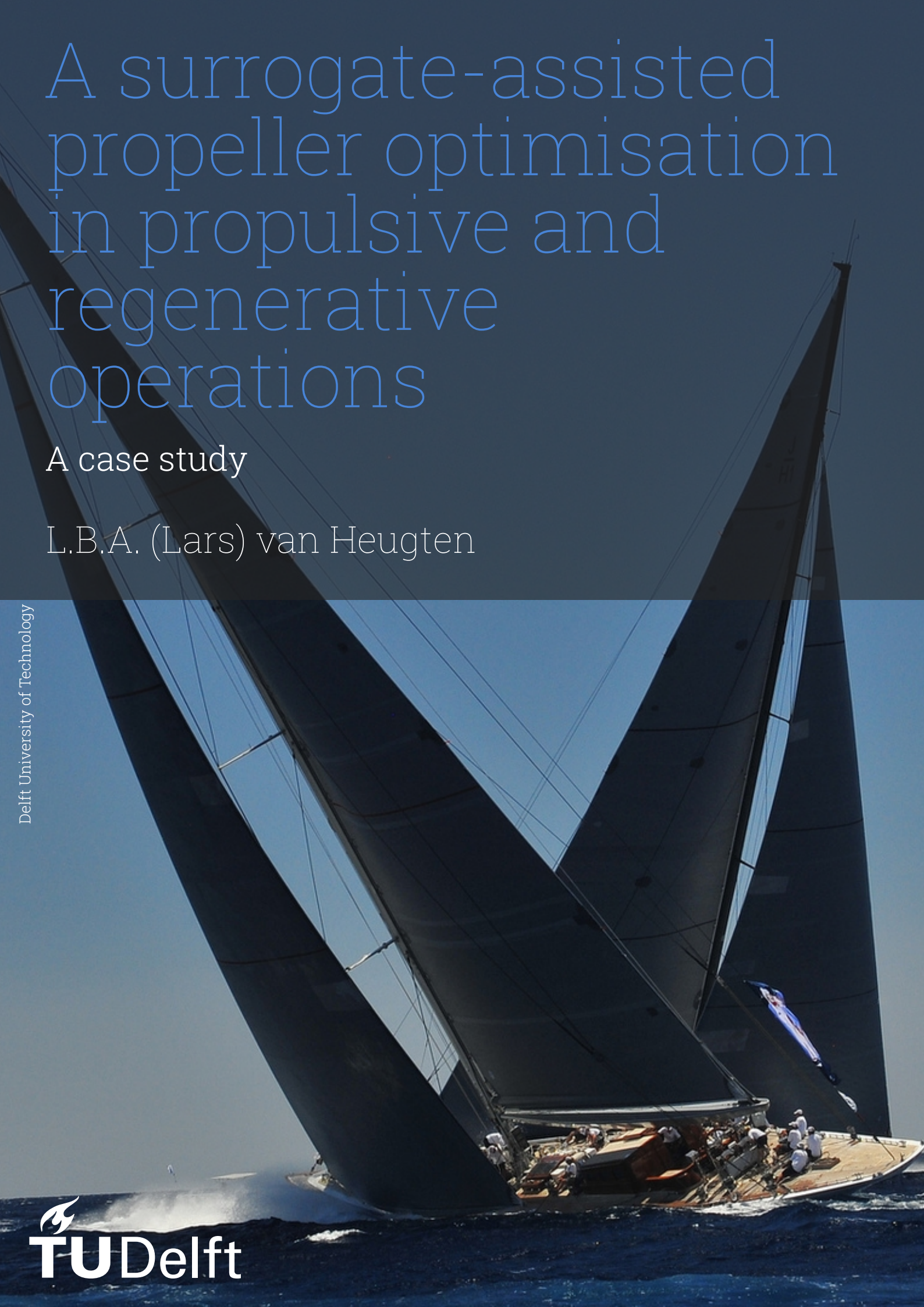
# A surrogate-assisted propeller optimisation in propulsive and regenerative operations

## A case study

L.B.A. (Lars) van Heugten

Delft University of Technology



TU Delft

Thesis for the degree of MSc in Marine Technology in the specialisation of Ship hydromechanics

# A surrogate-assisted propeller optimisation in propulsive and regenerative operations

## A case study

by

L.B.A. (Lars) van Heugten

Performed at:

MARIN

This thesis MT.22/23.035.M. is classified as confidential in accordance with general conditions for projects performed by the TU Delft.

July 12, 2023


| | |
|---|---|
| **Company supervisors** | |
| Responsible supervisor: | prof.dr.ir. J.T.C. van Terwisga |
| E-mail | T.v.Terwisga@marin.nl |
| Daily supervisors | dr.ir. E.J. Foeth, dr.ir. T.P. Scholcz |
| E-mail | E.J.Foeth@marin.nl, T.P.Scholcz@marin.nl |
| **Thesis exam committee:** | |
| chair/ Responsible professor: | prof.dr.ir. J.T.C. van Terwisga |
| Staff member: | dr.ir. H.J. de Koning Gans |
| Staff member: | ir. L.J. Lagendijk |
| company member: | dr.ir. E.J. Foeth |
| company member: | dr.ir. T.P. Scholcz |
| Company member: | ir. M. Leslie-Miller |
| **Author details:** | |
| Studynumber: | 4549422 |
| Author contact e-mail: | larsvanheugten@gmail.com |


Cover: Sailing Yacht HD Wallpaper - WallpaperFx

TU Delft

# Preface

This thesis was written to graduate from the master Marine Technology at Delft University. During the past nine months I had the pleasure to further broaden my knowledge in the field of ship hydromechanics and propulsion in combination with computational fluid dynamics while learning about surrogate-assisted optimisation methods.

This project would not have been possible without the help of multiple people. First I would like to thank Tom van Terwisga for proposing this research topic at MARIN. Secondly, I would like to express my appreciation for the daily help Evert-Jan Foeth and Thomas Scholcz provided throughout my thesis. Finally I would like to thank Jidde Huisinga and Mark Leslie-Miller for providing knowledge about sailing yachts and their propulsive needs. Lastly I owe a lot from the friends I made over the last years in Delft.

*L.B.A. (Lars) van Heugten*
*Delft, July 2023*

# Summary

Engineering has become more and more optimisation and simulation based. This caused an increase in the use of optimisation methods for complex problems. This research focuses on the implementation of a surrogate-assisted optimisation method for propeller optimisation, to design a better propeller in a shorter timespan. Therewith reducing both: computational expenses, and fuel consumption when a propeller has been built. This report was written to answer the research question of this thesis:

*"How can surrogate-assisted optimisation methods be used to design a propeller for both propulsive and regenerative operations?"*

The research question tries to reach two objectives. The first goal is to implement a surrogate-assisted method for propeller optimisation. The other objective is to design a propeller for both propulsive as well as for regenerative operations.
To succeed, this thesis has been limited to the testing and coupling of SAMO-COBRA (A Fast Surrogate Assisted Constrained Multi-objective Optimization Algorithm) [53] with PropArt. The testing has been based on benchmark tests where SAMO-COBRA was tested against CMOPSO and NSGA-II (the two algorithms that MARIN currently uses for propeller optimisation). The propeller optimisation is limited to the evaluation of three configurations for a provided Hundested propeller. The optimisation process is constrained regarding cavitation, geometry and separation of flow. For the propeller optimisation the scope of this research focuses on the modelling of the turbine while using the Boundary Element Method (BEM) based program PROCAL. This is needed since PROCAL is made for propeller design and therefore was not capable of modelling turbine wakes.
Within MARIN, propeller optimisation is currently based on genetic- and particle-swarm optimisers. These methods are fast but tend to throw away a lot of valuable data when closing in on the Pareto Front. SAMO-COBRA is proposed as an alternative since it is able to learn from all evaluated propellers. Initial benchmark tests where SAMO-COBRA is compared to NSGA-II and CMOPSO show that SAMO-COBRA is able to find a larger hypervolume in a fixed number of iterations for most computations. Therefore it is expected to save time and reduce computational expenses. Therewith reducing the carbon footprint of the optimisation process. The optimisation of a propeller that can be used for propulsion and regeneration is a new concept with a complex trade-off between two objectives. The relevance however is getting more important since it has great potential for vessels with sail assisted propulsion. Research in the maritime industry currently focuses on the potential of these sail assisted vessels and therefore it is to be expected that regenerative capabilities of a propeller will get more and more important over the coming years.
The next chapters will cover the content of this research. First the basics regarding propeller design and the possible configurations for propulsive and regenerative operation are covered. For the remainder of the research two separate designs, a conventional CPP where regeneration occurs at a negative angle-of-attack and a fully reversible propeller, will be considered.
The third chapter will cover the Boundary Element Momentum theory, the mathematical method that is the basis of PROCAL. It explains the principles and shortcomings of BEM together with the solutions that PROCAL uses to solve these.
Chapters four and five cover the corrections that are made to improve the wake expansion and wake alignment for high J values. For the wake expansion three methods are proposed. After comparing the open water diagram of a F4-63-0.6 propeller that is computed by PROCAL, and the one that was made after physical testing. It can be concluded that the disk theory is the most versatile implementation for propellers that operate at a range of advance ratio. Result show that from the two proposed methods, the method where the wake pitch is prescribed by the advance ratio results in the best wake alignment.
After implementing the changes needed to model turbine operations, the focus of the research switches to the optimisation process. First SAMO-COBRA is tested against NSGA-II and CMOPSO for the same variety of optimisation problems, using three different generation sizes. After a budget of 1000

evaluations SAMO-COBRA finds the largest hypervolume for the majority of the evaluated analytical functions.
As SAMO-COBRA outperformed NSGA-II and CMOPSO it is valuable to create a coupling between SAMO-COBRA and PropArt. The method used to allow for the collaboration of SAMO-COBRA and PropArt is explained. This chapter also provides an in depth explanation of the test case, its objectives and its constraints.
After evaluating the test case, the results are provided. SAMO-COBRA is not able to find feasible results for any of the optimisation problems. After removing the cavitational constraints a new analysis of the results is done, it is found that a propeller that is rotated 180 degrees can provide the highest regenerative power.
The discussion proposes multiple hypotheses that cause the lack of feasible results of the original test case. The proposed causes are: errors in the PropArt model, problems in the compiled PropArt version or a too low convergence level of COBYLA. The chapter also focusses on the limitations regarding the model and design problem as well as recommendations for future research.

# Contents

# List of Figures

# List of Tables

# Nomenclature

## Abbreviations

| Abbreviation | Definition |
|---|---|
| AoA | Angle-of-Attack |
| BEM | Boundary Element Method |
| CFD | Computational Fluid Dynamics |
| CPP | Controllable Pitch Propeller |
| DoE | Design of Experiments |
| DoF | Degrees of Freedom |
| FPP | Fixed Pitch Propeller |
| LHS | Latin Hypercube Sampling |
| MDoE | Modern Design of Experiments |
| OA | Orthogonal-Array |
| RANS | Reynolds-Averaged Navier-Stokes |
| RBF | Radial Basis Function |
| RSM | Response surface model |

## Symbols

| Symbol | Definition | Unit |
|---|---|---|
| $A$ | Cross sectional area | [m$^2$] |
| $a$ | acceleration | [m/s$^2$] |
| $a$ | The axial velocity induction factor | [-] |
| $a'$ | The radial velocity induction factor | [-] |
| $C_d$ | Drag coefficient | [-] |
| $C_l$ | Lift coefficient | [-] |
| $C_p$ | Power coefficient | [-] |
| $C_{p,min}$ | Minimum pressure coefficient | [-] |
| $c_r$ | Chord length | [m] |
| $C_T^*$ | Four quadrant thrust coefficient | [-] |
| $C_Q^*$ | Four quadrant torque coefficient | [-] |
| $C_x$ | Axial force coefficient | [-] |
| $D$ | Diameter | [m] |
| $dT$ | Elemental momentum | [kgm/s] |
| $F_d$ | Drag force | [N] |
| $F_L$ | Lift force | [N] |
| $g$ | local gravity | [m/s$^2$] |
| $h$ | Distance submerged | [m] |
| $h$ | particle height | [m] |
| $I$ | moment of inertia | [m$^4$] |
| $i_g$ | Geometry generated rake | [m] |
| $i_s$ | Skew induced rake | [m] |
| $i_t$ | Total rake | [m] |
| $J$ | Advance ratio | [-] |
| $K_T$ | Thrust coefficient | [-] |
| $K_Q$ | Torque coefficient | [-] |

| Symbol | Definition | Unit |
|---|---|---|
| $\dot{L}$ | Elemental momentum | [kgm/s] |
| $\dot{m}$ | Mass flow | [kg/s] |
| $n_p$ | propeller rps | [rps] |
| $P$ | Power | [W] |
| $P_{thrust}$ | Thrust power | [W] |
| $P_{added}$ | Added power | [W] |
| $p$ | Pressure | [N/m$^2$] |
| $p_{atm}$ | Atmospheric pressure | [N/m$^2$] |
| $p_v$ | Vapour pressure | [N/m$^2$] |
| $p_0$ | Upstream pressure | [N/m$^2$] |
| $Q$ | Torque | [N·m] |
| $R$ | Total blade span | [m] |
| $r$ | local blade span | [m] |
| $T$ | Thrust | [N] |
| $v$ | Velocity | [m/s] |
| $v_A$ | Advanced velocity | [m/s] |
| $w$ | wake factor | [-] |
| $Z$ | Number of blades | [-] |
| $\alpha$ | Angle-of-Attack | [°] |
| $\beta$ | Blade angle | [°] |
| $\beta$ | Angle of inflow | [°] |
| $\Gamma$ | Circulation | [m$^2$/s] |
| $\Delta x$ | Horizontal change | [m] |
| $\epsilon$ | Drag lift ratio | [-] |
| $\eta_0$ | Open water efficiency | [-] |
| $\eta_i$ | Ideal axial efficiency | [-] |
| $\theta_r$ | Rake angle | [°] |
| $\theta_s$ | Skew angle | [°] |
| $\theta_{sp}$ | Propeller skew angle | [°] |
| $\lambda$ | Tip speed ratio | [-] |
| $\rho$ | Density | [kg/m$^3$] |
| $\sigma$ | Cavitation number | [-] |
| $d\tau$ | angular momentum | [kgm$^2$/s$^2$] |
| $\phi$ | Incoming flow angle | [rad] |
| $\phi$ | Velocity potential | [-] |
| $\Psi$ | Blade loading coefficient | [-] |
| $\Omega$ | Radial velocity | [rad/s] |
| $\omega$ | Radial wake velocity | [rad/s] |

# 1 Introduction

This thesis is written to reach two goals. The first goal is to implement a surrogate-assisted optimisation method for propeller optimisation. The other objective is to design a propeller for both propulsive as regenerative operations. To succeed, the first objective is limited to the testing of and the coupling of SAMO-COBRA (a Fast Surrogate Assisted Constrained Multi-objective Optimization Algorithm) with Propart. The second objective is based on the test case that will be used to test SAMO-COBRA. The case optimises a Hundested propeller for propulsive and regenerative operation to be able to harvest energy from the wake while under sail. The propeller is to be designed for project ZERO, a sailing vessel that has been designed to cross the Atlantic ocean with net zero emissions. A cross sectional view of Project ZERO can be seen in Figure 1.1. To successfully complete this thesis the following research question is formulated:

*"How can surrogate-assisted optimisation methods be used to design a propeller for both propulsive and regenerative operations?"*

To answer this question, six sub-questions have been formulated.

1. How do turbines and propellers work, what do typical designs look like and what are typical design variables?
2. What possible propeller configurations would be able to perform as propulsor as well as a turbine?
3. What surrogate assisted optimisation methods are available and which one is the best to implement for this research?
4. What (if any) modifications are needed within the PROCAL software to allow optimisation for regenerative and propulsive mode?
5. How can surrogate-assisted optimisation be coupled to PROPART, and how does it perform in comparison to NSGA-II and CMOPSO?
6. Can surrogate-assisted optimisation be used for the optimisation of a propeller with propulsive and regenerative operations?

The research will use MARIN's (The Maritime Research Institute Netherlands) in-house propeller tool, PROCAL. PROCAL is a Boundary Element Method (BEM) based program that can be used to compute propellers. The design framework PropArt has been built around PROCAL to be able to optimise propellers. This optimisation is currently done by either a genetic algorithm (NSGA-II) or by a particle swarm optimiser (CMOPSO). NSGA-II and CMOPSO are both based on elitism, but their main difference is the way new particles are defined. NSGA-II uses a crossover principle between the variables of elite samples. CMOPSO uses the relative velocities of different elite particles. The main disadvantage regarding CMOPSO and NSGA-II is that non-elite particles will not be considered and therefore the algorithm will not learn from this evaluation. SAMO-COBRA on the other hand uses radial basis functions to predict the location of new samples. To train the radial basis functions, SAMO-COBRA will use all evaluated samples. Therefore, it is to be expected that SAMO-COBRA will be able to reach a better Pareto Front within the same computational budget.

This research proves to be relevant in two fields of active research. If fewer iterations are needed to reach the same solution this will reduce the computational costs, causing a reduction in power consumption and lead time for the optimisation process. The test case provides an interesting trade-off between two optimisation objectives. Although the application on luxury sailing yachts seems limited to a very specific market, this concept might prove useful in the design of wind assisted vessels. When operating in overpowered conditions it can be useful to regenerate energy that can be stored and later be used for propulsive purposes.
Within the first part of the thesis a definition study has been done, providing answers for sub-questions 1-3. This report will cover sub-questions 4-6. First the necessary knowledge about propeller design and PROCAL are covered. sub-question 4 will be covered in chapters 4 and 5. By comparing open water diagrams of an F-series propeller with PROCAL results it was found that the implementation of a new model for wake expansion and wake pitch alignment can result in a large improvement of the thrust- and torque coefficients at high advance ratios. Chapter 6 elaborates on SAMO-COBRA and explains how its performance was tested against NSGA-II and CMOPSO. Chapter 7 explains how SAMO-COBRA is coupled to PropArt. The testcase is introduced by explaining the constraints, objectives and different computations that are done. The results of the test case are presented in chapter 8. Since no feasible results are found with the SAMO-COBRA optimisation the discussion (chapter 9) mainly focuses on possible causes for this problem. Chapter 9 also discusses the limitations of this research and the recommendations for future research.

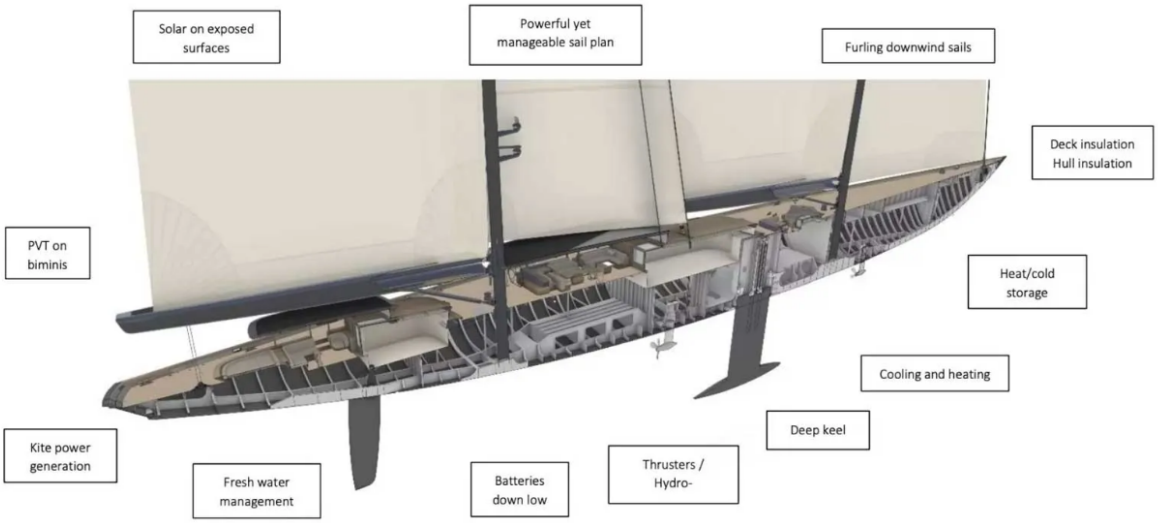



**Figure 1.1:** A cross sectional view of project ZERO [12]

# 2

# Propellers

To understand the optimisation of propellers the basics of propeller geometry and the different possible configurations should be known. Within the scope of this research there are four cases of interest.

1. Two separate designs, one for regeneration and one for propulsion.
2. A controllable pitch propeller (CPP) with a normal operational domain (up to feathering mode)
3. a CPP that can rotate over feathering mode
4. a CPP with the capability to rotate over 180 degrees and is therefore reversely loaded

In the propulsive domain all concepts have the same working principles and blade characteristics. Therefore these will be covered as one case. The challenges mainly occur when the propeller has to operate in its so-called turbine or trailing mode. These conditions have been visualised in Figure 2.1 where the y-axis is the advanced velocity and the x-axis is the rotational velocity at the propeller blade. As a turbine the CPP in the normal domain, the blades are loaded under a negative Angle-of-Attack, the CPP blades that can rotate over feathering mode can rotate further and therefore the blade will be loaded at a normal Angle-of-Attack but at a negative pitch (this would result in a propeller that is loaded in the fourth quadrant). For a CPP that can rotate over 180 degrees the blades can operate in a reversed load condition. In other words, the flow will enter the blade at the trailing edge and will leave at the leading edge. This section will first cover the blade geometry and the main aspects that can be altered. Later on the different load cases will be discussed together with their challenges.

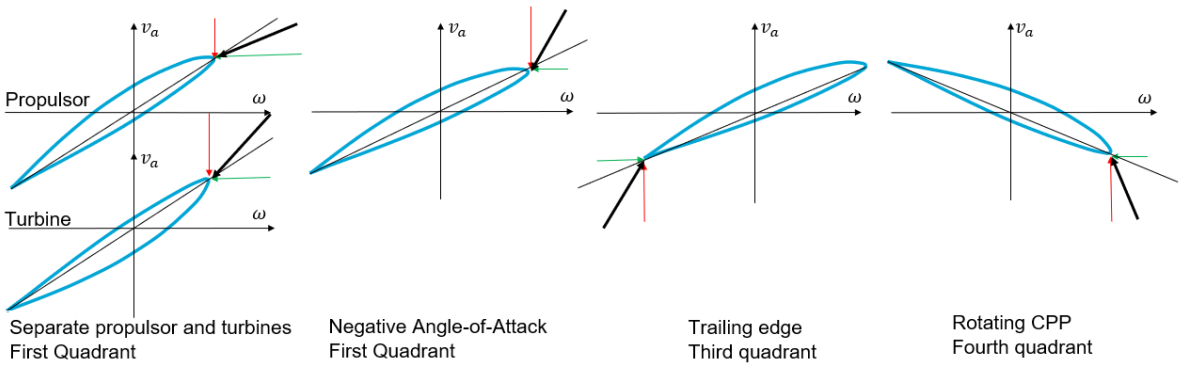



**Figure 2.1:** Four possible configurations for regeneration

## 2.1. Propeller geometry

The art of propeller design is based on tuning different design parameters such that a good trade-off between behind efficiency and cavitational behaviour is obtained. To be able to optimise a desired propeller it is important to understand the influence of the different parameters. Therefore this section will discuss the most important design parameters, these can be seen in Figure 2.2.

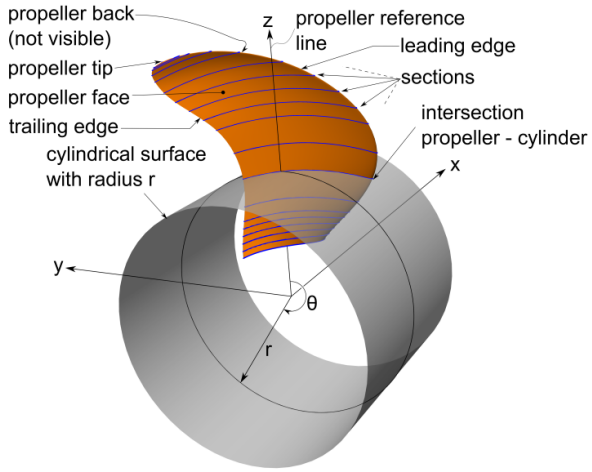



**Figure 2.2:** Some different propeller variables [16]

### 2.1.1. Pitch

Pitch is the ratio of the horizontally covered distance by a propeller blade over one rotation. This ratio can have multiple definitions since the pitch line is defined differently in different literature [8]:

- nose-tail pitch
- face pitch
- effective or no-lift pitch
- hydrodynamic pitch

Here the hydrodynamic pitch is the angle of the inflow vector ($\beta_i$), The definition of the other pitch lines can be seen in Figure 2.3. Today the most commonly used definition however is the nose-tail pitch. [8]

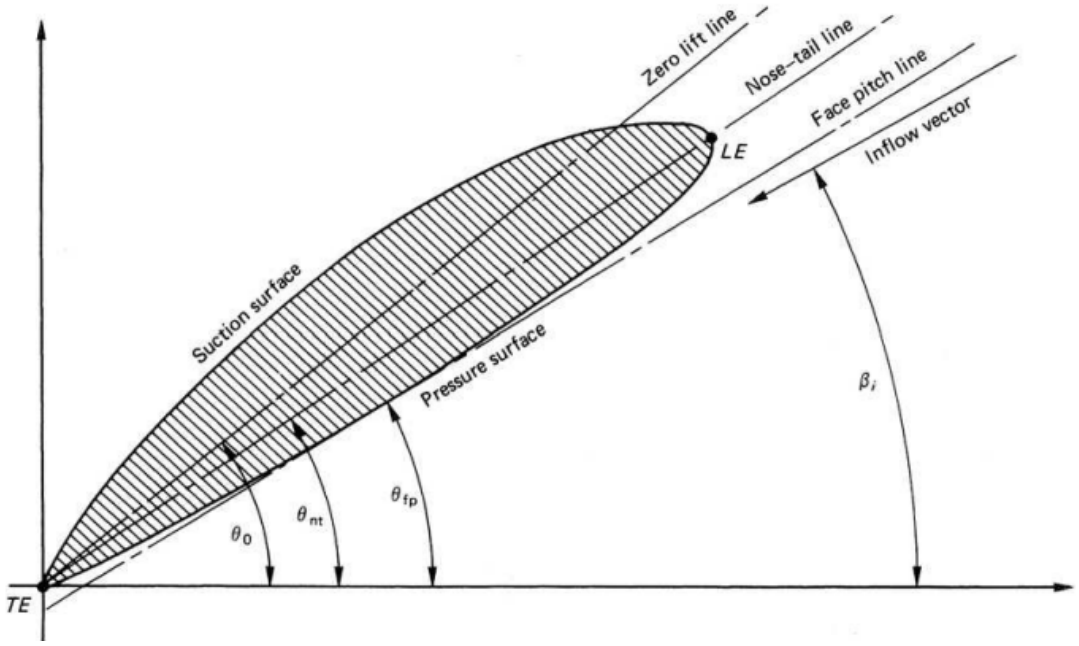



**Figure 2.3:** The different pitch lines [8]

The pitch is an important design parameter as it is of influence of the eventual Angle-of-Attack of the flow. Pitch should be considered together with the rotational speed. If the rotational speed is low the pitch should be high and visa versa. A visualisation can be seen in Figure 2.4. Since the variations

of Angle-of-Attack are often small, they can be tuned by the rotational speed (for normal operations). Therefore propeller stall is rarely a problem that has to be considered in propeller design.

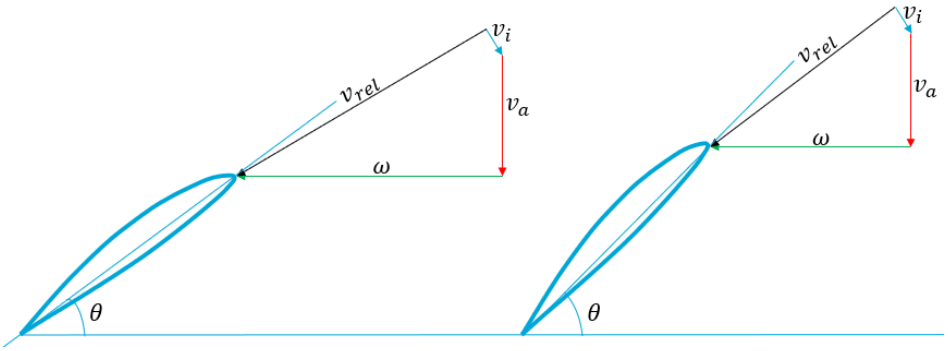



**Figure 2.4:** A change in relative velocity can be noticed when the rotational velocity is changed

The rotational speed and advance velocity, in combination with the variation in pitch over the blade will mainly determine the blade loading. It should be mentioned that the pitch angle of the blade varies over the radius since the rotational speed ($v_{rot} = \Omega r$) will increase linearly over the radius. In basic propeller design an engineer will try to avoid, or at least delay certain types of cavitation [8].

$$C_p = \frac{p - p_0}{\frac{1}{2}\rho v_{rel}^2} \tag{2.1}$$

A useful tool can be the cavitation bucket. This bucket is a collection of the local $C_{p,min}$ and therefore it is rather easy to observe the cavitational behaviour at certain Angles-of-Attack. One can recall the formulation of $C_p$ from. [28]
Nowadays the local $C_{p,min}$ is most often computed using RANS (Reynolds-Averaged Navier-Stokes) or BEM methods because these methods are easier to use, and more accurate when considering 3D-effects. Although the increase in accuracy, creating a cavitation bucket for a propeller in behind conditions is a tedious and time consuming job. In Figure 2.5 a high resolution graph of a cavitation bucket can be seen over a symmetrical foil. Every line is a function of $C_p$ and the Angle-of-Attack at a certain location on the foil. Therefore the total cavitation bucket will provide the smallest $C_p$ that was found on the entire profile. As a result, it is possible to find a $C_{p,min}$ value for a given Angle-of-Attack [28].

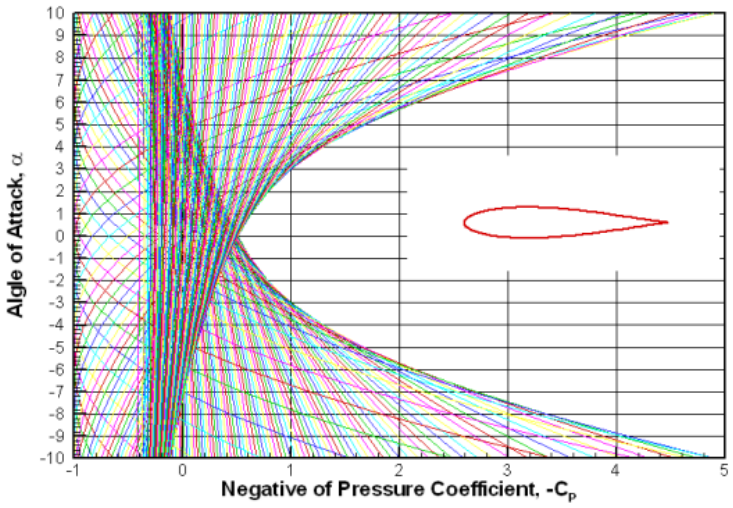



**Figure 2.5:** The cavitation bucket of a symmetric foil [28]

The cavitation bucket can be used in propeller optimisation as an approach to determine whether a profile will be cavitation free. This also clearly highlights the different parameters that have influence

on the prevention of cavitation: the Angle-of-Attack (so indirect the rotational speed and pitch), camber and thickness. If this is not possible the cavitation bucket can also display where cavitation will start to occur. Lets for example consider an Angle-of-Attack of $5°$. We can now see that this results in a $-C_{p,min} \approx 1.7$. Therefore cavitation can be expected from a cavitation number that is smaller than $\sigma = 1.7$. On the other hand, suppose $\sigma$ is fixed at $\sigma = 2.0$. Then it is evident that the propeller can be operated cavitation free if the angle of attack stays below $5.8°$ [28]. The different colours in the graph are the 180 chordwise locations that have been considered. The first line that is crossed is the most probable location where cavitation starts to occur [28].

### 2.1.2. Skew and rake

Skew is the asymmetry a blade has in the yz-plane (as defined in Figure 2.6). Here two definitions are governing, $\theta_s(x)$ and $\theta_{sp}$. The skew angle, $\theta_s(x)$, is the angle between the directrix and a line through the shaft centre line and the mid-chord point of a section as a function of the radius. Therefore the skew angle usually varies over the radius. The propeller skew angle $\theta_{sp}$ is the greatest angle between the shaft centre line and the mid-chord point of the entire blade. The definitions are illustrated in Figure 2.6 [8]. Skew (in combination with lowered tip loading) can be used to limit sheet cavitation as it immediately changes to tip cavitation [5]. A high skew propeller also spreads the pressure fluctuation on the hull due to the blade passing which will limit radiated noise and vibrations. Rake on the other hand is the backwards slope of a profile in the xz-plane of reference (as defined in Figure 2.7)[8]. Since propeller blades are designed taking into account helical effects rake and skew are linked. Therefore the total rake per section can be computed using [8]:

$$i_t(r) = i_s(r) + i_g(r) \tag{2.2}$$

Where $i_t$ is the total rake, $i_s$ is the rake caused by the skew angle and $i_g$ is rake created by the geometry itself. The rake angle ($\theta_r$) can be computed using:

$$\theta_r = tan^{-1}\Big(\frac{i_g(r=R)}{R}\Big) \tag{2.3}$$

The rake angle is the formulation of the angle with respect to the flow. If the blade orientation is exactly vertical the rake angle will be zero and if it tends forward or backwards it will get positive or negative respectively [25].

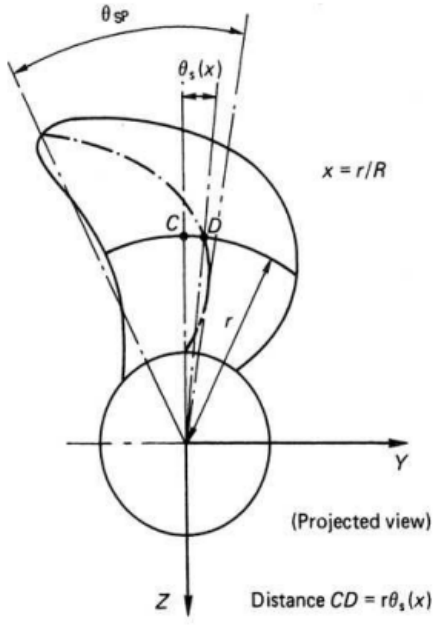



**Figure 2.6:** The skew and propeller skew angle [8]

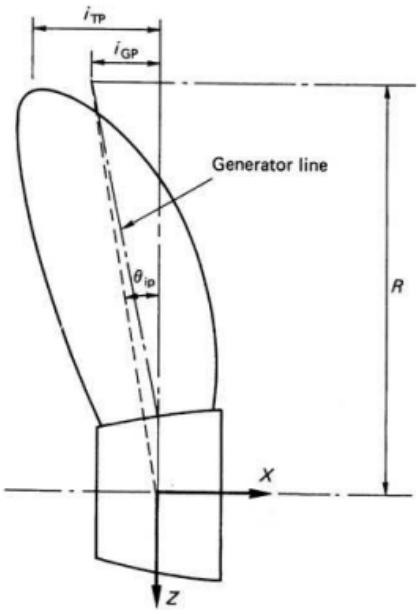



**Figure 2.7:** The definition of rake [8]

## 2.2. Propeller performance

Since the main goal of a propeller is to add sufficient momentum to a flow in the most efficient way. This will be explained using the Momentum Theory where the open water efficiency was introduced as:

$$\eta_0 = \frac{1}{2\pi}\frac{Tv_a}{Qn_p} \tag{2.4}$$

Open water efficiency is a widely accepted way to define propeller efficiency. To obtain the best propeller a part of the selection process is based on the matching of a propeller based on the open water diagram. This can be done using three non-dimensional parameters:

- The advance ratio ($J$)
- The thrust coefficient ($K_T$)
- The torque coefficient ($K_Q$)

All three are accompanied with their own formulation:

$$J = \frac{v_a}{n_p D} \tag{2.5}$$

$$K_T = \frac{T}{\rho n_p^2 D^4} \tag{2.6}$$

$$K_q = \frac{Q}{\rho n_p^2 D^5} \tag{2.7}$$

Here, $v_a$ is the advance velocity (the local velocity at the propeller given by $v_a = v(1-w)$), $n_p$ is the propeller rps and the propeller diameter $D$. $T$ represents the thrust and $Q$ the torque. This results in an open water efficiency formulation that depends on $K_T$ $K_Q$ and $J$:

$$\eta_0 = \frac{1}{2\pi}\frac{K_T J}{K_Q} \tag{2.8}$$

All lines can be plotted in an open water diagram as can be seen in Figure 2.8

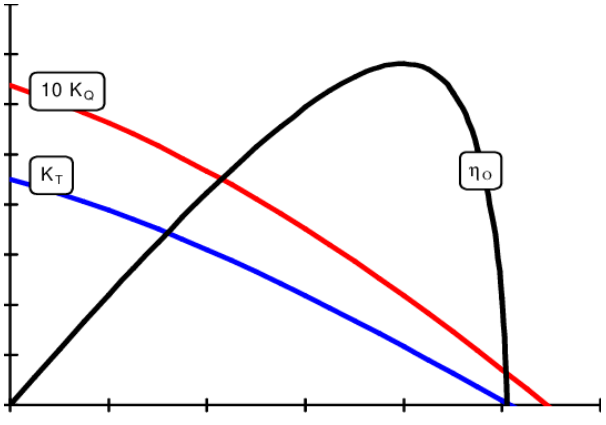



**Figure 2.8:** A typical open water diagram of the Wageningen B5-75 P/D=0.96 [29]

The goal of this research is to investigate the possibilities of regeneration, an open water diagram with negative thrust and torque would be interesting. This so called trailing condition can be included in such diagram. This is represented in a four-quadrant diagram. Which are the $K_T$ and $K_Q$ graphs at different loading conditions that can be seen in Table 2.1

**Table 2.1:** Ship speed and propeller speed at all four quadrants based on $\beta$ [31]

| Quadrant | Ship speed ($v_s$) | Propeller speed ($n_p$) | |
|---|---|---|---|
| $1^{st}$ 0-90 | + | + | Normal forward operation |
| $2^{nd}$ 90-180 | + | - | Reversing propeller during stopping manoeuvre |
| $3^{th}$ 180-270 | - | - | Normal astern operation and turbine mode |
| $4^{th}$ 270-360 | - | + | Forward thrust at negative speed (manoeuvring) |

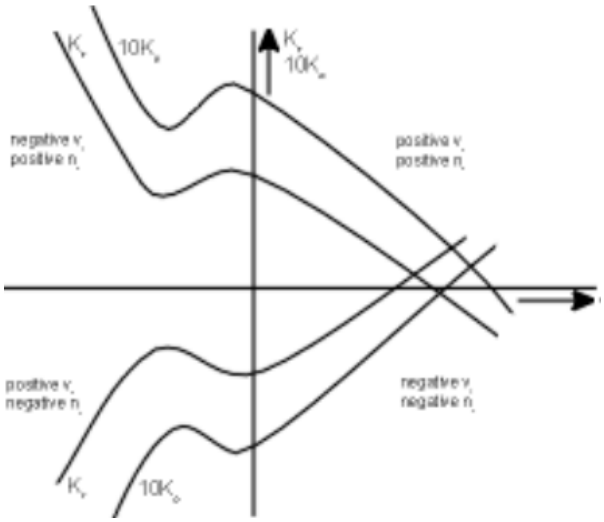



**Figure 2.9:** The open water diagram in four quadrants [29]

Notice that the $K_T$ and $K_Q$ lines in the first quadrant are for normal conditions, in fact these $K_T$ and $K_Q$ can also be seen in Figure 2.8. The open water diagram for four quadrants shows that the $K_T$ and $K_Q$ curves also provide coefficients for all different operating conditions. However, these are useful for the performance in off-design conditions and therefore in this research. A downside of the definition in Figure 2.8 is the odd definition of the quadrants (quadrant four for example provides a normal astern operation [29] while Table 2.1 states that this is in the third quadrant).

To provide a full description of propeller performance over all inflow angles MARIN has put great effort in creating a method to measure all four-quadrants. To do so, $C_T^*$ and $C_Q^*$ have been defined. These alternative thrust and torque coefficient can be plotted versus the angle of incidence at $70\%$ of the radius ($\beta$) [29] [31].

$$\beta = arctan(\frac{v_a}{0.7\pi n_p D}) \tag{2.9}$$

$$C_T^* = \frac{T}{\frac{1}{2}\rho(v_A^2 + (0.7\pi n_p D)^2)\frac{\pi}{4}D^2} \tag{2.10}$$

$$C_Q^* = \frac{Q}{\frac{1}{2}\rho(v_A^2 + (0.7\pi n_p D)^2)\frac{\pi}{4}D^3} \tag{2.11}$$

Note that the definitions of $C_T$ (Equation 4.8) and $C_T^*$ (Equation 2.10) are almost similar. The only difference in definition is the advance velocity squared, for $C_T^*$ $v_A^2 + (0.7\pi n_p D)^2$ is considered which of course is the squared inflow velocity at a propeller blade at $0.7D$. Since it is now known what $\beta$, $C_T^*$ and $C_Q^*$ are, the four quadrant diagram is useful, an example can be seen in Figure 2.10.

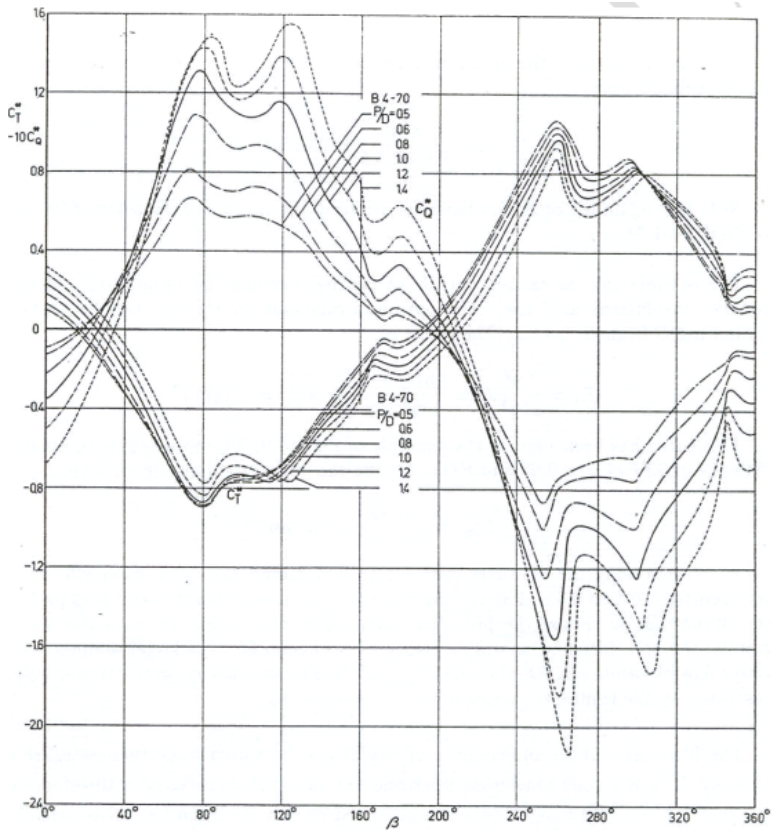



**Figure 2.10:** Four quadrants measurements of a B4-70 [29]

### 2.2.1. Turbine efficiency

Turbine efficiency tries to maximise the harvested power from a flow. A common used definition to define efficiency is based on the formulation of turbine power [57]:

$$P = \frac{1}{2}\rho v^3 A C_p \tag{2.12}$$

Here, $C_p$ is the power coefficient, which is the fraction of the power that is actually absorbed by the turbine and the total available power in the flow. Therefore the power coefficient is a direct representation of the turbine efficiency [57]. According to Betz this value could have a maximum of $C_p = 0.593$ [33]. In practice however turbines sometimes reach maximum values of $0.45$ while coefficients of $0.3 - 0.35$ should suffice for a well designed machine [14]. For hydrokinetic turbines $C_p$ values higher than the Betz limit for single turbine operations is not to be expected [51]. Most research regarding hydrokinetic turbines is based on tidal turbines [41]. For fixed turbines drag is only of importance for the foundation. However, for a turbine on a moving foundation, for instance a sailing vessel, drag is important. The importance is based on the connection between power, drag and velocity. If we recall the definition for drag [46]:

$$T = \frac{1}{2}\rho A v^2 C_T \tag{2.13}$$

If the sailing vessel increases its speed the harvested power will increase as well as the drag that is caused by the turbine. If both are considered the coupling will finally converge to a steady speed. The downside of this process is that it is very expensive to evaluate this for every single propeller. To simplify the assumption will be made that the propeller drag is considered to be constant while the power coefficient $C_p$ is maximised. chapter 4 and chapter 5 will provide more insight in the physics regarding turbines and turbine operation.

## 2.3. Negative Angle-of-Attack (first quadrant)

When considering a trailing propeller, it turns into a turbine with an inefficient blade design. For regeneration however, geometry such as camber might become an issue since the flow can be simulated as a foil loaded under a negative Angle-of-Attack. Considering the explanation of lift that has been provided in the literature review it is possible to understand the design criteria for a foil under negative Angles-of-Attack. If one considers camber as the only design criterion (over a profile with fixed thickness) one can to see that the correlation of the lift coefficient $\frac{C_{l-}}{C_{l+}}$ (at negative Angles-of-Attack $C_{l-}$ and positive Angles-of-Attack $C_{l+}$) decreases as camber increases. This can also be seen in Figure 2.11 [22] where lift curves for foils with negative, zero and positive camber are provided. The change in positive and negative camber changes the zero-lift line and since the nose-tail line does not change the Angle-of-Attack where the lift coefficient is at its maximum will change. A normal CPP that operates as a turbine is comparable to the foil with a negative camber. On the other hand $\frac{C_{l-}}{C_{l+}}$ will increase as the thickness increases (and the camber stays fixed) since it will result in a smaller $\frac{dp}{dn}$ over the leading edge on both sides [3]. This allows for a larger range of Angles-of-Attack, but will limit the maximum lift [28].

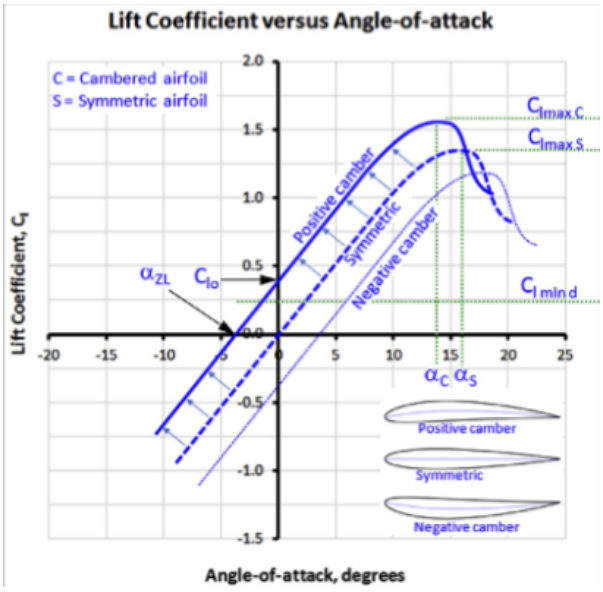



**Figure 2.11:** The effect of camber on the lift coefficient [22]

When observing the lift coefficient vs Angle-of-Attack graphs this can also be observed. Generally speaking the graph is symmetrical for symmetrical foils but as soon as camber is added the lift coefficient will tend towards positive Angles-of-Attack. For a propeller that should both, operate in propulsive- as well as in turbine mode this can pose a problem and should be taken into consideration.

### 2.3.1. Conclusions based on literature

Since the camber will be negative for at least one computation certain compromises have to be made to obtain a propeller that has sufficient propulsive efficiency as well as a high enough turbine efficiency. This is mainly based on the difference in blade design and the lack of ability to adapt the pitch angle and orientation of the blade. Either the turbine or the propeller has to serve at a 'negative Angles-of-Attack'. This implies that either, one of the two conditions will have a very limited lift coefficient, or the blade may only contain very limited camber. Taking into consideration the rotational speed when trailing as well as when in propulsive mode. the rotational speed when trailing is a result of the inflow velocity and harvested power whereas it is opposite for the propulsive mode where the ships velocity can be seen as a result of a certain rotational speed. This also exposes the challenge as it is seemingly impossible to design one propeller with a fixed pitch that can be used for propulsive operations as well as for turbine operations at various speeds.

## 2.4. Reversely loaded foils

Reversely loaded foils operate similar to forward loaded foils, however there are differences regarding maximum lift as well as for the leading and trailing edges. Marchand [38] clearly shows the effects of discontinuity of flow around the Leading edge of a reversed foil. Due to a sharp corner the flow shows discontinuity and laminar separation in the leading edge region on very small angles of attack. This behaviour can be observed in Figure 2.12 where the separation bubble is visible right at the Leading edge. Although this layer reattaches downstream the rest of the boundary layer remains turbulent [11]. This phenomena has one major implication, the $C_l$ changes almost instantly when the Angle-of-Attack is altered. This can be seen in Figure 2.13. As well it can be observed that loss of lift occurs at a lower angle than a forward loaded foil. This is due to the sharp angle (especially at high angles of attack) at the leading edge.

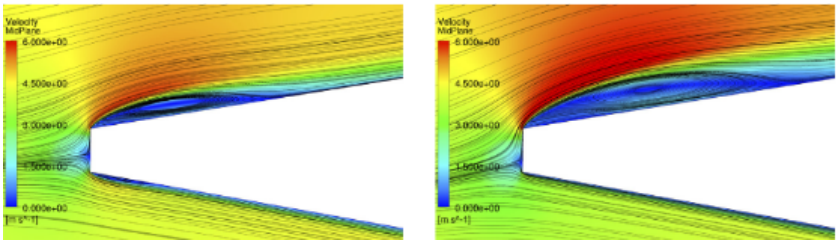


**Figure 2.12:** Separation at the leading edge for $\alpha = 0.5°$ (left) and $\alpha = 3°$ (right) [38]

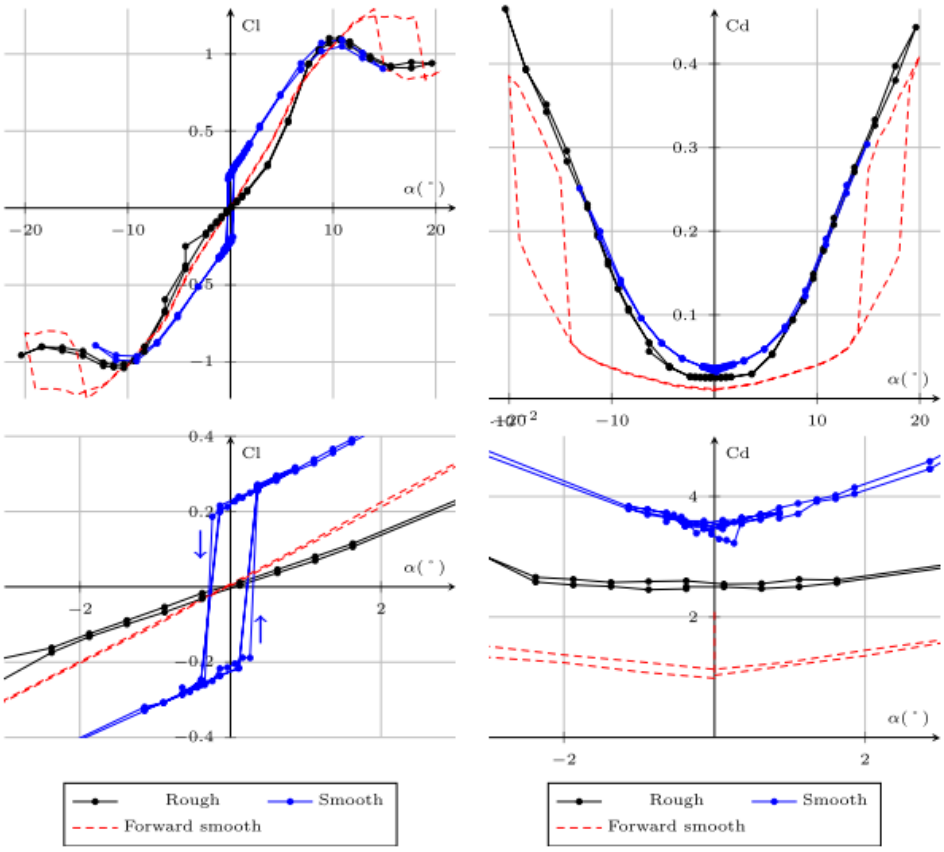



**Figure 2.13:** The influence of Angle-of-Attack on the $C_l$ (left) and $C_d$ (right) at a reversely loaded foil [38]

Another, still untouched issue is the geometry of the trailing edge. If the foil is reversed the trailing edge is heavily rounded, this will result in significant separation and turbulence. Both the NACA-0012 and SC-1095 profiles have been tested under angles of attack up to 180 degrees and shedding occurred

even at angles close to $180°$ This results in a lower stall angles than flow entering from the conventional leading edge [47].

## 2.5. Forward loaded CPP as a turbine (fourth quadrant)

To operate in the fourth quadrant a propeller has to be able to rotate over its feathering mode. The feathering mode of a propeller blade is when it is oriented such that it has a minimum drag. This causes the propeller blade to rotate past its zero lift condition and therefore it will start to generate negative lift. When considering one cross section (as depicted in the middle of Figure 2.1) this appears to be perfectly fine and it could be considered as the best condition. However, when considering the pitch distribution of an entire propeller blade this configuration starts to pose problems. Figures 2.14 and 2.15 display the propeller at positive and negative pitch angles, the red arrow is the inflow velocity and green is the rotational velocity.

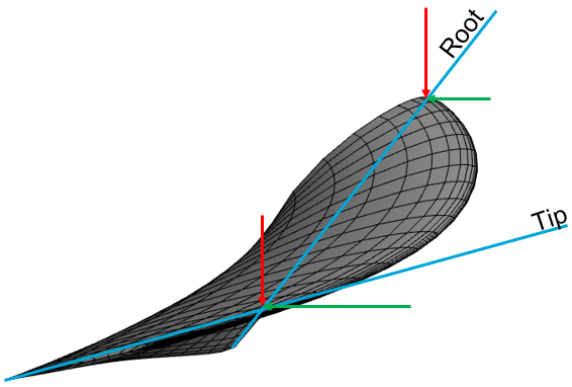



**Figure 2.14:** A top view of a propellerblade in normal condition (positive pitch)

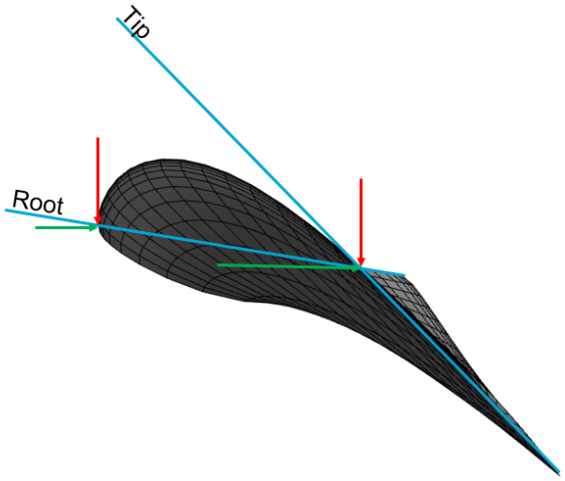



**Figure 2.15:** A top view of a propellerblade in turbine condition (negative pitch)

Figure 2.14 and Figure 2.15 show that the orientation of the blade changes if it is rotated through the feathering position. The pitch distribution is the wrong way around. This causes problems since the tangential velocity is increasing over the span while the pitch angle is increasing instead of decreasing (Figure 2.15 shows that the pitch angle at the trailing end is larger when compared to the root). This will cause tip overloading, which is something one wants to avoid when designing a propeller.

## 2.6. Configurations

To summarise, there are four possible configurations that can be investigated. These are a separate propeller and retractable turbine, a normal controllable pitch propeller that can rotate up to feathering mode, a CPP that can be reversed and lastly a a CPP with a negative pitch. After evaluating some sanity checks for case two (negative Angle-of-Attack) it was found that this propeller can just manage to satisfy the separation constraints. Therefore it is to be expected that the last configuration (CPP with negative pitch) is infeasible. This results in three configurations that will be considered for the remainder of the research. Furthermore, all designs are considered to be non-podded since that would significantly increase the complexity of the problem. The reversing of the propeller in the third quadrant can be achieved by using a fully rotating CPP (similar as done by the ship motion group [21]) or by using a rotating thruster. The final configurations are:

1. A separate propeller and retractable turbine, both optimised for their own task. This method is the most basic setup where a propeller is optimised for a specific design condition and a turbine will be optimised for real life sailing conditions.
2. A CPP that operates up to feathering mode but is optimised for both, propulsive, as well as for regenerative purposes. The propeller is loaded under a negative angle of attack and is therefore in the first quadrant of the four quadrant diagram Figure 2.10.

3. A CPP that is loaded under a reversed flow is investigated (third quadrant). If the flow is reversed the camber is orientated in the right direction making lift easier. The main challenges are regarding the design of the leading and trailing edge since they both should be able to fullfill both applications.

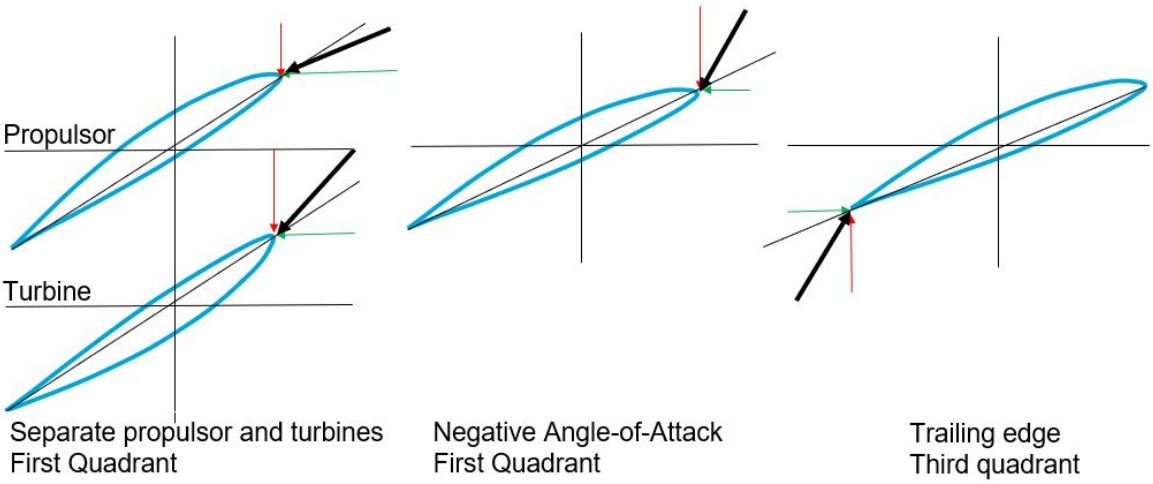



**Figure 2.16:** The three configurations that will be investigated further

# 3

# BEM - PROCAL

Propeller optimisation has modernised along with the increase in computational power. Propeller design can either be based on existing examples, but usually propellers are designed for specific operating conditions. This can be done by Simulation-based optimisation propeller design. Recent developments have been made in automated multi-objective optimisation techniques [7, 17, 15]. When optimising propeller design one can either use RANS methods or a BEM. Within MARIN PROCAL has been developed as a BEM based software to calculate propeller designs based on potential flow theory [40].

## 3.1. Potential flow

The theoretical basis of BEM can be found in the Navier-Stokes equations. This set of equations simplify due to assumptions that are made when using a BEM method, the flow is assumed to be irrotational and water is assumed to be inviscid. Therefore the disturbance velocity may be written as the gradient of the disturbance of the potential $\phi(\mathbf{x}, t)$. Which is space and time dependent [50]:

$$\mathbf{v}(\mathbf{x}, t) = \nabla\phi(\mathbf{x}, t) \tag{3.1}$$

Due to incompressibility the continuity equation simplifies to:

$$\nabla \cdot \mathbf{v}(\mathbf{x}, t) = \mathbf{0} \tag{3.2}$$

Which can be written as:

$$\nabla \cdot \nabla\phi(\mathbf{x}, t) \equiv 0 \tag{3.3}$$

This results in the Laplace equation:

$$\nabla^2\phi(\mathbf{x}, t) = 0 \tag{3.4}$$

Using Equation 3.1 it is possible to define the velocity at any point of the flow domain:

$$\mathbf{v}(\mathbf{x}, t) = \mathbf{v_0}(x, t) + \nabla\phi(\mathbf{x}, t) \tag{3.5}$$

Where $\mathbf{V_0} = \mathbf{V_w} - \Omega \times x$, $V_w$ is the undisturbed velocity and $\Omega \times x$ is the transport velocity. Taking into account the incompressibility and irrotational conditions for the flow the Navier-Stokes equation can be simplified to (which is equal to the unsteady Bernoulli equation): [50]

$$\frac{\partial\phi}{\partial t} + \frac{p}{\rho} + \frac{|\mathbf{v}|^2}{2} + gz = \frac{p_{ref}}{\rho} + \frac{|\mathbf{v_0}|^2}{2} \tag{3.6}$$

where $p_{ref}$ is the reference pressure far upstream. Now, reconsider the formulation for the pressure coefficient and the cavitation number:

$$C_p = \frac{p - p_0}{\frac{1}{2}\rho v_0^2} \tag{2.1}$$

$$\sigma = \frac{p - p_0}{\frac{1}{2}\rho v_0^2} \tag{3.7}$$

Using Equation 2.1 and Equation 3.7 the Navier stokes equation can be rewritten as [50]:

$$\frac{2}{v_{ref}^2}\frac{\partial \phi}{\partial t} + \frac{|\mathbf{v}|^2 - |\mathbf{v_0}|^2}{v_{ref}^2} + \frac{2gz}{v_{ref}^2} = -C_p \tag{3.8}$$

### 3.1.1. Lift and Potential flow

When considering potential flow theory one can model objects by means of sources and sinks. A source is a point that produces mass that follows a straight streamline emanating from the source origin. A sink is the opposite of a source, the streamlines are straight but moving directly towards the sink centre. By combining sinks and sources with uniform flow it is possible to model the flow around objects as can be seen in Figure 3.1 [2, 32, 30].

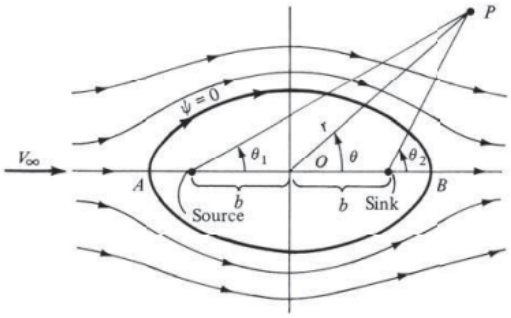



**Figure 3.1:** A rankine oval, described by a source-sink pair [2]

It can be seen that there is a distinct boundary where the tangential velocity is 0 ($\frac{\partial \phi}{\partial n} = 0$), this is the described surface. There are cases where the description of a source and sink is insufficient. Therefore it is possible to introduce a dipole (or doublet), which are a source and a sink that are infinitely close together, therefore they produce a flowfield while not producing or extracting fluid. This enables velocities to be modelled tangential to the origin (as can be seen in Figure 3.2) [2, 32].

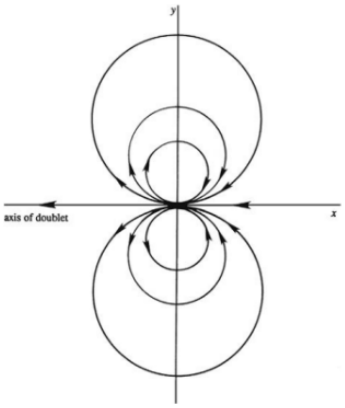



**Figure 3.2:** The flow field of an ideal two dimensional dipole [32]

## 3.2. Boundary Element Method

Although it is possible to model 2-dimensional geometries while using potential flow as in Figure 3.1. The construction of more complex structures is not straightforward. Therefore BEM is introduced. BEM is a method where a solid structure is modelled as a closed sheet existing of a finite number of elements. Every element is a source (or sink) and can contain a dipole. To ensure a closed sheet the influence of the total produced fluid (of all elements) at the center of element $m$ should satisfy the Neumann condition ($\frac{\partial \phi}{\partial n} = \mathbf{V_0 n}$). [30, 50]. The total sum of the influence of all sources and dipoles ($n_1 ... n_N$) at location $m$ can therefore be written and solved as a surface integral, this is called the Green's identity [30].

## 3.3. Challenges using BEM

Since BEM uses an inviscid and irrotational flow some physical effects that occur in flow can not be modelled. This poses challenges that often can be solved using approximations such as empirical models. The main limitations will be described below.

### 3.3.1. Trailing edge

Due to irrotational effects the flow will tend to move around the trailing edge if nothing is altered. This results in problems at the trailing edge causing a large pressure gradient. Due to the pressure gradient the local velocity can approach infinitely large numbers. As one can imagine, that is impossible in real life due to separation of flow. To solve this an amount of circulation ($\Gamma$) is added to the flow. This circulation is represented as a vortex sheet that causes the flow to accelerate on one side of the foil while the flow on the other side slows down. For foils where the flow must separate smoothly in normal conditions the Kutta-condition should be satisfied. This condition states that the pressure differences between the pressure and suction side of the trailing edge should be zero (to therefore ensure smooth separation of flow) [26] (see Figure 3.3 and Figure 3.4). The added circulation also provides a definition of the available lift by $F_L = \rho V \Gamma$.

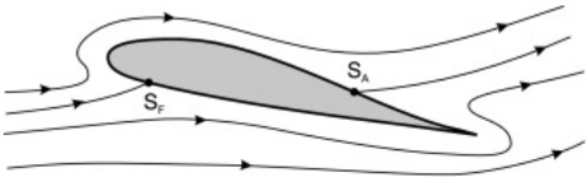



**Figure 3.3:** A foil in inviscid flow without circulation [26]

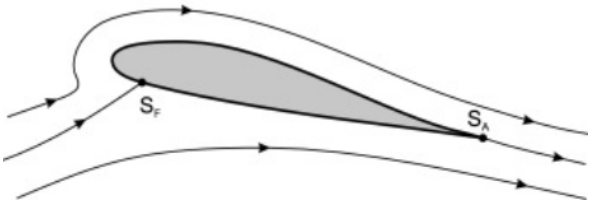



**Figure 3.4:** A foil in inviscid flow with circulation [26]

### 3.3.2. Wake modelling

The Kutta-condition introduces circulation around the foil. This circulation is needed to generate lift but as a consequence the wake will contain shed vorticity [4]. To be able to compute a propeller with an accurate thrust and drag while using BEM the vorticity should be maintained until the far field. This is done by considering a material wake. The modelling of this wake is a complex process and chapter 5 shows that choosing a different model can have significant influence on the propeller performance. Because the wake modelling has to be done for every timestep this will slow down BEM computations considerably, chapter 5 will explain more about this process.

## 3.4. PROCAL

Since PROCAL is based on potential theory, it is bound to its physical limitations. To solve the issues that have been described in the previous section PROCAL has been equiped with multiple modules. This section will elaborate more on the tools that have been implemented to compensate/ mitigate the limitations of a BEM approach to propeller design.

### 3.4.1. Trailing edge

To ensure the Kutta-condition at the trailing edge for a three dimensional computation it should be ensured that:

$$\Delta p = p^+ - p^- = 0 \tag{3.9}$$

Where $p^+$ is the pressure side of the blade and $p^-$ is the suction side of the blade at the trailing edge. To satisfy this, the non-linear equation below should be satisfied at the trailing edge.

$$\Delta p(\Delta \phi_{TE}) = 0 \tag{3.10}$$

This is done by applying the so-called Iterative Pressure Kutta-Condition (IPKC). To prevent a pressure jump the dipole strength in the wake is varied such that $\Delta p = 0$ at the trailing edge. The vorticity in the material wake is carried by the dipole strength in the wake elements. Since the relation between dipole strength and pressure is quadratic non-linear equations are needed to solve this. This is normally done by the Newton-Raphson iteration process. To do so the Jacobian matrix, which is a matrix containing all

partial derivatives of the pressure differences with respect to the dipole strength of every wake element is needed. Solving this matrix is needed to update the dipole strength and therefore solve the pressure difference. Iterating this problem is a time consuming, but accurate for satisfying the Kutta condition [52].

A similar problem as in Figure 3.3 can occur when the flow satisfies the Kutta-condition. If the trailing edge is rounded this will result in a sharp curvature (as can be seen in Figure 3.5). Similarly as with an unsatisfied Kutta-condition this will result in very strong pressure gradients and therefore, strong fluctuations in the velocity. To ensure a proper separation PropArt flattens the trailing edge, therefore altering the blade geometry. This poses an increase in uncertainty, especially when computing a propeller that is reversed and therefore has two flattened edges.

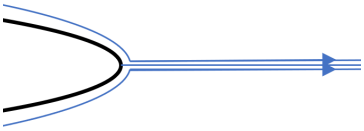


**Figure 3.5:** A rounded trailing edge, with its streamlines if nothing is altered

### 3.4.2. wake modelling

To ensure a consistent wake when using BEM, a material wake is modelled. This wake carries all vorticity that is shed by the propeller (to which it is attached). This can be seen in Figure 3.6. To achieve this, a kinematic and dynamic boundary condition need to be satisfied. To satisfy the dynamic boundary condition pressure differences between the two wake surfaces are not allowed, in other words Equation 3.9 has to be met. The kinematic boundary condition requires that the vortex wake is a stream-surface of the flow [50]. To satisfy the continuity equation (and therefore the actuator disk theory that will be discussed in section 4.1) the wake should contract for propulsive operation, as in Figure 3.7. This is achieved by implementing a contraction model, within PROCAL there are multiple options that vary from a self iterative wake to a fully prescribed wake [5]. Since PROCAL was originally written for propulsive operations the wake contraction models might have to be altered to manage wake expansion for turbine operations.

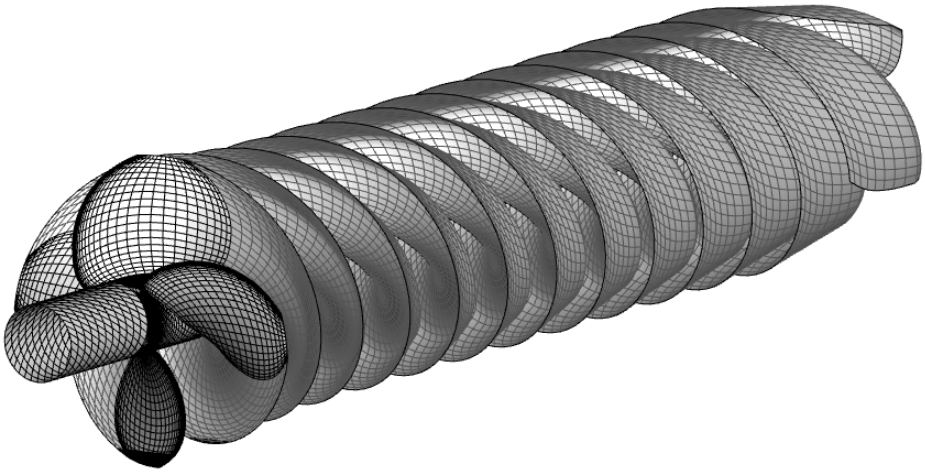


**Figure 3.6:** The propeller wake of a test optimisation

The last issue regarding the wake is the lack of sufficient rotation in the flow to roll and form a tip vortex. To compensate this an Empirical Tip Vortex model (ETV model) has been implemented as a post processor. [50] Although the use of empirical methods on BEM is a way of stretching the limitations of BEM it is still considered as a sufficient accurate method to model tip-vortices.

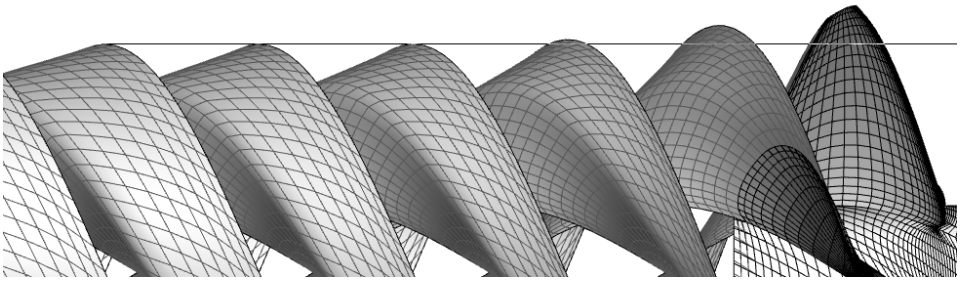


**Figure 3.7:** Wake contraction in PROCAL

### 3.4.3. Cavitation

Cavitation is very often accepted at the benefit of propeller efficiency or it may be unavoidable due to the large thrust density required. However, certain types of cavitation or excessive cavitation may lead to harmful effects such as propeller thrust breakdown and erosion and it can become a major source of noise and vibration causing discomfort to crew and passengers [50]. A problem that has not been covered in the first section is the modelling of cavitation within PROCAL. PROCAL is able to compute the local pressure distribution over the blade. The local pressures can be used for a model that implements cavitation based on the pressure coefficient and the cavitation number as can be seen in Equation 2.1 and Equation 3.7. If the pressure coefficient drops below the cavitation number cavitation will start to occur.

The pressure at the cavity surface is considered to be the vapour pressure $p_{vap}$. If cavitation occurs, PROCAL provides an output file that describes the extent of the cavity over one rotation. An example can be seen in Figure 3.8, where a cavitating propeller has been modelled to show the appearance of cavitation.

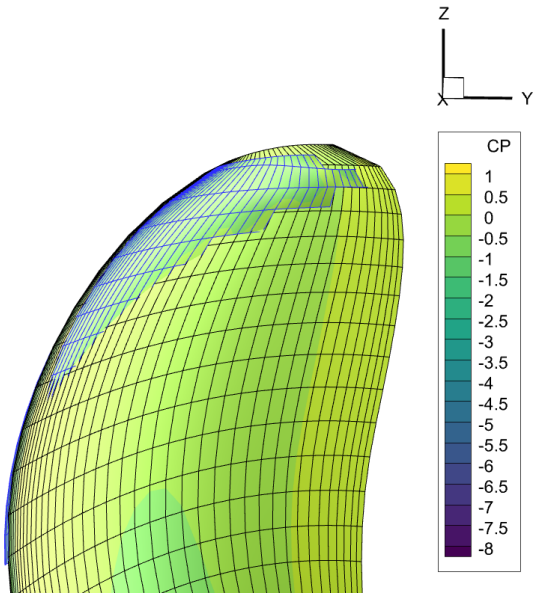



**Figure 3.8:** The appearance of cavitation in PROCAL

# 4

# Wake expansion

Since PROCAL was initially designed for propellers the wake models only support wake contraction. To be able to obtain a contracting wake two models, the Hoshino and the Min model were implemented [6] (an alternative is to compute the wake orientation iteratively, the wake contraction is still prescribed). Although these models are accurate in propulsion, they are not fit to model wake expansion. By altering two main settings the wake can be changed to an expanding wake. First, the contraction rate has to be defined manually, this can be done as a fraction of the original propeller radius. Secondly, the wake alignment can be changed by different settings. Literature proposes a few methods to estimate the wake expansion of a propeller in turbine mode. This chapter will cover two methods for wake expansion models that can be found in literature and one method that MARIN has used to compute the wake expansion of a propeller with high J values.

## 4.1. Momentum theory

When making basic calculations for propellers or turbines the momentum theory (or actuator disk theory) is a mathematical method that is often used to describe an ideal actuator disk. Using the momentum theory the propeller- or turbine blades have been replaced by an infinitely thin disk where a pressure jump over the upstream and downstream disk side induce either an increase or a decrease in flow velocity [14, 49]. The theory assumes an axisymmetric flow approach to balance in and outflow across the disk. This flow is assumed inviscid, irrotational and incompressible [49]. A graphical view of both modes according to the momentum theory can be seen in Figure 4.1 and Figure 4.2.

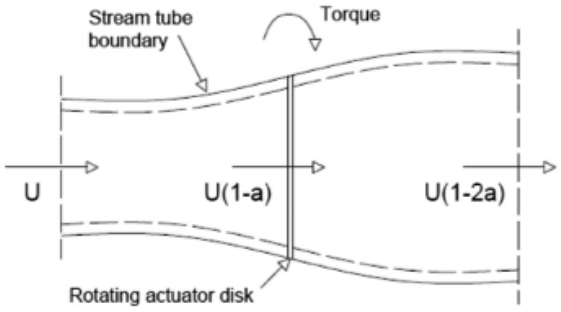



**Figure 4.1:** The momentum theory over a turbine disk [49]

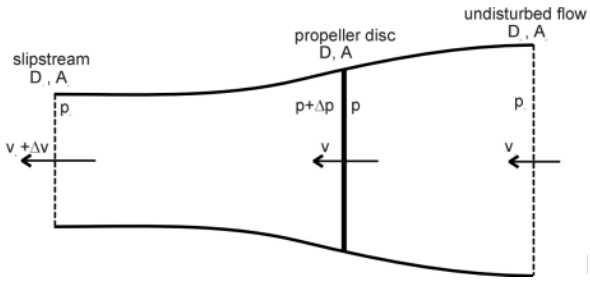



**Figure 4.2:** The momentum theory over a propeller disk [29]

Assuming the disk has a diameter $D$ and cross sectional area $A$ over this area a sudden pressure jump of $\Delta p$ is prescribed. The flow in both cases has an initial velocity that is changed over the pressure jump, as the pressure jump is positive the flow accelerates (propulsive mode) and as the pressure jump is negative the flow will decelerate (turbine mode). The thrust when in propulsive mode can be computed by:

$$T = A\Delta p \tag{4.1}$$

### 4.1.1. Expansion

Since the momentum theory is valid for both propeller and turbine operations it is possible to use this theory to determine a factor for the thrust coefficient in propeller operation that is valid in turbine operation as well. This factor can be used in the derivation of a wake expansion factor in turbine mode. Since the axial force coefficient is a proper way to describe the disk loading of a turbine the thrust loading coefficient can be introduced for a propeller. To do so a definition for the delivered thrust is needed. Taking into account the conservation of mass or the continuity equation and by applying this at the undisturbed flow [14]:

$$\frac{\dot{m}}{\rho} = v_A A_0 = vA = (v_A + \Delta v)A_1 \tag{4.2}$$

or:

$$v_A D_0^2 = vD^2 = (v_A + \Delta v)D_1^2 \tag{4.3}$$

Resulting in:

$$D_0^2 = \frac{v}{v_A}D^2 \text{ and } D_1^2 = \frac{v}{(v + \Delta v)}D^2 \tag{4.4}$$

From the momentum theory it follows that the force exerted on the fluid equals the net outflow of momentum. And since momentum is the product of mass flow and speed this can be computed by:

$$T = \dot{m}_{out}(v_A + \Delta v) - \dot{m}_{in}v_A \tag{4.5}$$

By eliminating $A_1$ and $A_0$ and rewriting $\dot{m}_{in}$ and $\dot{m}_{out}$ a new formulation for thrust can be formulated:

$$T = \frac{\pi}{4}D^2\rho v\Delta v \tag{4.6}$$

combining Equation 4.1 and Equation 4.6 one can find that the velocity equals $v = (v_A + \frac{1}{2}\Delta v)$ so the formulation for thrust is:

$$T = \frac{\pi}{4}D^2\rho(v_A + \frac{1}{2}\Delta v)\Delta v \tag{4.7}$$

By definition the non-dimensional thrust loading coefficient is [29]:

$$C_T^{def} = \frac{T}{\frac{1}{2}\rho v_A^2 \frac{\pi}{4}D^2} \tag{4.8}$$

combining the last two equations results in :

$$C_T = \frac{(v_A + \frac{1}{2}\Delta v)\Delta v}{\frac{1}{2}v_A^2} = 2\frac{\Delta v}{v_A} + (\frac{\Delta v}{v_A})^2 \tag{4.9}$$

This can be rewritten to:

$$\frac{\Delta v}{v_A} = -1 \pm \sqrt{1 + C_T} \tag{4.10}$$

As can be seen in Figure 4.1 the change in velocity at and behind the disk can be expressed by using the formulation given by Equation 4.10. As well the known relations regarding conservation of mass can be used to formulate the rate of expansion of the turbine wake:

$$v_d A_d = v_w A_w \tag{4.11}$$

where:

$$v_d = v_0(1 - a) = v_0(1 - \Delta v) \qquad v_w = v_0(1 - 2a) = v_0(1 - 2\Delta v) \tag{4.12}$$

and $A_d = \pi R_d^2$ and $A_w = \pi R_w^2$ and we want to determine $R_w$. As can be seen the fraction $\frac{\Delta v}{v_A}$ of Equation 4.10 equals $a$ and therefore combining Equation 4.10,4.11 and 4.12 simplifies to:

$$a = -1 \pm \sqrt{1 + C_T} \tag{4.13}$$

$$R_w = R_d\sqrt{\frac{1 - a}{1 - 2a}} \tag{4.14}$$

Therewith it is possible to determine the expansion of the wake at the end of the near field domain.

### 4.1.2. Integration

To compute the wake expansion for a certain J value, PROCAL has to run one iteration with an estimated expansion. When done, the iteration can be expanded by recalculating the expansion using the new $C_T$ value. Almost always the $C_T$ value converged to 0.01 accuracy within 1 (and sometimes 2) iterations, therefore it is safe to say that this iterative method is promising and should be tested for different propellers as well.

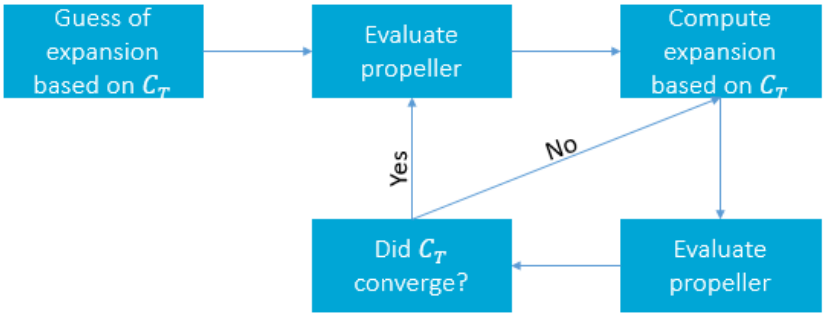



**Figure 4.3:** The iterational process of the wake expansion

## 4.2. Empirical method

Another method based on measurements is proposed by Salvatore, Sarichloo and Calcagni [46]. This method is mainly based on experimental data. The function that is suggested is:

$$R_w = R_d + r_0(1 - e^{-\tilde{\zeta}_x/C_2}) \tag{4.15}$$

Where $R_d$ is the propeller radius at the disk. $\tilde{\zeta}_x$ is a linear spacing from 0 to 1 from where 0 is the disk and 1 is the start of the far field. $r_0$ and $C_2$ are determined from test data and are proposed to be $0.35$ and $2.0$. The formulation was fitted against experimental data for a $700mm$ diameter tidal turbine in such a way that the wake expansion in the entire near field can be computed. Since it is currently not possible to model the shape of the wake expansion in PROCAL, only Salvatores expansion rate at the start of the far field is used. This was done because the rate of expansion that is defined is at the start of the far field. With the proposed values this yields an expansion factor of $0.39$ for a propeller with a diameter of $700mm$. This however is for one specific propeller in one specific operation. Since the method was fitted to this turbine the wake expansion factor of this propeller is used for testing later on.

## 4.3. Zero expansion

A third method is to neglect the wake expansion but admit that there is no contraction, in other words, set the expansion factor to 0. This method is assumed to be the least accurate but has been used within MARIN up to now. Therefore this method will be used as benchmark to see whether improvements can be made.

## 4.4. Testing rate of expansion

The three methods have been tested for a four bladed F-series propeller (F4-63-0.6) for increasing values of J since a wide range of J values has been experimentally tested in the basins at MARIN. The results can be seen in Figure 4.4,4.5 and 4.6.

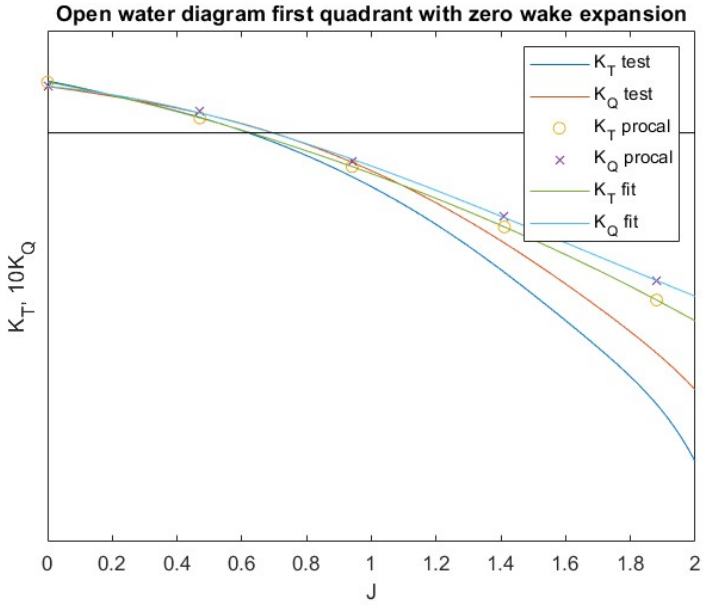



**Figure 4.4:** Open water diagram for the first quadrant and zero expansion and an iterative wake alignment

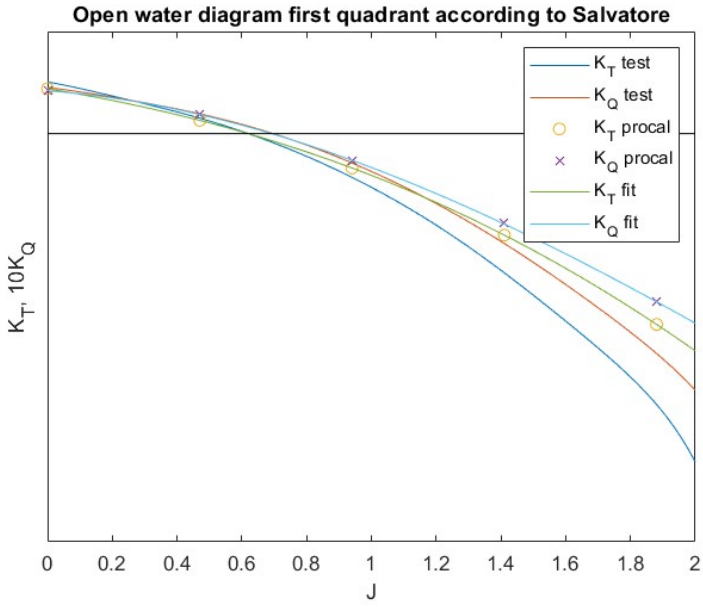



**Figure 4.5:** Open water diagram for the first quadrant with maximum expansion according to Salvatore and an iterative wake alignment

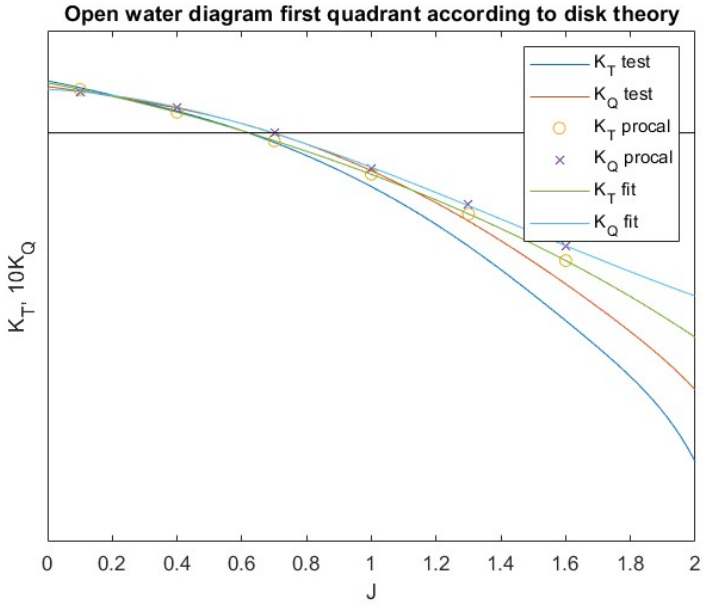



**Figure 4.6:** Open water diagram for the first quadrant with expansion according to the disk theory and an iterative wake alignment

The wake pitch alignment was chosen iteratively for these runs. As can be seen there are some slight deviations between Salvatore and the disk theory but these are only minor differences in $K_T$ and $K_Q$ values. The zero wake expansion however does differ noticeably for the higher J values. The similarity in results for Salvatore and the disk theory can be explained by the similarity in expansion factor for turbine operation. For Salvatore's empirical model an expansion factor of 0.39 was found while the range of the expansion factor for the momentum theory at negative $K_T$ and $K_Q$ values differs between $0.2$ and $0.47$ (with J between $1$ and $1.6$). Since Salvatores method was tuned for one turbine at one operating condition it is not very versatile. As well, Salvatores method focuses on the development of the propeller wake over the near field while this research investigates the wake expansion at the start of the far field. Therefore the disk theory is chosen for further computations. This was done since it only focuses on the wake expansion and can easily be used to compute the rate of wake expansion in different operating conditions.
The remainder of this research uses a manual iteration to determine the wake expansion factor. This will be done using the F4-63-0.6 propeller. Since it is shown that the iterative procedure does converge efficient it is advised to implement this method for a future version of PROCAL.

# 5

# Wake pitch alignment

There are multiple options to compute the wake alignment in turbine operations. Two methods will be considered, the iterative wake alignment option that was build into Procal can be used. Secondly a method that can be deducted from the Blade Element Momentum Theory is proposed.

## 5.1. Blade Element Momentum Theory

The operating principle of a horizontal axis turbine is classically based on the Blade Element Momentum theory (also known as the Strip theory or the Glauert/Wilson Method) [42, 48, 57]. The Momentum theory and the Blade Element theory are two separate theories that can be combined and are a widely accepted way to compute the performance of a turbine [49]. The only real alternative is to use a full computational fluid dynamics (CFD) approach using either RANS (Reynolds Averaged Navier Stokes) or BEM.
The Blade Element Theory, differing from the momentum theory, defines the forces on the blades as functions of lift and drag coefficients ($C_l$ and $C_d$ respectively) as well as the Angle-of-Attack. The blade is divided in small segments Figure 5.1 so the chord length, twist angles and blade geometry can be taken constant per segment. In this method frictional and viscous effects are neglected. This results in two forces that have to be considered, the lift and drag forces. [14, 48]

### Lift and drag

Using the sectional geometry the lift and drag coefficients can be estimated (this is often done by wind tunnel testing) which can be used to calculate the lift and drag:

$$dF_L = \frac{1}{2} C_l \rho v_{rel}^2 c_r dr \tag{5.1}$$

$$dF_d = \frac{1}{2} C_d \rho v_{rel}^2 c_r dr \tag{5.2}$$

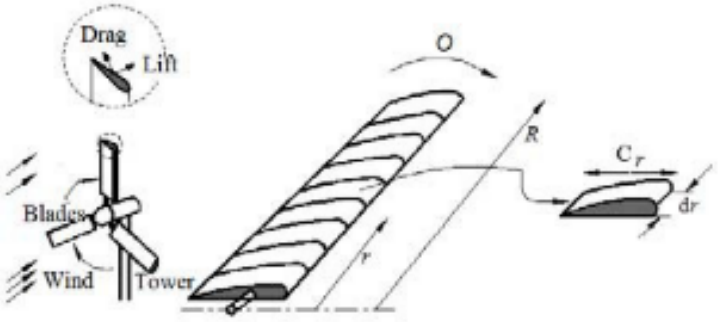



**Figure 5.1:** A representation of the Blade Element theory [48]

Here, $v_{rel}$ is the relative inflow velocity (including the rotational speed experienced at the blade), $c_r$ is the chord length and $dr$ is the segment width [48]. Using these equations elementary expressions for

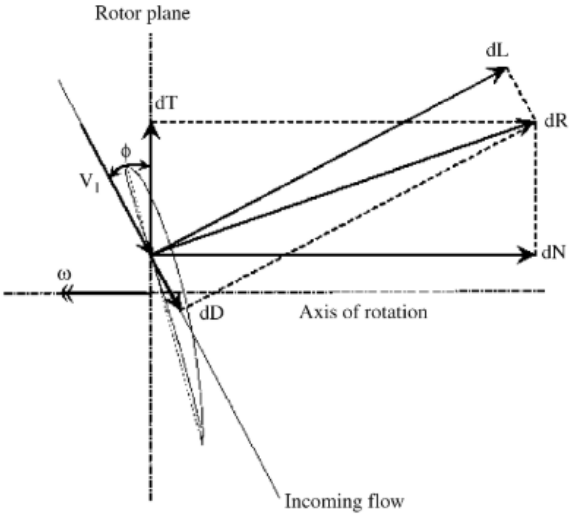



**Figure 5.2:** A graphical representation of the lift and drag forces on a blade segment [34]

lift and drag can be formulated:

$$dF_T = \frac{1}{2}\rho v_{rel}^2(C_l sin(\phi) - C_d cos(\phi))Zc_r dr \tag{5.3}$$

$$dF_N = \frac{1}{2}\rho v_{rel}^2(C_l cos(\phi) + C_d sin(\phi))Zc_r dr \tag{5.4}$$

Here, $\phi$ is the angle of $v_{rel}$ with respect to the rotor plane (as defined in Figure 5.2) and $Z$ is the number of blades present on the turbine. [14, 48].
Using these expressions finally the elementary power of the turbine can be computed by multiplying the elementary thrust with $\Omega r$:

$$dP = \frac{1}{2}\rho v_{rel}^2 \Omega r((C_l sin(\phi) - C_d cos(\phi))Zc_r dr \tag{5.5}$$

Where $\Omega$ is the rotational speed of the rotor [14]. The total Power can than be computed by the integral over the propeller span:

$$P = Z\int_0^r \rho v_{rel}^2 \Omega((C_l sin(\phi) - C_d cos(\phi))Zc_r r dr \tag{5.6}$$

### 5.1.1. Relative velocity

The geometry of turbines varies along the radial distance $r$. This is mainly due to the importance of the rotational speed. The inflow velocity that is experienced at the root of the blade will be roughly equal to the incident speed while the inflow velocity at the propeller tip will be dominated by the rotational velocity. The velocity can be written as:

$$v_{rel} = \sqrt{v^2(1-a)^2 + (\Omega r)^2(1+a')^2} \tag{5.7}$$

Where $a$ is the axial velocity induction factor and $a'$ is the tangential velocity induction factor where $a$ and $a'$ vary over the radius of the blade due to varying blade loading [14] and the $\Omega r$ term of course is linearly dependent on the distance from the root. These induction factors represent the rotational terms in the wake of the turbine. Since the wake behind the turbine does have rotary motion a formulation of the angular momentum can be formulated

$$d\tau = (d\dot{m})2\Omega r^2 = (2\pi r dr \rho v_{turb})2a'\Omega r^2 \tag{5.8}$$

Where $d\tau$ is the elemental torque of the wake based on the approach from the propeller and $\dot{m} = \rho v_{turb} A$ is the massflow through the streamtube in Figure 4.1 [14]. The elemental torque can also be defined from rotor perspective:

$$dT = \dot{L} = \frac{I\omega}{dt} = \dot{m} r^2 \omega a' dr \tag{5.9}$$

In this definition $dT$ and $\dot{L}$ are the angular momentum of fluid crossing the disk, $\omega$ is the rotational speed of the wake and $I$ is the moment of inertia. As well $a'$ is included, this is the induction factor that determines the part of the torque that is effectively transferred. [20] Since both formulations represent the same, they can be rewritten to define $a'$. Here the tangential induction factor is the fraction of the angular momentum present in the stream divided by the angular momentum of the turbine:

$$a' = \frac{\omega}{2\Omega} \tag{5.10}$$

The momentum theory is used to get a first estimate of $a$ and $a'$. Since both values are considered to be constant over the radius [14, 20]. However the Blade Element theory considers $a$ and $a'$ to be varying over the blade radius. Using the elementary torque formulation from the momentum theory:

$$dX = d\dot{m}(v_{in} - v_{out}) = d\dot{m} v_{turb} \frac{2a}{1-a} \tag{5.11}$$

and from the blade element theory we recall:

$$dF_N = dX = \frac{1}{2}\rho v_{rel}^2 (C_l cos(\phi) - C_d sin(\phi)) Z c_r dr \tag{5.4}$$

rewriting these equations provides us with:

$$\frac{a}{1-a} = \frac{Z c_r (C_L cos(\phi) + C_D sin(\phi))}{8\pi r sin^2(\phi)} \tag{5.12}$$

The same can be done for the tangential velocity induction factor. we recall the elemental torque from the momentum equation as [14, 42]:

$$d\tau = (2\pi r dr)\rho v_{tur} 2a'\Omega r^2 \tag{5.8}$$

By using Equation 5.3, Equation 5.8 can be rewritten to:

$$\frac{a'}{1+a'} = \frac{Z c_r (C_l sin(\phi) - C_d cos(\phi))}{8\pi r sin(\phi)\,\mathrm{cos}(\phi)} \tag{5.13}$$

These formulations still seem to be fairly complex, therefore three coefficients are introduced, the blade loading coefficient ($\Psi$), the drag over lift ratio ($\epsilon$) and the tip speed ratio ($\lambda$) [14]:

$$\Psi = \frac{Z c_r C_l}{8\pi r} \tag{5.14}$$

$$\epsilon = \frac{C_d}{C_l} \tag{5.15}$$

$$\lambda = \frac{\Omega R}{v_{rel}} \tag{5.16}$$

therefore Equation 5.12 and Equation 5.13 simplify to:

$$\frac{a}{1-a} = \frac{\Psi(cos(\phi) + \epsilon sin(\phi))}{sin^2(\phi)} \tag{5.17}$$

$$\frac{a}{1+a'} = \frac{\Psi(sin(\phi) - \epsilon cos(\phi))}{sin(\phi)cos(\phi)} \tag{5.18}$$

Lastly we can now define the relative flow angle [14]:

$$tan(\phi) = \frac{v_{in}}{r\Omega}\left(\frac{1-a}{1+a'}\right) \tag{5.19}$$

The relative flow angle is useful for determining the local angle of the flow. Therefore this formulation can be used in the pitch alignment.

## 5.2. Pitch alignment

To determine the wake pitch alignment based on the Blade Element Momentum theory Equation 5.19 is used. In this formulation the computation of $a$ can been explained in Equation 4.13, the estimation of $a'$ however is more difficult.

To model the turbine wake in normal conditions (no separation and stall occur) the Hansen model can be used. This model uses an implicit model to estimate the tangential induction factor. It is derived from the classic Blade Element Momentum theory and ignores drag. Although there is no optimal value given for $a$ there is a relationship between $a$ and $a'$. This relation can be given by optimising the harvested power as defined by the Blade Element Momentum Theory (this definition is based on the formulation of elemental torque, Equation 5.8). [24]

$$P = 4\pi\rho\Omega^2 V_0 \int_0^R a'(1-a)r^3 dr \tag{5.20}$$

Maximising the harvested power can be done by maximising the non-dimensional components: $f(a', a) = a'(1-a)$. According to potential flow theory the force is perpendicular to the local velocity and thus the induced local velocity must be parallel to the force. Therefore the following relationship exists [24]:

$$(\frac{\Omega r}{V_0})^2 a'(1+a') = a(1-a) \tag{5.21}$$

By maximising $f(a', a)$ while still satisfying Equation 5.21 the relation between $a$ and $a'$ is [24]:

$$a' = \frac{1-3a}{4a-1} \tag{5.22}$$

Using the Hansen model, the relation between $a$ and $a'$ can be computed and the actual flow angle (pitch) of the wake can be estimated. This has been done for the test case, where an advance ratio of $1.2$ is the design point when in regeneration. For the F4-63-0.6 propeller this results in a $C_T$ value of $0.701$, using Equation 4.13 a value for $a$ can be computed. This results in $a = 0.304$. Finally using Equation 5.22 $a'$ can be obtained: $0.402$. Therefore the alignment angle of the wake can be given by:

$$tan(\phi) = \frac{v_{in}}{r\Omega}\left(\frac{1-0.34}{1+0.402}\right) = 0.497\frac{v_{in}}{r\Omega} \tag{5.23}$$

This is very similar to the advance ratio. Therefore some tests have been computed using a wake pitch alignment that is similar to the advance ratio. This resulted in the two open water diagrams that can be seen in:

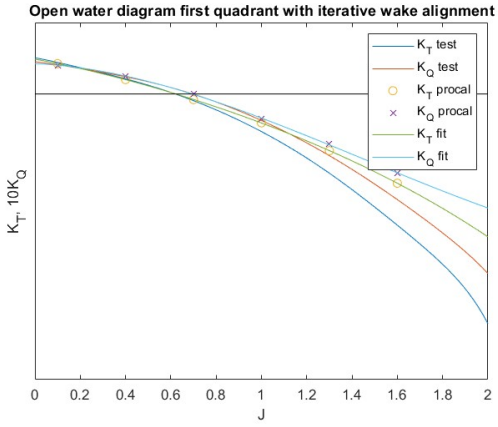



**Figure 5.3:** Open water diagram for the first quadrant with an iterative wake pitch alignment

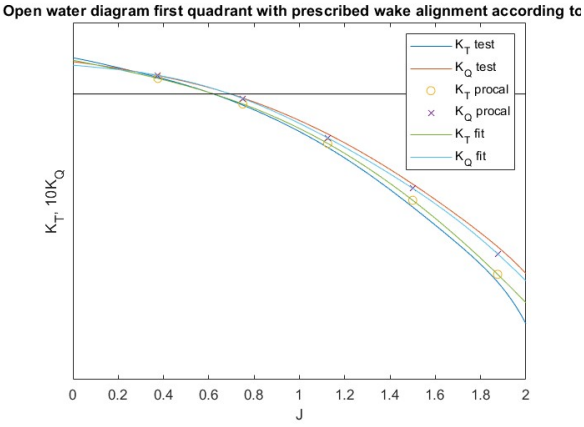



**Figure 5.4:** Open water diagram for the first quadrant with a wake pitch alignment according to J

When comparing these plots, it can be seen that the wake alignment according to the advance ratio performs best, especially for high J values. A visualisation of the wake shows a clear difference in the wake geometry. When taking a look at Figure 5.5 and Figure 5.6 it becomes clear that the iterative pitch alignment models a gradient of the velocity with respect to the radius, this is similar as in Figure 3.6

but now for a turbine instead of for a propulsor. The main difference is that the part of the wake that is dominated by the horizontal velocity is dominant over the wake that aligns the rotational effects. This might explain why the prescribed wake yields more accurate results than the iterative wake. This however also raises the quesion if the prescribed wake according to the advance ratio can be useful for all computations regarding high J values or that alternations should be made to implement the Hansen model.

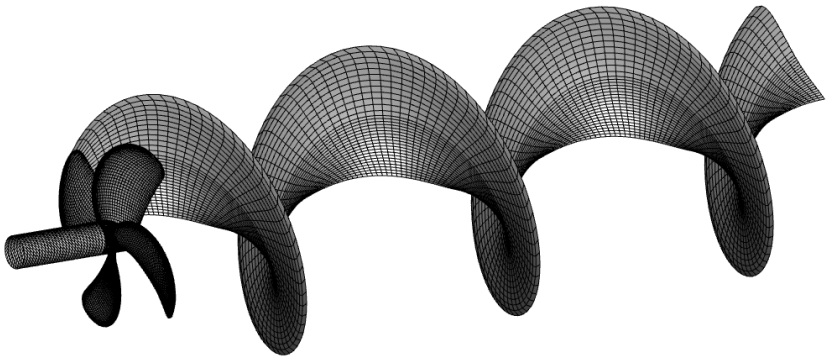


**Figure 5.5:** The prescribed wake alignment according to the advance ratio for a propeller in turbine operation

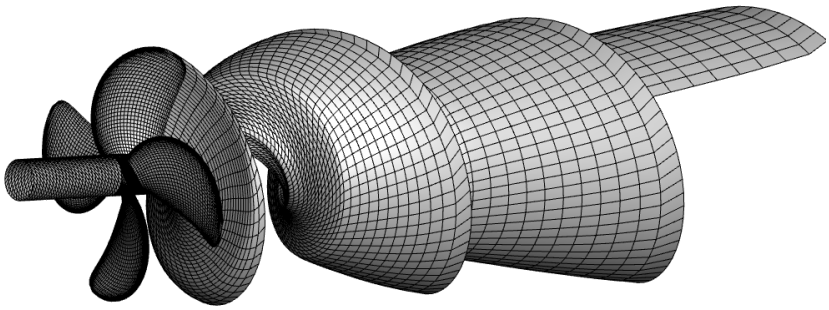


**Figure 5.6:** The iterative wake pitch alignment for a propeller in turbine operation

Since the wake alignment according to the advance ratio fits the open water diagram the fraction of $\frac{1-a}{1+a'}$ has been computed for a range of J values, these can be seen in Figure 5.7.

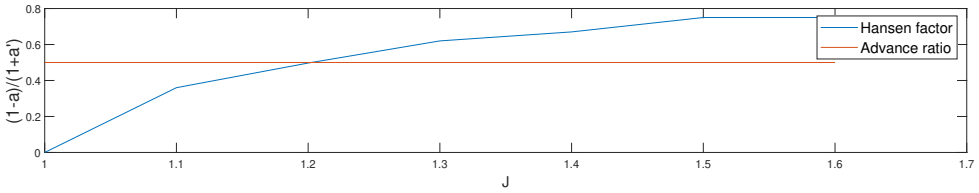



**Figure 5.7:** Fraction of the Hansen model for different advance ratios

The table shows a change in the value of the Hansen fraction for different advance ratios. Therefore it is worth investigating the effect of this fraction on the computation of propellers for really high values of J. To be able to do this, the Hansen model would first have to be implemented within PROCAL. Since this has not yet been done the wake pitch alignment method according to the advance ratio will be considered. This was done since Figure 5.7 shows a good match between the Hansen fraction and the advance ratio as wake pitch alignment and Figure 5.5 results in the best match between modelled data and test data.

# 6

# Surrogate methods

A surrogate is a function that approximates another function. Surrogates are useful in optimisation since the time needed for one evaluation is very short. Therefore its use is most common for the evaluation of time-consuming functions [19, 56]. To approximate a function the surrogate needs a number of input values so it can try to generate the best fit through the different points [23]. Typically the input values are values that have been computed using an expensive function. The exact location and spread of these variables are computed using Design of Experiments (DoE). Next these results are used to construct a surrogate model using data fit surrogate models. Examples of such models are regression splines, Kriging or Radial Basis Functions. A typical surrogate framework can be seen in Figure 6.1.

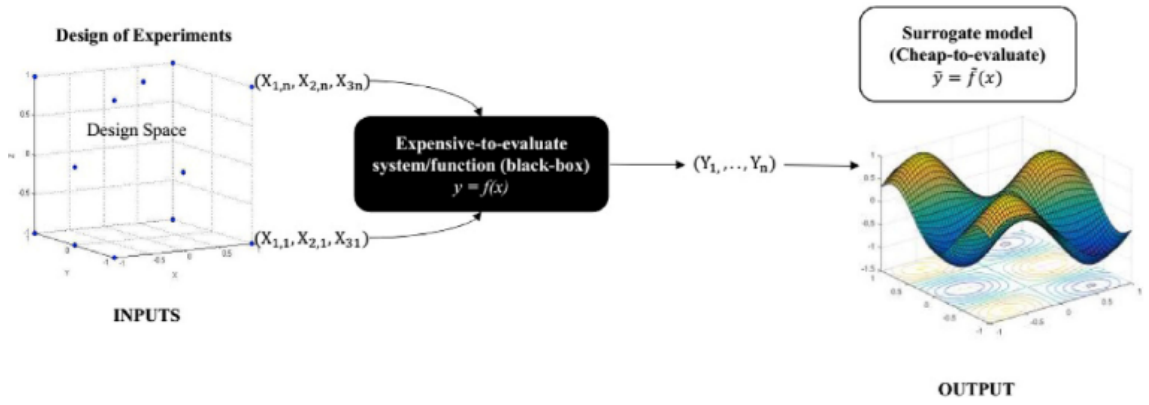



**Figure 6.1:** Basic surrogate framework [56]

Optimisation using surrogates attempts to find a Pareto optimal solution. In design problems there often are multiple local minima but just one global minimum. To find the global minimum multiple known (or computed) data points are provided to the surrogate and the surrogate will fit a function through the dataset. Next a simple minimizer can be used to estimate the minimum. The use of surrogates is a delicate trade-off between exploration and exploitation, the desire for new local minima (extra evaluations) and the exploitation of known data due to limited time or computational costs [23]. Considering Figure 6.1 it would be the last step, where the global minimum in the output would be the desired value.
To effectively implement surrogate optimisation a couple of key steps have to be taken into account:

- Design of experiments (DoE)
- Numerical simulations at selected locations
- Construction of surrogate model
- Model validation

## 6.1. Design of Experiments

The Design of experiments is the sampling plan within the design variable space [19]. The data used for the DoE are sample points that have either been determined by physical experiments or with computational expensive computer models [44]. Depending on the type of surrogate either all data can be used to train the surrogate [19] or part of the data can be stored separately to test the predicting abilities of the surrogates [44, 56]. The spacing of the data points can be based on different spacing methods, examples are random sampling, Latin hypercube or A-optimal sampling. These are merely examples to illustrate that there are multiple possibilities, also multiple methods can be combined to get a more diverse dataset. The spacing of the different datapoints were initially established to be able to control bias and variance in physical experiments where [56, 44]:

- Bias: quantifies the extent to which the surrogate model outputs (i.e.$\hat{f}(\mathbf{x})$)differ from the true values (i.e. $f(\mathbf{x})$) calculated as an average over all possible data sets $D$.

- Variance: measures the extent to which the surrogate model $\hat{f}(\mathbf{x})$ is sensitive to particular data set $D$. Each data set $D$ corresponds to a random sample of the function of interest [44].

Since DoE's were first introduced a change in data type has occurred. Today data is mostly gained via computer simulations instead of physical experiments. This changes the problem from a stochastic to a deterministic problem, eliminating the need for repetition [18, 56]. Since the different nature of the two DoE's origin the physical approach will be referred to as Classical Design of Experiments and the computational approach is called the Modern Design of Experiment (MDoE). MDoE's have been developed parallel to the evolution of computer science and ease access to computing facilities, they need different sampling plans with regard to Classical DoE's [56]. Since all experiments that will be considered in the current research are of computational origin sampling methods for MDoE's will be considered.

### 6.1.1. Random sampling

Random sampling is a method of generating training points that have the same probability to appear in any sample [56]. Therefore a random subset of size $n$ from set of size $N$ is considered. There are multiple methods that can be considered, examples are: simple random sampling, Monte Carlo methods (Markov Chain Monte Carlo Samplings, Gibbs sampling, Metropolis-Hastings sampling, etc.) When using random sampling one should always have a complete and thorough understanding of the entire design space (assuming no missing data) because random sampling often includes clustering of variables and sparsely represented areas if optimising in a higher dimensions [56]. An example of random sampling can be seen in Figure 6.2.

### 6.1.2. Quasi-random designs

Where random sampling takes random samples the quasi variant doesn't really do that but is not completely regular either. Quasi-random designs seek to bridge the gap between the flexibility of random design but without losing the structured advantages of grid designs. Using this technique it is ought to have a high level of uniformity in a multidimensional space (unit hypercube), while not being statistically independent [56]. Two examples of quasi random sampling methods are Sobol Sampling and the Halton Sampling method, an example of Sobol sampling can be seen in Figure 6.3.

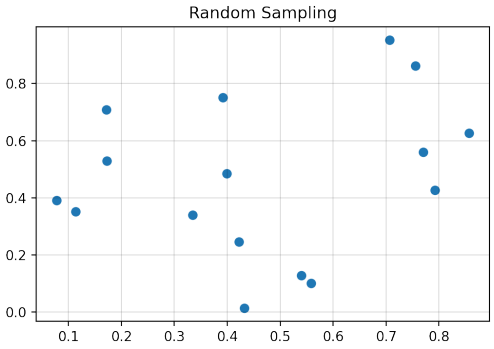



**Figure 6.2:** Random sampling with 16 data points

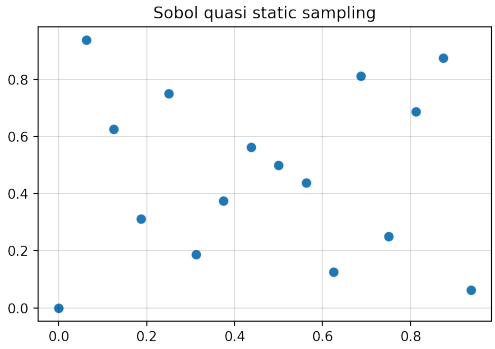



**Figure 6.3:** Quasi Random Sampling using Sobol Sampling

### 6.1.3. Uniform designs

Uniform designs were introduced to prevent the problems that occurred in random design, sparsely sampled areas in the design space. Uniform designs are often robust and flexible in terms of problem dimensionality [56].

### 6.1.4. Latin Hypercube Sampling

Not all MDoE's can be placed in one of the design groups described above. An example of an MDoE sampling method that was developed using key components of multiple designs is the Latin Hypercube sampling (LHS). A $D$-dimensional LHS is a generalisation of a Latin square, which is a two-dimensional grid with at most one point in each row and each column [55, 56]. Due to its variety and flexibility in application it is the most common experimental design used to study numerical models [35, 55, 56]. Within the use of LHS there may be great variation in the space-filling properties of LHS designs. The most prevailing sampling methods are the random, the orthogonal array-based and the maximun Latin hypercube sampling [35, 55, 18]. Here one can clearly see that multiple design components have been implemented in one method, an example can be seen in Figure 6.4.

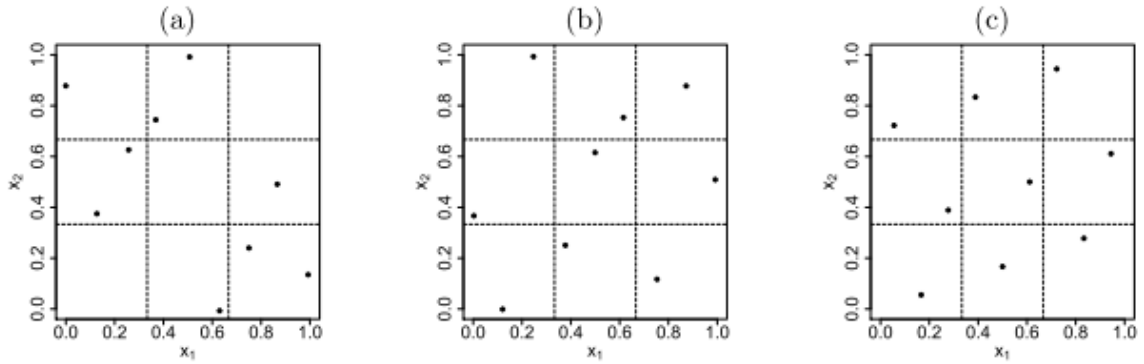



**Figure 6.4:** Latin hypercube samples with n = 9 and d = 2: (a) random LHS; (b) random LHS generated from an orthogonal array; (c) maximin LHS. [55]

### 6.1.5. Radial Basis Functions

Radial Basis Function models are a very accurate tool to interpolate high dimensional highly non-linear problem. RBF's are a different method to approximate $f$ by $\hat{f}$ in a sampled design space where not all $x$'s are known. Just as with polynomial regression and Kriging, we consider that $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{f} \in \mathbb{R}$ are given such that $n$ is any positive integer that is larger than the length of vectors $\mathbf{x}$ and $\mathbf{f}$. Using these formulations a function $\hat{\mathbf{f}}$ can be formulated [23]:

$$\hat{\mathbf{f}}(x) = \sum_{i=1}^{n} \lambda_i \phi(||x - x_i||) + p(x), \qquad x \in \mathbb{R}^n \tag{6.1}$$

This function interpolates the data $(x_1, f_1), ..., (x_n, f_n)$. The coefficients of $\lambda_i$ are the weights of certain basis functions and $|| \cdot ||$ is the Euclidean norm, the absolute distance between two points in $\mathbb{R}^n$. $p(x)$ is a formulation for a polynomial tail that originates from $\Pi_m$ (the space of polynomials) with a degree less or equal to $m$, i.e. it can be expressed as a linear combination of functions [10, 23]. The space of polynomials is an addition that can be used to ensure interpolation in case the basis matrix of the RBF is not invertible [53]. However, it is not necessary for all types of RBF's [45]. In case an RBF is unconditionally positive a polynomial tail is unnecessary as it will result in extra computational time. However, there are multiple RBF's that are not unconditionally positive, therefore a polynomial tail can be added to make them conditionally positive definite (examples are the thin plate spline and Hardy's Multiquadratic) [10]. For $\phi$ different functions can be considered, the most popular Radial Basis Functions can be seen in Table 6.1 [23]. $\xi$ are the basis functions shape parameters or the

**Table 6.1:** Common basis RBF functions [56]

| Name | Radial Function |
|---|---|
| Linear | $\phi(r) = cr$ |
| Cubic | $\phi(r) = (c + r)^3$ |
| Thin Plate Spline | $\phi(r) = r^2 ln(cr)$ |
| Gaussian | $\phi(r) = exp(-cr^2/(2\sigma^2))$ |
| Hardy's Multiquadric | $\phi(r) = \sqrt{(r + c)^2 + \sigma^2}$ |
| Hardy's inverse Multiquadric | $\phi(r) = ((r + c)^2 + \sigma^2)^{-\frac{1}{2}}$ |
| C4 Matérn | $\phi(r) = exp(-\xi r) \times (3 + 3\xi r + (\xi r)^2)$ |
| Wendland's C6 | $\phi(r) = (-\xi)^8 \times (32\xi^3 + 25\xi^2 + 8\xi + 1)$ |
| Euclid's hat | $\phi(r) = \pi(1/12\xi^3) - r^2\xi + (4/3r^3)$ |

variance $\sigma^2$ of the Gaussian basis function [56]. Here, the shape/width parameter ($c$) is approximated as the average Euclidean distance between the samples [56]:

$$c = \left[\frac{max(x) - min(x)}{n_s}\right]^{\frac{1}{n_{dv}}} \tag{6.2}$$

where:

$$\mathbf{\Phi}_{i,j} = \phi(||x_i - x_j||), \qquad i, j = 1, ..., n \tag{6.3}$$

In case the order of the polynomial tail ($\Pi_m$) is larger than $0$ the matrix of $\Phi$ has to be extended. Therefore matrix $P$ is introduced, where $p_1, ...p_{\hat{m}}$ is in the basis of the space. This matrix consists of [23, 53]:

$$\mathbf{P} = \begin{bmatrix} p_1(x_1) & \cdots & p_{\hat{m}}(x_1) \\ \vdots & & \vdots \\ p_1(x_n) & \cdots & p_{\hat{m}}(x_n) \end{bmatrix} \tag{6.4}$$

Using $\Phi$ and $P$ one can construct the matrix that describes the RBF [23, 53]:

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \tag{6.5}$$

Now the values of $\lambda$ and $\mu$ can be determined as they are the weights for the RBF. For each unseen input $x'$ the value of $f'$ can be predicted:

$$\hat{\mathbf{f}} = \mathbf{\Phi} \cdot \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} = \sum_{i=1}^{n} \lambda_i \phi(||\hat{x} - x_i||) + \mu_0 + \sum_{l=1}^{d} \mu_l \hat{x}_l + \sum_{l=1}^{d} \mu_l \hat{x}_l^2, \qquad x \in \mathbb{R}^d \tag{6.6}$$

RBF's yield good results for a wide range of sample sizes and designs. They tend to produce accurate approximations to response functions with various patterns of both random and deterministic errors. They are relatively accurate and robust for high-order nonlinear and small scale problems but the evaluation of the model parameters involve computational complications, especially with large numbers of design parameters, resulting in high expenses regarding computational costs as well as memory [1].

## 6.2. Surrogate-assisted optimisation

As explained at the start of this chapter, surrogates are used to approximate a function of $f(x)$ as $\hat{f}(x)$ by using interpolation or regression between training data. These values are spaced by design of experiments and can used for interpolation or regression by means of different Surrogate models. The final function $\hat{f}(x)$ can be used in the optimisation of design problems. This is done by:

$$\text{Minimise: } \hat{f}(\mathbf{x}) \qquad D \to \mathbb{R}^d, (f_1(x), ..., f_n(x)) \tag{6.7}$$

$$\text{subject to: } g_i(x) \leq 0 \forall i \in \{1, ..., m\} \qquad \mathbf{x} \in D \subset \mathbb{R}^n \tag{6.8}$$

Where $g_i(x)$ are the constraints within the design space. A maximised function can also be considered by just adding a minus [53]. To improve accuracy in the optimisation process one can start with a limited sample set that results in a coarse RBF fit. Next, based on the found results of iteration one a new sample set can be computed resulting in an updated function that is more accurate. Using this approach accuracy can easily be traded for computational costs and speed.

### 6.2.1. Pareto-optimal

At the start of this section a single objective optimisation is given as an example. However, for many cases there are multiple objectives. If there are more than one objective results will most likely move towards the two objectives until no improvements can be made for one objective without degrading another objective. This is called a Pareto-optimal solution. Often there are multiple Pareto-optimal solutions that are located on a so called 'front', this is called the Pareto front. If a Pareto front is reached a trade-off has to be made between the two objectives on how to determine the best solution [36].

There are multiple ways of measuring the performance of multiple-objective optimisation methods. The method that will be used in the upcoming sections is the hypervolume. The hypervolume is the area (2-objectives) or volume (3-objectives) that is mapped between a pre-defined reference point and the computed points on the Pareto-front. An example can be seen in Figure 6.5 [54]. Every extra evaluation that is an addition to the pareto frontier will increase the hypervolume. Therefore it can be concluded that the largest hypervolume is a fine measure to indicate algorithm performance.

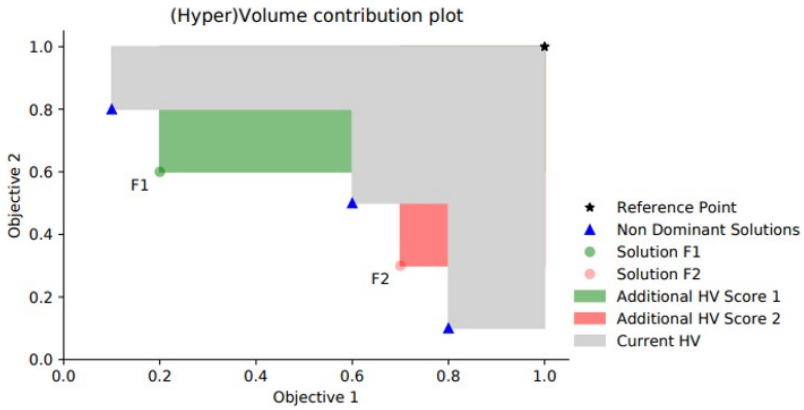



**Figure 6.5:** A representation of a hypervolume that is extended by two newly computed datapoints [54]

### 6.2.2. SAMO-COBRA

An example of a state of the art Surrogate Assisted Constrained Multi-objective Optimisation Algorithm is SAMO-COBRA [53]. SAMO-COBRA's optimisation process exists of two phases, first the *'Initialising-phase'* is started. Here SAMO-COBRA computes a DoE based on one of the built in sampling methods (LHS, Halton, Random sampling or Boundary sampling). After the DoE has been evaluated *'Phase-II'* is started. Here, SAMO-COBRA fits multiple RBF functions (cubic, gaussian, multiquadric inverse quadratic, inverse multiquadric and thinplatespline) for all objectives and constraints. All RBF's are

fitted normal, as an augmented RBF and with or without a logaritmic transformation [53].

$$Plog(y) = \begin{cases} +ln(1+y), & \text{if } y \geq 0 \\ -ln(1-y), & \text{if } y \leq 0 \end{cases} \tag{6.9}$$

Now Constrained optimisation by linear approximations (COBYLA) is used to search for new feasible Pareto-optimal solutions. COBYLA is a method that can be used for nonlinear contrained optimisation calculations when there are no derivatives. Each iteration uses a linear approximation of the objectives and constraints by interpolation between the known data points [43]. To measure the Pareto-optimal solution two methods can be used, the first method interpolates over the RBF to predict the largest possible increase in hypervolume. This method is called the predicted progress in hypervolume progress (PHV). The alternative is to compute all objective scores and subtract all uncertainties (which is similar as in Kriging). This is called the S-metric criterion [53]:

$$\hat{U}_{RBF} = \phi(||\mathbf{x}' - \mathbf{x}'||) - \Phi'^{T}\Phi^{-1}\Phi' \tag{6.10}$$

Within COBYLA a convergence criterion is given, if the approximations have converged or if a maximum number of iterations is exceeded. The newly predicted $\mathbf{x}'$ is evaluated and used to find the best RBF. This is done by minimising the Root mean squared error (RMSE) between the predicted and evaluated results. For this process all RBF functions that predicted a growth in hypervolume over the last $n$ generation are used. $n$ is a hyperparameter that is currently set to two but can be increased or decreased depending on the problem. This process is repeated until the computational budget has been exceeded.

## 6.3. Benchmark tests

To be able to optimise a propeller for propulsive and regenerative operations using surrogate-assisted optimisation methods a couple of steps have to be taken. First SAMO-COBRA has to be tested to check whether it really outperforms NSGA-II and CMOPSO, next SAMO-COBRA has to be coupled to PropArt to obtain a functioning process and finally some alternations to the wake modelling have to be made to ensure that PROCAL stays accurate when modelling turbines.

## 6.4. Benchmark algorithms

The two most promising algorithms that have already been implemented in PropArt are NSGA-II and CMOPSO. NSGA-II (non-dominated sorting genetic algorithm II [13]) is a genetic algorithm that generates random samples and determines which ones are elite based on a non dominated sorted method to determine the front. All samples are compared pairwise and the ones closest to the front are stored while the losing (non dominating) samples are neglected. Every generation new samples are generated. This process is repeated until the budget is exceeded. [13]
CMOPSO, a particle swarm optimiser, is a similar algorithm. Just as in NSGA-II elite particles are selected and stored for the next generation. The main difference occurs when new particles are computed. To determine the location two randomly selected elite particles are compared. The one that loses is updated based on the velocity of the other particle. This process is visualised in Figure 6.6. The non-random process of CMOPSO enables the algorithm to reach a front faster [58]. During the literature review alternative optimisation methods have been considered. Due to the high dimensional nature of the design problem of a propeller a surrogate assisted constrained multi-objective optimisation algorithm (SAMO-COBRA) was proposed.

## 6.5. Benchmark tests

To test if SAMO-COBRA is competitive it is compared to NSGA-II and CMOPSO for multiple analytical test functions. Since the algorithm will be used to evaluate an expensive function the budget was set to 1000 evaluations. For NSGA-II and CMOPSO generation sizes of 10, 50 and 100 were used. This was done since the number of generations and generation size within a computational budget do have influence on the outcome and performance of genetic- and particle swarm optimisers.
SAMO-COBRA on the other hand is a non-random process and was therefore given 100 evaluations, if it did not perform best more evaluations were allowed. In that case 200 and 1000 evaluations were

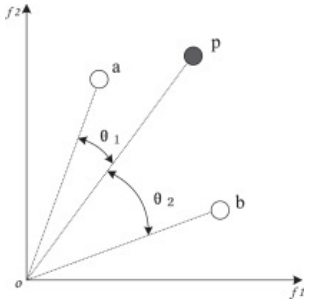



**Figure 6.6:** Pairwise competition between two elite particles [58]

computed. The performance was measured using the hypervolume which can be computed as a post processing step.

**Table 6.2:** Hypervolume for analytical testfunctions tested in NSGA-II, CMOPSO and SAMO-COBRA for different generation sizes the mean of 30 runs has been considered. All test functions have been added in Appendix A

| | NSGA-II | | | CMOPSO | | | SAMO-COBRA (n-iterations) | objectives/ constraints/ variables |
|---|---|---|---|---|---|---|---|---|
| | 10 | 50 | 100 | 10 | 50 | 100 | | |
| MW2 | 0.13 | **0.41** | 0.26 | 0.16 | 0.36 | 0.073 | 0.39 (1000) | 2/1/15 |
| MW3 | 0.025 | 0.21 | 0.51 | 0.028 | 0.19 | 0.42 | **0.77 (200)** | 2/2/15 |
| C3DTLZ4 | 2.65 | 3.32 | **3.33** | 2.66 | 3.00 | 3.00 | 2.28 (200) | 2/2/12 |
| BNH | 4536 | 5210 | 5247 | 4548 | 5225 | 5255 | **5262 (100)** | 2/2/2 |
| SRN | 13309 | 16177 | 17288 | 14777 | 17103 | 17343 | **17437 (1000)** | 2/2/2 |
| CEXP | 3.35 | 3.72 | 3.72 | 3.51 | 3.74 | 3.75 | **5.77 (100)** | 2/1/2 |
| DTLZ2 | **0.69** | 0.53 | 0.48 | 0.31 | 0.54 | 0.52 | **0.69 (200)** | 2/1/10 |
| WFG1 | **5.13** | 3.75 | 0.62 | 1.76 | 0.37 | 0 | 0.81 (1000) | 2/1/10 |

Overall, SAMO-COBRA performs best on the majority of the testfunctions. Therefore it is interesting to check its performance in a propeller optimisation. However, there are some more conclusions that can be made from the test functions that have been computed. For low dimensional problems (CEXP and BNH testfunction) it can be seen that SAMO-COBRA outperforms NSGA-II and CMOPSO with very few iterations. For the SRN testfunction SAMO-COBRA performs slightly better than CMOPSO, this is mainly due to the maximum possible hypervolume that can be achieved (17440), therefore it can be concluded that both performed similar. For high dimensional problems, more computations are needed. If SAMO-COBRA's computed hypervolume after 100 evaluations did not outperform NSGA-II and CMOPSO 200 and 1000 samples were computed. This showed that most problems can generate a larger hypervolume in fewer iterations. All results can be seen in Table 6.2. The biggest achieved hypervolume per test function has been printed in bold, as one can see only test problems MW2, C3DTLZ4 and WFG1 performed better using a genetic or particle swarm algorithm (DTLZ2 is considered a tie). It is interesting to see that all but one testfunctions with multiple constraints performed best with SAMO-COBRA while NSGA-II and CMOPSO did outperform SAMO-COBRA for most single constrained functions.
It can be seen that CMOPSO outperforms NSGA-II when using small generation sizes. For the smallest generation sizes (10) and a budget of 1000 evaluations CMOPSO beats NSGA-II on 5 out of 8 algorithms. This is also in line with the paper of Zhang [58] where it is shown that CMOPSO beats NSGA-II for relatively small generation sizes (50). This is probably due to the pairwise competition that is used in CMOPSO (Figure 6.6, This enables the algorithm to make accurate predictions for smaller generation sizes. For a generation size of 50 the algorithm performs similar to NSGA-II, for large generation sizes (and therefore a small number of generations) NSGA-II outperforms CMOPSO. This is probably due to the fact that pairwise competition is not yet profitable.
When analysing the hypervolume progress for the MW2 testproblem it can be seen that when CMOPSO finds a hypervolume it keeps increasing while the hypervolume can actually decrease for NSGA-II. This

is due to the random generation of new samples. The possible decrease in hypervolume in a new generation has an effect on the reliability of NSGA-II. This is due to the decrease in hypervolume that in fact shows a decrease in the ability of reaching the objective after more iterations. The variation in results for the 30 test runs is also striking, CMOPSO seems to be more consistent than NSGA-II throughout its procedure resulting in comparable results for most runs. For both, NSGA-II and CMOPSO multiple examples of data crowding have been found. An example can be seen in Figure 6.7. It occurs mostly on constrained problems where a part of the generation violates the constraint.

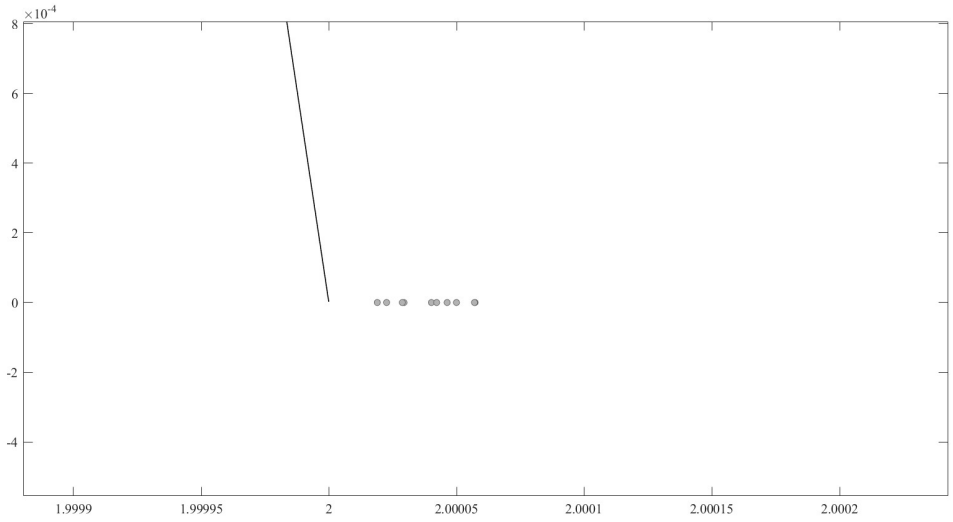



**Figure 6.7:** An example of data crowding close to the pareto front. The front spans from (0,2) to (2,0)

When zooming in on CMOPSO it can be seen that the hypervolume does start larger for a generation size of 50, probably due to a wider spread of initial samples. However, the growth does decrease fast for a larger generation size, whereas the progress stays more or less constant when looking at the generation size of 10. The test for SAMO-COBRA shows a fast increase in hypervolume at the start of the problem and afterwards only seems to be increasing slowly. SAMO-COBRA did find a hypervolume at 100 iterations for almost all problems, more iterations ensured an increase in hypervolume but the rate of hypervolumeprogress was low. The same can be seen for different testfunctions, an example is the BNH testfunction, it was run 100 iterations and the result can be seen in Figure 6.15 where it can also be seen that the hypervolumeprogress does increase fast at first (and early in the iterational progress) and stagnates later on.

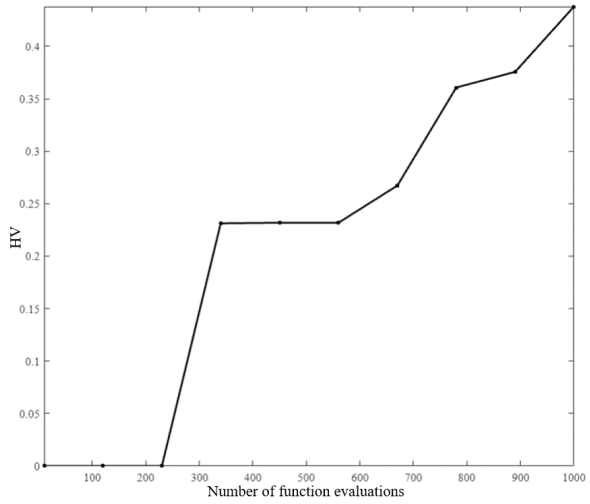



**Figure 6.8:** Hypervolume progress for MW2 with CMOPSO, generation size 10 and a budget of 1000

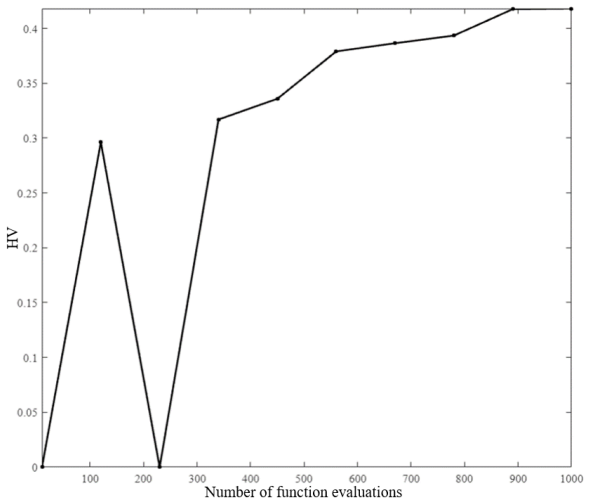



**Figure 6.9:** Hypervolume progress MW2 with NSGA-II, generation size 10 and a budget of 1000

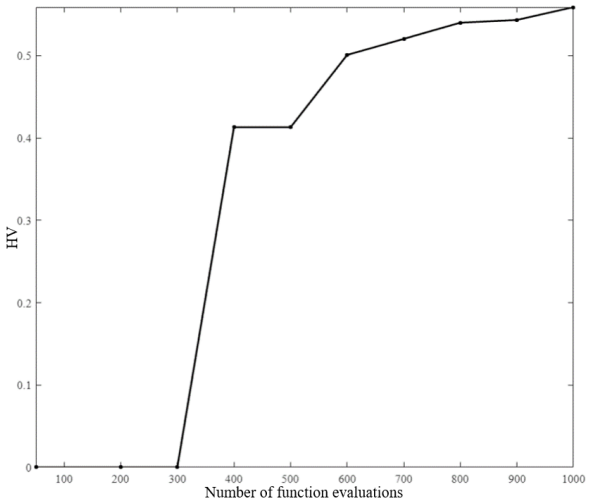



**Figure 6.10:** Hypervolume progress for MW2 with CMOPSO, generation size 50 and a budget of 1000

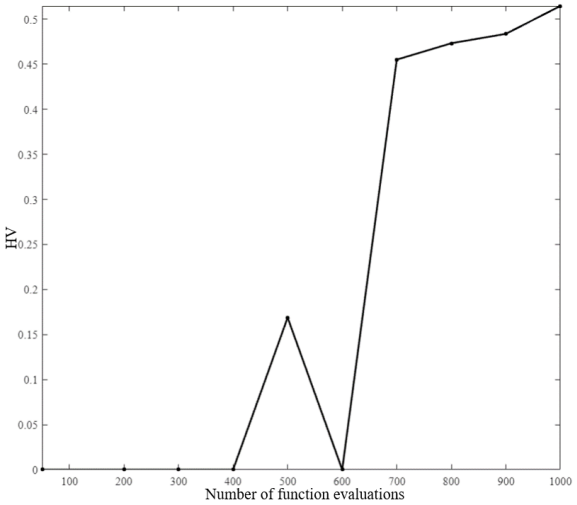



**Figure 6.11:** Hypervolume progress MW2 with NSGA-II, generation size 50 and a budget of 1000

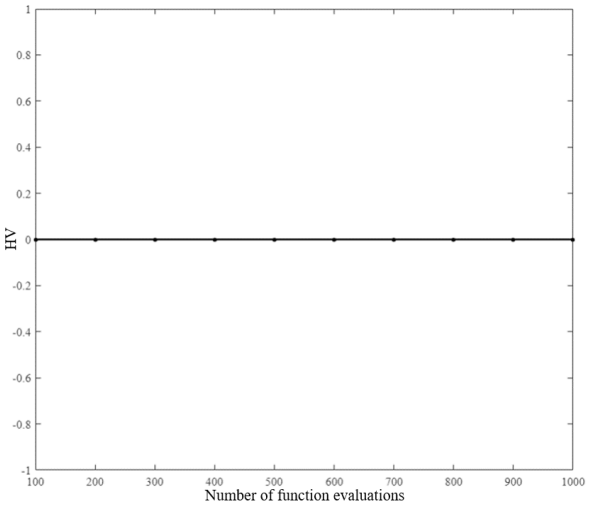



**Figure 6.12:** Hypervolume progress for MW2 with CMOPSO generation size 100 and a budget of 1000

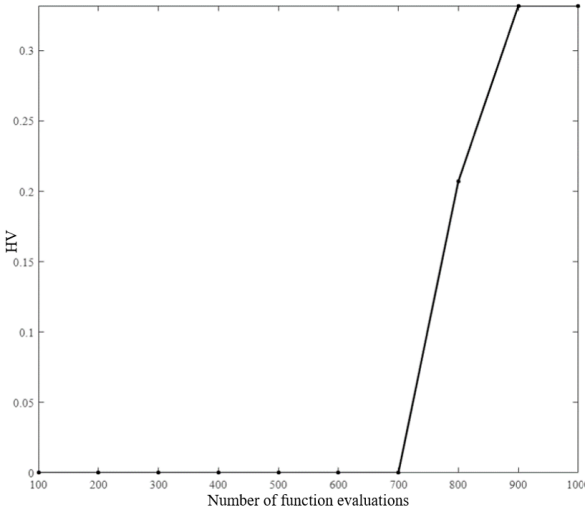



**Figure 6.13:** Hypervolume progress for MW2 with NSGA-II generation size 100 and a budget of 1000

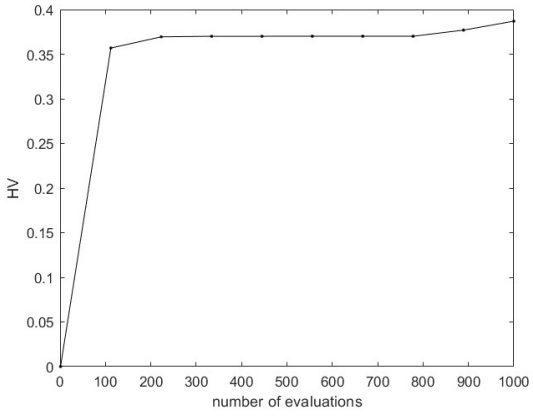



**Figure 6.14:** Hypervolume progress for MW2 with SAMO-COBRA, the budget is 1000 evaluations

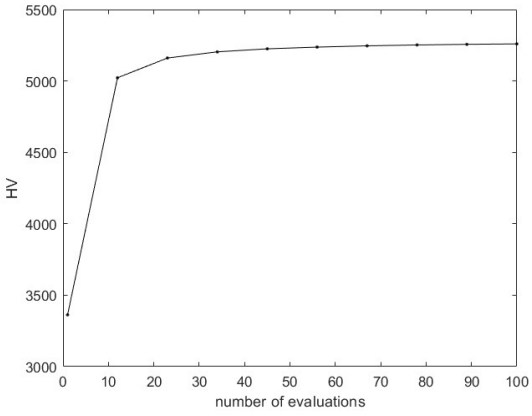



**Figure 6.15:** Hypervolume progress for BNH with SAMO-COBRA, the budget is 100 evaluations

# 7

# Model and Test case

## 7.1. Workflow

To create a model where SAMO-COBRA can communicate with PropArt, SAMO-COBRA should generate sampleplans based on the objectives and constraints from prior generations. To do so, a workflow is made. First, the problem is defined in PropArt. This is done since there are a lot of settings that are important for the propeller geometry that have to be defined in PropArt and are not needed for SAMO-COBRA. When using this method these settings do not have to be converted from python to MATLAB for every single iteration. As well, all *'Preflight'* checks (meaning, the checks that are performed to ensure a sound propeller) can easily be performed when the problem definition is done in PropArt.
The input and settings that are considered as variables are exported from MATLAB to python using the option *'Variablelist'* in PropArt. The variablelist contains all variables, their upper and lower limits, the number of objectives and the number of constraints. This file is imported to SAMO-COBRA, where the upper and lower bounds per variable are read. These parameters are used to determine the design space that can be mapped in the *'Intialising-phase'*. Here SAMO-COBRA computes its initial samples, this can be done with multiple sampling methods, however an LHS method with a center maximin approach and the Halton sampling method seem to perform best when comparing the different test problems. For these test problems LHS and Halton return the largest hypervolume after the *initialising-phase* . However, LHS seems to be able to return more feasible results when computing multiple constrained propellers and has therefore been chosen as DoE method. The default size is the number of variables+1, since there is some divergence within PropArt (not all propellers can be evaluated due to violation of equality constraints in PropArt) 40 samples were considered. When using LHS it was found that about $80-90$% of the samples can be evaluated while still satisfying the equality constraints. This will result in approximately the number of variables+1 samples that meet the equality constraints. The constructed initial sampleplan is an $m \times n$ matrix with $m$ samples and $n$ variables. The matrix is exported to a *.txt* file after which PropArt (in MATLAB) is started using python. PropArt now imports the sampleplan and performs all necessary checks on the propellers equality constraints (these are either true or false, for example whether a CPP blade can pass the other blades freely), if the propeller passes it will be evaluated and the resulting objectives and inequality constraints are written as a *.mat* structure. If an equality constraint can not be met PropArt will return *inf* for the objectives and constraints, SAMO-COBRA is altered to ignore these. Since these computations can not be used for the optimisation part SAMO-COBRA will not count these values as function evaluations and will evaluate more propellers to reach its budget. In case a design exceeds an inequality constraint the returned constraint value is positive (therewith making this result infeasible). The samples where only inequality constraints are violated do count as function evaluations because SAMO-COBRA can use the resulting objectives and constraints as data to train the RBF.
When the DoE is computed SAMO-COBRA continues by importing the values of the objectives and constraints. When this is done *'Phase-II'* starts. After phase-II has been started COBYLA is used to search for new feasible Pareto-optimal solutions. This is done by maximising the hypervolume by interpolating over the RBF to find the point where the largest predicted progress in hypervolume is expected (PHV) [53]. To do so, COBYLA is tested on convergence in its prediction, if that is reached the

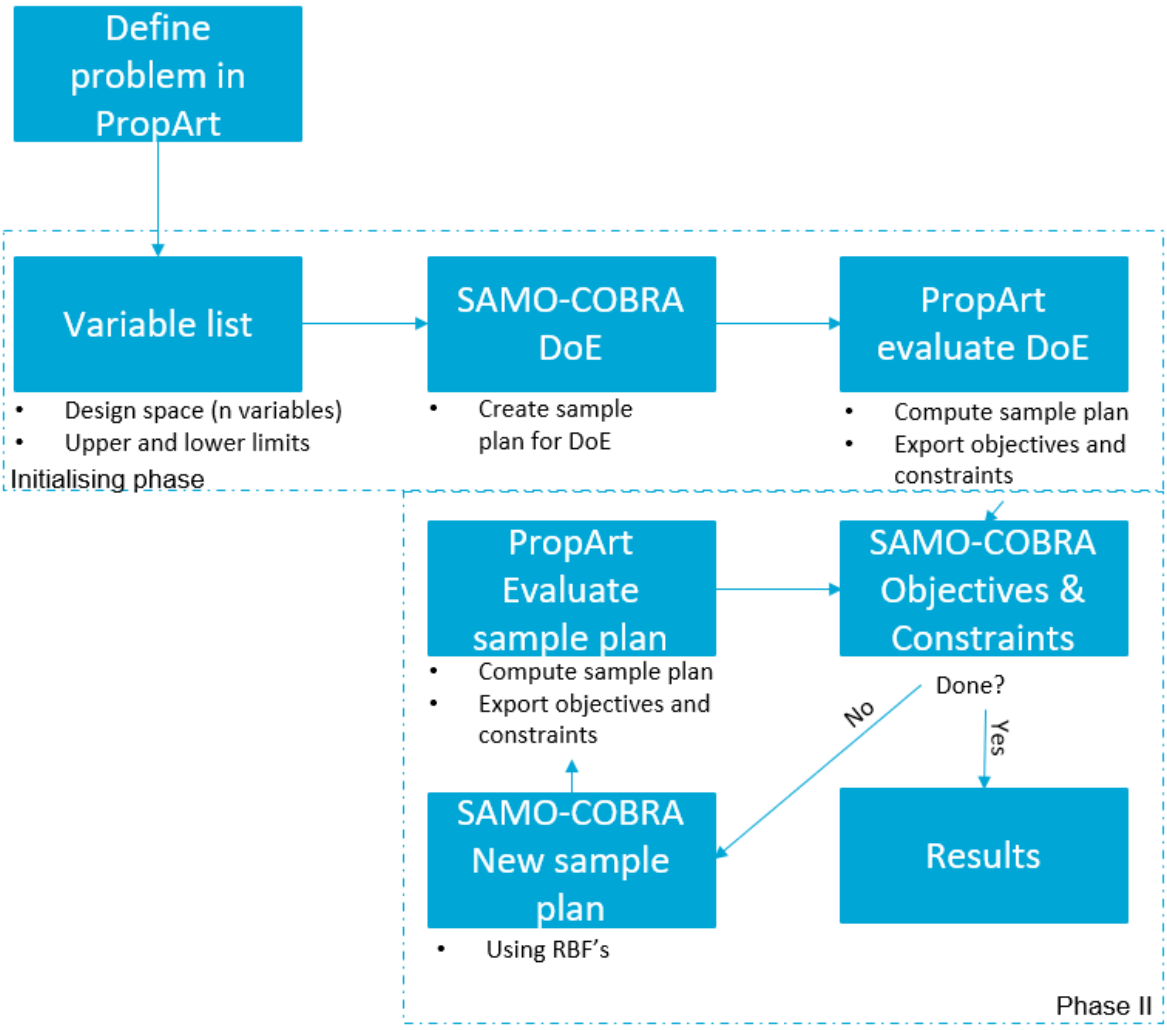



**Figure 7.1:** An overview of the SAMO-COBRA workflow

process continues to the next objective or constraint. If the convergence is not successful a maximum number of iterations is given, this number is defined by the sum of the number of variables, objectives and constraints multiplied by 50 [53].
The new predicted $\mathbf{x}'$ is evaluated and the RBF with the smallest RMSE is used to select the values ($\mathbf{x}'$) for new samples. This process is repeated until the budget is exceeded. A schematic overview of the structure can be seen in Figure 7.1. The testcase uses a batch size of 20 (to avoid divergence of an entire generation) out of which an average of $\pm 60 - 70\%$ is returned. 20 evaluations is also the maximum to evaluate on the cluster while still using one node. To try and reach a proper optimisation approximately 30 times the number of variables has been set as the budget.

## 7.2. Setup testcase

To test the three designs a case has been chosen, Project ZERO. This is a sailing vessel that has been designed by Dykstra Naval Architects. The aim of project ZERO was to design a sailing vessel that can travel the world with zero emission. To be able to do so, the propellers are to be optimised for propulsion as well as for regeneration. The initial design has two propellers, a small one in front of the keel that is primarily used for regeneration and a larger one behind the keel that is used for propulsion as well as for regeneration. For the optimisation only the aft propeller is considered. The submergence of the aft propeller is $2.94m$ at the shaft and the propeller diameter is $1.5m$. To optimise the propeller, six computations have to be computed per iteration, three for propulsion and three for regeneration.

These can be seen in Table 7.1. Within one computation it is assumed that the ship speed is constant and only the rps can be optimised within $80\%$ and $120\%$ of the initial propeller rpm.

**Table 7.1:** Design characteristics of the two design settings

| | Regenerative | | | Propulsive | | |
|---|---|---|---|---|---|---|
| | 1 | 3 | 5 | 2 | 4 | 6 |
| computation | turbine mode | overload turbine | separation analysis | propulsion mode | overload propulsion | separation analysis |
| Speed | 16 kts | 21.33 kts | 16 kts | 12 kts | 9 kts | 12 kts |
| rpm | 276 rpm | 276 rpm | 276 rpm | 345 rpm | 345 rpm | 345 rpm |
| Thrust | -26.9 kN | -26.9 kN | -26.9 kN | 32.2 kN | 32.2 kN | 32.2 kN |
| computation | unsteady | unsteady | steady | unsteady | unsteady | steady |

Computation one and two are for normal operations and will be used to determine the objectives where computation one is also used to determine the geometrical constraints regarding skew balance and blade thickness. The third and fourth computation are used to determine the cavitation bucket in an overloaded condition. The overload constraints are needed since the assumption is made that the vessel is sailing at a constant speed in calm water. Since this is not possible (there will be waves and wind gusts that have effect on the speed) a separate condition at a higher ship speed for regeneration and a lower ship speed for propulsion (there is more drag) are introduced. These two conditions are in overload since the blade loading will increase for an increase in speed at a constant rpm for a turbine and a decrease in speed at a constant rpm for propulsive operation. The fifth and sixth computation are needed to compute constraints regarding separation of flow. To be able to compute the separation a steady flow calculation is needed, since computations 1,2,3 and 4 are unsteady two more had to be added. When computing the two separate designs, for regeneration the uneven numbers are considered and the even numbers are considered for propulsion. Since the basic design of the hull has already been finished a propeller hub and wakefield have been provided as well, these can be seen in Figure 7.2 and Figure 7.3.

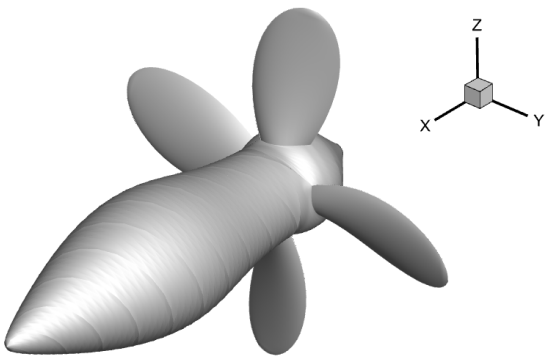



**Figure 7.2:** The propeller hub of Project ZERO

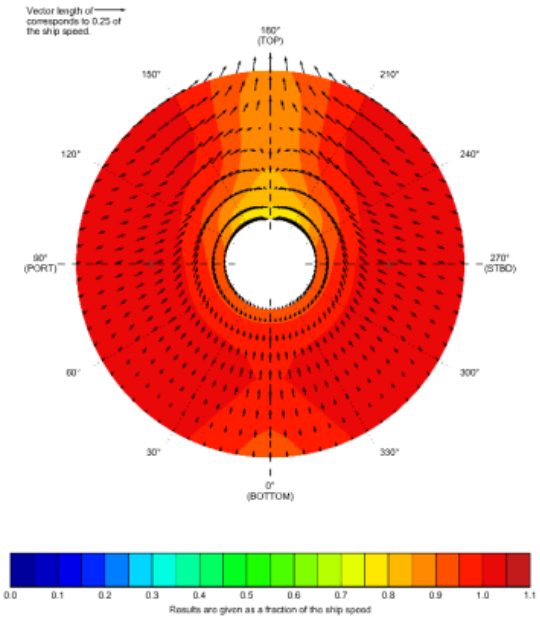



**Figure 7.3:** A view of the wake field of the propeller

### 7.2.1. Variables

To be able to optimise the propeller the propeller an initial Hundested propeller has been considered. This geometry is defined by the following variables referring to different sections of the blade:

- chord length
- pitch
- thickness
- camber
- rake
- skew

These define the blade geometry in the regenerative case. If the geometry is fixed, it is used to evaluate the propulsive case where only the rpm and the pitch can be altered. For the cases with one propeller the problem has 36 variables and for design case 1 the problem has 35 variables. A full list of the variables can be seen in Appendix B.

## 7.3. Objectives

When optimising the three design cases do need objectives as well. The separate design case strives a maximum regenerative power in turbine mode and a maximum behind efficiency for the propulsive computation. The normal CPP (first quadrant) and the reversely loaded case (third quadrant) are optimised for a maximum regenerative power (for the given maximum drag from Table 7.1) and a minimum propulsive power (that is needed for the thrust provided in Table 7.1).

### 7.3.1. Constraints

To ensure that the designed propeller meets all requirements multiple constraints have been included. These constraints can be categorised in three categories: there are geometrical limitations, cavitational limitations and constraints regarding seperation of flow. To satisfy the different classes multiple constraints have been added per class, a short summary can been seen below and the full list of constraints can be seen in Appendix C.

Cavitational constraints

- Computation one and two should be free of mid chord cavitation
- Computation three and four should be free of mid chord cavitation
- Circulation at the hub should be limited to avoid strong hub vortices
- Computation one and two should be free of cavitation

When optimising the blade, the pressure peak at the leading edge of the blade will be maximised. In other words the pressure will be spread more evenly over the blade. If the pressure can not be spread evenly the pressure level might exceed the vapour pressure at the leading edge (sheet cavitation) or mid chord (bubble cavitation). Therefore the first cavitational constraints are needed to prevent these phenomena. Secondly constraints are needed to limit the circulation around the hub to avoid erosive hub vortices.

Geometrical constraints

- The skew should be balanced (and no skew angles larger than 20 degrees are allowed)
- The propeller should satisfy Bureau Veritas class rules regarding blade thickness
- The trailing edge in computation one should have sufficient blade thickness to be used as a leading edge according to class rules

The geometrical constraints are used to balance the blade and satisfy class rules. Since the blade is rotated such that the trailing edge is used as a leading edge, the propeller in third quadrant has an additional geometrical constraint that the trailing edge should be thick enough according to class rules for a leading edge.

Seperation of flow constraints

- No flow separation should occur in computation five and six. This is determined by using the Twaites head criterion.

The Twaites Head criterion is a criterion that was developed in the seventies and could be used to predict separation over a blade. This is done by computing the momentum integral over the blade when in steady flow (equations of Thwaites). The ratio between the estimated momentum and displacement thickness is considered. For turbulent flow, when this shape factor exceeds a certain value flow separation is assumed to follow [9].

# 8

# Results

This chapter will provide more insight in the different Pareto plots and optimisation data regarding constraint violation and optimisation time as well as results about the optimised propeller geometries and the differences between the three cases. To get results from the different optimisation processes, SAMO-COBRA evaluated a DoE of 100 samples. After the DoE, a batch larger than one was needed to avoid divergence of the entire generation. To use the full extent of the cluster, a batch size of 20 was chosen (a node on the cluster has 20 cores). On average $\pm 70 - 80\%$ per generation is returned as feasible. To be able to obtain a sufficiently large data set over 600 feasible propellers have been evaluated per run. For all runs this resulted in approximately 50 evaluated generations.

## 8.1. Pareto plots

For every run a Pareto plot was made where the objectives are provided on the x- and y-axis. All Pareto plots try to maximise the objective on the x-axis while minimising the objective on the y-axis. For this set of evaluations no propellers were found that satisfy all constraints. Therefore this section will first provide the results of the computation with all constraints.

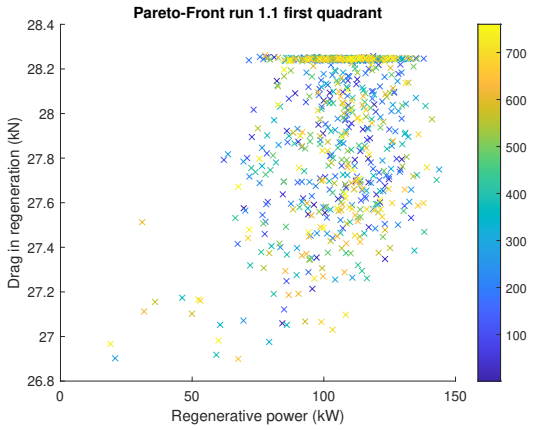



**Figure 8.1:** Pareto plot of the first run in regeneration in the first quadrant

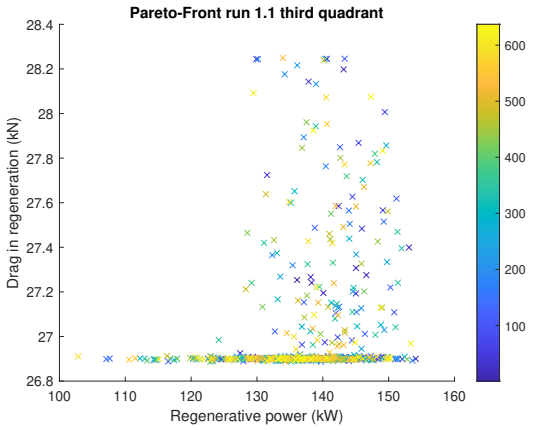



**Figure 8.2:** Pareto plot of the first run in regeneration in the third quadrant

Figure 8.1 shows that there is a hard limit at a drag of $28.2kN$. This is due to the setpoint that has been provided in the description. SAMO-COBRA tries to maximise the regenerative power while minimising the drag. The maximum regenerative power that can be obtained is close to the regenerative power that is found in Figure 8.4. Figure 8.2 shows a propeller that is optimised for regeneration with the propeller blades oriented as in configuration three. This run resulted in a lot of samples that have a drag of $26.9kN$, which is similar to the minimum value that has been found in Figure 8.1. The most interesting optimisation parameter for this optimisation is the regenerative power. When comparing Figure 8.1 and Figure 8.2 it is shown that the run in the third quadrant can achieve a larger regenerative power. For

configuration 1.2 the spread in behind efficiency can be seen. Towards the lower right corner of the plot, the start of a Pareto-Front is visible.

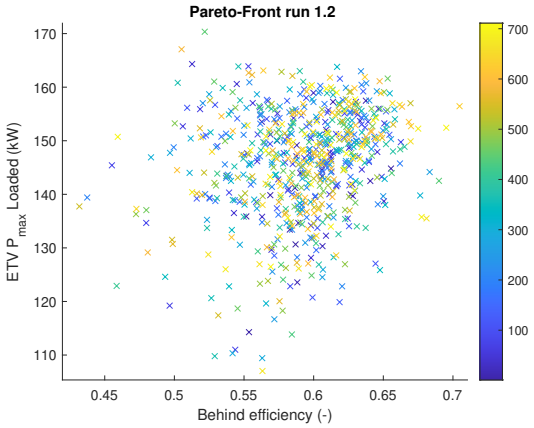



**Figure 8.3:** Pareto plot of the first run in propulsion

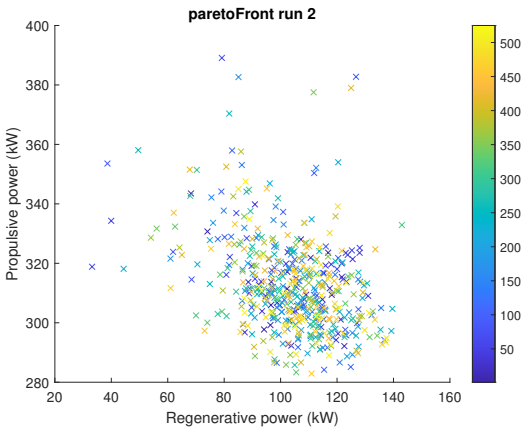



**Figure 8.4:** Pareto plot of the second configuration

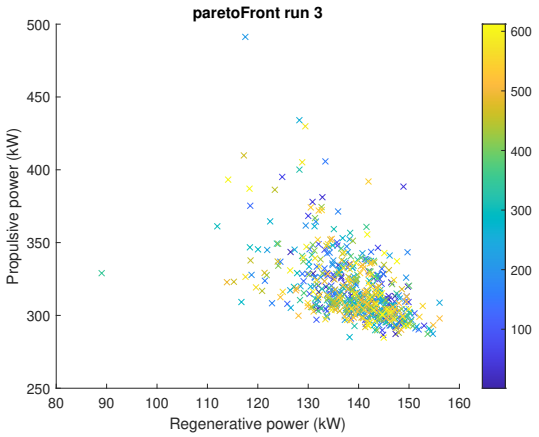



**Figure 8.5:** Pareto plot of the third configuration

Configuration two and three maximise the regenerative power while minimising the propulsive power. Figure 8.4 and Figure 8.5 present an interesting trade-off between the maximum achievable regenerative power and the needed propulsive power. Here it can be seen that configuration two can achieve the smallest propulsive power needed. The minimum needed propulsive power is $282kW$. For run three this is $285kW$. The regenerative power however shows a larger difference. Run two can achieve a maximum regenerative power of $143kW$ while run three can reach a maximum regenerative power of $156kW$. When comparing the points on the Pareto Front, run two shows a wider spread of values on a front such that a trade-off can be made between regenerative- and propulsive power while run two sees more convergence towards one point.

## 8.2. Constraint violation

Since every propeller violates at least one constraint, some alternations are needed to obtain feasible results. The figures below (left) show the constraint number per run versus the number of constraint violations. The figures on the right show a boxplot where the median is shown in red, the 25th and 75th percentile are shown in the box and the maximum values are shown using the dotted lines. Outliers are shown using a red cross [39].

### 8.2.1. Run 1.1 first quadrant

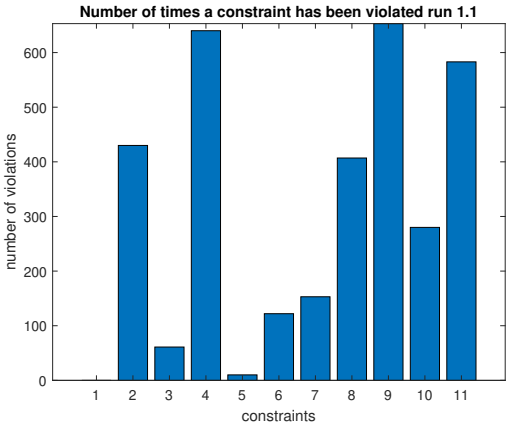



**Figure 8.6:** Number of times a constraint is violated for run 1.1

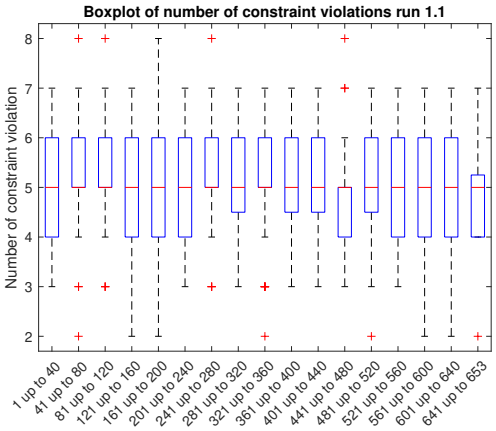



**Figure 8.7:** Boxplot of the constraint violation for run 1.1

Run 1.1 in the first quadrant shows that the violated constraints are most often constraint 4, 9 and 11. These constraints are:

- Constraint 4: Class compliance
- Constraint 9: The positive inception bucket in regeneration
- Constraint 11: Pressure side separation limit regeneration

In comparison to the other configurations run 1.1 differs in the violation of the class compliance and the separation analysis. Optimisation 1.1 optimises a regenerative propeller similar to optimisation 2. The literature in section 2.3 shows that it is best if the leading edge has a relatively large blade thickness and a limited camber. The violation of the separation constraint was to be expected for a blade that operates at a negative angle of attack. When taken a closer look at Figure 8.7 it can be seen that the minimum number of violated constraints is two (in total, 8 propellers). After analysing these propellers, one propeller was able to satisfy all but the cavitation constraints. This was the $336^{th}$ propeller. The violated constraints are constraint 2 (cavitation margin on the pressure side) and 9 (inception bucket constraint).

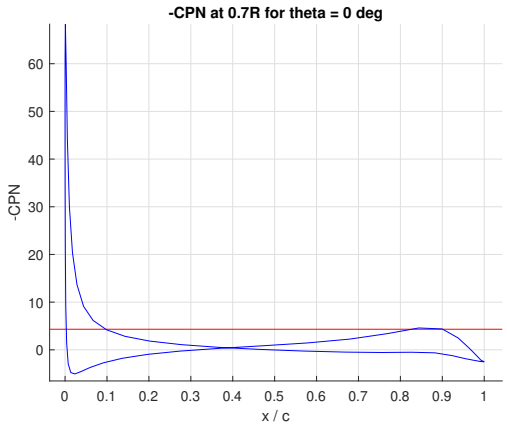



**Figure 8.8:** The pressure level over the blade at 0.7r

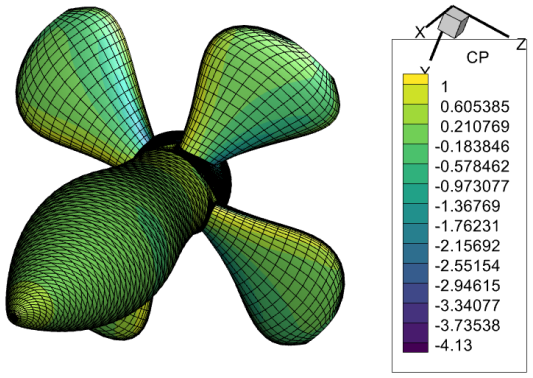



**Figure 8.9:** The minimum pressures plotted on a render of the propeller blade

Figure 8.8 shows the local pressures over the chord length in blue and the local cavitation number can be seen in red. It can be seen that there is a large pressure peak at the leading edge. Close to the trailing edge the pressure exceeds the cavitation number. This can also be seen in Figure 8.9. The objective that is of most interest is the regenerative power. The evaluated propeller can regenerate is $130kW$, while the drag is $28.2kN$

### 8.2.2. Run 1.1 third quadrant

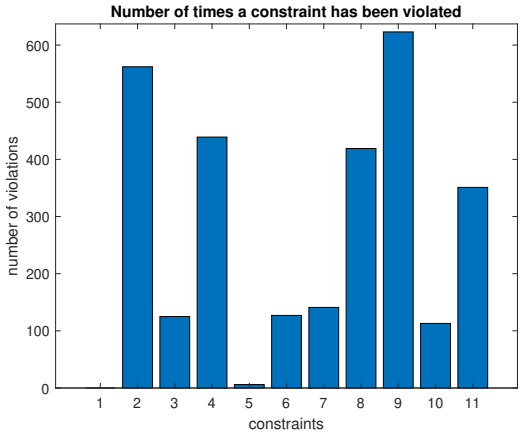



**Figure 8.10:** Number of times a constraint is violated for run 1.1 in third quadrant

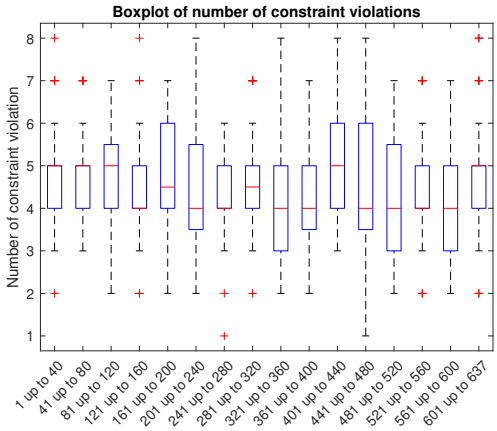



**Figure 8.11:** Boxplot of the constraint violation for run 1.1 in the third quadrant

Run 1.1 in the third quadrant shows similarities when comparing constraint violations between Figure 8.6 and Figure 8.10. For both runs the same constraints are frequently violated. However, there are differences. The run in the third quadrant shows more constraint violations for constraint 2 (cavitation margin mid chord on the pressure side of the blade) and constraint 8 (mean skew balance), while it shows less constraint violations regarding class compliance (4), the cavitational- (8) and separational (11) constraints.
Figure 8.11 shows that the regenerative operation in third quadrant can evaluate propellers that only violate one constraint. There are two propellers that violate one constraint. For these propellers constraint 7 (right skew balance) and constraint 9 (inception bucket constraint) are violated. The regenerative power for the first propeller is $137kW$ at $26.9kN$ drag while it is $144kW$ at $27.4kN$for the second propeller.

### 8.2.3. Run 1.2

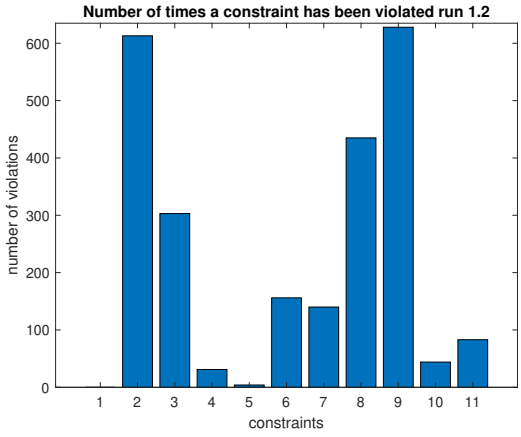



**Figure 8.12:** Number of times a constraint is violated for run 1.2


**Figure 8.13:** Boxplot of the constraint violation for run 1.2

For run 1.2 Figure 8.12 shows that constraint 2, 8 and 9 are often violated. These are:

- Constraint 2: Skew angle
- Constraint 8: Pressure side cavitation margin propulsion
- Constraint 9: Mid chord pressure level propulsion

Figure 8.13 shows that the propulsive operation can yield a propeller that only violates 1 constraint. In total, there are 42 propellers violating two constraints and three propellers that violate 1 constraint. Of the propellers that violate only one constraint, the first propeller violates the skew balance constraint,

the second one violates a pressure level constraint, and the last one violates the cavitation inception bucket constraint.

For the three propellers with one constraint violation the pressure over the propeller blade was evaluated. These evaluations can be seen in Figures 8.15, 8.16 and 8.17. Figure 8.15 shows that the blade is not perfectly balanced. There is more propeller blade area towards the trailing edge, causing a imbalance. This could pose problems regarding the spindle torque (torque needed to keep a CPP blade positioned). The pressures on the other hand stay below all cavitational limits. Figure 8.16 shows a more balanced propeller, but with lower pressures towards the trailing edge of the blade. The main problem can be seen near the hub on both the leading edge as well as on the trailing edge of the blade. On those locations on the propeller blade, the pressure is below the vapour pressure, causing a violation of the pressure margin constraint. Figure 8.17 shows a similar spread of pressures over the propeller blade as in Figure 8.16. The main problem for this propeller can be seen close to the trailing edge, this can also be seen in Figure 8.14.

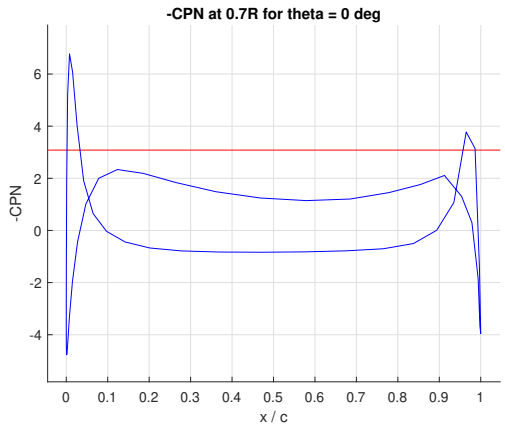



**Figure 8.14:** Pressures over the blade over the chord wise location

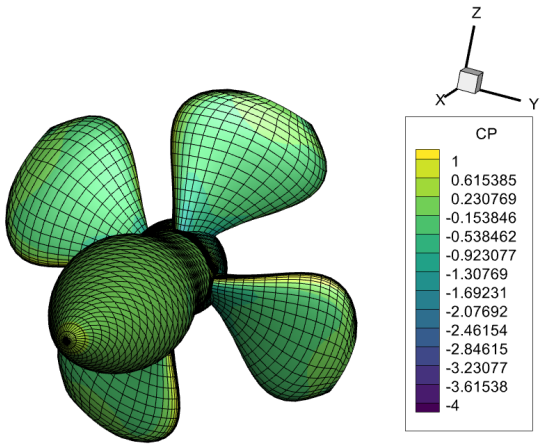



**Figure 8.15:** The local pressures on the propeller in propulsive operation where only the skew constraint was violated

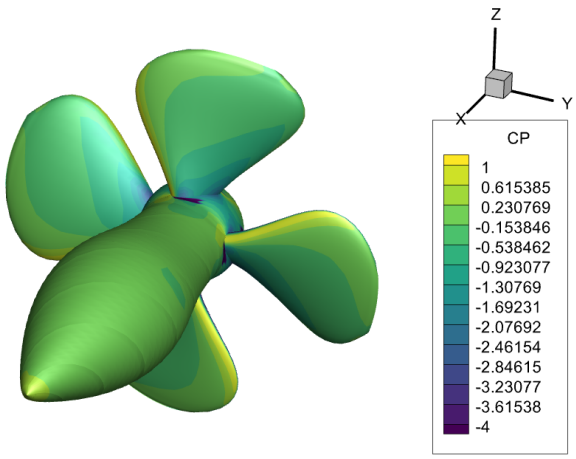



**Figure 8.16:** The local pressures on the propeller in propulsive operation where only the constraint regarding the pressure margin was violated

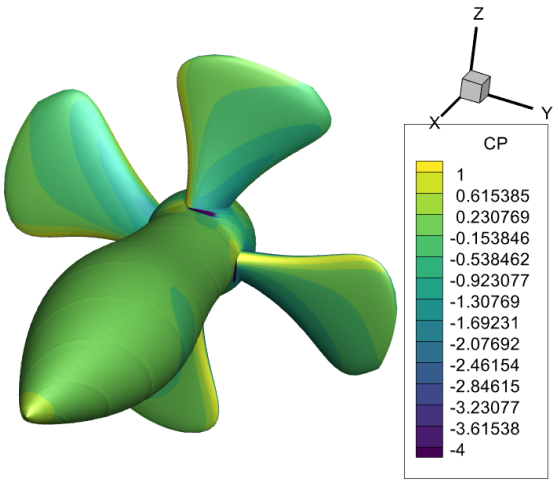



**Figure 8.17:** The local pressures on the propeller in propulsive operation where only the cavitation bucket constraint was violated

### 8.2.4. Run 2

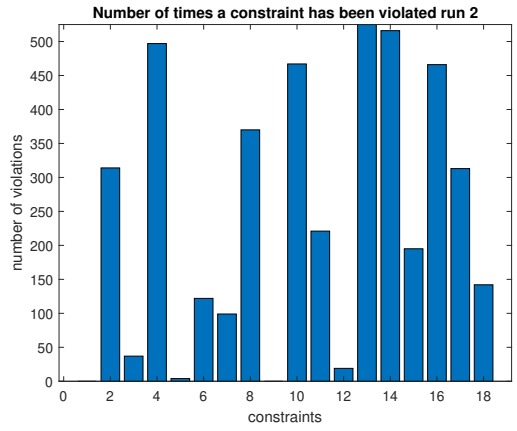



**Figure 8.18:** Number of times a constraint is violated for run 2

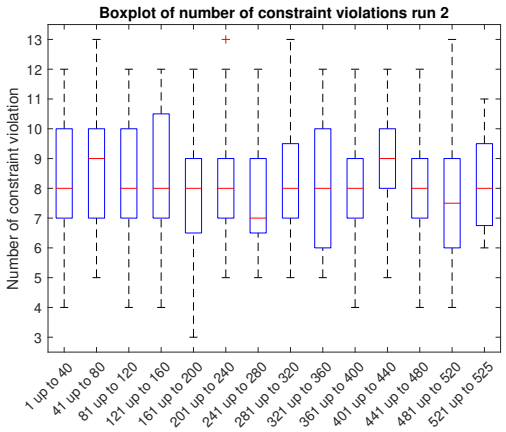



**Figure 8.19:** Boxplot of the constraint violation for run 2

Run 2 is the run where the pitch of the propeller blade can be varied but only within a conventional domain. Figure 8.18 shows that there are multiple constraints that are often violated. Constraint 4,10,13,14 and 16 pose difficulties. These are:

- Constraint 4: Class compliance
- Constraint 10: Pressure side cavitation margin propulsion
- Constraint 13: Positive inception bucket regeneration
- Constraint 14: Positive inception bucket propulsion
- Constraint 16: Pressure side separation limit regeneration

When comparing the constraint violations from run 1.1 and 1.2 with run 2 it can be seen that all constraints that are often violated in run two are also present for the separate designs. Constraint 4, 13 and 16 are the constraints that also pose problems for run 1.1 and constraint 10 poses a problem for run 1.2. It is interesting to see that the constraint violations for skew angle and mid chord pressure level in propulsion are often violated for run 1.2 while these are not often violated in run two (constraint 5 and 11 respectively). When comparing the number of constraint violations in Figure 8.19 it can be seen that there is one propeller with three constraint violations and there are 13 propellers with 4 constraint violations.
Just as for run 1.1 and run 1.2 the constraint violations have been checked without cavitational constraints. For configuration two however, no propellers have been found that do not violate any of the geometrical or separation constraints. Therefore, this layout is not feasible.

### 8.2.5. Run 3

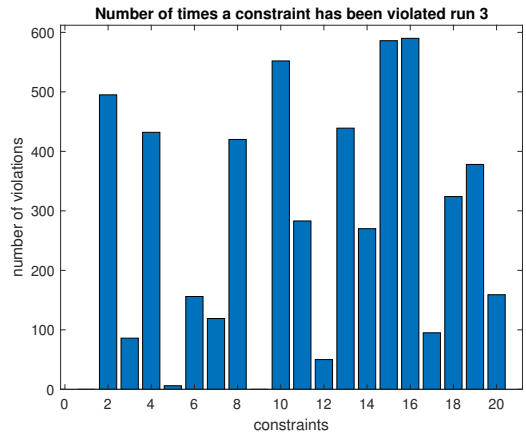



**Figure 8.20:** Number of times a constraint is violated for run 3

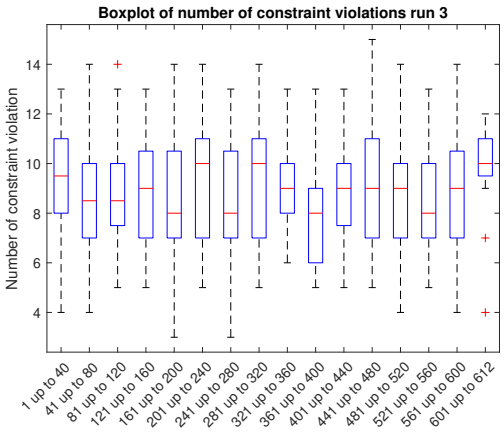



**Figure 8.21:** Boxplot of the constraint violation for run 3

Run three is the run where the propeller blade is rotated 180 degrees. Figure 8.20 shows that this run has the most constraints, of these constraints, several are often violated. These are:

- Constraint 2: Pressure side cavitation margin regeneration
- Constraint 4: Class compliance
- Constraint 8: Skew balance mean skew angle
- Constraint 10: Pressure side cavitation margin propulsion
- Constraint 13: Trailing edge thickness in regeneration
- Constraint 15: Positive inception bucket at regeneration
- Constraint 16: Positive inception bucket at propulsion

It is interesting to see the differences and comparisons in constraint violations with respect to run 2. Similar to run 2, most constraint violations occur regarding cavitation and pressure levels. The main difference is the trailing edge thickness and the mean skew angle. The trailing edge thickness was added since the propeller should satisfy leading edge class rules for both the leading- and the trailing edge. It is notable that the trailing edge in regeneration poses more of a problem than the trailing edge in propulsion. Figure 8.21 shows that the lowest number of constraint violations is three. In total there are two propellers that violate three constraints, while there are 9 propellers that violate 4 constraints.

When neglecting the cavitational constraints, three feasible propellers can be found. These can be seen in figures 8.22, 8.23 and 8.24.

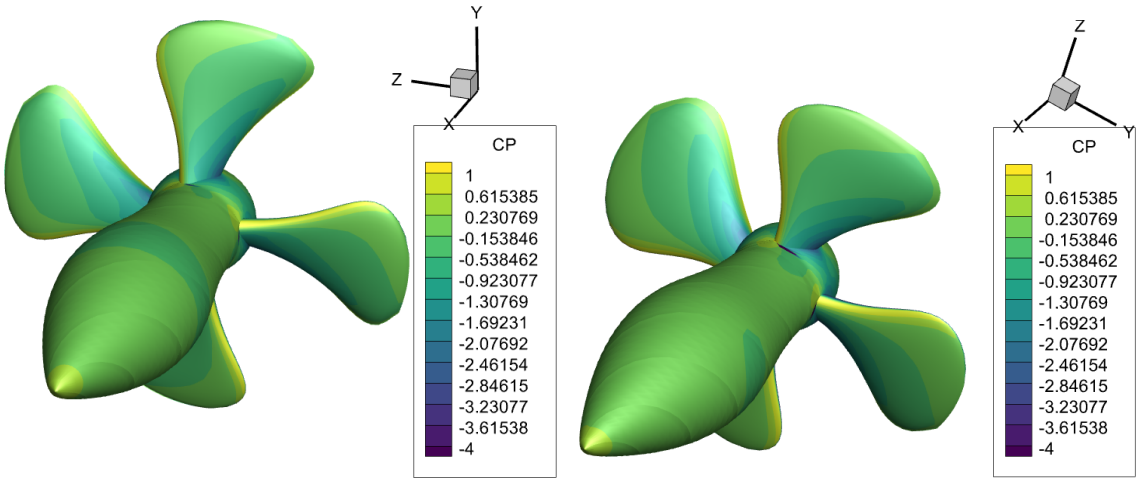



**Figure 8.22:** The local pressures on the propeller for propeller 1 run 3




**Figure 8.23:** The local pressures on the propeller for propeller 2 run 3

Figures 8.22 and 8.23 show similar blade designs. The main differences are the blade area ratio and the skew angles at the blade outline at the leading and trailing edge as a result of the increase in blade area ratio for Figure 8.23. Figure 8.24 shows a propeller with a difference in skew angle when compared to Figure 8.22 and Figure 8.23. The skew angle is oriented backwards for propulsion and forwards for regeneration for the last design.

The three propellers that are considered returned propulsive and regenerative powers, these can be seen in Table 8.1.

| Propeller | Regenerative power (kW) | Propulsive power (kW) |
|---|---|---|
| propeller 1 | 136 kW | 334kW |
| propeller 2 | 140 kW | 301kW |
| propeller 3 | 146.6 kW | 316kW |

**Table 8.1:** Propulsive and regenerative power of the presented propellers for run 3

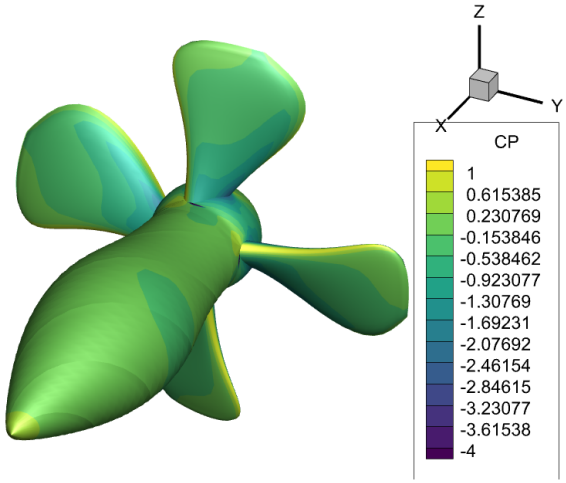



**Figure 8.24:** The local pressures on the propeller for propeller 3 run 3

## 8.3. CMOPSO

Since no feasible propellers emerged, an optimisation for run 3 was done using CMOPSO as optimiser. This was done on Marlcus 5. The run was evaluated on Marclus 5 to check whether origin of the problem is within the definition of the constraints in PropArt. The run considered 101 generations and it did return feasible solutions. The first figure shows that the optimisation finds a front with feasible results.

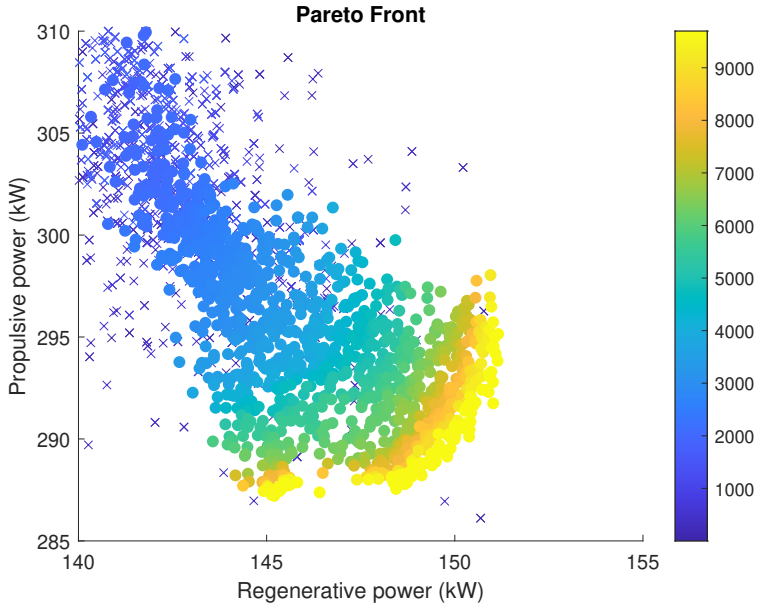



**Figure 8.25:** The Pareto-Front that CMOPSO found for run 3

Figure 8.25 shows a clear trend towards the front once feasible results have been found. Since CMOPSO is able to find feasible results using the same input and structure files, it can be concluded that there are no problems regarding the problem definition.

# 9 Discussion

This research proposes a method for propeller design by surrogate assisted optimisation. Initial benchmark tests have been done to validate the potential of SAMO-COBRA. It was shown that SAMO-COBRA outperforms NSGA-II and CMOPSO for most tests. Therefore, SAMO-COBRA was coupled to PropArt enabling surrogate-assisted propeller optimisation. To test the surrogate, a test case has been considered. The propeller should be optimised for propulsive as well as for regenerative operation. To be able to model an expanding wake, three methods for wake expansion have been proposed. After initial testing, an expanding wake based on the disk theory was shown to be the most versatile. However, the biggest improvement could be made using a wake that has been aligned using the advance ratio. After implementation of these improvements, the test case has been evaluated for three configurations. Using SAMO-COBRA, no feasible solutions have been found for the initial settings. After investigating what constraints were violated most often it turned out that they were the cavitational constraints. When analysing the results that violated the fewest constraints it was shown that the third configuration (fully rotating the blades) is able to generate the highest regenerative power.

## 9.1. Infeasible results

After computing the benchmark tests it was expected that SAMO-COBRA would be able to optimise a complex high dimensional test case. However, it turned out that SAMO-COBRA was not able to find a feasible solution. This raises a question as to what causes this problem. To answer this, a few hypotheses were made:

1. There is a mistake in the problem definition, the test case has no feasible solution
2. There is a problem in the compiled MATLAB version of PropArt on Marclus 4
3. The convergence criterion or the maximum number of iterations for COBYLA is too low, causing a bad RBF fit

### 9.1.1. PropArt model

To test whether there is a mistake in the problem statement in PropArt, an optimisation using CMOPSO was started on Marclus 5. This run did yield feasible results. Therefore, it can be concluded that there are no mistakes in the problem description. This run had to be performed on a computer or cluster with a MATLAB license. This is needed to isolate the problem definition from the compiled version.

### 9.1.2. Compiled PropArt version

The second hypothesis is somewhat more complex. Since the problem is too complex to be evaluated on a normal computer, it was evaluated on the student cluster (computational computer) at MARIN. This cluster does not have a MATLAB license and therefore a so called Runtime (compiled) version of MATLAB was needed to run PropArt. After analysing the cavitational constraint violations in a pressure plot over the chord thickness for multiple propellers it is observed that the trailing edge often violates the constraint. An example can be seen in Figure 9.1. Subsection 3.4.1 explains that a low pressure

at the trailing edge can be caused by a rounded trailing edge. This was checked using Tecplot, and a rounded propeller was found. The propeller can be seen in Figure 9.2.

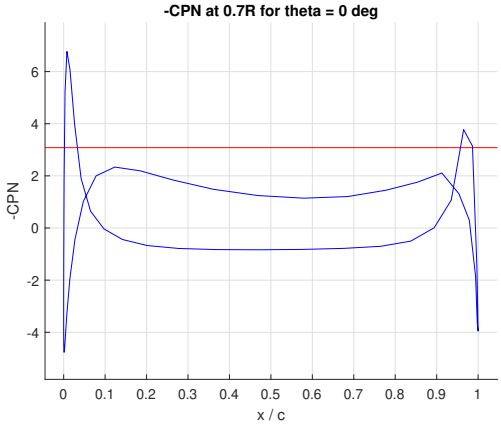



**Figure 9.1:** The pressure over the chord in blue and the cavitation number in red

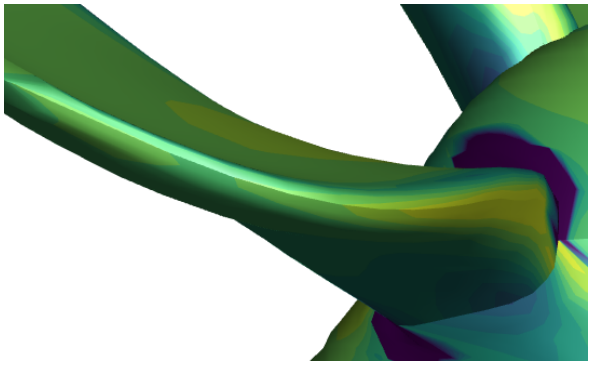


**Figure 9.2:** A blunt trailing edge of the propeller optimised for propulsion (run 1.2)

These results raise the suspicion that a bug was compiled in the Runtime version of PropArt causing a blunt Trailing edge. To check this, 20 propellers that were returned as feasible from the CMOPSO optimisation have been evaluated using the compiled PropArt version on Marclus 4. This resulted in infeasible propellers. One propeller did not pass PropArts equality constraints and the other 19 were not able to pass the cavitational constraints. The most violated constraints were two and 15, which is comparable to the constraint violations that were found using Figure 8.20. Therefore it can be concluded that there is an error in the compiled version of PropArt.

### 9.1.3. COBYLA

The last possibility is an error within the SAMO-COBRA algorithm. The most obvious option would be the maximum number of iterations for COBYLA. Since limited computational expenses are available, SAMO-COBRA provides COBYLA with an evolutionary budget. This budget was set at the value that was proposed by Winter $(50(m+d+k)$ [53], which is the sum of the number of objectives, variables and constraints. This evolutionary budget is lower than the default number of iterations. The number was lowered since SAMO-COBRA needed approximately one day to evaluate one batch, thereby increasing the computational expenses to such a level that the trade-off between computing extra expensive evaluations in a faster but less accurate optimiser versus absolute time is becoming problematic. The chosen settings allowed for a reasonable optimiser time where SAMO-COBRA needs approximately 20 to 30 minutes to determine the next generation. However, when looking at the boxplots (Figures 8.7, 8.13, 8.19 and 8.21) of the constraint violation, the algorithm does not seem to learn from prior generations. Therefore, the question is raised whether the RBF-fit is not accurate enough and therefore not able to predict the location of new samples properly. To test this hypothesis a couple of benchmark tests (BNH, SRN and CEXP) were re-evaluated with a decreased budget for COBYLA. The test cases showed that a slight, but no significant, decrease in achieved hypervolume was visible. However, it was noticable that SAMO-COBRA needed more iterations to find a feasible solution after initial testing. Since these problems are only double constrained it is to be expected that more constraints enlarge these effects and therefore it is likely that trading speed for accuracy can have significant effects on the performance of SAMO-COBRA.

## 9.2. Results

The results from the Pareto-plots (Figures 8.1 and 8.4 ) show that there is no significant difference between a turbine that is specifically designed for regeneration or one that should serve as both. When comparing run two (Figure 8.4) and three (Figure 8.5) it can be seen that a front is starting to appear in the lower right corner for Figure 8.4. Figure 8.5 results in a narrower front. Comparing the objectives after removing the cavitational constraints it is clear that run 3 is able to generate a higher regenerative power, while run 2 is just able to achieve the smallest propulsive power (with a difference of 2kW) while violating multiple constraints. Therefore, it is possible to conclude that the third configuration yields the

best result.

## 9.3. Limitations

When considering the limitations for the use of surrogate methods for propeller optimisation in propulsive and regenerative operations, it can be split into two parts, the limitations regarding the surrogate and the optimisation as well as limitations regarding the use of PROCAL and thus the Boundary Element Method.

### 9.3.1. Surrogates

When running SAMO-COBRA for high-dimensional problems, the computational demand increases with the dimensionality of the problem. To ensure a fast optimisation process the script must be evaluated on a cluster. This reduces the optimisation time as well as the time needed to evaluate the propellers, making it possible to evaluate more propellers. However, this also results in some limitations within the optimisation process. First, only part of the algorithm can be evaluated on multiple cores. This is due to SAMO-COBRA being a python-based optimiser that therefore is only able to run either on one core or in parallel on multiple cores. The problem with computing on multiple cores is that parallel computing starts separate child processes, these processes can not communicate with each other. As a result, only small sections of code can be evaluated using multiple cores. Secondly, the current implementation of SAMO-COBRA only uses the login node and extensive use of cores is therefore not possible.

Additionally, Figure 9.3 shows that the time needed per generation increases exponentially after 30 generations. This can have two causes, of which the first is a lack of working memory. The second cause is a possible memory leakage of the program (if a program allocates memory without actually using it).

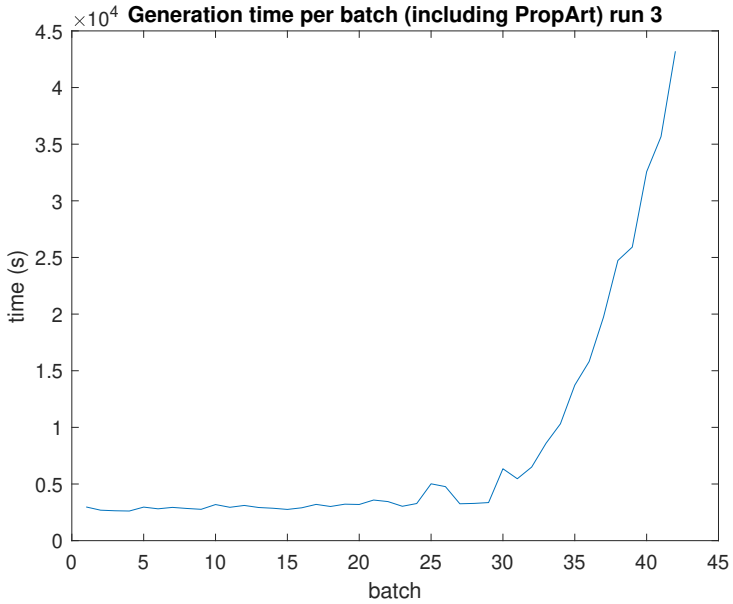



**Figure 9.3:** Runtime of SAMO-COBRA per batch

Another limitation that can be linked to the limitations of the login node of the cluster is the influence of the convergence limits for COBYLA. These have significant influence on the time needed to determine the variables for the next generation. There is a trade-off between speed, computational demand and accuracy and only two can be satisfied. Therefore, currently the accuracy has to be limited such that a reasonable optimisation speed can be satisfied.

### 9.3.2. Boundary Element Methods

There are certain limitations regarding the use of a Boundary Element method. Besides the fact that BEM assumes irrotational and inviscid flow. As a result, empirical models are needed to model cavitation and vortices. Another limitation is the fact that the propeller is modelled using a panelled method. Therefore geometrical compromises have been made while meshing the propeller. After the mesh is finished, the trailing edge is smoothened. For normal propellers this does not pose a problem, but for a propeller with two trailing edges (the leading edge is also used as trailing edge for run 3) this increases the uncertainties.

When zooming in on the propeller wake in regenerative operations, limitations within the changes that have been made become apparent. First, currently the expansion factor for the wake is iterated and evaluated manually, creating an external loop in the design. A solution would be to implement this procedure within PROCAL. Another question that has to be asked for both, the wake expansion as well as the wake alignment, is the validity. Both showed significant improvements for the tests that have been computed. Therefore it is advised to do tests with a wider range of propellers.

## 9.4. Recommendations

To conclude this research, there are multiple recommendations that can be made to improve both the integration of surrogate assisted optimisation methods and the physical modelling of propellers in turbine mode.

### 9.4.1. Model

To start with the improvements for the surrogates, either converting SAMO-COBRA to MATLAB or finding a similar surrogate that is available in MATLAB is suspected to contribute towards a better result. This is due to multiple reasons. Firstly, it would avoid transferring input and results from one coding language to another. Secondly, it would simplify running on the computational nodes of the cluster as well as making it possible to save all results in one structure. Currently, all results are saved in a SAMO-COBRA dict and the variables have to be rescaled and re-evaluated to get anything besides objectives, constraints or variables. Additionally, SAMO-COBRA runs using batch sizes larger than one to avoid the diverging of an entire generation. To be able to reach a feasible solution in the least possible iterations, a batch size of one would be optimal. This is hard to implement in two coding languages, but might be possible if PropArt and SAMO-COBRA would be written in one coding language. Considering the costs that come with the use of MATLAB, Python could be used as well. However, this would imply that PropArt has to be converted from MATLAB to Python

Secondly, when considering the relatively long computational time of SAMO-COBRA with respect to CMOPSO it might be worth trying to combine the two. When analysing the hypervolume progress for the benchmark tests, it can be seen that SAMO-COBRA finds a hypervolume within fewer iterations but after this initial phase the rate of hypervolume progress is often lower than for CMOPSO. Therefore, it is questionable if the use of finding feasible results using SAMO-COBRA and exploiting these results using CMOPSO might result in an even faster algorithm.

### 9.4.2. Test case

Regarding the test case, there are two possible causes for SAMO-COBRA not being able to find a feasible solution that are good candidates for further research. To start, it should be investigated whether there is a possible connection between the number of available iterations for COBYLA and the speed in which SAMO-COBRA can find a Pareto-front. Secondly it should be tested whether there are any problems regarding the sharpening of the trailing edge by the compiled PropArt version. When considering a new optimisation problem, considering multiple test cases with an increasing complexity (an increasing number of objectives, constraints and variables) can provide more insight in the scaling of the computational effort with the number of parameters.

### 9.4.3. Propellers and PROCAL

When considering the hydrodynamics of this thesis, implementation of the proposed models for wake expansion and pitch alignment in turbine mode hold promise to be of use. For both, great improvements were found when comparing PROCAL results to open water diagrams that were obtained after physical tests. However, tests with other propellers still lag. Since a large database of physical tests as well as

computer models of these propellers is available, more testing is recommended.
Additionally, the effect of the Hansen factor for J values other than 1.2 should be investigated. It was shown that the Hansen fraction changes from 0 to 0.75 when J is varied from 1 to 1.6, this has an effect on the wake pitch angle and might therefore influence the open water performance. Since this requires changes in the PROCAL software itself, it was not in the scope of this research to implement this change in PROCAL.
In conclusion, further research should be directed toward the blade spindle torque, for if a blade is rotated to operate in the third quadrant, the blade will encounter high spindle torque when operating in one of these two conditions. Therefore it should be known what effects regeneration in the third quadrant have on the maximum spindle torque and what adjustments can be made to limit the spindle torque with a minimum loss of efficiency.

# 10 Conclusion

This chapter will summarise the findings presented in the research. When looking for a configuration that can be used for both propulsion, as for regeneration. From all investigations there are three possible configurations that are worth researching:

- Two separate designs where one design satisfies the propulsive needs and one satisfies the regenerative demand.
- A controllable pitch propeller that operates in a normal domain (it can rotate up to feathering mode, meaning, the blades will be loaded at a negative angle of attack.
- A controllable pitch propeller where the blades can be rotated 180 degrees such that the blade is loaded from behind in turbine operation. This is operation in the third quadrant.

For the last two designs it is to be expected that a propeller that can rotate to operate in the third quadrant will be the most effective since the blade camber is in the right orientation.
To be able to properly model propellers in turbine operations, some alternatives have been proposed regarding wake expansion (chapter 4) and alignment (chapter 5 ) for the analysis tool PROCAL. For this wake expansion, three methods have been proposed: A case where the wake expansion is assumed to be zero, an empirical for wake expansion on tidal turbines was tested and lastly a method that is based on the disk theory. Tests were computed using an F-series propeller, it turned out that the empirical model and the disk theory performed similarly. However, the disk theory can be used to model all sizes of propellers at a wide range of advance ratios where the empirical model is mainly focused on the development of the wake over the nearfield wake and is fitted for one propeller at a specific J value.
To find a good method to align the pitch of the propeller wake, two methods were tested. First an iterative pitch alignment was tested. Here the dipole orientation and strength was iterated for the nearfield. The second method uses a prescribed wake pitch, derived from the Blade Element Momentum theory. Both methods have been tested for an F-series propeller. A comparison of the open water diagrams resulting from PROCAL and tank tests showed that the prescribed wake pitch performs better than the results obtained from the iterative wake alignment (which is slower too).
To test whether it is feasible to couple a surrogate assisted optimiser (SAMO-COBRA) with Procal, multiple benchmark tests have been computed. Here various analytical test functions with a range in number of variables, objectives, and constraints were used to test SAMO-COBRA against NSGA-II and CMOPSO. The two implemented optimisation algorithms. In the end SAMO-COBRA outperformed NSGA-II and CMOPSO on half of the algorithms. In the analytical tests where SAMO-COBRA did not perform better, different generation sizes of CMOPSO and NSGA-II won. From these tests it is concluded that SAMO-COBRA is a more general optimiser (Since no settings have to be edited). Especially in the first stages of the optimisation process SAMO-COBRA manages to beat NSGA-II and CMOPSO as it finds a hypervolume in fewer iterations.
To test the integration of SAMO-COBRA and PropArt, a test case has been evaluated. The test case tries to optimise the aft propeller of a sailing yacht for propulsive as well as for regenerative operation. Therefore this problem has 30+ variables, between 10 and 20 constraints and multiple objectives. The optimisation process using SAMO-COBRA did not find feasible results. The discussion therefore provides multiple hypotheses to investigate the cause of the problem. The most likely causes are a bug in

the compiled PropArt version on Marclus 4 as well as a problem in COBYLA where the limitations on the maximum number of iterations is too low.

# References


[1] M. Ahmed and N. Qin. “Surrogate-Based Aerodynamic Design Optimization: Use of Surrogates in Aerodynamic Design Optimization”. In: *International Conference on Aerospace Sciences and Aviation Technology* 13.AEROSPACE SCIENCES (2009), pp. 1–26. DOI: 10.21608/asat.2009.23442.

[2] J.D. Anderson. *Fundamentals of Aerodynamics*. 6th ed. McGraw Hill Education. ISBN: 978-1-259-12991-9.

[3] R.F. Anderson. “The Aerodynamic characteristics of airfoils at negative angles of attack”. In: (1932).

[4] J. Bosschers. *Procal v2.0 Theory manual*. MARIN, 2009.

[5] J. Bosschers. “Propeller tip-vortex cavitation and its broadband noise”. English. PhD thesis. Netherlands: University of Twente, Sept. 2018. ISBN: 978-94-92679-52-9. DOI: 10.3990/1.9789492679529.

[6] J. Bosschers. *User’s guide PROCAL version 2.416*. MARIN, 2020.

[7] S. Brizzolara and S. Gaggero. “Silent Propellers with Unconventional Profile Shapes. Examples Obtained with a New Automatic Optimization Method”. In: (Oct. 2009).

[8] J. Carlton. *Marine Propellers and Propulsion*. 2007. 2007. ISBN: 9780750681506.

[9] T. Cebeci, G.J Mosinskis, and A.M.O. Smith. “Calculation of Separation points in Incompressible Turbulent Flows”. In: *J. aircraft* 9.9 (1972).

[10] P Chandrashekarappa and Duvigneau. “Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization”. In: (2007). ISSN: 0249-6399. URL: https://hal.inria.fr/inria-00137602v2.

[11] C. Critzos, H. Heyson, and Jr.R. Boswinkle. “Characteristics of NACA 0012 Airfoil Section at Angles of Attack From 0 Degrees to 180 Degrees,” NACA TN-3361”. In: (1955).

[12] Katia Damborsky. *SDF 2023: Zero fossil fuel sailing yacht Project Zero in build at Vitters*. 2023. URL: https://www.boatinternational.com/yachts/news/sdf-2023-vripack-net-zero-sailing-yacht (visited on 06/23/2023).

[13] K Deb et al. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. In: *IEEE transactions on evolutionary computation, vol 6. no2* (2002).

[14] S.L. Dixon, B. Eng., and C.A. Hall. *Wind Turbines and Aerodynamics Energy Harvesters 2019*. May 2009. 2014, pp. 419–485. ISBN: 9780124159549. DOI: 10.1016/B978-0-12-415954-9.00010-3.

[15] E.J. Foeth. “Propeller optimization using an unsteady Boundary-Element Method”. In: (June 2015).

[16] E.J. Foeth and A. Lampe. “CRS Propagate II: T131 PropArt User Guide V1.605”. In: (Oct. 2020).

[17] S. Gaggero et al. “A Design by Optimization of Tip Loaded Propellers”. In: (June 2015).

[18] Anthony Giunta, Steven Wojtkiewicz, and Michael Eldred. “Overview of modern design of experiments methods for computational simulations”. In: January (2003), pp. 1–17.

[19] Tushar Goel et al. “Ensemble of surrogates”. In: *Structural and Multidisciplinary Optimization* 33.3 (2007), pp. 199–216. ISSN: 1615147X. DOI: 10.1007/s00158-006-0051-9.

[20] P. Greaves. “Design of offshore wind turbine blades”. In: *Offshore Wind Farms: Technologies, Design and Operation* (Mar. 2016), pp. 105–135. DOI: 10.1016/B978-0-08-100779-2.00006-4.

[21] Ship motion group. *Groundbreaking innovative (hybrid) Controllable Pitch Propulsion system*. URL: https://shipmotiongroup.com/sailyacht/controllable-pitch-propulsion/ (visited on 01/06/2023).

[22] Snorri Gudmundsson. *The Anatomy of the Airfoil*. 2014, pp. 235–297. ISBN: 9780123973085. DOI: 10.1016/b978-0-12-397308-5.00008-8.

[23] H. M. Gutmann. “A Radial Basis Function Method for Global Optimization”. In: *Journal of Global Optimization* 19.3 (2001), pp. 201–227. ISSN: 09255001. DOI: 10.1023/A:1011255519438.

[24] M.O.L Hansen. *Aerodynamics of Wind Turbines*. 2nd ed. earhtscan, 2008. ISBN: 978-1-84407-438-9.

[25] A.N. Hayati, S.M. Hashemi, and M. Shams. “A study on the effect of the rake angle on the performance of marine propellers”. In: (2011). DOI: 10.1177/0954406211418588. URL: https://doi.org/10.1177/0954406211418588.

[26] E.L. Houghton et al. *Aerodynamics for engineering students, seventh edition*. Buttleworth Heinemann, 2017. DOI: https://doi.org/10.1016/B978-0-08-100194-3.00006-7.

[27] S. Huband et al. “A review of multiobjective test problems and a scalable test problem toolkit”. In: *IEEE Transactions on Evolutionary Computation* 10.5 (2006), pp. 477–506. DOI: 10.1109/TEVC.2005.861417.

[28] J. Kerwin. *13.04 LECTURE NOTES HYDROFOILS AND PROPELLERS*. Jan. 2001.

[29] H. Klein Woud and D. Stapersma. *Design of Propulsion and Electric Power Generation Systems*. Imarest, 2015. ISBN: 1902536479.

[30] H.J. Koning Gans, de. “Introduction of Numerical Methods in Ship Hydromechanics”. In: (2012).

[31] G. Kuiper. *Resistance and Propulsion 1 Technical University Delft*. 2002.

[32] P.K. Kundu, I.M. Cohen, and D.R. Dowling. *Fluid Mechanics*. Elsevier, 2015. ISBN: 9780124059351.

[33] S. Laín, L. T. Contreras, and O. López. “A review on computational fluid dynamics modeling and simulation of horizontal axis hydrokinetic turbines”. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 41.9 (2019), pp. 1–24. ISSN: 18063691. DOI: 10.1007/s40430-019-1877-6. URL: https://doi.org/10.1007/s40430-019-1877-6.

[34] R. Lanzafame and M. Messina. “Fluid dynamics wind turbine design: Critical analysis, optimization and application of BEM theory”. In: *Renewable Energy* 32.14 (Nov. 2007), pp. 2291–2305. ISSN: 09601481. DOI: 10.1016/J.RENENE.2006.12.010.

[35] Stephen Leary, Atul Bhaskar, and Andy Keane. “Optimal orthogonal-array-based latin hypercubes”. In: *Journal of Applied Statistics* 30.5 (2003), pp. 585–598. ISSN: 02664763. DOI: 10.1080/0266476032000053691.

[36] Li Li et al. “On the estimation of pareto front and dimensional similarity in many-objective evolutionary algorithm”. In: *Information Sciences* 563 (2021), pp. 375–400. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2021.03.008. URL: https://www.sciencedirect.com/science/article/pii/S0020025521002450.

[37] Zhongwei Ma and Yong Wang. “Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons”. In: *IEEE Transactions on Evolutionary Computation* 23.6 (2019), pp. 972–986. DOI: 10.1109/TEVC.2019.2896967.

[38] J.B. Marchand, J.E. Astolfi, and P. Bot. “Discontinuity of lift on a hydrofoil in reversed flow for tidal turbine application”. In: *European Journal of Mechanics, B/Fluids* 63 (2017), pp. 90–99. ISSN: 09977546. DOI: 10.1016/j.euromechflu.2017.01.016. URL: http://dx.doi.org/10.1016/j.euromechflu.2017.01.016.

[39] Mathworks. *boxplot*. URL: https://nl.mathworks.com/help/stats/boxplot.html (visited on 06/26/2023).

[40] A. Mola et al. “Efficient reduction in shape parameter space dimension for ship propeller blade design”. In: (2019). DOI: https://doi.org/10.48550/arXiv.1905.09815.

[41] L. E. Myers and A. S. Bahaj. “Experimental analysis of the flow field around horizontal axis tidal turbines by use of scale mesh disk rotor simulators”. In: *Ocean Engineering* 37.2-3 (Feb. 2010), pp. 218–227. ISSN: 00298018. DOI: 10.1016/J.OCEANENG.2009.11.004.

[42] M. Nachtane et al. “Design and Hydrodynamic Performance of a Horizontal Axis Hydrokinetic Turbine”. In: 16.2 (2019), pp. 6453–6469.

[43] M. J. D. Powell. “A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation”. In: *Advances in Optimization and Numerical Analysis*. Ed. by Susana Gomez and Jean-Pierre Hennart. Dordrecht: Springer Netherlands, 1994, pp. 51–67. ISBN: 978-94-015-8330-5. DOI: 10.1007/978-94-015-8330-5_4. URL: https://doi.org/10.1007/978-94-015-8330-5_4.

[44] Néstor V. Queipo, Carlos J. Arévalo, and Salvador Pintos. “The integration of design of experiments, surrogate modeling and optimization for thermoscience research”. In: *Engineering with Computers* 20.4 (2005), pp. 309–315. ISSN: 01770667. DOI: 10.1007/s00366-004-0299-x.

[45] T. Rendall and C. Allen. “Evaluation of Radial Basis Functions for CFD Volume Data Interpolation”. In: (2010). DOI: 10.2514/6.2010-1414.

[46] F. Salvatore, Z. Sarichloo, and D. Calcagni. “Marine Turbine Hydrodynamics buy a Boundary Element Method with Viscous Flow Correction”. In: *Marine Science and Engineering* (2018).

[47] M.J. Smith, N.D. Liggett, and B.C.G. Koukol. “Aerodynamics of airfoils at high and reverse angles of attack”. In: *journal of aircraft* (2011). DOI: 10.2514/1.C031428.

[48] Xinzi Tang et al. “A direct approach of design optimization for small horizontal axis wind turbine blades”. In: *Procedia CIRP* 36 (2015), pp. 12–16. ISSN: 22128271. DOI: 10.1016/J.PROCIR.2015.01.047.

[49] D. Uşar and S. Bal. “Cavitation simulation on horizontal axis marine current turbines”. In: *Renewable Energy* 80 (Aug. 2015), pp. 15–25. ISSN: 18790682. DOI: 10.1016/J.RENENE.2015.01.060.

[50] G. Vaz and J. Bosschers. “Modelling three dimensional sheet cavitation on marine propellers using a boundary element method”. In: (2006).

[51] Ross Vennell. “Exceeding the Betz limit with tidal turbines”. In: *Renewable Energy* 55 (2013), pp. 277–285. ISSN: 09601481. DOI: 10.1016/j.renene.2012.12.016. URL: http://dx.doi.org/10.1016/j.renene.2012.12.016.

[52] Y. Wang, M. Abdel-Maksoud, and B. Song. “A fast method to realize the pressure Kutta condition in Boundary Element Method for lifting bodies”. In: *Ocean engineering* 130 (2017). DOI: https://doi.org/10.1016/j.oceaneng.2016.12.009.

[53] R. Winter, de, B. Stein, van, and T Bäck. “SAMO-COBRA: A fast surrogate assisted constrained multiple-objective optimization algorithm”. In: (2021). DOI: https://doi.org/10.1007/978-3-030-72062-9_22.

[54] *Multi-Point acquisition functionf for constraint parallel efficient multi-objective optimization*. English. July 2022.

[55] David C. Woods and Susan M. Lewis. “Design of experiments for screening”. In: *Handbook of Uncertainty Quantification* (2017), pp. 1143–1185. DOI: 10.1007/978-3-319-12385-1_33. arXiv: 1510.05248.

[56] Raul Yondo, Esther Andrés, and Eusebio Valero. “A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses”. In: *Progress in Aerospace Sciences* 96.December 2017 (2018), pp. 23–61. ISSN: 03760421. DOI: 10.1016/j.paerosci.2017.11.003.

[57] M. Ishak Yuce and Abdullah Muratoglu. “Hydrokinetic energy conversion systems: A technology status review”. In: *Renewable and Sustainable Energy Reviews* 43 (2015), pp. 72–82. ISSN: 13640321. DOI: 10.1016/J.RSER.2014.10.037.

[58] X. Zhang et al. “A competitive mechanism based multi-objective particle swarm optimizer with fast convergence”. In: *Information sciences, volume 427* (2018), pp. 63–76. DOI: https://doi.org/10.1016/j.ins.2017.10.037.

# A

# Test functions

## A.1. MW2

$$\text{min } f_1(\mathbf{x}) = x_1 \tag{A.1}$$

$$minf_2(\mathbf{x}) = g_1(1 - 0.85f_1/g_1) \tag{A.2}$$
$$s.t.c(\mathbf{x}) = 1 - f_1 - f_2 + 0.5sin(2\pi l)^8 \geq 0 \tag{A.3}$$
$$l = \sqrt{2}f_2 - \sqrt{2}f_1 \tag{A.4}$$

[37]

## A.2. MW3

$$minf_1(\mathbf{x}) = x_1 \tag{A.5}$$
$$\text{min } f_2(\mathbf{x}) = g_1(1 - f_1/g_1) \tag{A.6}$$
$$\text{s.t. } c(\mathbf{x}) = 1 - f_1 - f_2 + 0.5sin(3\pi l)^8 \geq 0 \tag{A.7}$$
$$l = \sqrt{2}f_2 - \sqrt{2}f_1 \tag{A.8}$$

[37]

## A.3. C3DTLZ4

$$g(\mathbf{x}) = \Sigma_i^n (x_i - 0.5)^2) \tag{A.9}$$
$$f_1(\mathbf{x}) = (1 + (g(\mathbf{x})))cos(x_1 * \pi/2) \tag{A.10}$$
$$f_2(\mathbf{x}) = (1 + (g(\mathbf{x}))) * sin(x_1 * \pi/2) \tag{A.11}$$
$$c_1(\mathbf{x}) = (f_1(\mathbf{x})^2)/4 + f_2(\mathbf{x})^2 - 1 \tag{A.12}$$
$$c_2(\mathbf{x}) = (f_2(\mathbf{x})^2)/4 + f_1(\mathbf{x})^2 - 1 \tag{A.13}$$

$$0 \leq x_i \leq 1 \qquad i = 1, 2, n \tag{A.14}$$

## A.4. BNH

$$\text{min } f_1(\mathbf{x}) = 4x_1^2 + 4x_2^2 \tag{A.15}$$
$$\text{min } f_2(\mathbf{x}) = (x_1 - 5)^2 + (x_2 - 5)^2 \tag{A.16}$$
$$\text{subject to } C_1(\mathbf{x}) = (x_1 - 5)^2 + x_2^2 \leq 25, \tag{A.17}$$
$$\text{subject to } C_2(\mathbf{x}) = (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7 \tag{A.18}$$
$$0 \leq x_1 \leq 5 \qquad 0 \leq x_2 \leq 3 \tag{A.19}$$

## A.5. SRN

$$f_1 = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \tag{A.20}$$

$$f_2 = 9x_1 - (x_2 - 1)^2 \tag{A.21}$$

$$c_1 = x_1^2 + x_2^2 - 225 \tag{A.22}$$

$$c_2 = x_1 - 3x_2 + 10 \tag{A.23}$$

$$-20 \leq x_1 \leq 20 \qquad -20 \leq x_2 \leq 20 \tag{A.24}$$

## A.6. CEXP

$$f_1 = x_1 \tag{A.25}$$

$$f_2 = (1 + x_2)/x_1 \tag{A.26}$$

$$g_1 = x_2 + x_1 - 6 \tag{A.27}$$

$$g_2 = -x_2 + 9x_1 - 1 \tag{A.28}$$

$$0.1 \leq x_1 \leq 1 \qquad 5 \leq x_2 \leq 9 \tag{A.29}$$

## A.7. DTLZ2

$$\text{Min. } f_1(\mathbf{x}) = (1 + g(x_M))cos(x_1\pi/2) \cdots cos(x_{M-2}\pi/2)cos(x_{M-1}\pi/2) \tag{A.30}$$

$$\text{Min. } f_2(\mathbf{x}) = (1 + g(x_M))cos(x_1\pi/2) \cdot cos(x_{M-2}\pi/2)sin(x_{M-1}\pi/2) \tag{A.31}$$

$$\text{Min. } f_M(\mathbf{x}) = (1 + g(x_M))cos(x_1\pi/2) \cdot sin(x_{M-2}\pi/2) \tag{A.32}$$

$$\textbf{with } g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \tag{A.33}$$

$$0 \leq x_i \leq 1 \qquad \text{for } i = 1, 2, n \tag{A.34}$$

[27]

## A.8. WFG1

$$\text{Min. } f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})) \tag{A.35}$$

with

$$f_1(\mathbf{x}) = 0.5(x_1^2 + x_2^2) + sin(x_1^2 + x_2^2) \tag{A.36}$$

$$f_2(\mathbf{x}) = \frac{(3x_1 + 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)}{27} + 15 \tag{A.37}$$

$$f_3(\mathbf{x}) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1exp(-(x_1 + x_2^2) \tag{A.38}$$

$$\text{with box constraints: } -3 \leq x_1, x_2 \leq 3 \tag{A.39}$$

[27]

# B

# Variables for the test case

| Variable | upper limit | lower limit | seperate turbine | separate propulsor | normal CPP | reversible CPP |
|---|---|---|---|---|---|---|
| pitch variable 1 | 0.55 | 0.85 | x | x | x | x |
| pitch variable 2 | 0.5 | 2 | x | x | x | x |
| pitch variable 3 | 0.5 | 2 | x | x | x | x |
| camber x-dir 1 | 0.3 | 0.7 | x | x | x | x |
| camber x-dir 2 | 0.8 | 1 | x | x | x | x |
| camber y-dir 1 | 0 | 0.1 | x | x | x | x |
| camber y-dir 2 | 0 | 0.1 | x | x | x | x |
| camber y-dir 3 | 0 | 0.1 | x | x | x | x |
| camber y-dir 4 | 0 | 0.02 | x | x | x | x |
| chord $LE_x$ | 0.5 | 1 | x | x | x | x |
| chord $LE_y$ | -0.4 | -0.025 | x | x | x | x |
| chord $TE_x$ | 0.5 | 1 | x | x | x | x |
| chord $TE_y$ | 0.025 | 0.4 | x | x | x | x |
| chord tip radius | 0.05 | 1.2 | x | x | x | x |
| chord tip skew | -0.1 | 0.1 | x | x | x | x |
| rake x-dir | -0.5 | 0.85 | x | x | x | x |
| rake y-dir 1 | -0.05 | 0.05 | x | x | x | x |
| rake y-dir 2 | -0.05 | 0.05 | x | x | x | x |
| thickness x-dir 1 | 0.6 | 0.8 | x | x | x | x |
| thickness y-dir 1 | 0.035 | 0.055 | x | x | x | x |
| thickness y-dir 2 | 0 | 0.02 | x | x | x | x |
| thickness y-dir 3 | 0.005 | 0.015 | x | x | x | x |
| sections camber x2 | 0.1 | 0.5 | x | x | x | x |
| sections camber x3 | 0.5 | 0.9 | x | x | x | x |
| sections camber y1 | 0 | 0.8 | x | x | x | x |
| sections camber y2 | 0 | 0.8 | x | x | x | x |
| sections thickness LE | 0.1 | 1.5 | x | x | x | x |
| sections thickness x1 | 0 | 0.4 | x | x | x | x |
| sections thickness x2 | 0.05 | 0.5 | x | x | x | x |
| sections thickness x3 | 0.5 | 0.95 | x | x | x | x |
| sections thickness x4 | 0.6 | 1 | x | x | x | x |
| sections thickness y1 | 0.7 | 1 | x | x | x | x |
| sections thickness y2 | 0.7 | 1 | x | x | x | x |
| sections thickness TE | 0.1 | 1.5 | x | x | x | x |
| rps regeneration | 3.68 | 5.52 | x | | x | x |
| rps propulsion | 4.6 | 6.90 | | x | x | x |

# C

# Constraints for the test case

| Constraint | separate turbine | separate propulsor | negative angle-of-attack | third quadrant |
|---|---|---|---|---|
| mid chord cavitation margin regen | x | | x | x |
| Pressure side cavitation margin regen | x | | x | x |
| Pressure margin mid chord regen | x | | x | x |
| Class compliance | x | x | x | x |
| Skew angle | x | x | x | x |
| Skew balance left | x | x | x | x |
| Skew balance right | x | x | x | x |
| Skew balance mean skew | x | x | x | x |
| Mid chord cavitation margin propul | | x | x | x |
| Pressure side cavitation margin propul | | x | x | x |
| Mid chord pressure level propul | | x | x | x |
| mean circulation hub | x | x | x | x |
| Trailing edge thickness regeneration radii 1 | | | | x |
| Trailing edge thickness propulsion radii 1 | | | | x |
| Positive inception bucket at regeneration | x | | x | x |
| Positive inception bucket at propulsion | | x | x | x |
| Suction side separation limit regeneration | x | | x | x |
| Pressure side separation limit regeneration | x | | x | x |
| Suction side separation limit propulsion | | x | x | x |
| Pressure side separation limit propulsion | | x | x | x |