Hull Form Design Optimization

Using Computational Fluid Dynamics Data-Driven Surrogate Models

Walker, J.M.

DOI
10.4233/uuid:2fee1b23-109d-40dc-9653-8f246800a539

Publication date
2025

Document Version
Final published version

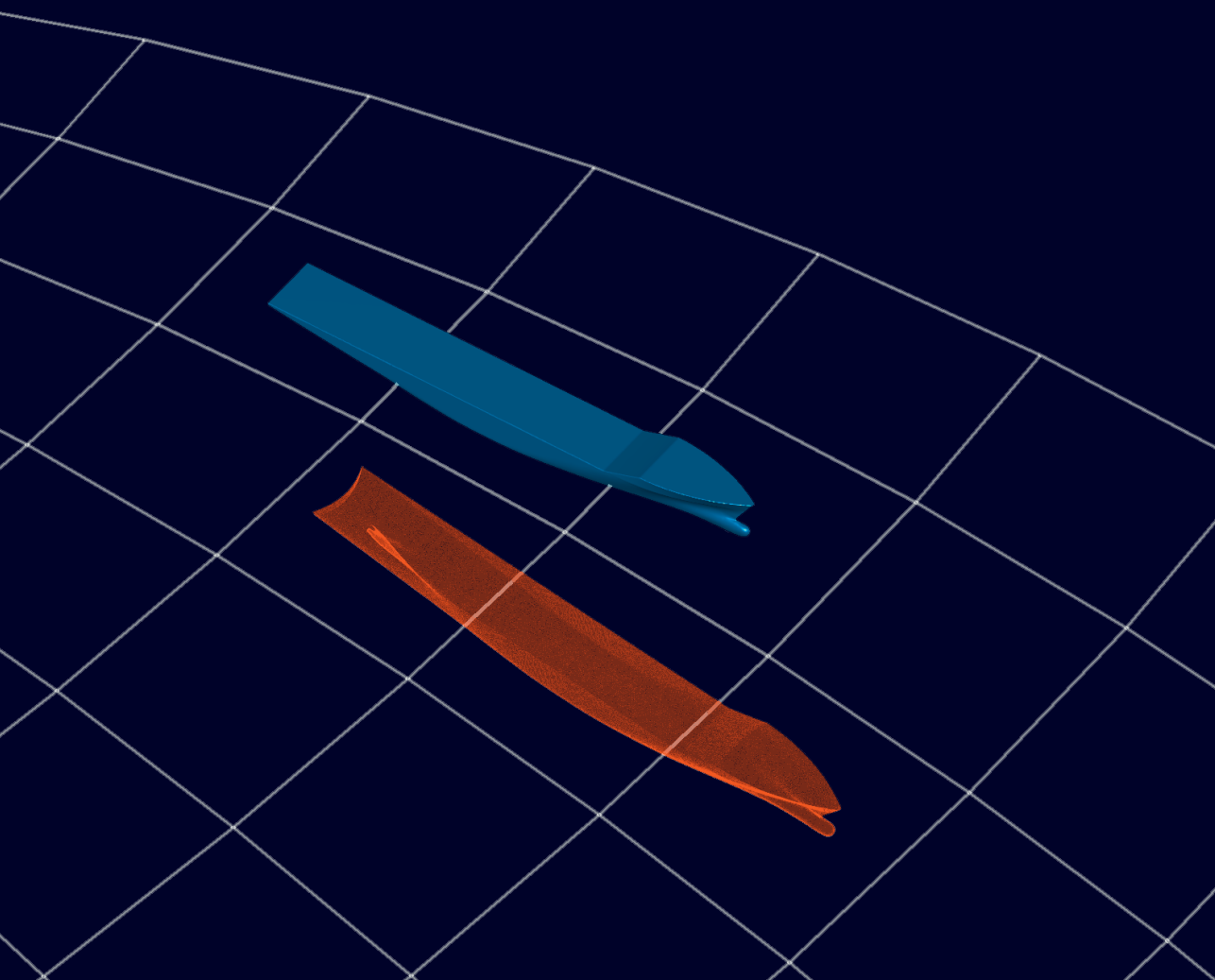
Citation (APA)
Walker, J. M. (2025). *Hull Form Design Optimization: Using Computational Fluid Dynamics Data-Driven Surrogate Models*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:2fee1b23-109d-40dc-9653-8f246800a539

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Hull Form Design Optimization

Using Computational Fluid Dynamics Data-Driven Surrogate Models

Jake M. Walker



Hull Form Design Optimization

Using Computational Fluid Dynamics Data-Driven Surrogate Models

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op 1 December 2025 om 17:30 uur

door

Jake Matthew WALKER

Master of Engineering in Mechanical Engineering,
University of Strathclyde, Schotland,
geboren te Edinburgh, Schotland.

Dit proefschrift is goedgekeurd door de

promotoren:	Dr. A. Coraddu
	Prof. dr. ir. B.J. Boersma
	Prof. dr. L. Oneto

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter	
Dr. A. Coraddu,	Technische Universiteit Delft	
Prof. dr. ir. B.J. Boersma,	Technische Universiteit Delft	
Prof. dr. L. Oneto,	Università degli Studi di Genova,	Italië
Onafhankelijke leden:		
Prof. dr. G. Weymouth,	Technische Universiteit Delft,	Nederland
Prof. dr. A. Sobey,	University of Southampton,	Engeland
Dr. D. Villa,	Università degli Studi di Genova,	Italië
Dr. A. Knobbe,	Universiteit Leiden,	Nederland
Prof. dr. R. Pecnik,	Technische Universiteit Delft,	Reserve member

Cover Design:	Jake M. Walker
Typesetting:	LaTeX
Printing:	Ridderpint
ISBN:	978-94-6522-960-7

A catalogue record is available from the Delft University of Technology Library.

Finally, from so little sleeping and so much reading,
his brain dried up and he went completely out of his mind.

— Miguel de Cervantes Saavedra, Don Quixote

Summary

This thesis addresses the growing demand for sustainable ship design by focusing on the optimization of hull forms. The hull form geometry directly influences the ship's hydrodynamic resistance, and consequently its fuel consumption and emissions. As the maritime industry faces increasingly stringent environmental regulations, there is a pressing need for design methodologies that can deliver high-performance hull forms across diverse operational conditions, while remaining computationally efficient and producing physically meaningful geometries.

The first part of the thesis critically reviews the vibrant and expanding literature on hull form optimization, structured around the four-step framework. This framework typically involves: (1) defining a geometric parameterization and performance indicators, (2) evaluating hydrodynamic performance via simulation, (3) constructing surrogate models to approximate those results, and (4) verifying that the optimized shapes are physically plausible. This review highlights the trade-offs involved in parameterization choice, the challenges of building accurate surrogate models in high-dimensional spaces, and the frequent absence of methods to enforce physical plausibility during optimization. It emphasizes that despite significant progress, current workflows remain data-hungry, computationally expensive, and poorly suited for iterative or comparative design investigations.

The second part of the thesis addresses the issue of physical realism by introducing a novel integration of the IMO Intact Stability Code directly into the optimization loop. This allows the optimization process to systematically exclude non-compliant designs during the search, rather than checking constraints only after optimization. As demonstrated using the KCS hull as a test case, this constraint-driven approach not only yields regulation-compliant designs but also reduces the design space and cuts overall computational effort.

The third part of the thesis addresses one of the biggest challenges with the four-step approach: the lack of ability to reuse existing data when changes are made to the parent hull geometry or to the shape parameterization. As a result, each new optimization study often requires generating a new simulation dataset from scratch, which leads to high computational cost and human effort. Therefore, in this chapter, we present a key methodological change: decoupling the geometric parameterization from the surrogate modeling phase. This allows existing simulation data to be reused across different design investigations, even when the underlying shape parameterization changes. This decoupled framework is demonstrated using the Delft Systematic Yacht Hull Series and is shown to construct accurate surrogate models without requiring new simulations for each new parameterization. The framework is particularly advantageous in extrapolation scenarios, where optimization is carried out in regions of the design space not explicitly covered by the training data. By enabling data reuse, the method not only accelerates the design process but also provides a natural starting point for new optimization campaigns, significantly lowering the barrier to entry for early-stage hull-form investigations. This part of the thesis highlights extrapolation as a central AI challenge and opportunity in the context

of hull form optimization. We show in this chapter how to develop predictive models to generalize beyond their training data without compromising physical plausibility.

The final part of the thesis extends this decoupled framework to more topologically complex hull forms, with a focus on container ships. We developed strategies to build surrogate models capable of predicting resistance for a wide range of designs, both within and outside the training distribution. In particular, we benchmarked models trained exclusively on in-class data against those built from reused, out-of-class datasets. Results showed that surrogate models trained on reused historical data could generalize well and perform competitively in optimization tasks. We also proposed a hybrid strategy that selectively combines reused data with a limited amount of newly generated simulations. This approach offers a flexible trade-off between computational cost and model fidelity. Results show that the hybrid approach led to approximately 30% reduction in computational effort compared to the standard four-step approach.

Taken together, the contributions of this thesis provide a framework for hull form optimization that is data-efficient and generalizable. By enabling the reuse of simulation data across different designs and parameterizations, the proposed methods establish a more sustainable and scalable foundation for future ship design workflows. The results support design exploration that is fast, interpretable, and regulation-compliant, helping to align the maritime industry's operational and environmental goals.

Samenvatting

Dit proefschrift speelt in op de groeiende behoefte aan duurzaam scheepsontwerp door zich te richten op de optimalisatie van rompvormen. De geometrie van de rompvorm beïnvloedt rechtstreeks de hydrodynamische weerstand van het schip en daarmee het brandstofverbruik en de uitstoot. Aangezien de maritieme sector wordt geconfronteerd met steeds strengere milieuregels, is er een dringende vraag naar ontwerpmethoden die onder uiteenlopende operationele omstandigheden hoogwaardige rompvormen kunnen leveren terwijl de methoden computationeel efficiënt blijven en fysiek realistische geometrieën genereren.

Het eerste deel van dit proefschrift geeft een kritische beschouwing van de levendige en zich uitbreidende literatuur over rompvormoptimalisatie, gestructureerd rond het vierstappenmodel. Dit model omvat doorgaans: (1) het definiëren van een geometrische parametrisering en prestatie-indicators, (2) het evalueren van hydrodynamische prestaties via simulaties, (3) het opstellen van surrogate modellen ter benadering van deze resultaten, en (4) het verifiëren dat de geoptimaliseerde vormen fysiek realistisch zijn. Deze literatuurstudie benadrukt de afwegingen die gepaard gaan met de keuze van parametrisering, de uitdagingen van het bouwen van nauwkeurige surrogaatmodellen in hoog-dimensionale ruimtes, en het veelvuldige gebrek aan methoden om fysieke aannemelijkheid tijdens de optimalisatie af te dwingen. Er wordt geconcludeerd dat ondanks aanzienlijke vooruitgang, de huidige workflows veel data vereisen, computationeel kostbaar zijn, en slecht aansluiten bij iteratieve of vergelijkende ontwerpstudies.

Het tweede deel van het proefschrift richt zich op de uitdaging van het waarborgen van fysieke realisme door een nieuwe integratie van de IMO Intact Stability Code direct in de optimalisatiecyclus voor te stellen. Hierdoor kunnen ontwerpen die niet aan de eisen voldoen systematisch worden uitgesloten tijdens het optimalisatieproces, in plaats van deze pas achteraf te controleren. Zoals aangetoond met de KCS-romp als testcase, levert deze, op beperkingen gebaseerde, aanpak niet alleen regelconforme ontwerpen op, maar verkleint het ook de ontwerpruimte en vermindert het de totale rekeninspanning.

Het derde deel van het proefschrift pakt één van de grootste beperkingen van het vierstappenmodel aan: het onvermogen om bestaande data opnieuw te gebruiken wanneer wijzigingen worden aangebracht aan de oorspronkelijke rompgeometrie of de gebruikte vormparametrisering. Als gevolg daarvan vereist elke nieuwe optimalisatiestudie over het algemeen het genereren van een volledig nieuwe simulatiedataset, wat leidt tot hoge rekenkosten en veel menselijke inspanning. Daarom introduceren we in dit hoofdstuk een belangrijke methodologische verandering: het loskoppelen van de geometrische parametrisering van de fase waarin surrogaatmodellen worden opgesteld. Hierdoor kan bestaande simulatiedata worden hergebruikt in verschillende ontwerpstudies, zelfs wanneer de onderliggende vormparametrisering verandert. Dit losgekoppelde framework wordt gedemonstreerd met behulp van de Delft Systematic Yacht Hull Series en blijkt in staat om nauwkeurige surrogaatmodellen op te stellen zonder nieuwe simulaties te vereisen

voor elke nieuwe parametrisering. Het framework is met name voordelig in extrapolatie-scenario's, waarbij optimalisatie plaatsvindt in gebieden van de ontwerpruimte die niet expliciet zijn opgenomen in de trainingsdata. Door het hergebruik van data te faciliteren, versnelt deze methode niet alleen het ontwerpproces, maar biedt deze ook een natuurlijk startpunt voor nieuwe optimalisatiecampagnes, waardoor de drempel ontwerpstudies in een vroeg stadium aanzienlijk wordt verlaagd. Dit deel van het proefschrift benadrukt extrapolatie als een centrale uitdaging en kans voor AI in de context van rompvormoptimalisatie. We tonen in dit hoofdstuk aan hoe voorspellende modellen kunnen worden ontwikkeld die buiten hun trainingsdomein kunnen generaliseren zonder dat dit ten koste gaat van de fysieke plausibiliteit.

Het laatste deel van het proefschrift breidt dit losgekoppelde framework uit naar meer topologisch complexe rompvormen, met een focus op containerschepen. We ontwikkelden strategieën voor het bouwen van surrogaatmodellen die in staat zijn om de weerstand van een breed scala aan ontwerpen te voorspellen, zowel binnen als buiten de oorspronkelijke trainingsverdeling. In het bijzonder hebben we modellen vergeleken die uitsluitend zijn getraind op in-class data met modellen die gebruikmaken van hergebruikte en out-of-class datasets. De resultaten tonen aan dat modellen getraind op historische, hergebruikte data goed kunnen generaliseren en concurrerend presteren bij optimalisatietaken. We stelden ook een hybride strategie voor die hergebruikte data combineert met een beperkte hoeveelheid nieuw gegenereerde simulaties. Deze aanpak biedt een flexibel evenwicht tussen rekenkosten en modelnauwkeurigheid. Resultaten tonen aan dat deze hybride aanpak leidde tot een reductie van ongeveer 30% in rekenkundige inspanning ten opzichte van het standaard vierstappenmodel.

Gezamenlijk bieden de bijdragen van dit proefschrift een data-efficiënt en generaliseerbaar framework voor rompvormoptimalisatie. Door het mogelijk te maken simulatiedata opnieuw te gebruiken voor verschillende ontwerpen en parametrisaties, leggen de voorgestelde methoden een duurzamere en beter schaalbare basis voor toekomstige scheepsontwerp-workflows. De resultaten ondersteunen ontwerpverkenningen die snel, interpreteerbaar en aan regelgeving conformzijn, waarmee de operationele- en milieudoelstellingen binnen de maritieme sector beter op elkaar kunnen worden afgestemd.

Contents

List of Figures

List of Tables

1

Introduction

Maritime transport (shipping) is widely recognized as the most efficient and cost-effective means of transporting goods internationally [112, 146]. Shipping currently facilitates approximately 80% of global trade [112], and within the European Union, around 90% of external and 40% of internal trade is conducted via sea routes [65]. As a result, the maritime sector is viewed as a cornerstone of the European economy.

Despite its efficiency, the shipping industry is under growing pressure to reduce its environmental impact. In recent decades, international efforts such as COP21–COP26 [221, 223] have highlighted the role of greenhouse gas (GHG) emissions in driving climate change [222]. The transport sector contributes nearly a quarter of the EU's total GHG emissions [64], and maritime emissions are projected to increase by 90–130% by 2050 in the absence of intervention [66]. To counter this trend, the European Commission has adopted a strategy aiming to reduce transport emissions by 60% by 2050 (compared to 1990 levels), while the International Maritime Organization (IMO) has committed to a 50% reduction in emissions from shipping by the same deadline [112].

Meeting these ambitious targets requires a transformation in how ships are designed and operated. Technological innovations such as alternative fuels [67], hybrid or full-electric propulsion [63], and novel operational strategies like smart steaming [78] are creating new design constraints and opportunities. Designing ships that are fit-for-purpose under these constraints is increasingly complex, as each combination of mission profile, fuel system, and operational condition may require a distinct optimal configuration. This complexity has made fully integrated, holistic ship design difficult to implement in practice.

As a result, much of the recent research effort has focused on developing modular design optimization techniques that target specific ship components. One of the most influential of these is the hull form, which directly determines hydrodynamic resistance and therefore has a significant impact on fuel consumption and emissions. By optimizing the hull geometry to reduce resistance, it is possible to achieve meaningful gains in both environmental performance and operating cost. Even modest improvements in hull resistance can yield significant fuel savings over a ship's operational lifetime.

1

1.1 Hull Form Shape Optimization

Shape optimization refers to the process of modifying the geometry of a component–such as a ship hull–to achieve optimal performance based on predefined objectives [25, 98, 216, 239, 260]. Typically, the goal is to minimize the energy requirements, e.g., drag forces, while ensuring manufacturability and adherence to regulatory standards. This process may be driven by economic [144, 165], environmental [22], or combined objectives [6], and must adapt to evolving conditions such as fluctuating fuel prices [254] and environmental policies [193].

Optimization is usually performed with respect to specific Key Performance Indicators (KPIs), including but not limited to: energy consumption, fuel usage, drag, lift-to-drag ratio, instability, overheating, volume, mass, cost, and operational risk [115, 120, 239, 258].

The hull form geometry directly influences the ship's hydrodynamic resistance, and consequently its fuel consumption and emissions. Currently, there is no end-to-end tool to perform hull form shape optimization (e.g., that can automatically generate a geometric parameterization, evaulate canidate designs, and explore a candidate design space). Therefore, the design of hull forms is typically divided into three main stages [133, 175]: early-stage, mid-stage, and late-stage design. Each stage has distinct goals and toolsets.

During early-stage design (or concept design), the mission requirements may still be fluid, and it is beneficial to explore a wide variety of ship types [14, 124]. Recent developments in generative AI and parametric design tools allow for rapid exploration of alternative hull families. However, these tools often lack the resolution and control necessary for fine-grained hydrodynamic optimization and are not yet widely adopted in industrial practice [133].

Mid-stage design (or contract design) begins once a baseline hull is selected. Here, the design space is defined as a set of perturbations around a parent geometry [81, 228, 260]. This is the stage where simulation-based optimization, supported by Data-Driven Surrogates (DDS) models, is most widely used [54, 238]. Design variants are generated using parameterized deformation techniques and evaluated using a surrogate model trained on Computational Fluid Dynamics (CFD) data.

Late-stage design focuses on integration and compliance. The hull form is typically fixed, and the focus shifts to internal layout [259], safety [174], and regulatory compliance [102]. Changes at this stage are expensive and usually avoided. A summary of the aforementioned stages of hull form design is depicted in Figure 1.1.

While all stages are important, this thesis focuses on mid-stage design, where the most significant potential exists for performance improvement through shape optimization.

Historically, mid-stage design optimization relied on human expertise or parametric tuning of a base geometry [68, 104, 189]. Although these methods are robust and interpretable, they often restrict the exploration of unconventional or high-performance shapes. CFD is typically used to evaluate performance metrics, but CFD evaluations are computationally expensive, especially for large design spaces. To reduce this burden, DDS are increasingly used to approximate CFD responses [98, 140, 151, 158]. DDS models require significant training effort but can make near-instantaneous predictions, enabling fast optimization workflows. This data-driven approach forms the backbone of most modern mid-stage design studies.

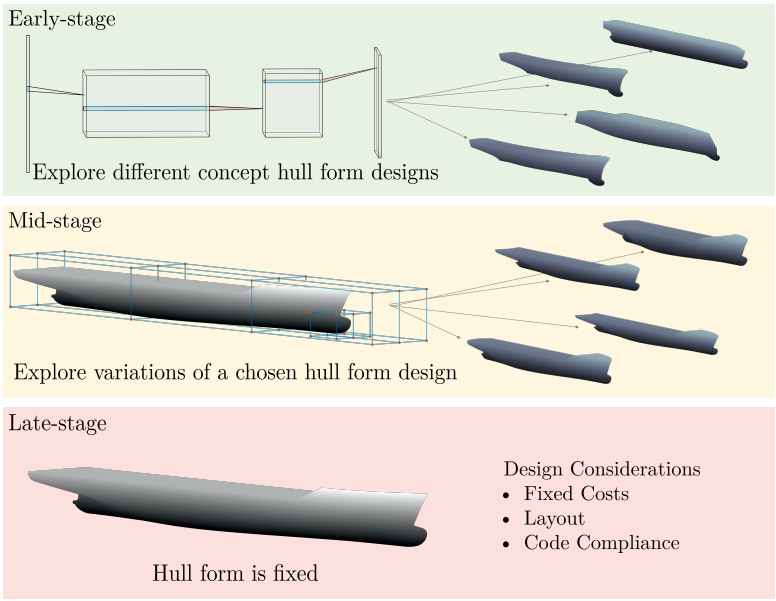


Figure 1.1: Hull form design stages.

1.2 Research gap and contribution

Despite recent advances, mid-stage hull-form optimization using data-driven surrogates still suffers from several key limitations. These limitations were identified through the literature review in Chapter 2 and are summarised below, together with the corresponding contributions of this thesis (Table 1.1).

Many existing DDS-based optimization frameworks generate candidate designs solely based on geometric parameter variation and hydrodynamic performance predictions. Physical feasibility checks, such as ensuring stability compliance, are often applied only as post-processing filters, meaning substantial computational effort may be spent on designs that are infeasible from the outset (G1). This thesis addresses this by embedding stability and hydrostatic constraints directly within the optimization loop, ensuring all evaluated designs remain physically plausible (Chapter 3).

Training accurate surrogates traditionally requires large CFD datasets covering the entire parameterized design space. This high computational burden restricts the number of scenarios and variants that can be evaluated, slowing the iterative design process (G2). Here, computational effort is reduced through a combination of stability-informed dimensionality reduction (Chapter 3), selective sampling (Chapter 3), and hybrid reuse of existing experimental (Chapter 4) and simulation data (Chapter 5).

In the literature, surrogate models are almost always tied to a specific geometric parametrization, such as a fixed Free-Form Deformation (FFD) lattice tied to a particular parent hull. When the parameterization changes, e.g., by modifying the FFD lattice or changing the parent hull, the surrogate must be retrained from scratch, requiring a new CFD dataset (G3). This thesis develops a decoupled, parameterization-agnostic surrogate modeling

framework that instead uses generalizable geometric and hydrostatic features, enabling model reuse across different encodings (Chapter 4, Chapter 5).

Current optimization pipelines treat each project in isolation, even when relevant CFD results from past studies exist. This results in repeated, costly CFD campaigns for similar hull types and operational ranges (G4). The framework proposed in this thesis integrates feature-based surrogate modeling to merge datasets from different sources and parameterizations, making simulation data reusable across design campaigns (Chapter 4, Chapter 5).

Most surrogate-based optimization studies focus on interpolation, where predictions are made within the boundaries (convex hull) of the training data. Extrapolation, predicting in regions outside the training set, is avoided or treated as unreliable (G5). In this thesis, extrapolation is reframed as a capability: by selecting suitable generalizable features and applying targeted training strategies, surrogates are made capable of reliable predictions beyond their original training domain, enabling exploration of new hull forms and operating conditions with limited additional CFD effort (Chapter 4, Chapter 5).

There is little guidance in the literature on how to transfer knowledge between hull families. Design and optimization efforts are often restarted from scratch when switching to a different parent hull (G6). This work demonstrates a novel framework for transferring knowledge across different hull families for resistance prediction, showing how surrogates trained on one hull family can be adapted to others, thus reducing the computational requirements for developing surrogates for the new designs (Chapter 5).

Table 1.1 maps these six research gaps to the specific contributions of this thesis and the chapters in which they are addressed.

Table 1.1: Summary of literature gaps and corresponding contributions of this thesis.

Gap	Current Literature	Contribution	Chapter(s)
G1	Surrogate models often lack integrated physical constraints, risking non-physical or infeasible designs.	Embedded physical plausibility checks (IMO stability code, hydrostatic constraints) directly into the optimization loop.	Ch. 3 – Plausibility integration into optimization.
G2	High computational cost in mid-stage design limits exploration of design space.	Reduced Data generation effort by via improved sampling, stability constraints, dimensionality reduction, and hybrid data reuse.	Ch. 3 – Selective sampling to improve plausibility; Ch. 4 - Training from existing experiental data; Ch. 5 – Training from existing simulation data.
G3	Surrogate models in hull-form optimization are tied to specific geometric parameterizations, preventing reuse across different parent hulls or encoding methods.	Developed a parameterization-agnostic (decoupled) framework for surrogate modeling, enabling simulation data reuse across studies with different parameterizations.	Ch. 4 – Learning surrogate models from independent feature space; Ch. 5 – Expanding independent feature space to describe more complex hull forms.
G4	Data reuse across projects is not addressed in most optimization pipelines, leading to repeated CFD campaigns.	Introduced feature-based surrogate modeling to integrate data from multiple sources and parameterizations, reducing the need for new CFD datasets.	Ch. 4 – Decoupled framework development; Ch. 5 – Simulation data reuse.
G5	Existing works focus on interpolation; extrapolation is avoided or treated as unreliable.	Treated extrapolation as a design capability, training surrogates to generalize to unseen hull forms and operational conditions.	Ch. 4 – Extrapolation experiments; Ch. 5 – Extgrapolation between families of cargo vessels.
G6	No established strategy for transferring models between hull families.	Demonstrated cross-hull transfer learning for resistance prediction, enabling knowledge transfer to new designs.	Ch. 5 – Generalizing surrogate development between hull forms.

1

1.3 Structure

This dissertation is organised to progressively address the six research gaps identified in Section 1.2, moving from a review of existing approaches to the development and application of a generalisable, parameterization-agnostic framework for hull-form optimization.

Chapter 2 presents a comprehensive review of the state of the art in optimization using computational fluid dynmaics data-driven surrogates. The review is structured around the four core stages common to most workflows:

1. geometric parameterization and KPI definition;
2. CFD-based data generation and surrogation;
3. surrogate-based optimization;
4. physical plausibility check.

For each stage, the limitations reported in the literature are analysed, leading directly to the formulation of research gaps G1–G6. It is worth mentioning that although airfoils differ geometrically and physically from ship hulls, airfoil optimization studies are deliberately included in the literature review. This is because both domains share a common methodological foundation based on using CFD DDS models to accelerate design-space exploration and performance optimization. Many of the surrogate-based optimization techniques were first developed and validated within the aerodynamic community, where data availability and standardized benchmarks enabled methodological progress. Therefore, by including these studies in the require we are able to cross-fertilize between airfoil and hull-form optimization research. For this reason, while numerical results from airfoil cases are not compared directly to ship data, their methodological insights underpin several of the approaches relevant to this work.

Chapter 3 addresses G1 and G2 by presenting a method for stability-constrained shape optimization using surrogate models. Physical feasibility constraints, specifically the IMO Intact Stability Code, are embedded directly into the optimization loop to prevent evaluation of non-viable designs. In parallel, computational cost is reduced through stability-informed dimensionality reduction and selective sampling strategies. The method is validated on a single parent hull family, allowing controlled assessment of improvements in plausibility and efficiency.

Chapter 4 addresses G2, G3, G4, and G5 by introducing a novel decoupled framework that removes the dependency between the surrogate model and the underlying geometric parameterization. The framework enables existing experimental data to be used to devlop the surrogates thereby reducing the computational load (G2). The framework uses generalizable hydrostatic and geometric features to develop the surrogates independent from the specific parameterizations and parent hulls (G3 and G4). It also reframes extrapolation as a design capability (G5), with different and increasingly challenging experiments showing how surrogates can be trained to generalize beyond the boundaries (convex hull) of the training set. Initial validation is performed using reduced-complexity hull families (sailing yacht hulls from the Delft Systematic Yacht Hull Series) to isolate and evaluate each methodological component.

Chapter 5 focuses on G2, G3, G4, G5, and G6 by applying the decoupled framework to more complex hull forms, specifically container ships. This chapter demonstrates reusing simultaion data to balance surrogate model accuracy and computational cost in an industrially relevant context (G2). This chapter expands on the meth-

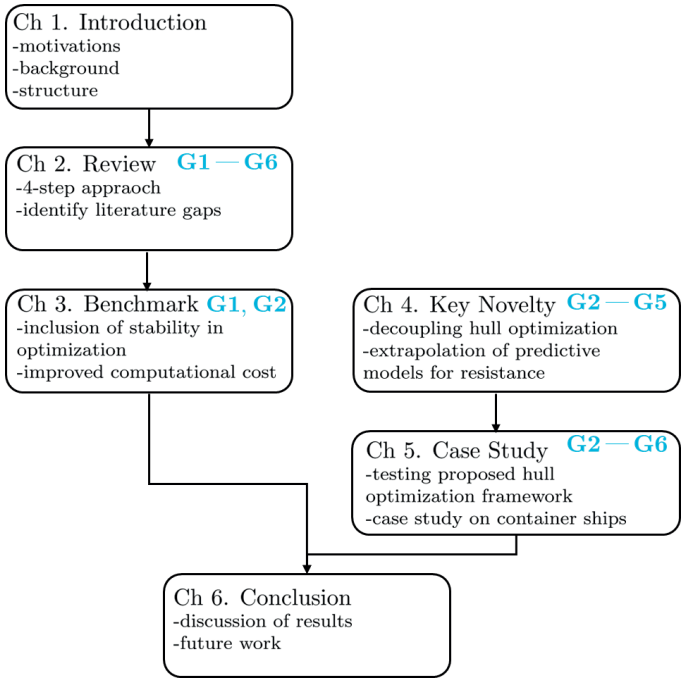


Figure 1.2: Structure of this thesis.

ods developed in Ch. 4 to adapt the surrogate models for more complex hull topologies (G3, G4). The extrapolation capability is evaluated for realistic operational conditions (G5), and the potential for cross-hull transfer learning (G6) is demonstrated by adapting models trained on one hull family to predict resistance for others.

Finally, Chapter 6 summarises the key contributions of the thesis, reflects on their implications for scalable and sustainable ship design, and outlines directions for future work.

2

Problem Formalization and State–of–the–art Review

Shape optimization of vessel hulls and airfoils is a critical step to ensure optimal performance and minimal environmental footprint. Usually, their design is an adaptation of an existing one, not optimized for the specific Key Performance Indicators (KPIs) like the drag of a hull or lift and drag of an airfoil, or the result of a mix between human experience and numerical optimization approaches. Nowadays, the state-of-the-art approach for shape optimization of hulls and airfoils is based on Computational Fluid Dynamics (CFD) Data-Driven Surrogate (DDS) models, and consists of four steps. First, a parametrization and parameter ranges are defined, with more or less human intervention, to build a shape design space. Unfortunately, an accurate estimation of desired KPIs (e.g., drag resistance for the hull or lift and drag for airfoils) based on the shape parameters is computationally expensive, requiring CFD, which prevents the direct numerical optimization of these KPIs. For this reason, the second step is to sample a few specific and representative shapes from the design space, based on human experience or through more complex strategies, and evaluate their KPIs using CFD. Then, a DDS is developed based on the generated data, and possibly already available data, which is computationally expensive to construct but inexpensive for making predictions. Thanks to this particular property of the DDS, as the third step, the latter can be directly incorporated into an optimization loop to retrieve a series of candidate geometries on the Pareto front of the KPIs. Finally, the fourth step consists of validating the physical plausibility of the results of this chain by verifying, with CFD, that the DDS and the optimizer did not induce physically implausible shapes. This work will review the literature on the just-described four steps of shape optimization of hulls and airfoils leveraging CFD DDS models. In particular, we have selected the top publications according to strict criteria, performed a critical review accompanied by informative summary tables, and finally, discussed the open problems and future perspectives in the field.

This chapter is reproduced from J. M. Walker, A. Coraddu, and L. Oneto. A review on shape optimization of hulls and airfoils leveraging Computational Fluid Dynamics Data-Driven Surrogate models. *Ocean Engineering*, 312: 119263, 2024.

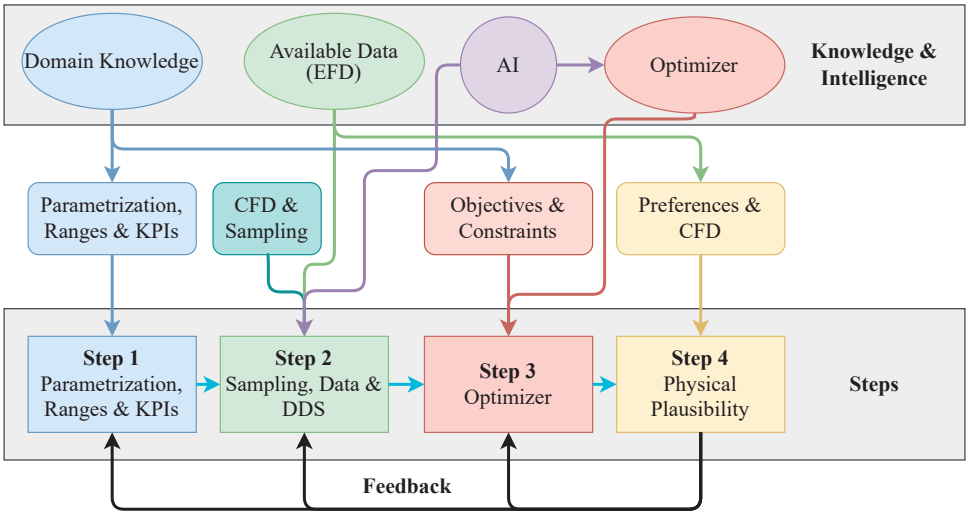


Figure 2.1: The graphical abstract of this review.

The application of the DDS to shape optimization of hulls and airfoils is an area of research that remains vibrant. Previous reviews and surveys on shape optimization methods [11, 73, 125, 130, 135, 243] have primarily focused on the optimization aspects, and, to the best of the authors' knowledge, there is no recent review or survey paper holistically summarizing the state-of-the-art on shape optimization of hulls and airfoils leveraging CFD DDS models. A graphical abstract of this review is reported in Figure 2.1. This review will focus on the four main steps related to shape optimization of hulls and airfoils leveraging CFD DDS models

Step (1) shape parametrization, parameter ranges, and KPIs definition;
Step (2) sampling, data generation with CFD possibly integrating already available shapes, and DDS construction;
Step (3) shape optimization leveraging the CDF DDS;
Step (4) physical plausibility of the candidate shapes and feedback.

Step (1) deals with the parametrization of the geometry and the definition of the parameter ranges and the desired KPIs for the specific novel hull or airfoil design. The parametrization is responsible for translating the geometry into a numerical representation and must be homomorphic (a unique set of parameters is matched to only one shape in the design space and vice-versa [148]). Parametrization must also be informative (to allow for the prediction of the desired KPIs), intelligible (to allow for interpretation and test the physical plausibility of the results), and synthetic (it must not contain redundant information) [200]. Parameter ranges should be defined based on domain knowledge by domain experts to construct a shape design space large enough to allow for meaningful modifications and small enough to allow for its actual manufacturing [70, 98]. KPIs should well characterize the performance of the geometry. They can be both explicit (e.g., drag of a hull or lift and drag of an airfoil) or implicit (e.g., the range of the parameters can impose the fact that some geometries should not be explored since they are too hard to

manufacture) [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261]. Parametrization and parameter ranges affect both the complexity of estimating the KPIs (see Section 2.2) and the effectiveness of the optimizer in finding designs that can be actually manufactured [224]. In fact, if we rely solely on CFD to explore a large design space, it is often tricky to calibrate the simulations accurately. If instead, we use a CFD DDS on a large design space, a vast amount of data is necessary that can be hard to retrieve (i.e., relying on EFD data) or computationally expensive to produce (i.e., relying on CFD simulations). For these reasons, Step (1) usually starts with the design requirements, mission profiles, mission constraints, and design constraints that define a raw starting point for the design process [7, 105]. It is not possible, nor meaningful, to think about a fully automated design process that does not exploit the domain and expert knowledge such as reference geometries and manufacturing constraints [7]. The level of human intervention in this phase depends on the type of project, its cost, its scope, and delivery time [46]. Note that these steps deeply influence the quality of final results: a well-constrained design process will facilitate the success of the next steps [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261]. As we will see later in Section 2.2.1, a number of approaches to parametrization exist in the literature, either based on modeling shapes in parametric design software [98, 140, 158] or deforming a specific parent shape [36, 70, 82, 140, 156, 260], with both approaches having varying degrees of human intervention and their associated strengths and weaknesses.

Step (2) deals with the accurate estimation of the desired KPIs based on the shape parameters defined at Step (1). In order to perform an accurate estimation, high-fidelity CFD [59] (e.g., Direct Numerical Simulation if computationally feasible [204], or an engineering approximation using Reynolds-averaged Navier–Stokes equations [19, 36, 140, 151, 156, 158, 172, 185, 218], or Large Eddy Simulation [16]) is needed, but its computational requirements are incompatible with an automated optimization process that usually requires several thousand shapes to be evaluated [73, 118, 209] (see Step (3)). In fact, high-fidelity CFD are based on iterative methods to solve the underlying physics of the problem at hand and require a significant computational effort [85, 118, 135, 243]. Low-fidelity CFD are much cheaper in terms of computational requirements [130], but most of the time much less accurate with respect to high-fidelity CFD [212] and still much more computationally demanding with respect to the DDS [85]. In fact, the DDS relies on a simple idea: instead of depending on physical knowledge of the phenomena to make predictions, a model is built based on examples (data) of the input/output (parameters/values or parameters/KPIs) relationships [114, 125] under exam leveraging state-of-the-art Artificial Intelligence (AI), and especially Machine Learning (ML) techniques [5, 80, 202]. The positive aspects of the DDS lie in the fact that the predictions are computationally inexpensive since the function approximating the input/output is cheap to evaluate. For this reason, DDS models have recently attracted the attention of researchers as accurate and computationally inexpensive surrogates of high-fidelity CFD [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261]. However, this advantage comes with a price. In fact, building a DDS is both data demanding and computationally expensive [85, 118, 135, 243]. For what concerns the data, in-field data requires performing trials [15, 147] (taking months), EFD data requires model scale tests [70, 98, 122] (taking weeks), and CFD data requires computationally expensive simulations [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261] (taking days). Moreover, building a DDS based on this data requires computationally intensive procedures,

the so-called training phase, which comes with its own computational burden (taking hours) even if this burden is negligible with respect to the one needed to generate or collect the data [253]. The key advantage of the DDS is that once it is constructed, the forward phase time is negligible (milliseconds or less), enabling its use with any state-of-the-art numerical optimizer [183, 191, 226]. For this reason, in Step (2), a few specific and representative shapes are sampled, based on human experience or through more complex strategies [9, 12], from the design space and fed into the CFD to produce the related data, which is added to possibly already existing data (e.g., in-field or EFD data). This step aims to build a representative dataset of examples to train the DDS [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261]. The more representative the input/output relation dataset is, the less data is needed for the DDS [202]. Consequently, there are both theoretical and practical reasons to pay appropriate attention at this stage. Theoretically, the database contains the phenomenon we want to learn, and careful consideration regarding the geometry sampling and KPIs estimation is required to capture the desired behavior. Building a representative dataset with a smart sampling strategy is an important and challenging problem per se [9, 12] and in this survey, we will review the most exploited strategies in the context of hulls and airfoils shape optimization leveraging CFD DDS models (see Section 2.2.2). Practically, an extensive CFD campaign is often required to build the database, which demands a significant amount of time and computational resources [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261]. In an ideal world we would rely on already available data (e.g., historical experimental campaigns or simulation) for the dataset to circumvent this challenge. However, at present, this is not always possible due to the lack of historical data and the reliance on very specific parametrizations and parameter ranges that are unique to their particular investigation [36, 70, 82, 140, 156].

In Step (3), we leverage the outputs of Step (1) and Step (2) to automatically search for optimal candidate geometries. For this reason, we are mainly concerned with two connected problems: (i) the formulation of the objective function we want to optimize and (ii) the formulation of the constraints we will apply to the objective. In the context of shape optimization of hulls and airfoils, the objective function is a multi-objective one composed of all the KPIs we want to optimize. Some of these KPIs will be simple functions of the shape geometry parameters (e.g., the volume of a hull), while others can be more complex functions needing a CFD DDS (e.g., the airfoil's drag and lift coefficients). The constraints, instead, are derived from the domain knowledge and represent the fact that not all the parameter space, defined at Step (1), correspond to a feasible shape (e.g., due to cost considerations [246], performance constraints [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261], manufacturing constraints [107], or regulatory constraints [108]).

It is worth mentioning that formulating the constraints may also require the computation of complex quantities via a CFD DDS. To deal with the multi-objective nature of the problem, the most common approach is to replace the multiple objectives with a weighted sum of the different objectives [62] but also other approaches exist [11, 23, 49, 62, 73, 114, 125, 130, 135, 230, 243]. In this way, the problem becomes a constrained single-objective optimization problem that can be solved with state-of-the-art optimizer [23, 127, 153, 183, 191, 215, 226]. Note that, nowadays, optimizers can also be empowered with AI-based techniques [215], which allows them to reduce their computational requirements or improve their effectiveness in finding good solutions. Nevertheless, given

the multi-objective nature of the original problem, it is required to create the so-called Pareto frontier allowing to find the set of all Pareto efficient solutions [23, 62], namely all the solutions for which is not possible to find more than one objective better than the one of the solutions itself. To tackle this problem, usually, multiple single-objective optimization problems need to be solved (e.g. single-objective problems resulting from the weighted sum of the different objectives of the original multi-objective problem with different configurations of the weights [62]) resulting in additional computational overhead [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261].

The last step, the Step (4), is probably the most important one not just because it takes the output of all the previous steps to produce a candidate shape but because it also provides feedback and refinements for previous steps able to empower them [73, 130]. In fact, by exploiting shape design preferences from experienced designers' expertise, it is possible to choose some candidate shapes on the Pareto front, potentially exhibiting the best trade-off between the KPIs for the specific application. This choice obviously does not conclude the design process. In fact, these new shape designs (i) must be physically plausible (e.g., it is actually possible to manufacture them and it is cost-effective) and (ii) must actually provide the required performance. This can be verified with different levels of accuracy [70, 98, 129]. In the first level, high-fidelity CFD are used since they can surely be exploited to study a few candidate shapes [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261]. Then a model scale test can be performed to confirm the CFD [70, 98]. Finally, a field trial will be performed to confirm the results of model scale and CFD results [157, 214]. If, during this verification, inconsistencies are encountered, or the candidate shape does not exhibit satisfactory performances, these results are used as feedback to modify and improve the previous steps (e.g., modifying the parameter ranges in Step (1), improving the DDS accuracy by including more data in Step (2), or improving the definition of the constraints in Step (3)). The procedure is repeated until a satisfactory design is obtained [73, 130].

In this review, it is not possible to consider all the publications in this vibrant field of research due to the large volume of works. Hence, we defined criteria, taking inspiration from [210], to narrow down the top publications according to

- problem category: we focus primarily on shape optimization of hulls and airfoils but, at times, may also include some adjacent shapes;
- publication date: we focus on works published after the year 2015;
- publication quartile: we favor publications in journals belonging to the first quartile (and in some specific cases also the second one);
- number of citations: we favor publications that have averaged at least three or more citations per year since issuing.

Subsequently, we performed a critical review of the narrowed literature accompanied by informative summary tables.

The rest of the review is organized as follows (see Figure 2.2). Section 2.1 contains the problem formalization which is necessary to understand the survey. Section 2.2 presents the analytical review on shape optimization of hulls and airfoils leveraging CFD DDS models following the four-step approach of Figure 2.1. Section 2.3 discusses the open problems and future perspectives of this vibrant field of research. Finally, Section 2.4 concludes the review.

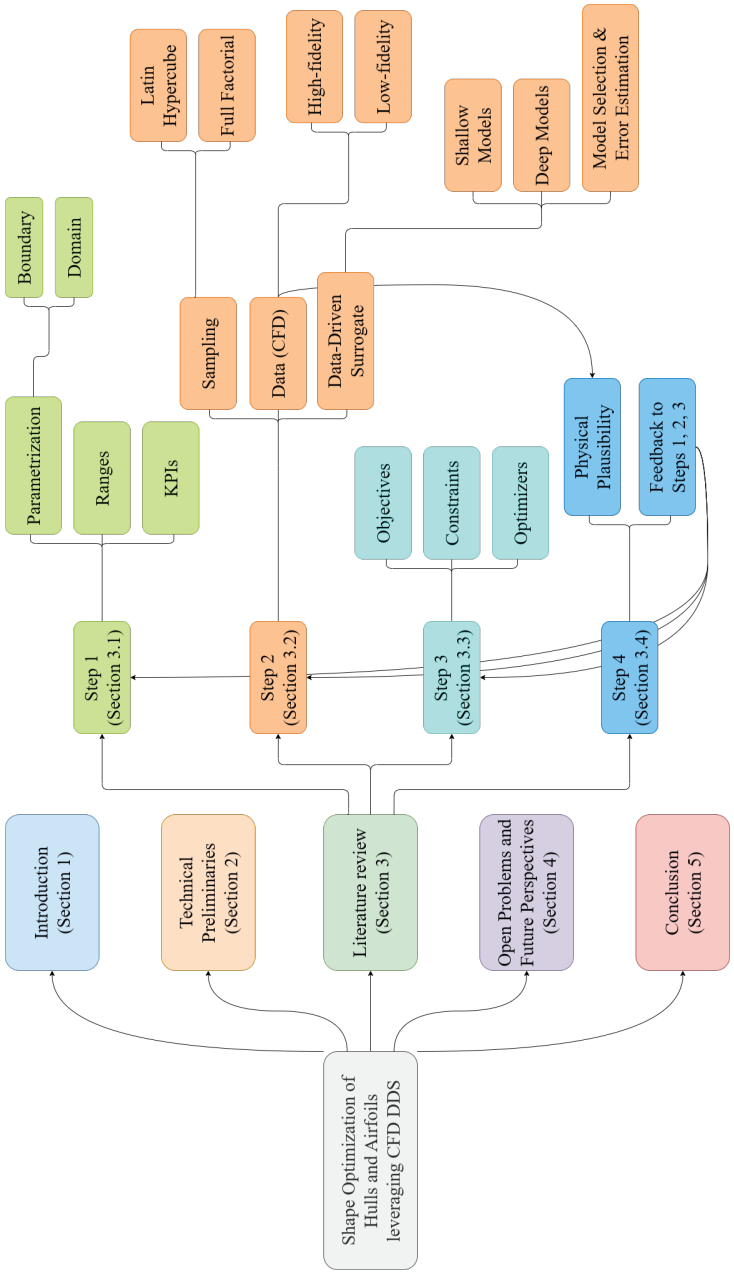


Figure 2.2: Graphical Table of Contents.

2.1 Problem formalization

The goal of shape optimization of hulls and airfoils is to find the optimal shape Ω^* in a set of possible ones $\mathcal{D}$, where $\mathcal{D}$ are all the possible candidate geometries that we can explore given, e.g., cost constraints, performance constraints, manufacturing constraints, or regulatory constraints. The optimality principle is guided by one or more KPIs depending on the specific application, e.g., the drag of a hull [25, 29, 36, 60, 98, 128, 140, 156, 158, 216, 244, 260] or the lift and drag of an airfoil [58, 75, 84, 88, 90, 129, 145, 151, 172, 185, 207, 218, 261]. We then define, in Step (1), k KPIs as functions of a particular shape as follows

$$\mathtt{J}_i(\Omega) : \mathcal{D} \to \mathbb{R}, i \in \{1, \cdots, k\}, \Omega \in \mathcal{D}, \tag{2.1}$$

and consequently, our final goal to optimize shape of hulls or airfoils can be formalized as follows

$$\min_{\Omega \in \mathcal{D}} \quad \mathtt{J}_1(\Omega), \cdots, \mathtt{J}_k(\Omega). \tag{2.2}$$

Note that, Ω is the actual 3D (or in some cases 2D) geometry, which, although technically possible [117], is hard to handle/modify directly through human intervention because there is a vast amount of possible modifications that can be made and it is preferable to structure the modifications with a method that is logical, consistent, and repeatable [36, 58, 84, 90, 140, 145, 156, 158, 172, 185, 207, 218, 261].

For this reason, in Step (1) a particular parent shape Ω^p is taken as reference geometry and then $\Omega \in \mathcal{D}$ are parametrized so that a modification vector $x \in \mathbb{R}^d$ can be applied to Ω^p according to a rule $\mathtt{R}$ to generate $\Omega \in \mathcal{D}$ [19, 36, 58, 84, 90, 98, 140, 145, 151, 172, 185, 207, 218, 261]. Note that this leads to an approximation since

- $\mathtt{R}$ may not be able to generate all $\Omega \in \mathcal{D}$ with $x \in \mathbb{R}^d$ [36, 98, 151, 158];
- some $x \in \mathbb{R}^d$ applied to the rule $\mathtt{R}$ may generate $\Omega \notin \mathcal{D}$ [19, 36, 58, 60, 84, 90, 98, 140, 145, 151, 172, 185, 207, 218, 261].

For the first problem, the trade-off between exploring more $\Omega \in \mathcal{D}$ and increasing the dimensionality of the parametrization, namely d, is determined by domain experts [36, 98, 151, 158]. For the second problem, some specific $x \in \mathbb{R}^d$ may lead to $\Omega \notin \mathcal{D}$ [60], so we always have to check for this consistency by adding constraints to $x \in \mathbb{R}^d$ [19, 36, 58, 84, 90, 98, 140, 145, 151, 172, 185, 207, 218, 261].

The modification vector x handles the modifications to Ω^p easily, retrieving a particular $\Omega \in \mathcal{D}$. So x is just instrumental. The elements in x must be independent because we must be able to change every element independently from the others, i.e., two dependent elements would result in a $(d-1)$-dimensional modification vector. In order to define $\mathtt{R}$, a number of approaches exist in the literature, the two most common being boundary [74] and domain-based parametrizations [197]. For the former, in its most naive form, $\mathtt{R} : \mathcal{D} \times \mathbb{R}^d \to \mathcal{D}$ can correspond to the displacement of coordinates on the surface of the shape. Whereas, a domain-based parametrization considers a control volume around the shape and the rule corresponds to the displacement of control points for a deformation-based, e.g., a Free-Form Deformation (FFD) parametrization [197]. However, regardless of the implementation, the rule $\Omega = \mathtt{R}(\Omega^p, x)$ must be homomorphic [247] and maps a parent shape $\Omega^p \in \mathcal{D}$ into another one $\Omega \in \mathcal{D}$ based on $x \in \mathbb{R}^d$. For what concerns the

homomorphism of R, what we require is a bijective homomorphism, namely

$$\exists\, R^{-1} :\ R^{-1}\big(R(\Omega, x), x\big) = \Omega, \qquad \forall\, x \in \mathbb{R}^d,\ \Omega \in \mathcal{D}, \tag{2.3}$$

or, in other words, a particular $x \in \mathbb{R}^d$ corresponds to one and only one modification of $\Omega \in \mathcal{D}$ and vice-versa. Consequently, Problem (2.2) can be reformulated subject to (s.t.) the described constraints as follows

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \mathtt{J}_1(\Omega), \cdots, \mathtt{J}_k(\Omega), \\ \text{s.t.:} \quad & \Omega = \mathsf{R}(\Omega^p, x), \Omega \in \mathcal{D}. \end{aligned} \tag{2.4}$$

Problem (2.4) remains mathematically and numerically challenging due to the complex relationships between the 3D (or 2D) geometries and the specific KPIs of interest [73, 130]. For this reason, in Step (1), we need to make some further reformulations and approximations to the problem at hand. For the multi-objectives of Problem (2.4), namely $\mathtt{J}_i(\Omega)$ with $i \in \{1, \cdots, k\}$, researchers formulate with different approaches (e.g., High/Low-fidelity CFD [129]) a more or less accurate estimation of $\mathtt{J}_i^{\mathsf{R}}(x)$ given x (which varies during optimization) and R (which is fixed) [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261]. For the constraints, namely $\Omega \in \mathcal{D}$ with $\Omega = \mathsf{R}(\Omega^p, x)$, researchers translate them into simple box constraints on x when possible, i.e., $l \leq x \leq u$ where $l, u \in \mathbb{R}^d$ and $l \leq u$ [36, 84, 140, 145, 151, 158, 172, 185, 218]. When this simplification is not possible (e.g., we need to have constraints on physical properties of Ω) researchers translate these constraints into c^{E} equality and/or $c^{\mathtt{I}}$ inequality constraints over more or less accurate functions of x that approximate the desired properties, namely $\mathtt{I}_i^{\mathsf{R}}(x)$ with $i \in \{1, \cdots, c^{\mathtt{I}}\}$ and $\mathtt{E}_i^{\mathsf{R}}(x)$ with $i \in \{1, \cdots, c^{\mathtt{E}}\}$ [19, 84, 98, 151, 156, 158, 185, 218]. As a consequence Problem (2.4) is reformulated as follows

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \mathtt{J}_1^{\mathsf{R}}(x), \cdots, \mathtt{J}_k^{\mathsf{R}}(x), \\ \text{s.t.:} \quad & \mathtt{I}_i^{\mathsf{R}}(x) \leq 0, i \in \{1, \cdots, c^{\mathtt{I}}\}, \\ & \mathtt{E}_i^{\mathsf{R}}(x) = 0, i \in \{1, \cdots, c^{\mathtt{E}}\}, \\ & l \leq x \leq u. \end{aligned} \tag{2.5}$$

Problem (2.5) usually raises computational challenges since, in order to compute some of the $\mathtt{J}_i^{\mathsf{R}}(x)$ with $i \in \{1, \cdots, k\}$, $\mathtt{I}_i^{\mathsf{R}}(x)$ with $i \in \{1, \cdots, c^{\mathtt{I}}\}$, and $\mathtt{E}_i^{\mathsf{R}}(x)$ with $i \in \{1, \cdots, c^{\mathtt{E}}\}$, the computational requirements may be prohibitive [85] (e.g., when using High-fidelity CFD [129]). In fact, these functions often approximate the physical properties and phenomena of Ω and while in some case an accurate enough estimation can be found with Low-fidelity CFD [128] (which require a comparatively low computational effort), in other cases, High-fidelity CFD are mandatory to reach a desirable level of accuracy [150].

To address this problem, in Step (2), researchers propose to use DDSs. DDSs focus on approximating a complex and computationally expensive function with another function that is computationally expensive to construct but computationally inexpensive to evaluate and is well suited to replace the ones in Problem (2.5) [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261]. In particular, given a complex and computationally

expensive relation $\mu : \mathbb{R}^d \to \mathbb{R}$, in our case one element in the subset of $\{\mathtt{J}_1^{\mathtt{R}}(x), \cdots, \mathtt{J}_k^{\mathtt{R}}(x), \mathtt{I}_1^{\mathtt{R}}(x), \cdots, \mathtt{I}_{c^{\mathtt{I}}}^{\mathtt{R}}(x), \mathtt{E}_1^{\mathtt{R}}(x), \cdots, \mathtt{E}_{c^{\mathtt{E}}}^{\mathtt{R}}(x)\}$ that are computationally expensive to evaluate, and a series of n samples, namely a dataset, of the input-output relation $\mathcal{S}_n = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ where $y_i = \mu(x_i)$ with $i \in \{1, \cdots, n\}$ we are able to generate a model $f(x)$ which is computationally expensive to build since

- building this model from $\mathcal{S}_n$ using ML is computationally expensive [187];
- retrieving $\mathcal{S}_n$ via CFD or EFD may be hard and is for sure computationally expensive [25, 229]. In fact, EFD suitable for the particular design may not be available due to its prohibitive cost [26, 176]. If instead we generate data via CFD, sampling a representative set of geometries is already a challenge, and then evaluating the performance of each geometry may take months due to the computational complexity of the CFD simulations [26];

but computationally inexpensive to estimate since, for the vast majority of ML algorithms, estimating $f(x)$ takes fractions of milliseconds [39].

Upon initial observation, one might question the logic behind this approach. In fact, in order to create the DDS we are spending a significant amount of computational power in both creating $\mathcal{S}_n$ with CFD simulations and $f(x)$ based on $\mathcal{S}_n$. It is easy to assume that this computational power can instead be used to solve Problem (2.5) directly, leveraging CFD simulations to estimate the complex relationships. This assumption is incorrect. In fact, when solving Problem (2.5) even using state-of-the-art optimizers, it is not uncommon to explore hundreds of thousands, if not millions, of values for x which may take years if we were to rely solely on CFD for the performance estimation [26]. Instead, to build the DDS, only a hundreds (maximum thousands) of samples are required for $\mathcal{S}_n$, needing only a few months of CFD simulations while building $f(x)$ may only take weeks [118]. Therefore, the computational savings are in order of magnitude thanks to the DDS [85, 118].

The problem of building a DDS can be then divided into two main sub-problems

- collecting/generating $\mathcal{S}_n$;
- building the actual DDS.

Regarding the first sub-problem, collecting/generating $\mathcal{S}_n$, it is possible to rely on already available EFD of CFD data when possible [69, 118], but, in most cases, data needs to be generated from scratch [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261]. In this case, the main problem is to decide how to sample the space induced by the modification vector x, which is already quite a challenge, and for which several proposals have been developed [9, 12].

Regarding the second sub-problem, building the actual DDS, it can easily mapped into a now-classical supervised ML problem, particularly an ML regression problem [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261]. In regression, we have an input space $\mathcal{X} \subseteq \mathbb{R}^d$ composed of d features (in our case the modification vector), an output space $\mathcal{Y} \subseteq \mathbb{R}$ (in our case one of the subsets of $\{\mathtt{J}_1^{\mathtt{R}}(x), \cdots, \mathtt{J}_k^{\mathtt{R}}(x), \mathtt{I}_1^{\mathtt{R}}(x), \cdots, \mathtt{I}_{c^{\mathtt{I}}}^{\mathtt{R}}(x), \mathtt{E}_1^{\mathtt{R}}(x), \cdots, \mathtt{E}_{c^{\mathtt{E}}}^{\mathtt{R}}(x)\}$ that are computationally expensive to evaluate), and a series of n samples $\mathcal{S}_n$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ $\forall i \in \{1, \cdots, n\}$. The scope is to learn the input/output relation $\mu : \mathcal{X} \to \mathcal{Y}$ based just on $\mathcal{S}_n$. Generally, μ is a probabilistic relation, but in our case, this relation is induced by, e.g., CFD models, so it is deterministic [41]. An ML regression algorithm $\mathcal{A}$, characterized by its hyperparameters $\mathcal{H}$, selects a model f inside a set of possible ones $\mathcal{F}$ based on the available data $\mathcal{A}_{\mathcal{H}} : \mathcal{S}_n \times \mathcal{F} \to f$.

Various ML algorithms exist in the literature [21, 80, 202, 206]. However, according to the no-free-lunch theorem [251], there is no *a priori* method for determining the best ML algorithm and best hyperparameters for a specific application for a particular problem, and the only option is to empirically test multiple approaches verifying which is actually the best one. This phase is referred to as model selection [169]. Broadly speaking, there are two main families of ML algorithms: shallow and deep models [80, 202]. For shallow models, $\mathcal{X}$ is first mapped, by means of an handcrafted feature engineering phase [30] or by means of a more or less elaborated procedure like the kernel trick [206], into a vector $\phi(X) \in \mathbb{R}^d$, named representation, able to well represent $\mathcal{X}$ while discarding the not useful information [170]. For deep models, $\mathcal{X}$ is mapped into a representation $\phi(X)$, but not with a fixed procedure, as ϕ is parameterized and learned directly from the data [32, 137]. Shallow models are the top-performing approaches in the case of non-structured (tabular) data while deep models are the top-performing approaches from structured (e.g., graphs and sequences) data[1]. Moreover, deep models usually require significantly more data with respect to the shallow ones to be trained [96, 137].

The error of f in approximating μ is measured by a prescribed metric $M : \mathcal{F} \to \mathbb{R}$. For what concerns the $M(f)$ many different metrics are available in literature [2, 205] both quantitative (e.g., the mean absolute error, the mean square error, the root mean square error, the coefficient of determination, and the mean absolute error in percentage to name a few [2]) and qualitative (e.g., scatter plots of the real versus predicted metrics [205]). In order to estimate the performance of the final model according to the desired metrics the error estimation phase needs to be performed [169].

Finally, in order to give some insights on what the algorithms actually learned from the data it is required to provide some explainability properties of the learned models [27, 57, 96]. For shallow models, feature ranking, namely how much the handcrafted features actually contribute to the prediction, is one of the most effective tools [57]. For deep models attention maps represent the state-of-the-art tools for explanations [91, 110].

At the end of Step (2) we obtain a new set of functions $\{\tilde{\mathtt{J}}^{\mathtt{R}}_1(x), \cdots, \tilde{\mathtt{J}}^{\mathtt{R}}_k(x), \tilde{\mathtt{I}}^{\mathtt{R}}_1(x), \cdots, \tilde{\mathtt{I}}^{\mathtt{R}}_{c^{\mathtt{I}}}(x), \tilde{\mathtt{E}}^{\mathtt{R}}_1(x), \cdots, \tilde{\mathtt{E}}^{\mathtt{R}}_{c^{\mathtt{E}}}(x)\}$ which are the computationally inexpensive counterparts of $\{\mathtt{J}^{\mathtt{R}}_1(x), \cdots, \mathtt{J}^{\mathtt{R}}_k(x), \mathtt{I}^{\mathtt{R}}_1(x), \cdots, \mathtt{I}^{\mathtt{R}}_{c^{\mathtt{I}}}(x), \mathtt{E}^{\mathtt{R}}_1(x), \cdots, \mathtt{E}^{\mathtt{R}}_{c^{\mathtt{E}}}(x)\}$ where, in some cases, we simply use the original function (if computationally inexpensive) or a surrogate (if computationally expensive). The result of this process is a reformulation of Problem (2.5) into the following one

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \tilde{\mathtt{J}}^{\mathtt{R}}_1(x), \cdots, \tilde{\mathtt{J}}^{\mathtt{R}}_k(x), \\ \text{s.t.:} \quad & \tilde{\mathtt{I}}^{\mathtt{R}}_i(x) \leq 0, i \in \{1, \cdots, c^{\mathtt{I}}\}, \\ & \tilde{\mathtt{E}}^{\mathtt{R}}_i(x) = 0, i \in \{1, \cdots, c^{\mathtt{E}}\}, \\ & l \leq x \leq u. \end{aligned} \tag{2.6}$$

Problem (2.6) is now computationally tractable and many approaches exist to tackle it [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261] and Step (3) exactly deals with this problem. The first challenge of Problem (2.6) is the fact that it is

[1]Results from the most popular machine learning website, Kaggle (`https://www.kaggle.com/`) show this to be the case when using real-world data.

characterized by multiple objectives. To deal with this challenge, multiple approaches exist [62, 145, 151, 156, 158, 172, 218]. Nonetheless, the vast majority of them reformulate the problem as a single objective problem as follows

$$\begin{aligned} \min_{x\in\mathbb{R}^d} \quad & \sum_{i=1}^{k} \lambda_i \tilde{\mathsf{J}}_i^{\mathsf{R}}(x) \\ \text{s.t.:} \quad & \tilde{\mathsf{I}}_i^{\mathsf{R}}(x) \leq 0, i \in \{1, \cdots, c^{\mathsf{I}}\}, \\ & \tilde{\mathsf{E}}_i^{\mathsf{R}}(x) = 0, i \in \{1, \cdots, c^{\mathsf{E}}\}, \\ & l \leq x \leq u. \end{aligned} \tag{2.7}$$

where $\lambda_i \in [0, 1], \forall i \in \{1, \cdots, k\}$ are constants such that $\sum_{i=1}^{k} \lambda_i = 1$ that weights more or less a particular objective [62]. Varying $\lambda_i \; \forall i \in \{1, \cdots, k\}$ in Problem (2.7) it is possible to find all the possible solutions of Problem (2.6) [62]. Note that, among all of the possible solutions, just some of them are actually meaningful: the ones on the Pareto front [138, 151], obtained via the Skyline operator [138], namely, the solutions that are not dominated by any other one according to at least one objective of Problem (2.6) [138, 151].

Problem (2.7) is now a computationally tractable single objective constrained optimization problem, namely a standard optimization problem to be addressed [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261]. In general, Problem (2.7) is both non-linear and non-linearly constrained [19, 84, 98, 151, 156, 158, 185, 218]. To address Problem (2.7), many approaches exists [183, 191, 226] but they can be grouped in two main families

- Problem (2.7) can be relaxed, either globally or in iterative local stages, into a convex formulation. This transformation enables the use of highly efficient algorithms designed specifically for convex optimization. Specifically
  - global convex relaxation reformulates Problem (2.7) into another problem which tries to approximate it at best with a single convex objective and a series of convex constraints [19, 84, 145, 185]. This approach is not always possible since there is an obvious trade-off between accuracy of the approximation of Problem (2.7) and convexity of the associated resulting optimization problem [139];
  - iterative local convex relaxation means that, starting from a point the domain of Problem (2.7), the problem is locally approximated with a linear or quadratic or convex objective and with a convex domain (defined by linear or quadratic or convex constraints). Then, the solution of this convex problem is used as a new starting point, and the steps are repeated until convergence to a local minima [191]. To improve the quality, a multiple starting point approach (colloquially, multi-start), is adopted [185];
- Problem (2.7) is directly addressed with optimizers that can directly handle non-linear and non-linearly constraint optimization problems [36, 140, 151, 158, 172]. Examples of these algorithms are the evolutionary algorithms [226].

Finally, Step (4) focuses on checking the physical plausibility of the solutions on the Pareto front of Problem 2.7. In particular, it is important that these solutions do not degenerate into a physically implausible one due to the many different approximations that stem from Problem 2.2 to Problem 2.7 (i.e., due to parametrization in Step (1), surrogation

in Step (2) [27, 41, 47, 116], and optimization in Step (3) [19, 36, 70, 84, 98, 98, 129, 140, 145, 151, 156, 158, 172, 185, 218]). This check is usually performed in two stages.

The first stage of checks is performed during surrogation in Step (2) [27, 41, 47, 116]. In fact, surrogating CFD with a DDS may lead to a loss in physical plausibility due to the fact that the DDS can take shortcuts [27] hidden in the data in order to learn the desired relation [41, 47, 116]. This is extremely counterproductive when DDSs are exploited in optimization since these shortcuts can trick the optimizers into solutions that are physically implausible [41]. Therefore, it is worthwhile to check the DDS for physical plausibility by means of two approaches

- the first, is to challenge the DDS into solving increasingly complex extrapolation scenarios [41, 47, 116, 239]: the DDS is first trained on a subset of data that represent particular physical conditions (e.g., low cruise speed when predicting the hull resistance) and then tested for accuracy in a different physical conditions (e.g., high cruise speed when predicting the hull resistance). A high accuracy in challenging extrapolation scenarios increases the trust in the DDS and checks for hidden shortcuts in the data [41, 116, 239];
- the second one is to inspect the behavior of the DDS through explainability, that tries to open the black box [76] of the DDS for an expert to check the learned relation [162].

The second stage of checks is performed during optimization in Step (3) [19, 36, 70, 84, 98, 98, 129, 140, 145, 151, 156, 158, 172, 185, 218]. In particular, given the solution of the Pareto front, they are checked for physical plausibility and quality by a series of increasingly complex, time consuming, and costly procedures [19, 36, 70, 84, 98, 129, 140, 145, 151, 156, 158, 172, 185, 218]. The first one is a check by a human expert who searched for implausible or non-manufacturable features of the geometry [19, 36, 84, 98, 140, 145, 156, 158, 172, 185, 218]. This step is crucial since it allows both to verify the quality of the pipeline and gives insights to experts on ideas for geometries that can be reused in other projects. Then, once this step is passed, the geometries are verified to have the desired performance via CFD [19, 36, 84, 98, 140, 145, 156, 158, 172, 185, 218]. This check actually gives insights on both parametrization in Step (1)(e.g., the ranges were too tight/loose or the parametrization too rough/detailed), surrogation in Step (2) (e.g., some degenerate solution or inaccuracy of the DDS are discovered), and optimization in Step (3) (e.g., some approximations during optimization were too imprecise). If the CFD checks are passed, EFD are checked [70, 98, 122]. If also EFD checks are passed, sea trials are performed [15, 147]. At each one of these checks, significant human intervention is required to discover issues, criticalities, and provide feedback to the previous step for improvements [15, 19, 36, 70, 84, 98, 122, 129, 140, 145, 147, 151, 156, 158, 172, 185, 218].

2.2 Literature review

This section will report the actual review of the current literature on the four main steps described in Section 2.1, summarized in the graphical abstract reported in Figure 2.1, and structured as depicted in Figure 2.2. Specifically, Section 2.2.1 reviews Step (1), Section 2.2.2 reviews Step (2), Section 2.2.3 reviews Step (3), and Section 2.2.4 reviews Step (4).

2.2.1 Step (1) Shape parametrization, parameter ranges, and KPIs definition

As previously described, Step (1) mainly deals with three aspects: parametrization, parameters ranges, and KPIs definition that will be reviewed in this section. We will also summarize the main work in the literature according to the most critical aspects identified during the review of the three main aspects related to Step (1).

Parametrization

Parametrization deals with translating the modification of a particular parent geometry (e.g., a hull or airfoil) into a numerical representation through a modification vector. The dimensionality of the modification vector, in coordination with the parameter ranges, defines the shape design space.

In general, it is possible to find two dominant approaches to parametrization in the literature

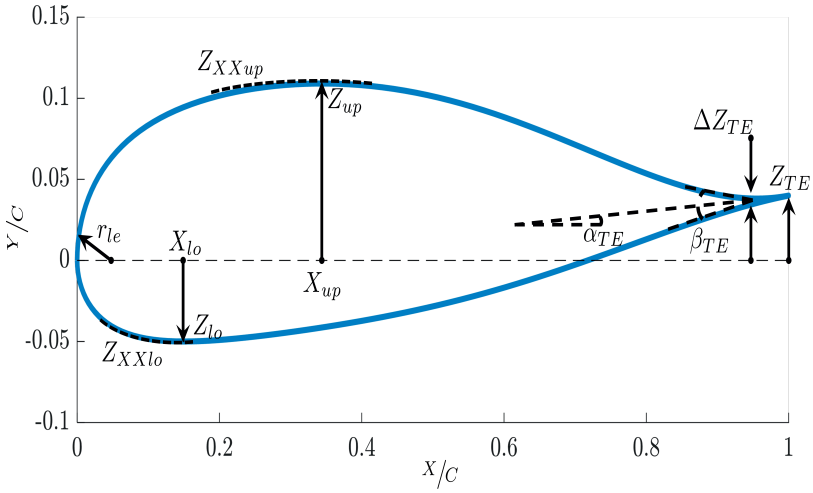
- Boundary-based parametrization: concerns parametrizing the boundary of the shape based on splines or physical attributes (e.g., length, breadth, depth, etc.) to drive the design;
- Deformation-based parametrization: concerns parametrizing the domain surrounding a parent geometry and uses non-physical descriptors to describe modifications from the parent shape (e.g., Free-Form Deformation).

Both of these approaches are demonstrated in Figure 2.3.

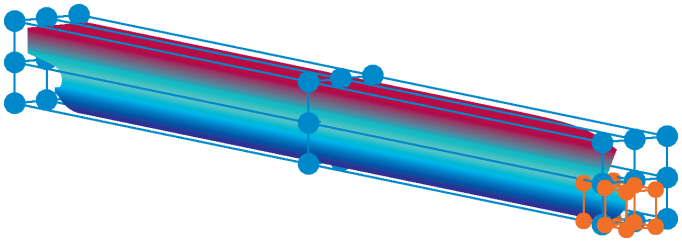
Boundary-based Parametrizations Boundary-based parametrizations exist in many forms in the literature. Every parametrization is specific to the considered shape.

For airfoils, it is possible to parametrize a geometry with the Class/Shape function Transformation (CST) approach [131] demonstrated in [84]. Similarly, the Hicks-Henne approach [145, 152] enables the parametrization of an airfoil with 2÷25 design variables. Whereas the PARSEC-11 approach [211] (leveraged in [185]) describes an airfoil with 11 parameters (shown in Figure 2.3a) by defining: the leading edge radius (r_{le}), the upper and lower curvature (Z_{XXup} and Z_{XXlo} respectively), the coordinates of the lowest point on the XZ plane (X_{lo}, Z_{lo}), the coordinates of the highest point on the XZ plane (X_{up}, Z_{up}), the wedge angle (β_{TE}), the trailing edge angle (α_{TE}), the height of the trailing edge (Z_{TE}), and finally, the trailing edge thickness (ΔZ_{TE}).

The most dominant boundary parametrizations for hulls and airfoils are curves or parametric models, or a combination of both. Parametrizations based on curves [142, 239, 242, 257], for example, Bezier curves [52] and Non-Uniform Rational B-Spline (NURBS) [203, 258], develop a relationship between the design parameters and the feature curves that define the geometry (either through control points or another functional relationship). It is worth mentioning that shape parametrizations based on this approach can lead to high dimensional parameter representations for complex geometries, and while there is a clear benefit to increasing the size of the shape design space, the resulting parametrization requires larger computational efforts with regard to the construction of a dataset and the optimization (see Sections 2.2.2 and 2.2.3). For this reason, the number of free parameters following this approach is usually reduced to around 6÷14 in the referenced works, and the performance of the DDS models is between $\sim$ 2÷4% [142, 257, 258]. However, preserving



(a) Boundary-based parametric model of an airfoil utilizing the PARSEC-11 parameters [211].



(b) Domain-based (e.g., FFD) parametrization of a hull with both global and local CVs.

Figure 2.3: Boundary and Domain-based parametrizations.

physical relevance in the parametrization scheme allows the geometries to be easily reconstructed from their parametric representations and the importance of each parameter remains intuitive [142, 143, 180, 257]. Maintaining the physicality of the parameters has a key advantage for reconstructing the geometry and preserving the knowledge obtained through shape optimization. Additionally, when the physical attributes of the model are used in the parametrization, it becomes easier for the human-machine interaction and to interpret the results of the optimization. The referenced works [98, 161] include using physical attributes in the parametrization and found performance increases of $\sim 2 \div 9\%$.

Domain-based Parametrizations Domain-based parametrizations [74] involves the definition of Control Volumes (CVs) on the geometry and can be placed on local areas of interest (i.e., the bulbous bow of a vessel hull [82, 260]), globally (i.e., covering the entire geometry), or in multiple parts to investigate a shape design space that covers global and local influences [156]. Figure 2.3b shows a hull with CVs defined over both global and local regions.

FFD [197] is a popular and effective choice for domain-based parametrizations as demonstrated in a number of studies [36, 51, 82, 97, 156, 244, 260, 260]. During this approach, a CV is defined on a parent shape, which is characterized by a number of control points, and the shape modification rule is directly related to the displacement of one or more control points. Often control points will be grouped together in the CV using a simple function (e.g., linear, quadratic) [51] to displace a group of control points according to a single parameter. This grouping of control points reduces both the dimensionality of the parametrization and the likelihood of inducing physically implausible designs.

In general, the referenced works were able to exploit deformation methods to optimize complex shapes based on the specified KPIs, and achieve performance improvements ranging from approximately $\sim 2 \div 12\%$. However, the key drawback of this approach is the relation between deformation parameters and their physical meaning to the original shape. Consequently, without careful tuning of the parametrization scheme through human intervention, there is a tendency for domain-based parametrizations to induce physically implausible designs which is often dealt with by imposing strict constraints on the design space (i.e., making very small modifications [97, 98, 156]) or via constraints based on a physical property of the shape [98, 156]. Furthermore, each parametrization scheme is unique to the parent shape and the specific domain. Consequently, it is not possible to generalize the findings from a deformation-based optimization into physically meaningful information that can be used in other shape optimization tasks.

Ranges

The parametrization used in shape optimization must meet several, sometimes conflicting, requirements. For instance, it should be sufficiently informative to predict geometry KPIs, be homomorphic, and avoid containing redundant information. The parameter ranges (in coordination with the parametrization dimensionality) define the shape design space around the parent design that the optimizer can explore to find the best candidate. Making the ranges suitably large is important to ensure an optimal candidate in the shape design space. However, too large ranges combined with too many features in the parametrization

can suffer from the curse of dimensionality [8] and prevent the optimizer from finding the best design in an adequate time.

For what concerns the ranges in the referenced works, exploring airfoil designs in 2-dimensions makes this task simple: as the parameter ranges are bound within the XZ plane, the design space is bound within the interval $[0, 1]$ along both axes (i.e., $[0, 1] \times [0, 1]$) [218].

For hulls, the ranges are more complex because the shapes are modeled in 3 dimensions instead of 2, and constrained more rigorously to reduce the likelihood of inducing infeasible designs (see Section 2.2.3). Additionally, because it is easy to induce infeasible designs, the parameters ranges are kept deliberately small in practice. This restricts the admissible space around a known feasible design and reduces the likelihood of the DDS inducing the optimizer into physically implausible candidate shapes.

KPIs

In most cases where the shape optimization aimed to minimize the energy requirements, the KPIs were obtained directly from the CFD model [19, 36, 84, 98, 158] (i.e., lift, drag, or both) or through coefficients derived from the output of the CFD model such as the Coefficients of lift and drag (C_L and C_D respectively) [145, 151, 172, 185, 218]. Other KPIs were also demonstrated for investigations for priorities outwith minimizing the energy requirements. The heat flux (Q) was minimized on the jet thermal protection system [172] to improve safety and performance. The stall angle of an airfoil was maximized in [185] to improve performance. The coefficient of discharge (C_{DC}) was optimized for the exhaust nozzle.

It is also worth mentioning that KPIs are often solved for different operating conditions, e.g., velocity [156], Mach number[2] (Ma) [218], or Froude Number[3] (Fr) [36, 98] according to the design requirements of the specific application.

Summary

Based on the review performed in this section we reported in Table 2.1 the most important works which deal with Step (1) considering the main different critical and fundamental aspect that raised during the review process

- Parent Geometry: the shape subject that has been optimized;
- Param.: the parametrization method leveraged;
- Param. Dim.: the dimensionality of the parametrization;
- Rule: the implementation of the parametrization;
- Ranges: the size of the shape design space;
- KPIs: the KPIs that that works decided to optimize.

[2]The Mach Number is the ratio between the flow velocity and the speed of sound.

[3]The Froude Number is the ratio between the flow inertia and the gravitational forces, and is proportional to the velocity.

Table 2.1: Most important works which deal with Step (1) considering the shape subject that has been optimized (Parent Geometry), the parametrization method leveraged (Param.), the dimensionality of the parametrization (Param. Dim.), the implementation of the parametrization (Rule), the size of the shape design space (Ranges), and the KPIs the work decided to optimize (KPIs).

Ref.	Parent Geometry	Param.	Param. Dim.	Rule	Ranges	KPIs
[151]	Airfoil	B-Spline	14	B-spline approach that allows the complete description of an airfoil using control points.	$x/c \in [0,1]$, $y/c \in [0,1]$	C_L
[98]	Hull	FFD/B-spline	5÷10	FFD & B-spline functions & 539 Control points governed by 10 parameters	Constraints on bulbous bow (width, height, and angle) and Section area curves of entrance, fore, run, and aft of vessel.	Drag
[19]	Hydrofoil	B-Splines	17	4 curves for 2d sections then scaled into 3d	Narrow bounds selected around the original design.	Drag
[172]	Airfoil (thermal protection system)	Design Parameters	5	25 RANS model simulations according to Orthogonal test design method DoE with 4 factors (5 levels per factor) and operating pressure.	$L/D \in [0.5, 1.5]$ $d/D \in [0.0625, 0.3125]$ $PR \in [0.2, 0.6]$ $t/D \in [0.016, 0.024]$	C_D & Q
[36]	Hull	FFD	7	7 parameters depending on how deformed the sections are $[0-1]$ with a design space characterized by 10^7 possible configurations.	FFD parameters were bound within $[-1, 1]$.	Drag
[218]	Airfoil	Design Parameters	5	Five-order CST (class/shape function transformation) to govern 39 design parameters.	$x \in [0,1]$ $y \in [-0.1, 0.2]$	C_D at Ma = $[0.765, 0.775, 0.785, 0.795, 0.805]$
[156]	Hull	FFD	5	FFD with 5 parameters functionally linked to groups of control points.	Geometric constrains so maximum variation in displacement or surface area is 1%	Drag at Fr = $[0.2, 0.26]$

[84]	Airfoil	Design Parameters	17	2d section of wing plotted in x/c and y/c.	Design space was defined by expanding coefficients by 1.5 times and narrowing by half.	Drag
[145]	Airfoil	Design Parameters	7	C-grid mesh deformed using a thin plate splines based radial basis function interpolation around the airfoil according to Hicks–Henne parameters.	All parameters bound in $[0, 1]$.	C_L & C_D
[158]	Hull	Design Parameters	6	Parametric model of a catamaran	Designs vary the length and area of the vessel.	Drag
[185]	Airfoil	PARSEC-11	6	8 Parameters linked to physical attributes and B-Splines.	Parameters bounded to ensure a typical shape is retrieved.	Stall Angle
[140]	Hull	FFD coupled with splines	5	3 FFD parameters & 2 splines parameters.	Ranges bound within ±15% of ship waterline length.	Drag
[239]	Hull	B-Spline Curves	32	32 Control points to define the B-Splines of different hull sections.	Ranges bound within margins determined from real vessels.	Drag
[143]	Hull	Design Parameters	5	Maximal radius, Coefficient of minimal radius of the rear body, parallel body length, tail fat index, and tail smoothing index.	Ranges bound within small margin determined by experimental results.	Drag
[242]	Hull	Design Parameters	6	3 parameters for longitudinal, transverse, and length ratio dimensions of hull, and 3 parameters based on fusion factors to apply deformations to the initial cross-section.	Ranges bound within $[0, 1]$.	Drag
[97]	Airfoil	FFD	10	5 FFD parameters on top and bottom sides of airfoil.	Ranges bound within ±10% of airfoil chord length.	Lift to Drag ratio.
[258]	Underwater Glider	NURBS	12	Thickness factor and Rotation angle in z-direction for 6 airfoil sections.	Ranges fixed empirically.	Lift to Drag ratio.

Summarizing Table 2.1, the critical aspect of Step (1) is to create a suitably large shape design space (in terms of dimensionality of the parametrization and the parameter ranges) so that an optimal design exists while ensuring the relationship between the parameters and KPIs can be easily captured in the ensuing steps. It is worth mentioning that while the shape design space is unique for each of the investigations, it is still meaningful to look at the state-of-the-art approaches to constructing the shape space for each parent geometry. For air/hydrofoils the referenced works usually relied on parametrizations with a dimensionality of 5÷17 [19, 172], while for hulls there were usually 5÷7 [36] parameters.

2.2.2 Step (2) Sampling, data generation, and DDS construction

Step (2) deals with three aspects: sampling techniques for the candidate geometries (Section 2.2.2), data generation with CFD (Section 2.2.2), and CFD DDS (Section 2.2.2) for the accurate estimation of the desired KPIs based on the shape parameters defined in Step (1). Section 2.2.2 will summarize the main work in the literature according to the most critical aspects identified during the review of the three main aspects related to Step (2).

Sampling Techniques

In the referenced works, there are two favored approaches to sampling. The first one is Latin Hypercube Sampling (LHS) [9, 225] where a multidimensional space is defined either randomly or structured so that the samples are distributed through all of the dimensions. The second one, which is the Full Factorial (FF) Design of Experiments (DoE) [12], involves constructing the full distribution of every possible factorial and then drawing (randomly) from that space. The LHS approach have been previously used in [19, 36, 84, 98, 156, 158, 185] and is overwhelmingly the preferred choice to fill vast multidimensional design spaces. It is also worth mentioning that Orthogonal Sampling (OS), which extends the principles of the LHS DoE, is also demonstrated in the literature [172]. The key difference between LHS and OS lies in the fact that OS specifically aims to uniformly distribute samples across the dimensions of the design space, making it a more structured approach compared to LHS and is reported to require less samples than LHS to well-represent the design space [172]. On the other hand, the FF DoE approach is also an effective choice [39]. The difference between LHS/OS and FF DoE is important because the LHS/OS involves drawing the samples from a distribution which is a subspace of the one we use in the case of FF DoE. In fact, no matter how many samples we draw to form the LHS/OS subspace, this distribution is fixed, whereas we can keep re-sampling from the FF approach without drawing from a separate distribution. Beyond LHS, OS, and FF DoE, it is worth mentioning that other sampling techniques exist and are demonstrated in the literature. In [140] authors employ the Sobol sampling method which is a method designed to space out samples more evenly than drawing from a purely random distribution. Additionally, in [218], authors leverage Gibbs sampling method based on Markov Chain Monte Carlo methods.

Apart from the sampling technique, it is also worth considering the proportion of the design space that is sampled to build the DDS. Obviously, this problem is dependent on the task at hand and depends not only on the dimensionality of the design space but also on the sensitivity of the model to changes in the design parameters (i.e., the complexity of the underlying phenomenon). Therefore, from the referenced works, we can only deduce heuristics in relation to the dimensionality of the problem and the task at hand. The best

DDS (i.e., that could approximate the output of the HF model with an error of just a few %) have a ratio ρ of

$$\rho = \frac{\text{number of samples}}{\text{dimensionality of the parametrization}} \tag{2.8}$$

of minimum 65 [19] and more typically 300÷500 [158]. Furthermore, in [36] the authors show the effect of the number of samples on the accuracy of the DDS. They found that when using a ρ ratio of between 4÷8 the accuracy of the DDS was relatively stable. However, the DDS accuracy was significantly improved when the ρ ratio was increased to 17 which is in line with expectations.

Data Generation with CFD

Regarding the data needed to develop the DDS we need to leverage CFD to estimate the KPIs defined in Section 2.2.1, for each of the sampled geometries defined in Section 2.2.2, based on their corresponding parameters value defined in Section 2.2.1.

In general, there is a wide degree of fidelities that can be obtained by using different CFD models. On one hand, Reynolds Averaged Navier Stokes (RANS) [10, 93] models are usually considered the gold standard High-Fidelity (HF) approach and can also include unsteady/turbulent extensions, but even this method can vary in trustworthiness [212] and fidelity [129]. On the other hand, Low-Fidelity (LF) models, e.g., boundary methods [31, 83], are much less accurate but can still make useful inferences for shape optimization problems.

Aerodynamic modeling using RANS CFD was demonstrated in [84, 151, 172, 185] and an LF approach (Euler methods) in [145]. Hydrodynamic modeling using RANS CFD was demonstrated in [19, 36, 156] and LF models, e.g., Neumann–Michell (NM) [166] were applied in [156] to reduce the computational complexity of an HF CFD model, and Boundary Element Method (BEM) [31, 83] is also an effective choice of model [39] if it is accurate enough for the problem at hand. Additionally, a multi-fidelity approach to shape optimization was proposed in [218] where the authors used a limited number of HF samples (5) and 19 LF samples to fill the design space.

One of the key concerns for the CFD models regards the validation of the ground truth and, subsequently, the trustworthiness of the data. CFD models are often validated on experimental data [19] which is the preferred method for validating virtual experiments [212]. Additionally, there are methods for validating mesh convergence during independent studies to improve the reliability and trustworthiness of the CFD [150, 212].

Regarding the accuracy of CFD models for shape optimization, the referenced works employed various methods depending on the problem at hand. For the resistance prediction or airfoils and hull forms, it is clear that HF RANS models are the favored tool, and the best models out of the referenced works (i.e., those characterized by DDS that could approximate the output of the HF model with a small error 1÷5%) typically used the unsteady RANS CFD simulations with turbulence [19, 156, 158]. This is explained by the fact this model determines a more accurate representation of the underlying phenomenon to capture the relationship between shape and performance accurately.

It is also worth mentioning that it is possible to develop the database using both LF and HF data, as was demonstrated in a couple of works [84, 140], but some careful consideration is required in this case. Too little HF data leads to an LF dominant DDS characterized

by insufficient accuracy for shape optimization. Too much HF data does not allow the LF data to play its part [140]. Although, if the correct balance between HF and LF is struck the variable fidelity approach can outperform single fidelity [84]. For this reason, and according to the current literature on the topic, single-fidelity sampling is the preferred approach for DDS-based shape optimization [19, 36, 98, 140, 145, 151, 156, 158, 172, 185].

CFD DDS

For the referenced works, the problem of building a DDS based on the data generated as described in Section 2.2.2 can be considered a conventional ML multi-output regression problem [202]. In this context the inputs of the regression problem are the vector of parameters that characterize the geometries (Section 2.2.1) and the outputs are the desired KPIs to estimate (Section 2.2.1).

The ML pipeline to address a multi-output regression problem consists of three main parts

- data-cleaning and data normalization [33, 69, 82, 87, 106];
- model development [19, 36, 80, 84, 98, 140, 151, 156, 158, 172, 185, 202, 206, 218];
- model selection and error estimation [169].

In the first phase, data are, mostly manually, explored and checked for inconsistency/outliers [33, 87]. This phase is useful to check for errors in EFD measurements or failures in CFD simulations [69, 82]. Subsequently, to avoid numerical issues, data is normalized.

In the second phase, a series of algorithms, shallow and/or deep, are chosen together with the range of their hyperparameters to learn the input-output relation [80, 202, 206]. In this setting, shallow models are preferred since they are less data-hungry [96, 137] and better suited for the tabular data that mostly characterizes the problem at hand [19, 36, 84, 98, 140, 151, 156, 158, 172, 185]. When input data are more structured (e.g., the input represents the actual 2D or 3D geometries), or we have a huge amount of data, deep models are preferred [3, 248, 249].

The third and final step is the model selection and error estimation phase [169]. Model selection is devoted to the selection of the optimal algorithms and associated hyperparameter [36, 39, 98, 169] while error estimation is devoted instead to providing an estimation of the future performance of the final model learned with the best optimal algorithms and associated hyperparameter on the available data [39, 169].

In the following paragraphs, we will review the use of shallow and deep models and how model selection and error estimation are actually performed in the current literature. We will not focus on data-cleaning and data normalization since this part is heavily handcrafted and usually not detailed in the works.

Shallow Models Gaussian Processes Regression (GPR) were leveraged in [19, 36, 84, 158, 172, 185, 218, 244] and specifically the Kriging algorithm (which is a type of GPR) was seen in [84, 158, 185, 244]. For what concerns the problem at hand, this approach treats the model parameters as random variables and thus determines a probability distribution for each parameter [18]. Since GPR is non-parametric, this approach is not concerned with the ability of a single function to fit the data but instead calculates the probability distributions according to all of the functions that can fit the data [184]. This means we have to specify a prior to bound the specific functions we consider. For the case of GPR,

the prior distribution is a Gaussian [164]. The posterior distribution is then obtained by looking at the data and determining the probability distribution based on Bayes' theorem [184]. In [36, 158, 172] the authors opt to use the Radial Basis Function (RBF) with two hyperparameters [196]: λ (the length-scale) and σ^2 (the signal variance), to map the data into a higher dimensional space where a linear solution exists to the problem at hand. Kriging and GPR are interpolation algorithms based on Gaussian processes and focused on fitting a function into every point in the data. For the construction of a DDS, the fact GPR passes through every point in the data set may be a useful attribute when the learning paradigm is focused within the bounds of the data (*in re* interpolation scenario as discussed in Section 2.1). On the other hand, this characteristic may pose some problems when it comes to extrapolating outside the training data in comparison to algorithms that are designed to generalize (the extrapolation approach). In the referenced works, which mostly addressed constructing the DDS to interpolate within the bounds of the database, the accuracy of the models was often characterized by 1% of Relative Error in Percentage (REP) and $R^2 \geq 0.9$ [19, 36] which suggests this approach is well suited for interpolating within the bounds of the experiments even when dealing with very complex functions.

Kernel methods [206], were used in [98, 158, 239]. Kernel Ridge Regression (KRR) was used in [98, 239]. Additionally, Support Vector Regression (SVR) [45] was leveraged in [158]. In both KRR and SVR it is possible to exploit kernel functions to learn a model for very complex functions [72, 231] although the kernel and model hyperparameters (γ, λ) [170] must be tuned according to an appropriate MS procedure [169]. When kernel methods were used to construct a DDS for the interpolation scenario within the bounds of the database, the accuracy of the surrogates was similar to the Bayesian GPR with the scatter plots [205] showing a good agreement between the real versus predicted resistance [98, 158].

Ensemble Methods, demonstrated in [239], are tree-based algorithms that group week predictors to generate robust ensembles. Random Forest (RF) [21, 171] is based on randomly sampling a subset of the training data to build different trees and averaging the models' outputs to reduce variance and improve performance. Alternatively, XGBoost [28], in contrast to RF, builds models sequentially, where each new tree attempts to correct the errors made by the previous ones. This technique gradually leads to improved model performance.

Shallow Neural Networks (SNNs) [80] are a versatile set of algorithms [1] that leverage different optimizers, e.g., backpropagation (BP) [24], ADAM [126], Levenberg–Marquard (LM) [132], etc., to train models containing neurons layered in a structure that resembles the human brain [155]. Basically, SNNs are Kernel Methods where the kernel is not fixed *a priori* but learned from the data [5, 80]. SNNs are widely adopted for their ability to perform well even on complex functions [95] and were used in [151, 218]. The referenced works found that SNNs worked well with a significant proportion of samples [151] and were able to learn a number of complex relationships [151] but were superseded by other approaches when the number of samples was low, e.g., in [218]. Another network based algorithm, based on the SNNs, is the Extreme Learning Machine (ELM) [56] used in [39, 239]. The approach of the ELM is very similar to Kernel Methods but the kernel is generated randomly (random projection) [101]. In [39], authors found the ELM was able to construct an effective DDS to predict the hydrodynamic response of a submersible substructure in

both an interpolation case (working within the bounds of the data) with a Mean Absolute Percentage Error (MAPE) of 3.75% and an extrapolation case where the DDS was evaluated using data outside the scope of the original problem with a MAPE of 5.78%.

Deep Models Since data for shape optimization of hulls and airfoils data are quite scarce in research, DDS based on deep models are not commonly used. A Deep Neural Network (DNN) based DDS was investigated in [218] where they used a database of 28 airfoils and reserved 4 geometries for testing. The REP using the DNN was 4.38% which was worse than using kriging (2.39%) for the DDS designed to interpolate within the boundaries of the database. However, authors of [208] demonstrated a DNN trained on the NACA airfoil dataset to predict the viscous flow field around the airfoil design. Results indicated that a highly accurate prediction can be obtained when sufficient data are available. Additionally, since the DNN is trained to predict a series of functions, it is invariant to the input space. Thanks to this particular property, authors show that the DNN can work with both low and high dimensional shape parametrizations without the need for retraining.

Model Selection and Error Estimation One of the main lacks in current research is that the model selection and error estimation phases are often undocumented or incomplete in the referenced works. This lack has been raised before in research [39]. On occasion, authors do not report the accuracy metrics on the error estimation which makes it difficult to evaluate the DDS [151]. On the other hand, some of the referenced work [98, 156, 158] presented scatter plots [205] to illustrate the performance of the DDS models instead of conventional accuracy metrics. In the referenced works, different statistical approaches were used for the EE: a Monte Carlo (MC) simulation was used in [98], Leave-One-Out (LOO) repetition was used in [36], and K-Fold Cross validation (KCV) was used in [36, 39].

Summary
Based on the review performed in Sections 2.2.2, 2.2.2, and 2.2.2 we reported in Table 2.2 the most important works which deal with Step (2) considering the main different critical and fundamental aspect that raised during the review process

- Parent Geometry: the shape subject that has been optimized;
- Sampling: the strategy exploited and the number of samples drawn from the design space;
- CFD: the method(s) used for data generation;
- DDS: the algorithm(s) exploited to build the DDS(s);
- KPI: the KPI(s) informed by the CFD and estimated by the DDS(s);
- Performance: the performance of the DDS(s).

Table 2.2: Most important works which deal with Step (2) considering the shape subject that has been optimized (Parent Geometry), the sampling strategy exploited and the number of samples sampled (Sampling), the CFD method(s) used for data generation (CFD), the algorithm(s) exploited to build the DDS(s) (DDS), the KPI(s) informed by the CFD and estimated by the DDS(s) (KPI), and the performance of the DDS(s) (Performance).

Ref.	Parent Geometry	Sampling	CFD	DDS	KPI	Performance
[151]	Airfoil	320 samples drawn from 8 generations of optimization	Initial population of points evaluated using RANS Solver (500 x 150 mesh) Spalart-Allmaras turbulence for an SC1095 helicopter blade.	SNN	Cl	–
[98]	Hull	200 samples drawn from LHS DoE	CFD simulations and Steady Ship Flow solver based on Neumann-Michell validated against other literature.	RBF Interpolation and Multi- quadratic kernel method	Drag	Cross-validated scatter plot shows agreement between CFD model and DDS.
[19]	Hydrofoil	1100 samples made up of 1000 LF and 100 HF simulations drawn from LHS DoE	URANS CFD performed in OpenFOAM and validated on experimental data. LF coarse grid simulations (0.5M cells) and HF fine grid simulations (3.5M cells).	GPR Kriging model with Matern 5/2 kernel.	Drag	DDS error of approximately 1% from HF model.
[172]	Airfoil	25 samples drawn from OS DoE	2d RANS CFD model in ANSYS Fluent with k-w SST turbulence and ICEM mesh with 129÷143 k cells.	DDS with Quadratic Response Surface Method developed in Isight 5.5	Cd & Q	R2 = 0.964 for Cd and 0.965 for Q.
[36]	Hull	30÷90	RANS CFD in OpenFOAM with a mesh containing 2.5×10^6 elements. Mesh validated against workshop averages.	GPR Response Surface Method.	Drag	LOO resampling R2=0.947÷0.973; RMSE=0.09÷0.05 for DDS.
[218]	Airfoil	5 samples for HF approach and 19 for LF drawn out of design space sampled with Gibbs sampling method.	CFD RANS model with 22m cells for HF approach and k-w SST turbulence for LF. Models validated against experimental data with 0.88÷1.89 % error between physical and virtual.	SNN (BP), Kriging, DBN, and Multi-Fidelity DBN	Cd at Ma = [0.765, 0.775, 0.785, 0.795, 0.805]	DDS Errors: BP 5.91% Kriging 4.71% DBN 4.38% Multi-Fidelity DBN 2.39%.
[156]	Hull	40 samples chosen from LHS DoE	CFD RANS and NM. Models validated against experimental results and maximum error is 2.45%.	RBF Interpolation, SVR, and MARS	Drag at Fr= [0.2, 0.26]	LOO cross-validation showed good agreement between experiment and DDS prediction.

[84]	Airfoil	2105 samples across 3 levels of fidelity using LHS DoE	HF model with 131k cells; MF model with 33k cells; and LF model with 8k cells.	Multi-heriarchical Kriging (MHK) with varied numbers of samples for each fidelity model.	Drag	Error of 0.008% at 46% reduction in CD; MHK DDS outperforms regular kriging.
[145]	Airfoil	544 samples ($N0 + k\Delta N$ training samples, for $N0 = 64$ and $\Delta N = 16$, k up to 30)	Euler equations using NUWTUN finite volume solver for RAE2822 wing solved for steady-state convergence using implicit time scheme.	DNN	Cl & Cd	–
[158]	Hull	2000 samples drawn from LHS DoE in 6 dimensions.	CFD using FreSCo+ solver with HF RANSE model with 2.1m cells used to develop LF in-house Rankine-panel method for reduced computational time. Final models validated on HF RANS models and sit within ±2% error bounds.	GPR, SVM, MARS	Drag	–
[185]	Airfoil	60÷80 drawn from LHS DoE	CFD RANS model in OpenFoam of NACA0012 wing with BlockMesh utility and 387k cells. Model validated against literature (LES and EFD) and full grid and time independence studies conducted to validate spatial and temporal resolutions.	Kriging	Stall Angle	RNMSE: 10% for DDS to predict objective and > 5% for constraint.
[140]	Hull	20÷50 drawn with Sobol sampling	CFD RANS model in (naoe-FOAM)/NMShip-SJTU with a mesh using 3 coarseness levels. Grid independence study and experimental validation reported to validate CFD models.	(Co-)Kriging	Drag	DDS RMSE: 0.7÷0.45 using 20÷50 samples. Results show mixed fidelity (HF + LF) DDS did not always outperform single-fidelity model.
[239]	Hull	FF DoE	CFD RANS model in StarCCM+ with a grid quality assessment. Validation of CFD with experimental results is reported.	RF, XGBoost, KRR, ELM	Drag at both high and low speeds.	DDS MAE = 0.9 ÷ 1.8[N] based on increasingly complex LOO extrapolation scenarios.
[143]	Hull	Optimal LHS DoE	CFD RANS model in Ansys Fluent with a grid quality assessment. Validation of CFD with experimental results is reported.	RBF	Drag and Energy Consumption	DDS R2 = 0.99 for Drag and 0.97 for Energy Consumption reported on 10 samples.

[242]	Hull	60 samples drawn from LHS DoE for 2 scenarios (120 total samples)	CFD RANS model in StarCCM+. Grid independence study with 3 levels of mesh coarsening reported.	4^{th}-order Response Surface Method	Drag at surface and underwater at $0.42m$	DDS R2 = 0.97 and 0.95 respectively.
[97]	Airfoil	500 samples drawn from LHS DoE	CFD RANS model in StarCCM+. Grid independence study with 3 levels of mesh coarsening reported for 3 different KPIs. Final CFD model is also validated with experimental results.	ANN	Lift to Drag ratio and static-stability height	DDS MSE = 0.05 averaged over the 2 outputs.
[258]	Underwater Glider	70 samples drawn from LHS DoE	CFD RANS model in Fine/Marin. Grid independence study with 2 levels of coarsening is reported.	Kriging	Lift to Drag ratio	–

2.2.3 Step (3) Shape optimization

Step (3) deals with mainly two aspects: the definition of the optimization problem (as previously described in Section 2.1) to define objectives and constraints (Section 2.2.3) and then the choice of the optimizer (Section 2.2.3). Section 2.2.3 will summarize the main work in the literature according to the most critical aspects identified during the review of the two main aspects related to Step (3).

Objectives and Constraints

The optimality of particular hull or airfoil varies greatly depending on the specific application and is assessed according to a number of different KPI(s). For the reasons described in Chapter 1, shape optimization is usually concerned with minimizing the energy requirements of a candidate shape [19, 36, 84, 98, 145, 151, 158, 172, 185, 218]. Typically, the design of the most efficient hull or airfoil is aimed at achieving minimal energy usage (i.e., efficient fuel usage) which is particularly relevant given current global considerations [193, 254]. Therefore, when it comes to the objectives of shape optimization of hulls and airfoils we are mostly concerned with minimizing the drag of a hull [25, 29, 36, 98, 128, 140, 156, 158, 216, 244, 260] or the lift and drag of an airfoil [58, 75, 84, 88, 90, 129, 145, 151, 172, 185, 207, 218, 261]. However, in addition to energy efficiency, various studies consider optimality through the lens of maximizing safety [172] and performance [172, 185].

Considering the constraints applied to CFD DDS-based shape optimization of hulls and airfoils, researchers and practitioners often prefer to bound the optimization problem with simple box constraints [36, 84, 140, 145, 151, 158, 172, 185, 218]. This is certainly a feasible approach when the parameter ranges are small [36, 140, 172]. However, this is often not possible when the shape design space is quite large and filled with unfeasible or inadmissible shapes, because of the approximations during the parametrization (see Section 2.2.1) [19, 36, 58, 84, 90, 98, 140, 145, 151, 172, 185, 207, 218, 261]. Therefore, additional constraints are imposed based on the physical properties of the parent geometry (e.g., thickness) or problem under exam (e.g., Angle of Attack - AoA ans static stability) [19, 84, 97, 98, 151, 156, 158, 185, 218]. Imposing constraints based on the physical properties of the shape was shown to be particularly important when dealing with parametrizations based on non-physical descriptors (e.g., FFD) [98, 156].

Optimizers and Performance

When it comes to the choice of optimizer the state-of-the-art approaches for CFD DDS-based shape optimization belong to, broadly speaking, three main categories [183, 191, 226]: i) Gradient-based Algorithms [191], ii) Swarm Intelligence Algorithms [183], iii) Evolutionary Algorithms [226].

Nevertheless, the Shape Optimization of Hulls and Airfoils problem mainly results in a mixed-integer, non-convex objective, and non-linearly constrained optimization problems [19, 36, 58, 84, 90, 98, 140, 145, 151, 156, 158, 172, 185, 207, 218, 261] that reduces the choices for optimizer selection [183, 191, 226].

Regarding Evolutionary Algorithms [226], Genetic Algorithms (GAs) [140] and variants [36, 151, 156, 158, 172, 239], are far and beyond the most popular algorithm in the literature for solving hull and airfoil shape optimization problems. Investigations utilizing GAs show they are particularly effective in mixed-integer, non-convex, and non-linearly constrained problems, common in hull and airfoil optimization [36, 151, 156, 158, 172]. The

core strength of GAs lies in their evolutionary-inspired mechanism to explore a broad spectrum of possible candidate designs. This approach iteratively generates new candidates by combining characteristics from existing designs in the population [226]. The effectiveness of this strategy to find optimal designs is shown in the referenced works leveraging GAs, which found significant performance improvements of $7 \div 10\%$ [19, 36, 156]. However, while evolutionary strategies are proven to be effective, this success comes with a price. In fact, new candidate designs are introduced within the shape optimization by means of combining traits from existing designs in the population. This strategy leads to a generating a significant proportion of infeasible and sub-optimal designs due to the complex relationship between the shape design parameters and the candidate's performance. For this reason, GAs can be computationally demanding and may require significant time to converge, especially in complex, multi-dimensional search spaces such as with the problem at hand.

Regarding Swarm Intelligence Algorithms [183], such as Particle Swarm [218] and Artificial Bee Colony [98] optimization, were also demonstrated to be a popular and effective choice for solving hull and airfoil shape optimization problems. These algorithms also excel in exploring complex search spaces, but in a fundamentally different way to evolutionary approaches. Rather than combining different traits of candidates to form new designs, Swarm Intelligence Algorithms leverage the collective behavior of a group of individuals or particles. In the case of Particle Swarm Optimization, each particle adjusts its position in the search space based on its own experience and the best experiences of its neighbors, mimicking the behavior of a flock of birds [183]. On the other hand, Artificial Bee Colony Optimization, is inspired by the foraging behavior of honey bees, where a number of potential solutions are explored and the information is shared within the optimization [220]. The advantage of Swarm Intelligence Algorithms is the ability to quickly converge to promising regions in the search space due to the collaboration among individuals or particles. The referenced works found performance improvements on $10 \div 15\%$ [98]. However, Swarm Intelligence Algorithms have been noted to struggle in cases where the search space is highly non-convex or contains multiple local optima [79].

Finally, regarding Gradient-based optimizers [19, 84, 145, 185], it is worth mentioning that these algorithms are typically preferred for learning the weights of ML algorithms rather than solving hull and airfoil shape optimization problems. However, in the referenced works, they were shown to be an effective choice for some problems [19, 84, 145], leading to performance increases of $7 \div 50$. Although, it is fair to say that the shape optimization problems considered in this review are not convex, so it is worth mentioning that this approach relies on relaxing the shape optimization problem, either globally or in iterative local stages, into a convex formulation [191]. Global relaxation is usually not preferred due to the fact this approximation leads to a severe penalty on the accuracy of solving the problem. Therefore, local-convexity is assumed and the optimization is repeated using a multi-start strategy to find a global optimal [185]. Although, it is worth mentioning that dealing with the local-convexity approximation with multi-start may lead to significant computational overhead due to performing a large number of simulations.

It is worth mentioning that stochastic optimizers, e.g., Adaptive Simulated Annealing (ASA) [143] and Interval Optimization [242], are also found in the relevant literature. These algorithms leverage randomness to explore the search space effectively, making

them particularly useful for complex, non-convex, and multi-modal optimization. In the referenced work, to improve computational efficiency, the ASA approach is coupled with a gradient-based approach, the modified method of feasible direction (MMFD), to improve computational efficiency [143]. The interval optimization approach is based on taking into account a range of accuracy of the DDS. It was shown under certain conditions (e.g., a small interval [95% ÷ 105%]) to outperform GA-based techniques; however, larger intervals of ±10% and ±15% did not outperform GA.

Regardless of the choice of optimizer, not all of the possible solutions found during Step (3) are actually meaningful. In fact, we are only interested in the solutions that are not dominated by any other one according to at least one objective of shape optimization problem [138, 151], which are identified via the Skyline operator [138]. Therefore, the meaningful output of Step (3) is actually a Pareto Front of candidate designs that represent all feasible solutions to the optimization depending on the priority of different objectives [62, 151]. In one sense, all of the solutions on the Pareto front are actually optimal according to a specific trade-off between objectives. In rare cases, often with few objectives, it may be that one solution far outperforms the rest, i.e., the Pareto frontier is formed of only one feasible solution, namely a global optimal solution. However, it is more common that the design preferences of human experts guide a selection of one or more candidate designs from the Pareto front to take into Step (4) depending on the specific application [156].

Summary

Based on the review performed in Sections 2.2.3 and 2.2.3 we reported in Table 2.3 the most important works which deal with Step (3) considering the main different critical and fundamental aspect that raised during the review process

- Parent Geometry: the shape subject that has been optimized;
- Objectives: the objective(s) that have been selected;
- Constraints: the constraint(s) that have been implemented;
- Optimizer: the optimization algorithm(s) that have been exploited;
- Performance: the achieved results in terms of improvement of the KPI(s)

Table 2.3: Most important works which deal with Step (3) considering the shape subject that has been optimized (Parent Geometry), the objective(s) that have been selected (Objectives), the constraint(s) that have been implemented (Constraints), the optimization algorithm(s) that have been exploited (Optimizer), and the achieved results in terms of improvement of the KPI(s) (Performance).

Ref.	ParentGeometry	Objectives	Constraints	Optimizer	Performance
[151]	Airfoil	Minimize C_L in two conditions with different free-stream velocities and angles of attack.	Geometrical constraint to ensure thickness > 9.5%.	GA variant	No improvement found from baseline geometries after 20 generations.
[98]	Hull	Minimize total Drag over 3 Froude Numbers.	DDS developed for displacement constraint.	Artificial Bee Colony	Optimizer found a 10 ÷ 15% reduction in Drag for low and high speeds.
[19]	Hydrofoil	Minimize ratio of C_D/C_L to the C_D/C_L ratio of a benchmark solution.	Constrained $0.09 \leq C_L \leq 0.11$ and geometric constraint on thickness.	Gradient Descent	Improved performance by 7%
[172]	Airfoil	Minimize the C_D and the Heat flux (Q).	Box constraints on parameter ranges.	GA variant	Optimal designs found for single- and multi-objective problems.
[36]	Hull	Minimize the Drag.	Box constraints on parameter ranges.	GA variant	Optimal candidate found by optimizer and DDS reduced Drag by 3N (8%).
[218]	Airfoil	Minimize the Drag at 3 Mach numbers.	Geometric constraints on AoA and wing thickness at different points.	Particle Swarm	Optimal candidate found that outperformed the baseline geometry.
[156]	Hull	Minimize the Drag at two Froude numbers.	Geometric constraints fix main dimensions of the hull. Additionally, maximum variation for displacement and surface area is fixed at ±1%.	GA variant	3 Pareto-optimal candidates selected with 5 ÷ 10% reduction in drag according to the DDS.
[84]	Airfoil	Minimize C_L	$C_D \geq C_L$ of baseline geometry. Thickness constraints over the width of the design.	Gradient-based	Cd reduced between 23 ÷ 25% according to DDS models developed with varying fidelity data.
[145]	Airfoil	Minimize C_D	C_L is kept constant according to a benchmark design (around 0.9). Strict penalty on deviating C_L constrains the shape design space around the reference airfoil.	Gradient-based	Optimal candidate found with very similar C_L (0.88) and 50% reduction in C_D.

[158]	Hull	Minimize Drag at 5 speeds.	Constant displacement.	GA variant	At design speed of 23 knots DDS models predicted reduction in drag of 1 ÷ 1.5%.
[185]	Airfoil	Minimize dynamic stall.	Constrained the dynamic stall point of the airfoil shape (informed by a DDS).	Gradient-based	Optimized airfoil increased stall angle by 3°
[140]	Hull	Minimize Drag	Box constraints on parameter ranges.	GA	Drag reduced by 4N with single-fidelity DDS and 2N with mixed-fidelity DDS.
[239]	Hull	Minimize weighted sum of Drag at high and low speeds	Box constraints on parameter ranges.	GA variant	Significant Drag reductions found based on the extent of variation from the baseline design.
[143]	Hull	Minimize weighted sum of Drag and Energy Consumption	Box constraints on parameter ranges.	ASA+MMFD	Drag reduced by 9 ÷ 11% and Energy Consumption by 3 ÷ 4%.
[242]	Hull	Minimize Drag	Box constraints on parameter ranges.	GA and Interval Optimization.	Drag reduced by 4% and 6% with GA and Interval Optimization respectively.
[97]	Airfoil	Maximize Lift over Drag	Box constraints on parameter ranges.	GA	Lift to Drag ratio decreased compared to baseline design due to static stability constraint.
[258]	UnderwaterGlider	Maximize Lift over Drag	Box constraints on parameter ranges.	GA	Lift to Drag ratio increased by 18.98% compared to baseline design.

2.2.4 Step (4) Physical plausibility and feedback

Step (4) deals with the validation of the final output of the optimizer in terms of physical plausibility and provides feedback to the preceding steps (e.g., to decide if the candidate design meets the specified criteria). Note that, in practical applications, this is likely the most important step of the four, but it also involves more human feedback, making it the most challenging to automate [19, 36, 84, 98, 140, 145, 156, 158, 172, 185, 218].

The validation in terms of physical plausibility is an important step in the pipeline since, using particular parametrization, KPIs computation, constraints, and especially DDS we are making an approximation of the physics (CFD) [19, 36, 84, 98, 116, 140, 145, 156, 158, 172, 185, 218]. Verifying that all these approximations did not induce numerical artifacts is vital [19, 36, 84, 98, 140, 145, 156, 158, 172, 185, 218].

If the DDS performance is a numerical artifact because a vast number of the induced geometries are not physically plausible [27, 41, 47, 116], feedback to Step Step (1) might suggest augmenting the design space (reducing the dimensionality) to reduce these artifacts.

If the predictions from the DDS are significantly different from the CFD, the optimizer may be induced into false minima due to the imprecision of the DDS. Therefore, feedback to Step Step (2) may infer that more data is required to improve the reliability of the DDS to reduce the likelihood of numerical artifacts [27, 41, 47, 116].

Feedback to Step Step (3) [19, 36, 70, 84, 98, 98, 129, 140, 145, 151, 156, 158, 172, 185, 218] may infer that the dimensionality of the parametrization is too small, and more free variables are required to find an optimal design. Similarly, the box constraints on the parametrization may be improved to facilitate exploring a larger number of candidate designs. Another feedback could be to change the optimizer hyperparameters if the solutions found are sub-optimal.

Based on the review performed in this section, we reported in Table 2.4 the most important works which deal with Step (4) considering

- Parent Geometry: the shape subject that has been optimized;
- Physical Plausibility: the adopted method to check for the physical plausibility of the generated geometry.

Table 2.4: Most important works which deal with Step (4) considering the shape subject that has been optimized (Parent Geometry) and the adopted method to check for the physical plausibility of the generated geometry (Physical Plausibility).

Ref.	**Parent Geometry**	**Physical Plausibility**
[98]	Hull	Validation with numerical (CFD) and experimental (EFD) methods showed candidate shape increased drag by 6% at low speeds and decreased drag between 6 ÷ 13% at medium to high speeds.
[19]	Hydrofoil	Validation with LF and HF solvers informed the candidate shape performance.
[172]	Airfoil	Validation of the candidate design with CFD showed the C_D predicted by the optimizer was 7% lower than the HF model and 1% lower for Q.
[36]	Hull	Validation with CFD showed the optimal candidate had a Drag reduction of 0.5N (1%).
[218]	Airfoil	Validation showed good agreement between CFD and DDS predicted C_D for the optimal candidate.
[156]	Hull	Results validated with NM plus CFD method and pressure distribution around the candidates compared to original for plausibility.
[84]	Airfoil	Validation with CFD showed good agreement with DDS. Error of DDS prediction on optimal candidate was 0.2 ÷ 0.5%.
[145]	Airfoil	Validation shows physical plausibility and explains large reduction in C_D by reducing in upper shock on upper surface.
[158]	Hull	Final design validated with HF RANS model showed 1 design failed physical plausibility (drag actually increased) and the other passed (error between DDS and CFD of 0.4%).
[185]	Airfoil	Final model validated against HF CFD showed physical plausibility.
[140]	Hull	Validation with HF CFD showed actual reduction of 1N for single-fidelity candidate and agreement with mixed-fidelity prediction.
[239]	Hull	Validation with HF CFD showed strong physical plausibility between DDS and real Drag when close to known designs. As the hull design deviated from known examples physical plausibility decreased.
[97]	Airfoil	Validation with HF CFD showed 3.6% deviation between DDS and real Lift to Drag ratio and 0.0$ error for stability prediction.
[258]	Underwater Glider	Validation with HF CFD affirmed that the actual Lift to Drag ratio outperformed the baseline design.

2.3 Open problems and future perspectives

Following the review carried out in Section 2.2, this section elaborates on the open problems of future perspectives of the shape optimization of hulls and airfoils leveraging CFD and DDSs.

Regarding Step (1), it is clear from the review that each work develops a unique parametrization schema (e.g., particularly when using domain-based parametrizations) requiring a new database to be collected/generated from EFD and CFD simulations. While this problem is overcome by standardized parametrizations (e.g., Hicks-Henne and PARSEC-11), there is a lack of standardized schemes for domain-based parametrizations. The latter represents an important open challenge (especially for hull shape optimization). Establishing a new database for every parametrization scheme presents a considerable challenge, as it greatly increases computational costs and necessitates human oversight. This issue is further complicated when devising a dedicated parametrization scheme alongside CFD models. Additionally, such a high degree of customization poses a significant hurdle for the community in terms of developing shared datasets. These shared resources are crucial for continually expanding toward larger and more diverse design spaces. Decoupling parametrization from parameters leveraged by DDSs to make prediction [69, 134, 198, 208, 239] may help the community in reusing the same DDSs in optimization with different parametrization schema.

Regarding Step (2), the DDS models reviewed in this work show the learning paradigm is almost always focused on interpolating rather than extrapolating beyond the bounds of the data used to construct the models. This means that little effort has been made to assess the performance of DDS models with a specific setting or application. In fact, there are a number of interesting extrapolation scenarios (unseen families of geometries, unseen geometries, or unseen speeds) that are useful to develop models that work well when applied outside the boundaries of the data used to construct them [239]. Extrapolation is a particularly valuable attribute of DDS models since, in practice, shape optimization is the adaptation of an existing (not optimized) design for specific KPIs. Existing designs for hulls and airfoils are based on historical design requirements (e.g., using conventional fuels), and changing design requirements (inspired by fluctuations in fuel prices or addressing climate change) requires exploring a wider design space leveraging DDS models that remain accurate even in extrapolating conditions. Hence, during DDS-based shape optimization it is beneficial to produce a geometry for novel, yet-to-be-explored candidates, rather than confining ourselves to established (not optimized) designs.

Regarding Step (3), the current approach to optimization largely relies on well established methods for solving multi-objective non-linear and non-linearly constrained problems. Given the context of increased design uncertainty for hulls and airfoils, there is a necessity for a more comprehensive exploration of certain aspects within the domain of shape optimization. One promising avenue to achieve this is the cross-fertilization of methodologies from operations research into this field of research. State-of-the-art techniques from operations research and other related fields, such as physics-based optimization and hybrid strategies [160], offer a number of methods to enhance the current approaches to optimization in CFD-DDS based shape optimization of hulls and airfoils. Moreover, the increase in quantum computing resources presents an opportunity for enhanced optimization methods in the future. Quantum computers, which can perform

complex computations at speed, offer the potential to reduce the time required for optimization tasks [154]. The integration of these advanced computational techniques with existing CFD DDS shape optimization could lead to significant breakthroughs in the field.

Finally, regarding Step (4), there are a number of instances in the referenced works where the validation of the candidate designs is not confirmed using the high-fidelity CFD. This remains an open problem in the field until this procedure is common practice and must include a comparison between the KPI predicted by the DDS and the real one informed by the CFD. Ultimately, this approach will lead to a better understanding of the performance of DDS models.

2.4 Chapter Summary

This paper reviews the current research in shape optimization of vessel hulls and airfoils, which is a critical step to ensure optimal performance and minimal environmental footprint. In fact, their design is usually an adaptation of an existing one, not optimized for specific KPIs like the drag of a hull or lift and drag of an airfoil or the result of a mix between human experience and numerical optimization approaches. Nowadays, the state-of-the-art approach for shape optimization of hulls and airfoils is based on CFD DDS models and consists of four steps. First, parametrization and parameter ranges are defined, with more or less human intervention, to build a shape design space.

Accurate estimation of KPIs, such as drag resistance for hull designs or lift and drag for airfoils, is a cornerstone of modern aerodynamic and hydrodynamic engineering. However, the direct computation of these KPIs based on shape parameters involves complex simulations using CFD, which is notably resource intensive. This high computational demand traditionally limits the feasibility of direct numerical optimization of these KPIs in real-time design workflows. To address this challenge, a pragmatic approach is typically employed, involving a multi-step process that integrates both human expertise and advanced data-driven strategies. Initially, a selective sampling of distinct and representative shapes from the design space is conducted. This selection can be based on experienced human judgment or through sophisticated algorithmic strategies that aim to cover the potential variability in the design space. These selected designs are then subjected to detailed CFD analysis to evaluate their performance against the desired KPIs. Based on the insights gained from these analyzes, a DDS model is developed. While the initial creation of the DDS is computationally demanding - requiring substantial data processing and model training - the resultant model offers a significant reduction in computational expense for subsequent predictions. This efficiency is achieved by approximating the complex relationships between shape parameters and their resultant KPIs through learned data patterns, thus circumventing the need for direct CFD simulation in the early stages of design exploration. Recent advancements in machine learning have further enhanced the capabilities of DDS models, incorporating techniques such as deep learning and reinforcement learning to refine prediction accuracy and model robustness. Studies have demonstrated that such models can effectively predict hydro/aerodynamic properties with high reliability, approaching the fidelity of direct CFD simulations under varied operational conditions. Once the DDS is established, it can then seamlessly integrate into an optimization loop. This integration facilitates efficient exploration of the design space, enabling the rapid generation and assessment of candidate geometries. The objective is to

identify designs that approximate the Pareto front, optimizing multiple conflicting KPIs simultaneously, a method that has seen significant refinement in recent years through multi-objective optimization algorithms. The final step in this process involves a rigorous validation of the proposed designs. Using CFD, the physical plausibility of each candidate geometry suggested by the DDS and the optimization framework is verified. This validation is crucial, as it ensures that the surrogate model and the optimization algorithms have not proposed non-viable or physically implausible shapes. This comprehensive evaluation not only reinforces the reliability of the DDS, but also aligns the theoretical models with practical, real-world applicability. This approach, which blends sophisticated data-driven models with traditional simulation techniques, represents a significant shift in the way hull and airfoil designs are developed. By reducing reliance on extensive CFD simulations and leveraging recent research in surrogate modeling and optimization, this methodology promises to accelerate the design cycle, reduce costs, and enhance the innovation process in fields demanding high precision and efficiency.

Based on our review, we identified several important discussion points. In general, DDS-based shape optimization is an effective strategy for the optimization of hulls and airfoils and will remain an important area of research given the current motivations towards minimizing the energy requirements of hulls and airfoils. However, a key acknowledgment is that the current approaches to parametrization (especially when using domain-based parametrizations) often lack standardization. Consequently, much of the generated data remains unusable for future work despite a large degree of similarity among many existing designs. This redundancy not only lengthens the time frame of individual projects but also restricts the broader community's capacity to build upon prior findings. Additionally, future changes to conventional design requirements opens the door for shape optimization to explore novel design concepts, and improving the extrapolation capabilities of DDS models will significantly benefit this endeavor. However, at the time of writing, these challenges have not yet been extensively investigated or overcome.

3

Physically Plausible and Computationally Aware Hull Resistance Optimization

Optimizing the hull form is crucial for enhancing vessel performance, requiring a careful exploration of design variations once an initial concept is chosen. To address this, a well-defined parameterization and search space must be established, followed by a numerical optimization process. This phase involves balancing three key requirements: improving performance indicators like resistance (and consequently emissions), ensuring physical plausibility (e.g., designs that meet stability standards), and maintaining computational efficiency. While existing methods partially address these challenges using surrogate models to optimize performance, conducting a-posteriori checks for physical plausibility, and reducing the search space a-priori, they fall short of fully integrating these aspects. In this work, we introduce a two-fold approach to address these challenges. First, we integrate a key physical plausibility criterion directly into the optimization loop: the International Maritime Organization (IMO) Intact Stability Code. For the first time, we use a surrogate model to include this constraint within the numerical optimization process. This enables the generation of designs that inherently meet stability requirements. Second, we reduce the computational demands by applying data-driven methods to limit the search space in a problem-specific manner. This targeted reduction reduces the computational load without compromising the overall optimization performance. We test our proposal by optimizing the KCS hull form. Our results demonstrate two main achievements. First, we find that current optimization pipelines fail to meet the IMO stability guidelines, whereas our method achieves compliance by design. Second, our approach reduces the computational burden by 30% without sacrificing performance, representing a significant leap forward in practical hull design optimization.

This chapter is reproduced from J. M. Walker, A. Coraddu, and L. Oneto. Computationally Aware and Physically Plausible Hull Form Optimization Incorporating the IMO Intact Stability Code. *Ocean Engineering*, 341:122607, 2025.

During the mid-stage of hull-form optimization (see 1.1), a DDS-based approach to hull optimization is preferred. This approach follows four-steps (for a detailed overview, please refer to [238]). The main idea of the four-step approach is to establish a well-defined parameterization and search space, perform a limited number of high-fidelity simulations that serve as the foundation for the DDS, perform the numerical optimization leveraging the DDS, and check the output using a high-fidelity simulation (e.g., Computational Fluid Dynamics - CFD). This output assumes the form of a series of optimal designs, the ones on the Pareto frontier, which are then validated in terms of physical plausibility providing feedback (sometimes also integrated with late-stage designs [102]) able to refine the mid-stage design [140, 239, 245, 256].

The concept of physical plausibility is crucial; however, it is not easy to formally define. For example, it is possible to check that the performance of the designs on the Pareto frontier are not actually numerical artifacts (e.g., too small resistance), induced by the use of the DSS, by performing high-fidelity simulation [239]. Some checks are more complex to perform, for example, the a-posterior verification of the performance of the geometry (e.g., seakeeping, stability, structural integrity) [103]. Humans must also perform some other checks. For example, when the parameter space is defined too large, the optimizer can induce a geometry unsuitable for a specific application (e.g., too tight or too round to transport goods) [14]. Finally, some checks are not easy to code into a mathematical framework, such as regulatory aspects [103].

The two main lacks of current approaches to mid-stage hull form design are

Lack (1) the limited ability to take into account the seakeeping-related properties of the design when focusing mainly on resistance [103];

Lack (2) the computational requirements of the tool even when a DDS is employed [54].

Lack (1) has already been addressed by incorporating the seakeeping-related properties a-posteriori [167, 255] or by including Response Amplitude Operator (RAO) to account for the vessels' seakeeping capabilities in the objective of the optimization problem [55]. Lack (2), instead, has already been addressed by applying Design Space Dimensionality Reduction (DSDR) [54] to the parameter search space to speed up the optimization (i.e., the smaller the number of parameters to be optimized, the faster the optimization is). See Section 3.1 for more details.

The problem with the current approach to Lack (1) is that it is not trivial to trade-off the current seakeeping metrics (e.g., RAO) with resistance, as there are no clear, predefined criteria to determine, a priori, what constitutes a good or bad RAO, nor how much resistance should be sacrificed for improved seakeeping performance. Instead, these trade-offs are often empirically evaluated by human experts. The problem with the current approach to addressing Lack (2) instead is that DSDR is often performed a priori to optimization. Although the DSDR approach preserves variability in the design space, it does so without incorporating the necessary knowledge to retain the best-performing designs (in terms of at least one of seakeeping or resistance) while discarding the poorly performing designs. Moreover, to the best of the authors' knowledge, no work in the literature has suggested an approach to address both lacks simultaneously.

The purpose of our work is to take a step forward in finding the solution in both Lack (1) and Lack (2) and to blend the solutions to address them simultaneously. Specifically, to ad-

dress Lack (1), we consider, for the first time in the literature, the International Maritime Organization (IMO) Intact Stability Code [113, 149, 181, 182], recently updated in 2020, directly in the hull optimization problem. This constraint on stability allows us to easily distinguish between acceptable and unacceptable geometries. However, the problem with this constraint is that it is computationally challenging to assess, so we propose, for the first time, to surrogate it to include it in the optimization loop with minimal impact on the computational requirements. In order to address Lack (2), we compared different DSDR techniques able to reduce the number of parameters to optimize, i.e., Principal Component Analysis (PCA) and Neural Networks. Finally, in order to blend these solutions toward addressing Lack (1) and Lack (2) we incorporate the stability constraints in the DSDR and show that this further improves both the quality and computational requirements of the mid-stage hull form optimization schema. In particular, we build on our previous work [239] which represents the state-of-the-art pipeline for mid-stage hull form optimization without considering the stability constraints. We test the quality of our proposal by optimizing the KCS hull form which has been well analyzed in the literature [35, 71, 156, 228]. Our results demonstrate two main achievements. First, we find that current optimization pipelines fail to meet the IMO Intact Stability Code, whereas our method achieves compliance by design. Second, our approach reduces the computational burden by 30% without sacrificing performance, representing a significant step forward in practical hull form design optimization.

The rest of this paper is as follows. Section 3.1 reviews the most relevant work to our proposal that supports the statements we made in the introduction; Section 3.2 describes the specific problem we examine in this work and the dataset we leverage to address it; Section 3.3 decribes the method we use to solve the problem using the data described in the very same section; Section 3.4 presents the results; finally, Section 3.5 concludes the work.

3.1 Related Work

This section reviews the main works in the literature related to the mid-stage hull form design. In particular, we review the main works that deal with DSDR, to reduce computational requirements of the optimization, and the ones that incorporate seakeeping into optimization as, to the best of the authors' knowledge, no work in the literature blends the two aspects.

Several authors have demonstrated how DSDR methods can be applied to reduce the computational burden of hull form optimization [54]. The most popular approaches are linear dimensionality reduction methods [60, 228, 263].

In [228], the authors considered Proper Orthogonal Decomposition (POD) basis functions for DSDR of a Free-Form Deformation design space. Two approaches were examined: first, authors performed DSDR using geometric-based criteria highlighting new problem-dependent criteria for ranking POD modes, significantly reducing the design space size. Second, the authors apply POD to Design Velocities, directly correlated to shape variations, which demonstrated reducing a 68-dimensional parametrization of the DTC hull form to 30 variables for shape reconstruction. This work showed that the POD-based approach was effective for generating hull shape variations for optimization processes.

In [60], DSDR is performed using the Karhunen-Loève Expansion [54], which is based

on eigenfunction decomposition, for assessing the design space. Authors use a quantitative measure to show the design variability in the eigenvalue space and reduce the dimensionality of a parametrizaion of the DTMB5415 from 100 design variables to 25. In fact, authors generate 40000 candidate hulls but later reduce that to 2400 feasible samples and perform the hull form optimization. This method was able to generate a reparameterization of the shape modification vector for efficient optimization. However, this reparametrization does not allow the reduced-dimensionality representation to be projected in the original design space. This presents a challenge because well-established parametric models are commonly used by practitioners to preserve specific knowledge of existing designs [200].

In [263], the authors parameterized the KCS hull form using 37 design variables controlling a T-spline geometry representation. They performed DSDR using PCA and showed that, for varying levels of data compression (e.g., preserving between 68% and 100% of the original design variance), it was possible to obtain optimized hull forms (in terms of resistance) compared to the parent design. The study demonstrated that 6-dimensional parametrizations could achieve optimized hull forms with minimal computational effort (e.g., within tens of iterations). However, lower-dimensional parametrizations were consistently outperformed according to the key performance indicators when more than hundreds of iterations were used.

Regarding the integration of seakeeping constraints into the mid-stage hull form design, it is possible to find several milestone works [55, 81, 188, 256].

In [55], authors presented the multi-objective hull form optimization of the DTMB5415. In particular, the study focused on improving hydrodynamic performance, both in terms of resistance and seakeeping, using low fidelity solvers. A seakeeping merit factor based on the vertical acceleration of the bridge (located 27[m] forward amidships and 24.75[m] above keel) at 30[kn] in head waves and is the roll angle at 18[kn] in stern long-crested wave was used as the objective function during optimization. Results showed resistance and seakeeping improvements of 6÷23% and high-fidelity simulations verified the findings.

In [256], authors performed multi-level hull-propulsor optimization integrating medium- and low-fidelity solvers. A bi-fidelity Co-Kriging surrogate model and a multi-objective evolutionary algorithm were used to optimize the lifetime fuel consumption and energy efficiency design index of the S175 containership with KP505 propeller and MAN B&W engine. In particular, the authors imposed constraints on metacentric height (GM) so that the value would be greater than a given value of $GM_0 = 0.2$[m] based on the IMO Intact Stability Code.

In [188], performed surrogate-based multi-objective optimization of a SWATH hull form, considering seakeeping performance and ship resistance. A parametric model of the SWATH ship, with varying torpedo semi-axis and strut angles, is optimized using Gaussian process surrogate models. Three fidelity levels for ship resistance are considered, ranging from wetted surface calculations to high-fidelity CFD simulations with stabilizing fins. Results showed it was possible to optimize for seakeeping, resistance, and trade-off in terms of the median comprimise of both objectives.

In [81], authors presented the multi-objective optimization method for unmanned surface vehicle hull design, focusing on resistance and seakeeping performance. The optimization considers wave resistance, vertical acceleration, and pitch amplitude, using a

parametric model to generate candidate designs. To reduce computation time, a second-order response surface method surrogated numerical simulations for the resistance and seakeeping parameters, and the optimization was performed with the Particle Swarm and Sequential Quadratic Programming algorithms. Results on a 7[m] vessel demonstrated the effectiveness of the proposed method integrating the seakeeping into the objective function.

3.2 Problem and Data Description

In this work, we propose a novel computationally aware and physically plausible hull form optimization framework to address the two main gaps of the current literature Lack (1) and Lack (2).

For Lack (1), we propose to integrate the IMO Intact Stability Code [113, 149, 181, 182] directly into the optimization problem. For Lack (2), we propose using DSDR techniques [54, 60] to reduce the number of parameters we need to optimize. Moreover, for the first time in the literature, we simultaneously address both identified lacks by blending the two solutions. Specifically, we propose incorporating the stability constraints into the DSDR and show that this further improves both the quality and computational requirements of the mid-stage hull form optimization schema. We test our approach by optimizing the KCS hull form in a state-of-the-art mid-stage hull form optimization pipeline. For this purpose, we build on our previous work [239] that leveraged the mid-stage hull form optimization pipeline without considering the stability constraints.

In this work, we optimize the KCS hull form which has a complex topology (i.e., a cargo ship with a bulbous bow).

The pipeline for Mid-Stage Hull Form Optimization (MSHFO) consists of four main steps [238]

MSHFO (1) Shape parametrization, parameter ranges, and KPIs definition;
MSHFO (2) Sampling, data generation, and DDS;
MSHFO (3) Shape optimization;
MSHFO (4) Physical plausibility and feedback;

that we will improve to address Lack (1) and Lack (2).

Regarding MSHFO (1), the KCS requires a suitable parameterization and parameter ranges. To this end, we utilize the Free-Form Deformation (FFD) method to parameterize the KCS hull. FFD is an effective way to create a diverse design space while ensuring that the designs maintain geometric continuity and feasibility [34, 60, 228]. For the FFD parametrization, we overlay a control network, characterized by control points, over the parent design. We then induce various candidate hull form designs by displacing the control points. We utilized an in-house developed FFD tool that leverages subdivision surfaces, as referenced in [34], along with an empirically defined control network. The control network consists of two overlapping subnetworks: a global network that covers the entire hull and includes 24 control points, and a local network focused on the bulbous bow comprising 12 control points. By structuring the networks hierarchically, the density of control points is increased around the bulbous bow due to its intricate geometry. Figure 3.1 illustrates the FFD control network showing the global and local control networks, which account for a total of 36 control points.

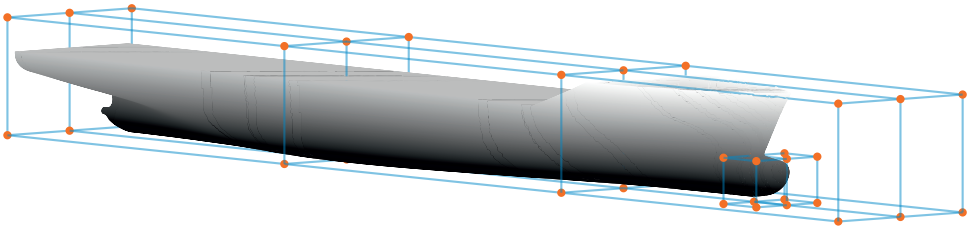


Figure 3.1: The FFD control network shows the global and local control networks, which account for a total of 36 control points.

A symmetry constraint is applied along the XZ plane to reduce complexity, bringing the number of independent control points down to 12. For the global control network, we restrict the Degrees of Freedom (DoF) of the control points to allow displacements only along the X and Y axes. However, we do not restrict the control points in the local control network. Hence, the independent control points were adjusted according to a vector $x = [x_1, \cdots, x_{29}]$, where $[x_1, \cdots, x_{16}]$ govern the displacements of the global control points (e.g., 2 DoF for each of the 8 independent global control points) and $[x_{17}, \cdots, x_{29}]$ manage the displacements of the local ones (e.g., 3 DoF for each of the 4 independent local control points). The original placement of these control points establishes the parametrization rule, $\mathtt{R}$ (which is fixed during optimization). $\mathtt{R}$ allows for a specific candidate geometry, Ω, to be created from the parent geometry, Ω^{p}, according to the displacement vector x (which varies during optimization), such that $\Omega = \mathtt{R}(\Omega^{\mathrm{p}}, x)$. It is important to note that $\mathtt{R}$ is homomorphic, as it should be according to the literature on the topic [238, 239]. For the parameter ranges, we defined the lower and upper bounds of each parameter of x in such a way as to ensure the solution is inside the range, i.e., a posteriori, we check that each solution we found was actually inside the range and none actually belonged to the borders. We set the lower (x^l) and upper (x^u) bounds of the parameters in the global control network, e.g., $x^l_{1,\cdots,16}$ and $x^u_{1,\cdots,16}$, so that they could move ±10% along the length of the vessel and up to ±20% the width of the vessel. Similarly, we set the lower and upper bounds of the parameters for the local control network, e.g., $x^l_{17,\cdots,29}$ and $x^u_{17,\cdots,29}$, so that they could move ±10% in each direction of the bulbous bow. Figure 3.2 shows the linesplans of the KCS hull and the geometries on the border of the representation space.

In MSHFO (1), apart from leveraging the symmetrical properties, we can further reduce the parameters to be optimized. This allows us to address Lack (2) and reduce the computational cost of the optimization phase. To this end, we employ DSDR algorithms such as the PCA and the Neural Networks to reduce the dimensionality of the parametrization. The first sanity check of DSDR is to ensure that the inverse operator of the PCA or the Neural Networks, which maps the reduced dimensional representation back to the original parameterization, performs well (e.g., it should reconstruct the original parameters with minimal error), see Section 3.3.3 for details. Note that this sanity check does not guarantee that the optimizer will still perform effectively (e.g., find the optimal hull in the original parameterization), as the impact of reconstruction error on the ability of the optimizer to evaluate the performance of candidate designs is unknown. A more informative check will be performed later in the process that is not standard in the literature, see

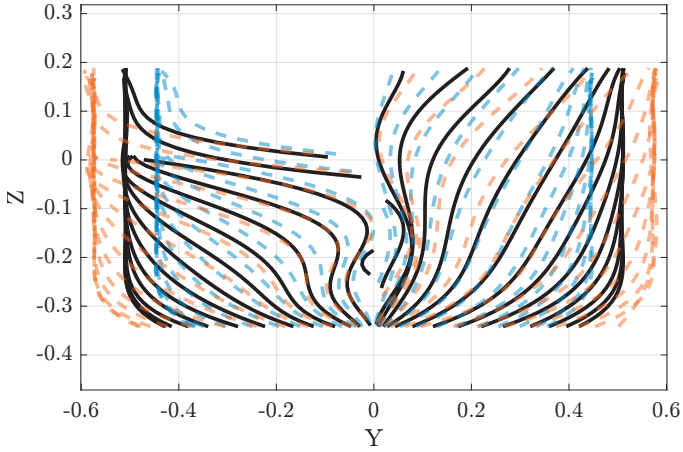


Figure 3.2: Linesplans of the KCS hull in black and the geometries represented by x^u and x^l in dashed orange and blue, respectively.

explanation of MSHFO (3).

Regarding MSHFO (2), we need to generate the data from scratch by performing a limited number of CFD simulations. Selecting the right simulation to be performed, i.e., sampling a representative selection of hulls, is a critical step in the process [238]. In this work, we opt for a Full Factorial Design of Experiments [13]. Due to computational constraints, we randomly sample 1000 geometries from the full factorial. The choice of 1000 samples, in line with the literature [53], has been made since it allows for adequate sampling of the design space defined by the parameterization, ensuring that the range of possible hull forms is sufficiently represented, covering a wide variety of designs that may meet the desired objectives and constraints. Subsequently, for each sampled hull, we performed

- a high-fidelity CFD simulation to estimated total Resistance (R_t) at three different Froude numbers (Fr) selected uniformly random in a range $0.108 \div 0.282$ [$^m/_s$];
- a panel code simulation to assess the stability performance according to the IMO Intact Stability Code.

For what concerns the high-fidelity CFD, we employ the same approach and software as in [239] with some additional fine-tuning and validation (described in Section 3.3.1) for the particular problem under exam. Due to the high computational cost of the CFD model, we performed the CFD for 218 randomly selected geometries out of 1000. The choice of 218 samples exceeds the typical number of samples used in the literature to develop a DDS [238]. A description of the dataset to learn R_t and the number of examples for each feature can be found in Table 3.1. For what concerns the IMO Intact Stability Code, we focus on a single Stability Failure Mode (SFM): the Dead Ship Condition. We leverage an in-house developed panel code, as referenced in [37, 38], to assess the Dead Ship SFM considering the International Code on Intact Stability Section 2.3 Severe wind and rolling criterion (weather criterion) [111] using the righting-arm approach [37]. The weather criterion [111] mandates that for a steady wind heeling lever (l_{w1}), angle of equilibrium (φ_0), wave action roll (φ_1), angle of heel (φ_2), and gust wind heeling lever (l_{w2}) evaluated as

Table 3.1: Dataset Desription to learn R_t (218 samples for $R_t(x, 0.108)$ and 218 samples for $R_t(x, 0.282)$).

Type	Feature	Symbol	Range
Input	Global parametrization [−]	$x_1, \cdots, x_{16}$	$\{-0.74, -0.71, \cdots, 0.74\}$
	Local parametrization [−]	$x_{17}, \cdots, x_{29}$	$\{-0.05, -0.02, \cdots, 0.05\}$
	Froude Number [−]	Fr	$[0.108 \div 0.282]$
Output	Total Resistance $[N]$	$R_t(x, \text{Fr})$	$[12.9 \div 221.2]$

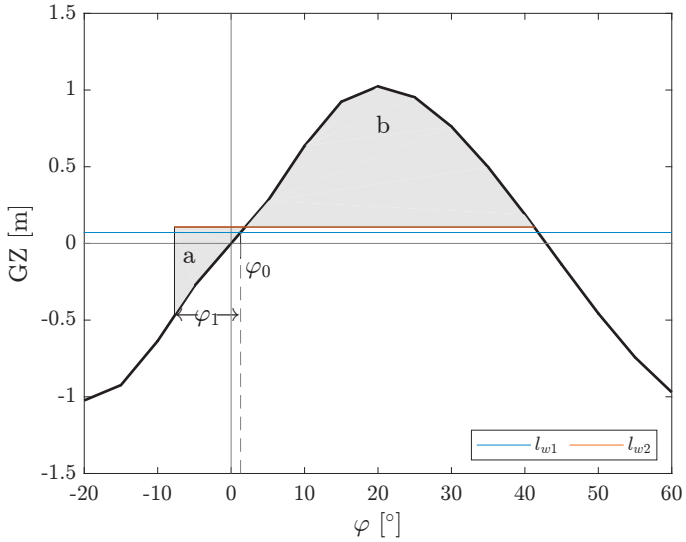


Figure 3.3: Example Righting-arm.

$^3/_2\, l_{w1}$, the area b (bounded between l_{w2} on the bottom and the GZ curve on the top) shall be equal to or greater than area a (bounded between GZ curve on the bottom l_{w2} on the top, and up to φ_1), as indicated in Figure 3.3.

We also paid attention to four additional criteria on top of the Dead Ship Condition coming from the same code section [111]

SFM (1) The Dead Ship Condition;
- SFM (1.1) The area under the GZ curve from 0° to 30°;
- SFM (1.2) Minimum GZ at 30°;
- SFM (1.3) The angle at which the max righting arm occurs;
- SFM (1.4) The minimum initial GM.

The SFM criteria for each specific geometry were determined in the following way. Initially, we computed the actual values for GZ (for 17 different transversal inclination angle φ sampled uniformly in the range $[-20^\circ \div 60^\circ]$) and GM, with the panel code because each of the criteria, i.e., SFM (1), SFM (1.1) ÷ SFM (1.4), are related to these quantities. A description of the dataset to learn GZ and GM and the number of examples for each feature can be found in Table 3.2. Subsequently, we derived the actual values for each of the SFMs,

Table 3.2: Dataset Desription to learn GZ and GM (1000 samples for GM and 17000 samples for GZ namely 1000 geometries for 17 transversal inclination angle $\varphi \in [-1.25 \div 1.25]$).

Type	Feature	Symbol	Range
Input	Global parametrization [−]	$x_1, \cdots, x_{16}$	$\{-0.74, -0.71, \cdots, 0.74\}$
	Local parametrization [−]	$x_{17}, \cdots, x_{29}$	$\{-0.05, -0.02, \cdots, 0.05\}$
	Transversal Inclination [°]	φ	$[-20 \div 60]$
Output	Righting Arm [m]	GZ(x, φ)	$[-1.5 \div 1.6]$
	Metacentric Height [m]	GM(x)	$[11.5 \div 16.8]$

i.e., the areas a and b, the area under the GZ curve from 0° to 30°, the GZ at 30°, the angle of the maximum righting arm, and GM. Finally, these values are compared against predefined thresholds from the IMO Intact Stability Code and the design is classified as either "pass" or "fail" indicating whether it meets the thresholds. In this way the SFM criteria are mapped from their actual values into a binary ("pass" 0 and "fail" 1) condition (or constraint). The last step is to condense the five binary conditions, i.e., SFM (1), SFM (1.1) ÷ SFM (1.4), into a single function IMO(x) that defines if a particular hull, generated by the parametrization x, passes all of the stability criteria

$$\text{IMO}(x) = \begin{cases} 1 & \text{if} \quad \begin{cases} a(x) \leq b(x), \\ \displaystyle\int_{0^\circ}^{30^\circ} \text{GZ}(x, \varphi)\, d\varphi \geq 0.055, \\ \text{GZ}(x, 30^\circ) \geq 0.2\text{m}, \\ \text{GZ}(x, \varphi_{\max}) \text{ occurs at } \varphi_{\max} \geq 25^\circ, \\ \text{GM}(x) \geq 0.15\text{m} \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Leveraging the datasets described in Tables 3.1 and 3.2, we build DDSs of R_t and IMO(x) and validate its performance (see Section 3.3.1).

After MSHFO (2), we can move to MSHFO (3) (see Section 3.3.2). Once the numerical optimization is performed, and the series of optimal designs, i.e., the Pareto frontier of solutions, have been obtained, we have to verify that the optimizer has performed effectively and found the optimal hulls in the original parameterization. When the optimization is performed without DSDR, we skip directly to validating the solutions in terms of physical plausibility, MSHFO (4). However, when the optimization had been performed in a reduced dimensionality representation, i.e., obtained through DSDR, we needed to assess the impact of the reconstruction error on the optimizer's ability to evaluate the performance of candidate designs. To perform this assessment, we proceed as follows

- we obtained the Pareto frontier of solutions expressed in the reduced dimensionality representation;
- we mapped the reduced dimensionality representation back to the original parameterization using the inverse operator (i.e., PCA or Neural Networks);
- we evaluated the performance (in terms of resistance and stability) of the candidate solutions using the DDS built from the datasets described in Tables 3.1 and 3.2. This

provides us with the estimated performance, including the reconstruction error.
- we check the performance without the reconstruction error. For this purpose, we estimated the performance based on the reduced dimensionality representation. Hence, we retrained the DDS using the same data as in Tables 3.1 and 3.2, but we performed DSDR on the input space, so the DDS accepted the reduced dimensionality representation as input. Using the DDS trained on the reduced dimensionality input, we evaluated the performance of the Pareto optimal solutions without the reconstruction error
- we compared the performance estimates obtained from the DDSs trained on the different input spaces to assess the impact of the reconstruction error on the optimizer's ability to identify optimal hulls in the original parameterization

This approach provided a more informative check than merely evaluating the reconstruction error, which had not been standard in the literature, and ensured that the optimization had been effective.

After MSHFO (3) we can move to MSHFO (4). Following the approach of [239] we first check that the performance of the designs on the Pareto frontier, in terms of R_t, are not actually numerical artefacts induced by the DSS and match the performance indicated by high-fidelity CFD. Furthermore, following the same principle as the check on R_t, we check that the performance of the DDS matches the panel code when assessing Eq. (3.1).

To summarize, we modified the MSHFO pipeline to simultaneously address Lack (1) and Lack (2). For Lack (1), we propose to integrate the IMO Intact Stability Code directly into the optimization problem, i.e., modifying MSHFO (3). For Lack (2), we propose using DSDR techniques [54, 60] to reduce the number of parameters we need to optimize, i.e., modifying both MSHFO (2) and MSHFO (3).

In more detail, we can summarize our modification to the MSHFO pipeline as follows. The hard constraint ("pass" or "fail") derived from the IMO Intact Stability Code, i.e., according to Eq. (3.1), is inserted directly in the optimization problem. The same constraint is used to remove from the dataset the geometries that do not satisfy it, reducing the number of samples in the dataset of Table 3.2, to be used during the DSDR. This allows us to build a low dimensional search space that accounts for the IMO Intact Stability Code. At first sight, it may seem incomplete that we only reduced the dataset based on the IMO Intact Stability Code and did not account for the geometric constraints on ∇, T, and θ. However, this is a deliberate consideration, as the geometric constraints remain tunable by the user (depending on design preferences), while the IMO Intact Stability constraints are hard ones.

Figure 3.4 visually depicts the classic MSHFO pipeline and Figure 3.5 depicts the proposed pipeline with modifications to address Lack (1) and Lack (2).

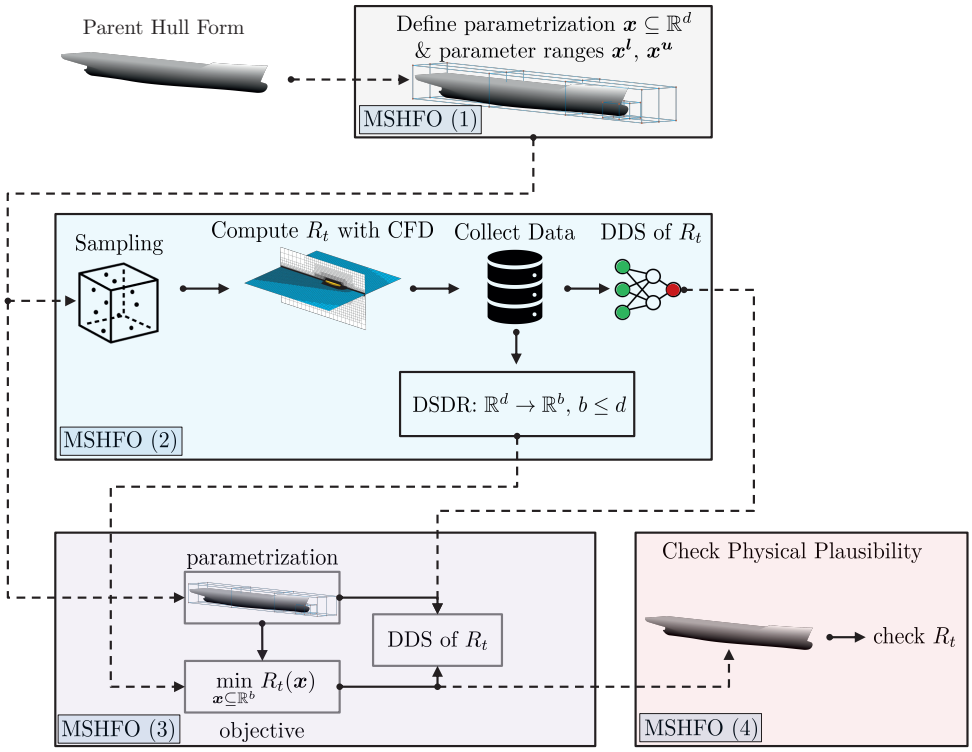


Figure 3.4: Classic MSHFO pipeline.

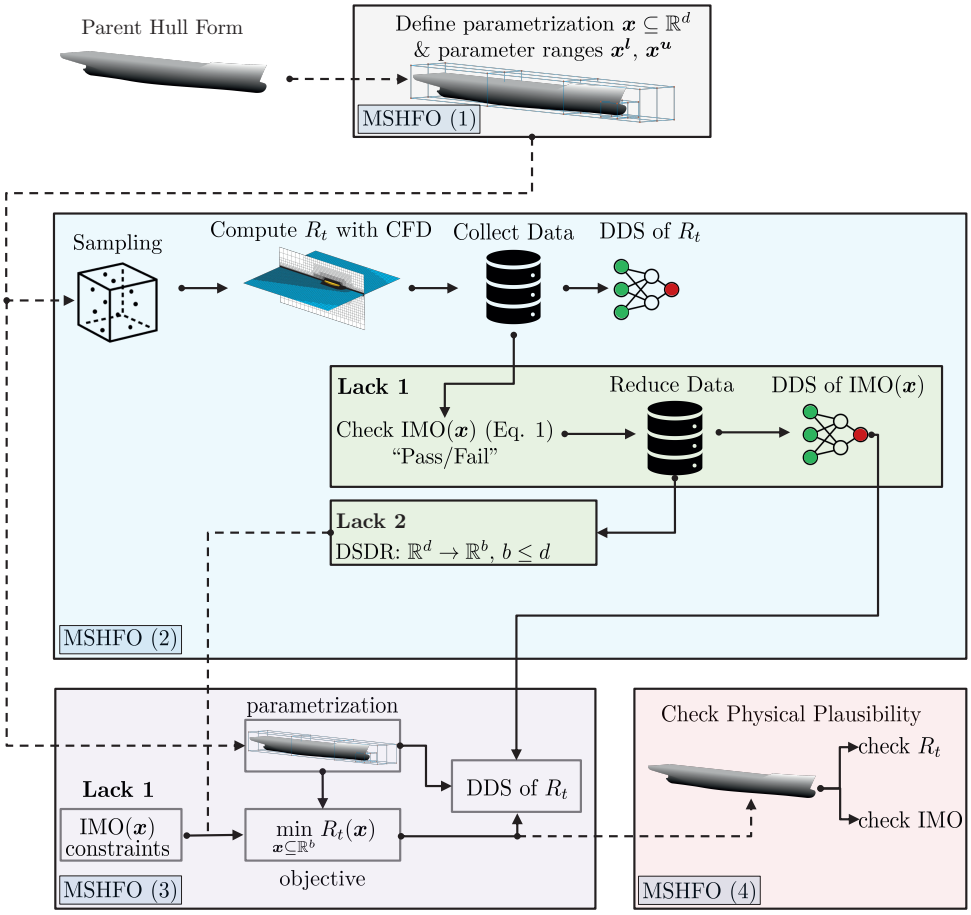


Figure 3.5: MDHFO pipeline with the proposed modifications to address Lack (1) and Lack (2).

3.3 Method

In this section, we have to describe how to fully address the problem described in Section 3.2 with the dataset described in the very same section. In particular, to solve Lack (1) and Lack (2) of the current approach to hull form optimization and blend them together there are a number of steps we need to explain in detail. Section 3.3.1 describes how we build the DDSs of R_t, GZ, and GM so they can be included in the optimization and how we validate their performance; Section 3.3.2 describes how we address Lack (1) by directly incorporating the IMO Intact Stability Code into MSHFO pipeline to optimize the KCS hull form. Section 3.3.2 describes how we address Lack (2), in line with the approaches described in Section 3.1, by using DSDR to reduce the computational demand of optimization. Moreover, for the first time in the literature, we describe how to simultaneously address Lack (1) and Lack (2) by incorporating the stability constraints directly into DSDR.

3.3.1 Data-driven Surrogates for Hull Form Optimization

In this section we describe how we develop the DDSs of R_t, GZ, and GM.

For what concerns the data generation, namely, the CFD model to determine R_t, the mesh generation, computation of the solution, and post-processing of the results is carried out in the commercial CFD package Star CCM+[1]. The baseline High-Fidelity (HF) mesh consists of approximately 1.5 million cells for half of the hull. The mesh is clustered around the hull, with extra refinement around the bulb and stern, across the free surface, and along the Kelvin angle. Figure 3.6 provides an overview of the mesh configuration used in this study. To compute the solution, a wall function approach was used to compute near-wall velocities, along with a prism-layer mesh in the boundary layer zone to achieve an average wall distance of approximately 60 [−]. This baseline configuration also serves as the finest mesh used in the sensitivity analysis. The calculations were performed at a time step of 0.01 [s] to compute steady-state regimes for multi-phase flows using the Volume of Fluid technique. The k-ω shear stress transport model was applied for turbulence closure. For what concerns the degrees of freedom of the simulation, we leveraged the Dynamic Fluid Body Interaction module of Star CCM+ to allow for sink and trim in calm water across a range of velocities (v) and consequently, Fn. To validate the state-of-the-art CFD model for the KCS hull form, we perform two checks commonly found in the literature [35]

- First, we carry out a mesh sensitivity analysis to determine simulation convergence;
- Second, we compare the predictions from our baseline mesh with available Experimental Fluid Dynamics (EFD) measurements of the calm water resistance of the model-scale KCS hull.

For the first task, namely, the mesh coarsening analysis, we created two coarser meshes by reducing the number of cells by approximately $1/3$ each time, i.e., Medium-Fidelity (MF) and Low-Fidelity (LF) respectively. We reported the resistance in Figure 3.7 and the sink and trim in Figure 3.8 for each mesh under analysis. Table 3.3 reports the cell count for each mesh and computational requirements as the core hours using an Intel XEON E5-6248R 24C 3.0GHz CPU.

For the second task, since we validated the mesh convergence, we compared our baseline CFD prediction with the available EFD measurements. Table 3.4 reports the total drag,

[1] www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html

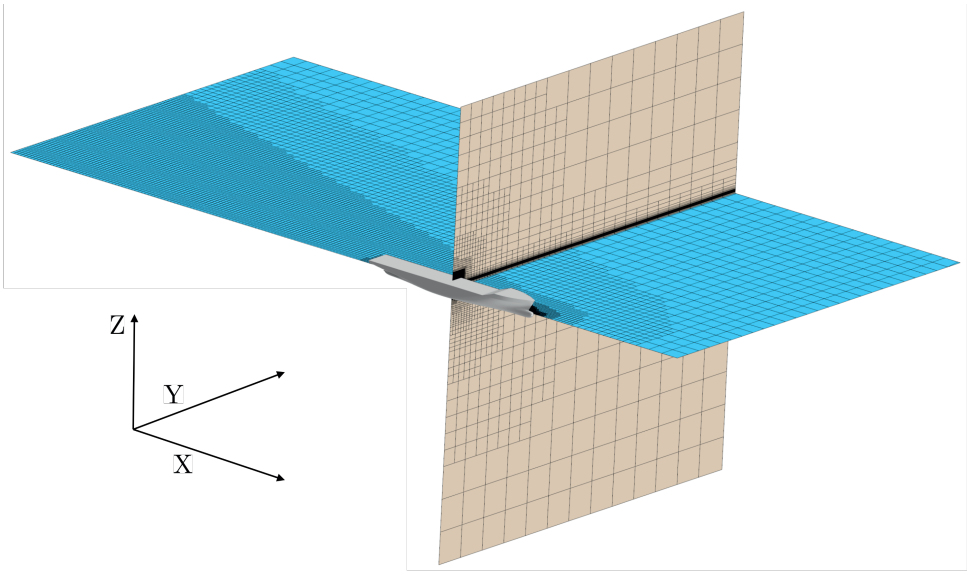


Figure 3.6: The mesh exploited for the CFD simulations with the Star CCM+[1] package. The mesh included a surface mesh refinement on the vessel hull and on the boundaries of the domain, in addition to volume mesh refinements around the hull, wake, and free surface.

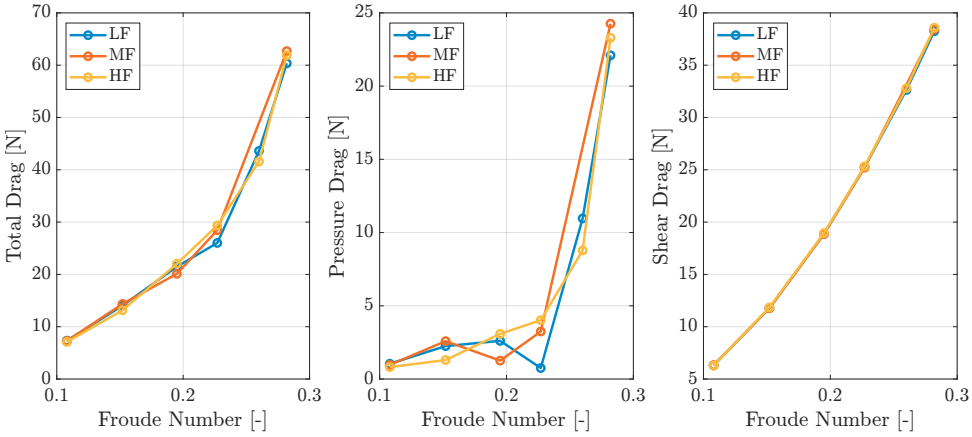


Figure 3.7: Comparison of the Total, Pressure, and Shear Drag for the 3 levels of mesh refinement over varying Fr number.

Table 3.3: Cell count and computational requirements for each Mesh.

Mesh	**$\times 10^6$ cells**	**Core Hours**
HF	1.5	140
MF	1.0	95
LF	0.7	84

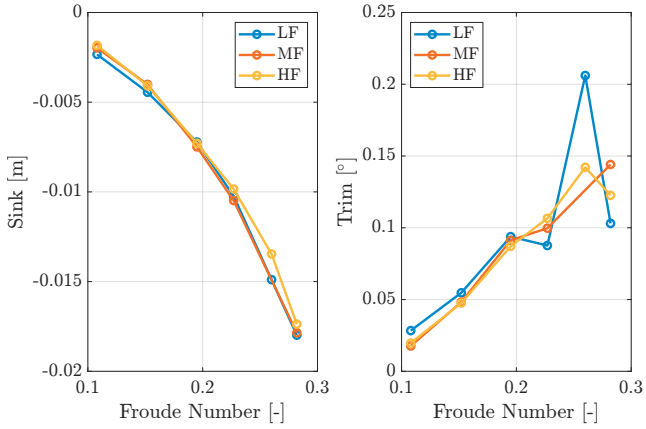


Figure 3.8: Comparison of the Sink and Trim for the 3 levels of mesh refinement over varying Fr number.

Table 3.4: Comparison of EFD and CFD results.

Model test		**EFD**			**CFD**		
v	**Fn**	**Total Drag**	**Sink**	**Trim**	**Total Drag**	**Sink**	**Trim**
$[m/s]$	$[-]$	$[N]$	$[cm]$	$[^\circ]$	$[N]$	$[cm]$	$[^\circ]$
0.915	0.108	15.3	−0.09	−0.017	14.2	−0.20	−0.028
1.281	0.152	27.8	−0.27	−0.053	26.3	−0.40	−0.051
1.647	0.195	45.6	−0.60	−0.097	44.0	−0.72	−0.094
1.922	0.228	65.5	−0.95	−0.127	58.7	−1.00	−0.130
2.196	0.260	98.4	−1.40	−0.170	83.2	−1.39	−0.174
2.379	0.282	127.6	−1.70	−0.160	123.8	−1.74	−0.152

sink, and trim metrics for both the CFD and EFD predictions. Figure 3.9 shows a visual comparison of these same metrics.

It is worth reiterating that while the computational demand of the CFD simulation is manageable for evaluating the performance of a limited number of hull designs, it becomes a significant obstacle in an optimization process where thousands of hull designs must be evaluated. It is also worth mentioning that one of the key limitations of the CFD model is that it evaluates R_t at a model scale (the approximate ratio for this simulation is 1 : 31.6 due to computational constraints) and scaling resistance prediction from model-scale to full-scale ships is still a challenging task [219], but this approach remains an appropriate and effective approximation while running hundreds of full-scale CFD simulations is not computationally tractable.

In this work, to build the DDSs of R_t (see Table 3.1), GZ and GM (see Table 3.2), we test two Machine Learning algorithms which come from the family of kernel methods [206] and neural networks [80]. Specifically, we test Kernel Ridge Regression (KRR) [206] and Extreme Learning Machine (ELM) [100, 101]. These two models share an effective training phase and a smooth functional form [190] which is simple to optimize [20] with respect to

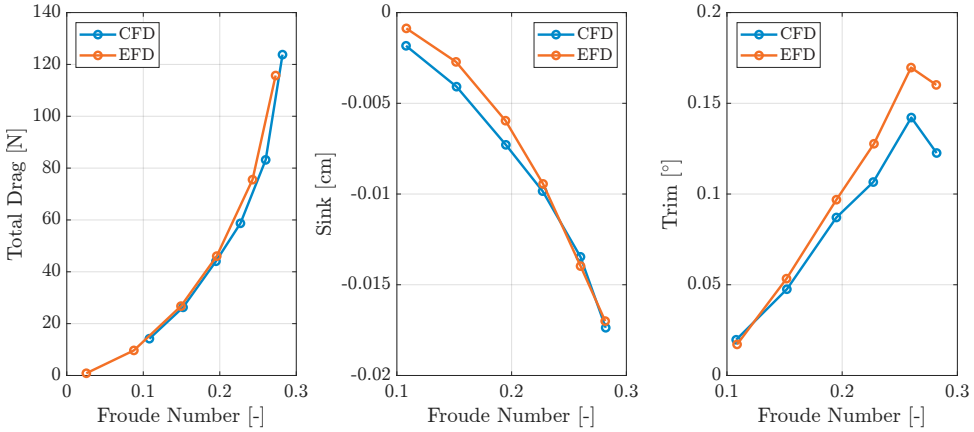


Figure 3.9: Comparison of the Total Drag, Sink and Trim for the selected CFD model and EFD results over varying Fr number.

Table 3.5: Hyperparameters and hyperparameters search space for the learning from data algorithms.

Algorithms	Hyperparameters
KRR	$\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$
	$\gamma : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$
ELM	$h_l : \{2^{5}, 2^{6}, \cdots, 2^{16}\}$
	$\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$

other methods like XGBoost [28] and Random Forests [21]. To build the DDSs using the data in Tables 3.1 and 3.2, we still have to face the problem of how to tune the hyperparameters for each learning from data algorithm (namely, model selection) and how to estimate the performance of the final model (namely, error estimation) [169]. During model selection, the main consideration is how to tune the hyperparameters of each algorithm. For KRR, we rely on the Gaussian Kernel for theoretical and practical reasons in [119]. Thus we have to tune the regularization hyperparameter (λ) and the kernel hyperparameter (γ). Lastly, for ELM, we chose to rely on the Sigmoid activation function and tune the number of neurons in the hidden layer (h_l) and the regularization hyperparameter (λ). The hyperparameters and search space for each algorithm are found in Table 3.5. Since we fixed the boundaries of the representation space sufficiently large to ensure that the solution was not on the border, we only need to validate the performance of the DDS when interpolating within the boundaries of the dataset. Therefore, we validate the performance of the DDS by performing a Leave One Out cross-validation [169]. For what concerns error estimation, the accuracy is measured by different metrics: three quantitative (the Mean Absolute Error - MAE, the Mean Square Error - MSE, and the Mean Absolute Percentage Error - MAPE) [163] and one qualitative (the scatter plot actual versus predicted value) [192]. In terms of computational requirements, the performance is measured using time to build the model (Train Time) and time to make a prediction (Test Time). Since our

surrogate will be leveraged in an optimizer, the most important computational metric is the Test Time.

3.3.2 Hull Form Optimization and Physical Plausibility

This section describes the optimization problem under exam and how we integrate the IMO Intact Stability Code into the MSHFO schema to achieve compliance by design. We formulate a multi-objective hull form optimization problem at high and low Fr subject to (s.t.) constraints on the geometry generated by the parameters x, and properties of the actual geometry, namely the draft (T), the trim (θ), and the displacement (∇). To trade off a fast cruise speed versus a slow steaming case we selected the high and low Fr as $\mathrm{Fr}^{\mathrm{High}} = 0.282$ and $\mathrm{Fr}^{\mathrm{Low}} = 0.108$ respectively. For the constraints on T, θ, and ∇, note that, unlike the hard constraints informed by the IMO Intact Stability Code, these ones are tunable according to design preference. For simplicity, we constrained $\mathrm{T}(x)$ and $\theta(x)$ to match the conditions of the parent design, i.e., $\mathrm{T}^p = 10.8\ [m]$ for the KCS at full-scale and $\theta^p = 0\ [^\circ]$ for the even keel position. Whereas, we constrained $\nabla(x)$ to vary within $\pm 10\%$ of the conditions of the parent design, e.g., a lower bound of $0.9 \cdot \nabla^p$ and upper bound of $1.1 \cdot \nabla^p$, where $\nabla^p = 1.08 \times 10^4\ [m^3]$ is the displacement of the KCS at full-scale. Consequently, the optimization problem becomes

$$\begin{aligned} \min_{x^l \le x \le x^u} \quad & (\lambda)\tilde{R}_t(x, \mathrm{Fr}^{\mathrm{High}}) + (1-\lambda)\tilde{R}_t(x, \mathrm{Fr}^{\mathrm{Low}}), \\ \text{s.t.} \quad & \mathrm{T}(x) = \mathrm{T}^p, \\ & \theta(x) = \theta^p, \\ & 0.9\nabla^p \le \nabla(x) \le 1.1\nabla^p. \end{aligned} \tag{3.2}$$

Furthermore, for the reasons described in Section 3, and to solve Lack (1), we propose constraining the optimization problem with the IMO Intact Stability Code. In particular, we add another constraint to Problem (3.2) to check that the geometry induced by x passes the IMO Intact Stability Code according to Eq. (3.1), namely

$$\begin{aligned} \min_{x^l \le x \le x^u} \quad & (\lambda)\tilde{R}_t(x, \mathrm{Fr}^{\mathrm{High}}) + (1-\lambda)\tilde{R}_t(x, \mathrm{Fr}^{\mathrm{Low}}), \\ \text{s.t.} \quad & \mathrm{T}(x) = \mathrm{T}^p, \\ & \theta(x) = \theta^p, \\ & 0.9\nabla^p \le \nabla(x) \le 1.1\nabla^p, \\ & \mathrm{IMO}(x) = 1 \quad \text{see Eq. (3.1).} \end{aligned} \tag{3.3}$$

Problem (3.3) is now a non-linear and non-linearly constrained optimization problem and a number of approaches exist in literature to address it [252]. We take advantage of some of the most popular gradient-based optimization algorithms [89]. Table 3.6 outlines the optimization algorithms and their hyperparameter settings. Note that, for all of the approaches summarized in Table 3.6, we manually implemented a multi-start with 10 repetitions.

Table 3.6: Hyperparameter settings for the different optimization algorithms.

Algorithm	Parameter	Value
IP	Maximum number of function evaluations allowed	10^6
	Maximum number of iterations allowed	10^6
AS	Maximum number of function evaluations allowed	10^6
	Maximum number of iterations allowed	10^5
	Maximum number of SQP iterations allowed	600

3.3.3 Computational Awareness and Physical Plausiblity for Hull Form Optimization

This section addresses how we use DSDR to address Lack (2) of the current approach to MSHFO and, for the first time in the literature, how we blend the solutions to address both Lack (1) and Lack (2) simultaneously.

Regarding Lack (2), we still need to describe how to validate our approach to using PCA or Neural Networks to reduce the complexity of the design space (namely, DSDR) and how to validate the reconstruction. The most straightforward approach for DSDR is to use PCA [54].

PCA assumes that the data resides in a low-dimensional informative space that has been transformed into a higher-dimensional space. Basically, PCA fits a d-dimensional ellipsoid to the data, with each ellipsoid axis representing a new component. The longer the axis, the greater the data variance along that dimension, making that component more significant due to its higher variability. Conversely, components with lower variance are less informative. However, it is worth mentioning that this method is simplistic considering linear combinations of the original design variables. On the one hand, this means that PCA has a very low computational cost, which is ideal for the problem at hand. On the other hand, PCA has been shown to be unsuitable for complex data in other applications [194].

Another approach to DSDR is to use Neural Networks [141]. This approach tries to map the input data into a lower dimensional space, which can maintain the information to reconstruct the original input data (auto-encoder) [195]. The advantage of Neural Networks is that they can also find non-linear relations that are not findable with PCA. By learning a compressed data representation, the Neural Network effectively reduces dimensionality while preserving essential features. However, despite its efficacy, this approach requires careful tuning of a number of hyperparameters, e.g., the number of hidden layers, the number of nodes in each layer, and the choice of activation function, to ensure optimal performance. This can result in a large computational burden during both the training phase and a more significant computational overhead than PCA during the test time [201].

To perform the DSDR, using PCA or Neural Networks, we first randomly sample 1000 geometries from the design space and generate a realistic sample set of the design space x [53]. We then leverage the approach of PCA or Neural Networks on x, to reduce the dimensionality of the parametrization from $x = \mathbb{R}^d \rightarrow x' = \mathbb{R}^b$. Where x' is the projection of x into a lower-dimensional representation, i.e., $b \leq d$. When we perform the DSDR, we have to tune the hyperparameters of PCA or Neural Networks to perform the projection

from $x \rightarrow x'$. For PCA, the hyperparameter is simply the percentage of variance in the x we want to preserve in x' (%var) which we set to 99%. For Neural Networks, we use a single hidden layer with the sigmoid activation function and have to tune the number of neurons in the hidden layer (h_l) which we set to match the dimensionality of the data projected by the PCA. Note that, for Neural Networks, we also have to select the optimizer, in our case ADAM [126], the learning rate l_r, in our case 0.001, and the number of epochs, in our case 1000. Note that the DSDR is fully unsupervised, so there is no risk of overfitting.

Once we have performed the DSDR, for the reasons described in Section 3.2, we have to validate that the representation in $\mathbb{R}^b$ can be retrieved and well-represented in $\mathbb{R}^d$. To perform this check is quite simple. For PCA, we reconstruct the data using the covariance matrix's eigenvectors to project the lower-dimensional data back into the original design space. For Neural Networks, the reconstruction comes for free from the auto-encoders. Then we check the reconstruction error, which is the difference between the original data in $\mathbb{R}^d$ and the data projected first into $\mathbb{R}^b$ and then back to $\mathbb{R}^d$. We measure this error using the same three quantitative metrics as before: the MAE, MSE, and MAPE.

3.4 Results

In this section, we will present the result of applying the methods described in Section 3.3 to the problem described in Section 3.2 using the data described in this very same section. The results are organized as follows. First, we report the performance of the developed DDSs to predict R_t, GZ, and GM. Subsequently, to address Lack (1), we report the results of optimization problem 3.3, with and without the stability constraints informed by IMO(x) - Eq. (3.1). Lastly, to address Lack (2), we report the results of our optimization problem with DSDR and show that we can simultaneously address Lack (1) and Lack (2) to further improve the computational complexity of optimization.

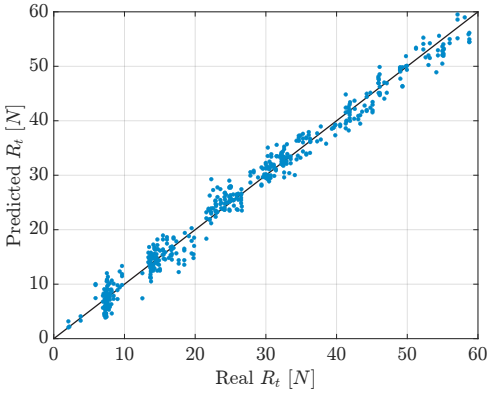
Table 3.7 reports the performance of the DDS of R_t (in terms of accuracy - MAE, MSE, and MAPE - and time - Train Time and Test Time) using the learning algorithms described in Section 3.3.1 (KRR and ELM). Figure 3.10 reports the associated scatter plots.

From Table 3.7 it can be observed that KRR outperforms ELM in terms of MAE and MSE accuracy (visually represented in Figure 3.10). Moreover, both KRR and ELM have a test time in fractions of milliseconds, which makes them well suited for use in MSHFO (3).

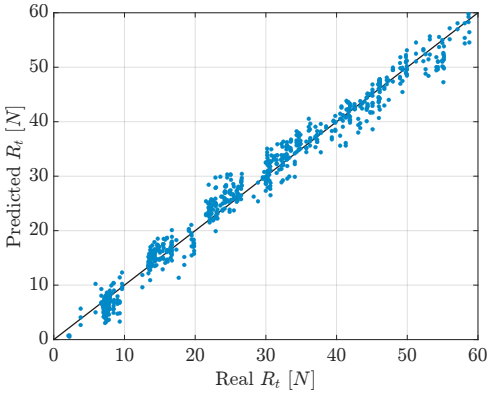
For what concerns the IMO constraint, we developed DDSs (according to Section 3.3.1) for GZ and GM as each of the constraint criteria - SFM (1), SFM (1.1), SFM (1.2), SFM (1.3), and SFM (1.4) - can be directly related to these quantities. Table 3.8 reports the results for the GZ and GM DDSs analogously to the ones of Table 3.7 for the DDS of R_t. Figure 3.11,

Table 3.7: DDS of R_t performance (in terms of accuracy - MAE, MSE, and MAPE - and time - Train Time and Test Time) for the different learning algorithms (KRR and ELM).

Learning	**Accuracy**			**Time**	
Algorithm	**MAE** $[N]$	**MSE** $[N^2]$	**MAPE** $[\%]$	**Train** $[s]$	**Test** $[\mu s]$
KRR	1.04 ± 0.07	1.32 ± 0.64	1.76 ± 0.09	322 ± 3	12.6 ± 5.6
ELM	1.59 ± 0.20	2.02 ± 0.83	4.09 ± 1.15	10095 ± 105	4.6 ± 2.7



(a) KRR



(b) ELM

Figure 3.10: Scatter plot of DDS of R_t for the models in Table 3.7.

Table 3.8: GZ and GM DDSs: performance (in terms of accuracy - MAE, MSE, and MAPE - and time - Train Time and Test Time) for the different learning algorithms (KRR and ELM).

DDS	**Learning**	**Accuracy**			**Time**	
Target	**Algorithm**	**MAE** $\times 10^{-3}[m]$	**MSE** $\times 10^{-3}[m^2]$	**MAPE** [%]	**Train** [s]	**Test** [μs]
GZ	KRR	4.5 ± 0.3	58.0 ± 5.6	1.00 ± 0.00	453 ± 5	9.2 ± 5.0
	ELM	2.7 ± 0.2	24.6 ± 3.8	1.00 ± 0.01	10245 ± 216	5.2 ± 2.9
GM	KRR	4.8 ± 0.2	26.1 ± 7.5	1.00 ± 0.00	429 ± 4	9.1 ± 4.6
	ELM	3.3 ± 0.4	18.5 ± 3.6	1.00 ± 0.01	9383 ± 921	16.4 ± 5.3

Table 3.9: IMO (SFM (1), SFM (1.1), SFM (1.2), SFM (1.3), SFM (1.4), and the IMO(x) of Eq. (3.1)) DDS based on the GZ and GM DDDs: Confusion Matrices (values are in %).

(a) SFM (1).

		DDS	
		Pass	**Fail**
Real	**Pass**	100	0
	Fail	0	0

(b) SFM (1.1).

		DDS	
		Pass	**Fail**
Real	**Pass**	100	0
	Fail	0	0

(c) SFM (1.2).

		DDS	
		Pass	**Fail**
Real	**Pass**	80	1
	Fail	0	19

(d) SFM (1.3).

		DDS	
		Pass	**Fail**
Real	**Pass**	89	0
	Fail	0	11

(e) SFM (1.4).

		DDS	
		Pass	**Fail**
Real	**Pass**	49	2
	Fail	1	43

(f) IMO(x) - Eq. (3.1).

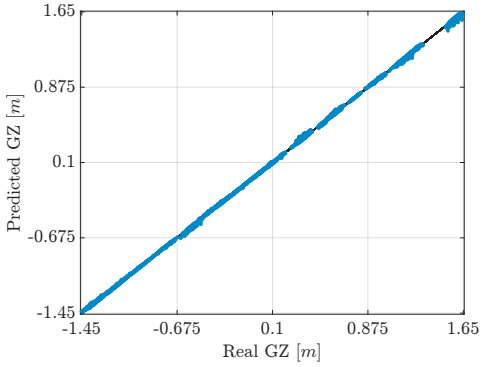
		DDS	
		Pass	**Fail**
Real	**Pass**	49	4
	Fail	2	45

instead, is the equivalent of Figure 3.10. From Table 3.8 it can be observed that ELM outperform KRR in terms of MAE and MSE accuracy (visually represented in Figure 3.11) for both GM and GZ and both KRR and ELM have a test time in fractions of milliseconds which makes them well suited for use in MSHFO (3).

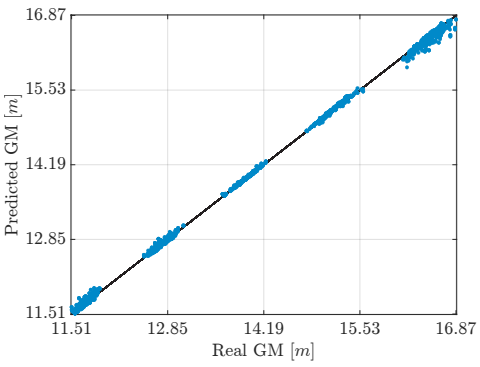
Since the constraint criteria (SFM (1), SFM (1.1), SFM (1.2), SFM (1.3), SFM (1.4), and IMO(x) - Eq. (3.1)) are used to determine feasible geometries (e.g., "pass"/"fail" a particular hull depending on the criteria), we test the efficacy of the best DDSs in this "pass"/"fail" binary classification scenario [163]. In Table 3.9 we compare the "pass"/"fail" criteria informed by the Real method (panel code) and the best DDS using Confusion Matricies [163]. From Table 3.9 it can be observed that

- for SFM (1) and SFM (1.1) all of the hulls pass the criteria;
- for SFM (1.2), SFM (1.3), SFM (1.4), and the IMO(x) - Eq. (3.1) the DDS performs well (low false positives and false negatives).

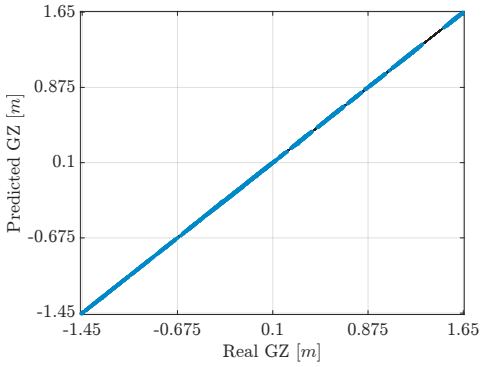
At this point, we have developed and statistically validated the different DDSs. The next step is to highlight the results of the multi-objective hull form optimization of the KCS vessel (parent design) according to Section 3.3.2. We first solved the multi-objective optimization excluding the stability constraints, e.g., solving Problem (3.2) and show that the current approach fails to meet the IMO stability guidelines. Subsequently, we solve the



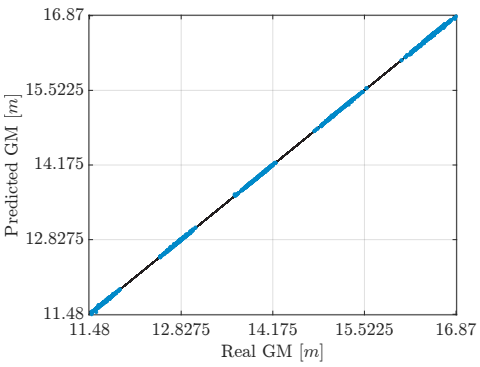
(a) KRR model for GZ.



(b) KRR model for GM.



(c) ELM model for GZ.



(d) ELM model for GM.

Figure 3.11: Scatter plot of DDSs of GZ and GM for the models in Table 3.8.

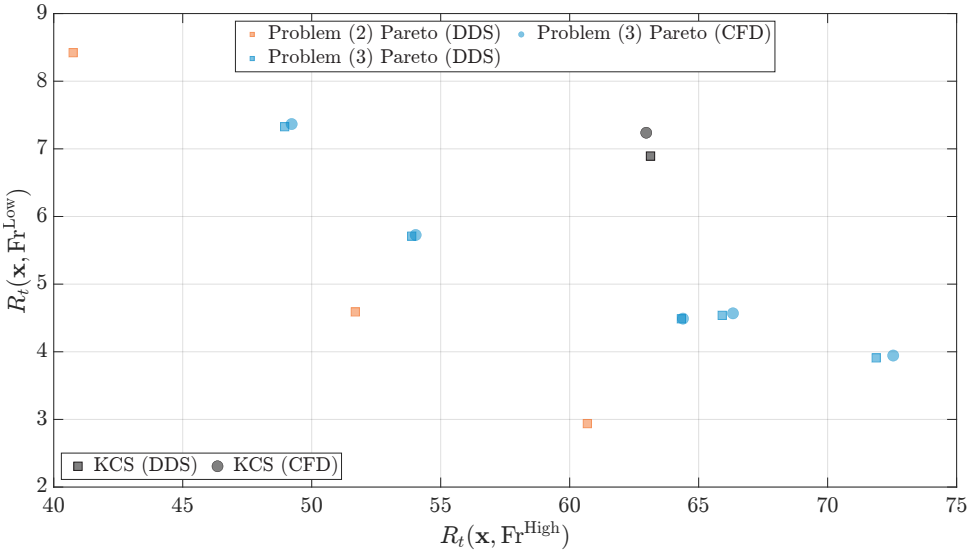
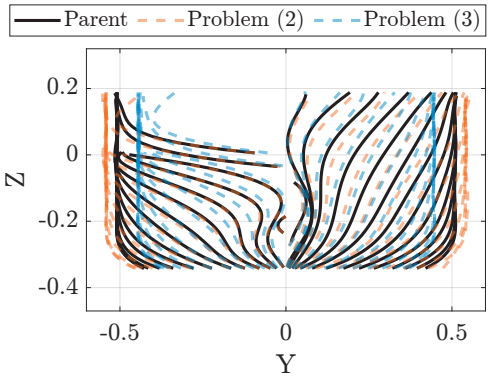


Figure 3.12: Pareto Front for solving Problems (3.2) and (3.3) and the KCS (parent design).

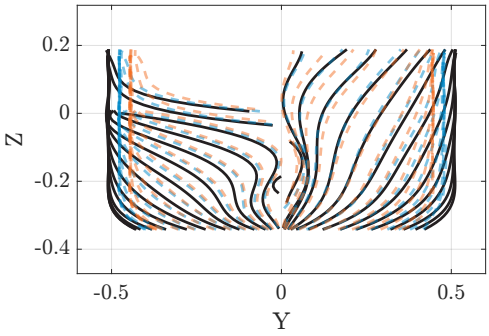
Table 3.10: Average Number of Function Calls and Running Time to solve Problem (3.2) and (3.3).

Problem	**Optimization Algorithm**	**Calls** $[-]$	**Time** $[s]$
Problem (3.2)	IP	3006 ± 786	164 ± 47
	AS	1218 ± 198	153 ± 24
Problem (3.3)	IP	4471 ± 434	223 ± 16
	AS	1893 ± 1108	224 ± 128

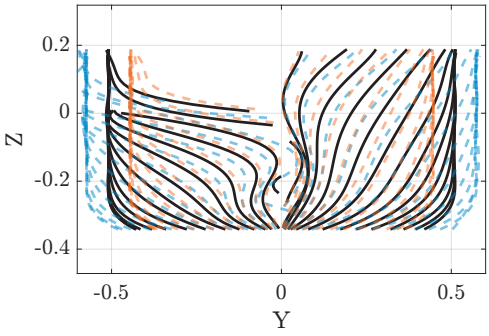
multi-objective optimization, including the stability constraints, e.g., solving Problem (3.3) and show that our approach achieves compliance by design. Figure 3.12 shows the Pareto front for solving Problems (3.2) and (3.3) and the KCS (parent design). Table 3.10 shows the average number of function calls and running time for Problems (3.2) and (3.3). Figure 3.13 reports the comparison between the KCS (parent) design and the optimized designs. Figure 3.14 shows a comparison between DDS and Panel Code informed GZ curves for Problem (3.2) and Problem (3.3) designs. Note that, since the geometries found by solving Problem (3.2) failed IMO(x) - Eq. (3.1), there is no need to validate them with CFD; however, the cost informed by the DDS for Problem (3.3) is checked with the CFD model (see Section 3.3.1) for physical plausibility. From Figure 3.12 it can be observed that without including the stability constraints, i.e., solving Problem (3.2), the DDS induces the optimizer to find geometries that significantly outperform (according to the DDS) the KCS (parent design). Moreover, when validating the geometries found by solving Problem (3.2) with IMO(x) - Eq. (3.1), we find that these geometries are in fact not feasible (see Figures 3.13 and 3.14 for more details). Finally, by including the stability constraints, i.e., solving Problem (3.3), the DDS induces the optimizer to find geometries that perform worse than the ones found by solving Problem (3.2) but still outperform the KCS (parent design).



(a) $\lambda = 0.$

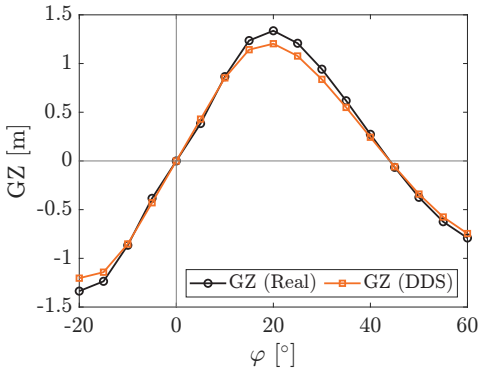


(b) $\lambda = 0.5.$

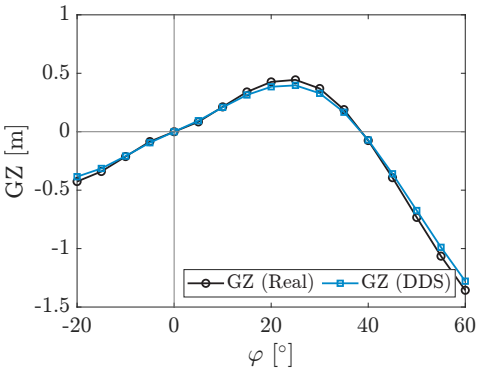


(c) $\lambda = 1.0.$

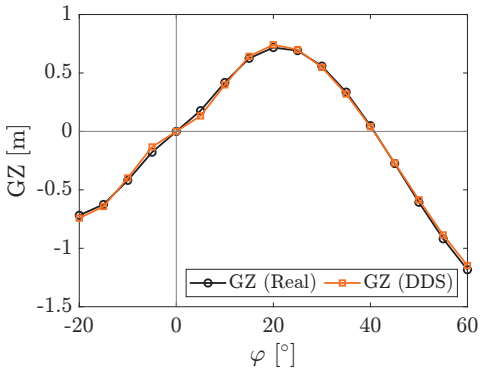
Figure 3.13: Comparison between KCS (parent) design in black and optimized (Problem (3.2) and Problem (3.3)) designs.



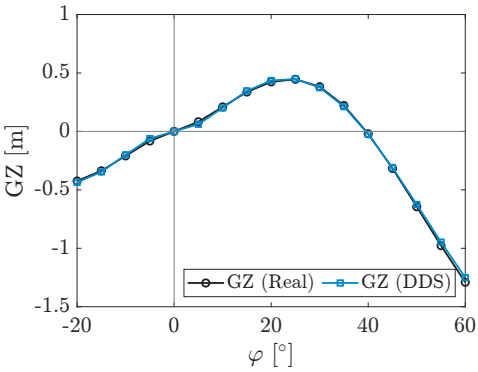
(a) Problem (3.2) $\lambda = 0.0$.



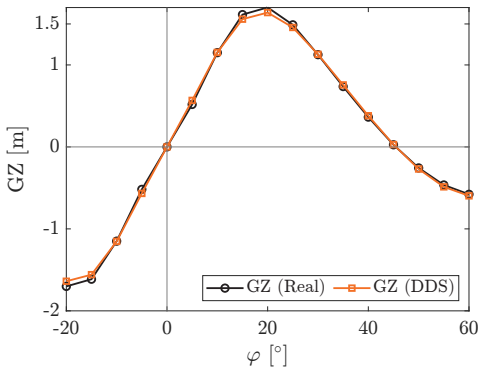
(b) Problem (3.3) $\lambda = 0.0$.



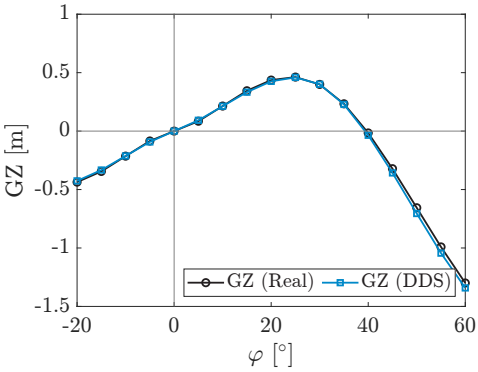
(c) Problem (3.2) $\lambda = 0.5$.



(d) Problem (3.3) $\lambda = 0.5$.



(e) Problem (3.2) $\lambda = 1.0$.



(f) Problem (3.3) $\lambda = 1.0$.

Figure 3.14: Comparison between DDS and Panel Code Stability for Problem (3.2) and Problem (3.3) designs.

Table 3.11: DSDR reconstruction error (in terms of MAE, MSE, and MAPE) and the resulting dimension (DIM) of the new low dimensional subspace using PCA and NN algorithms. We perform the dimensionality reduction using two datasets: (i) the entire dataset (i.e., DSDR_{STD}) and (ii) just the subset of the dataset that satisfies the IMO Intact Stability Code (i.e., DSDR_{IMO}).

Dataset	**DIM**	**DSDR Algorithm**	**Reconstruction Error**		
			MAE $[\times 10^{-2} N]$	**MSE** $[\times 10^{-2} N^2]$	**MAPE** [%]
DSDR_{STD}	7	PCA	22.15 ± 0.03	10.50 ± 0.01	0.70 ± 0.03
		NN	21.56 ± 0.04	10.48 ± 0.00	0.67 ± 0.04
DSDR_{IMO}	4	PCA	20.45 ± 0.04	10.90 ± 0.02	0.50 ± 0.04
		NN	20.32 ± 0.05	10.88 ± 0.01	0.46 ± 0.05

From Figure 3.13 it can be observed that for $\lambda = 1$ the design found from solving Problem (3.2) extended further in the transverse direction than the parent design and the one found by solving Problem (3.3) (see Figure 3.13c). We validate the stability of the designs in Figure 3.13 by comparing the DDS and Panel Code Stability in Figure 3.14. For $\lambda = 0$, $\lambda = 0.5$, and $\lambda = 1.0$ it can be observed that the GZ curves found by solving Problem (3.2) fail Eq. (3.1) because the angle of the maximum GZ, namely, φ_{max}, fails the condition to occur at $\varphi_{\text{max}} \geq 25^{\circ}$ (see Figures 3.14a, 3.14c, and 3.14e). The GZ curves found by solving Problem (3.3) are in line with the expectation of IMO(x) - Eq. (3.1) (see Figures 3.14b, 3.14d, and 3.14f).

From Table 3.10 it can be observed that including IMO(x) - Eq. (3.1) in the optimization problem increases the computational complexity of optimization by approximately 30%.

After addressing Lack (1), by showing the importance of including the IMO(x) of Eq. (3.1) in the optimization problem, we need to address Lack (2), namely, the computational cost of optimization. For the reasons described in Section 3.2 and according to the method described in Section 3.3.3 we test different DSDR algorithms (PCA and Neural Networks - NN) to reduce the computational complexity of solving Problem (3.3).

Table 3.11 reports the reconstruction error (in terms of MAE, MSE, and MAPE) and the resulting dimension (DIM) of the new low dimensional subspace using PCA and NN algorithms. We perform the dimensionality reduction using two datasets: (i) the standard one that uses the entire dataset (i.e., DSDR_{STD}) and (ii) our new proposal that used just the subset of the dataset that satisfies the IMO Intact Stability Code (i.e., DSDR_{IMO}). The case DSDR_{IMO} allows simultaneously addressing Lack (1) and Lack (2). From Table 3.11 we can observe that

- both the DSDR_{STD} and DSDR_{IMO} cases lead to a reduced dimensionality search space that can be well reconstructed back to the original dimensionality (small reconstruction errors);
- the DSDR_{IMO} case leads to a smaller search space than the DSDR_{STD} case because there is less variance among the examples preserved in the dataset.

Now that we have established we can project the data into a reduced complexity space (e.g., $\mathbb{R}^7$ in the DSDR_{STD} case or $\mathbb{R}^4$ in the DSDR_{IMO} case) we want to test the impact of the dimensionality reduction on the ability to predict R_t. Table 3.12 reports the surrogate performance (in terms of accuracy - MAE, MSE, and MAPE - and time - Train Time and

Table 3.12: Surrogate performance (in terms of accuracy - MAE, MSE, and MAPE - and time - Train Time and Test Time) for the different learning algorithms to build DDS for R_t when we use all the inputs, or a subset of the input induced by the DSDR (with NN - i.e., the best performing DSDR method according to Table 3.11) for both the DSDR_{STD} and DSDR_{IMO} cases.

Case	DIM	Learning Algorithm	Accuracy			Time	
			MAE $[N]$	**MSE** $[N^2]$	**MAPE** $[\%]$	**Train** $[s]$	**Test** $[\mu s]$
No DSDR	Original	KRR	1.04 ± 0.07	1.32 ± 0.64	1.76 ± 0.09	322 ± 3	12.6 ± 5.6
		ELM	1.59 ± 0.20	2.02 ± 0.83	4.09 ± 1.15	10095 ± 105	4.6 ± 2.7
DSDR_{STD}	7	KRR	0.99 ± 0.14	0.99 ± 0.53	1.36 ± 0.11	205 ± 7	11.3 ± 3.8
		ELM	1.45 ± 0.27	0.99 ± 0.23	3.38 ± 0.08	99987 ± 113	4.5 ± 3.0
DSDR_{IMO}	4	KRR	0.98 ± 0.14	0.99 ± 0.36	1.38 ± 0.09	207 ± 6	11.1 ± 1.8
		ELM	1.45 ± 0.36	0.99 ± 0.21	3.41 ± 0.13	99827 ± 103	4.2 ± 3.1

Test Time) for the different learning algorithms to build DDS for R_t when we use all the inputs, or a subset of the input induced by the DSDR (with NN - i.e., the best performing DSDR method according to Table 3.11) for both the DSDR_{STD} and DSDR_{IMO} cases. From Table 3.12 we can observe that the accuracy remains high with low MAE, MSE, and MAPE errors for the two different low dimensional search spaces. Notably, performance slightly improves, meaning that some dimensions in the original space are actually not useful.

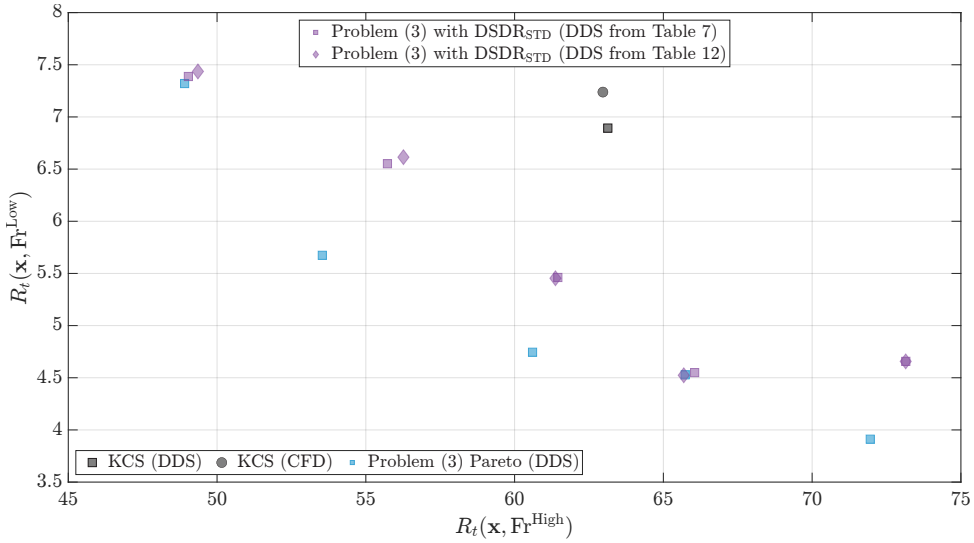
The next step is to perform the actual numerical optimization, i.e., solve Problem (3.3), with the two different low dimensional search spaces (i.e., the DSDR_{STD} and the DSDR_{IMO}), to address Lack (2). Figure 3.15 shows the results of solving Problem (3.3) with DSDR_{STD} and DSDR_{IMO}. In particular, Figure 3.15 reports

- gray marks: the KCS (parent design) both using the DDS (with no DSDR, i.e., Table 3.7) with a square and the CFD with a circle;
- blue square marks: the results of Figure 3.12 for Problem (3.3) (i.e., no DSDR);
- purple square marks: solutions of Problem (3.3) with DSDR and DDS trained on the original input dimension but fed with the input reconstructed from the smaller dimensional space;
- purple diamonds marks: solutions of Problem (3.3) with DSDR and DDS trained on the smaller input dimension induced by the DSDR.

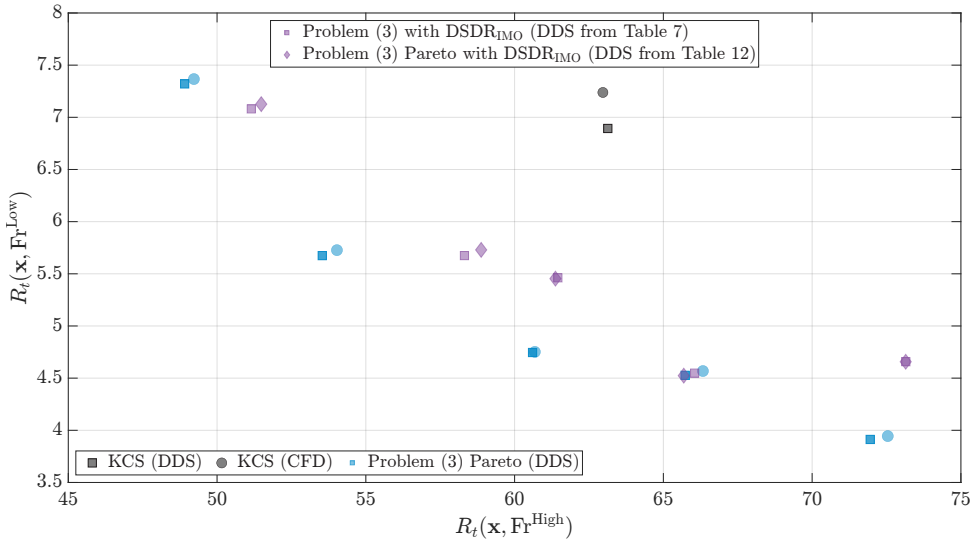
From Figures 3.15 we can observe the following

- using the original dimension of the lower dimension based DDS are approximately equal, showing that the reconstruction does not negatively impact the optimization;
- the two different low dimensional search spaces (DSDR_{STD} and DSDR_{IMO}) both find geometries that outperform the KCS (parent) design;
- the two different low dimensional search spaces (DSDR_{STD} and DSDR_{IMO}) do not outperform solving Problem (3.3) in the original dimensionality.

The results of solving Problem (3.3) with the two different low dimensional search spaces (DSDR_{STD} and DSDR_{IMO}) still need to be checked for physical plausibility (i.e., using the CFD model in Section 3.3.1). Figure 3.16 shows the same results as Figure 3.15 but



(a) DSDR$_{\text{STD}}$



(b) DSDR$_{\text{IMO}}$

Figure 3.15: Results of solving Problem (3.3) with DSDR$_{\text{STD}}$ and DSDR$_{\text{IMO}}$. Gray marks: the KCS (parent design) both using the DDS (i.e., Table 3.7) with a square and the CFD with a circle. Blue square marks: the results of Figure 3.12 for Problem (3.3) (i.e., no DSDR). Purple square marks: solutions of Problem (3.3) with DSDR and DDS trained on the original input dimension but fed with the input reconstructed from the smaller dimensional space. Purple diamonds marks: solutions of Problem (3.3) with DSDR and DDS trained on the smaller input dimension induced by the DSDR.

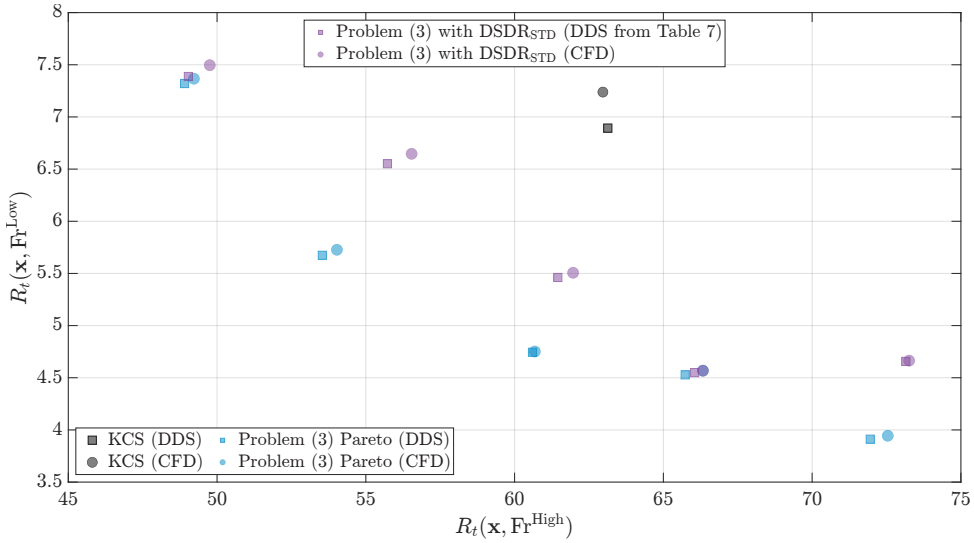
Table 3.13: Average number of Function Calls and Running Time for solving Problem (3.3) for the two different low dimensional search spaces (DSDR and DSDR+IMO) and Dimensionality (Dim.) of the problem and Optimization Algorithm. The difference from solving the problem in the original dimensionality (Table 3.10) is also reported.

Case	DIM	Optimization Algorithm	Calls	Time $[s]$	Savings	
					Calls	Time $[s]$
No DSDR	Original	IP	4471 ± 434	223 ± 16	-	-
		AS	1893 ± 1108	224 ± 128		
$DSDR_{STD}$	7	IP	2808 ± 331	194 ± 24	-37.2%	-13.0%
		AS	875 ± 413	186 ± 82	-53.8%	-17.0%
$DSDR_{IMO}$	4	IP	2137 ± 332	172 ± 22	-52.2%	-22.8%
		AS	621 ± 221	134 ± 76	-66.2%	-39.9%

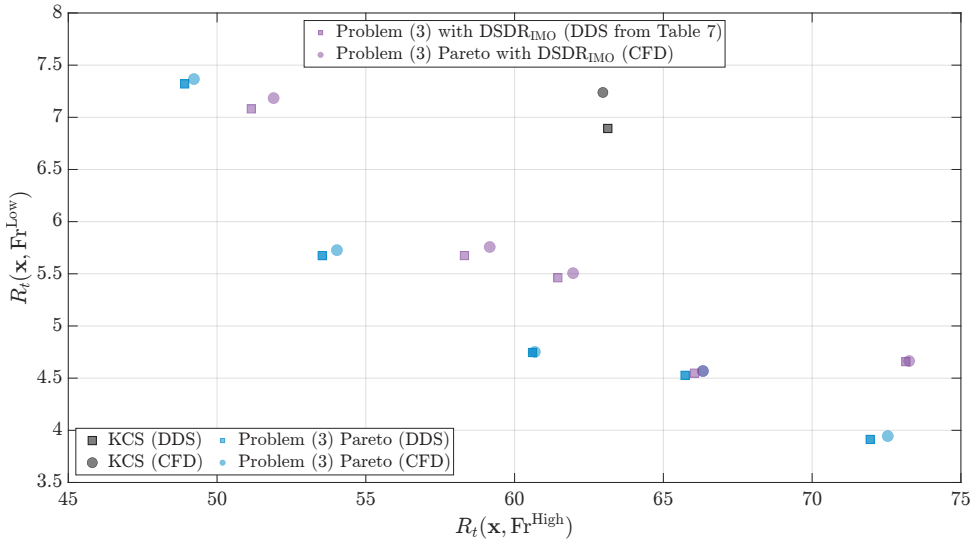
- we removed purple diamonds marks (i.e., solutions of Problem (3.3) with DSDR and DDS trained on the smaller input dimension induced by the DSDR) as there is no need to define different surrogates for different dimensionalities;
- we added purple circle marks for the CFD physical plausibility check for each of the purple square marks (i.e., solutions of Problem (3.3) with DSDR and DDS trained on the original input dimension but fed with the input reconstructed from the smaller dimensional space).

From Figure 3.16 it is possible to observe that the CFD model validates the results of solving Problem (3.3) with the two different low dimensional search spaces ($DSDR_{STD}$, and with $DSDR_{IMO}$).

Finally, Table 3.13 reports the average number of function calls and running time for solving Problem (3.3) for the two different low dimensional search spaces ($DSDR_{STD}$, and with $DSDR_{IMO}$) and optimization algorithms. The time and call savings with respect to solving the problem without DSDR are also reported. Figure 3.17 provides a graphical representation of Table 3.13. From Table 3.13 and figure 3.17 it can be observed that the $DSDR_{IMO}$ approach leads to significant computational time savings ($20 \div 40\%$) compared to solving Problem (3.3) in its original dimensionality. $DSDR_{IMO}$ also outperforms $DSDR_{STD}$ by approximately $10 \div 20\%$ in terms of computational time.



(a) DSDR$_{\text{STD}}$



(b) DSDR$_{\text{IMO}}$

Figure 3.16: Physical plausibility for solving Problem (3.3) with DSDR$_{\text{STD}}$ and DSDR$_{\text{IMO}}$. We report the same results of Figure 3.15 but (i) we removed purple diamonds marks (i.e., solutions of Problem (3.3) with DSDR and DDS trained on the smaller input dimension induced by the DSDR) as there is no need to define different surrogates for different dimensionalities and (ii) we added purple circle marks for the CFD physical plausibility check for each of the purple square marks (i.e., solutions of Problem (3.3) with DSDR and DDS trained on the original input dimension but fed with the input reconstructed from the smaller dimensional space).

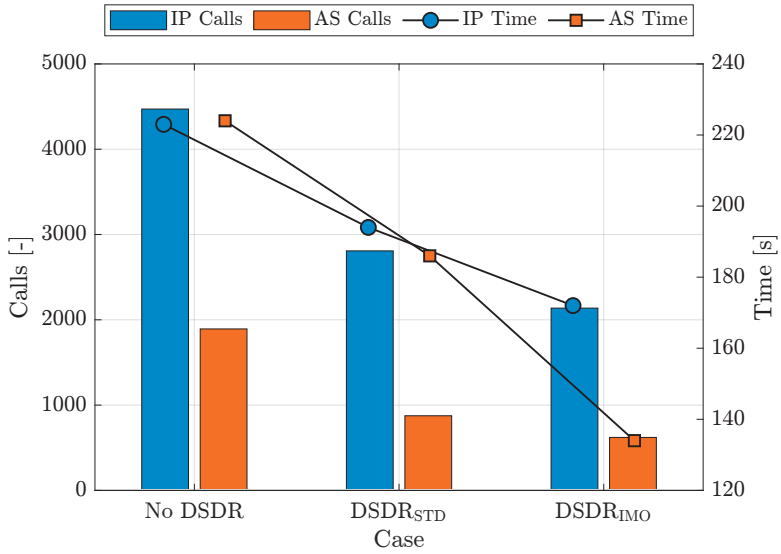


Figure 3.17: Graphical representation of Table 3.13.

3.5 Chapter Summary

This study has presented a novel approach to hull form optimization that directly incorporates the IMO Intact Stability Code into the numerical optimization loop. By developing a data-driven surrogate model tailored to the stability constraint, we have ensured that all resulting designs inherently meet critical stability requirements without needing a-posteriori checks. Our results, obtained through the optimization of the KCS hull form, highlight two principal advantages of this method. The first one is that, unlike traditional optimization pipelines—where stability and other physical plausibility criteria are often verified after the fact, our integrated approach systematically excludes non-compliant designs. This marks a pivotal step toward more reliable, real-world-ready solutions. The second one is that, through a data-driven reduction in the search space, we successfully cut the computational cost by approximately 30% while maintaining the quality of the optimized solutions. This reduction not only accelerates design workflows but also opens the door for more frequent or larger-scale optimization campaigns within the same computational budget.

Taken together, these contributions start to address the longstanding challenge of reconciling performance goals with practical, real-world constraints in a computationally efficient manner. By embedding a key regulatory requirement directly into the design process, our work paves the way for a new generation of design tools that balance hydrodynamic performance, physical plausibility, and computational efficiency.

In future investigations, several avenues remain open. First, extending the set of embedded constraints to other critical design rules (e.g., safety-related or structural considerations) could produce even more robust solutions. Second, coupling data-driven techniques directly into high-fidelity simulations may further improve the accuracy and reliability of

the surrogate models, i.e., physics-informed data-driven techniques. Finally, exploring applications to different vessel types or operating conditions could broaden the applicability of the proposed framework, contributing to a more sustainable and compliant maritime industry.

4

Decoupling Yacht Hull Resistance Optimization

Current approaches to vessel hull resistance optimization rely on a mix of human experience and Data-Driven Models (DDMs): human experts define, via parametrization and parameter ranges, a series of geometries; a surrogate of the relationship between these parameters and the resistance, based on data from Experimental Fluid Dynamics (EFD) or Computational Fluid Dynamics (CFD), is built to interpolate within the defined parameter ranges; finally, the optimal parameters are found by optimizing, with more or less human intervention, the surrogate and used to retrieve the optimal geometry. This approach has its limitations: the need for human intervention in the geometry parametrization and optimization, the need for extensive computational efforts (for CFD), costs (for EFD), and time (for both CFD and EFD), and the limited ability of the approach to work beyond the specific setting (e.g., changes in the family of geometry or extrapolation outside the parameter ranges). For these reasons, in this chapter, we propose to address the current limitations by first leveraging on a parametrization able to accurately describe the entire Delft Systematic Yacht Hull Series (DSYHS). In particular, we decouple this parametrization from the one needed to create the DDM, showing that the DDM can be directly trained on the DSYHS EFD dataset, avoiding the need for both CFD and new EFD customized for the specific problem, matching the performance of state-of-the-art CFD models even in extrapolating conditions (i.e., for geometries and parameter ranges beyond the boundaries used to construct the surrogate), with physical plausibility and minimal human intervention. The results of different and increasingly challenging extrapolating conditions validated with statistical methods on the DSYHS EFD dataset and using state-of-the-art CFD models will support our proposal. Finally, we show that it is possible to optimize the hull resistance by exploring geometric parameters beyond the boundaries of the DSYHS validating the results via state-of-the-art CFD.

This chapter is reproduced from J. M. Walker, A. Coraddu, and L. Oneto. A Decoupled Approach to AI-based Design and Optimization of the Delft Systematic Yacht Hull Series. In *Conference on Computer and IT Applications in the Maritime Industries*, pages 209–222, 2023 and J. M. Walker, A. Coraddu, and L. Oneto. Data-Driven Models for Yacht Hull Resistance Optimization: Exploring Geometric Parameters Beyond the Boundaries of the Delft Systematic Yacht Hull Series. *IEEE Access*, 12:76102–76120, 2024.

Current approaches to vessel hull resistance optimization rely on a mix between human experience and DDMs [36, 70, 82, 98, 136, 140, 156, 158, 262]. As the first step, human experts define a specific parametrization, i.e., a rich yet synthetic quantitative descriptors of a set of candidate geometries, and parameter ranges, i.e., the geometry design space [36, 70, 82, 98, 136, 140, 156, 158, 262]. For this purpose, a number of approaches exist in the literature: from Free-form Deformation (FFD) [36, 70, 82, 140, 156, 262] to B-Splines [98, 140], and model design parameters [136, 158] each one having its strengths and weaknesses (see Chapter 2 for the details). Once the parametrization and parameter ranges have been defined, a dataset composed of parameters' values (using the selected parametrization) and associated resistance (measured with EFD or estimated using CFD) is built [36, 70, 82, 98, 136, 140, 156, 262]. This process is time, computational, and financially demanding [92, 186, 213].

For this reason, it is necessary to carefully select a minimal number of parameters configuration in the parameters ranges, i.e., a small number of candidate geometries, selected with more or less complex strategies [9, 12] and then perform the EFD or run the CFD simulations. EFD are seldom used because of the very specific parametrization, and parameter ranges. Moreover, EFD data are seldom shared and available to researchers and practitioners [217] in many cases due to confidentiality issues. In some cases, an already available set of EFD or CFD is available, and it is possible to enrich it with very few new candidate geometries performing EFD or running CFD simulations [264], but to the best of the authors' knowledge, no one in the literature is proposing this approach. Most, if not all of the work relies just on CFD simulations [36, 82, 98, 136, 140, 156, 262]. Based on the dataset of candidate geometries and their resistance, a DDM-based surrogate of the relationship between the parametrization and the resistance is built, which allows estimating the resistance for a new parameter configuration at a fraction of the time, computational, and financial requirements of the EFD or CFD or both [36, 70, 82, 98, 136, 140, 156, 158, 262]. The resulting surrogate is then exploited, with different levels of human supervision, by an optimizer to search for the optimal parameters configuration in the parameter range, retrieving then the associated optimized geometry [36, 70, 82, 98, 136, 140, 156, 262]. In practical cases, resistance is one of the different design optimality conditions (e.g., resistance at high and low speed), therefore, multiple optimal solutions are retrieved according to the Pareto front [23]. Figure 4.1 summarizes the current approach we just described.

The current approach has its limitations. The first one is the need for more or less partial human supervision in geometry parametrization and optimization [36, 70, 82, 98, 136, 140, 156, 158, 262]. In fact, the parametrization needs to satisfy multiple functional requirements: it must be informative enough to allow for the prediction of the resistance and to be homomorphic (i.e., one geometry corresponds to a particular value of the parameters and vice-versa), but it should be synthetic and intelligible enough to allow for interpretation and test (e.g., for physical plausibility of the results) [36, 70, 82, 98, 136, 140, 156, 158, 262]. Moreover, human intervention should also be limited during the optimization phase: the parametrization and the surrogate should be accurate and physically plausible enough to not induce the optimizer into unfeasible, physically implausible, or degenerate solutions [200]. The second limitation is the need for extensive computational efforts (for CFD), costs (for EFD), and time (for both CFD and EFD) needed to build the dataset required in the surrogation [36, 70, 82, 98, 136, 140, 156, 158, 262]. The ideal situation would

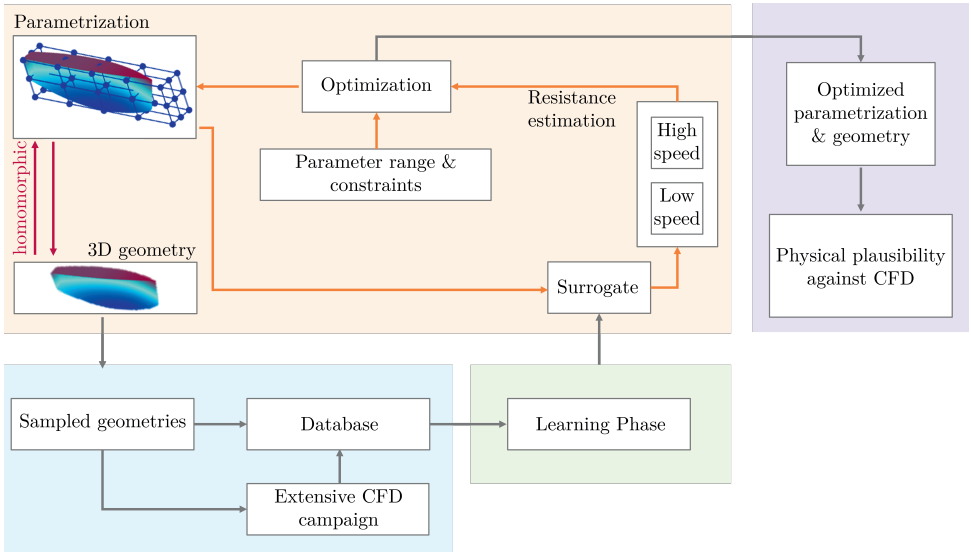


Figure 4.1: Current approach to vessel hull optimization.

be to just rely on previous CFD and EFD and not requiring new CFD and EFD for a new design. The last limitation is the limited ability of the approach to work beyond the specific setting (e.g., changes in the family of geometry or extrapolation outside the parameter ranges) as observed in many works [36, 70, 82, 98, 136, 140, 156, 158, 262].

To overcome the limitations discussed above, we propose a novel approach to vessel hull optimization, summarized in Figure 4.2.

As the first step, we propose a parametrization approach able to cover a large set of geometries (i.e., parent hulls) and not just a variation of a particular parent hull. More specifically, our parametrization is a homomorphy not only able to well represent the entire Delft Systematic Yacht Hull Series (DSYHS) (composed of 6 parent hulls) but, as described later, is also able to perform well beyond the DSYHS (i.e., extrapolate). While requiring some human intervention, this step has the capability to minimize it. In fact, this parametrization can be used for all designs around the 6 parent hulls of the DSYHS, namely, the parametrization step should not be performed every time we change the parent hull as it happens now. This approach paves the way toward more general homomorphic parametrization able to cope with the largest possible sets of parent hulls, allowing us to easily plug them into our pipeline, further decreasing the need for human intervention.

Then, in order to further minimize the human intervention in the parametrization phase, we decoupled the parametrization exploited to define the geometry and to define the optimization parameters from the features necessary to predict the hull resistance based on DDMs. In particular, from a hull geometry defined by a particular configuration of parameters, we exploit the Nautilus code[1] which is able to automatically extract a series of features able to cover and extrapolate over multiple parent hulls while being informa-

[1] https://github.com/mai-lab-tud/nautilus

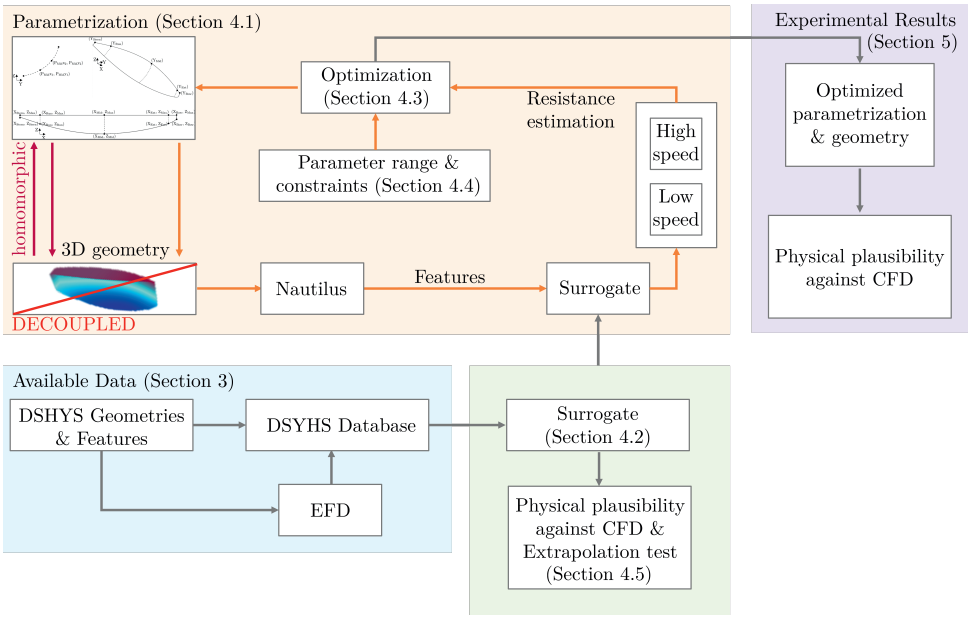


Figure 4.2: Proposed approach to vessel hull optimization.

tive enough to allow for effective and physically plausible predictions of the resistance associated with a particular hull [69, 134, 198, 199]. This decoupling is a fundamental and key contribution to our approach. In fact, the features extracted by Nautilus should not meet the requirement of the geometry parametrization to be homomorphic. This, on one hand, facilitates the ability to create a rich and informative features set that can be used to predict the hull resistance via DDMs without any human intervention (using Nautilus). On the other hand, the homomorphic parametrization of the geometry just needs to focus on the parameters to optimize during the optimization step reducing its complexity and minimizing the human intervention in those cases when new parent hulls need to be covered. This decoupling reduces the original complex and constrained problem into two simpler ones.

Thanks to the decoupled approach to the parametrization, which is able to cover multiple parent hulls, we can train DDMs based on the already available EFD of the DSYHS requiring no additional EFD or CFD. However, CFD has been used to check the physical plausibility of the trained DDMs in both synthetic extrapolating scenarios inside the DSYHS and also with a more realistic test outside the DSYHS. For the first case, we defined three, increasingly challenging, extrapolation cases by removing part of the EFD during the DDMs training phase and using those data for testing purposes

- Leave One Velocity Out (LOVO) where we remove all the EFD corresponding to a particular velocity;
- Leave One Geometry Out (LOGO) where we remove all the EFD corresponding to a particular geometry (variation of a particular hull);
- Leave One Series Out (LOSO) where we remove all the EFD corresponding to a

particular series (all variations of a particular parent hull).
For the more realistic test outside the DSYHS, we rely on all but one series belonging to DSYHS to train the DDM, and then we tested it with variations of a particular parent hull that was not used to train the DDM and explore geometric parameters $\delta\%$ larger than the ones covered by the DSYHS.

The proposed surrogate (tested in terms of different extrapolating scenarios and physical plausibility against CFD) is exploited (with minimal levels of human supervision to define the parameters range and constraints) by an optimizer (chosen according to the best options in the literature) to search for the optimal parameters configuration, and retrieve the associated optimized geometry. In particular, we will search for the Pareto front in terms of resistance at high and low speeds. Furthermore, we show that it is possible to optimize the hull resistance by exploring geometric parameters beyond the boundaries of the DSYHS and validating the results via state-of-the-art CFD.

The rest of the paper is organized as follows. Section 4.1 describes the available data; Section 4.2 outlines the proposed methodology; Section 4.3 contains the results; and finally, Section 4.4 concludes the work.

4.1 DSYHS Database

In this section we will describe the data that we will exploit in this study. In particular, we leverage the DSYHS database [168] (available upon request to the Delft University of Technology Ship Hydromechanics Laboratory[2]) which has been exploited in a number of works [48, 69, 134, 198, 199].

In [77] the authors present the original series of the DSYHS which included 22 systematically varied sailing yacht hulls, alongside a polynomial expression they developed to determine the residual hull resistance in terms of the hull geometry, over a range of Froude numbers. In the successive years many more experiments were added to the DSYHS database and now, to the best of the authors' knowledge, the DSYHS database is currently the largest collection of sailing yacht EFD in the world.

The current DSYHS database contains the hull collections for Series $1 \div 7$ ($S_1 \div S_7$) where S_5 does not exist in the database. The 6 series, composed of parent hulls and their derivatives, are in model scale (which is the scale which the experiments were performed at) and span approximate lengths of $2.100 \div 2.500$m, widths of $0.440 \div 0.660$m, and depths of $0.270 \div 0.350$m. Table 4.1 shows the geometric boundaries of each series of the DSYHS. From these 6 Series, namely the parent hulls, 54 different geometries G have been derived. For each geometry, the total resistance R_t over a range of speeds v had been retrieved via EFD. A visual representation of this description is reported in Figure 4.3. The total number of EFD in the DSYHS dataset is 702.

The 54 geometries contained in the DSYHS are described through the use of hydrostatic coefficients common for naval architecture applications, see Table 4.2 for details. Note that, for a general geometry, the parameters reported in Table 4.2 can be easily retrieved with Nautilus[1].

[2] http://dsyhs.tudelft.nl/dsyhs.php

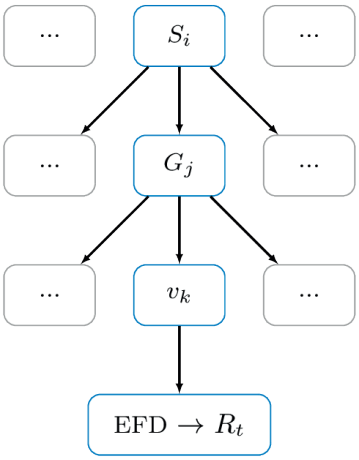


Figure 4.3: The database contains the hull collections for Series $1 \div 7$ ($S_1 \div S_7$) where S_5 does not exist in the database. From these 6 Series, namely the parent hulls, 54 different geometries G have been derived. For each geometry, the total resistance R_t over a range of speeds v had been retrieved via EFD.

Table 4.1: Geometric boundaries of each series of the DSYHS.

Series	**Geometries**	**Length** [m]	**Breadth** [m]	**Depth** [m]	**Volume** $\times 10^{-3}$[m^3]	**Draft** [m]
S_1	22	1.94÷2.30	0.51÷0.68	0.29÷0.36	37.60÷37.61	0.13÷0.15
S_2	6	2.31÷2.35	0.50÷0.66	0.23÷0.35	37.53÷37.60	0.11÷0.15
S_3	11	2.28÷2.40	0.48÷0.80	0.20÷0.28	37.53÷37.53	0.11÷0.11
S_4	9	2.09÷2.38	0.37÷0.74	0.27÷0.34	30.92÷37.53	0.11÷0.12
S_6	3	2.42÷2.43	0.66÷0.66	0.30÷0.35	30.92÷30.92	0.12÷0.12
S_7	3	2.51÷2.57	0.38÷0.45	0.21÷0.33	30.92÷30.92	0.12÷0.12

Table 4.2: Hydrostatic coefficients provided in the DSYHS to describe the geometries.

Parameter	**Symbol**
Length of waterline	L_{wl}
Breadth of waterline	B_{wl}
Draft of canoe body	T
Volume of canoe body	∇
Longitudinal center of buoyancy	LCB
Longitudinal center of flotation	LCF
Area of waterplane	A_w
Area of cross-section	A_x
Wetted surface area of canoe body	S
Block coefficient	Cb
Midship area coefficient	Cm
Prismatic coefficient	Cp
Waterplane area coefficient	Cw

4.2 Methodology

In this section, we will deepen the description of the methodology we propose starting from the schema presented in Figure 4.2.

In particular, Section 4.1 already focused on the available data, while the following aspects of the methodology will be the subjects of this section:

- the development of the surrogate to estimate R_t based on the parameters reported in Table 4.2 that can also be retrieved with Nautilus[1] for any hull geometry (Section 4.2.1);
- the validation in different extrapolating scenarios and the physical plausibility against CFD of the surrogate (Section 4.2.2);
- the homomorphic parametrization of the hull and the parameters range and constraints generating the parameter space (Section 4.2.3);
- the optimization framework which searches in the homomorphic parameters space simultaneously optimizing R_t for both a high v^{High} and low v^{Low} estimated with the surrogate (Section 4.2.4);
- the verification of the physical plausibility against CFD of the geometries on the Pareto front generated by the optimizer (Section 4.2.5).

Note that, with the aid of the proposed decoupling strategy between the parametrization exploited by the optimizer and the one exploited by the surrogate, given a point in the homomorphic parameters space it is possible to extract the input of the surrogate with Nautilus[1] and estimate the R_t for v^{High} and v^{Low} with minimal computational requirements making the optimization fast and cheap to perform.

4.2.1 Surrogate development

The problem of predicting R_t based on the parameters reported in Table 4.2 and the velocity v can be mapped to a typical regression problem by Machine Learning [202, 206].

The no-free-lunch theorem [4] guarantees us that, in order to find the best algorithm for a particular application, it is necessary to test multiple algorithms. In our case, we will test 4 state-of-the-art algorithms[3] [72, 231]: Random Forests (RF) [21, 171], XGBoost [28], Kernel Ridge Regression (KRR) [206], and the Extreme Learning Machine (ELM) [99, 100] namely a Single Layered Neural Network [17, 80] where the weights of the first layers have been randomly set reducing the computational burden of the training phase with minimal, if not absent, effect on accuracy.

In RF we need to tune the number of features to randomly sample from the whole features during each node of each tree creation n_f and the maximum number of elements in each leaf of each tree n_l. As RF performance improves by increasing the number of trees n_t, we set it to 1000 as a reasonably large number yet computationally tractable.

In XGBoost, we need to tune the learning rate of the gradient l_r, the max depth of each tree n_d, the minimum loss reduction m_l, the number of points to randomly sample from the whole training set for each tree creation n_b, and the number of features to randomly sample from the whole training set during the creation of each node for each tree n_f.

[3] Results in Kaggle `www.kaggle.com`, the most popular Machine Learning competition website, shows that these algorithms are the top winners.

Table 4.3: Hyperparameters and hyperparameters search space for all algorithms tested in this work, $d = 13$ denotes the number of features in the dataset (see Table 4.2).

Algorithm	Hyperparameters
RF	$n_f : \{d^{1/5}, d^{1/2}, d^{3/4}\}$ $n_l : \{1, 3, 5, 10\}$ $n_t : \{1000\}$
XGBoost	$l_r : \{0.01, 0.02, 0.03, 0.04, 0.05\}$ $n_d : \{3, 5, 10\}$ $m_l : \{0, 0.1, 0.2\}$ $n_b : \{0.6n, 0.8n, 1n\}$ $n_f : \{0.5d, 0.8d, 1d\}$
KRR	$\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$ $\gamma : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$
ELM	$h_l : \{2^5, 2^6, \cdots, 2^{16}\}$ $\lambda : \{10^{-6}, 10^{-5.8}, \cdots, 10^{3}\}$

In KRR we chose to rely on the Gaussian kernel for the reason described in [119], and then the regularisation hyperparameter λ and the kernel coefficient γ need to be tuned.

In ELM, we use the sigmoid activation function in the hidden layer and the linear activation in the output layer. Then we need to tune the number of hidden neurons h_l and then the regularisation hyperparameter λ on the weights of the last layer.

The summary of these hyperparameters with the associated search space is reported in Table 4.3.

Note that, the selection of the best performing algorithm and the best hyperparameters, will depend on the scenario under consideration and on two different metrics, namely accuracy and computational requirements (see Section 4.2.2).

The performances, in terms of accuracy, will be measured in accordance with different metrics: three quantitative (the Mean Absolute Error - MAE, the Mean Absolute Percentage Error - MAPE, and the Pearson Product-Moment Correlation Coefficient - PPMCC) [163] and one qualitative (the scatter plot actual versus predicted value) [192].

The performances, in terms of computational requirements, will be measured by means of time to build the model (Training Time) and time to make a prediction (Test Time). Since our surrogate will be leveraged in the optimization phase (see Section 4.2.4), the most important computational metric is the Test Time.

4.2.2 Surrogate validation and physical plausibility

In our work, we will study three different extrapolating scenarios based on the intrinsic hierarchy of the dataset. This will allow us to understand the extrapolation ability and the robustness of the different models described in the previous section (see Figure 4.4 for a visual representation):

- LOVO: where we remove all the EFD corresponding to a particular velocity. Since the EFD, for each geometry and each series, has been performed at different speeds, we create an histogram of the velocities with 16 bins. For the sake of replicability, one can find the final binning (with lower v_l and upper v_u bounds) reported in Table 4.4. The LOVO scenario, then, is actually leaving out all the EFD following in one of these bins. The scope of this scenario is to test the extrapolation ability of

Table 4.4: Histogram of the velocities with 16 bins for the DSYHS EFD.

Bin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\mathbf{v_l}$	0.00	0.30	0.62	0.94	1.26	1.58	1.90	2.22	2.54	2.86	3.18	3.50	3.82	4.14	4.46	4.78
$\mathbf{v_u}$	0.30	0.62	0.94	1.26	1.58	1.90	2.22	2.54	2.86	3.18	3.50	3.82	4.14	4.46	4.78	5.10

the model in terms of velocity, namely to estimate the resistance at a velocity never observed before in the dataset;

- LOGO: where we remove all the EFD corresponding to a particular geometry (variation of a particular hull). The scope of this scenario is to test the extrapolation ability of the model in terms of geometry, namely to estimate the resistance of a geometry never observed before in the dataset;
- LOSO: where we remove all the EFD corresponding to a particular series (all variations of a particular parent hull). The scope of this scenario is to test the extrapolation ability of the model in terms of series, namely to estimate the resistance for a series never observed before in the dataset.

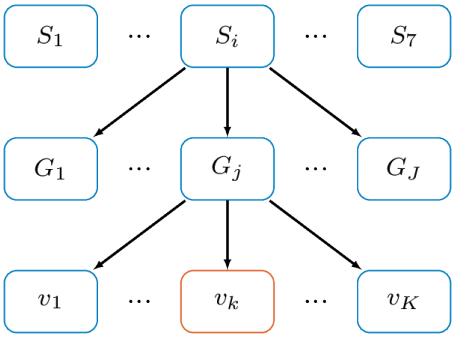
Note that the LOSO scenario is, in our work, the most interesting and useful one in practical applications. In fact, in practice, what we want to do is to generate geometry for a new, previously unexplored series, and this is precisely the scope of the LOSO scenario: we assume to have developed a few series, and we try to infer something for a new series that was previously unexplored.

What remains to be addressed is how to tune the hyperparameters of the different Machine Learning algorithms that we tested to generate the surrogate (see Section 4.2.1) and how to assess their final performance [169].

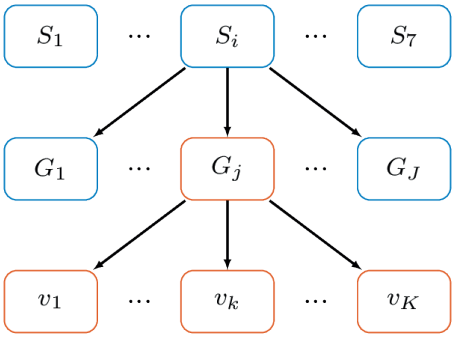
For what concerns the last point, the answer is easy. Based on the different scenarios (LOVO, LOGO, and LOSO) we have to split the data in Training $\mathcal{D}_n$ and Test $\mathcal{T}_t$ sets using the principle of the different extrapolating scenarios. For example, in the LOVO scenario, we put all the EFD corresponding to one of the histogram bins in $\mathcal{T}_t$ while the remaining ones are kept in the $\mathcal{D}_n$. Then we can use $\mathcal{D}_n$ to both train the model and select the associated best hyperparameters and use $\mathcal{T}_t$ to assess the performance of the final model. Repeating multiple times, this procedure will give us the average performance in the different scenarios.

Instead, for tuning the hyperparameters of the different Machine Learning algorithms, we proceeded as follows. We took $\mathcal{D}_n$ and split it into Learning $\mathcal{L}_l$ and Validation $\mathcal{V}_v$ sets considering the very same extrapolating scenario that we use for assessing the final performance. Then we train each model with $\mathcal{L}_l$ with many different hyperparameters configurations and measure its performance on $\mathcal{V}_v$ according to the MAE. Then we repeated the experiment multiple times and selected the hyperparameters' configuration which gives the best average MAE on the validation sets. Finally, we retrained the model with the selected best configuration of the hyperparameters on the whole $\mathcal{D}_n$ which is the model that will be used for testing purposes (see the previous paragraph).

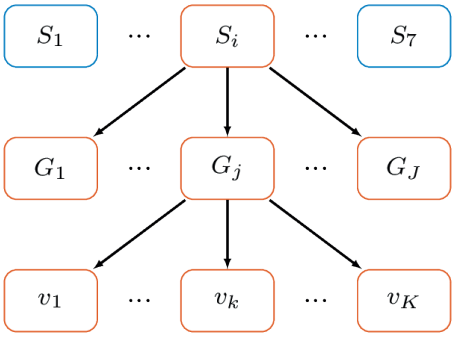
To ensure the physical plausibility of the proposed surrogate, we leveraged a state-of-the-art CFD model. For the DSYHS, EFD have been carried out by means of a large



(a) LOVO.



(b) LOGO.



(c) LOSO.

Figure 4.4: Visual representation of the three different extrapolating scenarios we investigated in this work based on the intrinsic hierarchy of the dataset. In particular we highlighted data hidden from the training phase and exploited just for testing purposes in orange.

experimental campaign carried out at the Delft University of Technology towing tank[4] and for this reason, they possess some level of uncertainty that cannot be removed.

Namely, the EFD were carried out using tangible models and the hydrostatic features (of Table 4.2) were determined by human observation. Therefore, to measure the quality of our surrogate we need to compare its performance against a baseline which, in our case, is a state-of-the-art CFD model.

Unfortunately, the CFD model is too computationally expensive to run for all the geometries and velocities in the databases. For this reason, we will compare our surrogate on a subset of them. In particular, we will consider the most challenging scenario, i.e., the LOSO, and we will perform the comparison between the CFD and the proposed surrogate models on the series which exhibit the largest deviation between the surrogate and the EFD results.

For the CFD model, the mesh generation, the computation of the solution, and the post-processing of results was carried out in Star CCM+[5] which is a state-of-the-art commercial CFD package. The simulation domain was created to satisfy the following constraints: the depth under the vessel was greater than twice the draft, the length of the domain after the vessel was longer than twice the length of the vessel, and the width of the domain was 50% larger than the length of the vessel. To reduce the computational demand of the simulation, the hull was divided symmetrically along the longitudinal axis and only half of the problem was simulated to assess the hydrodynamic performance. The CFD model is a finite volume based viscous RANS solver which can compute the hull resistance in various calm-water conditions by solving the underlying partial differential equations. For the problem at hand, a RANS solver was used considering the $k-\omega$ shear-stress turbulence model with wall functions [250]. The boundaries of the domain were set so the symmetric and far-field boundaries were considered as symmetry planes. The top, bottom, and inlet boundaries were considered as velocity inlets while the outlet boundary was considered as a pressure outlet. The volume of fluid technique was used to establish a free surface in the solution and solving the underlying equations with the volume fraction of both water and air [94]. To find the solution of the hull resistance, the vessel was simulated using the dynamic fluid body interaction module in Star CCM+[5] with two degrees of freedom (sink and trim), which is in line with the experimental campaign outlined in [77, 121, 122]. The simulation was set-up at a time-step of 0.001s and the behavior of the vessel simulated for a period of 60s. The solution of the simulation was then taken as the time averaged response over this period. The described CFD simulation was validated against the original EFD results for a number of geometries to ensure it could be used for the physical plausibility of the surrogate. A mesh coarsening procedure was carried out with $3 \cdot 10^5$, $9 \cdot 10^5$, and $3 \cdot 10^6$ cells respectively to ensure there was grid independence. Results using the highest fidelity mesh with $3 \cdot 10^6$ cells are presented in Section 4.3. Figure 4.5 shows the exploited mesh for the CFD simulations with the Star CCM+[5] package. The mesh included a surface mesh refinement on the vessel hull and on the boundaries of the domain in addition to volume mesh refinements around the hull, wake, and free surface.

[4] www.tudelft.nl/3me/over/afdelingen/maritime-and-transport-technology/research/ship-hydromechanics/facilities/towing-tank-no-1

[5] www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html

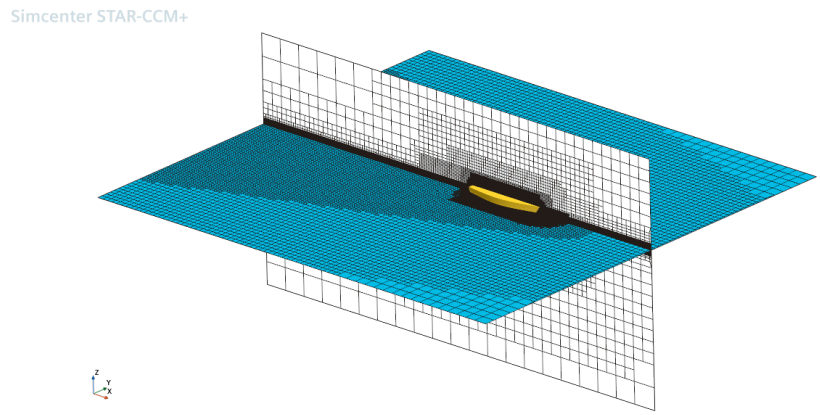


Figure 4.5: The mesh exploited for the CFD simulations with the Star CCM+[5] package. The mesh included a surface mesh refinement on the vessel hull and on the boundaries of the domain in addition to volume mesh refinements around the hull, wake, and free surface.

4.2.3 Hull parametrization and parameter ranges

In this section, we will describe the adopted homomorphic parametrization, together with the associated parameter range, that will be leveraged during the optimization phase (see Section 4.2.1) to search for the best hull, i.e., the hull that will exhibit the best R_t at v^{Low} and v^{High}. It is worth noting that this parametrization is decoupled from the one exploited in the definition of the surrogate (see Section 4.2.4) as described in the introduction (see Figure 4.2).

In particular, a parametric model for a sailing yacht hull [86, 109, 123, 173, 177–179] was developed with the Siemens NX[6] software leveraging on 32 parameters. The full list of parameters together with their description is reported in Table 4.5 and visualized in Figure 4.6.

The 32 parameters govern the hull geometry through the use of B-Spline curves [203], which in turn, drive the design of the yacht hull surface inside the parametric model. The parametrization is directly related to control points on the B-Spline curves which allows the parameters to be modified independently and ensures the desired homomorphic properties. Geometric constraints were imposed on the model to ensure G0 (positional) and G1 (tangential) continuity at the intersection between adjacent splines to assist in producing feasible designs. Additionally, the G2 (curvature) continuity was also applied to ensure a smooth surface was retrieved from the model [109]. Figure 4.6 includes: an example cross-section of the mid section (top left), an isometric view of the parametric hull (top right), and a planar view in the xz plane of the parametric hull (bottom). Parameters denoted with an x or z define features in the xz plane and parameters denoted with y define features in the xy plane. The parameters preceded by P refer to the B-spline control points in the yz plane of each section.

For what concerns the parameters ranges, they have been designed following this prin-

[6] www.plm.automation.siemens.com/global/en/products/nx/

Table 4.5: The 32 parameters together with their description characterizing the adopted homomorphic parametrization for a sailing yacht hull developed with the Siemens NX [6].

Symbol	Description	Symbol	Description
X_{Bow}	Length along the center line from the origin to the bow.	P^{y1}_{Entr}	Width from the center line to the first control point of entrance section.
Y_{Bow}	Width from the center line to the bow section at the maximum height.	P^{z1}_{Entr}	Height from Z_{Entr} to the first control point of entrance section.
Z_{Bow}	Height from the origin to the bottom of the bow section.	P^{y2}_{Entr}	Width from the center line to the second control point of entrance section.
Z_{Max}	Height from the origin to maximum height.	P^{z2}_{Entr}	Height from Z_{Entr} to the second control point of the entrance section.
X_{Entr}	Length along the center line from the origin to the entrance section.	P^{y1}_{Mid}	Width from the center line to the first control point of the first mid section.
Y_{Entr}	Width from the center line to the entrance section at the maximum height.	P^{z1}_{Mid}	Height from Z_{Mid} to the first control point of the first mid section.
Z_{Entr}	Height from the origin to the bottom of the entrance section.	P^{y2}_{Mid}	Width from the center line to the first second point of the first mid section.
X_{Mid}	Length along the center line from the origin to the first mid section.	P^{z2}_{Mid}	Height from Z_{Mid} to the second control point of the first mid section.
Y_{Mid}	Width from the center line to the first mid section at the maximum height.	P^{y1}_{Run}	Width from the center line to the first control point of the run section.
Z_{Mid}	Height from the origin to the bottom of the first mid section.	P^{z1}_{Run}	Height from Z_{Run} to the first control point of the run section.
X_{Run}	Length along the center line from the origin to the run section.	P^{y2}_{Run}	Width from the center line to the second control point of the run section.
Y_{Run}	Width from the center line to the run section at the maximum height	P^{z2}_{Run}	Height from Z_{Run} to the second control point of the run section.
Z_{Run}	Height from the origin to the bottom of the run section.	P^{y1}_{Stern}	Width from the center line to the first control point of the stern section.
X_{Stern}	Length along the center line from the origin to the stern section.	P^{z1}_{Stern}	Height from Z_{Stern} to the first control point of the stern section.
Y_{Stern}	Width from the center line to the stern section at the maximum height.	P^{y2}_{Stern}	Width from the center line to the second control point of the stern section.
Z_{Stern}	Height from the origin to the bottom of the stern section.	P^{z2}_{Stern}	Height from Z_{Stern} to the second control point of the stern section.

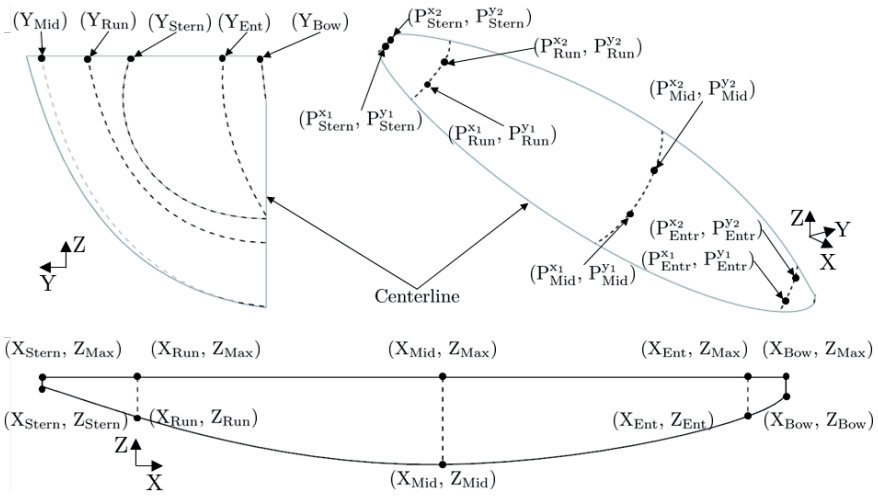


Figure 4.6: A visual representation of the 32 parameters characterizing the adopted homomorphic parametrization for a sailing yacht hull developed with Siemens NX [6].

ciple. First, for each parameter, we search for the minimum and maximum value in a specific series S_i, i.e., the series that we want to optimize (see Section 4.2.4). Then we increased that range by $\delta\%$. This extrapolation is especially useful because, in practice, we want to be able to generate a geometry for a new, previously unexplored series rather than restrict ourselves to preexisting designs. In the experiments, we will show that the $\delta = 30\%$ is the limit threshold beyond which the surrogate starts to induce the optimizer into degenerate solutions. The parameter ranges extracted from the original 54 hulls belonging to the DSYHS database are reported in Table 4.6.

The proposed homomorphic parametrization does not succumb to the limitations of the current approaches, i.e., the need to re-parametrize each parent geometry, and is able to cover the whole DSHYS database and beyond (i.e., up to $\delta\%$ of the DSHYS).

4.2.4 Optimisation framework

In this section we will present the proposed approach to the search for the best hull in a series S_i, in terms of the best R_t at v^{High} and v^{Low}, in line with the referenced works [82, 158], leveraging the parametrization described in Section 4.2.3 and the surrogate described in Section 4.2.1.

Since it is not a fair trade-off between the resistance R_t and the volume of the hull ∇, i.e., having no submerged body will correspond to the case where the resistance is zero, and in line with the original investigations of [77, 121, 122], we are concerned about optimizing the relative resistance to the submerged volume (i.e., $\frac{R_t}{\nabla}$).

Again, in line with the referenced works [98, 156], the optimization problem is subject to a constraint to bound the volume according to a lower and upper boundary ∇_l and ∇_u respectively. ∇_l and ∇_u have been set by searching for the minimum and maximum value in a specific series S_i, i.e., the series that we want to optimize, because we aim to optimize the geometry of a hull that fits within a particular series i.e., conforms to the same volume

Table 4.6: The parameter ranges for the 32 geometric design parameters of the values extracted from the DSYHS database. The parameter ranges are reported in mm.

Parameter	Range	Parameter	Range
X_{Bow}	[1650, 2200]	P^{Y1}_{Entr}	[6, 43]
Y_{Bow}	[1, 62]	P^{Z1}_{Entr}	[31, 85]
Z_{Bow}	[156, 355]	P^{Y2}_{Entr}	[12, 71]
Z_{Max}	[196, 364]	P^{Z2}_{Entr}	[60, 165]
X_{Entr}	[1600, 2050]	P^{Y1}_{Mid}	[154, 349]
Y_{Entr}	[18, 97]	P^{Z1}_{Mid}	[63, 121]
Z_{Entr}	[22, 179]	P^{Y2}_{Mid}	[166, 368]
X_{Mid}	[800, 1050]	P^{Z2}_{Mid}	[127, 244]
Y_{Mid}	[173, 371]	P^{Y1}_{Run}	[72, 274]
Z_{Mid}	[−97, 4]	P^{Z1}_{Run}	[28, 79]
X_{Run}	[−200, 50]	P^{Y2}_{Run}	[128, 307]
Y_{Run}	[146, 327]	P^{Z2}_{Run}	[64, 163]
Z_{Run}	[29, 180]	P^{Y1}_{Stern}	[9, 248]
X_{Stern}	[150, 400]	P^{Z1}_{Stern}	[4, 71]
Y_{Stern}	[26, 292]	P^{Y2}_{Stern}	[17, 273]
Z_{Stern}	[38, 282]	P^{Z2}_{Stern}	[7, 139]

constraints.

At this point, we can formalize our problem as follows

$$\begin{aligned} \min_{\mathbf{p}} \quad & \left\{ \frac{R_t(\mathbf{p}, v^{\text{High}})}{\nabla(\mathbf{p})}, \frac{R_t(\mathbf{p}, v^{\text{Low}})}{\nabla(\mathbf{p})} \right\}, \\ \text{s.t.} \quad & \nabla_l \leq \nabla(\mathbf{p}) \leq \nabla_u, \\ & \mathbf{p}_l(\delta) \leq \mathbf{p} \leq \mathbf{p}_u(\delta), \end{aligned} \tag{4.1}$$

where $\mathbf{p}$ is the vector of the 32 parameters of the homomorphic parametrization of Table 4.5, $\mathbf{p}_l(\delta)$ and $\mathbf{p}_u(\delta)$ are their lower and upper bounds of the parameters as a function of δ, $\nabla(\mathbf{p})$ is the volume of the hull we want to optimize as a function of $\mathbf{p}$ estimated with Nautilus[1], ∇_l and ∇_u are the upper and lower bound of $\nabla(\mathbf{p})$. Finally, $R_t(\mathbf{p}, \cdot)$ is the total resistance as a function of $\mathbf{p}$ and the velocity (computed at v^{High} and v^{Low}) estimated via the surrogate described in Section 4.2.1 but where $\mathbf{p}$ induces the geometry and, based on the geometry, Nautilus[1] estimates the quantities of Table 4.2 that together with the velocity are the actual inputs of the surrogate.

Problem (4.1) is a non-linear non-linearly constrained multi-objective optimization problem that is hard to optimize in practice.

The first step toward the solution of Problem (4.1) is to reformulate the problem as a single objective one. For this purpose we will rely on a classical approach: replace the multiple objectives with a weighted sum of the different objectives (changing the sign in

Table 4.7: Parameters setting for the optimization algorithm exploited to solve Problem (4.2).

Algorithm	Matlab Function	Parameter	Value
EA	ga	PopulationSize	5000
		MaxGenerations	200
		CrossoverFraction	0.8
		EliteCount	1
		Multi start (manually implemented)	10

front to the objective so as to have all minimization or maximization) [62]

$$\begin{aligned} \min_{\mathbf{p}} \quad & \lambda \frac{R_t(\mathbf{p}, v^{\text{High}})}{\nabla(\mathbf{p})} + (1-\lambda)\frac{R_t(\mathbf{p}, v^{\text{Low}})}{\nabla(\mathbf{p})}, \\ \text{s.t.} \quad & \nabla_l \leq \nabla(\mathbf{p}) \leq \nabla_u, \\ & \mathbf{p}_l(\delta) \leq \mathbf{p} \leq \mathbf{p}_u(\delta), \end{aligned} \tag{4.2}$$

where $\lambda \in [0, 1]$ defines the importance of the different objectives, i.e., for $\lambda \to 1$ we care more about $R_t(\mathbf{p}, v^{\text{High}})$ than $R_t(\mathbf{p}, v^{\text{Low}})$ and vice-versa for $\lambda \to 0$. Solving Problem (4.2) for different values of λ allows for the creation of the so-called Pareto frontier in a computationally efficient way [62].

Problem (4.2) is a non-linear non-linearly constrained optimization problem. In order to solve this problem different approaches can be exploited [127]. In the literature, there are a number of state-of-the-art algorithms available that are able to deal with this problem, e.g., gradient descent [191], swarm [183], and evolutionary [226]. A series of no-free-lunch theorems [252] ensure us that there is no way to choose a-priori the best optimization algorithms for a particular problem and the only option is to empirically test multiple approaches verifying which is actually the best one. As a consequence, to the best of the authors' knowledge and according to the literature on the subject [36, 70, 82, 140, 156], we opt for the Evolutionary Algorithm (EA) as it showed to be the best approach for these class of problems. In particular, we relied on an EA-based optimization framework built in MATLAB[7] using the function ga which is a variant implementation of the NSGA-II [49, 153] Genetic Algorithm. Moreover, we customize the optimizer adding a multi-start approach, running the algorithm multiple times keeping the best solution found in the different starts. For the sake of repeatability, Table 4.7 reports the parameters' set that empirically produced the best results in the paper.

4.2.5 Optimisation framework physical plausibility

In this section, we will present the proposed approach to demonstrate the physical plausibility of the solution (i.e., hull geometry) retrieved by solving Problem (4.1) through Problem (4.2) with different λ (see the previous section).

First, we need to better specify our definition of physical plausibility. In particular, in this work, we consider the ability of the optimizer to find non-degenerate geometries, namely geometries that in EFD will exhibit $R_t(\mathbf{p}, v^{\text{High}})$ and $R_t(\mathbf{p}, v^{\text{Low}})$ far away from

[7] https://mathworks.com/products/matlab.html

the one suggested by the optimizer. Such geometry is then considered non-physically plausible. This outcome may happen for two main reasons, which are also connected

- the first one is because $R_t(\mathbf{p}, v^{\text{High}})$ and $R_t(\mathbf{p}, v^{\text{Low}})$ inserted in Problems (4.1) and (4.2) are not the real resistances but a surrogate characterized by no infinite precision and limited extrapolation abilities (this has been already tested in Section 4.2.2). As a consequence, during exploration, the EA can spot false minima induced by the imprecision and the extrapolation limitations of the surrogate model
- the second one is that the parameter space defined by $\mathbf{p}_l(\delta)$ and $\mathbf{p}_u(\delta)$, namely by δ is too large, requesting the optimizer to search within a parameter space that has more risk of imprecise extrapolation of the surrogate.

For this reason, analogously to what has been done for the surrogate in Section 4.2.2, we will test the geometries found by the optimizer with the Star CCM+[5] package checking the deviation between the estimated $R_t(\mathbf{p}, v^{\text{High}})$ and $R_t(\mathbf{p}, v^{\text{Low}})$ and the one identified by the surrogate and then the optimizer. In the CFD simulation based on the Star CCM+[5] package, we exploited the same setting described in Section 4.2.2.

4.3 Experimental results

In this section, we will report the results of applying the methodology described in Section 4.2 to solve the problem faced in this work using the data described in Section 4.1.

Specifically, we performed the following experiments

- in Section 4.3.1 we tested the quality of the surrogate model in the different extrapolating scenarios (LOVO, LOGO, and LOSO);
- in Section 4.3.2 we focused on the LOSO scenario, the most challenging and useful in practice, testing the physical plausibility of the results against the CFD;
- in Section 4.3.3 we tested the quality of the optimization framework on a particular series of the DSYHS showing that we can improve the current geometries with new designs that we tested using CFD to verify their physical plausibility.

All experiments have been run on the DelftBlue supercomputer at the Delft High-Performance Computing Center [50], which hosts 238 Compute nodes with a total of 476 Intel XEON E5-6248R 24C 3.0GHz CPUs and 192 GB of Memory per node.

4.3.1 Surrogate validation in extrapolation

In this section, we will report the performance of the surrogate models described in Section 4.2.1 using the validation approaches described in Section 4.2.2 in the different extrapolating scenarios. In particular, we will compare the results of the different algorithms employed to build the surrogate (RF, XGB, KRR, and ELM) on the different extrapolating scenarios (LOVO, LOGO, and LOSO) using different metrics. For the metrics, we measured the accuracy with both quantitative (MAE, MAPE, and PPMCC) and qualitative (scatter plot) measures and the computational requirements (Training Time and Test Time).

Table 4.8 reports for all algorithms employed to build the surrogate (RF, XGB, KRR, and ELM) and for all the different extrapolating scenarios (LOVO, LOGO, and LOSO) the different metrics employed to evaluate the performance (MAE, MAPE, PPMCC, Training Time, and Test Time). Figure 4.7, instead, reports the scatter plot for the best algorithm in each scenario (ELM for LOVO and KRR for LOGO and LOSO) where we considered just

Table 4.8: Surrogate Models Validation in the Extrapolating Scenarios: metrics employed to evaluate the performance (MAE, MAPE, PPMCC, Training Time, and Test Time) for all algorithms employed to build the surrogate (RF, XGB, KRR, and ELM) and for all the different extrapolating scenarios (LOVO, LOGO, and LOSO).

Scenario	Surrogate	Accuracy			Time	
		MAE [N]	MAPE [%]	PPMCC [–]	Training 10^3 [s]	Test 10^{-6} [s]
LOVO	RF	2.56 ± 1.14	0.96 ± 0.62	0.89 ± 0.09	2.0 ± 0.1	2.2 ± 0.2
	XGB	2.35 ± 1.07	0.89 ± 0.63	0.96 ± 0.02	1.8 ± 0.1	5.0 ± 0.3
	KRR	4.75 ± 5.26	2.14 ± 2.77	0.85 ± 0.19	0.1 ± 0.1	9.6 ± 7.7
	ELM	2.28 ± 0.82	1.15 ± 0.45	0.94 ± 0.03	0.9 ± 0.0	5.2 ± 0.7
LOGO	RF	2.22 ± 1.39	0.12 ± 0.05	0.99 ± 0.00	1.9 ± 0.0	5.0 ± 0.3
	XGB	1.65 ± 0.97	0.12 ± 0.04	0.99 ± 0.00	1.6 ± 0.0	5.2 ± 0.4
	KRR	0.94 ± 0.21	0.48 ± 0.09	0.99 ± 0.01	0.5 ± 0.0	7.5 ± 0.5
	ELM	1.67 ± 0.35	0.63 ± 0.09	0.99 ± 0.00	5.9 ± 0.0	4.5 ± 0.5
LOSO	RF	2.84 ± 1.67	0.16 ± 0.06	0.99 ± 0.01	0.5 ± 0.0	1.2 ± 0.3
	XGB	2.33 ± 1.47	0.14 ± 0.08	0.99 ± 0.00	0.4 ± 0.0	4.8 ± 0.3
	KRR	1.83 ± 1.07	0.11 ± 0.03	0.99 ± 0.00	0.1 ± 0.0	7.6 ± 5.4
	ELM	3.11 ± 1.99	0.16 ± 0.07	0.96 ± 0.01	0.1 ± 0.0	1.5 ± 0.1

the Accuracy as a metric since the Test time differences are negligible.

From Table 4.8 and Figure 4.7 it is possible to observe that

- as the complexity of the extrapolation scenario increases (i.e., from LOVO to LOSO) the average accuracy of the models, across all of the algorithms, decreases;
- the ELM is the best performing algorithm for the LOVO scenario, while the KRR is the best performing algorithm for the LOGO and LOSO scenarios;
- despite the fact RF was demonstrated as the best algorithm overall in terms of Test Time, differences with the other methods are negligible for our application (well below fractions of milliseconds);
- final performance both in terms of accuracy (well below 1% of error) and Test Time (less than 10^{-5} [s]) even in the most challenging scenario (LOSO) make these surrogates perfect to be employed inside and automatic optimization framework (see Section 4.2.4). In fact, in order to reach this level of accuracy, usually a CFD simulation is required, but the same prediction takes around 1 hour with CFD.

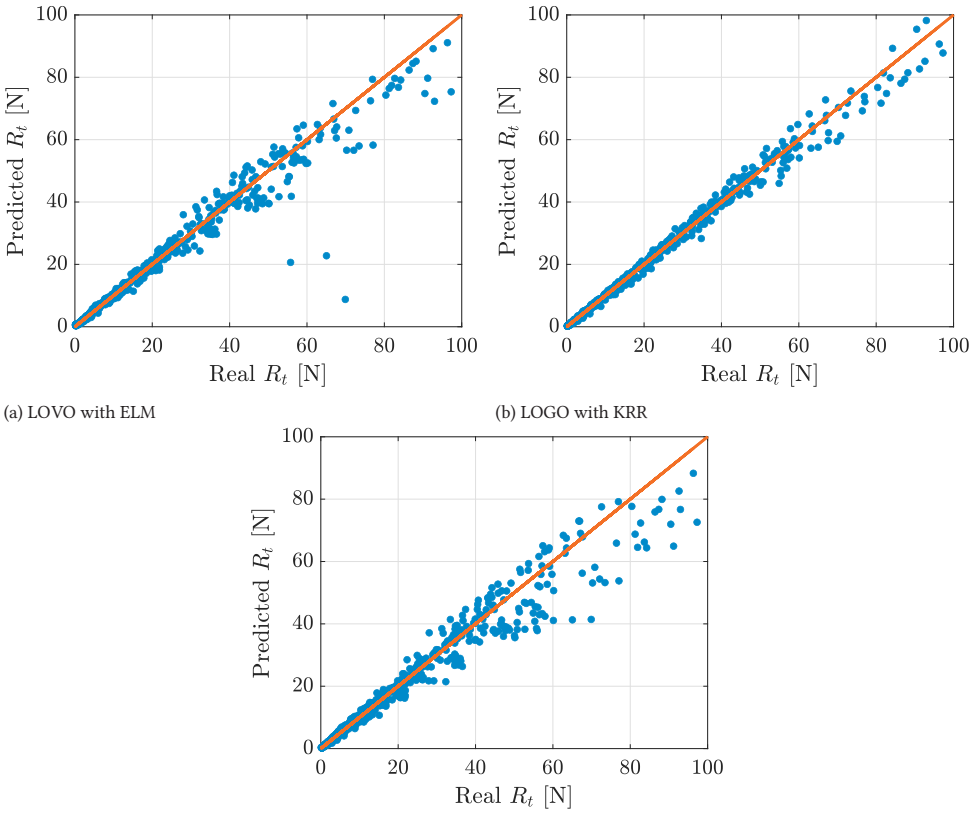
4.3.2 Surrogate validation and physical plausibility in the LOSO scenario

In this section, we will deepen the analysis of the performance of the best algorithm identified in Section 4.3.1 for the LOSO scenario (KRR), because, in practice, this is the most interesting scenario. In fact, in practice, what we want to do is to generate geometry for a new, previously unexplored series.

Let us start by validating the quality of the model on the different series and on the different geometries.

Table 4.9 reports, for the KRR in the LOSO scenario, the different metrics of accuracy (MAE, MAPE, and PPMCC) for each of the series. Instead, Figure 4.8 reports, for the KRR in the LOSO scenario, the scatter plot for each of the series.

From Table 4.9 and Figure 4.8 it is possible to observe that the surrogate performs



(a) LOVO with ELM

(b) LOGO with KRR

(c) LOSO with KRR

Figure 4.7: Surrogate Models Validation in the Extrapolating Scenarios: scatter plot for the best algorithm in each scenario (ELM for LOVO and KRR for LOGO and LOSO) considering just the Accuracy as a metric since the Test time differences are negligible (see Table 4.8).

Table 4.9: Surrogate Validation in the LOSO Scenario: metrics of accuracy (MAE, MAPE, and PPMCC) for each of the series of the KRR (the best algorithm identified in Section 4.3.1).

Series	**Accuracy**		
	MAE $[N]$	**MAPE** $[\%]$	**PPMCC** $[-]$
S_1	3.47 ± 0.94	1.26 ± 0.29	0.99 ± 0.00
S_2	3.13 ± 1.85	0.61 ± 0.24	0.99 ± 0.02
S_3	0.62 ± 0.34	0.62 ± 0.35	1.00 ± 0.00
S_4	2.28 ± 0.49	0.88 ± 0.17	1.00 ± 0.00
S_6	0.86 ± 0.38	0.49 ± 0.11	1.00 ± 0.00
S_7	0.62 ± 0.18	0.37 ± 0.15	1.00 ± 0.00

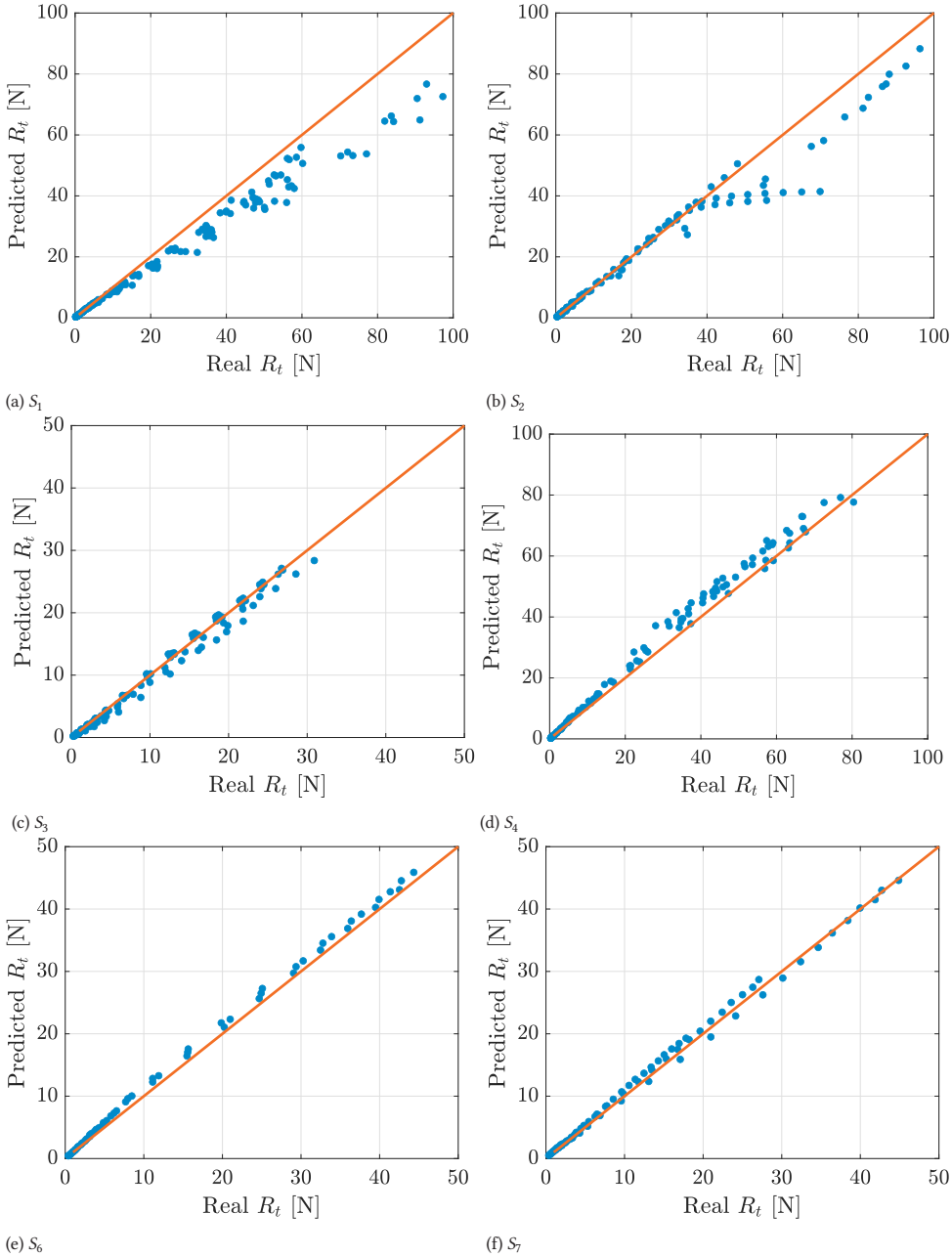


Figure 4.8: Surrogate Validation in the LOSO Scenario: scatter plot for the KRR (the best algorithm identified in Section 4.3.1) for each of the series (see Table 4.8).

Table 4.10: Surrogate Validation in the LOSO Scenario: different metrics of accuracy (MAE, MAPE, and PPMCC) for the best (i.e., the one exhibiting the smallest error) and worst (i.e., the one exhibiting the most significant error) geometries in each of the series for the KRR in the LOSO scenario.

Series	**Best**	**Accuracy**			**Worst**	**Accuracy**		
		MAE $[N]$	**MAPE** $[\%]$	**PPMCC**		**MAE** $[N]$	**MAPE** $[\%]$	**PPMCC**
S_1	G_7	1.50	0.60	0.99	G_6	10.47	1.83	0.99
S_2	G_4	0.34	0.29	0.99	G_5	5.31	1.00	0.99
S_3	G_1	0.11	0.14	0.99	G_7	1.47	1.52	0.99
S_4	G_9	0.77	0.86	0.99	G_5	3.63	1.42	0.99
S_6	G_3	0.58	0.37	0.99	G_1	0.98	0.53	0.99
S_7	G_1	0.51	0.27	0.99	G_2	0.72	0.44	0.99

better on some series than on others. This is due to several reasons

- the performance of the surrogate decreases at higher speeds as less experiments have been performed at higher speeds. As a matter of fact, for S_1, S_2, and S_4, the poor performance is exhibited around Resistances in the range from 40–100N (see Figures 4.8a, 4.8b, and 4.8d);
- the poor performance for S_1 as the LOSO (Figure 4.8a) is related to the fact that there are significantly more geometries in this series than in any other (22 out of 47 geometries according to Table 4.1). Consequently, when we check S_1 in the LOSO scenario, we have very few geometries to learn our model.

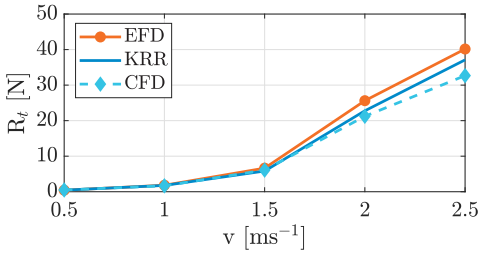
Let us continue this section with the test of the physical plausibility of the surrogate.

First, we have to look in detail at the performance of the surrogate in each geometry of the series. Since reporting all the errors for all the geometries of all the series is not meaningful, we decided to report in Table 4.10, for the KRR in the LOSO scenario, the different metrics of accuracy (MAE, MAPE, and PPMCC) for the best (i.e., the one exhibiting the smallest error) and worst (i.e., the one exhibiting the most significant error) geometries in each of the series.

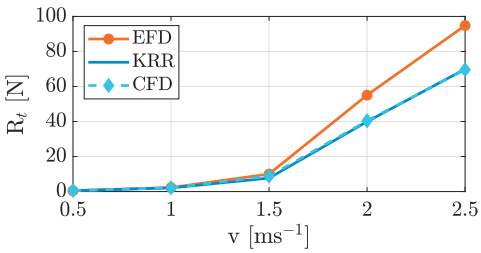
From Table 4.10, it is possible to observe how the gap between the best and the worst geometries can be significant. Nevertheless, large or small here is not a concept that we can define without having a baseline.

For this reason, in Figure 4.9, we reported, for the best and worst geometry in each of the series as in Table 4.10, the comparison between the EFD (the available data), the KRR surrogate (that we learned from the EFD in the LOSO scenario), and the CFD (using the StarCCM+[5] package as described in Section 4.2.5). Note that, for the EFD just few points are actually known and we linearly interpolate between them. For the CFD, we have the same issue since making one prediction, as pointed out in Section 4.3.1, takes a few hours. Instead, for the KRR, we can make prediction for a huge number of points since only fractions of milliseconds are needed (see Table 4.8).

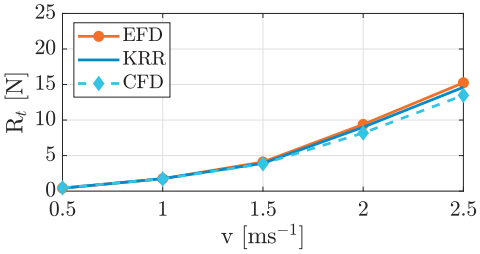
From Table 4.10 and Figure 4.9, we can observe that the deviation of the KRR-based surrogate from the EFD is, in terms of magnitude, similar, when not better, than the one of the CFD even when we consider the geometry in which the surrogate performs worse. Moreover, the resistance behaviour as a function of the speed is quantitatively aligned with the expectations. In conclusion, the KRR-based surrogate performance and physical plausibility can be considered at the level of a state-of-the-art CFD-based model at a



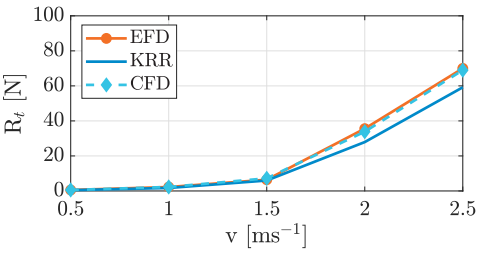
(a) S_1: G_7



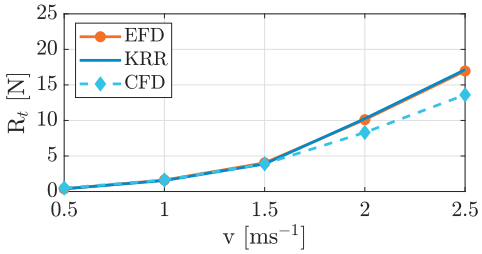
(b) S_1: G_6



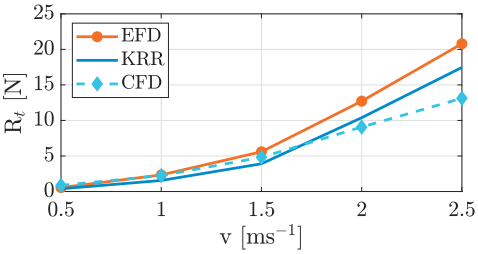
(c) S_2: G_4



(d) S_2: G_5

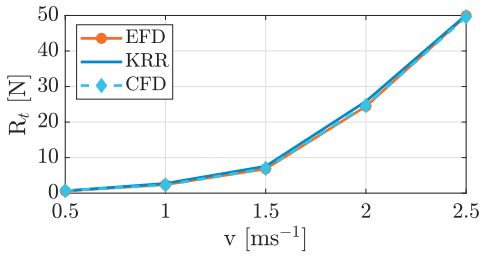


(e) S_3: G_1

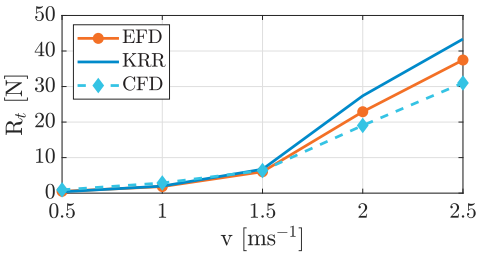


(f) S_3: G_7

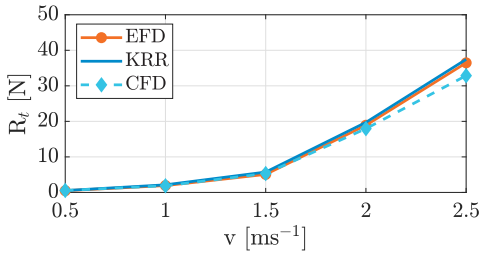
The figure continues on the next page ↓



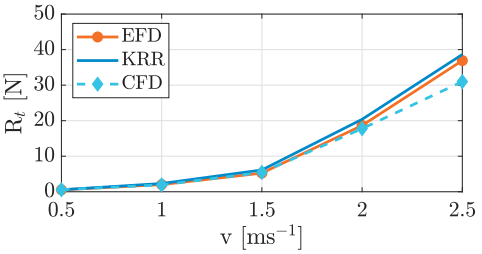
(g) S_4: G_9

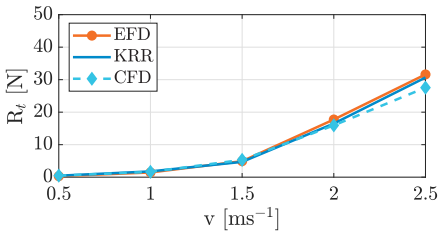


(h) S_4: G_5

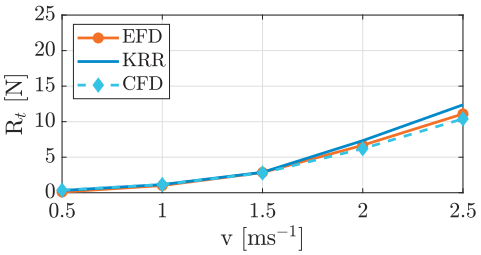


(i) S_6: G_3





(j) S_7: G_1



(k) S_7: G_2

Figure 4.9: Surrogate Physical Plausibility in the LOSO Scenario: comparison between the EFD (the available data), the KRR surrogate (that we learned from the EFD in the LOSO scenario), and the CFD (using the StarCCM+[5] package as described in Section 4.2.5) for the best and worst geometry in each of the series as in Table 4.10.

fraction of its computational requirements: from hours to a fraction of milliseconds.

4.3.3 Optimization framework validation and physical plausibility

At this point, we have empirically shown that the proposed parametrization and the surrogate are able to work well also in extrapolating scenarios matching the performance, in terms of accuracy and physical plausibility, of state-of-the-art CFD models at a fraction of their computational requirements. In this section, we will leverage this surrogate in the optimization framework proposed in Section 4.2.4, validating its performance by means of the approach described in Section 4.2.5.

For computational constraints (i.e., using the CFD too many times would result in months of simulations) in this section we limit the analysis to the optimization of a single series S_j. In order to have a realistic baseline (i.e., EFD data) we designed a specific experiment: we trained the surrogate with the EFD of all the series in the DSYHS except S_j simulating the need to design a vessel exactly in the missing series. In this way, the EFD data of S_j will function as a realistic baseline to compare with the results of our optimization. Note that, with this approach, we are actually using the surrogate as in the LOSO scenario. Then we solved the optimization problem of Section 4.2.4 using this surrogate as resistance predictor for different values of $\lambda \in [0, 1]$ and with the parameter range induced by the S_j (see Section 4.2.4) computing the Pareto frontier of the geometries. The Pareto frontier of the geometries is then compared with the EFD data of the S_j (where we linearly interpolated between the available data). Moreover, for each one of the geometries on the Pareto we computed the resistance at high and low speed with the CFD.

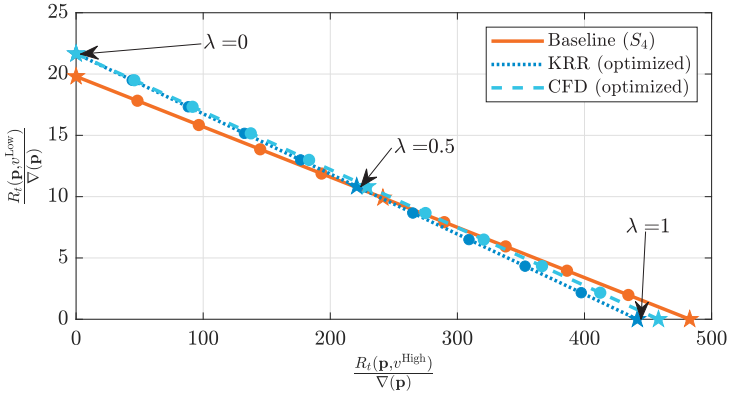
We set $S_j = S_4$: this choice is based on Table 4.9 as this is the series that exhibits approximately the average performance of the surrogate in the LOSO scenario (i.e., it is not the most challenging nor the simplest series to predict but is an average to challenging one). For S_4 the $\mathbf{p}_l(\delta)$ and $\mathbf{p}_u(\delta)$ are reported in Table 4.6 while $\nabla_l = 19 \cdot 10^{-3}\ m^3$ and $\nabla_u = 48 \cdot 10^{-3}\ m^3$. We reported the results for different values of $\delta \in \{10, 20, 30\}\%$ and $\lambda \in \{0, 0.1, \cdots, 1\}$ linearly interpolating between this value.

Figure 4.10 reports the Pareto frontier ($\frac{R_t(\mathbf{p}, v^{\text{High}})}{\nabla(\mathbf{p})}$ on the x-axis and $\frac{R_t(\mathbf{p}, v^{\text{Low}})}{\nabla(\mathbf{p})}$ on the y-axis) for different values of λ and δ together with the EFD data and the CFD validation as just described. Additionally, Figure 4.11 reports a comparison of the body plans[8] for the baseline geometry belonging to S_4 and the optimized ones with $\lambda = 1$ and $\delta \in \{10, 20, 30\}\%$. Setting $\lambda = 1$ implies that we prefer to minimise the resistance at v^{High}, representing a typical velocity for high-speed operations where we should observe the most significant differences in optimal performance.

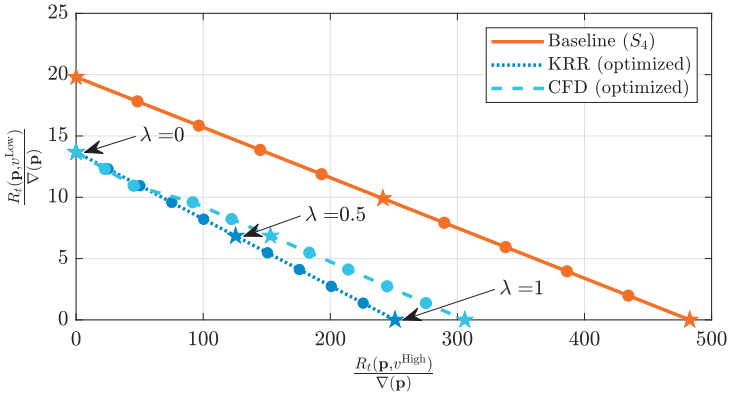
From Figures 4.10 and 4.11 we can observe that

- when δ is small (Figure 4.10a, $\delta = 10\%$) the optimization framework coupled with the surrogate is able to find geometries that match the performance of the one in S_4 without any a-priori knowledge of the geometries belonging to S_4. Nevertheless, it is worth noting how the geometry found by the optimizer (Figure 4.11a), even if having a similar performance, is quite different. This is due to the fact that the

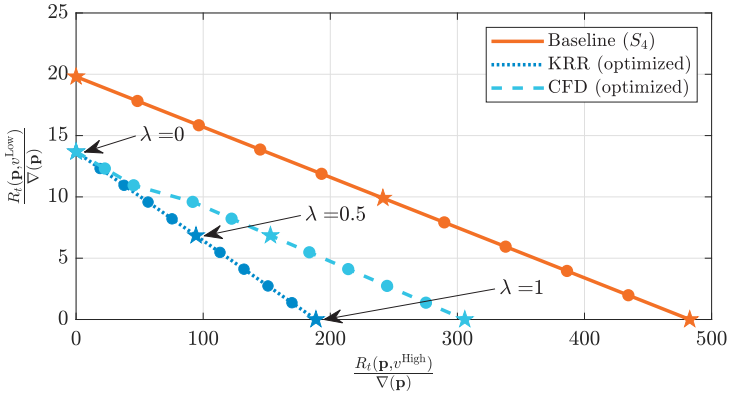
[8] The body plan is commonly used in naval architecture to display hull geometries and contains the set of transverse sections (the fore of the hull is on the left, and the aft on the right).



(a) δ = 10%

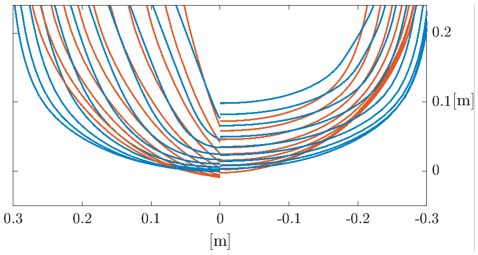


(b) δ = 20%

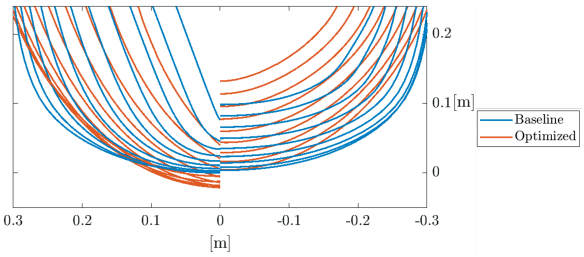


(c) δ = 30%

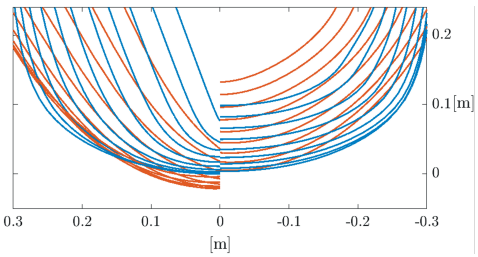
Figure 4.10: Optimization Framework Validation and Physical Plausibility: Pareto frontier ($\frac{R_t(\mathbf{p},v^{\text{High}})}{\nabla(\mathbf{p})}$ on the x-axis and $\frac{R_t(\mathbf{p},v^{\text{Low}})}{\nabla(\mathbf{p})}$ on the y-axis) for different values of λ and δ together with the EFD data and the CFD.



(a) $\delta = 10\%$



(b) $\delta = 20\%$



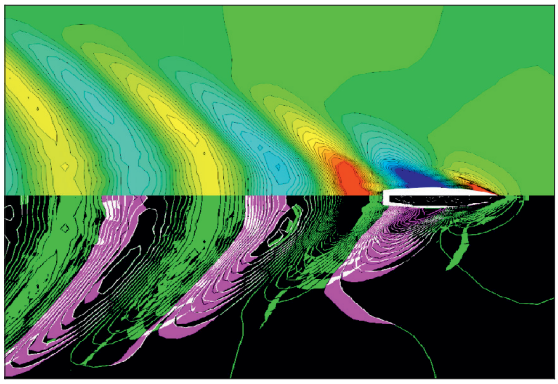
(c) $\delta = 30\%$

Figure 4.11: Optimization Framework Validation and Physical Plausibility: a comparison of the body plans[8] for the baseline geometry belonging to S_4 and the optimized ones with $\lambda = 1$ and $\delta \in \{10\%, 20\%, 30\%\}$.

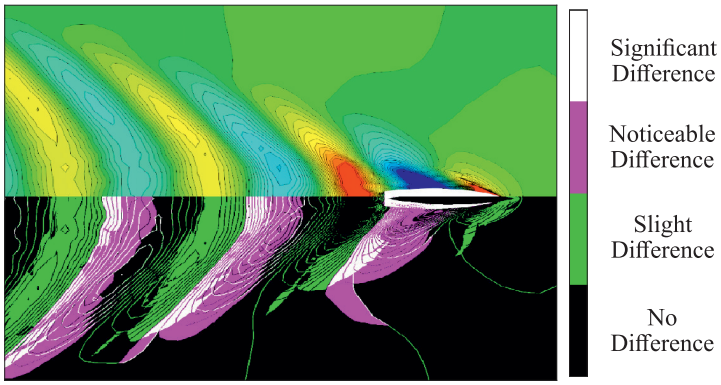
optimization problem is surely simplified, not taking into account all the realistic constraints that impact the design of a hull geometry (e.g., stability and seakeeping);
- when δ is a bit larger (Figure 4.10b, $\delta = 20\%$) the surrogate is able to exceed remarkably, according to the surrogate, the performance of the S_4 geometry. However, this is a bit optimistic when checking the resistance at high speed: when using the CFD to estimate the resistance of the geometry found with the surrogate there is a reduction of this performance gain which remains still remarkable. Also in this case note that the differences in the geometries (Figure 4.11b) stars to enlarge;
- when we further increase δ (Figure 4.10c, $\delta = 30\%$) the surrogate can exceed even more, according to the surrogate, the performance of the S_4 geometry. However, this is just a numerical artefact when checking the resistance at high speed due to the extrapolation limits of the surrogate. In fact, when using the CFD to estimate the resistance of the geometry found with the surrogate, there is a reduction of this performance that brings us back to the gain found when δ was smaller. Note that in this case the geometry (Figure 4.11b) is quite similar to the case of $\delta = 20\%$ (Figure 4.11b).

Finally, for the sake of completeness, a qualitative indicator of the quality of the optimized geometry with $\lambda = 1$ (for the same reasons as before) and $\delta \in \{10, 20, 30\}\%$ is reported in Figure 4.12 which shows the wave profile at v^{High} of the original S4 hull (top half) and the difference with the optimized parametric hulls (bottom half).

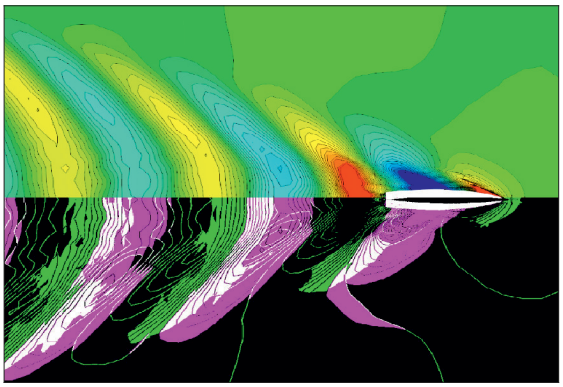
From Figure 4.12 we can observe that
- in all cases there is a noticeable difference between the original hulls and the optimized hulls;
- when $\delta = 10$ (Figure 4.12a) there is little significant difference (indicated by the lack of white color in the bottom half of the figure) which is expected due to the fact that the representation space is constrained around that of the original hull;
- when $\delta = 20$ or $\delta = 30$ (Figures 4.12b and 4.12c) there is a more significant difference between the original and optimized wave profiles (indicated by the presence of white color in the bottom half of the figures) which is in agreement with the results of Figure 4.10.



(a) $\delta = 10\%$



(b) $\delta = 20\%$



(c) $\delta = 30\%$

Figure 4.12: Optimization Framework Validation and Physical Plausibility: quality of the geometry generated in S_4 with $\lambda = 1$ and $\delta \in \{10, 20, 30\}\%$ using the wave profile at v^{High} of the baseline hull belonging to S_4 (top half) and the difference with the optimized one (bottom half).

4.4 Chapter Summary

In this work, we tackled the problem of vessel hull resistance optimization, which is crucial for achieving optimal performance and reducing environmental impact. First, we reviewed the current approach in the literature that mostly relies on a mix of human experience and DDMs: human experts define, via parametrization and parameter ranges, a series of geometries; a surrogate of the relationship between these parameters and the resistance, based on data from EFD or CFD, is built to interpolate within the defined parameter ranges; finally, the optimal parameters are found by optimizing, with more or less human intervention, the surrogate and used to retrieve the optimal geometry. Several limitations of the existing approaches were identified, including the need for human intervention in geometry parametrization and optimization, extensive computational efforts and costs, and limited ability to work beyond the specific settings. To overcome those limitations, and to the best of the authors' knowledge, for the first time in the literature, we proposed a parametrization able to accurately describe the entire DSYHS that was decoupled from the one needed to create the DDM. We showed that the DDM can be directly trained on the DSYHS EFD dataset, avoiding the need for new CFD or EFD customized for the specific problem, and match the performance of state-of-the-art CFD models even in extrapolating conditions (i.e., for geometries and parameter ranges beyond the boundaries used to construct the surrogate), with physical plausibility and minimal human intervention. Apart from the methodological contribution, we also validated our approach to developing DDMs on different and increasingly challenging extrapolating conditions with statistical methods using the DSYHS EFD dataset and for physical plausibility using state-of-the-art CFD models. We demonstrated the effectiveness of our proposal by showing that it is possible to optimize the hull resistance by exploring geometric parameters beyond the boundaries of the DSYHS.

5

Decoupling Hull Resistance Optimization for Computational Efficiency and Physical Plausibility

Accurate and computationally tractable prediction of calm-water resistance is essential for effective hull-form optimization. While Data-Driven Surrogates (DDSs) provide efficient inference during optimization, their construction is hindered by time-intensive Computational Fluid Dynamics (CFD) simulations or Experimental Fluid Dynamics (EFD) simulations, and their reliance on parametrization-specific design spaces. Building on the previous work done in Chapter 4, which tackled this challenge for sailing yacht hull forms, we expand our approach to more topologically complex hull forms. We propose a novel surrogate modeling approach that decouples geometry representation from performance prediction using hydrostatic and stability features as input, enabling transfer across parametrizations and parent hulls. This allows leveraging historical CFD datasets to construct DDSs and reduces new simulation costs. We validate the proposed pipeline across interpolation, extrapolation, and hybrid enrichment scenarios on three benchmark hulls (KCS, S175, DTC). Results show that the proposed decoupled approach has comparable accuracy to parametrization-based surrogates in interpolation settings, and superior performance in extrapolation scenarios when we enrich historical data sets with problem specific data. This approach represents a step towards improving the generalizability of DDSs for hull-form optimization, reducing the computational cost of building DDSs, and improving the physical interpretability of DDS input features.

This chapter is partly based on J. M. Walker, A. Coraddu, and L. Oneto. Decoupled Hull Resistance Prediction: A Computationally Aware and Physically Plausible Approach for Data-Driven Surrogates. In *Proceedings of the 12th Conference on Computational Methods in Marine Engineering, Edinburgh, Scotland,* 2025.

Designing efficient ship hulls under evolving operational, economic, and environmental conditions remains a central challenge in naval architecture. Classical designs are increasingly misaligned with modern demands shaped by changing trade routes, emission regulations, volatile fuel prices [193, 223, 254], alternative propulsion systems [67], and mission-specific constraints. To remain competitive, ship designers must generate novel hull forms or refine existing ones to meet a growing set of Key Performance Indicators (KPIs), including regulatory compliance, safety, and energy efficiency.

A critical factor optimizing the energy efficiency is the accurate prediction of calm-water resistance (R_t). High-fidelity Computational Fluid Dynamics (CFD) techniques deliver dependable predictions; however, they impose an excessive computational load for the scales needed for use within the context of hull-form optimization. As a result, Data-Driven Surrogates (DDSs) have emerged as effective approximators trained on CFD data, offering fast inference and low computational cost in the forward phase (i.e., predictions in milliseconds or less) [238].

The DDS-based hull-form optimization (DSO) workflow [238] typically consists of the following four sequential steps:

1. **Parametrization and KPI definition**: Establishing geometric parameters and performance indicators to characterize the hull design;
2. **Data sampling and surrogate model training**: Generating design samples (e.g., via DoE or Latin Hypercube) and constructing surrogate models to approximate performance metrics;
3. **Numerical optimization**: Solving the optimization problem using surrogate models to identify promising hull forms;
4. **Expert validation and feedback**: Evaluating selected designs through expert judgment and high-fidelity simulations to refine the design loop.

In Step 1, the starting point for the optimization is chosen (i.e., a parent hull-form), parametrized with a finite-dimensional modification vector, and the KPI to be optimized is selected (e.g., R_t).

In Step 2, a limited number of example hull-form variations are generated, and their performance is assessed using CFD to construct a representative database. A DDS is trained to learn the input-output relationship between the hull-form modification parameters and R_t, producing a surrogate with negligible computational load during optimization. This makes the DDS ideally suited for Step 3, where the numerical optimization is performed to optimize the modification parameters and identify the best-performing hull variant. Finally, in Step 4, the optimal hull is validated using high-fidelity CFD or expert review, with feedback optionally passed to earlier stages.

However, the current DSO workflow exhibits two fundamental limitations. First, DDSs are built on features derived from specific parametrization schemes such as Free-Form Deformation (FFD) or Radial Basis Function (RBF) networks. This tightly couples each DDS to a specific parent hull and modification framework, limiting their reusability when changes are made to the parametrization. Changing the parametrization or hull family typically requires generating new data and retraining from scratch. Second, most DDSs are trained to interpolate. They perform well within the boundaries of the training data, but their predictions deteriorate rapidly when extrapolating to unseen geometries. This limits their usefulness in exploratory design, where optimal solutions may lie outside the

sampled parameter space.

Prior efforts to address these limitations have made partial progress. In [235, 239], the authors demonstrated that DDSs trained on historical CFD datasets can be reused for hull-form optimization by decoupling geometry modifications from KPI predictions in a simplified design scenario. Using the Delft Systematic Yacht Hull Series, they trained models on hydrostatics-derived inputs to predict calm-water resistance, showing promising extrapolation across systematic variations. However, the underlying hulls shared high topological similarity, limiting the generalizability of the findings.

In [69], authors developed DDSs using hydrostatics-based features across three hull series (79 total hulls). Although their dataset was broader, extrapolation was only tested on a limited number of unseen configurations, and again in the context of geometrically similar sailing yachts.

In [159], authors investigated the prediction of added-wave resistance for dissimilar hull types (e.g., container ships and tankers), enriching hydrostatics-based features with non-linear transformations informed by domain knowledge. Their DDSs achieved good alignment with CFD and experiments, though the focus was on added resistance rather than calm-water R_t.

Building on these insights, our work extends the applicability of hydrostatics-based DDSs to a higher-complexity problem, extrapolating predictions between benchmark parent hull-forms. In particular, to address the limitations of the current DSO workflow, we propose a DDS framework that is decoupled from the geometry modification parameters, allowing for a more computationally aware approach (because our approach is less data-hungry) and physically plausiblity (because our approach is trained on parameters which have a physical meaning). Rather than relying on the geometry modification parameters, we use hydrostatic and stability features, such as centers of buoyancy, waterplane area, and moments of inertia that are intrinsic to the hull geometry, to train the DDS. These features, such as buoyancy centers, waterplane area, or moments of inertia, are computable for any hull shape regardless of how it was generated. This allows the DDS to generalize across parent hull-forms, reuse historical CFD datasets, and remain valid when the parametrization scheme is modified.

We test the proposed decoupled approach to hull-form resistance prediction on three canonical hulls: KCS, S175, and DTC. For each, we generate CFD-evaluated datasets based on Free-Form Deformation (FFD) induced datasets, and extract the hydrostatic and stability descriptors from the geometry using the publicly available Nautilus code[1] [239]). We then train DDSs to predict R_t from these features under four evaluation settings: i) interpolation within a given parent hull's design space (benchmarked against the standard approach using the geometry modification parameters); ii) a 1-to-1 extrapolation learning from one parent hull-form and extrapolating to unseen hull families; iii) a leave-one-out extrapolation learning from two parent hull-forms and extrapolating to an unseen family; iv) a hybrid approach based on enriching the leave-one-out scenario with varying samples from the unseen family. To the best of the authors' knowledge, this is the first study to demonstrate that DDSs can be constructed from historical CFD datasets and adapted to new problems via data enrichment. The key rationale is to reduce the computational cost of training DDSs by leveraging existing data and requiring fewer new CFD samples than

[1] https://github.com/mai-lab-tud/nautilus

the conventional approach, which relies entirely on generating new CFD simulations for each optimization problem.

Our results show that the decoupled DDS framework based on the hydrostatic input matches the performance of traditional geometry modification parameter-based DDSs for resistance prediction in interpolation settings. In extrapolation scenarios, where conventional DDSs are not able to perform, we show that the DDSs developed with the decoupled approach are still able to perform well in optimizaztion tasks. Crucially, we show that with modest data enrichment (10–30% of new CFD simulations), we are able to significantly improve performance, and when data enrichment reaches 70% of new CFD simulations, we are able to match the performance of DDSs built 100% These findings demonstrate that embedding physical knowledge in the feature space—rather than in the parametrization—enables transferable, interpretable, and computationally efficient surrogate models, and can lower the computational cost of training DDSs by up to 30% by requiring less data.

The remainder of this paper is structured as follows: Section 5.1 describes the problem formulation and dataset construction; Section 5.2 presents the surrogate modeling approach and optimization framework; Section 5.3 reports and analyzes the experimental results; and Section 5.4 summarizes the conclusions and outlines directions for future work.

5.1 Problem and Data

We consider the problem of predicting calm water resistance R_t for a given hull geometry using DDS based on an input space that is decoupled from geometry modifications and is physically interpretable. Conventionally, the DDS input features are derived solely from the geometry modification parameters (e.g., FFD control points), which are specific to the parametrization scheme and parent hull. This makes the DDS sensitive to reparametrization and unable to generalize across different parent hulls. Instead, we propose to decouple geometry representation from performance prediction by using hydrostatic and stability features as input. By using these features, our DDS becomes parametrization-agnostic and is better suited for extrapolation and data reuse.

To test the proposed modifications to the DSO pipeline (see introduction to Chapter 5) we leverage three parent hull-forms commonly found in the literature (KCS [36, 71, 156], S175 [71], and DTC [61, 71, 228]). We generate the database by creating hull-form variations of each of the parent hull-forms. We employ the Free-Form Deformation (FFD) method [227] for this task, which provides an effective means to generate diverse designs while ensuring geometric continuity and feasibility. FFD approach overlays a control network, consisting of control points, onto the parent hull design, enabling the generation of candidate hull forms by displacing these control points.

Table 5.1: Parent Hulls and Main Particulars.

Name	λ[-]	T[m]	V[m^3]	Lwl[m]	B_{wl}[m]
KCS	31.599	0.342	1.649	7.356	1.020
S175	40	0.238	0.364	4.382	0.621
DTC	59.407	0.244	0.826	6.090	0.858

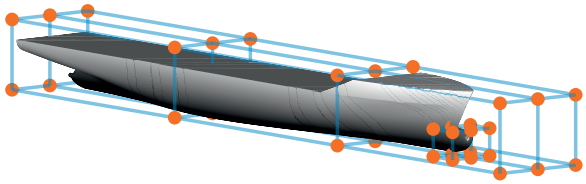


Figure 5.1: FFD parametrization grid.

We use an in-house developed FFD tool based on subdivision surfaces [34]. The control network consists of two overlapping subnetworks: a global network covering the entire hull with 24 control points, and a local network focused on the bulbous bow with 12 control points. The hierarchical structure of the networks increases the density of control points around the bulbous bow to better capture its intricate geometry.

A symmetry constraint along the XZ plane reduces the number of independent control points to 12. The degrees of freedom (DoF) of the global control points are restricted to displacements along the X and Y axes, while the local control points are unrestricted. The parameter vector $x = [x_1, \cdots, x_{29}]$ defines the displacements of the control points: the first 16 elements govern the global network, and the last 13 control the local network. The FFD Parametrization grid is vizualized in Figure 5.1.

The FFD parametrization, which is fixed during optimization, translates a parent hull-form into a variation based on the modification vector x, which is varied during optimization. Parameter bounds are defined to ensure the hull variations are bounded within a feasible range. The global control network parameters are constrained to move up to $\pm 10\%$ along the vessel's length and $\pm 20\%$ along its width, while the local control network parameters for the bulbous bow are restricted to $\pm 10\%$ in all directions. This approach ensures efficient parameterization for optimization while maintaining geometric consistency across various hull designs. For each parent hull-form, we sampled 90 variation hull-forms from the design space and simulated them using a state-of-the-art commercial CFD solver, Star CCM+, at two independently random Froude numbers in the range $[0.108, 0.280]$. The mesh in Star CCM+ is the same as reported in Chapter 3. The details of the dataset are reported in 5.2.

Table 5.2: Overview of the datasets generated via Free-Form Deformation (FFD) for each parent hull. The input range corresponds to the normalized bounds of the 29-dimensional design variable vector x, used to displace control points and generate geometric variations. The output range refers to the computed total resistance R_t from CFD simulations over the Froude Range.

Parent	Samples	Froude Range [-]	Input Range [-]	Output Range [N]
DTC	180	$[0.108, 0.280]$	$[-0.891, 0.891]$	$[8.250, 230.100]$
KCS	180		$[-0.737, 0.737]$	$[4.008, 81.968]$
S175	180		$[-0.428, 0.428]$	$[2.850, 53.177]$

The dataset in Table 5.2 supports the standard DDS construction approach using geometry modification parameter-based inputs. However, our objective is to address the limitations of the current DSO workflow by decoupling the DDS from the geometry modification parameters, allowing for a more computationally aware approach and physical

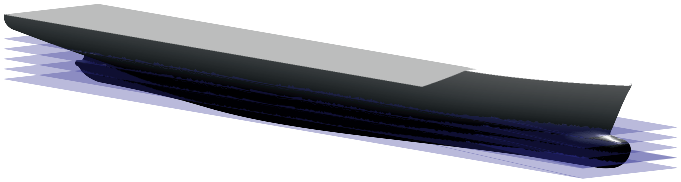


Figure 5.2: Graphical representation of segmenting a hull into $h = 5$ sections along the vertical axis.

plausibility. To support this, we modify the DSO pipeline by redefining the input space using hydrostatic and stability properties. Inspired by our earlier work [235, 239]; however, we adapt the construction of the features in this work to suit the complex geometries of the selected parent hull-forms.

We extract hydrostatic and stability features of the variant hull-forms as follows:

- We segment each hull into h vertical sections extending from the waterline to the keel;
- For each segment, we compute hydrostatic quantities such as buoyancy centers, waterplane area, and moments of inertia;
- We also extract global stability metrics to evaluate design feasibility.

The complete list of geometry, hydrostatics, and stability features is shown in Table 5.3, and a visual example of hull segmentation is shown in Figure 5.2. This procedure results in a feature vector of $\mathbb{R}^{100}$ used as input to the DDS (i.e., the 20 features of Table 5.3 for $h = 5$ vertical sections - drafts).

Table 5.3: Geometry, Hydrostatics, and Stability Features.

Feature	Symbol	Feature	Symbol
Draft	T	Waterplane length	Lwl
Volume	V	Waterplane beam	Bwl
Wetted Surface	Sw	Center of Flotation in X	FX
Center of Buoyancy in X	BX	Center of Flotation in Y	FY
Center of Buoyancy in Y	BY	Center of Flotation in Z	FZ
Center of Buoyancy in Z	BZ	Moment of inertia in X	Ix
Area	A	Moment of inertia in Y	Iy
Minimum Coordinate in X	X_{min}	Maximum Coordinate in X	X_{max}
Minimum Coordinate in Y	Y_{min}	Maximum Coordinate in Y	Y_{max}
Minimum Coordinate in Z	Z_{min}	Maximum Coordinate in Z	Z_{max}

With the modified input space in place, we now need to test its impact on DDS performance using four experimental setups:

- Benchmarking against the conventional DDS approach using FFD parameters (Table 5.2);
- Training DDS models using hydrostatic and stability inputs;
- Assessing extrapolation performance by testing on unseen hull families;
- Evaluating enrichment strategies where extrapolated DDS models are partially supplemented with samples from the target hull family.

Subsequently, to validate the performance of the proposed approach, we apply these DDS models to an optimization task:

- First, we optimize with the baseline DDS trained on FFD inputs;

- Second, we optimize using a DDS trained on hydrostatic features;
- Third, we evaluate optimization with an extrapolated DDS (trained on unrelated hulls);
- Finally, we analyze optimization results under DDS models enriched with partial samples from the target hull.

5.2 Method

5.2.1 DDS Construction

The problem of predicting R_t based on the FFD parametrization or the hydrostatic and stability features, and Fr, can be mapped to a typical regression problem by Machine Learning [202, 206]. We use a popular and effective Machine Learning algorithm, Kernel Ridge Regression (KRR) [206], to build the DDS in this work. In KRR we chose to rely on the Gaussian kernel for the reason described in [119], and then the regularization hyperparameter λ and the kernel coefficient γ need to be tuned. Note that the selection of the best-performing algorithm and the best hyperparameters will depend on the scenario under consideration and the model's performance.

The performance, in terms of accuracy, will be measured in accordance with different metrics: three quantitative (the Mean Absolute Error - MAE, the Mean Absolute Percentage Error - MAPE, and the Pearson Product-Moment Correlation Coefficient - PPMCC) [163] and one qualitative (the scatter plot of the actual versus predicted value) [192].

5.2.2 DDS Validation

In our work, we will study different DDSs based on either interpolating (i.e., when data comes from a single parent hull-form) or extrapolating (i.e., training data comes from one parent hull-form and the DDS is tested on another parent hull-form). This will allow us to understand the extrapolation ability of the DDSs and address the problems described in Section 5.1. The problems we still have to face are how to tune the hyperparameters for this approach (λ, γ) and how to estimate the performance of the final model. Model Selection (MS) and Error Estimation (EE) deal exactly with these problems [169].

When it comes to the interpolation approach, we can use resampling techniques, which are commonly used by researchers and practitioners because they work well in most situations [169]. Resampling techniques are based on a simple idea: the original dataset is resampled once or many times, with or without replacement, to build three independent datasets called learning, validation, and test sets. The simple idea is that the set of hyperparameters that allows achieving a small error on the validation set, which is independent from the learning set used to train the model, should also allow us to perform well on the independent test set. In this work, we will rely on Complete k-fold cross validation, and the resampling must be done without replacement [169].

Instead, for the extrapolation approaches, we take inspiration from our previous work [239]. We perform a Leave One Out (LOO) extrapolation test based on the parent hull-form and the validation is performed according to the method described in our previous work.

5.2.3 Optimization Problem

To test the DDSs in an example optimization scenario, we have constructed a multi-objective problem where the total resistance is evaluated at both a low and high Froude number to account for operational trade-offs between, e.g., slow steaming and fast cruising. Let $\mathrm{Fr}^{\mathrm{Low}} = 0.108$ and $\mathrm{Fr}^{\mathrm{High}} = 0.260$ be the two selected operating conditions. The optimization seeks a parameter vector x that defines the free-form deformation of the parent hull geometry, minimizing a weighted combination of total resistance $\tilde{R}_t(x, \mathrm{Fr})$ at both speeds:

$$\min_{x^l \leq x \leq x^u} \quad \lambda, \tilde{R}_t(x, \mathrm{Fr}^{\mathrm{High}}) + (1 - \lambda), \tilde{R}_t(x, \mathrm{Fr}^{\mathrm{Low}}), \tag{5.1}$$

where $\lambda \in [0, 1]$ controls the trade-off between the objectives.

The optimization is subject to constraints on draft, trim, and displacement:

$$\begin{aligned} \mathrm{T}(x) &= \mathrm{T}^p, \\ \theta(x) &= \theta^p, \\ 0.9 \cdot \nabla^p \leq \nabla(x) &\leq 1.1 \cdot \nabla^p \end{aligned}$$

where T^p, θ^p, and ∇^p denote the draft, trim, and displacement of the parent hull, respectively.

Problem (5.1) is a non-linear optimization problem with both non-linear equality and inequality constraints. To solve it, we adopt an evolutionary algorithm (EA)-based approach implemented in MATLAB using the ga function, which provides a variant of the NSGA-II Genetic Algorithm [49, 153]. To improve robustness and mitigate the risk of convergence to local minima, we incorporate a manually implemented multi-start strategy, executing the algorithm independently across ten runs and retaining the best solution found. The optimizer is configured with a population size of 5000, a maximum of 200 generations, a crossover fraction of 0.8, and an elite count of 1.

5.3 Results

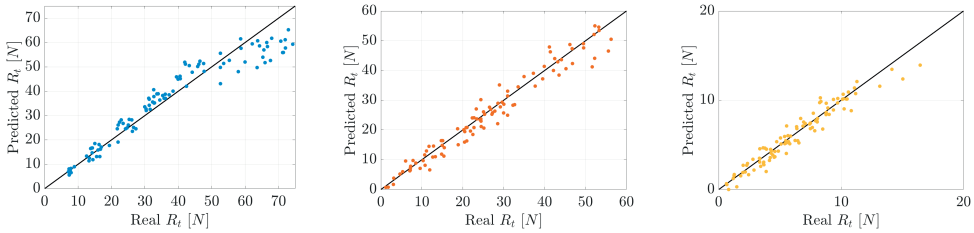
In this section, we evaluate the performance of the proposed DDS configurations and optimization strategies described in Section 5.2 to address the problem and data described in Section 5.1. Our goal is to assess how well different DDS input spaces generalize across parent hull-forms and how effectively they support the design optimization process. We organize the results into two parts. First, in Section 5.3.1, we address the modifications to the DSO pipeline by proposing an alternative decoupled approach to predict R_t with DDSs. Then, in Section 5.3.2, we test our proposed DDSs by performing the numerical optimization of an example problem.

5.3.1 DDS Results

Table 5.4 reports the benchmarking for the DDS trained on the FFD parametrization (x) as input (i.e., Table 5.2) to predict R_t for the hull-forms derived from each parent (DTC,

Table 5.4: Performance of DDS for each parent hull using an input space composed of Froude number and FFD parameters x. The models are trained and tested using data belonging to just one parent-hull (i.e., interpolation) and the reported metrics refer to predictions of total resistance R_t.

Target	MAE [N]	MAPE [%]	R2 [-]	PPMCC [-]
DTC	4.37 ± 2.13	1.00 ± 0.00	0.91 ± 0.05	0.96 ± 0.03
KCS	2.34 ± 0.83	0.97 ± 0.05	0.96 ± 0.06	0.98 ± 0.02
S175	0.65 ± 0.15	0.99 ± 0.05	0.93 ± 0.07	0.97 ± 0.03



(a) DTC. (b) KCS. (c) S175.

Figure 5.3: Scatter Plot of real versus predicted total resistance R_t for each parent hull using an input space composed the of Froude number and FFD parameters x.

S175, and KCS). The models are trained and tested using data belonging to just one parent-hull (i.e., the interpolation scenario) and the reported metrics refer to predictions of total resistance R_t.

From Table 5.4 it can be observed that

- DDS models trained on FFD parameters demonstrate strong interpolation performance;
- KCS achieves the lowest MAE (2.34 N) and highest R^2 (0.96), indicating high accuracy;
- All models show strong and consistent performance with MAPE around 1%;

Figure 5.3 shows the Scatter Plot of real versus predicted total resistance R_t for each parent hull using an input space composed the of Froude number and FFD parameters x.

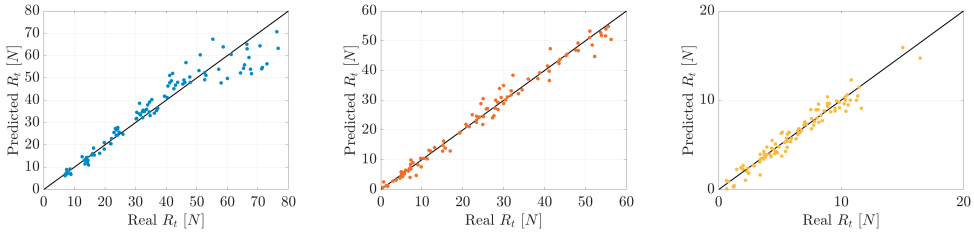
Table 5.5 shows the same experiment as Table 5.4, but the input space of the DDS is now composed of the hydrostatic and stability features. The models are trained and tested using data belonging to just one parent-hull (i.e., the interpolation scenario), and the reported metrics refer to predictions of total resistance R_t.

From Table 5.5 it can be observed that

- Hydrostatic features offer competitive or superior interpolation performance com-

Table 5.5: Performance of DDS for each parent hull using an input space composed of the Froude number and the hydrostatic and stability features. The models are trained and tested using data belonging to just one parent-hull (i.e., interpolation), and the reported metrics refer to predictions of total resistance R_t.

Target	MAE [N]	MAPE [%]	R2 [-]	PPMCC [-]
DTC	4.04 ± 2.10	1.00 ± 0.00	0.91 ± 0.12	0.96 ± 0.06
KCS	1.48 ± 0.70	0.97 ± 0.05	0.98 ± 0.04	0.99 ± 0.01
S175	0.58 ± 0.22	1.00 ± 0.00	0.95 ± 0.04	0.97 ± 0.02



(a) DTC. (b) KCS. (c) S175.

Figure 5.4: Scatter Plot of real versus predicted total resistance R_t for each parent hull using an input space composed of the Froude number and the hydrostatic features.

pared to FFD inputs.

- KCS again achieves the best performance (MAE = 1.48 N, $R^2 = 0.98$).
- For S175, both MAE and R^2 improve relative to the FFD input case.
- The DTC performance is stable, showing the robustness of hydrostatic inputs.

Figure 5.4 shows the Scatter Plot of real versus predicted total resistance R_t for each parent hull using an input space composed of the Froude number and the hydrostatic and stability features.

Now that we have addressed the interpolation DDSs, we want to test the extrapolation capabilities. The first extrapolation models are trained on data belonging to a particular parent hull, and the reported metrics refer to predictions of total resistance R_t on different parent hulls.

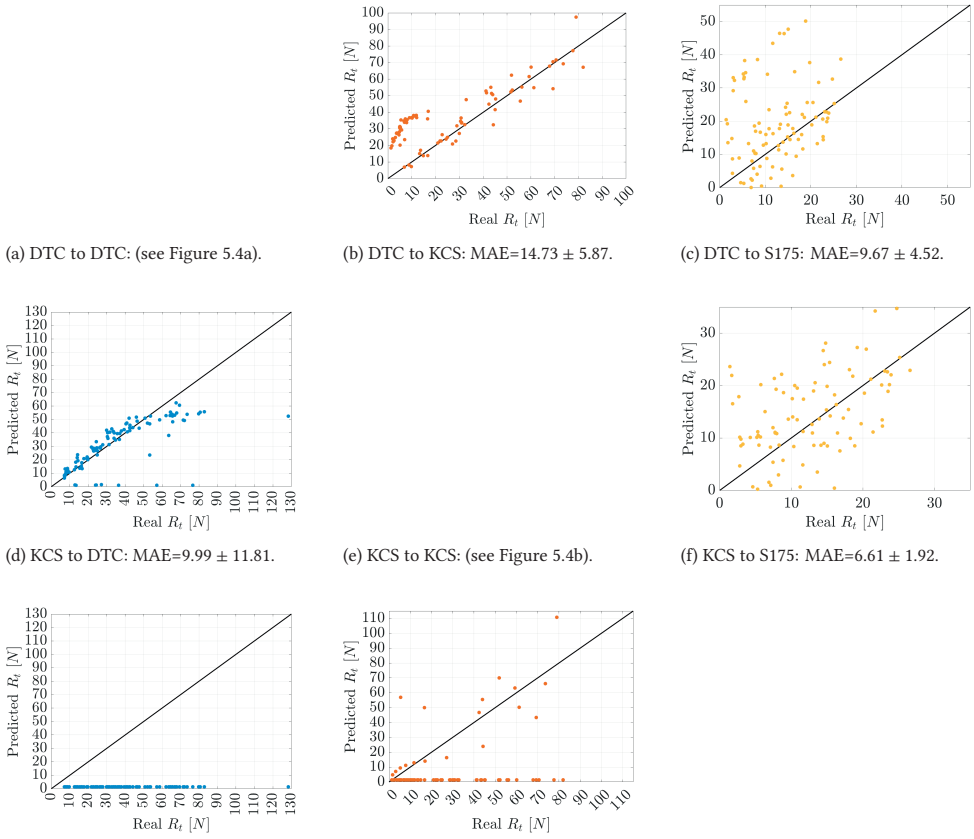
Figure 5.5 shows the Scatter plots comparing predicted and true total resistance R_t for all 1-to-1 evaluation cases. Each row corresponds to a surrogate model trained on a single parent hull (DTC, KCS, S175), and each column shows predictions made on a different hull (i.e., in the extrapolation scenario). The interpolation scenarios are found in Figure 5.4. MAE values evaluated with a 95% confidence interval are shown in the sub-captions.

From Figure 5.5 it can be observed that

- DDSs exhibit variable extrapolation capability across parent hulls.
- Models trained on S175 generalize poorly to DTC and KCS (high MAE, low R^2).
- DTC→KCS and KCS→S175 are more successful, suggesting moderate geometry similarity.
- Worst case: S175→DTC (MAE = 32.74 N, $R^2 = 0.32$).

Actually, these results are reasonable considering that there is only one parent hull-form in the training data. It is also reasonable that the S175 generalizes worse than the other two parent-hulls because the bulbous bow is significantly different on this hull-form, whereas there is a more similar topology between the KCS and the DTC hull forms.

The next step to the extrapolation test is to train the DDS on data belonging to two parent hulls and test it on unseen data belonging to another parent hull (i.e., the LOO scenario). Table 5.6 shows the Performance of DDS for each parent hull using an input space composed of the Froude number and the hydrostatic and stability features. The models are trained on two-class data (i.e., corresponding to two parent hulls), and the reported metrics refer to predictions of total resistance R_t on out-of-class data (i.e., leave one parent hull out scenario).



(a) DTC to DTC: (see Figure 5.4a).

(b) DTC to KCS: MAE=14.73 ± 5.87.

(c) DTC to S175: MAE=9.67 ± 4.52.

(d) KCS to DTC: MAE=9.99 ± 11.81.

(e) KCS to KCS: (see Figure 5.4b).

(f) KCS to S175: MAE=6.61 ± 1.92.

(g) S175 to DTC: MAE=32.74 ± 12.61.

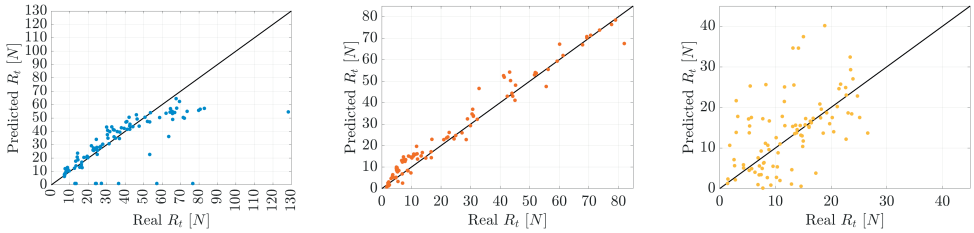
(h) S175 to KCS: MAE=18.09 ± 7.81.

(i) S175 to S175: (see Figure 5.4c).

Figure 5.5: Scatter plots comparing predicted and true total resistance R_t for all 1-to-1 evaluation cases. Each row corresponds to a surrogate model trained on a single parent hull (DTC, KCS, S175), and each column shows predictions made on a different hull (i.e., in an extrapolation scenario). The interpolation scenarios are found in Figure 5.4. MAE values evaluated with a 95% confidence interval are shown in the sub-captions.

Table 5.6: Performance of DDS for each parent hull using an input space composed of the Froude number and the hydrostatic and stability features. The models are trained on two-class data (i.e., corresponding to two parent hulls), and the reported metrics refer to predictions of total resistance R_t on out-of-class data (i.e., leave one parent hull out scenario).

Train	Target	MAE [N]	MAPE [%]	R2 [-]	PPMCC [-]
KCS+S175	DTC	9.67 ± 5.45	1.00 ± 0.05	0.32 ± 0.80	0.62 ± 0.55
DTC+S175	KCS	3.26 ± 1.15	0.96 ± 0.10	0.96 ± 0.10	0.98 ± 0.01
DTC+KCS	S175	5.61 ± 2.17	0.78 ± 0.30	0.41 ± 0.16	0.56 ± 0.33



(a) DTC. (b) KCS. (c) S175.

Figure 5.6: Scatter Plot of real versus predicted total resistance R_t for each parent hull in the leave one out scenario using an input space composed of the Froude number and the hydrostatic features.

Table 5.7: Performance of DDS for each parent hull using an input space composed of the Froude number and the hydrostatic features. The models are trained on data belonging to two parent hulls and enriched with varying amounts of data corresponding to another parent hull. The Enrichment Fraction (r) is varied to include different fractions of the testing parent hull-form data. The metric refers to the MAE of total resistance R_t on out-of-class unseen data (i.e., corresponding to a different parent hull).

Train	**Target**	**Enrichment Fraction** (r) [-]				
		0.1	0.3	0.5	0.7	0.9
KCS+S175	DTC	6.00 ± 1.51	4.24 ± 0.61	3.03 ± 0.70	2.58 ± 1.25	1.58 ± 1.29
DTC+S175	KCS	2.85 ± 0.29	1.80 ± 0.23	1.29 ± 0.27	0.86 ± 0.20	0.54 ± 0.18
DTC+KCS	S175	5.31 ± 1.57	1.92 ± 0.80	1.22 ± 0.55	0.89 ± 0.40	0.53 ± 0.26

From Table 5.6 it can be observed that

- Leave-one-out (LOO) performance varies widely based on the excluded hull.
- Best generalization when KCS is the target ($R^2 = 0.96$).
- Leaving out DTC significantly degrades accuracy ($R^2 = 0.32$), even with two hulls in training.
- S175 remains a challenging target despite two-source training data.

Figure 5.6 shows the Scatter Plot of real versus predicted total resistance R_t for each parent hull in the leave one out scenario using an input space composed of the Froude number and the hydrostatic features.

These results indicate that due to the complex and varying topology of the three parent hull-forms, pure extrapolation based on LOO may have limited potential (due to finite DDS accuracy). Therefore, we investigate trading off the computational benefits of enriching the LOO datasets with varying amounts of data belonging to the testing class. Table 5.7 shows the performance of DDSs for each parent hull using an input space composed of the Froude number and the hydrostatic and stability features. The models are trained on data belonging to two parent hulls and enriched with varying amounts of data corresponding to another parent hull. The Enrichment Fraction (r) is varied to include different fractions of the testing parent hull-form data. The metric refers to the MAE of total resistance R_t on out-of-class unseen data (i.e., corresponding to a different parent hull).

From Table 5.7 it can be observed that

- Incremental enrichment with target hull data (r from 0.1 to 0.9) improves performance across the board.
- DTC sees largest gain (MAE: 6.00 → 1.58).

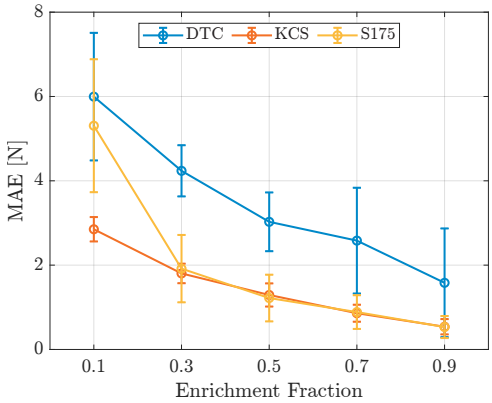


Figure 5.7: Graphical representation of Table 5.7: Line Plot of the MAE when models are trained on two-class data (i.e., corresponding to two parent hulls) and enriched with varying amounts of out-of-class data (i.e., corresponding to another parent hull). Captions are for the out-of-class parent hull.

- Diminishing returns observed beyond $r = 0.7$.
- Shows how minimal sampling from out-of-class hulls can drastically boost DDS accuracy.

Figure 5.7 shows a graphical representation of Table 5.7.

Figures 5.8, 5.9, and 5.10 show the scatter plot of real versus predicted total resistance R_t for the each parent hull (DTC, KCS, and S175 respectively) in the leave one out scenario with varying amounts of enrichment with testing parent-hull data.

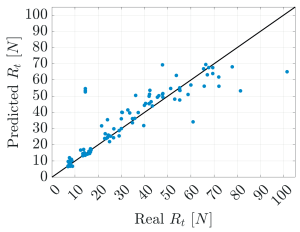
5.3.2 Optimization

This section tests the DDSs shown in Section 5.3.1 in the optimization problem described in Section 5.2.3. First, we decided to pick one optimization case (out of DTC, KCS, and S175) to demonstrate our proposed approach. We selected the KCS because it's well benchmarked in the literature. We performed the optimization with the four different surrogates (belonging to Table 5.4, Table 5.5, Table 5.6, and Table 5.7, respectively).

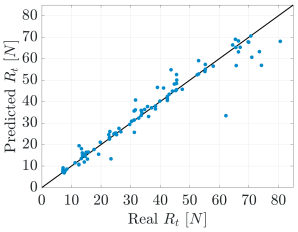
Figure 5.11 shows the convergence of the multi-objective optimization problem using the Genetic algorithm (GA) for the KCS hull using different Data-Driven Surrogates (i.e., from Tables 5.4, 5.5, 5.6, and 5.7). Each trace corresponds to the mean convergence trajectory across 10 multi-start runs for a given λ. Shaded regions indicate the standard deviation around the mean according to multi-start. Objective values are normalized to facilitate comparison across weights.

From Figure 5.11 it can be observed that

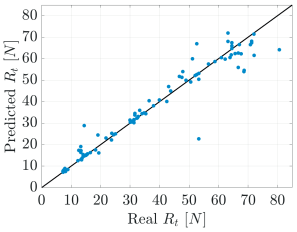
- Using the DDS trained on the FFD parametrization converges to a steady solution for each λ.
- Using the DDS trained on the geometry, hydrostatics, and stability features (b), (c), and (d) leads to faster convergence and lower variance compared to (a), suggesting more stable optimization behavior.
- The DDS trained in the LOO scenario (c) converges quicker than the other DDSs.



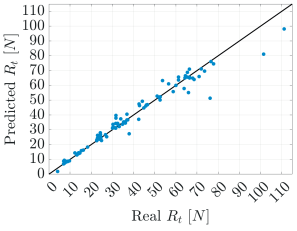
(a) Leave-DTC-out ($r = 0.1$).



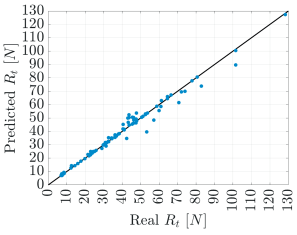
(b) Leave-DTC-out ($r = 0.3$).



(c) Leave-DTC-out ($r = 0.5$).

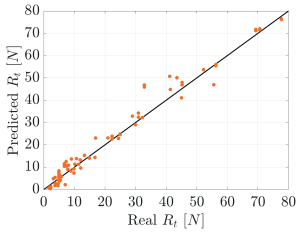


(d) Leave-DTC-out ($r = 0.7$).

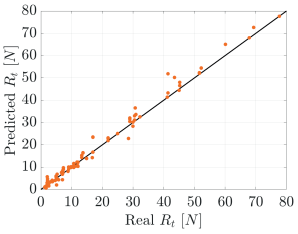


(e) Leave-DTC-out ($r = 0.9$).

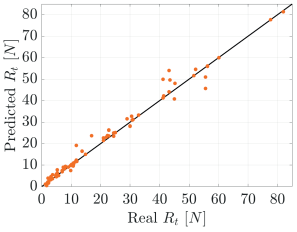
Figure 5.8: Scatter Plot of real versus predicted total resistance R_t for the DTC parent hull in the leave one out scenario with varying amounts of enrichment with in-class data.



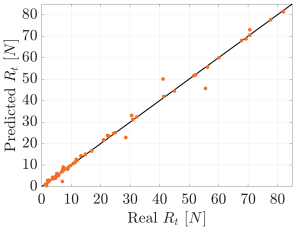
(a) Leave-KCS-out ($r = 0.1$).



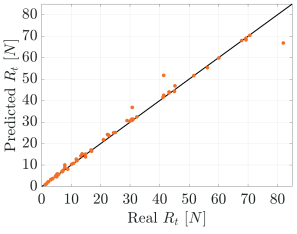
(b) Leave-KCS-out ($r = 0.3$).



(c) Leave-KCS-out ($r = 0.5$).

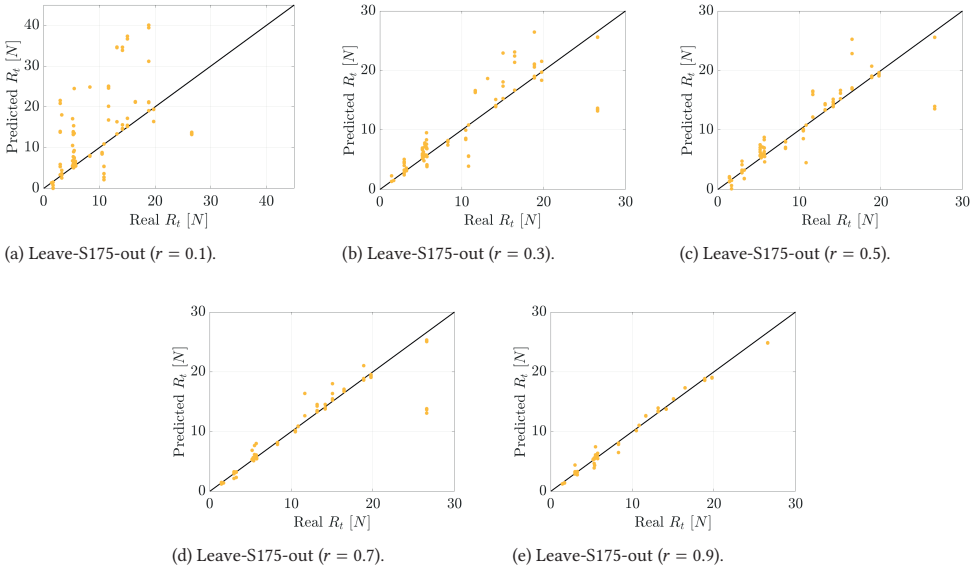


(d) Leave-KCS-out ($r = 0.7$).



(e) Leave-KCS-out ($r = 0.9$).

Figure 5.9: Scatter Plot of real versus predicted total resistance R_t for the KCS parent hull in the leave one out scenario with varying amounts of enrichment with in-class data.



(a) Leave-S175-out ($r = 0.1$).

(b) Leave-S175-out ($r = 0.3$).

(c) Leave-S175-out ($r = 0.5$).

(d) Leave-S175-out ($r = 0.7$).

(e) Leave-S175-out ($r = 0.9$).

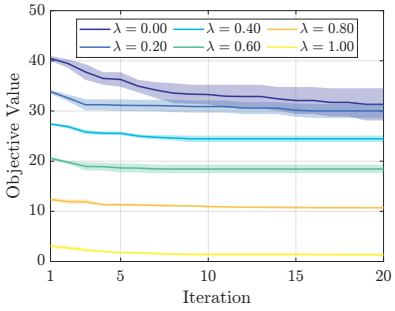
Figure 5.10: Scatter Plot of real versus predicted total resistance R_t for the S175 parent hull in the leave one out scenario with varying amounts of enrichment with in-class data.

Figure 5.12 shows the cost of a multi-objective optimization problem using the Genetic algorithm (GA) for the KCS hull using different Data-Driven Surrogates (i.e., from Tables 5.4, 5.5, 5.6, and 5.7). A boxplot is plotted for the cost found across 10 multi-starts for a given convex weight λ.

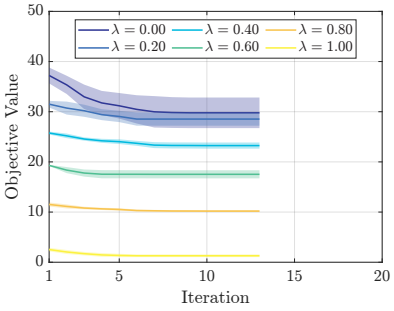
Figure 5.13 shows the Pareto Front of a multi-objective optimization problem using the Genetic algorithm (GA) for the KCS hull using different Data-Driven Surrogates (i.e., from Tables 5.4, 5.5, 5.6, and 5.7). Each round marker represents the best design found across 10 multi-starts for a given convex weight λ, with lighter shaded bars indicating confidence intervals; the corresponding diamond markers are the result informed by CFD.

From Figure 5.13 it can be observed that

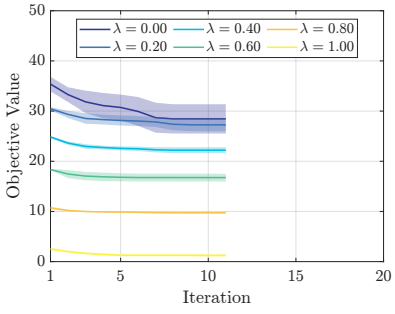
- For the DDS trained on the FFD parametrization (Figure 5.13a):
  - The Pareto front shows that for all values of λ the optimizer was able to obtain a solution.
  - CFD validation shows a small but consistent underestimation of true resistance, suggesting minor model bias.
  - Confidence intervals are narrow, reflecting good optimization stability across multi-starts.
- For the DDS trained on the geometry, hydrostatics, and stability features (Figure 5.13b):
  - The Pareto front is mostly comparable to the Pareto found with the DDS trained on the FFD-parameters (Figure 5.13a), indicating that in both cases the optimizer found similar geometries.
  - CFD-validation markers are slightly further from the round ones, showing worse DDS prediction capabilities.



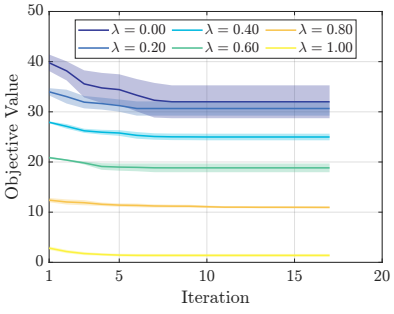
(a) DDS using an input space composed the of Froude number and FFD parameters x (Table 5.4).



(b) DDS using an input space composed the of Froude number and the hydrostatic features $\mathcal{F}$ (Table 5.5).

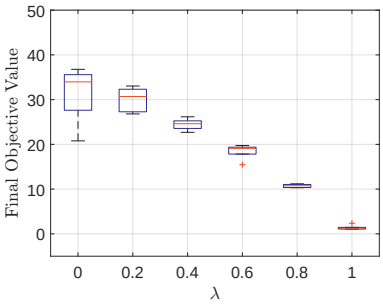


(c) DDS using an input space composed the of Froude number and the hydrostatic features $\mathcal{F}$ with KCS as the Leave One Parent Hull Out (Table 5.6).

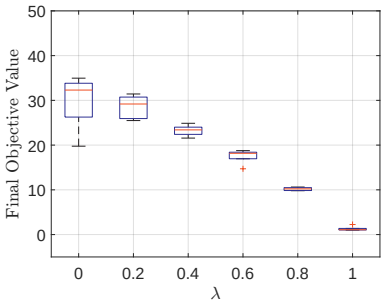


(d) DDS using an input space composed the of Froude number and the hydrostatic features $\mathcal{F}$ with KCS as the Leave One Parent Hull Out enriched with a fraction ($r = 0.7$) of the KCS data (Table 5.7).

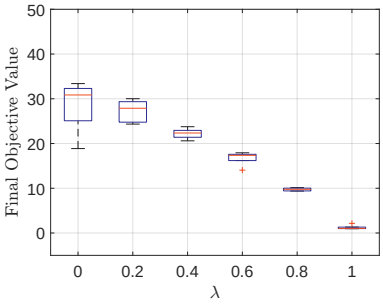
Figure 5.11: Convergence of the multi-objective optimization problem using the Genetic algorithm (GA) for the KCS hull using different Data-Driven Surrogates (i.e., from Tables 5.4, 5.5, 5.6, and 5.7). Each trace corresponds to the mean convergence trajectory across 10 multi-start runs for a given λ. Shaded regions indicate the standard deviation around the mean according to multi-start. Objective values are normalized to facilitate comparison across weights.



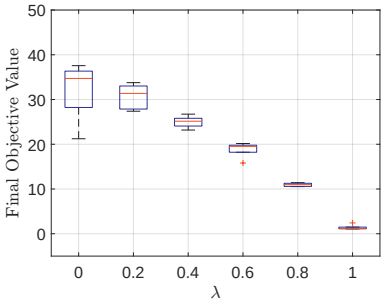
(a) DDS using an input space composed the of Froude number and FFD parameters x (Table 5.4).



(b) DDS using an input space composed of the Froude number and the hydrostatic features $\mathcal{F}$ (Table 5.5).

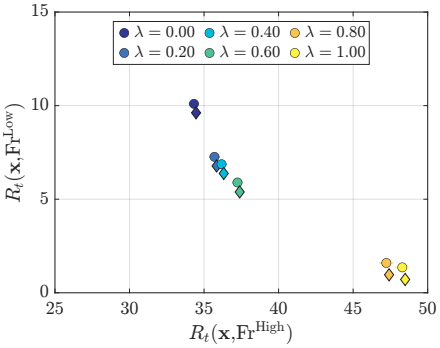


(c) DDS using an input space composed the of Froude number and the hydrostatic features $\mathcal{F}$ with KCS as the Leave One Parent Hull Out (Table 5.6).

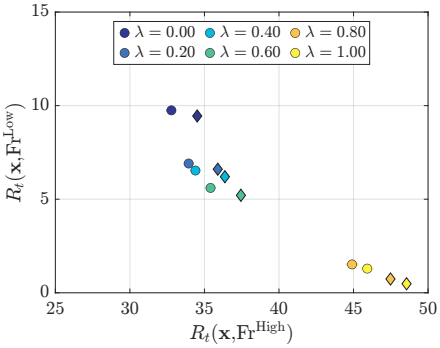


(d) DDS using an input space composed of the Froude number and the hydrostatic features $\mathcal{F}$ with KCS as the Leave One Parent Hull Out enriched with a fraction ($r = 0.7$) of the KCS data (Table 5.7).

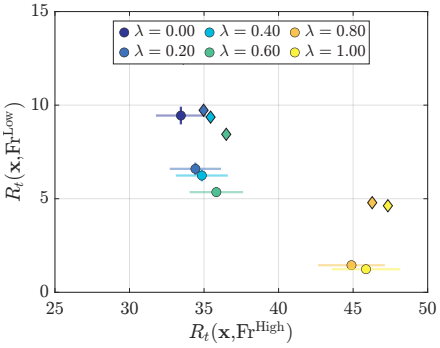
Figure 5.12: Cost of multi-objective optimization problem using the Genetic algorithm (GA) for the KCS hull using different Data-Driven Surrogates (i.e., from Tables 5.4, 5.5, 5.6, and 5.7). A boxplot is plotted for the cost found across 10 multi-starts for a given convex weight λ.



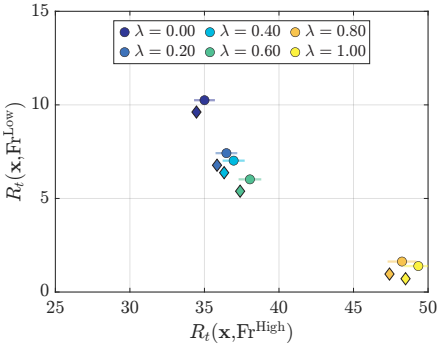
(a) DDS using an input space composed the of Froude number and FFD parameters x (Table 5.4).



(b) DDS using an input space composed the of Froude number and the hydrostatic features $\mathcal{F}$ (Table 5.5).



(c) DDS using an input space composed the of Froude number and the hydrostatic features $\mathcal{F}$ with KCS as the Leave One Parent Hull Out (Table 5.6).



(d) DDS using an input space composed of the Froude number and the hydrostatic features $\mathcal{F}$ with KCS as the Leave One Parent Hull Out enriched with a fraction (r = 0.7) of the KCS data (Table 5.7).

Figure 5.13: Pareto Front of multi-objective optimization problem using the Genetic algorithm (GA) for the KCS hull using different Data-Driven Surrogates (i.e., from Tables 5.4, 5.5, 5.6, and 5.7). Each round marker represents the best design found across 10 multi-starts for a given convex weight λ, with lighter shaded bars indicating confidence intervals, the corresponding diamond markers are the result informed by CFD.

- For the DDS trained on the geometry, hydrostatics, and stability features in LOO scenario (Figure 5.13c):
  - Pareto front degrades in quality and smoothness due to generalization limitations in out-of-class prediction.
  - CFD-informed markers deviate more strongly, confirming weak extrapolation behavior of this DDS.
- For the DDS trained on the geometry, hydrostatics, and stability features in the LOO scenario, then enriched with data from the LOO parent hull-form (Figure 5.13d):
  - Pareto front recovers and even surpasses the DDS trained only on hydrostatic and stability features.
  - CFD predictions align well with DDS predictions, confirming the benefit of incorporating out-of-class data on DDS prediction accuracy.

Overall, Figure 5.13 underscores the importance of training data composition and feature choice on the quality of optimization outcomes. Enrichment with target-hull data significantly enhances both DDS accuracy and the resulting design diversity.

5.4 Chapter Summary

This chapter introduced a new approach to develop Data-Driven Surrogates (DDSs) for hull resistance prediction within the DDS-based hull-form optimization (DSO) workflow. We modified the conventional DSO pipeline, which relies solely on geometry-specific parametrizations such as Free-Form Deformation (FFD), by proposing a decoupled input space composed of hydrostatic and stability features. This change enables the DDS to be agnostic to the parametrization scheme and facilitates extrapolation to unseen hull-form families.

To validate our approach, we generated a dataset for three canonical parent hull-forms (DTC, KCS, and S175), each perturbed via FFD and evaluated with CFD across a range of Froude numbers. We benchmarked DDSs trained on conventional FFD parameters against those trained on the proposed hydrostatic input space under interpolation conditions. We then tested the DDS performance under extrapolation scenarios, including leave-one-out and hybrid enrichment cases, and assessed their impact within a full DSO workflow. Uniquely, we proposed and demonstrated a method to reduce the computational burden of DDS construction by enriching historical CFD datasets with a minimal number of new simulations targeted at the problem of interest. This hybrid strategy outperformed conventional DDS training approaches in both accuracy and efficiency.

Our results show that DDSs trained on hydrostatic features retain strong predictive accuracy within the interpolation regime and, critically, offer consistent and reliable performance under extrapolation. This makes the method particularly attractive for industrial contexts, where CFD data from prior design campaigns can be reused. We demonstrated that this enrichment strategy can reduce the need for new CFD simulations by up to 30%, without sacrificing predictive performance.

The proposed decoupling of geometry and performance spaces broadens the applicability of DDS techniques, enabling more flexible, data-efficient, and reusable optimization pipelines. Future work should explore how this input space can support other stages of the DSO process.

6

Conclusion

This dissertation has investigated how data-driven surrogate models can enhance mid-stage hull-form optimization by reducing reliance on computationally expensive CFD simulations while preserving the physical credibility and engineering relevance of the resulting designs. The central aim was not only to accelerate the optimization process but to do so in a way that respects the fundamental principles of naval architecture, enables data reuse across projects, and extends predictive capability into previously unexplored regions of the design space.

Chapter 2 reviewed the state of the art in hull-form optimization, structured around four core steps: geometric parameterization, data generation, surrogate modelling, and physical plausibility evaluation. This synthesis identified six key research gaps (Table 1.1): the absence of integrated physical constraints in surrogate-based workflows (G1), high computational cost limiting exploration of the design space (G2), the tight coupling of surrogates to specific parameterizations (G3), the lack of systematic data reuse across projects (G4), the general avoidance of extrapolation in most studies (G5), and the absence of strategies for transferring models between hull families (G6).

Chapter 3 addressed G1 and G2 by embedding IMO intact stability and hydrostatic constraints directly into the optimization loop. This ensured that all candidate designs satisfied key physical and regulatory requirements without the need for post-processing, while dimensionality reduction and constraint-driven pruning reduced CFD effort by up to 30% without degrading prediction accuracy. Chapter 4 addressed G3 and G4 through the development of a parameterization-agnostic, decoupled framework that cleanly separates geometry encoding from surrogate modelling. This allowed simulation data to be reused across different parameterizations and hull families, improving scalability and significantly reducing the need for new CFD campaigns. Chapter 5 addressed G5 and G6 by treating extrapolation as a design capability and demonstrating cross-hull transfer learning for resistance prediction. Surrogates trained on generalizable feature spaces were able to capture hydrodynamic trends for unseen hull forms and operational regimes, maintaining physical plausibility even in challenging extrapolation scenarios.

Together, these contributions establish a modular, physically informed, and computationally efficient methodology for hull-form optimization. By systematically addressing

G1–G6, the work moves beyond narrowly specialized, parameterization-locked pipelines toward reusable frameworks that integrate domain constraints, reduce simulation costs, leverage hybrid data reuse, and extend predictive reach beyond the immediate training domain. These capabilities directly support the maritime sector's decarbonization and efficiency targets by enabling rapid, reliable evaluation of novel hull designs under diverse operational constraints. The methods developed here provide a foundation for scalable and sustainable AI-assisted ship design workflows that balance performance, cost-effectiveness, and innovation, while opening the door to broader exploration of unconventional hull configurations in pursuit of next-generation maritime transport solutions.

It is worth addressing the fact that this work has also opened several avenues for further research and development.

First, the decoupled surrogate modeling framework could be extended to integrate with alternative models to modify hull geometries. For instance, generative adversarial networks (GANs) are one increasingly popular approach for producing novel geometries in other engineering domains, with growing traction in the martime domain, but often suffer from producing designs that are not physically realistic. Combining predictive models (as developed in this work) with GANs is a potentially vibrant area of future research in the maritime domain to generate and assess diverse and physically plausible hull forms. Similarly, a reinforcement learning framework could be used to adapt existing parameterization schemes (e.g., Free-Form Deformation lattices or parametric models) during online hull form optimization. Uniquely, the predictive models developed in this work are parametrization-agnostic, thereby enabling optimization algorithms to evolve both the geometry and the design space itself in response to performance feedback.

Second, a significant step forward would be to eliminate the intermediate feature-engineering stage by training predictive models directly on geometric representations (e.g., surface meshes or volumetric grids). The principal bottlenecks for this approach are the need for substantially larger datasets and the increased complexity of deep models, both in terms of training stability and computational cost.

Lastly, a natural progression of this work is to increase the complexity of the optimization problem to cover additional performance criteria, such as seakeeping, manoeuvrability, structural performance, or life-cycle cost. This would require integrating multiple predictive models, each possibly trained on different datasets, within a unified optimization framework. Such integration would further reduce the exploration of infeasible designs and improve the readiness of optimized hulls for practical application.

Bibliography

References

[1] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), 2018.

[2] E. Acar. Effect of error metrics on optimum weight factor selection for ensemble of metamodels. *Expert Systems with Applications*, 42(5):2703–2709, 2015.

[3] G. Achour, W. J. Sung, O. J. Pinon-Fischer, and D. N. Mavris. Development of a conditional generative adversarial network for airfoil shape optimization. In *AIAA Scitech 2020 Forum*, page 2261, 2020.

[4] S. P. Adam, S. A. N. Alexandropoulos, P. M. Pardalos, and M. N Vrahatis. No free lunch theorem: A review. In *Approximation and optimization: Algorithms, complexity and applications*, 2019.

[5] C. C. Aggarwal. *Neural networks and deep learning*. Springer, 2018.

[6] M. Ahmadzadehtalatapeh and M. Mousavi. A review on the drag reduction methods of the ship hulls for improving the hydrodynamic performance. *International Journal of Maritime Technology*, 4:51–64, 2015.

[7] P. F. Albuquerque, P. V. Gamboa, and M. A. Silvestre. Mission-based multidisciplinary aircraft design optimization methodology tailored for adaptive technologies. *Journal of aircraft*, 55(2):755–770, 2018.

[8] N. Altman and M. Krzywinski. The curse (s) of dimensionality. *Nat Methods*, 15(6): 399–400, 2018.

[9] M. J. Anderson and P. J. Whitcomb. *RSM Simplified: Optimizing Processes Using Response Surface Methods for Design of Experiments, Second Edition*. Productivity Press, 2016.

[10] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, and B. van Wachem. *Computational Fluid Dynamics for Engineers*. Cambridge University Press, 2011.

[11] J. H. Ang, C. Goh, and Y. Li. Key challenges and opportunities in hull form design optimisation for marine and offshore applications. In *International Conference on Automation and Computing: Automation, Computing and Manufacturing for New Economic Growth, ICAC 2015*, 2015.

[12] J. Antony. *Design of Experiments for Engineers and Scientists.* Elsevier, 2014.

[13] J. Antony. *Design of Experiments for Engineers and Scientists (Second Edition).* Elsevier, 2014.

[14] N. J. Bagazinski and F. Ahmed. ShipGen: A Diffusion Model for Parametric Ship Hull Generation with Multiple Objectives and Constraints. *Journal of Marine Science and Engineering*, 11(12):2215, 2023.

[15] C. Bai and W. Wang. Review of computational and experimental approaches to analysis of aerodynamic performance in horizontal-axis wind turbines (HAWTs). *Renewable and Sustainable Energy Reviews*, 63:506–519, 2016.

[16] C. J. Barnes and M. Visbal. High-fidelity LES simulations of self-sustained pitching oscillations on a NACA0012 airfoil at transitional Reynolds numbers. In *54th AIAA Aerospace Sciences Meeting*, page 1353, 2016.

[17] C. M. Bishop et al. *Neural networks for pattern recognition.* Oxford university press, 1995.

[18] W. M. Bolstad. *Introduction to Bayesian Statistics, Second Edition.* John Wiley & Sons, Inc., 2007.

[19] L. Bonfiglio, P. Perdikaris, S. Brizzolara, and G. E. Karniadakis. Multi-fidelity optimization of super-cavitating hydrofoils. *Computer Methods in Applied Mechanics and Engineering*, 332:63–85, 2018.

[20] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

[21] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[22] H. P. Buckley, . Y. Zhou, and D. W. Zingg. Airfoil optimization using practical aerodynamic design requirements. *Journal of Aircraft*, 47(5):1707–1719, 2010.

[23] R. S. Burachik, C. Y. Kaya, and M. M. Rizvi. Algorithms for generating Pareto fronts of multi-objective integer and mixed-integer programming problems. *Engineering Optimization*, 54(8):1413–1425, 2022.

[24] M. Buscema. Back Propagation Neural Networks. *Substance use & misuse*, 33:233–70, 02 1998.

[25] E. F. Campana, D. Peri, Y. Tahara, and F. Stern. Shape optimization in ship hydrodynamics using computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 196:634–651, 2006.

[26] P. Casalone, O. Dell'Edera, B. Fenu, G. Giorgi, S. A. Sirigu, and G. Mattiazzo. Unsteady RANS CFD simulations of Sailboat's hull and comparison with full-scale test. *Journal of Marine Science and Engineering*, 8(6):394, 2020.

[27] D. Castelvecchi. Can we open the black box of AI? *Nature News*, 538(7623):20, 2016.

[28] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[29] X. Cheng, B. Feng, Z. Liu, and H. Chang. Hull surface modification for ship resistance performance optimization based on Delaunay triangulation. *Ocean Engineering*, 153:333–344, 2018.

[30] D. Chicco, L. Oneto, and E. Tavazzi. Eleven quick tips for data cleaning and feature engineering. *PLOS Computational Biology*, 18(12):e1010718, 2022.

[31] Y. R. Choi, S. Y Hong, and H. S. Choi. An analysis of second-order wave forces on floating bodies by using a higher-order boundary element method. *Ocean Engineering*, 28(1):117–138, 2001.

[32] F. Cipollini, L. Oneto, A. Coraddu, and S. Savio. Unsupervised deep learning for induction motor bearings monitoring. *Data-Enabled Discovery and Applications*, 3: 1–13, 2019.

[33] B. Clarke, E. Fokoue, and H. H. Zhang. *Principles and theory for data mining and machine learning*. Springer, 2009.

[34] A. Coppedé, G. Vernengo, and D. Villa. A combined approach based on Subdivision Surface and Free Form Deformation for smart ship hull form design and variation. *Ships and Offshore Structures*, 13(7):769–778, 2018.

[35] A. Coppedè, S. Gaggero, G. Vernengo, and D. Villa. Hydrodynamic shape optimization by high fidelity CFD solver and Gaussian process based response surface method. *Applied Ocean Research*, 90:101841, 2019.

[36] Coppedè, A. and Gaggero, S. and Vernengo, G. and Villa, D. Hydrodynamic shape optimization by high fidelity cfd solver and gaussian process based response surface method. *Applied Ocean Research*, 90(May 2018):101841, 2019.

[37] A. Coraddu, P. Gualeni, and D. Villa. Investigation about wave profile effects on ship stability. In *Sustainable Maritime Transportation and Exploitation of Sea Resources: Proceedings of the 14th International Congress of the International Maritime Association of the Mediterranean*, pages 143–149, 2011.

[38] A. Coraddu, D. Villa, and P. Gualeni. Vulnerability assessment for the loss of stability in waves: some application cases for a further insight into the problem. In *Proceedings of the 11th International Conference on the Stability of Ships and Ocean Vehicles*, pages 23–28, 2012.

[39] A. Coraddu, L. Oneto, M. Kalikatzarakis, D. Ilardi, and M. Collu. Floating Spar-Type Offshore Wind Turbine Hydrodynamic Response Characterisation: A Computational Cost Aware Approach. In *OCEANS 2020 Singapore – U.S. Gulf Coast*, pages 1–8. IEEE, 2020.

[40] A. Coraddu, M. Kalikatzarakis, J. Walker, D. Ilardi, and L. Oneto. Data science and advanced analytics for shipping energy systems. In *Sustainable Energy Systems on Ships*, pages 303–349. Elsevier, 2022.

[41] A. Coraddu, S. Gaggero, D. Villa, and L. Oneto. A Non-Deterministic Propeller Design Optimization Framework Leveraging Machine Learning Based Boundary Element Methods Surrogates. In *International Conference on Computational Methods in Marine Engineering: MARINE2023*, 2023.

[42] A. Coraddu, L. Oneto, J. Walker, K. Patryniak, A. Prothero, and M. Collu. Floating offshore wind turbine mooring line sections health status nowcasting: From supervised shallow to weakly supervised deep learning. *Mechanical Systems and Signal Processing*, 216:111446, 2024.

[43] A. Coraddu, J. M. Walker, R. Dick, and L. Oneto. From Fouling to Fuel Savings: A Data-driven Assessment of Hull & Propeller Cleaning Effectiveness. *Modelling and Optimisation of Ship Energy Systems 2025*, Sept. 2025.

[44] A. Coraddu, J. M. Walker, A. Mikelic, A. Ponte, and L. Oneto. Artificial Intelligence-Based Surrogate Models for Ship Resistance and Wave Pattern Prediction during Early-Stage Ship Design. *Ocean Engineering (Submitted)*, 2025.

[45] C. Cortes, V. Vapnik, and L. Saitta. Support-Vector Networks Editor. *Machine Learning*, 20:273–297, 1995.

[46] H. Cui, O. Turan, and P. Sayer. Learning-based ship design optimization approach. *Computer-Aided Design*, 44(3):186–195, 2012.

[47] V. D'Amato, L. Oneto, A. Camurri, and D. Anguita. The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–10. IEEE, 2022.

[48] J. de Baar, S. Roberts, R. Dwight, and B. Mallol. Uncertainty quantification for a sailing yacht hull, using multi-fidelity kriging. *Computers & Fluids*, 123:185–201, 2015.

[49] K. Deb. *Multiobjective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[50] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 1). `https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1`, 2022.

[51] N. Demo, M. Tezzele, G. Gustin, G. Lavini, and G. Rozza. Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition. *Technology and Science for the Ships of the Future - Proceedings of NAV 2018: 19th International Conference on Ship and Maritime Research*, pages 212–219, 2018.

[52] R. W. Derksen and T. Rogalsky. Bezier-PARSEC: An optimized aerofoil parameterization for design. *Advances in Engineering Software*, 41(7):923–930, 2010.

[53] M. Diez and A. Serani. From uncertainty quantification to shape optimization: Cross-fertilization of methods for dimensionality reduction. In *Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications*, 2021.

[54] M. Diez and A. Serani. Design-space dimensionality reduction in global optimization of functional surfaces: recent developments and way forward. *Ship Technology Research*, 71(2):141–152, 2023.

[55] M. Diez, A. Serani, E. F. Campana, O. Goren, K. Sarioz, D. B. Danisman, G. Grigoropoulos, E. Aloniati, M. Visonneau, P. Queutey, and F. Stern. Multi-Objective Hydrodynamic Optimization of the DTMB 5415 for Resistance and Seakeeping. In *SNAME 13th International Conference on Fast Sea Transportation, Washington, DC*, page D021S005R012, 2015.

[56] S. Ding, X. Xu, and R. Nie. Extreme learning machine and its applications. *Neural Computing and Applications*, 25:549–556, 2014.

[57] G. Dong and H. Liu. *Feature engineering for machine learning and data analytics*. CRC press, 2018.

[58] X. Du, P. He, and J. Martins. Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. *Aerospace Science and Technology*, 113:106701, 2021.

[59] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual review of fluid mechanics*, 51:357–377, 2019.

[60] D. D'Agostino, A. Serani, and M. Diez. Design-space assessment and dimensionality reduction: An off-line method for shape reparameterization in simulation-based optimization. *Ocean Engineering*, 197:106852, 2020.

[61] O. El Moctar, V. Shigunov, and T. Zorn. Duisburg test case: Post-panamax container ship for benchmarking. *Ship Technology Research*, 59(3):50–64, 2012.

[62] M. Emmerich and A. H. Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing*, 17(3):585–609, 2018.

[63] EPRS | European Parliamentary Research Service. Decarbonising maritime transport: The EU perspective. `https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659296/EPRS_BRI(2020)659296_EN.pdf`, 10 2020.

[64] European Commission. Transport emissions. `https://climate.ec.europa.eu/eu-action/transport-emissions_en`, 2016. Accessed: 10th April 2022.

[65] European Commission. Greening shipping. `https://europa.eu/newsroom/events/greening-shipping_en`, 2020. Accessed: 10th April 2022.

[66] European Commission. Reducing emissions from the shipping sector. `https://climate.ec.europa.eu/eu-action/transport-emissions/reducing-emissions-shipping-sector_en`, 2020. Accessed: 10th April 2022.

[67] European Commission Directorate General for Mobility and Transport, CE Delft, and ECORYS. Assessment of impacts from accelerating the uptake of sustainable alternative fuels in maritime transport: final report. https://data.europa.eu/doi/10.2832/76878, 2021.

[68] J. H. Evans. Basic design concepts. *Journal of the American Society for Naval Engineers*, 71(4):671–678, 1959.

[69] S. F. Fahrnholz and J. D. Caprace. A machine learning approach to improve sailboat resistance prediction. *Ocean Engineering*, 257:111642, 12 2022.

[70] Y. Feng, Z. Chen, Y. Dai, F. Wang, J. Cai, and Z. Shen. Multidisciplinary optimization of an offshore aquaculture vessel hull form based on the support vector regression surrogate model. *Ocean Engineering*, 166:145–158, 10 2018.

[71] Feng, Y. and el Moctar, O. and Schellin, T. E. Parametric hull form optimization of containerships for minimum resistance in calm water and in waves. *Journal of Marine Science and Application*, 20(4):670–693, 2021.

[72] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

[73] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.

[74] J. Gain and D. Bechmann. A Survey of Spatial Deformation from a User-Centered Perspective. *ACM Trans. Graph.*, 27(4), 11 2008.

[75] A. García-Gutiérrez, J. Gonzalo, D. Domínguez, and D. López. Stochastic optimization of high-altitude airship envelopes based on kriging method. *Aerosp ace Science and Technology*, 120:107251, 2022.

[76] R. Geirhos, J. H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[77] J. Gerritsma, R. Onnink, and A. Versluis. Geometry, resistance and stability of the Delft Systematic Yacht Hull Series. *International shipbuilding progress*, 28:276–297, 1981.

[78] R. Giusti, D. Manerba, and R. Tadei. Smart Steaming: A New Flexible Paradigm for Synchromodal Logistics. *Sustainability*, 13(9):4635, Apr 2021.

[79] L. Goel. An extensive review of computational intelligence-based optimization algorithms: trends and applications. *Soft Computing*, 24:16519–16549, 2020.

[80] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[81] G. Guan, L. Wang, J. Geng, Z. Zhuang, and Q. Yang. Parametric automatic optimal design of USV hull form with respect to wave resistance and seakeeping. *Ocean Engineering*, 235:109462, 2021.

[82] J. Guerrero, A. Cominetti, J. Pralits, and D. Villa. Surrogate-Based Optimization Using an Open-Source Framework: The Bulbous Bow Shape Optimization Case. *Mathematical and Computational Applications*, 23(4):60, 2018.

[83] W. S. Hall. *Boundary Element Method.* Springer Netherlands, 1994.

[84] Z. Han, C. Xu, L. Zhang, Y. Zhang, K. Zhang, and W. Song. Efficient aerodynamic shape optimization using variable-fidelity surrogate models and multilevel computational grids. *Chinese Journal of Aeronautics*, 33:31–47, 2020.

[85] S. Harries and C. Abt. CAESES–The HOLISHIP Platform for Process Integration and Design Optimization. In A. Papanikolaou, editor, *A Holistic Approach to Ship Design: Volume 1: Optimisation of Ship Design and Operation for Life Cycle*, pages 247–293. Springer International Publishing, 2019.

[86] S. Harries, C. Abt, and K. Hochkirch. Hydrodynamic Modeling of Sailing Yachts. In *Chesapeake Sailing Yacht Symposium*, pages 1–13, 2001.

[87] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer New York Inc., 2001.

[88] X. He, J. Li, C. A. Mader, A. Yildirim, and J. R.R.A. Martins. Robust aerodynamic shape optimization–From a circle to an airfoil. *Aerospace Science and Technology*, 87:48–61, 2019.

[89] L. Hei, Jorge Nocedal, and R. A Waltz. A numerical study of active-set and interior-point methods for bound constrained optimization. In *Modeling, Simulation and Optimization of Complex Processes*, 2008.

[90] A. J. Herrema, N. M. Wiese, C. N. Darling, B. Ganapathysubramanian, A. Krishnamurthy, and M. C. Hsu. A framework for parametric design optimization using isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316: 944–965, 2017.

[91] S. A. Hicks, J. L. Isaksen, V. Thambawita, J. Ghouse, G. Ahlberg, A. Linneberg, N. Grarup, I. Strümke, C. Ellervik, M. S. Olesen, T. Hansen, C. Graff, N.H. Holstein-Rathlou, P. Halvorsen, M. M. Maleckar, M. A. Riegler, and J. K. Kanters. Explaining deep neural networks for knowledge discovery in electrocardiogram analysis. *Scientific Reports*, 11(1):10949, 2021.

[92] T. Hino, F. Stern, L. Larsson, M. Visonneau, N. Hirata, and J. Kim. *Numerical Ship Hydrodynamics: An Assessment of the Tokyo 2015 Workshop*, volume 94. Springer Nature, 2020.

[93] C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics.* Elsevier Ltd., 2007.

[94] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.

[95] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[96] J. P. Horwath, D. N. Zakharov, R. Mégret, and E. A. Stach. Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images. *npj Computational Materials*, 6(1):108, 2020.

[97] H. Hu, G. Zhang, D. Li, Z. Zhang, T. Sun, and Z. Zong. Shape optimization of airfoil in ground effect based on free-form deformation utilizing sensitivity analysis and surrogate model of artificial neural network. *Ocean Engineering*, 257:111514, 2022.

[98] F. Huang and C. Yang. Hull form optimization of a cargo ship for reduced drag. *Journal of Hydrodynamics*, 28(2):173–183, 2016.

[99] G. B. Huang and L. Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16-18):3056–3062, 2007.

[100] G. B. Huang, L. Chen, and C. K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transaction on Neural Networks*, 17(4):879–892, 2006.

[101] G. B. Huang, Q. Y. Zhu, and C. K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[102] J. Huang, Q. Wei, and Y. Liu. Ship design optimization considering probabilistic compliance of decarbonization regulations. In *Advances in the Analysis and Design of Marine Structures*, page 819. CRC Press, 2023.

[103] L. Huang, B. Pena, Y. Liu, and E. Anderlini. Machine learning in sustainable ship design and operation: A review. *Ocean Engineering*, 266, 2022.

[104] L. Huyse and R. M. Lewis. Aerodynamic shape optimization of two-dimensional airfoils under uncertain conditions. Technical report, ICASE, NASA Langley Research Center, 2001.

[105] J. Hwang and J. R.R.A. Martins. Allocation-mission-design optimization of next-generation aircraft using a parallel computational framework. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1662, 2016.

[106] I. F. Ilyas and X. Chu. *Data cleaning*. Morgan & Claypool, 2019.

[107] International Maritime Organization (IMO). Sub-Committee on Ship Design and Construction (SDC). `https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/SDC-Default.aspx`, 2022.

[108] International Maritime Organization (IMO). Statutory Documents - IMO Publications and Documents - International Conventions - SOLAS - International Convention for the Safety of Life at Sea - Chapter II-1 - Construction - Structure, subdivision and stability, machinery and electrical installations. https://www.imorules.com/SOLAS_REGII-1.html, 2023.

[109] T. Ingrassia, A. Mancuso, A. Nigrelli, V. Saporito, and D. Tumino. Parametric hull design with rational Bézier curves. In *Advances on Mechanics, Design Engineering and Manufacturing III: Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing, JCM 2020, June 2-4, 2020*, pages 221–227. Springer, 2021.

[110] M. Innat, M.F. Hossain, K. Mader, and A.Z. Kouzani. A convolutional attention mapping deep neural network for classification and localization of cardiomegaly on chest X-rays. *Scientific Reports*, 13(1):6247, 2023.

[111] International Maritime Organization. 2008 IS Code - International Code on Intact Stability, 2008. In *Statutory Documents - IMO Publications and Documents - International Codes - Part A - Mandatory Criteria - Chapter 2 - General Criteria - 2.3 Severe wind and rolling criterion (weather criterion)*, volume Part A – Mandatory Criteria of *Statutory Documents - IMO Publications and Documents - International Codes*, London, 2008. International Maritime Organization.

[112] International Maritime Organization. Introduction to IMO. https://www.imo.org/en/About/strategy/Pages/default.aspx, 2019. Accessed: 10th April 2022.

[113] International Maritime Organization. Finalization of Second Generation Intact Stability Criteria-Report of the Drafting Group on Intact Stability. In *International Maritime Organization Sub-Committee on Ship Design and Construction*, London, 2020.

[114] P. Jiang, Q. Zhou, and X. Shao. *Surrogate Model-Based Engineering Design and Optimization*. Springer Singapore, 2020.

[115] M. Júnior, J. M, G. L. Halila, Y. Kim, T. Khamvilai, and K. G. Vamvoudakis. Intelligent data-driven aerodynamic analysis and optimization of morphing configurations. *Aerospace Science and Technology*, 121:107388, 2022.

[116] M. Kalikatzarakis, A. Coraddu, M. Atlar, S. Gaggero, G. Tani, and L. Oneto. Physically plausible propeller noise prediction via recursive corrections leveraging prior knowledge and experimental data. *Engineering Applications of Artificial Intelligence*, 118:105660, 2023.

[117] L. B. Kara and K. Shimada. Construction and Modification of 3D Geometry Using a Sketch-based Interface. In T. Stahovich and M.C. Sousa, editors, *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2006.

[118] A. J. Keane and I. I. Voutchkov. Robust design optimization using surrogate models. *Journal of Computational Design and Engineering*, 7(1):44–55, 2020.

[119] S. S. Keerthi and C. J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[120] G. K.W. Kenway and J. R.R.A. Martins. Multipoint aerodynamic shape optimization investigations of the common research model wing. *AIAA Journal*, 54(1):113–128, 2016.

[121] J. A. Keuning. Approximation of the hydrodynamic forces on a sailing yacht based on the 'Delft Systematic Yacht Hull Series'. In *15th International symposium on Yacht Design and Yacht Construction, Amsterdam, 16 November 1998: proceedings*, pages 99–152, 1998.

[122] J. A. Keuning and M. Katgert. A bare hull resistance prediction method derived from the results of the delft systematic yacht hull series extended to higher speeds. In *Internation conference International conference innovation in high performance sailing yachts, Lorient, France*, pages 13–21, 2008.

[123] S. Khan, E. Gunpina, and K. Mert Dogan. A novel design framework for generation and parametric modification of yacht hull surfaces. *Ocean Engineering*, 136:243–259, 2017.

[124] S. Khan, K. Goucher-Lambert, K. Kostas, and P. Kaklis. ShipHullGAN: A generic parametric modeller for ship hull design using deep convolutional generative model. *Computer Methods in Applied Mechanics and Engineering*, 411:116051, 2023.

[125] S.H. Kim and F. Boukouvala. Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. *Optimization Letters*, 14:989–1010, 2020.

[126] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2015.

[127] M. J. Kochenderfer and T. A. Wheeler. *Algorithms for decision making*. MIT Press, 2022.

[128] K. V. Kostas, A. I. Ginnis, C. G. Politis, and P. D. Kaklis. Ship-hull shape optimization with a T-spline based BEM-isogeometric solver. *Computer Methods in Applied Mechanics and Engineering*, 284:611–622, 2 2015.

[129] S. Koziel and L. Leifsson. Multi-level CFD-based Airfoil Shape Optimization With Automated Low-fidelity Model Selection. *Procedia Computer Science*, 18:889–898, 2013.

[130] S. Koziel and L. Leifsson. *Simulation-driven design by knowledge-based response correction techniques*. Springer, 2016.

[131] B. Kulfan. Universal Parametric Geometry Representation Method. *Journal of Aircraft - J AIRCRAFT*, 45:142–158, 01 2008.

[132] B. Kumaraswamy. Neural networks for data classification. In *Artificial Intelligence in Data Mining*, 2021.

[133] A. G. La Rosa, P. Simpson, and R. Zammit. Exploring the Opportunities of Generative Artificial Intelligence in Concept Ship Design. In *International Marine Design Conference. Amsterdam, Netherlands*, 2024.

[134] E. Lazarevska. Comparison of Different Models for Residuary Resistance Prediction. *EUROSIM Congress on Modelling and Simulation, EUROSIM*, pages 511–517, 2018.

[135] L. Leifsson and S. Koziel. Surrogate modelling and optimization using shape-preserving response prediction: A review. *Engineering Optimization*, 48:476–496, 2016.

[136] Y. Lin, J. He, and K. Li. Hull form design optimization of twin-skeg fishing vessel for minimum resistance based on surrogate model. *Advances in Engineering Software*, 123:38–50, 2018.

[137] H. Liu and B. Lang. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences*, 9(20), 2019.

[138] J. Liu, L. Xiong, J. Pei, J. Luo, H. Zhang, and W. Yu. Group-Based Skyline for Pareto Optimal Groups. *IEEE Transactions on Knowledge and Data Engineering*, 33(7):2914–2929, 2021.

[139] X. Liu and P. Lu. Solving nonconvex optimal control problems by convex optimization. *Journal of Guidance, Control, and Dynamics*, 37(3):750–765, 2014.

[140] X. Liu, W. Zhao, and D. Wan. Multi-fidelity Co-Kriging surrogate model for ship hull form optimization. *Ocean Engineering*, 243:110239, 2022.

[141] Z. Liu, J. Wang, and D. Wan. Nonlinear design-space dimensionality reduction in shape optimization. In *International Ocean and Polar Engineering Conference*, 2022.

[142] W. Luo and L. Lan. Design Optimization of the Lines of the Bulbous Bow of a Hull Based on Parametric Modeling and Computational Fluid Dynamics Calculation. *Mathematical and Computational Applications*, 22:4, 2017.

[143] W. Luo, X. Guo, J. Dai, and T. Rao. Hull optimization of an underwater vehicle based on dynamic surrogate model. *Ocean Engineering*, 230:109050, 2021. ISSN 0029-8018.

[144] T. Lutz and S. Wagner. Drag reduction and shape optimization of airship bodies. *Journal of Aircraft*, 35(3):345–351, 1998.

[145] K. O. Lye, S. Mishra, D. Ray, and P. Chandrashekar. Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 374:113575, 2021.

[146] D. J. C. MacKay. Chapter 15: Stuff. In *Sustainable Energy – without the hot air*. Green Books, 2009.

[147] D. C. Maniaci, P. J. Moriarty, M. F. Barone, M. J. Churchfield, M. A. Sprague, and S. Arunajatesan. Wind energy high-fidelity model verification and validation roadmap. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.

[148] I. Marinić-Kragić, D. Vučina, and M. Ćurković. Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape optimization workflow. *Computer-Aided Design*, 80:61–75, 2016.

[149] K. E. Marlantes, S. Kim, and L. A. Hurt. Implementation of the IMO Second Generation Intact Stability Guidelines. *Journal of Marine Science and Engineering*, 10(1):41, 2021.

[150] D. Martineau, S. Stokes, S. Munday, A. Jackson, B. Gribben, and N. Verhoeven. Anisotropic hybrid mesh generation for industrial RANS applications. In *Aerospace Sciences Meeting and Exhibit*, 2006.

[151] A. Massaro and E. Benini. A surrogate-assisted evolutionary algorithm based on the genetic diversity objective. *Applied Soft Computing Journal*, 36:87–100, 2015.

[152] D. A. Masters, N. J. Taylor, T. C. S. Rendall, C. B. Allen, and D. J. Poole. Geometric Comparison of Aerofoil Shape Parameterization Methods. *AIAA Journal*, 55(5): 1575–1589, 2017.

[153] J. McCall. Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184(1):205–222, 2005.

[154] A. D. Meglio, E. F. Combarro, S. González-Castillo, and A. D. Meglio. A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms, 2023.

[155] K. Mehrotra. *Artificial Neural Networks*. MIT Press, 1996.

[156] A. Miao and D. Wan. Hull Form Optimization Based on an NM+CFD Integrated Method for KCS. *International Journal of Computational Methods*, 17(10):1–14, 2020.

[157] K.S. Min and S.H. Kang. Study on the form factor and full-scale ship resistance prediction method. *Journal of marine science and technology*, 15:108–118, 2010.

[158] M. Mittendorf and A. D. Papanikolaou. Hydrodynamic hull form optimization of fast catamarans using surrogate models. *Ship Technology Research*, 68(1):14–26, 2021.

[159] M. Mittendorf, U. D. Nielsen, and H. B. Bingham. Data-driven prediction of added-wave resistance on ships in oblique waves—A comparison between tree-based ensemble methods and artificial neural networks. *Applied Ocean Research*, 118:102964, 2022.

[160] A. Mohammadi and F. Sheikholeslam. Intelligent optimization: Literature review and state-of-the-art algorithms (1965-2022). *Engineering Applications of Artificial Intelligence*, 126:106959, 2023.

[161] W. Moore, H. Mala-Jetmarova, M. Gebreslassie, G. Tabor, M. Belmont, and D. Savic. Comparison of Multiple Surrogates for 3D CFD Model in Tidal Farm Optimisation. *Procedia Engineering*, 154:1132–1139, 2016.

[162] M. Moradi and M. Samwald. Post-hoc explanation of black-box classifiers using confident itemsets. *Expert Systems with Applications*, 165:113941, 2021.

[163] M. Z. Naser and A. H. Alavi. Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. *Architecture, Structures and Construction*, 3(4):1–19, 2021.

[164] R.M. Neal. Regression and Classification Using Gaussian Process Priors. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*. Oxford University Press, 1998.

[165] M. Nelson, D.W. Temple, J.T. Hwang, Y.L. Young, J.R.R.A. Martins, and M. Collette. Simultaneous optimization of propeller-hull systems to minimize lifetime fuel consumption. *Applied Ocean Research*, 43:46–52, 2013.

[166] F. Noblesse, F. Huang, and C. Yang. The Neumann-Michell theory of ship waves. *Journal of Engineering Mathematics*, 79, 04 2013.

[167] H. Nowacki. On the History of Ship Design for the Life Cycle. In *A Holistic Approach to Ship Design: Volume 1: Optimisation of Ship Design and Operation for Life Cycle*, pages 43–73. Springer, 2019.

[168] Delft University of Technology. Delft Systematic Yacht Hull Series Database website. `https://dsyhs.tudelft.nl/`, 2021.

[169] L. Oneto. *Model Selection and Error Estimation in a Nutshell*. Springer, 2020.

[170] L. Oneto, A. Ghio, S. Ridella, and D. Anguita. Support Vector Machines and Strictly Positive Definite Kernel: The Regularization Hyperparameter is More Important than the Kernel Hyperparameters. In *IEEE International Joint Conference on Neural Networks*, 2015.

[171] I. Orlandi, L. Oneto, and D. Anguita. Random Forests Model Selection. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 441–446, 2016.

[172] M. Ou, L. Yan, W. Huang, and T. Zhang. Design exploration of combinational spike and opposing jet concept in hypersonic flows based on CFD calculation and surrogate model. *Acta Astronautica*, 155:287–301, 2019.

[173] S. Özgel Felek. Parametric sailing yacht exterior and interior design. *Tasarım Kuram*, 16(29):1–15, 2020.

[174] A. Papanikolaou. *Risk-based ship design - methods, tools and applications*. Springer Publishers, 2009.

[175] A. Papanikolaou, E. Boulougouris, S. Erikstad, S. Harries, and Austin A. Kana. Ship Design in the Era of Digital Transition: A State-of-the-Art Report. In *Proceedings of 15th International Marine Design Conference. Amsterdam, Netherlands*, 2024.

[176] B. Pena and L. Huang. A review on the turbulence modelling strategy for ship hydrodynamic simulations. *Ocean Engineering*, 241:110082, 2021. ISSN 0029-8018.

[177] F. L. Pérez, J. A. Clemente, J. A. Suárez, and J. M. González. Parametric generation, modeling, and fairing of simple hull lines with the use of nonuniform rational b-spline surfaces. *Journal of Ship Research*, 52(01):1–15, 2008.

[178] P. Pérez and J.A. Suárez. Quasi-developable B-spline surfaces in ship hull design. *Computer-Aided Design*, 39(10):853–862, 2007.

[179] F. Pérez-Arribas. Parametric generation of planing hulls. *Ocean Engineering*, 81: 89–104, 2014.

[180] D. Peri. Optimal ship hull via optimal parameterisation. *Ship Technology Research*, 63(3):159–170, sep 2016.

[181] N. Petacco and P. Gualeni. IMO Second Generation Intact Stability Criteria: General Overview and Focus on Operational Measures. *Journal of Marine Science and Engineering*, 8(7):494, 2020.

[182] W. S. Peters and V. I. Belenky. Second Generation Intact Stability Criteria: An Overview. In *SNAME Maritime Convention*, page D021S012R002, 2022.

[183] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization An overview. *Swarm Intelligence*, 1:33–57, 2007.

[184] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes in Machine Learning*. the MIT Press, 2006.

[185] V. Raul and L. Leifsson. Surrogate-based aerodynamic shape optimization for delaying airfoil dynamic stall using Kriging regression and infill criteria. *Aerospace Science and Technology*, 111:106555, 2021.

[186] Raven, H_C and Van der Ploeg, A and Starke, AR and Eça, L. Towards a CFD-based prediction of ship performance—progress in predicting full-scale resistance and scale effects. *International Journal of Maritime Engineering*, 150(A4):31–42, 2008.

[187] S. Ray. A quick review of machine learning algorithms. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 35–39. IEEE, 2019.

[188] P. Renaud, M. Sacher, and Y. M. Scolan. Multi-objective hull form optimization of a SWATH configuration using surrogate models. *Ocean Engineering*, 256:111209, 2022.

[189] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. In *34th aerospace sciences meeting and exhibit*, page 94, 1996.

[190] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural computation*, 16(5):1063–1076, 2004.

[191] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[192] K. L. Sainani. The Value of Scatter Plots. *PM&R*, 8(12):1213–1217, 2016.

[193] A. Sala, D. Damalas, L. Labanchi, J. Martinsohn, F. Moro, R. Sabatella, and E. Notti. Energy audit and carbon footprint in trawl fisheries. *Scientific data*, 9(1):428, 2022.

[194] N. Salem and S. Hussein. Data dimensional reduction and principal components analysis. *Procedia Computer Science*, 163:292–299, 2019.

[195] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[196] E. Schulz, M. Speekenbrink, and A. Krause. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, 85:1–16, 2018.

[197] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics (ACM)*, 20(4):151–160, 1986.

[198] S. B. Šegota, N. Andelić, J. Kudláček, and R. Čep. Artificial neural network for predicting values of residuary resistance per unit weight of displacement. *Journal of Maritime & Transportation Science*, 57(1):9–22, 2019.

[199] S. B. Šegota, I. Lorencin, M. Šercer, and Z. Car. Determining residuary resistance per unit weight of displacement with symbolic regression and gradient boosted tree algorithms. *Pomorstvo*, 35(2):287–296, 2021.

[200] A. Serani and M. Diez. Parametric model embedding. *Computer Methods in Applied Mechanics and Engineering*, 404:115776, 2023.

[201] A. Serani and M. Diez. A survey on design-space dimensionality reduction methods for shape optimization. *arXiv preprint arXiv:2405.13944*, 2024.

[202] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[203] S. M. Shamsuddin, M. A. Ahmed, and Y. Samian. NURBS skinning surface for ship hull design based on new parameterizaUon method. *Int Journal of Advanced Manufacturing Technologies*, 28:936–941, 2006.

[204] H. Shan, L. Jiang, and C. Liu. Direct numerical simulation of flow separation around a NACA 0012 airfoil. *Computers & fluids*, 34(9):1096–1114, 2005.

[205] L. Shao, A. Mahajan, T. Schreck, and D. J. Lehmann. Interactive regression lens for exploring scatter plots. In *Computer Graphics Forum*, 2017.

[206] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[207] Y. Shi, C. A. Mader, S. He, G. L. O. Halila, and J. R. R. A. Martins. Natural Laminar-Flow Airfoil Optimization Design Using a Discrete Adjoint Approach. *AIAA Journal*, 58(11):4702–4722, 2020.

[208] K. Shukla, V. Oommen, A. Peyvan, M. Penwarden, N. Plewacki, L. Bravo, A. Ghoshal, R. M. Kirby, and G. E. Karniadakis. Deep neural operators as accurate surrogates for shape optimization. *Engineering Applications of Artificial Intelligence*, 129:107615, 2024.

[209] S. N. Skinner and H. Zare-Behtash. State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing Journal*, 62:933–962, 2018.

[210] H. Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339, 2019.

[211] H. Sobieczky. Parametric Airfoils and Wings. In K. Fujii and G. S. Dulikravich, editors, *Recent Development of Aerodynamic Design Methodologies: Inverse Design and Optimization*, pages 71–87. Vieweg+Teubner Verlag, 1999.

[212] F. Stern, R. V. Wilson, H. W. Coleman, and E. G. Paterson. Comprehensive Approach to Verification and Validation of CFD Simulations–Part 1: Methodology and Procedures. *Journal of Fluids Engineering*, 123(4):793–802, 07 2001.

[213] F. Stern, J. Yang, Z. Wang, H. Sadat-Hosseini, M. Mousaviraad, S. Bhushan, and T. Xing. Computational Ship Hydrodynamics: Nowadays and Way Forward. *International Shipbuilding Progress*, 60:3–105, 07 2013.

[214] G. Strasser, K. Takagi, S. Werner, U. Hollenbach, T. Tanaka, K. Yamamoto, and K. Hirota. A verification of the ITTC/ISO speed/power trials analysis. *Journal of Marine Science and Technology*, 20:2–13, 2015.

[215] A. Swarnkar and A. Swarnkar. Artificial Intelligence Based Optimization Techniques: A Review. In A. Kalam, K. R. Niazi, A. Soni, S. A. Siddiqui, and A. Mundra, editors, *Intelligent Computing Techniques for Smart Energy Systems*, pages 95–103. Springer Singapore, 2020.

[216] Y. Tahara, S. Tohyama, and T. Katsui. CFD-based multi-objective optimization method for ship design. *International Journal for Numerical Methods in Fluids*, 52: 499–527, 2006.

[217] Y. Tahara, D. Peri, E. F. Campana, and F. Stern. Single- and multiobjective design optimization of a fast multihull ship: numerical and experimental results. *Journal of Marine Science and Technology*, 16(4):412–433, 12 2011.

[218] J. Tao and G. Sun. Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. *Aerospace Science and Technology*, 92: 722–737, 2019.

[219] M. Terziev, T. Tezdogan, and A. Incecik. Scale effects and full-scale ship hydrodynamics: A review. *Ocean Engineering*, 245:110496, 2022.

[220] K. Thirugnanasambandam, M. R., D. Bhattacharyya, and J. Kim. Directed Artificial Bee Colony algorithm with revamped search strategy to solve global numerical optimization problems. *Automated Software Engineering*, 29(1), 01 2022.

[221] United Nations. COP 21. `https://unfccc.int/event/cop-21`, 2015. Accessed: 10th April 2022.

[222] United Nations. Report of the conference of the parties on its twenty-first session, held in paris from 30 november to 13 december 2015. `https://unfccc.int/documents/9097`, 2016. Accessed: 10th April 2022.

[223] United Nations. COP 26. `https://unfccc.int/event/cop-26`, 2021. Accessed: 10th April 2022.

[224] B. D. Upadhyay, S. S. Sonigra, and S. D. Daxini. Numerical analysis perspective in structural shape optimization: A review post 2000. *Advances in Engineering Software*, 155:102992, 2021.

[225] F. A. C. Viana. A Tutorial on Latin Hypercube Design of Experiments. *Quality and Reliability Engineering International*, 32(5):1975–1985, 2016.

[226] P. A. Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *International Conference on Global Trends in Signal Processing, Information Computing and Communication*, pages 261–265, 2016.

[227] D. Villa, S. Gaggero, A. Coppedè, and G. Vernengo. Parametric hull shape variations by Reduced Order Model based geometric transformation. *Ocean Engineering*, 216 (February 2019):107826, 2020. ISSN 00298018. doi: 10.1016/j.oceaneng.2020.107826. URL `https://doi.org/10.1016/j.oceaneng.2020.107826`.

[228] D. Villa, S. Gaggero, A. Coppede, and G. Vernengo. Parametric hull shape variations by Reduced Order Model based geometric transformation. *Ocean Engineering*, 216: 107826, 2020.

[229] R. Vinuesa and S. L. Brunton. The potential of machine learning to enhance computational fluid dynamics. *arXiv preprint arXiv:2110.02085*, pages 1–13, 2021.

[230] K. Vu, C. D'Ambrosio, Y. Hamadi, and L. Liberti. Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3):393–424, 2017.

[231] M. Wainberg, B. Alipanahi, and B. J. Frey. Are random forests truly the best classifiers? *The Journal of Machine Learning Research*, 17(1):3837–3841, 2016.

[232] J. Walker, A. Coraddu, L. Oneto, and S. Kilbourn. Digital Twin of the Mooring Line Tension for Floating Offshore Wind Turbines. In *IEEE OCEANS, San Diego–Porto*, 2021.

[233] J. Walker, A. Coraddu, M. Collu, and Oneto L. Digital twins of the mooring line tension for floating offshore wind turbines to improve monitoring, lifespan, and safety. *Journal of Ocean Engineering and Marine Energy*, 8:1–16, 2022.

[234] J. M. Walker, A. Coraddu, V. Garofano, and L. Oneto. Artificial Intelligence Based Short-Term Motions Forecasting for Autonomous Marine Vehicles Control. In *Proceedings of the International Ship Control Systems Symposium*, volume 16, page 35, 2022.

[235] J. M. Walker, A. Coraddu, and L. Oneto. A Decoupled Approach to AI-based Design and Optimization of the Delft Systematic Yacht Hull Series. In *Conference on Computer and IT Applications in the Maritime Industries*, pages 209–222, 2023.

[236] J. M. Walker, A. Coraddu, S. Savio, and L. Oneto. Shallow and Deep Learning Models for Vessel Motions Forecasting during Adverse Weather Conditions. *Modelling and Optimisation of Ship Energy Systems 2023*, Dec. 2023.

[237] J. M. Walker, A. Coraddu, and L. Oneto. Power Demand Forecasting of a Hybrid Marine Energy System using Shallow and Deep Learning. In *IEEE OCEANS, Singapore*, 2024.

[238] J. M. Walker, A. Coraddu, and L. Oneto. A review on shape optimization of hulls and airfoils leveraging Computational Fluid Dynamics Data-Driven Surrogate models. *Ocean Engineering*, 312:119263, 2024.

[239] J. M. Walker, A. Coraddu, and L. Oneto. Data-Driven Models for Yacht Hull Resistance Optimization: Exploring Geometric Parameters Beyond the Boundaries of the Delft Systematic Yacht Hull Series. *IEEE Access*, 12:76102–76120, 2024.

[240] J. M. Walker, A. Coraddu, and L. Oneto. Computationally Aware and Physically Plausible Hull Form Optimization Incorporating the IMO Intact Stability Code. *Ocean Engineering*, 341:122607, 2025.

[241] J. M. Walker, A. Coraddu, and L. Oneto. Decoupled Hull Resistance Prediction: A Computationally Aware and Physically Plausible Approach for Data-Driven Surrogates. In *Proceedings of the 12th Conference on Computational Methods in Marine Engineering, Edinburgh, Scotland*, 2025.

[242] Y. Wan, Y. Hou, Y. Xiong, Z. Dong, Y. Zhang, and C. Gong. Interval optimization design of a submersible surface ship form considering the uncertainty of surrogate model. *Ocean Engineering*, 263:112262, 2022.

[243] G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design, Transactions of the ASME*, 129(4):370–380, 2007.

[244] P. Wang, Z. Chen, and Y. Feng. Many-objective optimization for a deep-sea aquaculture vessel based on an improved RBF neural network surrogate model. *Journal of Marine Science and Technology (Japan)*, 26, 2020.

[245] P. Wang, Z. Chen, and Y. Feng. Many-objective optimization for a deep-sea aquaculture vessel based on an improved RBF neural network surrogate model. *Journal of Marine Science and Technology*, 26(2):582–605, 2021.

[246] Q. Wang, J. Wang, J. Chen, S. Luo, and J. Sun. Aerodynamic shape optimized design for wind turbine blade using new airfoil series. *Journal of Mechanical Science and Technology*, 29:2871–2882, 2015.

[247] S. Wang, J. Ren, X. Fang, H. Lin, G. Xu, H. Bao, and J. Huang. IGA-suitable planar parameterization with patch structure simplification of closed-form polysquare. *Computer Methods in Applied Mechanics and Engineering*, 392, 2022.

[248] Y. Wang, J. Joseph, T. P. Aniruddhan Unni, S. Yamakawa, A. Barati Farimani, and K. Shimada. Three-dimensional ship hull encoding and optimization via deep neural networks. *Journal of Mechanical Design*, 144(10):101701, 2022.

[249] Y. Wang, K. Shimada, and A. Barati Farimani. Airfoil GAN: encoding and synthesizing airfoils for aerodynamic shape optimization. *Journal of Computational Design and Engineering*, 10(4):1350–1362, 2023.

[250] D. Wilcox. A half century historical review of the k-omega model. In *29^{th} aerospace Sciences meeting*, page 615, 1991.

[251] D. H. Wolpert. The supervised learning no-free-lunch theorems. *Soft computing and industry: Recent applications*, pages 25–42, 2002.

[252] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[253] X. Yan, J. Zhu, M. Kuang, and X. Wang. Aerodynamic shape optimization using a novel optimizer based on machine learning techniques. *Aerospace Science and Technology*, 86:826–835, 2019.

[254] N. Yilmaz. Comparative energy and environmental assessment of battery technologies and alternative fuels in sustainable aviation. *International Journal of Green Energy*, pages 1–10, 2022.

[255] A. Yrjänäinen, T. Johnsen, J. S. Dæhlen, H. Kramer, and R. Monden. Market conditions, mission requirements and operational profiles. In *A Holistic Approach to Ship Design: Volume 1: Optimisation of Ship Design and Operation for Life Cycle*, pages 75–121. Springer, 2019.

[256] H. Zakerdoost and H. Ghassemi. Probabilistic surrogate-based optimization of ship hull-propulsor design with bi-level infill sampling technique. *Ocean Engineering*, 286:115614, 2023.

[257] B. J. Zhang and Z. X. Zhang. Research on theoretical optimization and experimental verification of minimum resistance hull form based on rankine source method. *International Journal of Naval Architecture and Ocean Engineering*, 7(5):785–794, 2015.

[258] D. Zhang, Z. Wang, H. Ling, and X. Zhu. Kriging-based shape optimization framework for blended-wing-body underwater glider with NURBS-based parametrization. *Ocean Engineering*, 219:108212, 2021.

[259] H. Zhang, Y. Yu, Q. Zhang, Y. Yang, H. Liu, and Y. Lin. A bidirectional collaborative method based on an improved artificial fish swarm algorithm for ship pipe and equipment layout design. *Ocean Engineering*, 296:117045, 2024.

[260] S. Zhang, T. Tezdogan, B. Zhang, and L. Lin. Research on the hull form optimization using the surrogate models. *Engineering Applications of Computational Fluid Mechanics*, 15(1):747–761, 2021.

[261] T. Zhang, W. Huang, Z. Wang, and L. Yan. A study of airfoil parameterization, modeling, and optimization based on the computational fluid dynamics method. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 17(8):632–645, 2016.

[262] X. Zhang, F. Xie, T. Ji, Z. Zhu, and Y. Zheng. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 373:113485, 2021.

[263] Y. Zhang, N. Ma, X. Gu, and Q. Shi. Geometric space construction method combined of a spline-skinning based geometric variation method and PCA dimensionality reduction for ship hull form optimization. *Ocean Engineering*, 302:117604, 2024.

[264] H. Zhao, T. Icoz, Y. Jaluria, and D. Knight. Application of data-driven design optimization methodology to a multi-objective design optimization problem. *Journal of Engineering Design*, 18(4):343–359, 2007.

Glossary

CFD Computational Fluid Dynamics: a branch of fluid mechanics that uses numerical analysis and algorithms to solve and analyze problems involving fluid flows.

DDS Data-Driven Surrogate: a data-based model that approximates a computationally expensive input–output relationship, often used to replace simulations in optimization.

DOE Design of Experiments: a statistical approach used to plan, conduct, and analyze experiments efficiently, especially for exploring large design spaces.

EFD Experimental Fluid Dynamics: physical experiments used to study fluid flow behavior, often used to validate CFD results.

FFD Free-Form Deformation: a shape parametrization method where an object is embedded in a flexible lattice that can be manipulated to deform the object smoothly.

GHG Greenhouse Gas: gases such as CO_2 and NO_x that trap heat in the atmosphere and contribute to global warming and climate change.

IMO International Maritime Organization: a specialized agency of the United Nations responsible for regulating shipping and maritime safety.

KPI Key Performance Indicator: a quantifiable measure used to evaluate the performance or effectiveness of a design, system, or process.

LOO Leave-One-Out: a cross-validation technique where each sample is used once as a test set while the remaining samples form the training set.

PCA Principal Component Analysis: a dimensionality reduction technique that transforms a dataset into a set of linearly uncorrelated variables called principal components.

RANS Reynolds-Averaged Navier–Stokes: a set of equations used in CFD that model the average flow behavior by decomposing turbulent flow into mean and fluctuating components.

Curriculum Vitæ

Jake Matthew Walker

Jake was born in Edinburgh, Scotland, in 1997. He attended Madras College Secondary School in St. Andrews from 2009 to 2015. He then studied at the University of Strathclyde in Glasgow from 2015 to 2020, where he obtained an MEng in Mechanical Engineering (with distinction). Between 2020 and 2021, he began his research at the Department of Naval Architecture and Marine Engineering at the University of Strathclyde, under the supervision of Dr. Andrea Coraddu (Delft University of Technology, Netherlands) and Prof. Luca Oneto (University of Genoa, Italy). In November 2021, he commenced his doctoral studies at Delft University of Technology, continuing under the supervision of Dr. Coraddu and Prof. Oneto. The results of his research are presented in this thesis. He defended his Ph.D. in December 2025 and currently lives in The Hague, Netherlands.

List of Publications

Journal Articles

J6. A. Coraddu, J. M. Walker, A. Mikelic, A. Ponte, and L. Oneto. Artificial Intelligence-Based Surrogate Models for Ship Resistance and Wave Pattern Prediction during Early-Stage Ship Design. *Ocean Engineering (Submitted)*, 2025

J5. J. M. Walker, A. Coraddu, and L. Oneto. Computationally Aware and Physically Plausible Hull Form Optimization Incorporating the IMO Intact Stability Code. *Ocean Engineering*, 341:122607, 2025

J4. J. M. Walker, A. Coraddu, and L. Oneto. A review on shape optimization of hulls and airfoils leveraging Computational Fluid Dynamics Data-Driven Surrogate models. *Ocean Engineering*, 312:119263, 2024

J3. J. M. Walker, A. Coraddu, and L. Oneto. Data-Driven Models for Yacht Hull Resistance Optimization: Exploring Geometric Parameters Beyond the Boundaries of the Delft Systematic Yacht Hull Series. *IEEE Access*, 12:76102–76120, 2024

J2. A. Coraddu, L. Oneto, J. Walker, K. Patryniak, A. Prothero, and M. Collu. Floating offshore wind turbine mooring line sections health status nowcasting: From supervised shallow to weakly supervised deep learning. *Mechanical Systems and Signal Processing*, 216:111446, 2024

J1. J. Walker, A. Coraddu, M. Collu, and Oneto L. Digital twins of the mooring line tension for floating offshore wind turbines to improve monitoring, lifespan, and safety. *Journal of Ocean Engineering and Marine Energy*, 8:1–16, 2022

Conference Papers

C7. A. Coraddu, J. M. Walker, R. Dick, and L. Oneto. From Fouling to Fuel Savings: A Data-driven Assessment of Hull & Propeller Cleaning Effectiveness. *Modelling and Optimisation of Ship Energy Systems 2025*, Sept. 2025

C6. J. M. Walker, A. Coraddu, and L. Oneto. Decoupled Hull Resistance Prediction: A Computationally Aware and Physically Plausible Approach for Data-Driven Surrogates. In *Proceedings of the 12th Conference on Computational Methods in Marine Engineering, Edinburgh, Scotland*, 2025

C5. J. M. Walker, A. Coraddu, and L. Oneto. Power Demand Forecasting of a Hybrid Marine Energy System using Shallow and Deep Learning. In *IEEE OCEANS, Singapore*, 2024

C4. J. M. Walker, A. Coraddu, S. Savio, and L. Oneto. Shallow and Deep Learning Models for Vessel Motions Forecasting during Adverse Weather Conditions. *Modelling and Optimisation of Ship Energy Systems 2023*, Dec. 2023 💬🏆

C3. J. M. Walker, A. Coraddu, and L. Oneto. A Decoupled Approach to AI-based Design and Optimization of the Delft Systematic Yacht Hull Series. In *Conference on Computer and IT Applications in the Maritime Industries*, pages 209–222, 2023 💬📄

C2. J. M. Walker, A. Coraddu, V. Garofano, and L. Oneto. Artificial Intelligence Based Short-Term Motions Forecasting for Autonomous Marine Vehicles Control. In *Proceedings of the International Ship Control Systems Symposium*, volume 16, page 35, 2022

C1. J. Walker, A. Coraddu, L. Oneto, and S. Kilbourn. Digital Twin of the Mooring Line Tension for Floating Offshore Wind Turbines. In *IEEE OCEANS, San Diego–Porto*, 2021

Book Chapters

BC1. A. Coraddu, M. Kalikatzarakis, J. Walker, D. Ilardi, and L. Oneto. Data science and advanced analytics for shipping energy systems. In *Sustainable Energy Systems on Ships*, pages 303–349. Elsevier, 2022

📄 *Included in this thesis.*
🏆 *Won a best paper award.*
💬 *Conferences where I presented.*

Acknowledgments

We stand on the shoulders of giants.

First and foremost, I want to acknowledge the supervision, guidance, and friendship of my promotor over the last 5 years. Andrea, thank you. Luca, thank you for your continued support and guidance. The completion of this thesis marks the culmination of an extremely rewarding journey. My heartfelt thanks go to the friends and family who made this possible. *Y a mi esposa, Christy, gracias por hacer de cada día un lugar al que pertenezco.*

Jake
Den Haag. November 1, 2025.