



NTNU – Trondheim
Norwegian University of
Science and Technology

Empirical prediction of resistance of fishing vessels

Kristoffer Kleppesø

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

Kristoffer Kleppesø

Empirical prediction of resistance of fishing vessels

In the early design phase, it is essential for the ship designer to have access to quick and reasonably accurate predictions of the resistance of the ship, so that powering predictions can be made as quickly and accurately as possible. For this purpose, empirical resistance prediction methods are often applied.

Fishing vessels often have hull forms which differs quite significantly from typical merchant vessels. Therefore, the classical empirical methods, like Guldhammer-Harvald, Holtrop, and Hollenbach, aren't really suitable. Thus, a need for identifying better methods, and possibly development of new empirical methods, has been expressed by the practitioners in the field.

In the project thesis, the most commonly used empirical resistance prediction methods were evaluated against a dataset of 48 fishing vessels that have been model tested at MARINTEK in recent years. In the master thesis the aim is to provide improved empirical prediction methods tailored for modern fishing vessels.

On this basis, the objectives of this master thesis are to:

- Update the simple resistance prediction formula by Digernes to match the most up-to-date data material available
- Develop an updated Artificial Neural Network (ANN) prediction method for fishing vessels.
 - For this purpose, the thesis should contain an overview and explanation of the principles of ANN, as well as explanation of the actually used method.
- Develop a simple method to include propulsive efficiency in the predictions, so that prediction of required engine power can be easily done.

The developed methods shall be evaluated and validated against experimental (and possibly numerical) results. The agreement shall be commented, and any need for further development shall be identified.

The ANN method shall be implemented such that it can be easily used in practical work, and also so that it might be integrated in existing computer programs, like ShipX.

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



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

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 : 09.06.2015

Trondheim, 15.01.2015

Sverre Steen
Supervisor

Preface

The motivation for this thesis has been to see if it is possible to improve the results from existing empirical methods. Last semester, I investigated the performance of empirical methods for fishing vessels in a pre project and the results motivated me to see if ANN could give better and more reliable results.

This work has been a steep learning curve in Artificial Neural Networks and a great experience for me. From the work with the pre project, this thesis has developed into a comprehensive study of the possibilities that Artificial Neural Networks gives. I have worked steadily throughout the semester and this has given me a good understanding of the problems I encountered and made it possible to solve them in a good way. I'm pleased with the result, and hope that this master thesis can be of use for those that are interested to see the possibilities with artificial neural networks.

There are many people that have helped me with this thesis, and I'm very grateful for all this support. I cannot write all the names here, but some deserves to be mentioned. The results would not have been possible to produce without much appreciated help from MARINTEK, and their kindness to give me access to their model test results and answer all my questions. I would also like to thank my supervisor Professor Sverre Steen, he has been a good mentor through the last two semesters and helped me to find good solutions to many problems

Kristoffer Kleppestø



Trondheim, 9th July 2015

Sammendrag

Muligheten for å benytte Artificial Neural Network til å estimere motstand og propulsjonsvirkningsgrad for skip er testet. Ulike ANN er testet med varierende input data, netverksstørrelse og struktur samt ulik oppdeling av datamaterialet mellom trening- og testgrupper. Metoder er utviklet for å estimere motstand (C_R), total propulsjonsvirkningsgrad (η_D), propellvirkningsgrad (η_0), skrogvirkningsgraden (η_H), medstrømsfaktor (w), thrustdeduksjon (t) og relativ rotasjonsvirkningsgrad (η_R). Datamaterialet for denne masteroppgaven består av modellforsøksresultater ved MARINTEK fra 193 ulike fiskefartøy og lastkondisjoner.

Det er funnet den beste nettverkstrukturen for hvert problem. Et nettverk med totalt 45 celler har vist seg å gi generelt gode resultater, men for skrogvirkningsgraden har mindre nettverk gitt best resultater.

Oppsummert prestereer ANN metodene svært godt. Resultatene for motstandspreddisjon er sammenlignet mot andre empiriske metoder, og ANN presterer alltid best, ofte med god margin. Av metodene for propulsjonsvirkningsgrader var det metodene for den totale propulsjonsvirkningsgraden og propellvirkningsgrad som gav minst avvik. Ulike sett av metoder ble satt sammen for å estimere effektbehovet, og best resultat ble oppnådd ved å benytte kun metodene C_R og total propulsjonsvirkningsgrad. I tabellen nedenfor er prosentvis middelvik og standardavvik for alle ANN metodene oppgitt:

	Middelvik[%]	Standardavvik
C_R	1.4827	0.1745
η_D	0.0423	0.0409
η_0	-1.5626	0.0602
η_R	0.1688	0.0372
η_H	0.5992	0.0684
thrust	2.7184	0.1714
medstrøm	6.4155	0.2914

Det er også utført en regressjonsanalyse for Digernes Formel. Nye koeffisienter er beregnet og avvikene for metoden ble redusert slik at metoden nå skal gi bedre resultater for moderne fiskefartøy.

Konklusjonen fra dette arbeidet er at ANN er i stand til å gi gode estimater for et bredt spekter av problemer. Det er enkelt å trene nye nettverk, og brukeren har mulighet til å lage spesialiserte metoder for spesielle fartøykategorier.

Summary

The possibility to use Artificial Neural Network for estimating ship resistance and propulsion coefficients are investigated. Different ANNs are tested by varying input parameters, network size and complexity and division of data material into training and testing sets. ANN prediction methods are trained for Resistance (C_R), total propulsion efficiency (η_D), open water efficiency (η_0), hull efficiency (η_H), wake fraction (w), thrust deduction (t) and relative rotative efficiency (η_R). The data material for the thesis are model test results from MARINTEK and consist of 193 fishing vessels and loading conditions.

Each problem has its own network structure that give best performance. A network size of 45 neurons generally performs well, while for the hull efficiency coefficients better performance are achieved with smaller networks.

The performance for ANN methods are very good. The resistance prediction method are compared to other empirical methods, and the performance are often much better. For the propulsion coefficients, the methods for total propulsion efficiency and open water efficiency gave best performance. Different combinations of prediction methods are combined to make an estimate for installed engine power, the combination of ANN methods for C_R and total propulsion efficiency gave the best result. The table gives mean error and standard deviation in percentage for all the trained ANN methods:

	Mean error[%]	Standard Deviation
C_R	1.4827	0.1745
η_D	0.0423	0.0409
η_0	-1.5626	0.0602
η_R	0.1688	0.0372
η_H	0.5992	0.0684
thrust	2.7184	0.1714
wake	6.4155	0.2914

A new regression analyse was performed for Digernes formula. The new coefficients improved the formula, and it should now perform better for modern fishing vessels.

The conclusion from this work is that ANN are capable of making good predictions for a broad range of problems. It is easy to train new methods, and one can easily make individual methods for specific vessel categories.

Contents

1	Introduction	1
2	Previous work	2
3	Resistance theory	3
3.1	Residuary resistance coefficient	3
3.2	Form factor	4
3.3	Frictional resistance	4
3.4	Air resistance	5
3.5	Base drag	5
3.6	Appendage resistance	5
3.7	Correlation factor	6
4	Propulsion theory	7
4.1	Open water efficiency	7
4.2	Hull efficiency	7
4.3	Wake fraction	8
4.4	Thrust deduction	8
4.5	Relative rotative efficiency	9
5	Empirical methods	10
5.1	Resistance prediction methods	10
5.1.1	Digernes formula	10
5.1.2	Hollenbachs resistance estimate	11
5.1.3	Holtrop statistical method	13
5.1.4	Oortmerssen power prediction method	14
5.2	Propulsion prediction methods	16
5.2.1	Wageningen B-series	16
5.2.2	BSRA Propulsion factors	17
5.2.3	Harvald graphical method for propulsion factors	18
5.2.4	Holtrop estimate of propulsion factors	21
5.2.5	Taylor wake formula	22

6	ANN theory	24
6.1	Introduction	24
6.2	The neuron	25
6.3	Network structure	26
6.4	ANN learning	28
7	Background data	32
8	Updated Digernes Formula	34
9	Network structures	37
9.1	C_R estimation	38
9.1.1	A second iteration	41
9.2	Total Propulsion efficiency	43
9.3	Open Water efficiency	45
9.4	Hull Efficiency	45
9.5	relative rotative efficiency	46
10	ANN programming	48
10.1	Input processing	48
10.2	Matlab NeuralNetwork toolbox	49
10.3	Final script	49
11	Results	51
11.1	Individual methods	52
11.2	Estimating delivered power	54
12	Discussion	56
12.1	Future work	56
A	Digernes coefficients	I
B	Error Plots	III
C	ANN.m	VI
D	MyNeuralNetworkFunction.m	IX

List of Figures

5.1	Hollenbach’s modes for «maximum», «mean» and «minimum» resistance, is tested on fishing vessels. Error between formula and model tests are plotted against the Froude number. The horizontal black line indicates no error (0%) and y-axis range is from -100% to 250%, where negative error indicates underprediction of resistance. (Kleppestø, 2014)	12
5.2	Oortmerssen uses the diagonal of the hull to calculate his form factor for a vessel. This figure illustrated how two of the coefficient is to be found	15
5.3	Percentage error plotted against Froude number for Oortmerssen method. Zero error is marked with a horizontal line, and y-axis goes from -100% til 250% (Kleppestø, 2014)	16
5.4	Harvalds graphical representation of the wake and thrust deduction factor for single-crew ships: $w = w_1 + w_2 + w_3$; $t = t_1 + t_2 + t_3$ (Harvald, 1983, p. 166)	19
5.5	Propeller arrangements as defined by Harvald to estimate the thrust deduction (Harvald, 1983, p. 184)	20
6.1	Comparison of a stylized biological and an artificial neuron (Ünal et al., 2013, ch. 1)	26
6.2	By combining sigmoids functions it is possibl to reproduce complex continuous functions	28
6.3	Schematic overview of a Neural Network	29
6.4	AN example of an overfitted function. The dataset consist of an sinus curve with a random error added to the data points. The solid line are the results from an overfitted ANN, while the dotted line are the underlying function. (Beale et al., 2015, p. 8-32)	30
6.5	Flowchart of that illustrates the process of supervised learning for a FFANN (Kattan et al., 2011, p. 18)	30
7.1	Graphical representation of some non-dimensional coefficients in the data material	33
8.1	Calculated performance for the different Robust fitting weight functions given in Table 8.1. The left axis are the RMSE while the right axis are the calculated regression coefficient	36

9.1	Graphical representation for the performance of different ANN architectures, with different number and distribution of neurons. Performance is represented with mean-square-error, and low values are best performance	40
9.2	Graphical representation for the performance of different Neural Network architectures, with different number and distribution of neurons. Performance is represented with mean-square-error	42
9.3	Graphical representation for the performance of different Input parameter sets for estimating η_D (is this needed?)	44
B.1	The plots illustrates all the calculated errors for delivered power in percentage on the z-axis (upwards). Each vessel are on the x-axis (0-200) and the speeds in the y-axis (0-25). The vessels are testes for different number of speeds.	IV
B.2	The plots illustrates all the calculated errors for η_0 in percentage on the z-axis (upwards). Each vessel are on the x-axis (0-200) and the speeds in the y-axis (0-25). The vessels are testes for different number of speeds. The figure are included to show that an ANN network can have a very good performance in general, but fail considerable when it misses. In these two cases, there are an error (zero value) for one of the input parameters that creates the two large errors	V

List of Tables

5.1	Coefficient values for Digernes method, (Formula 5.2)	11
5.2	Field of validity for Digernes method as it is defined by Digernes	11
5.3	Field of validity for Hollenbach's method as they are defined for the «minimum resistance» mode (Hollenbach (1998))	13
5.4	Field of validity for Holtrops statistical method for fishing vessels, as it is defined in his first article (Holtrop, 1977)	14
5.5	Field of validity for Oortmerssen method which is based on the bar charts of the data material as it's given in the article (Van Oortmerssen, 1971)	16
5.6	Field of validity for Wageningen B-series	17
5.7	Field of validity for Harvalds graphical method which is based on the presentation of the data material as it is indicated in (Harvald, 1983). Valid for both wake and thrust deduction	21
7.1	Outer limits for data material, who also will function as field of validity for ANN method	33
8.1	Matlabs weight functions available in the Robust fit option. All these are tested to see which gave the best results for Digernes Formula	35
8.2	Coefficient values for Digernes Formula found from regression analysis with a modern data material	35
9.1	Hardware and Software used to design and train ANN network	37
9.2	Three dimensionless parameter sets that are tested for C_R prediction, to identify necessary number of input parameters. Input set 2 was meant to be larger, but because of holes in the database some less common parameters was removed	39
9.3	An initial set of network structures to test how the ANN results depends on node distribution	39
9.4	New set of Neural Network structures iterated from performance to Networks in Table 9.3, with 40-50 neurons. This test set investigate if the shape of the network has any influence on the results	41
9.5	Six input sets, that are tested for η_D network. #4-6 are dimensionless	44
9.6	Performance data for η_D ANN	45
9.7	Properties of η_0 ANN	45

9.8	Performance of Thrust deduction	47
9.9	Performance of wake	47
9.10	Performance of Hull efficiency	47
9.11	Mean estimation of Hull efficiency	47
11.1	Error calculation for the tested C_R estimation methods. All the results are converted to the total resistance in Newton before the errors are calculated. The errors are given as percentage compared to the model test results.	52
11.2	Error calculation for the tested wake estimation methods. The errors are given as percentage compared to the model test results. The mean value are given as an alternative to the ANN. The reason for this are explained in Section 9.4	53
11.3	Error calculation for the tested thrust estimation methods. The errors are given as percentage compared to the model test results. The mean value are given as an alternative to the ANN. The reason for this are explained in Section 9.4	53
11.4	Error calculation for the tested η_R estimation methods. The errors are given as percentage compared to the model test results. The mean value are given as an alternative to the ANN. The reason for this are explained in Section 9.4	53
11.5	Error calculation for the tested η_0 estimation methods. The errors are given as percentage compared to the model test results for all tested J-values	54
11.6	An overview of the calculated errors for all trained ANNs. The errors are given as percentage compared to model test results, and the standard deviation are calculated for the percentage errors	54
11.7	An overview of the calculated errors for the delivered power for three different combinations of methods. #1 C_R and η_D are calculated from ANN. #2 C_R , wake, thrust deduction, η_0 and η_R from ANN. #3 R_{Tot} from updated Digernes formula, wake from Taylors formula, thrust deduction from Holtrop's formula, η_0 from Wageningen B-series and η_D as the mean value from data material.	55
A.1	The resulting coefficient values for different robust weight function from regression analysis in Matlab. See Section 8	II

List of symbols

Hull geometry

L_{WL}	– Length of Waterline
L_{PP}	– Length between perpendiculars
L_{OS}	– Length overall submerged
L_D	– Displacement length (Oortmerssen)
B	– Moulded Breadth
T	– Draught amidship
T_{AP}	– Draught at aft perpendicular
T_{FP}	– Draught at forward perpendicular
S	– Wetted surface
S_B	– Area of transom stern
∇	– Volume displacement
LCB	– Longitudinal center of buoyancy from $0.5 \cdot L_{PP}$ [m]
lcb	– Longitudinal center of buoyancy relative to $0.5 \cdot L_{PP}$ [%]
H	– Hull roughness [μm]
C_M	– Midship section coefficient
C_B	– Block coefficient
C_P	– Prismatic coefficient
ψ	– Angle of the midship diagonal (Oortmerssen)
D_w	– BSRA wake coefficient
D_t	– BSRA thrust coefficient

Propeller geometry

P/D	– Pitch ratio
A_E/A_0	– Expanded blade area ratio
Z	– Number of propeller blades
D_P	– Propeller diameter

Resistance

C_{Tot}	– Total resistance coefficient
C_R	– Residuary resistance coefficient
C_W	– Wave resistance coefficient
C_F	– Frictional resistance coefficient
ΔC_F	– Hull roughness allowance
C_B	– Bulb pressure resistance coefficient
C_{TR}	– Transom stern resistance coefficient (Holtrop)
C_{AA}	– Air resistance coefficient
C_{BD}	– Base drag coefficient (due to transom stern)
C_{App}	– Appendage resistance coefficient
C_A	– Correlation factor coefficient
<i>Resistance components, R, uses the same subscripts as resistance coefficients</i>	
$1 + k_0$	– Form factor
ϕ	– Fullness parameter (MARINTEK)

Propulsion

T	– Propeller thrust
η_D	– Total propulsive efficiency ($= \eta_H \cdot \eta_0 \cdot \eta_R$)
η_H	– Hull efficiency ($= (1 - t)/(1 - w)$)
η_0	– Open water efficiency
η_R	– Relative rotative efficiency
w	– Wake fraction (effective)
w_P	– Potential wake
w_F	– Frictional wake
w_W	– Wave wake
t	– Thrust deduction
n	– Propeller revolutions [Hz]
V_A	– Advance velocity ($= V(1 - w)$)
K_T	– Thrust coefficient
K_Q	– Torque coefficient
P_D	– Power delivered to the propeller
P_E	– Effective power from the propeller

Characteristic numbers

F_N	– Froude number
R_N	– Reynolds number
J	– Advance ratio

Robust fitting regression

<i>resid</i>	–	Residuals
<i>tune</i>	–	Tuning constant
<i>s</i>	–	Estimate of standard deviation
<i>h</i>	–	Leverage values from least square fit
<i>MAD</i>	–	Median absolute deviation

ANN

NN	–	Neural Network
AN	–	Artificial Neuron
ANN	–	Artificial Neural Network
FFANN	–	Feed Forward Artificial Neural Network
Performance	–	Matlab measure performance with MSE
MSE	–	Mean square error
RMSE	–	Root mean square error
R^2	–	Regression coefficient
x	–	Sample
X	–	Relative error [%]
\bar{X}	–	Mean error
σ	–	Standard deviation

Chapter 1

Introduction

When someone wants to build a new vessel today, they will normally start with having a design established that can fulfill the requirements of the ships intended purpose at the most affordable price.. In the process of proposing a design, the Naval Architect needs good tools that can estimate the performance of his design before the detailed drawings are made. It is of particular importance to have reliable tools when estimating the ships resistance and machinery power. The machinery plant of a ship is one of the main cost drivers when building a new vessel. As such, being able to estimate the installed power accurately for a certain design and speed, is very important to get a correct and cost efficient price for the new vessel.

Throughout history, there has been many ideas and theories presented for how a ships resistance could be estimated accurately. . Some of these methods produce good estimates, but without being updated along with modern theories, the confidence in their results weakens. Traditionally the methods have been based on mathematical regression, sometimes in a hybrid solution with a theoret- ically deduced element. More recently we have seen increased interest into utilizing the possibilities that comes with increased computer power related to estimating resistance. CFD and finite element methods have been developed, but these types of methods need an actual hull to perform calculations. In the early design phase the naval architect need a tool where he can test combinations of parameters (displacement, length, breath and draught etc.) to get a best possible starting point before designing the hull.

Artificial Neural Network (ANN) stands out as an interesting tool to use in making of a modern empirical method. The method is generic and identifies relations in complex problems very well, such as resistance estimation.. This thesis will investigate if ANN can be used to develop reliable estimation methods based on model test data of fishing vessels. The results will be compared to the existing methods to see if any improvements have been achieved.

Chapter 2

Previous work

This thesis fall into line with previously developed empirical methods. Many has tried to develop empirical methods to predict the necessary power for a ship in the early design phase, some has succeeded and are used today. It is important to remember that most empirical methods are developed for a narrow vessel category, and there exist a lot of methods for high speed vessels, multiple hulled boats and different categories of vessels. This thesis discusses traditional displacement vessels, with special attention to fishing vessels.

There are two empirical methods that are particularly well recognized and extensively in use today. One is the Hollenbach's method (Hollenbach, 1998) which was published in 1998, and considered reliable and with a broad area of application. The other major method is Holtrop's statistical method (Holtrop, 1984), with four publicly available revisions. Holtrop's method is kept up to date by the Netherlands model basin (MARIN) and are their property. It's newest revisions is therefore not open to public. From a project assignment from last semester (Kleppestø, 2014), I also discovered methods that are more specialized towards fishing vessels. The oldest is Oortmerssen's power prediction method (Van Oortmerssen, 1971), that is still useful for small vessels. Another method which is much simpler, but gives very good results, is the Digernes formula (Digernes, 1982). MARINTEK tested Artificial Neural Network methods in 2008, and made it available through their software package ShipX. This gave promising results and is an important reason why I have chosen to investigate this field further in a master thesis.

I have also found some empirical methods that predict propulsion coefficients. These are less known, and I have essentially used them to compare my results from the trained Artificial Neural Networks. They are presented in Section 5.2.

Chapter 3

Resistance theory

Today, there is consensus for the principles used to calculate a vessels hull resistance. The basis is Froude scaling of the resistance, which is represented with dimensionless coefficients that are defined as per Formula 3.1. This chapter will provide a brief presentation of how the resistance is divided into dimensionless coefficients and how these coefficients are used to scale the resistance. Formula 3.2 show how the total resistance coefficient is divided to account for different resistance components (Fathi et al., 2012, p. 56).

$$C_X = \frac{R_X}{0.5 \cdot \rho \cdot S \cdot V^2} \quad (3.1)$$

$$C_{Tot} = C_R + (1 + k_0)(C_F + \Delta C_F) + C_{AA} + C_{BC} + C_{App} + C_A \quad (3.2)$$

C_{Tot}	–	Total resistance coefficient
C_R	–	Residuary resistance coefficient
$1 + k_0$	–	Form factor
C_F	–	Frictional resistance coefficient
ΔC_F	–	Hull roughness allowance
C_{AA}	–	Air resistance coefficient
C_{BD}	–	Base drag coefficient
C_{App}	–	Appendage resistance coefficient
C_A	–	Correlation factor coefficient

3.1 Residuary resistance coefficient

The residuary resistance represents effects that are hard to distinguish as unique components. It mainly consist of the wave resistance which is a result of the wave pattern

(pressure distribution) around a moving hull. Another component is called the viscous-pressure resistance and is the resistance mainly resulting of the flow separation behind the hull. MARINTEK assumes that this component can be Froude scaled and can be included in the C_R . The C_R coefficient is identical for a model and a full scale vessel. The C_R coefficient is found in model tests, while all the other components can be calculated by using known theories. It is essential to establish a good way to estimate the residual coefficient as this has a major impact on the result when calculating the total hull resistance.

3.2 Form factor

The calculation of frictional resistance assumes a flat plate in a uniform flow, and it's important to compensate for the fact that a hull has curvature. The form factor is related to the friction resistance coefficient, and corrects for the fullness of the hull. The form factor can be established from model test, then Prohaskas method (ITTC, 2011) is often applied. In this thesis MARINTEKs approach is applied, and MARINTEK uses a self developed formula to find the form factor (Formula 3.3)

$$k_0 = 0.6\phi + 145\phi^{3.5} \quad (3.3)$$

$$\phi = \frac{C_B(L_{WL})}{L_{WL}} \cdot \sqrt{(T_{AP} + T_{FP}) \cdot B}$$

3.3 Frictional resistance

The frictional resistance consist of two components for full scale ships. The friction resistance for a flat plate, and the increased frictional resistance from surface roughness. The frictional resistance coefficient is calculated from the ITTC'57 correlation line (Formula 3.4), and represents the viscous resistance on a flat plate in turbulent flow. For full scale vessel, an increase in resistance is expected due to increased roughness (which is assumed to be zero for a polished model). This roughness allowance is formulated as Formula 3.5, where H is the surface roughness in μ (10^{-3} mm), which often is set to 150 as a standard value.

$$C_F = \frac{0.075}{(\log_{10} R_N - 2)^2} \quad (3.4)$$

$$\Delta C_F = [110.31 \cdot (H \cdot V)^{0.21} - 403.33] \cdot C_F^2 \quad (3.5)$$

3.4 Air resistance

The air resistance depends on the speed and size of the vessel's superstructure. It can be calculated with varying accuracy and details depending on the requirements and available resources. MARINTEK uses a fairly simple formula given as Formula 3.6. A_T is defined as the transverse projected area above the waterline. One could also deduce a drag coefficient for the superstructure, but it's hard to get a good accuracy without an independent study.

$$C_{AA} = 0.001 \cdot \frac{A_T}{S} \quad (3.6)$$

3.5 Base drag

For ships with a submerged transom stern, the resistance is increased due to suction. This suction force is a result of the external flow and the boundary layer around the hull. The base drag must be scaled to Reynolds number due to its dependency of the boundary layer. As long as the stern is wet, the base drag coefficient is expressed as Formula 3.7, where S_B is the area of the wetted transom stern.

$$C_{BD} = 0.029 \cdot \frac{(S_B/S)^{3/2}}{\sqrt{C_F}} \quad (3.7)$$

3.6 Appendage resistance

Most ships do have some kind of appendages which are not part of the hull, but will affect the resistance. Example of appendages are propeller shafts, shaft brackets, stabilizer fins, motion damping foils, tunnel thruster opening, bilge keels and rudders. Not all appendages are present on models, so the full scale appendage resistance coefficient is larger than for a typical model scale. The total appendage resistance is the sum of the resistance for all the included appendages. The appendage resistance coefficient is formulated as Formula 3.8, where w_{App} is the hull wake fraction at the position of the appendage, $C_{F,App}$ is the frictional resistance for the appendage and k_{App} is the form factor of the appendage (Steen, 2013). It is possible to find tables which estimate the resistance coefficient for different types of appendages.

The appendage resistance can be hard to calculate with desired accuracy. For this reason MARINTEK do not scale the appendage resistance separately. Instead it's included in the residual resistance, and scale effects are included in the correlation factor.

$$C_{App} = \frac{(1 - w_{App})^2 \cdot (1 + k_{App})}{S_{App}} \cdot C_{F,App} \quad (3.8)$$

3.7 Correlation factor

The correlation factor is a purely empirical coefficient which is used to compensate for the fact that not all effects present in the ships resistance are possible to identify and scale separately. The correlation coefficient depends on both the methods applied, and the test facilities used in the model test. In this thesis, test results and methodology from MARINTEK are applied and therefore also their correlation factor, C_A .

$$C_A = -0.228 \cdot 10^{-3} \tag{3.9}$$

Chapter 4

Propulsion theory

In this thesis I'm also interested in looking at the propulsion efficiency with the purpose of determining the necessary machinery power for the vessel. This chapter will define the total propulsion efficiency and its components. The total propulsion efficiency can be defined as per Formula 4.1.

$$\eta_D = \eta_0 \cdot \eta_H \cdot \eta_R \quad (4.1)$$

- η_D – Total propulsion efficiency
- η_0 – Open water efficiency
- η_H – Hull efficiency
- η_R – Relative rotative efficiency

4.1 Open water efficiency

The open water efficiency is the propeller efficiency found from an open water test in a test basin. It indicates the optimum efficiency that the propeller can deliver in a uniform and laminar flow. The open water efficiency can be defined as per Formula 4.2, where P_D is the power delivered *to* the propeller and P_E is the effective power delivered *from* the propeller. The open water efficiency depends on the J-number and is often expressed in an open water diagram together with K_Q and K_T .

$$\eta_0 = \frac{P_E}{P_D} \quad (4.2)$$

4.2 Hull efficiency

The hull efficiency, quantifies the effect from the hull on a working propeller's efficiency. It is found in a self-propulsion test, where a model of the propeller delivers the necessary

thrust needed to run the ship model at a desired speed (except for a tow force that is applied to compensate for the increased friction resistance in full scale). The hull efficiency is defined as per Formula 4.3. The effects that are included in this efficiency, wake fraction and thrust deduction, are both described in more detail in Section 4.3 and 4.4.

$$\eta_H = \frac{1 - t}{1 - w} \quad (4.3)$$

4.3 Wake fraction

When a propeller is placed behind a hull, the flow velocity through the propeller disc is less than the speed of the vessel (free flow velocity). This speed difference is called wake, and is defined as per Formula 4.4. This reduction in flow velocity is recognized to have three causes: Potential wake, Frictional wake and Wave wake (Molland et al., 2011, p. 146) and (Harvald, 1983, p. 152). The reduced speed over the propeller disc increases the propeller efficiency.

The wake fraction used in this thesis, is deduced from the propulsion test and the open water diagram. The expected thrust at a given flow velocity (model speed) are known, and the actual thrust is measured. One can easily deduce the speed difference (ΔJ) that separate the two thrust values. This difference in speed is used to find the effective wake.

$$w = \frac{V - V_A}{V} \quad (4.4)$$

4.4 Thrust deduction

The thrust required to move a vessel at a certain speed is greater than the rope force needed to tow the vessel at the same speed. This means that a working propeller increases the resistance on the hull. This is referred to as thrust deduction (Defined in Formula 4.5). This increase in resistance is mainly caused by the fact that a propeller accelerates the flow speed ahead of itself. This causes two effects: (1) The increase in flow velocity at the stern of the vessel, results in increased shear forces, the same as more frictional resistance and (2) The increase of velocity reduces the pressure (explained by the Bernoulli equation) which can be explained as a suction force on the stern (Molland et al., 2011, p. 145). This implies that the thrust deduction reduces the propeller efficiency.

The thrust deduction used in this thesis is found from the propulsion test, where the trust measured on the propeller shaft, and the resistance are known from a towing test is used.

$$t = \frac{T - R_{Tot}}{T} \quad (4.5)$$

4.5 Relative rotative efficiency

The relative rotative efficiency compensate for the efficiency loss that is caused by other disturbances in the wake field. When thrust identity is assumed, the relative rotative efficiency can be expressed as per Formula 4.6. Mainly two components are present (Molland et al., 2011, p.145) (1) Turbulence in full scale is much more severe than in model scale because of the difference in Reynolds number. This will have negative effect on the propeller efficiency (2) The flow behind the hull is non-uniform and the propeller gives the flow rotation which is left as lost energy behind the propeller. This will also have a negative effect on the propeller efficiency. To what extent. will depend on how optimized the propeller is for the given wake situation and if there are any contra-rotating devices installed around the propeller.

$$\eta_R = \frac{K_Q}{K_{Q0}} \tag{4.6}$$

Chapter 5

Empirical methods

As a reference in this thesis, I have chosen some common empirical methods to compare the results from the developed Artificial Neural Networks. The described methods are regarded as reliable and are often used in an early design phase. I have divided the empirical methods in two groups; (1) methods for predicting ship resistance and (2) propulsion and hull efficiency.

5.1 Resistance prediction methods

Many different methods have been developed for prediction of ship resistance. While some methods calculate the total resistance, others calculate the residual coefficient. To compare the methods on the same premises, I have calculated the total resistance based on the ITTC procedure by dividing the resistance in several components that can be calculated based on the ship geometry as seen in Equation 5.1.

$$C_{Tot,s} = (1 + k_0) \cdot (C_{F,s} + \Delta C_F) + C_R + C_{AA,s} + C_{DB,s} + C_{App,s} + C_A \quad (5.1)$$

5.1.1 Digernes formula

Digernes formula is a pure regression formula (Formula 5.2), which was formulated at MARINTEK in 1982. This formula is based on the experience showing that resistance, to a large extent, is decided by the displacement and speed of the vessel (Digernes, 1982). By developing an equation that is logarithmic linear, one gets the possibility to apply linear regression to find the coefficients. There are also some other geometrical relations added which are assumed to influence the resistance. The values for the coefficients are listed in Table 5.1.

$$R_{Tot} = a \cdot \left(\frac{L_{WL}}{B}\right)^b \cdot \left(\frac{B}{T}\right)^c \cdot \nabla^\delta \cdot e^{\beta' \cdot F_n} \quad (5.2)$$

Table 5.1: Coefficient values for Digernes method, (Formula 5.2)

a	b	c	δ	β'
$2.956 \cdot 10^{-4}$	0.802	0.745	1.113	15.605

Table 5.2: Field of validity for Digernes method as it is defined by Digernes

$L_{WL}[m]$	$B[m]$	$T[m]$	$\nabla[m^3]$	L_{WL}/B	B/T	F_N
11.9–53.9	3.55–13.6	0.68–5.65	17.6–1376	1.8–4.5	1.9–10.0	0.223–0.494

This regression is based on resistance results from 33 model tests, and later controlled against 280 other vessels (Digernes, 1983). The regression coefficient for this regression is found to be $R^2 = 0.995$, which is very impressive. A regression coefficient value close to 1, indicates that this equation is able to follow the curvature of resistance curves very well for the vessels included in the original dataset. It's worth to mention that the dataset consist of vessels without bulb.

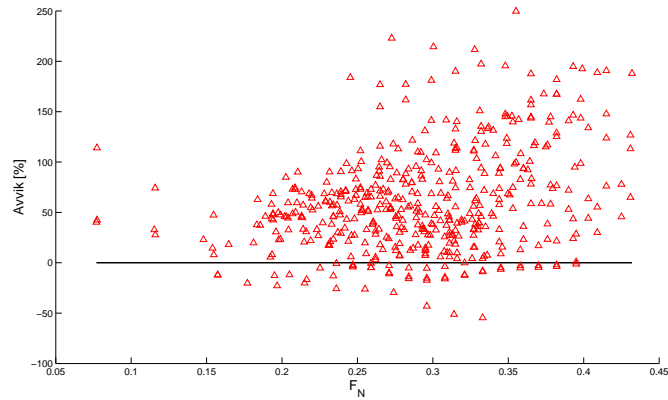
Field of validity

Digernes defines the field of validity for his method based on six geometrical parameters and one Froude number interval (see Table 5.2). These are the outer limits for the original dataset. The method is based on fishing vessels only, and this characterizes the field of validity, and especially the size of the vessels.

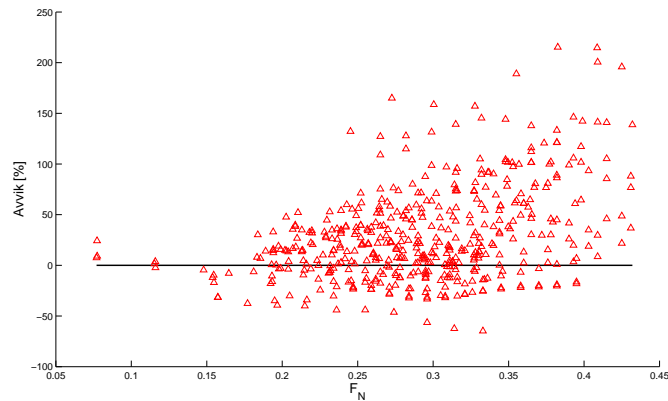
In this assignment, I have made an updated regression for Dignernes Method which is described in Section 8

5.1.2 Hollenbachs resistance estimate

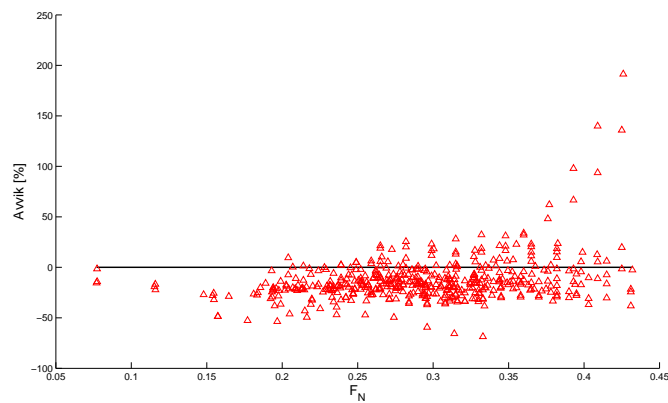
Hollenbachs resistance estimate is a modern method released in 1998, and is widely used by Naval Architects today. This method is primarily made for merchant vessels, but with a wider field of validity than Holtrop. It's developed from model results at the Wiena towing-tank. The method is quite comprehensive and utilizes a larger dataset and more complex formulas than the other methods reviewed in this thesis. This method separate the best and poorest 5% of the results, and gives the user the possibility to decide whether the specified vessel have good or poor hull lines. This is related to the amount of constrains a vessel are subject to in the design process (Hollenbach, 1998) and how optimized the hull is for minimum resistance. Because this method is primarily designed for merchant vessels, it's hard to tell which type of hull lines apply for fishing vessels. These modes are tested on a group of fishing vessels (Kleppestø, 2014) and it was clear that fishing vessels should use the «minimum» resistance option. Figure 5.1 show the results for the three options. Fishing vessels do have extensive constrains regarding stability and payload, so this finding is somehow surprising, but may indicate that the hull lines are good *relative* to their extreme hull coefficients (i.e. B/L , C_B).



(a) Maximum Resistance



(b) Mean Resistance



(c) Minimum resistance

Figure 5.1: Hollenbach's modes for «maximum», «mean» and «minimum» resistance, is tested on fishing vessels. Error between formula and model tests are plotted against the Froude number. The horizontal black line indicates no error (0%) and y-axis range is from -100% to 250%, where negative error indicates underprediction of resistance. (Kleppestø, 2014)

Table 5.3: Field of validity for Hollenbach’s method as they are defined for the «minimum resistance» mode (Hollenbach (1998))

	$L_{PP}[m]$	$L_{PP}/\nabla^{1/3}$	C_B	L/B	B/T	L_{OS}/L_{WL}	L_{WL}/L_{PP}
Single Screw	42–205	4.49–6.01	0.60–0.83	4.71–7.11	1.99–4.00	1.00–1.05	1.00–1.06
Twin Screw	30.6–206.8	4.41–7.27	0.51–0.78	3.96–7.13	2.31–6.11	1.00–1.05	1.00–1.07

Field of validity

Hollenbachs resistance estimate is an advanced prediction method that combines several empirical methods and require relatively many input parameters. This gives a quite generous field of validity. Even so, Table 5.3 indicates that the limits are designed for merchant vessels and that the L/B and $L_{pp}/\nabla^{1/3}$ limits are not suitable for fishing vessels.

5.1.3 Holtrop statistical method

Holtrops statistical analysis of performance results are more than ten years older than Hollenbachs method, but is still widely used today. The method is also developed for merchant vessels, which utilize other types of hull-forms than fishing vessels. This method is based on model tests and full scale tests from the Nederlands Ship Model Basin. It uses the same division of the total resistance as ITTC (as explained in Section 3), the total resistance can then be formulated as Formula 5.3.

$$R_{Total} = R_F(1 + k_1) + R_{APP} + R_W + R_B + R_{TR} + R_A \quad (5.3)$$

- R_F = Friction resistance according to ITTC-1957 formula
- $1 + k_1$ = Form factor of the hull
- R_{APP} = Appendage resistance
- R_W = Wave resistance
- R_B = Additional pressure resistance of bulbous bow near the water surface
- R_{TR} = Additional pressure resistance due to transom immersion
- R_A = Model-Ship correlation resistance

As a result of where Holtrop includes viscous-pressure resistance in the resistance, it’s a difference in the form factor of Holtrop versus the one from MARINTEK. Holtrop presents a resistance component for the wave resistance (R_W) alone and includes the viscous pressure resistance in the form factor. In contrast, MARINTEK base their form factor on Prohaskas method, and includes viscous pressure resistance together with the wave resistance in the residual resistance coefficient. Because of this, one should not compare the resistance coefficients directly, but rather look at the total resistance which should be comparable.

Table 5.4: Field of validity for Holtrops statistical method for fishing vessels, as it is defined in his first article (Holtrop, 1977)

L_{WL}/B	C_P	$F_{N,maks}$
3.9–6.3	0.55–0.65	0.38

Holtrops statistical method apply some uncommon input values, which are not always available in the initial design phase. This might make the method somewhat harder to use, but with some experience it should be possible to estimate those values without affecting the results too much. Some sources also recommend not to use the bulb option this method offer.

As a final remark, I want to point out that Holtrops statistical method for resistance prediction has been upgraded several times. The 1984 version is the latest version publicly available. Newer versions of this method is the property of Nederlands Ship Model Basin (MARIN), and has not been published.

Field of validity

Since Holtrops statistical method is not made for, or from, fishing vessels, it is not expected that the field of validity cover typical fishing vessels. This method want to be a general method, and the field of validity is given in the first (of four) article (Holtrop, 1977), and then specified for fishing vessels. How the field of validity has changed through the later revisions is not commented, and I assume that it hasn't changed. Holtrop defines the field of validity with only a few parameters, but they are non-dimensional and give a good representation of the hull types that are used in the development of the method.

5.1.4 Oortmerssen power prediction method

Oortmerssen method for power prediction (Van Oortmerssen, 1971) is the oldest resistance prediction method in this thesis. The method is based on both a mathematical derivation and regression to a data material of fishing vessels and tugs. The formula and input parameters are deduced from the wave resistance for a short vessel. Coefficients are calculated with regression of a data material consisting of 93 vessels that are random fishing vessels and tugs. The method is somewhat similar to Digernes formula, but the approach is more advanced.

Oortmerssen tries to deduce his own formula for the form factor based on Granvilles (Granville, 1956) empirical formula. The fundamental idea is that the form factor, to a large extent, depends on the relationship $\frac{C_B}{L/B}$, and that this is a good starting point. The deduced formula is seen in Formula 5.4. The meaning of the parameter ψ is illustrated in Figure 5.2, while C is empirically determined from model tests. The recommended value for C is 139. Oortmerssen concludes himself that the form factor does not improve the

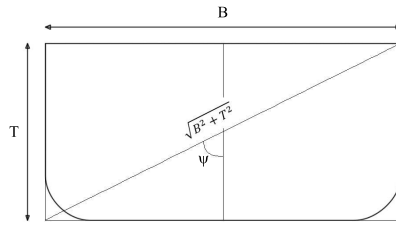


Figure 5.2: Oortmerssen uses the diagonal of the hull to calculate his form factor for a vessel. This figure illustrated how two of the coefficient is to be found

results, and removes it from the final formula. For a modern group of fishing vessel, the form factor does actually increase the errors (Kleppestø, 2014).

$$\begin{aligned}
 k_0 &= C \cdot \left[\frac{C_B}{L_D / \sqrt{B^2 + T^2}} \right]^2 \cdot \sin^2 2\psi \\
 &= C \cdot \frac{C_B^2}{\left(\frac{L_D}{B}\right)^2 \cdot \left[\left(\frac{B}{T}\right)^2 + 1\right]} \quad (5.4)
 \end{aligned}$$

The residual resistance of Oortmerssen is deduced from the wave pattern along the hull. Oortmerssen has deduced the wave resistance for a wave pattern along a two dimensional pressure distribution that he uses as a basis for his formula. This method is made from small ships, so a simplification is applied by removing the parallel midship in the deduction. He expresses the divergent wave pattern in a polynomial, before he simplify the whole equation to make a better fit to the data material.

When I tested this method on a modern group of fishing vessels, it seems like the applied formula contains an error. This assumption is based on two facts. (1) I cannot get the same answer for the example calculation given in the article. (2) The errors seem to have a noticeable «bump» in a Froude number interval as illustrated in Figure 5.3. My best guess, after some back tracing, is that one of the coefficients are misprinted in the article. I have tried, but not succeeded, in finding experienced naval architects that are familiar with this method which I could compare my results with. Mainly because of the age of this method (44 years), it's not well known anymore and seldom used.

Field of validity

This method is deduced to predict resistance for tugs and small fishing vessels (Van Oortmerssen, 1971), and it's a result from mathematical deduction validated on model tests. Since the method is published in 1971, one should also be aware that design trends has changed considerable since then. Oortmerssen does not himself define a field of validity for his method, but gives bar charts for his data material, and based on this I have formulated a field of validity that can be seen in Table 5.5.

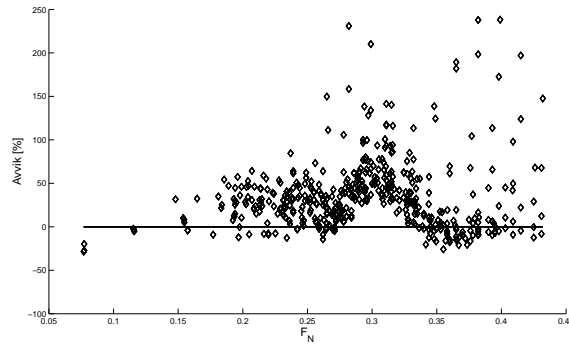


Figure 5.3: Percentage error plotted against Froude number for Oortmerssen method. Zero error is marked with a horizontal line, and y-axis goes from -100% til 250% (Kleppestø, 2014)

Table 5.5: Field of validity for Oortmerssen method which is based on the bar charts of the data material as it's given in the article (Van Oortmerssen, 1971)

$L_D[m]$	$LCB[\%]$	B/T	L_D/B	$\nabla[m^3]$	$S[m^2]$	C_P	C_M
16–72	-4.2–7	1.9–2.95	3.4–6.2	0–2850	0–1200	0.55–0.7	0.76–0.94

5.2 Propulsion prediction methods

There are empirical formulas that estimate propulsion coefficients, but they are not as well known as the resistance methods. This section presents a selection of methods that I have tested against my data material.

As for the resistance methods, one must keep in mind that this thesis deal with modern fishing vessels with length 30-90m and does not always hold the same properties as traditional merchant vessels. For propulsion calculations, I must emphasize that all the vessel in this thesis are equipped with ducted propellers, which is quite uncommon for other vessel types than fishing vessels and tugs.

For the propulsion prediction methods, I have typed the formulas in the document, this in contrast to the resistance methods. There are two reasons for this (1) The wake and thrust deduction formulas are much simpler and therefore easier to include in the text, and (2) Wake and thrust deduction formulas are not so commonly known today, so it may be harder for the reader to find the formulas on his own.

5.2.1 Wageningen B-series

Wageningen B-series are a comprehensive and systematic propeller series that are frequently used to find an early estimate for the propeller geometry. The series consist of more than 120 propellers with a systematic change in the propeller geometry. In this thesis I take advantage of the fact that open water diagrams are already established for

Table 5.6: Field of validity for Wageningen B-series

Z	A_E/A_0	P/D
2–7	0.3–1.05	0.5–1.4

this series, and the propeller efficiency can be estimated. A regression analysis has been performed, so that the results are interpolated between the tested propellers (Bernittsas et al., 1981). This polynom makes it simple to estimate the open water propeller efficiency for the actual propellers in my data material.

Field of validity

The field of validity are the limits of the propeller series tested, and can be seen in Table 5.6. Because of the polynom that are applied, one must be careful when approaching the boundaries of the series.

5.2.2 BSRA Propulsion factors

The British Ship Research Association conducted a methodical series of vessels (Parker and Eng., 1966), which they later used to develop both a resistance method and an equation for wake estimation. The resistance method gave so poor results for my data material that only the wake formula is included. The series represent a systematic variation of hull proportions and LCB for a range of block coefficients from 0.65 to 0.80. The full scale ships are all scaled to the same dimension; $L_{PP} \times B \times T = 400\text{ft} \times 55\text{ft} \times 26\text{ft} = 122\text{m} \times 16,8\text{m} \times 8\text{m}$.

The equation is the result of a linear regression where the authors have identified hull parameters that have the most impact on the thrust and wake. These are the block coefficient, Froude number and a «home made» coefficient D for both wake and thrust. The longitudinal position of buoyancy (LCB) forward of amidships is also included in an intricate manner which is meant to illustrate how much the position of LCB deviates from the basis position forward of amidships. These three parameters can be seen in Formula 5.5

$$D_w = \frac{B}{\nabla^{1/3}} \cdot \sqrt{\frac{\nabla^{1/3}}{D}} \quad (5.5)$$

$$D_t = \frac{B \cdot D}{\nabla^{2/3}} \quad (5.6)$$

$$\delta LCB = LCB - 0.2 \cdot (C_B - 0.675) \cdot L_{PP} \quad (5.7)$$

In the same journal there is a discussion where the complex formula is criticized and a simpler version is suggested with similar performance. It's this simplified version that is later adapted by Prof. Knut Minsaas in his ship propulsion compendiums. I will present the simplified version (because this seems to be the one commonly used). The

simplified formulation has been fitted for two independent data sets, and therefore two sets of coefficients are found, as seen in Formula 5.8 and 5.9.

$$w = -0.478 + 0.79 \cdot C_B - 0.10 \cdot \frac{V}{\sqrt{L}} + 0.170 \cdot D_w - 0.0072 \cdot LCB \quad (5.8)$$

$$w = -0.508 + 0.67 \cdot C_B - 0.05 \frac{V}{\sqrt{L}} + 0.234 \cdot D_w - 0.0028 \cdot LCB \quad (5.9)$$

Field of validity

There is not much emphasis on this method's field of validity, so the user should know that the data set consists of ocean going vessels with a single propeller tested before 1966. For Formula 5.8, the dataset is specified as merchant vessels with a Block Coefficient of $C_B \in [0.65, 0.8]$, while the dataset for Formula 5.9 consists of unspecified vessel categories with a block coefficient of $C_B \in [0.55, 0.8]$. Since I'm not familiar with these formulas, I have included both in this thesis and tested them to my data material.

5.2.3 Harvald graphical method for propulsion factors

Harvald developed a prediction method for the wake fraction and thrust deduction in 1949, from 200 model experiments, consisting of single-screw ships carried out at the model basin in Wageningen (Harvald, 1983, p. 165-186). The derivation of this method is based on the theory that one can divide the wake coefficient into three separate parts, like it's common to do with resistance. Because the definition of the total wake coefficient are straight forward (Formula 4.4), this should not have any big impact. I will later give a brief explanation of the wake coefficients that Harvald base his diagrams on.

In the development of this method, Harvald tried to find which coefficients that the wake fraction depends most on. He sorted his data material by block coefficient and breath-to-length ratio which are the two most important parameter for the potential wake (Figure 5.4). This was not enough for a method in itself because of the extensive scattering of the points.

[...] the scattering of the points was so extensive that it was extremely difficult to draw mean curves. (Harvald, 1983, p. 166)

The two lower diagrams in Figure 5.4 try to identify the cause of this extensive scattering by sorting the data further. The third diagram tries to quantify the frictional resistance based on boundary layer calculations, by normalizing the material to $D_P/L = 0.04$. The middle diagram is a correction for ships with pronounced U- or V-shaped sections. He also attempted to make corrections based on the B/T or T/L ratios, without success.

Harvald also try to make a method to estimate trust deduction factor (see Formula 4.5). He looks for a relationship between wake and thrust deduction, but cannot find any valid generalization for this relationship. Because thrust and wake should be

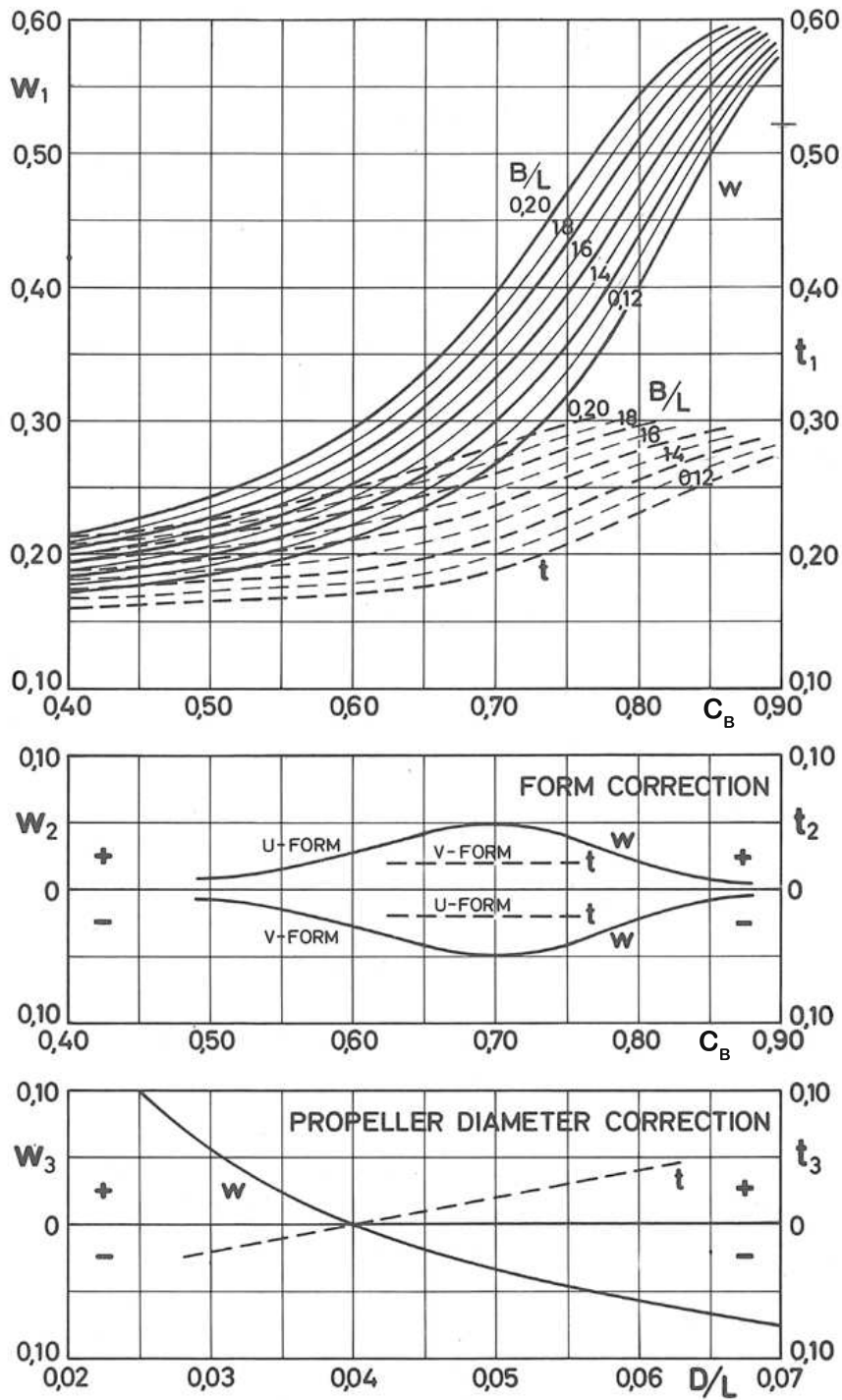


Figure 5.4: Harvalds graphical representation of the wake and thrust deduction factor for single-crew ships: $w = w_1 + w_2 + w_3$; $t = t_1 + t_2 + t_3$ (Harvald, 1983, p. 166)

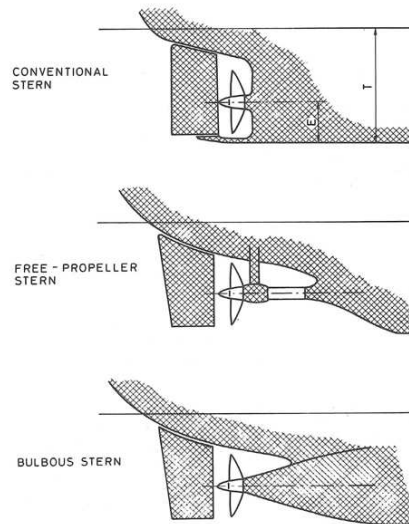


Figure 5.5: Propeller arrangements as defined by Harvald to estimate the thrust deduction (Harvald, 1983, p. 184)

correlated per definition, he presents an individual method to estimate the thrust deduction that seems to have much in common with the wake method. It is therefore presented in the same diagram as the wake method (Figure 5.4). But as the thrust deduction is more dependent on the propeller arrangement, the curves are only valid for a conventional stern (defined in Figure 5.5). For a Free-propeller stern the thrust deduction is divided in half ($\Delta t = -0.5t$), and for bulbous stern it is reduced by a quarter ($\Delta t = -0.25t$). Because it's impossible for me to identify the stern shapes in my data material in this thesis, Harvalds method for thrust deduction is not investigated further.

Wake components

As commented earlier, Harvald divides the wake coefficient into three components. Where w_P , w_F and w_W are the potential wake fraction, the friction wake fraction and the wave wake fraction.

$$w = w_P + w_F + w_W \quad (5.10)$$

The potential wake is the wake obtained in an ideal fluid (potential flow) without any wave making. This should be possible to calculate mathematically by using potential theory. *The wave wake* comes from the particle movement in the wave system from the vessel, which is stationary relative to the propeller. *The frictional wake* is the wake that is caused by effects from friction on the afterbody of the vessel. This is assumed to be the main component, while at the same time hard to quantify because it's not certain how much of the boundary layer that needs to be taken into account.

Table 5.7: Field of validity for Harvalds graphical method which is based on the presentation of the data material as it is indicated in (Harvald, 1983). Valid for both wake and thrust deduction

C_B	B/L_{OS}	D_P/L_{OS}	B/T	T/L_{OS}
0.4–0.9	0.12–0.2	0.02–0.07	2.4–2.5	about 0.06

Harvald criticize this way of dividing the wake fraction, so I will spend no more time discussing this method, and finish the discussion with quoting him:

This method of definition is the one mostly employed, but it is far from good, as it is uncertain how much of the frictional belt must be taken into account. (Harvald, 1983, p. 166)

In this thesis I have performed a mathematical curve fit for the curves to make calculations easier for the large amount of vessels in the data material.

Field of validity

There are not given any specific field of validity for this method. As a graphical method, one should not try to extrapolate the curves as this probably indicates quite well the field of validity. As mentioned, the data material consists of single-screw merchant vessels. Harvald also gives a quite narrow B/T and T/L interval for this method, as seen in Table 5.7. For thrust deduction the field of validity is the same, but in addition the stern shape is important, and three propeller configurations are defined as seen in Figure 5.5.

5.2.4 Holtrop estimate of propulsion factors

Holtrop developed a statistical method for propulsion factors, and together with the resistance this has been updated several times. I have used the latest published article (Holtrop, 1984). This method is quite advanced compared to the other propulsion methods with many terms (Formula 5.11). Since this is a statistical regression, the connection to real life phenomena's are not considered in other ways than in the choice of input coefficients. This gives the impression to be messy and difficult to use, but this is not the case when using computers. This method is hard to use with a pocket calculator, but with a simple computer script or excel spreadsheet it is very simple to reuse this formula for many vessels.

$$w = C_9 \cdot C_{20} \cdot C_V \cdot \frac{L}{T_A} \left(0.050776 + 0.93405 \cdot C_{11} \cdot \frac{C_V}{1 - C_{P1}} \right) + 0.27915 \cdot C_{20} \sqrt{\frac{B}{L(1 - C_{P1})}} + C_{19} \cdot C_{20} \quad (5.11)$$

Where the coefficients are defined as:

$$\begin{aligned}
 C_8 &= \begin{cases} (B \cdot S)/(L \cdot D \cdot T_A) & : B/T_A < 5 \\ S \cdot (7B/T_A - 25)/(L \cdot D(B/T_A - 3)) & : B/T_A > 5 \end{cases} \\
 C_9 &= \begin{cases} C_8 & : C_8 < 28 \\ 32 - \frac{16}{C_8 - 24} & : C_8 > 28 \end{cases} \\
 C_{11} &= \begin{cases} T_A/D & : T_A/D < 2 \\ 0.0833333(T_A/D)^3 + 1.33333 & : T_A/D > 2 \end{cases} \\
 C_{19} &= \begin{cases} 0.12997/(0.95 - C_B) - 0.11056/(0.95 - C_P) & : C_P < 0.7 \\ \frac{0.18567}{1.3571 - C_M} - 0.71276 + 0.38648 \cdot C_P & : C_P > 0.7 \end{cases} \\
 C_{20} &= 1 + 0.015 \cdot C_{stern} \\
 C_{P1} &= 1.45 \cdot C_P - 0.315 - 0.0225 \cdot LCB \\
 C_V &= (1 + k)C_F + C_A \\
 C_A &= 0.000352
 \end{aligned}$$

Holtrop has also formulated an equation for the thrust deduction that is much simpler than the wake formula (Formula 5.12)

$$\begin{aligned}
 t &= 0.25014 \left(\frac{B}{L}\right)^{0.28956} \left(\frac{\sqrt{BT}}{D}\right)^{0.2624} \\
 & / (1 - C_P + 0.0225 \cdot LCB)^{0.01762} + 0.0015 \cdot C_{stern}
 \end{aligned} \tag{5.12}$$

Field of validity

As for the resistance prediction method, Holtrop only gives a field of validity in his first article from 1977. Even if the formulas are revisited in the later articles there are given no or little information of the types of ships these updates are based on. The field of validity is assumed to be the same as for the resistance method which are shown in Table 5.4.

5.2.5 Taylor wake formula

The simplest formula for wake estimation I will present is Taylors formula (Formula 5.13). This formula is very old (1910) and not very well explained in the original source (Taylor, 1910, p. 201). As such, I can not give much information related to this formula. But with its simple appearance, a deep understanding is not necessary. However, Taylor does note that this method has a tendency of over predicting the full scale wake fraction.

$$w = 0.5C_B - 0.05 \tag{5.13}$$

Field of validity

There is not much information about the field of validity, but it is made for single-screw ships. It is also noted that the formula is derived from model tests where some are of the bare hulls where the propeller configuration does not interfere with the natural flow of water (i.e. vessels fitted with struts or bossing).

Chapter 6

ANN theory

6.1 Introduction

McCulloch and Pitts took the first step towards Artificial Neural Networks in 1943, when they devised a simple mathematical model of a neuron (Fausett and Hall, 1994, p. 22). Neurons are the building blocks in a neural network, and they process the input-data through the network. The fundamental idea behind a neural network is to replicate the complex network of biological neurons in the human brain and its ability to solve complex non-linear problems in an efficient manner.

Today, there exists over thousand artificial neural network models, and more than forty are considered primary ones (He and Xu, 2010, p. 4). The reason that there are so many possible ways to design an ANN is the simple nature of a neuron. Because a neuron can process an unlimited number of signals, the network designer can choose which signals that should be sent to each neuron and make recursive and feedback loops in the network. By utilizing feedback loops one can design a network which is capable of learning from each calculation. Some of the problems that an ANN can solve are; pattern recognition, system simulation, automatic control and curve fitting.

The human brain is made up of many relative simple neurons which are connected in a complex network. By utilizing parallel computation this makes it capable of solving very complex problems with a relative small amount of energy. The human brain has approximately $10^{10} - 10^{11}$ neurons. Each biological neuron is interconnected with $10^3 - 10^5$ other neurons (including itself). By taking into the fact that that interconnections between the neurons is in constant change through human learning and experience, one can get the idea of how powerful and flexible the human brain is. The switching time of a biological neuron is several milliseconds (in the order of 10^{-3} s), which is millions of times longer than that of a current computer (in the order of 10^{-10} s). Even so, the human brain can produce an accurate response to a complex task in less than one second This is much faster than a modern computer. This illustrates that although the processing and transmission-speed of a single neuron is rather slow, the brain can respond quickly due to its high parallelism. (He and Xu, 2010, ch. 2.1).

One of the ANNs many strengths is that the user don't need a mathematical under-

standing of the problem he needs to solve. In contradiction to making regression and simulation with traditional methods, a user making use of ANN, don't need to know how these relations are formulated as long as he is aware of the relations in the problem and has an accurate dataset. With the large number of neurons, comes the advantage of parallel computation which the network makes easy to utilize. However, the black box approach of ANNs means that the user only sees the results, and not how the network has chosen to model the relations. This can make some advanced users skeptical in embracing the method since one cannot supervise the computation.

Because ANN uses association in solving problems, and, as many other mathematical methods, do not look for an exact recognition, it is capable of solving more diffuse problems, like handwriting recognition. Since handwritings are quite unique, but at the same time similar, they are tough problems to solve with traditional methods. An ANN will look after if a letter «looks like» an a or b etc. not if it matches a predefined value (letter).

6.2 The neuron

Figure 6.1 compare the artificial neuron to a stylized biological neuron. The figure show that the fundamental structure is somehow equal, while the biological version is more advanced as it can mix several inputs to a joint input and have the bias value dependent on inputs. In this text, we will concentrate on the artificial neuron and its characteristics.

The artificial neuron is built up by five elements; inputs (p_i), bias weights ($w_{i,j}$), bias value (b), activation function ($f(n)$) and output value (a). The neuron can receive an unlimited number of input signals, where each has an unique bias weight assigned from training. All the weighted input signals are summed up and a bias value is added to the sum. This sum is then used as input to an activation function that create a normalized output value from the neuron.

The bias weight represents the relative effect each input has on the output value. The bias is unique for each input and neuron, and the same input signal has different weights for each neuron. The bias weight can be zero or negative. The bias value is added to the weighted sum, and its purpose is to shift the value of the sum before it is fed into the activation function. To change the bias value is equivalent of sliding the activation function sideways, which gives the possibility to give an output value of zero without the weighted sum of inputs being zero. Because the activation function normally is symmetrical, the bias value is essential to decide the zero-output point (Kohl, 2010). The activation function is used to normalize the output value from the neuron. The function can be discrete, linear or of other shapes. In this assignment, I use the common sigmoid function (Equation 6.1) which normalize the output between $[-1, +1]$. The sigmoid function are plotted in Figure 6.2a.

$$F(n) = \frac{2}{1 + \exp(-2n)} - 1 \quad (6.1)$$

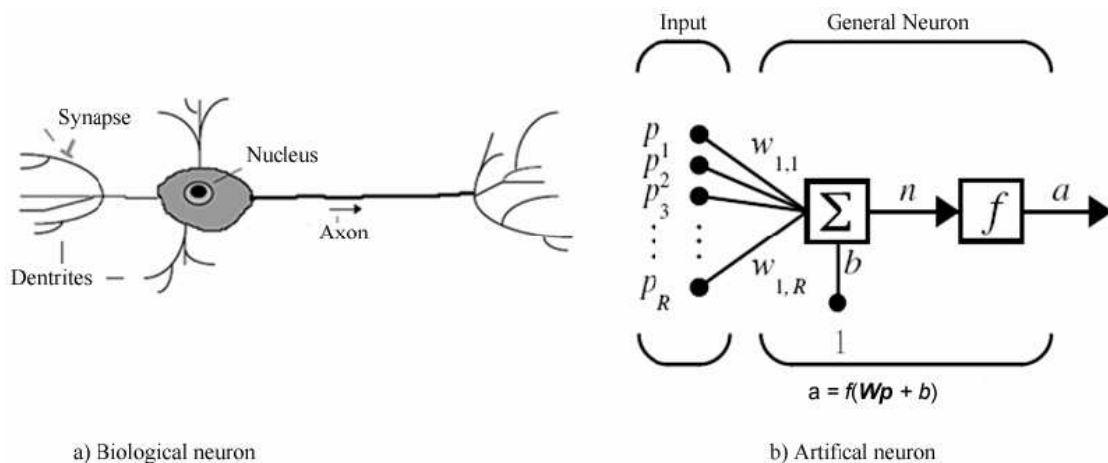


Figure 6.1: Comparison of a stylized biological and an artificial neuron (Ünal et al., 2013, ch. 1)

This structure gives a very flexible neuron that can communicate with many other neurons in different ways and networks. It can, as stated, process signals from an unlimited amount of other neurons and share its output value with an the same number of neurons. Because the weights for each neuron is decided individually, the neuron can handle both discrete and continuous values. By assigning a discrete activation function (a step function), the problem can be made fully discrete as well. It's common to standardize the neurons in a network with one activation function, and individual values for the bias values and weights.

6.3 Network structure

The way neurons interact with each other defines the network structure. There are many ways to design a network, and finding the optimum structure depends on the type of problem at hand. The network design needs to define include the number of neurons and how they are distributed in the network. Complex problems needs more neurons and a more advanced structure.

Two common neural networks are (1) feedforward and (2) recurrent networks. In this assignment I use feed forward artificial neural network (FFANN), which is the most common signal path and works well for pattern recognition (Ünal et al., 2013). In a FFANN all the signal paths go the same way (i.e. from the inputs towards the outputs). In this one-way signal flow, the neurons are structured in layers (input-, output- and hidden layers) and each output value from the previous layer is sent to every neuron in the next layer. This creates a lot of possible combinations to process the input values into the desired output values. The structure of a FFANN is illustrated in Figure 6.3.

Recurrent network is a signal feedback network which uses a neurons output signal as one of its input signals. This gives the system a dynamic behavior with a time delay

for the memory processing. A Recurrent network is suitable for simulation problems and regulators. A recurrent network is able to make decisions based on the real time situation and experience. An example is backing up a trailer, where a recurrent network can assess the state of the trailer (position of the cab of the truck, the position of the rear of the trailer, the (fixed) position of the loading dock, and the angles that the truck and the trailer make with the loading dock) to steer the trailer in place. As for a human, an untrained network will perform this task poorly, but will gradually improve with training when it learns how the trailer react on steering (Fausett and Hall, 1994, ch. 1.3). This is a problem where traditional methods will need to calculate many complete paths of steering the trailer for then to find feasible and optimum solutions, while a recurrent network takes real times decisions based on the situation.

For a FFANN, the number of neurons and hidden layers is equivalent to the number of Degrees of Freedom (DOF). If we increase the number of neurons in the hidden layer(s), it gives the possibility to generate more complex functions, but this also increase the risk of overfitting the network and decrease its generalization ability.

«Each sigmoid function represents a soft threshold in its input space, as shown in Figure 6.2a. With one hidden layer and one output layer, each output unit computes a soft-thresholded linear combination of several functions. For example, by adding two opposite-facing soft threshold functions and thresholding the result, we can obtain a "ridge" function as shown in Figure 6.2b. Combining two such ridges at right angle to each other (i.e. combining the outputs from four hidden units), we obtain a "bump" as shown in Figure 6.2c.

With more hidden units, we can produce more bumps of different sizes in more places. In fact, with a single sufficiently large hidden layer, it is possible to represent any continuous function of the inputs with arbitrary accuracy; with two layers, even discontinuous functions can be represented. (The proof is complex, but the main point is that the required number of hidden units grows exponentially with the number of inputs. For example, $\frac{2^n}{n}$ hidden units are needed to encode all Boolean functions of n inputs). Unfortunately, for any particular network structure, it is harder to characterize exactly which functions can be represented and which one cannot.»

(Russel and Norvig, 2010, p. 732)

Even though the fact that a complex problem needs a complex network structure to be reproduced, I have not found any sources that gives a good rule of thumb to how large the network should be for certain complexity of problems. The general advice seems to be the trial-and-error approach for each problem one wants to solve by dividing the dataset into two groups, one for training the network and one to validate the trained network. A poor fit to the training group calls for a larger network structure, while a poor fit to the validation group indicates overfitting and calls for a simpler network structure. The complexity can be varied by changing the number of hidden layers, the total number of neurons and the distribution of neurons over the hidden layers.

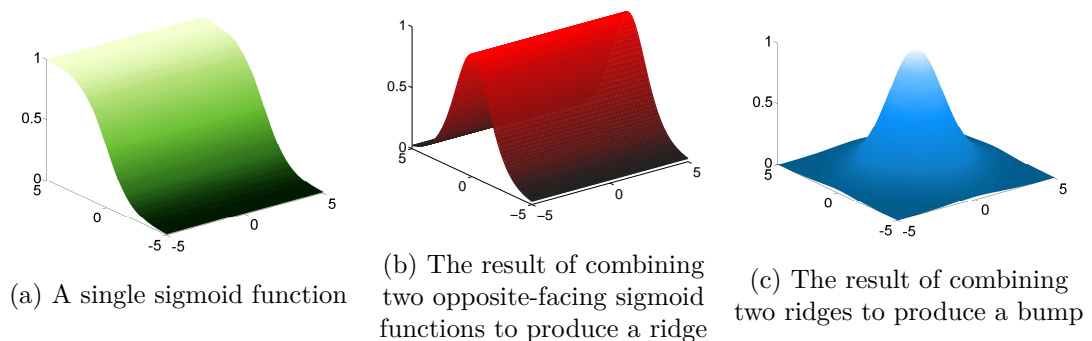


Figure 6.2: By combining sigmoid functions it is possible to reproduce complex continuous functions

6.4 ANN learning

Neural networks improve their performance by changing the way input signals are treated through the network in order to reduce the output error. This is done by changing the bias weights and bias values through training. The larger the network is, the more demanding training becomes to make a good network. This also provides the possibility to use more biases which make the network capable of distinguish between finer nuances in the result domain.

Training of an ANN is only based on experience with a given training dataset, and it's not possible to improve the performance beyond what these two factors determine. This implies that in order to get a good, well trained ANN, one needs a good network and a good representative, dataset for training. A FFANN is therefore only optimized to solve the problems it have already encountered, and don't improve with experience as some other types of ANNs. If the training dataset is biased, the outputs from the neural net will also be biased. The same applies for overtraining where a complex network is used with a simple problem and with a non satisfactory dataset. The network will then be able to make a perfect fit to the dataset while it may not represent the underlying trend very well. This is illustrated in Figure 6.4.

When we want to train an ANN against a dataset with given inputs and outputs, it's called *supervised learning*. This is an iterative method where the network start with initial bias weights and values and calculates the errors this initial network gives. Then the bias weights and values are changed to minimize the error in an iterative process until a stop condition is met (i.e. number of iterations, error magnitude, error gradient). A flow chart of this process is shown in Figure 6.5.

Recurrent networks learn by looking at the error produced by a «choice» versus the anticipated result from the action. From the trailer example, this would be that the trailer moves to the point that was anticipated to get towards the loading dock. If the trailer doesn't get where the network anticipated it would go, the network has to change. This is the same as learning by experience. The trailer problem is also divided in two,

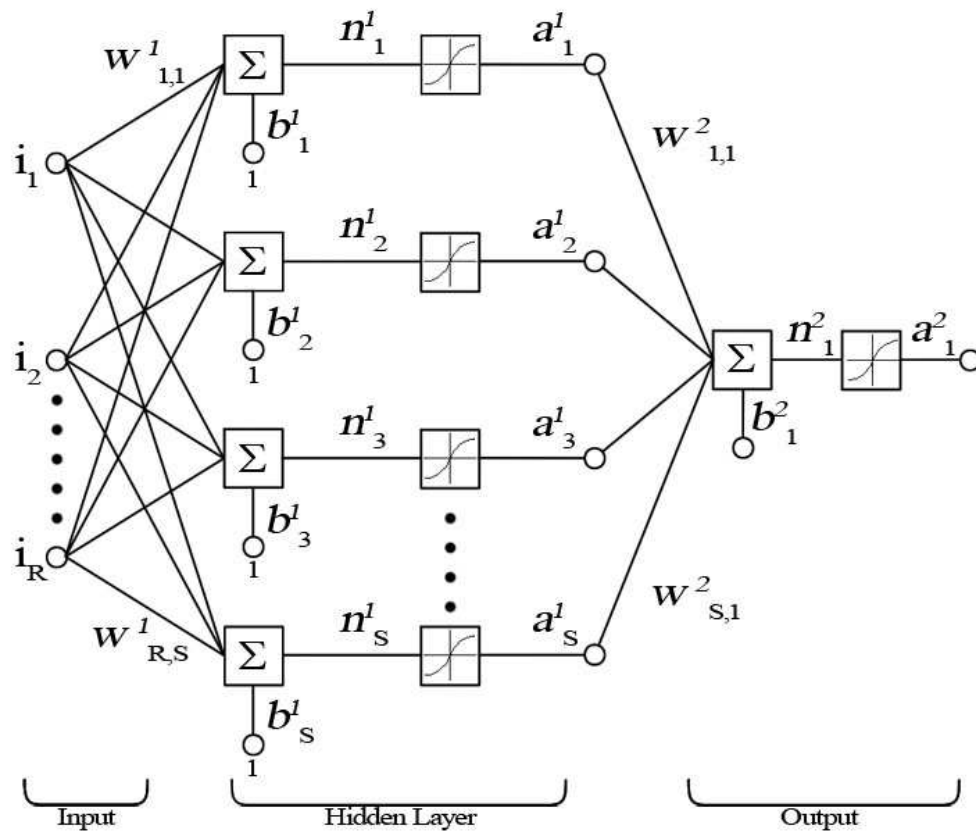


Figure 6.3: Schematic overview of a Neural Network

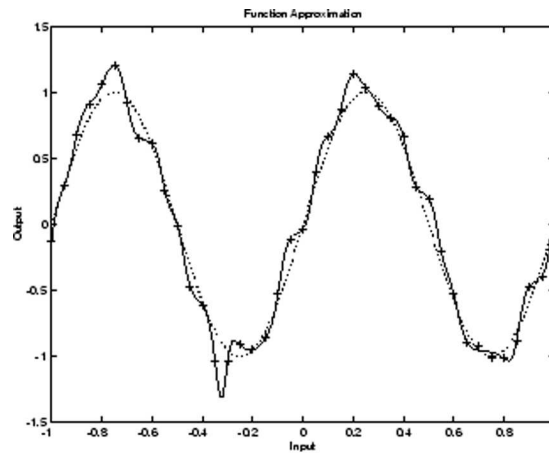


Figure 6.4: AN example of an overfitted function. The dataset consist of an sinus curve with a random error added to the data points. The solid line are the results from an overfitted ANN, while the dotted line are the underlying function. (Beale et al., 2015, p. 8-32)

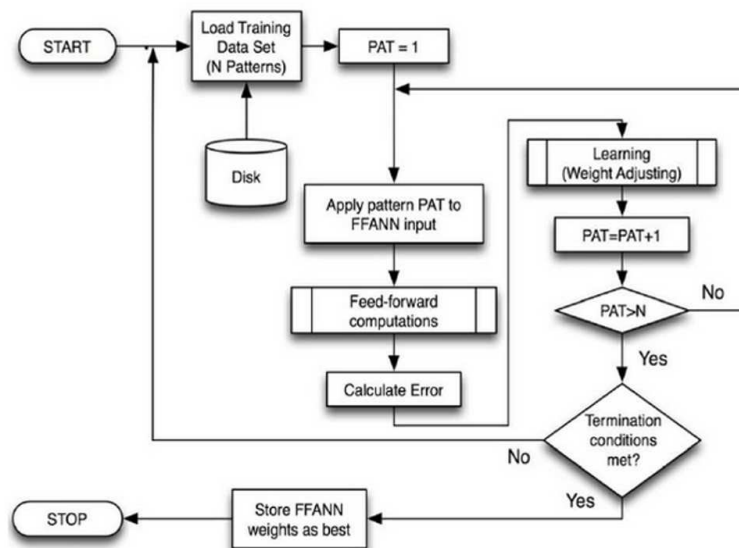


Figure 6.5: Flowchart of that illustrates the process of supervised learning for a FFANN (Kattan et al., 2011, p. 18)

one part which focus on what is needed to get the trailer to the loading ramp, and one that focus on steering the trailer and how it react on certain controls (i.e. gas, steering)

Chapter 7

Background data

The basis for a good empirical method is a good data material. Because it's not possible to manually control the training of an ANN, the outcome solely rely on the training data set. In general, ANNs can be considered as a representation of the training data set, and the quality of the data set is the quality of the ANN. To be able to make a good generalized ANN method, it is necessary with a large data set with a certain variance. For this thesis, MARINTEK has been so kind to give access to their model results of fishing vessels from the time period 1987-2013, in total 193 vessels and loading conditions. To give the reader a better understanding of the types of hull that this datamaterial consist of, I have plotted the distribution of some dimensionless parameters in Figure 7.1. A field of validity are important to define for empirical methods, and because I have used the same data material for all ANN methods, I give one field of validity for all the methods in Table 7.1.

Because of the nature of ANN, one must be careful when approaching the edges of the validity field. Especially the C_R ANN seems to be sensitive for the $L_{WL}/\nabla^{(1/3)}$ parameter. I have done some testing on vessels wchich are outside the field of validity. Due to the nature of ANN the results that fail, fails significantly and should be easy to identify as unreliable to the user.

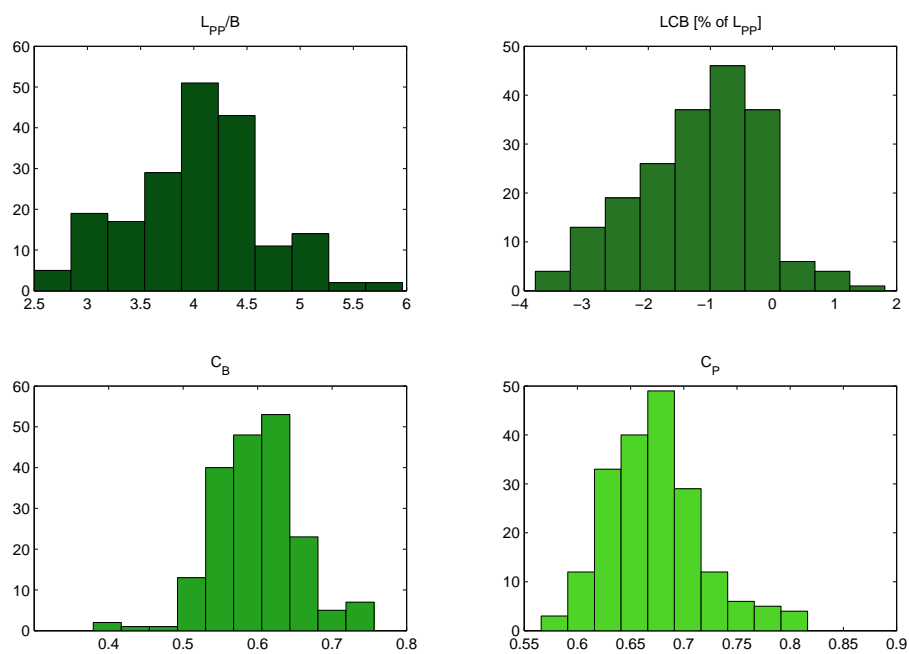


Figure 7.1: Graphical representation of some non-dimensional coefficients in the data material

Table 7.1: Outer limits for data material, who also will function as field of validity for ANN method

F_N	C_P	$L_{WL}/\nabla^{(1/3)}$	B/T	$(T_{AP} - T_{FP})/L_{PP}[\%]$	D_P/T
0.1–0.42	0.6–0.75	3.5–5.2	1.5–3.5	-2 – 3.5	0.45–0.9

Chapter 8

Updated Digernes Formula

The Digernes Formula, is a logarithmic linear formula (Formula 8.1) which simplify regression analysis. Because of the minimal additional work, I have performed a new regression of the Digernes Formula to see how much improvements can be achieved by using new coefficients from a modern dataset. There has been shown that Digernes formula does not predict the resistance well for low Froude numbers (Kleppestø, 2014). Of this reason, I have filtered out Froude numbers less than 0.15 in the data material.

$$\ln(R_{Total}) = A + b \cdot \ln\left(\frac{LWL}{B}\right) + c \cdot \ln\left(\frac{B}{T}\right) + \delta \cdot \ln(\nabla) + \beta \cdot F_N \quad (8.1)$$

Matlabs regression function gives a robust fitting option, and I have tested different implemented weight functions to see how much they improve the results. Robust fitting is a method to decrease the influence of outliers in the regression model. Matlabs user guide describe the process as:

«Robust regression works by assigning a weight to each data point. Weighting is done automatically and iteratively using a process called iteratively reweighted least squares. In the first iteration, each point is assigned equal weight and model coefficients are estimated using ordinary least squares. At subsequent iterations, weights are recomputed so that points farther from model predictions in the previous iteration are given lower weight. Model coefficients are then recomputed using weighted least squares. The process continues until the values of the coefficient estimates converge within a specified tolerance.» (Mathworks, 2015, ch.9)

The weight functions are dependent on the variable r , which is defined in Formula 8.2. In this formula, $resid$ is the vector of residuals from the previous iteration, h is the vector of leverage values from a least-square fit, and s is an estimation of the standard deviation of the error term given by $s = MAD/0.6745$. MAD is the median absolute deviation of the residuals from their median. The constant 0.6745 makes the estimate unbiased

Table 8.1: Matlabs weight functions available in the Robust fit option. All these are tested to see which gave the best results for Digernes Formula

#	Weight function	Equation
1	andrews	$w = (r < \pi) \cdot \sin(r)/r$
2	bisquare	$w = (r < 1) \cdot (1 - r^2)^2$
3	cauchy	$w = 1/(1 + r^2)$
4	fair	$w = 1/(1 + r)$
5	huber	$w = 1/\max(1, r)$
6	logistic	$w = \tanh(r)/r$
7	ols	Ordinary least square
8	talwar	$w = 1 \cdot (r < 1)$
9	welsch	$w = \exp(-(r^2))$

Table 8.2: Coefficient values for Digernes Formula found from regression analysis with a modern data material

a	b	c	δ	β
0.795154	0.401732	0.438858	0.959433	12.535261

for the normal distribution. $Tune$ is a tuning constant that is divided into the residual vector before computing weights, and has a default value for each weight function.

$$r = resid/(tune \cdot s \cdot \sqrt{1 - h}) \quad (8.2)$$

From Figure 8.1 we can see that weight function #8 gives significantly better results than the other functions. It gives a regression coefficient closer to one, and a lower error value (RMSE). I have also logged the coefficient values that each weight function gives, and there are only minor differences in the values (coefficients for all weight functions are given in Appendix A). This is easy to understand when we look at the resulting small differences in performance. The final coefficients from this new regression are given in Table 8.2.

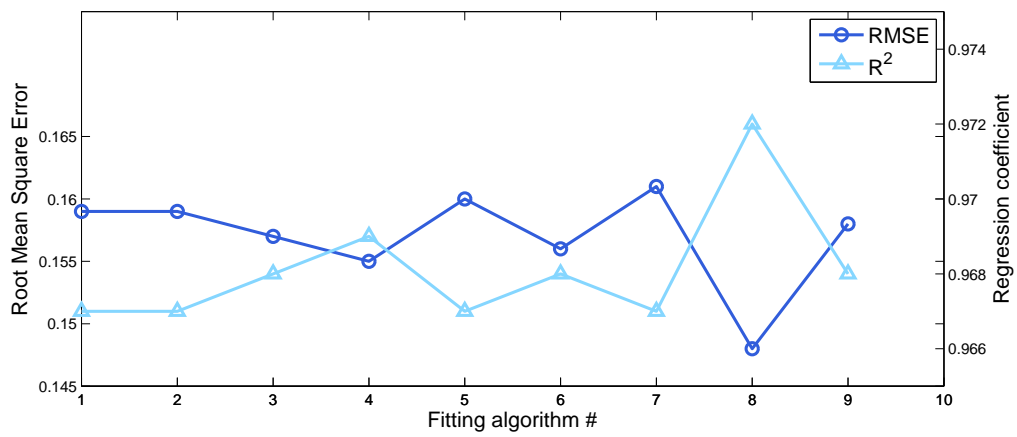


Figure 8.1: Calculated performance for the different Robust fitting weight functions given in Table 8.1. The left axis are the RMSE while the right axis are the calculated regression coefficient

Chapter 9

Network structures

The network design have a major influence of the types of problems and complexity that an ANN can handle. There are a lot of litterature for what type of network that is suitable for different kind of problems. There are however little information of how a multi layer feed forward network should be sized for specific types of problems, beyond the principles presented in Section 6.3. It was therefore no references I could use to determine the network structure to apply for my ANN methods in the beginning. The sources I found which gave concrete example of node distribution, was quite old and was therefor limited by computational power. In this work available computer power and computational time has not been restricted as I had access to a powerful computer (Table 9.1). The working period that stretched over a whole semester gave the possibility to train networks over long time periods.

To find a good network for each problem, I used a systematic trial and error method to identify any trends for what kind of networks that solved the different kind of problems the best. I will in the following text present the process to identify good network sizes and the final structure for each prediction method.

Table 9.1: Hardware and Software used to design and train ANN network

Hardware	
CPU model	Intel Xeon E5-2680 v3
CPU speed	2.50 GHz
RAM	128 GB
Software	
Operating System	Windows Server 2012 R2
Matlab + Neural Network toolbox	R2014a

9.1 C_R estimation

There are many methods existing today, which can predict the C_R coefficients behavior quite well, but they often have problems with the accuracy. The residual resistance is a large component in the total resistance, particularly for higher Froude numbers. As a result, even small percentage errors will produce large absolute errors and have a major effect on estimated calculated machinery power. From other methods, I could find a good collection of parameters that affect the resistance. Based on the most promising coefficients I have tested three different groups of input parameters, to determine how many are necessary to predict C_R well. When I selected the input parameters, I emphasized that they should be easy to estimate early in the design process. I then test the three parameter sets over a broad range of network structures to get a first impression of what is necessary to get good results. In the end I will use this knowledge on an improved set of network structures, to find the optimum network for C_R prediction.

It's important to remember that ANN training is a generic training process, which produces different results for every training session. There are possible to improve the results from a given network by repeatedly training it until the results are satisfactory. Of this reason, the following graphs should be treated as indications of a possible trend. Of this reason it is important to test the networks in parallel for different setups so that one can get an indication of the accuracy of a given network structure. This makes it possible to identify some trends and make conclusions.

For the first iteration, I tested the parameter sets presented in Table 9.2. Because C_R is a dimensionless coefficient, I have decided to use dimensionless input parameters as well. A test without dimensionless input parameters gave significantly poorer results. Parameter set 1 and 2 are large and quite similar and parameter set 3 is to test the effect of a small input set. Originally parameter set 2 was larger, but because of holes in the database some coefficients was removed to not make disturbances in the training of the networks. In Table 9.3, the node distribution and total nodes are given for 17 different networks. The performance of these networks are plotted in Figure 9.2, where the performance are given for both the training and validation sets. The performance are calculated by Matlab under training as the mean square error (Formula 10.1)

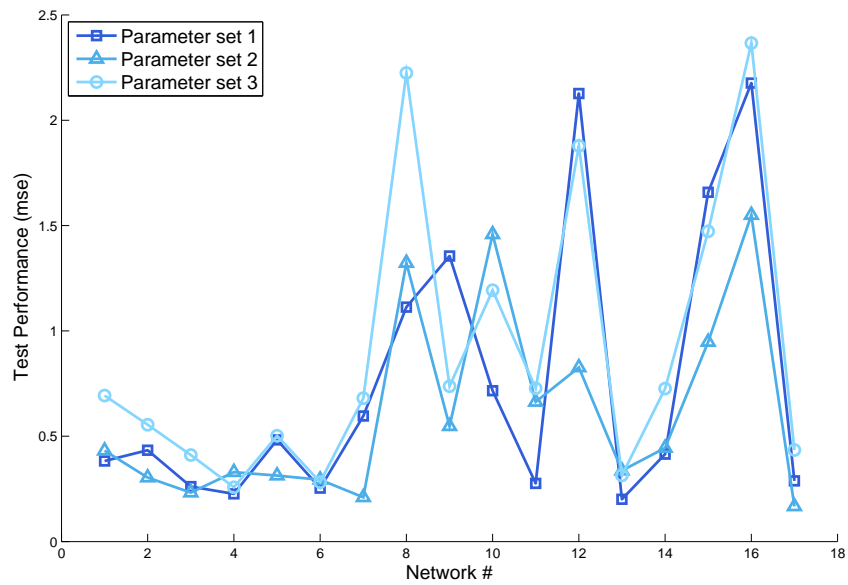
As Figure 9.2 illustrates, the training performance gets better as the number of neurons increases. This is expected, as more neurons gives more possibilities to find a network that fits the training set well. What is more interesting, is to look at how the networks are able to generalize to the test set (Figure 9.1a). This graph show that the number of neurons alone are no good measure of the quality of a network. When I look at both graphs, I note that network #13 performs well for both training and testing, and at the same time perform well for all three parameter sets. From this observation I conclude that this network gives stable and good results. I can also see a vague trend that networks with 45 nodes performs well in general. When the result for different parameter sets are evaluated, it's clear that parameter set #2 performs best for almost all networks for best train and test performance.

Table 9.2: Three dimensionless parameter sets that are tested for C_R prediction, to identify necessary number of input parameters. Input set 2 was meant to be larger, but because of holes in the database some less common parameters were removed

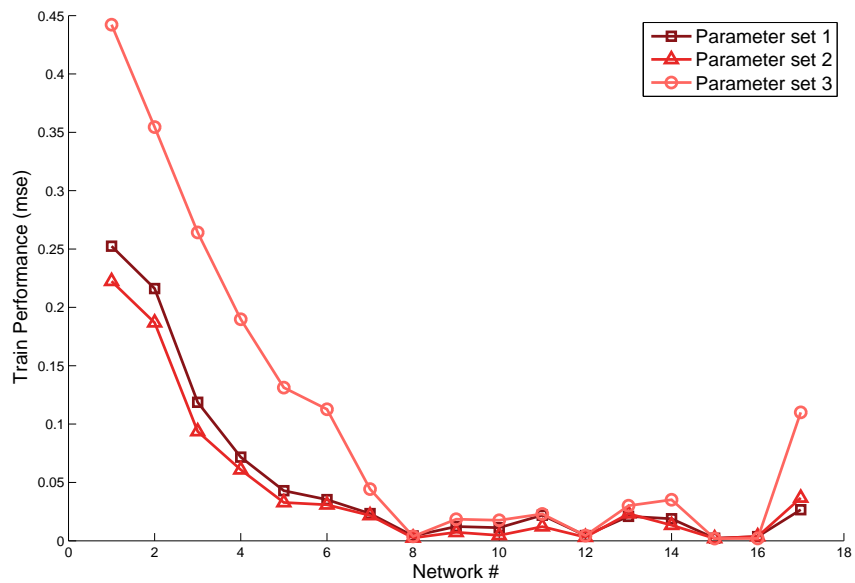
Parameter	Set 1	Set 2	Set 3
F_N	X	X	X
$L_{WL}/\Delta^{1/3}$		X	
L_{WL}/L_{PP}	X	X	
L_{WL}/B	X	X	X
B/T	X	X	X
$(T_{AP} - T_{FP})/L_{PP}$	X	X	
LCB/L_{PP}	X	X	X
C_M	X	X	
C_B	X	X	X

Table 9.3: An initial set of network structures to test how the ANN results depend on node distribution

Node distribution	No. of neurons	#
15-0-0	15	1
20-0-0	20	2
30-0-0	30	3
40-0-0	40	4
50-0-0	50	5
60-0-0	60	6
20-20-0	40	7
30-30-0	60	8
20-30-0	50	9
30-20-0	50	10
15-15-15	45	11
20-20-20	60	12
10-15-20	45	13
20-15-10	45	14
20-30-20	70	15
30-15-30	75	16
100-0-0	100	17



(a) Test Performance



(b) Train Performance

Figure 9.1: Graphical representation for the performance of different ANN architectures, with different number and distribution of neurons. Performance is represented with mean-square-error, and low values are best performance

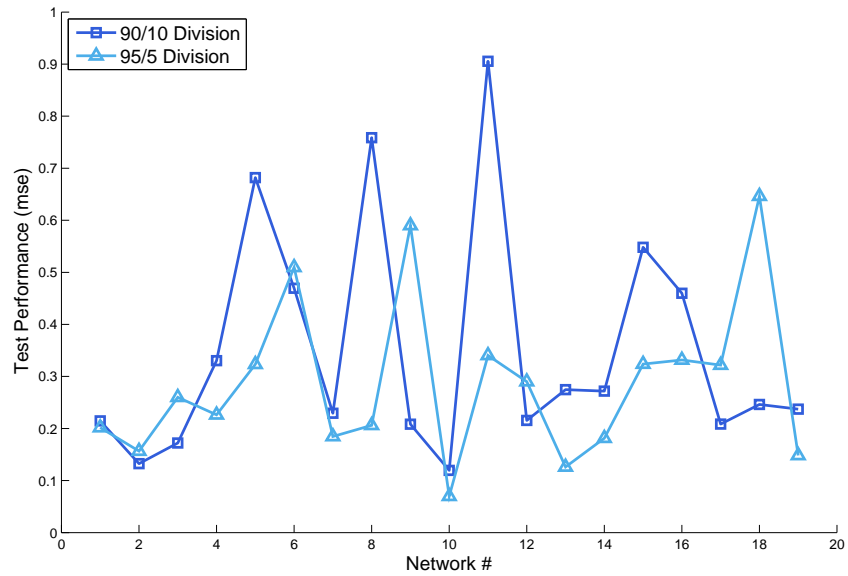
Table 9.4: New set of Neural Network structures iterated from performance to Networks in Table 9.3, with 40-50 neurons. This test set investigate if the shape of the network has any influence on the results

Node distribution	No. of neurons	#
40-0-0	40	1
45-0-0	45	2
50-0-0	50	3
20-20-0	40	4
20-25-0	45	5
25-20-0	45	6
15-30-0	45	7
30-15-0	45	8
5-40-0	45	9
40-5-0	45	10
15-15-15	45	11
10-10-20	40	12
20-10-10	40	13
10-20-10	40	14
10-15-20	45	15
20-15-10	45	16
10-20-15	45	17
15-30-5	50	18
5-30-15	50	19
15-5-30	50	20

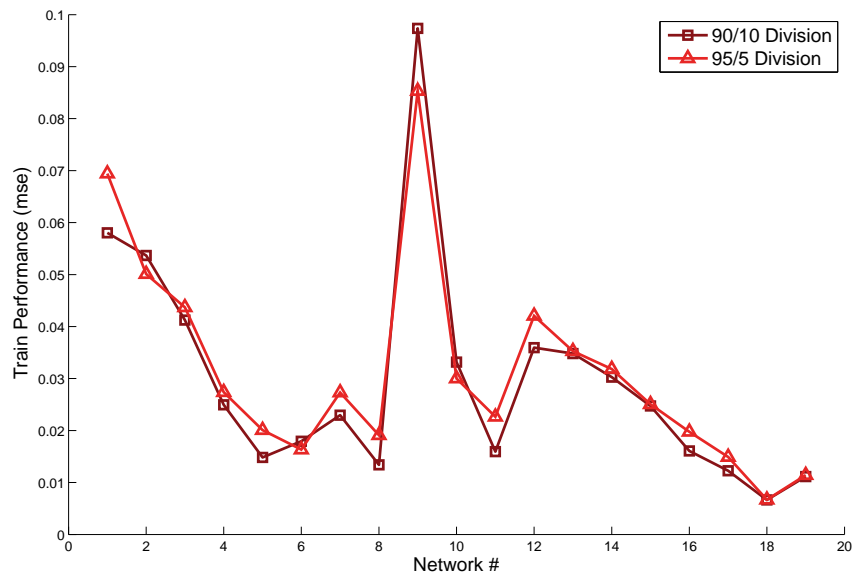
9.1.1 A second iteration

From this first iteration, I get an impression of what size of network that performs well, but there is no obvious trend in the test performance. Since this ANN will be applied for new vessels, it's important that the Test Performance are good and stable. Based on the first iterateion and with basis in Figure 9.2, I tested another set of networks that are similar to network #13. What I also want to investigate is if the shape of the network has any influence of the results. The new set of networks can be seen in Table 9.4. For these networks I have only tested input set #2, and instead focused on the division ratio of the total dataset between testing and training. My hypothesis are that there is an actual (but complex) relation between the input parameters and the C_R coefficient, and it is important that the network gets access to as many training samples as possible to find this relation. But at the same time, a certain amount of samples to test the networks ability to generalize are also required. The two division ratios plotted are 95%/5% and 90%/10%.

As for the more general investigation, we can see a good correlation between the size of the network and the Training performance, while there is not so easy to draw



(a) Test Performance



(b) Train Performance

Figure 9.2: Graphical representation for the performance of different Neural Network architectures, with different number and distribution of neurons. Performance is represented with mean-square-error

any conclusion from the Test performance. There is not even possible to see a clear correlation between Train- and Test performance. By looking at the Train performance, it seems that a neural net should have the smallest number of neurons in the last layer. An extreme effect, are the difference between train performance of network #9 and #10. The same trend are not so obvious by looking at the Test performance.

To pick a network architecture to go ahead with, I note the fact that network #15 perform very well under training with a good test performance for the 90/10 division ratio. With the aim to get the best possible performance, the best network would be three hidden layers with a neuron distribution 15-30-5. But, because I'm not very experienced with ANNs and afraid to recommend an over fitted ANN, I will choose network #2 with just one hidden layer and 45 neurons in this thesis. The reason for this are the behavior of the test performance in Figure 9.1a and 9.2a. These figures illustrates that the test performance are steady and good for the one layered ANN, while it gets quite unstable for the more complex networks. Because I cannot identify if this behavior are caused by difference in network architecture or over fitting, there is impossible for me to recommend a multi layered FFANN. The performance for the 90/10 division of the dataset are slightly weaker, but not worse than that I prefer the security a larger test set gives against over fitting.

9.2 Total Propulsion efficiency

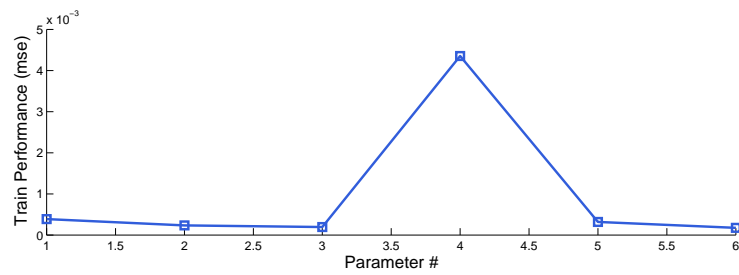
There might be ambitious to make a ANN method for the total propulsion, without taking it's component (as described in Section 4) into account. I have assumed that the problem is much similar to the C_R -estimation problem, so that I can use the same network structure to get good results. For this network, I will look closer into which input parameters that are important to get a good relation. It should, in theory, not matter if the parameters are dimensionless or not because the training process should be able to find the relation anyway. There are, as I will show, not indifferent when the result are dimensionless. There is possible to let the efficiency be speed or J-number dependent. I have chosen to take the mean value for each vessel, and predict this. This is simply because the variation for each vessel are small, and there was no clear correlation to the J-number in my material.

For the propulsion efficiency, the inflow to the propeller are the single most important factor. Parameters that can effect this inflow is therefor important to capture. This implies that coefficients that sais something about the fullness should be investigated. Some propeller data are obvious also important, together with the number of propellers. The suggested input parameter sets are given in Table 9.5, and their performance are given in Table 9.6.

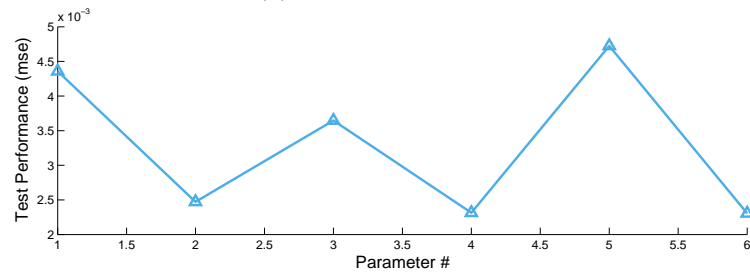
There is easy to see that input set # 6 that performs best on all parameters, and is the set that I will choose as my input set for η_D ANN. There seems to be little gain to choose dimensionless parameters, but because η_D is dimensionless I believe there are more consistent to use dimensionless input parameters. It is also interesting to note how good the results are, by remember how complex the total propulsion efficiency are.

Table 9.5: Six input sets, that are tested for η_D network. #4-6 are dimensionless

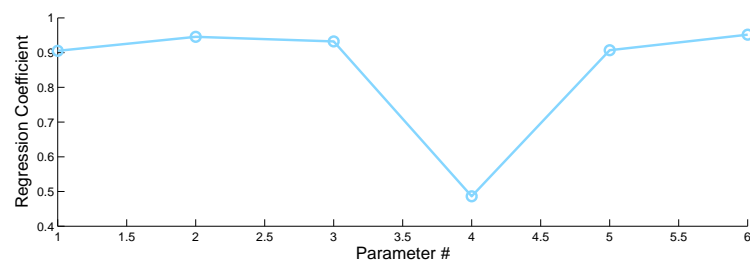
#	Input parameters									
1	No Prop	C_B	L_{WL}	B		∇				AeA0
2	No Prop	C_B	L_{WL}	B	T	∇				AeA0 Pitch
3	No Prop	C_B	L_{WL}	B	T	lcb	∇	D_P		AeA0 Pitch
4	No Prop	C_B	L_{WL}/B			lcb	$L_{WL}/\nabla^{1/3}$			Pitch
5	No Prop	C_B	L_{WL}/B	B/T		lcb	$L_{WL}/\nabla^{1/3}$			AeA0 Pitch
6	No Prop	C_B	L_{WL}/B	B/T		lcb	$L_{WL}/\nabla^{1/3}$	D_P/L_{WL}		AeA0 Pitch



(a) Train Performance



(b) Test Performance



(c) Regression coefficient

Figure 9.3: Graphical representation for the performance of different Input parameter sets for estimating η_D (is this needed?)

Table 9.6: Performance data for η_D ANN

#	Train Performance	Test Performance	R^2
1	3,880239E-04	4,356840E-03	0,90537
2	2,348041E-04	2,471690E-03	0,94563
3	1,936928E-04	3,646710E-03	0,93252
4	4,353770E-03	2,313880E-03	0,48618
5	3,170786E-04	4,723520E-03	0,90721
6	1,756066E-04	2,305300E-03	0,95184

Table 9.7: Properties of η_0 ANN

Input data	J -no, $Pitch$, $AeA0$, Z , D_P
Network	15-30-5
TrainPerformance	1.110776E-04
TestPerformance	2.042546E-04
R^2	0.98698

9.3 Open Water efficiency

The Open water efficiency are found from an open water test, for an individual propeller. The open water efficiency are better understood than the other parameters, this is because there are fewer unknown effects involved (like turbulence and flow pattern behind a hull) and well known propeller series that has investigated the effect of propeller geometry. There is however worth noting that Fishing vessels uses Ducted Propellers which is not so well investigated.

For the η_0 ANN, I used the same input parameters as Wagening Propeller series and added propeller diameter as a parameter. This network gave so good results that I did not perform a broad testing of networks and input parameters. The summary of the network can be seen in Table 9.7.

9.4 Hull Efficiency

Hull efficiency are a result of wake and thrust deduction, and it is usual to calculate these two parameters individually. I have tried to make ANN's for all three (η_H , wake, thrust deduction), but for the same network structure as C_R the training gives the strange result that the mean value gives the best performance. I have therefore looked at some simpler network structures to see how the results react on this.

I have tested some input sets and chosen one for all the parameters, which are *Number of Propellers*, C_B , lcb , B/T , L_{WL}/B . This is inspired from the methods described in Section 5.2, combined with the idea to define the fullness of the stern by using common hull parameters. The stern fullness should be possible to define with the block coefficient and the longitudinal center of buoyancy (lcb).

Let me, in words, explain how this attempt went. When I tried to train the network with the same architecture as I found for the C_R estimation, the best result are given as the mean value. This indicates that it cannot find a more reliable relation. By reducing the complexity of the network (10 neurons in one layer), it produces a relation which are not good compared to the networks mentioned above. This behavior are interestingly equal for η_H , wake and thrust deduction. After discussing these findings with experienced professionals I conclude that this is because these parameters depends mostly on the stern curvature and how the propeller are positioned. This is information that I can't include because the information I have access to, are given in table form and does not include stern shape. The methods developed, are also meant to be used in the early design phase, where the sterns hape may not have been decided. To give the reader some numbers to substantiate this phenomena, Table 9.8-9.11 gives the performance of some relatively simple networks against the C_R network.

We can see that the performance of the trained network are not much better than the mean value. I will, however continue to use the network with the best performance for the following calculations. This mean that I will use network #3 for thrust deduction and network #1 for wake and hull efficiency. But there should be noted that the mean value may be a more reliable estimate which can be seen from the Test Performance values for the 15-30-5. I have therefor given the mean values in Table 9.10.

9.5 relative rotative efficiency

I have not found any methods that predicts η_R . After investigating this parameter and talking to experienced professionals I understand that η_R are quite similar for each vessel type. On this basis, and after trying to train different ANNs I have decided that the mean value are a good and reliable estimate for η_R . When I tried to train an ANN, I used the same input parameters as for hull efficiency, and added an I/O parameter to tell if an propeller duct was installed or not. Most of the vessels uses propeller duct. The mean value for η_R are 1.0675284.

Table 9.8: Performance of Thrust deduction

#	Network	Train Performance	Test Performance	R^2
1	10-0-0	2,04482E-03	1,59440E-03	0,54188
2	40-0-0	2,49086E-03	2,28405E-03	0,33909
3	45-0-0	1,74665E-03	3,29041E-03	0,53307
4	40-5-0	1,70810E-03	2,78777E-03	0,56871
5	15-30-5	2,64608E-03	3,41059E-03	0,00000

Table 9.9: Performance of wake

#	Network	Train Performance	Test Performance	R^2
1	10-0-0	3,06063E-03	2,16921E-03	0,52795
2	40-0-0	3,22665E-03	1,26019E-03	0,52432
3	45-0-0	2,75511E-03	3,94356E-03	0,52698
4	40-5-0	3,05110E-03	2,21737E-03	0,52657
5	15-30-5	4,24294E-03	2,89303E-03	0,00000

Table 9.10: Performance of Hull efficiency

#	Network	Train Performance	Test Performance	R^2
1	10-0-0	4,71171E-03	5,02722E-03	0,61749
2	40-0-0	6,80309E-03	3,00503E-03	0,40452
3	45-0-0	6,21724E-03	8,80279E-03	0,39797
4	40-5-0	4,46053E-03	1,11394E-02	0,57586
5	15-30-5	8,14787E-03	3,28945E-03	0,00000

Table 9.11: Mean estimation of Hull efficiency

<i>Parameter</i>	<i>Mean value</i>
η_H	0.9967
wake	0.2528
thrust deduction	0.2590

Chapter 10

ANN programming

In Section 6, I have presented the general theory for ANN, but in the scope of a master thesis it's relevant to inform the reader of how this theory is applied. Rather than giving a user manual for the tools and methods I have applied. Instead I will share my experience and use this chapter to discuss what worked well and what I believe can be done in a better way. I hope this can be of help to those who want to train their own ANNs and would like to hear someone else's experience.

The software I have used in this work is mainly Matlab with Neural Network toolbox where I have performed all the calculations and training of ANNs. To process the data material to the right format (in my case .csv format), I have also used textprocessing tools that are able to configure .csv files (MS Excel and TextPad have worked well for me).

10.1 Input processing

The most important part of training an ANN is to have a large and reliable data material. Especially for resistance and propulsion estimation it is difficult to predict which input parameters that will give the best results. So to have the possibility efficient change the input parameters. I therefore converted my data source to a .csv file, and used Matlabs handy «csvread» command, that automatically transform a .csv file to a matrix. When the data material is imported as a matrix in Matlab, it is trivial to sort and process the data before beginning the training of the network. I gained good experience in structuring the data into *tables*, which gives the possibility to assign names to the row and columns so that it is easier to call for the right data (i.e. `Geometry.Lwl(i)` if one wants to use the waterline length of model i that is stored in a table called «Geometry»).

Early in the process, I decided to make a few and general tables to store a lot of information in one place. This made it easy to know where to get the information I needed, and to know that the methods was tested and trained on the same premises. Late in the process, I learned that it might have been better to also make individual tables for each ANN method. The reason for this hindsight is that sometimes there are missing data in some of the inputs, but it's undesirable to remove the vessel from the

table and thereby reducing the data material for all the ANNs.

10.2 Matlab NeuralNetwork toolbox

The NN toolbox is a comprehensive tool package that is needed to handle the complete process of training an ANN. It includes a lot of commands, but also a very handy graphical userface to design and train ANNs (By using the command «nnstart»). This graphical interface is a good way to get familiar with the NN toolbox and gives easy access to the important operations, but gives limited options to do complex network designs. I used the included options of codegeneration, that convert the choices made in the graphical userface to a script. In this script it's possible to exploit the advanced possibilities in the toolbox.

In Appendix C, I have given an example of the script I used to train ANNs. I made one script for each ANN. This way I could store different input/output matrices and then it was easier to train several networks simultaneously. The script makes it easier to customize (1)the neuron distribution over several layers, (2)the division of the dataset in a training and testing fraction and (3)the training limits (Like maximum training epochs, stop conditions and such). The structure of the script are standarized from the export option, but there are alsopossible to train and design more complex networks (Beale et al., 2015, p.8-44). In my work, I limited myself to using the the configurations that I understand and not tested very advanced networks, this is because I need to be committed to the results I present and be able to explain them. To take the ANN knowledge and experience furthter, I believe that these options should be investigated by users that are familiar with both the use and desing of ANNs because I'm conviced that they might produce even better results when used right.

A parameter that is important in the use of NN toolbox is the performance parameter. In this thesis I have used mean square error (MSE, Formula 10.1) as the performance calculations. It increases the importance of outliers in the predictions, and therefor reduces the standard deviation in the training. This is an important quality for this types of problems where I want a stable and thrustworthy prediction method.

$$MSE = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \quad (10.1)$$

10.3 Final script

Another nice feature of ANNs, is that they are so easy to use after they are trained. A trained FFANN basically consist of matrices of bias weights and values that is to be added and multiplied. This is an operation that is very simple for any computer. The training requires a lot of processing power to find these matrices, but once found an ANN can be used with a very small processor (even a smartphone can use ANNs). The final script which consist of matrices and some transfer functions, can also easily be

translated to any programming language that best suits the intended use. An example of an FFANN with a neuron distribution of 15-30-5 is attached in Appendix D

Chapter 11

Results

This chapter will compare the developed ANN methods against established methods to quantify the improvements that ANN gives. Because I have used 90% of my data material for training the networks, it is expected that they will perform better than the other methods. I have tried to account for this when choosing error calculations for comparison, which should indicate both the precision of the methods and dispersion of the results. The aim is to give the reader a good basis to identify if an ANN method could be a better tool in estimating resistance and installed power than what are publicly available today. This chapter is divided in two parts, first I investigate the estimation of individual parameters, before I look at how the methods work together to predict the necessary delivered power.

To indicate the precision of the methods I use the mean error (Formula 11.1), this is a simple but effective indicator of how close the prediction are to the actual value. The mean value can however be tricked by an estimate that over predict and under predict equally, so that independent if the scattering are high, the mean error will be low. To correct for this pitfall, I have also calculated the standard deviation (Formula 11.2). This is a good parameter to reveal if there is large dispersion in the results, because it sums up the deviation from the mean error in each sample. So the case with large deviations that are equally distributed for positive and negative errors will produce a small mean error, but a large standard deviation. These two variables work well together to give the reader a good impression of the reliability of each method. To make comparisons easier, both between methods and parameters, I will calculate errors in percentage against the model results (Formula 11.3).

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad (11.1)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2} \quad (11.2)$$

Table 11.1: Error calculation for the tested C_R estimation methods. All the results are converted to the total resistance in Newton before the errors are calculated. The errors are given as percentage compared to the model test results.

	Mean error [%]	Standard Deviation
ANN	1.4539	4.4158
Digernes	13.0064	14.3315
Hollenbach	-13.6922	16.4739
Holtrop	4.8853	15.6854
Oortmerssen	34.126	30.3128

$$X = 100 \cdot \frac{x_{calculated} - x_{model}}{x_{model}} \quad (11.3)$$

11.1 Individual methods

I will first calculate the error for each method independently, and compare the difference between the ANN and established methods I have investigated. In the next section, I will use the described methods to calculate the necessary delivered power.

In the project assignment last semester (Kleppestø, 2014), I calculated the errors for the empirical methods. The data material was then smaller than the one used in this thesis. It consist of 48 fishing vessels (Where 194 vessels are present in this data material). Because some of the empirical methods calculate the total resistance directly, I have calculated the total resistance based on the C_R coefficient so I can compare all the methods on the same basis. As Table 11.1 illustrates, the developed ANN for C_R method perform significantly better than the other methods.

Table 11.2 gives the calculated errors for wake estimation. The methods that are tested are discussed in Section 5.2, and one can see that there is not necessary the most advanced formulas that are best. Especially interesting is it to see that the simple formula of Taylor gives so good results. As discussed in Section 9.4. Because the mean value seems to be a competitive alternative to estimate the wake estimation, I have calculated the error for the mean value for the vessels with one propeller which applies for 186 of the vessels, and we can see that it actually performs better than the ANN.

I have not investigated more than one empirical method for thrust deduction, and in Table 11.3 it's easy to see that the result are not very good for Holtrop's thrust deduction estimate. The ANN method are fairly good, and one can see that the mean value are not so good in prediction as for wake. I have also here removed the seven vessels with twin propellers, and can inform the reader that the effect on error was much less than for wake.

Another coefficient that I was curious if the mean value could compete with the ANN results are the relative rotative efficiency (η_R). This is the parameter where the effect of sorting out the twin-screw ships has the most influence on the errors. We can see that

Table 11.2: Error calculation for the tested wake estimation methods. The errors are given as percentage compared to the model test results. The mean value are given as an alternative to the ANN. The reason for this are explained in Section 9.4

	Mean error [%]	Standard Deviation
ANN	6.4155	0.2914
Mean Value	5.3373	0.2640
BSRA	454.4822	30.2287
Harvald	9.8757	0.6322
Holtrop	-21.4035	0.3452
Taylor	8.2889	0.5312

Table 11.3: Error calculation for the tested thrust estimation methods. The errors are given as percentage compared to the model test results. The mean value are given as an alternative to the ANN. The reason for this are explained in Section 9.4

	Mean error [%]	Standard Deviation
ANN	2.7184	0.1714
Mean Value	4.1971	0.2159
Holtrop	-112.017	0.1590

the mean value performs very well, but the ANN are slightly better. The reader should be aware that η_R varies much for different kinds of vessel types.

To predict the propeller efficiency, it is common to use the systematic Wageningen B-series. When we compare the results to the trained ANN, it's interesting to note that the performance are not very bad. Even when we know that the tested propellers are equipped with ducts, something the Wageningen B-series propellers are not. When we look at Table 11.5, it seems it's easy to conclude that the ANN does not perform any better than the Wageningen B-series. But this is not necessarily true, because we here got an interesting case where the ANN in general performs well, but has two results that increase the total error significantly. After investigating these two errors, I found that the A_E/A_0 input value was zero for these two propellers. This was discovered rather late in the process, and because of the way I process the data's in my Matlab program its not trivial to remove individual results from the calculations. To illustrate the case, I have plotted all the errors in Figure B.2 where it's clear that for all the other vessels,

Table 11.4: Error calculation for the tested η_R estimation methods. The errors are given as percentage compared to the model test results. The mean value are given as an alternative to the ANN. The reason for this are explained in Section 9.4

	Mean [%]	Standard Deviation
ANN	0.1688	0.0372
Mean Value	0.2872	0.053

Table 11.5: Error calculation for the tested η_0 estimation methods. The errors are given as percentage compared to the model test results for all tested J-values

	Mean error [%]	Standard Deviation
ANN	-1.5626	0.0602
Wageningen	-0.0436	0.0675

Table 11.6: An overview of the calculated errors for all trained ANNs. The errors are given as percentage compared to model test results, and the standard deviation are calculated for the percentage errors

	Mean error[%]	Standard Deviation
C_R	1.4827	0.1745
η_D	0.0423	0.0409
η_0	-1.5626	0.0602
η_R	0.1688	0.0372
η_H	0.5992	0.0684
thrust	2.7184	0.1714
wake	6.4155	0.2914

the ANN performs much better than Wageningen B-series. This can also be used as an example that ANNs can give quite extreme results with poor input data, in this example the results from Wageningen B-series don't show such a major effect with the same input values.

In general, it is very interesting to see how well ANN is able to estimate the total propulsion efficiency. It removes many potential error sources to use only one method, compared to calculate each component with several methods. When all the results for the ANNs are examined, there is impressive to see how good the results are for all the different methods (Table 11.6). Especially the very low standard deviation for all the methods gives a good impression that the ANN are a comprehensive method that can handle different problems very well.

11.2 Estimating delivered power

The goal for this thesis is to predict the necessary machinery power. To estimate the machinery power, the results from several methods must be put together. Then small individual errors can either be strengthened or weakened by errors in other methods. It's therefore not enough to look at the methods independently, but necessary to see how they work together.

I have tested many methods, and this gives many possible combinations to calculate the total machinery power. To keep it simple, I have chosen just three combinations that seems most relevant. A obvious choice is to test the effect to calculate η_D directly, because if this works it is the most efficiency way to estimate with as few methods as

Table 11.7: An overview of the calculated errors for the delivered power for three different combinations of methods. **#1** C_R and η_D are calculated from ANN. **#2** C_R , wake, thrust deduction, η_0 and η_R from ANN. **#3** R_{Tot} from updated Digernes formula, wake from Taylors formula, thrust deduction from Holtrop's formula, η_0 from Wageningen B-series and η_D as the mean value from data material.

	Mean error [%]	Standard Deviation
# 1	-3.0009	0.0258
# 2	12.5282	0.8768
# 3	-79.3519	0.1344

possible. To compare this, I have performed a calculation with the other ANN methods. The last combination I have chosen is to calculate η_D from the best performing empirical methods from Section 11.1, that is Taylors formula for wake, Holtrops formula for thrust, Wageningen B-series for η_0 and the mean value for η_R . I have used the updated Digernes formula to calculate R_{Tot} for the empirical combination because it performs well and was easy accessible for me.

From Table 11.7 we can see that the mean error are not so different, while there are significantly differences for the standard deviation. This type of scenario are discussed in the introduction to this chapter. To assist the reader further, I have included a 3-D plot of all the calculated errors for all vessels and speed in Appendix B (Figure B.1). On the basis of this results, it's clear that it's recommended to use only the ANN method for C_R and η_D to estimate the necessary machinery power.

Chapter 12

Discussion

When the results are summerized, all the ANN methods perform better than existing methods. This illustrates how powerful and flexible a ANN is, and this supports the impression I have from this work. As long as one uses a good and sufficiently large data material, anyone can produce good and reliable ANN's that are tailored for their problem.

My main concern when training ANNs, has been to control overfitting. An overfitted ANN loses its generalization ability, and therefore loses reliability as a prediction method. I have taken several measures to prevent overfitting, and believe that I have succeeded. A small test with vessels not included in the data material, show that the resistance prediction are very good, but a few vessels that are slightly outside the field of validity fails significantly. It is however so obvious that these results are wrong, that any user would dismiss them. The same phenomenon is observed for some of the propulsion estimation methods, most of the errors are very low with a few results are way off when some of the input parameters are abnormal. This indicates that there are still more to learn about ANNs, but with the promising results in this thesis, ANN has the potential to become the dominant tool for empirical prediction in the near future.

The estimation of machinery power, gave very good result by combining just C_R and η_D estimation. Overall, I would recommend everyone that needs a reliable and precise estimate of resistance or machinery power to try these methods.

I have also improved Digernes Formula for modern fishing vessels. This formula can be recommended for everyone that needs a fast and simple estimate of resistance.

12.1 Future work

This work has shown the great potential of ANN, but it has not been able to produce excellent results for all type of problems. There is still a need for a deeper understanding of what kind of network architecture that suits the kind of problems encountered in ship resistance and propulsion estimation. I have not been able to justify if my errors are caused by not choosing the right input parameters, or the wrong network architecture. A deeper study into commercial use of ANN should aim to answer these problems.

Another interesting topic for future work, would be to check the capability to do a parameter study with an ANN. Because neural network training is about finding connections both between the input parameters and the output parameters. This means that a neural network not only look at how each input affect the output, but how the input parameters *together* affect the output value. This should mean that there is possible to do a parameter study from an ANN if the training sets are good.

Bibliography

- Beale, M. H., Hagan, M. T., and Demuth, H. B. (2015). *Neural Network Toolbox: User's Guide*. Mathworks, 8.3 edition. http://se.mathworks.com/help/releases/R2015a/pdf_doc/nnet/nnet_ug.pdf.
- Bernitsas, M. M., Ray, D., and Kinley, P. (1981). Kt, kq and efficiency curves for the wageningen b-series propellers. Technical report, The University of Michigan.
- Digernes, T. (1982). *An analytical approach to evaluating fishing vessel design and operation*. PhD thesis, Norwegian Institute of Technology.
- Digernes, T. (1983). Motstand, fremdrifts og styring. Technical Report 2, Fiskeriteknisk Forskingsinstitutt (FTFI). Resistance regressions - results using norwegian danish and FAO - material.
- Fathi, D., Ringen, E., Alterskjær, A., and Berget, K. (2012). *ShipX Ship Speed & Powering Manual*. MARINTEK.
- Fausett, L. V. and Hall, P. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*, volume 40. Prentice-Hall Englewood Cliffs.
- Granville, P. (1956). The viscous resistance of surface vessels and the skin friction of flat plates. *Trans. SNAME*, 64:209–227.
- Harvald, S. A. (1983). *Resistance and Propulsion of Ships*. Ocean Engineering. John Wiley & Sons, Inc.
- He, X. and Xu, S. (2010). *Process neural networks: theory and applications*. Springer Science & Business Media.
- Hollenbach, K. U. (1998). Estimating resistance and propulsion for single-screw and twin-screw ships-ship technology research 45 (1998). *Schiffstechnik*, 45(2):72.
- Holtrop, J. (1977). Statistical analysis of performance test results. *International Shipbuilding Progress*, 24(270).
- Holtrop, J. (1984). A statistical re-analysis of resistance and propulsion data. *International Shipbuilding Progress*, 31(363):272–276.

- ITTC (2011). Ittc – recommended procedures and guidelines: Resistance test. Technical report, ITTC.
- Kattan, A., Abdullah, R., and Geem, Z. W. (2011). *Artificial Neural Network training and software implementation techniques*. Computer Networks. Nova Science Publisher, Inc.
- Kleppestø, K. (2014). Empirisk motstandsberregning på fiskefartøy. Pre-Project for Master Thesis.
- Kohl, N. (2010). Re: Role of bias in neural networks [online forum comment]. <http://stackoverflow.com/questions/2480650/role-of-bias-in-neural-networks>.
- Mathworks (2015). *Statistics and Machine learning Toolbox: User's guide*. Mathworks, 10.0 edition.
- Molland, A. F., Turnock, S. R., and Hudson, D. A. (2011). *Ship Resistance and Propulsion*. Cambridge university press.
- Parker, M. N. and Eng., B. (1966). The b.s.r.a. methodical series - an overall presentation propulsion factors. *Transactions of the royal institution of naval architects*, 108:389.
- Russel, S. and Norvig, P. (2010). *Artificial Intelligence - A modern Approach*. Pearson Education, 3rd edition.
- Steen, S. (2013). *Ship Resistance - Lecture notes TMR4220*. Dep. Marine Technology, NTNU.
- Taylor, D. W. (1910). *The speed and power of ships*. John Wiley & Sons, Inc.
- Ünal, M., Ak, A., Topuz, V., and Erdal, H. (2013). *Optimization of PID Controllers Using Ant Colony and Genetic Algorithms*. Springer-Verlag.
- Van Oortmerssen, G. (1971). A power prediction method and its application to small ships. *International Shipbuilding Progress*, 12(207).

Appendix A

Digernes coefficients

The resulting coefficients from the updated regression analysis of Digernes Formula. Different robust fitting weight functions are tested, and only minor differences are produced.

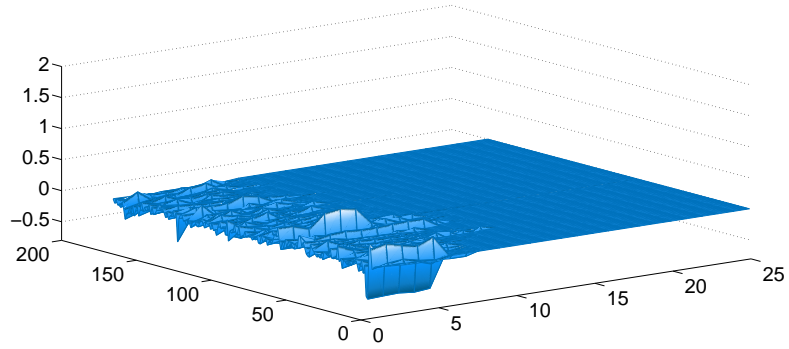
Table A.1: The resulting coefficient values for different robust weight function from regression analysis in Matlab. See Section 8

#	Weight function	a	b	c	δ	β
1	andrews	-0,184409	0,402914	0,431905	0,955804	12,506959
2	bisquare	-0,184274	0,402638	0,431659	0,955863	12,507215
3	cauchy	-0,179162	0,392988	0,431762	0,957102	12,508442
4	fair	-0,175876	0,385251	0,429369	0,958457	12,510439
5	huber	-0,180830	0,394678	0,434129	0,956721	12,509730
6	logistic	-0,177884	0,390039	0,431725	0,957521	12,509021
7	ols	-0,174844	0,345410	0,434087	0,966449	12,494959
8	talwar	-0,223922	0,401732	0,438858	0,959433	12,535261
9	welsch	-0,191959	0,399156	0,431529	0,956268	12,507599

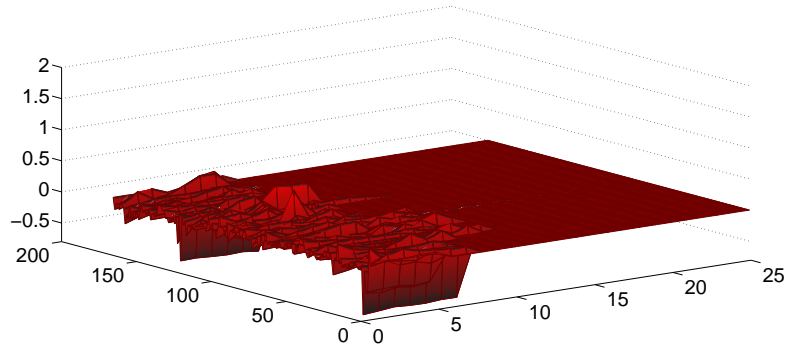
Appendix B

Error Plots

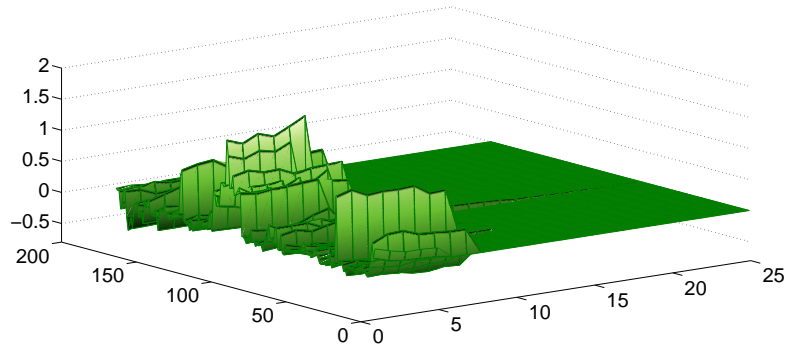
Sometimes it's easier to explain with a figure than tables. The following figures tries to show the distribution of the error for some of the threated methods in this thesis.



(a) Visualization of the errors calculated for the #1 combinations of methods for machinery power estimation

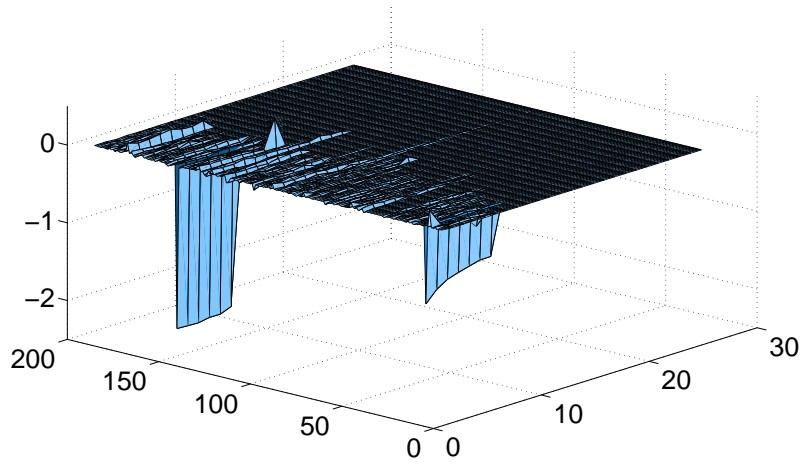


(b) Visualization of the errors calculated for the #2 combinations of methods for machinery power estimation

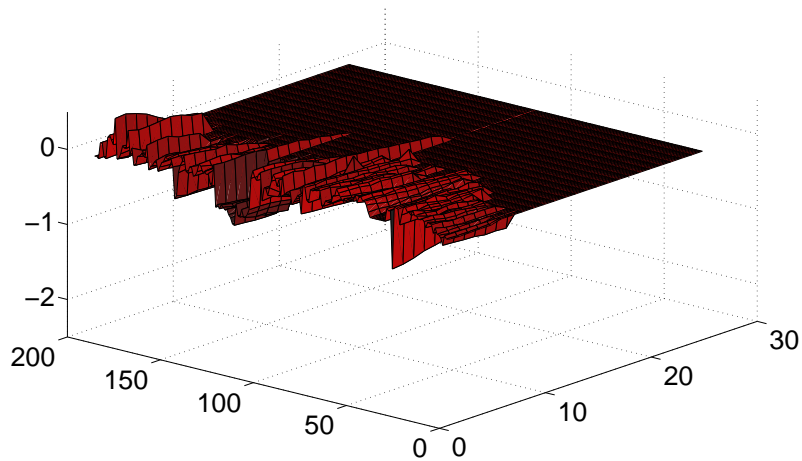


(c) Visualization of the errors calculated for the #3 combinations of methods for machinery power estimation

Figure B.1: The plots illustrates all the calculated errors for delivered power in percentage on the z-axis (upwards). Each vessel are on the x-axis (0-200) and the speeds in the y-axis (0-25). The vessels are testes for different number of speeds.



(a) Visualization of the errors calculated for the ANN method for η_0 estimation



(b) Visualization of the errors calculated for the Wageningen B-series used for η_0 estimation

Figure B.2: The plots illustrates all the calculated errors for η_0 in percentage on the z-axis (upwards). Each vessel are on the x-axis (0-200) and the speeds in the y-axis (0-25). The vessels are testes for different number of speeds. The figure are included to show that an ANN network can have a very good performance in general, but fail considerable when it misses. In these two cases, there are an error (zero value) for one of the input parameters that creates the two large errors

Appendix C

ANN.m

Matlab code that is used to train a three layered FFANN network

```

% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
%
% This script assumes these variables are defined:
%
% ANN_in - input data.
% ANN_out - target data.

%global ANN_in ANN_out

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. NFTOOL falls back to this in low memory situations.
trainFcn = 'trainbr'; % Bayesian Regularization

% Create a Fitting Network
hiddenLayerSize = [15 30 5]; % Neuron distribution over three layers
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
% !Hides post- & preprocessing as described in Stackoverflow
% thread so the length of weight vector is known!
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 89/100;
net.divideParam.valRatio = 1/100;
net.divideParam.testRatio = 10/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse'; % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression','plotfit'};

% Train the Network
net.trainParam.epochs = 5000; % Max training epochs
net.trainParam.showWindow = 0; % If graphical window of training progress should show up
net.trainParam.showCommandLine = 1; % Sho progress as command line output
net.trainParam.show = 500; % Choose how often progress should be written

% Train the network
[net,tr] = train(net,ANN_in,ANN_out);

```

```

% Test the Network
y = net(ANN_in);
e = gsubtract(ANN_out,y);
performance = perform(net,ANN_out,y)

% Recalculate Training, Validation and Test Performance
trainTargets = ANN_out .* tr.trainMask{1};
valTargets = ANN_out .* tr.valMask{1};
testTargets = ANN_out .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
%valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
figure, plotperform(tr) % Performance progress
%figure, plottrainstate(tr)
%figure, plotfit(net,ANN_in,ANN_out)
figure, plotregression(ANN_out,y) % Regression coefficient
%figure, ploterrhist(e)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
if (true)
    % Generate MATLAB function for neural network for application deployment
    % in MATLAB scripts or with MATLAB Compiler and Builder tools, or simply
    % to examine the calculations your trained neural network performs.
    genFunction(net,'Mastergrad/myNeuralNetworkFunction');
    y = myNeuralNetworkFunction_Cr(ANN_in);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(ANN_in);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end

```

Appendix D

MyNeuralNetworkFunction.m

Example file of how a trained FFANN with neuron distribution of 15-30-5 would look like. The values are set to one because the results are confidential.

```

function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION_CR neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction.
%
% [Y] = myNeuralNetworkFunction_Cr(X,~,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 8xQ matrix, input #1 at timestep ts.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

% ===== NEURAL NETWORK CONSTANTS =====
% The matrix values are replaced with ones. This is because the final
% results are confidential but at the same time it should be of interest
% to the reader to see the matrix sizes.

% Input 1
x1_step1_xoffset = [1;1;1;1;1;1;1;1];
x1_step1_gain = [1;1;1;1;1;1;1;1];
x1_step1_ymin = -1;

% Layer 1
b1 = [1;1;1;1;1;1;1;1;1;1;1;1;1;1;1];
IW1_1 = [ 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1];

% Layer 2
b2 = [1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1];
LW2_1 = [ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; ...
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];

% Layer 3

```



```

end

% Final Delay States
Xf = cell(1,0);
Af = cell(4,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x, settings_gain, settings_xoffset, settings_ymin)
    y = bsxfun(@minus, x, settings_xoffset);
    y = bsxfun(@times, y, settings_gain);
    y = bsxfun(@plus, y, settings_ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
    a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings_gain, settings_xoffset, settings_ymin)
    x = bsxfun(@minus, y, settings_ymin);
    x = bsxfun(@rdivide, x, settings_gain);
    x = bsxfun(@plus, x, settings_xoffset);
end

```