NTNU – Trondheim
Norwegian University of
Science and Technology

Development of a Simulation Model for Propeller Performance

Øyvind Øksnes Dalheim

Marine Technology
Submission date: June 2015
Supervisor: Sverre Steen, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE TECHNOLOGY

SPRING 2015

FOR

Øyvind Øksnes Dalheim

Development of a simulation model for propeller performance

As part of the research project HyDynPro, which is aiming at finding the reasons for in-service problems of the lower bevel gear of azimuthing thrusters, a simulation model for the forces and dynamic response of the thruster drive train is under development. The propeller is the main source of excitation of the dynamic response of the thruster drive train, so a simulation model of the propeller forces is an essential part of the mentioned simulation model of the propeller.

In the project thesis, the candidate developed a simplified 6 DoF propeller simulation model, using only curve-fit methods for the average forces and a fully empirical method for adding high (blade-pass) frequency harmonic variations.

The aim of the master thesis is to develop a simulation model for the 6 DoF propeller forces, which is more physically based. It shall provide 6 DoF forces in the time-domain, being function of propeller speed, water inflow velocity and direction, as well as propeller submergence. Being a more physically based model, the propeller geometry shall also be taken into account. The chosen method shall be computationally efficient, so that implementation in an efficient time-domain simulation model is possible. The resulting method shall be implemented in a suitable form, and validated against measurements (to be provided to the candidate).

Furthermore, it is expected that the state of knowledge in the field of hydrodynamic simulation models of propellers is established in the thesis, based on a thorough literature study.

The propeller model shall output propeller forces at the propeller shaft in all six degrees of freedom. It is recommended that the propeller simulation model is kept fairly simple. However, there are a number of effects that still need to be included. These effects include the change of operating point (advance number), effect of oblique inflow due to azimuth angle change, relative vertical motions due to ship motions and waves, and due to changes in submergence, including ventilation and propeller out of water. This means that response in three different frequency ranges shall be covered, even though the representation might be simplified. These frequency ranges are: very low frequency (change of speed of the ship and propeller, and change of azimuth angle), wave encounter frequency, and propeller blade passing frequency.

The model shall be tuned and validated against measurement data (model and/or full scale), which will be made available to the candidate.

The model shall be implemented in Matlab in a form suitable for integration with the global thruster simulation model.

In the thesis the candidate shall present his personal contribution to the resolution of problem within the scope of the thesis work.

Theories and conclusions shall be based on mathematical derivations and/or logic reasoning identifying the various steps in the deduction.

The thesis work shall be based on the current state of knowledge in the field of study. The current state of knowledge shall be established through a thorough literature study, the results of this study shall be written into the thesis. The candidate should utilize the existing possibilities for obtaining relevant literature.

The thesis should be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, reference and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work. The plan should include a budget for the use of computer and laboratory resources that will be charged to the department. Overruns shall be reported to the supervisor.

The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

The thesis shall be submitted electronically (pdf) in DAIM:

- Signed by the candidate
- The text defining the scope (signed by the supervisor) included
- Computer code, input files, videos and other electronic appendages can be uploaded in a zip-file in DAIM. Any electronic appendages shall be listed in the main thesis.

The candidate will receive a printed copy of the thesis.

Supervisor : Professor Sverre Steen
Start : 15.01.2015
Deadline : 10.06.2015

Trondheim, 15.01.2015

Sverre Steen
Supervisor

Preface

This work is the result of my master thesis at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU) in Trondheim during spring 2015. The thesis is the final part for getting the degree of *Master of Science*.

The work has its foundation in the research project HyDynPro, and is aimed at developing a time domain simulation model for propeller performance, with calculation of six degrees of freedom propeller forces in various types of environments and inflow conditions.

First I would like to thank my supervisor, Professor Sverre Steen, for his guidance and support throughout the project. I appreciate the way he has managed to balance between providing useful discussions and suggestions, while still encouraging me to face the challenges by myself. I also want to give a thanks to PhD candidate Kevin Koosup Yum for supporting me with the Simulink software, and PhD candidate Bhushan Taskar for providing necessary data and help with the AKPA software. Dr. Kourosh Koushan is acknowledged for providing experimental data for validation purposes, and an additional thanks goes to Dr. Luca Savio for his contributions during the initial period of the work.

Trondheim June 10, 2015

Øyvind Øksnes Dalheim

Abstract

The simulation model PropSiM has been developed and implemented into Simulink. The simulation model generates a time domain solution to the six degree of freedom propeller forces in varying operation conditions, including:

- Change of operating point, including controllable pitch.
- Unsteady axial and tangential inflow wake field.
- Effect of oblique inflow in manoeuvring conditions.
- Reduced propeller submergence.
- Wagner effect.
- Propeller ventilation.

The basic hydrodynamic propeller calculations are performed using propeller vortex lattice lifting line theory. The control and vortex points are cosine spaced along the lifting line, providing efficient convergence even for a limited number of vortex panels.

Solution to the unsteady inflow wake field is found using a quasi-steady approach, where calculations from all the respective blade positions are superimposed to get a representation of the unsteady wake field. This approach has been verified using simulations with the software akpa.

The effect of oblique inflow is found by adapting the inflow wake field to the inflow angle and perform a skewed propeller wake correction. The effect of partly submergence is treated by forcing the circulation to be zero at the dry parts of the blade. In addition the Wagner effect is included. Ventilation is considered using an analogy to a blade experiencing only static pressure on the pressure side and atmospheric pressure on the suction side, and adding a weight factor for the amount of ventilated propeller disk area.

Comparison of open water test of the model scale KVLCC2 propeller, akpa and PropSiM calculations shows that the predicted thrust and torque values generally are conservative. Yet the development with varying advance number J agrees well with the experiments. In addition the ability to reproduce an accurate spanwise circulation and lift distribution has been validated against akpa calculations. In general PropSiM overestimates the lift, but multiple runs indicate that the lift and circulation distribution are reproduced sufficiently for a wide range of operating points.

The simulation model has been validated against experimental results for four different cases of varying propeller submergence and ventilation mechanisms. The first case is a non-ventilating condition of a propeller being forced in sinusoidal heave motion. The other three cases are all ventilating conditions, while the last two cases are in addition surface piercing conditions. The results show that PropSim has the ability to catch the effects of abrupt changes in operating conditions in a very convincing way.

The model is also validated for oblique inflow calculations against model experiments. The results show that thrust and torque are sufficiently predicted by the simulation model. The vertical side force and bending moment is not as well predicted as the horisontal side force and bending moment. However the total side force and bending moment coefficients are predicted with similar behaviour as found in experiments for increasing advance number and azimuth angles.

Sammendrag

Simuleringsmodellen PropSiM har blitt utviklet og implementert i Simulink. Simuleringsmodellen beregner propellkrefter i seks frihetsgrader i tidsdomene ved forskjellige kondisjoner, inkludert:

- Endret operasjonspunkt, inkludert kontrollerbar stigningsvinkel.
- Inhomogent aksielt og tangentielt innstrømningsfelt.
- Effekt av skrå innstrømning ved manøvreringsoperasjoner.
- Redusert neddykking av propell.
- Wagner effekt.
- Ventilerende propell.

Den grunnleggende hydrodynamiske beregningen er basert på løftelinje teori med sirkulasjonspaneler. Kontroll- og sirkulasjonspunktene er cosinusfordelt langs løftelinjen, noe som sørger for effektiv konvergens selv for et begrenset antall sirkulasjonspaneler.

Løsning av propellkrefter i et inhomogent innstrømningsfelt blir funnet ved en kvasi-statisk tilnærming, der beregninger fra alle respektive bladposisjoner superponeres. Denne tilnærmingen har blitt verifisert ved hjelp av simuleringer i programvaren akpa.

Effekt av skrå innstrømning beregnes ved å dekomponere innstrømningsfeltet til innstrømningsvinkelen, og legge til en korreksjon for skrå propellstrøm. Effekt av delvis neddykking blir behandlet ved å tvinge sirkulasjonen rundt bladet til å være null for bladseksjoner som er ute av vannet. I tillegg blir Wagner effekten lagt til. Ventilasjon blir behandlet ved å bruke en analogi til et blad som opplever kun statisk trykk på trykksiden og atmosfærisk trykk på sugesiden, og legge til en vektfaktor for størrelsen på det ventilerte propellarealet.

Sammenligning med friprøve av KVLCC2-propellen i modellskala, akpa- og PropSiM-beregninger viser at den simulerte trusten og dreiemomentet for det meste er konservativt beregnet. Likevel stemmer utviklingen med fremgangstallet godt overens med forsøk. I tillegg har evnen til å gjengi en nøyaktig spennvis sirkulasjons- og løftefordeling blitt validert mot akpa-beregninger. Generelt overpredikerer simuleringsmodellen løft og sirkulasjon, men en stor samling av beregninger viser at fordelingen av løft og sirkulasjon langs bladet predikeres tilfredsstillende for et bredt spekter av operasjonspunkter.

Simuleringsmodellen har blitt validert mot eksperimentelle forsøk for fire ulike scenarioer av varierende neddykking og ventilasjonshendelser. Det første scenarioet er en ikkeventilerende kondisjon for en propeller som tvinges i en vertikal sinusbevegelse. De andre tre scenarioene er alle ventilerende kondisjoner, mens de to siste i tillegg er overflatepenetrerende. Resultatene viser at PropSim klarer å fange effekter av hurtige trusttap på en overbevisende måte.

Simuleringsmodellen har også blitt validert for beregninger av skrå innstrømning mot eksperimentelle forsøk. Resultatene viser at trusten og dreiemomentet predikeres på en tilfredsstillende måte. Den vertikale sidekraften og bøyemomentet blir ikke like nøyaktig beregnet som den horisontale. Likevel viser resultatene at den totale sidekraften og bøyemomentet predikeres med tilsvarende oppførsel som er funnet gjennom forsøk for økende fremgangstall og innstrømningsvinkel.

Contents

Nomenclature

Abbreviations

BEMT	Blade Element Momentum Teory
BET	Blade Element Teory
CPP	Controllable pitch propeller
DoF	Degrees of freedom
EAR	Blade Area Ratio
HyDynPro	Hydroelastic effects and dynamic response of propellers and thrusters
LL	Lifting Line
MBS	Multibody Simulation

Greek

α_i	Ideal angle of attack of camber profile	[rad]
β	Hydrodynamic angle without induced velocities	[rad]
β_m	Hydrodynamic angle without induced velocities at vortex panel m	[rad]
$\beta_{i,m}$	Hydrodynamic angle inclusive induced velocities at vortex panel m	[rad]
β_i	Hydrodynamic angle inclusive induced velocities	[rad]
β_V	β-factor for ventilation	[-]
β_w	β-factor for lift hysteresis effect	[-]
δ	Azimuth angle	[deg]
Δ_{rel}	Relative change of circulation from previous iteration	[-]
Δ_{tol}	Accepted relative change of circulation from previous iteration	[-]
Δy_v	Relative spanwise size of vortex panel	[-]
Γ	Bound vortex circulation	[m^2/s]
Γ_0	Bound vortex circulation at hub root	[m^2/s]
Γ_m	Bound vortex circulation at vortex panel m	[m^2/s]
∇	Mean free surface	[-]

ϕ	Pitch angle	[rad]
ρ	Water density	[kg/m^3]
θ_e	Angle when blade section entered the water	[deg]

Lowercase

c	Chord length	[m]
c_m	Chord length at vortex panel m	[m]
f_y	Horizontal force local to azimuth	[N]
f_z	Vertical force local to azimuth	[N]
g	Gravitational acceleration	[m/s^2]
h	Propeller submergence	[m]
k	Exponent of torque loss related to thrust loss	[-]
m_y	Horizontal bending moment local to azimuth	[Nm]
m_z	Vertical bending moment local to azimuth	[Nm]
n	Propeller shaft frequency	[Hz]
r_h	Radius of propeller hub	[m]
s_{r_A}	Submergence ratio amplitude	[-]
$s_{r_{max}}$	Maximum submergence ratio	[-]
$s_{r_{min}}$	Minimum submergence ratio	[-]
s_{r_T}	Submergence ratio period	[s]
s_r	Submergence ratio	[-]
t	Time	[s]
t_e	Elapsed time since water entry of blade section	[s]
t_{max}	Highest recommended time step in simulation model	[s]
x_h	Hub radius to propeller radius ratio	[-]
y_c	Relative spanwise coordinate of control point	[-]
y_v	Relative spanwise coordinate of vortex point	[-]
$z_{max(C_{Li})}$	Maximum camber height of the camber profile	[m]
z_{max}	Maximum camber height of blade section	[m]

Uppercase

A_0	Propeller disk area	[m^2]
A_{nv}	Non-ventilated area of propeller disk	[m^2]
A_s	Submerged propeller disk area, non-ventilated	[m^2]
A_v	Ventilated area of propeller disk	[m^2]
C_L	Sectional lift coefficient	[-]
C_Q	Non-dimensional torque of propeller	[-]
C_T	Non-dimensional thrust of propeller	[-]
$C_{Dv,m}$	Viscous drag coefficient at vortex panel m	[-]
C_{Dv}	Viscous drag coefficient	[-]
C_{Fh}	Hub vortex drag coefficient	[-]
C_{fy}	Non-dimensional horisontal force of blade	[-]
C_{fz}	Non-dimensional vertical force of blade	[-]
$C_{L\alpha}$	Sectional lift coefficient due to angle of attack	[-]
C_{L_V}	Fully ventilated lift coefficient	[-]
$C_{L_{PV}}$	Partially ventilated lift coefficient	[-]
C_{Lc}	Sectional lift coefficient due to camber	[-]
C_{Li}	Sectional lift coefficient at ideal angle of attack of camber profile	[-]
C_{my}	Non-dimensional horisontal bending moment of blade	[-]
C_{mz}	Non-dimensional vertical bending moment of blade	[-]
F_h	Hub vortex drag	[N]
J	Advance number	[-]
J_e	Effective advance number	[-]
K_Q	Torque coefficient	[-]
K_T	Thrust coefficient	[-]
$K_b = \sqrt{K_{my}^2 + K_{mz}^2}$	Total bending moment coefficient local to azimuth	[-]
K_{Fy}	Horizontal force coefficient	[-]
K_{fy}	Horizontal force coefficient local to azimuth	[-]
K_{Fz}	Vertical force coefficient	[-]
K_{fz}	Vertical force coefficient local to azimuth	[-]
K_{My}	Horizontal bending moment coefficient	[-]
K_{my}	Horizontal bending moment coefficient local to azimuth	[-]

K_{Mz}	Vertical bending moment coefficient	[-]
K_{mz}	Vertical bending moment coefficient local to azimuth	[-]
K_{Q_0}	Nominal torque coefficient	[-]
$K_s = \sqrt{K_{fy}^2 + K_{fz}^2}$	Total side force coefficient local to azimuth	[-]
K_{T_0}	Nominal thrust coefficient	[-]
M	Number of vortex panels	[-]
R	Propeller radius	[m]
S_{wet}	Wetted span of partly submerged propeller blade	[-]
V	Forward speed	[m/s]
$V_{\infty,m}$	Velocity seen by propeller blade at vortex panel m	[m/s]
V_{∞}	Velocity seen by propeller blade	[m/s]
$V_{a,\delta}$	Axial velocity local to blade at oblique inflow	[m/s]
$V_{a,a}$	Axial velocity local to blade due to axial inflow	[m/s]
$V_{a,t}$	Axial velocity local to blade due to tangential	[m/s]
V_a	Axial inflow velocity	[m/s]
$V_{t,\delta}$	Tangential velocity local to blade at oblique inflow	[m/s]
$V_{t,a}$	Tangential velocity component local to blade due to axial inflow	[m/s]
$V_{t,t}$	Tangential velocity component local to blade due to tangential inflow	[m/s]
V_t	Tangential inflow velocity	[m/s]
Z	Number of propeller blades	[-]
G	Non-dimensional bound vortex circulation	[-]
G_0	Non-dimensional bound vortex circulation at hub root	[-]
G_m	Non-dimensional bound vortex circulation at vortex panel m	[-]

1 Introduction

Most marine vessels are equipped with propellers and thrusters to ensure propulsion, manoeuvring and station-keeping capabilities. Safe and efficient operation of marine vessels are thus highly depending on how these systems perform. In order to provide long life operation of the propellers and thrusters, knowledge of the hydrodynamic forces and responses is important. This master thesis will focus on how to model and simulate hydrodynamic loads on azimuthing thrusters operating in various extreme load conditions. The aim is to develop a hydrodynamic component for a multibody simulation (MBS) model of an azimuthing thruster. The MBS model involves modelling of complete propulsion system, from the engine to the propeller. That is, coupling and interaction between propeller and the engine through gear, shaft and other structural components.

1.1 Motivation and background

The project is a part of HyDynPro, Hydroelastic effects and Dynamic response of Propellers and thrusters, which is a project run by *Rolls-Royce University Technology Centre*. The overall objective is to find the real dimensioning loads on azimuthing thrusters in extreme situations, including propeller-ice impacts and dynamic response effects, for the purpose of solving the in-service problems related to bevel gears, shaft bearings and seals on azimuthing thrusters.

The problems are believed to be related to extreme dynamic loads on the propellers, caused by waves, intermittent ventilation, and strongly oblique inflow. A propeller operating in such conditions is the main source of excitation of the dynamic response of a thruster drive train. A simulation model of the propeller forces is therefore an essential part of the MBS model.

Excitation loads and succeeding responses in the thruster drive train components is largely connected to environmental conditions. The environmental conditions are actually time dependent disturbances causing time dependent propeller reaction forces. The reaction forces are input to the propeller control system, however a more important aspect is that the time varying forces will excite dynamic loads in the thruster drive train. Catching these dynamic variations is important for understanding how the propeller loads are transmitted through multiple mechanical couplings. In addition the study of dynamic variations of excitation loads can be crucial for determining whether or not the dynamic response is extreme enough to cause a translation or change in rotation in all or parts of the propulsion system and drive train (Hutchison et al., 2014).

1.2 Previous work

An attempt of developing an advanced physical based model for hydrodynamic loads on a propeller in various extreme conditions has been made, but it turned out to be unsuccessful. The model became too complex, slow with respect to computational time and also some instability and convergence problems were experienced. During spring 2014 a new project was initiated with the intention of developing a quite simple and computational fast simulation model for time domain calculation of propeller loads. The project evolved through an employment during summer 2014, for which the author was hired. The author continued with the model related to his project thesis during fall 2014, which resulted in the well-functioning simplified simulation model presented in (Dalheim, 2014). The experience gained through this development formed an important foundation for the work related to a more physical based simulation model.

1.3 Scope of work

Dalheim (2014) presented a simplified simulation model, mainly based on curve-fit methods for average forces, geometrical considerations and a fully empirical method for adding high frequency harmonics to the average loads. The aim of the master thesis is to develop a more physical based simulation model, providing more accurate propeller forces in time domain. The simulation model shall take into account:

- Change of operating point, including controllable pitch angle.
- Angular inhomogeneous axial and tangential inflow velocity.
- Propeller submergence, including ventilation and Wagner effect.
- Effect of oblique inflow in manoeuvring conditions.

Subsequent to the establishment of the simulation model the model shall be tuned and validated against experimental data.

2 Propeller performance simulation

There are several approaches on how to do a propeller analysis that can be used in a propeller performance simulation model. The approaches differ in the variety of computational effort, accuracy, necessary simplifications and limitations, and the applicability varies with the purpose of the model. We mainly distinguish between empirical and numerical methods, although semi-empirical methods exist as well.

This chapter summarises the most common approaches in the field of propeller analysis, and respectively relate them to their applicability for use in a time domain propeller simulation model. A view on the present state in the field of propeller simulation is further established, followed by a discussion and conclusion on principal strategy for building the simulation model.

2.1 Empirical approaches

An empirical approach to propeller analysis is by far the most computational efficient. Such methods utilise preprocessed experimental results in order to calculate the basic propeller characteristics. Despite the superior computational speed, empirical methods suffer from lack of physical behaviour. That is, such methods are only capable of replicating average forces, meaning that all the propeller harmonics are lost. In addition the empirical approaches can only deal with homogeneous inflow. Dalheim (2014) successfully presented a simplified time domain propeller simulation model based on purely empirical approaches. The accuracy of the model obviously were limited and the relation to the propeller harmonics non-physical, however the computational speed were superior and made the model suitable for several purposes. The following sections introduce two empirical approaches that can be implemented in a simulation model.

2.1.1 Open water model test

One empirical approach is to use open water model test data for the relevant propeller, represented by polynomial fitted curves. In the case of a controllable pitch propeller (CPP) the open water characteristics should be available in a multiple set of relevant pitch angles. This approach requires that open water model test data is available for the propeller subject to the simulation.

2.1.2 Propeller series

A second approach is to use open water model test data connected to the main parameters of the propeller. From time to time, propeller models with systematically changes of pitch, blade area and number of blades have been tested. Open water test results with such propellers have formed basis for propeller diagrams, and the primary example of such propeller diagrams is the Wageningen B-series. This series is made by a curvefit to open water characteristics of 120 propellers tested at Netherland Ship Model Basin in Wageningen (Bernitsas et al., 1981). Figure 2.1 shows an example of such a propeller diagram. This approach does not require that open water model test data is available for the propeller subject to the simulation. Only the main parameters of the propeller is relevant for the analysis.

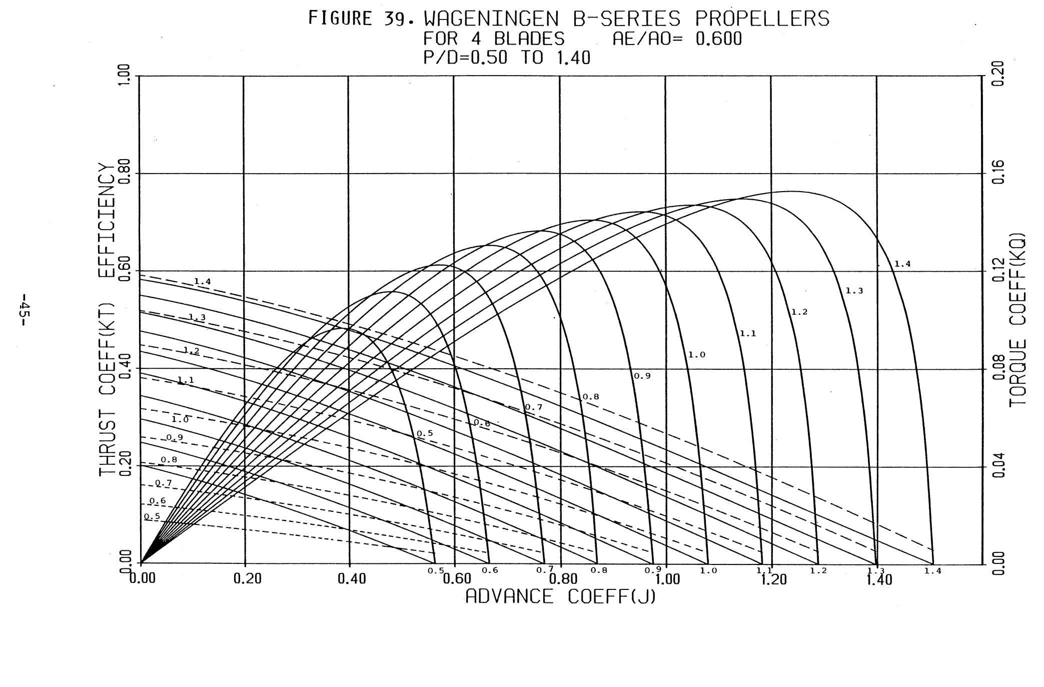


Figure 2.1: Example of Wageningen B-series propeller diagram (Bernitsas et al., 1981).

2.2 Numerical approaches

A number of numerical methods can be applied for the purpose of finding the hydrodynamic force and moment generated by a propeller. Numerical methods include both potential and viscous flow methods. The different methods spans over a large field regarding computational effort, basic assumptions, simplifications and validity ranges. The aim of this section is to briefly look at the most relevant numerical methods used in propeller design and analysis, and connect them to the applicability related to a time domain simulation model of propeller forces.

2.2.1 Momentum Theory

The simplest possible idealisation of a propeller is by treating the propeller as an actuator disk. That is, the physical propeller is replaced by a permeable disk of equal radius. The actuator disk causes an instantaneous uniform pressure jump Δp in the fluid, which can be related to the change in the fluid velocity within the slipstream. The thrust, torque and delivered power can be attributed to this change in fluid velocity (Rankine, 1865; Froude, 1911, 1889). The method can be useful for maximum efficiency calculations as well as an estimate for the propeller induced velocity. In addition the computational speed is superior to other methods. However, momentum theory does not provide any information on the differential propeller thrust and torque at a given blade section. Thus it is considered too simple and non-physical for application to propeller design or propeller analysis. In relation to a 6 DoF time domain propeller simulation model all the physics behind propeller harmonics, variable inflow conditions, ventilation effects and propeller side forces and bending moments will be completely lost. Thus the momentum theory is considered to be far off the level of accuracy wanted for the simulation model.

2.2.2 Blade Element Momentum Theory

The Blade Element Momentum Theory (BEMT) is based on a combination of Blade Element Theory (BET) and Momentum Theory. BET calculates the forces and moments acting on a blade from a finite number of independent blade sections. The blade sections are treated as two-dimensional foils subject to an angle of attack relative to incoming fluid flow. Unlike the momentum theory, BET considers the geometrical properties of the blade sections in order to determine the differential blade forces. Thus the forces exerted on the blade elements are determined from the two-dimensional lift and drag characteristics of the blade section shape in addition to the orientation relative to the incoming flow. A BEMT model combines the two-dimensional behaviour of the blade provided by BET with the change in fluid momentum found using momentum theory. An iterative process between calculating blade section thrust and torque using BET and finding the increase of axial and angular momentum using momentum theory is done for each blade section. This consequently leads to a total thrust and torque by integration over the propeller disc. The advantage of BEMT over more advanced methods is that it allows the lift and drag properties of the two-dimensional sections to include viscous effects such as stall and the effect of laminar separation at low Reynolds numbers by using empirically based lift and drag curves for the blade sections (Amini, 2011).

With respect to simplicity of the propeller calculations, BEMT has the great advantage of modelling the blade as a set of two-dimensional independent foil sections. However, due to the fact that the blade sections are treated independently, the method is less valid for high spanwise pressure variations. That is, the accuracy decreases with increased propeller loading due to large pressure gradients across the span. In relation to a 6 DoF time domain propeller simulation model BEMT is considered to be applicable. Sufficient accuracy can be achieved within acceptable limits of computational time, and information regarding sectionwise thrust and torque enables calculation of side forces and bending moments. A modification can also be made for the purpose of treating oblique inflow (Amini, 2011), and the influence of propeller ventilation can be implemented. For the propeller simulation model the drawback of BEMT reveals itself when it comes to reducing the propeller submergence. Due to the fact that each blade section is solved independently of the adjacent sections there will be no corrections made to the submerged part of the propeller blade as parts of the blade gets dry. The change of propeller forces is a result of the reduced fluid momentum change only, and the physics related to lack of induced velocity from dry parts of the blade is completely lost.

2.2.3 Lifting Line

Lifting Line Theory represents the propeller blades as lifting lines, which allows for a spanwise varying circulation distribution. Similar to BEMT, lifting line theory has the powerful advantage of simplifying a three-dimensional problem down to a finite number of two-dimensional problems. In BEMT however the amount of influence that the blade sections have on each other is completely lost, while lifting line theory is able to account for the flow characteristics at all blade sections. Lifting line theory is considered to be applicable in relation to a 6 DoF time domain propeller simulation model. Sufficient accuracy can be achieved within acceptable limits of computational time, and information regarding sectionwise thrust and torque enables calculation of side forces and bending moments. In addition, lifting line theory has the advantage over BEMT in the way that the physical behaviour of reduced propeller induces velocities is retained even at a reduced propeller submergence. The lifting line approach is only valid for moderately loaded high aspect ratio propellers, and is unable to capture the behaviour of stall (Lerbs, 1952).

2.2.4 Lifting Surface

In the Lifting Surface method the propeller blade is represented by an infinitely thin surface lying on the blade mean camber line. This means that the problem no longer is two-dimensional. Similar to the lifting line approach a distribution of circulation is used in the spanwise direction. However, in the lifting surface method the circulation is also distributed in the chordwise direction, leading to a sheet of vorticity lying on the camber line. The lifting surface method accounts for the blade geometry to a larger degree. However, due to the larger amount of unknown circulation, the computational effort is significantly larger compared to the lifting line approach. In relation to a time domain simulation model the lifting surface method would be more accurate than lifting line, due to the chordwise representation of the blade geometry. The lifting surface approach includes the physical behaviour of the propeller induced velocities to a larger degree, and is able to account for inhomogeneous inflow in an unsteady propeller calculation (Kerwin and Lee, 1978).

2.2.5 Panel method

Panel methods are very similar to Lifting Surface methods, however that the problem is further extended by including the blade thickness and the hub by a finite number of vortex panels. Instead of using the mean camber surface, the blade surface is discretised with a distribution of source or dipole panels. The use of panel method for a propeller in open water condition is known to predict the propeller torque and thrust with good accuracy (Amini, 2011). In relation to a time domain simulation model a panel method would be more accurate than lifting line and lifting surface, due to the thickness representation of the blade geometry in addition to the chordwise representation. The computational time would however be a major drawback for time domain simulation.

2.2.6 Reynolds Averaged Navier Stokes

Reynolds Averaged Navier Stokes (RANS) calculations solves the averaged flow field by modelling a full three-dimensional viscous flow field using a finite volume or finite element approach. Time domain solution of propeller forces by utilising RANS calculations would be very time consuming, and not suitable for the purpose of this model. In addition the complexity would expand to a whole new level when it comes to including different variable working conditions as ventilation and surface penetration. Califano (2010) discussed the difficulty to accurate capture the pressure in the tip vortex, which both experiments and numerical calculations have shown to be a key mechanism for ventilation. This is due to the fact that every attempt to refine the grid in order to better capture the low pressures within the tip vortex, result in an increase of the already long computational time in addition to worsen the stability of the simulation. Califano (2010) therefore suggests to use simpler, less computationally expensive methods to capture the main mechanisms of a ventilating propeller.

2.3 Present state

Available methods for propeller force modelling are numerous, and the present knowledge on propeller performance simulation models is wide and extensive. A considerable amount of the knowledge originates from aerodynamics, where simulation models used for aeroelastic studies have been sought after for several years. For marine propellers the objective has been related to methods for propeller design, and high quality approaches for detailed study of hydrodynamical effects. That is, boundary element methods and other three-dimensional approaches has been developed and further improved for the purpose of accurate prediction of propeller characteristics in various conditions. The continuous development of computational capacity has gradually also introduced computational fluid dynamics (CFD) to the field of marine propellers.

The present state of propeller performance modelling is extensive, however the objective of the succeeding sections is to reflect on today's knowledge related to propeller simulation and introduce some established approaches on this field.

2.3.1 OpenProp

OpenProp is a free software that can be used for design and analysis of marine propellers (Epps and Kimball, 2013). Regarding propeller performance analysis it provides performance curves for a given rotor geometry, blade cavitation analysis and propeller design optimisation using lifting line theory. Several functionalities are available in a graphical user interface, which offer a variety of options for two- and three-dimensional graphical representations. That is, circulation distribution, three-dimensional propeller geometry, sectional lift curve, induced velocities etc.

OpenProp has achieved good reviews and is acknowledged for the related work. Further enhancement of the code is also planned, which will increase the knowledge on adapting lifting line theory to account for additional physical effects, like for example compressible flow corrections and implementation of rotors with rake and skew. The software is currently not adapted to angular inhomogeneous inflow conditions, and does not cover any effects of changing the propeller submergence or inflow angle. Implementation of OpenProp into a time domain simulation model is possible, however the amount of necessary modification is substantial.

2.3.2 AeroDyn

AeroDyn is a set of routines used to predict the aerodynamics of horisontal axis wind turbines (Moriarty and Hansen, 2005). The routines are used in combination with several aeroelastic simulation codes, that is, YawDyn, FAST, SymDyn and ADAMS.The aeroelastic simulator calls for the AeroDyn routines at each time step, enabling time domain simulation for investigation of change in aerodynamic forces.

A number of different models are used in AeroDyn, each of them applicable to different simulation purposes. The most important feature is the wake model implementation, which contains two options for wake modelling: the blade element momentum theory and the generalised dynamic wake theory. The generalised dynamic wake model is an acceleration potential method, which allows for a more general distribution of pressure than blade element momentum theory. Other advantages is inclusion of dynamic wake effect, tip losses and skewed wake dynamics (Moriarty and Hansen, 2005). However as with other rotor simulation models the method was developed for lightly loaded rotors, and suffers from computational instability at low inflow velocities.

2.3.3 Boundary Element Methods

Boundary element methods have been used for the solution of propeller design and analyses for several years. The first three-dimensional boundary element model developed for investigation of steady flow around a marine propeller can be attributed to Hess and Valarezo (1985). Their approach was related to the representation of a steadily translating and rotating propeller, where the trailing vortex surface from the trailing edge was modelled using a prescribed wake shape (Politis, 2004). After the publication of this pioneering work a large number of publications regarding different forms of boundary element methods and free wake modelling appeared. Increasing computer power enabled the application of the wake relaxation method to the solution of steady state flow problem, which further resulted in a number of papers on using boundary element approaches on unsteady flows.

Politis (2004) developed a boundary element method for simulation of unsteady motion of a propeller using a time-stepping approach. The problem was formulated for the flow of a propeller performing an unsteady translation and rotation with instantaneous velocity and rotational vectors. The model was used to simulate a propeller in inclined flow and a propeller in heaving motion, and the conclusion was that the model could provide very accurate results compared to experiments.

The time-stepping approach of the boundary element method is well suited for application to a time-domain simulation model. As the translation and rotation velocities are instantaneous vectors, the method is already prepared for implementation of various inflow conditions, that is, manoeuvring situations, heaving and surface piercing propeller and other transient operational conditions. The drawback is however the amount of computational effort required to carry out a full time-domain simulation of the operating condition of interest. In combination with a MBS-model of a complete thruster drive train the computational time will be considerable. Hence this approach may be unbalanced in terms of matching the required accuracy of the model with the demand of computational power.

2.3.4 Blade Element Momentum Methods

Several applications of blade element momentum theory in terms of propeller performance simulation can be found in the literature, especially connected to aerodynamics. BEMT is in fact one of the most common engineering models for computation of the aerodynamic loads on wind turbine rotors (Madsen et al., 2007). Among the varieties of publications, Rwigema (2010) presented a BEMT-model for analyses of the aerodynamic performance of an aircraft propeller which combined the behaviour of the resultant slip-stream by coupling a vortex wake deflection representation to the propeller forces. More related to marine propellers was the BEMT-model established by Amini (2011) for investigation of azimuth propulsors in off-design conditions, with emphasis on the effect of inclined inflow. The purpose of this model was however not related to a time domain simulation, but to a thorough study of side forces and bending moments due to inclined inflow. The approach to this propeller investigation reveals how a BEMT-approach is suitable for implementation into a simulation model. In addition the overall knowledge on the field of BEMT is comprehensive. Yet, there is shortage of available documentation or publications on the utilisation of BEMT for combining hydrodynamic effects in the time domain.

2.3.5 Nonlinear Dynamic Propeller Model

The knowledge on the field of propeller simulation has not only developed through applications to aerodynamics and aircraft propellers. There is a large amount of publications related to analysis of propulsion control units that treat the problem of modelling propeller hydrodynamics accurately within an acceptable amount of computational power. Traditionally the mathematical models of propeller thrust and torque have been based on model tested steady state thrust and torque characteristics. Accurate thrust control in dynamic positioning and underwater robotics depends on both how well and how fast the propeller hydrodynamics are provided by the model. In that relation the transient effects in a thruster have been found to be important for the control performance.

Blanke et al. (2000) developed a large signal dynamic model of a propeller that included the effects of transients in the flow over a wide range of operation. The model was based on propeller characteristics from open water tests, modified to obtain a dynamic model including additional states. With the intention of creating reliable control performance units, the balance between computational effort and accuracy focuses more towards high computational speed, simply due to the fact that the control units must be able to minimise the time delay in operation. Improved hydrodynamic models which satisfy demands on rapid response are longed for in the field of thruster control. Yet, use of open water propeller characteristics combined with transient states is reliable and utterly computational efficient, which makes it difficult to find an adequate hydrodynamic replacement.

2.4 Strategy for building the simulation model

The two approaches considered to be the most relevant for the purpose of the simulation model are BEMT and lifting line theory. RANS calculations are considered to require an unacceptable amount of computational effort, at least with respect to the range of application for the simulation model. Not to mention the additional modelling trouble that will be present when environmental effects such as ventilation and reduced propeller submergence are included. The analysis will regardless of modelling strategy be exposed to prominent uncertainties. Thus the gain in simulation accuracy with increasing demand of computational effort will be uncertain. Boundary element methods are however better suited for time domain simulation of propeller forces, because the method is more applicable for implementation into a MBS model, and in addition less computational demanding. There is however a risk of getting a slow simulation model using this approach, and combining this method with additional effects such as reduced propeller submergence can be hard to implement within a physical approach.

Both BEMT and lifting line theory is found to be suited for the simulation model, simply because these methods stand out as the only ones with the potential of offering acceptable computational speed. The choice of simulation strategy among these to approaches is however more uncertain, because both accuracy and speed is expected to be on the same level. Yet, the final choice of model strategy is lifting line theory. The theory can handle a general case of a propeller in radially non-uniform inflow with an arbitrary distribution of circulation. As with BEMT the lifting line approach has the powerful advantage of simplifying a three-dimensional problem down to a finite number of two-dimensional problems. However, the lifting line method can also account for the amount of influence between the sections. This is expected to be of particular importance for a propeller operating in the free surface, where parts of the propeller blade will vary between wet and dry.

3 Theory for propeller analysis

Prior to building the complete simulation strategy and structure of the simulation model it is necessary to go into the knowledge and methods of basic propeller characteristics. Further the fundament of the lifting line approach must be established, including relevant equations, assumptions and limitations. In order to obtain a physical approach for including various environmental conditions that a propeller is subject to, the characteristics of these conditions and the strategy of implementing them must be established. The subsequent sections will go into and present all the theories and methods found to be relevant for the propeller simulation model.

3.1 Propeller characteristics

It is convenient to express propeller characteristics by use of dimensionless numbers. Three of the most important numbers are the advance number J, thrust coefficient K_T and the torque coefficient K_Q, shown in (3.1), (3.2) and (3.3), respectively. For a specific propeller geometry, K_T and K_Q are often given as functions of the advance number J. This relationship is commonly referred to as open water propeller characteristics.

$$J = \frac{V}{nD} \tag{3.1}$$

$$K_T = \frac{T}{\rho n^2 D^4} \tag{3.2}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{3.3}$$

where n is propeller shaft frequency, ρ is water density and D is propeller diameter.

In the study of finding the design loads of azimuthing thrusters, all propeller forces in six degrees of freedom (DoF) are of interest. The propeller side forces and bending moments can be expressed dimensionless in a similar way as for thrust and torque.

$$K_{fy} = \frac{fy}{\rho n^2 D^4} \tag{3.4}$$

$$K_{fz} = \frac{fz}{\rho n^2 D^4} \tag{3.5}$$

$$K_{my} = \frac{m_y}{\rho n^2 D^5} \tag{3.6}$$

$$K_{mz} = \frac{m_z}{\rho n^2 D^5} \tag{3.7}$$

Azimuthing thrusters are mostly equipped with controllable pitch propellers (CPP). CPP propellers can offer higher propulsion efficiency, and are particularly suitable for vessels that require good manoeuvrability and other conditions related to variable propulsion power. There are two control parameters for a CPP, the shaft speed and the pitch angle. The pitch angle ϕ consists of one blade geometry dependent part, and one part that can be controlled by adjusting the rotation of the blade root fastening point, hereby referred to as the angle ψ_{cpp}. ψ_{cpp} is defined as positive if it increases the blade pitch angle. The blade geometry dependent part of the pitch angle is often given in terms of the propeller pitch ratio P/D. P/D varies along the span, and it is convenient to express the pitch angle as a function of the relative spanwise coordinate y.

$$\phi(y) = \tan^{-1}\left(\frac{P/D}{y\,\pi}\right) + \psi_{cpp} \tag{3.8}$$

where $y = \frac{r}{R}$

There are other ways of describing the thrust and torque as non-dimensional coefficients. While K$_T$ and K$_Q$ are made non-dimensional by use of the propeller shaft frequency, we can rather express the forces related to the propeller disc area and the forward speed of the ship. The coefficients are referred to as C$_T$ and C$_Q$. Similar expressions can also be established for the side forces and bending moments, based on (3.9) and (3.10) respectively.

$$C_T = \frac{T}{\frac{1}{2}\rho V^2 \pi R^2} = \frac{8}{\rho V^2 \pi D^2} \cdot T \tag{3.9}$$

$$C_Q = \frac{Q}{\frac{1}{2}\rho V^2 \pi R^3} = \frac{16}{\rho V^2 \pi D^3} \cdot Q \tag{3.10}$$

3.2 Lifting Line Theory used in propeller modelling

A sectionwise two-dimensional analysis of a lifting wing has the powerful advantage of simplifying a three-dimensional problem down to a finite number of independent two-dimensional sections. While this finite number of independent two-dimensional sections can be solved quite easy, the amount of influence that the two-dimensional sections have on each other is completely lost. On a three-dimensional finite wing, the local lift on a wing section is strongly affected by the lift on the neighbouring sections. Thus, much of the physics is actually lost by treating the sections independently.

The Lifting Line theory is a two-dimensional mathematical model for predicting the lift distribution over a three-dimensional wing based on its geometry. This theory is powerful because it can treat the wing as two-dimensional sections while still accounting for the influence the two-dimensional sections have on each other. In a propeller analysis the aim is to find the propeller forces related to a certain inflow velocity to the propeller blades. In other words, there is a matter of finding the distribution of lift on the propeller blade for the relevant characteristics of the flow. The wing section lift can effectively be related to the wing section circulation by making use of the Kutta-Joukowski theorem (3.11). The unknown in the propeller analysis then becomes the spanwise distribution of circulation rather than the distribution of lift. This is a crucial step in building the lifting line approach, because the substitution enables to take the wing section reciprocal influence into consideration.

$$dL(y) = \rho V \Gamma(y) \tag{3.11}$$

As a starting point it is important to understand the two-dimensional features of the flow around a blade section. Figure 3.1 shows a two-dimensional velocity diagram of a blade section. $\mathrm{V_a}$ is the undisturbed inflow velocity normal to the propeller disk, and $\mathrm{V_t}$ is the undisturbed in-plane velocity to the propeller disk in tangential direction of the blade. ωr is the velocity component due to the propeller rotational speed. Together these velocities form the undisturbed inflow velocity $\mathrm{V_0}$ oriented with the undisturbed hydrodynamic angle β.

$$\tan\beta = \frac{V_a}{\omega r + V_t} \tag{3.12}$$

In order to generate momentum, and hence propeller thrust and torque, the propeller blade induces velocities in axial and tangential direction. The axial and tangential induced velocity is denoted u_a^* and u_t^* respectively. Together the undisturbed and induced velocities form the total resultant velocity seen by the blade section V$_\infty$ oriented with the hydrodynamic angle β_i, where

$$\tan\beta_i = \frac{V_a + u_a^*}{\omega r + V_t - u_t^*} \tag{3.13}$$

$$V_\infty = \sqrt{(V_a + u_a^*)^2 + (\omega r + V_t - u_t^*)^2} \tag{3.14}$$

The hydrodynamic angle of attack α is then found from the blade section pitch angle ϕ and the hydrodynamic angle β_i.

$$\alpha = \phi - \beta_i \tag{3.15}$$

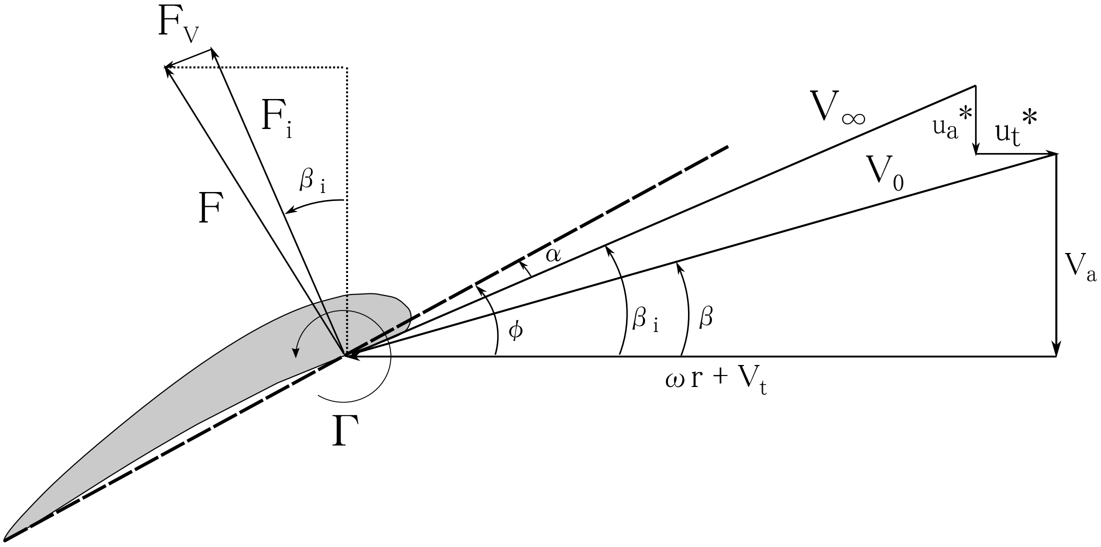


Figure 3.1: Propeller blade section velocity diagram.

Using the Kutta-Joukowski relation (3.11) the inviscid lift force F$_\mathrm{i}$ is found as

$$F_i = \rho V_\infty \Gamma \tag{3.16}$$

The viscous drag force of the blade section is equal to

$$F_V = \frac{1}{2}\rho V_\infty^2 C_{Dv} c \tag{3.17}$$

where

$\mathrm{C_{Dv}}$: Viscous drag coefficient.

c : Chord length.

The total thrust and torque of the propeller is then found by integrating the lift force $\mathrm{F_i}$ and drag force $\mathrm{F_v}$ parallel and normal to $\mathrm{V_\infty}$ over the span of the blade, and multiply with the number of propeller blades Z.

$$T = \rho Z \int_{r_h}^{R} \left[V_\infty \Gamma \cos(\beta_i) - \frac{1}{2}(V_\infty)^2 c\, C_{Dv} \sin(\beta_i) \right] \mathrm{d}r \tag{3.18}$$

$$Q = \rho Z \int_{r_h}^{R} \left[V_\infty \Gamma \sin(\beta_i) + \frac{1}{2}(V_\infty)^2 c\, C_{Dv} \cos(\beta_i) \right] r\mathrm{d}r \tag{3.19}$$

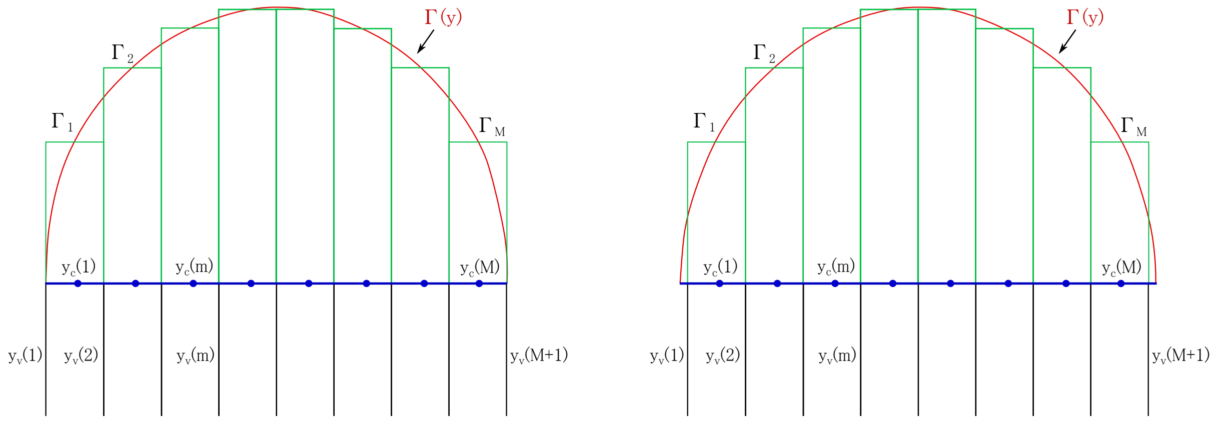
3.2.1 Vortex lattice method

Planar lifting line

Analytic solutions to the linearised problem of a two-dimensional foil section can be found. However, for a three-dimensional foil this is generally not possible, and there is necessary to involve numerical methods for solving such problems (Kerwin, 2001). An efficient approach is by representing the continuous vortex sheet by a lattice of concentrated, straight line vortex elements. The first step is simply to divide the span of the lifting line into M panels. This allows for the continuos vortex sheet $\Gamma(y)$ to be concentrated into discrete point vortices Γ_m within each panel.

The simplest spanwise arrangement of the vortex panels consists of equally spaced panels with no tip inset, and with the control points located in the middle of each panel. An example is shown in Figure 3.2 (a). This arrangement is in general able to predict a circulation distribution with reasonable shape, however the magnitude will be too high. This problem originates from the strength of the continuous free vortex sheet, which has a square root singularity at the tips. This singularity is not well treated by the simple approach of equally spaced vortex panels.

Much better results can be obtained if the tip panels are inset by one quarter of a panel width, see Figure 3.2 (b). This is a very easy modification, which does not cause any



(a) Without panel inset.

(b) With 1/4 tip panel inset.

Figure 3.2: Discrete representation of a continuous circulation distribution $\Gamma(y)$ with equally spaced vortex and control points with and without panel inset.

additional computational effort. Changing the tip inset will influence the accuracy of tip induced velocity, total lift and induced drag. However, no single value of the tip inset can be optimal for all three at the same time, so a tip inset of one quarter panel is considered to be the best (Kerwin, 2001).

Another vortex panel arrangement can be established by using a relationship between the physical coordinate y to an angular coordinate $\tilde{y}$ as shown in (3.20). This approach is called *cosine spacing*, and Lan (1974) proved that this arrangement can give remarkably good results. Yet, if the control points are located midway between the vortices the singularity replication will still be present. The control points should rather be mapped with the same cosine transformation as the vortices, giving the *real cosine spacing* where control points are biased towards the tips (Kerwin, 2001). An example of real cosine spacing is shown in Figure 3.3, and equations for cosine spaced vortex ($y_v(m)$) and control ($y_c(m)$) points distributed between the propeller hub x_h and blade tip are given in (3.22) and (3.21).

$$y = -\frac{s}{2}\cos(\tilde{y}) \tag{3.20}$$

where s is the span of the foil.

$$y_c(n) = x_h + \frac{1}{2}(1 - x_h)\left[1 - \cos((2n-1)\frac{\pi}{2M})\right] \tag{3.21}$$

$$y_v(m) = x_h + \frac{1}{2}(1 - x_h)\left[1 - \cos(2(m-1)\frac{\pi}{2M})\right] \tag{3.22}$$

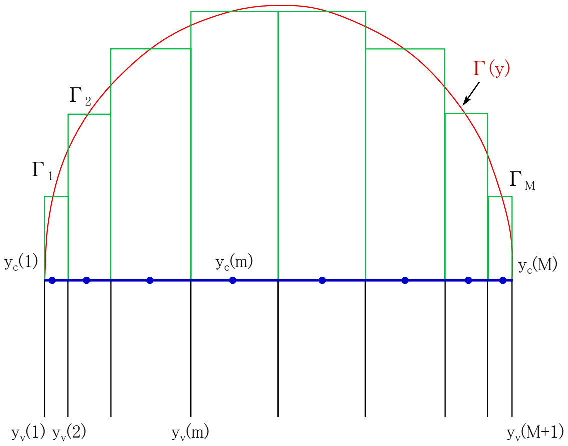


Figure 3.3: Discrete representation of a continuous circulation distribution $\Gamma(y)$ with cosine spaced vortex and control points.

Regardless of panel spacing method the continuous distribution of circulation along the span can be replaced by a stepped distribution. The step in circulation strength takes place at the vortex point, leaving the circulation strength within each panel constant. The value of the circulation for each panel is equal to the continuous circulation distribution at a specified y-value, see Figure 3.3. The purpose of the control points is to find the induced velocity at these locations.

A piecewise circulation distribution introduces a set of concentrated vortex lines shed from the boundaries of each vortex panel. For a continuous circulation distribution the strength of the shed vorticity is equal to the rate of change of the bound vorticity. For a stepped circulation distribution however, the strength of the shed vorticity is equal to the difference in bound vorticity strength across the panel boundary. Each panel has a set of two vortex lines shed from the boundaries. This means in fact that the continuous vortex distribution is replaced by a set of discrete horseshoe vortices, consisting of a bound vortex line (the lifting line) and two concentrated tip vortices. By introducing the discrete circulation distribution the singularity in the induced velocity field integral is avoided. The induced velocity at control point n is rather found by summation over the M vortex panels.

$$w_n^* = \sum_{m=1}^{M} \Gamma_m w_{n,m} = \sum_{m=1}^{M} \frac{\Gamma_m}{4\pi\left(y_v(m) - y_c(n)\right)} - \frac{\Gamma_m}{4\pi\left(y_v(m+1) - y_c(n)\right)} \tag{3.23}$$

where $w_{n,m}$ is the velocity induced at control point n by a unit horseshoe vortex at vortex panel m. We call $w_{n,m}$ the influence function, and it consists of the contribution of two semi-infinite trailing vortices of opposite sign.

Propeller vortex lattice lifting line

The propeller vortex lattice lifting line approach is very similar to a planar lifting line problem. The span of the propeller blade is divided into M vortex panels between the propeller hub and tip. See Figure 3.4 for an illustration of a propeller blade with eight real cosine spaced vortex panels. The distribution of bound circulation becomes, equivalent to the planar lifting line problem, a function of the radial position on the blade, and the continuous distribution is approximated by a set of M vortex elements. The vortex elements are constant in strength on each panel. The vortex system is considered to be built from a set of M horseshoe vortex elements. The horseshoe elements consist of a bound vortex Γ_m and two free vortices of strength $\pm\Gamma_m$. An additional consideration that is crucial for a vortex lattice lifting line approach used on a propeller blade, is the fact that each horseshoe element actually represents a set of Z identical elements originating from each propeller blade. Each one of the horseshoe elements induces both axial and tangential velocity at the control points, and the total contribution comes from a summation of the M vortex panels, as shown in (3.24) and (3.25) (Kerwin, 2001).

$$u_a^*(y_c(n)) = \sum_{m=1}^{M} \Gamma_m \overline{u}_a(n,m) \tag{3.24}$$

$$u_t^*(y_c(n)) = \sum_{m=1}^{M} \Gamma_m \overline{u}_t(n,m) \tag{3.25}$$

where $\overline{u}_a(n,m)$ and $\overline{u}_t(n,m)$ are the horseshoe influence functions.

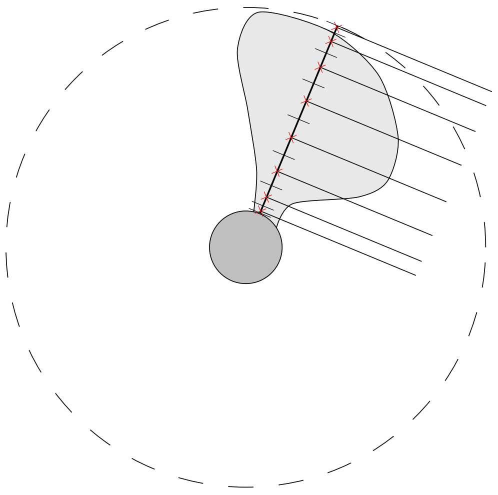


Figure 3.4: Propeller blade with cosine spaced vortex and control points. M=8.

The induced velocity at the control point y_c by a set of Z unit strength helical vortices shed at the vortex point y_v can be expressed as an integral using Biot-Savarts law:

$$\overline{u}_a(r_c, r_v) = \frac{1}{4\pi} \sum_{k=1}^{Z} \int_0^{\infty} \frac{r_v \left[r_v - r_c \cos(\phi + \delta_k)\right]}{\left[(r_v \phi \tan\beta_w)^2 + r_v^2 + r_c^2 - 2 r_v r_c \cos(\phi + \delta_k)\right]^{3/2}} \, \mathrm{d}\phi \tag{3.26}$$

$$\overline{u}_t(r_c, r_v) = \frac{1}{4\pi} \sum_{k=1}^{Z} \int_0^{\infty} \frac{r_v \tan\beta_w \left[(r_c - r_v \cos(\phi + \delta_k)) - (r_v \phi \sin(\phi + \delta_k))\right]}{\left[(r_v \phi \tan\beta_w)^2 + r_v^2 + r_c^2 - 2 r_v r_c \cos(\phi + \delta_k)\right]^{3/2}} \, \mathrm{d}\phi \tag{3.27}$$

where

β_w : Pitch angle of helix at vortex point r_v.

ϕ : Angular coordinate of general point on the helix shed from the actual blade.

δ_k : Angular coordinate of general point on the helix shed from the k'th blade.

The expressions for the induced velocities from Biot-Savarts law is extremely hard to evaluate analytically, so a numerical solution is necessary. Lerbs (1952) solved the potential problem for this type of flow in terms of modified Bessel functions. However, direct evaluation of the modified Bessel functions requires the same amount of computational effort as numerical integration of (3.26) and (3.27). Fortunately highly accurate asymptotic formulas for the sums of modified Bessel functions exist, which enabled Wrench (1957) to develop closed form approximations to the induced velocities. The closed form approximations are given in appendix A.

Knowing the induced velocities enables to determine the hydrodynamic inflow angle β_i. From Figure 3.1 the following kinematic relationship can be established:

$$\omega r \tan\beta_i = \omega r \tan\beta + u_a^* - u_t^* \tan\beta_i \tag{3.28}$$

which can be written as

$$\frac{u_a^*}{V} - \frac{u_t^*}{V} \tan\beta_i = \frac{V_a}{V} \left(\frac{\tan\beta_i}{\tan\beta} - 1 \right) \tag{3.29}$$

By combining (3.29) with (3.24) and (3.25) an equation system of the unknown circulation values Γ_m can be established. The equations can be solved simultaneously in a matrix equation system.

$$\sum_{m=1}^{M} \left[\overline{u}_a(n,m) - \overline{u}_t(n,m) \tan\beta_i(n)\right] \Gamma_m = \frac{V_a}{V} \left(\frac{\tan\beta_i(n)}{\tan\beta(n)} - 1 \right) \quad n = 1 \ldots M \tag{3.30}$$

Both the integral expression of Biot-Savarts law and the closed form approximations of Wrench depends on the pitch angle β_w of the helical surfaces. Linear theory valid for lightly loaded propellers assumes that $\beta_w = \beta$, that is, the inflow angle of the undisturbed flow. The induced velocities however increase with the propeller loading, such that the difference between β and β_i increases. The moderately loaded propeller theory thus makes a better assumption that the free vortices follow helical paths with pitch angle β_i, rather than β (Kerwin, 2001). Using the moderately loaded propeller assumption complicates the calculation of induced velocities, due to the fact that β_i depends on the induced velocities itself. An iterative solution is therefore required.

The resulting circulation distribution found from (3.30) does not take into account the presence of the propeller hub. This is equivalent to assuming that the propeller blade has a free end and that the circulation goes to zero towards the hub. In reality the circulation has a non-zero finite value, and the presence of the hub boundary is equivalent to having zero crossflow through a circle of radius r_h. In a two-dimensional flow it is known that if a vortex located at radius r has an image vortex located at a radius r_i equal to

$$r_i = \frac{r_h^2}{r} \tag{3.31}$$

the total velocity normal to r_h is zero. However, numerical calculations show that (3.31) is amazingly well suited for a helical vortex as well (Kerwin, 2001). To account for the propeller hub using a vortex lattice lifting line approach it is simply sufficient to supplement each helical horseshoe vortex with its image inside the hub. The velocity induced by the image horseshoe vortex located at r_i can be combined with the influence function of the horseshoe vortices along the lifting line, such that no additional unknowns are introduced.

Once the discrete distribution of circulation is known the forces can be calculated by summing up contribution from all the vortex panels. The integral equations for thrust and torque given by (3.18) and (3.19), respectively, can be converted to a discrete sum over the M vortex panels.

$$T = \rho Z \sum_{m=1}^{M} \left[V_{\infty,m}\Gamma_m \cos(\beta_{i,m}) - \frac{1}{2}(V_{\infty,m})^2 c_m C_{Dv,m} \sin(\beta_{i,m}) \right] \mathrm{d}r_m \tag{3.32}$$

$$Q = \rho Z \sum_{m=1}^{M} \left[V_{\infty,m}\Gamma_m \sin(\beta_{i,m}) + \frac{1}{2}(V_{\infty,m})^2 c_m C_{Dv,m} \cos(\beta_{i,m}) \right] r_m \mathrm{d}r_m \tag{3.33}$$

By utilising (3.9) and (3.10) the non-dimensional form of thrust and torque can be expressed. The side forces can be found by decomposing the tangential blade force in horisontal and vertical direction, and the bending moments can be found by multiplying the thrust with the horisontal and vertical distance to the point of attack, i.e. the respective distance to the control points. C_T and C_Q consider the contribution from all the propeller blades, while C_{fy}, C_{fz}, C_{my} and C_{mz} consider only the contribution from the actual propeller blade. Derivation of the formulas is given in Appendix B. For simplicity the following relations are used:

$$F_1 = \frac{V_{\infty,m}}{V}\frac{\Gamma_m}{V\pi D}$$

$$F_2 = \frac{1}{2\pi}\left(\frac{V_{\infty,m}}{V}\right)^2\frac{c_m}{D}C_{Dv,m}$$

The final expressions for the non-dimensional 6 DoF force coefficients are:

$$C_T = 4Z\sum_{m=1}^{M}\Big[F_1\cos(\beta_{i,m}) - F_2\sin(\beta_{i,m})\Big]\,\Delta y_{v,m} \tag{3.34}$$

$$C_Q = 4Z\sum_{m=1}^{M}\Big[F_1\sin(\beta_{i,m}) + F_2\cos(\beta_{i,m})\Big]\,y_{c,m}\,\Delta y_{v,m} \tag{3.35}$$

$$C_{fy} = -4\cos\theta\sum_{m=1}^{M}\Big[F_1\sin(\beta_{i,m}) + F_2\cos(\beta_{i,m})\Big]\,\Delta y_{v,m} \tag{3.36}$$

$$C_{fz} = -4\sin\theta\sum_{m=1}^{M}\Big[F_1\sin(\beta_{i,m}) + F_2\cos(\beta_{i,m})\Big]\,\Delta y_{v,m} \tag{3.37}$$

$$C_{my} = -4\cos\theta\sum_{m=1}^{M}\Big[F_1\cos(\beta_{i,m}) - F_2\sin(\beta_{i,m})\Big]\,y_{c,m}\,\Delta y_{v,m} \tag{3.38}$$

$$C_{mz} = -4\sin\theta\sum_{m=1}^{M}\Big[F_1\cos(\beta_{i,m}) - F_2\sin(\beta_{i,m})\Big]\,y_{c,m}\,\Delta y_{v,m} \tag{3.39}$$

where $\Delta y_{v,m} = y_v(m+1) - y_v(m)$

The hub image correction, involving a finite circulation towards the hub, is equivalent to having a circulation inside the hub. In reality this circulation must be shed into the flow downstream of the hub, which forms a concentrated hub vortex. The presence of a concentrated hub vortex contributes to the propeller thrust. In the core of the hub vortex there is a low pressure region, causing a drag force. To obtain a physically realistic result of the drag force the hub vortex must be modelled as one single vortex of finite strength and core radius. Wang (1985) developed an expression for the resulting pressure force acting on the downstream end of the hub.

$$F_h = \frac{\rho}{16\pi}\left(\ln(\frac{r_h}{r_{hv}}) + 3.0\right)(Z\Gamma_0)^2 \tag{3.40}$$

where r_h is the hub radius, r_{hv} is the core radius of the hub vortex and Γ_0 is the blade root circulation.

In non-dimensional form the hub vortex drag force is:

$$C_{Fh} = \frac{1}{2}\left(\ln(\frac{r_h}{r_{hv}}) + 3.0\right)(ZG_0)^2 \tag{3.41}$$

where $G_0 = \frac{\Gamma_0}{\pi V D}$

The equation system in (3.30) does not take into account the geometry of the propeller blades, the circulation distribution is just related to the induced velocities through the inflow angle β_i and the horseshoe influence functions $\overline{u}_a$ and $\overline{u}_t$. The relation to the blade geometry can be established by use of the blade section lift coefficient from the law of Kutta-Joukowski.

$$C_L = \frac{dL}{\frac{1}{2}\rho {V_\infty}^2} = \frac{\rho V_\infty \Gamma}{\frac{1}{2}\rho {V_\infty}^2} = \frac{2\Gamma}{V_\infty c} \tag{3.42}$$

The sectional lift coefficient can also be expressed by superposition of lift from camber and angle of attack, given in (3.43).

$$C_L = C_{Lc} + C_{L\alpha} = \frac{z_{max}}{z_{max(C_{Li})}} C_{Li} + 2\pi(\phi - \beta_i - \alpha_i \frac{z_{max}}{z_{max(C_{Li})}}) \tag{3.43}$$

where

C_{Li} :	Lift coefficient at ideal angle of attack of camber profile.
α_i :	Ideal angle of attack.
z_{max} :	Maximum camber height of blade section.
$z_{max(C_{Li})}$:	Maximum camber height of the camber profile.

By equating (3.42) and (3.43) the effective inflow angle β_i can be expressed as a function of the circulation Γ, blade section pitch angle ϕ and ideal angle of attack α_i.

$$\beta_i = \left(\frac{C_{Li}}{2\pi} - \alpha_i\right)\frac{z_{max}}{z_{max(C_{Li})}} + \phi - \frac{\Gamma}{V_\infty\, c\, \pi} \tag{3.44}$$

Solving for the unknown circulation distribution

In moderately loaded propeller theory, the induced velocities become function of the effective inflow angle β_i which itself depends on the induced velocities. The relation between the circulation and the effective inflow angle β_i is established in (3.44). The solution of the circulation distribution from the equation system in (3.30) results in an effective inflow angle β_i using (3.44). This is again input to the equation system, which provides a new β_i. Thus the solution to the unknown circulation distribution is found in an iterative process.

Fundamental assumptions

The moderately loaded lifting line theory has the following fundamental assumptions:

- Incompressible and inviscid flow
- Circumferentially homogeneous flow.
- High aspect ratio blades.
- Trailing vortex considered as helix with fixed radius and pitch.

3.3 Thrust loss

A ship operating in high waves can experience large vertical motions relative to the free surface during both low speed operations as well as transit conditions. For the relevant propellers this can lead to a rather frequent change of working conditions. The result is large and abrupt thrust losses, which can be up to 70%-80% of the nominal thrust (Califano, 2010). With respect to the propeller shaft loads, and hence the forces acting on the lower bevel gear of azimuthing thrusters, the dynamics and magnitude of these thrust losses can be of major importance.

To explain the physics of the thrust losses we can separate into contribution from three effects: *loss of effective propeller disc area*, *ventilation* and the *lift hysteresis effect*. With respect to the mean thrust reduction during a complete propeller revolution we can establish reduction factors for each of the three contributions. The total factor is denoted

β, which is the ratio between the ventilating and non-ventilating thrust, where

$$\beta = \beta_0 \beta_V \beta_H \tag{3.45}$$

and

β_0 : Reduction factor due to loss of effective propeller disc area
β_V : Reduction factor due to ventilation.
β_H : Reduction factor due to the lift hysteresis effect.

Thrust and torque losses are closely related. However, due to the drag force on the propeller the change in K_Q is not the same as the change in K_T. Model tests indicate that

$$K_Q = \beta^k K_{Q0} \tag{3.46}$$

where k is a constant between 0.80 and 0.85 for open propellers and K_{Q0} is the torque coefficient for the deeply submerged propeller (Faltinsen et al., 1981). This empirical based relationship has later been confirmed by Kozlowska et al. (2009) based on new experimental results. A physical explanation of (3.46) is the fact that the efficiency should not increase with the thrust losses, i.e. the torque reduction factor should always be larger than the thrust reduction factor.

3.3.1 Loss of propeller disc area

The propeller submergence ratio s_r is considered as the ratio between the propeller submergence h and the propeller radius R.

$$s_r = \frac{h}{R} \tag{3.47}$$

The propeller submergence h is defined as the distance from the centre of propeller boss to the free surface ∇, with positive direction measured downwards from the free surface, see Figure 3.5. This means that $s_r = 1$ as the propeller blade penetrates the free surface, and $s_r = -1$ as the propeller leaves the water, as shown in Figure 3.6.

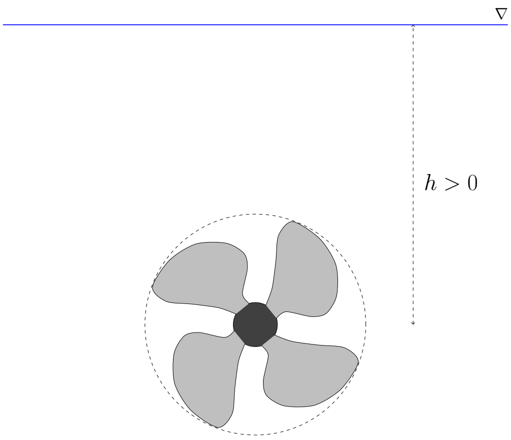


Figure 3.5: Definition of propeller submurgence $h > 0$.

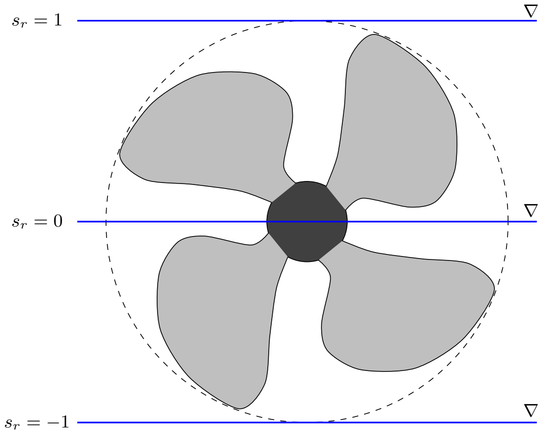


Figure 3.6: Definition of the submurgence ratio s_r for a propeller going out of water.

The propeller will experience a thrust loss during the water exit phase of each blade. This comes from loss of the effective disc area (Gutsche, 1967; Fleischer, 1973). Koushan (2004) derived a formula for the thrust diminuation factor due to loss of propeller disk area, by assuming that the resulting thrust is proportional to the submerged propeller disc area. In his formula he included the effect of the propeller hub.

$$\beta_0 = \frac{A_s}{A_0} = \left[0.5 + \frac{\sin^{-1}(s_r)}{\pi} + \frac{s_r}{\pi}\sqrt{1-(s_r)^2}\right]\left[1 - \frac{|s_r + x_h| - (s_r + x_h)}{2(1-x_h)}\right] \tag{3.48}$$

where

A_s : Submerged propeller disk area.

A_0 : Propeller disk area.

s_r : Submergence ratio.

x_h : Propeller hub ratio.

Dalheim (2014) discussed this formula in relation to simple geometric considerations, and presented a new formula for the loss of propeller disc area. For simplicity the formula is presented as a tripartite expression for submergences above, inside and beneath the propeller hub.

$$\beta_0 = \begin{cases} \dfrac{1-2{x_h}^2}{2(1-{x_h}^2)} + \dfrac{\sin^{-1}(s_r)+s_r\sqrt{1-{s_r}^2}}{\pi(1-{x_h}^2)} & \text{if } s_r \geq x_h \\ \dfrac{1}{2} + \dfrac{\sin^{-1}(s_r)+s_r\sqrt{1-{s_r}^2}}{\pi(1-{x_h}^2)} \\ \quad - \dfrac{{x_h}^2\sin^{-1}(s_r/x_h)+s_r\sqrt{{x_h}^2-{s_r}^2}}{\pi(1-{x_h}^2)} & \text{if } -x_h < s_r < x_h \\ \dfrac{\pi-\cos^{-1}(s_r)+s_r\sqrt{1-{s_r}^2}}{\pi(1-{x_h}^2)} & \text{if } s_r \leq -x_h \end{cases} \tag{3.49}$$

Use of either (3.48) or (3.49) is convenient if the propeller dynamics are of low importance, i.e. only the the average forces are of interest. Using these geometrical properties in fact means that the propeller harmonics for a partially submerged propeller are assumed equal to the deeply submerged propeller. For the aim of developing a propeller simulation model the propeller dynamics are of considerable importance. Due to the fact that fluctuating water entry and exit will lead to important propeller dynamics, neither one of the presented formulas is suitable for the simulation model. The thrust loss must rather be treated local to each blade. The strategy relevant for the simulation model is discussed in section 5.4.1.

3.3.2 Ventilation

When the propeller submergence gets sufficiently low the propeller will not only experience thrust loss due to reduced propeller disk area, but also due to ventilation. During ventilation air gets dragged down to the suction side of the propeller disk, causing an increase in pressure, and hence a thrust loss. Treating ventilation in a delicate way is very difficult, and the common practice in propeller analysis is to use simplified formulas. Minsaas et al. (1983) developed an expression for the reduced thrust due to ventilation by assuming that the suction side of the propeller blade was fully ventilated, and that the pressure side was covered by static pressure. Using a mathematical interpretation this can be expressed as:

$$C_{L_V} = C_L(\sigma_V = 0) + \sigma_V \tag{3.50}$$

where

C_{L_V} :	Fully ventilated lift coefficient.
σ_V :	Suction side contribution to lift coefficient at fully ventilated case.
$C_L(\sigma_V = 0)$:	Pressure side contribution to lift coefficient at fully ventilated case.

By assuming that the pressure side of the blade has no camber, the pressure side lift coefficient can be expressed as:

$$C_L(\sigma_V = 0) = \frac{\pi}{2}\alpha \tag{3.51}$$

where α [rad] is the blade section angle of attack.

The suction side contribution to lift coefficient is found by neglecting effect of camber and angle of attack, and finding the pressure difference between static pressure on the pressure side and atmospheric pressure on the suction side.

$$\sigma_V = \frac{2gh}{V_\infty^2} \tag{3.52}$$

where

g :	Gravitational acceleration.
h :	Propeller submergence.
V_∞^2 :	Velocity seen by the blade section.

From this, the lift coefficient of a fully ventilated propeller blade section can be expressed as:

$$C_{L_V} = \frac{\pi}{2}\alpha + \frac{2gh}{V_\infty^2} \tag{3.53}$$

A very rough estimate of the thrust coefficient of a typically propeller can be found as (Minsaas et al., 1983):

$$K_T = 1.5\ EAR\ C_L(0.7) \tag{3.54}$$

where

EAR :	Expanded blade area ratio.
$C_L(0.7)$:	Lift coefficient at 70% radius.

From this the thrust loss factor due to propeller ventilation can be found as:

$$\beta_V = \frac{K_T}{K_{T_0}} = \frac{1.5\ EAR}{K_{T_0}} \left(\frac{\pi}{2}\alpha + \frac{2gh}{V_\infty^2} \right) \tag{3.55}$$

where α and V_∞^2 are evaluated at 70% radius.

Kozlowska et al. (2009) concluded that this formula underestimates the thrust loss when the propeller is close to the free surface, and overestimates the thrust loss for deeply submerged propellers. When the propeller is deeply submerged, the assumption of fully ventilated suction side is not satisfactory. Kozlowska and Steen (2010) proposed corrections to (3.55) based on the assumption that thrust loss also depends on how much of the propeller blade area that is covered by air. This resulted in the thrust loss equation for partially ventilated propellers. Kozlowska and Steen (2010) proposed that the lift coefficient for partially ventilated propeller can be approximated by lift coefficient for non-ventilated flat plate and fully ventilated flat plate, weighted by the ventilated and non-ventilated areas of the blade. In a mathematical interpretation this was formulated as:

$$C_{L_{PV}} = \frac{K_{T_0}}{1.5\ EAR} \left(\frac{A_{nv}}{A_0} \right) + \left(\frac{\pi}{2}\alpha + \frac{2gh}{V_\infty^2} \right) \left(\frac{A_v}{A_0} \right) \tag{3.56}$$

This resulted in a new expression for the thrust loss factor due to partial ventilation:

$$\begin{aligned} \beta_V &= \frac{1.5\ EAR}{K_{T_0}} \left[\frac{K_{T_0}}{1.5\ EAR} \left(\frac{A_{nv}}{A_0} \right) + \left(\frac{\pi}{2}\alpha + \frac{2gh}{V_\infty^2} \right) \left(\frac{A_v}{A_0} \right) \right] \\ &= \frac{1.5\ EAR}{K_{T_0}} \left(\frac{\pi}{2}\alpha + \frac{2gh}{V_\infty^2} \right) \left(\frac{A_v}{A_0} \right) + \left(\frac{A_{nv}}{A_0} \right) \end{aligned} \tag{3.57}$$

where

A_0 : Propeller disk area.

A_v : Ventilated area of propeller disk.

A_{nv} : Non-ventilated area of propeller disk.

Experiments (Koushan, 2006, 2007; Kozlowska et al., 2009; Olofsson, 1996) on varying submerged propellers have identified a small amount of ventilation regimes, showing the same type of ventilation patterns in terms of applied loads or type of air-drawing. The ventilation regimes are illustrated in Figure 3.7. The partially ventilating regime is characterised by having and unstable amount of propeller blade area covered by air. In this regime the propeller thrust typically fluctuates rapidly. The fully ventilated regime takes place if the submergence is sufficiently low and the propeller loading high. More or less the complete suction side of the blade is covered by air, which causes a significant thrust loss. The loss is however quite stable in time in the fully ventilating regime. Propellers will also typically experience an unstable ventilation regime, where rather abrupt changes of thrust occurs.

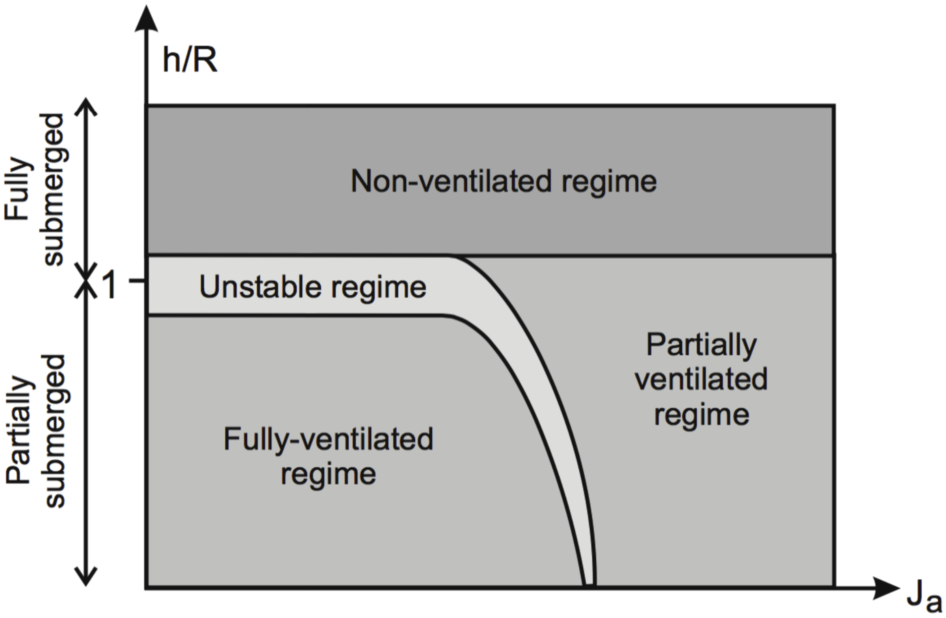


Figure 3.7: Ventilation flow regimes. Adapted from Olofsson (1996).

Determination of the amount of ventilated area of the propeller blades is a very demanding and, not to mention, difficult process. Califano (2010) used RANS simulations to investigate the mechanisms of ventilation of a marine propeller by vortex formation, and Kozlowska and Steen (2010) used high speed video during model experiments for visual observations. The visual observations led to an approximation of the ventilated blade areas for different propeller submergences, given in Table 3.1.

Table 3.1: Ventilated and non-ventilated area of propeller blade. Adopted from (Kozlowska and Steen, 2010).

h/R	$\left(\frac{A_{nv}}{A_0}\right)$ min - max	$\left(\frac{A_v}{A_0}\right)$ min - max
1	0	1
1.40	0 - 0.05	1 - 0.95
1.72	0 - 0.50	1 - 0.50
2.60	0.20 - 1	0 - 0.80
3.40	0.20 - 1	0 - 0.80

3.3.3 Lift hysteresis

When the propeller goes in and out of water there will be a hysteresis effect in the thrust generation. This causes a loss of thrust, denoted β_w. The lift hysteresis effect is connected to the time delay before the lift on the blade builds up. When a propeller blade splashes into the water, there will be a sudden increase in the lift. However, after the blade has penetrated the water surface there will take some time for the lift to reach its full value. Wagner studied a related problem for a two dimensional foil, and found that the sudden increase amounted to 50% of the fully developed lift (Faltinsen et al., 1981).

The Wagner function gives the ratio β_w between the instantaneous lift and the fully developed lift for a propeller blade immersing into the water. The original function was a rather mathematical complicated analytical solution to the problem, so a curve fit approximation to the analytical solution is mostly used. The curve fit approximation is given in (3.58) and is referred to as Minsaas' Wagner function. A plot of this function is given in Figure 3.8.

$$\beta_w = 0.5 + 0.5\sqrt{1 - \left(\frac{155 - \frac{V_\infty t}{c}}{155}\right)^{27.59}} \tag{3.58}$$

where

$\frac{V_\infty t}{c}$: number of chord lengths travelled by the propeller section

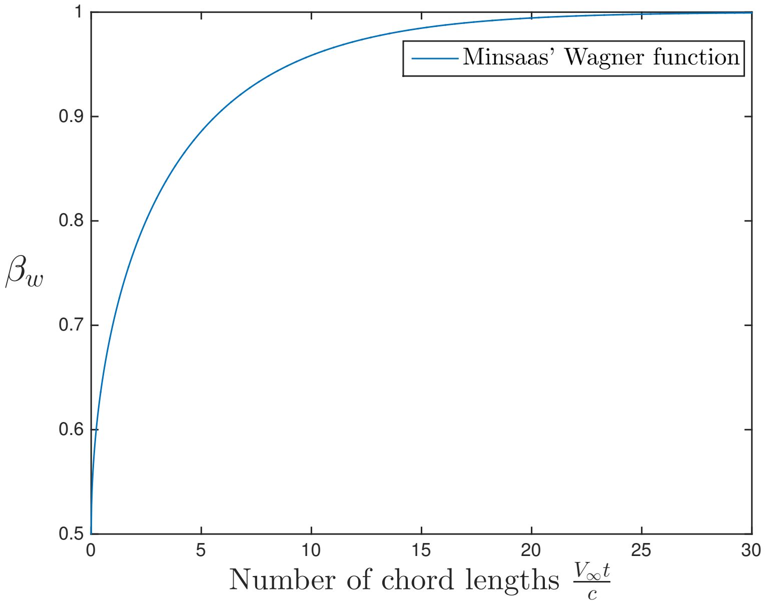


Figure 3.8: Minsaas' curve fit to the Wagner function.

3.4 Oblique inflow

Oblique inflow is an operating condition that a propeller most certainly will experience during operation. The situation occurs due to ship manoeuvring, either at high advance numbers during transit or at low advance numbers during port operations and dynamic positioning. The general aspect of oblique inflow is that the inflow field seen by the propeller changes with the azimuth angle δ. This is in fact rather general and applicable to all advance numbers. A more sophisticated aspect is that the propeller wake will leave the propeller along an axis not normal to the disc plane. The propeller wake will be skewed to one side of the propeller with the wake skew angle χ, see Figure 3.9. The wake skew angle is largely depending on the propeller loading.

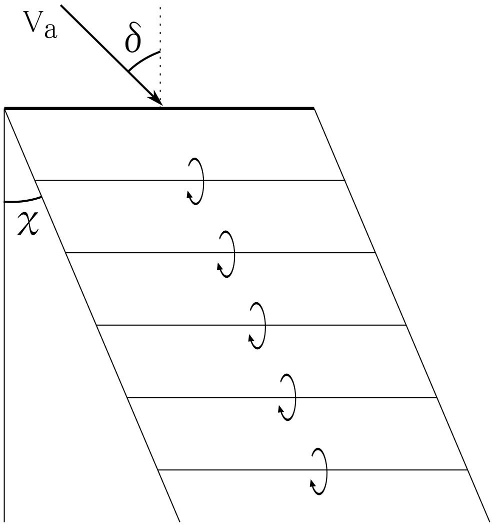


Figure 3.9: Propeller disk at inflow angle δ and skewed vortex rings at angle χ. Reproduced from (Amini, 2011).

3.4.1 Decomposition of axial and tangential inflow field

The condition of axial and tangential inflow field change when the propeller is inclined with an azimuth angle relative to the ship velocity. The axial and tangential inflow components must be decomposed into the local coordinate system rotating along with the azimuth angle δ. The axial inflow velocity V_a is decomposed into two components, one in-plane component and one component normal to the propeller disc, see Figure 3.10. The normal component is a source to an axial inflow velocity field independent of local blade position. Following the directions as illustrated Figure 3.10, the velocity normal to the propeller disc due to axial inflow velocity in case of oblique inflow can be expressed by (3.59). Similar, the in-plane component creates a tangential and radial inflow velocity. However this will be local to the blade, thus dependent on the local blade position θ. The model neglects all radial velocity components, so only the tangential component will be further discussed. Following simple geometrical considerations, the tangential velocity local to the blade due to axial inflow can be expressed as in (3.60). The decomposition of the in-plane velocity is also illustrated in Figure 3.11.

$$V_{a,a} = V_a \cos\delta \tag{3.59}$$

$$V_{t,a} = V_a \sin\delta\cos\theta \tag{3.60}$$

where

V_a : Undisturbed axial inflow component local to blade.

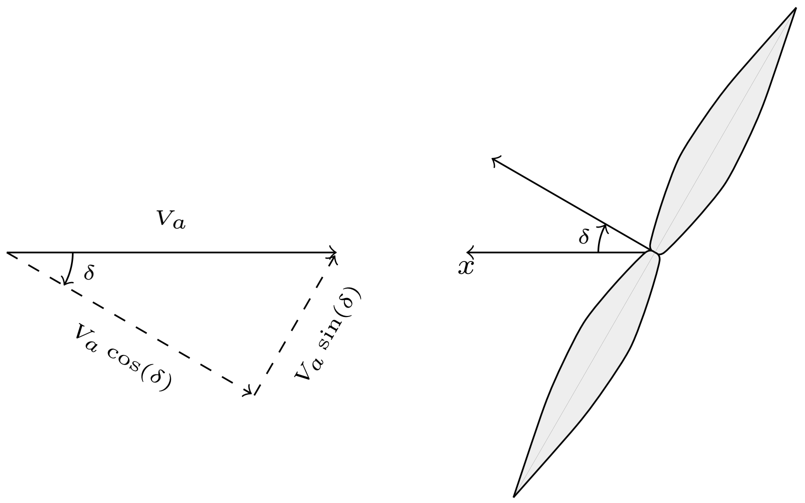


Figure 3.10: Decomposition of axial wake field due to azimuth angle δ. Top view.

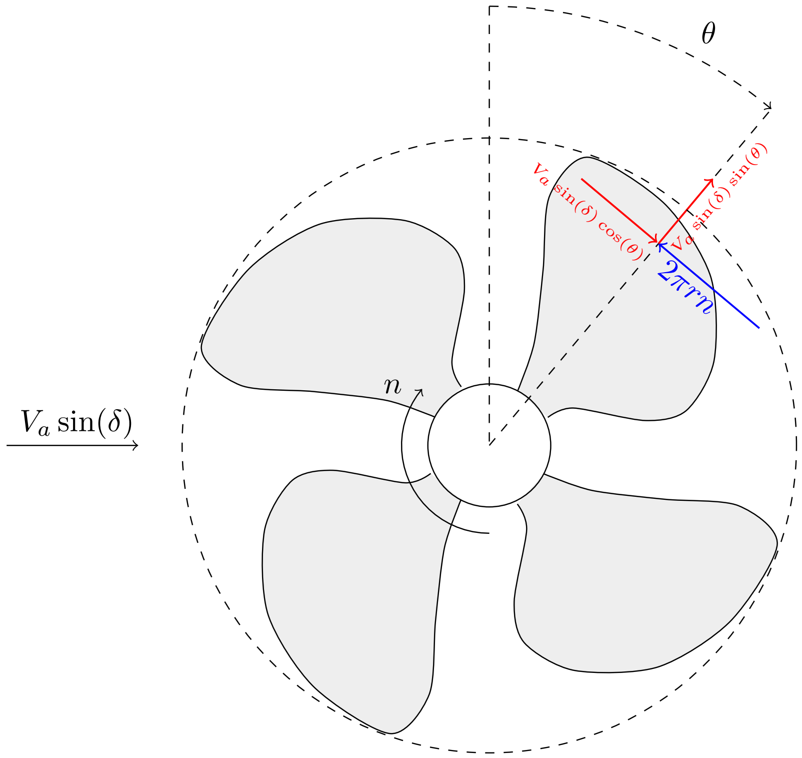

θ
$V_a \sin(\delta) \cos(\theta)$
$V_a \sin(\delta) \sin(\theta)$
$2\pi r n$
n
$V_a \sin(\delta)$

Figure 3.11: Decomposition of axial wake field into tangential and radial velocity components as function of angular propeller blade position θ. View from behind.

If the propeller operates in a tangential velocity field, the tangential velocity will contribute to both a normal component $V_{a,t}$ and a tangential component $V_{t,t}$ local to the blade, dependent on the azimuth angle δ. The contribution from the tangential velocity to the axial inflow component local to the propeller blade can be expressed by (3.61). The contribution from the tangential velocity to the tangential inflow component local to the propeller blade can be expressed by (3.62).

$$V_{a,t} = V_t \sin\delta \cos\theta \tag{3.61}$$

$$V_{t,t} = V_t \sqrt{(\cos\theta \cos\delta)^2 + \sin^2\theta} \tag{3.62}$$

where

V_t : Undisturbed tangential inflow component local to blade.

For a propeller operating in both an axial and a tangential undisturbed velocity field, the total axial and tangential inflow velocity local to the blade during oblique inflow can be found as:

$$V_{a,\delta} = V_a \cos\delta + V_t \sin\delta \cos\theta \tag{3.63}$$

$$V_{t,\delta} = V_t \sqrt{(\cos\theta \cos\delta)^2 + \sin^2\theta} + V_a \sin\delta \cos\theta \tag{3.64}$$

Note that for radial and circumferentially inhomogeneous axial and tangential velocity fields, V_a and V_t are functions of both radius r and rotational angle θ.

3.4.2 Vortex wake deflection

As illustrated in Figure 3.9 the propeller wake will be skewed to one side of the propeller if the incoming velocity is inclined relative to the propeller disk. Amini (2011) discussed the properties of the skewed vortex wake using theory of Coleman et al. (1945), which formulated an expression for the distribution of induced velocity over the propeller disk. The final expression was presented as:

$$\frac{u_i(r,\theta)}{u_0} = 1 + K \tan(\frac{\chi}{2}) \frac{r}{R} \cos(\psi) \tag{3.65}$$

where

$u_i(r,\theta)$: Local axial induced velocity at (r,θ) due to skewed propeller wake.

u_0 : Average axial induced velocity over the propeller disk.

ψ : Circumferential position of blade, zero at downstream position.

K : Form factor.

χ : Propeller wake skew angle.

$\frac{r}{R}$: Relative radial position on the propeller blade.

Different values for the K factor have been suggested. Pitt and Peters (1981) proposed $K = \frac{15\pi}{32}$.

Note that the circumferential position of the blade in (3.65) is measured from the most downstream position of the propeller disk when subject to oblique inflow. This is due to the fact that this position has the greatest amount of induced velocity. If the blade position θ is defined as shown in Figure 3.11, the angle ψ will be shifted 90° relative to θ, and the relation will be:

$$\psi = 90 - \theta \tag{3.66}$$

By relating the mean induced velocity to the incoming velocity, the wake skew angle was found to be equal to:

$$\chi = \arctan\left(\frac{V_a \sin\delta}{V_a \cos\delta + u_0}\right) \tag{3.67}$$

Amini (2011) also discussed the momentum theory for oblique inflow presented in (Glauert, 1935). Glauert studied a lifting rotor with an inclination angle to the incoming flow, and formulated that the differential thrust of an annular ring of the propeller can be expressed as:

$$T(r) = 4\rho\pi r u_0(r)\sqrt{V_a^2 + 2V_a u_0(r)\cos\delta + u_0^2(r)}\,\mathrm{d}r \tag{3.68}$$

where

$u_0(r)$: Average value of the axial induced velocity for an annular ring.

(3.68) relates the annular average induced velocity to the annular thrust, which means that the $u_0(r)$ can be found by knowing the thrust of each annular ring. That is, if the thrust on a propeller blade section is found by for example using BEMT or lifting line theory, the average induced velocity on the blade section can be found. The wake skew angle can the by found using (3.67), which finally leads to the local axial induced velocity using (3.65). Since the thrust depends on the induced velocity, and the induced velocity again depends on the thrust, and iteration procedure is necessary to obtain a final result of the annular propeller induced velocity.

4 Evaluating simplifications

A simulation model built for investigation of propeller performance can get very accurate, however the accuracy of the simulation result is very often related to the required computational effort. Depending on the purpose of the model it is necessary to decide on a balance between the desired accuracy and the tolerance for computational effort. Thus, investing some time for evaluation of possible simplifications prior to building the simulation model can be beneficial. The numerical software AKPA has been applied for this objective, and this section will present the motive, strategy and findings of the study.

4.1 AKPA

The AKPA software is a velocity based boundary element method intended for analysis of two-staged podded marine propulsors with controllable pitch open or ducted propellers (Krasilnikov et al., 2011). The program consists of two main calculation modules: The Steady Propeller Analysis (SPA) and the Quasi-Steady Propeller Analysis (QSPA). The SPA module performs the analysis of propeller in circumferentially homogeneous velocity field, while the QSPA module performs unsteady propeller simulation in a circumferentially inhomogeneous velocity field, allowing both radial and angular variations. The QSPA module originally used a quasi-steady approach, however as from AKPA version 5.4 and later on the module performs a complete unsteady simulation.

4.1.1 Analysed data

The propeller used in the AKPA analysis is the KVLCC2 propeller. This is a four-bladed fixed pitch propeller of regular design, as shown in Figure 4.1. The global propeller geometry and blade section geometry of the propeller is given in Appendix C1.2. The unsteady axial wake field used in the analysis is shown in Figure 4.2, and input data can be found in Appendix E3.

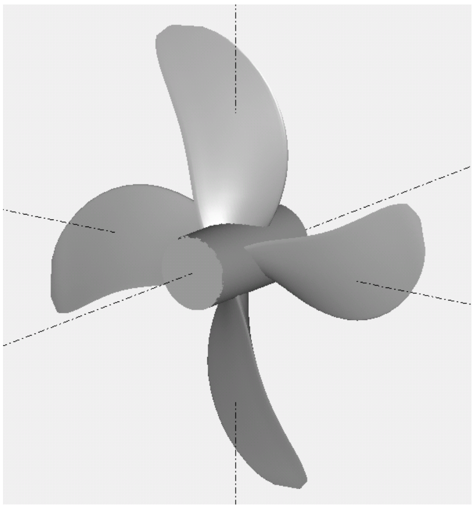


Figure 4.1: Propeller three-dimensional geometry view of KVLCC2 propeller.

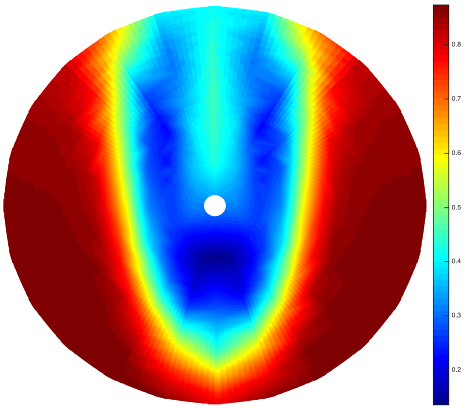


Figure 4.2: Unsteady axial wake field used in the AKPA calculations.

4.1.2 Unsteady effects

A propeller operating in a circumferentially inhomogeneous wake field will experience unsteady effects. The unsteady effects are related to vortex generation due to change of local inflow speed at different propeller blade angular positions. When the circulation around the blade changes, vortices are shed from the trailing edge causing additional induced velocities on the blade. This will affect the resulting flow characteristics around the blades, hence also the propeller forces. Performing fully unsteady propeller calculations are very time consuming, because it requires to keep track of the vortices shed due to circumferentially varying blade circulation.

Several studies have performed analyses of unsteady effects on propellers by relating the wavelength of the oscillatory downwash to either the span or the chord length of the blade. Sclavounos (1987) developed a method of matched asymptotic expansions to account for unsteady effects, which generalised the existing approaches and increased their valid frequency range. Common for the varieties of developed approaches is the fact that they are potential methods.

Using simpler approaches for propeller simulation makes it more challenging to account for the unsteady effects. The potential methods are not suited for application to lifting line or BEMT, hence this effect must be considered in other ways. The most simple way of treating the circumferentially inhomogeneous wake field is by using a quasi-steady approach. The concept behind this approach is that the instantaneous flow field local to a propeller blade is treated as if it was covering the entire propeller disk. Because the local flow field changes among the propeller blades, the calculations are applied to each one of the propeller blades. The total propeller forces are then found by averaging the contributions from the number of blades in question. An illustration is shown in Figure 4.3.

The quasi-steady approach takes into account the instantaneous flow conditions for each blade, however the unsteadiness of the problem is completely lost. Experience has however shown that if the radial distribution of axial and tangential momentum is taken into consideration, the influence from the vortices shed due to circumferentially varying inflow can be neglected (Phillips et al., 2009).

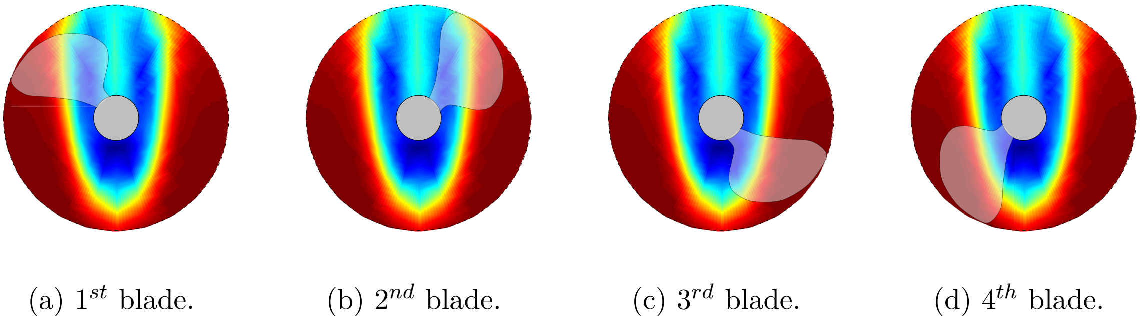
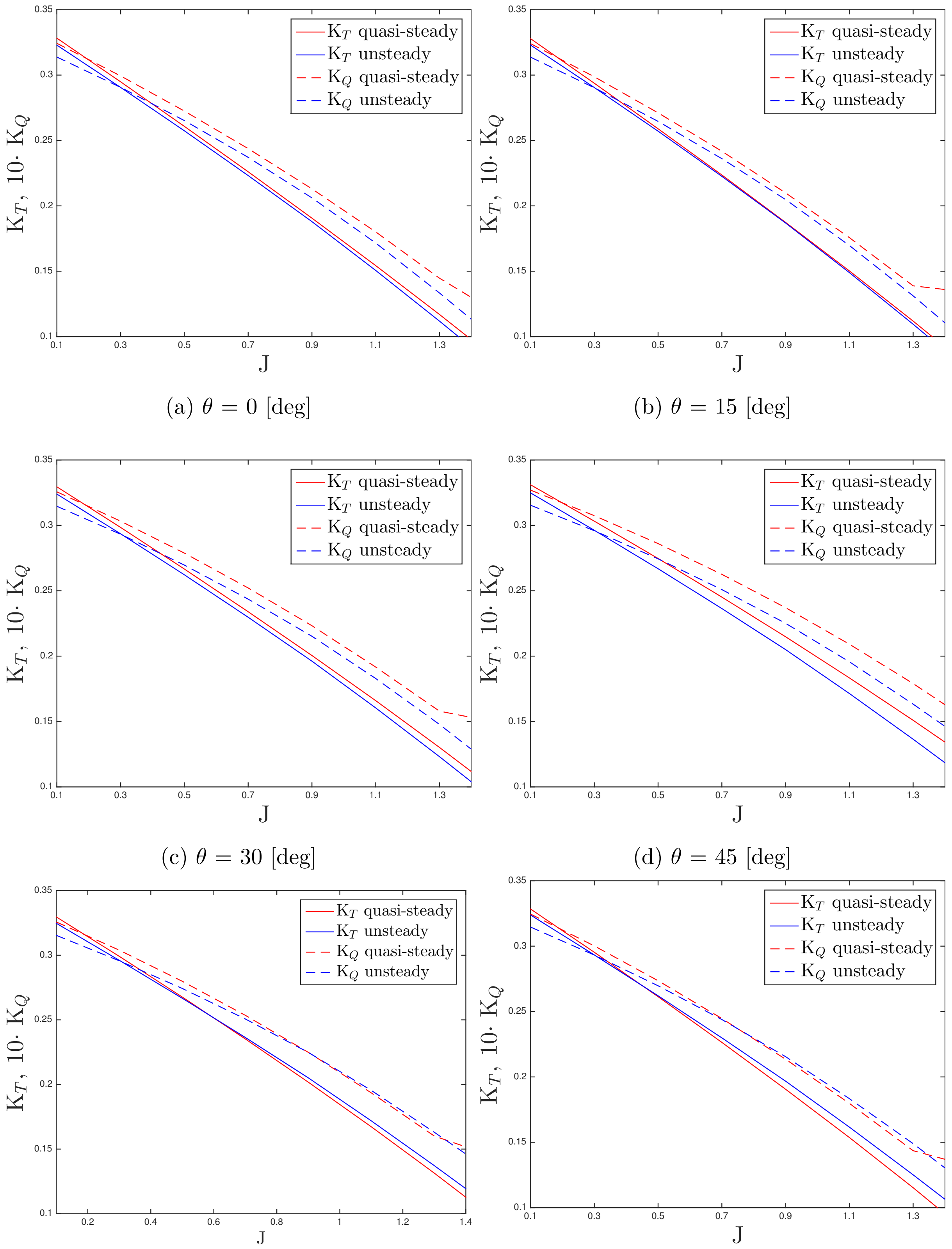

(a) 1st blade.
(b) 2nd blade.
(c) 3rd blade.
(d) 4th blade.

Figure 4.3: Quasi-steady approach for lifting line calculation in unsteady wake field.

In addition to the present experience regarding unsteady analyses, the actual contribution from the unsteady effects has been investigated using AKPA. Fully unsteady simulations for six different propeller blade positions at advance numbers ranging from J=0.1 to J=1.4 have been carried out. Further, quasi-steady simulations for the corresponding circumferentially homogeneous inflow fields and same range of advance numbers have been performed. For the four-bladed KVLCC2 propeller this corresponds to 4 x 6 = 24 simulations for each advance number. Figure 4.4 shows the result of the unsteady and quasi-steady AKPA simulations.

The simulations clearly verify the initial assumption that the unsteady effects can be neglected. Figure 4.4 shows that the difference between quasi-steady and fully unsteady analysis is low, and that this assumption holds for both low and high propeller loadings.



(a) θ = 0 [deg]

(b) θ = 15 [deg]

(c) θ = 30 [deg]

(d) θ = 45 [deg]

(e) θ = 60 [deg]

(f) θ = 75 [deg]

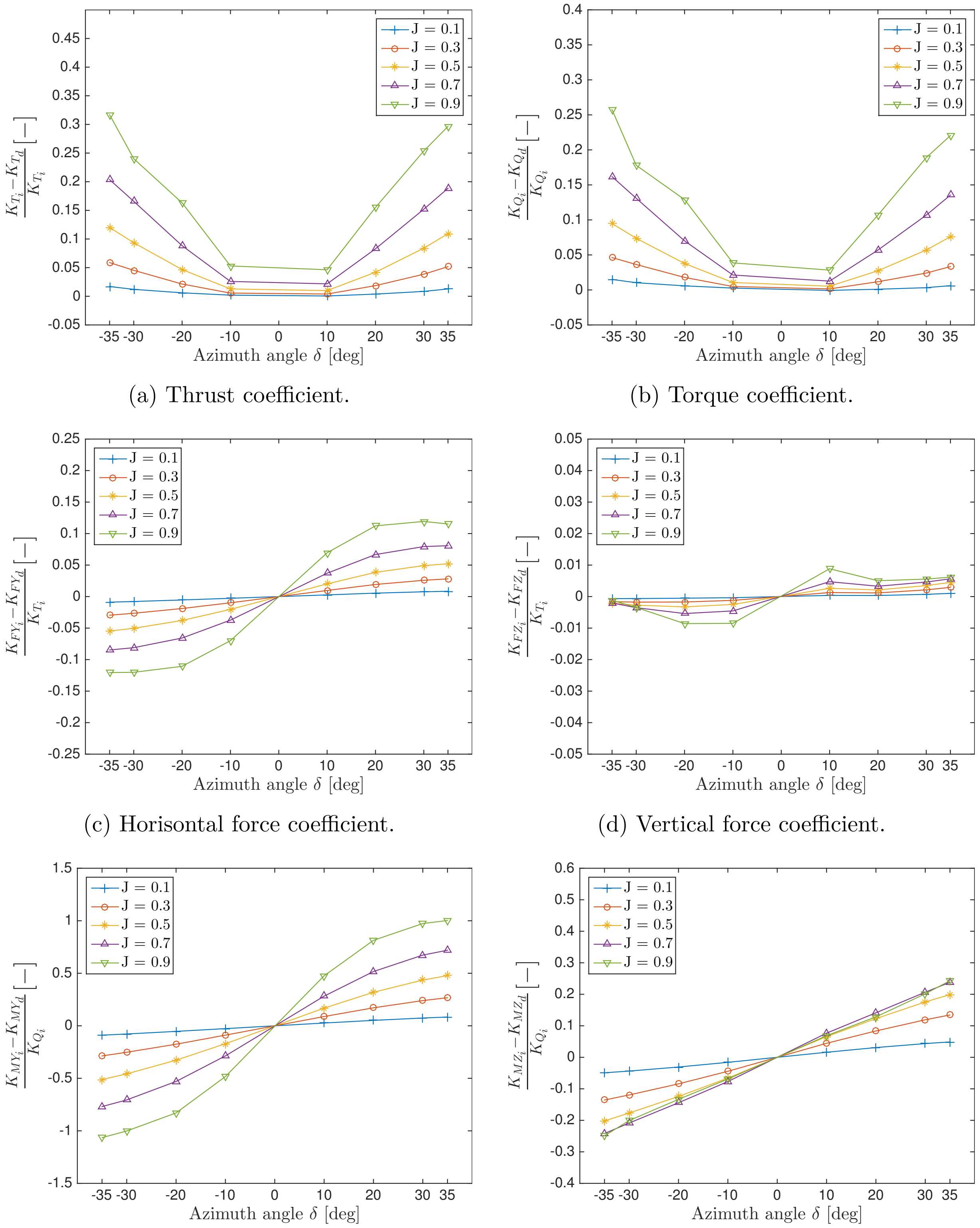
Figure 4.4: Comparison between quasi-steady and fully unsteady propeller force calculation in AKPA for four-bladed propeller at six different angular positions θ [deg].

4.1.3 Effect of skewed wake corrections

When the propeller is inclined to the flow with an angle δ, the effective inflow to the propeller disk will consist of a normal component and an in-plane component, see Figure 3.10. The in-plane component contributes to a circumferentially dependent tangential inflow velocity component, while the normal component will be uniform over the propeller disk. The normal velocity component will decrease with increased azimuth angle δ, which means that the effective advance number decreases. The circumferentially varying tangential components cause the generated propeller forces to vary with blade position, which in reality means that side forces and bending moments are generated.

In addition to the fact that the inflow conditions depend on the azimuth angle δ, the properties of the propeller wake will also change with the azimuth angle, as discussed in section 3.4.2. The effect is referred to as the *skewed wake effect*.

Taking the skewed propeller wake into consideration will introduce more complex behaviour of propeller induced velocities, hence also additional demands on computational effort. For the purpose of establishing a computational efficient propeller simulation model, it would be practical if the skewed propeller wake effect could be neglected, and within an acceptable range of error. Multiple combinations of circumferential blade position θ, advance number J and azimuth angle δ have been simulated using AKPA to test whether or not this simplification can hold, and if so, to what extent. The strategy has been to compare the fully unsteady calculation including built-in azimuth angle functionality with a fully unsteady calculation only adjusting the inflow velocity field to the azimuth angle. For simplicity the fully unsteady calculation including built-in azimuth angle functionality, hereby referred to as the *inclined calculation*, is considered to be the correct value. The fully unsteady calculation only adjusting the inflow velocity field to the azimuth angle, hereby referred to as the *decomposed calculation*, is considered a simplification. Figure 4.5 shows the relative error between the decomposed (subscript d) and inclined (subscript i) simulation for all six DoF at propeller rotation angle $\theta = 0$. The complete set of results for six different propeller rotation angles can be found in Appendix D. Note that the coordinate system used is fixed to the propeller plane, i.e. local to the propeller shaft, as illustrated in Figure 5.1.



(a) Thrust coefficient.

(b) Torque coefficient.

(c) Horisontal force coefficient.

(d) Vertical force coefficient.

(e) Horisontal moment coefficient.

(f) Vertical moment coefficient.

Figure 4.5: Comparison between decomposed wake field (subscript d) and inclined (subscript i) calculation using AKPA for the KVLCC2 propeller in oblique inflow.

What we clearly can identify from Figure 4.5 is that the effect of skewed propeller wake is of significant importance. Starting with the thrust (a) and torque (b) coefficients we see how the relative error is quite low for high propeller loadings, that is, low advance number. For increasing advance number the relative error grows significantly for both thrust and torque, and we can identify errors up to 30%. This can be explained by the fact that the incoming velocity has less importance for higher propeller loadings. The relative error also increases with the azimuth angle.

Horisontal (c) and vertical (d) side forces show the same trend as for the thrust and torque. That is, the relative error is quite low for high propeller loadings, but increases with the advance number. Note that the relative error of side force is plotted against the thrust coefficient from the inclined calculation. This enables a logical representation of how the relative error changes with propeller loading and azimuth angle. The drawback is however that the amount of error in vertical and horisontal side force is more difficult to compare.

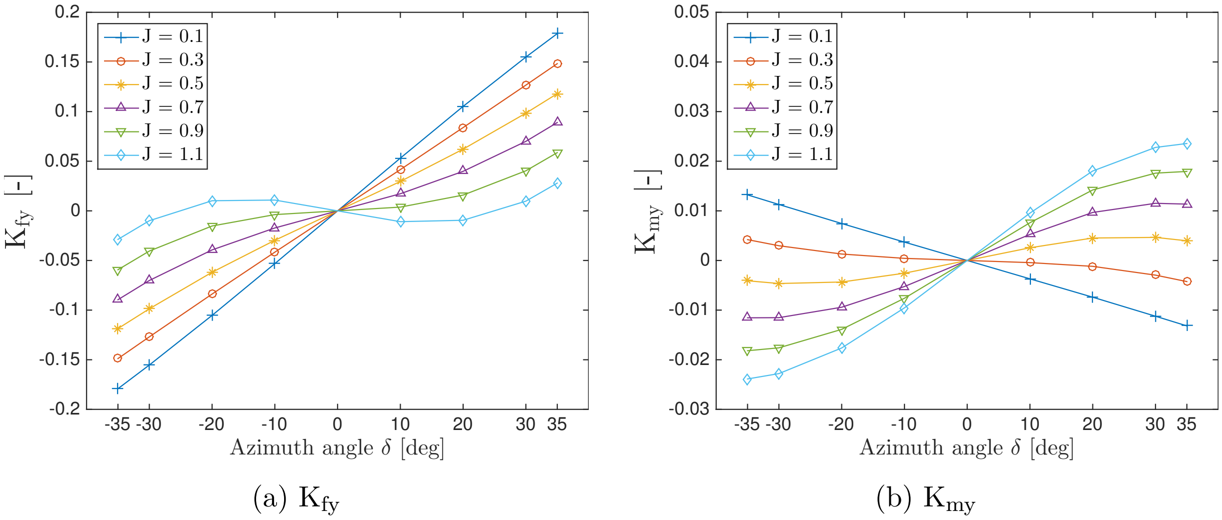
Similarly, the horisontal (e) and vertical (f) bending moments show the same trend that the relative error grows significantly with propeller loading. Note that the relative error of bending moment is plotted against the torque coefficient from the inclined calculation. This enables a logical representation of how the relative error changes with propeller loading and azimuth angle. The drawback is however that the amount of error in vertical and horisontal bending moment is more difficult to compare.

The AKPA calculations show evidently that the effect of skewed propeller wake is hard to neglect, because the accuracy of the simulation model would be significantly reduced, especially for low propeller loadings and large azimuth angles. Hence, the conclusion from this investigation is that the skewed propeller wake effect should be taken into consideration.

4.1.4 Correction of AKPA-results

According to the AKPA user manual (Krasilnikov et al., 2011) the output propeller side forces and bending moments are in a local coordinate system fixed to the propeller disk. Multiple simulations for different combinations of azimuth angles δ, inclination angles χ and advance numbers however revealed that the forces stated to be local to the propeller could not possibly be that. Figure 4.6 shows the horisontal side force and bending moment post-processed directly from AKPA results, which are stated to be local to the propeller. The first consideration is how the side force increases with propeller loading. For high propeller loading, the dependence on the azimuth angle is evident and almost linear. Yet, the physical interpretation of high propeller loading suggest only modest dependence on the azimuth angle. This is because the contribution from inflow field becomes small when the loading is high, resulting in lower dependence on azimuth angle. If we rather consider the result shown in Figure 4.6 (a) as side force fixed to a global coordinate system, the result makes more sense. In a global coordinate system the main contribution to side force is the side force component of the thrust. Because the thrust increases with propeller loading the side force in the global coordinate system increases its dependency on azimuth angle as the propeller loading increases. Exactly the same argument can be established for the horisontal bending moment, because in a global coordinate system it will mainly be a decomposed component of the propeller torque.

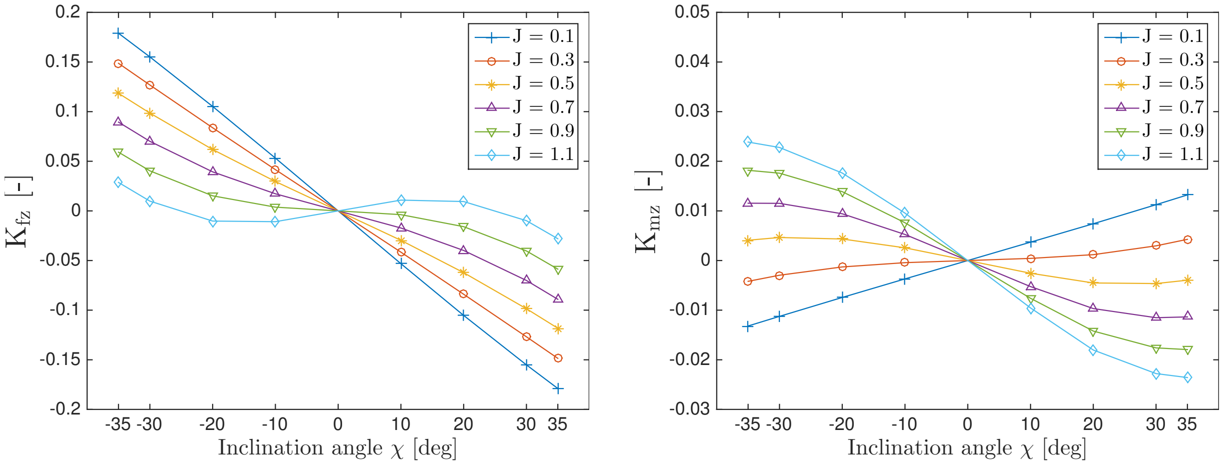
Similar findings were made related to the inclination angle χ. Figure 4.7 shows the vertical side force and bending moment post-processed directly from AKPA results, which are stated to be local to the propeller. The same strong dependence on the inclination angle for high propeller loading is found for the vertical side force and bending moment. Due to the same reasons as for the horisontal side force and bending moment at non-zero azimuth angles, one can argue that the vertical side force and bending moment is given in a global fixed coordinate system rather than local to the propeller disk.



(a) K_{fy}

(b) K_{my}

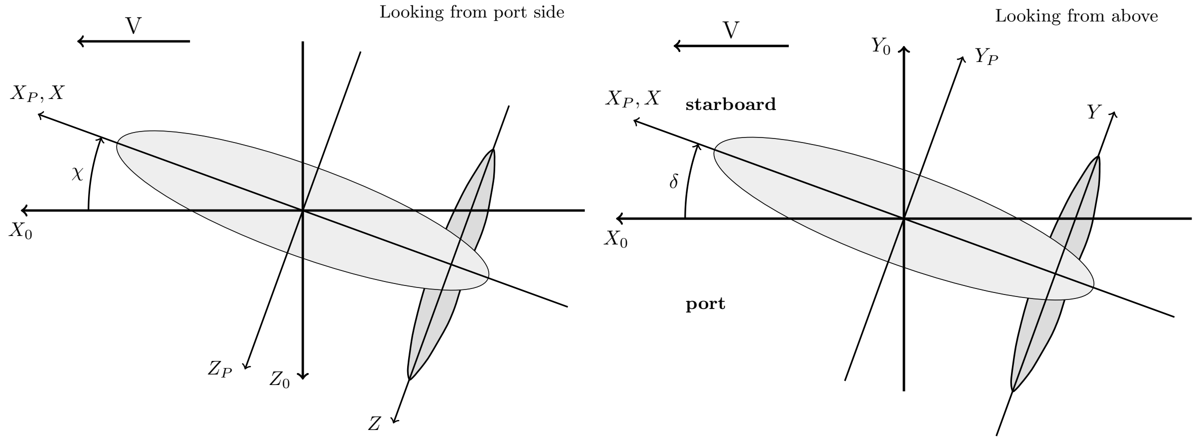
Figure 4.6: Horisontal side force and bending moment coefficients as function of azimuth angle δ. Post-processed directly from AKPA results, stated to be local to the propeller.



(a) $\mathrm{K_{fz}}$ (b) $\mathrm{K_{mz}}$

Figure 4.7: Vertical side force and bending moment coefficients as function of inclination angle χ. Post-processed directly from AKPA results, stated to be local to the propeller.

From this detailed investigation of the AKPA-calculations there is sufficient evidence that the results are given in a global coordinate system ($\mathrm{X_0}$,$\mathrm{Y_0}$,$\mathrm{Z_0}$). However, a more feasible way of presenting the 6 DoF propeller forces in a condition of oblique inflow is in the propeller plane fixed coordinate system (X,Y,Z), see Figure 4.8. This requires a coordinate system transformation of the results. The transformation matrices are given in (4.1) and (4.2).



(a) Shaft inclination angle χ (b) Azimuth angle δ

Figure 4.8: Definition of global and local coordinate systems, with corresponding inclination angle χ and azimuth angle δ.

$$T_\chi = \begin{bmatrix} \cos(\chi) & 0 & -\sin(\chi) \\ 0 & 1 & 0 \\ \sin(\chi) & 0 & \cos(\chi) \end{bmatrix} \tag{4.1}$$

$$T_\delta = \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

Transformation of forces and moments from the global coordinate system ($\mathrm{X_0}$,$\mathrm{Y_0}$,$\mathrm{Z_0}$) to the local coordinate system (X,Y,Z) then follows from matrix multiplication as shown in (4.3). Note that this transformation holds for any combination of azimuth and inclination angles.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = T_\delta \, T_\chi \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \tag{4.3}$$

5 Structure of the simulation model

The 6 DoF time domain propeller simulation model is built based on the theory, assumptions and simplifications found to be valid and appropriate for the purpose of the model. In the succeeding sections the structure and properties of the model will be presented. The intention is to give understanding and knowledge of what the model is capable of doing and how the simulation model works. The propeller simulation model has been implemented into Simulink and has been named PropSim, which it from now on will be referred to as.

5.1 Specifications

The forces calculated in the simulation model are reaction forces, i.e. forces acting from the water on the propeller. The coordinate system that is used follows a right handed system, as shown in Figure 5.1. Note that the coordinate system is persistent regardless of direction of propeller rotation. Due to the choice of coordinate system, one should have in mind that the torque becomes negative as reaction force for a clockwise rotating propeller.

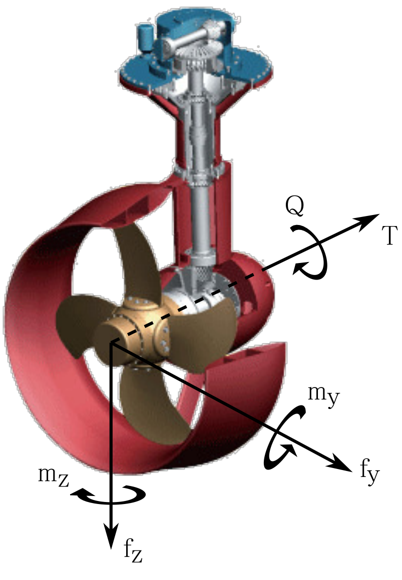


Figure 5.1: Coordinate system used in the simulation model for both clockwise and counterclockwise rotating propeller. Reproduced from (Helicetude, 2015).

5.2 Simulation input pre-processing

The simulation model retrieves information about the propeller, incoming wake field and lifting line properties from separate input files. The routine readinput.m is responsible for reading the input data, and is attached as Appendix F2. The input data is read as an initialisation of the simulation, that is, prior to the simulation starts. Hence the input data can be modified between each simulation, but not during the simulation.

5.2.1 Propeller

The propeller data is given in the file propeller.txt. That is, both global propeller information and blade section details are read from this file. The structure of the propeller file is illustrated in Figure 5.2. The first part of the file contains the global propeller data. There is a column for propeller diameter D, Expanded blade Area Ratio EAR, number of blades Z, propeller hub ratio x_h, direction of rotation, initial position of the propeller at simulation start and the density of Wagner calculation. The direction of rotation can either be RIGHT or LEFT. Note that the simulation model does not distinguish between left and right rotating propellers in the current version, the default choice is right rotating. The initial position of the propeller is the circumferential position θ [deg] when the simulation starts. The density of Wagner calculation is how dense, in degrees, the instantaneous velocity seen by the propeller blade should be calculated during Wagner effect calculation.

Beneath the global propeller data follows the blade section geometry details. The first column r_R contains the relative spanwise positions where the blade geometry is defined. The number of spanwise positions and their respective values can be user-defined, however the positions must be relative to the propeller radius, hence between zero and one.

c_D : Chord length relative to propeller diameter.

tmax_D : Maximum section thickness relative to propeller diameter.

P_D : Section pitch relative to propeller diameter.

f0_D : Maximum section camber relative to propeller diameter.

cD : Section drag coefficient.

The last part of the propeller input file contains information for the camber line using a selected NACA-profile. The first row specifies the lift coefficient at ideal angle of attack for the profile. The second row specifies the maximum camber to chord length of the camber line. The third and last row specifies the ideal angle of attack for the profile.

```
/* Propeller characteristics */
D    EAR   Z   Xh  Rotation Init.Pos Wagner
x     x    x   x   RIGHT    x        x
=============================================
/* Blade section geometry */
r_R    c_D    tmax_D   P_D     f0_D      cD
0.155   x      x        x        x        x
0.160   x      x        x        x        x
0.250   x      x        x        x        x
0.300   x      x        x        x        x
0.400   x      x        x        x        x
0.500   x      x        x        x        x
0.600   x      x        x        x        x
0.700   x      x        x        x        x
0.800   x      x        x        x        x
0.900   x      x        x        x        x
0.950   x      x        x        x        x
1.000   x      x        x        x        x
=============================================
/* Camber profile */
 x        | Lift coefficient for chosen camber profile (Cl)
 x        | Max camber relative to chord length (f0max_c) [%]
 x        | Ideal angle of attack [deg]
```

Figure 5.2: Structure of propeller input file.

5.2.2 Incoming wake field

The wake field data is given in the file wakefield.txt. The simulation model can handle both an axial and a tangential inflow wake field, which follows the structure illustrated in Figure 5.3. The first row contains the relative spanwise positions of the wake input, and the first column contains the circumferential positions, in degrees, of the wake input. The remaining rows and columns holds the actual wake input data, formatted as the incoming velocity relative to the forward speed of the ship. For axial inflow positive sign means towards the propeller, and for tangential inflow positive sign means clockwise seen from behind the propeller. The number of circumferential and spanwise positions are user-defined, however the wake input data must cover the whole area of the propeller disk.

```
/* Axial wakefield */
T\R 0.05 0.23 0.39 0.51 0.63 0.72 0.80 0.92 1.00
0    x    x    x    x    x    x    x    x    x
30   x    x    x    x    x    x    x    x    x
60   x    x    x    x    x    x    x    x    x
90   x    x    x    x    x    x    x    x    x
120  x    x    x    x    x    x    x    x    x
150  x    x    x    x    x    x    x    x    x
180  x    x    x    x    x    x    x    x    x
210  x    x    x    x    x    x    x    x    x
240  x    x    x    x    x    x    x    x    x
270  x    x    x    x    x    x    x    x    x
300  x    x    x    x    x    x    x    x    x
330  x    x    x    x    x    x    x    x    x
360  x    x    x    x    x    x    x    x    x


/* Tangential wakefield */
T\R 0.05 0.23 0.39 0.51 0.63 0.72 0.80 0.92 1.00
0    x    x    x    x    x    x    x    x    x
30   x    x    x    x    x    x    x    x    x
60   x    x    x    x    x    x    x    x    x
90   x    x    x    x    x    x    x    x    x
120  x    x    x    x    x    x    x    x    x
150  x    x    x    x    x    x    x    x    x
180  x    x    x    x    x    x    x    x    x
210  x    x    x    x    x    x    x    x    x
240  x    x    x    x    x    x    x    x    x
270  x    x    x    x    x    x    x    x    x
300  x    x    x    x    x    x    x    x    x
330  x    x    x    x    x    x    x    x    x
360  x    x    x    x    x    x    x    x    x
```

Figure 5.3: Structure of wake field input file.

5.2.3 Lifting Line

The properties of the lifting line calculations are given in the file LLproperties.txt. An illustration of the input file is given in Figure 5.4. The first input value is the number of vortex panels M the propeller blade should be divided into. The second input value is the ratio between the hub vortex radius and the hub radius. Typically this can be set to 0.25. The third input value decides the maximum number of iterations allowed in solving for the unknown circulation. The fourth input value is the swirl cancellation factor, where 1 = no cancellation. The fifth input value is the density of water [kg/m^3]. The sixth and last input value is the accuracy tolerance Δ_{tol} for determination of the unknown circulation, that is, the accepted relative change of circulation from the previous iteration, which can be expressed as:

$$\Delta_{rel} = \frac{|\text{new} - \text{old}|}{\text{new}} \tag{5.1}$$

The iteration aborts when $\Delta_{rel} \leq \Delta_{tol}$, or when the maximum number of iterations is exceeded.

```
/* Lifting Line properties */
 x         | Number of blade vortex panels
 x         | Hub vortex radius/Hub radius
 x         | Maximum number of iterations in wake alignment
 x         | Swirl cancellation factor: 1=no cancellation
 x         | Density of water [kg/m^3]
 x         | Circulation accuracy tolerance
```

Figure 5.4: Structure of input file for lifting line properties.

5.2.4 Important notes on input files

It is crucial that the structure of the input files are maintained during modifications. The routine that reads the input data needs to understand the structure of the files in order to allocate the data correctly. Hence it is recommended that the original files attached to the model are stored for later reuse. The following features are applicable and common to all the input files:

- All input files must be stored in the input folder.
- Numbers are separated by one or several spaces, **not** tabular spacing.
- Headerlines can not be removed, and must start with /* and end with */.
- Numbers and their respective descriptions are separated by |.
- The sequence of the input data must be maintained.
- The numbers can have an unspecified number of decimal places.
- The decimal point must be period, not comma.

5.2.5 Pre-processing

The routine readinput.m is responsible for assigning the information in the input files to appropriate parameters. In addition the span of the blade is divided into M vortex panels by the routine cosspace.m, using real cosine spaced control and vortex points as defined in (3.21) and (3.22). The final step in pre-processing the input data is to interpolate the blade geometry to the control points and the axial and tangential wake field to both the vortex and control points.

5.3 Main body

The main body of PropSiM starts with the Simulink module shown in Figure 5.5. The user-defined input to the block is the propeller rotational speed [RPM], ship speed [m/s], pitch control angle [deg], propeller submergence [m] and the azimuth angle [deg]. This input can have individual arbitrary time dependency, however it is recommended that they are kept continuous. The output from the block is the 6 DoF propeller forces and moments in [N] and [Nm], respectively. PropSiM gives output for each time step. The subsequent sections will go through the various calculation steps of the model. The steps can be related to the PropSiM flowchart presented in 5.7.

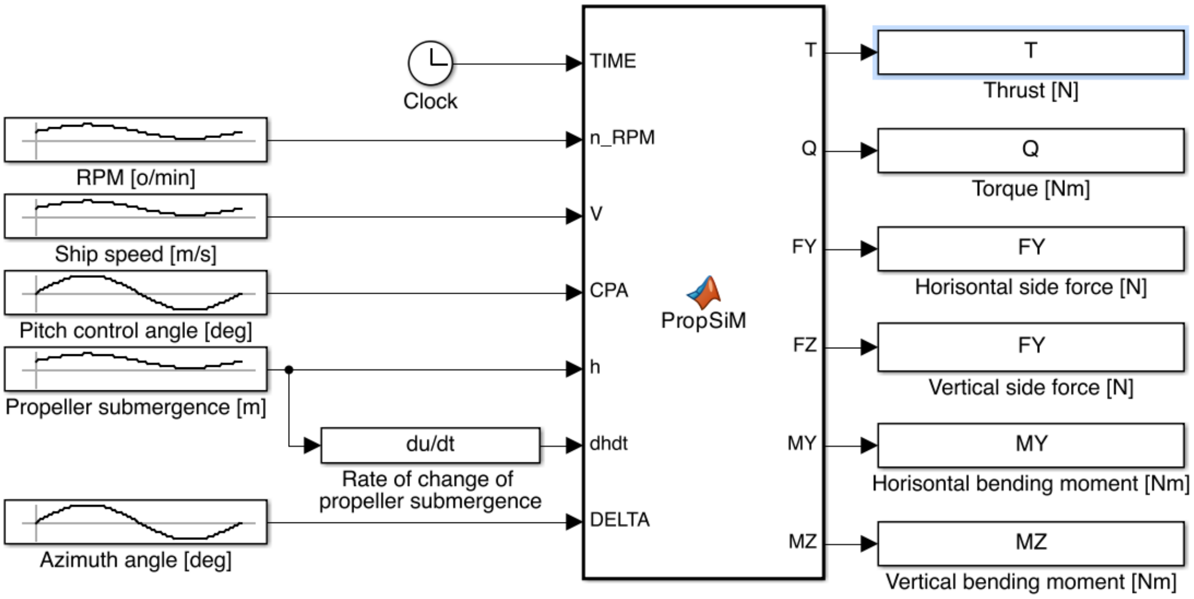

Clock
TIME
RPM [o/min]
n_RPM
Ship speed [m/s]
V
Pitch control angle [deg]
CPA
Propeller submergence [m]
h
du/dt
Rate of change of propeller submergence
dhdt
Azimuth angle [deg]
DELTA
PropSiM
T
Thrust [N]
Q
Torque [Nm]
FY
Horisontal side force [N]
FZ
FY
Vertical side force [N]
MY
Horisontal bending moment [Nm]
MZ
Vertical bending moment [Nm]

Figure 5.5: Representation of PropSiM as it is shown for the user.

5.3.1 Interpolation of input data

For each time step the circumferential position of the blades must be determined. The angle θ [deg] for blade number k is found from the expression:

$$\theta(k) = 360\left(n\,t + \frac{\theta_0}{360} + \frac{k-1}{Z} - \left\lfloor n\,t + \frac{\theta_0}{360} + \frac{k-1}{Z} \right\rfloor\right) \tag{5.2}$$

where

n : Propeller rotation speed [Hz].

t : Real time [s].

k : Propeller blade number.

θ_0 : Initial circumferential position of propeller [deg].

Z : Number of propeller blades.

$\lfloor x \rfloor$: Rounding x down to nearest integer.

After the blade positions are determined the instantaneous wake field local to each propeller blade is found by interpolating the input wake field to the circumferential positions. The routine uses a spline interpolation method. If the azimuth angle δ is non-zero, the axial and tangential wake fields are corrected for the azimuth angle using (3.63) and (3.64).

5.3.2 Finding blade circulation

When the inflow velocities local to each blade are determined, the undisturbed hydrodynamic angle β of each blade is found using (3.12). PropSiM then enters the quasi-steady analysis of the blade circulation by treating each blade subsequently in a loop. A first estimate of the hydrodynamic angle β_i is found as a multiplier of the undisturbed hydrodynamic angle β equal to 1.1, simply due to the fact that $\beta_i > \beta$ for a moderately loaded propeller. The horseshoe influence functions are then found using the approach of Wrench's closed form approximations discussed in section 3.2.1. The closed form approximations are given in Appendix A. An important aspect of finding the influence functions is that cosine spacing of the vortex and control points leads to variable distance between vortex and control points along the span. Hence the influence function from two adjacent horseshoe vortices must be averaged relative to their individual distance to the control point. The contribution to the horseshoe influence functions due to the hub image is found using the approach described in section 3.2.1.

When the horseshoe influence functions are found the distribution of circulation is determined using the equation system in (3.30) by inverse matrix multiplication. An updated value of the hydrodynamic angle β_i is then found using (3.44), which then again is input to finding new horseshoe influence functions. This procedure runs in a loop until either: (i) the maximum number of iterations is reached or (ii) the circulation distribution has converged sufficiently. Both (i) and (ii) depends on the input specified in LLproperties.txt, which were discussed in section 5.2.3. The propeller induced velocities local to control point y_c are determined from the circulation by using (3.24) and (3.25).

5.3.3 6 DoF propeller forces

When a satisfactory accuracy of the circulation distribution has been obtained, the 6 DoF non-dimensional force coefficients can be found from the discrete sum over the vortex panels given in (3.34) - (3.39). Note that C_T and C_Q are thrust and torque coefficients from all the blades, while C_{fy}, C_{fz}, C_{my} and C_{mz} are for the actual blade subject to the calculation. The thrust is corrected for the hub vortex, using (3.41).

5.4 Effect of propeller submergence

5.4.1 Loss of propeller disc area

In section 3.3.1 simple geometrical considerations were presented with the purpose of treating the thrust loss of a partly submerged propeller. However, using these geometrical properties was found to be insufficient for the purpose of the propeller simulation model. PropSiM rather treats the loss of thrust and torque local to each blade. Based on the propeller submergence the wetted length of the span S_{wet} is found using geometric identities.

$$S_{wet} = \begin{cases} \left[x_h, \dfrac{s_r}{\cos\theta}\right] & \text{for } s_r \geq 0 \\ \left[\dfrac{s_r}{\cos\theta}, 1\right] & \text{for } s_r < 0 \end{cases} \tag{5.3}$$

where

S_{wet} : Interval of wetted span of propeller blade.

x_h : Relative size of propeller hub.

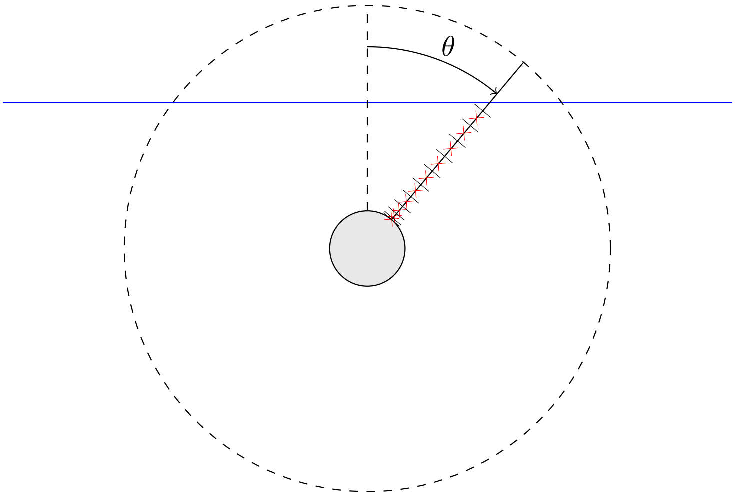


Figure 5.6: Wetted span of partly submerged propeller blade.

From this the instantaneous submerged vortex points are found solving for $y_v \in S_{wet}$. The change of submerged vortex points actually leads to a reduction in the circulation equation system (3.30). However, when the number of submerged vortex points is less than four the blade is considered dry. This is due to the fact that the numerical solution to the equation system gets unstable for less than four points. The simplification is however considered to have very low influence on the result, since the vortex and control points are very dense towards the propeller hub and tip using the real cosine spacing method.

When $s_r < 0$ and parts of the propeller blade is out of water the hub image effect is turned off. This is because it is assumed that there will be a pressure leak on the part of the blade operating close to the free surface. Hence, it is more correct to let the circulation approach zero towards the free surface rather than having a finite value of circulation. The same argument holds for $s_r > 0$: there will be a pressure leak on the parts operating close to the surface. However, the blade tip is in any case modelled with a free tip, so no modification is necessary.

5.4.2 Ventilation

The effect of ventilation is treated according to the approach outlined in section 3.3.2 by use of (3.57). The angle of attack α and velocity V_∞ are evaluated at the control point with y_c closest to 0.70. Most of the variables in (3.57) are already known from the calculation of circulation distribution. However, the amount of ventilated propeller disk area is undetermined. As there is no easy and still accurate way of predicting the ventilated propeller disk area, a simplified approach using data from experiments is implemented. The experimental results from Kozlowska and Steen (2010) presented in Table 3.1 has been used to generate polynomial approximations to the amount of ventilated propeller disk area. The polynomials take two different characteristics into consideration, a propeller increasing its submergence and a propeller decreasing its submergence. The polynomials are illustrated in Figure 5.7 and expressed mathematically in (5.4). Experiments have shown that for a propeller approaching the free surface, ventilation does not occur until $s_r \approx 1.7$. From the moment ventilated area starts to form, the change in ventilated area is rather abrupt. This is reproduced in the polynomial for decreasing submergence. For a propeller decreasing its submergence the behaviour is different. When the propeller enters the water completely the amount of ventilation will be considerable. For increasing submergence the propeller will still have access to air, so the amount of ventilation will be quite large. As the polynomial shows, some ventilated area will remain until $s_r \approx 3.4$. The equation for the ventilated area is directly implemented into PropSim.

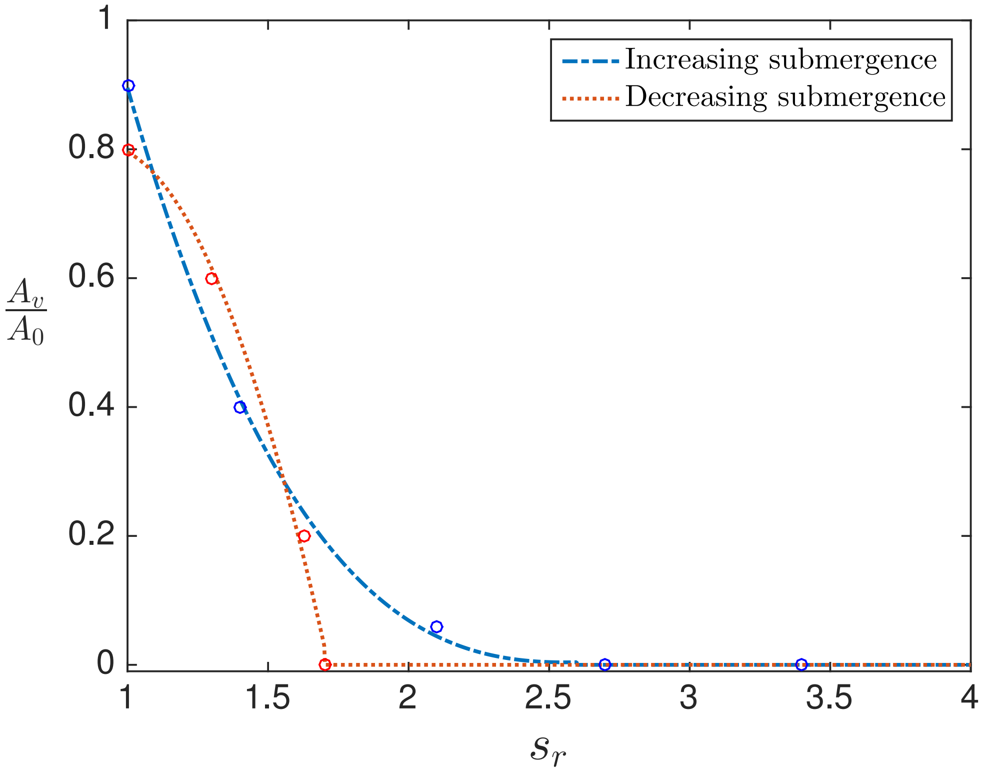


Figure 5.7: Polynomial approximations to the ventilated area during ventilation.

$$\frac{A_v}{A_0} \approx \begin{cases} -1.2276\, {s_r}^2 + 2.2201\, s_r - 0.1969 & \text{if } \dfrac{dh}{dt} < 0,\ s_r \in [1, 1.7] \\[2ex] -0.1593\, {s_r}^3 + 1.3420\, {s_r}^2 - 3.7357\, s_r + 3.4483 & \text{if } \dfrac{dh}{dt} > 0,\ s_r \in [1, 3.4] \end{cases} \tag{5.4}$$

PropSim calculates the thrust loss factor due to ventilation (β_V) for $1 \le s_r < 3.4$. For $s_r < 1$ β_V is kept as it was at the time the propeller disk penetrated the free surface. New calculation of β_V initiates when the propeller is fully submerged again.

5.4.3 Lift hysteresis

The third contribution to the thrust loss is related to the Wagner effect. PropSiM calculates the instantaneous reduction in lift force due to Wagner effect directly using Minsaas' expression given in (3.58). Since the lift force on a blade section is proportional to the bound circulation, expressed by the Kutta-Joukowski relation in (3.42), the loss factor β_w can be directly applied to the bound circulation rather than the lift coefficient.

The instantaneous number of chord lengths travelled by a blade section since water entry is necessary as input to the Wagner function. PropSiM evaluates this in the routine wagnereffect.m, which is attached in Appendix F6. For circumferential inhomogeneous inflow the velocity seen by the blade section changes with circumferential blade position. This means that integration is required in order to find the total number of chord lengths travelled from water entry to the position of the blade. In the Wagner routine the angular distance between water entry of a blade section and the circumferential blade position is divided into a certain number of points, based on user-specified input. The axial and tangential inflow wake fields are then interpolated at these points using a spline interpolation method. For each circumferential point between water entry and the instantaneous blade position the velocity seen by the propeller blade section is found using the equation system given in (3.30), i.e. by the routine circulation.m. The total number of chord lengths travelled by the blade section is then found by averaging V_∞ over the total number of circumferential points, multiplying with the time since water entry and dividing by the chord length of the blade section. The time since a blade section entered the water is found using the following relationship:

$$t_e = \frac{\theta - \theta_e}{360\, n} \tag{5.5}$$

where

t_e : Elapsed time since blade section entered the water [s].

n : Shaft rotational speed [Hz].

θ_e : Angle when blade section entered the water [deg].

θ : Propeller blade position [deg].

For saving of computational effort, the Wagner effect is only calculated for the blade sections that vary between wet and dry during one revolution. Yet since V_∞, and hence the circulation distribution, must be determined at several positions for each propeller blade, the Wagner effect influences the computational effort significantly.

5.4.4 Additional inflow velocity component

As a result of heave and pitch motion of the ship, the propeller can experience large vertical motions during operation. The vertical motion is denoted h, and is considered positive downwards. Vertical motion of the propeller causes an additional inflow velocity component $\mathrm{V_h} = \frac{d\mathrm{h}}{dt}$ to the propeller, which is the time derivative of the propeller submergence. For simplicity the model assumes that the vertical velocity is uniform over the propeller disc, and that it only contributes to a tangential velocity component $\mathrm{V_{h,t}}$ local to the propeller blade, i.e. the radial component is neglected. The tangential component depends on the rotational angle of the blade as shown in Figure 5.8 and is expressed in (5.6). PropSim treats this additional inflow velocity component by adding it to the tangential inflow wake field. Note the sign convention that is used. $\frac{d\mathrm{h}}{dt}$ is positive when the propeller increases its submergence.

$$V_{\mathrm{h,t}} = V_{\mathrm{h}} \sin(\theta) = \frac{d\mathrm{h}}{dt} \sin(\theta) \tag{5.6}$$

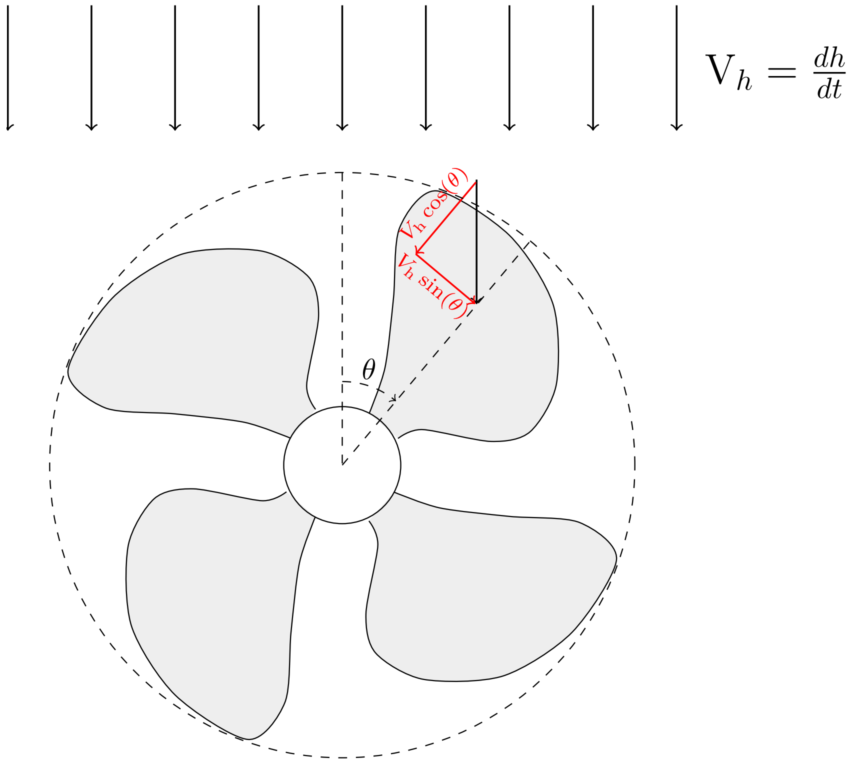

$\mathrm{V}_h = \frac{dh}{dt}$
$\mathrm{V_h}\cos(\theta)$
$\mathrm{V_h}\sin(\theta)$
θ

Figure 5.8: Decomposition of rate of change of propeller submergence $\mathrm{V_h}$=$\frac{dh}{dt}$ into tangential and radial velocity component local to propeller blade.

5.5 Effect of oblique inflow

If the azimuth angle is different from zero the simulation model activates calculation of side forces and bending moments due to oblique inflow. The axial and tangential inflow wake fields are decomposed into the new propeller plane using (3.63) and (3.64). The effect of the skewed propeller wake is then considered by forcing the axial induced velocity to follow the distribution given in (3.65). This is implemented into the simulation model by forcing the distribution of hydrodynamic angle β_i along the span to follow the distribution of axial induced velocity in (3.65). The annular mean axial induced velocity is found by equating the mean differential thrust expressed in (3.68) with the actual differential thrust found using the lifting line approach. The thrust is evaluated for each vortex panel, so the summation of the thrust coefficient in (3.34) is removed. The differential thrust of a vortex panel is found as:

$$T\,\mathrm{d}r_m = \rho\Big[V_{\infty,m}\Gamma_m\cos(\beta_{i,m}) - \frac{1}{2}(V_{\infty,m})^2 c_m C_{Dv,m}\sin(\beta_{i,m})\Big]\,\mathrm{d}r_m \tag{5.7}$$

Solving for the mean annular induced velocity u_0 requires an iteration procedure. In the simulation model the iteration continues until an error less than 5% is achieved. When u_0 is found the propeller wake skew angle χ can be found using the relationship in (3.67). With the wake skew angle available a new distribution of circulation, and hence a new distribution of hydrodynamic angle β_i can be found. This loop continues until either the required convergence of circulation is fulfilled or until the maximum number of iterations is exceeded.

The effect of oblique inflow is implemented in the routine circulation.m, simply because it is the distribution of circulation that is altered due to the skewed propeller wake. When the propeller is only partly submerged the calculation of oblique inflow is turned off. This is mainly because a situation of partly submergence introduces so many additional effects that the contribution from oblique inflow will be of low importance. Another aspect is simply that treating the skewed propeller wake when parts of the propeller is out of water will require an undesirable amount of computational effort.

5.6 Iterative solution of induced velocities

5.6.1 Improved iteration procedure

The solution of the equation system expressed in (3.30) is found by iteration. In the first outline of the lifting line routine the iteration procedure simply followed an algorithm of updating the hydrodynamic inflow angle β_i to the induced velocities. The development of the circulation at 70% propeller radius during the iteration process is shown as a blue line in Figure 5.9. However, an improved iteration method was implemented into the model to ensure faster convergence. This is shown as an orange line in Figure 5.9. The improvement was created based on observations that the circulation is alternately overestimated and underestimated. By memorising the previous steps in the iteration, a new guess on the induced velocities can be found from the two preceding predictions. The simulation model utilise this by finding the third estimate on the circulation distribution as the mean value of the first and second estimate. As we clearly can observe from Figure 5.9 this forces a much faster convergence of the circulation distribution.

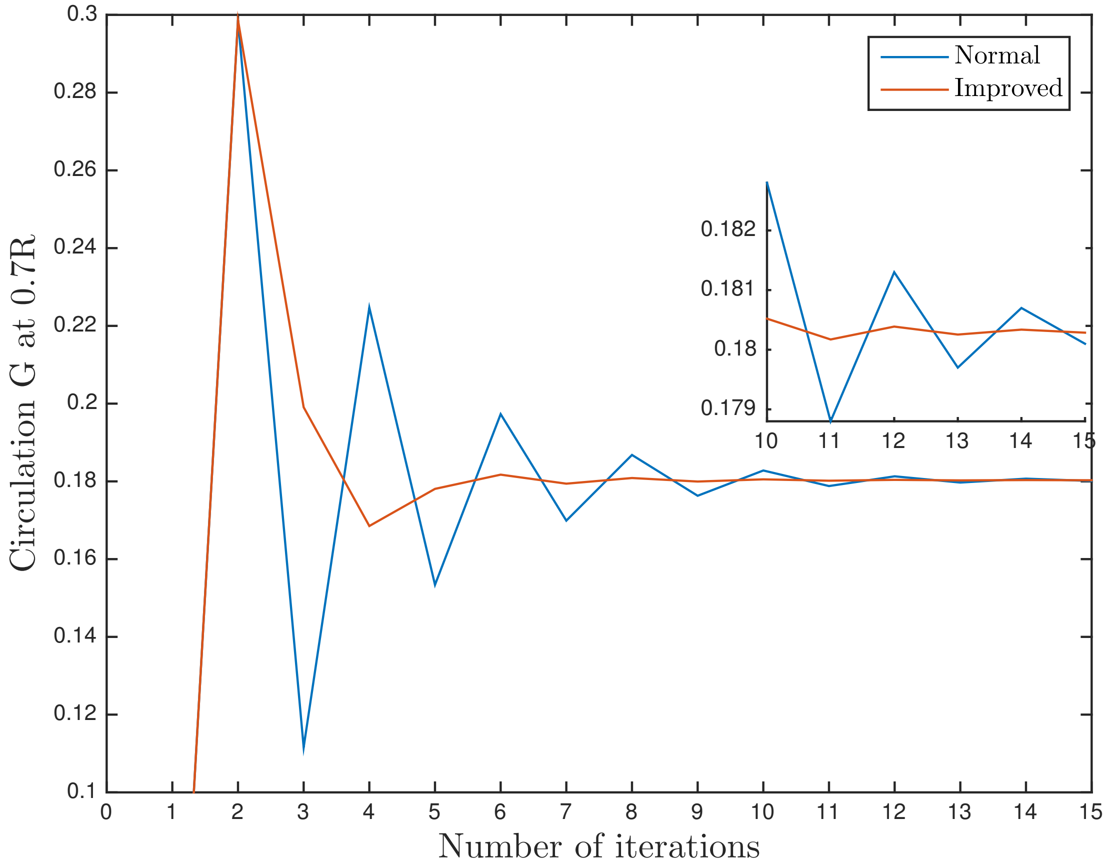


Figure 5.9: Convergence of circulation at 70% propeller radius for normal and improved iteration procedure.

5.6.2 Numerical instability - reduction factor

Some numerical instability in the iteration procedure was observed for high propeller loading. This occurs because the first estimate of the induced velocities gets very large, causing large negative angles of attack. In the second iteration these large negative angles of attack causes the calculation of induced velocities using Wrench's approximations to be unstable. Wrench's approximations includes logarithmic expressions, which are sensitive when the input approaches zero. If the input becomes negative the calculation of induced velocities crashes.

To avoid this problem for high propeller loadings, a reduction factor is included in the iteration procedure. This means that the first estimate of induced velocities is tuned down to limit the extent of overestimation of induced velocities in the first cycle of iteration. The reduction factor is set to 0.8.

5.7 Flowchart

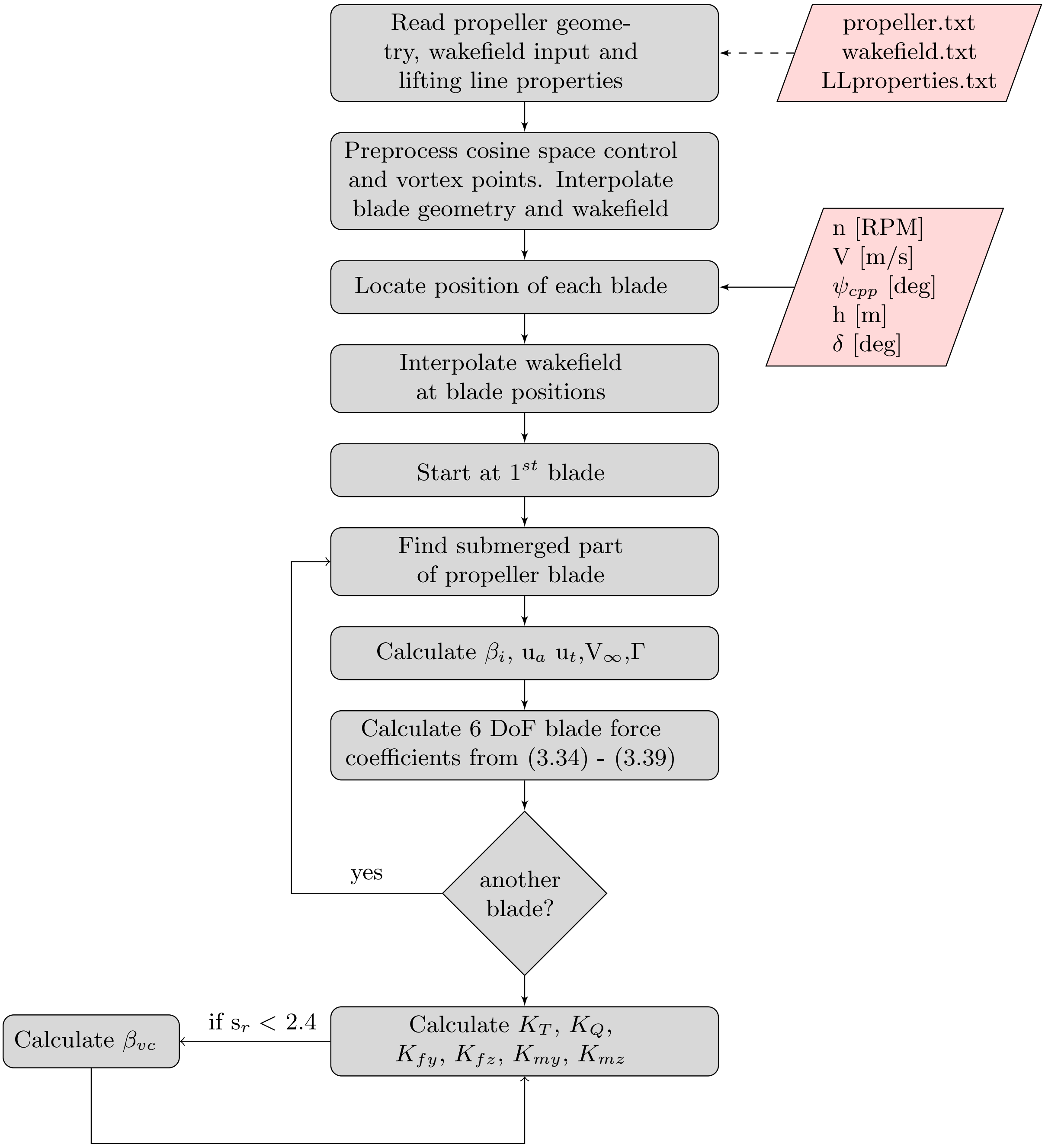

Read propeller geometry, wakefield input and lifting line properties
propeller.txt
wakefield.txt
LLproperties.txt
Preprocess cosine space control and vortex points. Interpolate blade geometry and wakefield
Locate position of each blade
n [RPM]
V [m/s]
ψ_{cpp} [deg]
h [m]
δ [deg]
Interpolate wakefield at blade positions
Start at 1^{st} blade
Find submerged part of propeller blade
Calculate β_i, u_a u_t,V_∞,Γ
Calculate 6 DoF blade force coefficients from (3.34) - (3.39)
another blade?
yes
Calculate K_T, K_Q, K_{fy}, K_{fz}, K_{my}, K_{mz}
if $s_r < 2.4$
Calculate β_{vc}

Figure 5.10: Flowchart PropSim

5.8 Recommendations

The number of vortex panels M used in the simulation will affect both computational time and accuracy. For a deeply submerged propeller, not experiencing any loss of propeller disk area, it is sufficient to use $M = 8$. If the simulation is aimed at studying the effect of partly submergence it is recommended to use $M = 16$. Depending on the propeller parameters, $M = 12$ can also be sufficient, however if the calculation gets unstable the number of panels must be increased. Note that a propeller blade is considered dry if less than four vortex points are submerged, hence the thrust loss will be unphysically abrupt when the propeller hub is close to the free surface and the number of vortex panels is low. The value of M is set in the input file LLproperties.txt.

The density of the Wagner calculation will also have a major contribution to computational time. If this number is decreased the Wagner effect is calculated more dense, i.e. there will be more points where to evaluate the induced velocities for each time step. A sufficient choice of the Wagner density is 15 [deg]. Increasing this number decreases the computational time. The value of the Wagner density is set in the input file propeller.txt.

The simulation model can not handle velocities equal to zero. The propeller loading will be very high, and the numerical calculation will be unstable. Hence it is recommended to keep the advance number $J > 0.05$ to ensure that unstability is avoided. How low advance number the simulation model can handle is depending on other parameters, such as the heave velocity of the propeller.

6 Results and validation

PropSiM has been tested and validated using both commercial software and model experiments. It is important to ensure that the model delivers expected results, that is, results that both agrees with physical behaviour and with more advanced numerical tools. More important is it also to ensure that the simulation model is able to produce results in agreement with model experiments. This section will cover all the testing of the simulation model, and compare results from simulations with both other software and model experiments.

6.1 Main body

The main body of the simulation model, that is the basic lifting line code with all additional effects excluded, has been validated in an open water condition. The validation considers both model experiments and simulations using the software akpa.

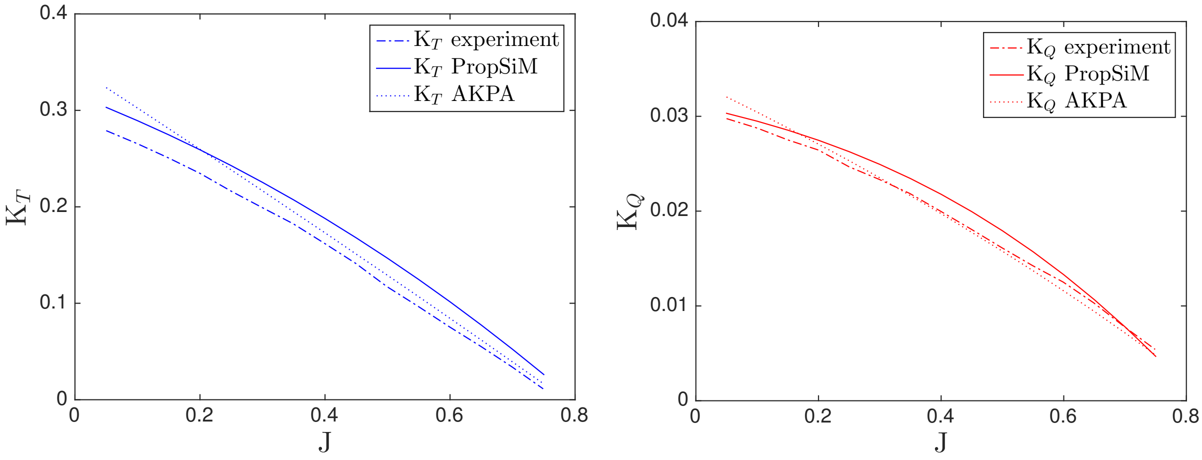
6.1.1 Open water thrust and torque coefficients

Thrust and torque coefficients as function of advance number have been evaluated for both the KVLCC2 propeller and the P1374 propeller in an open water condition. The geometry of the propellers is attached in Appendix C1.

The simulation of the KVLCC2 propeller is shown in 6.1 along with results from model experiments and akpa calculations. We see that the shape of the curves for both thrust and torque obtained with PropSiM is nearly identical to the experiment, however it seems that especially the thrust coefficient is somewhat overestimated. The akpa calculations seem to produce results more in agreement with the experiment, however the shape of the curves are not predicted that good.

A similar simulation of the P1374 propeller is shown in 6.2. akpa calculations for this propeller has not been achieved, so the validation is only performed against model experiments presented in (Amini, 2011). For this propeller we can identify that the result from simulation agrees very well with the experiments, especially in magnitude of the thrust coefficient. The shape of the curves are however not so good as for the KVLCC2 propeller.

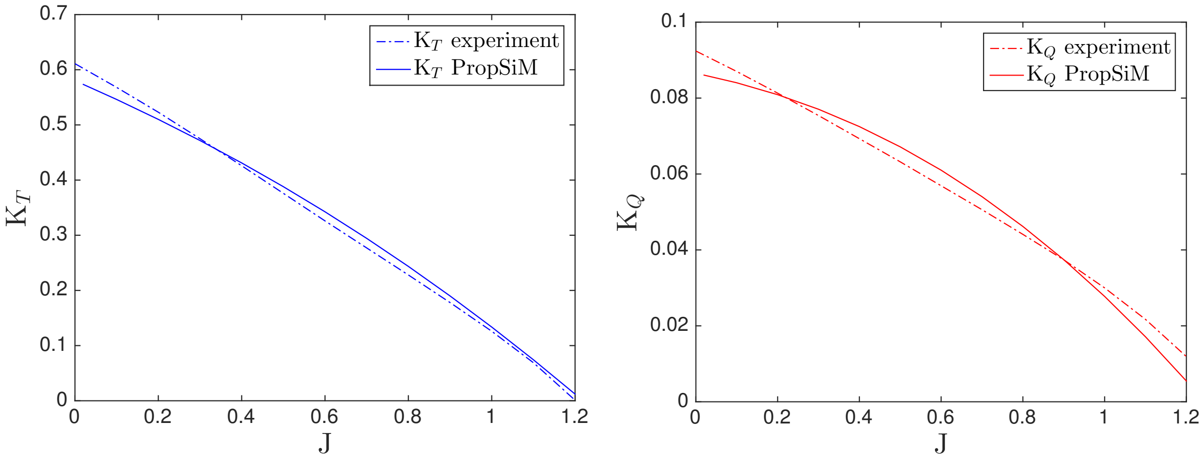
The open water comparison of the simulation model against AKPA calculations and model experiments gives good reason to argue that the main body of the simulation model works sufficiently, and that the lifting line approach is implemented correctly.



(a) Thrust coefficient K$_T$ from experiment, PropSiM and AKPA calculations.

(b) Torque coefficient K$_Q$ from experiment, PropSiM and AKPA calculations.

Figure 6.1: Comparison of open water thrust and torque coefficients from experimental results, PropSiM and AKPA calculations for the KVLCC2 propeller.



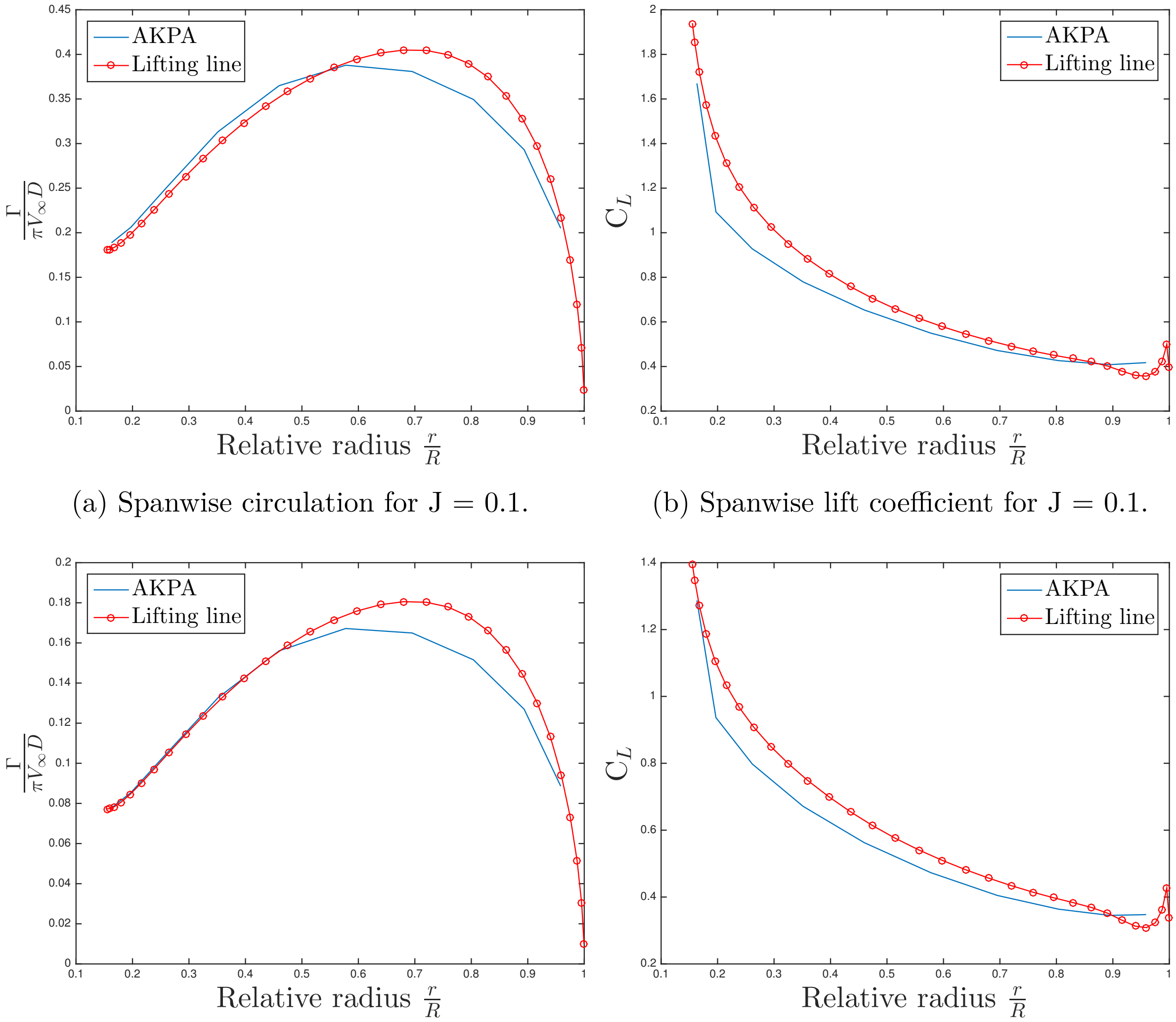
(a) Thrust coefficient K$_T$ from experiment and PropSiM calculations.

(b) Torque coefficient K$_Q$ from experiment and PropSiM calculations.

Figure 6.2: Comparison of open water thrust and torque coefficients from experimental results and PropSiM calculations for the P1374 propeller.

6.1.2 Open water spanwise circulation and lift distribution

Spanwise distribution of circulation and lift coefficient for the KVLCC2 propeller in open water has been validated against AKPA calculations for advance numbers between 0.1 and 0.8. The results are shown in Figure 6.3. We see a fairly good match between simulation in PropSiM and AKPA for high propeller loading, that is for advance numbers below 0.4. The match gets somewhat poor for $J = 0.5$ and $J = 0.6$, however the agreement returns for $J = 0.7$ and $J = 0.8$. For all advance numbers we see that the simulation model tends to overestimate the circulation and the lift coefficient, which already has been emphasized for the thrust and torque coefficients in section 6.1.1.



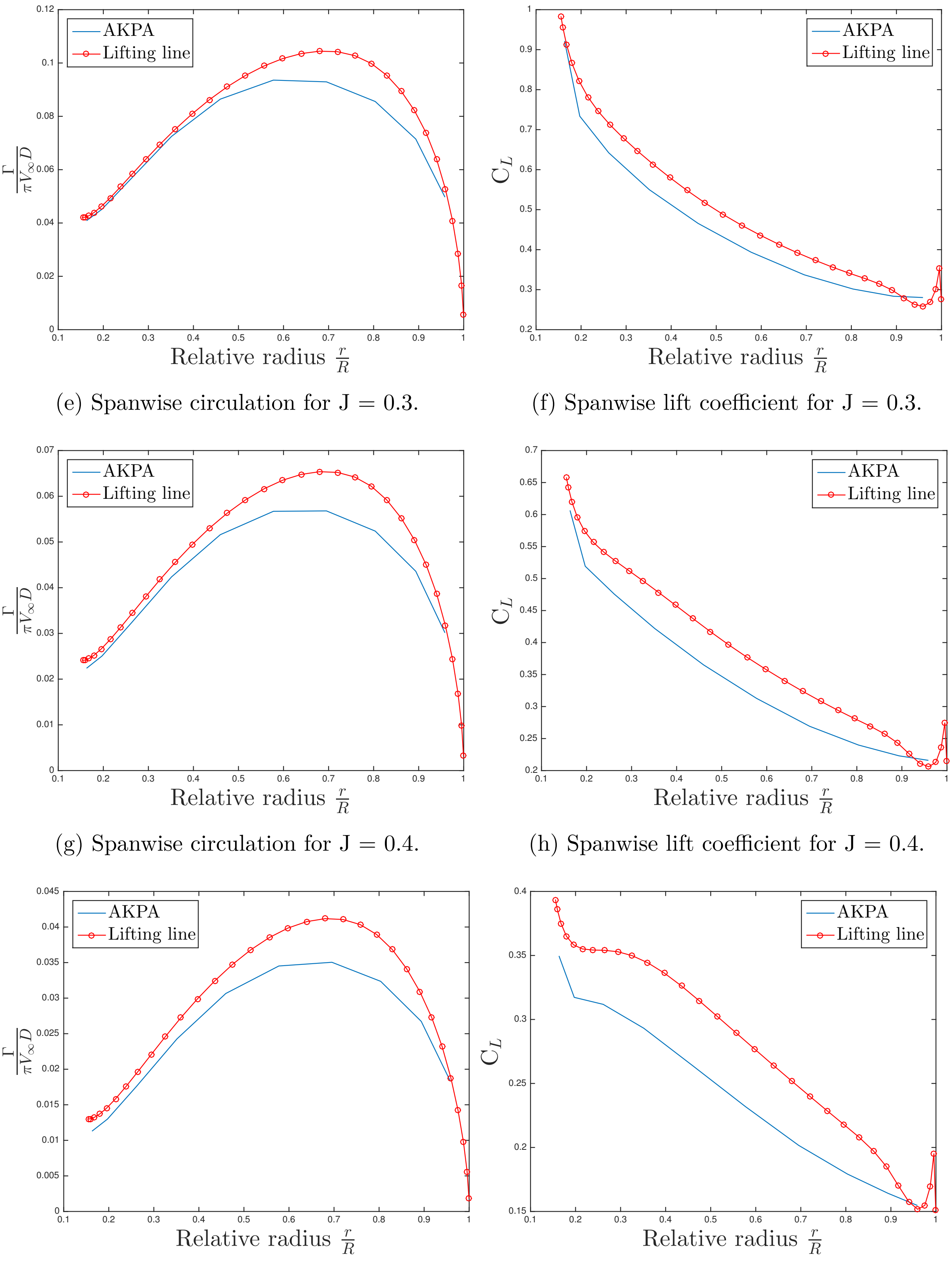
(a) Spanwise circulation for J = 0.1.

(b) Spanwise lift coefficient for J = 0.1.

(c) Spanwise circulation for J = 0.2.

(d) Spanwise lift coefficient for J = 0.2.

Figure 6.3: Spanwise circulation and lift coefficient from PropSiM and AKPA calculations.



(e) Spanwise circulation for J = 0.3.

(f) Spanwise lift coefficient for J = 0.3.

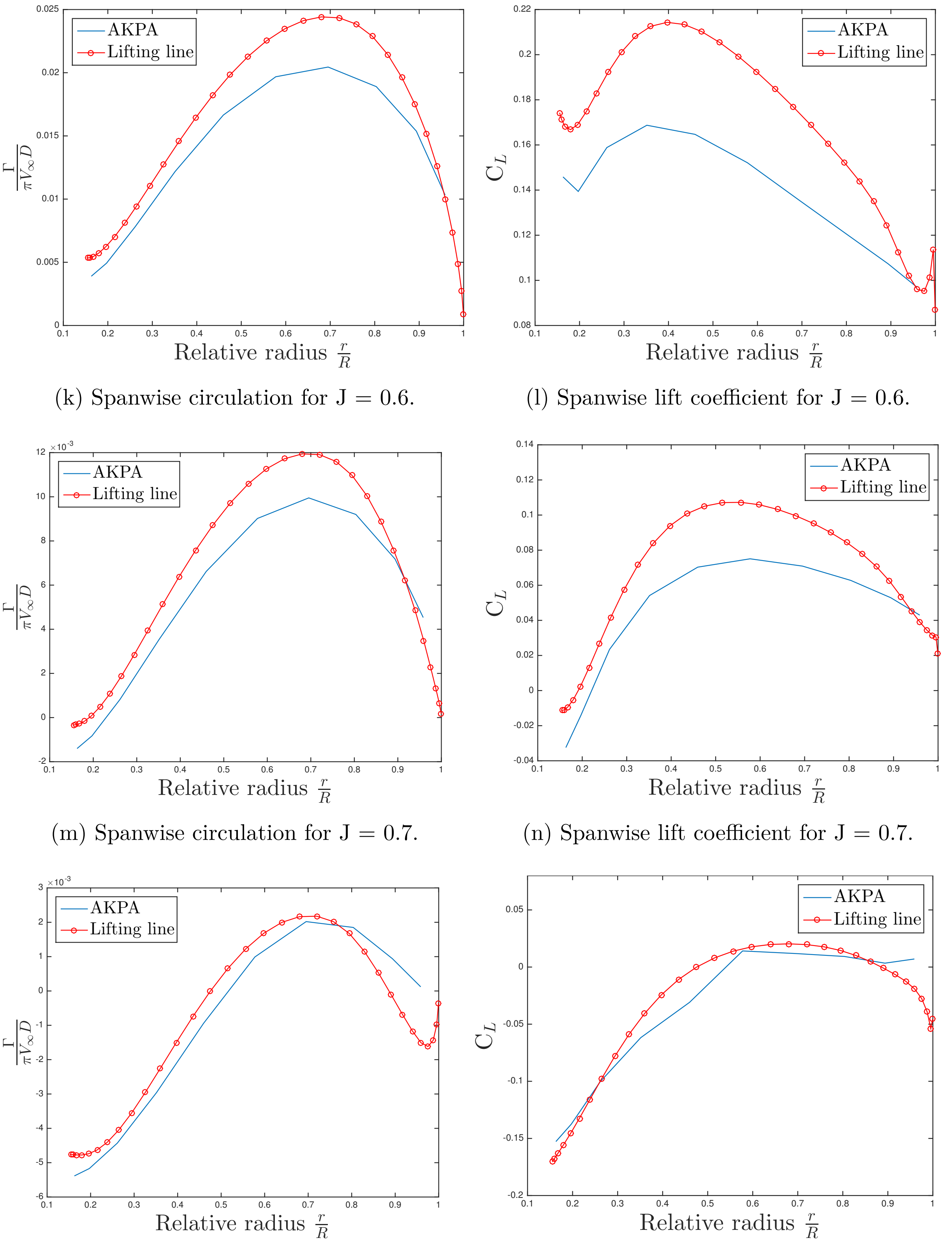
(g) Spanwise circulation for J = 0.4.

(h) Spanwise lift coefficient for J = 0.4.

(i) Spanwise circulation for J = 0.5.

(j) Spanwise lift coefficient for J = 0.5.

Figure 6.3: Spanwise circulation and lift coefficient from PropSiM and AKPA calculations.



(k) Spanwise circulation for J = 0.6.

(l) Spanwise lift coefficient for J = 0.6.

(m) Spanwise circulation for J = 0.7.

(n) Spanwise lift coefficient for J = 0.7.

(o) Spanwise circulation for J = 0.8.

(p) Spanwise lift coefficient for J = 0.8.

Figure 6.3: Spanwise circulation and lift coefficient from PropSiM and AKPA calculations.

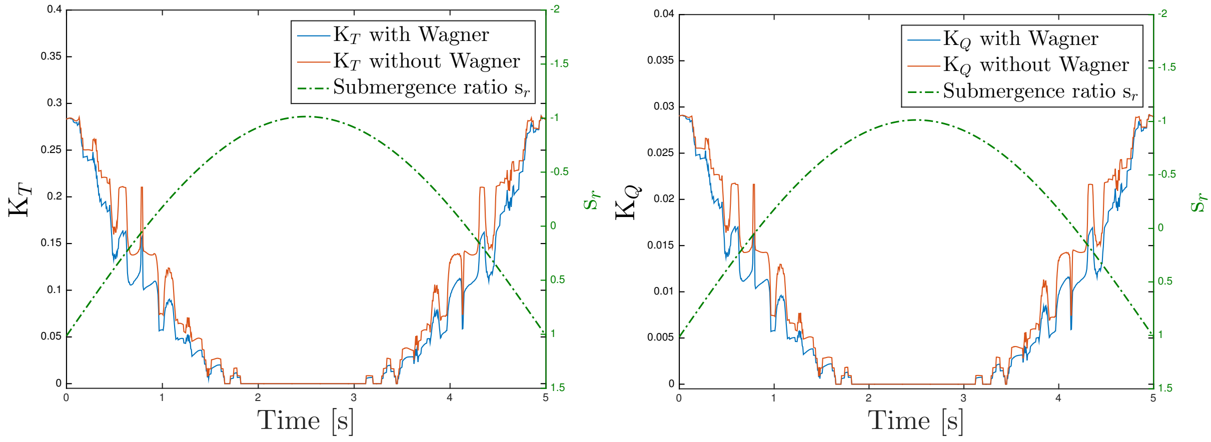
6.2 Physical behaviour of special effects

Different effects implemented into the simulation model have been isolated and tested for investigation of their physical behaviour. The subsequent sections present the results from the simulations, and related comments regarding the physical behaviour.

6.2.1 Wagner effect

The KVLCC2 propeller has been used to investigate the Wagner effect predicted by the simulation model. The strategy was to design two similar cases of a partly submerged condition, with turning the Wagner effect on/off as the only difference. The propeller was simulated in a sinusoidal heave motion, being completely out of water at its highest position. The ventilation model was deactivated for both of the simulations in order to isolate the Wagner effect as much as possible.

Figure 6.4 shows the thrust and torque coefficients with and without the Wagner effect, along with the submergence ratio of the propeller. The propeller had a shaft speed of $n = 76$[RPM] and advance number $J = 0.1$. The simulation results show that the Wagner effect leads to loss of both thrust and torque. In addition some more abrupt changes in thrust and torque can be identified when Wagner effect is turned off. The somewhat non-harmonical behaviour of the propeller harmonics can be explained by the very slowly rotating propeller. With a shaft frequency of only 76 [RPM] = 1.27 [Hz] the propeller submergence, and hence the Wagner effect, will change a lot during one propeller revolution. In addition one must have in mind that the calculation of Wagner effect depends on the wet vortex and control points, which will experience a stepwise rather than continuous change for varying propeller submergence.



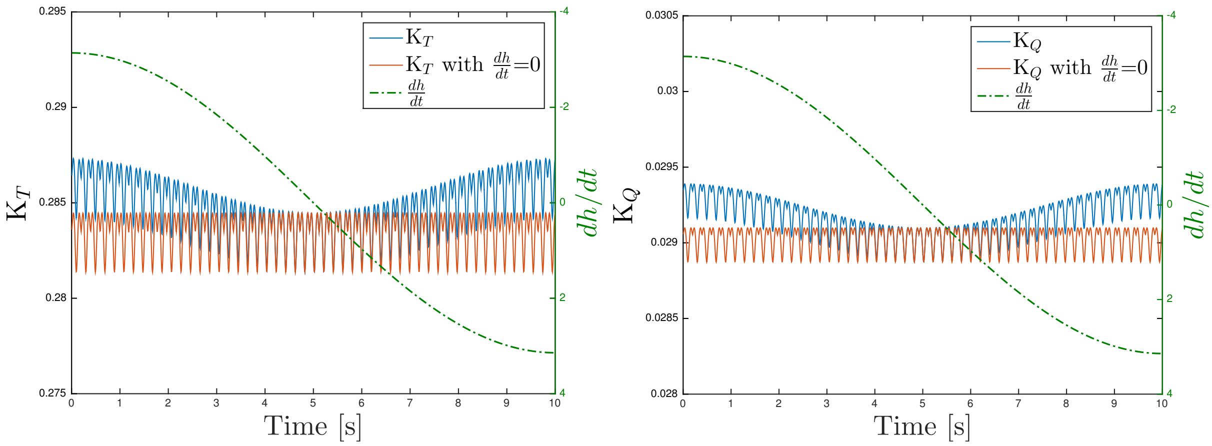
(a) Wagner effect on thrust coefficient K_T.

(b) Wagner effect on torque coefficient K_Q.

Figure 6.4: Effect of turning Wagner effect calculation on/off on thrust and torque coefficient for propeller in forced sinusoidal heave motion. n = 76 [RPM], J = 0.1.

6.2.2 Deeply submerged with harmonic variation

The KVLCC2 propeller has also been used to evaluate the effect of the vertical inflow component resulting from vertical motion of the propeller. During one revolution the propeller blade experiences a local harmonic varying tangential velocity component. If the vertical speed of the propeller is not constant, i.e. $\frac{d^2\mathrm{h}}{dt^2} \neq 0$, this will cause harmonic variation of the blade lift force, resulting in low frequent harmonic fluctuations of thrust and torque. This effect is shown in Figure 6.5 for a sine varying propeller submergence, which has a non-constant cosine varying $\frac{dh}{dt}$. In addition the same propeller has been simulated at constant submergence, which is shown in Figure 6.5. We can clearly identify the low frequent fluctuation in thrust and torque due to the vertical motion of the propeller. We also see that the vertical motion does not affect the amplitude of the high frequency propeller harmonics.



(a) Thrust coefficient K$_T$.

(b) Torque coefficient K$_Q$.

Figure 6.5: K$_T$ and K$_Q$ for propeller at constant submergence compared to propeller exposed to sine varying propeller submergence, showing low frequent harmonic fluctuations of thrust and torque.

6.3 Partly submerged propeller

Validation of the simulation model in case of partly submerged condition is carried out using four model experimental cases collected from Koushan (2006). The experiments were performed on an open pulling propeller exposed to forced sinusoidal heave motion. Tests were conducted at different propeller rate of revolutions and carriage speeds at both constant submersions as well as with periodically varying submersions. The propeller simulation results presented in Dalheim (2014) are also shown along with the four validation cases. Data for the submergence conditions are given in Table 6.1.

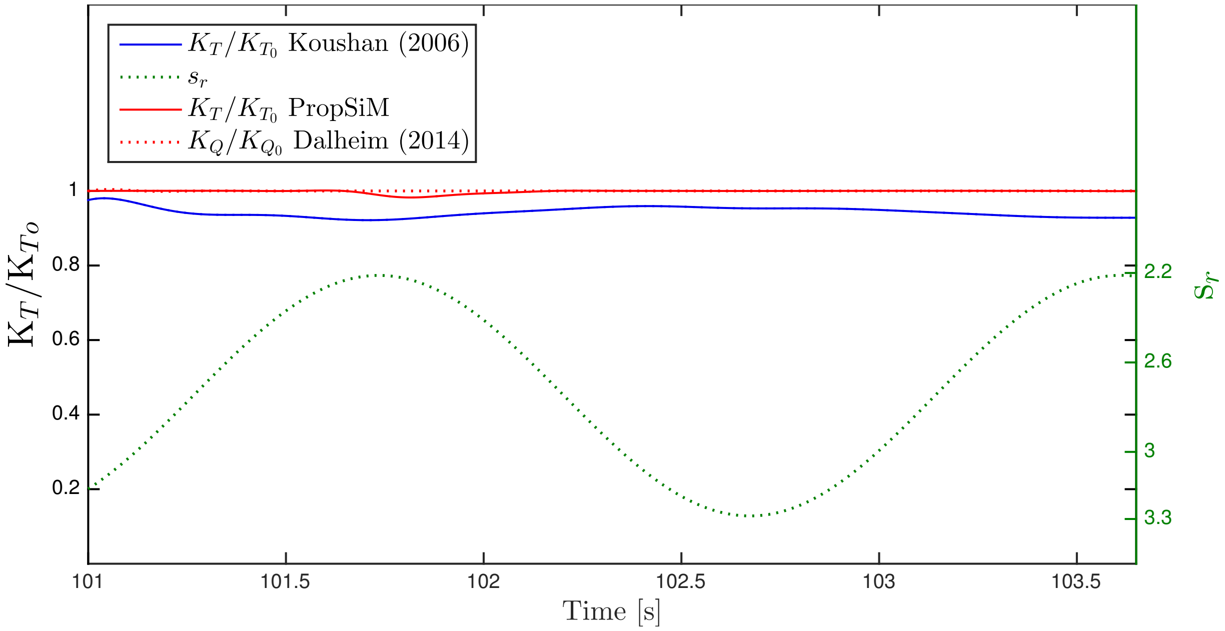
The first validation case is a non-ventilating condition of a propeller being forced in sinusoidal heave motion. The other three cases are all ventilating conditions, while the last two cases are in addition surface piercing conditions.

Table 6.1: Four experimental test conditions for propeller forced in sinusoidal heave motion.

Case	$\frac{h}{R}$ at highest position ($s_{r_{min}}$)	$\frac{h}{R}$ amplitude (s_{r_A})	$\frac{h}{R}$ period (s_{r_T})
1	2.2	0.55	1.87
2	1.1	1.075	2.0
3	-0.15	1.075	2.0
4	-1.0	1.075	2.0

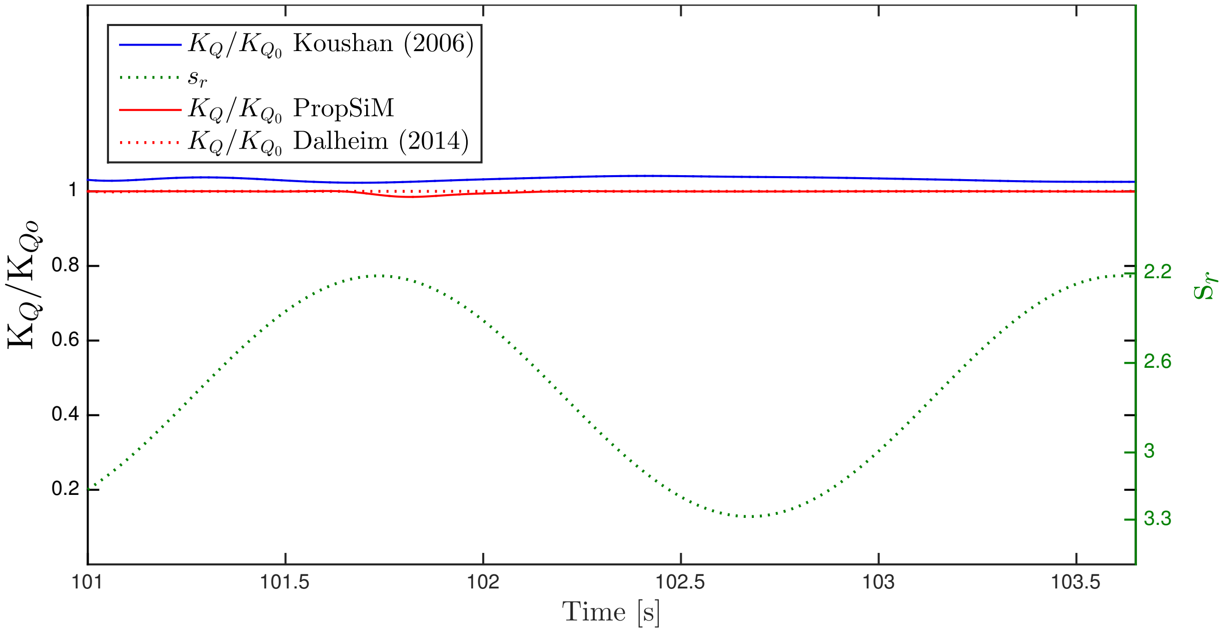
The results from both model experiments and propeller simulation are filtered using a low-pass filter. The motive is to investigate the ventilation effects, so the high frequency harmonics are of secondary importance.

6.3.1 Case 1



(a) Relative thrust coefficient K_T/K_{T0}.

Figure 6.6: Validation of simulation model with experimental results from Koushan (2006) for propeller subject to forced sinusoidal heave motion.
Case 1: $s_{r_{max}} = 3.30$, $s_{r_{min}} = 2.20$, $s_{r_A} = 0.55$, $s_{r_T} = 1.87$ [s].

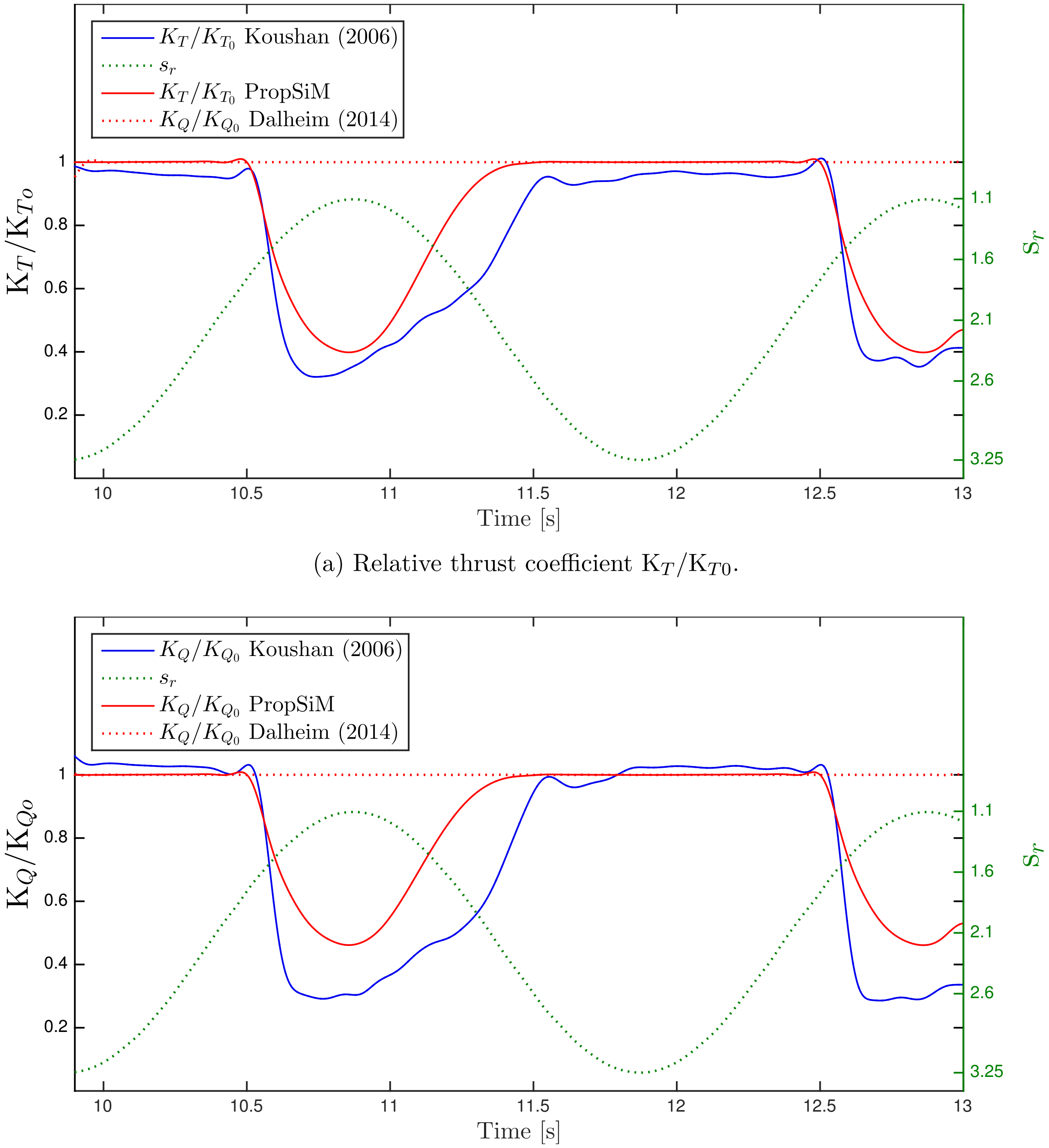


(b) Relative torque coefficient $\mathrm{K}_Q/\mathrm{K}_{Q0}$.

Figure 6.6: Validation of simulation model with experimental results from Koushan (2006) for propeller subject to forced sinusoidal heave motion.
Case 1: $s_{r_{max}} = 3.30$, $s_{r_{min}} = 2.20$, $s_{r_A} = 0.55$, $s_{r_T} = 1.87$ [s].

The simulation results in the non-ventilating condition match very well with the experimental result. Only small deviations can be found. Part of this deviation could possibly be due to the blade force measurements.

6.3.2 Case 2



(a) Relative thrust coefficient K_T/K_{T0}.

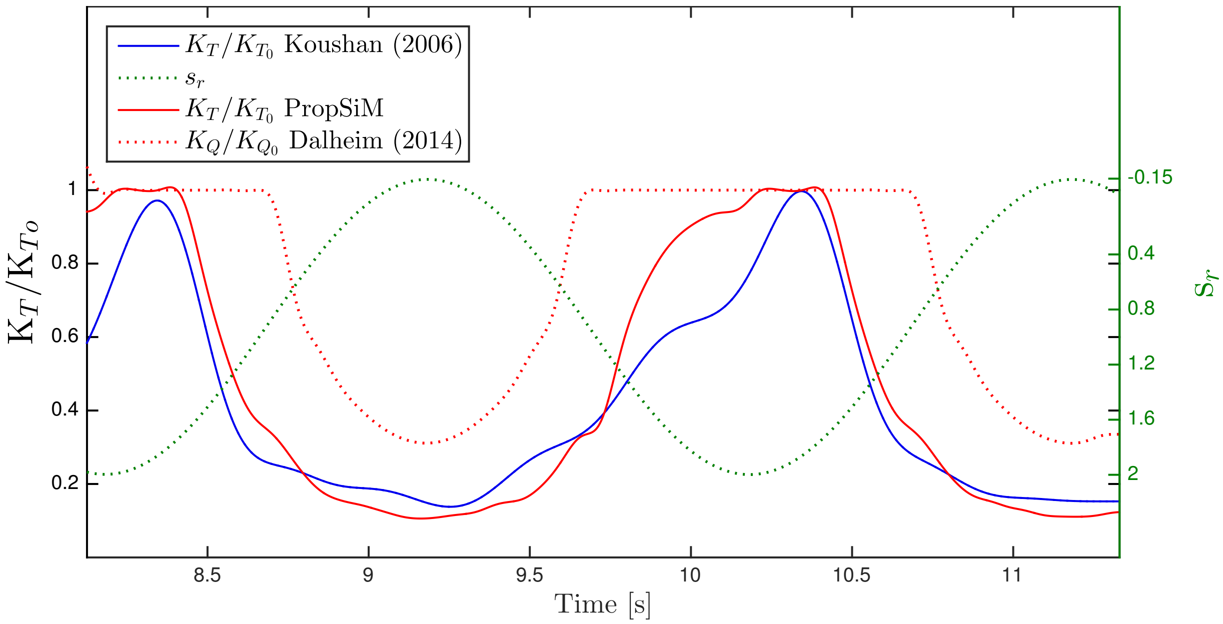
(b) Relative torque coefficient K_Q/K_{Q0}.

Figure 6.7: Validation of simulation model with experimental results from Koushan (2006) for propeller subject to forced sinusoidal heave motion.
Case 2: $s_{r_{max}} = 3.25$, $s_{r_{min}} = 1.10$, $s_{r_A} = 1.075$, $s_{r_T} = 2.0$ [s].

In the second case the lowest submergence ratio is $s_{r_{min}} = 1.10$. Dalheim (2014) predicted no effect of ventilation in this case, however the experiment shows that the amount of ventilation will be substantial even if the propeller is submerged during the complete forced heave motion. PropSim is able to reproduce this ventilation mechanism in a very convincing way, and we see a very good agreement on both how the thrust drops and how the thrust reconstructs. The deviation in torque is somewhat larger than for the thrust. However, as the loss factor for torque is related to the loss factor for thrust through a constant (see (3.46)), some better agreement with the experimental result can be achieved. This simulation was run with $k = 0.82$.

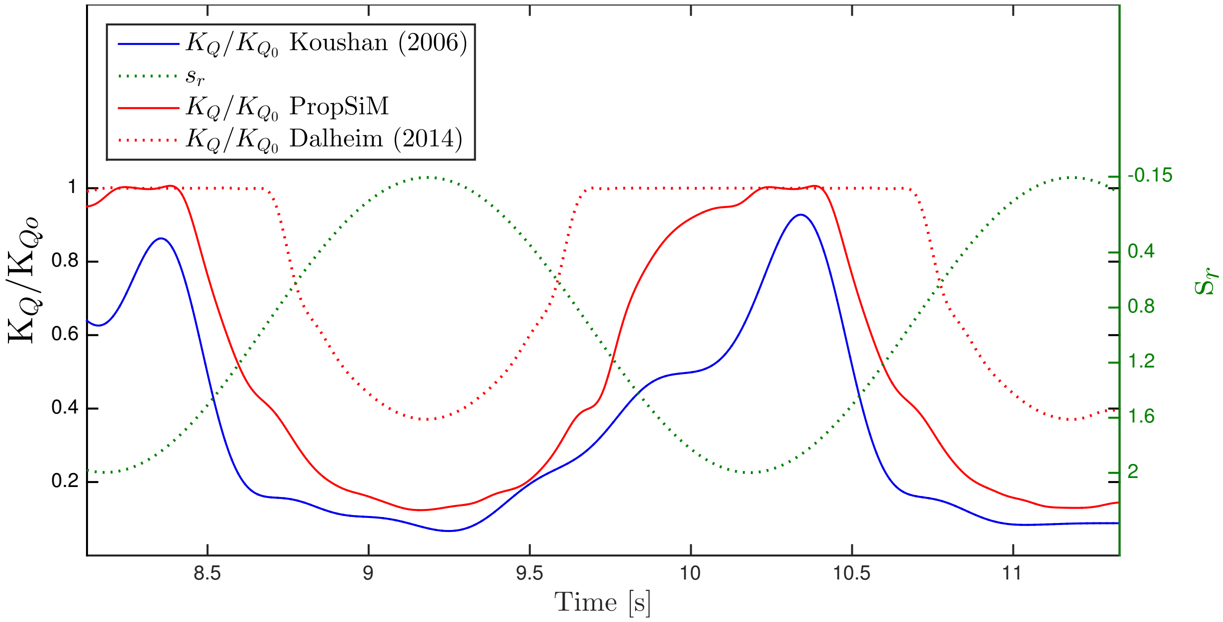
6.3.3 Case 3

In this ventilation case even the propeller hub leaves the water, and results from Dalheim (2014) are in better agreement with the experiment. However we clearly see that the simplified simulation model recovers from the thrust and torque loss very soon after the propeller enters the water again, and that thrust and torque loss are delayed when the propeller penetrates the water surface on its way up. This was expected to be due to massive suction of air down to the propeller, especially as the propeller is leaving its highest position and going down into the water again. The simulation result from PropSim shows again a much better agreement with the experiments. We see that the ventilation occurs almost at the same time as in the experiment, and that it takes some time to rebuild the thrust again after the propeller is fully submerged. The new model behaves more physically, and much of the same mechanisms that are found in the experiment are reproduced by this model. The torque is almost in similar agreement with the experiments as the thrust.



(a) Relative thrust coefficient K$_T$/K$_{T0}$.

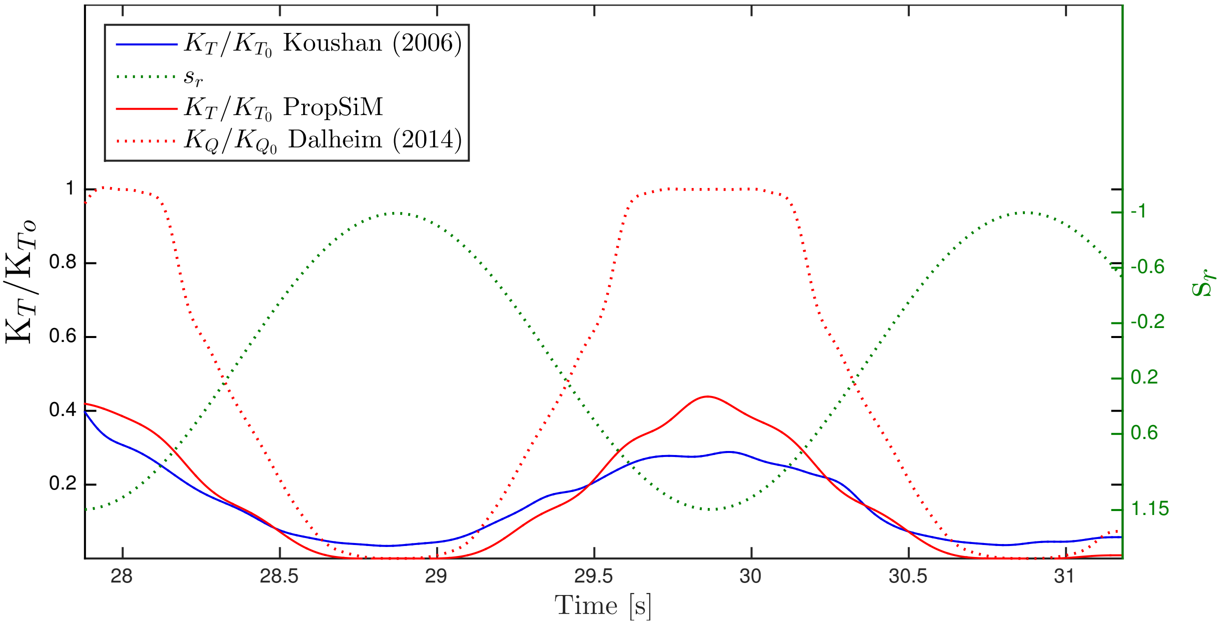
Figure 6.8: Validation of simulation model with experimental results from Koushan (2006) for propeller subject to forced sinusoidal heave motion.
Case 3: $s_{r_{max}}$ = 2.0, $s_{r_{min}}$ = -0.15, s_{r_A} = 1.075, s_{r_T} = 2.0 [s].



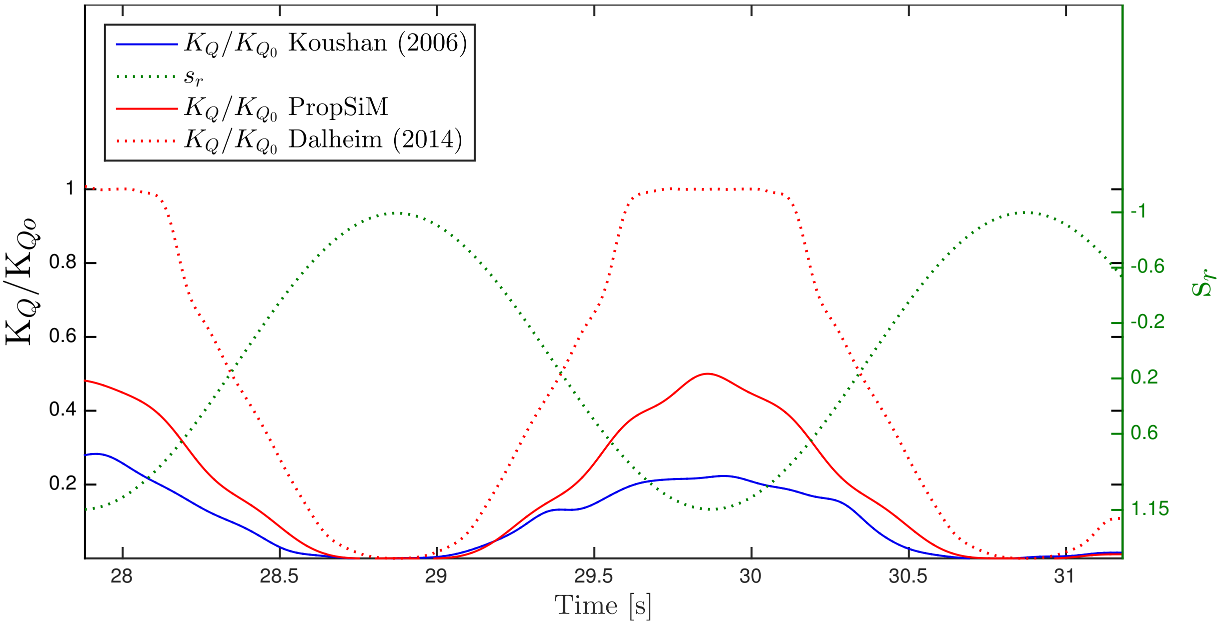
(b) Relative torque coefficient K$_Q$/K$_{Q0}$.

Figure 6.8: Validation of simulation model with experimental results from Koushan (2006) for propeller subject to forced sinusoidal heave motion.
Case 3: $s_{r_{max}}$ = 2.0, $s_{r_{min}}$ = -0.15, s_{r_A} = 1.075, s_{r_T} = 2.0 [s].

6.3.4 Case 4



(a) Relative thrust coefficient K$_T$/K$_{T0}$.



(b) Relative torque coefficient K$_Q$/K$_{Q0}$.

Figure 6.9: Validation of simulation model with experimental results from Koushan (2006) for propeller subject to forced sinusoidal heave motion.
Case 4: $s_{r_{max}}$ = 1.15, $s_{r_{min}}$ = -1.00, s_{r_A} = 1.075, s_{r_T} = 2.0 [s].

In the last validation example of ventilation events, the propeller is going completely in and out of the water for each cycle. Dalheim (2014) found that the simulated maximum loss of thrust and torque matched well with the experiment, but that the same development as for case 3 could be seen: the simulation model recovers from thrust and torque loss much too soon. In this case the thrust and torque never recover from the loss, and suffer from substantial ventilation during the complete heave motion. PropSiM is able to reproduce this effect in a very convincing way. The amount of thrust loss is almost correct, and the time of incidence for the thrust loss follows the experiment completely. The torque loss is somewhat underestimated, however as the loss factor for torque is related to the loss factor for thrust through a constant (see (3.46)), some better agreement with the experimental result can be achieved. This simulation was run with $k = 0.82$.

In total these four cases show that the new simulation model can predict very physically correct ventilation patterns, and that the model produces time domain propeller forces very similar to what is found in experiments.

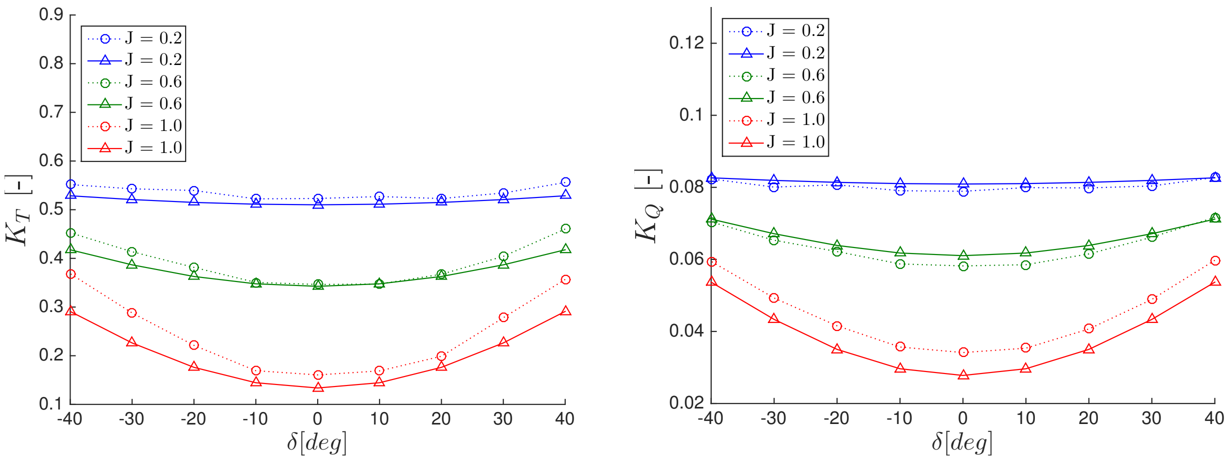
6.4 Oblique inflow

For conditions of oblique inflow the validation of the simulation model is based on experimental results published by Amini (2011). The validation has been carried out for advance numbers $J = 0.2$, $J = 0.6$ and $J = 1.0$ and for azimuth angles δ varying in a step of 10° from -40° to 40° for the P1374 propeller. Figure 6.10 (a) to (f) show how the PropSiM simulation behaves relative to experiments. Continuous lines are used for results from simulation model, while the experimental results are shown with dashed lines.

Figure 6.10 (a) and (b) show the propeller thrust and torque coefficients for the simulated conditions. As expected, the thrust and torque coefficients increase with increasing absolute value of the azimuth angle δ. This is because the component of the incoming velocity in the direction of the propeller axis decreases with increasing azimuth angle. In other words, the effective advance number seen by the propeller will be decreased, causing higher thrust and torque. The agreement with the experimental results is good for $J = 0.2$ and $J = 0.6$, also for large azimuth angles. For $J = 1.0$ the difference between simulation and experiments is somewhat larger, at least for large azimuth angles. Parts of the difference can be explained by the inaccuracy in the lifting line approach, since there is a difference between simulated result and experiment for $\delta = 0$ as well. The contribution from the oblique inflow calculation is thus in better agreement with experiments than Figure 6.10 (a) and (b) indicate, especially with respect to the torque coefficient where almost a parallel shift of the curve can be observed for $J = 1.0$.

Comparison of propeller side force coefficients is shown in Figure 6.10 (c) and (d). As for the thrust and torque the agreement is good for $J = 0.2$. For the horisontal side force the agreement is also sufficient for $J = 0.6$. For $J = 1.0$ the horisontal side force is underestimated compared to the experiments. The agreement of the vertical side force for lower propeller loading is not as good as for the horisontal side force. The results show that the vertical side force is largely underestimated for lower propeller loadings. The reason for this is expected to be related to the implementation of the simplified oblique momentum theory outlined in section 3.4.2. The same observation can be made for the bending moment coefficients, shown in Figure 6.10 (e) and (f). The agreement in horisontal bending moment is much better than for the vertical bending moment, and we can observe that the horisontal bending moment is very well predicted also for low propeller loadings. The vertical bending moment is largely underestimated, especially for low propeller loadings.

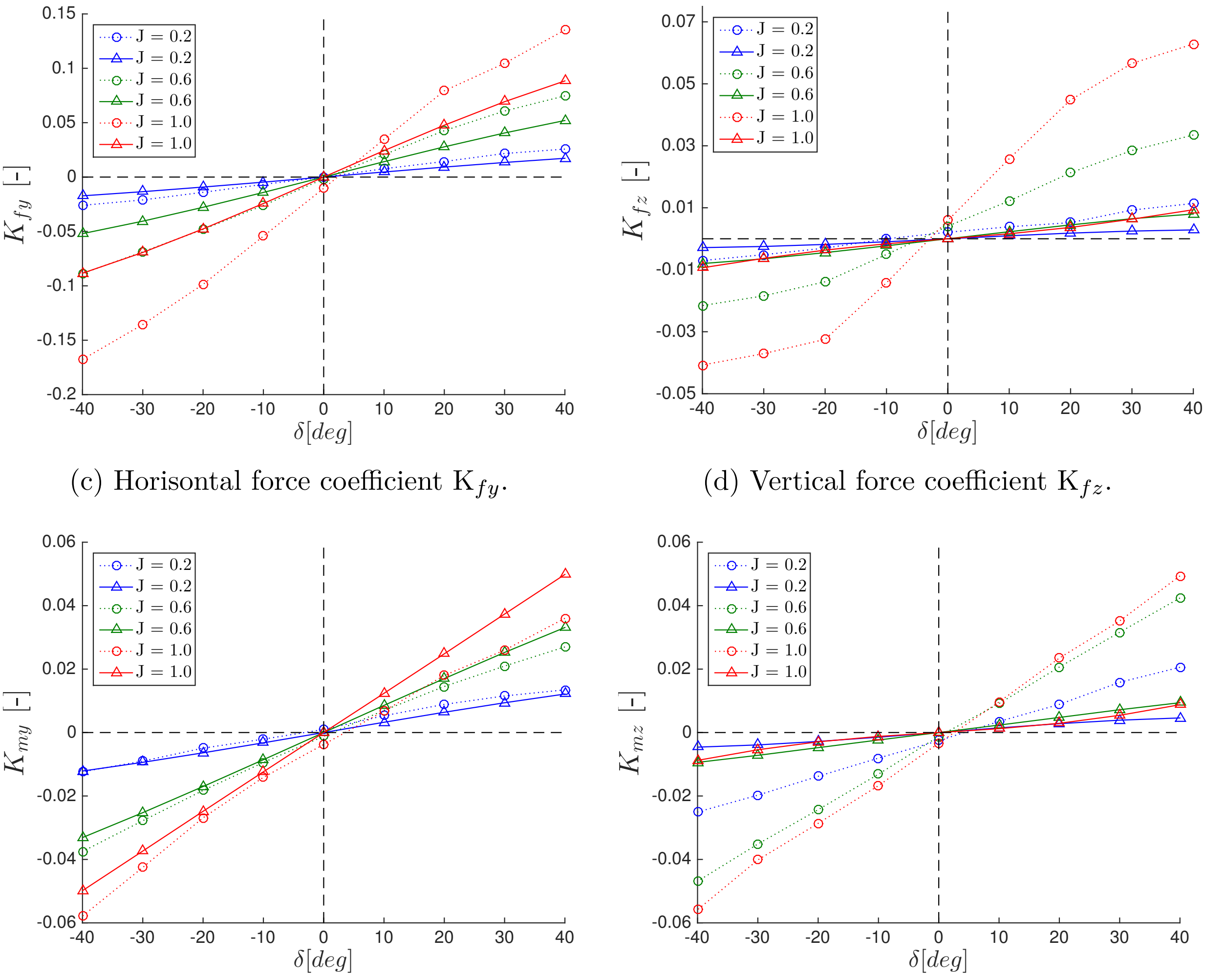
In Figure 6.11 (a) and (b) the propeller shaft total side force coefficient K_s and bending moment coefficient K_b are shown. In general we can observe that the simulated result is underestimated compared to experiments. This can mainly be explained by the error in vertical side force and bending moment. Yet we can observe that the simulation model predicts similar behaviour as found in experiments for the total side force and bending moment coefficients for increasing advance number and azimuth angles.



(a) Thrust coefficient K$_T$.

(b) Torque coefficient K$_Q$.

Figure 6.10: Validation of simulation model (——△——) with experimental results from Amini (2011) (· · · ○ · · ·) for 6 DoF propeller forces in oblique inflow condition with propeller P1374. Inflow angle δ between -40° and 40°. Advance number J=0.2, J=0.6 and J=1.0.



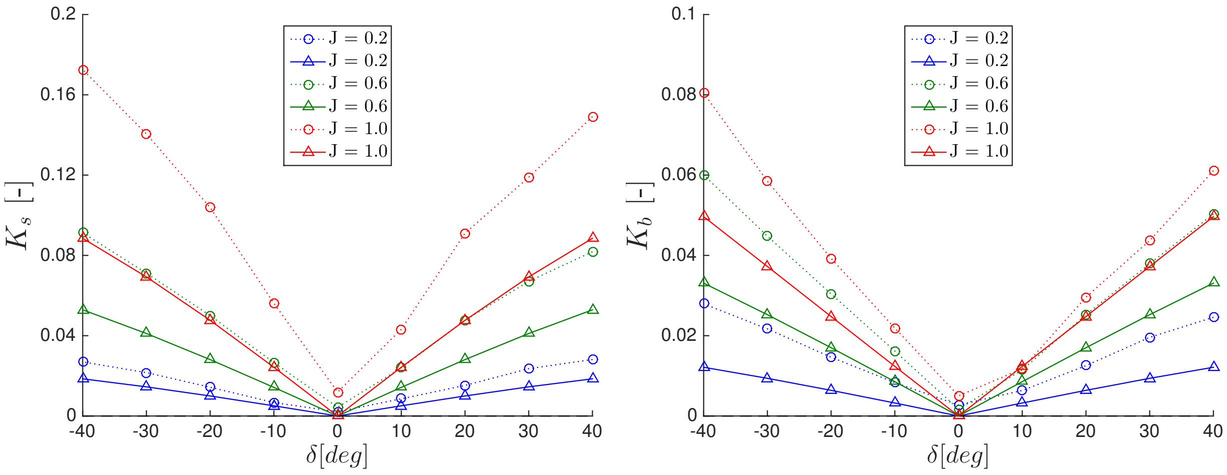
(c) Horisontal force coefficient K_{fy}.

(d) Vertical force coefficient K_{fz}.

(e) Horisontal moment coefficient K_{my}.

(f) Vertical moment coefficient K_{mz}.

Figure 6.10: Validation of simulation model (—△—) with experimental results from Amini (2011) (· · · ○ · · ·) for 6 DoF propeller forces in oblique inflow condition with propeller P1374. Inflow angle δ between -40° and 40°. Advance number J=0.2, J=0.6 and J=1.0.



(a) Total side force coefficient K$_s$.

(b) Total bending moment coefficient K$_b$.

Figure 6.11: Validation of simulation model (—△—) with experimental results from Amini (2011) (· · · ○ · · ·) for propeller shaft total side force and bending moment coefficients in oblique inflow condition with propeller P1374. Inflow angle δ between -40° and 40°. Advance number J=0.2, J=0.6 and J=1.0.

6.5 Computational effort

For a deeply submerged propeller the computational time has been found to be satisfactory for implementation into a MBS model. However, inclusion of various types of effects has considerable influence on the computational speed. The most important effect with respect to demand of computational effort is the Wagner effect. This has to do with the fact that the unknown circulation distribution must be determined at several blade positions for each time step. If the simulation model is processing very slowly, the density of the Wagner effect calculation should be reduced. How this can be done is further described in section 5.2.3.

7 Conclusions and recommendations

7.1 Conclusion

Based on a study of the present state of knowledge regarding time domain simulation of propeller performance, the simulation model PropSiM has been built based on a lifting line approach. Various operating conditions have been implemented into the simulation model, such that time domain simulation of a propeller operating in a physical environment is possible. The propeller simulation model has shown to be able to predict 6 DoF propeller forces with satisfactory results, and the computational efficiency of the model has shown to be satisfactory, especially for a deeply submerged propeller. Overall, the work can conclude on the following:

- PropSiM provides 6 DoF propeller forces in time domain for an arbitrary propeller. The model catches important effects on a propeller operating in various environmental conditions.

- The lifting line approach has shown a tendency of overestimating the propeller thrust and torque, however, the approach has been found to be suitable for implementation into a time domain simulation model, giving satisfactory results.

- The vertical side force and bending moment has shown not to be conservative in the simulation model when the propeller operates in oblique inflow. Lifting line theory has been found to be challenging and computational demanding when it is used in combination with momentum theory for skewed propeller wake.

- The required computational effort is largely depending on the effects included in the simulation. Computational time is saved when parts of a propeller blade is out of water, however the calculation of the Wagner effect is very demanding and increases the computational effort substantially.

- The ventilation model works very well, and is able to predict important physical effects during ventilation. The model has been validated against different types of ventilation incidents, and has proved to replicate the magnitude and time of incidence of both thrust loss and thrust regeneration in a convincing manner.

7.2 Recommendations for future work

The work related to the propeller simulation model has expanded the author's knowledge on the field of study. Important experience regarding different effects on propellers has been achieved, however additional experience related to combining different effects efficiently in a time domain has been more important. The propeller simulation model has shown to be able to predict 6 DoF propeller forces with satisfactory results, however there are several improvements that can be implemented. The author's thoughts on recommended improvements for a future release of PropSiM are:

- Include memory blocks to the simulation model for improvement of computational speed during constant and repeatedly operating conditions.

- Improve both speed and precision in calculation of the Wagner effect, by increasing the density of computational points towards the water entry phase and lower the density towards the position of the blade.

- Avoid superfluous calculation of Wagner effect by implementing a routine that shuts down the Wagner calculation if the propeller blade is sufficiently far away from the position of water entry.

- Implement calculation of impact loads on a propeller operating in waves by adding a slamming model for partly submerged propeller.

- Modify the simulation model to account for direction of propeller rotation.

- Improve the model for conditions of oblique inflow such that vertical bending moments are more accurately simulated.

- Further validation of complete simulation model against experiments, with particularly emphasis on combining different effects.

- Include a physical model for estimating the ventilated area of the propeller disc based on propeller loading.

- Develop a correction routine to the quasi-steady analysis to account for unsteady effects.

Bibliography

Amini, H. (2011). *Azimuth propulsors in off-design conditions.* Doctoral thesis, NTNU, Trondheim.

Bernitsas, M., Ray, D., and Kinley, P. (1981). KT, KQ and Efficiency Curves for the Wageningen B-Series Propellers. Technical report, Department of Naval Architecture and Marine Engineering. The University of Michigan.

Blanke, M., Lindegaard, K.-P., and Fossen, T. I. (2000). Dynamic model for thrust generation of marine propellers. *5th IFAC Conference on Manoeuvring and Control of Marine Craft*, pages 363–368.

Califano, A. (2010). Dynamic loads on marine propellers due to intermittent ventilation.

Coleman, R. P., Feingold, A. M., and Stempin, C. W. (1945). Evaluation of the induced velocity field of an idealized helicoptor rotor. Technical report.

Dalheim, Ø. Ø. (2014). *Development of a simplified simulation model for propeller performance.* Project thesis, Norwegian University of Science and Technology.

Epps, B. P. and Kimball, R. W. (2013). OpenProp v3: Open-source software for the design and
analysis of marine propellers and horizontal-axis turbines.
URL: http://engineering.dartmouth.edu/epps/openprop, Retrieved: 2015-06-05.

Faltinsen, O. M., Minsaas, K. J., Liapis, N., and Skjø rdal, S. O. (1981). Prediction of Resistance and Propulsion of a Ship in a Seaway. *Proceedings of 13th symposium on naval hydrodynamics*, pages 505–529.

Fleischer, K. P. (1973). Untersuchungen über das Zusammenwirken von Schiff und Propeller bei teilgetauchten Propellern (Investigations of the interaction between ship and propeller for partly submerged propellers).

Froude, R. E. (1889). *On the part played in propulsion by differences of fluid pressure.* Transactions of the Royal Institution of Naval Architects.

Froude, R. E. (1911). *The acceleration in front of a propeller.* Transactions of the Royal Institution of Naval Architects.

Glauert, H. (1935). Airplane propellers. In *Aerodynamic theory*, pages 169–360. Springer.

Gutsche, F. (1967). Einfluss der Tauchung auf Schub und Wirknungsgrad von Schiffpropellern. Schiffbauforchung 6 (5/6), 256—277. *German."Influence of Submergence on Thrust and Efficiency of Ship Propellers.*

Helicetude (2015). URL: www.helicetude.com/english/RudderPropeller.asp, Retrieved: 2015-06-05.

Hess, J. L. and Valarezo, W. O. (1985). Calculation of steady flow about propellers using a surface panel method. *Journal of Propulsion and Power*, 1(6):470–476.

Hutchison, S., Steen, S., and Savio, L. (2014). Modelling of propeller hydrodynamics for implementation with multi-body simulation. Technical report, Trondheim.

Kerwin, J. E. (2001). Hydrofoils and Propellers. (January).

Kerwin, J. E. and Lee, C. S. (1978). Prediction of teady and unsteady marine propeller performance by numerical lifting-surface theory. *Publication of: Society of Naval Architects and Marine Engineers.*

Koushan, K. (2004). Environmental and Interaction Effects on Propulsion Systems used in Dynamic Positioning, an overview. *PRADS 2004: 9th International Symposium on Practical Design of Ships and Other Floating Structures*, pages 1013–1020.

Koushan, K. (2006). Dynamics of Ventilated Propeller Blade Loading on Thrusters due to Forced Sinusoidal Heave Motion. *Proceedings of 26th Symposium on Naval Hydrodynamics*, (September):17–22.

Koushan, K. (2007). Dynamics of Propeller Blade and Duct Loadings on Ventilated Ducted Thrusters due to Forced Periodic Heave Motion. *Proceedings of International Conference on Violent Flows (VF-2007).*

Kozlowska, A. M. and Steen, S. (2010). Ducted and Open Propeller Subjected to Intermittent Ventilation. In *Eighteenth International Conference on Hydrodynamics in Ship Design, Safety and Operation*, Gdansk.

Kozlowska, A. M., Steen, S., and Koushan, K. (2009). Classification of different type of propeller ventilation and ventilation inception mechanism. *First International Symposium on Marine Propulsors, smp'09*, (June):2–9.

Krasilnikov, V. I., Berg, A., Achkinadze, A. S., Stepanov, I. E., and Sun, J. (2011). AKPA-CP-DT 5.6 Manual.

Lan, C. E. (1974). A Quasi-Vortex-Lattice Method in Thin Wing Theory. *Journal of Aircraft*, 11(9):518–527.

Lerbs, H. W. (1952). *Moderately loaded propellers with a finite number of blades and an arbitrary distribution of circulation.* Society of Naval Architects and Marine Engineers.

Madsen, H. A., Mikkelsen, R., Øye, S., Bak, C., and Johansen, J. (2007). A Detailed investigation of the Blade Element Momentum (BEM) model based on analytical and numerical results and proposal for modifications of the BEM model. *Journal of Physics: Conference Series*, 75:012016.

Minsaas, K., Faltinsen, O., and Persson, B. (1983). On the importance of added resistance, propeller immersion and propeller ventilation for large ships in a seaway. In *Proceedings of International Symposium on Practical Design of Ships and other Floating Structures - PRADS'83*.

Moriarty, P. and Hansen, A. (2005). AeroDyn theory manual. Technical Report January, National Renewable Energy Laboratory.

Olofsson, N. (1996). *Force and Flow Characteristics of a Partially Submerged Propeller*. Chalmers University of Technology.

Phillips, A. B., Turnock, S. R., and Furlong, M. (2009). Evaluation of manoeuvring coefficients of a self-propelled ship using a blade element momentum propeller model coupled to a Reynolds averaged Navier Stokes flow solver. *Ocean Engineering*, 36(15-16):1217–1225.

Pitt, D. M. and Peters, D. A. (1981). Theoretical prediction of dynamic-inflow derivatives. *Vertica*, 5(1):21–34.

Politis, G. K. (2004). Simulation of unsteady motion of a propeller in a fluid including free wake modeling. *Engineering Analysis with Boundary Elements*, 28(6):633–653.

Rankine, W. J. M. (1865). *On the mechanical principles of the action of propellers*. Transactions of the Royal Institution of Naval Architects.

Rwigema, M. (2010). Propeller blade element momentum theory with vortex wake deflection. *Proceedings of the 27th Congress of the International ...*, pages 1–9.

Sclavounos, P. D. (1987). An unsteady lifting-line theory. *Journal of Engineering Mathematics*, 21(3):201–226.

Wang, M.-H. (1985). *Hub Effects in Propeller Design and Analysis*. PhD thesis, MIT.

Wrench, J. W. (1957). The calculation of propeller induction factors. Technical report, Washington.

A Wrench's closed form approximations

For $r_c < r_v$:

$$\overline{u}_a(r_c) = \frac{Z}{4\pi r_c}\left(y - 2Zr_vF_1\right) \tag{A.1}$$

$$\overline{u}_t(r_c) = \frac{Z^2}{2\pi r_c}y_0F_1 \tag{A.2}$$

For $r_c > r_v$:

$$\overline{u}_a(r_c) = -\frac{Z^2}{2\pi r_c}yy_0F_2 \tag{A.3}$$

$$\overline{u}_t(r_c) = \frac{Z}{4\pi r_c}\left(1 + 2Zy_0F_2\right) \tag{A.4}$$

where

$$\begin{aligned} F_1 \approx &-\frac{1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{0.25}\left\{\frac{1}{U^{-1}-1} + \frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}} + \frac{3y^2-2}{(1+y^2)^{1.5}}\right]\right. \\ &\left.\ln\left(1 + \frac{1}{U^{-1}-1}\right)\right\} \end{aligned} \tag{A.5}$$

$$\begin{aligned} F_2 \approx &-\frac{1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{0.25}\left\{\frac{1}{U-1} - \frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}} + \frac{3y^2-2}{(1+y^2)^{1.5}}\right]\right. \\ &\left.\ln\left(1 + \frac{1}{U-1}\right)\right\} \end{aligned} \tag{A.6}$$

and

$$U = \left\{\frac{y_0(\sqrt{1+y^2}-1)}{y(\sqrt{1+y_0^2}-1)}\exp\left(\sqrt{1+y^2} - \sqrt{1+y_0^2}\right)\right\}^Z \tag{A.7}$$

$$y = \frac{r_c}{r_v\tan\beta_w} \tag{A.8}$$

$$y_0 = \frac{1}{\tan\beta_w} \tag{A.9}$$

Collected from Kerwin (2001).

B Derivation of discrete sum of thrust and torque

Starting with

$$C_T = \frac{T}{\frac{1}{2}\rho V^2 \pi R^2} = \frac{T}{\frac{1}{2}\rho V^2 \pi \frac{D^2}{4}} = \frac{8T}{\pi \rho V^2 D^2} \tag{B.1}$$

$$C_Q = \frac{Q}{\frac{1}{2}\rho V^2 \pi R^3} = \frac{Q}{\frac{1}{2}\rho V^2 \pi \frac{D^3}{8}} = \frac{16Q}{\pi \rho V^2 D^3} \tag{B.2}$$

$$T = \rho Z \sum_{m=1}^{M} \left[V_\infty \Gamma \cos(\beta_i) - \frac{1}{2}(V_\infty)^2 c C_{Dv} \sin(\beta_i) \right] \mathrm{d}r \tag{B.3}$$

and

$$Q = \rho Z \sum_{m=1}^{M} \left[V_\infty \Gamma \sin(\beta_i) + \frac{1}{2}(V_\infty)^2 c C_{Dv} \cos(\beta_i) \right] r \mathrm{d}r \tag{B.4}$$

Then combining the equations to get:

$$\begin{aligned} C_T &= \frac{8}{\pi \cancel{\rho} V^2 D^2} \cancel{\rho} Z \sum_{m=1}^{M} \left[V_\infty \Gamma \cos(\beta_i) - \frac{1}{2} V_\infty^2 c\, C_{Dv} \sin(\beta_i) \right] \mathrm{d}r \\ &= \frac{4\,\cancel{8}}{\cancel{\pi} \cancel{V^2}\, \cancel{D^2}} Z \sum_{m=1}^{M} \left[\frac{V_\infty}{V} \frac{\Gamma}{V \pi D} \cos(\beta_i) - \frac{1}{2\pi} \left(\frac{V_\infty}{V} \right)^2 \frac{c}{D}\, C_{Dv} \sin(\beta_i) \right] \frac{\mathrm{d}r}{\cancel{2}R} \\ &= 4\, Z \sum_{m=1}^{M} \left[\frac{V_\infty}{V} \frac{\Gamma}{V \pi D} \cos(\beta_i) - \frac{1}{2\pi} \left(\frac{V_\infty}{V} \right)^2 \frac{c}{D}\, C_{Dv} \sin(\beta_i) \right] \Delta y_v \end{aligned} \tag{B.5}$$

$$\begin{aligned} C_Q &= \frac{16}{\pi \cancel{\rho} V^2 D^3} \cancel{\rho} Z \sum_{m=1}^{M} \left[V_\infty \Gamma \sin(\beta_i) + \frac{1}{2} V_\infty^2 c\, C_{Dv} \cos(\beta_i) \right] r \mathrm{d}r \\ &= \frac{4\,\cancel{16}}{\cancel{\pi} \cancel{V^2}\, \cancel{D^3}} Z \sum_{m=1}^{M} \left[\frac{V_\infty}{V} \frac{\Gamma}{V \pi D} \sin(\beta_i) + \frac{1}{2\pi} \left(\frac{V_\infty}{V} \right)^2 \frac{c}{D}\, C_{Dv} \cos(\beta_i) \right] \frac{r}{\cancel{2}R} \frac{\mathrm{d}r}{\cancel{2}R} \\ &= 4\, Z \sum_{m=1}^{M} \left[\frac{V_\infty}{V} \frac{\Gamma}{V \pi D} \sin(\beta_i) + \frac{1}{2\pi} \left(\frac{V_\infty}{V} \right)^2 \frac{c}{D}\, C_{Dv} \cos(\beta_i) \right] y_c\, \Delta y_v \end{aligned} \tag{B.6}$$

Similarly, the side force contribution from one blade can be found by decomposing the tangential force in horisontal and vertical direction.

$$\begin{aligned}
C_{fy} &= \frac{-8\cos\theta}{\pi\cancel{\rho}V^2D^2}\cancel{\rho}\sum_{m=1}^{M}\left[V_\infty\Gamma\sin(\beta_i)+\frac{1}{2}V_\infty^2 c\,C_{Dv}\cos(\beta_i)\right]\mathrm{d}r \\
&= \frac{-4\cos\theta\,\cancel{8}}{\cancel{\pi}\cancel{V^2}\cancel{D^2}}\sum_{m=1}^{M}\left[\frac{V_\infty}{V}\frac{\Gamma}{V\pi D}\sin(\beta_i)+\frac{1}{2\pi}\left(\frac{V_\infty}{V}\right)^2\frac{c}{D}\,C_{Dv}\cos(\beta_i)\right]\frac{\mathrm{d}r}{\cancel{2}R} \qquad \text{(B.7)} \\
&= \quad -4\cos\theta\sum_{m=1}^{M}\left[\frac{V_\infty}{V}\frac{\Gamma}{V\pi D}\sin(\beta_i)+\frac{1}{2\pi}\left(\frac{V_\infty}{V}\right)^2\frac{c}{D}\,C_{Dv}\cos(\beta_i)\right]\Delta y_v
\end{aligned}$$

Hence the vertical component is:

$$C_{fz} \;=\; -4\sin\theta\sum_{m=1}^{M}\left[\frac{V_\infty}{V}\frac{\Gamma}{V\pi D}\sin(\beta_i)+\frac{1}{2\pi}\left(\frac{V_\infty}{V}\right)^2\frac{c}{D}\,C_{Dv}\cos(\beta_i)\right]\Delta y_v \qquad \text{(B.8)}$$

The bending moment contribution from one blade can be found by multiplying the thrust force with the horisontal and vertical distance to where the thrust force attacks, i.e. the control points.

$$\begin{aligned}
C_{my} &= \frac{-16\cos\theta}{\pi\cancel{\rho}V^2D^3}\cancel{\rho}\sum_{m=1}^{M}\left[V_\infty\Gamma\cos(\beta_i)-\frac{1}{2}V_\infty^2 c\,C_{Dv}\sin(\beta_i)\right]r\mathrm{d}r \\
&= \frac{-4\cos\theta\,\cancel{16}}{\cancel{\pi}\cancel{V^2}\cancel{D^3}}\sum_{m=1}^{M}\left[\frac{V_\infty}{V}\frac{\Gamma}{V\pi D}\cos(\beta_i)-\frac{1}{2\pi}\left(\frac{V_\infty}{V}\right)^2\frac{c}{D}\,C_{Dv}\sin(\beta_i)\right]\frac{r}{\cancel{2}R}\frac{\mathrm{d}r}{\cancel{2}R} \qquad \text{(B.9)} \\
&= \quad -4\cos\theta\sum_{m=1}^{M}\left[\frac{V_\infty}{V}\frac{\Gamma}{V\pi D}\cos(\beta_i)-\frac{1}{2\pi}\left(\frac{V_\infty}{V}\right)^2\frac{c}{D}\,C_{Dv}\sin(\beta_i)\right]y_c\,\Delta y_v
\end{aligned}$$

Hence the vertical component is:

$$C_{mz} \;=\; -4\sin\theta\sum_{m=1}^{M}\left[\frac{V_\infty}{V}\frac{\Gamma}{V\pi D}\sin(\beta_i)-\frac{1}{2\pi}\left(\frac{V_\infty}{V}\right)^2\frac{c}{D}\,C_{Dv}\cos(\beta_i)\right]y_c\,\Delta y_v \qquad \text{(B.10)}$$

C Setup of validation experiments

C1 Propeller model geometry

C1.1 P1374

Geometry of propeller P1374:

Propeller diameter	250 mm
Hub diameter	60 mm
Design pitch ratio P/D	1.10
Skew	25 deg
EAR	0.60
Number of blades	4

Table C.1: Main specifications of propeller P1374.

r/R	c/D	P/D	f_0/D *10^3
0.240	0.13340	1.07800	1.200
0.250	0.14135	1.07895	3.685
0.300	0.17995	1.08335	7.050
0.350	0.21635	1.08725	8.750
0.400	0.25040	1.09065	9.815
0.500	0.31040	1.09585	10.800
0.600	0.35755	1.09895	10.715
0.700	0.38790	1.10000	9.880
0.800	0.39395	1.08820	8.450
0.900	0.35640	1.03765	6.250
0.950	0.30020	0.99350	4.530
0.975	0.24515	0.96610	3.245
0.990	0.18370	0.94790	2.070
1.000	0.02500	0.93500	0.000

Table C.2: Blade section geometry of propeller P1374.

C1.2 KVLCC2

Geometry of propeller KVLCC2:

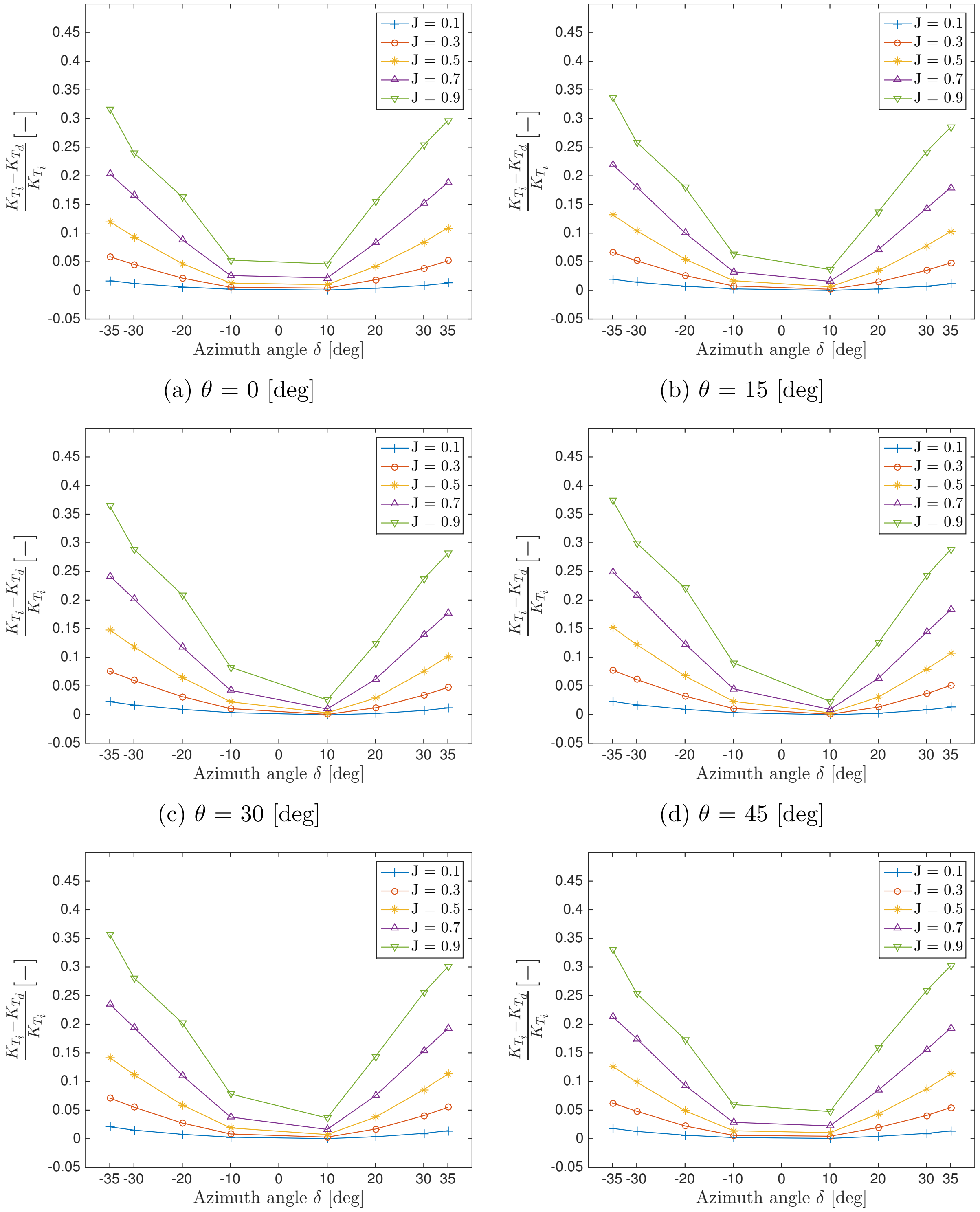
Propeller diameter	9.86 m
Hub diameter	0.31 m
Design pitch ratio P/D	1.10
EAR	0.431
Number of blades	4

Table C.3: Main specifications of propeller KVLCC2.

r/R	c/D	P/D	f_0/D *10^3
0.155	0.14992	0.57430	4.650
0.160	0.15150	0.57650	4.740
0.250	0.17720	0.61300	6.185
0.300	0.18920	0.63100	6.735
0.400	0.20930	0.66300	7.075
0.500	0.22470	0.69150	6.585
0.600	0.23350	0.71200	5.840
0.700	0.23380	0.72120	5.120
0.800	0.21920	0.71600	4.340
0.900	0.18080	0.69270	2.910
0.950	0.14220	0.67480	1.820
1.000	0.01000	0.65100	0.030

Table C.4: Blade section geometry of propeller KVLCC2.

D Investigation of skewed wake effect



(a) $\theta = 0$ [deg]

(b) $\theta = 15$ [deg]

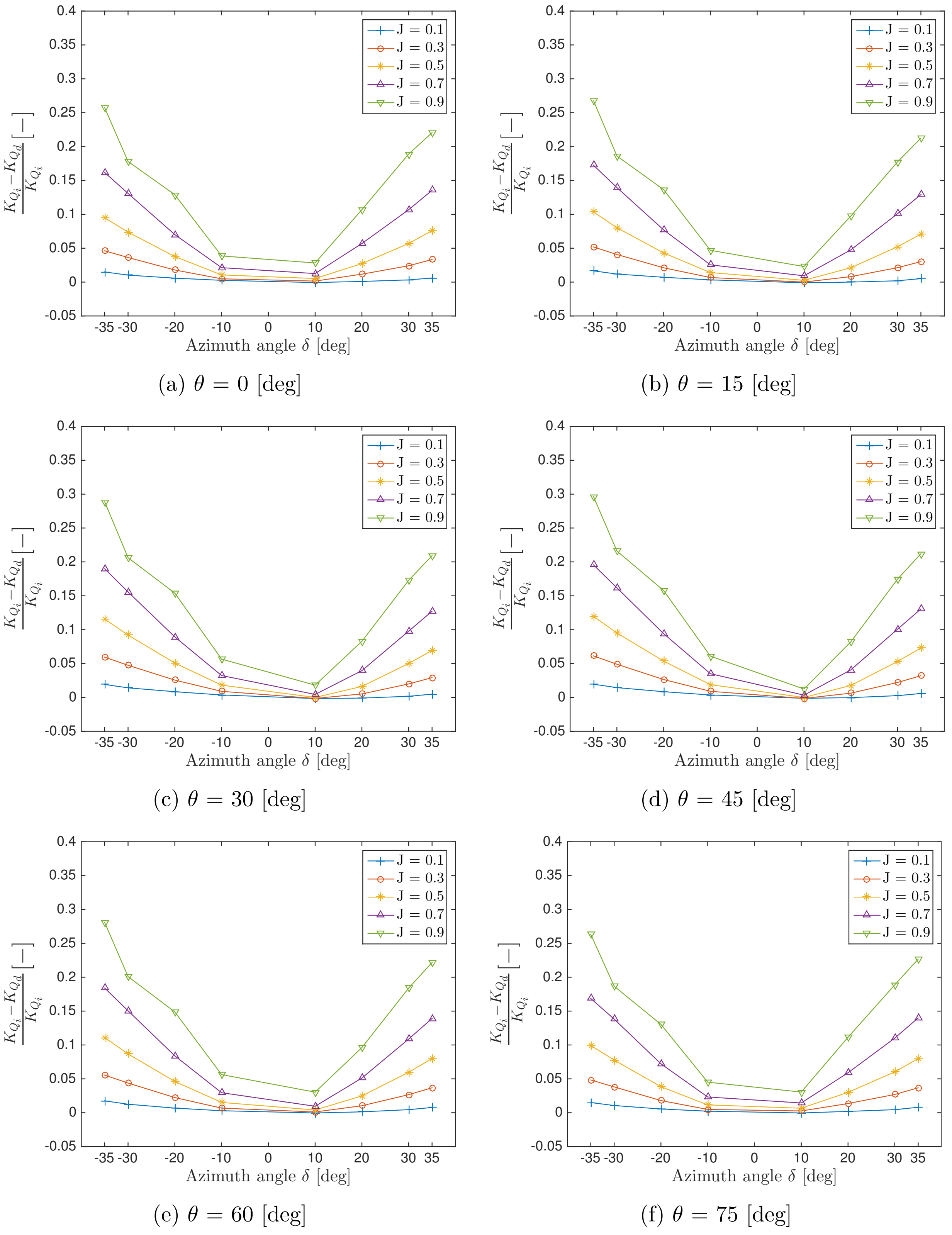
(c) $\theta = 30$ [deg]

(d) $\theta = 45$ [deg]

(e) $\theta = 60$ [deg]

(f) $\theta = 75$ [deg]

Figure D.1: K_T comparison between decomposed wakefield (with no radial flow) due to oblique inflow (K_{T_d}) and AKPA azimuth angle built-in calculation (K_{T_i}) for four bladed propeller at six different angular blade positions θ [deg]. Presented as relative difference with respect to K_{T_i}.



(a) $\theta = 0$ [deg]

(b) $\theta = 15$ [deg]

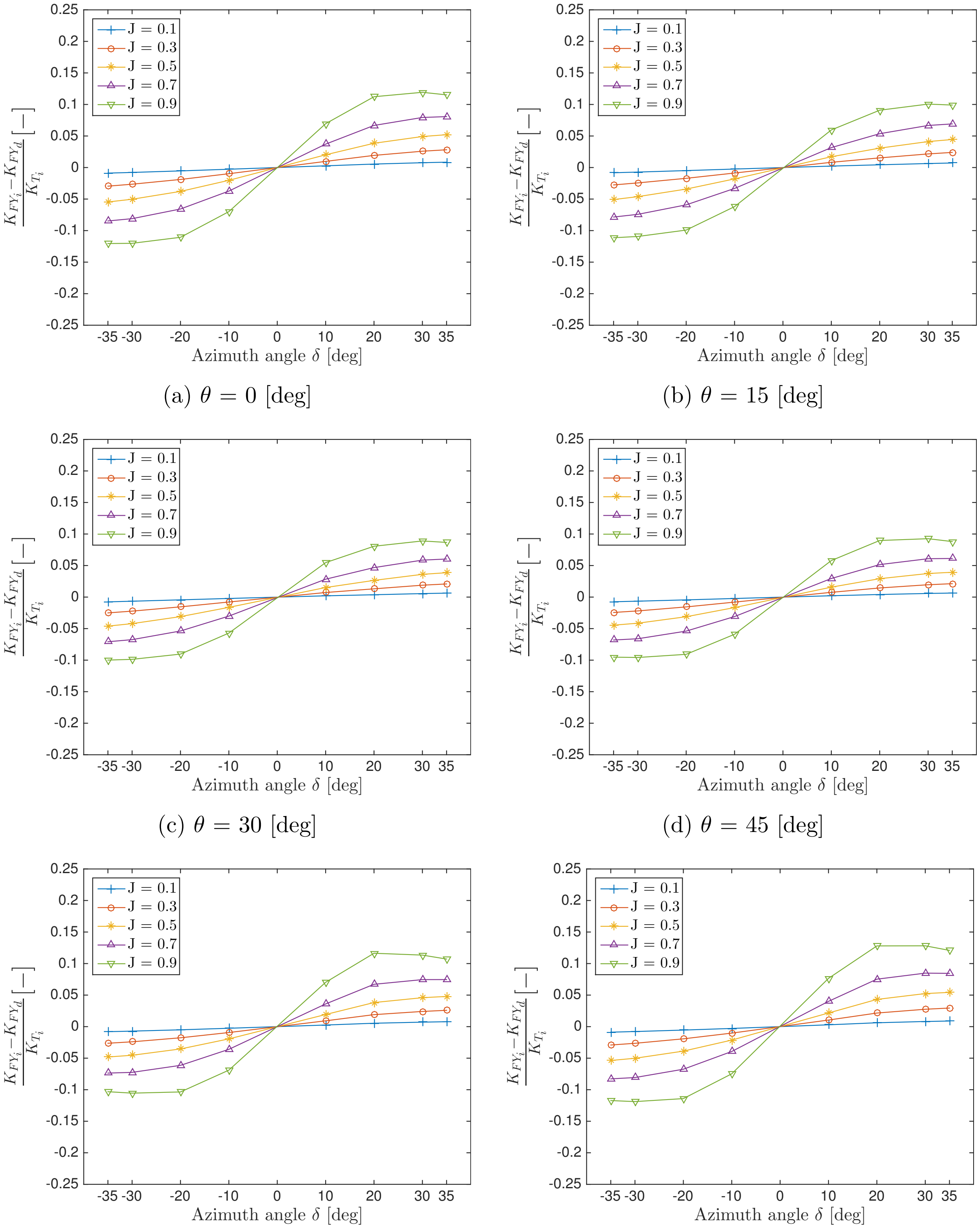
(c) $\theta = 30$ [deg]

(d) $\theta = 45$ [deg]

(e) $\theta = 60$ [deg]

(f) $\theta = 75$ [deg]

Figure D.2: K_Q comparison between decomposed wakefield (with no radial flow) due to oblique inflow (K_{Q_d}) and AKPA azimuth angle built-in calculation (K_{Q_i}) for four bladed propeller at six different angular blade positions θ [deg]. Presented as relative difference with respect to K_{Q_i}.



(a) θ = 0 [deg]

(b) θ = 15 [deg]

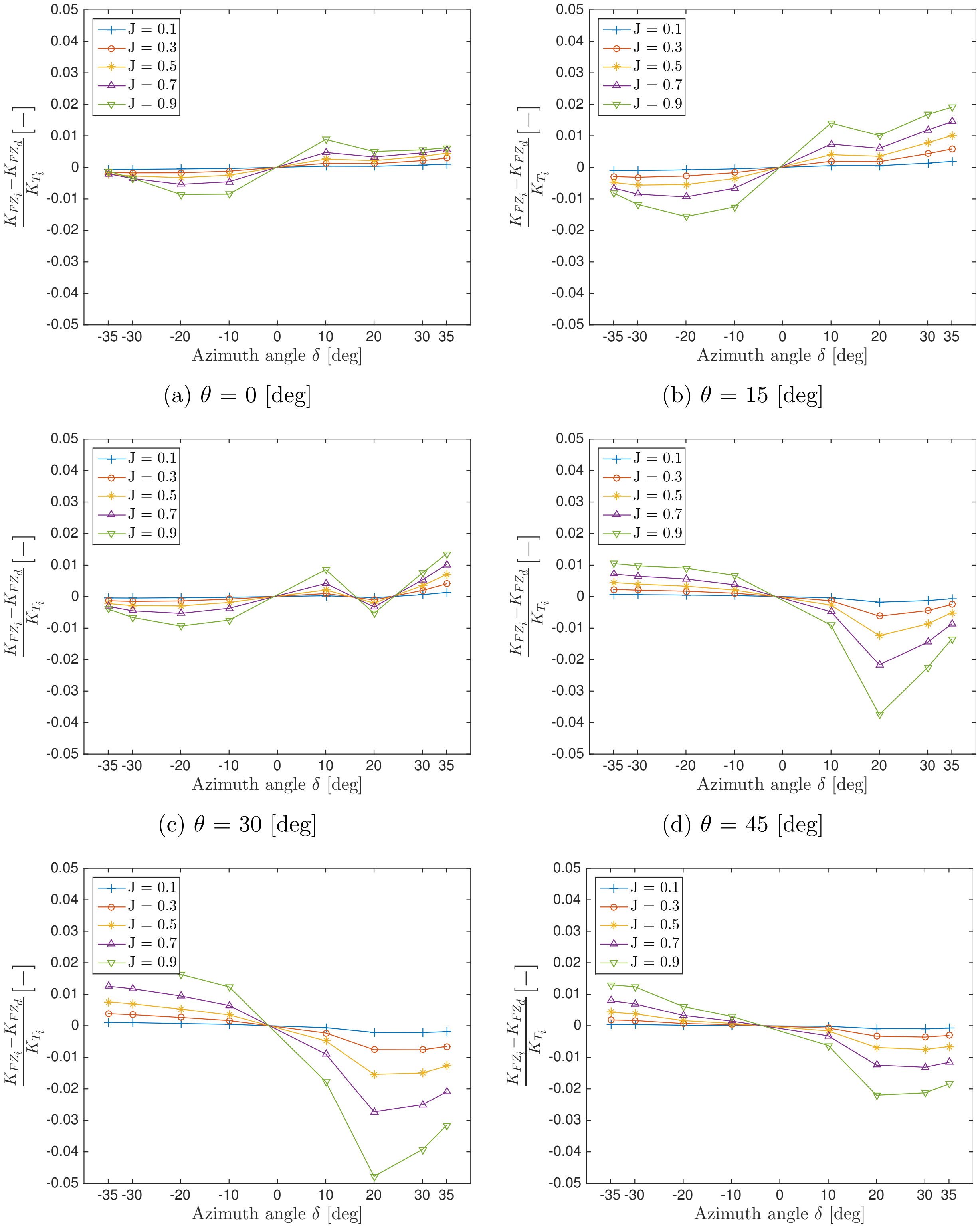
(c) θ = 30 [deg]

(d) θ = 45 [deg]

(e) θ = 60 [deg]

(f) θ = 75 [deg]

Figure D.3: K_{FY} comparison between decomposed wakefield (with no radial flow) due to oblique inflow (K_{FY_d}) and AKPA azimuth angle built-in calculation (K_{FY_i}) for four bladed propeller at six different angular blade positions θ [deg]. Presented as relative difference with respect to K_{T_i}.



(a) θ = 0 [deg]

(b) θ = 15 [deg]

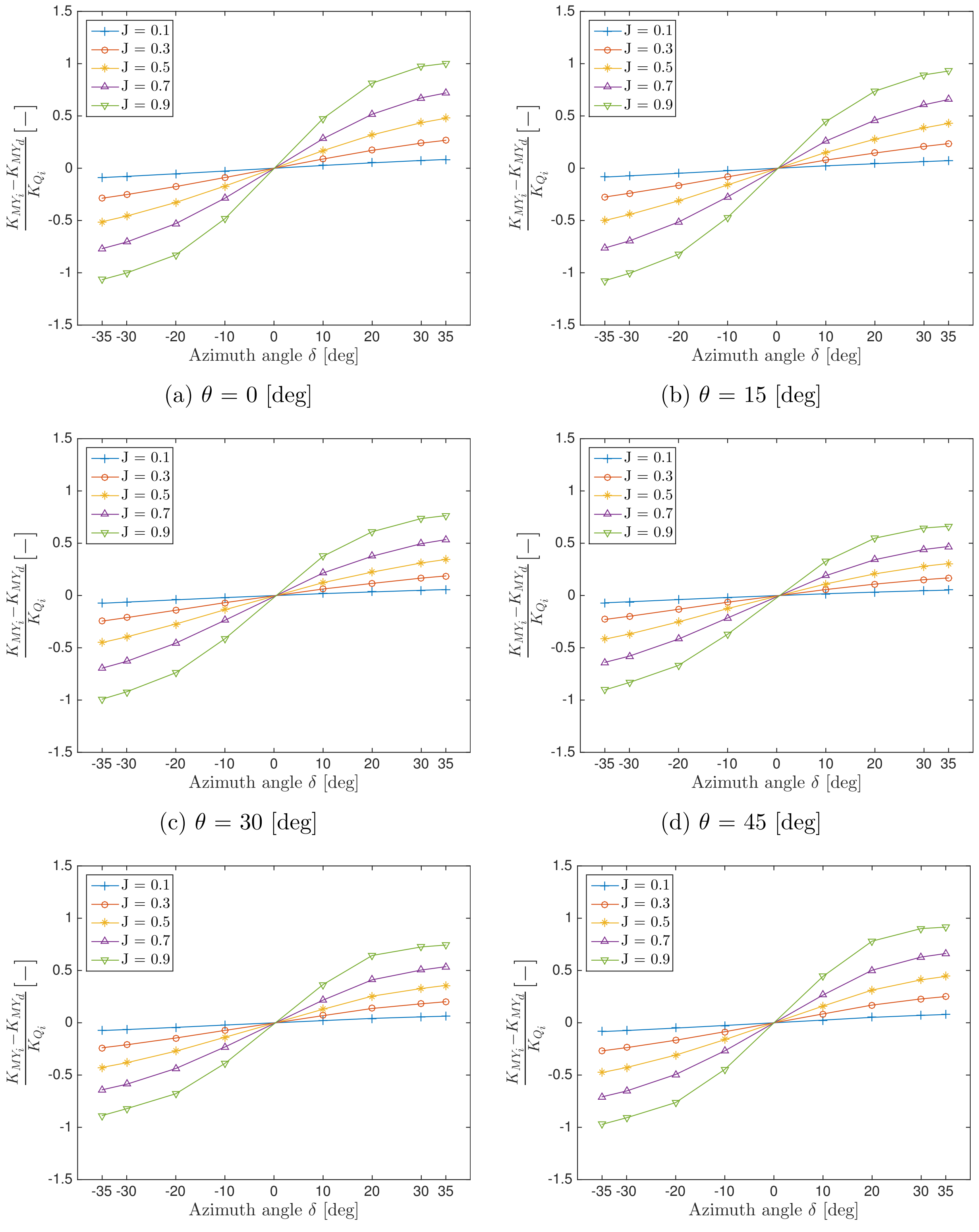
(c) θ = 30 [deg]

(d) θ = 45 [deg]

(e) θ = 60 [deg]

(f) θ = 75 [deg]

Figure D.4: K_{FZ} comparison between decomposed wakefield (with no radial flow) due to oblique inflow (K_{FZ_d}) and AKPA azimuth angle built-in calculation (K_{FZ_i}) for four bladed propeller at six different angular blade positions θ [deg]. Presented as relative difference with respect to K_{T_i}.



(a) θ = 0 [deg]

(b) θ = 15 [deg]

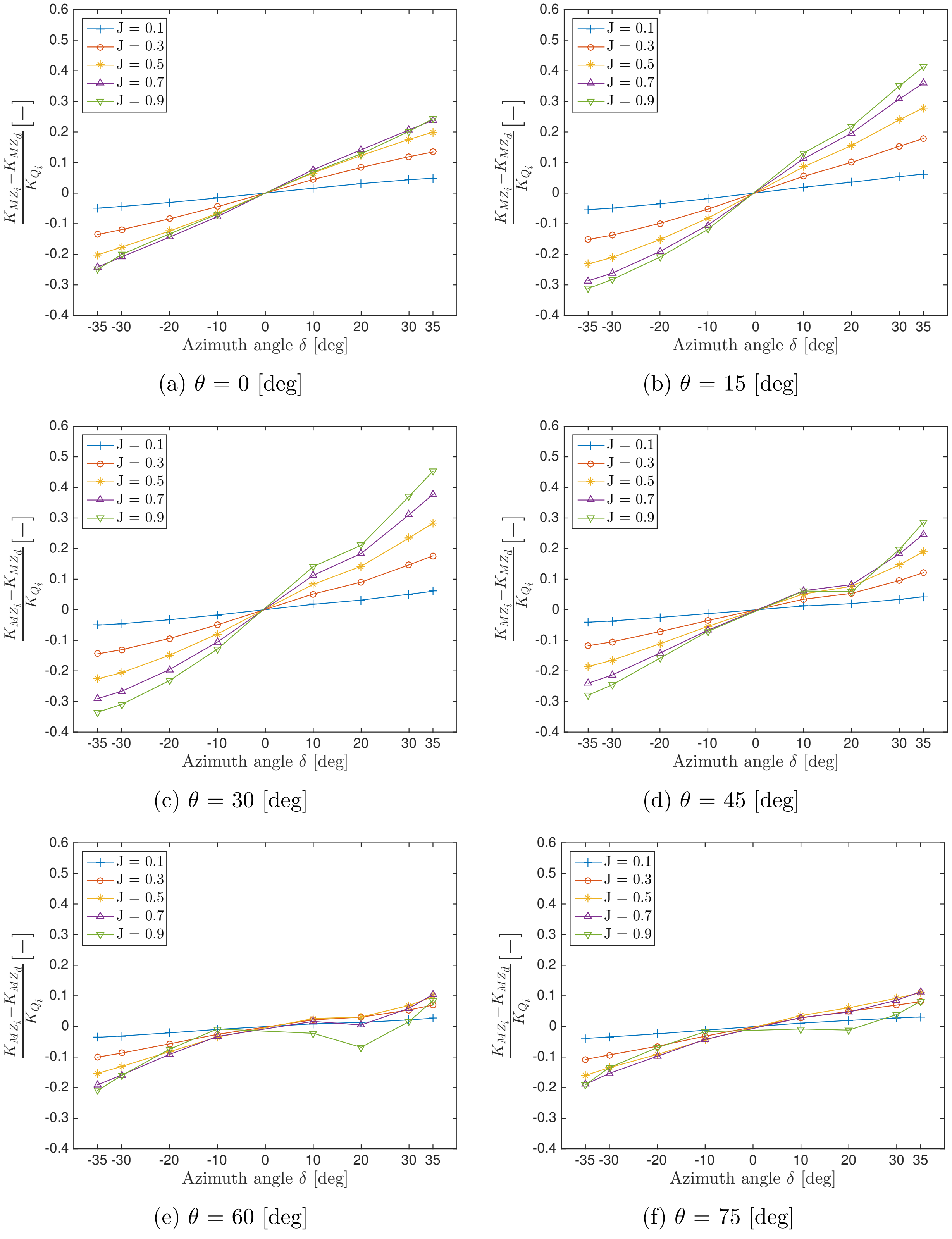
(c) θ = 30 [deg]

(d) θ = 45 [deg]

(e) θ = 60 [deg]

(f) θ = 75 [deg]

Figure D.5: K_{MY} comparison between decomposed wakefield (with no radial flow) due to oblique inflow (K_{MY_d}) and AKPA azimuth angle built-in calculation (K_{MY_i}) for four bladed propeller at six different angular blade positions θ [deg]. Presented as relative difference with respect to K_{Q_i}.



(a) $\theta = 0$ [deg]

(b) $\theta = 15$ [deg]

(c) $\theta = 30$ [deg]

(d) $\theta = 45$ [deg]

(e) $\theta = 60$ [deg]

(f) $\theta = 75$ [deg]

Figure D.6: K_{MZ} comparison between decomposed wakefield (with no radial flow) due to oblique inflow (K_{MZ_d}) and AKPA azimuth angle built-in calculation (K_{MZ_i}) for four bladed propeller at six different angular blade positions θ [deg]. Presented as relative difference with respect to K_{Q_i}.

E Input files

E1 Lifting Line properties

```
/* Lifting Line properties */
  16         | Number of blade vortex panels
   0.25      | Hub vortex radius/Hub radius
  10         | Maximum number of iterations in wake alignment
   1         | Swirl cancellation factor: 1=no cancellation
1000         | Density of water [kg/m^3]
   0.0000001 | Circulation accuracy tolerance
```

E2 Propeller

```
/* Propeller characteristics */
D      EAR    Z  Xh     Rotation Init.Pos Wagner
9.86   0.431  4  0.155  RIGHT    0        15
=============================================
/* Blade section geometry */
r_R   c_D     tmax_D   P_D     f0_D     cD
0.155 0.14992 0.046995 0.57430 0.004650 0.0092
0.160 0.15150 0.046810 0.57650 0.004740 0.0091
0.250 0.17720 0.042170 0.61300 0.006185 0.0077
0.300 0.18920 0.038520 0.63100 0.006735 0.0071
0.400 0.20930 0.032020 0.66300 0.007075 0.0062
0.500 0.22470 0.026020 0.69150 0.006585 0.0056
0.600 0.23350 0.020550 0.71200 0.005840 0.0052
0.700 0.23380 0.015600 0.72120 0.005120 0.0049
0.800 0.21920 0.011050 0.71600 0.004340 0.0047
0.900 0.18080 0.007000 0.69270 0.002910 0.0046
0.950 0.14220 0.004720 0.67480 0.001820 0.0047
1.000 0.01000 0.000000 0.65100 0.000030 0.0049
=============================================
/* Camber profile */
1.0        | Lift coefficient for chosen camber profile (Cl)
6.651      | Max camber relative to chord length (f0max_c) [%]
1.40       | Ideal angle of attack [deg]
```

E3 Wakefield

```
/* Tangential wakefield */
T\R 0.05265  0.10530 0.15796 0.21061 0.26326 0.31591 0.36856 0.42122 0.47387 0.52652 0.57917 0.63183 0.68448 0.73713 0.78978 0.84243 0.89509 0.94774 1.00039
0   -0.0035  0.0095  0.0046 -0.0003 -0.0051 -0.0086 -0.0121 -0.0156 -0.0191 -0.0172 -0.0123 -0.0073 -0.0023  0.0032  0.0115  0.0197  0.0278  0.0360  0.03940
15  -0.0384 -0.0261 -0.0136 -0.0051  0.0072  0.0096  0.0115  0.0134  0.0141  0.0134  0.0090  0.0046  0.0019 -0.0018 -0.0046 -0.0068 -0.0093 -0.0098 -0.01000
30  -0.0766 -0.0252 -0.0166 -0.0071  0.0006  0.0023 -0.0032 -0.0142 -0.0248 -0.0302 -0.0280 -0.0174 -0.0068  0.0038  0.0144  0.0252  0.0405  0.0191  0.02230
45  -0.1073 -0.0737 -0.0563 -0.0449 -0.0388 -0.0326 -0.0237 -0.0191 -0.0145 -0.0097 -0.0043  0.0011  0.0095  0.0226  0.0356  0.0482  0.0580  0.0671  0.07580
60  -0.1283 -0.1007 -0.0903 -0.0836 -0.0768 -0.0631 -0.0497 -0.0386 -0.0344 -0.0300 -0.0128  0.0119  0.0363  0.0572  0.0767  0.0904  0.0991  0.1081  0.11540
75  -0.1385 -0.1283 -0.1263 -0.1238 -0.1162 -0.0978 -0.0738 -0.0611 -0.0474 -0.0225  0.0041  0.0319  0.0591  0.0851  0.1019  0.1195  0.1360  0.1523  0.16170
90  -0.1385 -0.1442 -0.1495 -0.1564 -0.1448 -0.1226 -0.0991 -0.0771 -0.0534 -0.0240  0.0047  0.0333  0.0623  0.0861  0.1081  0.1304  0.1522  0.1730  0.18390
105 -0.1300 -0.1477 -0.1574 -0.1622 -0.1560 -0.1328 -0.1089 -0.0852 -0.0605 -0.0315 -0.0041  0.0221  0.0476  0.0711  0.0950  0.1188  0.1419  0.1650  0.18410
120 -0.1138 -0.1400 -0.1503 -0.1544 -0.1561 -0.1396 -0.1163 -0.0917 -0.0674 -0.0429 -0.0189  0.0019  0.0222  0.0423  0.0606  0.0796  0.1004  0.1217  0.14290
135 -0.0911 -0.1204 -0.1306 -0.1384 -0.1391 -0.1335 -0.1185 -0.0999 -0.0805 -0.0610 -0.0414 -0.0208 -0.0018  0.0145  0.0286  0.0427  0.0566  0.0683  0.08500
150 -0.0388 -0.0896 -0.1056 -0.1091 -0.1103 -0.1067 -0.1001 -0.0940 -0.0849 -0.0720 -0.0592 -0.0453 -0.0315 -0.0173 -0.0026  0.0155  0.0363  0.0568  0.07370
165 -0.0404 -0.0509 -0.0657 -0.0729 -0.0738 -0.0711 -0.0692 -0.0654 -0.0615 -0.0573 -0.0520 -0.0469 -0.0404 -0.0335 -0.0234 -0.0130 -0.0011  0.0107  0.02380
180  0.0145  0.0095  0.0046 -0.0003 -0.0051 -0.0086 -0.0121 -0.0156 -0.0191 -0.0172 -0.0123 -0.0073 -0.0023  0.0032  0.0115  0.0197  0.0278  0.0360  0.03940
195 -0.0404 -0.0261 -0.0136 -0.0051  0.0072  0.0096  0.0115  0.0134  0.0141  0.0134  0.0090  0.0046  0.0019 -0.0018 -0.0046 -0.0068 -0.0093 -0.0098 -0.01000
210 -0.0388 -0.0252 -0.0166 -0.0071  0.0006  0.0023 -0.0032 -0.0142 -0.0248 -0.0302 -0.0280 -0.0174 -0.0068  0.0038  0.0144  0.0252  0.0405  0.0191  0.02230
225 -0.0911 -0.0737 -0.0563 -0.0449 -0.0388 -0.0326 -0.0237 -0.0191 -0.0145 -0.0097 -0.0043  0.0011  0.0095  0.0226  0.0356  0.0482  0.0580  0.0671  0.07580
240 -0.1138 -0.1007 -0.0903 -0.0836 -0.0768 -0.0631 -0.0497 -0.0386 -0.0344 -0.0300 -0.0128  0.0119  0.0363  0.0572  0.0767  0.0904  0.0991  0.1081  0.11540
255 -0.1300 -0.1283 -0.1263 -0.1238 -0.1162 -0.0978 -0.0738 -0.0611 -0.0474 -0.0225  0.0041  0.0319  0.0591  0.0851  0.1019  0.1195  0.1360  0.1523  0.16170
270 -0.1385 -0.1442 -0.1495 -0.1564 -0.1448 -0.1226 -0.0991 -0.0771 -0.0534 -0.0240  0.0047  0.0333  0.0623  0.0861  0.1081  0.1304  0.1522  0.1730  0.18390
285 -0.1385 -0.1477 -0.1574 -0.1622 -0.1560 -0.1328 -0.1089 -0.0852 -0.0605 -0.0315 -0.0041  0.0221  0.0476  0.0711  0.0950  0.1188  0.1419  0.1650  0.18410
300 -0.1283 -0.1400 -0.1503 -0.1544 -0.1561 -0.1396 -0.1163 -0.0917 -0.0674 -0.0429 -0.0189  0.0019  0.0222  0.0423  0.0606  0.0796  0.1004  0.1217  0.14290
315 -0.1073 -0.1204 -0.1306 -0.1384 -0.1391 -0.1335 -0.1185 -0.0999 -0.0805 -0.0610 -0.0414 -0.0208 -0.0018  0.0145  0.0286  0.0427  0.0566  0.0683  0.08500
330 -0.0766 -0.0896 -0.1056 -0.1091 -0.1103 -0.1067 -0.1001 -0.0940 -0.0849 -0.0720 -0.0592 -0.0453 -0.0315 -0.0173 -0.0026  0.0155  0.0363  0.0568  0.07370
345 -0.0384 -0.0509 -0.0657 -0.0729 -0.0738 -0.0711 -0.0692 -0.0654 -0.0615 -0.0573 -0.0520 -0.0469 -0.0404 -0.0335 -0.0234 -0.0130 -0.0011  0.0107  0.02380
360 -0.0035  0.0061  0.0158  0.0254  0.0264  0.0271  0.0277  0.0283  0.0287  0.0288  0.0290  0.0291  0.0292  0.0305  0.0319  0.0332  0.0345  0.0359  0.03750

/* Axial wakefield */
T\R 0.05265 0.10530 0.15796 0.21061 0.26326 0.31591 0.36856 0.42122 0.47387 0.52652 0.57917 0.63183 0.68448 0.73713 0.78978 0.84243 0.89509 0.94774 1.00039
0   0.34739 0.37490 0.40443 0.43065 0.43608 0.44151 0.44556 0.44905 0.45098 0.45108 0.45071 0.44859 0.44657 0.4427  0.43884 0.43341 0.42752 0.41989 0.41069
15  0.34721 0.36644 0.38355 0.39532 0.39302 0.39146 0.38723 0.38401 0.36533 0.34905 0.33893 0.32439 0.31492 0.30774 0.31501 0.32467 0.34721 0.37186 0.40048
30  0.34252 0.35199 0.36211 0.36082 0.33718 0.30314 0.26726 0.22752 0.22770 0.23819 0.25723 0.27195 0.31841 0.37113 0.42881 0.50039 0.56911 0.62891 0.67537
45  0.33718 0.34344 0.33994 0.31004 0.27039 0.22982 0.24720 0.26533 0.29891 0.38530 0.46239 0.54078 0.62330 0.70573 0.76535 0.78485 0.80629 0.81659 0.81871
60  0.32918 0.33157 0.31188 0.27444 0.24573 0.26671 0.28640 0.38281 0.49165 0.59717 0.69304 0.78246 0.80666 0.83297 0.84079 0.84511 0.84806 0.84750 0.84732
75  0.32218 0.32228 0.29642 0.26165 0.27048 0.29845 0.39026 0.52293 0.63618 0.73582 0.81328 0.83619 0.85072 0.85413 0.85781 0.85845 0.85910 0.85937 0.85974
90  0.31611 0.31464 0.28879 0.26202 0.28695 0.32053 0.44666 0.59809 0.70389 0.79626 0.83674 0.85174 0.85937 0.86204 0.86360 0.86416 0.86471 0.86498 0.86563
105 0.31197 0.30489 0.28216 0.26514 0.29274 0.34104 0.44933 0.60564 0.72248 0.79930 0.84944 0.85643 0.86241 0.86434 0.86664 0.86719 0.86774 0.86848 0.86912
120 0.30443 0.28796 0.27526 0.25944 0.26744 0.32715 0.39146 0.52578 0.66994 0.77464 0.81935 0.86112 0.86508 0.86535 0.86701 0.86912 0.87060 0.87115 0.87170
135 0.29560 0.27490 0.25052 0.23626 0.24481 0.26137 0.32080 0.42329 0.52090 0.65127 0.75422 0.82883 0.84806 0.86195 0.86977 0.87032 0.86940 0.87032 0.87262
150 0.29127 0.25944 0.23028 0.20268 0.19274 0.20157 0.24233 0.28345 0.35724 0.44666 0.54906 0.66461 0.74630 0.79838 0.83435 0.86186 0.86830 0.87106 0.87299
165 0.28695 0.25456 0.21795 0.18234 0.14849 0.16155 0.17977 0.19237 0.22080 0.26570 0.34739 0.43295 0.51925 0.60186 0.67381 0.74106 0.78908 0.82414 0.86167
180 0.28584 0.24766 0.20921 0.17038 0.13579 0.16523 0.19844 0.22678 0.25512 0.29054 0.32973 0.36818 0.40554 0.44767 0.52366 0.60922 0.68190 0.75458 0.79184
195 0.28695 0.25456 0.21795 0.18234 0.14849 0.16155 0.17977 0.19237 0.22080 0.26570 0.34739 0.43295 0.51925 0.60186 0.67381 0.74106 0.78908 0.82414 0.86167
210 0.29127 0.25944 0.23028 0.20268 0.19274 0.20157 0.24233 0.28345 0.35724 0.44666 0.54906 0.66461 0.74630 0.79838 0.83435 0.86186 0.86830 0.87106 0.87299
225 0.29560 0.27490 0.25052 0.23626 0.24481 0.26137 0.32080 0.42329 0.52090 0.65127 0.75422 0.82883 0.84806 0.86195 0.86977 0.87032 0.86940 0.87032 0.87262
240 0.30443 0.28796 0.27526 0.25944 0.26744 0.32715 0.39146 0.52578 0.66994 0.77464 0.81935 0.86112 0.86508 0.86535 0.86701 0.86912 0.87060 0.87115 0.87170
255 0.31197 0.30489 0.28216 0.26514 0.29274 0.34104 0.44933 0.60564 0.72248 0.79930 0.84944 0.85643 0.86241 0.86434 0.86664 0.86719 0.86774 0.86848 0.86912
270 0.31611 0.31464 0.28879 0.26202 0.28695 0.32053 0.44666 0.59809 0.70389 0.79626 0.83674 0.85174 0.85937 0.86204 0.86360 0.86416 0.86471 0.86498 0.86563
285 0.32218 0.32228 0.29642 0.26165 0.27048 0.29845 0.39026 0.52293 0.63618 0.73582 0.81328 0.83619 0.85072 0.85413 0.85781 0.85845 0.85910 0.85937 0.85974
300 0.32918 0.33157 0.31188 0.27444 0.24573 0.26671 0.28640 0.38281 0.49165 0.59717 0.69304 0.78246 0.80666 0.83297 0.84079 0.84511 0.84806 0.84750 0.84732
315 0.33718 0.34344 0.33994 0.31004 0.27039 0.22982 0.24720 0.26533 0.29891 0.38530 0.46239 0.54078 0.62330 0.70573 0.76535 0.78485 0.80629 0.81659 0.81871
330 0.34252 0.35199 0.36211 0.36082 0.33718 0.30314 0.26726 0.22752 0.22770 0.23819 0.25723 0.27195 0.31841 0.37113 0.42881 0.50039 0.56911 0.62891 0.67537
345 0.34721 0.36644 0.38355 0.39532 0.39302 0.39146 0.38723 0.38401 0.36533 0.34905 0.33893 0.32439 0.31492 0.30774 0.31501 0.32467 0.34721 0.37186 0.40048
360 0.34739 0.3749  0.40443 0.43065 0.43608 0.44151 0.44556 0.44905 0.45098 0.45108 0.45071 0.44859 0.44657 0.4427  0.43884 0.43341 0.42752 0.41989 0.41069
```

F PropSiM source code

The simulation model is built on multiple functions that handles the different hydrodynamic effects. The main function ”PropSiM” controls the sequence of the calculations, ensuring that the all effects are included both when they should and in the right order.

This appendix contains the source code of the main function, along with the accompanying subroutines.

The Simulink model block and the subroutines are published with MATLAB® R2015a.

F1 PropSiM

```
function [T, Q, fy, fz, my, mz] = PropSiM(TIME,n_RPM,V,CPANGLE,h,dhdt,DELTA,RHO,...
    INIT,NBLADES,NITER,NPANELS,ALPHAi,ZCi,CLi,WTP,AXWYC,AXWYV,TAWYC,TAWYV,BPAC,DIAM,CDYC,...
    ZCYC,CRP,DRAGYC,DTHETA,EAR,HUBVRAD,ROT,TOL,XHUB,YC,YV,PANELSIZE)
%*****************************************************************************************
% ROUTINE:  PropSiM
% SUBCALL:  circulation.m
%           wagnereffect.m
% ----------------------------------------------------------------------------------------
%
% Object:   Calculate 6 DoF propeller forces.
%
% Method:   Use Lifting Line theory to calculate 6 DoF propeller forces.
%           Include effect from reduced propeller submergence, Wagner effect, ventilation,
%           change in operating point and oblique inflow.
%
% PARAMETERS
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% T             output  float   Thrust [N]
% Q             output  float   Torque [Nm]
% fy            output  float   Horisontal side force [N]
% fz            output  float   Vertical side force [N]
% my            output  float   Horisontal bending moment [Nm]
% mz            output  float   Vertical bending moment [Nm]
% TIME          input   float   Simulation time [s]
% n_RPM         input   float   Shaft speed [RPM]
% V             input   float   Forward speed of vessel [m/s]
% CPANGLE       input   float   Controllable pitch angle at blade root [deg]
% h             input   float   Propeller submergence [m]
% dhdt          input   float   Rate of change of propeller submergence [m]
% NWPANELS      input   float   Number of wet vortex panels
% BLADE         input   float   Blade number
% BLADEPOS      input   float   Circumferential position of blades
% DELTA         input   float   Azimuth angle [deg]
% RHO           input   float   Density of water
% INIT          input   float   Initial position of propeller blade at simulation start
% NBLADES       input   float   Number of propeller blades
% NITER         input   float   Maximum number of iterations in wake alignment
% NPANELS       input   float   Number of cosine spaced vortex panels
% ALPHAi        input   float   Ideal angle of attack of NACA section
% ZCi           input   float   Maximum camber of NACA section
% ZCYC          inout   float   Maximum camber at control points
% CLi           input   float   Lift coefficient at ideal angle of attack for NACA section
% WTP           input   float   Angular position of wake input
% AXWYC         input   float   Interpolated axial wake field at control points
% AXWYV         input   float   Interpolated axial wake field at vortex points
% TAWYC         input   float   Interpolated tangential wake field at control points
% TAWYV         input   float   Interpolated tangential wake field at vortex points
% BPAC          input   float   Interpolated blade pitch angle at control points
% DIAM          input   float   Propeller diameter
% CDYC          input   float   Interpolated chord length at control points
% CRP           input   float   Swirl cancellation factor
% DRAGYC        input   float   Interpolated drag coefficient at control points
% DTHETA        input   float   Wagner density angle
% EAR           input   float   Expanded blade Area Ratio
% HUBVRAD       input   float   Relative radius of hub vortex
% ROT           input   float   Direction of rotation for propeller
% TOL           input   float   Relative tolerance of circulation accuracy
% XHUB          input   float   Relative hub radius
% YC            input   float   Coordinates of control points
% YV            inout   float   Coordinates of vortex points
% PANELSIZE     input   float   Length of the vortex panels
% ----------------------------------------------------------------------------------------
% INTERNAL VARIABLES
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% BLADEPOS              float   Circumferential position of blades
% CT                    float   Non-dimensional thrust
% CQ                    float   Non-dimensional torque
% CFY                   float   Non-dimensional horisontal side force
% CFZ                   float   Non-dimensional vertical side force
% CMY                   float   Non-dimensional horisontal bending moment
% CMZ                   float   Non-dimensional vertical bending moment
% CTH                   float   Hub vortex drag coefficient
% VENTFACTOR            float   Thrust loss due to ventilation
% AVAO                  float   Ventilated area/disk area
% ANVAO                 float   Non-ventilated area/disk area
% k                     float   Relation between thrust and torque loss
% g                     float   Gravitational acceleration
% n_Hz                  float   Rotational speed [Hz]
% SR                    float   Submergence ratio
% J                     float   Advance number
% dSRdt                 float   Derivative of submergence ratio
% WETSPAN               float   Wetted span interval of blades
% BLADEENT              float   Angle when blade enters the water
% TANBC                 float   tan(beta) at control points
% TANBV                 float   tan(beta) at vortex points
% YCWETVARINDX          float   Index to control points that vary between wet and dry
% YVWETVARINDX          float   Index to vortex points that vary between wet and dry
% YCSUBINDX             float   Index to submerged control points
% YVSUBUNDX             float   Index to submerged vortex points
% NWPANELS              float   Number of wet vortex panels
% VASTAR                float   Resulting axial inflow velocity
% VTSTAR                float   Resulting tangential inflow velocity
% VINF                  float   Velocity seen by propeller blade
% G                     float   Nondimensional circulation
% CHIYC                 float   Wake skew angle at control point
% U0                    float   Mean annular axial induced velocity
% CTH                   float   Hub vortex drag coefficient
% DVISC                 float   Viscous drag force coefficient
% WAGNER                float   beta_factor for Wagner effect at each blade section
% YCWAGNER              float   Index to control points that are affected by Wagner
% YVWAGNER              float   Index to vortex points that are affected by Wagner
% WAGFLAG               float   Flag that inform if this is a Wagner calculation or not
% axwbyc                float   Temporary axial wake field at control point for each blade
% axwbyv                float   Temporary axial wake field at vortex point for each blade
% tawbyc                float   Temporary tangential wake field at control point for each blade
% tawbyv                float   Temporary tangential wake field at vortex point for each blade
% PART                  float   Number of revolutions
% ----------------------------------------------------------------------------------------
% ========================================================================================
% Written by:  Oyvind Oksnes Dalheim
% Last edited: 10.06.15
% ****************************************************************************************

coder.extrinsic('cosspace.m', 'induction.m', 'circulation.m', 'wagnereffect.m');
%% Default settings
ROT = 1;

%% Preallocation
BLADEPOS  = zeros(NBLADES,1);
CT        = zeros(NBLADES,1);
CQ        = zeros(NBLADES,1);
```

```
CFY       = zeros(NBLADES,1);
CFZ       = zeros(NBLADES,1);
CMY       = zeros(NBLADES,1);
CMZ       = zeros(NBLADES,1);
CIRC      = zeros(NBLADES,NPANELS);
ANGLE     = zeros(NBLADES,1);
SPEED     = zeros(NBLADES,1);
persistent VENTFACTOR AVAO ANVAO;
if isempty(VENTFACTOR)
    VENTFACTOR = 1;
    AVAO = 0;
    ANVAO = 1-AVAO;
end

coder.varsize('YCWAGNER',[1 NPANELS],[0 1]);
coder.varsize('YVWETVARINDX',[1 NPANELS+1], [0 1]);
coder.varsize('YCWETVARINDX',[1 NPANELS], [0 1]);

%% Unit conversion
k = 0.84;
g = 9.81;
n_Hz = n_RPM/60;
SR = (2*h)/(DIAM);
dSRdt = (-dhdt)/V;
J = V/(n_Hz*DIAM);
PART = n_Hz*TIME+INIT/360;

%% Finding position of each blade [deg]
for i = 1:NBLADES
    BLADEPOS(i) = 360*(PART+((i-1)/NBLADES)-floor(PART+((i-1)/NBLADES)));
end
%% Finding wetted span interval for all blades
WETSPAN = [ones(NBLADES,1).*XHUB, ones(NBLADES,1)];
if SR<1 && SR >= 0
    BLADEENT = acosd(SR);
    WETSPAN(:,2) = ceil((SR)./(cosd(BLADEPOS-(BLADEPOS-BLADEENT).*logical((BLADEPOS>=BLADEENT)...
        & (BLADEPOS<=(360-BLADEENT))))).*10^6)/10^6;
    WETSPAN(~isfinite(WETSPAN)) = 1;
elseif SR<0 && SR > -1
    BLADEENT = acosd(SR);
    WETSPAN(:,1) = floor((1-(1-SR./cosd(BLADEPOS)).*logical((BLADEPOS>BLADEENT)...
        & (BLADEPOS<(360-BLADEENT)))).*10^6)/10^6;
elseif SR <= -1
    WETSPAN = ones(NBLADES,2);
end

%% Interpolating at each blade position
axwbyc = zeros(NBLADES,NPANELS);
tawbyc = zeros(NBLADES,NPANELS);
axwbyv = zeros(NBLADES,NPANELS+1);
tawbyv = zeros(NBLADES,NPANELS+1);
TANBC = zeros(NBLADES,NPANELS);
TANBV = zeros(NBLADES,NPANELS+1);

for i = 1:NPANELS
    axwbyc(:,i) = interp1(WTP,AXWYC(:,i),BLADEPOS,'spline');
    tawbyc(:,i) = interp1(WTP,TAWYC(:,i),BLADEPOS,'spline');
end
for i = 1:(NPANELS+1)
    axwbyv(:,i) = interp1(WTP,AXWYV(:,i),BLADEPOS,'spline');
    tawbyv(:,i) = interp1(WTP,TAWYV(:,i),BLADEPOS,'spline');
end

%% Decomposing wake field due to azimuth angle
```

```
AXWBYC = axwbyc.*cosd(DELTA) + tawbyc.*sind(DELTA).*cosd(repmat(BLADEPOS,1,NPANELS));
AXWBYV = axwbyv.*cosd(DELTA) + tawbyv.*sind(DELTA).*cosd(repmat(BLADEPOS,1,NPANELS+1));
TAWBYC = tawbyc.*sqrt((cosd(repmat(BLADEPOS,1,NPANELS)).*cosd(DELTA)).^2+...
         (sind(repmat(BLADEPOS,1,NPANELS))).^2)+ axwbyc.*sind(DELTA).*...
         cosd(repmat(BLADEPOS,1,NPANELS))+dSRdt.*sind(repmat(BLADEPOS,1,NPANELS));
TAWBYV = tawbyv.*sqrt((cosd(repmat(BLADEPOS,1,NPANELS+1)).*cosd(DELTA)).^2+...
         (sind(repmat(BLADEPOS,1,NPANELS+1))).^2)+axwbyv.*sind(DELTA).*...
         cosd(repmat(BLADEPOS,1,NPANELS+1))+dSRdt.*sind(repmat(BLADEPOS,1,NPANELS+1));
VNORMAL = axwbyc.*cosd(DELTA);
VINPLANE = tawbyc.*cosd(DELTA).*cosd(repmat(BLADEPOS,1,NPANELS)).*logical(DELTA~=0) + axwbyc.*sind(DELTA);

for i = 1:NBLADES
    TANBC(i,:) = AXWBYC(i,:)./((pi/J).*YC-TAWBYC(i,:).*ROT);
    TANBV(i,:) = AXWBYV(i,:)./((pi/J).*YV-TAWBYV(i,:).*ROT);
end

YVWETVARINDX = find(YV>abs(SR));
if isempty(YVWETVARINDX)
    YCWETVARINDX = zeros(1,0);
else
    YCWETVARINDX = find(YC>(min(YV(YVWETVARINDX))));
end

%% Ventilated propeller area
if SR < 3.4 && SR >= 1
   if dSRdt > 0
       AVAO = polyval([-1.2276 2.2201 -0.1969],SR)*logical(SR<1.7 && SR>1)+logical(SR<=1);
   elseif dSRdt < 0
       AVAO = polyval([-0.1593 1.3420 -3.7357 3.4483],SR)*logical(SR<2.6 && SR>1)+logical(SR<=1);
   end
   ANVAO = 1-AVAO;
end

%% Running loop for all blades
for BLADE = 1:NBLADES
    WAGFLAG = 0;
    YVSUBINDX = find(YV>=WETSPAN(BLADE,1) & YV<=WETSPAN(BLADE,2));
    NWPANELS = length(YVSUBINDX)-1;
    if NWPANELS<4
        % Blade is considered dry
        continue
    end
    YVSUB = YV(YVSUBINDX);
    YCSUBINDX = find(YC>min(YVSUB) & YC<max(YVSUB));
    YCSUB = YC(YCSUBINDX);

    %% Finding circulation for submerged parts of the blade
    [G,VINF,VTSTAR,VASTAR,alpha] = circulation(NITER,TOL,NBLADES,NPANELS,NWPANELS,...
        BLADE,BLADEPOS(BLADE),CRP,YCSUBINDX,YVSUBINDX,YC,YV,TANBC,TANBV,AXWBYC,...
        VNORMAL,VINPLANE,WAGFLAG,CLi,ALPHAi,ZCYC,ZCi,BPAC,...
        CPANGLE,DIAM,CDYC,XHUB,DELTA,HUBVRAD,DRAGYC,PANELSIZE);

    %% Wagner effect
    WAGNER = ones(1,NPANELS);
    if ~isempty(YCWETVARINDX)
        YCWAGNER = max(min(YCWETVARINDX),min(YCSUBINDX)):1:min(max(YCWETVARINDX),max(YCSUBINDX));
        YVWAGNER = max(min(YVWETVARINDX),min(YVSUBINDX)):1:min(max(YVWETVARINDX),max(YVSUBINDX));
        if ~isempty(YCWAGNER)
            WAGFLAG = 1;
            WAGNER = wagnereffect(NITER,TOL,NBLADES,CRP,CLi,ALPHAi,ZCYC,ZCi,BPAC,CPANGLE,DIAM,...
            CDYC,XHUB,NPANELS,SR,YC,YV,YCWAGNER,YVWAGNER,WAGNER,DTHETA,BLADEPOS(BLADE),WTP,AXWYC,...
            TAWYC,AXWYV,TAWYV,VNORMAL,VINPLANE,WAGFLAG,J,ROT,DELTA,HUBVRAD,DRAGYC,PANELSIZE);
        end
    end
    %% For ventilation
    [~,indx07] = min(abs(YCSUB-0.7));
    ANGLE(BLADE,1) = alpha(indx07);
    SPEED(BLADE,1) = VINF(indx07);

    %% TOTAL OF BLADE EFFECTS
    G = G.*WAGNER(YCSUBINDX)';
    CIRC(BLADE,YCSUBINDX) = G';
    CTH = 0.5*(log(1/HUBVRAD)+3)*(NBLADES*CIRC(BLADE,1))^2;
    DVISC = (1/(2*pi)).*VINF.*(CDYC(YCSUBINDX)./DIAM).*DRAGYC(YCSUBINDX);
    tmp1 = -4*sum((VASTAR.*G'+DVISC.*VTSTAR).*PANELSIZE(YCSUBINDX));
    tmp2 = -4*sum((VTSTAR.*G'-DVISC.*VASTAR).*PANELSIZE(YCSUBINDX).*YC(YCSUBINDX));
    CT(BLADE)  = 4*NBLADES*sum((VTSTAR.*G'-DVISC.*VASTAR).*PANELSIZE(YCSUBINDX))-CTH;
    CQ(BLADE)  = 4*NBLADES*sum((VASTAR.*G'+DVISC.*VTSTAR).*YCSUB.*PANELSIZE(YCSUBINDX));
    CFY(BLADE) = cosd(BLADEPOS(BLADE))*tmp1;
    CFZ(BLADE) = sind(BLADEPOS(BLADE))*tmp1;
    CMY(BLADE) = cosd(BLADEPOS(BLADE))*tmp2;
    CMZ(BLADE) = sind(BLADEPOS(BLADE))*tmp2;
end %each blade

%% Thrust loss due to ventilation
if SR < 3.4 && SR >= 1
    cl07fv = (pi/2)*mean(ANGLE) + (2*g*h)/((mean(SPEED)*V)^2);
    VENTFACTOR = ANVAO + ((1.5*EAR)/(sum(CT*J^2*pi/8)/NBLADES))*cl07fv*AVAO;
    if VENTFACTOR > 1
        VENTFACTOR = 1;
    end
end

%% 6 DoF propeller forces
KT  = VENTFACTOR*sum(CT*J^2*pi/8)/NBLADES;
KQ  = (-VENTFACTOR^k)*sum(CQ*J^2*pi/16)/NBLADES;
KFY = sum(CFY*J^2*pi/8);
KFZ = sum(CFZ*J^2*pi/8);
KMY = sum(CMY*J^2*pi/16);
KMZ = sum(CMZ*J^2*pi/16);

T = KT*RHO*n_Hz^2*DIAM^4;
Q = KQ*RHO*n_Hz^2*DIAM^5;
fy = KFY*RHO*n_Hz^2*DIAM^4;
fz = KFZ*RHO*n_Hz^2*DIAM^4;
my = KMY*RHO*n_Hz^2*DIAM^5;
mz = KMZ*RHO*n_Hz^2*DIAM^5;
```

F2 readinput.m

```
%*****************************************************************************************
% ROUTINE:  readinput.m
% ----------------------------------------------------------------------------------------
%
% Object:   Read wake field, propeller and blade section geometry and lifting line input.
%           input.
%
% Method:   The input is made available to the workspace.
%
% PARAMETERS
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% ALPHAi        output  float   Ideal angle of attack of NACA section
% CLi           output  float   Lift coefficient at ideal angle of attack for NACA section
% AXWYC         output  float   Interpolated axial wake field at control points
% AXWYV         output  float   Interpolated axial wake field at vortex points
% TAWYC         output  float   Interpolated tangential wake field at control points
% TAWYV         output  float   Interpolated tangential wake field at vortex points
% BPAC          output  float   Interpolated blade pitch angle at control points
% CDYC          output  float   Interpolated chord length at control points
% CRP           output  float   Swirl cancellation factor
% DIAM          output  float   Propeller diameter
% DRAGYC        output  float   Interpolated drag coefficient at control points
% DTHETA        output  float   Wagner density angle
% EAR           output  float   Expanded blade Area Ratio
% HUBVRAD       output  float   Relative radius of hub vortex
% INIT          output  float   Initial position of propeller blade at simulation start
% NBLADES       output  float   Number of propeller blades
% NITER         output  float   Maximum number of iterations in wake alignment
% NPANELS       output  float   Number of cosine spaced vortex panels
% PANELSIZE     output  float   Length of the vortex panels
% RHO           output  float   Density of water
% ROT           output  float   Direction of rotation for propeller
% TOL           output  float   Relative tolerance of circulation accuracy
% WTP           output  float   Angular position of wake input
% XHUB          output  float   Relative hub radius
% YC            output  float   Coordinates of control points
% YV            outout  float   Coordinates of vortex points
% ZCi           output  float   Maximum camber of NACA section
% ZCYC          outout  float   Maximum camber at control points
% ----------------------------------------------------------------------------------------
% ========================================================================================
% Written by:  Oyvind Oksnes Dalheim
% Last edited: 10.04.15
% ****************************************************************************************

%% Find the directory of the input files
p1 = mfilename;
p2 = mfilename('fullpath');
inputpath = p2(1:length(p2)-length(p1));

%% Read propeller geometry
fid = fopen(strcat(inputpath,'input/propeller.txt'));
temp = textscan(fid, '%f%f%f%f%s%f%f', 'Delimiter', ' ', 'MultipleDelimsAsOne', 1,...
    'HeaderLines', 2, 'CollectOutput', 0, 'CommentStyle', '/*');
ROTATION = temp{5}{1};
PropGeometry  = [cell2mat(temp(1,1:4)),cell2mat(temp(1,6:7))];
BladeGeometry = cell2mat(textscan(fid, '%f%f%f%f%f%f', 'Delimiter', ' ', 'MultipleDelimsAsOne',
    1,...
    'HeaderLines', 3, 'CollectOutput', 0, 'CommentStyle', '/*'));
CamberProfile = cell2mat(textscan(fid, '%f%*s', 'Delimiter', '|', 'MultipleDelimsAsOne', 1,...
    'HeaderLines', 2, 'CollectOutput', 0, 'CommentStyle', '/*'));
[~] = fclose(fid);
clearvars fid temp

%% Read wakefield input
fid = fopen(strcat(inputpath,'input/wakefield.txt'));
[~] = fgets(fid); [~] = fgets(fid);
formatspec = repmat('%f', 1, length(strsplit(fgets(fid)))-1);
fseek(fid, 0, 'bof');
AxialWake = cell2mat(textscan(fid, formatspec, 'Delimiter', ' ', 'MultipleDelimsAsOne', 1,...
    'HeaderLines', 2, 'CollectOutput', 0, 'CommentStyle', '/*'));
TangWake  = cell2mat(textscan(fid, formatspec, 'Delimiter', ' ', 'MultipleDelimsAsOne', 1,...
    'HeaderLines', 1, 'CollectOutput', 0, 'CommentStyle', '/*'));
fseek(fid, 0, 'bof');
formatspec = strcat('%*s', formatspec(3:end));
WRP     = cell2mat(textscan(fid, formatspec, 1, 'Delimiter', ' ', 'MultipleDelimsAsOne', 1,...
    'HeaderLines', 1, 'CollectOutput', 0, 'CommentStyle', '/*'));
[~] = fclose(fid);
WTP = AxialWake(:,1);
clearvars fid formatspec

%% Read Lifting Line properties
fid = fopen(strcat(inputpath,'input/LLproperties.txt'));
[~] = fgets(fid);
LLproperties = cell2mat(textscan(fid, '%f%*s', 'Delimiter', '|', 'MultipleDelimsAsOne', 1,...
    'HeaderLines', 0, 'CollectOutput', 0, 'CommentStyle', '/*'));
[~] = fclose(fid);
clearvars fid

%% Allocating to variables
%Input to the lifting line calculation
NPANELS   = LLproperties(1);  %Number of panels
NITER     = LLproperties(3);  %Maximum number of iterations
HUBVRAD   = LLproperties(2);  %Hub vortex radius/Hub radius
CRP       = LLproperties(4);  %Swirl cancellation factor
RHO       = LLproperties(5);  %Density of water
TOL       = LLproperties(6);  %Circulation accuracy tolerance

%Propeller input
DIAM    = PropGeometry(1);   %Propeller diameter
EAR     = PropGeometry(2);   %Expanded blade area ratio
NBLADES = PropGeometry(3);   %Number of propeller blades
XHUB    = PropGeometry(4);   %Hub ratio
INIT    = PropGeometry(5);   %Initial revolution angle of propeller [deg]
XR      = BladeGeometry(:,1);%Input blade section positions
DTHETA  = PropGeometry(6); %Density of Wagner calculation [deg]

if strcmp(ROTATION,'RIGHT') == 1
    ROT = 1;
elseif strcmp(ROTATION,'LEFT') == 1
    ROT = -1;
else
    error('Invalid direction of rotation input. Check propeller input file.')
end

%Camber line input
CLi     = CamberProfile(1);     %Lift coefficient at ideal angle of attack
ZCi     = CamberProfile(2)/100; %max camber/chord for camber profile input (NACA)
ALPHAi  = CamberProfile(3)*(pi/180);

NWRP = size(AxialWake,2)-1; %Number of wake radial positions in input
NWAP = size(AxialWake,1);   %Number of wake angular positions in input
```

```
%% Preprocessing
AXWYC = zeros(NWAP, NPANELS);   %Axial wake at control points
TAWYC = zeros(NWAP, NPANELS);   %Tangential wake at control points
AXWYV = zeros(NWAP, NPANELS+1); %Axial wake at vortex points
TAWYV = zeros(NWAP, NPANELS+1); %Tangential wake at vortex points

%Establishing cosine spaced control points and vortex panels
[YC, YV] = cosspace(XHUB, 1, NPANELS);
PANELSIZE = diff(YV);

%Interpolating blade geometry at control points
CDYC   = interp1(XR, BladeGeometry(:,2).*DIAM, YC, 'spline');
ZCYC   = interp1(XR, BladeGeometry(:,5)./BladeGeometry(:,2), YC, 'spline');
PDYC   = interp1(XR, BladeGeometry(:,4), YC, 'spline');
DRAGYC = interp1(XR, BladeGeometry(:,6), YC, 'spline');
BPAC   = atan(PDYC./(YC.*pi));

%Interpolating wake field at control, vortex and blade section points
for i = 1:NWAP
    AXWYC(i,:) = interp1(WRP, AxialWake(i,2:end), YC);
    TAWYC(i,:) = interp1(WRP, TangWake(i,2:end), YC);
    AXWYV(i,:) = interp1(WRP, AxialWake(i,2:end), YV);
    TAWYV(i,:) = interp1(WRP, TangWake(i,2:end), YV);
end
clearvars ans i inputpath p1 p2 LLproperties PropGeometry BladeGeometry
clearvars CamberProfile ROTATION TangWake AxialWake PDYC NWRP NWAP XR WRP
```

F3 cosspace.m

```
function [YC,YV] = cosspace(startPoint, endPoint, M)
%*****************************************************************************************
% ROUTINE:  cosspace.m
% ----------------------------------------------------------------------------------------
%
% Object:   Calculate cosine spaced control and vortex points.
%
% Method:   Divide the length of the blade into M vortex panels from hub to
%           blade tip. Concentrates samples at the ends while
%           producing fewer sample points in the center,
%
% PARAMETERS
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% startPoint    input   float   Starting value of the vortex point coordinates.
% endPoint      input   float   Ending value of the vortex point coordinates.
% M             input   float   Number of vortex panels
% YC            output  float   Coordinates of control points.
% YV            outout  float   Coordinates of vortex points.
% ----------------------------------------------------------------------------------------
% INTERNAL VARIABLES
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% del                   float   Angular panel increment.
% ----------------------------------------------------------------------------------------
% ========================================================================================
% Written by:  Oyvind Oksnes Dalheim
% Last edited: 22.03.15
% ****************************************************************************************

%% Checking input
if endPoint <= startPoint
    disp('Cosine spacing failed, because end point is lower than start point');
    YV = [];
    return;
end

%% Calculate control and vortex points
YV = zeros(1,M+1); YC = zeros(1,M);
del = pi/(2*M);

for N = 1:M+1
    YV(N) = (0.5*(endPoint-startPoint))*(1-cos(2*(N-1)*del));
end
for N = 1:M
    YC(N) = (0.5*(endPoint-startPoint))*(1-cos((2*N-1)*del));
end
YV = YV + startPoint;
YC = YC + startPoint;
end
```

F4 circulation.m

```matlab
function [G,VINF,VTSTAR,VASTAR,alpha] = circulation(NITER,TOL,NBLADES,NPANELS,NWPANELS,BLADE,...
    BLADEPOS,CRP,YCSUBINDX,YVSUBINDX,YC,YV,TANBC,TANBV,AXWBYC,VNORMALIN,VINPLANEIN,WAGFLAG,...
    CLi,ALPHAi,ZCYC,ZCi,BPAC,CPANGLE,DIAM,CDYC,XHUB,DELTA,HUBVRAD,DRAGYC,PANELSIZE)
%*****************************************************************************************
% ROUTINE:  circulation.m
% SUBCALL:  induction.m
% ----------------------------------------------------------------------------------------
%
% Object:   Find unknown circulation distributon and induced velocities.
%
% Method:   Iterate between induced velocities and circulation.
%
% PARAMETERS
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% VASTAR        output  float   Resulting axial inflow velocity
% VTSTAR        output  float   Resulting tangential inflow velocity
% VINF          output  float   Velocity seen by propeller blade
% G             output  float   Nondimensional circulation
% alpha         output  float   Angle of attack at 70% radius
% NBLADES       input   float   Number of propeller blades
% NITER         input   float   Maximum number of iterations in wake alignment
% NPANELS       input   float   Number of cosine spaced vortex panels
% NWPANELS      input   float   Number of wet vortex panels
% BLADE         input   float   Blade number
% BLADEPOS      input   float   Circumferential position of blades
% CRP           input   float   Swirl cancellation factor
% YCSUBINDX     input   float   Index to submerged control points
% YVSUBUNDX     input   float   Index to submerged vortex points
% YC            input   float   Coordinates of control points
% YV            inout   float   Coordinates of vortex points
% TANBC         input   float   tan(beta) at control points
% TANBV         inout   float   tan(beta) at vortex points
% AXWYC         input   float   Interpolated axial wake field at control points for actual blade
% WAGFLAG       input   float   Flag that inform if this is a Wagner calculation or not
% CLi           input   float   Lift coefficient at ideal angle of attack for NACA section
% ALPHAi        input   float   Ideal angle of attack of NACA section
% ZCi           input   float   Maximum camber of NACA section
% ZCYC          inout   float   Maximum camber at control points
% BPAC          input   float   Interpolated blade pitch angle at control points
% CPANGLE       input   float   Controllable pitch angle at blade root [deg]
% CDYC          input   float   Interpolated chord length at control points
% DIAM          input   float   Propeller diameter
% DRAGYC        input   float   Interpolated drag coefficient at control points
% XHUB          input   float   Relative hub radius
% DELTA         input   float   Azimuth angle [deg]
% HUBVRAD       input   float   Relative radius of hub vortex
% PANELSIZE     input   float   Length of the vortex panels
% TOL           input   float   Relative tolerance of circulation accuracy
% ----------------------------------------------------------------------------------------
% INTERNAL VARIABLES
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% YCSUB                 float   Submerged control points
% YVSUB                 float   Submerged vortex points
% BETAiC                float   Hydrodynamic inflow angle
% UAHIF                 float   Axial horseshoe influence coefficient
% UTHIF                 float   Tangential horseshoe influence coefficient
% UASTAR                float   Axial induced velocity
% UTSTAR                float   Tangential induced velocity
% BETAiC                float   Hydrodynamic inflow angle
% TANBCB                float   tan(beta) at control point for specific blade
% TANBVB                float   tan(beta) at vortex point for specific blade
% TANBIC                float   tan(beta_i) at control point for specific blade
% TANBIV                float   tan(beta_i) at vortex point for specific blade
% CHIYC                 float   Wake skew angle at control point
% U0                    float   Mean annular axial induced velocity
% CTH                   float   Hub vortex drag coefficient
% VNORMAL               float   Inflow velocity normal to propeller
% VINPLANE              float   Inflow velocity in propeller plane
% ----------------------------------------------------------------------------------------
% ========================================================================================
% Written by:  Oyvind Oksnes Dalheim
% Last edited: 10.06.15
% ****************************************************************************************

%% Preprocessing
YCSUB = YC(YCSUBINDX);
YVSUB = YV(YVSUBINDX);

%% Preallocating for speed
tempC    = zeros(NITER,NWPANELS);
tempV    = zeros(NITER,NWPANELS+1);
memCIRC  = zeros(NITER,NWPANELS);
UAQ      = zeros(1,NWPANELS+1);
UTQ      = zeros(1,NWPANELS+1);
UAHIF    = zeros(NWPANELS);
UTHIF    = zeros(NWPANELS);
AXWYCB   = AXWBYC(BLADE,YCSUBINDX);
TANBCB   = TANBC(BLADE,YCSUBINDX);
TANBVB   = TANBV(BLADE,YVSUBINDX);
U0       = zeros(1,size(AXWYCB,2));
VNORMAL  = zeros(1,length(YCSUBINDX));
VINPLANE = zeros(1,length(YCSUBINDX));

%% Initialising effect of oblique inflow
if WAGFLAG ~= 1 && DELTA ~= 0
    U0 = 0.5.*AXWYCB;
    VNORMAL  = VNORMALIN(BLADE,YCSUBINDX);
    VINPLANE = VINPLANEIN(BLADE,YCSUBINDX);
    CHIYC  = atand(VINPLANE./(VNORMAL+U0));
    UASTAR = U0.*(1+((15*pi)/32).*tand(CHIYC./2).*YCSUB.*-1*cosd(BLADEPOS-90));
    UTSTAR = zeros(1,NWPANELS);
    TANBIC = (AXWYCB+UASTAR)./((AXWYCB./TANBCB)+UTSTAR);
    if length(YCSUBINDX)>=2
        TANBIV = interp1(YCSUB,TANBIC,YVSUB,'spline','extrap');
    else
        TANBIV = repmat(TANBIC,1,length(YCSUBINDX)+1);
    end
else
    CHIYC  = zeros(1,NWPANELS);
    TANBIC = TANBCB.*1.2;
    TANBIV = TANBVB.*1.2;
end


%% Iteration procedure
for ITER=1:NITER
    if ITER == 3 || ITER == 5
        TANBIC = (tempC(ITER-2,:)+tempC(ITER-1,:))/2;
        TANBIV = (tempV(ITER-2,:)+tempV(ITER-1,:))/2;
    end

    %% Compute axial and tangential horseshoe influence coefficients
    for m = 1:NWPANELS
        RCQ = YCSUB(m);
        for n = 1:NWPANELS+1
            %Induction of trailing vortices shed at YV
            TANBIQ = TANBIV(n);
            RVQ = YVSUB(n);
            [UAIF, UTIF] = induction(NBLADES,TANBIQ,RCQ,RVQ);
            UAQ(n) = -0.5*(UAIF/(YCSUB(m)-YVSUB(n)));
            UTQ(n) =  0.5*((UTIF*CRP)/(YCSUB(m)-YVSUB(n)));

            %Induction of hub-image
            if floor(YVSUB(1)*10^8)/10^8>XHUB
                continue
            end
            RVQ=YVSUB(1)^2/YVSUB(n);
            TANBIQ = TANBIV(1)*YVSUB(1)/RVQ;
            [UAIFH, UTIFH] = induction(NBLADES,TANBIQ,RCQ,RVQ);
            UAQ(n) = UAQ(n)+0.5*(UAIFH/(YCSUB(m)-RVQ));
            UTQ(n) = UTQ(n)-0.5*((UTIFH*CRP)/(YCSUB(m)-RVQ));
        end
        %Final building of influence functions
        UAHIF(m,:) = diff(UAQ);
        UTHIF(m,:) = diff(UTQ);
    end

    %% Solve system of equations
    A = UAHIF-UTHIF.*repmat(TANBIC',1,size(UTHIF,2));
    B = (AXWYCB.*((TANBIC./TANBCB)-1))';
    G = A\B;

    %% Evaluating induced velocities from the circulation
    UASTAR = (UAHIF*G)';
    UTSTAR = (UTHIF*G)';

    %% Account for skewed propeller wake effect
    if DELTA ~= 0 && WAGFLAG ~= 1
            VASTAR = (AXWYCB+UASTAR);
            VTSTAR = (AXWYCB./TANBCB)+UTSTAR;
            VINF = sqrt(VASTAR.^2+VTSTAR.^2);
            DVISC = (1/(2*pi)).*VINF.*(CDYC(YCSUBINDX)./DIAM).*DRAGYC(YCSUBINDX);
            tmp  = (NBLADES*(VTSTAR.*G'-DVISC.*VASTAR))./(2*YCSUB);
            old = U0.*1.2;
            while max(abs((U0-old)./U0)) > 0.05
                old = U0;
                U0 = tmp./sqrt(VINPLANE.^2+(VNORMAL+U0).^2);
            end
            CHIYC = atand(VINPLANE./(VNORMAL+U0));
    end

    VASTAR = (AXWYCB+UASTAR);
    VTSTAR = (AXWYCB./TANBCB)+UTSTAR;
    VINF = sqrt(VASTAR.^2+VTSTAR.^2);

    BETAiC = ((CLi/(2*pi)-ALPHAi).*(ZCYC(YCSUBINDX)/ZCi))+BPAC(YCSUBINDX)+...
             CPANGLE*(pi/180)-((G'*DIAM)./(VINF.*CDYC(YCSUBINDX)));

    if WAGFLAG ~= 1
        tmp = (tan(BETAiC).*VTSTAR-AXWYCB).*(1+((15*pi)/32)...
              .*tand(CHIYC./2).*YCSUB.*-1*cosd(BLADEPOS-90));
        TANBIC = ((AXWYCB+tmp)./VTSTAR);
        if ITER == 1 %Including reduction factor
            TANBIC = 0.8.*TANBIC;
        end
    else
        TANBIC = tan(BETAiC);
    end

    if length(YCSUBINDX)>=2
        TANBIV = interp1(YCSUB,TANBIC,YVSUB,'spline','extrap');
    else
        TANBIV= repmat(TANBIC,1,length(YCSUBINDX)+1);
    end
    tempC(ITER,:) = TANBIC;
    tempV(ITER,:) = TANBIV;
    memCIRC(ITER,:) = G';

    %% Check if the circulation has converged sufficiently
    if ITER > 2
        if abs(max((memCIRC(ITER,:)-memCIRC((ITER-1),:))./memCIRC(ITER,:))) < TOL
            continue
        end
    end

end
alpha = (BPAC(YCSUBINDX)+CPANGLE*(pi/180))-BETAiC;
end
```

F5 induction.m

```
function [UAIF, UTIF] = induction(NBLADES,TANBIQ,RCQ,RVQ)
%*****************************************************************************************
% ROUTINE:  induction.m
% ----------------------------------------------------------------------------------------
%
% Object:   Find horseshoe influence coefficients.
%
% Method:   Wrench's approximations to Bessel functions.
%
% PARAMETERS
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% UAHIF         output  float   Axial horseshoe influence coefficient
% UTHIF         output  float   Tangential horseshoe influence coefficient
% RCQ           input   float   Coordinate of control point
% RVQ           input   float   Coordinate of vortex point
% TANBIQ        input   float   tan(beta_i) at vortex point
% NBLADES       input   float   Number of propeller blades
% ----------------------------------------------------------------------------------------
% ========================================================================================
% Written by:  Oyvind Oksnes Dalheim
% Last edited: 22.04.15
% ****************************************************************************************

%% Calculates horseshoe influence coefficients using Wrench's approximations
Y0 = 1/TANBIQ;
ETA = RVQ/RCQ;
Y = Y0/ETA;
XS = 1+Y^2;
T = sqrt(XS);
V = 1+Y0^2;
W = sqrt(V);
R = (((Y0*(T-1))/(Y*(W-1)))*exp(T-W))^NBLADES;
XX = (1/(2*NBLADES*Y0))*((V/XS)^0.25);
YY = ((9*Y0^2)+2)/(V^1.5)+((3*Y^2-2)/(XS^1.5));
Z = (1/(24*NBLADES))*YY;
if (Y >= Y0)
    F2 = XX*(1/(R-1)-Z*log(1+1/(R-1)));
    UAIF = 2*NBLADES^2*Y0*Y*(1-ETA)*F2;
    UTIF = NBLADES*(1-ETA)*(1+2*NBLADES*Y0*F2);
else
    if (R > 1.0E-12)
        RAT = 1/((1/R)-1);
        if RAT<-1
            RAT=0;
        end
    else
        RAT = 0;
    end
    F1 = -XX*(RAT+Z*log(1+RAT));
    UAIF = NBLADES*Y0*(1-1/ETA)*(1-2*NBLADES*Y0*F1);
    UTIF = 2*NBLADES^2*Y0*(1-ETA)*F1;
end
end
```

F6 wagnereffect.m

```
function WAGNER = wagnereffect(NITER,TOL,NBLADES,CRP,CLi,ALPHAi,ZCYC,ZCi,BPAC,CPANGLE,DIAM,...
    CDYC,XHUB,NPANELS,SR,YC,YV,YCWAGNER,YVWAGNER,WAGNER,DTHETA,BLADEPOS,WTP,AXWYC,TAWYC,...
    AXWYV,TAWYV,VNORMAL,VINPLANE,WAGFLAG,J,ROT,DELTA,HUBVRAD,DRAGYC,PANELSIZE)
%*****************************************************************************************
% ROUTINE:  wagnereffect.m
% SUBCALL:  circulation.m
% ----------------------------------------------------------------------------------------
%
% Object:   Find the thrust loss factor due to Wagner effect.
%
% Method:   Find the velocity seen by the propeller blade at certain user-defined
%           circumferential positions by solving for the unknown
%           circulation. Calculate number of chord lengths travelled by
%           each section since water entry, and use Minsaas' Wagner function.
%
% PARAMETERS
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% WAGNER        output  float   beta_factor for Wagner effect at each blade section
% NBLADES       input   float   Number of propeller blades
% NITER         input   float   Maximum number of iterations in wake alignment
% CRP           input   float   Swirl cancellation factor
% CLi           input   float   Lift coefficient at ideal angle of attack for NACA section
% ALPHAi        input   float   Ideal angle of attack of NACA section
% ZCi           input   float   Maximum camber of NACA section
% ZCYC          inout   float   Maximum camber at control points
% BPAC          input   float   Interpolated blade pitch angle at control points
% CPANGLE       input   float   Controllable pitch angle at blade root [deg]
% DIAM          input   float   Propeller diameter
% CDYC          input   float   Interpolated chord length at control points
% XHUB          input   float   Relative hub radius
% NPANELS       input   float   Number of cosine spaced vortex panels
% SR            input   flot    Submergence ratio
% YC            input   float   Coordinates of control points
% YV            input   float   Coordinates of vortex points
% YCWAGNER      input   float   Index to control points that are affected by Wagner
% YVWAGNER      input   float   Index to vortex points that are affected by Wagner
% WAGNER        input   float   beta_factor for Wagner effect at each blade section
% DTHETA        input   float   Wagner density angle
% BLADEPOS      input   float   Circumferential position of blades
% WTP           input   float   Angular position of wake input
% AXWYC         input   float   Interpolated axial wake field at control points
% AXWYV         input   float   Interpolated axial wake field at vortex points
% TAWYC         input   float   Interpolated tangential wake field at control points
% TAWYV         input   float   Interpolated tangential wake field at vortex points
% WAGFLAG       input   float   Flag that inform if this is a Wagner calculation or not
% J             input   float   Advance number
% ROT           input   float   Direction of rotation for propeller
% DELTA         input   float   Azimuth angle [deg]
% HUBVRAD       input   float   Relative radius of hub vortex
% DRAGYC        input   float   Interpolated drag coefficient at control points
% PANELSIZE     input   float   Length of the vortex panels
% TOL           input   float   Relative tolerance of circulation accuracy
% ----------------------------------------------------------------------------------------
% INTERNAL VARIABLES
% Name          Note    Type    Description
% ----------------------------------------------------------------------------------------
% WAGANENTYV            float   Angle when vortex point affected by Wagner enters the water
% WAGANENTYC            float   Angle when control point affected by Wagner enters the water
% WAGBLADEPOS           float   Circumferential positions where Wagner effect is calculated
% NWAGBLADEPOS          float   Number of WAGBLADEPOS
% AXWBYCWAG             float   Interpolated axial wake field at the Wagner positions
% AXWBYVWAG             float   Interpolated axial wake field at the Wagner positions
% TAWBYCWAG             float   Interpolated tangential wake field at the Wagner positions
% TAWBYVWAG             float   Interpolated tangential wake field at the Wagner positions
% WAGVINF               float   Velocity seen by propeller blade at all Wagner positions
% NUMCHORD              float   Number of chord lengths travelled by each blade section
% BLADEENT              float   Angle when blade enters the water
% WETSPAN               float   Wettet span interval of blade
% YCSUBINDX             float   Index to submerged control points
% YVSUB                 float   Submerged vortex points
% YVSUB                 float   Submerged vortex points
% NWPANELS              float   Number of wet vortex panels
% noData                float   Number of desired Wagner positions
% ---------------------------------------------------------------------------------------
% =======================================================================================
% Written by:  Oyvind Oksnes Dalheim
% Last edited: 10.06.15
% ***************************************************************************************

coder.extrinsic('circulation.m', 'induction.m');
coder.varsize('WAGBLADEPOS', [1 360],[0 1])

%% Determining the number of Wagner positions
assert (NPANELS<=64);
WAGANGENTYC = acosd(SR./YC(YCWAGNER));
if SR >= 0
    WAGANGENTYV = acosd(SR./YV(YVWAGNER(YVWAGNER>4)));
else
    WAGANGENTYV = acosd(SR./YV(YVWAGNER(YVWAGNER<NPANELS-2)));
end

noData = floor((BLADEPOS - min(WAGANGENTYV))/DTHETA)+1;
assert (noData<360);
WAGBLADEPOS  = linspace(min(WAGANGENTYV),BLADEPOS,noData);
NWAGBLADEPOS = length(WAGBLADEPOS);
assert (NWAGBLADEPOS<=360);

%% Preallocating variables for speed
AXWBYCWAG = zeros(NWAGBLADEPOS,NPANELS);
AXWBYVWAG = zeros(NWAGBLADEPOS,NPANELS+1);
TAWBYCWAG = zeros(NWAGBLADEPOS,NPANELS);
TAWBYVWAG = zeros(NWAGBLADEPOS,NPANELS+1);
TANBCWAG  = zeros(NWAGBLADEPOS,NPANELS);
TANBVWAG  = zeros(NWAGBLADEPOS,NPANELS+1);
WAGVINF   = zeros(NWAGBLADEPOS,NPANELS);
NUMCHORD  = zeros(1,NPANELS);

%% Interpolating wake field at each Wagner position
for i = 1:NPANELS
    AXWBYCWAG(:,i) = interp1(WTP,AXWYC(:,i),WAGBLADEPOS,'spline');
    TAWBYCWAG(:,i) = interp1(WTP,TAWYC(:,i),WAGBLADEPOS,'spline');
end
for i = 1:NPANELS+1
    AXWBYVWAG(:,i) = interp1(WTP,AXWYV(:,i),WAGBLADEPOS,'spline');
    TAWBYVWAG(:,i) = interp1(WTP,TAWYV(:,i),WAGBLADEPOS,'spline');
end
for i = 1:NWAGBLADEPOS
    TANBCWAG(i,:) = AXWBYCWAG(i,:)./((pi/J)*YC-TAWBYCWAG(i,:)*ROT);
    TANBVWAG(i,:) = AXWBYVWAG(i,:)./((pi/J)*YV-TAWBYVWAG(i,:)*ROT);
end

%% Finding wetted span length for each Wagner position
```

```
WETSPAN = [ones(NWAGBLADEPOS,1).*XHUB, ones(NWAGBLADEPOS,1)];
BLADEENT = acosd(SR);
if SR >= 0
    WETSPAN(:,2) =
        ceil(SR./(cosd(WAGBLADEPOS-(WAGBLADEPOS-BLADEENT).*logical((WAGBLADEPOS>=BLADEENT)...
        & (WAGBLADEPOS<=(360-BLADEENT))))).*10^6)/10^6;
    WETSPAN(~isfinite(WETSPAN)) = 1;
elseif SR < 0
    WETSPAN(:,1) = floor((1-(1-SR./cosd(WAGBLADEPOS)).*logical((WAGBLADEPOS>BLADEENT)...
        & (WAGBLADEPOS<(360-BLADEENT)))).*10^6)/10^6;
end

%% Calculating Wagner effect
for k=1:NWAGBLADEPOS
    YVSUBINDX = find(YV>=WETSPAN(k,1) & YV<=WETSPAN(k,2));
    YVSUB = YV(YVSUBINDX);
    YCSUBINDX = find(YC>min(YVSUB) & YC<max(YVSUB));
    YCSUB = YC(YCSUBINDX);
    NWPANELS = length(YCSUB);
    [~,WAGVINF(k,YCSUBINDX),~,~,~] =
        circulation(NITER,TOL,NBLADES,NPANELS,NWPANELS,k,WAGBLADEPOS(k),...
    CRP,YCSUBINDX,YVSUBINDX,YC,YV,TANBCWAG,TANBVWAG,AXWBYCWAG,VNORMAL,VINPLANE,WAGFLAG,CLi,ALPHAi,...
    ZCYC,ZCi,BPAC,CPANGLE,DIAM,CDYC,XHUB,DELTA,HUBVRAD,DRAGYC,PANELSIZE);
end
for i=1:length(YCWAGNER)
    j = YCWAGNER(i);
    NUMCHORD(j) = J*DIAM*(mean(WAGVINF(:,j))./CDYC(j)).*((BLADEPOS-WAGANGENTYC(i))/360);
end
WAGNER(YCWAGNER) = 0.5+0.5.*sqrt(1-((155-NUMCHORD(YCWAGNER))./155).^(27.59));
end
```

G Electronic appendages

The propeller simulation model is submitted in an electronic appendage as a zip-file along with the master thesis. In addition a poster presentation of the project is attached. The electronic appendage contains the following files:

Poster

Input files:

```
propeller.txt
wakefield.txt
LLproperties.txt
```

Matlab files:

```
readinput.m
cosspace.m
circulation.m
induction.m
wagnereffect.m
```

Simulink model:

```
PropSiM.slx
```