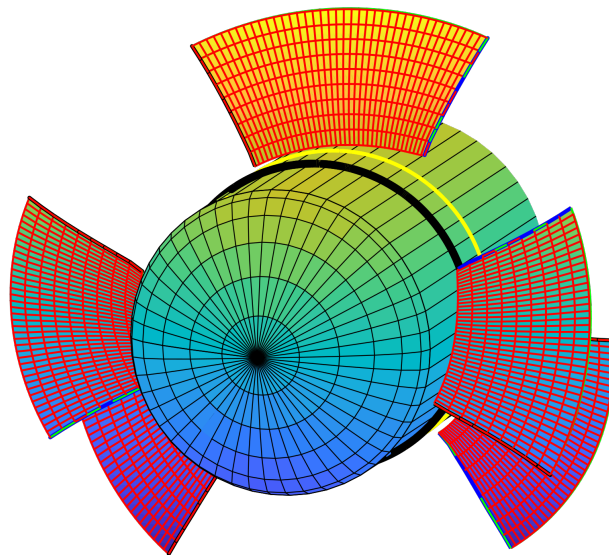


Pål Dahle

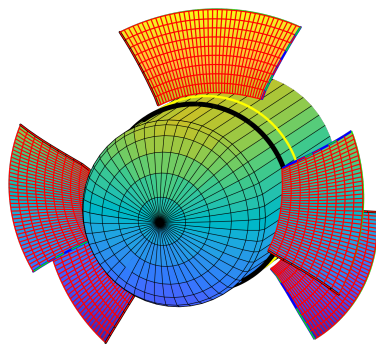
Pump turbines in existing powerplants

Master's thesis in Mechanical Engineering
Supervisor: Pål-Tore Selbo Storli
June 2019



Pål Dahle

Pump turbines in existing powerplants



Master's thesis in Mechanical Engineering
Supervisor: Pål-Tore Selbo Storli
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

 **NTNU**
Norwegian University of
Science and Technology

MASTER WORK

for

student Pål Dahle

Spring 2019

Pump turbines in existing powerplants
Pumpeturbiner i eksisterende kraftverk

Background

Norway has 50% of the European hydro reservoir energy storage, and many of these sites are highly suitable for retrofitting of pump-storage capabilities. To be able to reuse existing power plants by retrofitting with Reversible Pump Turbines (RPTs), the problem of cavitation in pumping mode must be solved. This has been proposed solved by the use of an axial booster pump in front of the RPT. An axial pump for the use as such booster pump for the use at Roskrepp power plant must be designed. Preliminary findings suggest that for the needed performance of a booster pump at Roskrepp, conventional single stage axial machines will have very low efficiencies. Other design solutions are available and must be investigated to determine if they are applicable at Roskrepp.

Objective

The objective of the master degree work is to investigate if a two-stage counter-rotating axial machine can be used to provide the needed performance, and find a preliminary design for such a machine.

The following tasks are to be considered:

1. Literature search regarding design of axial hydraulic machines
2. Develop a user-friendly way of creating counter-rotating propeller designs in Matlab
3. Create new a design code (Possibly extend the open code OpenProp, Matlab-based) for the design of hubless rim-driven thrusters.
4. Obtain a design with acceptable performance for Roskrepp power plant.
5. If the student will go to Nepal for an excursion, earlier and further work will be presented as a publication and presented at the conference; 9th *International symposium on Current Research in Hydropower Technologies (CRHT-IX)* at Kathmandu University

-- “ --

The master work comprises 30 ECTS credits.

The work shall be edited as a scientific report, including a table of contents, a summary in Norwegian, conclusion, an index of literature etc. When writing the report, the candidate must emphasise a clearly arranged and well-written text. To facilitate the reading of the report, it is important that references for corresponding text, tables and figures are clearly stated both places.

By the evaluation of the work the following will be greatly emphasised: The results should be thoroughly treated, presented in clearly arranged tables and/or graphics and discussed in detail.

The candidate is responsible for keeping contact with the subject teacher and teaching supervisors.

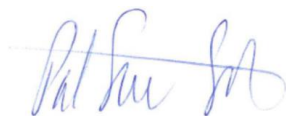
Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report. If the documentation on risk assessment represents a large number of pages, the full version is to be submitted electronically to the supervisor and an excerpt is included in the report.

According to "Utfyllende regler til studieforskriften for teknologistudiet/sivilingeniørstudiet ved NTNU" § 20, the Department of Energy and Process Engineering reserves all rights to use the results and data for lectures, research and future publications.

Submission deadline: 11 June 2019.

- Work to be done in lab (Water power lab, Fluids engineering lab, Thermal engineering lab)
 Field work

Department for Energy and Process Engineering, 10/1 2019



Pål-Tore Storli
Supervisor

Co-Supervisor(s):

ABSTRACT

A user-friendly method of designing contra-rotating axial flow pumps have been developed in Matlab. Graphical user interfaces have been developed for inputting design parameters and viewing the design results. The design method utilizes existing lifting-line design theory for the optimization of a set of contra-rotating propellers but is modified to facilitate axial pump design.

Replacing the current Francis runner with a reversible pump turbine in the same location is considered as a cost-saving way of retrofitting Roskrepp hydropower plant with pumping capacity. Preliminary simulations estimate a pressure head deficit when operating the reversible pump turbine in pump mode. A contra-rotating axial booster pump for pumping in series with the reversible pump turbine is proposed as a way of making up this head deficit. Therefore, a preliminary design of a booster pump was created using the developed design method. The booster pump design was only able to deliver part of the head deficit, but it is argued that the booster pump's head requirement is set unrealistically high. The attained booster pump head may therefore be sufficient, but an assessment of the real head deficit of the reversible pump turbine must be conducted to know for sure. The proposed pump design showed a suboptimal hydraulic efficiency, however, the total hydraulic efficiency of the booster pump and reversible pump turbine pumping in series is considered acceptable. Swirl velocities at the aft impeller outlet is believed to be the cause of the low booster pump efficiency.

Differences in propeller and axial pump optimization restricts the application of the created design method to high-efficiency pumps, i.e. pumps with no outlet swirl. The obtained design's attainable head is therefore assumed to be overpredicted. High-efficiency designs are believed to achieve more accurate results with the developed method. Further assessment of the design of both the reversible pump turbine and the axial booster pump is needed before deciding on a final booster pump design.

SAMMENDRAG

En brukervennlig metode for å designe kontraroterende aksialpumper har blitt utviklet ved bruk av Matlab. Grafiske brukergrensesnitt har blitt laget for å skrive inn designparametre og å vise designresultater. Designmetoden benytter seg av eksisterende løftelinjeteori for design og optimalisering av kontraroterende propeller, men den er modifisert for å muliggjøre aksialpumpedesign.

Å erstatte det nåværende Francis-løpehjulet med en reversibel pumpeturbin på samme sted blir vurdert som en besparende måte å ettermontere pumpekapasitet på Roskrepp vannkraftverk. Innledende simuleringer estimerer at den reversible pumpeturbinen vil levere for lav løftehøyde når den opererer i pumpemodus. En kontraroterende aksial boosterpumpe til å pumpe i serie med den reversible pumpeturbinen er foreslått for å gjøre opp for den manglende løftehøyden. Et førstedesign av en boosterpumpe er derfor laget ved bruk av den utviklede designmetoden. Boosterpumpedesignet klarte bare å levere en del av den nødvendige løftehøyden, men det argumenteres for at løftehøydekravet til booster-pumpen er valgt urealistisk høyt. Trykkhøyden levert av førstedesignet er derfor muligens tilstrekkelig, men en evaluering av den reelle løftehøydemangelen til den reversible pumpeturbinen må gjennomføres for å vite dette sikkert. Det foreslåtte førstedesignet har en ikke-optimal hydraulisk virkningsgrad, men den totale hydrauliske virkningsgraden for booster-pumpen og den reversible pumpeturbinen i serie er derimot regnet som akseptabel. Rotasjon i strømmingen ved utløpet av den bakerste impelleren er anslått å være årsaken til den lave virkningsgraden til booster-pumpen.

Forskjeller i propell- og pumpeoptimalisering begrenser bruken av den utviklede designmetoden til pumper med høy virkningsgrad, altså pumper med lite utløpsrotasjon. Den oppnåelige løftehøyden til førstedesignet av booster-pumpen antas derfor å være overestimert. Design med høye virkningsgrader antas å være mer nøyaktige med den utviklede designmetoden. Designet av både den reversible pumpeturbinen og booster-pumpen må evalueres på nytt før et endelig boosterpumpedesign kan bestemmes.

ACKNOWLEDGEMENTS

I wish to thank my supervisor Pål-Tore Selbo Storli for his guidance throughout this project and for his part in our paper for the CRHT-IX conference in Nepal.

I would also like to thank all the students at the Waterpower Laboratory for a year filled with laughter, interesting discussions and an amazing trip to visit Kathmandu University in Nepal. A special thank you to my fellow student Rune Haugen Larsen for countless discussions about our projects and for continuously sparking my interest in our work.

Pål Dahle
Trondheim, Juni 2019

CONTENTS

Abstract	i
Sammendrag	ii
Acknowledgements	iii
Table of Contents	v
List of Tables	vi
List of Figures	viii
Nomenclature	ix
Abbreviations	xii
1 Introduction	1
1.1 Background and motivation	1
1.1.1 Case power plant	3
1.2 Objective	3
2 Literature review regarding axial hydraulic machine design	4
2.1 Historical development	4
2.2 Modern axial hydraulic machines	7
3 Basic Theory	9
3.1 Pumped hydroelectric storage	9
3.2 External flow	10
3.2.1 Drag	10
3.2.2 Lift	10
3.2.3 Wing theory	11
3.3 Cavitation	13
3.3.1 Cavitation in an axial flow pump	13

3.4	Axial hydraulic machinery	15
3.4.1	Axial flow pumps	15
4	Contra-rotating axial pump design method	18
4.1	Lifting line theory	18
4.2	Wake model	20
4.3	Optimization	22
4.4	CRP modifications	23
4.5	Design inputs	25
5	Method	27
5.1	Facilitating user-friendly CRP designs in Matlab	27
5.1.1	Design input	27
5.1.2	Displaying design results	29
5.2	Design of booster pump for application at Roskrepp	30
5.2.1	Determining design inputs	30
5.2.2	Design Iterations	38
6	Results and Discussion	39
6.1	Preliminary booster pump design	39
6.2	Relationship between propellers and pumps	44
6.3	Booster pump configuration at Roskrepp hydropower plant	46
7	Conclusions	48
8	Further work	50
	Bibliography	51
	Appendix	54
A	Preliminary booster pump design geometry	55
B	Explanation of graphical user interfaces - Coupled CRP design method	57
B.1	GUI for inputting design parameters	58
B.2	GUI for viewing optimization results	59
C	Matlab code for design input GUI - Coupled CRP design method	60
D	Matlab code for results GUI - Coupled CRP design method	83
E	Derivatives of auxiliary function for optimum circulation calculations	94
F	Research paper for CRHT-IX conference at Kathmandu University in Nepal	96

LIST OF TABLES

3.1	Empirically derived values for parameters a and b for calculation of $NPSH_R$. Collected from [1].	14
4.1	Input parameters for CRP design using the coupled method.	26
6.1	Efficiency of booster pump and thrust and torque coefficients for each stage. Results obtained by the coupled CRP design optimization.	39

LIST OF FIGURES

1.1 RPT configuration with booster pump to increase inlet pressure of RPT in pump mode.	2
1.2 Intended placement of contra-rotating axial booster pump at Roskrepp hydropower plant.	3
2.1 Leonardo da Vinci's "Aerial Screw" helicopter. Image from Ligett [2]. . .	4
2.2 Illustration of Froude's blade element theory.	5
2.3 Illustration of Prandtl's lifting line theory. A set of circulation vortices is shed from the wing along the flow direction.	6
2.4 Water jet propulsor.	8
3.1 Foil section displaying foil terminology.	11
3.2 Lift coefficient to drag coefficient ratio plotted versus angle of attack for the NACA 66-209 foil. Plot data extracted from http://airfoiltools.com/airfoil/details?airfoil=naca66209-il	12
3.3 Rake and skew of a propeller blade (wing) illustrated.	12
3.4 Suction head of a pumping application.	14
4.1 Lifting line representation of a propeller with three blades in a constant axial inflow. Trailing vortices are shed from the blades at a constant wake pitch angle.	18
4.2 Velocity and force diagram of a blade section at an arbitrary radius r from the axis of rotation.	19
4.3 Present wake model illustrated for arbitrary control panels $r_c(n - 1)$ and $r_c(n)$ along a lifting line.	21
4.4 Flowchart of coupled CRP design optimization. Method implemented by Laskos [3].	24
5.1 Graphical user interface for inputting CRP design parameters.	28
5.2 Graphical user interface for displaying, plotting and saving design results.	29
5.3 Selected CRP design foil section with NACA 65A010 thickness distribution and NACA $a = 0.8$ meanline.	31

5.4	Diagram for selection of hub ratio. Reproduced from [4]. Here, the dimensional specific speed is calculated with imperial units as $N_s = \frac{n\sqrt{Q}}{H^{0.775}}$.	34
5.5	Blade cascade for booster pump illustrating the blade deflection angle and how the forward impeller determines the inflow velocity of the aft impeller.	35
5.6	Chord distributions for both impeller stages from impeller hub to tip. Radial positions nondimensionalized by tip radius and chord lengths nondimensionalized by impeller diameter.	38
6.1	3D geometry of booster pump seen from multiple angles.	40
6.2	Swirl velocity distribution from hub to tip at outlet of aft impeller stage. .	41
6.3	Nondimensional circulation distribution from impeller hub to tip for both impeller stages.	42
6.4	Radial distribution of the total resultant inflow velocities of each impeller stage.	43
6.5	Radial distribution of axial and tangential self-induced velocities for both impellers.	44
6.6	Radial distribution of axial and tangential interaction velocities.	45
6.7	Illustration of the difference between a high-head and a low-head runner. Elongating the blades will increase the lifting height of a centrifugal pump or an RPT.	47

NOMENCLATURE

Symbols

α	Angle of attack [$^{\circ}$]
β_i	Pitch angle of resultant inflow velocity [$^{\circ}$]
β_w	Wake pitch angle [$^{\circ}$]
$\Delta\beta$	Deflection angle [$^{\circ}$]
η_h	Hydraulic efficiency []
Γ	Circulation [$\frac{m^2}{s}$]
λ	Lagrange multiplier [m]
λ_Q	Lagrange multiplier [$\frac{rad}{s}$]
λ_T	Lagrange multiplier [$m\frac{rad}{s}$]
ω	Angular velocity [$\frac{rad}{s}$]
ρ	Density [$\frac{kg}{m^3}$]
τ	Torque [Nm]
ITER	Number of iterations []
$NPSH_A$	Available net positive suction head [m]
$NPSH_R$	Required net positive suction head [m]
θ	Blade pitch angle [$^{\circ}$]
\vec{l}	Distance vector [m]
A	Area [m^2]
a	Empirical constant for calculation of $NPSH_R$ []

B	Auxiliary function [Nm] or [W]
b	Empirical constant for calculation of $NPSH_R$ []
c	Chord length [m]
C_D	Drag coefficient []
c_m	Meridional velocity [$\frac{m}{s}$]
D	Diameter [m]
F_D	Drag force [N]
F_i	Kutta-Joukowski lift force [$\frac{N}{m}$]
F_L	Lift force [N]
F_v	Viscous drag force [$\frac{N}{m}$]
g	Gravitational acceleration [$\frac{m}{s^2}$]
H	Head [m]
h	Pressure head [m]
H_s	Suction head [m]
J	Advance ratio []
K_Q	Torque coefficient []
K_T	Thrust coefficient []
M	Number of vortex panels []
n	Rotational speed [$\frac{rev}{min}$]
N_s	Specific speed []
P	Power [W]
p	Pressure [Pa]
Q	Volumetric flow rate [$\frac{m^3}{s}$]
q	Torque ratio []
R	Blade tip radius [m]
r	Radius [m]
R_h	Hub radius [m]
R_{hv}	Hub vortex radius [m]

T	Thrust [N]
T_r	Required thrust [N]
U	Blade section velocity [$\frac{m}{s}$]
u^*	Induced velocity [$\frac{m}{s}$]
u_{tip}	Blade tip velocity [$\frac{m}{s}$]
V	Velocity [$\frac{m}{s}$]
V^*	Total resultant inflow velocity [$\frac{m}{s}$]
V_s	Ship speed [$\frac{m}{s}$]
X_f	Axial distance between impellers [m]
Z	Number of blades []
z	Elevation above a zero-level [m]

Subscripts

1	Forward impeller
2	Aft impeller
a	Axial
avg	Average
c	Control point
max	Maximum
sub	Submergence
t	Tangential
tot	Total
v	Vortex point

ABBREVIATIONS

PHS	=	Pumped Hydroelectric Storage
VRE	=	Variable Renewable Energy
RPT	=	Reversible Pump Turbine
CRP	=	Contra-Rotating Propeller
CFD	=	Computational Fluid Dynamics
DNS	=	Direct Numerical Simulation
PAT	=	Pump As Turbine
NACA	=	National Advisory Committee for Aeronautics
$NPSH_A$	=	Available Net Positive Suction Head
$NPSH_R$	=	Required Net Positive Suction Head
GUI	=	Graphical User Interface

INTRODUCTION

1.1 Background and motivation

The vast majority of the world's installed energy storage capacity is accounted for by pumped hydroelectric storage (PHS) [5]. PHS has been used to achieve load-shifting and to balance inflexible sources of power generation since the beginning of the 1900s. The wide range of power and energy capacity offered by PHS, as well as the long lifetime of PHS facilities, often makes it the preferred form of energy storage when compared to battery storage. Furthermore, it has a significant economic advantage over conventional hydroelectricity when electricity prices vary periodically. A PHS plant can profit from the price variations in the electricity market by pumping water to the upper reservoir when prices are low and producing electricity when prices are high. A shift in the balance between energy production and consumption will cause these price variations. Such shifts tend to have a certain periodicity, and they often occur daily, weekly or seasonally, depending on the underlying cause of the shift. Traditionally, PHS plants have absorbed surplus power and generated peaking capacity when the base-load power generation was provided by inflexible sources such as coal-fired or nuclear power plants. As the world now strives to utilize more variable renewable energy (VRE) sources, the balance between consumption and generation will shift more swiftly. VRE sources such as solar photo-voltaic, wind and wave energy are intermittent sources, and they will lead to an increase in fluctuations in power generation. A fluctuating nature on both the generation and consumption side will boost the need for PHS plants operating on short time-scales. The International Hydropower Association estimates an increase of 50% of the world's PHS capacity by 2030 [5].

In Norway, conventional hydropower plants far outnumber the PHS plants [6]. Despite this, several of Norway's existing conventional hydropower plants are built in sites well suited for PHS plants. These sites already have an upper and a lower reservoir, and they have infrastructure that can be reused for a PHS plant. Being able to reuse existing infrastructure will severely cut the costs of retrofitting pumping capacity in an existing hy-

dropower plant site. The biggest costs of installing a new hydropower plant are usually related to civil works such as the construction of the dam(s), tunnels, canals and powerhouse [7]. These costs can account for up to three quarters of the total costs, and represent an especially high fraction of the total costs for large capacity plants. Over 100 Norwegian hydropower plants are located between two regulated reservoirs and around 20 of these lay between reservoirs of more than a hundred million cubic meters [8]. This gives Norway a big advantage in developing PHS plants, as opposed to for instance Germany, where most new PHS plants are built with the construction of new artificial reservoirs.

Despite having locations where no dam or reservoir construction is needed, developing a new PHS plant is still very expensive. Excavating a new generator hall, surge tank and new waterways usually dominates the costs of construction. These expenses can, however, be avoided by reusing the infrastructure of an existing hydropower plant, thus making the development far more economically viable. There are several ways a conventional Francis hydropower plant could be retrofitted with pumping capacity, but many of these require large changes to be made to the infrastructure of the plant. Even constructing a whole separate waterway with a pumping station can be necessary. A possibility that minimizes infrastructure changes is to replace the Francis turbine with a reversible pump turbine (RPT) in the same location, thus removing the need for civil works almost entirely. This solution, however, poses several problems that must be solved. First of all, an RPT with the same main dimensions as the current Francis turbine will produce a too low lifting height in pump mode when operating at the same rotational speed as the Francis runner. This problem can partly be solved by increasing the rotational speed of the generator (motor), which will increase the lifting height of the RPT. Increasing the rotational speed will, however, increase the risk of cavitation on the impeller. Therefore, a configuration of an axial booster pump and an RPT pumping in series is proposed. The configuration is shown in Figure 1.1. In addition to relieving the RPT load, the booster pump will increase the inlet pressure of the RPT, thus allowing it to rotate at higher speeds without cavitating.

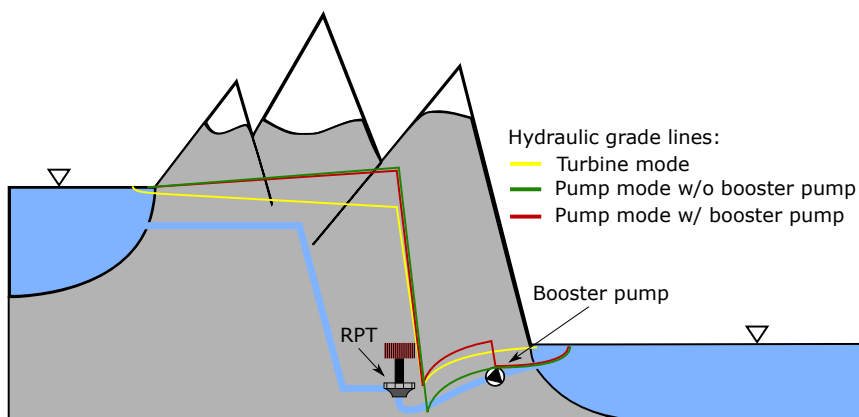


Figure 1.1: RPT configuration with booster pump to increase inlet pressure of RPT in pump mode.

1.1.1 Case power plant

Roskrepp hydropower plant is located between the two reservoirs Roskreppfjorden and Øyarvatn in Agder county in the southwestern part of Norway. It is operated by Sira-Kvina power company and has a head of 83 m [9]. Currently, Roskrepp hydropower plant utilizes a Francis turbine, but the possibility of installing an RPT instead to retrofit pumping capacity is being investigated. A preliminary study suggests that an RPT with the same main dimensions and rotational speed as the current Francis turbine will produce 20 m too low lifting height when operating in pump mode. Part of this deficit can be made up for by increasing the rotational speed of the RPT, but a booster pump might still be needed to reach the target lifting height. Therefore, a contra-rotating axial booster pump design is being investigated. The pump is intended to be placed at the exit of the draft tube of the existing Francis runner, as shown in Figure 1.2. Here, an existing access hatch will facilitate the pump installation.

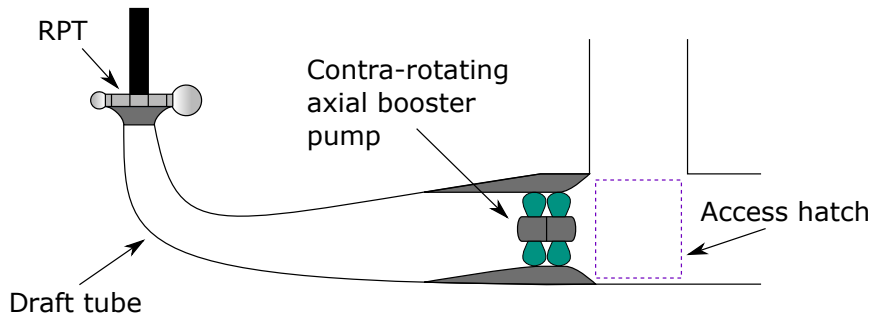


Figure 1.2: Intended placement of contra-rotating axial booster pump at Roskrepp hydropower plant.

1.2 Objective

The objective of this work is to investigate the design of axial hydraulic machines and develop a user-friendly way of designing contra-rotating axial pumps in Matlab. A contra-rotating propeller design method will be made available for user-friendly, rapid designs of contra-rotating axial pumps. The possibility of designing hub-less pumps shall be included. Furthermore, a straightforward manner of displaying and post-processing the design results will be developed. Finally, a preliminary contra-rotating axial pump design shall be created for the intended use as a booster pump at Roskrepp hydropower plant. The validity and performance of the booster pump design will be discussed on the basis of theoretical and practical considerations. As a part of the work, the project was presented and submitted as a research paper for the *9th International symposium on Current Research in Hydropower Technologies (CRHT-IX)* at Kathmandu University in Nepal. The presented research paper is available in Appendix F.

LITERATURE REVIEW REGARDING AXIAL HYDRAULIC MACHINE DESIGN

2.1 Historical development

Axial hydraulic machines have been around for more than two thousand years, and several well-known mathematicians have pioneered axial machine designs throughout history. The simple screw propeller can be dated all the way back to Archimedes of Syracuse, who used the elementary design to lift water from lower to higher elevation [10, p.120]. Famously, Leonardo da Vinci sketched his simple screw propelled helicopter centuries before any helicopter was ever built [11, pp.1-2]. The sketch is shown in Figure 2.1. Despite these early ideas and simple designs, no theoretical advances were made in axial hydraulic machine design until the middle of the nineteenth century. In 1865, W.J.M Rankine suggested a theory based on an ideal flow of water passing through a propeller actuator disk with no slipstream rotation [12]. He considered the propeller to absorb all of the engine power and then dissipate this power as a pressure jump of the water flowing across the disk, i.e. increasing the total head of the flow. This theory is generally known as the axial momentum theory. Some years later, in 1887, R.E. Froude removed the assumption of no slipstream rotation, which made the model more realistic for propellers.

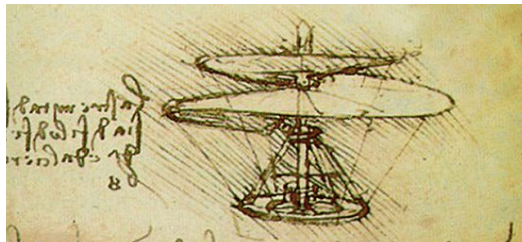


Figure 2.1: Leonardo da Vinci's "Aerial Screw" helicopter. Image from Ligett [2].

It was William Froude who first proposed a theory accounting for the geometry of the axial machine blades, thus making it the first theory applicable for design purposes [13]. In 1878, he developed the so-called blade element theory, where a propeller blade is divided into a large number of two-dimensional strips, as can be seen in Figure 2.2. Each strip is then viewed as an airfoil subject to a resultant uniform inflow. As for any external flow, there will be lift and drag forces related to these airfoils. By knowing experimental values for the lift and drag coefficients, this allowed Froude to calculate both the thrust and torque of a propeller. Froude's theory forms the basis for all modern axial hydraulic machine design theory.

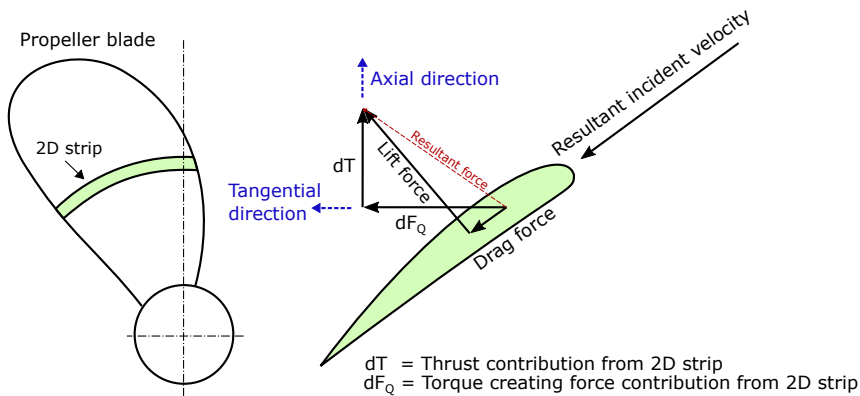


Figure 2.2: Illustration of Froude's blade element theory.

In the early 1900s, Prandtl had proposed that the lift of a wing was caused by an induced circulation around the wing section and that free trailing vortices were shed from the blade and the blade tip [14]. An illustration of Prandtl's lifting line formulation is shown in Figure 2.3. He confirmed his theory experimentally in 1919 and concluded that by calculation of the velocities induced by the free vortices, the forces acting on the blade sections would be equal to the ones in two-dimensional flow. Betz investigated further and found that having the trailing vortices follow a constant-pitch helical wake would minimize propeller losses and maximize efficiency.

Later on, in 1944, Burrill presented a propeller design procedure that combined the basis of momentum theory and blade element theories with parts of Prandtl's vortex method [13]. This was the first consistent progress in propeller design theory, and it proved to work well for moderately loaded propellers at near-design condition. To this day, Burrill's method remains the latest progress in developing a propeller theory combining blade element and momentum theories. Several methods have introduced correction factors to make use of Burrill's method for heavily or lightly loaded propellers.

Generally, methods succeeding Burrill's procedure have utilized the lifting line and lifting surface theories from aerodynamics. The perhaps most important one of these newer methods was proposed by Lerbs in 1952. He developed what was the most advanced lift-

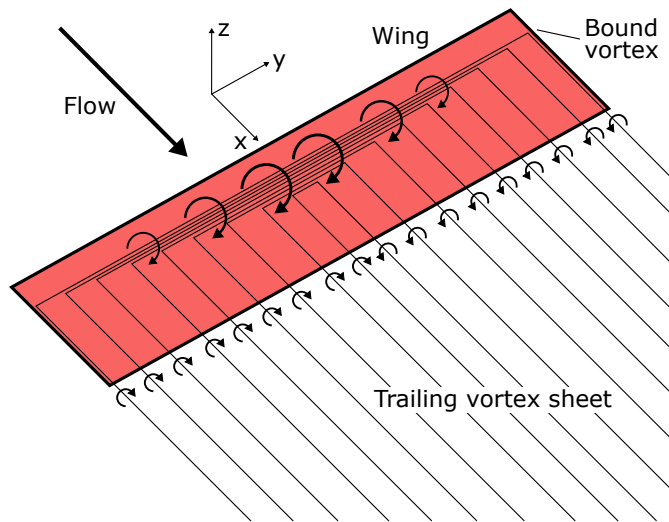


Figure 2.3: Illustration of Prandtl's lifting line theory. A set of circulation vortices is shed from the wing along the flow direction.

ing line method to date. His method considered the propeller blades to be represented by lifting lines of radially varying bound vorticity [13]. Previous lifting line methods had assumed the vorticity to be constant. Lerbs' method was created for a moderately loaded propeller working in an inviscid fluid, and thus the induced velocities from the trailing vortices needed to be accounted for. He related the induced velocities to the radial circulation distribution of the lifting lines by an analogy to Biot-Savart's law, an equation describing the magnetic field generated by a constant electric current. Another new feature of Lerbs' method was the inclusion of a propeller hub. He assumed the circulation to be zero at the hub, which is not physically realistic, but it was argued that the pressure side of one blade and the suction side of another would cause the circulation here to tend towards zero. The method of Lerbs is considered the basis for modern, moderately-loaded lifting line theory.

In the 1960s, a series of lifting surface methods spawned from the already well-established lifting line methods. The lifting surface methods modeled a propeller blade as an infinitely thin surface with the shape of the blade camber line. A distribution of vorticity was placed upon this surface, varying in both radial and chordal (along the blade chord) direction. Both propeller design and analysis were possible with lifting surface methods, as with lifting line methods. An advantage of the lifting surface models is the ability to model blade thickness. This is done by placing a distribution of sinks and sources along the chord of a blade section, a known method used in potential flow theory to model solid objects in an external inviscid flow. Lifting surface theories assume the radial distributions of chord length, thickness, skew and rake to be known. The goal is to determine the radial and chordal camber distributions and the radial pitch distribution. As computational capabilities developed after the 1960s, new, more advanced, lifting surface variations appeared. Since lifting surface methods often require large-capacity computers, there has also been

developed hybrid lifting line/lifting surface models. These methods mainly utilize lifting line theory, but they apply lifting surface correction factors to approximate the computationally heavier lifting surface methods.

Existing lifting line and lifting surface methods have also been adapted for special applications such as ducted propellers, controllable pitch propellers, contra-rotating propellers (CRPs) and supercavitating propellers. For some of these applications, e.g. controllable pitch propellers, the same design methods can be used. Special performance analysis methods are, however, required because they have different operating conditions. For others, such as CRPs and ducted propellers, the design methods have been modified to model the new geometrical features that set them apart from conventional open-water propellers.

Following the development of faster computers, the use of computational fluid dynamic (CFD) design methods for axial hydraulic machines has increased. Multiple methods exist, and most of these are finite volume methods with a turbulence model and a cavitating flow model. Often, the choice of turbulence or cavitating flow model is what separates the different CFD methods, and the designer chooses a method based on the available computing resources. Some methods, such as Direct Numerical Simulation (DNS), are computationally heavy and only expedient for research purposes. Other methods are computationally lighter but might be too inaccurate for the application at hand. Choosing the most expedient method for a certain design has become part of the design process. Also, the numerical properties of each individual method are of great importance. These properties determine when the method is numerically stable, and also other effects such as smearing of the results.

2.2 Modern axial hydraulic machines

Today, a multitude of uses for axial hydraulic machines exists, and the design of a machine is usually limited by the intended application's requirements. In hydropower, for instance, the outer diameter of an axial turbine or pump may be limited by access roads and tunnels for transportation of the machine, whereas for a yacht propeller, the noise and vibrations may be the limiting factor due to comfort considerations. Some other factors that can affect axial hydraulic machine designs are cavitation properties, price and structural properties. The performance requirements also vary from application to application. Efficiency will be of great importance for applications such as axial water turbines, whereas a water cooling pump might allow a lower efficiency if that means saving money on the design process.

Modern axial hydraulic machines vary greatly in both configuration and size because of the abundance of suitable applications. At one end of the spectrum, there are extremely large machines such as the Annapolis STRAFLO tidal turbine, which delivers nearly 20MW of power at a rated head of 5,5 m and discharge of $408 \frac{\text{m}^3}{\text{s}}$ [15], and a lot of large Kaplan turbines around the world. These machines are usually engineered to order and have high demands for performance and life expectancy. On the other side of the spectrum, there exists a range of mass-produced axial machines, such as small boat propulsors and small water pumps, that are off-the-shelf items. These are cheap, and often don't have very high

performance or life expectancy requirements. In addition to size variations, a lot of different configurations, such as CRPs, water jet propulsors, rim-driven machines and hubless rotors, are adopted by engineers designing axial hydraulic machines. CRPs are used to remove torque on the body supporting the shaft on which they are mounted, as well as to increase efficiency by removing rotational components from the flow [16]. A pair of propellers in close proximity will undoubtedly affect each other, and thus this effect must be accounted for in the design process. Another configuration is the water jet propulsor, which is often used in marine applications. A common water jet configuration is shown in Figure 2.4. For a water jet design method to be accurate, the effects of the duct must be accounted for.

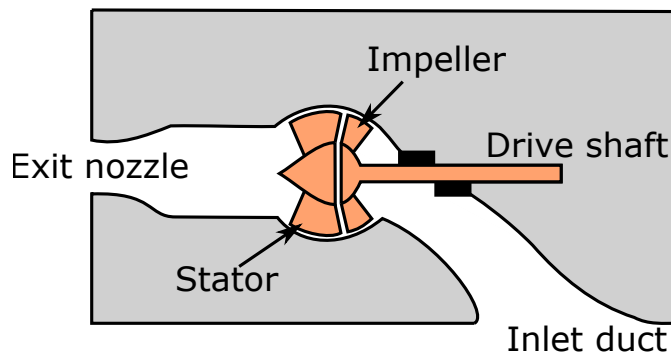


Figure 2.4: Water jet propulsor.

Regardless of configuration, it is usually CFD, blade element theory or lifting line theory that forms the foundation for modern axial hydraulic machine design. Correction factors, specific models and approximations are added to best fit the application at hand.

BASIC THEORY

3.1 Pumped hydroelectric storage

Pumped hydroelectric storage is a manner of hydroelectric storage where electricity is generated by employing water that has previously been pumped from a lower origin to an upper reservoir. When water is pumped from lower elevation to a higher elevation reservoir, the water is given gravitational potential energy. This potential energy can be converted to electric energy by allowing the water to flow to lower elevation through a water turbine. Because of losses in the waterway and the hydro machinery, PHS is a net consumer of energy. Despite this, price variations in the electricity market can make PHS economically profitable. PHS plants are divided into two types: pure and combined. Pure plants rely entirely on water previously pumped to the upper reservoir, whereas combined plants have both pumped water and natural streamflow water [17]. Various configurations of PHS exist, and the most important ones are the separate pump and turbine, reversible pump turbine and pump as turbine (PAT) configurations. The separate pump and turbine configuration has two separate waterways for pumping and energy production. This requires more civil work which is often expensive. The reversible pump turbine configuration uses a centrifugal pump that is designed to also have a good efficiency when operated as a turbine in the other rotational direction. This is a more cost-effective solution than having the pump and turbine separated, but it is often at the expense of turbine efficiency [18, p.262]. PAT configurations, like RPTs, have a centrifugal pump that works as a turbine in the other rotational direction. A PAT configuration, however, uses pumps that are exclusively designed for pumping purposes. As a result, PAT configurations often have lower turbine mode efficiency than other PHS configurations. For state of the art PHS facilities, this is inexpedient, but many isolated, rural areas, developing countries and other places where efficiency is less of a priority can make great use of PAT configurations [19].

3.2 External flow

A solid body fully submerged in a flowing fluid with no boundary layer effect from external boundaries is said to be in an external flow. An airplane flying, a power line in the wind and an underwater pipeline are examples of solid bodies in external flows. These bodies are often classified as either streamlined or bluff bodies. A body is considered to be streamlined if its shape is aligned with the anticipated streamlines of the fluid flowing past it. Examples of streamlined bodies are race cars, arrows and airplanes when they are operated as intended. The opposite of a streamlined body is a body that tends to block the flow. This is called a bluff body. Spheres and cylinders are often used as examples when describing bluff bodies. When a solid body is in a passing flow, it experiences a force from the flow that is usually decomposed into a force in the flow direction and a force perpendicular to the flow. These forces are called drag and lift respectively. The magnitudes of lift and drag are highly dependent on the body's orientation in the incoming flow.

3.2.1 Drag

Drag is generally an unwanted effect as it is associated with higher energy consumption and unwanted consequences such as higher fuel consumption of a car or lower efficiency in an axial hydraulic machine. Special attention is often paid to developing streamlined designs and avoiding bluff bodies. The drag force exerted on a bluff body is usually dominated by pressure drag, caused by the pressure difference of the front and back of the body, whereas the drag force acting on streamlined bodies is dominated by friction drag due to a viscous fluid flowing over the body surface [20]. Developing a streamlined design will have opposite effects on pressure and friction drag. Pressure drag will be reduced but friction drag will increase with a larger surface area, which often is a side effect of streamlining a design. It is therefore only expedient to minimize the sum of the drag forces when designing, not one or the other. Drag is often stated by a dimensionless drag coefficient, C_D , which is defined in Equation 3.1.

$$C_D = \frac{F_D}{\frac{1}{2}\rho V^2 A} \quad (3.1)$$

Here, F_D is the total drag force on the body, ρ is the fluid density, V is the velocity of the fluid and A is an arbitrary area, usually the projected frontal area of the body, i.e. the area seen from the incoming flow [20, p.565].

3.2.2 Lift

The lift force on a solid body works in a direction perpendicular to the flow direction and is mainly caused by a pressure difference between the top and bottom surfaces of the body. Viscous forces also contribute to lift, just as they do to drag, but for most engineering applications these forces are often negligible. To elaborate, bodies in engineering applications are often streamlined to minimize drag which causes the viscous forces to be primarily aligned with the flow direction. Similar to the drag coefficient in Equation 3.1, lift is often stated as a dimensionless coefficient, C_L , defined in Equation 3.2.

$$C_L = \frac{F_L}{\frac{1}{2}\rho V^2 A} \quad (3.2)$$

Here, F_L is the lift force on the body and A is again an arbitrary area, but most commonly the area that would be seen from above the body in a direction normal to the flow direction [20, p.588]. The rest of the parameters are the same as in Equation 3.1.

3.2.3 Wing theory

Applications such as race cars and airplanes utilize wings shaped as airfoils that are designed to maximize lift and minimize drag. A vast number of airfoil sections have been created and tested, and a great part of existing airfoil data was recorded by NASA's predecessor, the National Advisory Committee for Aeronautics (NACA). An arbitrary foil section is shown in Figure 3.1. The chord line is a straight line from the leading to the trailing edge of the foil. The angle of the chord line with the incoming flow direction is the most important parameter in lift generation and is called the geometric angle of attack. All foils have a certain angle of attack called the zero-lift angle, where only drag forces are exerted on the foil. The camber line also goes from the leading edge to the trailing edge of the foil, but unlike the chord line, the camber line follows a path that is at an equal distance from the upper and lower surfaces of the foil. For a symmetrical foil, the camber and chord lines are coincident. The local camber of a foil is the normal distance from the chord line up to the camber line at a certain chordal position. Another important property of a foil is its thickness, which is the distance between the upper and lower surface at a certain chordal position. The combination of the thickness and camber distributions along the chord of a foil is what sets it apart from other foils. There is no exact science describing the shape of an optimum foil, but depending on the flow conditions in a given application, some foils will perform better than others. Consequently, foils have been tested at different Reynolds numbers and a range of angles of attack to create databases that can help engineers pick a good foil for their application. Typically, plots of the ratio of the lift coefficient and the drag coefficient of a foil plotted against the angle of attack are utilized. Such a plot for the foil NACA 66-209, a foil developed and tested by NACA, is shown in Figure 3.2. The apex of each Reynolds number line gives the optimum angle of attack and the corresponding lift to drag ratio for the foil. The points where $\frac{C_L}{C_D}$ is zero yield the zero-lift angles for each Reynolds number. Cavitation and strength considerations also matter when choosing a foil for an application.

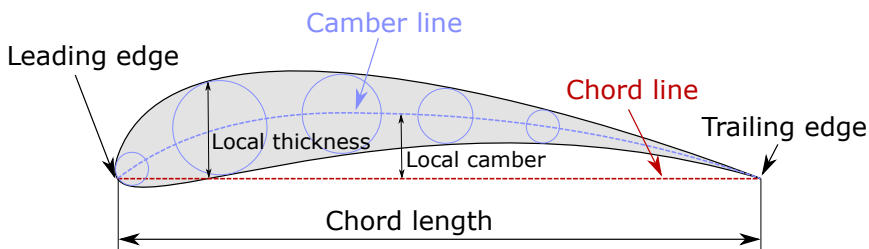


Figure 3.1: Foil section displaying foil terminology.

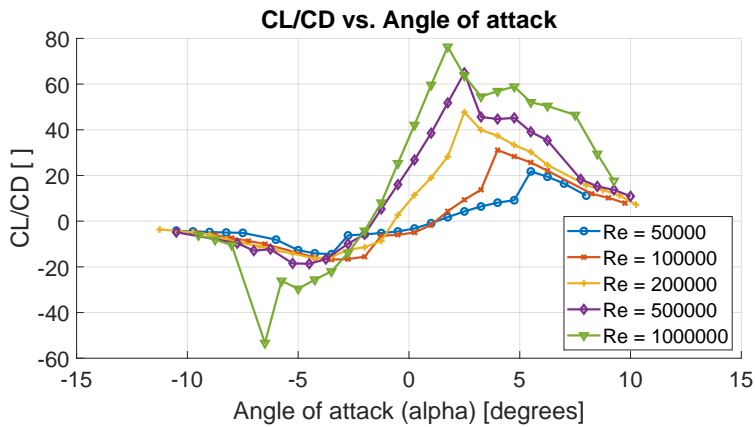


Figure 3.2: Lift coefficient to drag coefficient ratio plotted versus angle of attack for the NACA 66-209 foil. Plot data extracted from <http://airfoiltools.com/airfoil/details?airfoil=naca66209-il>.

A wing is comprised of a foil section extruded for a certain span length. Along the span of the wing, the foil section can change chord length, twist and even change shape. All of these changes will alter the properties of the wing. Engineers optimize these parameters to, among other things, increase efficiency, distribute the wing loading or change cavitation conditions. Two important parameters for wings used in rotating configurations are rake and skew, illustrated in Figure 3.3. Skew will for some applications reduce vibrations on the propeller shaft and is also used to impede cavitation [13]. Rake is often added in marine propellers to increase the distance between the propeller and the ship hull [21].

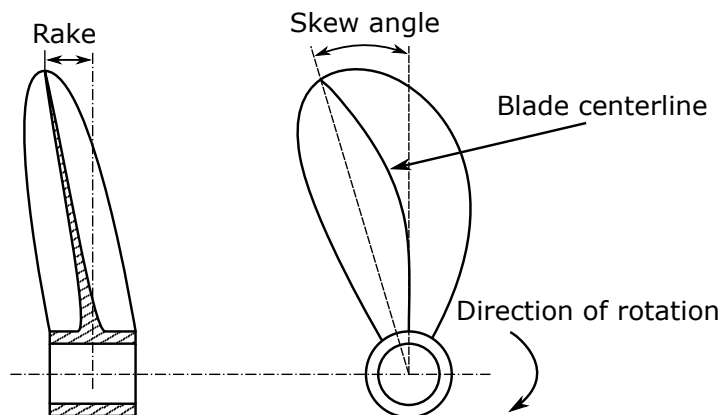


Figure 3.3: Rake and skew of a propeller blade (wing) illustrated.

3.3 Cavitation

Cavitation is the partial evaporation of liquid in a flow system [22]. When the static pressure in a flow drops below the vapor pressure of the liquid, the liquid evaporates locally and forms a small vapor-filled cavity. A small region in the flow field will thus have a two-phase flow for a small period of time. The vapor cavity will abruptly condensate if it is transported into a region of static pressure exceeding the vapor pressure. This is often referred to as an implosion of the vapor cavity and will create a strong local shock in the flow. When such an implosion occurs close to a surface, it can break loose pieces of the surface if the shock exceeds the failure limit of the surface material. This phenomenon is called cavitation erosion, and it can be devastating for engineering applications such as pumps and turbines. Furthermore, the vapor cavities will block the flow path which will decrease the efficiency of a pump or turbine. Frequently, the condensation of vapor cavities occurs when the flow velocity is reduced and pressure is regained, which naturally happens when the flow area increases. Bernoulli's equation, stated in Equation 3.3, describes the conservation of energy of a fluid particle flowing along a streamline.

$$\frac{V^2}{2} + gz + \frac{p}{\rho} = \text{constant} \quad (3.3)$$

Here, g is the gravitational acceleration and ρ is the density of the flowing liquid. It is evident that increasing the flow velocity, V , will decrease the pressure, p , if the elevation, z , remains constant. Cavitation can therefore be minimized by designing flow paths that avoid large local velocities.

3.3.1 Cavitation in an axial flow pump

Cavitation can be a severe problem for an axial flow pump, but if the inlet pressure of the pump is sufficiently increased or local velocities are reduced, the problem can be removed entirely. There are several ways of accomplishing this, including submerging the pump, altering the blade geometries or reducing the rotational speed of the pump. When assessing the cavitation properties of an axial flow pump, the available Net Positive Suction Head ($NPSH_A$) is a defining quantity. The $NPSH_A$ has the same unit as a pressure head, namely meters, and expresses how close the fluid at a certain point in a flow is to cavitation. In most cases, the $NPSH_A$ is provided by the pump site data [1], and it is calculated by Equation 3.4.

$$NPSH_A = -H_s + h_b - h_{va} \quad (3.4)$$

Here, H_s is the suction head of the pump, h_b is the barometric pressure at the altitude of the plant, and h_{va} is the vapor pressure of the fluid at the given flow conditions. The barometric pressure is normally 10.3 m at sea level, but decreases with 0.12 m for each 100 m above sea level. The suction head accounts for the submergence of the pump and the average velocity of the flow entering the pump, as described in Equation 3.5.

$$H_s = h_{sub} + \frac{V_{avg}^2}{2g} \quad (3.5)$$

In Equation 3.5, h_{sub} is the level of submergence of the pump, as shown in Figure 3.4, V_{avg} is the average flow velocity and g is the gravitational acceleration. The submergence is negative for pumps located below the water surface.

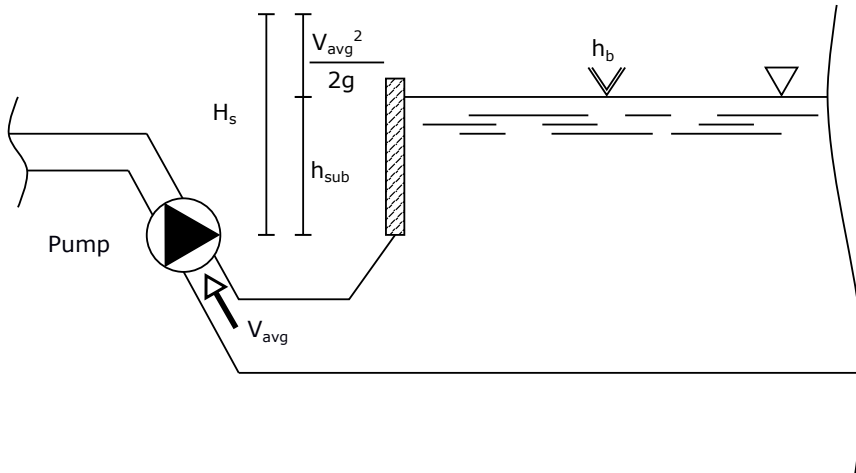


Figure 3.4: Suction head of a pumping application.

The $NPSH_A$ describes the cavitation conditions in the location where the pump is placed, but it doesn't factor in the pump itself. Therefore, a required Net Positive Suction Head ($NPSH_R$) that assures no cavitation in the pump must be established. Normally, this value is specified by the pump manufacturer. An empirical formula, shown in Equation 3.6, has been developed to estimate the $NPSH_R$ [1].

$$NPSH_R = a \frac{c_m^2}{2g} + b \frac{u_{tip}^2}{2g} \quad (3.6)$$

Here, c_m is the meridional velocity of the flow, and u_{tip} is the velocity of the pump blade tip. The constants a and b are dependant on the specific rotational speed, outlet angle and runner geometry, and common values are displayed in Table 3.1, collected from [1].

	Turbines	Pumps
a	$1.05 < a < 1.15$	$1.6 < a < 2.0$
b	$0.05 < b < 0.15$	$0.2 < b < 0.25$

Table 3.1: Empirically derived values for parameters a and b for calculation of $NPSH_R$. Collected from [1].

To assure that no cavitation occurs in the pump impeller, the required net positive suction head must necessarily be less than the available net positive suction head, as stated in Equation 3.7.

$$\text{NPSH}_R < \text{NPSH}_A \quad (3.7)$$

The blade tip speed, u_{tip} , in Equation 3.6 is dependant on the rotational speed of the pump. Equation 3.7 can therefore be used to estimate the maximum rotational speed that is below the cavitation limit of the pump. If a pump is required to have a high rotational speed, the NPSH_R might be too high to avoid cavitation. In such cases, multiple pumps can be placed in series. A slowly rotating pump with a lower NPSH_R can be used to increase the pressure sufficiently for the next pump to operate at higher rotational speeds. This is a normal configuration for high-speed pumps with large lifting heights, as they are more prone to cavitation.

3.4 Axial hydraulic machinery

Hydraulic machines are machinery that either extracts energy from a flowing liquid or provides energy to the liquid. These energy transfers are accomplished by either a generator or a motor that is connected to the rotating hydraulic machine. Hydraulic machines are often categorized as either radial, semi-radial or axial according to the way liquid flows through them. In axial hydraulic machines, the liquid particles follow streamlines with an approximately constant radial distance to the axis of rotation. The flow may, however, have a velocity component tangential to the axis of rotation. Momentum is provided to or extracted from the flow by lift forces on foil shaped blades. This is different from radial and semi-radial hydraulic machines where centrifugal forces are of great importance to the momentum transfer between the machine and the flow. Centrifugal forces play little or no role in the momentum transfer in axial machines. Commonly, axial hydraulic machines are used in applications such as tidal turbines, low-head hydropower turbines or water pumps, however, a range of other applications exists.

3.4.1 Axial flow pumps

Axial flow pumps are generally used for pumping applications with high flow rates and low pumping heads. In such applications, they provide higher efficiencies than both radial and semi-radial pumps. The specific speed, N_s , of a pump is used to determine what type of pump is suitable for a pumping application and is often a starting point for the pump design process. Equation 3.8 asserts the specific speed of a pump given a set of head-capacity requirements and a rotational speed.

$$N_s = \frac{\omega\sqrt{Q}}{(gH)^{\frac{3}{4}}} \quad (3.8)$$

Since the rotational speed, ω , often is limited by cavitation considerations, the specific speed is often determined by the head, H , and capacity, Q , requirements. For multistage pumps, the specific speed is computed for the first stage only [23]. N_s in Equation 3.8 is dimensionless if ω , H and Q have the units $\frac{\text{rad}}{\text{s}}$, m and $\frac{m^3}{\text{s}}$ respectively, and the gravitational acceleration, g , is given in $\frac{m}{\text{s}^2}$. Axial pumps have higher dimensionless specific speeds than radial and semi-radial pumps, with values starting at approximately $N_s \approx 2.6$

[24, p.55]. For lower values than this semi-radial pumps will have better efficiency, and for very high values, typically above $N_s \approx 6$, hydraulic losses can become very large [4].

Axial flow pumps are essentially the same as marine propellers, at least as far as the impeller geometry goes. They do, however, differ slightly, as an axial pump needs a set of inlet guide vanes or exit vanes to increase the pressure across the pump. A propeller is only required to produce a thrust force, and can therefore exist as a solitary impeller. Despite this difference, the underlying theory is the same for propeller and pump design. A useful aspect to understand is that propeller thrust and pump head are closely coupled together. Epps [25] states the relationship between useful propeller power, P , and propeller thrust, T , for a ship moving at constant speed, V_s , as shown in Equation 3.9.

$$P = TV_s \quad (3.9)$$

The useful power of a pump is given in Equation 3.10 [22, p.46], where ρ is the pumping liquid density.

$$P = \rho g Q H \quad (3.10)$$

An equation describing the relation between pump thrust and head can then be derived, shown in Equation 3.11. The ship speed, V_s , is set to the average axial flow velocity entering the pump for pumping applications. In marine propellers, the ship speed is related to the axial velocity seen by the propeller by a wake fraction. This wake fraction is intended to represent the effects of the boundary layer from the ship hull on the flow entering the propeller. For a pump application, this wake fraction is assumed to be zero, and therefore the axial flow velocity and ship velocity are set equal.

$$T = \frac{\rho g H Q}{V_s} \quad (3.11)$$

Equation 3.11 is an important result as it removes one of the barriers between pump and propeller design. Propeller design methods usually strive to meet a certain thrust requirement, whereas pump design operates with head and flow rate requirements. By recognizing this result, propeller design methods are made available for pumping applications.

Axial pumps, as other axial machines, are often categorized by their advance ratio, J , which is the nondimensional ratio of the axial flow velocity, V_a , through the impeller and the impeller tip speed. The advance ratio can be calculated by Equation 3.12, where n is the impeller's rotational speed in *rpm* and D is the impeller diameter in meters.

$$J = \frac{60V_a}{nD} \quad (3.12)$$

The advance ratio is high for large incoming flow velocities and slowly rotating impellers, and low for small incoming flow velocities and quickly rotating impellers. Often, the efficiency of an axial flow pump is highly affected by the advance ratio of the impeller [26].

For a pump, the hydraulic efficiency is of great importance. It is defined as the ratio of useful power delivered to the flow and power delivered to the pump. The useful power

delivered to the flow is only in the form of an increase in liquid pressure, as an increase of the flow's kinetic energy is considered a loss of energy. Power delivered to the pump is in the form of torque on the impeller, which can either be transferred by a shaft or by an induced magnetic field. Equation 3.13 expresses the hydraulic efficiency of a pump.

$$\eta_h = \frac{P_{useful}}{P_{delivered}} = \frac{\rho g Q H}{\tau \omega} \quad (3.13)$$

Here, H is the pressure head delivered to the flow, Q is the volumetric flow rate, τ is the torque on the pump impeller and ω is the angular velocity of the impeller. When several pumps are pumping in series, the hydraulic efficiency is computed with the total head, i.e. the sum of the pressure head of all of the pumps, and the delivered power is the sum of the delivered power of all of the pumps. A conventional way of stating the torque acting on an impeller is by a torque coefficient, K_Q , defined by Equation 3.14.

$$K_Q = \frac{\tau}{\rho \left(\frac{n}{60}\right)^2 D^5} \quad (3.14)$$

Similarly, the thrust of an impeller is often expressed by the thrust coefficient K_T , defined in Equation 3.15.

$$K_T = \frac{T}{\rho \left(\frac{n}{60}\right)^2 D^4} \quad (3.15)$$

CONTRA-ROTATING AXIAL PUMP DESIGN METHOD

In this work, the design method of Laskos [3] is used for a preliminary design of an axial contra-rotating booster pump. In his work, a lifting-line method, based on Lerbs' formulation, has been modified for the design of a two-stage contra-rotating propeller. This chapter highlights the fundamental theory and assumptions of lifting-line propeller design and optimization, and it details the extension to multiple-component impellers.

4.1 Lifting line theory

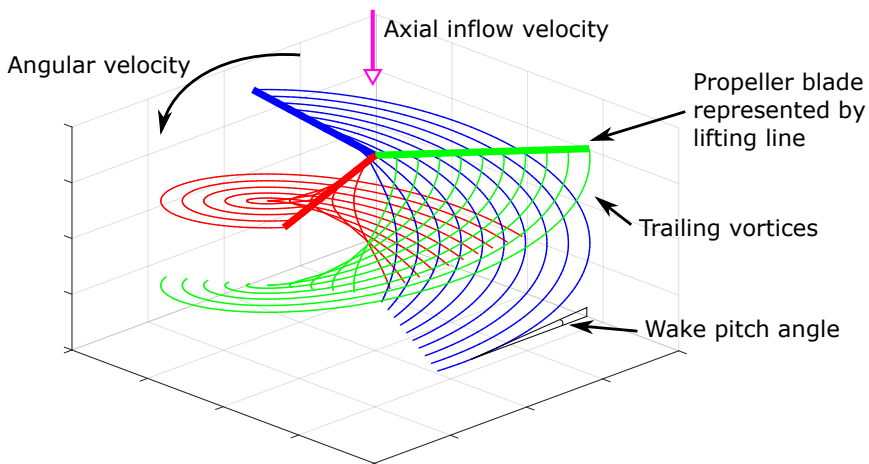


Figure 4.1: Lifting line representation of a propeller with three blades in a constant axial inflow. Trailing vortices are shed from the blades at a constant wake pitch angle.

In Lerbs' lifting line method, all Z blades of a propeller are modeled as straight radial lifting lines from the blade hub to the tip. The lifting lines consist of radially varying

vorticity, $\Gamma(r)$, and have equal angular spacing and identical loading. Trailing behind each of the lifting lines is a vortex sheet, assumed to have the shape of a helix with a fixed pitch angle, β_w , and radius. This is shown in Figure 4.1. The trailing vortices induce velocities in the flow, which must be included when calculating the lift and drag forces on the blades. The induced velocities affect both the magnitude and incident angle of the total resultant flow velocity. Figure 4.2 shows a two-dimensional blade section at radius r from the axis of rotation with all related velocity components and forces. The total resultant incident velocity, V^* , is a sum of all the velocity components present in the flow. This includes the inflow velocity, V_0 , which has the components V_a and V_t in the axial and tangential direction respectively. Furthermore, the induced velocities in the axial and tangential directions, u_a^* and u_t^* , as well as the tangential contribution from the blade rotation, ωr , is included. The expression for the magnitude of the total resultant incident velocity thus becomes:

$$V^* = \sqrt{(V_a + u_a^*)^2 + (\omega r + V_t + u_t^*)^2} \quad (4.1)$$

The angle of attack, α , of the resultant incident velocity with the chord line of the foil section is the difference of the pitch angle of the blade, θ , and the pitch angle of V^* , namely β_i :

$$\alpha = \theta - \beta_i \quad (4.2)$$

From Figure 4.2, the expression for β_i becomes:

$$\tan \beta_i = \frac{V_a + u_a^*}{\omega r + V_t + u_t^*} \quad (4.3)$$

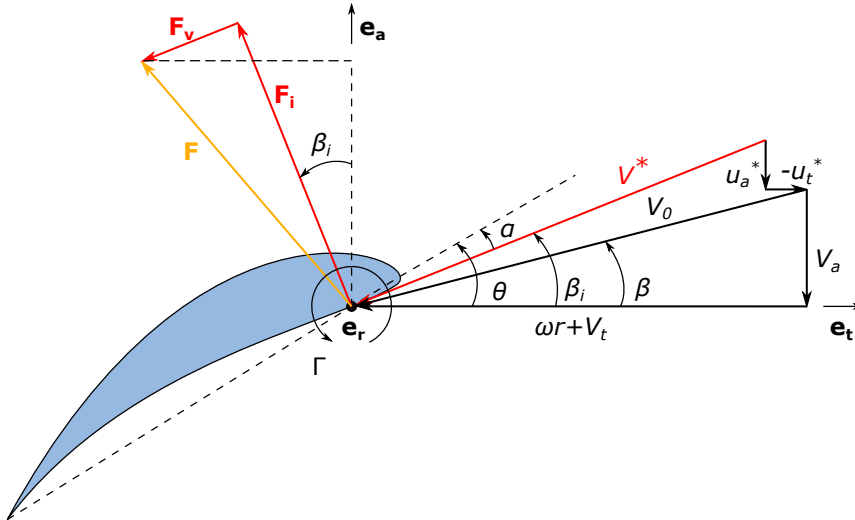


Figure 4.2: Velocity and force diagram of a blade section at an arbitrary radius r from the axis of rotation.

The Kutta-Joukowski theorem for steady, two-dimensional flow states an equation for the inviscid lift force, F_i , per unit span length acting on a cylinder of any cross-section [14]. This force acts perpendicular to the total resultant incident velocity and its magnitude is computed by Equation 4.4.

$$F_i = \rho V^* \Gamma \quad (4.4)$$

Here, Γ is the circulation around the cylinder, i.e. the integral of the velocity field around the cylinders border. Equation 3.1 is used to compute the viscous drag force, F_v , on the foil section. The area A is divided by the blade span to get the force per unit span length, and the length used in the flow direction becomes the chord length, c . The equation becomes:

$$F_v = \frac{1}{2} \rho (V^*)^2 C_D c \quad (4.5)$$

As expressions for the forces per unit span have been derived, the total thrust and torque on the propeller can be computed by integrating the forces along the blades from the hub radius, R_h , to the tip radius, R . The thrust, computed by Equation 4.6, is the sum of the forces acting on the propeller in the axial direction. The torque, computed by Equation 4.7, is computed by summing the product of the tangential forces on the propeller with their respective radial distance from the axis of rotation.

$$\mathbf{T} = Z \int_{R_h}^R [F_i \cos \beta_i - F_v \sin \beta_i] dr \quad (4.6)$$

$$\tau = Z \int_{R_h}^R [F_i \sin \beta_i + F_v \cos \beta_i] r dr \quad (4.7)$$

4.2 Wake model

As previously stated, the wake of each blade is modeled as a trailing vortex sheet. The purpose of the wake model is to compute the induced velocities in the axial and tangential direction, $u_a^*(r)$ and $u_t^*(r)$ respectively, due to the trailing vortices.

The lifting lines themselves do not contribute to the induced velocities, which is evident when considering the assumptions of equal angular spacing between the lines and all the lines being straight radial lines. By analogy to the Biot-Savart law, an equation relating a magnetic field to the constant current that is generating it, the velocities induced by a vortex filament can be described by [27]:

$$\vec{u} = \frac{\Gamma}{4\pi} \int \frac{d\vec{s} \times \vec{l}}{|\vec{l}|^3} \quad (4.8)$$

Here, Γ is the circulation, $d\vec{s}$ is a piece of the filament and \vec{l} is the vector from the piece to where the induced velocity is computed. Since the radial circulation distribution is continuous, the trailing vortex strength is given by Equation 4.9.

$$\bar{\Gamma}(r) = \left[-\frac{d\Gamma(r)}{dr} \right] dr \quad (4.9)$$

To compute the induced velocity at a certain control point r_c , the sum of the influences from all the trailing vortices of all the propeller blades must be summed. By defining the axial influence of a unit-strength trailing vortex filament shed from each of the Z blades as $\bar{u}_a(r_c, r_v, \beta_w)$, we can write the following expression for the axial induced velocity at control point r_c :

$$u_a^*(r_c) = \int_{Rh}^R \bar{u}_a(r_c, r_v, \beta_w) \left[-\frac{d\Gamma(r_v)}{dr_v} \right] dr_v \quad (4.10)$$

Here, the influence on control point r_c of a trailing vortex from each interior vortex point radius, r_v , is integrated. \bar{u}_a is given by a version of the Biot-Savart law, Equation 4.8, but with a unit strength circulation.

$$\bar{u}_a(r_c, r_v, \beta_w) = \frac{1}{4\pi} \sum_{k=1}^Z \int \hat{e}_a \cdot \frac{d\vec{s} \times \vec{l}}{|\vec{l}|^3} \quad (4.11)$$

The same calculations can be done for the tangential induced velocity at a given control point. Lerbs derived an analytical solution for Equation 4.11 in 1952 [27], a solution only dependent on the radii of the control points and interior vortex points and the blade wake angle, β_w .

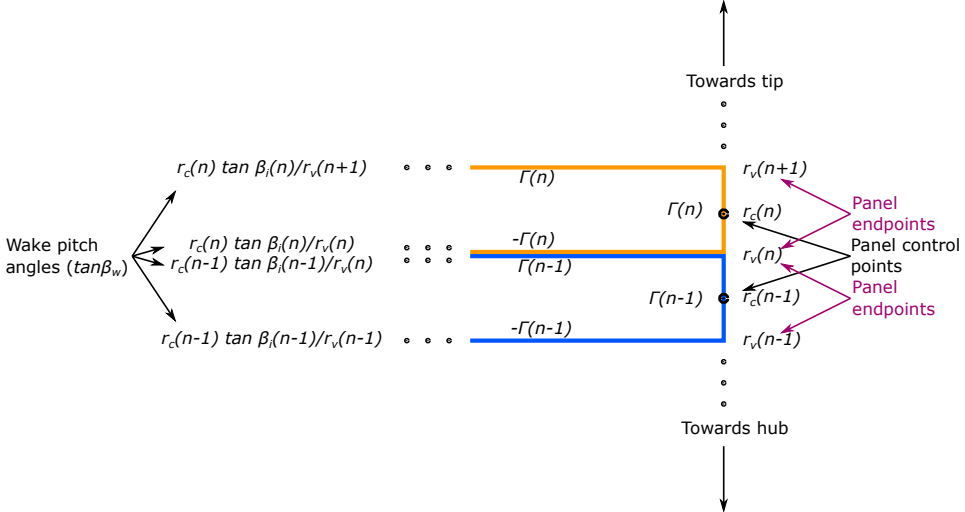


Figure 4.3: Present wake model illustrated for arbitrary control panels $r_c(n-1)$ and $r_c(n)$ along a lifting line.

Discretizing Equation 4.10 has proven to be advantageous, as it makes the equation extendable to propellers with multiple components. In the discretized formulation, the lifting lines are divided into M panels of constant strength. A control point, $r_c(m)$ ($m =$

$1 \dots M$), is placed on each panel, usually on the panel midpoint. The induced velocities are only calculated at these control points. Each panel has endpoints denoted by $r_v(m)$ ($m = 1 \dots M + 1$), so panel n goes from $r_v(n)$ to $r_v(n + 1)$ and contains the control point $r_c(n)$. The model utilized in this work is referred to as the present wake model [27] and is illustrated in Figure 4.3. It states that each panel is surrounded by a complete horseshoe vortex, as shown in the figure. A trailing vortex is shed by each horseshoe vortex at the panel endpoints. This means that two trailing vortices will be shed from each of the interior panel endpoints ($r_v(m)$ ($m = 2 \dots M$)), whereas only one vortex will be shed from the two panel endpoints at the ends of the lifting line.

The trailing vortices of each horseshoe vortex are oriented at a wake pitch angle that is mathematically consistent with the pitch at the control point of the panel, i.e. $2\pi r_c \tan \beta_i = 2\pi r \tan \beta_w$ for the entire panel. At the endpoints of panel n , the wake pitch angle is therefore given by:

$$\begin{aligned} \tan \beta_w(r_v(n + 1)) &= r_c(n) \tan \beta_i(n) / r_v(n + 1) \\ \tan \beta_w(r_v(n)) &= r_c(n) \tan \beta_i(n) / r_v(n) \end{aligned} \quad (4.12)$$

4.3 Optimization

The optimization procedure in lifting line theory aims to determine the optimum propeller blade pitch at each radial section. The optimum blade pitch is the value that produces the required propeller thrust, or pump head, at the lowest torque value. This is intuitive, as it is desired for the motor to spend as little energy as possible rotating the axial machine, but still be able to deliver the required performance. The propeller blade pitch is dependant on the circulation, Γ , and therefore, the optimum circulation distribution along the blade, $\Gamma(r)$, will yield the optimum pitch distribution. Epps [25] and Laskos [3] have both utilized the method of Lagrange multipliers for the circulation optimization. In this method, an auxiliary function, stated in Equation 4.13, is formed.

$$B = \tau + \lambda(T - T_r) \quad (4.13)$$

Here, λ is a Lagrange multiplier, τ and T are the propeller torque and thrust respectively, and T_r is the design-specified required thrust. The Lagrange multiplier's purpose is to introduce a thrust constraint on the optimization, namely that $T = T_r$. It is clear that if this constraint is fulfilled, B is minimum when the torque is minimum. The thrust and torque values are computed from discretized versions of Equations 4.6 and 4.7 respectively. By setting the partial derivatives of B to zero, and solving the resulting system of equations, the optimum circulation distribution is determined. The partial derivatives of H are shown in Equation 4.14, and they form a system of $M + 1$ non-linear equations.

$$\begin{aligned} \frac{\partial B}{\partial \Gamma^{(i)}} &= 0, \quad (i = 1, \dots, M) \\ \frac{\partial B}{\partial \lambda} &= 0 \end{aligned} \quad (4.14)$$

This system of equations is non-linear due to terms containing products of V^* , $\frac{\partial V^*}{\partial \Gamma}$ and λ . Epps [25] solves this problem by freezing these parameters for each iteration of the iterative solution process, so that Equation 4.14 becomes linearized. Whenever the equations are solved for the optimum circulation, the parameters V^* , $\frac{\partial V^*}{\partial \Gamma}$ and λ are updated by their respective equations, as described previously in this chapter. The iterative process is continued until convergence or until a specified maximum number of iterations has been reached.

4.4 CRP modifications

Laskos [3] extended the method of Epps [25] to a set of two contra-rotating propellers. He proposed two methods, the 'coupled' and the 'uncoupled' method, that each perform lifting line optimization to a CRP set. This work utilizes the coupled method, a method that regards the CRP set as a single unit. The method yields both the radial load distribution (circulation) for the two propeller components simultaneously, as well as the load division between the components. An advantage of this method is that it can be used for propellers with inlet guide vanes or exit stay vanes. Thus, it will be applicable for both single-stage and contra-rotating axial pump design. What separates the design of one CRP stage and a conventional single propeller is that the CRP stage will be affected by the other stage. Each stage of a CRP will induce velocities that affect the inflow of the other stage, and these velocities are referred to as interaction velocities. The interaction velocities are similar to the aforementioned, self-induced velocities, but they do not affect the propeller that is inducing them. Hough and Ordway [28] developed a method for calculation of the induced interaction velocities, which has been implemented in the coupled CRP design method.

As for the single propeller case described in Section 4.3, the optimization of a CRP concerns itself with finding the optimum circulation distribution, which in turn provides the optimum blade pitch of each propeller stage. The optimum blade pitch for a CRP set is the value that requires the least power, P , to produce the required thrust, T_r . The power is the sum of the products of torque and angular velocity of each propeller stage, $P = \omega_1 \tau_1 + \omega_2 \tau_2$. Here, subscripts 1 and 2 indicate the forward and aft propeller stage respectively. Since there now are two propeller stages, the required thrust is a sum of the thrust of both stages, $T_r = T_1 + T_2$. In a CRP, controlling the torque distribution between the two stages is often desired. Therefore, a torque ratio, $q = \frac{\tau_2}{\tau_1}$, is introduced in the optimization scheme, which is used as a new constraint on the design. The auxiliary function of Equation 4.13 is updated with the new thrust and torque constraints imposed on the design.

$$B = (\omega_1 \tau_1 + \omega_2 \tau_2) + \lambda_T (T_1 + T_2 - T_r) + \lambda_Q (q \tau_1 - \tau_2) \quad (4.15)$$

Here, λ_T and λ_Q are Lagrange multipliers that enforce the thrust and torque constraints on the design optimization. For the CRP case, the two propeller stages are individually divided into a set of horseshoe vortex panels. The forward and aft stages are divided into M_1 and M_2 panels, respectively, so a total set of $M_1 + M_2$ circulations represent the

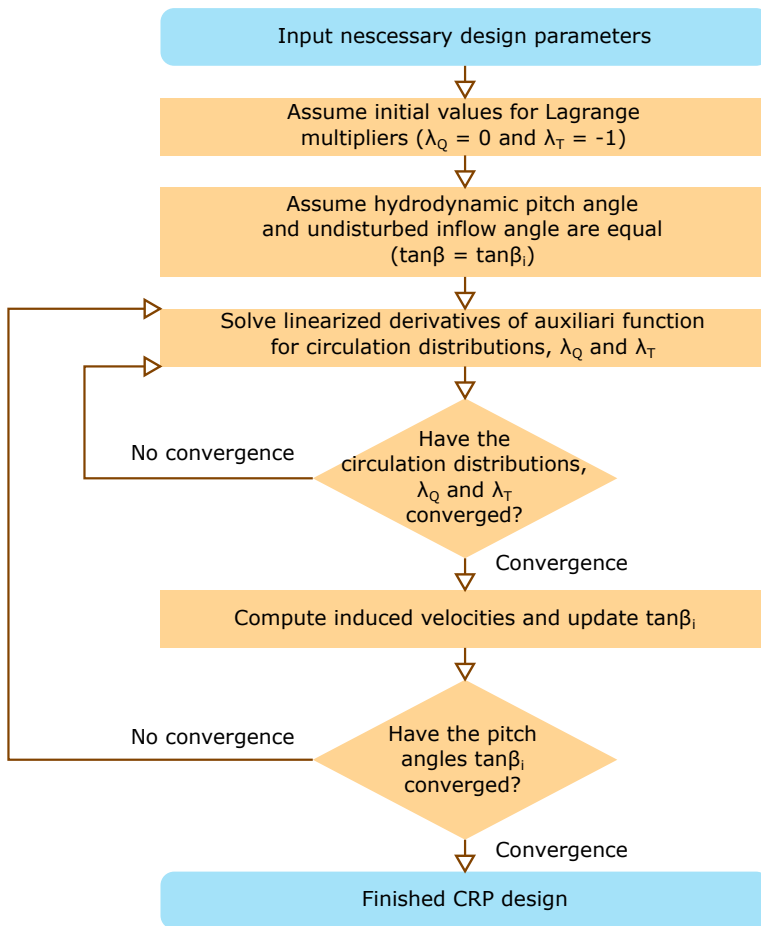


Figure 4.4: Flowchart of coupled CRP design optimization. Method implemented by Laskos [3].

lifting lines of the CRP set. The derivatives of 4.15 with respect to the circulations and the two Lagrange multipliers are set to zero to find the optimum circulation distribution. This constitutes a set of $M_1 + M_2 + 2$ non-linear equations, as shown in Equation 4.16. The equations are written in their entirety in Appendix E. Initial values for the two multipliers are set to $\lambda_Q = 0$ and $\lambda_T = -1$ as these values tend to converge rapidly. The circulations are initially assumed to be zero.

$$\begin{aligned}
\frac{\partial B}{\partial \Gamma_j(i)} &= 0 \\
\frac{\partial B}{\partial \lambda_T} &= 0 \\
\frac{\partial B}{\partial \lambda_Q} &= 0
\end{aligned} \tag{4.16}$$

for $j = 1, 2$ and $i = 1, \dots, M$

The set of equations is linearized by freezing the value of the Lagrange multipliers where they form product terms with the circulation values. After assuming initial values for the Lagrange multipliers and the circulations, the linearized equations are solved until convergence of all parameters. Subsequently, both the self-induced and the interaction velocities are computed for both propeller stages. This yields the total resultant inflow velocity on each stage and the pitch angles of the inflow, β_i . Convergence of the optimized design is reached if the freshly computed pitch angle is equal to the angle used in the computation of the circulations and Lagrange multipliers. If not, Equations 4.16 must be solved again with the new value of β_i . This process is continued until convergence of β_i . The entire coupled design optimization process for a CRP set is highlighted in the flow chart in Figure 4.4.

4.5 Design inputs

Several input parameters are needed to implement the uncoupled method of optimum CRP design, most of which are related to the CRP geometry, flow conditions or performance requirements. Most of these parameters are shown in Table 4.1, and the method for selecting the parameters will be highlighted in the following chapter. For both stages of the CRP, the radius, number of blades and rotational speed must be specified. A hub can be included if desired. If so, a common hub ratio for both impellers, $\frac{R_h}{R}$, is selected. Moreover, the hub will introduce a concentrated vortex in the flow, ending infinitely far downstream from the hub. The ratio of this hub vortex diameter to the hub diameter needs to be specified. Furthermore, a blade section of the propeller blades must be chosen, with a corresponding section drag coefficient for the specified flow conditions. The axial distance between the two stages must be stated, as well as the desired load distribution, q , as previously mentioned. Some performance requirements also need to be specified. These are the required head, the rated discharge and the flow density.

Since the uncoupled CRP method is developed for marine propeller design, a ship velocity, V_s , is a required design input. This velocity is, among other things, used to normalize the inflow velocity inputs, and is assumed equal to the axial flow velocity for pumps, as previously stated. The axial and tangential inflow velocities have to be specified for each of the stages, not including velocity components induced by circulation, as they are to be calculated. The inflow velocities are given at a selected number of radial locations. At the same radial locations, the blade chord lengths must be specified. Both the inflow velocity

components and the chord lengths are interpolated to obtain smoothly varying distributions of these properties from the propeller hub to the tip.

For the optimization process, the number of vortex panels, M , needs to be chosen for each of the stages. A high number of panels yields higher accuracy in the results but will require more computational time. Finally, the maximum number of iterations in the iterative optimization process is specified. Reaching this number of iterations supersedes the convergence criteria of the optimization, and the optimization process will move on.

Symbol	Parameter	Unit
R_1	Radius of forward propeller	m
R_2	Radius of aft propeller	m
Z_1	Forward propeller's number of blades	-
Z_2	Aft propellers's number of blades	-
n_1	Rotational speed of forward propeller	rpm
n_2	Rotational speed of aft propeller	rpm
M_1	Number of vortex panels of forward propeller	-
M_2	Number of vortex panels of aft propeller	-
H	Total head	m
Q	Discharge	$\frac{m^3}{s}$
X_f	Axial distance between stages	m
q	Torque ratio	-
C_D	Blade section drag coefficient	-
V_s	Ship speed equivalent velocity	$\frac{m}{s}$
R_{hv}	Hub vortex radius	m
R_h	Hub radius	m
ρ	Fluid density	$\frac{kg}{m^3}$
ITER	Maximum iterations for optimization	-

Table 4.1: Input parameters for CRP design using the coupled method.

5.1 Facilitating user-friendly CRP designs in Matlab

Despite the existence of a CRP design method implemented in Matlab, little or no conscious effort has been made to provide a user-friendly manner of creating new designs. Currently, the existing code is only accessible through the command window of Matlab, with no clear documentation on the needed input parameters, and no way of easily saving, loading and exporting completed designs. A prerequisite for a design method to be useful is that it is made readily available to the intended user. With this in mind, an effort has been made to develop a user-friendly, straightforward and logical fashion of utilizing the coupled CRP design method of Laskos [3]. The focus was put on constructing multiple practical and convenient graphical user interfaces (GUIs) that connect the back end calculations with a front end user. Moreover, the GUIs should provide an easy way of organizing current and previous CRP designs. This is accomplished by enabling the saving and loading of designs. Moreover, it simplifies iterative design procedures where parametric studies can be performed. To sum up, a user-friendly way to input parameters and display results is lacking from the current design code. The functionality of the developed GUIs will be described in this chapter and is highlighted in Appendix B.

5.1.1 Design input

As a method of straightforwardly inputting design parameters for a CRP design, a GUI was developed in Matlab using Matlab's built-in GUIDE framework. The interface was created using drag-and-drop functionality in the GUIDE Layout Editor and can be seen in Figure 5.1. Editable text boxes, check boxes, tables and pop-up menus are used for the inputs of all design parameters. The most expedient method of inputting a value was carefully chosen for each parameter, e.g. a checkbox for selection of hub/no hub or a table input for radially varying parameters such as inflow velocities and chord lengths. Also, the GUI is split into several panels, each one representing a set of coherent parameters. For instance, all the properties common to both propellers are placed in one panel.

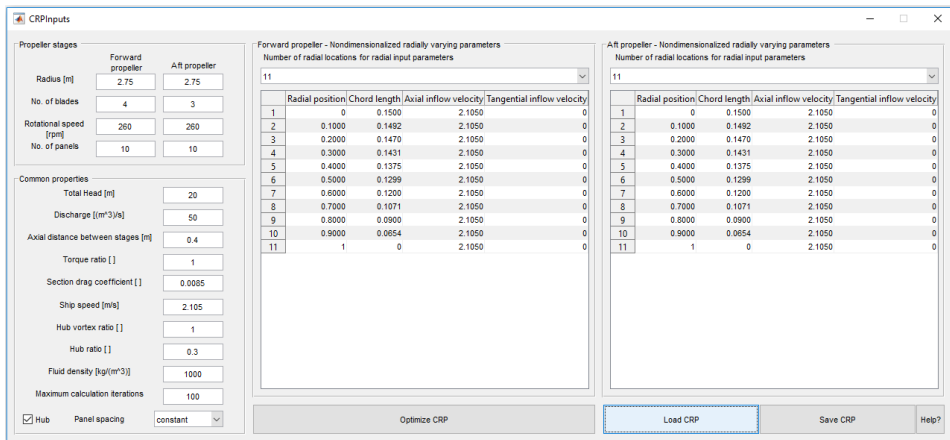


Figure 5.1: Graphical user interface for inputting CRP design parameters.

To make the app work, functionality was programmed in function callbacks triggered by the buttons in the user interface. The Matlab code is included in Appendix C. In the design input GUI, the user can perform multiple actions. Firstly, when design parameters have been set, the user can save the CRP design values to a *.mat* file, so the design can be reused and modified later. By pressing the *Save CRP* button, the user will be prompted to provide a filename and a save directory, and subsequently the *.mat* file will be saved.

A previously created CRP design can be loaded into the GUI by pressing the *Load CRP* button. The user is prompted to choose a file to load, which must be a *.mat* file with parameters that are named in the same manner as they are saved from the GUI's save functionality. The desired parameters to load can be chosen from a list of checkboxes, so the user doesn't have to include all parameters of a previous design. When finishing the loading, all chosen values are entered in their corresponding input locations in the GUI. The user is then free to modify all parameters at will.

The final functionality of the input GUI is to optimize a CRP design for the entered design parameters, and this is performed by pressing the *Optimize CRP* button. Effectively, this performs the optimization method created by Laskos [3], but before anything else, the head requirement of the design is converted into a thrust requirement by Equation 3.11. After the optimization process is completed, another GUI for displaying and analyzing the optimization results is automatically opened.

As support for designers using the user interface, a help button has been added. The sole purpose of this button is to open a *.txt* document containing information about all design parameters and their corresponding units and nomenclature. This is implemented to combat misunderstandings about design inputs. The help button is labeled *Help?*.

5.1.2 Displaying design results

A clear and convenient way of displaying the design results is crucial to a user-friendly design code. Consequently, a practical graphical user interface was developed to display, plot and save results of the CRP design optimization. This GUI was developed in the same way as the one for inputting design parameters, but it has entirely different functionality. The Matlab code of the results GUI is shown in Appendix D. Figure 5.2 shows the user interface, which opens automatically after the CRP optimization is completed.

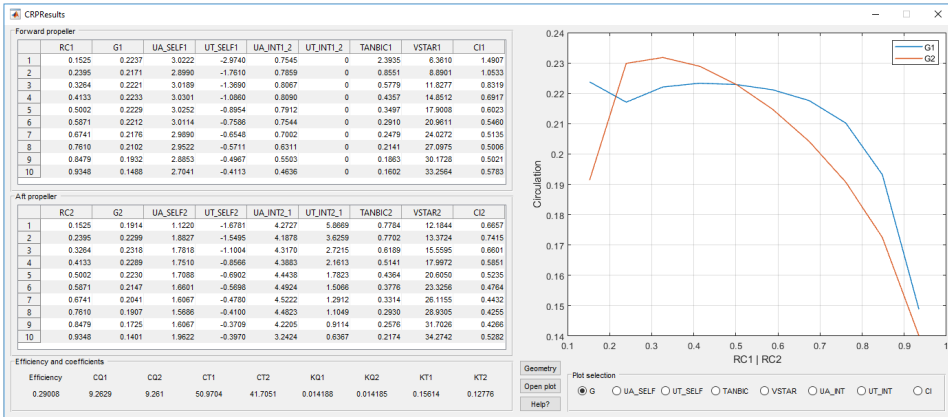


Figure 5.2: Graphical user interface for displaying, plotting and saving design results.

The optimization yields several results, most of which are specific to each of the propeller stages. Consequently, the GUI is split into two main result tables displaying all values for one propeller each. For each propeller, the circulation, self-induced velocities, interaction velocities, total resultant inflow angle and magnitude and lift coefficients are displayed in a table. All these parameters vary radially and are therefore given at all of the control points of each propeller's blades. Furthermore, the CRP efficiency and each stage's thrust and torque coefficients are shown in a separate panel.

All the radially varying properties are conveniently displayed in a plot. The x-axis tracks the non-dimensional radial positions of the control points, and the y-axis tracks the selected property. The desired property for plotting can be selected in the *Plot selection* panel. Having the plot incorporated in the GUI serves the purpose of quickly visualizing values, but post-processing, editing, zooming, exporting and other plot-related actions may be desired by the designer. Thus, the ability to open each plot in a separate, standard Matlab figure was included in the GUI. By pressing the *Open plot* button in the GUI, the selected plot, i.e. the one showing in the incorporated plot, will open in a separate figure. In the separate figure, all default figure actions are available. This includes panning, zooming, exporting, tracking data points, rotating and many others.

The final three-dimensional geometry of the CRP is visualized by surface plots of the propeller stages and a common hub surface plot. This geometry plot is implemented in the

design code of Laskos [3], but upon closing the plot, the optimization must be redone to view it again. For this reason, a *Geometry* button was implemented in the CRP results GUI. Pressing this button re-executes the code creating the 3D propeller geometry plot. The 3D plot opens in a separate figure, so it can easily be exported or modified as desired by the designer.

Similar to the design parameter input GUI, the result GUI has a *Help?* button to clarify all misunderstandings related to the result parameters. A *.txt* file containing information about all parameters, their units and nomenclature will open in a separate window. Furthermore, the expressions for the different torque and thrust coefficients are stated.

Upon exiting the GUI, the user will be prompted to save the results in a *.mat* file. This will save all the parameters, including the design parameters and the current state of the GUI, to a file. Opening this file will reopen the results GUI with all results parameters loaded. Moreover, all parameters will also be loaded into the Matlab Workspace, thus enabling post-processing of the data.

5.2 Design of booster pump for application at Roskrepp

5.2.1 Determining design inputs

A significant part of theoretical axial pump design is choosing the right design parameters. Theoretical assumptions may not be valid for all design parameters. Moreover, optimizing the design of an axial pump is rendered useless if the intended application or requirements are unfit for an axial pump. Therefore, all design parameters should be chosen based on theoretical knowledge and or empirical experience. By doing so, a design will likely have greater performance, and if it doesn't, the designer has a basis for deciding why not. All design parameters used in the coupled CRP design method have been carefully considered. Several of the parameters are interrelated, meaning that they depend on each other. For such parameters, a combination of empirical knowledge from previous CRP designers and short parametric studies have been used to attempt to determine the optimum combination of parameters.

Performance requirements

A set of performance requirements has resulted from a preliminary investigation of retrofitting Roskrepp hydropower plant with pumping capacity. The proposed solution of a booster pump in series with a reversible pump turbine puts forth requirements regarding both lifting height and capacity of the booster pump. CFD simulations of Roskrepp hydropower plant as a PHS plant have been conducted. When replacing the Francis turbine with an RPT with the same dimensions, the RPT produced 20 m too low lifting height in pump mode. Part of this height can be made up for by changing the rotational speed of the RPT and/or redesigning it, but it is believed that a booster pump is needed to supply the remaining lifting height that the RPT cannot provide. The aim was therefore to produce an as high lifting height as possible with an acceptable booster pump efficiency. The rated flow

of Roskrepp, which is $Q = 50 \frac{m^3}{s}$, was maintained for the booster pump design. When an acceptable design was achieved for a certain booster pump head, the design was adjusted to increase the head further with equal or greater pump efficiency. Naturally, the pumping liquid is fresh water, so the liquid density for the design input is set to $\rho = 1000 \frac{kg}{m^3}$.

Blade 2D foil section

The two-dimensional foil sections of the pump blades are equal for both stages of the booster pump when using the coupled design method. When deciding on a blade section for a pumping application, several considerations must be made. Firstly, cavitation will occur easier on blade sections with high thickness ratios [22], i.e. a large ratio of maximum local thickness to chord length. These foils will have higher local velocities on the suction side, which in turn will cause lower static pressure, in accordance with Bernoulli's equation 3.3. Furthermore, a larger maximum camber will cause the same effect. Thin foils with weak curvature are therefore preferred when considering an axial pump's cavitation properties. Another consideration that should be made is regarding blade strength. Thinner blades have lower yield strength, thus making them more prone to mechanical failure. Rake and skew is possible to add for the booster pump blades, but will not be included in this work.

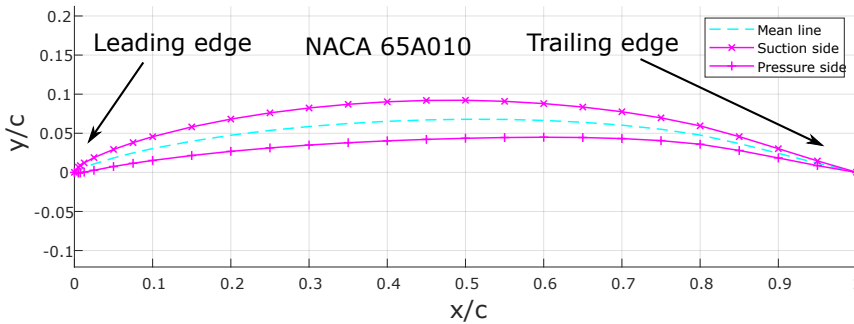


Figure 5.3: Selected CRP design foil section with NACA 65A010 thickness distribution and NACA $a = 0.8$ meanline.

For a two-stage contra-rotating booster pump at Roskrepp, the focus has been put on empirical knowledge when selecting the blade foil section. A foil with NACA 65A010 thickness distribution and a NACA $a = 0.8$ mean line was chosen. This is a foil that is widely adopted for marine applications such as propellers on large ships [26], especially due to its flat suction side pressure distribution. This pressure distribution leads to wide cavitation-free areas. The NACA $a = 0.8$ mean line is regarded as well suited for propeller applications as it has a lower effect of viscosity on lift than other mean lines [13, p.449]. The digits in the thickness distribution reveal that the foil is part of NACA's 6-series of

foils, has minimum pressure at 50% of the chord, maintains laminar flow over 80% of the chord, has a design lift coefficient of $C_L = 0$ and has a maximum thickness of 10% of the chord. The foil is modified from its original shape, and the letter A indicates that another thickness distribution and mean line are actually used. The selected foil section is seen in Figure 5.3.

Unfortunately, no drag coefficient data has been obtained for the selected NACA 65A010 foil. However, Kravitz [26] has used $C_D = 0.01$ for a marine contra-rotating propulsor design using the same foil section. Foil section drag coefficients tend to vary less at higher Reynolds numbers, and furthermore, they generally decrease as the Reynolds number rises. The Reynolds numbers will be significantly larger in the booster pump than in the marine propulsor design of Kravitz since both the chord lengths and rotational speeds will be larger. Increasing these parameters will increase the Reynolds number. Therefore, it is considered a safe assumption that the drag coefficient will be less than or equal to the value used by Kravitz, and it is therefore set to $C_D = 0.01$.

Radii of propeller stages

For a CRP operating in open water, the aft propeller should have a smaller radius than the forward one due to slipstream contraction [3]. Tip vortices from the forward propeller impinge on the aft propeller, creating a need for the aft propeller to be smaller. However, the coupled CRP design method assumes that the streamlines do not contract, and the radii of the two booster pump stages should therefore be equal. This decision is reinforced by the CRP application being a booster pump, i.e. a ducted configuration.

The radius of an axial impeller affects the optimum efficiency greatly. In general, the maximum efficiency increases with an increasing radius [29], however, a peak efficiency will be reached for a finite radius [3]. For a marine propeller, the efficiency will approach 100% in an ideal fluid when the radius approaches infinity. This is because as the mass flow increases, a required increase in momentum leads to a smaller and smaller increase in speed. The radius of the booster pump application at Roskrepp is limited by the discharge tunnel at the proposed pump placement. Without having to excavate a larger tunnel, the maximum radius possible is $R = 2.75\text{ m}$. Setting the booster pump radius to this large value will make tip speeds very high, and consequently, pressures will be very low near the impeller tip. With the cavitation limitations posed by Equation 3.7, the maximum compliant rotational speed will be restricted by the booster pump radius. The booster pump radius was set as part of an iterative procedure where the rotational speeds, pump radius and hub ratio were adjusted to increase efficiency and meet cavitation requirements. The optimum booster pump radius was found to be $R = 1.625\text{ m}$.

Rotational speeds

A primary concern when determining the rotational speeds of the propeller is the cavitation restrictions. If the diameter and hub ratio are set, then the maximum rotational speed of an axial pump is given by combining Equations 3.6 and 3.7 and solving for the rotational speed. The resulting equation for the rotational speed in rpm becomes:

$$n_{max} = \frac{60}{2\pi} \frac{1}{R} \sqrt{\frac{2g}{b} \left(\text{NPSH}_A - a \frac{c_m^2}{2g} \right)} \quad (5.1)$$

Brekke [1] states that the empirical coefficients a and b are $a = 1.8$ and $b = 0.23$ at the onset of cavitation for a pump. The available net positive suction head, NPSH_A , is computed for the conditions at Roskrepp. The barometric pressure head at the pump elevation is $h_b = 9.3136 \text{ m}$, the vapor pressure head of water at 10°C is $h_{va} = 0.125 \text{ m}$ and the minimum submergence is $h_{sub} = -3 \text{ m}$. The meridional velocity component c_m at the pump inlet is equal to the axial inflow velocity ($c_m = V_a = V_{avg}$) and can be computed from the flow rate, Q , the selected hub ratio and the pump diameter. With these values, the available net positive suction head at the pump inlet becomes $\text{NPSH}_A = 11.779 \text{ m}$. Equation 5.1 then yields a maximum rotational speed of $n_{max} = 181 \text{ rpm}$. This value was finally selected as the rotational speed of the pump as a result of the iterative process of determining a suitable hub ratio, diameter and rotational speed.

Hub properties

Based on axial machines of similar dimensions, a hub should be present for the booster pump at Roskrepp. The design will be of a considerable size, and large thrust forces will be exerted on the pump blades, so a hub should be included to assure mechanical integrity during operation. Several hub-less marine propulsors are available for ships of considerable size, but the main argument for removing the hub is a reduction of noise and vibration, which will be of less importance than structural integrity in a PHS plant. Single impeller hub-less designs have been shown to have slightly higher hydrodynamic efficiencies [30], but little research exists on hubless contra-rotating configurations.

When selecting the diameter of the hub, several considerations must be addressed. Primarily, the blade outlet angle, angle β_2 in Figure 5.5, at the hub should be less than 90° [22]. This is to ensure that the tangential velocity component at the outlet of the forward impeller is pointing in the same direction for all radii. Secondly, the attainable head, i.e. the maximum possible head, from an axial impeller is limited by the hub ratio. Increasing the hub ratio will increase the attainable head. Regarding efficiency, increasing the hub ratio will slightly increase the peak efficiency, but at the cost of lower efficiency at off-design conditions [31]. Empirical data for determining the hub ratio exist for single stage machines, and Stepanoff [4, p.145] published a diagram for selecting an appropriate value based on the vane spacing, $\frac{Zc}{\pi D}$, and the specific speed. The diagram, which is reproduced in Figure 5.4, shows the appropriate hub ratio (marked with a red circle) according to the selected diameter and specific speed. As this value for the hub ratio produced blade exit angles higher than 90° ($\beta_2 > 90^\circ$) at the hub, the selected hub ratio was increased to $\frac{R_{hub}}{R} = 0.5$ (marked with a blue circle).

A concentrated vortex will emanate from the hub, terminating infinitely downstream of the hub. This hub vortex will cause a drag force on the hub which decreases the efficiency [3]. The efficiency decreases with decreasing diameter of the vortex core. The drag force on the hub will also increase with decreasing vortex core diameter, however, the variation

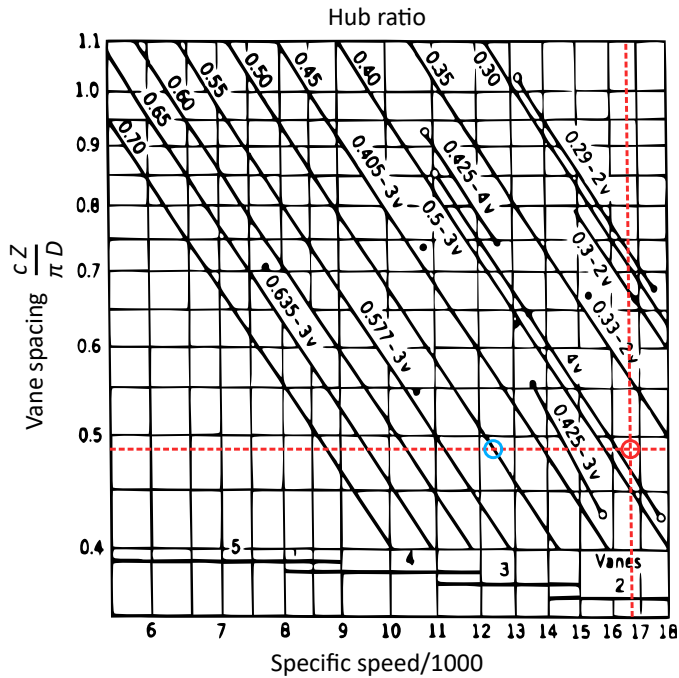


Figure 5.4: Diagram for selection of hub ratio. Reproduced from [4]. Here, the dimensional specific speed is calculated with imperial units as $N_s = \frac{n\sqrt{Q}}{H^{0.75}}$.

in drag is small if the hub diameter is large enough [32]. Laskos [3] states that the hub vortex ratio is set to $\frac{R_{hv}}{R_h} = 1$ if a hub is present, which will be used in this work as well.

Blade numbers

Often, the blade number of an axial machine is chosen based on avoiding damaging resonant frequencies of the intended application [13]. No such frequencies have been investigated at Roskrepp for this work, and thus, the number of blades is chosen on a different basis. Generally, increasing the number of blades will decrease both the efficiency and the optimum impeller diameter. The blade loading will, however, increase with fewer blades. Diagrams, such as the one shown in Figure 5.4, are available for empirically determining the blade number of single impellers. Based on the selected hub ratio and the diagram presented by Stepanoff, the forward impeller was chosen to have 3 blades (called vanes in the diagram). For contra-rotating configurations, the efficiency increases with the number of blades on the aft impeller, but for higher rotational speeds a lower number of aft impeller blades gives the highest optimum efficiency [29]. The aft pump stage was therefore chosen to have 2 blades, which was proven to yield higher efficiency than 3 blades for the selected design parameters.

Inflow velocities

The inflow velocities are required for each of the two booster pump stages, and neither the self-induced velocities nor the interaction velocities are included in the inflow. All induced velocity components are accounted for in the optimization instead. Since the inflow will vary radially, the inflow velocities are given at a finite number of radial positions and assumed constant in the circumferential direction. A common design assumption is that the inflow of the forward impeller only has an axial component, which is constant for the entire cross-section of the pipe. That is to say, the axial velocity does not change in the radial or circumferential direction. For the inflow of the aft impeller, an approximation of the flow leaving the forward impeller must be estimated. Such an approximation can be found by knowing the blade section, rotational speed and inflow of the forward impeller.

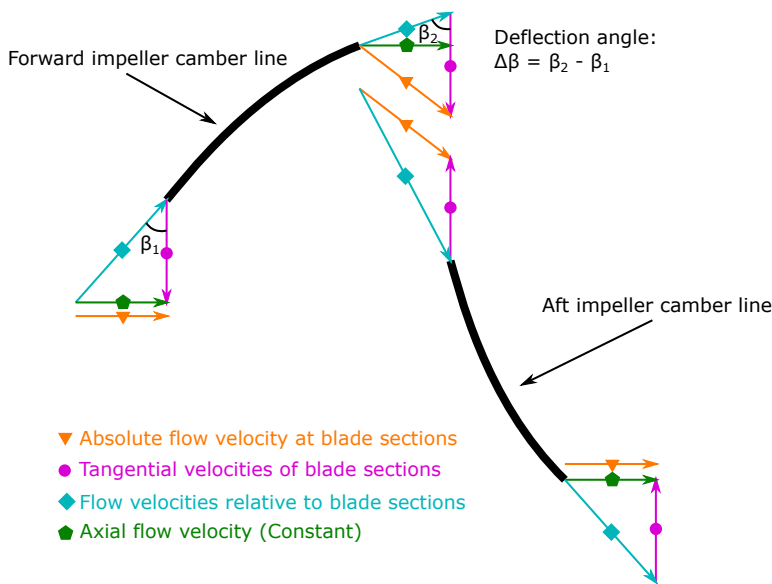


Figure 5.5: Blade cascade for booster pump illustrating the blade deflection angle and how the forward impeller determines the inflow velocity of the aft impeller.

The blade section of the forward impeller yields the deflection angle, $\Delta\beta$, which describes how many degrees the flow turns when flowing through the impeller, i.e. the difference in slope of the camber line at the leading and the trailing edge of the blade profile. The flow is assumed to enter and leave the impeller blade at the angle of the camber line at the leading and trailing edge respectively, as shown in Figure 5.5. For the blade section used in this work, the deflection angle is $\Delta\beta = 37.4^\circ$. Since the mass flow is constant, and the fluid is water, the axial velocity component will remain unchanged through the booster pump. The axial velocity is computed by dividing the volumetric flow rate, Q , with the cross-sectional area between the impeller shroud and the hub. The velocity approaching the forward impeller only has an axial component, but the blade section will move in the tangential direction with a velocity of $U = \omega r$, where ω is the angular velocity of the blade

section and r is the section's distance from the axis of rotation. These two velocity vectors give the relative inflow velocity seen from the blade and the relative inflow pitch angle, β_1 in Figure 5.5.

The flow leaves the forward impeller along the direction of the camber line at the trailing edge, which is oriented at a pitch angle of $\beta_2 = \Delta\beta - \beta_1$. The velocity magnitude of the water leaving the impeller is so that the axial velocity remains constant. By again accounting for the blade velocity, U , of the forward impeller, the absolute flow velocity leaving the forward impeller is computed. This velocity is given as inflow velocity for the aft impeller for the optimization. To sum up, the inflow parameters are given at certain radial sections and are the absolute velocities approaching the impeller stages without effects of circulation or slip at the blades.

Axial distance between stages

The axial distance, X_f , between the two stages for contra-rotating propellers have been assessed by several studies. Kravitz [26] found that the effect of the distance between propeller stages on CRP efficiency was almost negligible. He utilized the method of Laskos for the design of a contra-rotating marine propulsor. Similarly, Coleman [33] states that the distance between coaxial helicopter rotors is not very noticeable for practical operation. Tang [29] concluded that the individual performance of each propeller stage was greatly affected by the axial distance, but that the total CRP efficiency was only slightly affected. He attributed the individual propellers' performance variations to the induced interaction velocities. Contrary to the aforementioned studies, Roy [34] et al. concluded that the optimum distance between a set of ducted, contra-rotating, axial fans is 50% of the chord length of the forward fan stage. Doubling the axial distance to 100% of the first stage's chord length leads to a decrease in fan efficiency of 10%. With this in mind, a small parametric study of the axial distance was conducted for the booster pump. The result showed an up to 1% increase in efficiency with the optimum axial distance between the stages. The optimum value, which was used in the final design, was found to be $X_f = 0.5 m$.

Torque ratio

Since the two stages will be of equal size and have the same hub ratio, the torque ratio is set to $q = \frac{\tau_2}{\tau_1} = 1$, i.e. the torque acting on each pump stage will be the same. Tang [29] found that the maximum efficiency of a set of contra-rotating propellers is achieved for a torque ratio of $q = 1$. He proposed that when a CRP set operates with equal torque and rotational speed on both propellers, the tangential interaction velocity induced on the aft propeller exactly counteracts the self-induced velocity of the aft propeller, thus increasing the efficiency. Furthermore, he argued that the aft propeller can recover the rotational energy losses caused by the forward propeller more efficiently under equal torque conditions. Selecting equal torque ratio and rotational speed for both stages also reduced the complexity of pump motor design, which is a convenient design bonus.

Ship speed

The ship speed is not an appropriate parameter for pump applications. It is utilized in marine propeller design because the velocity of a ship often is larger than the axial inflow velocity seen by the propeller, an effect caused by the wake of the ship's hull. For this work, the ship speed is assumed to be equal to the axial inflow velocity of the booster pump. The theoretical design in this work assumes a completely axial inflow velocity with no radial or circumferential variation, which is the same inflow conditions an open water ship propeller would encounter with no influence of the ship hull. For this reason, setting the ship speed equal to the axial inflow velocity is considered a fair assumption. This assumption is also used by Tang et al. [29] for CRP design.

Optimization scheme parameters

The maximum number of iterations of each of the converging processes is set to $ITER = 50$. This has proven to be a reasonable number, both with respect to being high enough to achieve convergence in most cases and regarding computational time. The panel spacing is set to constant, meaning that each vortex panel is of equal length for a propeller stage. The alternative is to have cosine spacing, where the panels vary in length with a cosinusoidal nature. The constant panel spacing is chosen based on convergence problems with the cosine spacing experienced by Laskos [3].

Chord distributions

Radial chord distributions for axial machines vary greatly for different applications. Wind turbines have very long blades with short chord lengths, fans often have more or less constant chord lengths from hub to tip and propellers and pumps usually have wider blades with chord lengths increasing radially. Longer chord lengths usually yield more rigid blades that are less prone to cavitation on the suction side, a result of reducing the blade loading [26]. Blades with longer chord lengths are, however, associated with larger drag forces, as can be seen from Equation 3.1, and thus, they often have lower efficiencies than blades with shorter chords. Therefore, a compromise between strength and/or cavitation limitations and performance is often the preferred design.

No strength calculations have been conducted for the booster pump design, which is common for preliminary designs made with lifting line theory, so the chord lengths have been chosen on a basis of performance, with a focus on designing realistic blade shapes for an axial pump. Güllich [22] states that the chord lengths of an axial pump should be so that the blade's leading and trailing edges are smooth, i.e. no abrupt changes in chord length or high curvature along blade edges should occur. Additionally, he proposes a third-degree polynomial with appropriately chosen coefficients for the radial variation of blade chord. In this work, a second-degree polynomial was chosen instead. Güllich also proposes that the ratio of the chord length at the hub to the tip chord length should be around 0.7 to 0.9. For the Roskrepp booster pump design, this ratio was set to 0.6 as it yielded higher efficiency. The optimum chord length distribution tends to differ for the forward and aft pump stages, and the aft impeller should have slightly shorter chords than the forward impeller

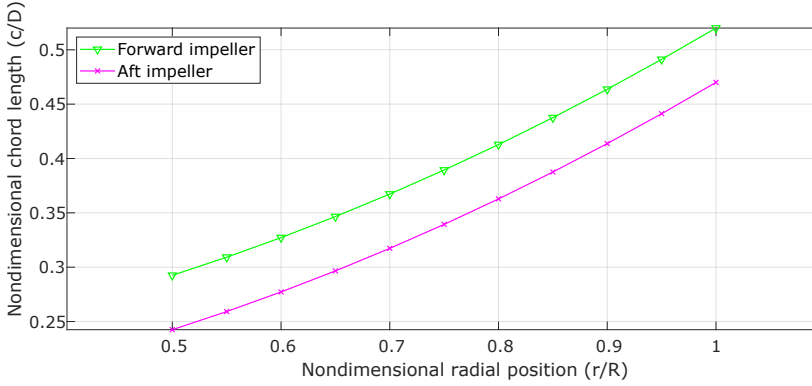


Figure 5.6: Chord distributions for both impeller stages from impeller hub to tip. Radial positions nondimensionalized by tip radius and chord lengths nondimensionalized by impeller diameter.

[29]. The blades of both impellers should, however, have similar shapes. For this reason, the aft impeller chord lengths were designed a bit shorter, but with a similar shape as the forward impeller. The radial chord length distributions for both impeller stages are shown in Figure 5.6. The second degree polynomials describing the chord distributions $c(r)$ for the forward and aft impellers are given in Equations 5.2 and 5.3 respectively.

$$\frac{c_1(r)}{D_1} = 0.25 \left(\frac{r}{R_1} \right) + 0.27 \left(\frac{r}{R_1} \right)^2 \quad (5.2)$$

$$\frac{c_2(r)}{D_2} = 0.2 \left(\frac{r}{R_2} \right) + 0.27 \left(\frac{r}{R_2} \right)^2 \quad (5.3)$$

5.2.2 Design Iterations

By using the developed GUI, the iterative process of designing the axial contra-rotating booster pump was efficient and effortless. As previously stated, several of the design parameters are interrelated, and time was therefore spent experimenting with the effects of changing these parameters. The primary focus was put on increasing the pump efficiency, with a secondary objective of increasing the head of the pump as much as possible. During the design process, the design of an optimum impeller was prioritized over practical solutions regarding the entire booster pump configuration at Roskrepp. To elaborate, no attention was paid to components such as the converging-diverging duct needed to lead the flow of water through the pump or the motor of the pump. Seeing that this is a preliminary design, these aspects of the pump configuration are of less importance than investigating if a booster pump can be expedient for retrofitting Roskrepp with pumping capacity.

RESULTS AND DISCUSSION

6.1 Preliminary booster pump design

The final contra-rotating booster pump design provides a pressure head of $H = 6.2 \text{ m}$ at a discharge of $Q = 50 \frac{\text{m}^3}{\text{s}}$, and has an efficiency of 59.7%. The efficiency and the torque and thrust coefficients for each stage are shown in Table 6.1. Figure 6.1 shows the geometry of the designed contra-rotating booster pump, and more images of the geometry can be seen in Appendix A. The forward and aft stages have 3 and 2 impeller blades respectively, with the leading edges plotted with a blue line. The green blade surfaces face the incoming flow and the red surfaces are directed downstream of the pump. For the use at Roskrepp hydropower plant, with the intended pump placement, the green surfaces will face the discharge tunnel and the red will face the draft tube.

η_h	59.7%
$K_{Q,1}$	0.0397
$K_{Q,2}$	0.0397
$K_{T,1}$	0.1917
$K_{T,2}$	0.2260

Table 6.1: Efficiency of booster pump and thrust and torque coefficients for each stage. Results obtained by the coupled CRP design optimization.

Ideally, a contra-rotating axial machine could have an efficiency of above 80%, but for too low advance ratios of the aft impeller stage, the peak efficiency will be much lower [26]. The advance ratio of the forward impeller is also of significance for the efficiency. Rotational losses in the impeller will dominate over viscous losses for high advance ratios. It is believed that the relatively low efficiency of the booster pump is due to a too low advance ratio for both impeller stages. An obvious way of increasing the advance ratio is to lower the rotational speed of the pump stages, but this brings several unwanted consequences.

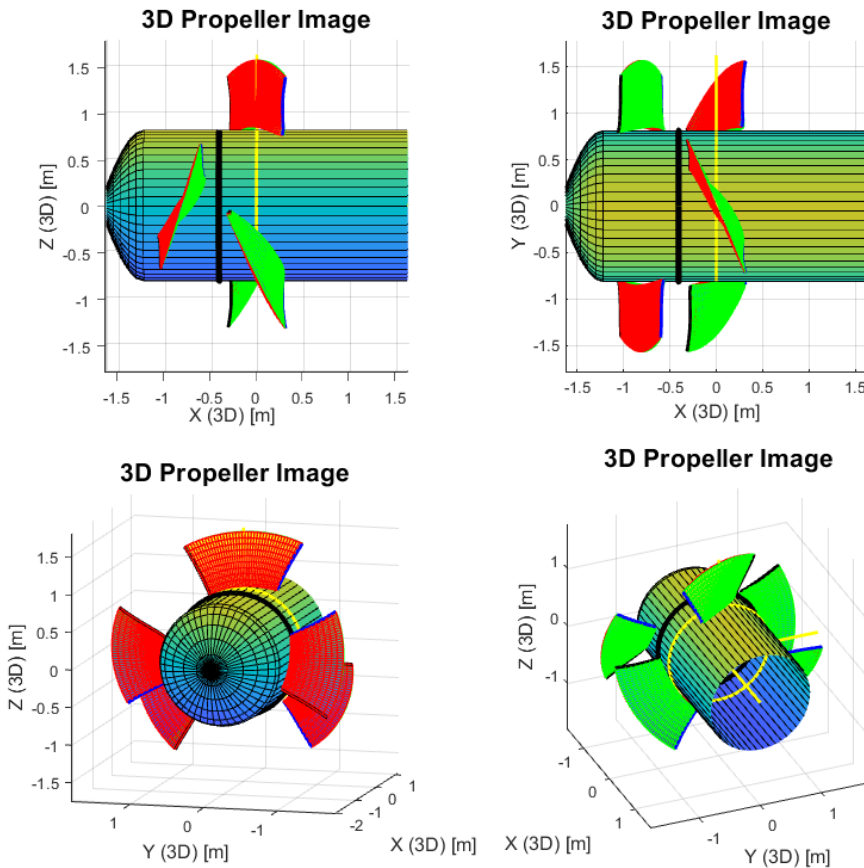


Figure 6.1: 3D geometry of booster pump seen from multiple angles.

First of all, the rotational speed is set high to avoid a blade outlet angle above 90° at the hub of the first impeller. Lowering the rotational speed would force the hub ratio to be raised to remain compliant with this outlet angle requirement. By increasing the hub ratio, the blade span is shortened accordingly, and the blade loading will increase. Naturally, this can cause unwanted cavitation and/or structural problems, however, no assessment of these problems has been conducted, thus the severity of the problems remains uncertain. Secondly, the lifting height provided by the booster pump is restricted by the rotational speed. For this reason, setting the rotational speed too low might render the booster pump useless, i.e. there exists a lower limit for the lifting height of the booster pump where it no longer serves its purpose. Another way of increasing the advance ratios of the impellers is to increase the volumetric flow rate, Q , but since this is a design requirement at Roskrepp, no changes have been made to this parameter.

Ideally, the aft impeller of a contra-rotating pump removes all kinetic swirl energy from the forward stage by converting it into pressure energy. For the final booster pump design, this is not the case. There is a significant swirl at the outlet of the aft impeller, which can be seen in Figure 6.2. The results clearly show a linearly increasing swirl velocity distribution from the hub to the tip, which indicates that the rotational velocity of the aft stage is suboptimal. Swirl at the pump outlet is considered energy loss, and the hydraulic efficiency decreases when the swirl increases. A set of exit stay vanes could be a suitable way of converting the swirling kinetic energy to pressure energy. This would add complexity to the booster pump design, which makes the design process more difficult. Another manner of removing the swirl is to change the foil section of the aft impeller to a section with a different deflection angle. A lower deflection angle would cause the flow to leave the aft impeller blades at a more favorable angle, given the present rotational speed.

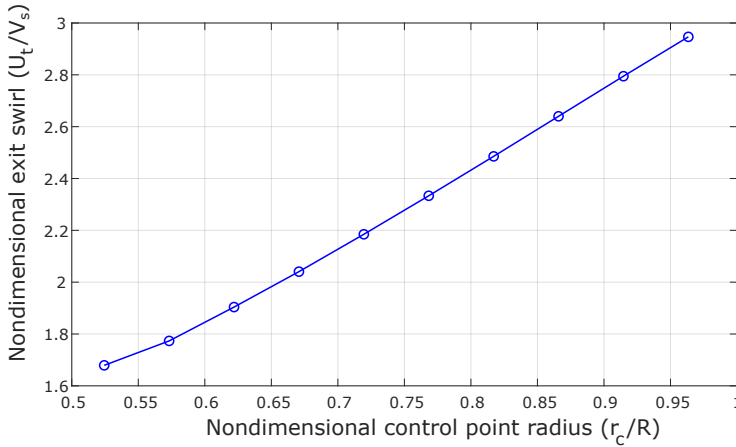


Figure 6.2: Swirl velocity distribution from hub to tip at outlet of aft impeller stage.

In spite of a slightly low booster pump efficiency, the total efficiency of the booster pump and RPT pumping in series can still be acceptable. Since the booster pump only provides a small fraction of the needed lifting height to pump water to the upper reservoir, its efficiency will be of less importance than the RPT's on the total efficiency. For Roskrepp, the elevation difference of the two reservoir surfaces is approximately $H_{tot} = 105 \text{ m}$ when the upper reservoir is at its highest regulated water level and the lower reservoir is at its lowest water level for normal operation. Assuming that a pressure head corresponding to this elevation difference needs to be provided by an RPT pumping in series with the designed booster pump, the RPT would need to provide $H_{RPT} = 98.8 \text{ m}$ of pressure head. Typically, RPTs have a pump efficiency in the range 90 – 92% [35]. Assuming an efficiency of $\eta_{h,RPT} = 90\%$, the shaft power of the RPT becomes $P_{RPT} = \frac{\rho g H_{RPT} Q}{\eta_{h,RPT}} = 53.846 \text{ MW}$. The required shaft power of the booster pump can be calculated from the torque coefficients of the design optimization, $K_{Q,1}$ and $K_{Q,2}$, by using Equation 3.14. This yields a

total booster pump power of $P_{booster\ pump} = 4.969\ MW$. The total hydraulic efficiency for the pumping system is then calculated by Equation 3.13.

$$\eta_{h,tot} = \frac{\rho g H_{tot} Q}{P_{RPT} + P_{booster\ pump}} = \frac{1000 \frac{kg}{m^3} \cdot 9.81 \frac{m}{s^2} \cdot 105m \cdot 50 \frac{m^3}{s}}{(53.846 + 4.969) \cdot 10^6 W} = 87.57\% \quad (6.1)$$

Because the booster pump only is required to deliver a small fraction of the total lifting height, the poor efficiency is of little importance for the entire pump system. 59.7% booster pump efficiency can therefore be regarded as acceptable if the RPT cannot be designed to provide the 6.2 m of pressure head provided by the booster pump on its own.

The thrust and torque coefficients of the two impellers are both within the range of expected values [12], which usually means that a high efficiency is achievable. Despite this, a too low or high advance ratio could decrease the efficiency.

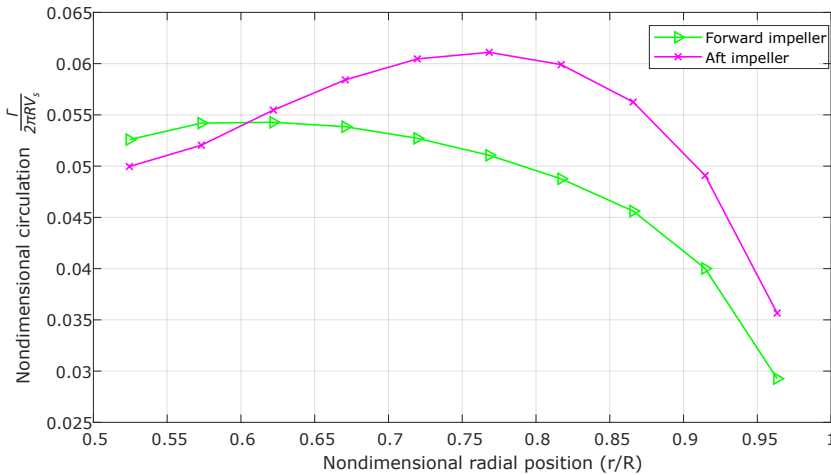


Figure 6.3: Nondimensional circulation distribution from impeller hub to tip for both impeller stages.

Equation 4.4 shows that the inviscid lift force used in the calculation of thrust and torque on the impeller is proportional to the circulation around the foil. Hence, the circulation distribution along the span of the blades is a good indicator of the blade load distribution. Figure 6.3 shows the radial circulation distributions of the two impeller stages. For the forward impeller, the circulation is decreasing with the shape of a second-degree polynomial towards the tip of the blade, where it approaches zero. For a finite wing, the pressure difference between the suction and pressure side is zero at the tip, and consequently, the lift at the wing tip must be zero. From Equation 4.4, it is evident that the circulation needs to be zero at the wing tip as well, so this is an expected result. The circulation distribution of the aft impeller blades has a different shape. It increases towards the center of the blade

span before it starts decreasing towards the blade tip. Even though the torque is equal for both impellers, the forces acting on them are very different. This can also be seen from the thrust coefficients presented in Table 6.1. A prevailing reason for this is that the aft impeller has higher velocities flowing across it than the forward impeller. This is shown by the resultant inflow velocities plotted in Figure 6.4. Substantial swirl velocities are created by the forward impeller in the opposite direction of the aft stage's rotation. Reducing the aft impellers rotational speed will lead to an even load distribution between the two stages, which is desired because the blades of both impellers are fairly equal in shape and size.

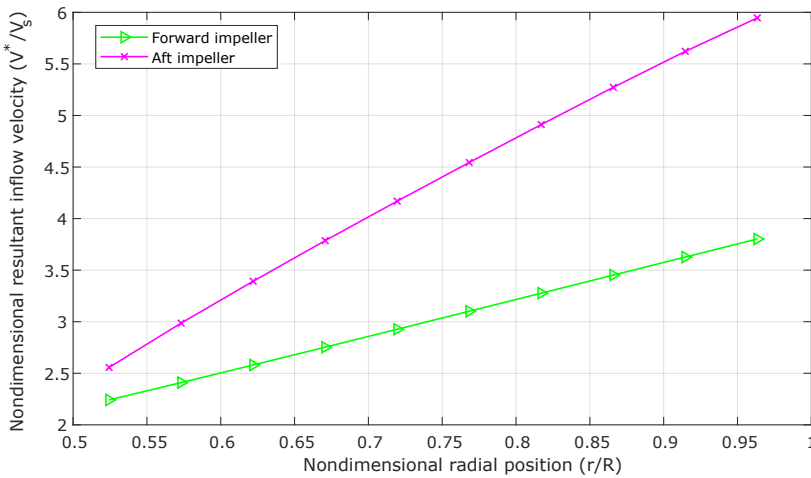


Figure 6.4: Radial distribution of the total resultant inflow velocities of each impeller stage.

The induced velocities are of small magnitudes compared to the resultant inflow velocities. Figures 6.5 and 6.6 show the radial distributions of self-induced and interaction velocities, respectively. For the induced inflow of both impellers, it is clear that the self-induced axial velocities are of greatest importance. The distributions of these show an almost linear increase from hub to tip. The tangential self-induced velocities are of less importance for the total inflow, and they have no common trend.

Figure 6.6 clearly shows that the forward impeller affects the aft impeller way more than the converse. Axial velocities induced on the aft from the forward impeller are greater near the hub, and they decrease progressively towards the tip. The axial interaction velocities of the forward impeller are almost constant along the blade span and of very small magnitude. Similar results are shown for the tangential interaction velocities, but the aft impeller induces no tangential interaction velocities on the forward impeller at all. To sum up, the forward impeller clearly induces significant velocities on the aft impeller, whereas the aft impeller nearly doesn't affect the forward one. Contrary to this, the axial self-induced velocities affect both impellers, and the magnitudes of these velocities are higher at the blade

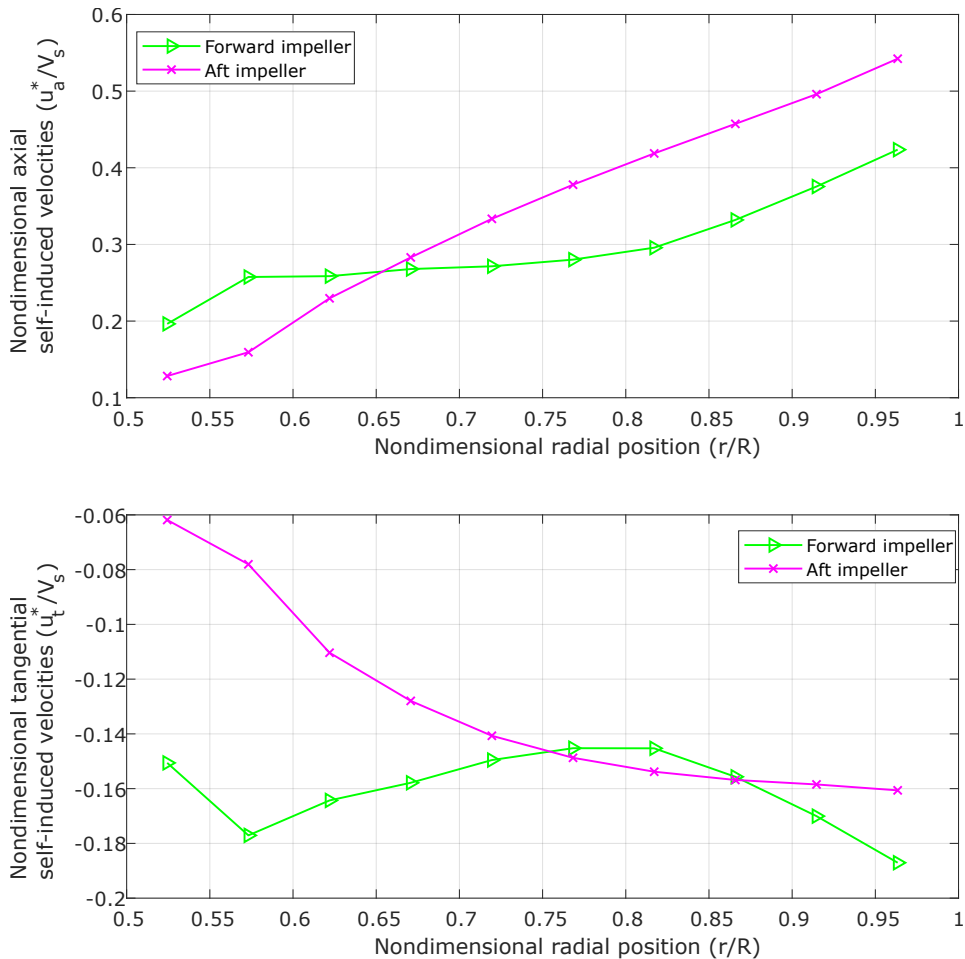


Figure 6.5: Radial distribution of axial and tangential self-induced velocities for both impellers.

tips.

6.2 Relationship between propellers and pumps

One of the assumptions made in the proposed booster pump design method is that the useful power for propellers and axial pumps are equivalent, but this assumption is not always valid. The optimization of a propeller aims to minimize the torque required to produce a certain thrust force, whereas optimization of a pump is concerned with minimizing the torque required to increase the flow pressure a certain amount. These goals are not necessarily coincident. A marine propeller, for instance, will produce a large thrust force without any increase in flow pressure. Instead of pressure energy, swirling kinetic energy is given to the flow. The useful power of a pump and a propeller is therefore different, and

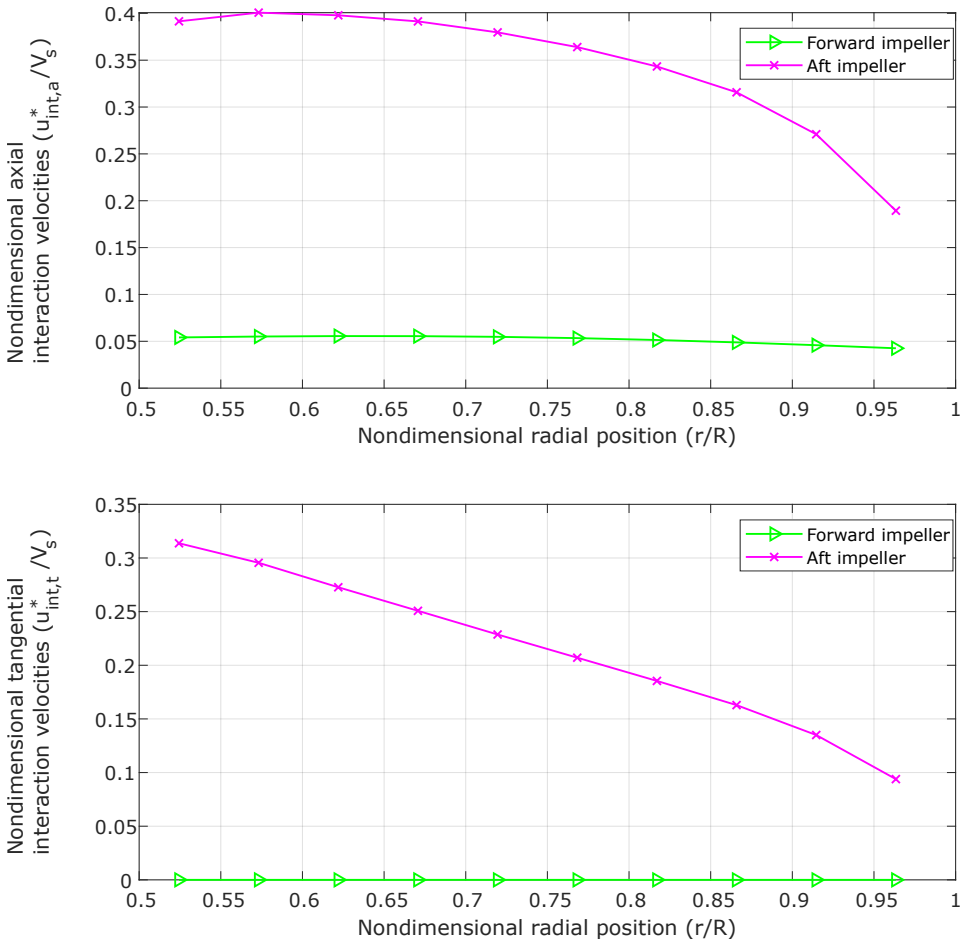


Figure 6.6: Radial distribution of axial and tangential interaction velocities.

their efficiencies are therefore calculated differently. Propeller efficiency is concerned with the thrust force produced by the propeller since this is what matters for propulsion. Pump efficiency, on the other hand, cares about the pressure energy given to the flow. For this reason, the assumption of Equation 3.11 is only valid for contra-rotating pumps of very high hydraulic efficiency, i.e. pumps with almost no exit swirl. When all swirl is removed, the useful power will be equal for both a propeller and a pump, and their efficiencies will therefore also be equal. The proposed booster pump design is therefore created with the assumption of no swirl at the exit of the aft impeller. As seen by the swirl at the aft impeller exit in Figure 6.2, this is clearly not a valid assumption. The booster pump design should therefore be modified to remove the majority of the swirl at the aft impeller exit.

Equation 3.11 can be modified to account for lower hydraulic efficiency pumps as well. Stepanoff [4] stated Equation 6.2 for the axial thrust of an axial pump, which includes the

hydraulic efficiency, η_h . If this equation was to be used with the coupled CRP method, a hydraulic efficiency would have to be assumed before designing. The design process would need another iterative loop checking for convergence of the final design efficiency to the assumed efficiency. Unfortunately, Equation 6.2 is not physically realistic, except for very high hydraulic efficiencies. If the efficiency of the pump abruptly drops to zero, which can happen e.g. if the pump goes to runaway, the equation indicates that the thrust force will go to infinity. This is not physically realistic. Therefore, both methods of calculating the thrust of an axial pump from the produced head are only valid for high-efficiency designs. It is likely that the proposed booster pump design will deliver a lower head than the required input head value. A more accurate design can be achieved if the hydraulic efficiency is sufficiently raised.

$$T = \frac{\rho g Q H}{V_s \eta_h} \quad (6.2)$$

6.3 Booster pump configuration at Roskrepp hydropower plant

With the initial booster pump requirement of 20 m of head at a discharge of $50 \frac{m^3}{s}$, the proposed design will not be expedient. It is, however, likely that the required lifting height is estimated too high. The head requirement was created by simulating an RPT in pump mode with the same main dimensions and rotational speed as the currently installed Francis runner. Because of the fixed rotational speed, a head deficit of 20 m was identified. Since no cavitation was found when conducting this simulation, there is no reason why the rotational speed cannot be increased. If this is done, the lifting height of the RPT in pump mode can be raised, thus lowering the head requirement of the booster pump.

The current turbine at Roskrepp is a relatively low-head Francis turbine. As with most low-head Francis runners, there is some space between the runner blades and the inlet guide vanes. For high-head runners, the blades are usually longer, and the space between the blades and the guide vanes is therefore smaller. When designing an RPT for Roskrepp, the blades can be elongated into this space. Figure 6.7 illustrates this concept. This will also increase the lifting height of the RPT, which further lowers the head requirement of the booster pump.

Even though no cavitation was found in the initial simulations of the RPT, cavitation is likely to occur when the rotational speed is sufficiently increased. For this reason, a booster pump is presumably useful at Roskrepp hydropower plant after all. It is normal for high-lift pumps to be assisted by a slowly rotating pump with a lower required net positive suction head. The assisting pump will increase the available net positive suction head for the high-lift pump, which allows it to operate at higher speeds without cavitating.

Due to the diameter of the designed booster pump being smaller than the draft tube dimensions at Roskrepp, a converging-diverging duct must be created to lead the water through the axial pump. It is unknown how this will affect both the inflow of the booster pump as

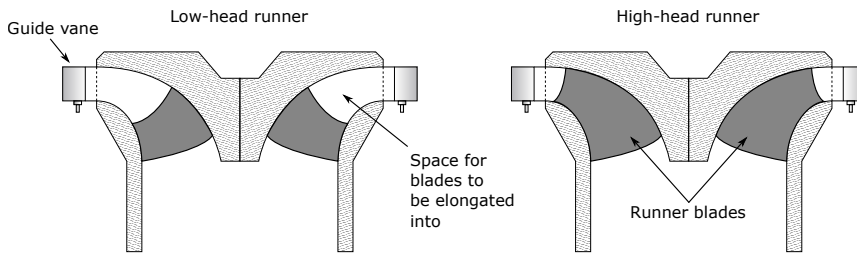


Figure 6.7: Illustration of the difference between a high-head and a low-head runner. Elongating the blades will increase the lifting height of a centrifugal pump or an RPT.

well as the power production when operating the RPT in turbine mode. CFD simulations of the duct effects should be conducted if a final design is decided upon. For operation in turbine mode, a possible solution is to make the booster pump retractable, i.e. facilitate moving the pump out of the waterway. If the booster pump is made to be rim-driven, like a permanent magnet thruster, no driving shaft is necessary to apply torque to the pump. This makes retracting the pump much easier. Such a configuration is already used by several propulsor design companies. A rim-driven pump is possible to design with the developed method, however, the blades need to be mounted on a surrounding ring and will therefore have different chord distributions. Another possibility is to let the booster pump go to run-away speed or even using it as a contra-rotating straight-flow turbine for power generation. These problems are, however, not important to solve in a preliminary design process.

CONCLUSIONS

A user-friendly method of designing contra-rotating axial pumps was developed in Matlab by modifying the coupled CRP design method proposed by Laskos [3]. Graphical user interfaces were created to facilitate a straightforward design process and post-processing of the results. The developed method permits an iterative design process and rapid parametric studies for minor design changes, which proved to be very useful for the design of a booster pump.

A preliminary design of a booster pump for Roskrepp hydropower plant was developed using the aforementioned method with a set of carefully chosen design parameters. The hydraulic efficiency of the proposed design was lower than what can be expected of an optimum contra-rotating axial machine. Despite this, the total pumping efficiency of the booster pump and an RPT pumping in series is still acceptable since the booster pump only needs to supply a small fraction of the total lifting height. High swirl velocities at the aft impeller outlet are believed to cause the low booster pump efficiency. By lowering the advance ratios of the impellers, the outlet swirl velocities are expected to decrease, and the efficiency is expected to increase. Doing this could, however, limit the attainable head of the booster pump, which may render the pump useless for its intended purpose.

The aft impeller displayed a higher blade loading than the forward one, which is believed to be caused by higher flow velocities across the blades. Because the blades are similar in shape and have the same foil section, the blade loading should ideally be relatively equal for the forward and aft impellers. Reducing the aft impeller's advance ratio, i.e. reducing its rotational speed, will decrease the flow velocities across the aft impeller blade. This will lead to a lower blade loading.

Induced velocities were shown to be of greater importance for the aft impeller than the forward one. Both impellers showed significant self-induced velocities in the axial direction, but the tangential self-induced velocities were of less importance. The forward impeller was shown to affect the aft impeller greatly, whereas the aft impeller almost doesn't induce velocities on the forward impeller at all.

Due to differences in propeller and pump design optimization, the proposed design method is only valid for contra-rotating pumps with very little swirl at the aft impeller outlet, i.e. high-efficiency pumps. For pumps with high hydraulic efficiency, the axial thrust force on the pump is related to the delivered head by Equation 3.11. As the proposed design has significant swirl at the aft impeller outlet, the booster pump is expected to deliver a lower pressure head to the flow than it is designed for. Effort should be put into designing a booster pump with higher hydraulic efficiency if a more accurate design is to be obtained.

The lifting height of the proposed booster pump is lower than the original requirement at Roskrepp, but the requirement is likely to be too high. By increasing the rotational speed of the RPT and altering its blade design, the head requirement is likely to be decreased. Further investigations into a realistic lifting height for the RPT is needed to conclude if the proposed booster pump delivers enough pressure head.

FURTHER WORK

Before the booster pump design process continues, a final design of an RPT for Roskrepp hydropower plant should be completed. The booster pump requirements depend heavily on the lifting height that the RPT can deliver without cavitating, and therefore, the RPT design should be completed first. If the lifting height needed from the booster pump is too high, an axial booster pump might not be appropriate for the considered application.

For the configuration of a booster pump and an RPT pumping in series at a PHS plant, an investigation should be made into the operation of the RPT in turbine mode. The proposed booster pump design is only designed for one flow direction. What should happen in the other flow direction is not yet determined. If the booster pump can work as a straight-flow turbine, if it should go to runaway speed or if it should be retracted out of the waterway needs to be determined.

BIBLIOGRAPHY

- [1] Hermod (NTNU) Brekke. *Introduction to Hydraulic Machinery*. NTNU, 2000.
- [2] S. Liggett, R. A. Earnshaw, E. Thompson, P. S. Excell, and K. Heald. Collaborative research in art, design and new media-challenges and opportunities. In *2015 Internet Technologies and Applications, ITA 2015 - Proceedings of the 6th International Conference*, pages 503–508. IEEE, 9 2015.
- [3] Dimitrios Laskos. *Design and Cavitation Performance of Contra-Rotating Propellers*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [4] Alexey J. (Ingersoll-Rand Company) Stepanoff. *Centrifugal and Axial Flow Pumps*. John Wiley & Sons, Inc., New York, 2nd editio edition, 1948.
- [5] Mathis Rogner and Nicholas Troja. The world’s water battery: Pumped hydropower storage and the clean energy transition. *IHA Working Paper*, 1(December):1–15, 2018.
- [6] NVE. Vannkraft. <https://www.nve.no/energiforsyning/vannkraft/?ref=mainmenu>, 2019.
- [7] International Renewable Energy Agency. Renewable Energy Technologies: Hydropower. Technical Report 3/5, International Renewable Energy Agency, 2012.
- [8] NVE/Vattenfall. Pumpekraft i Noreg - Kostnader og utsikter til potensial. Technical report, NVE, 2011.
- [9] Sira-Kvina Kraftselskap. Roskrepp kraftverk. <https://www.sirakvina.no/roskrepp-kraftverk/roskrepp-kraftverk-article257-921.html>, 2019.
- [10] Lucio Russo. *The Forgotten Revolution*. Springer-Verlag, 2011.
- [11] John Seddon and Simon Newman. *Basic Helicopter Aerodynamics: Third Edition*. John Wiley & Sons, Inc., 2011.
- [12] John Carlton. *Marine Propellers and Propulsion*. Elsevier Ltd., 2012.
- [13] J.S. Carlton and J.S. Carlton. Propeller Design, 1 1924.

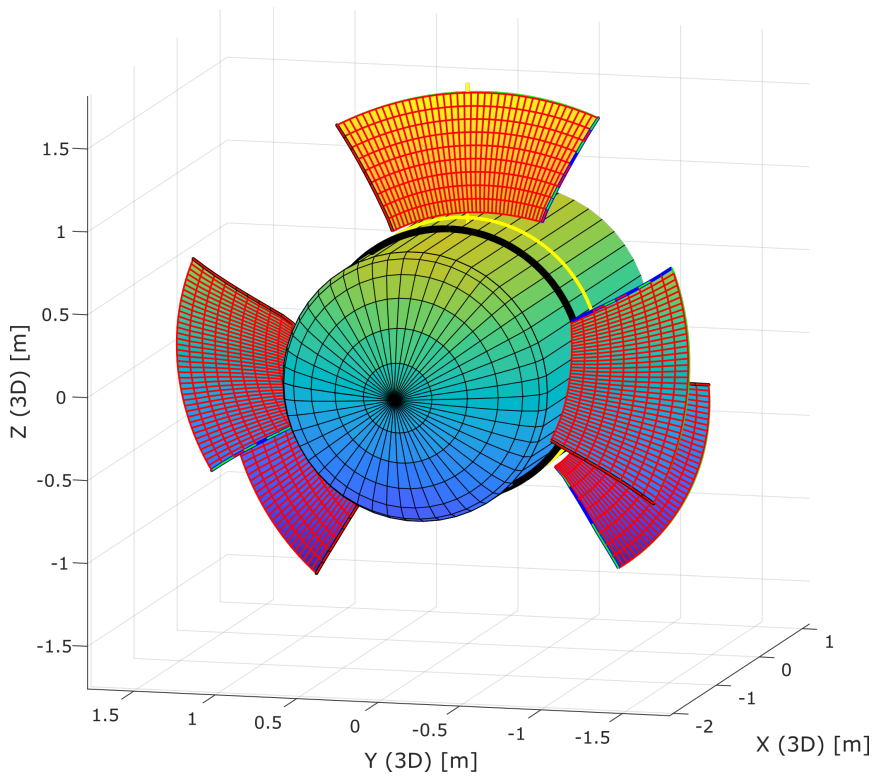
-
- [14] James C Wu. *Springer Tracts in Mechanical Engineering Elements of Vorticity Aerodynamics*. Springer-Verlag, 2004.
- [15] Michael J. Dadswell, Roger A. Rulifson, and Graham R. Daborn. Potential Impact of Large-Scale Tidal Power Developments in the Upper Bay of Fundy on Fisheries Resources of the Northwest Atlantic. *Fisheries*, 11(4):26–35, 7 2004.
- [16] Nowrouz Mohammad Nouri, Saber Mohammadi, and Masoud Zarezadeh. Optimization of a marine contra-rotating propellers set. *Ocean Engineering*, 167:397–404, 11 2018.
- [17] Chi Jen Yang. Pumped Hydroelectric Storage. In *Storing Energy: With Special Reference to Renewable Energy Sources*, pages 25–38. Elsevier, 1 2016.
- [18] A Morabito, J Steimes, and P Hendrick. Pumped hydroelectric energy storage: A comparison of turbomachinery configurations. In *Sustainable Hydraulics in the Era of Global Change*, pages 261–268. CRC Press, 7 2016.
- [19] Øyvind Albert. *Pump as Turbine - Symmetry Prediction Method for Pump as Turbine Characteristics*. PhD thesis, Norwegian University of Science and Technology, 2018.
- [20] Yunus Cengel and John M. Cimbala. *Fluid mechanics Fundamentals and Applications Ed 4*. McGraw-HillHigher Education, 2017.
- [21] Donald R Smith and John E Slater. The Geometry of Marine Propellers. *Defence Research Establishment Atlantic*, 1988.
- [22] Johann Friedrich Gülich. Centrifugal Pumps. In *Hermetic Pumps*, pages 5–96. Springer-Verlag, 2007.
- [23] Steve Wilson. Grundfos White Paper - Specific Speed. Technical report, Grundfos, 2019.
- [24] Roger E.A. Arndt and Leonardo P. Chamorro. Hydraulic turbines. In *Energy Conversion, Second Edition*, pages 257–279. Elsevier Inc., 2017.
- [25] Brenden Epps. OpenProp v2.4 Theory Document. *OpenProp v2.4 Theory document*, 44:32, 2010.
- [26] Eyal Kravitz. *Analysis and Experiments for Contra-Rotating Propeller*. PhD thesis, Massachusetts Institute of Technology, 2011.
- [27] Brenden Epps. On the Rotor Lifting Line Wake Model. *Journal of Ship Production and Design*, 33(1):31–45, 2017.
- [28] G. R. Hough and D. E. Ordway. The generalized actuator disk, 1964.
- [29] Jiwei Tang, Xiaoliang Wang, Dengping Duan, and Weicheng Xie. Optimization and analysis of efficiency for contra-rotating propellers for high-altitude airships. *The Aeronautical Journal*, pages 1–21, 2019.

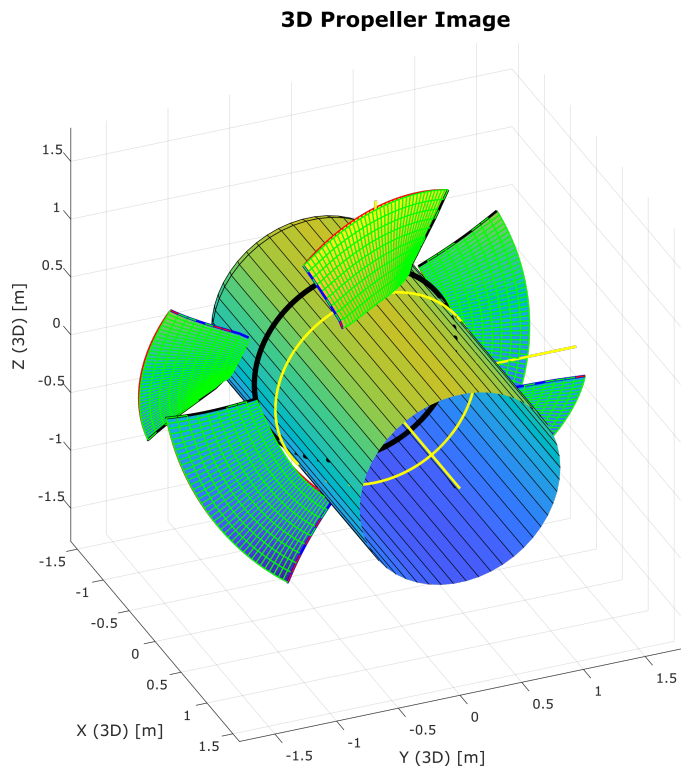
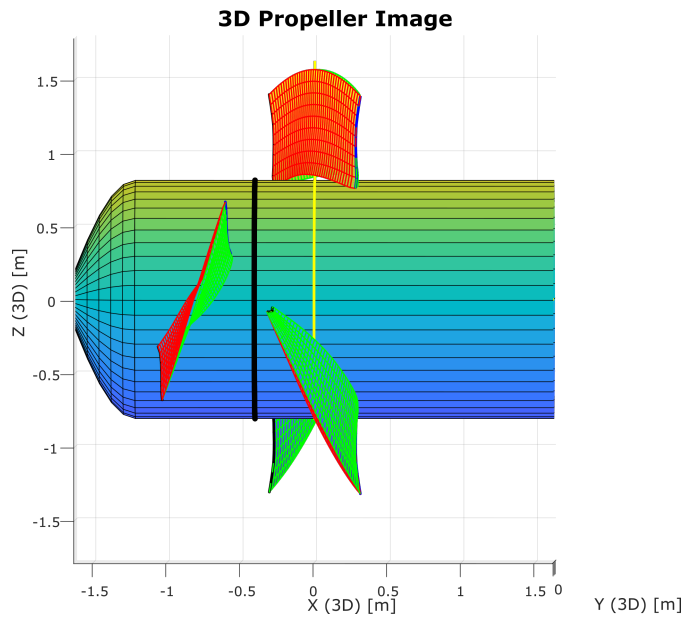
-
- [30] Bao Wei Song, You Jiang Wang, and Wen Long Tian. Open water performance comparison between hub-type and hubless rim driven thrusters based on CFD method. *Ocean Engineering*, 103:55–63, 7 2015.
- [31] Lijian Shi, Fangping Tang, Jiren Zhou, Rongsheng Xie, and Lilong Qi. The influence of hub/tip ratio on hydraulic performance of an axial-flow impeller. *Proceedings of the 3rd International Conference on Industrial Application Engineering 2015*, 2015.
- [32] Takashi Kanemaru, Tomohiro Ryu, Akira Yoshitake, Jun Ando, and Kuniharu Nakatake. The Modeling of Hub Vortex for Numerical Analysis of Marine Propeller Using a Simple Surface Panel Method "SQCM". Technical report, Kyushu University, Japan, 2013.
- [33] Colin P Coleman. A Survey of Theoretical and Experimental Coaxial Rotor Aerodynamic Research. *NASA TP 3675*, 3675(March):1–25, 1997.
- [34] Bhaskar Roy, K Ravibabu, P Srinivasa Rao, S Basu, A Raju, and P N Murthy. Flow Studies in Ducted Twin-Rotor Contra-Rotating Axial Flow Fans. Technical report, Indian Institute of Technology Bombay, 1992.
- [35] MWH Americas Inc. Technical analysis of pumped storage and integration with wind power in the pacific northwest. *Technical report, MWH-HDC-T12*, 2009.

Appendix

PRELIMINARY BOOSTER PUMP DESIGN GEOMETRY

3D Propeller Image



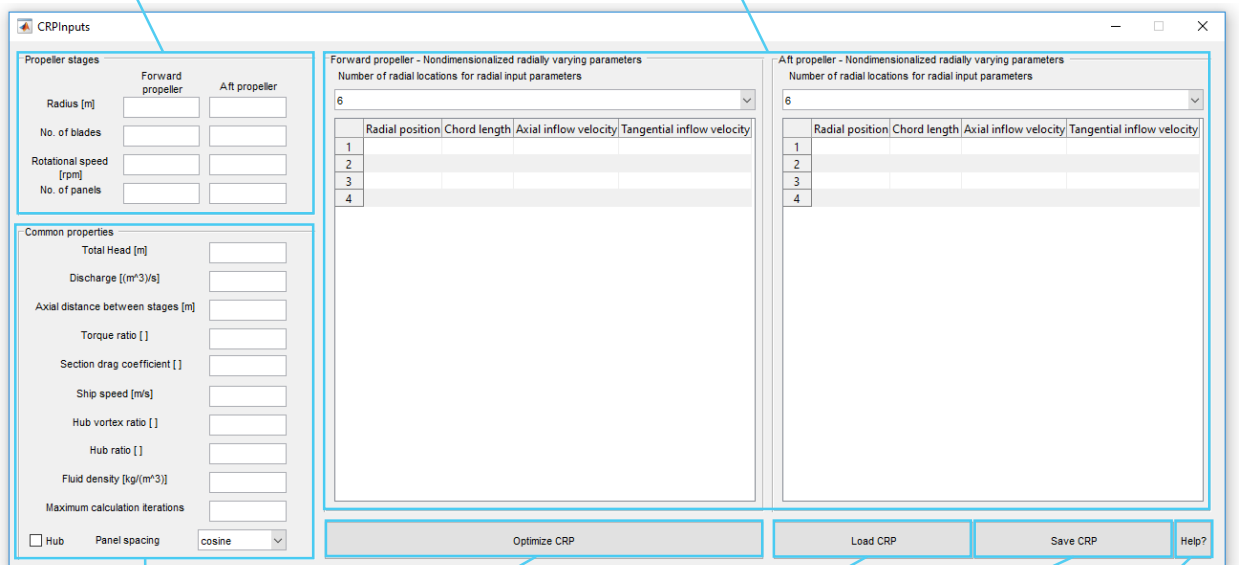


EXPLANATION OF GRAPHICAL USER INTERFACES - COUPLED CRP DESIGN METHOD

B.1 GUI for inputting design parameters

Individual properties for each impeller stage

Radially varying input properties



Properties that aren't restricted to a single impeller

Run optimization

Load previous design

Save current design

Open .txt file with nomenclature, units and explanations

B.2 GUI for viewing optimization results

Results for forward and aft impeller

Efficiency, thrust and torque coefficients

Open 3D geometry plot

View nomenclature, units and explanations of parameters

Open current axes plot in new figure

Select which property to plot

Axes plot showing radial distribution of selected property

The screenshot displays the CRPResults application interface. It features three main data sections: 'Forward propeller', 'Aft propeller', and 'Efficiency and coefficients'. The 'Forward propeller' and 'Aft propeller' sections each contain a table with 10 rows of parameters. The 'Efficiency and coefficients' section provides a summary of key performance indicators. On the right side, there is a plot titled 'Circulation' showing the radial distribution of circulation for two properties, G1 (blue line) and G2 (orange line), across the range of RC1 and RC2. The plot shows G1 peaking around RC1=0.65 and G2 peaking around RC1=0.75. Below the plot is a 'Plot selection' control with radio buttons for G, UA_SELF, UT_SELF, TANBIC, VSTAR, UA_INT, and UT_INT. A 'Geometry' button is also present, which opens a 3D plot of the propeller geometry.

	RC1	G1	UA_SELF1	UT_SELF1	UA_INT1_2	UT_INT1_2	TANBIC1	VSTAR1	CI1
1	0.5244	0.0548	0.1964	-0.1506	0.0541	0	0.6737	2.2418	0.5115
2	0.5732	0.0565	0.2575	-0.1771	0.0551	0	0.6508	2.4099	0.4638
3	0.6220	0.0565	0.2586	-0.1643	0.0556	0	0.5929	2.5804	0.4103
4	0.6707	0.0561	0.2679	-0.1579	0.0555	0	0.5492	2.7529	0.3604
5	0.7195	0.0548	0.2715	-0.1496	0.0548	0	0.5091	2.9268	0.3133
6	0.7683	0.0530	0.2802	-0.1452	0.0534	0	0.4769	3.1016	0.2701
7	0.8171	0.0505	0.2957	-0.1453	0.0514	0	0.4516	3.2767	0.2300
8	0.8659	0.0470	0.3320	-0.1557	0.0488	0	0.4370	3.4519	0.1920
9	0.9146	0.0412	0.3755	-0.1700	0.0459	0	0.4265	3.6272	0.1512
10	0.9634	0.0302	0.4236	-0.1871	0.0426	0	0.4186	3.8025	0.0999

	RC2	G2	UA_SELF2	UT_SELF2	UA_INT2_1	UT_INT2_1	TANBIC2	VSTAR2	CI2
1	0.5244	0.0523	0.1283	-0.0619	0.3915	0.3137	0.7407	2.5570	0.5132
2	0.5732	0.0545	0.1593	-0.0781	0.4007	0.2955	0.6138	2.9864	0.4290
3	0.6220	0.0582	0.2297	-0.1104	0.3979	0.2727	0.5478	3.3925	0.3772
4	0.6707	0.0613	0.2830	-0.1279	0.3914	0.2508	0.4939	3.7863	0.3335
5	0.7195	0.0635	0.3334	-0.1407	0.3797	0.2287	0.4514	4.1690	0.2938
6	0.7683	0.0642	0.3779	-0.1487	0.3640	0.2071	0.4157	4.5438	0.2553
7	0.8171	0.0630	0.4187	-0.1538	0.3433	0.1854	0.3848	4.9117	0.2172
8	0.8659	0.0592	0.4573	-0.1568	0.3157	0.1629	0.3576	5.2728	0.1783
9	0.9146	0.0517	0.4960	-0.1585	0.2709	0.1350	0.3317	5.6214	0.1370
10	0.9634	0.0375	0.5423	-0.1608	0.1894	0.0939	0.3050	5.9460	0.0884

Efficiency and coefficients								
Efficiency	CQ1	CQ2	CT1	CT2	KQ1	KQ2	KT1	KT2
0.60126	0.31289	0.31291	0.75185	0.84589	0.041276	0.049672	0.19837	0.26856

MATLAB CODE FOR DESIGN INPUT GUI - COUPLED CRP DESIGN METHOD

```
function varargout = CRPInputs(varargin)
% CRPInputs MATLAB code for CRPInputs.fig
%     CRPInputs, by itself, creates a new CRPInputs or
%     raises the existing
%     singleton*.
%
%     H = CRPInputs returns the handle to a new CRPInputs
%     or the handle to
%     the existing singleton*.
%
%     CRPInputs('CALLBACK', hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in CRPInputs.M with the
%     given input arguments.
%
%     CRPInputs('Property','Value',...) creates a new
%     CRPInputs or raises the
%     existing singleton*. Starting from the left,
%     property value pairs are
%     applied to the CRPInputs before CRPInputs_OpeningFcn
%     gets called. An
%     unrecognized property name or invalid value makes
%     property application
%     stop. All inputs are passed to CRPInputs_OpeningFcn
%     via varargin.
%
```

```

%      *See CRPInputs Options on GUIDE's Tools menu.
%      Choose "CRPInputs allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
% CRPInputs

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CRPInputs_OpeningFcn, ...
                  'gui_OutputFcn',  @CRPInputs_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
                                         varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before CRPInputs is made visible.
function CRPInputs_OpeningFcn(hObject, eventdata, handles,
                              varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
%            of MATLAB
% handles    structure with handles and user data (see
%            GUIDATA)
% varargin   command line arguments to CRPInputs (see
%            VARARGIN)

% Choose default command line output for CRPInputs
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CRPInputs wait for user response (see
    UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the
    command line.
function varargout = CRPInputs_OutputFcn(hObject, eventdata
    , handles)
% varargout    cell array for returning output args (see
    VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version
    of MATLAB
% handles      structure with handles and user data (see
    GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function R2_Callback(hObject, eventdata, handles)
% hObject      handle to R2 (see GCBO)
% eventdata    reserved - to be defined in a future version
    of MATLAB
% handles      structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of R2 as
    text
%           str2double(get(hObject,'String')) returns contents
    of R2 as a double

% --- Executes during object creation, after setting all
    properties.
function R2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to R2 (see GCBO)
% eventdata    reserved - to be defined in a future version
    of MATLAB
% handles      empty - handles not created until after all
    CreateFcns called

```

```

% Hint: edit controls usually have a white background on
    Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Z2_Callback(hObject, eventdata, handles)
% hObject     handle to Z2 (see GCBO)
% eventdata   reserved - to be defined in a future version
    of MATLAB
% handles     structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of Z2 as
    text
%     str2double(get(hObject,'String')) returns contents
    of Z2 as a double

% --- Executes during object creation, after setting all
    properties.
function Z2_CreateFcn(hObject, eventdata, handles)
% hObject     handle to Z2 (see GCBO)
% eventdata   reserved - to be defined in a future version
    of MATLAB
% handles     empty - handles not created until after all
    CreateFcns called

% Hint: edit controls usually have a white background on
    Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function N2_Callback(hObject, eventdata, handles)
% hObject     handle to N2 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version
% of MATLAB
% handles structure with handles and user data (see
% GUIDATA)

% Hints: get(hObject,'String') returns contents of N2 as
% text
% str2double(get(hObject,'String')) returns contents
% of N2 as a double

% --- Executes during object creation, after setting all
% properties.
function N2_CreateFcn(hObject, eventdata, handles)
% hObject handle to N2 (see GCBO)
% eventdata reserved - to be defined in a future version
% of MATLAB
% handles empty - handles not created until after all
% CreateFcns called

% Hint: edit controls usually have a white background on
% Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function R1_Callback(hObject, eventdata, handles)
% hObject handle to R1 (see GCBO)
% eventdata reserved - to be defined in a future version
% of MATLAB
% handles structure with handles and user data (see
% GUIDATA)

% Hints: get(hObject,'String') returns contents of R1 as
% text
% str2double(get(hObject,'String')) returns contents
% of R1 as a double

% --- Executes during object creation, after setting all
% properties.
function R1_CreateFcn(hObject, eventdata, handles)

```

```

% hObject      handle to R1 (see GCBO)
% eventdata   reserved - to be defined in a future version
               of MATLAB
% handles      empty - handles not created until after all
               CreateFcns called

% Hint: edit controls usually have a white background on
       Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Z1_Callback(hObject, eventdata, handles)
% hObject      handle to Z1 (see GCBO)
% eventdata   reserved - to be defined in a future version
               of MATLAB
% handles      structure with handles and user data (see
               GUIDATA)

% Hints: get(hObject,'String') returns contents of Z1 as
       text
%           str2double(get(hObject,'String')) returns contents
               of Z1 as a double

% --- Executes during object creation, after setting all
       properties.
function Z1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Z1 (see GCBO)
% eventdata   reserved - to be defined in a future version
               of MATLAB
% handles      empty - handles not created until after all
               CreateFcns called

% Hint: edit controls usually have a white background on
       Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function N1_Callback(hObject, eventdata, handles)
% hObject      handle to N1 (see GCBO)
% eventdata    reserved - to be defined in a future version
%               of MATLAB
% handles      structure with handles and user data (see
%               GUIDATA)

% Hints: get(hObject,'String') returns contents of N1 as
%         text
%         str2double(get(hObject,'String')) returns contents
%         of N1 as a double

% --- Executes during object creation, after setting all
%         properties.
function N1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to N1 (see GCBO)
% eventdata    reserved - to be defined in a future version
%               of MATLAB
% handles      empty - handles not created until after all
%               CreateFcns called

% Hint: edit controls usually have a white background on
%         Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0, '
    defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function H_Callback(hObject, eventdata, handles)
% hObject      handle to H (see GCBO)
% eventdata    reserved - to be defined in a future version
%               of MATLAB
% handles      structure with handles and user data (see
%               GUIDATA)

% Hints: get(hObject,'String') returns contents of H as
%         text
%         str2double(get(hObject,'String')) returns contents
%         of H as a double

```

```

% --- Executes during object creation, after setting all
% properties.
function H_CreateFcn(hObject, eventdata, handles)
% hObject    handle to H (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    empty - handles not created until after all
%           CreateFcns called

% Hint: edit controls usually have a white background on
%       Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Q_Callback(hObject, eventdata, handles)
% hObject    handle to Q (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    structure with handles and user data (see
%           GUIDATA)

% Hints: get(hObject,'String') returns contents of Q as
%       text
%       str2double(get(hObject,'String')) returns contents
%       of Q as a double

% --- Executes during object creation, after setting all
% properties.
function Q_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Q (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    empty - handles not created until after all
%           CreateFcns called

% Hint: edit controls usually have a white background on
%       Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Xf_Callback(hObject, eventdata, handles)
% hObject    handle to Xf (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of Xf as
    text
%          str2double(get(hObject,'String')) returns contents
    of Xf as a double

% --- Executes during object creation, after setting all
    properties.
function Xf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Xf (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
    CreateFcns called

% Hint: edit controls usually have a white background on
    Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function qlq2ratio_Callback(hObject, eventdata, handles)
% hObject    handle to qlq2ratio (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of
        qlq2ratio as text
%         str2double(get(hObject,'String')) returns contents
        of qlq2ratio as a double

% --- Executes during object creation, after setting all
        properties.
function qlq2ratio_CreateFcn(hObject, eventdata, handles)
% hObject    handle to qlq2ratio (see GCBO)
% eventdata  reserved - to be defined in a future version
        of MATLAB
% handles    empty - handles not created until after all
        CreateFcns called

% Hint: edit controls usually have a white background on
        Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
        defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function CD_Callback(hObject, eventdata, handles)
% hObject    handle to CD (see GCBO)
% eventdata  reserved - to be defined in a future version
        of MATLAB
% handles    structure with handles and user data (see
        GUIDATA)

% Hints: get(hObject,'String') returns contents of CD as
        text
%         str2double(get(hObject,'String')) returns contents
        of CD as a double

% --- Executes during object creation, after setting all
        properties.
function CD_CreateFcn(hObject, eventdata, handles)
% hObject    handle to CD (see GCBO)
% eventdata  reserved - to be defined in a future version
        of MATLAB

```

```

% handles    empty - handles not created until after all
              CreateFcns called

% Hint: edit controls usually have a white background on
              Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Vs_Callback(hObject, eventdata, handles)
% hObject    handle to Vs (see GCBO)
% eventdata  reserved - to be defined in a future version
              of MATLAB
% handles    structure with handles and user data (see
              GUIDATA)

% Hints: get(hObject,'String') returns contents of Vs as
              text
%           str2double(get(hObject,'String')) returns contents
              of Vs as a double

% --- Executes during object creation, after setting all
              properties.
function Vs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Vs (see GCBO)
% eventdata  reserved - to be defined in a future version
              of MATLAB
% handles    empty - handles not created until after all
              CreateFcns called

% Hint: edit controls usually have a white background on
              Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Rhv_Callback(hObject, eventdata, handles)
% hObject      handle to Rhv (see GCBO)
% eventdata    reserved - to be defined in a future version
%               of MATLAB
% handles      structure with handles and user data (see
%               GUIDATA)

% Hints: get(hObject,'String') returns contents of Rhv as
%         text
%         str2double(get(hObject,'String')) returns contents
%         of Rhv as a double

% --- Executes during object creation, after setting all
%         properties.
function Rhv_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Rhv (see GCBO)
% eventdata    reserved - to be defined in a future version
%               of MATLAB
% handles      empty - handles not created until after all
%               CreateFcns called

% Hint: edit controls usually have a white background on
%         Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in spacing.
function spacing_Callback(hObject, eventdata, handles)
% hObject      handle to spacing (see GCBO)
% eventdata    reserved - to be defined in a future version
%               of MATLAB
% handles      structure with handles and user data (see
%               GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
%         spacing contents as cell array
%         contents{get(hObject,'Value')} returns selected
%         item from spacing

```

```

% --- Executes during object creation, after setting all
% properties.
function spacing_CreateFcn(hObject, eventdata, handles)
% hObject    handle to spacing (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    empty - handles not created until after all
%           CreateFcns called

% Hint: popupmenu controls usually have a white background
%       on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0, '
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function M2_Callback(hObject, eventdata, handles)
% hObject    handle to M2 (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    structure with handles and user data (see
%           GUIDATA)

% Hints: get(hObject,'String') returns contents of M2 as
%       text
%       str2double(get(hObject,'String')) returns contents
%       of M2 as a double

% --- Executes during object creation, after setting all
% properties.
function M2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to M2 (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    empty - handles not created until after all
%           CreateFcns called

% Hint: edit controls usually have a white background on
%       Windows.
%       See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function M1_Callback(hObject, eventdata, handles)
% hObject    handle to M1 (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of M1 as
    text
%          str2double(get(hObject,'String')) returns contents
    of M1 as a double

% --- Executes during object creation, after setting all
    properties.
function M1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to M1 (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
    CreateFcns called

% Hint: edit controls usually have a white background on
    Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in HubFlag.
function HubFlag_Callback(hObject, eventdata, handles)
% hObject    handle to HubFlag (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of
      HubFlag

function Rhub_Callback(hObject, eventdata, handles)
% hObject      handle to Rhub (see GCBO)
% eventdata    reserved - to be defined in a future version
      of MATLAB
% handles      structure with handles and user data (see
      GUIDATA)

% Hints: get(hObject,'String') returns contents of Rhub as
      text
%      str2double(get(hObject,'String')) returns contents
      of Rhub as a double

% --- Executes during object creation, after setting all
      properties.
function Rhub_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Rhub (see GCBO)
% eventdata    reserved - to be defined in a future version
      of MATLAB
% handles      empty - handles not created until after all
      CreateFcns called

% Hint: edit controls usually have a white background on
      Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0, '
      defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function rho_Callback(hObject, eventdata, handles)
% hObject      handle to rho (see GCBO)
% eventdata    reserved - to be defined in a future version
      of MATLAB
% handles      structure with handles and user data (see
      GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of rho as
text
%         str2double(get(hObject,'String')) returns contents
of rho as a double

% --- Executes during object creation, after setting all
properties.
function rho_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rho (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in OptimizeButton.
function OptimizeButton_Callback(hObject, eventdata,
handles)
% hObject    handle to OptimizeButton (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get design input variables
[R1,R2,Z1,Z2,N1,N2,M1,M2,H,Q,Xf,q,CD,...
    Vs,Rhv,Rhub,rho,ITER,spacing, Hub_Flag, XR1,XR2,XCoD1,
    ...
    XCoD2,XVA1,XVA2,XVT1,XVT2] = getGUIInputs(handles);

% Required thrust computation
% Calculation only valid for high-efficiency pumps
pumpPower = rho*9.81*Q*H;
Tr = pumpPower/Vs;

% Run Coupled CRP design method developed by Laskos

```

```

[EFFY, CT1, CT2, CQ1, CQ2, KT1, KT2, KQ1, KQ2, RC1, RC2, G1, G2,
  UA_SELF1, UT_SELF1, ...
  UA_INT1_2, UT_INT1_2, UA_SELF2, UT_SELF2, UA_INT2_1,
  UT_INT2_1, TANBIC1, ...
  TANBIC2, VSTAR1, VSTAR2, C11, C12]=CoupledCRP (Rhub, R1, R2, M1
  , M2, Z1, Z2, ...
  Tr, q, N1, N2, XR1, XR2, XCoD1, XCoD2, CD, XVA1, XVA2, XVT1, XVT2,
  Vs, Xf, ITER, ...
  spacing, Hub_Flag, Rhv);

% Save design results to .mat file for opening in results
  GUI
save("OptimizerResults.mat", 'EFFY', 'CT1', 'CT2', 'CQ1', 'CQ2',
  'KT1', ...
  'KT2', 'KQ1', 'KQ2', 'RC1', 'RC2', 'G1', 'G2', 'UA_SELF1', '
  UT_SELF1', ...
  'UA_INT1_2', 'UT_INT1_2', 'UA_SELF2', 'UT_SELF2', '
  UA_INT2_1', ...
  'UT_INT2_1', 'TANBIC1', 'TANBIC2', 'VSTAR1', 'VSTAR2', 'C11'
  , 'C12', ...
  'Rhub', 'R1', 'R2', 'M1', 'M2', 'Z1', 'Z2', 'Tr', 'q', 'N1', 'N2'
  , 'XR1', ...
  'XR2', 'XCoD1', 'XCoD2', 'CD', 'XVA1', 'XVA2', 'XVT1', 'XVT2',
  'Vs', ...
  'Xf', 'ITER', 'spacing', 'Hub_Flag', 'Rhv');

% Start results GUI to display optimization results
run('CRPResults.m');
% Close self
close(handles.figure1);

% --- Executes on selection change in NoRadPos1.
function NoRadPos1_Callback(hObject, eventdata, handles)
% hObject    handle to NoRadPos1 (see GCBO)
% eventdata  reserved - to be defined in a future version
  of MATLAB
% handles    structure with handles and user data (see
  GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
  NoRadPos1 contents as cell array
%         contents{get(hObject,'Value')} returns selected
  item from NoRadPos1

popup3val = get(handles.NoRadPos1, 'Value');

```

```

set(handles.RadForward, 'Data', cell(popup3val, 4));

% --- Executes during object creation, after setting all
% properties.
function NoRadPos1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NoRadPos1 (see GCBO)
% eventdata  reserved - to be defined in a future version
% of MATLAB
% handles    empty - handles not created until after all
% CreateFcns called

% Hint: popupmenu controls usually have a white background
% on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in NoRadPos2.
function NoRadPos2_Callback(hObject, eventdata, handles)
% hObject    handle to NoRadPos2 (see GCBO)
% eventdata  reserved - to be defined in a future version
% of MATLAB
% handles    structure with handles and user data (see
% GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
% NoRadPos2 contents as cell array
%         contents{get(hObject,'Value')} returns selected
%         item from NoRadPos2

popup4val = get(handles.NoRadPos2, 'Value');
set(handles.RadAft, 'Data', cell(popup4val, 4));

% --- Executes during object creation, after setting all
% properties.
function NoRadPos2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NoRadPos2 (see GCBO)
% eventdata  reserved - to be defined in a future version
% of MATLAB
% handles    empty - handles not created until after all
% CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background
    on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, '
    defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in LoadButton.
function LoadButton_Callback(hObject, eventdata, handles)
% hObject     handle to LoadButton (see GCBO)
% eventdata   reserved - to be defined in a future version
    of MATLAB
% handles     structure with handles and user data (see
    GUIDATA)

% Choose file to load
CRPLoad = uiimport;

% Set methods
set(handles.R1, 'String', num2str(CRPLoad.R1));
set(handles.R2, 'String', num2str(CRPLoad.R2));
set(handles.Z1, 'String', num2str(CRPLoad.Z1));
set(handles.Z2, 'String', num2str(CRPLoad.Z2));
set(handles.N1, 'String', num2str(CRPLoad.N1));
set(handles.N2, 'String', num2str(CRPLoad.N2));
set(handles.M1, 'String', num2str(CRPLoad.M1));
set(handles.M2, 'String', num2str(CRPLoad.M2));
set(handles.H, 'String', num2str(CRPLoad.H));
set(handles.Q, 'String', num2str(CRPLoad.Q));
set(handles.Xf, 'String', num2str(CRPLoad.Xf));
set(handles.q1q2ratio, 'String', num2str(CRPLoad.q));
set(handles.CD, 'String', num2str(CRPLoad.CD));
set(handles.Vs, 'String', num2str(CRPLoad.Vs));
set(handles.Rhv, 'String', num2str(CRPLoad.Rhv));
set(handles.Rhub, 'String', num2str(CRPLoad.Rhub/CRPLoad.R1
));
set(handles.rho, 'String', num2str(CRPLoad.rho));
set(handles.ITER, 'String', num2str(CRPLoad.ITER));
if (CRPLoad.spacing == "constant")
    set(handles.spacing, 'Value', 1);
elseif (CRPLoad.spacing == "cosine")
    set(handles.spacing, 'Value', 2);

```

```

end
set(handles.HubFlag, 'Value', CRPLoad.Hub_Flag);

% Radial properties of forward impeller
forwardCells = cell(length(CRPLoad.XR1),4);
forwardCells(:,1) = transpose(num2cell(CRPLoad.XR1));
forwardCells(:,2) = transpose(num2cell(CRPLoad.XCoD1));
forwardCells(:,3) = transpose(num2cell(CRPLoad.XVA1));
forwardCells(:,4) = transpose(num2cell(CRPLoad.XVT1));
set(handles.RadForward, 'Data', forwardCells);
set(handles.NoRadPos1, 'Value', length(CRPLoad.XR1));

% Radial properties of aft impeller
aftCells = cell(length(CRPLoad.XR2),4);
aftCells(:,1) = transpose(num2cell(CRPLoad.XR2));
aftCells(:,2) = transpose(num2cell(CRPLoad.XCoD2));
aftCells(:,3) = transpose(num2cell(CRPLoad.XVA2));
aftCells(:,4) = transpose(num2cell(CRPLoad.XVT2));
set(handles.RadAft, 'Data', aftCells);
set(handles.NoRadPos2, 'Value', length(CRPLoad.XR2));

% --- Executes on button press in SaveButton.
function SaveButton_Callback(hObject, eventdata, handles)
% hObject    handle to SaveButton (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    structure with handles and user data (see
%           GUIDATA)

[R1,R2,Z1,Z2,N1,N2,M1,M2,H,Q,Xf,q,CD,...
 Vs,Rhv,Rhub,rho,ITER,spacing, Hub_Flag, XR1,XR2,XCoD1,
 ...
 XCoD2,XVA1,XVA2,XVT1,XVT2] = getGUIInputs(handles);

% CRPSave = struct('R1', R1, 'R2', R2, 'Z1', Z1, 'Z2', Z2,
%   'N1', N1,...
%   'N2', N2, 'M1', M1, 'M2', M2, 'H', H, 'Q', Q, 'Xf',
%   Xf, 'q', q,...
%   'CD', CD, 'Vs', Vs, 'Rhv', Rhv, 'Rhub', Rhub, 'rho',
%   rho,...
%   'spacing', spacing, 'XR1', XR1, 'XR2', XR2, 'XCoD1',
%   XCoD1,...
%   'XCoD2', XCoD2, 'XVA1', XVA1, 'XVA2', XVA2, 'XVT1',
%   XVT1, 'XVT2', XVT2);

```

```

% Save all design parameters in .mat file. Filename is
  prompted from the
% user
uisave({'R1', 'R2', 'Z1', 'Z2', 'N1', 'N2', 'M1', 'M2', 'H', 'Q', 'Xf
      ', 'q', 'CD', ...
      'Vs', 'Rhv', 'Rhub', 'rho', 'ITER', 'spacing', 'Hub_Flag', '
      XR1', 'XR2', 'XCoD1', ...
      'XCoD2', 'XVA1', 'XVA2', 'XVT1', 'XVT2'});

function [R1,R2,Z1,Z2,N1,N2,M1,M2,H,Q,Xf,q,CD,...
        Vs,Rhv,Rhub,rho,ITER,spacing, Hub_Flag, XR1,XR2,XCoD1,
        ...
        XCoD2,XVA1,XVA2,XVT1,XVT2] = getGUIInputs(handles)

% Get methods for saving CRPInputs input values
% Values converted to correct type from input GUI
R1 = str2num(get(handles.R1, 'String'));
R2 = str2num(get(handles.R2, 'String'));
Z1 = str2num(get(handles.Z1, 'String'));
Z2 = str2num(get(handles.Z2, 'String'));
N1 = str2num(get(handles.N1, 'String'));
N2 = str2num(get(handles.N2, 'String'));
M1 = str2num(get(handles.M1, 'String'));
M2 = str2num(get(handles.M2, 'String'));
H = str2num(get(handles.H, 'String'));
Q = str2num(get(handles.Q, 'String'));
Xf = str2num(get(handles.Xf, 'String'));
q = str2num(get(handles.q1q2ratio, 'String'));
CD = str2num(get(handles.CD, 'String'));
Vs = str2num(get(handles.Vs, 'String'));
Rhv = str2num(get(handles.Rhv, 'String'));
Rhub = str2num(get(handles.Rhub, 'String'))*R1;
rho = str2num(get(handles.rho, 'String'));
ITER = str2num(get(handles.ITER, 'String'));
spacing = "";
if (get(handles.spacing, 'Value') == 1)
    spacing = "constant";
elseif (get(handles.spacing, 'Value') == 2)
    spacing = "cosine";
end
Hub_Flag = get(handles.HubFlag, 'Value');

% Radial properties of forward impeller
forwardCells = cell2mat(get(handles.RadForward, 'Data'));

```

```

XR1 = transpose(forwardCells(:,1));
XCoD1 = transpose(forwardCells(:,2));
XVA1 = transpose(forwardCells(:,3));
XVT1 = transpose(forwardCells(:,4));

% Radial properties of aft impeller
aftCells = cell2mat(get(handles.RadAft, 'Data'));
XR2 = transpose(aftCells(:,1));
XCoD2 = transpose(aftCells(:,2));
XVA2 = transpose(aftCells(:,3));
XVT2 = transpose(aftCells(:,4));

function ITER_Callback(hObject, eventdata, handles)
% hObject      handle to ITER (see GCBO)
% eventdata    reserved - to be defined in a future version
%              of MATLAB
% handles      structure with handles and user data (see
%              GUIDATA)

% Hints: get(hObject,'String') returns contents of ITER as
%         text
%         str2double(get(hObject,'String')) returns contents
%         of ITER as a double

% --- Executes during object creation, after setting all
%         properties.
function ITER_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ITER (see GCBO)
% eventdata    reserved - to be defined in a future version
%              of MATLAB
% handles      empty - handles not created until after all
%              CreateFcns called

% Hint: edit controls usually have a white background on
%         Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0, '
defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version
%              of MATLAB
% handles      structure with handles and user data (see
%              GUIDATA)

% Open .txt file with nomenclature and units for all
%              parameters
winopen('DesignInputs.txt');
```

MATLAB CODE FOR RESULTS GUI - COUPLED CRP DESIGN METHOD

```
function varargout = CRPResults(varargin)
% CRPRESULTS MATLAB code for CRPResults.fig
%     CRPRESULTS, by itself, creates a new CRPRESULTS or
%     raises the existing
%     singleton*.
%
%     H = CRPRESULTS returns the handle to a new
%     CRPRESULTS or the handle to
%     the existing singleton*.
%
%     CRPRESULTS('CALLBACK', hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in CRPRESULTS.M with the
%     given input arguments.
%
%     CRPRESULTS('Property','Value',...) creates a new
%     CRPRESULTS or raises the
%     existing singleton*. Starting from the left,
%     property value pairs are
%     applied to the GUI before CRPResults_OpeningFcn gets
%     called. An
%     unrecognized property name or invalid value makes
%     property application
%     stop. All inputs are passed to
%     CRPResults_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI
```

```

        allows only one
%       instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
    CRPResults

% Last Modified by GUIDE v2.5 24-May-2019 13:53:51

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',     gui_Singleton, ...
                  'gui_OpeningFcn',    @CRPResults_OpeningFcn
                  , ...
                  'gui_OutputFcn',     @CRPResults_OutputFcn,
                  ...
                  'gui_LayoutFcn',     [] , ...
                  'gui_Callback',      []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
                                         varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before CRPResults is made visible.
function CRPResults_OpeningFcn(hObject, eventdata, handles,
                               varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
%             of MATLAB
% handles    structure with handles and user data (see
%            GUIDATA)
% varargin   command line arguments to CRPResults (see
%            VARARGIN)

```

```

% Choose default command line output for CRPResults
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CRPResults wait for user response (see
    UIRESUME)
% uiwait(handles.figure1);

% Run initialization of GUI values from design input GUI
initializeResults(handles);

set(handles.figure1, 'CloseRequestFcn', @closeGUI);

% --- Outputs from this function are returned to the
    command line.
function varargout = CRPResults_OutputFcn(hObject,
    eventdata, handles)
% varargout cell array for returning output args (see
    VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version
    of MATLAB
% handles structure with handles and user data (see
    GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function initializeResults(handles)

% Load default optimizer results file into the workspace
load("OptimizerResults.mat");

% Update table values for forward propeller
forwardCells = cell(length(RC1), 9);
forwardCells(:, 1) = num2cell(RC1);
forwardCells(:, 2) = num2cell(G1);
forwardCells(:, 3) = num2cell(UA_SELF1);
forwardCells(:, 4) = num2cell(UT_SELF1);
forwardCells(:, 5) = num2cell(UA_INT1_2);
forwardCells(:, 6) = num2cell(UT_INT1_2);
forwardCells(:, 7) = num2cell(TANBIC1);

```

```

forwardCells(:,8) = num2cell(VSTAR1);
forwardCells(:,9) = num2cell(C11);
set(handles.forwardTable, 'Data', forwardCells);

% Update table values for aft propeller
aftCells = cell(length(RC2),9);
aftCells(:,1) = num2cell(RC2);
aftCells(:,2) = num2cell(G2);
aftCells(:,3) = num2cell(UA_SELF2);
aftCells(:,4) = num2cell(UT_SELF2);
aftCells(:,5) = num2cell(UA_INT2_1);
aftCells(:,6) = num2cell(UT_INT2_1);
aftCells(:,7) = num2cell(TANBIC2);
aftCells(:,8) = num2cell(VSTAR2);
aftCells(:,9) = num2cell(C12);
set(handles.aftTable, 'Data', aftCells);

% Update coefficient values
set(handles.EFFY, 'String', num2str(EFFY));
set(handles.CQ1, 'String', num2str(CQ1));
set(handles.CQ2, 'String', num2str(CQ2));
set(handles.CT1, 'String', num2str(CT1));
set(handles.CT2, 'String', num2str(CT2));
set(handles.KQ1, 'String', num2str(KQ1));
set(handles.KQ2, 'String', num2str(KQ2));
set(handles.KT1, 'String', num2str(KT1));
set(handles.KT2, 'String', num2str(KT2));

% Initialize plot
axes(handles.resultPlot);
plot(RC1, G1, RC2, G2);
leg = legend('G1', 'G2');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Circulation");
grid on;

% --- Executes on button press in radio_G.
function radio_G_Callback(hObject, eventdata, handles)
% hObject    handle to radio_G (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    structure with handles and user data (see
%           GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of
    radio_G

load("OptimizerResults.mat");

% Update GUI axes with nondimensional circulation values
axes(handles.resultPlot);
plot(RC1, G1, RC2, G2);
leg = legend('G1', 'G2');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Circulation");
grid on;

% --- Executes on button press in radio_UA_SELF.
function radio_UA_SELF_Callback(hObject, eventdata, handles
    )
% hObject    handle to radio_UA_SELF (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
    radio_UA_SELF

load("OptimizerResults.mat");

% Update GUI axes with self-induced axial velocity values
axes(handles.resultPlot);
plot(RC1, UA_SELF1, RC2, UA_SELF2);
leg = legend('UA_SELF1', 'UA_SELF2');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Axial self induced velocities");
grid on;

% --- Executes on button press in radio_UT_SELF.
function radio_UT_SELF_Callback(hObject, eventdata, handles
    )
% hObject    handle to radio_UT_SELF (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB

```

```

% handles      structure with handles and user data (see
    GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
    radio_UT_SELF

load("OptimizerResults.mat");

% Update GUI tangential with self-induced axial velocity
    values
axes(handles.resultPlot);
plot(RC1, UT_SELF1, RC2, UT_SELF2);
leg = legend('UT_SELF1', 'UT_SELF2');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Tangential self induced velocities");
grid on;

% --- Executes on button press in radio_TANBIC.
function radio_TANBIC_Callback(hObject, eventdata, handles)
% hObject      handle to radio_TANBIC (see GCBO)
% eventdata    reserved - to be defined in a future version
    of MATLAB
% handles      structure with handles and user data (see
    GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
    radio_TANBIC

load("OptimizerResults.mat");

% Update GUI axes with tangent of pitch angles
axes(handles.resultPlot);
plot(RC1, TANBIC1, RC2, TANBIC2);
leg = legend('TANBIC1', 'TANBIC2');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Tangent of hydrodynamic pitch angle");
grid on;

% --- Executes on button press in radio_VSTAR.
function radio_VSTAR_Callback(hObject, eventdata, handles)
% hObject      handle to radio_VSTAR (see GCBO)

```

```

% eventdata reserved - to be defined in a future version
% of MATLAB
% handles structure with handles and user data (see
% GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
% radio_VSTAR

load("OptimizerResults.mat");

% Update GUI axes with total resultant inflow velocity
% values
axes(handles.resultPlot);
plot(RC1, VSTAR1, RC2, VSTAR2);
leg = legend('VSTAR1', 'VSTAR2');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Total inflow velocity");
grid on;

% --- Executes on button press in radio_UA_INT.
function radio_UA_INT_Callback(hObject, eventdata, handles)
% hObject handle to radio_UA_INT (see GCBO)
% eventdata reserved - to be defined in a future version
% of MATLAB
% handles structure with handles and user data (see
% GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
% radio_UA_INT

load("OptimizerResults.mat");

% Update GUI axes with axial interaction velocity values
axes(handles.resultPlot);
plot(RC1, UA_INT1_2, RC2, UA_INT2_1);
leg = legend('UA_INT1_2', 'UA_INT2_1');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Axial interaction velocities");
grid on;

% --- Executes on button press in radio_UT_INT.

```

```

function radio_UT_INT_Callback(hObject, eventdata, handles)
% hObject      handle to radio_UT_INT (see GCBO)
% eventdata    reserved - to be defined in a future version
%             of MATLAB
% handles      structure with handles and user data (see
%             GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
%       radio_UT_INT

load("OptimizerResults.mat");

% Update GUI axes with tangential interaction velocity
%       values
axes(handles.resultPlot);
plot(RC1, UT_INT1_2, RC2, UT_INT2_1);
leg = legend('UT_INT1_2', 'UT_INT2_1');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Tangential interaction velocities");
grid on;

% --- Executes on button press in radio_C1.
function radio_C1_Callback(hObject, eventdata, handles)
% hObject      handle to radio_C1 (see GCBO)
% eventdata    reserved - to be defined in a future version
%             of MATLAB
% handles      structure with handles and user data (see
%             GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
%       radio_C1

load("OptimizerResults.mat");

% Update GUI axes with required lift coefficient
%       distribution
axes(handles.resultPlot);
plot(RC1, C11, RC2, C12);
leg = legend('C11', 'C12');
set(leg, 'Interpreter', 'none');
xlabel("RC1 | RC2");
ylabel("Required lift coefficient");
grid on;

```

```

function closeGUI(hObject, eventdata, handles)

% Prompt user to save results
answer = questdlg('Save results to another file before
    quitting?', 'Yes', 'No');

switch answer
    case 'Yes'
        load("OptimizerResults.mat"); % Load variables into
            workspace
        uisave; % Save values to ,mat file
        delete(hObject); % Close GUI
    case 'No'
        delete(hObject); % Close GUI
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

load("OptimizerResults.mat");

% Geometry module of Laskos
CoD1 = pchip(XR1,XCoD1,RC1);
CoD2 = pchip(XR2,XCoD2,RC2);

Gamma1=G1*2*pi*R1*Vs;
Gamma2=G2*2*pi*R2*Vs;
C11= 2*Gamma1'./(VSTAR1*Vs.*CoD1.*2*R1);
C12= 2*Gamma2'./(VSTAR2*Vs.*CoD2.*2*R2);
% ===== Inputs necessary for geometry generation
    =====
skew01 = zeros(1,11); % Skew [deg]
skew02 = zeros(1,11);
rake01 = zeros(1,11); % Xs/D, Rake
rake02 = zeros(1,11);
% ----- t0/c, thickness / chord
    -----
t0oc01 = [.2056 .1551 .1181 .0902 .0694 .0541 .0419 .0332

```

```

        .0324...
        .0204 .005];
        %[0.0815 0.0771 0.0731 0.0664 0.0608 0.0561 0.0522...
        % 0.0489 0.0457 0.0457 0.005];
t0oc02 = t0oc01;
%
-----

BetaI_c1=atand(TANBIC1);
BetaI_c2=atand(TANBIC2);
Np=40; % Number of points over the chord
% ===== Generate Propeller Geometry
% =====
Geometry(XR1,XR2,t0oc01,t0oc02,skew01,skew02,rake01,rake02,
        RC1,RC2,C11,...
        C12,BetaI_c1,BetaI_c2,Xf,Z1,Z2,Rhub,CoD1,CoD2,R1,R2,M1,M2,
        Np)

% --- Executes on button press in openPlotButton.
function openPlotButton_Callback(hObject, eventdata, handles)
% hObject    handle to openPlotButton (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    structure with handles and user data (see
%           GUIDATA)

% Open plot in external figure for editing
f1 = figure; % Open a new figure with handle f1
copyobj(handles.resultPlot,f1); % Copy axes object h into
figure f1
set(gca,'ActivePositionProperty','outerposition')
set(gca,'Units','normalized')
set(gca,'OuterPosition',[0 0 1 1])
set(gca,'position',[0.1300 0.1100 0.7750 0.8150])

% --- Executes on button press in helpButton.
function helpButton_Callback(hObject, eventdata, handles)
% hObject    handle to helpButton (see GCBO)
% eventdata  reserved - to be defined in a future version
%           of MATLAB
% handles    structure with handles and user data (see
%           GUIDATA)

```

```
% Open .txt file with nomenclature and unit explanations  
winopen('DesignOutputs.txt');
```

DERIVATIVES OF AUXILIARY FUNCTION FOR OPTIMUM CIRCULATION CALCULATIONS

The derivatives of the auxiliary function B with respect to the unknown circulations and the Lagrange multipliers are set equal to zero by the following equations

$$\begin{aligned}
 \frac{\partial B}{\partial \Gamma_j(i)} &= 0 \\
 &= \omega_1 \frac{\partial \tau_1}{\partial \Gamma_j(i)} + \omega_2 \frac{\partial \tau_2}{\partial \Gamma_j(i)} + \lambda_T \left[\frac{\partial T_1}{\partial \Gamma_j(i)} + \frac{\partial T_2}{\partial \Gamma_j(i)} \right] + \lambda_Q \left[q \frac{\partial \tau_1}{\partial \Gamma_j(i)} + \frac{\partial \tau_2}{\partial \Gamma_j(i)} \right] \\
 &= (\omega_1 + q\lambda_Q) \frac{\partial \tau_1}{\partial \Gamma_j(i)} + (\omega_2 - \lambda_Q) \frac{\partial \tau_2}{\partial \Gamma_j(i)} + \lambda_T \left[\frac{\partial T_1}{\partial \Gamma_j(i)} + \frac{\partial T_2}{\partial \Gamma_j(i)} \right] \\
 &\text{for } j = 1, 2 \text{ and } i = 1, \dots, M_j
 \end{aligned} \tag{E.1}$$

and,

$$\frac{\partial B}{\partial \lambda_T} = 0 = (T_{I_1} + T_{V_1}) + (T_{I_2} + T_{V_2}) - T_r \tag{E.2}$$

$$\frac{\partial B}{\partial \lambda_Q} = 0 = q(\tau_{I_1} + \tau_{V_1}) - (\tau_{I_2} + \tau_{V_2}) \tag{E.3}$$

Here, the subscripts I and V are used to indicate the inviscid and viscous thrust and torque of the forward and aft impellers (subscripts 1 and 2). The inviscid and viscous thrust forces are given by the following equations

$$T_{I_j} = \rho Z_j \sum_{n=1}^{M_j} \left[V_{t_j}(n) + \omega_j r_j(n) + u_{t_j}^*(n) \right] \Gamma_j(n) \Delta r_j \tag{E.4}$$

$$T_{V_j} = -\frac{1}{2} \rho Z_j \sum_{n=1}^{M_j} V_j^*(n) \left[V_{a_j}(n) + u_{a_j}^*(n) \right] c_j(n) C_{D_{V_j}}(n) \Delta r_j \tag{E.5}$$

The inviscid and viscid torques are given by the following equations

$$\tau_{I_j} = \rho Z_j + \sum_{n=1}^{M_j} \left[V_{a_j}(n) + u_{a_j}^*(n) \right] r_j(n) \Gamma_j(n) \Delta r_j \quad (\text{E.6})$$

$$\tau_{V_j} = \frac{1}{2} \rho Z_j \sum_{n=1}^{M_j} V_j^*(n) \left[V_{t_j}(n) + \omega_j r_j(n) + u_{t_j}^*(n) \right] r_j(n) c_j(n) C_{D_{V_j}}(n) \Delta r_j \quad (\text{E.7})$$

for j = 1,2

RESEARCH PAPER FOR CRHT-IX CONFERENCE AT
KATHMANDU UNIVERSITY IN NEPAL

Pump turbines in existing powerplants

Pål Dahle¹ and Pål-Tore Selbo Storli¹

¹Department of energy and process engineering, Norwegian University of Science and Technology, Trondheim, Norway

E-mail: paal.dahle@gmail.com

March 2019

Abstract.

Norway has 50% of the European hydro reservoir energy storage [1], and many of these sites are highly suitable for retrofitting of pump-storage capabilities. To be able to reuse existing hydropower plants with Francis turbines by retrofitting with Reversible Pump Turbines (RPTs), the problem of cavitation in pumping mode must be solved. This has been proposed solved by placing an axial booster pump in front of the RPT. As a case, such a booster pump has been designed for the use at Roskrepp hydropower plant in Norway. Preliminary findings suggest that for the needed performance of a booster pump at Roskrepp, conventional single stage axial machines will have very low efficiency. Other design solutions are available and must be investigated to determine if they are applicable at Roskrepp. A counter-rotating two-stage pump design will be investigated.

Keywords: Booster pump, cavitation, pumped storage hydropower, counter-rotating pump, axial pump

1. Introduction

1.1. Background

Dammed hydroelectricity is the largest form of energy storage in the world [2]. This form of storage consists of both conventional hydroelectric generation as well as pumped storage. Where conditions are favourable, pumped storage hydropower (PSH) has a significant economical advantage over conventional hydroelectric generation when electricity prices vary periodically. Pumping water to the upper reservoir when prices are low, and producing energy when prices are high allows a PSH plant to profit from the price variations in the electricity market. The causes for price variations is a shift in the balance between the energy consumption and energy production within the day, days, weeks and seasons. Historically, PHS plants operating on the shorter time-scales have been economically viable because they have balanced base-load power generation (coal-fired or nuclear power plants) with fluctuating power consumption. The green transition of the energy system will require fossil fuel base-load generation to be substituted with renewable intermittent sources such as solar photo-voltaic, wind and wave

energy. This will increase fluctuations in the generation side as well, making the balance between consumption and generations shift more rapidly. This implies more fluctuating energy prices, which subsequently would favour more PHS capacity and investments. Being able to balance the power production with the consumption will be increasingly important as the world strives to utilize greener energy sources such as wind power, solar power and wave energy.

In Norway, conventional hydroelectric generation is by far the most common form of hydropower. With a total of 1676 hydropower plants, only nine have pumping capacity [3]. Almost 10 GW of Norway's hydropower is installed in hydropower plants with a reservoir and a discharge tunnel leading to another reservoir or lake. These power plants have an infrastructure consisting of waterways, access tunnels and generator halls that can be reused for installing pumped hydropower capacity. Reusing as much as possible of the existing infrastructure reduces the cost of retrofitting with pump capacity, which can make it a viable option.

An existing hydropower plant with a Francis turbine can be rebuilt as a pumped hydropower plant in multiple ways. One possibility is to replace the Francis turbine with a reversible pump turbine (RPT) in its current location, however this can present several problems. An RPT with the same dimensions as a Francis turbine will produce a too low lifting height when using the existing generator as a motor, however this can be solved using a frequency converter or a new generator to increase the rotational speed in pumping mode. A more difficult problem is that of cavitation in pumping mode caused by a too low submergence of the RPT. Cavitation will occur at the inlet of the RPT, and will inevitably destroy the unit after a short time of operation. A proposed solution to this cavitation problem is being investigated for the Norwegian hydropower plant Roskrepp. Placing an axial booster pump at the discharge side of the draft tube to increase the inlet pressure of the RPT in pumping mode is being looked into. The intended effect of the booster pump is to increase the inlet pressure at the RPT in pump mode, as can be seen in figure 1. Preliminary findings suggests that a single-stage axial flow machine will have a very low efficiency for the needed performance requirements at Roskrepp. Moving forward, it will be investigated if a two-stage counter-rotating axial pump will be a better solution for such a booster pump.

1.2. Objective

The objective of the work is to investigate if a two-stage counter-rotating axial pump can be used as a booster pump to avoid having to submerge a reversible pump turbine when retrofitting a hydropower plant with a reversible pump turbine. Roskrepp hydropower plant in Norway has been chosen as a case power plant. It has been estimated that such a booster pump would need to supply a head of $h = 20\text{m}$ at a volumetric flow rate of $Q = 50 \frac{\text{m}^3}{\text{s}}$ at Roskrepp [4].

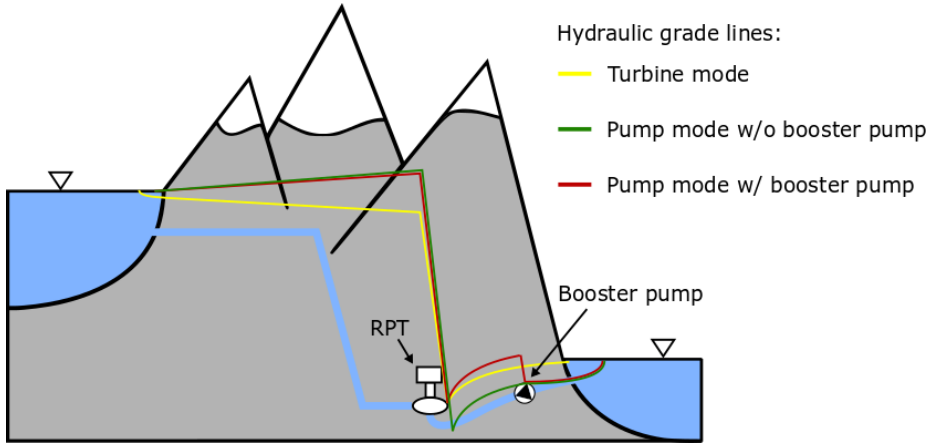


Figure 1. Booster pump effect on hydraulic grade line of a pumped hydropower plant in pump mode.

A user-friendly way of creating counter-rotating propeller designs in Matlab is going to be developed. The possibility of designing hubless, rim-driven thrusters shall be included. Finally, the design method is to be used for the case at Roskrepp to obtain an initial design with acceptable performance. As part of making the design method user-friendly, effort will be put into making a simple graphical user interface.

1.3. Previous work

An initial design for an axial booster pump was made using the open-source code OpenProp, which is made for the design of marine propellers. All the underlying theory of OpenProp was documented to validate the designs made by OpenProp. The initial single stage design created showed a low efficiency and an impractical geometry for the intended use in an existing hydropower plant. Both of these impracticalities are believed to be partly caused by designing outside the valid area of OpenProp's underlying theory. Also, some limiting geometrical constraints regarding the outer diameter of the pump were imposed on the design. These constraints have since been changed.

2. Axial pump design theory and method

2.1. Axial flow pump design

The method utilized in this thesis for designing axial flow pumps is well known propeller design theory. A propeller blade and an axial pump blade are equivalent, and can be represented by a radial lifting line with a radial circulation vortex distribution, as seen in figure 2. This is called moderately loaded lifting line theory. Given a pump diameter

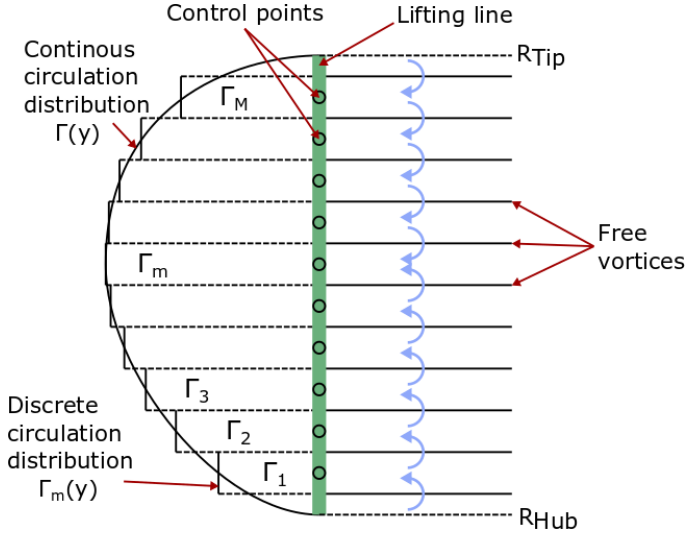


Figure 2. Vortex lattice lifting line representation of an axial pump blade.

and rotational speed, the goal is to optimize a pump for given inflow conditions and a thrust(or head) requirement. The pump is optimized when the torque required to produce the required thrust is at a minimum. Both torque and thrust are functions of the radial blade circulation distribution, so the optimization reduces to finding the optimum circulation distribution. The method of Lagrange multipliers is used to find this distribution along the blades of the axial pump. This optimum distribution enables calculation of the self-induced velocities due to circulation, which together with the inflow and rotational speed enables calculation of the optimum pitch of the pump blades. After the pump pitch is determined, chord length and thickness distributions of the blades can be determined. The entire design procedure can be iterated to find the best combination of pump diameter and rotational speed, which are design input parameters. Often, one of these inputs are fixed by some external factor such as a geometrical constraint. Figure 3 shows a flow chart of the design process utilized in the full thesis. A more detailed version of the design method is described by Epps [5].

2.2. Validity of moderately loaded lifting line theory

The lifting line theory described in section 2.1 is restricted to what is called moderately loaded propellers and pumps. Among other things, this means that a high advance ratio, J_S , is assumed, as is stated by Arán Menéndez [6]. The advance ratio is given by

$$J_S = \frac{V_a}{nD} \quad (1)$$

where V_a is the axial inflow velocity, n is pump rotational speed and D is the pump outer diameter. Values of J_S typically range between 0 and 0.95, so values above 0.8

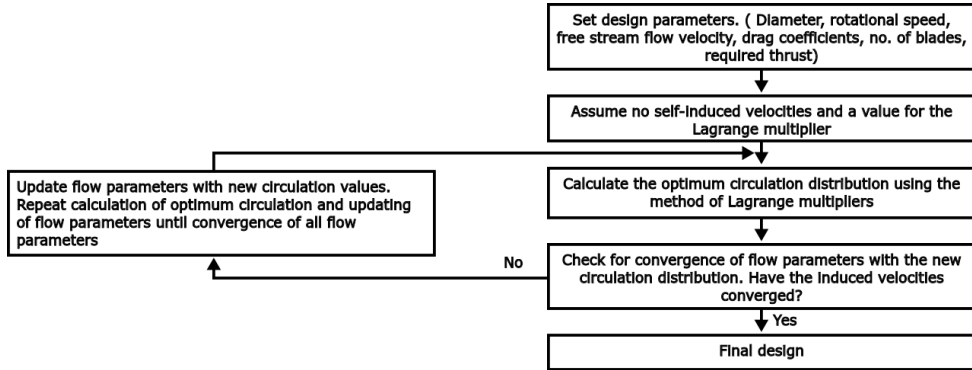


Figure 3. Simplified flow chart for lifting line theory design process.

will be considered high.

Another assumption of the lifting line theory is that the wake of the pump has a helical shape with constant radius and pitch [7]. This is not true for the far wake, but is a valid approximation for near wake calculations.

2.3. Specific speed limitation

The specific speed of a pump is a quasi dimensionless parameter that is used to characterize the pump type and proportions. The definition of the specific speed, N_s , is

$$N_s = \frac{n\sqrt{Q}}{H^{\frac{3}{4}}} \quad (2)$$

where n is the rotational speed in revolutions per minute, Q is the flow in $\frac{l}{s}$ and H is the pump head in meters. Ranges in the pump specific speed define what impeller type is suited for a certain application, based on empirical knowledge. As the specific speed increases, the suitable pump type gradually changes from pumps with radial impellers to axial propeller pumps. It follows from equation (2) that for a given requirement of head and capacity (volumetric flow rate), a higher specific speed pump will have higher rotational speed. The specific speed range well suited for axial propellers is $194 < N_s < 291$ [8]. For the requirements estimated for Roskrepp hydropower plant, this limits the rotational speed of a booster pump to $260 \text{ rpm} < n < 390 \text{ rpm}$, since both Q and H are given.

2.4. Extension to two-stage counter-rotating propeller

A goal of this project is to create a design code in Matlab that facilitates the design of two-stage counter-rotating axial pumps. For such a machine, the interaction between the two stages must be accounted for in the flow conditions of each stage. This is done

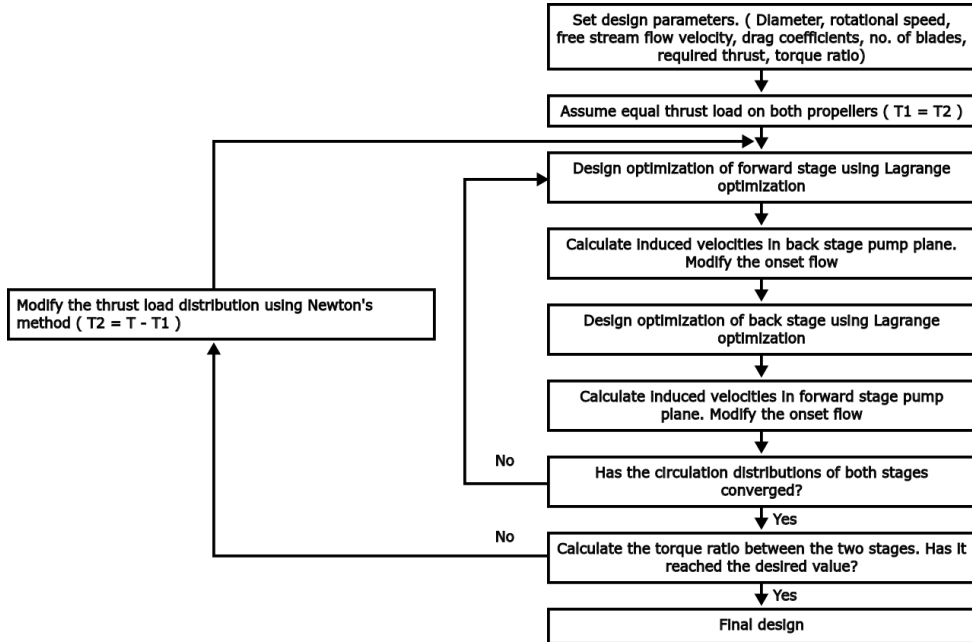


Figure 4. Simplified flow chart showing the design process for two-stage counter-rotating axial pump design.

by computing a set of axial and tangential interaction velocities on each stage from the other stage. These velocities are similar to the self-induced velocities described in section 2.1, but unlike the self-induced velocities, the interaction velocities are not steady. They are therefore time-averaged to make calculations easier.

The optimization scheme of a two-stage counter-rotating pump is similar to the single stage optimization, however it is now the sum of the torques acting on the two components that is to be minimized for a prescribed thrust (or head) requirement. An additional requirement of a torque ratio between the two components is often stated for two-component pumps. This is to ensure that both stages have a desired percentage of the load. The method of Lagrange multipliers is once again used to determine the circulation distribution along the blades. A simplified flow chart showing the design process for two-stage counter-rotating axial pump design is shown in figure 4. A more detailed design process is described by Laskos [9].

3. Results

The preliminary single stage axial booster pump design had an open water efficiency of $\eta_o = 0.49571$. This is lower than what can be expected for a single stage pump without guide vanes. Usual optimum efficiency values are in the range $0.7 < \eta_o < 0.8$ [9]. The

advance ratio, J_S , was 0.53. This is not compliant with the moderate load assumption of lifting line propeller design theory, as stated by Menéndez [6].

4. Discussion

The preliminary pump design shows too poor performance to be an expedient design. Both the validity of the design theory and the design constraints can be causes for this low efficiency design. A constraint was put on the outer pump diameter to make the pump fit snugly in the draft tube exit, thus making the outer diameter of the pump very large. This constraint is not needed if there is built a narrowing of the draft tube right before the booster pump. Not having a constraint on the pump diameter allows for more freedom when designing the pump by allowing more combinations of diameter and rotational speed, which are the two main design input parameters. A smaller diameter will increase the axial inflow velocity, and from equation (1) we see that this will increase the advance ratio. A higher advance ratio will possibly increase the validity of the design with respect to the moderately loaded assumption.

A two-stage counter-rotating axial pump design has several advantages over a single-stage design. First and foremost, counter-rotating designs usually have higher efficiencies than single-stage designs [10]. Additionally, a two-stage pump can have smaller dimensions than a single stage pump. This can make transportation and installation of the pump easier. If the dimensions are too big, boring may be required to place the booster pump in the desired location. This is more costly and more time-consuming than being able to fit the booster pump through the existing tunnels in the hydropower plant. Also, having a counter-rotating pump will allow for flow control that would otherwise be impossible with only one stage. With two counter-rotating stages the rotation in the flow leaving the pump can be controlled to either give a desirable inflow for the RPT in pump mode, or the flow rotation can be minimized to increase the efficiency of the booster pump. Furthermore, multiple stages will reduce the load on each stage.

There are also some disadvantages with a counter-rotating two-stage design. They are more mechanically complex than single-stage designs and can therefore be more expensive to build. Furthermore, they have more moving parts, which can cause more noise and vibrations.

5. Conclusion

Adding a booster pump at the draft tube exit of a hydropower plant to avoid having to submerge a reversible pump turbine is being investigated. A single stage axial booster pump has been designed for Roskrepp hydropower plant, but the design showed poor performance. The poor performance is believed to be caused by designing outside the valid area of the applied design theory, and also by geometrical constraints put on the

design. A counter-rotating axial pump design is speculated to have better performance for the intended use, and will be looked into.

6. Further work

There are several interesting possibilities for further work. Many practical issues need to be investigated for the booster pump project to advance, such as how to place the booster pump at the exit of the draft tube. Does it have to be mounted directly in the discharge tunnel or can it be transported in one piece into the tunnel? The pump will have a smaller diameter than the outlet of the draft tube. Therefore, some sort of convergent-divergent pipe needs to be placed at the draft tube exit to guide the water through the booster pump. Another interesting continuation of this project is to determine what should happen to the booster pump when the RPT is working in turbine mode. Some of the prevailing possibilities include removing the pump from the draft tube, letting the pump go to runaway speed or using it as a turbine for power production.

Furthermore, the design code can be extended to represent the wake flow more realistically. As of now, the wake is represented by trailing vortices with constant pitch and constant radius. A more realistic representation will include a certain narrowing of the wake flow and/or wake roll up effects [9]. The optimization scheme for counter-rotating propellers does not include chord optimization as of now. A natural step in the chord optimization would be to calculate the local blade stresses of the booster pump.

References

- [1] Norwegian Ministry of Petroleum and Energy. Features of the norwegian power supply system, 2019. [Online; accessed 18-March-2019].
- [2] Pumped storage hydropower to turbocharge the clean energy transition. *Contify Energy News*, 2018.
- [3] Norges vassdrags- og energidirektorat. NVE Vannkraft utbygd og ikke utbygd, 2019. [Online; accessed 4-March-2019].
- [4] Pål-Tore Selbo Storli. Planning meeting for booster pump project, 2019. Requirements estimated by Rainpower for Roskrepp hydropower Plant.
- [5] B.P. Epps and R.W. Kimball. Unified rotor lifting line theory. *Journal of Ship Research*, 57(4):181–201, 2013.
- [6] D.H. Menéndez Arán and S.A. Kinmas. On fully aligned lifting line model for propellers: An assessment of betz condition. *Journal of Ship Research*, 58(3):130–145, 2014.
- [7] B. Epps. On the rotor lifting line wake model. *Journal of Ship Production and Design*, 33(1):31–45, 2017.
- [8] A. J Stepanoff. *Centrifugal and axial flow pumps : theory, design, and application*. Wiley, New York, 2nd ed. edition, 1957.
- [9] Dimitrios Laskos. *Design and cavitation performance of contra-rotating propellers*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [10] JS Vanderover and KD Visser. Analysis of a contra-rotating propeller driven transport aircraft. *AIAA Student Paper Competition*, 2006.

