-	0	No preferences but safety and compliance system has to be re-evaluated because it contains the high-fog system and the emergency generator. Hi-fog system is mostly close to engine system but no regulations and emergency generator has to be above the waterline but this has to be checked with Sr. specialist B. Jongepier.	-	0	
Wheelhouse System	High	9	Bow	9	-	0	Bridge has to be at least at the owner deck and preferable as high as possible. This is because the command center is left outside the scope for this research and therefore the captain needs a plain view on the sea and it's surrounding.	-	0	
Crew Operations System	Low	6	-	0	-	0	No preferences for the crew operation system, but because no preferences it is expected to find the crew operations on the lower deck.	-	0	
Crew Accommodation System	Low	6	Bow	9	-	0	No preferences and therefore expected to find the crew operations on the lower deck.	Power System	6	Noise regulations, check with sr. Specialist B.Jongepier.
Tender Garage & Toy Storage Systen	-	0	-	0	-	0	Tender garage needs to be above the waterline and there are no further preferences.	-	0	
Beachclub system	Low	9	Stern	9	-	0	Beachclub has to be on the lower deck located as close to the stern as possible. Otherwise it would be to far from the water or to close to the bow and thereby causing difficulties with the ship bow structure.	-	0	
Dining & Galley System	High	3	-	0	-	0	No regulatory or preferences standard requirements for the dining&galley system	-	0	
Owner Accommodation System	High	9	Bow	6	-	0	Owner accommodation positioned on owner or bridge deck as high as possible and preferably facing the bow of the ship. Custom requirement of the client could be facing the stern of the ship.	Power System	9	Because of noise generation from the power system
Guest Accommodation system	High	6	-	0	-	0	Guest accommodation system has to be placed on the main deck or highter because of the need for big windows. Lower deck can't supply big enough windows. This is concluded due to the fact that there aren't any vessels found with guest accommodation on the lower deck. This has to be checked with Sr. Specialist B. Jongepier.	Power System	9	

Figure 4.14: Arrangement Conditions

4.4.1. Arrangement conditions

To determine the appropriate arrangement conditions for the application of the investigated method, research on similar studies was conducted. Previous research by [Nam et al., 2010] successfully demonstrates a similar application of ship layout based on arrangement conditions. In the study, the location of various modules in the ship is determined based on preference for deck height (y-direction) and longitudinal position (x-direction). Since it is expected that considering only deck height and longitudinal position will not result in reliable layouts, further research was conducted. Research by [Helvacioglu and Insel, 2005] indicates that not only the location of different modules but also the distance from one module to another is significant. This relationship between different modules is referred to as topological relations and is also included as an arrangement condition in determining the yacht layout. Because of the limited timeframe and simple nature of the research it can be assumed that for a effective functioning of the packing tool, the arrangement conditions are determined by height preference, length preference, and topological relations. The placement of the modules depends on their vertical position, horizontal position, and distance from other modules. To further specify these properties for each module, a unique classification system has been developed. Each system is individually assessed based on height preference, length preference, close to system, and far from system, with each preference rated on a scale from 1 to 3 to determine the weight of the property. Each system has a single vertical and a single horizontal preference but can have multiple topological relations (close to and far from), each with its own score. Figure 4.14 presents the arrangement conditions of the various systems. Appendix K displays the complete set of arrangement conditions. The scores of the arrangement conditions were determined through a literature review, the MIM matrix, and confirmations from Sr. Specialist B. Jongepier. Because of the experience of the Sr. specialist it is assumed these arrangement conditions meet the scope of the research.

4.4.2. Packing Algorithm

The objective of the packing algorithm is to generate a number of arrangements that are both innovative and optimized. As such, the packing algorithm can serve as a support tool for the designer during the early phase of yacht design. A side view (xz-plane) of the yacht was chosen to keep the packing algorithm simple and feasible. This choice is justified by the fact that most systems extend across the full width of the yacht, making the addition of a top view (xy-plane) less relevant. This was assumption is supported by a comprehensive study of General Arrangements (GA's) of existing yachts.

Packing algorithms are computational methods designed to optimize the arrangement of objects within a defined space, aiming to minimize wasted space or maximize utilization. These algorithms are widely applied in logistics, manufacturing, and other fields where efficient space utilization is critical, such as bin packing, container loading, and resource allocation problems. The effectiveness of a packing algorithm is often evaluated based on its packing score, which measures how efficiently the space is used.

Three packing algorithms

In this study, various packing methods were applied and tested on the research case. To determine which packing approach is most suitable for the research case, an investigation was conducted to identify which packing approach achieves the highest packing score. The packing approaches selected for this study include the greedy approach, heuristic approach, and simulated annealing. These three approaches are commonly chosen for packing problems due to their distinct advantages in balancing solution quality, computational efficiency, and ease of implementation.
The **greedy approach** is a simple, straightforward method that places each item in the first available space where it fits, typically following a pre-determined order such as by size or arrival sequence. The algorithm makes decisions based solely on local optima, without considering the global implications of those choices. While this approach is computationally efficient and easy to implement, it often yields suboptimal solutions due to its lack of foresight and inability to revise earlier decisions once made [Coffman Jr et al., 1984].
Heuristic approaches, on the other hand, are chosen because they provide a balanced trade-off between solution quality and computational effort. Heuristics incorporate problem-specific knowledge or rules, which enable them to efficiently navigate the search space and find near-optimal solutions. This makes them especially valuable in scenarios where exact methods are impractical due to time constraints or the complexity of the problem. Heuristics are adaptable and can be tailored to specific problem characteristics, enhancing their effectiveness across a wide range of packing scenarios [Wäscher et al., 2007].
Simulated annealing is included in the selection due to its powerful optimization capabilities, particularly in escaping local optima. Its iterative process and the probabilistic acceptance of worse solutions provide a robust mechanism for exploring a wider solution space, making it well-suited for complex and high-dimensional packing problems. Despite being more computationally intensive, simulated annealing's ability to eventually converge to high-quality solutions makes it an appealing choice when the goal is to achieve as close to optimality as possible[Kirkpatrick et al., 1983].
Upon comparing these approaches, based on their packing scores, it was found that the heuristic approach achieved the highest performing packing score, outperforming both the greedy approach and simulated annealing (Appendix G.1). This outcome was unexpected, as simulated annealing, with its complex and thorough search mechanism, was anticipated to perform best. However, the heuristic approach proved to be more effective in this specific application, likely due to its tailored adaptability and efficient exploration of the solution space. This emphasizes that the best algorithmic strategy depends on the particular characteristics of the problem, as increased complexity does not always equate to superior performance.

Input

The input for the packing algorithm consists of the predicted top view xy-plane areas from the area prediction tool. These surface areas are used in combination with the known input parameters, length and width, and are supplemented with the fixed deck height. From the predicted top view areas, the surface area in the sideview xz-plane is calculated by dividing them by the width and multiplying them by the deck height. Additionally, it is specified whether the systems consist of a single unit or can be divided into multiple units. The arrangement conditions (figure I) are subsequently imported via Excel.

Packing algorith variables

As described in Section 4.2, there are system areas that cannot be predicted using the prediction tool and input parameters. Consequently, these systems are considered owner-specific. Therefore, the design brief must account for the owner's preferences regarding these areas. To incorporate this into the packing tool, a margin has been implemented. For each owner-specific system, it can be specified whether the owner desires a relatively large or small space. This allows the arrangement tool to also approximate the owner-specific areas.

Additionally, the arrangement tool includes an optimization function. This function generates arrangements based on the input from the design brief, but for a yacht with a 10% reduced length. In other words, it examines whether all modules calculated for a yacht of a certain length can also fit into a yacht that is 90% of that length. This allows for exploring the potential of arrangements with optimized lengths and the corresponding innovative layouts.

4.4.3. Packing script

The packing script optimizes the arrangement of yacht systems using a heuristic packing approach based on user preferences extracted from the Excel file (Appendix I). It involves several key steps, including data loading and preprocessing, feasibility checks, dynamic scoring, heuristic packing, arrangement generation, and visualization. The script iteratively adjusts and scores the placement of systems based on height, length, and proximity preferences to maximize the total arrangement score. A comparative analysis is performed for different ship configurations, including an optimized length scenario. The flowchart in figure 4.15 shows the different steps and loops within the packing script. The flowchart shows the script without the optimized length scenario. The optimized length scenario looks the same but with 90% of the original length for arrangement space. The complete script with detailed explanations can be found in Appendix D.

Data Loading and Preprocesssing

The script reads system placement preferences from an Excel file, extracting height and length preferences along with proximity constraints to other systems. These preferences are organized into a structured dictionary for use in subsequent calculations.

Feasibility Checks and Area Adjustments

The system areas are adjusted based on predefined margins, and lengths are calculated by dividing the adjusted areas by the ship's width. Feasibility checks ensure that the total system length does not exceed the available deck space on the yacht, considering the ship's length, width, and number of decks.

Dynamic Scoring and Placement

Each systems placement is scored dynamically based on height and length preferences, as well as proximity to other systems. The scoring function incorporates user-defined weights and adjusts scores to prioritize systems that meet the desired configurations, enhancing the overall arrangement score.

Heuristic Packing and Arrangement Generation

A heuristic packing algorithm generates multiple system arrangements by iteratively evaluating potential placements and selecting the highest-scoring configurations. The script accommodates separable systems by dividing them into smaller blocks, optimizing their distribution across the yacht's decks.

Visualisation and Comparative Analysis

Arrangements are visualized in a 2D layout, illustrating system placements across the yacht's decks. The script also generates length-optimized configurations by reducing the ship's length, comparing the performance of these configurations against the original ones based on total and average scores.

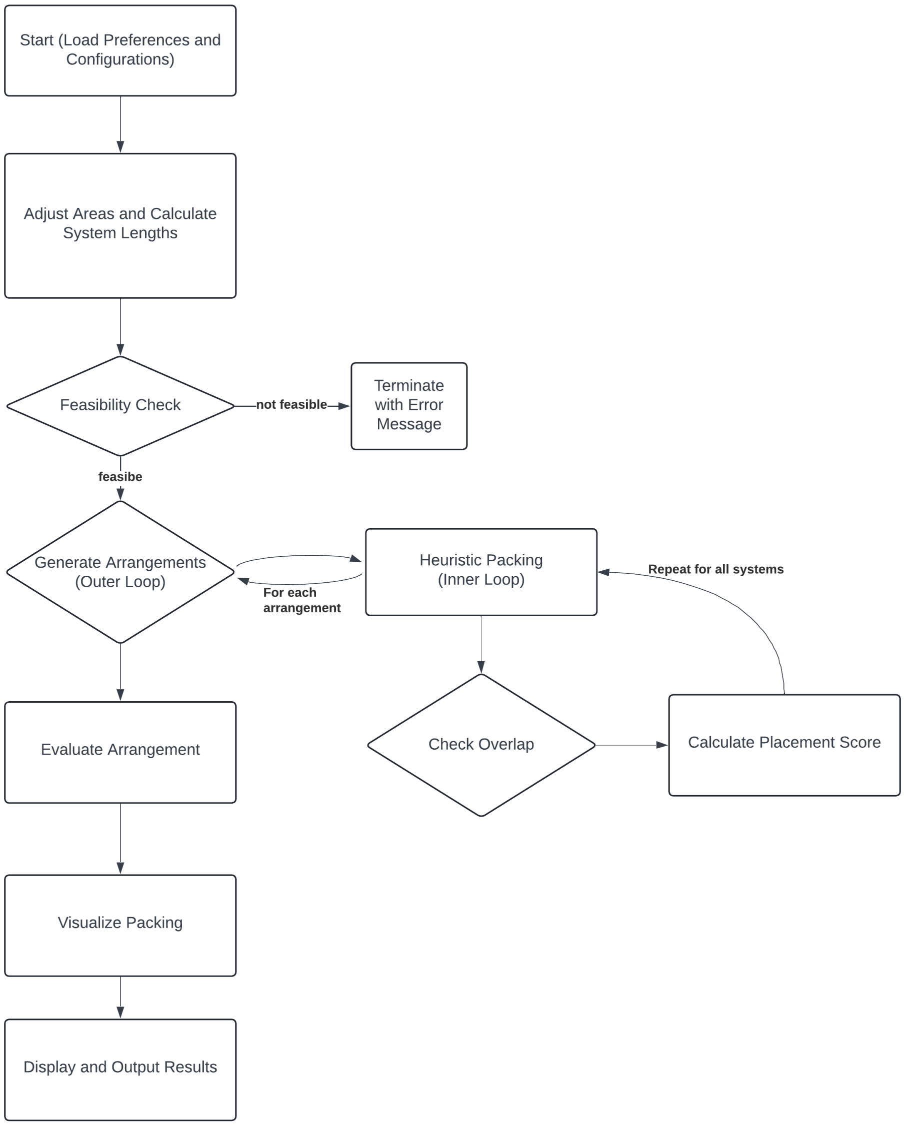
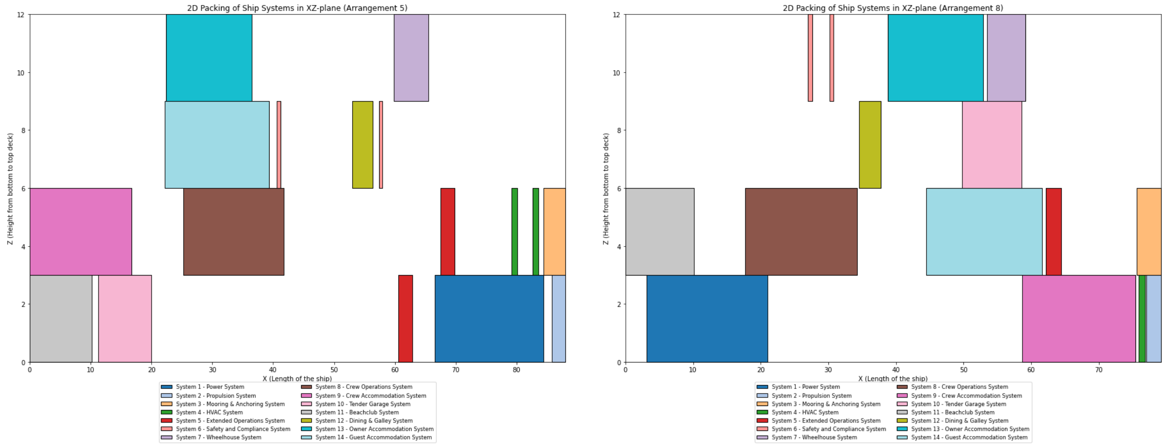


Figure 4.15: Flowchart Packing Algorithm Arrangement Tool

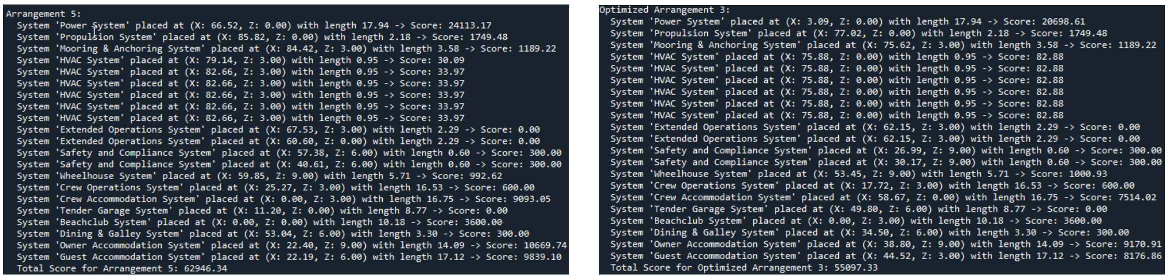
4.4.4. Module Arrangement visualisation results

To generate arrangements based on a specific design brief, the information from the design brief is first analyzed. This information, in combination with the area prediction tool, is used to gather the input for the packing algorithm. For this process, the length, width, and height of the yacht's decks are required, along with the areas obtained from the prediction tool and the preferences outlined in the Excel file. When the packing algorithm is executed within Spyder (Python 3.11), the desired results are visualized as plots. In the IPython console, the scores for each arrangement are displayed, including both the total score of the arrangements and the score per system. This approach allows for the identification of which system might be responsible for arrangements with significantly higher or lower scores. Figure 4.16 illustrates the highest-scoring arrangement for the case study yacht DN3408, along with the highest-scoring arrangement optimized for length. Margins for owner-specific systems were set to zero in this case due to the lack of access to the initial design brief. In addition to the visualized arrangements, Figure 4.17 presents the arrangement scores from the IPython console for both standard and length optimized arrangement. In Chapter 5, the results from the Arrangement tool are discussed and analyzed in collaboration with Senior Designer Ruud Bakker and Senior Specialist Bram Jongepier.



(a) Arrangement 5 **(b)** Arrangement 8 length optimized

Figure 4.16: Arrangement visualisation



```
Arrangement 5:
  System 'Power System' placed at (X: 66.52, Z: 0.00) with length 17.94 -> Score: 24113.17
  System 'Propulsion System' placed at (X: 85.82, Z: 0.00) with length 2.18 -> Score: 1749.48
  System 'Mooring & Anchoring System' placed at (X: 84.42, Z: 3.00) with length 3.58 -> Score: 1189.22
  System 'HVAC System' placed at (X: 79.14, Z: 3.00) with length 0.95 -> Score: 30.09
  System 'HVAC System' placed at (X: 82.66, Z: 3.00) with length 0.95 -> Score: 33.97
  System 'HVAC System' placed at (X: 82.66, Z: 3.00) with length 0.95 -> Score: 33.97
  System 'HVAC System' placed at (X: 82.66, Z: 3.00) with length 0.95 -> Score: 33.97
  System 'HVAC System' placed at (X: 82.66, Z: 3.00) with length 0.95 -> Score: 33.97
  System 'HVAC System' placed at (X: 82.66, Z: 3.00) with length 0.95 -> Score: 33.97
  System 'Extended Operations System' placed at (X: 67.53, Z: 3.00) with length 2.29 -> Score: 0.00
  System 'Extended Operations System' placed at (X: 60.60, Z: 0.00) with length 2.29 -> Score: 0.00
  System 'Safety and Compliance System' placed at (X: 57.38, Z: 6.00) with length 0.60 -> Score: 300.00
  System 'Safety and Compliance System' placed at (X: 40.61, Z: 6.00) with length 0.60 -> Score: 300.00
  System 'Wheelhouse System' placed at (X: 59.85, Z: 9.00) with length 5.71 -> Score: 992.62
  System 'Crew Operations System' placed at (X: 25.27, Z: 3.00) with length 16.53 -> Score: 600.00
  System 'Crew Accommodation System' placed at (X: 0.00, Z: 3.00) with length 16.75 -> Score: 9093.05
  System 'Tender Garage System' placed at (X: 11.20, Z: 0.00) with length 8.77 -> Score: 0.00
  System 'Beachclub System' placed at (X: 0.00, Z: 0.00) with length 10.18 -> Score: 3600.00
  System 'Dining & Galley System' placed at (X: 53.04, Z: 6.00) with length 3.30 -> Score: 300.00
  System 'Owner Accommodation System' placed at (X: 22.40, Z: 9.00) with length 14.09 -> Score: 10669.74
  System 'Guest Accommodation System' placed at (X: 22.19, Z: 6.00) with length 17.12 -> Score: 9839.10
  Total Score for Arrangement 5: 62946.34
```

```
Optimized Arrangement 3:
  System 'Power System' placed at (X: 3.09, Z: 0.00) with length 17.94 -> Score: 20698.61
  System 'Propulsion System' placed at (X: 77.02, Z: 0.00) with length 2.18 -> Score: 1749.48
  System 'Mooring & Anchoring System' placed at (X: 75.62, Z: 3.00) with length 3.58 -> Score: 1189.22
  System 'HVAC System' placed at (X: 75.88, Z: 0.00) with length 0.95 -> Score: 82.88
  System 'HVAC System' placed at (X: 75.88, Z: 0.00) with length 0.95 -> Score: 82.88
  System 'HVAC System' placed at (X: 75.88, Z: 0.00) with length 0.95 -> Score: 82.88
  System 'HVAC System' placed at (X: 75.88, Z: 0.00) with length 0.95 -> Score: 82.88
  System 'HVAC System' placed at (X: 75.88, Z: 0.00) with length 0.95 -> Score: 82.88
  System 'HVAC System' placed at (X: 75.88, Z: 0.00) with length 0.95 -> Score: 82.88
  System 'Extended Operations System' placed at (X: 62.15, Z: 3.00) with length 2.29 -> Score: 0.00
  System 'Extended Operations System' placed at (X: 62.15, Z: 3.00) with length 2.29 -> Score: 0.00
  System 'Safety and Compliance System' placed at (X: 26.99, Z: 9.00) with length 0.60 -> Score: 300.00
  System 'Safety and Compliance System' placed at (X: 30.17, Z: 9.00) with length 0.60 -> Score: 300.00
  System 'Wheelhouse System' placed at (X: 53.45, Z: 9.00) with length 5.71 -> Score: 1000.93
  System 'Crew Operations System' placed at (X: 17.72, Z: 3.00) with length 16.53 -> Score: 600.00
  System 'Crew Accommodation System' placed at (X: 58.67, Z: 0.00) with length 16.75 -> Score: 7514.02
  System 'Tender Garage System' placed at (X: 49.80, Z: 6.00) with length 8.77 -> Score: 0.00
  System 'Beachclub System' placed at (X: 0.00, Z: 3.00) with length 10.18 -> Score: 3600.00
  System 'Dining & Galley System' placed at (X: 34.50, Z: 6.00) with length 3.30 -> Score: 300.00
  System 'Owner Accommodation System' placed at (X: 38.80, Z: 9.00) with length 14.09 -> Score: 9170.91
  System 'Guest Accommodation System' placed at (X: 44.52, Z: 3.00) with length 17.12 -> Score: 8176.86
  Total Score for Optimized Arrangement 3: 55097.33
```

(a) Arrangement 5 **(b)** Arrangement 8 length optimized

Figure 4.17: Arrangement scores

4.4.5. Analyzing packing tool

Upon analyzing the packing tool, several notable irregularities have surfaced that are difficult to explain. In-depth investigation has indicated that these are the result of errors within the method. When examining the scores for the extended operations and tender garage in figure 4.17 for both the standard and optimized lengths, it is noticeable that these consistently receive a score of 0. Investigation into the preferences of these systems in the Excel file from Appendix I reveals that these are systems without preferences. However, a system without preferences should not be rated with a score of 0, as this suggests an extremely poor performance. Instead, a system without preferences should be evaluated based on its placement relative to other systems that do have preferences.

Research into this discrepancy has shown that the current system operates according to the following scoring mechanism:

- A system's placement is awarded points if it matches its height preference, with the weight defined by the score in the Excel file.
- Points are awarded if the system is closer to its preferred position, with the score adjusted according to the preference weight.
- Points are awarded based on how well this proximity requirement is met, with adjustments based on the proximity weight in the preferences.

It appears that the Tender Garage and Extended Operations systems lack preferences in the Excel file, and therefore do not receive any score. This is problematic because, even without explicit preferences, these systems should still receive a score based on their overall arrangement if they are placed logically within the ship. Research into this issue shows that the portion of the code responsible for calculating the score relies entirely on preferences. If a system has no preferences defined (such as Tender Garage or Extended Operations), it ends up with a score of zero, as there are no height preferences, length preferences, or proximity systems specified. Consequently, the dynamic score calculation returns zero, which is why systems without preferences always score zero.

To address this issue, it is suggested to introduce a baseline scoring system that assigns a reasonable score to any system, even if no specific preferences are provided. Systems should still receive a score based on general factors, such as whether they fit well within the available space and whether they are placed in logical areas of the ship. Plan for resolving the issues:

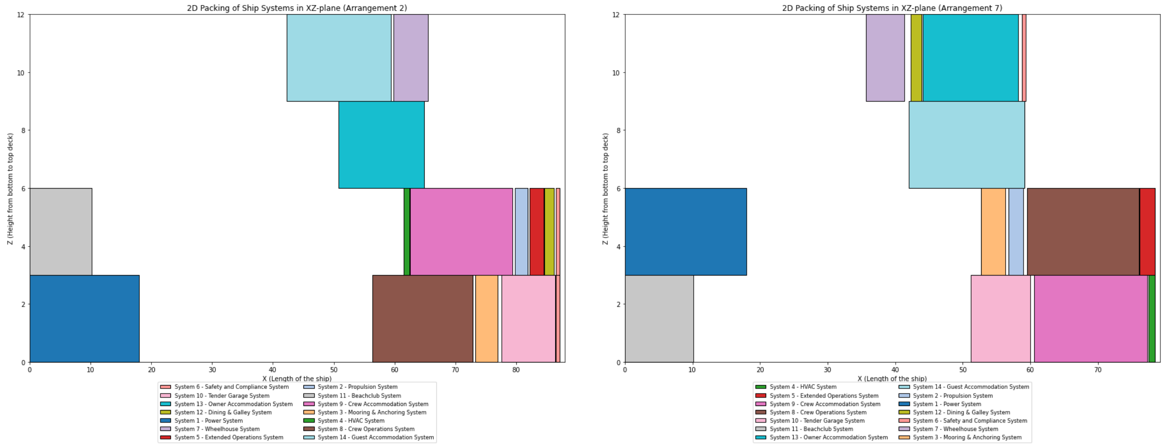
- **Add Baseline Scoring:** Introduce a default score for systems without preferences.
- **Revise Dynamic Score Calculation:** Modify the dynamic_score_placement function to assign baseline scores for systems with no specific preferences.
- **Prevent Over-Emphasis on Number of Preferences:** Ensure that systems with numerous preferences do not score disproportionately high, as the score should reflect how suitable the placement is rather than the number of preferences satisfied.

Script Adaptions

In the updated script, two key adjustments have been made to improve the scoring system for ship system placements. First, baseline scoring was introduced in the dynamic_score_placement function to ensure that systems without specific preferences are still evaluated fairly. A baseline score is now calculated based on how well the system fits within the available space on the ship. If the system fits well, it is rewarded, while systems that exceed the space are penalized. This ensures that even systems without preferences are judged based on their spatial efficiency. Second, the script addresses the issue of systems with numerous preferences receiving disproportionately high scores. In the updated dynamic_score_placement function, the score is now normalized by the number of active preferences. By dividing the total score by the preference count, the script ensures that systems with many preferences don't overshadow others, allowing the score to better reflect how suitable a placement is rather than just how many preferences are satisfied.

Results

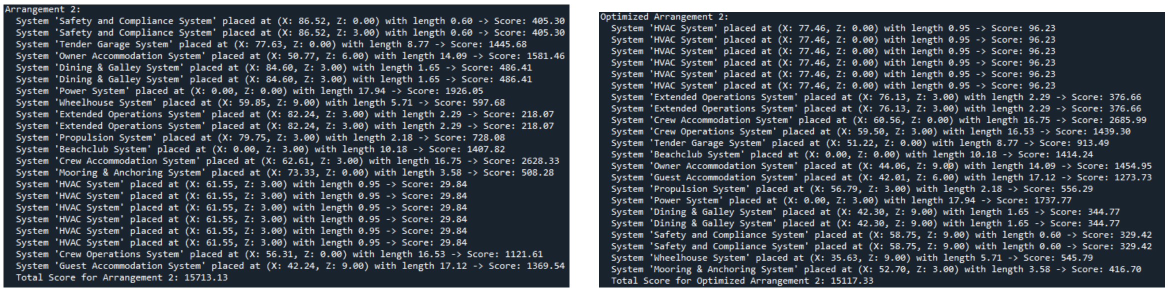
When the results of the improved script from figure 4.18 are compared with the results from the original script from figure 4.16, conclusions can be drawn. The updated arrangements generally show more logical layouts. This can be explained by the fact that the scores are less dependent on the number of preferences due to the normalization applied. When the scores from figure 4.19 are compared with those from the original script, this becomes clear. In the original script from figure 4.17, there are scores with values of 0 and notably high scores for owner, guest, and crew accommodations. These can be explained by the requirement for a large distance from the power system. When this is compared with the scores from the improved script, it is noticeable that the 0 values have disappeared, and the inexplicably high scores are no longer present. From this, it can be concluded that the issues identified in the original script are no longer present. This confirms that the improvements to the script have been successfully implemented in the new version.



(a) Updated Arrangement

(b) Updated Arrangement length optimized

Figure 4.18: Updated Arrangements



```
Arrangement 2:
  System 'Safety and Compliance System' placed at (X: 86.52, Z: 0.00) with length 0.60 -> Score: 405.30
  System 'Safety and Compliance System' placed at (X: 86.52, Z: 3.00) with length 0.60 -> Score: 405.30
  System 'Tender Garage System' placed at (X: 77.63, Z: 0.00) with length 8.77 -> Score: 1445.68
  System 'Owner Accommodation System' placed at (X: 50.77, Z: 6.00) with length 14.09 -> Score: 1581.46
  System 'Dining & Galley System' placed at (X: 84.60, Z: 3.00) with length 1.65 -> Score: 486.41
  System 'Dining & Galley System' placed at (X: 84.60, Z: 3.00) with length 1.65 -> Score: 486.41
  System 'Power System' placed at (X: 0.00, Z: 0.00) with length 17.94 -> Score: 1926.05
  System 'Wheelhouse System' placed at (X: 59.85, Z: 9.00) with length 5.71 -> Score: 597.68
  System 'Extended Operations System' placed at (X: 82.24, Z: 3.00) with length 2.29 -> Score: 218.07
  System 'Extended Operations System' placed at (X: 82.24, Z: 3.00) with length 2.29 -> Score: 218.07
  System 'Propulsion System' placed at (X: 79.75, Z: 3.00) with length 2.18 -> Score: 728.08
  System 'Beachclub System' placed at (X: 0.00, Z: 3.00) with length 10.18 -> Score: 1407.82
  System 'Crew Accommodation System' placed at (X: 62.61, Z: 3.00) with length 16.75 -> Score: 2628.33
  System 'Mooring & Anchoring System' placed at (X: 73.33, Z: 0.00) with length 3.58 -> Score: 508.28
  System 'HVAC System' placed at (X: 61.55, Z: 3.00) with length 0.95 -> Score: 29.84
  System 'HVAC System' placed at (X: 61.55, Z: 3.00) with length 0.95 -> Score: 29.84
  System 'HVAC System' placed at (X: 61.55, Z: 3.00) with length 0.95 -> Score: 29.84
  System 'HVAC System' placed at (X: 61.55, Z: 3.00) with length 0.95 -> Score: 29.84
  System 'HVAC System' placed at (X: 61.55, Z: 3.00) with length 0.95 -> Score: 29.84
  System 'HVAC System' placed at (X: 61.55, Z: 3.00) with length 0.95 -> Score: 29.84
  System 'Crew Operations System' placed at (X: 56.31, Z: 0.00) with length 16.53 -> Score: 1121.61
  System 'Guest Accommodation System' placed at (X: 42.24, Z: 9.00) with length 17.12 -> Score: 1369.54
  Total Score for Arrangement 2: 15713.13
```

```
Optimized Arrangement 2:
  System 'HVAC System' placed at (X: 77.46, Z: 0.00) with length 0.95 -> Score: 96.23
  System 'HVAC System' placed at (X: 77.46, Z: 0.00) with length 0.95 -> Score: 96.23
  System 'HVAC System' placed at (X: 77.46, Z: 0.00) with length 0.95 -> Score: 96.23
  System 'HVAC System' placed at (X: 77.46, Z: 0.00) with length 0.95 -> Score: 96.23
  System 'HVAC System' placed at (X: 77.46, Z: 0.00) with length 0.95 -> Score: 96.23
  System 'HVAC System' placed at (X: 77.46, Z: 0.00) with length 0.95 -> Score: 96.23
  System 'Extended Operations System' placed at (X: 76.13, Z: 3.00) with length 2.29 -> Score: 376.66
  System 'Extended Operations System' placed at (X: 76.13, Z: 3.00) with length 2.29 -> Score: 376.66
  System 'Crew Accommodation System' placed at (X: 60.56, Z: 0.00) with length 16.75 -> Score: 2685.99
  System 'Crew Operations System' placed at (X: 59.50, Z: 3.00) with length 16.53 -> Score: 1439.30
  System 'Tender Garage System' placed at (X: 51.22, Z: 0.00) with length 8.77 -> Score: 913.49
  System 'Beachclub System' placed at (X: 0.00, Z: 0.00) with length 10.18 -> Score: 1414.24
  System 'Owner Accommodation System' placed at (X: 44.06, Z: 9.00) with length 14.09 -> Score: 1454.95
  System 'Guest Accommodation System' placed at (X: 42.01, Z: 6.00) with length 17.12 -> Score: 1273.73
  System 'Propulsion System' placed at (X: 56.79, Z: 3.00) with length 2.18 -> Score: 556.29
  System 'Power System' placed at (X: 0.00, Z: 3.00) with length 17.94 -> Score: 1737.77
  System 'Dining & Galley System' placed at (X: 42.30, Z: 9.00) with length 1.65 -> Score: 344.77
  System 'Dining & Galley System' placed at (X: 42.30, Z: 9.00) with length 1.65 -> Score: 344.77
  System 'Safety and Compliance System' placed at (X: 58.75, Z: 9.00) with length 0.60 -> Score: 329.42
  System 'Safety and Compliance System' placed at (X: 58.75, Z: 9.00) with length 0.60 -> Score: 329.42
  System 'Wheelhouse System' placed at (X: 35.63, Z: 9.00) with length 5.71 -> Score: 545.79
  System 'Mooring & Anchoring System' placed at (X: 52.70, Z: 3.00) with length 3.58 -> Score: 416.70
  Total Score for Optimized Arrangement 2: 15117.33
```

(a) Updated Arrangement Score

(b) Updated Arrangement Score length optimized

Figure 4.19: Updated Arrangement scores

5

Case Study - Interview, Results & Discussion

The aim of this case study is to evaluate the functionality and effectiveness of the arrangement generator. With the assistance of Senior Designer Ruud Bakker and Senior Specialist Bram Jongepier, the arrangement generator will be analyzed and assessed using the presentation from Appendix J, comparisons with prospects from Appendix K, and a live demonstration of the arrangement generator. Given the combined 60 years of experience Bram and Ruud have in the designing and engineering of super yachts, it is assumed that strong conclusions can be drawn from this process. Based on several questions addressed during the interview, conclusions will be made regarding the performance of the arrangement generator. The interview will follow an open conversation format, where the presentation and live operation of the generator will be reviewed. The objective of the interview is to answer the following questions:

- What do the designer and specialist think about the working of the Arrangement Generator?
- Do the designer and specialist think the Arrangement Generator can be a tool to support the creative part of the design phase?
- Do the designer and specialist think the Arrangement Generator can be a tool to make quicker iterations in early phase design phase?
- What are the limitations of the Arrangement Generator
- What future adjustments can be made to improve the Arrangement Generator?
- In what areas performs the designer better than the arrangement generator?
- In what areas performs the arrangement generator better than the designer?

For the interview, the research was first explained using the presentation in Appendix J. This starts with an explanation of the development of modular systems based on the requirements outlined in Section 4.2. Subsequently, the functioning of the area prediction tool from Section 4.3 and the arrangement generator from Section 4.4 were explained. During this interview, results and arrangements were generated using the original packing tool, as shown in section 4.4.3. Since the improved version of the packing tool primarily focuses on detailed refinements rather than changes to its functionality or outcomes, it is assumed that this will not affect the results and conclusions drawn from the interview.To provide a comprehensive view of the arrangement generator's performance, three prospects (unbuilt yachts with highly developed designs) were analyzed in the case study. These three yachts are DN3408 - Bulldog, DN3524 and DN3554 - Komodo, with their corresponding input parameters shown in table 5.1. Komodo is particularly interesting to observe, as three detailed General Arrangements (GA's) are available for it. For area calculation and arrangement generation to compare Komodo the input from the studio and azure design is used.

Design Number	3408 - Bulldog	3524	3554 - Komodo - Studio	3554 - Komodo - Azure	3554 - Komodo - RWD
Length [m]	88	83	77,7	77,7	89,3
Width [m]	14,5	13,5	13,5	13,5	13,6
Draught [m]	4	3,85	3,85	3,85	4,05
Nr. of Crew [-]	17	30	23	23	30
Nr. of Guests [-]	14	18	14	14	16
Design Speed [kts]	16	17	17	17	-
Range [NM]	5000	5500	5500	5500	-

Table 5.1: Input parameters case study yachts

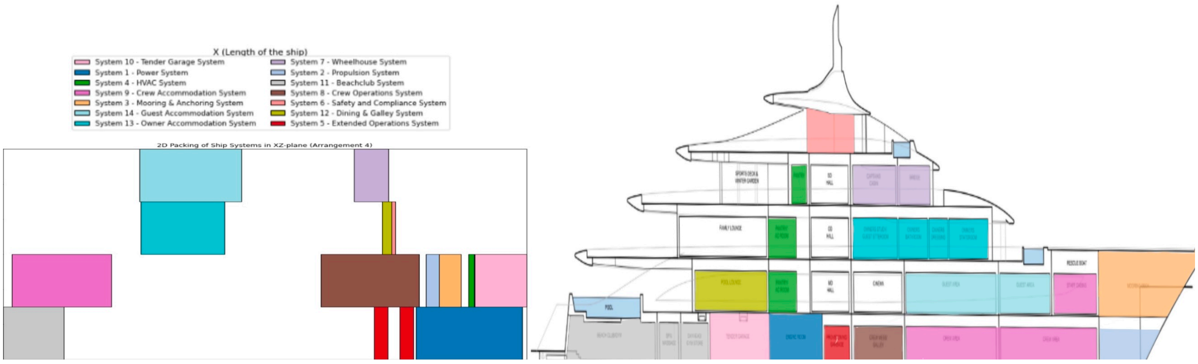


Figure 5.1: Comparison generation tool and Bulldog design

For all designs, the generated arrangements with the highest arrangement scores were collected and displayed alongside the general arrangements (GA's) created by the designer. This information was presented to the designer and specialist, as shown in figure 5.1 and Appendix K. A live demonstration of the arrangement generator was then given, after which the results were analyzed and compared with the designer's GA's. The pre-established questions mentioned above were answered during the interview based on the reactions and comments from the designer and specialist. Further conclusions and discussion points from the interview will also be presented, followed by a summarizing conclusion regarding the performance of the arrangement generator.

What do the designer and specialist think about the working of the arrangement generator?
For this question, the initial impressions of the designer and specialist regarding the arrangement generator are evaluated. The first impression is that the system shows potential but is challenging to interpret and understand in terms of its functionality. The specialist sees it as a tool that generates a wide array of starting points for a design, offering the possibility to explore creative and innovative new layouts. The designer adds that, in combination with the area prediction tool, the arrangement generator offers the potential for a very rapid initial cost analysis. This quick cost analysis is possible because the prediction tool and arrangement generator can swiftly estimate the amount of luxury space within the yacht, which in turn allows for a cost estimate. Such a rapid cost estimate could benefit the account manager while simultaneously saving the designer time.

Both the designer and the specialist further note that the arrangement tool only makes decisions based on the position within the ship and relative to other systems, whereas a designer considers countless factors when creating a design.

Does the designer think the arrangement generator can be a tool to support the creative part of the design phase?
When this question is posed to the designer, he expresses the expectation that the arrangement generator can help push the designer out of their comfort zone. Due to years of experience, fixed patterns and routines have developed in the designer's approach. By utilizing the arrangement generator, starting points outside of familiar designs are generated, which can then be considered for further exploration.

The designer also highlights that the current design process often does not sufficiently take the information from the design brief into account. Frequently, the design process begins without thoroughly analyzing the design brief, leading to the possibility that the owner's preferences may be overlooked. By analyzing the design brief and integrating its information into the arrangement generator, designs may emerge that the designer might not have otherwise considered.

Do the designer and specialist think the arrangement generator can be a tool to make quicker iterations in early stage design phase?
In response to the question of whether the arrangement generator can be used to enable faster iterations during the early phase of the design stage, the answer is no. The process of going through the design spiral is already well-established within the company, so it is not the area where time savings can be achieved. However, the arrangement generator does make it easier to generate new starting points, which can lead to the discovery of unique designs. Normally, finding a new and innovative design is a time-consuming process, but the arrangement generator can accelerate this.

Thus, the arrangement tool does not directly lead to faster iterations of the design spiral, but it facilitates the quicker discovery of new starting points for the design spiral. Ultimately, the arrangement generator can contribute to a faster generation of creativity within the design process.

What are the limitations of the arrangement generator?
The limitations of the arrangement generator primarily lie in its highly simplified functionality. Firstly, the designer points out that the arrangement generator is limited to four decks, with no option to remove a deck for optimization. Additionally, the specialist notes that in the length-optimized arrangement, the upper two decks of the ship become overly crowded. This makes the ship appear visually top-heavy and affects its weight distribution.

A general limitation of the arrangement generator is that it does not account for the weight distribution within the generated arrangements. Moreover, staircases and corridors are not generated by the tool, even though they have a significant impact on the layout of a yacht. The separation between crew and guest areas, as well as the connecting elements between different decks, such as the placement of the main staircase, greatly influence the overall arrangement of the yacht. However, these factors are not considered in the arrangement generator.

Another point of concern is that the arrangement generator is based on a dataset of yachts ranging between 70m and 110m in length. While the generator functions appropriately within this range, it will not produce useful results outside of it. This is due to the dataset not accounting for system spaces outside these lengths, and the arrangement generator itself does not consider critical regulatory boundaries. The most significant boundary is the 500GT limit, where, for example, fire insulation and the emergency generator, which occupy space within the design, are less critical. Additional important boundaries that are not considered include the 3000GT and 85m thresholds.

Making the arrangement generator applicable to all length categories, to cover the entire fleet, will require considerable effort.

What future adjustments can be made to improve the Arrangement Generator?
In addition to addressing the aforementioned limitations of the arrangement generator, the designer and specialist have identified several other potential areas for improvement. Both agree that the user interface could be made easier, clearer, and more efficient. While they acknowledge that the current version of the arrangement generator primarily serves to demonstrate its functionality, they believe that an improved interface would enhance this demonstration.

Furthermore, the specialist highlights the current functioning of the arrangement generator, which produces graphical designs based on numerical data. A future improvement could involve enabling the generator to create graphical designs using graphical input in addition to numerical input. This would allow the generator to produce higher-quality graphical arrangements as output.

In what areas performs the designer better than the arrangement generator?
When examining the areas where the designer outperforms the arrangement generator, the numerous considerations involved in yacht design become apparent. While the designer takes countless factors into account throughout the design process to achieve an optimal result, the arrangement generator only considers the placement of elements within the ship and their distance relative to other systems. The designer is familiar with all the unspoken conventions and unwritten rules of yacht design, which

are based on experience and knowledge that cannot be replicated by the arrangement generator. According to both the designer and the specialist, this indicates that the arrangement generator will always serve as a supportive tool and can never fully replace the designer.

In what areas performs the arrangement generator better than the designer?
The answer to the question of which areas the arrangement generator outperforms the designer primarily revolves around the generation of random designs. The arrangement generator analyzes the parameters for each design iteration and produces new arrangements, achieving a quantity of random designs that is unattainable for a designer. In a very short period, the generator can create an enormous number of innovative designs, something a designer would not be able to conceive or execute within the same timeframe. As a result, the arrangement generator enables the creation of a significantly higher level of innovative and creative arrangements.

Moreover, the arrangement generator ensures that the input parameters from the design brief are thoroughly analyzed and applied, a process that is currently less rigorously performed by designers.

Concluding:
The conclusion drawn with the designer and specialist is that the arrangement generator succeeds in generating basic early-stage design phase arrangements. However, the possibility that a more advanced tool could replace a designer in the early-stage design phase seems unrealistic. Although the tool can be further developed, it appears impossible for such a tool to learn and incorporate all the considerations a designer makes into a design. Nevertheless, the arrangement generator is viewed as a potentially powerful supportive tool. With further development, the tool could increasingly support the designer more easily and effectively.

The tool assists the designer by quickly generating multiple options, pushing the designer out of their comfort zone and enabling the rapid analysis of a wide range of possible innovative and creative designs. Additionally, the supportive tool could function as a tool for quickly analyzing the amount of luxury space present in a design. Therefore in the future, the tool could also assist in cost estimation, providing an account manager with a rapid price indication.

For further research, the tool could be studied more in-depth by making the tool generate arrangements until a design comparable to the GA is produced. By analyzing which systems score the highest in generated arrangement, it would be possible to determine which systems need to score high for a realistic and ”logical” design from the designer’s perspective. This could then be implemented in the arrangement generator through a toggle, allowing users to choose between a more creative or a more realistic design. This is supported by the observation that the generated design most similar to the actual design of Bulldog had the highest arrangement score. This suggests that a highly creative design often results in an illogical and lower-scoring design, while a conservative design similar to the designer’s arrangement scores higher.

Additionally, it is proposed to present the arrangement tool to younger, less experienced designers for further research. Young designers, having less established design patterns, may benefit more from such a supportive tool. They could be more easily inspired by the arrangement tool, given their greater flexibility in design approaches. Furthermore, younger designers are more accustomed to working with supportive tools, and it is expected that they would be more open to adopting the arrangement tool.

6 Conclusion

This research aimed to answer the question of whether the application of modularity can enhance design efficiency and creativity within the design process of custom-built luxury yachts. The main research question was formulated as follows:

How can a custom yacht company implement modularity to optimize design efficiency and creativity in the early-stage design process, while maintaining the high level of customization required for a custom yacht?

To investigate this main question, several sub-questions were developed, each introducing a new sub-study. These sub-questions are:

1. How can a modular design method be implemented in the design process to identify the required systems for luxury yachts, and evaluate which systems are suitable as a module?
2. What tool can predict the surface areas of modules based on design brief parameters, and how do owner-specific requirements influence these predictions?
3. How can designers be supported in exploring innovative arrangements of modules during the early stage design of luxury yachts?

The conclusions drawn from the sub-studies contribute to answering the main research question and support the final conclusion of the study.

To answer the question, **"How can a modular design method be implemented in the design process to identify the required systems for luxury yachts, and evaluate which systems are suitable as a module?"**, research was conducted on the functionality and applicability of Modular Function Deployment (MFD). In this study, MFD was applied to translate the requirements of a yacht into systems. Subsequently, using module drivers, an analysis was performed to determine which of these systems would be suitable as modules.

By analyzing the systems and modularity drivers, it became evident that all systems can benefit from modular principles to varying degrees, with some systems, such as the Tender Garage system, showing particularly high modular potential. Using the MIM matrix from the MFD method, it can be confirmed that the designated systems will provide added value as modules throughout the lifecycle of the yacht. This leads to the conclusion that these systems will function optimally as modules for the design process in the early stages of yacht development. On the other hand, the crew accommodation system, which was initially expected to score highly due to its simplicity and repetitive nature, scored lower. This outcome demonstrates that modularity is not just about constructing in blocks but involves considering the entire lifecycle of the yacht.

The study also revealed that modularity can enhance customization and personalization, contrary to initial assumptions that it might limit design creativity. This indicates that modular principles can not only streamline production and assembly but also encourage innovation and improve client satisfaction.

Further research is conducted to find an answer to the next subquestion: **"What tool can predict the surface areas of modules based on design brief parameters, and how do owner-specific requirements influence these predictions?"** The results of this analysis reveal that certain systems, such as the Beach Club, Dining & Galley, and Owner Accommodation, are consistently identified as owner-specific by both the predictive model and the designer's expertise. These systems exhibit high MSPE values (>30%), confirming their owner-specific nature. However, some systems, like the Tender Garage and Guest Accommodation, which were flagged by the designer as owner-specific, did not exceed the 30% MSPE threshold, indicating that while they are still influenced by owner preferences, their surface areas can be moderately predicted based on input parameters.

Additionally, the R^2 metric played a critical role in evaluating prediction accuracy. Systems with negative R^2 values, such as Safety & Compliance, indicate predictions that are worse than a simple mean prediction. This suggests that the inaccuracies are more likely due to data deficiencies or misalignment between the model and real-world conditions, rather than reflecting owner-specific requirements. For example, the high MSPE and negative R^2 values observed for the propulsion and Safety & Compliance systems underscore the complexity of these systems, where input parameters alone fail to capture the actual variations in space allocation. Due to this study, it can be assumed that modules with an MSPE >30% are strongly influenced by owner-specific requirements, while systems with lower but still significant MSPE values (10-30%) may also exhibit owner-specific characteristics. The R^2 analysis further highlights the limitations of the predictive model for certain systems, where prediction accuracy can be hindered by insufficient data or system complexity.

To complement the research and answer the main question, the final sub-study addresses the sub-question: **"How can designers be supported in exploring innovative arrangements of modules during the early stage design of luxury yachts?"**. The evaluation of the arrangement generator tool reveals its potential as a supportive tool for early-stage yacht design, particularly in fostering creative exploration by generating unconventional layout options. It encourages designers to move beyond established patterns, offering rapid ideation during the conceptual phase. However, the tool does not directly accelerate the iterative design process, it does facilitate a quicker discovery of new starting points. Current limitations include handling only up to four decks, neglecting spatial elements such as staircases and weight distribution, and being constrained to yachts within a specific length range (70-110 meters).

While the current arrangement generator serves as a valuable supportive tool in yacht design, certain parts are missing therefore limiting the working of the tool. Improvements to the user interface, the integration of graphical input capabilities, and the flexibility to toggle between creative and practical designs will significantly increase its utility. Although the generator is not a replacement for professional design expertise, with future advancements, it holds the potential to become an indispensable aid in both the creative and logistical processes of yacht design, expanding its applicability and versatility.

Based on the three sub-questions, a conclusive answer can be provided to the main research question: **"How can a custom yacht company implement modularity to optimize design efficiency and creativity in the early-stage design process, while maintaining the high level of customization required for a custom yacht?"**
The implementation of modularity in the design process of luxury yachts is feasible through the comprehensive application of Modular Function Deployment (MFD) in the early stage of the design phase. MFD is supported by the Area Prediction tool and the Arrangement Generation tool. Through the use of these supportive tools, it is demonstrated that, contrary to initial assumptions, modularity does not necessarily limit creativity. In fact, it offers the possibility to enhance creativity by encouraging innovative arrangements and offering designers a structured yet flexible framework to explore multiple configurations, pushing boundaries in the early stages of design.

7

Discussion and Recommendations

To complete the study within the given timeframe, assumptions and simplifications were made throughout the research. These assumptions are explained in this chapter, along with corresponding recommendations. The assumptions were made across the three sub-studies and will therefore be discussed in detail.

Modular Function Deployment (MFD)

The decision to apply a modular design method was made based on extensive literature research, which determined that MFD would be the most suitable approach. However, no practical research was conducted to compare different methods. The assumption that literature indicates MFD as the most appropriate method for this case might be challenged in practice, where other methods may prove more effective. For example methods like DSM, Function modelling and Axiomatic design as shown in section2.4.1. Further research into the practical application of these modular design methods is recommended to verify whether MFD can be proved to be the most suitable approach in practice.

For the analysis of the impact of the module drivers on different systems, the expertise of a single specialist was used due to time constraints. Although the specialist has over 30 years of experience, and thus his expertise is trusted, future research involving multiple specialists could provide stronger validation of the questionnaire results.

Area Prediction Tool

For predicting the surface areas, a simple random forest regression method was used, supported by cross-validation and bootstrap sampling. While the study shows that more complex predictive mechanisms might yield small improvements, it cannot be ruled out that a more advanced model could result in more reliable predictions. Since the nature of the research is to prove the method and not to perform it in detail, this assumption can be made. Future research should focus on identifying the best-fitting regression model for area predictions using the database to improve accuracy.

The dataset used for predictions consists of 25 yachts ranging from 70 to 100 meters in length. For realistic predictions over such a large range, a much larger dataset is required. Additionally, the study revealed that the dataset sometimes contained incorrect or missing values. Although efforts were made to improve the dataset, expanding the database with additional prospects or ship data from other shipyards could enhance the research. However, the use of ship data from other shipyards is strongly discouraged due to the different construction methods and quality standards. Furthermore, the current model expects accurate predictions from a dataset with a wide length range, future research could create separate datasets for smaller length ranges to improve accuracy.

The areas are predicted based on eight input parameters. Although these were identified as the most important parameters from the design brief, further research could study a wider range of design briefs to identify additional parameters that could enhance the dataset and improve the prediction tool's accuracy.

Evaluation metrics are used to determine which systems are owner-specific based on the prediction tool, and the results are compared with a designer's expertise. While this study relies on the experience of a single designer with over 30 years of expertise, future research involving multiple designers could lead to stronger conclusions regarding owner-specific systems.

Arrangement Generator

The arrangement generator is based on input from the prediction tool and specific arrangement conditions. These conditions were established through literature review and findings from MFD, and were finalized with the input of a specialist. Future research could incorporate the expertise of designers in addition to the specialist. Currently, the arrangement conditions are limited to spatial location within the yacht and relative positioning to other systems. To better reflect a designer's expertise, future research could consider more design factors in the arrangement conditions. Collaboration with designers would help identify which additional conditions are important and their relative significance in yacht design.

The heuristic approach was chosen for generating arrangements after testing it against two other methods: greedy approach and simulated annealing. While these are commonly used methods, further research could explore potentially better methods for packing algorithms.

Additionally, the heuristic approach was implemented in a significantly simplified version, the reason being the ability to quickly generate arrangements. This assumption was made in the research because evaluating the results, the functionality of the tool, and investigating its potential applications were deemed more important than obtaining highly accurate results from the tool. Now this has been proven in the conclusion, follow-up research could focus on improving the Packing Tool. Beyond exploring a more optimal method for applying packing to this case, the current packing tool and method could also be optimized by adding more complexity

The packing approach currently has many constraints, making it a relatively simple model. This assumption is made due to the limited time available for the research and the goal of proving the method's functionality rather than delivering a fully functional tool. One limitation is that the packing tool assumes uniform ship width across the entire length, leading to errors at the bow and stern, where ships typically taper. Additionally, the arrangement tool works in a 2D side view, assuming uniform width and that systems occupy the full width, which is not realistic. Future research could develop a 3D tool that accounts for varying widths and more accurate system placement.

The surface areas used in the arrangement generator come from the area prediction tool, but some systems with owner-specific areas are based on unrealistic predictions. While these areas are used in the arrangement generator, margins are added for owner-specific systems to reflect the owner's requirements. However, the predicted areas still include values with negative R^2 scores, indicating lower accuracy than a mean prediction. This assumption is made due to lack of owner specific information, resources and time. Future research could replace these areas with mean predictions or explore better methods to handle owner-specific system areas.

The conclusions about the arrangement generator's functionality are based on the assumption that a analysis conducted by a Senior Specialist and a Senior Designer is substantiated enough. Despite their combined 60 years of experience, a broader evaluation involving multiple specialists and designers could yield different insights. It is recommended that future research present the arrangement generator to a wider group of designers and specialists. Young designers, in particular, may offer fresh perspectives and could be more open to new tools. Additionally, the conclusions were drawn based on an outdated version of the packing tool, which contains errors. However, it has been assumed that these errors will not affect the final conclusions of the sub-study or the overall research. This assumption can be made because the update within the tool is on a certain level of detail, where conclusions from specialist and designer are made on overall working of the tool. Nonetheless, follow-up research using the most recent version of the packing tool will ensure that these errors are fully excluded.

Concluding Recommendations

The conclusion of the research indicates that the implementation of modularity in the design process of luxury yachts is feasible through the comprehensive application of Modular Function Deployment (MFD) in the early-stage design phase. MFD is supported by the Area Prediction tool and the Arrangement Generator tool. These tools demonstrate that, contrary to initial assumptions, modularity does not limit creativity. Instead, it can enhance creativity by encouraging innovative arrangements and providing designers with a structured yet flexible framework to explore multiple configurations, pushing boundaries in the early stages of design.

Further research should investigate whether the implementation of modularity can successfully address practical challenges. This includes studying whether clients can accept modularity without perceiving the designs as less bespoke. Additionally, research should explore whether designers and stakeholders can shift from traditional bespoke methods to modular design approaches without resistance. Another critical area for research is the potential conflict between modular design methods and maritime regulations. The transition to modular design could also impact Feadship's brand image, which should be thoroughly explored to find solutions to mitigate any negative effects. Lastly, research should focus on how investments in training, tools, and workflow adjustments can ensure a seamless transition to a modular design process.

When these studies are completed, a definitive conclusion can be drawn on whether modularity can optimize design efficiency and creativity within the yacht design process.

A important note that should be made here is that the research has focused on the current design process of yachts, specifically the existing Feadship fleet between 75 and 110 meters. However, the energy transition will require future yachts to meet different requirements and possess different characteristics compared to the current database. It is expected that yachts will evolve alongside this transition and will therefore need to comply with new standards.

Since the research, including the prediction tool and the subsequent arrangement generator, is based on the existing Feadship fleet, it cannot be guaranteed that these outcomes will remain relevant for the new generation of yachts that will be designed and built in the upcoming years. While this does not undermine the validity of the method, the reliability of the generator tool for future yachts cannot be assured. Further research will need to be conducted into the evolving set of requirements and properties for yachts as anticipated in the future.

Bibliography

[Bergman et al., 2009] Bergman, B., De Maré, J., Svensson, T., and Loren, S. (2009). *Robust design methodology for reliability: exploring the effects of variation and uncertainty*. John Wiley & Sons.

[Bishop, 2006] Bishop, C. M. (2006). Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

[Coffman Jr et al., 1984] Coffman Jr, E. G., Garey, M. R., and Johnson, D. S. (1984). Approximation algorithms for bin-packing—an updated survey. In *Algorithm design for computer system design*, pages 49–106. Springer.

[Davison and Hinkley, 1997] Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*. Number 1. Cambridge university press.

[Dobberfuhl and Lange, 2009] Dobberfuhl, A. and Lange, M. W. (2009). Interfaces per module: Is there an ideal number? In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 48999, pages 1373–1385.

[Efron and Tibshirani, 1997] Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560.

[Efron and Tibshirani, 1994] Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. Chapman and Hall/CRC.

[Ericsson and Erixon, 1999] Ericsson, A. and Erixon, G. (1999). *Controlling design variants: modular product platforms*. Society of Manufacturing Engineers.

[Erikstad, 2019] Erikstad, S. O. (2019). Design for modularity. *A Holistic Approach to Ship Design: Volume 1: Optimisation of Ship Design and Operation for Life Cycle*, pages 329–356.

[Erixon, 1998] Erixon, G. (1998). *Modular function deployment*.

[Erixon and Ostgren, 1993] Erixon, G. and Ostgren, B. (1993). Synthesis and evaluation tool for modular designs. In *International conference on engineering design*, pages 898–905.

[Erlandsson et al., 1992] Erlandsson, A., Erixon, G., and Ostgren, B. (1992). Product modules-the link between qfd and dfa. In *The International Forum on Product Design for Manufacture and Assembly, Newport, RI*.

[Feadship,] Feadship. The pursuit of dreams.

[Fuchs and Golenhofen, 2019] Fuchs, C. and Golenhofen, F. (2019). Mastering disruption and innovation in product management. *Switzerland: Springer International Publishing AG*.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

[Helvacioglu and Insel, 2005] Helvacioglu, S. and Insel, M. (2005). A reasoning method for a ship design expert system. *Expert systems*, 22(2):72–77.

[Hölttä et al., 2003] Hölttä, K., Tang, V., Seering, W. P., et al. (2003). Modularizing product architectures using dendrograms. In *DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm*, pages 343–344.

[Hyndman, 2018] Hyndman, R. (2018). *Forecasting: principles and practice*. OTexts.

[James et al., 2013] James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning*, volume 112. Springer.

[Jiao et al., 2007] Jiao, J., Simpson, T. W., and Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. *Journal of intelligent Manufacturing*, 18:5–29.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

[Kuhn, 2013] Kuhn, M. (2013). Applied predictive modeling.

[Lange and Imsdahl, 2013] Lange, M. W. and Imsdahl, A. (2013). Modular function deployment: using module drivers to impart strategies to a product architecture. In *Advances in product family and product platform design: methods & applications*, pages 91–118. Springer.

[Little and Rubin, 2019] Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

[Makridakis et al., 2008] Makridakis, S., Wheelwright, S. C., and Hyndman, R. J. (2008). *Forecasting methods and applications*. John wiley & sons.

[Malmström and Malmqvist, 1998] Malmström, J. and Malmqvist, J. (1998). Trade off analysis in product structures: A case study at celsius aerotech. In *Proceedings of NordDesign'98*, pages 187–196.

[Myers et al., 2012] Myers, R. H., Montgomery, D. C., Vining, G. G., and Robinson, T. J. (2012). *Generalized linear models: with applications in engineering and the sciences*. John Wiley & Sons.

[Nam et al., 2010] Nam, J.-H., Kim, D.-H., and Lee, H. J. (2010). An approach to computerized preliminary design procedure of mid-size superyachts from hull modeling to interior space arrangement. *International Journal of Naval Architecture and Ocean Engineering*, 2(2):96–103.

[Östgren, 1994] Östgren, B. (1994). Modularisation of the product gives effects in the entire production. *The Royal Institute of Technology, Stockholm*.

[Pahl and Beitz, 1988] Pahl, G. and Beitz, W. (1988). Engineering design: a systematic approach. *Nasa Sti/recon Technical Report A*, 89:47350.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

[Saltelli, 2008] Saltelli, A. (2008). *Global Sensitivity Analysis: the Primer*. John Wiley & Sons.

[Salvador et al., 2002] Salvador, F., Forza, C., and Rungtusanatham, M. (2002). Modularity, product variety, production volume, and component sourcing: theorizing beyond generic prescriptions. *Journal of operations management*, 20(5):549–575.

[Schafer and Graham, 2002] Schafer, J. L. and Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, 7(2):147.

[Smit, 2019] Smit, R. (2019). Modular platform: Towards a product family for a series of navy support vessels. *TU Delft*, (SDPO.19.035.m).

[Stjepandić et al., 2015] Stjepandić, J., Wognum, N., and Verhagen, W. J. (2015). Concurrent engineering in the 21st century. *Foundations, developments and challenges. Springer*.

[Stone, 1974] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133.

[Stone et al., 2000] Stone, R. B., Wood, K. L., and Crawford, R. H. (2000). A heuristic method for identifying modules for product architectures. *Design studies*, 21(1):5–31.

[Suh and Suh, 2001] Suh, N. P. and Suh, N. P. (2001). *Axiomatic design: advances and applications*, volume 4. Oxford university press New York.

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

[Ulrich, 1994] Ulrich, K. (1994). Fundamentals of product modularity. In *Management of design: engineering and management perspectives*, pages 219–231. Springer.

[Wäscher et al., 2007] Wäscher, G., Haußner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, 183(3):1109–1130.

[Willmott and Matsuura, 2005] Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82.

A

Paper

To be published...

B

Design Brief

FEADSHIP
ROYAL DUTCH SHIPYARDS

BRIEFING	
Design Number:	
Project Naam:	
Account manager:	
Designer DV	

1 Persoonlijk

- *Naam, nationaliteit, leeftijd*
- *Familie situatie*
- *Business*
- *Cultuur, interesses, lifestyle*
- *Kunst / Hobbies / auto's*
- *Belangrijke waarden - ecologie, innovatie, comfort etc.*
- *Hoe bij Feadship gekomen*
- *Huidige/vorige jachten / gecharterde jachten*
- *Overig – formeel/informeel*

2 Programma van eisen

2A Hoofd afmetingen

- *Exterior/, Interior designer*
- *Lengte, breedte, max. diepgang*
- *Snelheid / range / motoren*
- *Regelgeving, tonnage eis*

2B Gebruik

- *Privé / Zakelijk / Charter*
- *Welke ruimtes en waar*
    - *Gasten hutten, aantal gasten*
    - *Eigenaars hut, kantoor*
    - *Luxe ruimtes*
    - *Tenders (type/lokatie), helicopter, toys*
    - *Buiten ruimtes, jacuzzi, zwembad*
    - *Crew, aantal / lokatie*
    - *Etc. – lift*
- *Lange / korte reizen / vaargebied*
- *Veel weinig gasten, kinderen*
- *Nadruk op buiten / binnen ruimtes, zon / schaduw, privacy*

2C Vorm stijl

- *Klassiek / modern..*
- *Referentie jachten*
- *Wat juist niet*

3 Budget / Proces / presentatie

3a Budget

- Laag/hoog budget

3b Proces

- *Betrokkenen tijdens proces en taken / bevoegdheden*
- *Makelaar, Owners rep, Captain*

3c Laatste contact

- *Datum*
- *Wat is gepresenteerd*
- *Aanwezigen*

3d Volgende contact

- *Datum*
- *Wat wordt gepresenteerd*
    - *Ontwerp voorstellen*
    - *Plan (block layout / ingevuld)*
    - *Profiel*
    - *3D rendering / print model*
- *Aanwezigen*

DE VOOGT
NAVAL ARCHITECTS

C

Product Management Matrix

QFD

	Meal Preparation and Dining	Comfortable Sleeping	Provide Laundry Facilities	Provide Sanitary Facilities	Climate Control and Air Quality	Stability and Buoyancy	Emergency Self-Protection	Construction Integrity	Safe Navigation	Efficient Propulsion and Maneuvering	Secure Docking and Anchoring	Luxurious Acoommodation	Recreational Facilities	Operational Efficiency for Crew	Extended Operation Capabilities	Safety and Regulatory Compliance
Extended Stays Onboard	●	●	◉	○	○							○	○	○	●	○
Convenient Meal Preparation	●		○												●	
Comfortable Sleeping Environment		●	●	●	●	○						●				
Continuous Hygiene Maintenance			●	●								○		●	○	○
Comfortable Living Conditions	○	○	◉	○	●	◉						◉				
Safe Living Conditions on Sea						●	●	●	◉							●
Self-reliance During Emergencies							●							◉		◉
Structural Safety								●								●
Safe Navigation						◉			●	●				◉		
Independence in Movement						◉			●	●				◉		
Secure Mooring						○				●	●			●		
Luxurious Guest Accommodation		●	◉	◉	●							●				
Entertainmend and Leisure	○		○										●	○		
Crew Comfort and Efficiency		●	◉	○										●	○	◉
Long-Term Self-Sufficiency	◉		○											○	●	
Safety, Security and Compliance								●	◉					◉	○	●
Luxury Aesthetics, Craftsmanship & Tailored Design	◉	●										●				
Tailored Design																

○	Weak
◉	Moderate
●	Strong

DPM

Meal Preparation and Dining	Comfortable Sleeping	Provide Laundry Facilities	Provide Sanitary Facilities	Climate Control and Air Quality	Stability and Buoyancy	Emergency Self-Protection	Construction Integrity	Safe Navigation	Efficient Propulsion and Maneuvering	Secure Docking and Anchoring	Luxurious Guest Acoommodation	Recreational Facilities	Operational Efficiency for Crew	Extended Operation Capabilities	Safety and Regulatory Compliance	
				◉					●	◉			◉	○	○	Power System
								●	●	◉						Propulsion System
					○					●		○	◉	○		Mooring & Anchoring System
◉	●		○	●							●	◉	○		○	HVAC System
◉		◉	○						◉	○			◉	●	○	Extended Operations System
				◉	○	○		●	●	●			◉		◉	Wheelhouse System
												●	◉			Tender Garage & Toy Storage System
○		○	◉									●	○			Beachclub system
●		○	◉	◉							◉		◉	◉		Dining & Galley System
○	●	◉	●	◉							●					Owner Accommodation System
	●	◉	●	◉							●					Guest Accommodation system
	●	◉	●	◉								◉	○		○	Crew Accommodation System
◉		●				◉				●	○		●	○	◉	Crew Operations System
●	◉		○	○	●	●	●	○	◉				○	○	●	Safety and Compliance System

MIM

	Customization and Personalization	Luxurious Aesthetics	Advanced Technology Integration	Modular Adaptability	Efficient Production and Assembly	Crew Efficiency and Welfare	Independent Testing and Quality Assurance	Safety and Regulatory Compliance	Servicability and Mainenance	Upgradability and Flexibility	Sustainability and Environmental Compliance	Long-Term Durability
Power System	○		●	○	◉	○	◉	○	◉	◉	○	○
Propulsion System	○		●		◉	○	○	○	○	●	○	○
Mooring & Anchoring System					○	○	○	◉		○		
HVAC System	○		◉		○		◉		○	○	◉	○
Extended Operations System	◉		○						○		◉	
Wheelhouse System	◉	○	○	○	○	◉	◉	◉	○	○	○	◉
Tender Garage & Toy Storage System	●	○	◉	○	●	○	○	○	○	◉		◉
Beachclub system	●	●	○		○					○		
Dining & Galley System	◉	○				○			○		○	○
Owner Accommodation System	◉	◉	○	○	○		○					
Guest Accommodation system	◉	◉	○	○								
Crew Accommodation System						○		○	○			
Crew Operations System	○		○		○	○	○	○	○	○	○	○
Safety and Compliance System				◉		○	◉	●	○	○		

D

Regression prediction model script

This study applies machine learning techniques to analyze ship data, with a focus on predictive modeling of the yachts module areas based on the design parameters. The methodology involves the use of Linear Regression and Random Forest models to predict all areas simultaneously. The methodology builds on established principles of multi-output regression, which has been widely applied in complex engineering contexts requiring the simultaneous prediction of multiple interrelated outcomes[Tsoumakas and Katakis, 2007

Methodology

Data loading and preprocessing

Data import and preprocessing

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.multioutput import MultiOutputRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.impute import SimpleImputer
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import warnings
import matplotlib.pyplot as plt

warnings.filterwarnings("ignore", message="X does not have valid feature names, but")
```

The analysis begins with importing the necessary libraries for data manipulation, model training, and evaluation. The pandas library is utilized for data manipulation, while numpy is employed for numerical operations. Machine learning models and evaluation metrics are sourced from the scikit-learn library, a well-established library for machine learning in Python [Pedregosa et al., 2011]. Warnings related to feature names are suppressed to ensure cleaner output during model training and prediction.

Definition of system names

```
    system_names = [
    "Power System", "Propulsion System", "Mooring & Anchoring System",
    "HVAC System", "Extended Operations System", "Safety and Compliance System",
    "Wheelhouse System", "Crew Operations System", "Crew Accommodation System",
    "Tender Garage System", "Beachclub System", "Dining & Galley System",
    "Owner Accommodation System", "Guest Accommodation System"
]
```

The list of system names corresponds to the target variables, providing descriptive labels for the various ship systems under analysis.

Data Loading

```
filepath = r"C:\Users\jaap.marcus\OneDrive - De Voogt Naval Architects\Graduation Jaap Marcus\Onderz
df = pd.read_excel(filepath, sheet_name='Ship Data')
```

Data are loaded from the Excel file, the Excel file contains the information on ship design parameters and system areas. This data is collected from the Feadship Design Database.

Handling of Missing Values

```
df.replace(0, np.nan, inplace=True)
```

Zeros in the dataset are replaced with NaN values, assuming that zero represents missing data rather than true values. This is a standard perprocessing step in machine learning to handle incomplete data [Little and Rubin, 2019].

Data conversion and feature selection

Ensuring numeric data

```
feature_columns = ['Length', 'Width', 'Draught', 'Nr. of Crew', 'Nr. of Guests', 'Design Speed', 'Ra
target_columns = ['Area 1', 'Area 2', 'Area 3', 'Area 4', 'Area 5', 'Area 6', 'Area 7',
                  'Area 8', 'Area 9', 'Area 10', 'Area 11', 'Area 12', 'Area 13', 'Area 14']

for col in feature_columns + target_columns:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

To ensure all feature and target columns are suitable for analysis, the script convertgs them to numeric types, coercing errors into NaN values. This step is crucial for maintaining the integrity of the dataset, as non-numeric data can lead to errors during model training[James et al., 2013].

Separation of features and targets

```
X = df[feature_columns]
y = df[target_columns]
```

The dataset is split into features (x) and targets (y). Features consist of ship design parameters, while targets are the areas associated with the different ship systems.

Imputation of missing values

Imputation

```
imputer_X = SimpleImputer(strategy='mean')
imputer_y = SimpleImputer(strategy='mean')

X = pd.DataFrame(imputer_X.fit_transform(X), columns=X.columns)
y = pd.DataFrame(imputer_y.fit_transform(y), columns=y.columns)
```

Missing values in both the feature and target datasets are imputed using the mean strategy Imputation is a widely accepted technique for handling missing data, which helps prevent bias and maintains dataset size, ensuring robust model training[Schafer and Graham, 2002].

Data splitting

Train-Test split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

The dataset is split into training and testing sets using an 80/20 ratio. A random state is set to ensure the split is reproducible, which is critical for scientific experiments to allow for verification of results[Kuhn, 2013].

Model training

Training of models

```
lr_model = MultiOutputRegressor(LinearRegression())
lr_model.fit(X_train, y_train)

rf_model = MultiOutputRegressor(RandomForestRegressor(random_state=42))
rf_model.fit(X_train, y_train)
```

Two models are trained: a multi-output linear regression model and a multi-output random forest regression model. Because of its simplicity and interpretability linear regression is chosen. The linear regression is by training a MultiOutputRegressor on the data. For its robustness and ability to handle non-linear relationships a RandomForestRegressor is trained on the data as well.

Model prediction and evaluation metrics

Model predictions

```
lr_y_pred = lr_model.predict(X_test)
rf_y_pred = rf_model.predict(X_test)
```

Both models are used to predict the target variables on the test set, allowing for performance evaluation against the true values.

Evaulation metrics

```
def mean_absolute_percentage_error(y_true, y_pred):
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

def mean_squared_percentage_error(y_true, y_pred):
    return np.mean(((y_true - y_pred) / y_true) ** 2) * 100
```

The models are evaluated using multiple metrics, Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Percentage Error (MSPE) and R^2 score.

Performance evaluation

```
lr_mse_scores = []
rf_mse_scores = []
lr_mae_scores = []
rf_mae_scores = []
lr_mape_scores = []
rf_mape_scores = []
lr_mspe_scores = []
rf_mspe_scores = []
lr_r2_scores = []
rf_r2_scores = []

print("Model Performance Evaluation:")
for i in range(y.shape[1]):
    area_name = f'Area {i+1} - {system_names[i]}'
    lr_mse = mean_squared_error(y_test.iloc[:, i], lr_y_pred[:, i])
    lr_mae = mean_absolute_error(y_test.iloc[:, i], lr_y_pred[:, i])
    lr_mape = mean_absolute_percentage_error(y_test.iloc[:, i], lr_y_pred[:, i])
    lr_mspe = mean_squared_percentage_error(y_test.iloc[:, i], lr_y_pred[:, i])
```

```
    lr_r2 = r2_score(y_test.iloc[:, i], lr_y_pred[:, i])
    lr_mse_scores.append(lr_mse)
    lr_mae_scores.append(lr_mae)
    lr_mape_scores.append(lr_mape)
    lr_mspe_scores.append(lr_mspe)
    lr_r2_scores.append(lr_r2)
    print(f'Linear Regression - {area_name} - MSE: {lr_mse}, MAE: {lr_mae}, MAPE: {lr_mape}, MSPE: {

    rf_mse = mean_squared_error(y_test.iloc[:, i], rf_y_pred[:, i])
    rf_mae = mean_absolute_error(y_test.iloc[:, i], rf_y_pred[:, i])
    rf_mape = mean_absolute_percentage_error(y_test.iloc[:, i], rf_y_pred[:, i])
    rf_mspe = mean_squared_percentage_error(y_test.iloc[:, i], rf_y_pred[:, i])
    rf_r2 = r2_score(y_test.iloc[:, i], rf_y_pred[:, i])
    rf_mse_scores.append(rf_mse)
    rf_mae_scores.append(rf_mae)
    rf_mape_scores.append(rf_mape)
    rf_mspe_scores.append(rf_mspe)
    rf_r2_scores.append(rf_r2)
    print(f'Random Forest - {area_name} - MSE: {rf_mse}, MAE: {rf_mae}, MAPE: {rf_mape}, MSPE: {rf_m
```

The evaluation results highlight the strengths and weaknesses of each model for predicting different ship system areas, providing insights into model suitability based on system complexity and data characteristics.

Sensitivity analysis

Sensitivity analysis

```
def sensitivity_analysis(model, original_data, feature_names, target_names, perturbation_scale=0.01)
    sensitivity_results = {target: [] for target in target_names}
    original_predictions = model.predict(original_data)

    for i, feature in enumerate(feature_names):
        perturbation = original_data[0, i] * perturbation_scale
        if np.isnan(perturbation) or perturbation == 0:
            perturbation = perturbation_scale * np.nanstd(original_data[:, i])

        perturbed_data = original_data.copy()
        perturbed_data[0, i] += perturbation

        perturbed_predictions = model.predict(perturbed_data)

        for j, target in enumerate(target_names):
            sensitivity = np.abs((perturbed_predictions[0, j] - original_predictions[0, j]) / origin
            sensitivity_results[target].append(sensitivity)

    return sensitivity_results
```

Sensitivity analysis is conducted to assess the impact of disturbances in input features on the model predictions. This analysis identifies which features most significantly affect the target outputs, providing valuable insights into the relative importance of different design parameters[Saltelli, 2008].

Results

Visualization of sensitivity analysis

```
def plot_sensitivity_analysis(lr_sensitivity, rf_sensitivity, feature_names, target_names):
    area_labels = [f'Area {i+1} - {system_names[i].replace(" System", "")}' for i in range(len(targe
```

```
    for target, area_name in zip(target_names, area_labels):
        lr_values = lr_sensitivity[target]
        rf_values = rf_sensitivity[target]

        x = np.arange(len(feature_names))  # the label locations
        width = 0.35  # the width of the bars

        fig, ax = plt.subplots()
        rects1 = ax.bar(x - width/2, lr_values, width, label='Linear Regression')
        rects2 = ax.bar(x + width/2, rf_values, width, label='Random Forest')

        ax.set_xlabel('Features')
        ax.set_ylabel('Sensitivity (%)')
        ax.set_title(f'Sensitivity Analysis for {area_name}')
        ax.set_xticks(x)
        ax.set_xticklabels(feature_names, rotation=45, ha="right")
        ax.legend()

        fig.tight_layout()
        plt.show()
```

The results of the sensitivity analysis are visualized, comparing the sensitivity of each feature across the Linear Regression and Random Forest models. This analysis helps identify which features the models rely on most heavily, indicating areas where design parameters exert significant influence over system area predictions.

Visualization of evaluation metrics

```
def plot_metric_scores(lr_scores, rf_scores, target_names, metric_name, ylabel):
    area_labels = [f'Area {i+1} - {system_names[i].replace(" System", "")}' for i in range(len(targe

    x = np.arange(len(target_names))  # the label locations
    width = 0.35  # the width of the bars

    fig, ax = plt.subplots()
    rects1 = ax.bar(x - width/2, lr_scores, width, label='Linear Regression')
    rects2 = ax.bar(x + width/2, rf_scores, width, label='Random Forest')

    ax.set_xlabel('Target Areas')
    ax.set_ylabel(ylabel)
    ax.set_title(f'{metric_name} Scores for Each Area')
    ax.set_xticks(x)
    ax.set_xticklabels(area_labels, rotation=45, ha="right")
    ax.legend()

    fig.tight_layout()
    plt.show()
```

The evaluation metrics are plotted for each area, providing a visual comparison of model performance across the different ship systems.

E

Enhanced Regression Prediction Model Script

This enhanced script builds upon the initial machine learning methodology by incorporating cross-validation and bootstrap sampling techniques to improve the robustness and reliability of the predictive models. The enhanced approach uses Linear Regression and Random Forest models, evaluated using combined cross-validation and bootstrap sampling strategies, to predict ship system areas based on design parameters. The improvements are grounded in advanced statistical methods that enhance the generalizability of the model results[Efron and Tibshirani, 1997].

Methodology Enhancements

Cross-Validation and Bootstrap Integration

Cross-Validation Setup

```
from sklearn.model_selection import KFold

# Define cross-validation strategy
kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

Instead of using a single train-test split, the enhanced script employs K-Fold Cross-Validation with 5 folds, shuffling the data to ensure that the model evaluations are robust and less dependent on a single data split. This technique reduces the risk of overfitting and provides a more reliable estimate of the model's performance on unseen data.

Bootstrap Sampling Within Cross-Validation

```
from sklearn.utils import resample

def bootstrap_cross_val_predict(model, X, y, cv, n_bootstrap=100):
    bootstrap_predictions = np.zeros((len(X), y.shape[1]))

    for train_idx, test_idx in cv.split(X):
        X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
        y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]

        # Bootstrap sampling
        fold_predictions = []
        for _ in range(n_bootstrap):
            X_resampled, y_resampled = resample(X_train, y_train)
            model.fit(X_resampled, y_resampled)
            fold_predictions.append(model.predict(X_test))
```

```
        # Average predictions over bootstrap samples
        mean_fold_predictions = np.mean(fold_predictions, axis=0)
        bootstrap_predictions[test_idx] = mean_fold_predictions

    return bootstrap_predictions
```

Bootstrap sampling is integrated within each fold of the cross-validation process, where the training data is resampled multiple times to create diverse datasets for model training. The predictions from these bootstrap samples are averaged, which helps in stabilizing the model's performance metrics and reduces variance[Efron and Tibshirani, 1997]. This method combines the strengths of both cross-validation and bootstrap, making the model evaluation more robust.

Implementation of Cross-Validation with Bootstrap Sampling

```
# Create models
lr_model = MultiOutputRegressor(LinearRegression())
rf_model = MultiOutputRegressor(RandomForestRegressor(random_state=42))

# Use bootstrap sampling within cross-validation for predictions
lr_y_pred_cv_bootstrap = bootstrap_cross_val_predict(lr_model, X, y, kf, n_bootstrap=100)
rf_y_pred_cv_bootstrap = bootstrap_cross_val_predict(rf_model, X, y, kf, n_bootstrap=100)
```

Both Linear Regression and Random Forest models are evaluated using the combined cross-validation and bootstrap sampling approach. By applying this method, the models benefit from reduced sensitivity to random variations in the data, leading to more reliable performance metrics that better reflect the models' expected real-world performance.

Enhanced Performance Evaluation

Calculation of Evaluation Metrics

```
# Evaluate both models for each area and store results
lr_mse_scores = []
rf_mse_scores = []
lr_mae_scores = []
rf_mae_scores = []
lr_mape_scores = []
rf_mape_scores = []
lr_mspe_scores = []
rf_mspe_scores = []
lr_r2_scores = []
rf_r2_scores = []

print("Model Performance Evaluation with Cross-Validation and Bootstrap Sampling:")
for i in range(y.shape[1]):
    area_name = f'Area {i+1} - {system_names[i]}'

    # Linear Regression metrics
    lr_mse = mean_squared_error(y.iloc[:, i], lr_y_pred_cv_bootstrap[:, i])
    lr_mae = mean_absolute_error(y.iloc[:, i], lr_y_pred_cv_bootstrap[:, i])
    lr_mape = mean_absolute_percentage_error(y.iloc[:, i], lr_y_pred_cv_bootstrap[:, i])
    lr_mspe = mean_squared_percentage_error(y.iloc[:, i], lr_y_pred_cv_bootstrap[:, i])
    lr_r2 = r2_score(y.iloc[:, i], lr_y_pred_cv_bootstrap[:, i])
    lr_mse_scores.append(lr_mse)
    lr_mae_scores.append(lr_mae)
    lr_mape_scores.append(lr_mape)
    lr_mspe_scores.append(lr_mspe)
    lr_r2_scores.append(lr_r2)
    print(f'Linear Regression with Cross-Validation and Bootstrap -
```

```
    {area_name} - MSE: {lr_mse}, MAE: {lr_mae}, MAPE: {lr_mape}, MSPE:
    {lr_mspe}, R^2: {lr_r2}')

    # Random Forest metrics
    rf_mse = mean_squared_error(y.iloc[:, i], rf_y_pred_cv_bootstrap[:, i])
    rf_mae = mean_absolute_error(y.iloc[:, i], rf_y_pred_cv_bootstrap[:, i])
    rf_mape = mean_absolute_percentage_error(y.iloc[:, i], rf_y_pred_cv_bootstrap[:, i])
    rf_mspe = mean_squared_percentage_error(y.iloc[:, i], rf_y_pred_cv_bootstrap[:, i])
    rf_r2 = r2_score(y.iloc[:, i], rf_y_pred_cv_bootstrap[:, i])
    rf_mse_scores.append(rf_mse)
    rf_mae_scores.append(rf_mae)
    rf_mape_scores.append(rf_mape)
    rf_mspe_scores.append(rf_mspe)
    rf_r2_scores.append(rf_r2)
    print(f'Random Forest with Cross-Validation and Bootstrap - {area_name}
    - MSE: {rf_mse}, MAE: {rf_mae}, MAPE: {rf_mape}, MSPE: {rf_mspe}, R^2:
    {rf_r2}')
```

The evaluation metrics are calculated using the predictions obtained from the combined cross-validation and bootstrap approach. This enhanced evaluation method provides a comprehensive view of each model's performance across different areas, accounting for variability and increasing the reliability of the results.

Prediction and Sensitivity Analysis

Model Training for Final Predictions

```
# Train the models on the full dataset before predicting new data
lr_model.fit(X, y)
rf_model.fit(X, y)

lr_predicted_areas = lr_model.predict(new_ship_data)
rf_predicted_areas = rf_model.predict(new_ship_data)
```

After completing cross-validation with bootstrap sampling, the models are retrained on the full dataset to make final predictions for new data. This step ensures that the models have the benefit of learning from the entire available dataset, optimizing their performance for real-world predictions.

Enhanced Sensitivity Analysis

```
# Perform sensitivity analysis for Linear Regression and Random Forest
lr_sensitivity = sensitivity_analysis(lr_model, new_ship_data, feature_columns, target_columns)
rf_sensitivity = sensitivity_analysis(rf_model, new_ship_data, feature_columns, target_columns)
```

The sensitivity analysis remains a key component of the enhanced script, allowing for the evaluation of how perturbations in input features affect model predictions. This analysis continues to provide valuable insights into the importance of various design parameters, now based on more robustly evaluated models.

Results and Visualization

Visualization of Sensitivity and Performance Metrics

```
# Plotting the sensitivity analysis results
plot_sensitivity_analysis(lr_sensitivity, rf_sensitivity, feature_columns, target_columns)

# Plotting the MSE, MAE, MAPE, MSPE, and R² scores
plot_metric_scores(lr_mse_scores, rf_mse_scores, target_columns, 'MSE', 'MSE')
plot_metric_scores(lr_mae_scores, rf_mae_scores, target_columns, 'MAE', 'MAE')
plot_metric_scores(lr_mape_scores, rf_mape_scores, target_columns, 'MAPE', 'MAPE (%)')
plot_metric_scores(lr_mspe_scores, rf_mspe_scores, target_columns, 'MSPE', 'MSPE (%)')
plot_metric_scores(lr_r2_scores, rf_r2_scores, target_columns, 'R²', 'R²')
```

The enhanced script continues to provide detailed visualizations of both sensitivity analysis and model performance metrics. By integrating cross-validation and bootstrap sampling, these visualizations are now based on more stable and generalizable evaluation metrics, providing clearer guidance for decision-making based on the models' predictions.

F

Bootstrap Sampling & Cross-Validation

Bootstrap Sampling

Bootstrap Sampling is a resampling technique where data is randomly sampled with replacement from the original dataset to create a new dataset of the same size. It was introduced by Bradley Efron in 1979 as a way to estimate the sampling distribution of a statistic without needing to assume anything about the population distribution.
Bootstrap Sampling can be used to estimate uncertainty because it allows to estimate the variability of a models performance by repeatedly training it on different resampled versions of the data set. It cna also create multiple different datasets from a single dataset, even when the dataset is small. This way it helps to handle small datasets. Also Bootstrap Sampling can help to make more stable predictions by averaging predictions over many bootstrapped samples and reducing the effect of outliers or noise in the data. [Davison and Hinkley, 1997]

Working of Bootstrap Sampling

1. Assume you have a dataset $Z = \{x_1, x_2, ..., x_n\}$ of size n, where each x_i is an observation (a row in the dataset).
2. Create a new dataset Z^* of size n by randomly sampling with replacement from the original dataset Z. This means:
    - Each data point from the original dataset has an equal probability of being selected
    - Because of sampling with replacement, the same data point (row) can appear multiple times in the resampled dataset.

   For example, if $Z = \{x_1, x_2, x_3, x_4\}$, a possible bootstrapped sample could be $Z^* = \{x_2, x_2, x_4, x_1\}$
3. The process of resampling is repeated multiple times, creating multiple bootstrapped datasets $Z^{*1}, Z^{*2}, ..., Z^{*B}$, where B is the number of bootstrap iterations.

   Each bootstrapped dataset Z^{*b} can be used to train a new model, and the predictions from all models can be aggregated.
4. The training of the machine learning model on each of the dataset will have a different model for each dataset since each Z^{*b} is slightly different from the original Z.
5. After training the model on all bootstrapped samples, aggregating the predictions forms the final prediction. The bootstrap process begins with a statistic that has the interest ($\hat{a}^*$). A large number (B) of bootstrap samples are independently drawn. Each bootstrap sample is used to compute this statistic $\{\hat{a}^{*1}, \hat{a}^{*2}, ..., \hat{a}^{*B}\}$. All these bootstrapped data sets can be used to compute the

standard error of this desired statistic following:

$$\hat{SE}_B = \sqrt{\frac{\sum_{b=1}^{B}(\hat{a}^* - \overline{a}^{*B}{}^{2}}{)}(B-1} \tag{F.1}$$

Thus, $\hat{SE}_B$ serves as an estimate of the standard error of $\hat{a}$ estimated from the original dataset.

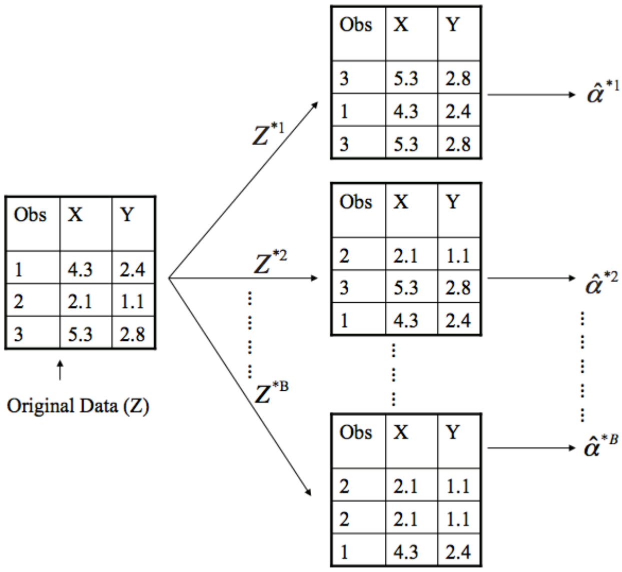


Figure F.1: Bootstrap Sampling

Cross-Validation

Cross-Validation is a technique for assessing the generalization performance of a model. It involves splitting the dataset into multiple subsets (folds), training the model on some subsets, and testing it on the remaining subsets. The key advantage is that every data point gets to be used for both training and testing, but never at the same time. most common used type of cross-validation is k-fold cross-validation where the dataset is split into k equal-sized folds. The model is trained on k-1 folds and tested on the remaining fold.

Cross-validation provides a more accurate estimate of how well the model will perform on unseen data. It reduces overfitting to a particular train-test split by ensuring that the model is tested on different subsets of the data. Also cross-validation makes better use of limited data by ensuring that each point is used for both training and testing.

Working of Cross-Validation

1. Assume the original dataset is $D = \{x_1, x_2, ..., x_n\}$, where n is the number of observations. This is the full dataset from which the cross-validation splits will be performed.
2. The dataset is randomly divided into K subsets, also called folds. Each fold contains $\frac{n}{K}$ observations.
3. The model is trained on K-1 folds of the data and tested on the remaining 1 fold. This process is repeated k times, each time using a different fold as the test set and the remaining K-1 folds as the training set.
4. Once al n iterations are complete, the performance scores from all iterations are averaged. This average gives a more robust estimate of the model's perfromance on unseen data, as the model has been tested on all parts of the dataset.
5. After cross-validation is complete, the model can be retrained on the entire dataset to produce a final model for deployment or further analysis. This final model benefits from using all available data for training.

Hybrid Cross-Validation & Bootstrap Sampling approach

In the prediction script, bootstrap sampling is applied within each fold of a K-fold Cross-Validation. This hybrid approach is designed to enhance model evaluation and make predictions more robust.

Working of Hybrid Cross-validation & Bootstrap Sampling

1. First the dataset is devided into training and testing sets using K-fold Cross-Validation.

```
for train_idx, test_idx in cv.split(X):
X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
```

2. Once the training set for a fold is defined (x_train, y_train), bootstrap sampling is applied to this training set to create multiple resampled versions of it.

```
for _ in range(n_bootstrap):
X_resampled, y_resampled = resample(X_train, y_train)
model.fit(X_resampled, y_resampled) fold_predictions.append(model.predict(X_test))
```

3. Once predictions have been made for all 50 bootstrap samples, the script averages the predictions for the test set to get a single robust prediction for each test point in that fold.

```
mean_fold_predictions = np.mean(fold_predictions, axis=0)
bootstrap_predictions[test_idx] = mean_fold_predictions
```

4. The cross-validation process is repeated for all folds. Each fold is used once as the test set, while bootstrap sampling is applied within the remaining k-1 folds to generate robust predictions. Once all folds are processed, the bootstrap_predictions array contains the final averaged predictions for the entire dataset, obtained through cross-validation with bootstrap resampling.

Each training set is modified through bootstrap sampling, which means the model is trained on slightly different versions of the training data during each bootstrap iteration. Some data points may be sampled multiple times, while others may not be included in certain bootstrap samples. The test set for each fold is never modified by bootstrap sampling. So the same test set is used for all 50 bootstrap iterations within that fold, allowing the model to be evaluated consistently across resampled versions of the training data.

By averaging predictions from multiple bootstrap samples, the final prediction becomes less sensitive to small fluctuations in the training data. This reduces the variance in the model's predictions and makes them more stable and reliable. The model is trained on slightly different versions of the training set in each bootstrap iteration, making it less likely to overfit to specific data points. This leads to a more robust model that generalizes better to unseen data. Instead of relying on a single model trained on one version of the training data, the final prediction is based on an ensemble of models trained on different bootstrapped training sets. This approach improves accuracy and robustness of predictions.

G

Performance metrics

```
Model Performance Evaluation:
Linear Regression - Area 1 - Power System - MSE: 4914.4678858972875, MAE: 56.38542847935155, MAPE: 21.2712289595848, MSPE: 7.227410137106995, R^2: -2.4529210876766308
Random Forest - Area 1 - Power System - MSE: 4639.736514013769, MAE: 62.7501064266668, MAPE: 21.771933611184657, MSPE: 5.345795043794544, R^2: -2.2598939340868927
Linear Regression - Area 2 - Propulsion System - MSE: 837.3559006619603, MAE: 25.086373230758344, MAPE: 80.78750098210718, MSPE: 106.41865263485089, R^2: -1.3831519670507242
Random Forest - Area 2 - Propulsion System - MSE: 515.5438273372549, MAE: 14.834165757575759, MAPE: 30.74791038296708, MSPE: 13.074506107274495, R^2: -0.4672605582027545
Linear Regression - Area 3 - Mooring & Anchoring System - MSE: 270.3641065799207, MAE: 13.358238059558722, MAPE: 26.45902610748359, MSPE: 10.714106353736096, R^2: -0.21534706163138595
Random Forest - Area 3 - Mooring & Anchoring System - MSE: 182.5686414521856, MAE: 10.745915380952363, MAPE: 20.19297544214005, MSPE: 5.25390457184712, R^2: 0.17931316866812752
Linear Regression - Area 4 - HVAC System - MSE: 878.6201241335872, MAE: 23.474513098003026, MAPE: 24.457889148239417, MSPE: 9.052680594208647, R^2: -0.03889817422762665
Random Forest - Area 4 - HVAC System - MSE: 1139.0503722113372, MAE: 29.415946960000014, MAPE: 28.614248924390168, MSPE: 9.484112513356298, R^2: -0.3468361576745955
Linear Regression - Area 5 - Extended Operations System - MSE: 443.65003109518995, MAE: 17.99017804699926, MAPE: 25.59710365932989, MSPE: 9.122812697228563, R^2: -7.273478192712258
Random Forest - Area 5 - Extended Operations System - MSE: 390.6487568875319, MAE: 15.489083354166668, MAPE: 21.07313962678251, MSPE: 7.450347345805892, R^2: -6.285075497776042
Linear Regression - Area 6 - Safety and Compliance System - MSE: 241.1794322665797, MAE: 10.90935458685673, MAPE: 46.32114533884021, MSPE: 27.590007503939447, R^2: -0.08448594709468438
Random Forest - Area 6 - Safety and Compliance System - MSE: 317.44511054692566, MAE: 12.49468334057971, MAPE: 48.664102649916956, MSPE: 29.385864523974014, R^2: -0.42742172550409596
Linear Regression - Area 7 - Wheelhouse System - MSE: 186.95328663065087, MAE: 9.703550666852394, MAPE: 13.896339959745665, MSPE: 3.1396125709497187, R^2: -0.6294162490978847
Random Forest - Area 7 - Wheelhouse System - MSE: 155.03058220989863, MAE: 8.699736, MAPE: 12.063885816429543, MSPE: 2.6190486866142515, R^2: -0.35118967049226035
Linear Regression - Area 8 - Crew Operations System - MSE: 941.3488175036979, MAE: 24.56348824556711, MAPE: 13.544535077815443, MSPE: 3.413703928002767, R^2: 0.11202029223156984
Random Forest - Area 8 - Crew Operations System - MSE: 580.0960157447992, MAE: 17.606929466666745, MAPE: 8.226464009096633, MSPE: 1.226614220280576, R^2: 0.45279212024221416
Linear Regression - Area 9 - Crew Accommodation System - MSE: 347.07748061916703, MAE: 14.580987953077576, MAPE: 11.972240562557728, MSPE: 2.9493833314824167, R^2: 0.6313918033450749
Random Forest - Area 9 - Crew Accommodation System - MSE: 77.14421044409386, MAE: 6.232913666666626, MAPE: 5.463660595421267, MSPE: 0.7219929806966952, R^2: 0.9180702007994375
Linear Regression - Area 10 - Tender Garage System - MSE: 2563.671766145058, MAE: 38.00991665667585, MAPE: 27.27641638069497, MSPE: 9.05346091039812, R^2: 0.04692238825971995
Random Forest - Area 10 - Tender Garage System - MSE: 2334.7209796296847, MAE: 35.50656821333333, MAPE: 28.659115319085117, MSPE: 13.215487403961177, R^2: 0.13203775743439605
Linear Regression - Area 11 - Beachclub System - MSE: 431.5811395692453, MAE: 18.277108031266494, MAPE: 23.542022267830205, MSPE: 7.2581641928974925, R^2: -5.980493179718514
Random Forest - Area 11 - Beachclub System - MSE: 190.9787132078459, MAE: 10.035938395833332, MAPE: 12.106074387504982, MSPE: 2.557824949245852, R^2: -2.0889338823966206
Linear Regression - Area 12 - Dining & Galley System - MSE: 3363.554951761939, MAE: 52.42683207998144, MAPE: 113.77857240384436, MSPE: 193.71174680489852, R^2: -7.09632379530904
Random Forest - Area 12 - Dining & Galley System - MSE: 2657.7365964923915, MAE: 43.61487467333334, MAPE: 112.11982232479374, MSPE: 207.24593599977123, R^2: -5.397367177418416
Linear Regression - Area 13 - Owner Accommodation System - MSE: 31861.238154342504, MAE: 161.22688090842306, MAPE: 91.30415369699378, MSPE: 121.01926145922491, R^2: -1.3938293610359591
Random Forest - Area 13 - Owner Accommodation System - MSE: 23607.090420283534, MAE: 143.79073318666659, MAPE: 91.48403720628481, MSPE: 140.92543151905198, R^2: -0.7736707501118649
Linear Regression - Area 14 - Guest Accommodation System - MSE: 436.16654699756856, MAE: 19.8038943802664, MAPE: 10.988135734277703, MSPE: 1.3849472579263924, R^2: 0.43640559779283483
Random Forest - Area 14 - Guest Accommodation System - MSE: 881.5682987661309, MAE: 27.29649016666681, MAPE: 14.18102645079273, MSPE: 2.2195504283921363, R^2: -0.13912211233993332
```

Figure G.1: Performance metrics initial regression methods

```
Model Performance Evaluation with Cross-Validation and Bootstrap Sampling:
Linear Regression with Cross-Validation and Bootstrap - Area 1 - Power System - MSE: 7716.903193871593, MAE: 69.57361075196428, MAPE: 28.121469733444705, MSPE: 13.159330359111362, R^2: -0.28295091931534566
Random Forest with Cross-Validation and Bootstrap - Area 1 - Power System - MSE: 4969.638529775655, MAE: 59.42991115673844, MAPE: 24.365213917123167, MSPE: 9.709687600074268, R^2: 0.17378744293384363
Linear Regression with Cross-Validation and Bootstrap - Area 2 - Propulsion System - MSE: 434.2888848381496, MAE: 15.907236621727163, MAPE: 59.199751512616004, MSPE: 76.32717111835153, R^2: -0.7586909531371511
Random Forest with Cross-Validation and Bootstrap - Area 2 - Propulsion System - MSE: 290.5031497472401, MAE: 11.661969158881123, MAPE: 37.94715845925048, MSPE: 27.16955269313962, R^2: -0.17641799998800778
Linear Regression with Cross-Validation and Bootstrap - Area 3 - Mooring & Anchoring System - MSE: 234.86066258372972, MAE: 13.158935068338126, MAPE: 29.4083152147477, MSPE: 11.349185080691583, R^2:
0.005939030695402248
Random Forest with Cross-Validation and Bootstrap - Area 3 - Mooring & Anchoring System - MSE: 197.61341611167356, MAE: 11.360210687692312, MAPE: 25.45139942009746, MSPE: 9.316192345805893, R^2:
0.16359009718142703
Linear Regression with Cross-Validation and Bootstrap - Area 4 - HVAC System - MSE: 717.1906433324867, MAE: 21.476946486822648, MAPE: 27.894026226646503, MSPE: 11.469467346845123, R^2: -0.039486570592323966
Random Forest with Cross-Validation and Bootstrap - Area 4 - HVAC System - MSE: 490.12822195030606, MAE: 17.858337676400005, MAPE: 23.990111988752844, MSPE: 8.307801395196366, R^2: 0.28961468011031577
Linear Regression with Cross-Validation and Bootstrap - Area 5 - Extended Operations System - MSE: 775.0812196757817, MAE: 22.299645495308898, MAPE: 41.658715346662035, MSPE: 32.92799356393223, R^2:
-0.6052534135831091
Random Forest with Cross-Validation and Bootstrap - Area 5 - Extended Operations System - MSE: 381.6151758976532, MAE: 15.850838156987171, MAPE: 31.754812212045717, MSPE: 22.694420445829653, R^2:
0.20964532718122897
Linear Regression with Cross-Validation and Bootstrap - Area 6 - Safety and Compliance System - MSE: 230.99456266525223, MAE: 11.498746016437128, MAPE: 118.57952302367573, MSPE: 653.732497796424, R^2:
-0.7733443614328874
Random Forest with Cross-Validation and Bootstrap - Area 6 - Safety and Compliance System - MSE: 168.99840571939907, MAE: 9.70948550130435, MAPE: 93.2859263600759, MSPE: 360.24615510770724, R^2:
-0.29740010507496484
Linear Regression with Cross-Validation and Bootstrap - Area 7 - Wheelhouse System - MSE: 156.24528532904128, MAE: 9.465860218898191, MAPE: 15.691840073926869, MSPE: 4.052459887785422, R^2:
-0.16674422075507134
Random Forest with Cross-Validation and Bootstrap - Area 7 - Wheelhouse System - MSE: 126.41271007091103, MAE: 7.6698910066769335, MAPE: 12.527063004959018, MSPE: 3.103771080038936, R^2: 0.05602720367135661
Linear Regression with Cross-Validation and Bootstrap - Area 8 - Crew Operations System - MSE: 2455.4072324895924, MAE: 35.30091137387389, MAPE: 18.36372347991412, MSPE: 6.740284243310815, R^2:
0.4045393370604604
Random Forest with Cross-Validation and Bootstrap - Area 8 - Crew Operations System - MSE: 2101.081763283291, MAE: 32.64882221861539, MAPE: 16.74551245095661, MSPE: 5.581591508679141, R^2: 0.4904667856719167
Linear Regression with Cross-Validation and Bootstrap - Area 9 - Crew Accommodation System - MSE: 858.095203392566, MAE: 23.459392928405023, MAPE: 18.02446905113331, MSPE: 4.595034685892788, R^2:
0.5539246976970692
Random Forest with Cross-Validation and Bootstrap - Area 9 - Crew Accommodation System - MSE: 591.7964711899535, MAE: 16.24630239215384, MAPE: 12.224761018110158, MSPE: 2.762836270454705, R^2:
0.6923583901364654
Linear Regression with Cross-Validation and Bootstrap - Area 10 - Tender Garage System - MSE: 2335.557984921135, MAE: 39.796161569347504, MAPE: 35.60566809199754, MSPE: 18.162882073574412, R^2:
-0.4519016131666389
Random Forest with Cross-Validation and Bootstrap - Area 10 - Tender Garage System - MSE: 1482.6606770557014, MAE: 28.382888442830758, MAPE: 27.65349097770996, MSPE: 16.101731652774212, R^2:
0.07830272564669216
Linear Regression with Cross-Validation and Bootstrap - Area 11 - Beachclub System - MSE: 851.1053387456583, MAE: 23.364937827368784, MAPE: 48.185153263270614, MSPE: 83.13322741915677, R^2: -0.6816352335625018
Random Forest with Cross-Validation and Bootstrap - Area 11 - Beachclub System - MSE: 691.8560490373388, MAE: 18.54628453548077, MAPE: 43.46419370513676, MSPE: 100.51497957229363, R^2: -0.36698650055374227
Linear Regression with Cross-Validation and Bootstrap - Area 12 - Dining & Galley System - MSE: 4888.761341381268, MAE: 56.83275063779339, MAPE: 107.55922865001469, MSPE: 189.45405852081367, R^2:
-0.901494599217403
Random Forest with Cross-Validation and Bootstrap - Area 12 - Dining & Galley System - MSE: 3018.171995656728, MAE: 43.62245051763077, MAPE: 91.65041596066096, MSPE: 141.30444162477676, R^2:
-0.17392471190442182
Linear Regression with Cross-Validation and Bootstrap - Area 13 - Owner Accommodation System - MSE: 18813.751535526604, MAE: 111.20534368824949, MAPE: 82.21487151489512, MSPE: 136.69103302514145, R^2:
-0.514657215829786
Random Forest with Cross-Validation and Bootstrap - Area 13 - Owner Accommodation System - MSE: 13270.79677299607, MAE: 96.27504430381539, MAPE: 70.64940990518275, MSPE: 108.29359109781262, R^2:
-0.06840510007120737
Linear Regression with Cross-Validation and Bootstrap - Area 14 - Guest Accommodation System - MSE: 992.6573021604622, MAE: 25.68362969303733, MAPE: 15.94726329292214, MSPE: 3.825880801539647, R^2:
0.25728884861508494
Random Forest with Cross-Validation and Bootstrap - Area 14 - Guest Accommodation System - MSE: 1337.1787380041621, MAE: 27.34511580569231, MAPE: 19.607338529331003, MSPE: 11.175172414908316, R^2:
-0.0004838104237903096
```

Figure G.2: Performance metrics bootstrap and cross-validation regression methods

H

Heuristic Packing Algorithm Script

This chapter provides a detailed explanation of the script developed to generate the arrangement of yacht systems using a heuristic packing approach. The script incorporates user preferences, spatial constraints, and dynamic scoring to generate and evaluate multiple arrangements, ensuring that yacht systems are placed in an optimal manner based on various design considerations.

Methodology

Data Loading and Preprocessing

Loading Preferences from Excel

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
import logging

# Suppress detailed INFO logs
logging.basicConfig(level=logging.CRITICAL, format='%(levelname)s:%(message)s')

# Function to read preferences from Excel
def read_preferences_from_excel(filepath):
    df = pd.read_excel(filepath, sheet_name=0)
    preferences = {
        'height_preference': [],  # High, Low, or None
        'length_preference': [],  # Bow or Stern
        'close_to_systems': [],
        'far_from_systems': []
    }
    for i in range(len(df)):
        # Extract height preference
        height_pref = df.iloc[i]['Height preference:']
        height_score = df.iloc[i]['Height score:']
        if height_pref == 'High':
            preferences['height_preference'].append(('High', height_score))
        elif height_pref == 'Low':
            preferences['height_preference'].append(('Low', height_score))
        else:
            preferences['height_preference'].append((None, 0))  # No preference

        # Extract length preference
```

```
        length_pref = df.iloc[i]['Length preference:']
        length_score = df.iloc[i]['Length score:']
        if length_pref == 'Bow':
            preferences['length_preference'].append(('Bow', length_score))
        elif length_pref == 'Stern':
            preferences['length_preference'].append(('Stern', length_score))
        else:
            preferences['length_preference'].append((None, 0))  # No preference

        # Extract close to systems preferences
        close_systems = {}
        for j in range(1, 7):
            close_system = df.iloc[i][f'Close to System {j}:']
            close_score = df.iloc[i][f'Close to Score {j}:']
            if pd.notna(close_system) and close_system != '-':
                system_index = df[df['System'] == close_system].index[0]
                close_systems[system_index] = close_score
        preferences['close_to_systems'].append(close_systems)

        # Extract far from systems preferences
        far_systems = {}
        for j in range(1, 7):
            far_system = df.iloc[i][f'Far from System {j}:']
            far_score = df.iloc[i][f'Far from Score {j}:']
            if pd.notna(far_system) and far_system != '-':
                system_index = df[df['System'] == far_system].index[0]
                far_systems[system_index] = far_score
        preferences['far_from_systems'].append(far_systems)

    return preferences
```

The script begins by importing necessary libraries for data manipulation, visualization, and randomization. It reads system placement preferences from an Excel file, which includes height preferences, length preferences, and proximity constraints. The extracted preferences are stored in a dictionary format, enabling structured access during subsequent computations.

Configuration Setup

Defining the Configuration Dictionary

```
# Define the configuration dictionary
config = {
    'ship': {
        'length': 88,
        'width': 14.5,
        'deck_height': 3,
        'decks': ['Lower Deck', 'Main Deck', 'Bridge Deck', 'Owner Deck']
    },
    'systems': {
        'areas': [260.11441804,  31.58688445,  51.94374071,  82.46643338,  66.50703283,
           17.44829461,  82.75784908, 239.68814354, 242.86259685, 127.18351904,
          147.55150076,  47.90980946, 204.32898304, 248.2602    ],
        'names': [
            "Power System", "Propulsion System", "Mooring & Anchoring System", "HVAC System",
            "Extended Operations System",
            "Safety and Compliance System", "Wheelhouse System", "Crew Operations System",
            "Crew Accommodation System",
            "Tender Garage System", "Beachclub System", "Dining & Galley System",
```

```
            "Owner Accommodation System", "Guest Accommodation System"
        ],
        'separable': [False, False, False, True, True, True, False, False, False,
        False, False, False, False, False],
        'num_blocks': [1, 1, 1, 6, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1],
        'preferences': {},  # This will be populated after reading the Excel file
        'margins': [0, 5, 10, 0, 15, 0, 5, 20, 0, 10, 5, 0, 15, 10]
    }
}
# Load preferences from Excel file
file_path = r"C:\Users\jaap.marcus\OneDrive - De Voogt Naval Architects\Graduation Jaap
Marcus\Onderzoek\Packing Tool\System_Scoring_Interface_Final_V9.xlsx"
preferences = read_preferences_from_excel(file_path)
config['systems']['preferences'] = preferences
```

The configuration dictionary defines the yacht's parameters, including its length, width, deck height, and deck names. It also specifies system attributes, such as their areas, names, separability (whether systems can be split into multiple blocks), and individual margins. The preferences extracted from the Excel file are integrated into this configuration to guide the arrangement process.

Feasibility Checks and Area Adjustments

Adjusting Areas and Calculating System Lengths

```
def adjust_areas(areas, margins):
    return [area * (1 + margin / 100) for area, margin in zip(areas, margins)]

def calculate_lengths(areas, width):
    return [area / width for area in areas]
```

The script adjusts system areas using predefined margins to account for spatial tolerances. It then calculates system lengths by dividing these adjusted areas by the ship's width. These lengths represent the linear space required by each system along the yacht's length.

Feasibility Function

```
def check_feasibility(system_lengths, ship_length, num_decks, deck_height, ship_width):
    total_system_length = sum(system_lengths)
    total_available_length = ship_length * num_decks

    if total_system_length > total_available_length:
        raise ValueError("Configuration not feasible: Total system length exceeds
        available ship space.")
```

A feasibility check is performed to ensure that the total length of all systems does not exceed the available space across the yacht's decks. This function compares the sum of system lengths with the total available length, defined as the product of the ship's length and the number of decks. If the total system length exceeds the available space, the configuration is deemed infeasible, and an error is raised.

Dynamic Scoring and Placement

Dynamic Scoring Function

```
ef dynamic_score_placement(system_idx, x, z, current_positions, preferences, deck_height,
ship_length, decks):
    score = 0
    deck_idx = int(z // deck_height)
    if deck_idx >= len(decks):
        return -1  # Out of bounds

    # Height preference score
```

```
    height_pref, height_weight = preferences["height_preference"][system_idx]
    if height_pref == 'High' and deck_idx >= 2:  # Bridge or Owner deck
        score += height_weight * 100  # Increased weight for height preference

    elif height_pref == 'Low' and deck_idx < 2:  # Lower or Main deck
        score += height_weight * 100  # Increased weight for height preference

    # Length preference score
    length_pref, length_weight = preferences["length_preference"][system_idx]
    if length_pref == 'Stern':
        score += (3 / (x + 1)) * length_weight * 100  # Stern preference
    elif length_pref == 'Bow':
        score += (3 / (ship_length - x + 1)) * length_weight * 100
        # Bow preference

    # Close to systems score
    close_to_systems = preferences["close_to_systems"][system_idx]
    for close_system_idx, close_score in close_to_systems.items():
        for placed_idx, placed_x, placed_z, placed_length in current_positions:
            if placed_idx == close_system_idx:
                distance = abs(placed_x - x) + abs(placed_z - z)
                score += close_score / (distance + 1) * 20  # Adjusted weight
                for close to systems

    # Far from systems score
    far_from_systems = preferences["far_from_systems"][system_idx]
    for far_system_idx, far_score in far_from_systems.items():
        for placed_idx, placed_x, placed_z, placed_length in current_positions:
            if placed_idx == far_system_idx:
                distance = abs(placed_x - x) + abs(placed_z - z)
                score += far_score * (distance + 1) * 20  # Adjusted weight for
                far from systems

    return score
```

The dynamic_score_placement function evaluates the placement of each system based on height, length, and proximity preferences. Height Preferences: Systems are scored higher if placed on preferred decks (e.g., upper decks for ”High” preference). Length Preferences: Scores are adjusted based on proximity to the bow or stern, as preferred. Proximity Preferences: Additional scores are added or subtracted based on whether systems are placed near or far from specified systems. The scoring incorporates user-defined weights, allowing customization of the importance of each preference, and adjusts placements to enhance the total score of the arrangement.

Heuristic Packing and Arrangement Generation

Heuristic Packing Algorithm

```
def heuristic_packing(length, width, heights, system_lengths, preferences, separable,
num_blocks, decks, system_names):
    current_positions = []

    for idx in range(len(system_lengths)):
        best_position = None
        best_score = float('-inf')

        # Handle separable systems by dividing them into multiple blocks
        if separable[idx]:
            block_length = system_lengths[idx] / num_blocks[idx]
```

```
            for block in range(num_blocks[idx]):
                for deck in decks:
                    deck_idx = decks.index(deck)
                    z = deck_idx * heights
                    for x in np.linspace(0, length - block_length, num=100):
                    # Evaluate positions in small steps
                        if not check_overlap(x, z, block_length, current_positions,
                        heights, length):
                            # Introduce a small random factor to create variations
                            in the arrangements
                            score = dynamic_score_placement(idx, x, z, current_positions,
                            preferences, heights, length, decks) + random.uniform(-10, 10)

                            if score > best_score:
                                best_position = (idx, x, z, block_length)
                                best_score = score
                if best_position:
                    current_positions.append(best_position)
        else:
            for deck in decks:
                deck_idx = decks.index(deck)
                z = deck_idx * heights
                for x in np.linspace(0, length - system_lengths[idx], num=100):
                # Evaluate positions in small steps
                    if not check_overlap(x, z, system_lengths[idx], current_positions,
                    heights, length):
                        # Introduce a small random factor to create variations in the
                        arrangements
                        score = dynamic_score_placement(idx, x, z, current_positions,
                        preferences, heights, length, decks) + random.uniform(-10, 10)
                        if score > best_score:
                            best_position = (idx, x, z, system_lengths[idx])
                            best_score = score
            if best_position:
                current_positions.append(best_position)

    total_score = sum(dynamic_score_placement(idx, x, z, current_positions, preferences,
    heights, length, decks)
                      for idx, x, z, _ in current_positions)
    return current_positions, total_score
```

The heuristic packing algorithm generates feasible arrangements by iteratively evaluating potential placements and selecting the configurations with the highest scores. For separable systems, the function divides them into smaller blocks and optimizes their distribution across decks. A random factor is introduced to the scores to create variability and avoid uniform arrangements.

Generate Arrangements

```
def generate_arrangements(num_arrangements, length, width, heights, system_lengths,
preferences, separable, num_blocks, decks, system_names):
    all_arrangements = []
    all_scores = []

    for _ in range(num_arrangements):
        arrangement, score = heuristic_packing(length, width, heights, system_lengths,
        preferences, separable, num_blocks, decks, system_names)
        all_arrangements.append(arrangement)
```

```
        all_scores.append(score)

    return all_arrangements, all_scores
```

This function generates multiple arrangements using the heuristic packing approach, storing each arrangement and its corresponding score for comparative analysis. It repeats the arrangement generation process for a specified number of iterations, allowing for a diverse set of configurations.

Visualization and Comparative Analysis

Visualization Function

```
dedef visualize_packing(ship_length, deck_height, packed_positions, system_names,
system_areas, arrangement_index):
    plt.figure(figsize=(15, 10))
    colors = plt.cm.get_cmap("tab20", len(system_areas))
    legend_labels = []

    for system_idx, x, z, length in packed_positions:
        plt.fill_between([x, x + length], [z, z], [z + deck_height, z + deck_height],
                         color=colors(system_idx), edgecolor='black', label=f'System
                         {system_idx + 1} - {system_names[system_idx]}' if system_idx
                         not in legend_labels else "")
        if system_idx not in legend_labels:
            legend_labels.append(system_idx)

    plt.xlim(0, ship_length)
    plt.ylim(0, deck_height * len(config['ship']['decks']))
    plt.xlabel("X (Length of the ship)")
    plt.ylabel("Z (Height from bottom to top deck)")
    plt.title(f"2D Packing of Ship Systems in XZ-plane (Arrangement {arrangement_index + 1})")
    plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.05), ncol=2, fontsize='small')
    plt.show()
```

The visualize_packing function creates 2D visualizations of the generated arrangements, displaying the system placements across the yacht's decks. The visual representation aids in assessing the spatial distribution of systems and verifying the feasibility of the arrangements.

Main Execution and Comparative Analysis

```
if __name__ == "__main__":
    # Check feasibility before attempting to place systems
    adjusted_areas = adjust_areas(config['systems']['areas'], config['systems']['margins']
    system_lengths = calculate_lengths(adjusted_areas, config['ship']['width']
    check_feasibility(system_lengths, config['ship']['length'], len(config['ship']['decks']),
    config['ship']['deck_height'], config['ship']['width'])

    # Generate arrangements with the original ship length
    num_arrangements = 5  # Number of arrangements to generate
    all_arrangements, all_scores = generate_arrangements(
        num_arrangements, config['ship']['length'], config['ship']['width'], config['ship']
        ['deck_height'],
        system_lengths, config['systems']['preferences'], config['systems']['separable'],
        config['systems']['num_blocks'], config['ship']['decks'], config['systems']['names']

    )

    total_scores = []
```

```
for i, (arrangement, score) in enumerate(zip(all_arrangements, all_scores)):
    print(f"\nArrangement {i + 1}:")
    total_score = 0
    for system_idx, x, z, length in arrangement:
        system_score = dynamic_score_placement(system_idx, x, z, arrangement, config['systems'][
        ['length'], config['ship']['decks'])
        total_score += system_score
        system_name = config['systems']['names'][system_idx]
        print(f"  System '{system_name}' placed at (X: {x:.2f}, Z: {z:.2f}) with
        length {length:.2f} -> Score: {system_score:.2f}")
    total_scores.append(total_score)
    print(f"  Total Score for Arrangement {i + 1}: {total_score:.2f}")
    visualize_packing(config['ship']['length'], config['ship']['deck_height'],
    arrangement, config['systems']['names'], adjusted_areas, i)

# Calculate and display the average score
average_score = sum(total_scores) / len(total_scores)
print(f"\nAverage Score of All Arrangements: {average_score:.2f}")
```

In the main execution block, the script adjusts system areas, calculates lengths, and performs feasibility checks before generating arrangements. It then compares the total and average scores of the generated configurations. The script also evaluates optimized configurations using 90% of the original ship length, providing insights into how reduced space impacts system arrangement quality.

Optimized Configurations

```
# Generate arrangements with 90% of the original ship length
optimized_length = config['ship']['length'] * 0.9
logging.info(f"Generating additional arrangements with 90% of the original ship length:
{optimized_length} meters.")

# Check feasibility for the optimized length
check_feasibility(system_lengths, optimized_length, len(config['ship']['decks']),
config['ship']['deck_height'], config['ship']['width'])

# Generate arrangements with the optimized ship length
optimized_arrangements, optimized_scores = generate_arrangements(
    num_arrangements, optimized_length, config['ship']['width'], config['ship']
    ['deck_height'],
    system_lengths, config['systems']['preferences'], config['systems']['separable'],
    config['systems']['num_blocks'], config['ship']['decks'], config['systems']['names']
)

optimized_total_scores = []

for i, (arrangement, score) in enumerate(zip(optimized_arrangements, optimized_scores)):
    print(f"\nOptimized Arrangement {i + 1}:")
    total_score = 0
    for system_idx, x, z, length in arrangement:
        system_score = dynamic_score_placement(system_idx, x, z, arrangement,
        config['systems']['preferences'], config['ship']['deck_height'], optimized_length,
        config['ship']['decks'])
        total_score += system_score
        system_name = config['systems']['names'][system_idx]
        print(f"  System '{system_name}' placed at (X: {x:.2f}, Z: {z:.2f}) with length
        {length:.2f} -> Score: {system_score:.2f}")
    optimized_total_scores.append(total_score)
```

```
        print(f"  Total Score for Optimized Arrangement {i + 1}: {total_score:.2f}")
        visualize_packing(optimized_length, config['ship']['deck_height'], arrangement,
        config['systems']['names'], adjusted_areas, i + num_arrangements)

    # Calculate and display the average score for optimized arrangements
    optimized_average_score = sum(optimized_total_scores) / len(optimized_total_scores)
    print(f"\nAverage Score of All Optimized Arrangements: {optimized_average_score:.2f}")
```

Optimized arrangements are generated by reducing the ship's length to 90% of its original size, allowing the script to evaluate how well the systems can still be placed efficiently under spatial constraints. These optimized configurations are scored and compared with the original ones to determine their feasibility and overall effectiveness.

I

Arrangement Conditions

System	Height preference:	Height score:	Length preference:	Length score:	Close to System 1:
Power System	Low	9	-	0	-
Propulsion System	Low	9	Bow	9	-
Mooring & Anchoring System	Low	6	Bow	9	-
HVAC System	-	0	-	0	Owner Accommodation
Extended Operations System	-	0	-	0	Dining & Galley System
Safety and Compliance System	High	3	-	0	-
Wheelhouse System	High	9	Bow	9	-
Crew Operations System	Low	6	-	0	-
Crew Accommodation System	Low	6	Bow	9	-
Tender Garage & Toy Storage System	-	0	-	0	-
Beachclub system	Low	9	Stern	9	-
Dining & Galley System	High	3	-	0	-
Owner Accommodation System	High	9	Bow	6	-
Guest Accommodation system	High	6	-	0	-

Close to Score 1:	Reasoning:	Far from System 1:	Far from Score 1:	Reasoning:
0	T connection transfers via electricity cables so no need for physical connection to be close to a certain system.	Owner Accommodation :	9	Because of noise generation from power system
0	Bowthruster has to be under waterline at the bow of the ship.	-	0	
0	Mooring has to be on the main deck in the bow of the ship.	-	0	
9	Research done in General arrangements and due to interview with Sr. Specialist B. Jongepier.	-	0	
0	To get resources to the galley	-	0	
0	No preferences but safety and compliance system has to be re-evaluated because it contains the high-fog system and the emergency generator. Hi-fog system is mostly close to engine system but no regulations and emergency generator has to be above the waterline but this has to be checked with Sr. specialist B. Jongepier.	-	0	
0	Bridge has to be at least at the owner deck and preferable as high as possible. This is because the command center is left outside the scope for this research and therefore the captain needs a plain view on the sea and it's surrounding.	-	0	
0	No preferences for the crew operation system, but because no preferences it is expected to find the crew operations on the lower deck.	-	0	
0	No preferences and therefore expected to find the crew operations on the lower deck.	Power System	6	Noise regulations, check with sr. Specialist B.Jongepier.
0	Tender garage needs to be above the waterline and there are no further preferences.	-	0	
0	Beachclub has to be on the lower deck located as close to the stern as possible. Otherwise it would be to far from the water or to close to the bow and thereby causing difficulties with the ship bow structure.	-	0	
0	No regulatory or preferences standard requirements for the dining&galley system	-	0	
0	Owner accommodation positioned on owner or bridge deck as high as possible and preferably facing the bow of the ship. Custom requirement of the client could be facing the stern of the ship.	Power System	9	Because of noise generation from the power system
0	Guest accommodation system has to be placed on the main deck or highter because of the need for big windows. Lower deck can't supply big enough windows. This is concluded due to the fact that there aren't any vessels found with guest accommodation on the lower deck. This has to be checked with Sr. Specialist B. Jongepier.	Power System	9	

Close to System 2:	Close to Sco	Reasoning:	Far from System 2:	Far from	Reasoning:	Close to System 3:	Close to	Reasoning:
			Guest Accommodati	9	Because of noice generation of power system			
Wheelhouse System	9					Guest Accommodati	9	
Crew Operations Sys	0	To get resources to the crew in operation.						

Far from System 3:	Far from Sc	Reasoning:	Close to System 4:	Close to Score 4:	Reasoning:	Far from Sys	Far from Score	Reasoning:
Crew Accommodatio	3	Noise regulations, check with Bram						
			Crew Accommodatio	9				

Close to System 5:	Close to Score	Reasoning:	Far from	Far from Sc	Reasoning:	Close to System 6	Close to Score 6:
Dining & Galley Syste	6					Propulsion System	6

Reasoning:	Far from System 6	Far from Score 6:	Reasoning:
Attachment physical connection between Propulsion system and HVAC, Also same score and reason for crew operations, luxury space and beachclub system.			

J

Modular Arrangement Generator Tool Presentation

Modular Arrangement Generator Tool

A tool to support the designer in the early stage of the design phase.
Capturing all customer requirements and transforms them into modular systems.
Predicting the required area for these modular systems.
And generating optimized arrangement based on these modular systems

Purpose

- Supporting the designer during the early stage design phase
  - Faster Design Iterations
  - Exploring the design phase
  - Improve creativity
  - Supporting the designer
- Doing This by generating Innovative 2D yacht layouts
- Layouts are based on design brief input & corresponding area

Modular systems from requirements

- From standard yacht requirements, following Modular systems are formed:
  - Power System
  - Propulsion System
  - Mooring & Anchoring System
  - Extended Operations System
  - Safety & Compliance System
  - Wheelhouse System
  - Crew Operations System
  - Crew Accomodation System
  - Tender Garage System
  - Beachclub System
  - Dining & Galley System
  - Owner Accomodation System
  - Guests Accomodation System

Area prediction

- Predicting the modular system area’s using a python regression model
- By predicting the areas from parameters, the required space for the modules can be determined by using design brief and the known Feadship database.
- Input based on Design Brief
  - Lenth, Width, Draught, Nr. of Crew, Nr. of Guests, Design Speed, Range
- Prediction of the required area’s of all systems
  - Based on input parameters and Feadship Database

Valid Area Prediction, or Error??

- Some area’s show big a big error in evaluation metrics.
- Possibly these area’s are not depending just on the input parameters.
- These Area’s are more depending on “Owner Specific Requirements”
- Are these “Owner Specific Area’s” also seen as owner specific by the designer?
  - Propulsion, Safety & Compliance, Beachclub, Dining & Galley, Owner Accomodation

Question:

What are owner specific systems in a yacht design following the Desginer, and does this correspond with the predictec owner specific area’s?

Predicted Area’s DN3408

Input parameters DN3408
from Design Brief:

Length	88	[m]
Width	14,5	[m]
Draught	4	[m]
Nr. of Crew	17	[-]
Nr. of Guests	14	[-]
Design Speed	16	[kts]
Range	5000	[NM]

System	Area[m^2]	Valid prediction[Y/N]
Power	285.35	Y
Propulsion	44.69	N
Moorin & Anchoring	61.98	Y
HVAC	113.48	Y
Extended Operations	67.52	Y
Safety & Compliance	24.41	N
Wheelhouse	73.78	Y
Crew Operations	191.34	Y
Crew Accomodation	113.94	Y
Tender Garage	127.49	Y
Beachclub	86.21	N
Dining & Galley	61.15	N
Owner Accomodation	178.18	N
Guest Accomodation	183.21	Y

Arrangement Generation tool

- Generating Arrangements based on a Packing Algorithm
  - Doing this by using the predicted Modular system area’s
- Packing based on input, arrangement conditions & variables
- Output is 2D Side View of yacht arrangement
- This 2D yacht arrangement visualisation can be analysed

Input Arrangement Generator

- Length, width, deck height
- Predicted System Area's
- System separable or not
- Margin
  - If owner specific area
  - % of bigger or smaller then mean
- Preferences
  - Preferences of every system in terms of placement
    - Height preff, length preff, close to & far from other system preference

```
# Define the configuration dictionary
config = {
    'ship': {
        'length': 88,
        'width': 14.5,
        'deck_height': 3,
        'decks': ['Lower Deck', 'Main Deck', 'Bridge Deck', 'Owner Deck']
    },
    'systems': {
        'areas': [260.11441804,  31.58688445,  51.94374071,  82.46643338,  66.50703283,
          17.44829461,  82.75784908, 239.68814354, 242.86259685, 127.18351904,
         147.55150076,  47.90980946, 204.32898304, 248.2602    ],
        'names': [
            "Power System", "Propulsion System", "Mooring & Anchoring System", "HVAC System", "Extended Operations System",
            "Safety and Compliance System", "Wheelhouse System", "Crew Operations System", "Crew Accommodation System",
            "Tender Garage System", "Beachclub System", "Dining & Galley System", "Owner Accommodation System", "Guest Acco
        ],
        'separable': [False, False, False, True, True, True, False, False, False, False, False, False, False, False],
        'num_blocks': [1, 1, 1, 6, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1],
        'preferences': {},  # This will be populated after reading the Excel file
        'margins': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  # Individual margins for each system
    }
}

# Load preferences from Excel file
file_path = r"C:\Users\jaap.marcus\OneDrive - De Voogt Naval Architects\Graduation Jaap Marcus\Onderzoek\Packing Tool\Syster
preferences = read_preferences_from_excel(file_path)
config['systems']['preferences'] = preferences  # Update the preferences in the config
```

Preferences

- Are captured in a Excell file and loaded in Packing tool

System	Height preference:	Height score:	Length preference:	Length score:	Close to System 1:	Close to Score 1:	Reasoning:	Far from System 1:	Far from Score 1:	Reasoning:
Power System	Low	9	-	0	-	0	T connection transfers via electricity cables so no need for physical connection to be close to a certain system.	Owner Accommodatio	9	Because of noise generation from power system
Propulsion System	Low	9	Bow	9	-	0	Bowthruster has to be under waterline at the bow of the ship.	-	0	
Mooring & Anchoring System	Low	6	Bow	9	-	0	Mooring has to be on the main deck in the bow of the ship.	-	0	
HVAC System	-	0	-	0	Owner Accommodatio	9	Research done in General arrangements and due to interview with Sr. Specialist B. Jongepier.	-	0	
Extended Operations System	-	0	-	0	Dining & Galley Syste	0	To get resources to the galley	-	0	
Safety and Compliance System	High	3	-	0	-	0	No preferences but safety and compliance system has to be re-evaluated because it contains the high-fog system and the emergency generator. Hi-fog system is mostly close to engine system but no regulations and emergency generator has to be above the waterline but this has to be checked	-	0	
Wheelhouse System	High	9	Bow	9	-	0	Bridge has to be at least at the owner deck and preferable as high as possible. This is because the command center is left outside the scope for this research and therefore the captain needs a plain view on the sea and it's surrounding.	-	0	
Crew Operations System	Low	6	-	0	-	0	No preferences for the crew operation system, but because no preferences it is expected to find the crew operations on the lower deck.	-	0	
Crew Accommodation System	Low	6	Bow	9	-	0	No preferences and therefore expected to find the crew operations on the lower deck.	Power System	6	Noise regulations, check with sr. Specialist
Tender Garage & Toy Storage Sys	-	0	-	0	-	0	Tender garage needs to be above the waterline and there are no further preferences.	-	0	
Beachclub system	Low	9	Stern	9	-	0	Beachclub has to be on the lower deck located as close to the stern as possible. Otherwise it would be to far from the water or to close to the bow and thereby causing difficulties with the ship bow	-	0	
Dining & Galley System	High	3	-	0	-	0	No regulatory or preferences standard requirements for the dining&galley system	-	0	
Owner Accommodation System	High	9	Bow	6	-	0	Owner accommodation positioned on owner or bridge deck as high as possible and preferably facing the bow of the ship. Custom requirement of the client could be facing the stern of the ship.	Power System	9	Because of noise generation from the power system

Margin

- As stated in the Area prediction tool there are systems with owner specific area's
- This means area's do not depend on input parameters but on preferences of the specific owner.
- Therefore these area's have the possibility to add a margin to
  - With this margin the mean area can be increased of decreased with a certain percentage

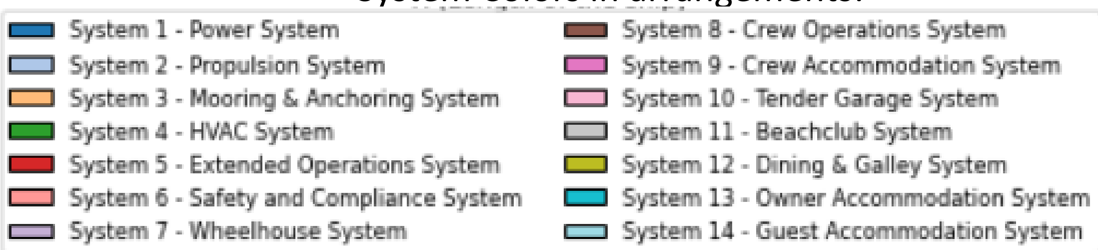
Visualisation

- The arrangement tool generates 5 arrangements based on input and preferences
- The arrangement tool also generates 5 arrangements based on input but generates a 10% shorter yacht
  - This is to see if the requirements could possibly also fit in a shorter yacht.
- All arrangements are generated and shown
  - Arrangements also come with arrangement scores
    - These scores show how much the arrangement applyes to its input and preferences
    - Scores for total arrangement and seperated systems are shown
      - This way it can be checked which system is the reason for a low or a high score.

Visualisation DN3408

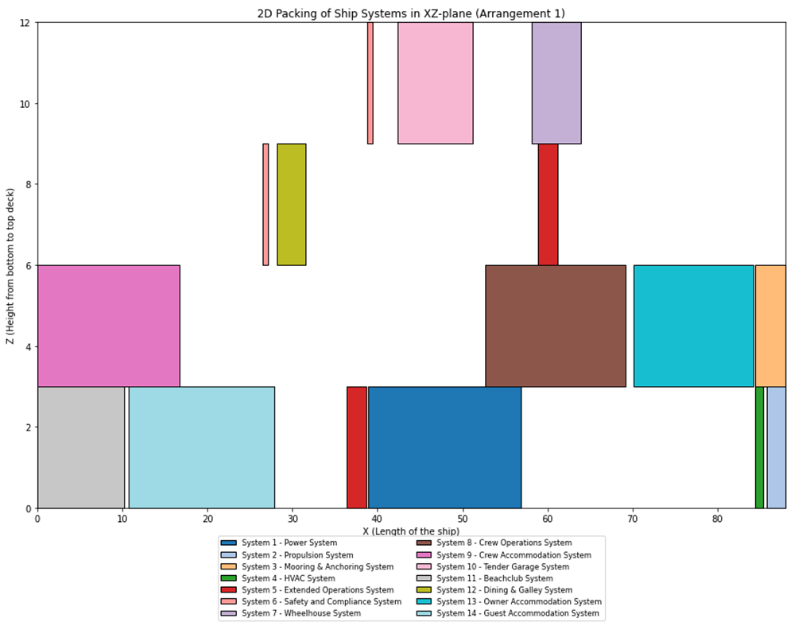
- Example visualisation of arrangement for input from DN3408
- These arrangements can be compared with final GA from designer
- In the meeting Thursday 19-09 new arrangements can be generated live to discuss.
- In the meeting Thursday 19-09 DN3442 arrangements are generated as well.

System Colors in arrangements:

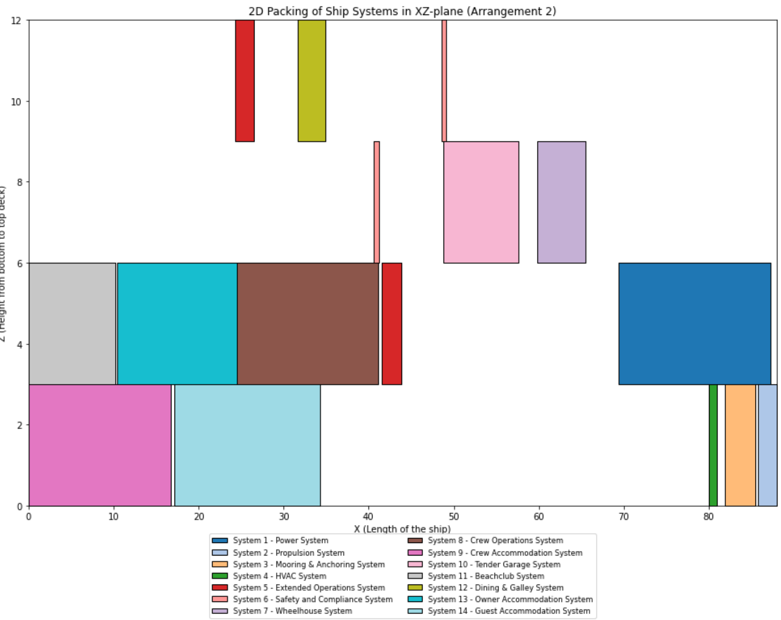


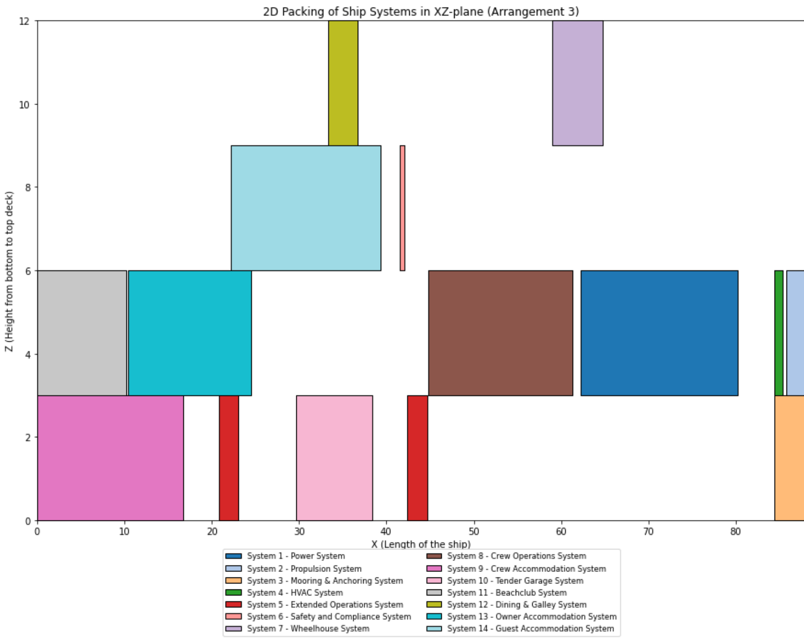
Visualisation DN3408

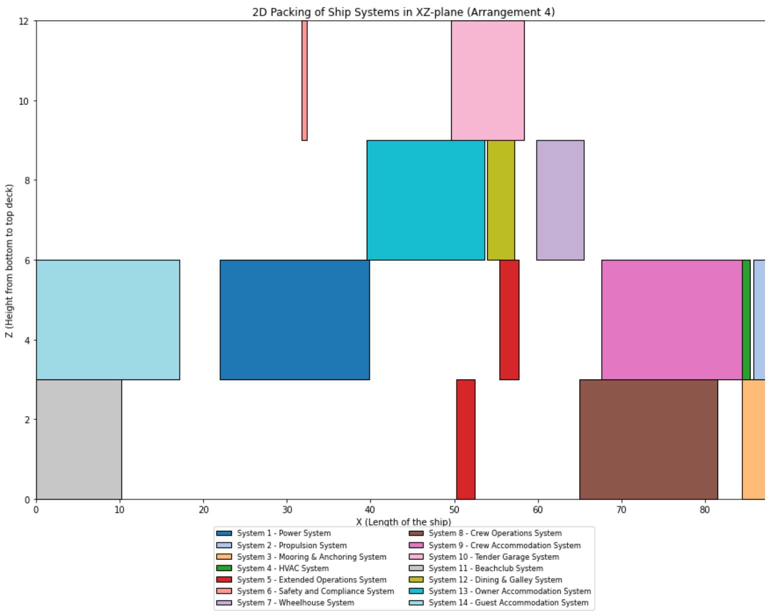
- 2D side view xz-plane
- Visualisation shows 4 decks
  - Lower, Main, Bridge, Owner
- X-axis shows Length all over
- Y-Axis shows height of decks
- 2 upper decks are only available from 1/4L – 3/4L
- Close your eyes and visualize a yacht from the building blocks…



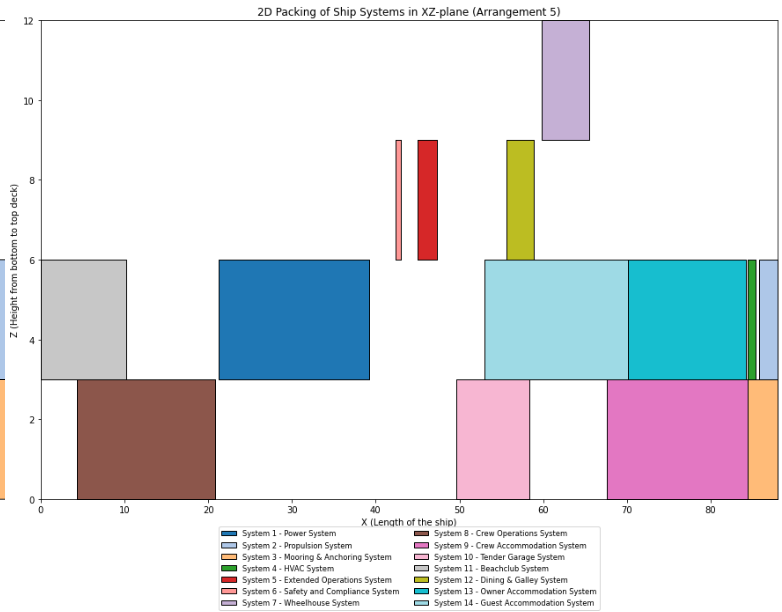
Visualisation DN3408







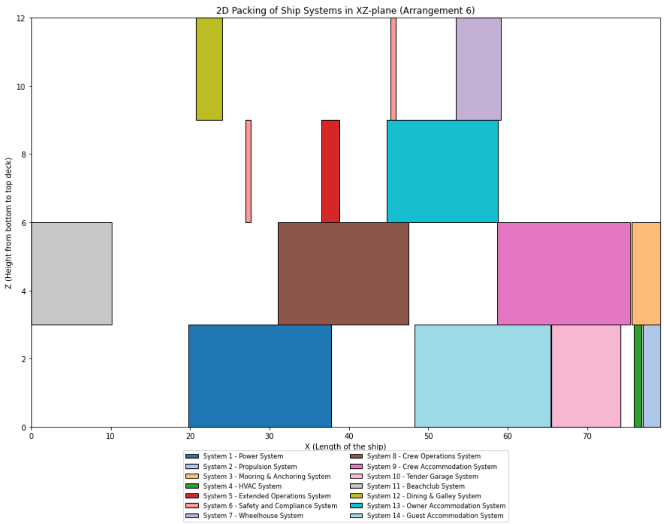
2D Packing of Ship Systems in XZ-plane (Arrangement 4)
Z (Height from bottom to top deck)
X (Length of the ship)
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System

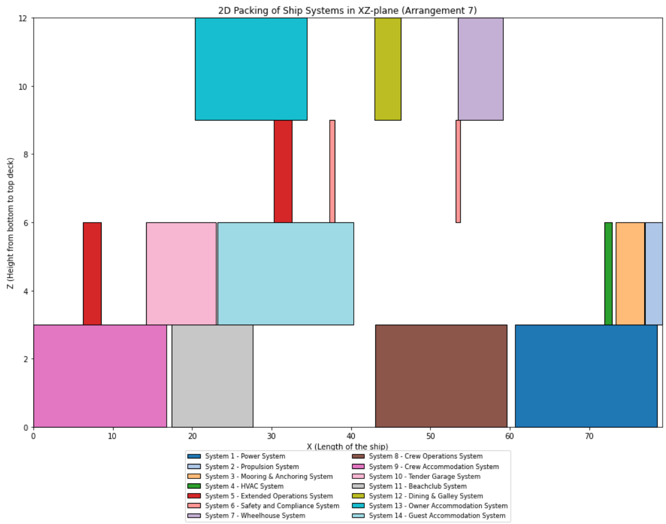


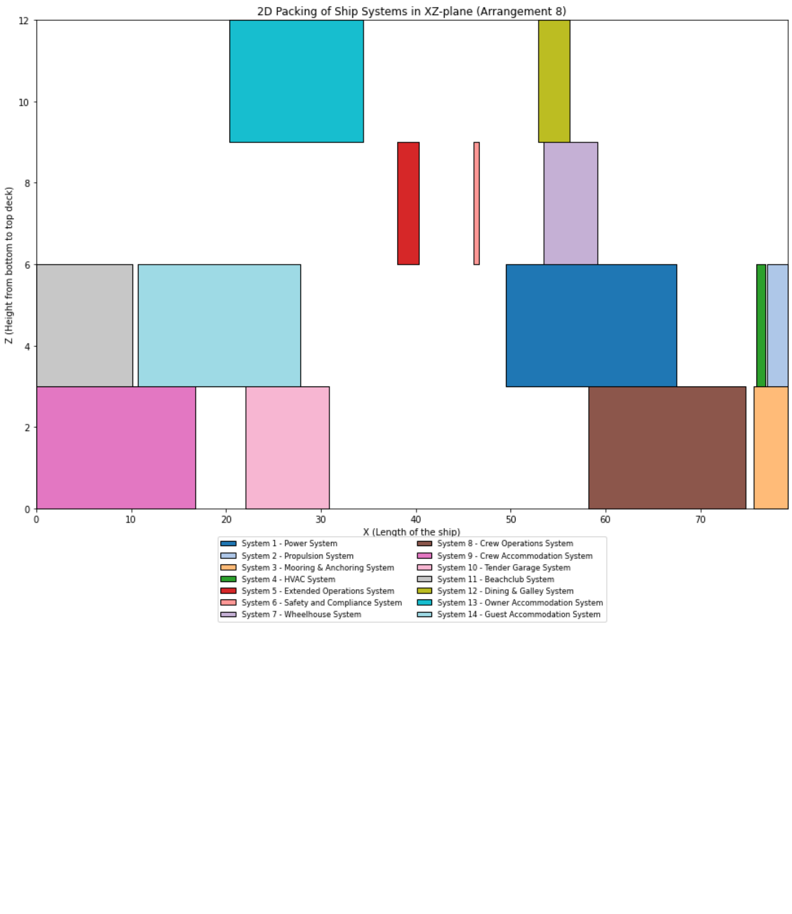
2D Packing of Ship Systems in XZ-plane (Arrangement 5)
Z (Height from bottom to top deck)
X (Length of the ship)
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System

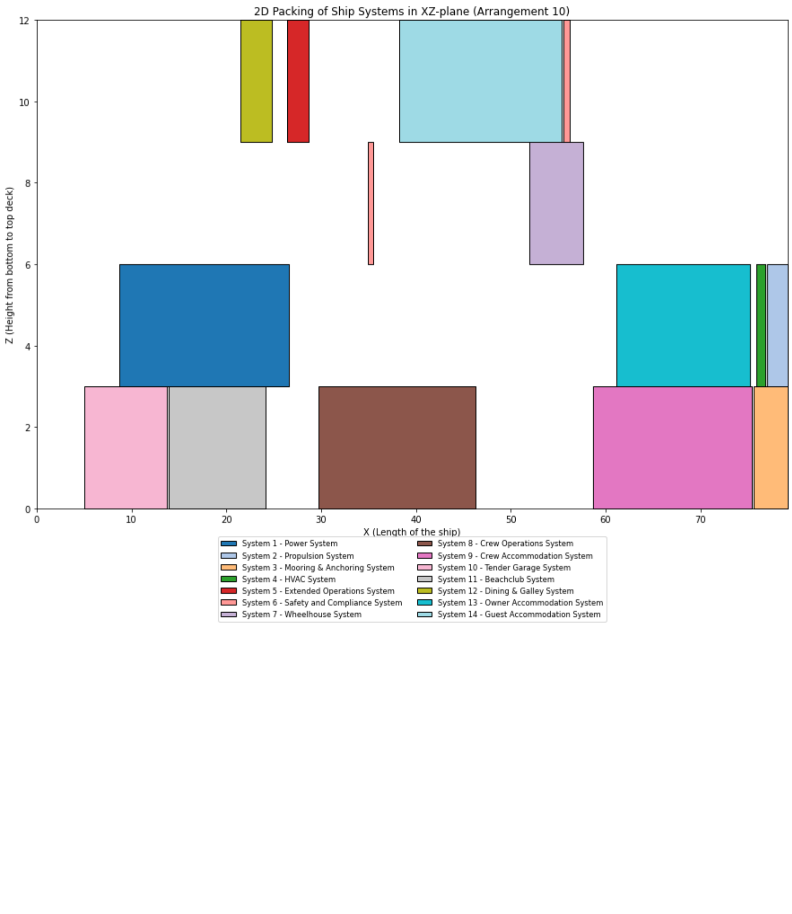
Visualisation DN3408 Optimized length

- These are the visualisations with input of DN3408 but with 10% length optimization.






2D Packing of Ship Systems in XZ-plane (Arrangement 8)
Z (Height from bottom to top deck)
X (Length of the ship)
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System


2D Packing of Ship Systems in XZ-plane (Arrangement 10)
Z (Height from bottom to top deck)
X (Length of the ship)
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System

Questions:

- What do you think about the working of the Arrangement Generator?
- Do you think the Arrangement Generator can be a tool to support the creative part of the design phase?
- Do you think the Arrangement Generator can be a tool to make quicker iterations in early phase design space?
- What are the limitations of the Arrangement Generator?
- What future adjustments can me made to improve the Arrangement Generator?
- In what areas performs the designer better than the arrangement generator?
- In what areas performs the arrangement generator better than the designer?

Questions for me?

- You can always reach me on jaap.marcus@devoogt.feadship.nl
- Thank you very much in advance and see you on Thursday!

- Thursday September 19
- 09:00 @ Pi

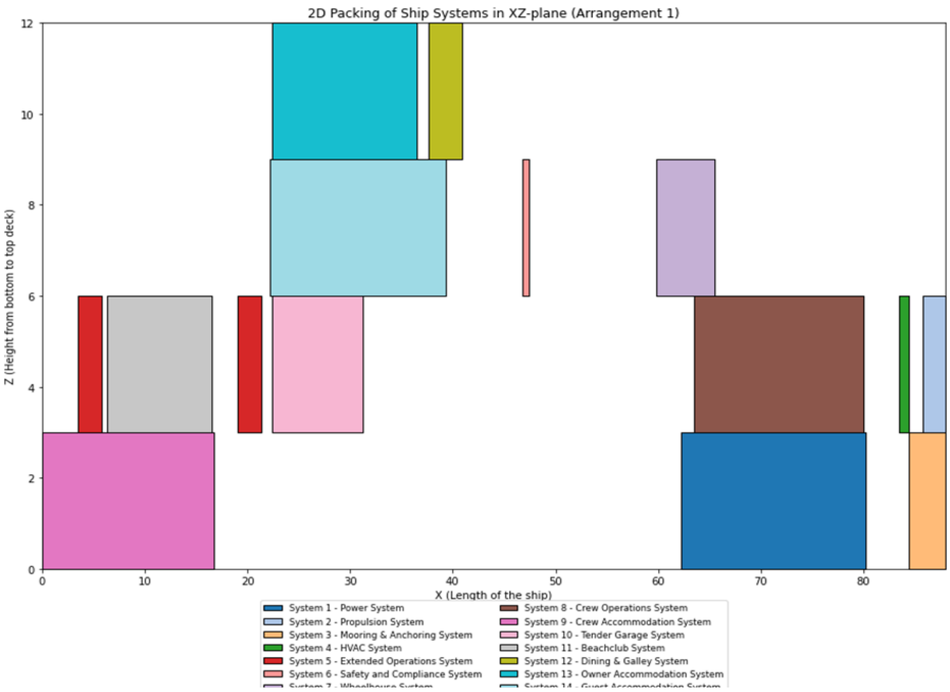
K

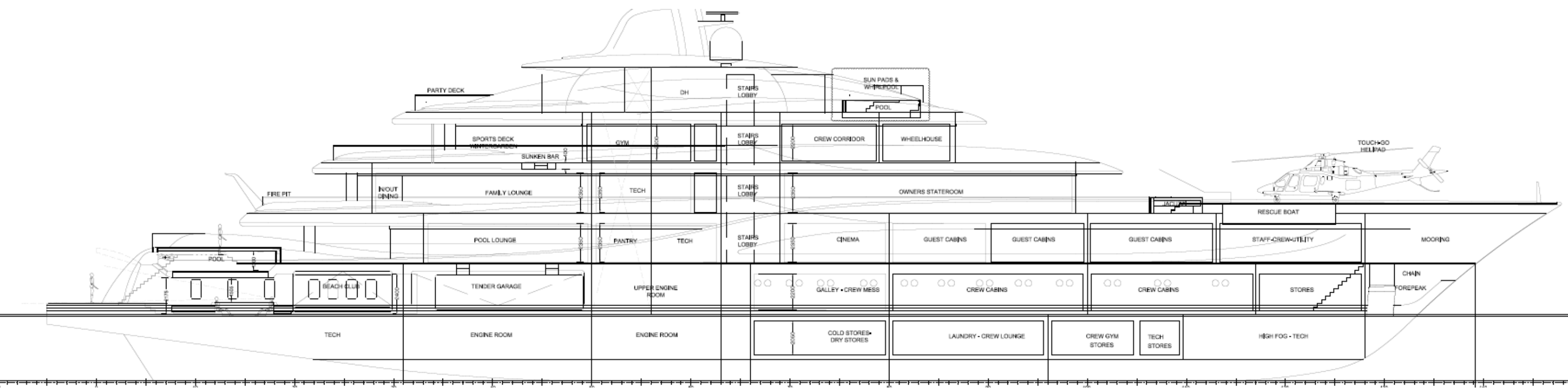
Modular Arrangement Tool Results

Arrangement Tool Results

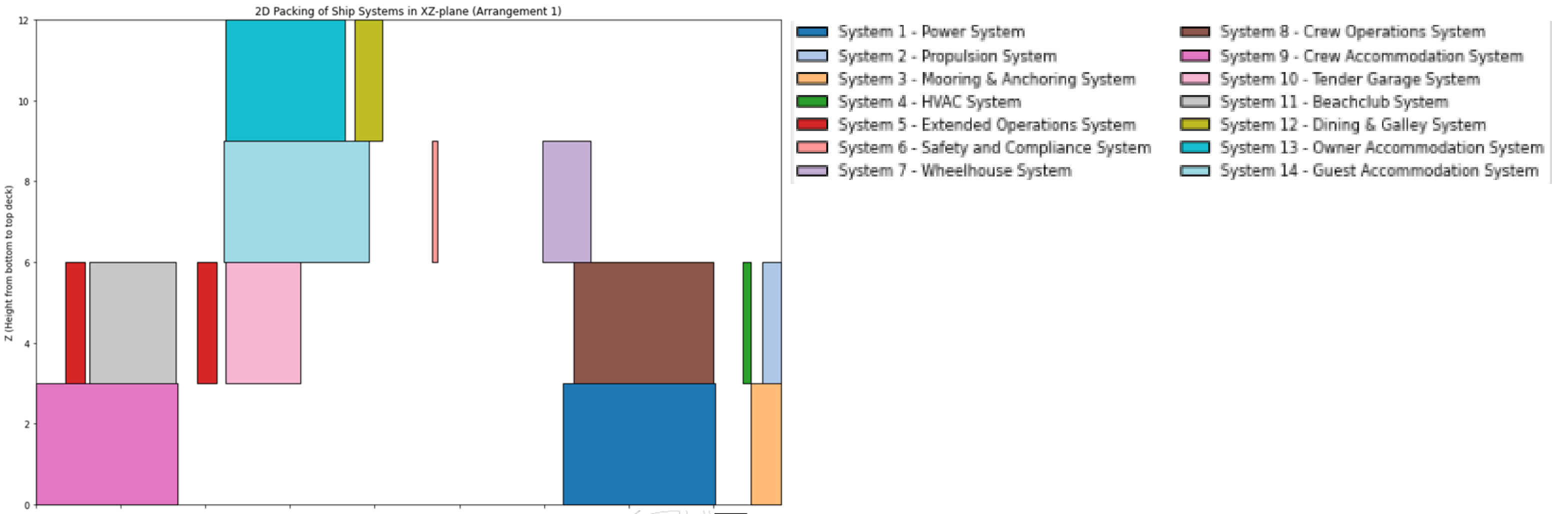
DN3408 - Bulldog

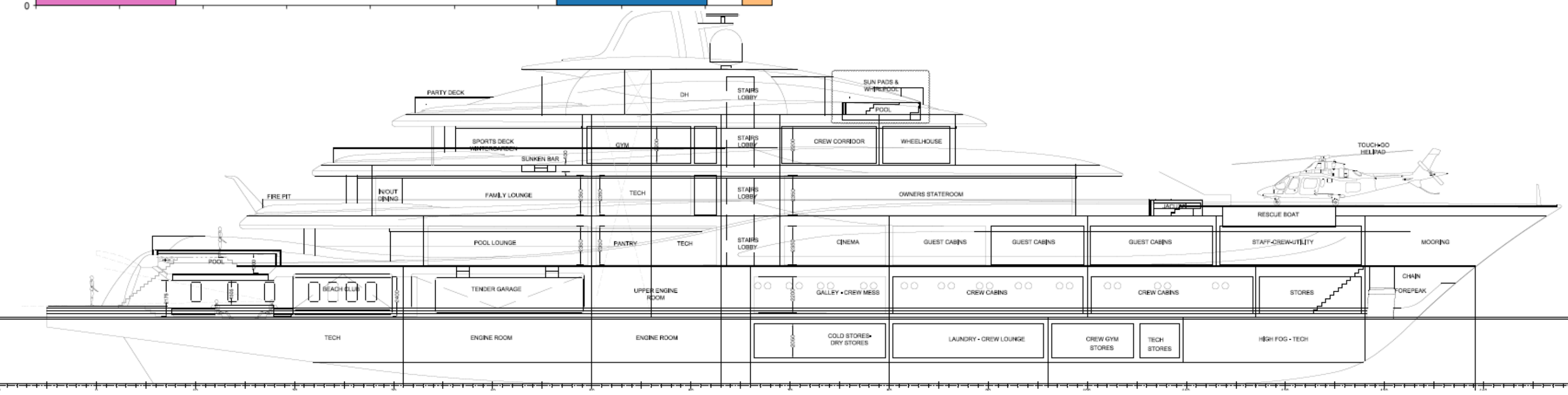
- Highest Score:





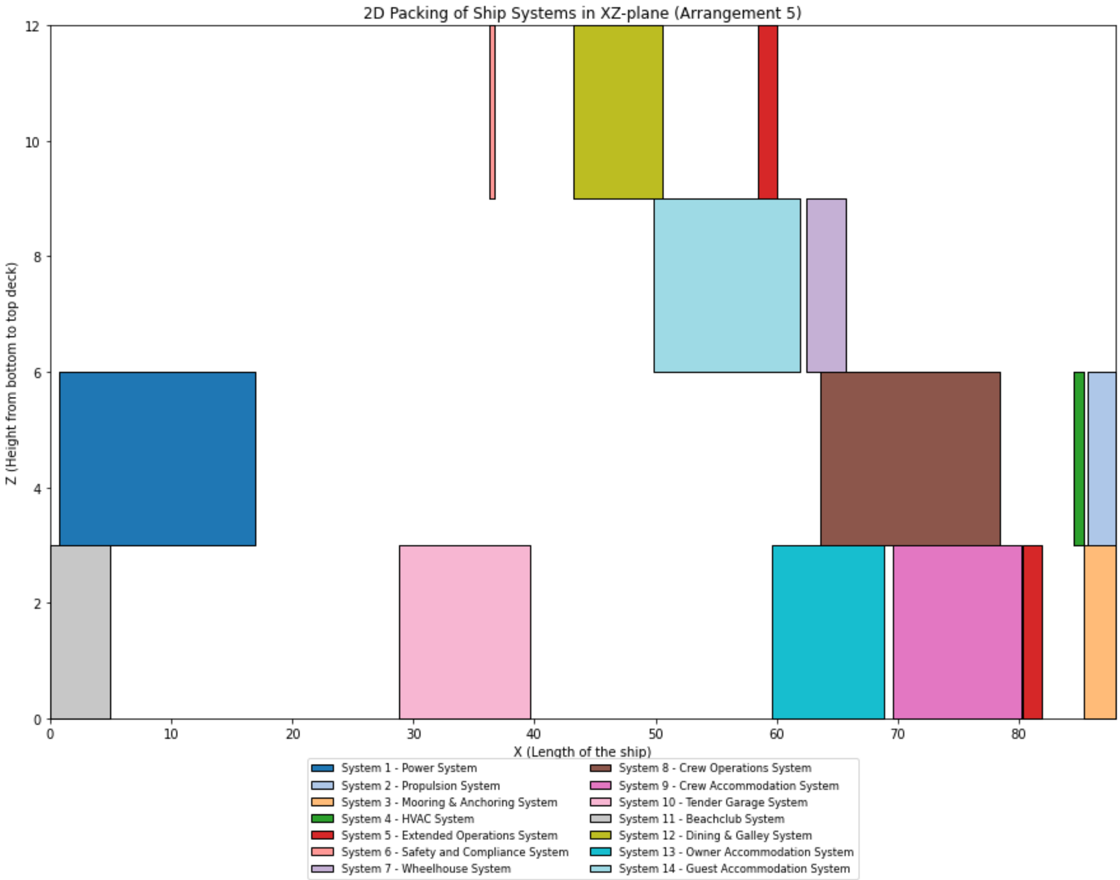
PARTY DECK
DH
STAIRS LOBBY
SUN PADS & WHIRLPOOL
POOL
SPORTS DECK
GYM
CREW CORRIDOR
WHEELHOUSE
TOUCH+GO HELIPAD
SUNKEN BAR
FIRE PIT
IN/OUT DINING
FAMILY LOUNGE
TECH
OWNERS STATEROOM
RESCUE BOAT
POOL LOUNGE
PANTRY
CINEMA
GUEST CABINS
STAFF+CREW+UTILITY
MOORING
BEACH CLUB
TENDER GARAGE
UPPER ENGINE ROOM
GALLEY + CREW MESS
CREW CABINS
STORES
CHAIN
FOREPEAK
ENGINE ROOM
COLD STORES+ DRY STORES
LAUNDRY + CREW LOUNGE
CREW GYM STORES
TECH STORES
HIGH FOG + TECH


2D Packing of Ship Systems in XZ-plane (Arrangement 1)
Z (Height from bottom to top deck)
0
2
4
6
8
10
12
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System

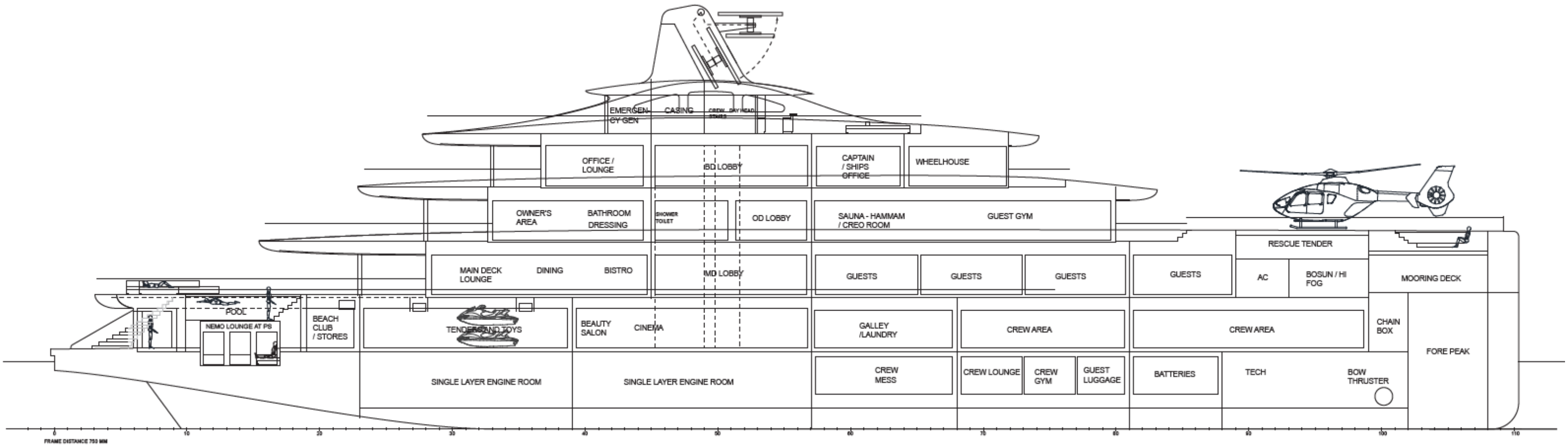

PARTY DECK
SUN PADS &
POOL
SPORTS DECK
SUNKEN BAR
GYM
CREW CORRIDOR
WHEELHOUSE
TOUCH+GO
HELIPAD
FIRE PIT
FAMILY LOUNGE
TECH
OWNERS STATEROOM
RESCUE BOAT
POOL LOUNGE
PANTRY
CINEMA
GUEST CABINS
STAFF+CREW+UTILITY
MOORING
POOL
BEACH CLUB
TENDER GARAGE
UPPER ENGINE
ROOM
GALLEY + CREW MESS
CREW CABINS
STORES
CHAIN
FOREPEAK
TECH
ENGINE ROOM
COLD STORES+
DRY STORES
LAUNDRY + CREW LOUNGE
CREW GYM
TECH
STORES
HIGH FOG + TECH

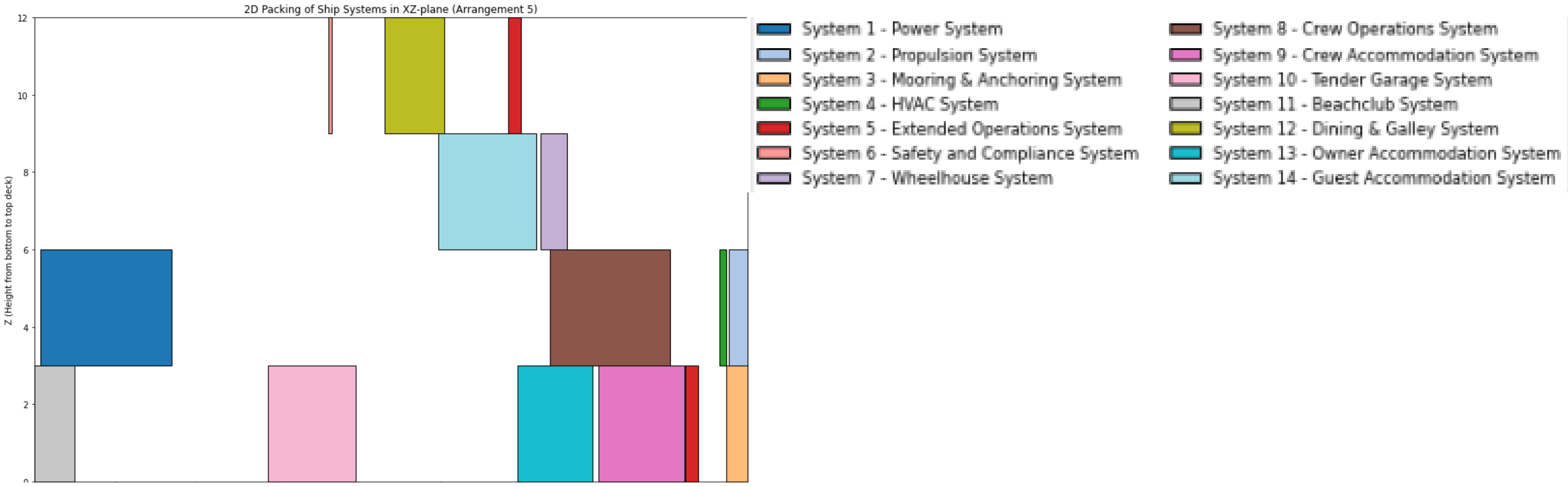
DN3524

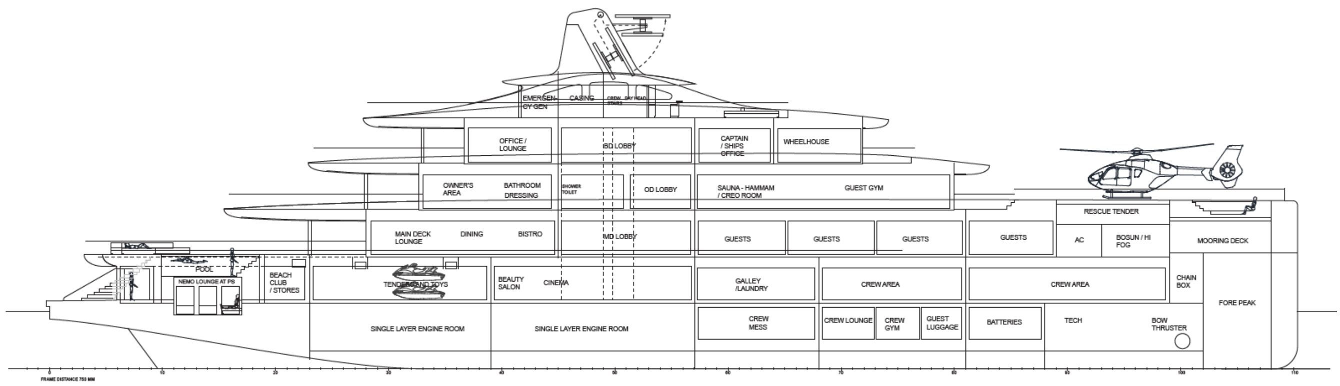
- Highest score:



DN3524

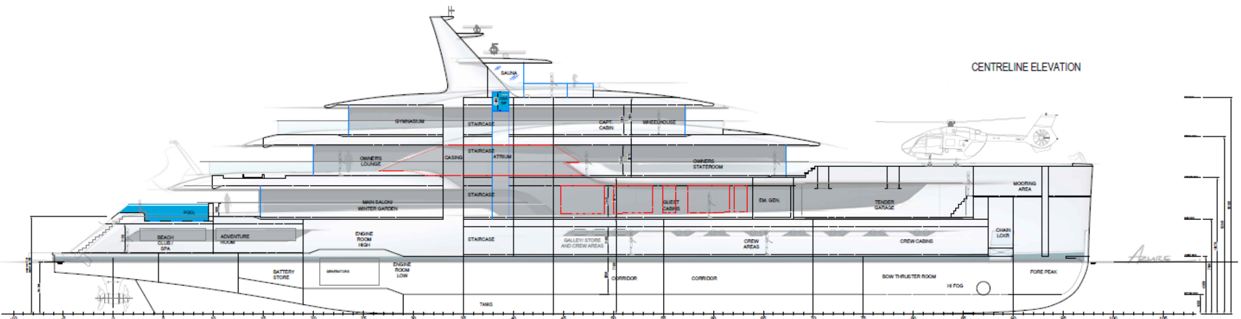

EMERGEN-CY GEN
CASING
OFFICE / LOUNGE
BD LOBBY
CAPTAIN / SHIPS OFFICE
WHEELHOUSE
OWNER'S AREA
BATHROOM DRESSING
SHOWER TOILET
OD LOBBY
SAUNA - HAMMAM / CREO ROOM
GUEST GYM
RESCUE TENDER
MAIN DECK LOUNGE
DINING
BISTRO
MD LOBBY
GUESTS
GUESTS
GUESTS
GUESTS
AC
BOSUN / HI FOG
MOORING DECK
POOL
NEMO LOUNGE AT PS
BEACH CLUB / STORES
TENDERS AND TOYS
BEAUTY SALON
CINEMA
GALLEY /LAUNDRY
CREW AREA
CREW AREA
CHAIN BOX
FORE PEAK
SINGLE LAYER ENGINE ROOM
SINGLE LAYER ENGINE ROOM
CREW MESS
CREW LOUNGE
CREW GYM
GUEST LUGGAGE
BATTERIES
TECH
BOW THRUSTER

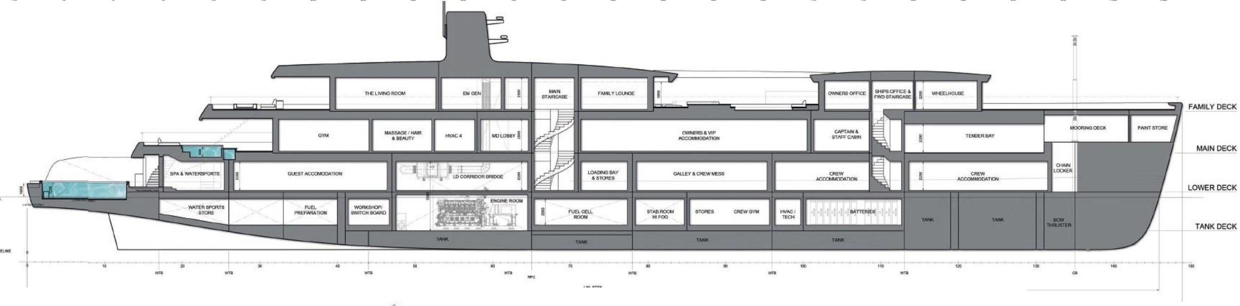

2D Packing of Ship Systems in XZ-plane (Arrangement 5)
Z (Height from bottom to top deck)
0
2
4
6
8
10
12
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System

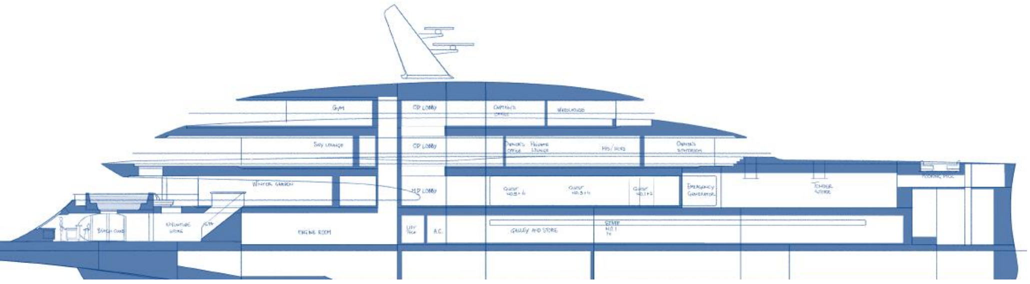

CASING
OFFICE / LOUNGE
OD LOBBY
CAPTAIN / SHIPS OFFICE
WHEELHOUSE
OWNER'S AREA
BATHROOM
DRESSING
OD LOBBY
SAUNA - HAMMAM / CREO ROOM
GUEST GYM
RESCUE TENDER
MAIN DECK LOUNGE
DINING
BISTRO
MD LOBBY
GUESTS
GUESTS
GUESTS
GUESTS
AC
BOSUN / HI FOG
MOORING DECK
POOL
NEMO LOUNGE AT PS
BEACH CLUB / STORES
BEAUTY SALON
CINEMA
GALLEY /LAUNDRY
CREW AREA
CREW AREA
CHAIN BOX
FORE PEAK
SINGLE LAYER ENGINE ROOM
SINGLE LAYER ENGINE ROOM
CREW MESS
CREW LOUNGE
CREW GYM
GUEST LUGGAGE
BATTERIES
TECH
BOW THRUSTER

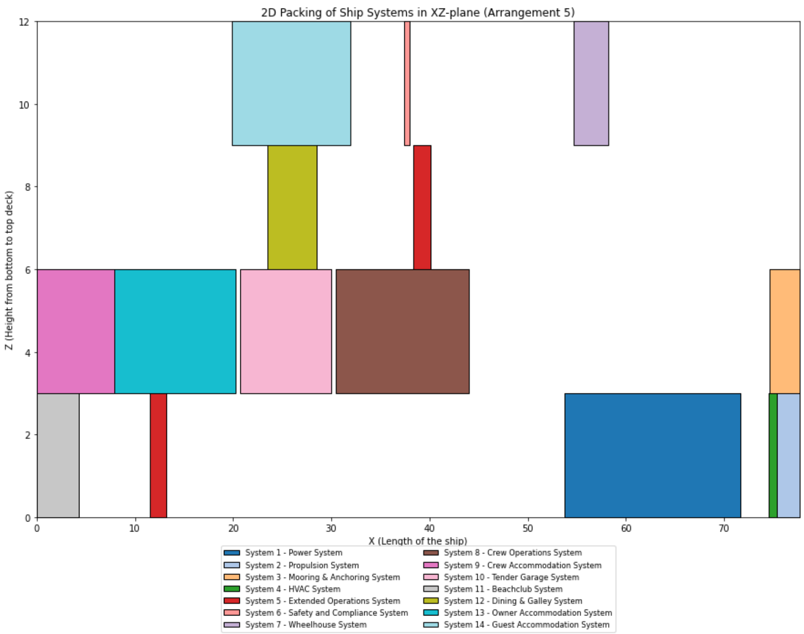
DN3554 - Komodo

- Three Designs and a Generated Design

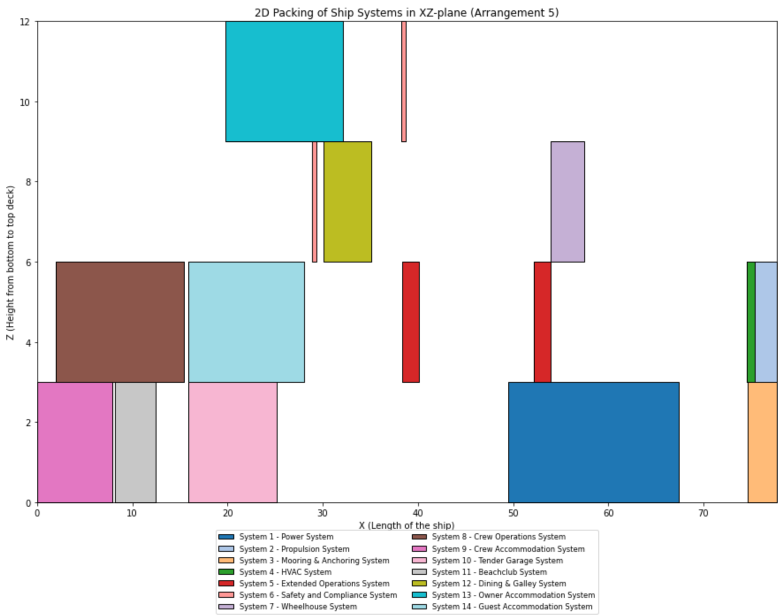

CENTRELINE ELEVATION
SAUNA
GYMNASIUM
STAIRCASE
CAPT. CABIN
WHEELHOUSE
OWNERS LOUNGE
CASING
ATRIUM
OWNERS STATEROOM
POOL
MAIN SALON / WINTER GARDEN
GUEST CABINS
EM. GEN
TENDER GARAGE
MOORING AREA
BEACH CLUB SPA
ADVENTURE ROOM
ENGINE ROOM HIGH
GALLEY STORE AND CREW AREAS
CREW AREAS
CREW CABINS
CHAIN LCKR
BATTERY STORE
ENGINE ROOM LOW
CORRIDOR
BOW THRUSTER ROOM
FORE PEAK
HI FOG
TANK


THE LIVING ROOM
EM GEN
MAIN STAIRCASE
FAMILY LOUNGE
OWNERS OFFICE
SHIPS OFFICE & FWD STAIRCASE
WHEELHOUSE
FAMILY DECK
GYM
MASSAGE / HAIR & BEAUTY
HVAC 4
MD LOBBY
OWNERS & VIP ACCOMMODATION
CAPTAIN & STAFF CABIN
TENDER BAY
MOORING DECK
PAINT STORE
MAIN DECK
SPA & WATERSPORTS
GUEST ACCOMODATION
LD CORRIDOR BRIDGE
LOADING BAY & STORES
GALLEY & CREW MESS
CREW ACCOMMODATION
CHAIN LOCKER
LOWER DECK
WATER SPORTS STORE
FUEL PREPARATION
WORKSHOP/ SWITCH BOARD
ENGINE ROOM
FUEL CELL ROOM
STAB ROOM HI FOG
STORES
CREW GYM
HVAC / TECH
BATTERIES
TANK
BOW THRUSTER
TANK DECK





2D Packing of Ship Systems in XZ-plane (Arrangement 5)
Z (Height from bottom to top deck)
X (Length of the ship)
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System



2D Packing of Ship Systems in XZ-plane (Arrangement 5)
Z (Height from bottom to top deck)
X (Length of the ship)
System 1 - Power System
System 2 - Propulsion System
System 3 - Mooring & Anchoring System
System 4 - HVAC System
System 5 - Extended Operations System
System 6 - Safety and Compliance System
System 7 - Wheelhouse System
System 8 - Crew Operations System
System 9 - Crew Accommodation System
System 10 - Tender Garage System
System 11 - Beachclub System
System 12 - Dining & Galley System
System 13 - Owner Accommodation System
System 14 - Guest Accommodation System