Determination of water speeds of ships

by

Michaël Mersie

Abstract

The aim of this research is to evaluate two different current correction methods: the "Iterative" Method, that has (partly) replaced the traditional "Mean of means" Method and is currently in use, and the "Direct" Method, a more fundamental and mathematical approach to the problem. These methods were both developed to determine the speed of a ship relative to the water, during so-called speed trials. This information is of great importance, since a delivered ship has to satisfy the contract speeds, recorded in the purchase agreement. If the ship does not satisfy these requirements, it could be rejected by the buyer in the worst case. Needless to say, it is very important that the speed trial analysis is as accurate as possible.

Both methods make use of two assumptions: an exponential relationship between ship power and the ship's water speed and a sinusoid relationship between the current speed and time. It is known that these assumption do not describe reality in an exact way, but they are simply the best approximations available at this moment.

The methods have been evaluated by simulations of different data input. This data is taken from model tests performed at MARIN. This means that these are cases where the water speed is known (makes it possible to analyse the true error) and no corrections need to be made for waves or wind. Additionally an academic example of a current is used. Furthermore realistic measurement noise was added to the input data. In order to evaluate if the methods correctly approximate this current function, 1000 simulations were carried out for every data set.

An interesting finding was that the "Direct" Method always converged to a good fit, where the "Iterative" Method failed in some situations. In the case that both methods converged rightfully, not much can be said about which method is the most accurate. Sometimes the "Iterative" Method provided a better result, sometimes the "Direct" Method did. Important to note is that in the case of good convergence of both methods, the differences were reasonably small between the two. This could be caused by errors on measurement noise level, the existence of multiple local minima or a premature stop of the method, because of computer limitations.
These computer limitations are involved with the following. For the non-linear fitting and the stopping criterion, values need to be put in by the user. Too large values can result in a premature stop of the algorithm or wrong convergence of the "Iterative" Method. Too small values on the other hand can lead to extensively long calculations time and thus computer limitations. Furthermore the chosen values should not be smaller than the computer precision, since the stopping criterion is then based on measurement noise level, which is a bad thing. The "Direct" Method only needs one (non-linear) fitting and consequently has a way smaller calculation time: a huge plus.

The performance of different non-linear fitting methods, that are being used in the current correction methods, was also analysed. In most cases both the Trust Region Reflective- and the Levenberg-Marquardt algorithm gave similar results. In an initial value analysis, performed in this research, it was shown that the Trust Region Reflective is more reliable if the initial value is chosen far off the true value. It is thus recommended to use the Trust Region Reflective algorithm.

Altogether, the "Direct" Method seems to be a more robust, stable and theoretically substantiated algorithm. It is recommended to use this algorithm as the main method for speed trial analysis or to use it as an initial value estimator.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor prof. dr. ir. Kees Vuik for his continuous support, knowledge and patience throughout this research.

Additionally I would like to thank ir. Hans Huisman, ir. Henk van den Boom and dr. ir. Martin van Hees, who provided insight, that greatly assisted the research, with their expertise in the subject. Furthermore they showed me the importance of the subject and hereby created a valuable link in between the theoretical research and the maritime industry.

Contents

List of abbreviations

MARIN	Maritime Research Institute Netherlands
ISO	International Organization for Standardization
ITTC	International Towing Tank Conference

LIST OF PHYSICAL CONSTANTS

knot	$kn = 0.51444 ms^{-1}$
gravitational acceleration	$g = 9.81 ms^{-2}$
period of tidal current	$T_C = 12.42 hr$

List of Symbols

V	velocity	ms^{-1}
p	pressure	$kgs^{-2}m$
ρ	density	kgm^{-3}
A	area	m^2
S	wet surface area	m^2
R	resistance force	$kgms^{-2}$
μ	dynamic viscosity	$kgm^{-1}s$
ν	kinematic viscosity	m^2s^{-1}
τ	non-dimensional time	

INTRODUCTION

For centuries humans have been engaged in ship design and -building. Naturally ships always serve a certain purpose and one of the key purposes has always been the transportation of cargo. Nowadays huge bulk carriers and oil tankers are built by shipyards. The performance of these ships is of great importance for the buyer of this ship, both from a commercial- and an economical perspective. International laws have been established with regard to the emission a ship is allowed to emit. Furthermore the buyer obviously wants the ship to sail as fast as possible, with the least possible use of fuel. That is why the shipyard and the buyer settle on contract speeds: certain water speeds (speed relative to the water) the ship needs to reach for certain power settings.

To assess whether these requirements are met with, speed trials are conducted. These speed trials are necessary, since the water speed of a ship is an unknown variable that is hard to measure. The ground speed (speed that the ship travels on the map) can be measured accurately using GPS, but unfortunately this speed is usually not equal to the water speed, because of current effects. The speed of the current is an unknown parameter and complicates the problem.

Traditionally the "Mean of means" Method was used as a correction method for the current. This method is explained in section 2.3. However, once the ships were getting bigger, the speed trials were taking longer. This is because bigger ships are less viable and have a bigger mass inertia. The assumptions where the 'Mean of means' Method was based on do not hold for this longer time span. A recent new method that has been introduced is the "Iterative" Method, explained in section 2.4. At present this method is already being used in sea trial analysis. Unfortunately this method experiences issues, while being used for real speed trial data. These problems mainly come down on divergence of the method or convergence to an incorrect result.
The "Direct" Method is another method to determine the water speed. This method is based on a fundamental mathematical way of looking at the problem. The method is explained in section 2.5.
In this report both the "Iterative" Method and the "Direct" Method will be analysed. The central research question is as follows:

> What is the best possible current correction method for speed trial analysis?

To answer this question, the question is parsed into multiple sub questions:

- What is the cause of wrong convergence/divergence of the "Iterative" Method?
- What can be said about the reproducibility of both methods?
- Which algorithm can best be used for non-linear fitting?
- Which of the two methods has the best performance?

In order to answer these questions, both methods needed to be implemented in MATLAB, using various non-linear fitting techniques. Supplementary literature study was necessary in order to understand these algorithms and to rightfully apply them. These numerical methods are described in section 4. The MATLAB code can be found in appendix B.
Subsequently both methods were tested, using partly real data of model tests performed at MARIN (appendix C). For the function of the current, an academic example is taken. The MATLAB code was used in various ways in order to observe various aspects of both methods.

1

Theoretical ship behaviour

1.1. THEORETICAL SHIP BEHAVIOUR

1.1.1. INTRODUCTION

The aim of this research is to make an continued evaluation of the Iterative Method, since there are still some unanswered questions left. For understanding the motivation behind this method, a theoretical background is necessary. Only having insight in this, the chosen model of the Iterative Method can be justified.

For the completeness of this report, a brief overview of ship resistance is given in this chapter.

1.1.2. TYPES OF RESISTANCE

Newton's first law states that if a body experiences a netto force equal to zero, the body will move with a constant speed. Thus if approximations can be made for the resistance force that a ship experiences, the engine must produce a force that is exactly equal to this resistance force for the ship to move at a constant velocity. From this engine force, the delivered power of the engine can be calculated. This is the desired quantity for the Iterative Method. Needless to say, it is necessary to have insight in the nature of this resistance force.

When a body moves through water it experiences resistance forces opposing the motion. Since a ship moves through both water and air, it experiences both water and air forces that oppose the motion. Initially, the resistance in 'still water' without wind will be considered. For additional wind forces and water movements, adjustments can be made to correct the extra resistance due to these forces. For wind, the resistance will be extended with an extra wind force. Regarding water movement, the travelled distance can be corrected for the distance travelled due to water movement. However, unless the winds are very strong, the water resistance will be the most dominant component of the total resistance. Especially for big ships that will be considered in this study, the wind force is neglectable.

Van Manen and Oossanen [1] state that the total 'still water' resistance can be divided into three fundamental components:

- The frictional resistance due to the motion of a body through a viscous fluid or gas
- The wave-making resistance, caused by a varying pressure field that a body creates while moving through a 'still water' surface , which manifests itself in a wave pattern
- The eddy making resistance, which is due to the loss of energy by eddies shed from the appendages or the hull. An eddy is a contrary- or circular current, produced when a fluid flows past an obstacle that is not properly streamlined and in alignment with the flow.

FRICTIONAL RESISTANCE

With up to 80 percent of total resistance for slow-speed ships like supertankers and around 50 percent of total resistance in high-speed ships ([2]) , the frictional force is by far the most important resistance component that a ship endures.

This frictional resistance occurs due to the viscosity of the water the ship propagates through. The fluid moves past the body, such that the spacing of the streamlines gets more dense close to the body. Since the mass flow within a streamline is constant, the velocity of the flow changes. Evidently, this results in pressure changes as well. Bernouilli's principle of conservation of energy states the relation between these two quantities. In differential form, this equation is given by:

$$\frac{1}{2}\Delta V^2 + g\Delta z + \frac{\Delta p}{p} = 0 \tag{1.1}$$

where

ΔV : velocity difference of the fluid between two points in a streamline in $\left[\frac{m}{s}\right]$

Δp : pressure difference between those two points in a streamline in $\left[\frac{N}{m^2}\right]$

Δz : height difference between those two points in a streamline in $[m]$

ρ : fluid density in $\left[\frac{N\cdot s^2}{m^4}\right]$

g : gravitational acceleration in $\left[\frac{m}{s^2}\right]$

This form of Bernouilli's principle only holds for steady, incompressible fluids with negligible viscous forces and non-intersecting streamlines.
It can be shown that the netto force a ship experiences equals the following integral:

$$R_{drag} = \int_{A_{hull}} \Delta p \cdot n dA \tag{1.2}$$

where

$\Delta p = p - p_0$: pressure field on the hull, resulting from the ship's velocity in $\left[\frac{N}{m^2}\right]$

A_{hull} : area of the hull in $\left[m^2\right]$

n : normal vector of the hull surface

p_0 : hydrostatic water pressure in $\left[\frac{N}{m^2}\right]$

As a result of the viscosity of the water, the particles directly adjacent to the hull surface will be drawn to the surface and tend to move with the same speed of the ship. These particles altogether are called the boundary layer. At a distance from the hull, the water will be at rest. Keeping this in mind and assuming that the pressure is constant along the surface of the hull, the expression can be simplified to:

$$R_{drag} = \frac{1}{2} S\rho V^2 \tag{1.3}$$

where

$$\begin{aligned}
S &: \text{ wet frontal area of the ship in } \left[m^2\right] \\
\rho &: \text{ water density in } \left[\frac{N \cdot s^2}{m^4}\right] \\
V &: \text{ ship's velocity relative to the flow in } \left[\frac{m}{s}\right]
\end{aligned}$$

In the simplified model that was introduced here, this derivation of the drag force using Bernouilli's principle is valid. However, in reality the pressure along the hull surface is not constant, the boundary layer does not have the exact same velocity as the ship and the viscous forces can not be neglected. To correct the drag force for these deviations, a drag coefficient is used. This will be explained later in greater detail.

Furthermore there will also be a frictional force experienced by the above water part of the ship. However this resistance is caused by air instead of water. Since the density of air is about a thousand times less than that of water, the air resistance is likely to be small in comparison with the frictional force caused by water. As a result the air resistance is often neglected in ship resistance studies.

Wave-making resistance

When a ship moves through a fluid, the normal pressure acting on the hull varies along the hull. This variation in pressure causes a wave pattern, since the pressure at the surface needs to be constant and equal to atmospheric pressure. The waves in the wave system have both mass and velocity and as a result contain energy. This energy can only originate from the movement of the ship. Because of this reason, the resulting force of the wave pattern must always have the character of a drag force. Locally however, the pressure field can apply either a resistance force or a thrust force on the ship's hull. This drag force is called the wave-making resistance. The magnitude of the wave-making resistance is considered to be a function of both the ship's velocity and its length ([1]).

Kelvin showed that the wave system created by a ship is comparable to that of a moving pressure point in a fluid [3]. The wave pattern consists of two main features, namely diverging- and transverse waves. The crests of the diverging waves are inclined at an angle of almost 20° to the direction of motion. The crests of the transverse waves intersect the center line of the ship at a right angle.

The wave system of a ship can be approximated by the wave patterns of two pressure points: one located on the bow- and one located on the stern of the ship. Local discontinuities along the ship's wetted surface can also produce additional waves, but are negligible compared to the waves produced by bow and stern.

Since the wavelength is dependent on the ship's velocity and both wave patterns are moving with the ship, interference between the transverse waves of both systems can occur. This can result in a constructive interference, where crests of the two wave patterns coincide and produce waves of greater magnitude. Simply said: in this case wave energy of two systems are combined into one wave system. It can also work in an opposite fashion: where transverse waves of both wave patterns (partly) cancel each other out.

A characterisation of this behaviour is represented by the Froude number Fr, defined by:

$$Fr = \frac{V}{\sqrt{gL}} \tag{1.4}$$

where

$$\begin{aligned} V &: \text{ relative velocity of the ship in } \left[\frac{m}{s}\right] \\ L &: \text{ waterline length of the ship in } \left[\frac{m}{s}\right] \\ g &: \text{ gravitational acceleration of } 9.81\frac{m}{s^2} \end{aligned}$$

Clearly, the interference of the wave patterns is manifested in the several hump and hollows that occur in the resistance curve. Without this effect the resistance curve would be smoothly increasing.

According to van Manen and van Oossanen [1], the magnitude of the wave-making resistance rises strongly from a Froude number of 0.35 and maximizes at a Froude number around 0.5.

1.1.3. THE EDDY MAKING RESISTANCE

As said before, eddies are defined as being a swirling in a fluid or a reverse current that occurs when a fluid moves past an object. These appear when flow lines do not close in a body from behind properly, to balance out the pressure forces acting on the frontal part of the body. This is because of the viscosity of the fluid. Eddies can also be generated when the flow breaks away from the hull, because of rapid changes of section in the hull surface.

Ships have a lot of features that generate eddies. These features can include appendages, for example stabilizers rudders and shaft brackets. Eddies only appear in turbulent flows, so there is clearly a strong relationship between the Reynolds number and the eddy resistance. Appendages all have a characteristic length and the influence on the resistance of these features can be analysed by observing these appendages individually and scaling this to the ship. Luckily the additional resistance experienced due to the eddies is relatively little, about 10 to 15 per cent of the hull resistance according to Tupper [2]. Ignoring the influence of these appendages does therefore not appear to be critical.

1.1.4. DIMENSIONAL ANALYSIS

Sometimes a dimensional analysis can provide a lot of insight in what the actual function would approximately look like. In order to investigate if a dimensional analysis is of any use, the physical properties of interest will have to be analysed. In ship resistance studies, these properties are the gravitational constant g, the waterdensity ρ, the waterviscosity μ and the static waterpressure p. Consequently, if R is taken as the resistance, V as the velocity and L as a so-called typical length, a dimensional analysis will result in the following expression for resistance, according to Tupper [2]:

$$R = f[L^a V^b \rho^c \mu^d g^e p^f] \tag{1.5}$$

If the quantities are expressed in the fundamental terms time T, mass M and length L, the expression for resistance is equivalent to:

$$\frac{ML}{T^2} = f\left[L^a\left(\frac{L}{T}\right)^b\left(\frac{M}{L^3}\right)^c\left(\frac{M}{LT}\right)^d\left(\frac{L}{T^2}\right)^e\left(\frac{M}{LT^2}\right)^f\right] \tag{1.6}$$

The six unknown parameters can be reduced to three by equating the indices of the fundamental dimensions on both sides of the equation and substituting those expressions. Writing the expression in terms of the physical properties, the following relation holds:

$$R = \rho V^2 L^2 f\left[\left(\frac{\mu}{\rho VL}\right)^d, \left(\frac{gL}{V^2}\right)^e\left(\frac{p}{\rho V^2}\right)^f\right] \tag{1.7}$$

Furthermore the expression can also be written as:

$$R = \rho V^2 L^2\left[f_1\left(\frac{\mu}{\rho VL}\right), f_2\left(\frac{gL}{V^2}\right), f_3\left(\frac{p}{\rho V^2}\right)\right] \tag{1.8}$$

It is evident that the non-dimensional ratios $\frac{R}{\rho V^2 L^2}$, $VL\frac{\rho}{\mu}$, $\frac{V}{\sqrt{gL}}$, $\frac{p}{\rho V^2}$ are of great significance. These ratios are termed as follows:

- Resistance coefficient: $Rc = \frac{R}{\rho V^2 L^2}$
- Reynold's number: $Re = \frac{VL\rho}{\mu}$
  Reynold's number represents the ratio between the inertial- and the viscous forces within a fluid.
- Froude number: $Fr = \frac{V}{\sqrt{gL}}$
  The Froude number represents the ratio between the inertial force and the gravitational force and is used to write speed as a dimensionless quantity.
- Euler number: $Eu = \frac{p}{\rho V^2}$
  This ratio represents a characterization of the energy loss in a flow due to pressure differences.

As is shown the dimensional analysis results in a relationship between dimensionless quantities. Consequently there can not be said anything about the function form of the ship resistance. Nevertheless the dimensionless ratios contain some important information. One can expect that the frictional resistance is a function of Re, the wave-making resistance of Fr and the eddy- and appendage resistance of Eu and Rc.

1.1.5. EMPIRICAL EXPRESSIONS FOR RESISTANCE

Since ships have plenty of complex components and are thus often inconvenient shaped bodies, it is quite hard to calculate the flow around it. Especially analytically since there has not been found an analytic expression for the solution of the general Navier-Stokes equation. Obviously approximations of total ship resistance were necessary for the field of Naval Architecture, thus engineers were looking for empirical relations between ship characteristics and the ship resistance. In the course of the years, hundreds of papers and books are written about this subject. In this report, only the empirical Power-Velocity relation will be discussed, since this assumption is made for the "Iterative" method. The Power-Velocity is of the next form:

$$P(V_s) = a + bV_S^q \tag{1.9}$$

where

$$\begin{aligned} P &: \text{Delivered power by the ship in } [kW] \\ V_S &: \text{Water speed of the ship in } [kts] \\ a, b \text{ and } q &: \text{real constants} \end{aligned}$$

2

Existing current correction methods for speed trials

2.1. INTRODUCTION

In the course of the years there were a lot of methods invented to determine the current effect during speed trials. The most commonly used method for a long period of time was the so-called "Mean of means" Method. This method models the current speed as a polynomial. However this approximation is only valid for short periods of time. Once ships were getting bigger, the time duration of speed trials extended too. The polynomial approximation did not hold anymore and consequently the "Means of means" Method did not produce a reliable approximation. It became essential to find new accurate approximation methods.

2.2. SPEED TRIALS

Current correction methods are all based on data obtained through speed trials. There are certain requirements for a speed trial. In this section the procedure of speed trials will be explained. According to [4], the environmental conditions of the location need to be constant or at least expected to be constant. In that way the environment has only the smallest possible impact on the ship, in order to avoid unexpected environmental effects in the measurements. With constant environmental conditions it is meant that the location is a sheltered area (i.e. limited wind, waves and current), ideally free from hindrance by small boats and commercial traffic.

The speed trials are always performed in such a way that the ship sails exactly with the wind or in the exact opposite direction. If this was not the case, the shape of the topside of the ship (above the water) would provide a wind drift angle, since the biggest above-water part of the ship is usually located on the back of the ship. A wind drift results in an extra resistance force, since the rudder of the ship has to provide a counter force. If the current or waves make an angle with the ship, the ship will make a current drift. The effect of this drift is cancelled out, because the headway distance is taken for calculations. As stated in [4]: "the ship's speed during a speed run is derived from the headway distance between start and end position and the elapsed time of the speed run". Moreover the start point of each run will be taken exactly the end point of the run before. Each speed trial is performed with a specific power setting. When the ship is sailing with the right power and the measurement commences, the power handle will be released, i.e. during the speed trial no attempts will be made to let the ship remain sailing in the right power setting. This is because the fluctuations in delivered power will only worsen the accuracy of the measurement. As a result, the difference between the two Runs of a Double Run can differ widely.

During the speed trial, environment- and ship characteristics are measured, like the shift draft, wind speed and direction, and so forth. This report will not go into detail about those measurements and the analysis of those. Only the measurement of the time, the ground speed and the delivered power will be important for the current correction methods. The following assumptions are made:

- The time a Run takes place is taken to be the exact time on which the Run is halfway through

- The delivered power of a Run is taken to be the average power that the ship delivers during the Run
- The ground speed of a Run is taken to be the average ground speed of the ship during the Run

2.3. "Mean of means" Method

As said before, the "Mean of means" Method was traditionally used to determine the relative ship's speed. The method originates from the idea that if the current speed is constant, the effect of the current cancels out once you take the average of the measured ground speeds of the Double Run ([5]). However, evidently the current speed is not constant in general. If the speed trials are executed within a certain time frame for the same power setting, the current speed can be approximated by a polynomial (you could see this as the Taylor polynomial of an either unknown or difficult function). In this case, the idea behind the "Mean of means" Method can be extended in a way that it also works for these types of functions. For an approximation with a n^{th} order polynomial, a minimum of $n+2$ runs are needed. Note that a approximation with a polynomial of a higher order does not necessarily lead to higher accuracy, because if the trials take too much time, the polynomial approximation of the current speed is not valid anymore. Often the polynomial is chosen of order 2, 3 or 4.

This extension can be described as follows. If $n+2$ runs are carried out, the means will be the average of the 1st and 2nd measured speeds, the average of the 2nd and 3rd measured speeds and so forth. This will result in $n+1$ means. If the same procedure is applied for these means, there will be n mean of means left in the next step. This process can be repeated until there is only one value left. This value will be the approximation of the relative ship's speed.

By induction, it can be easily proven that the 'final' mean is calculated by:

$$V_{MoM} = \frac{1}{2^{n+1}} \sum_{i=1}^{n+2} \binom{n+1}{i-1} V_i \tag{2.1}$$

where

V_{MoM} : "Mean of means" approximation of relative ship's speed in $\left[\frac{m}{s}\right]$

V_i : measured (average) speed for run i in $\left[\frac{m}{s}\right]$

n : number of runs

The proof can be found in section 1 of appendix A.

2.3.1. Pascal's Triangle

From formula 2.1 follows that the different measurements are not represented equally in the final solution, because of the varying value of the binomial coefficient. Since this weight for a specific measurement is only influenced by this coefficient, this representation follows Pascal's Triangle. This will not make a difference in the final value of the algorithm, as will be proved in the following section 2.3.2.

2.3.2. CONVERGENCE OF "MEAN OF MEANS" METHOD

In this section, the convergence of the "Mean of means" Method will be proven for the case that the current speed is approximated by a polynomial of order n, i.e.:

$$V_C(t) = \sum_{j=0}^{n} a_j t^j \tag{2.2}$$

where $\{a_j\}_{j=0}^{n}$ is a series of unknown constants. Since the polynomial is of order n and thus has $n+1$ unknown constants, there are at least $n+2$ runs needed for the method to converge. The following assumptions are made:

- During the ten minutes of measurement in a run, the current speed is considered to be constant
- The time interval Δt between every measurement will be kept exactly the same. In that way the i^{th} measurement is measured on time $i\Delta t$.

Thus the measured speed V_i for the i^{th} run can be expressed as a polynomial, using the polynomial approximation for the current 2.2 at the time of the i^{th} measurement, i.e. at time $i\Delta t$:

$$V_i = V_S + (-1)^{i+1} \sum_{j=0}^{n} a_j (i\Delta t)^j \tag{2.3}$$

The contribution of the current to the measured speed is taken to be positive, if i is an odd integer. If i is an even integer, the contribution is negative. Obviously this is because in a Double Run one run is upstream and the other run is downstream.

Substituting 2.3 into 2.1, using the fact that $n+2$ runs are carried out, results in the next formula for the "Mean of means" approximation:

$$V_{MoM} = \frac{1}{2^{n+1}} \sum_{i=1}^{n+2} \left[\binom{n+1}{i-1} V_S + (-1)^{i+1} \sum_{j=0}^{n} a_j (i\Delta t)^j \right] \tag{2.4}$$

This formula can be rewritten in such a way that the "Mean of means" approximation equals the sum of V_S, the velocity that needs to be determined, and a remainder term, which appears to be zero. It is proven as follows:

$$\begin{aligned}
\frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}\left[V_S+(-1)^{i+1}\sum_{j=0}^{n}a_j(i\Delta t)^j\right] &= \frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}V_S+\frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}\sum_{j=0}^{n}a_j(i\Delta t)^j\\
&= \frac{V_S}{2^{n+1}}\sum_{i=0}^{n+1}\binom{n+1}{i}+\frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}\sum_{j=0}^{n}a_j(i\Delta t)^j\\
&= \frac{V_S}{2^{n+1}}2^{n+1}+\frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}\sum_{j=0}^{n}a_j(i\Delta t)^j\\
&= V_S+\frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}\sum_{j=0}^{n}a_j(i\Delta t)^j\\
&= V_S+\frac{1}{2^{n+1}}\sum_{j=0}^{n}a_j\Delta t^j\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}i^j
\end{aligned}$$

The third equality follows from the Binomial identity:

$$(x+y)^n=\sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k \tag{2.5}$$

with $x, y = 1$.
If the remainder term

$$\frac{1}{2^{n+1}}\sum_{j=0}^{n}a_j\Delta t^j\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}i^j=0 \quad \forall n\in\mathbb{N}$$

, the "Mean of means" Method converges to the right value. More specifically, it even holds that:

$$\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}i^j=0 \quad \forall n\in\mathbb{N}, j=0,\ldots,n \tag{2.6}$$

Consequently $V_{MoM} = V_S$. The proof for statement 2.6 can be found in appendix 2.

2.3.3. Application "Mean of means" Method

The "Mean of means" Method is still used nowadays, because it is very simple to apply and gives reasonable results. The Method particularly works well if the time between Runs is little, since a linear approximation of the current speed is very accurate. The "Mean of means" Method is applied as follows:

1. For every Double Run the averages of the powers and ground speeds are calculated
2. These averages form new approximation of (P, V_S) points
3. The curve from the model test is fitted to these (P, V_S) points

With the help of this fitted curve there can be decided whether or not the ship satisfies the contract conditions.

2.4. "ITERATIVE" METHOD

The "Iterative" Method has an advantage over the traditional "Mean of means" method, namely that it takes information about the tidal current into account. This improves the quality of the results for longer sea trials. In [4], it is stated that for official sea trials the tidal current will be of the form:

$$V_C(t) = A\cos\left(\frac{2\pi}{T_C}t\right) + B\sin\left(\frac{2\pi}{T_C}t\right) + Ct + D, \tag{2.7}$$

where

$V_C(t)$: current speed on time t in $[kts]$,
t : time in units of time $[T]$,
T_C : period of the dominant tidal constituent in units of time, i.e. $[T]$,

where the period of the dominant tidal constituent, called "the principal lunar semidiurnal period", is given by T_C = 12 hours, 25 minutes and 12 seconds. A, B, C and D are unknown constants.

ISO [4] states that a minimum of four Double Runs is needed, of which two carried out at contract speed. The "Iterative" Method can be schematically presented as in figure 2.1. In the next section the "Iterative" Method will be carefully explained step by step ([5], [6]).

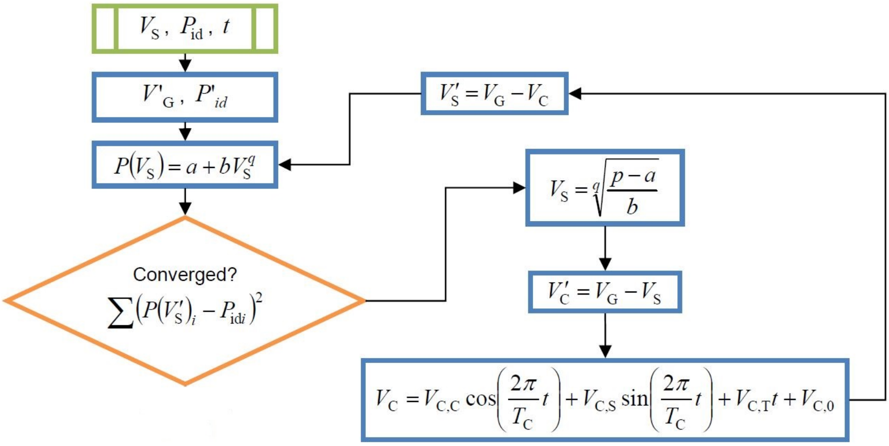
V_S, P_{id}, t
V'_G, P'_{id}
$P(V_S) = a + bV_S^q$
Converged?
$\sum(P(V'_S)_i - P_{idi})^2$
$V'_S = V_G - V_C$
$V_S = \sqrt[q]{\frac{p-a}{b}}$
$V'_C = V_G - V_S$
$V_C = V_{C,C}\cos\left(\frac{2\pi}{T_C}t\right) + V_{C,S}\sin\left(\frac{2\pi}{T_C}t\right) + V_{C,T}t + V_{C,0}$

Figure 2.1: Flow chart of the "Iterative" Method [4]

2.4.1. STAGE 1: INITIAL APPROXIMATION OF POWER/SPEED FUNCTION

It is assumed that delivered power by the engine and the ship's speed are related as follows:

$$P(V_s) = a + b \cdot V_s^q \tag{2.8}$$

or equivalently:

$$V_s = \left(\frac{P(V_s) - a}{b} \right)^{\frac{1}{q}} \tag{2.9}$$

where

$$\begin{aligned} P(V_s): &\text{ Engine power of the ship in } [kW] \\ V_s: &\text{ Speed of the ship relative to the water in } [kts] \end{aligned}$$

and a, b and q unknown constants. In reality the relation between these variables is a bit more complex, but this formula provides an accurate approximation.
As said before, a minimum of eight Runs (four Double Runs) are carried out. The values for both the measured powers and measured ground speeds will be corrected for the weather conditions (as described in [4]). Corrections that are made can be found in [4] as well. For each Double Run, the averages of both Runs will result in a (P, V_S) pair. Of course this start value for the iteration is not perfect, since the P-V_S relation is exponential and not linear. However this approximation should be reasonably close to the true value of the (P, V_S) pair. This process is represented in figure 2.1 as the arrow from the green box to the blue box directly below.

Consequently a first approximation of the power/speed regression curve is made by determining the unknown constants a, b, and q by fitting the (P, V_S) pairs to formula 2.8 through a non-linear least squares method. Often used non-linear least squares algorithms are the Levenberg-Marquardt algorithm and Trust Region Reflective algorithm. Both methods are described in chapter 4. This process is represented in figure 2.1 as the two arrows that lead from the blue box directly underneath the green box to the orange box. Often this approximation is not good enough yet, i.e. the error term is not converged. That is where the loop starts.

2.4.2. STAGE 2: INITIAL APPROXIMATION OF CURRENT SPEED FUNCTION

Applying the found constants on formula 2.9, new approximations of the ship's water speed run can be calculated. Now that the ship's speed is approximated, the current speed can be approximated too, since the ground speed V_G is measured and known. The current speed is thus simply approximated by:

$$V_C' = V_G \pm V_S \tag{2.10}$$

Since the time for each run is also known, this will result in (τ, V_C') pairs. The unknown constants in the current speed function 2.7 can now be approximated by fitting the formula to the pairs, using simple linear least squares fitting. Once the constants are found, the updated current speed V_C can be calculated using formula 2.10. This stage is represented by the three arrows in figure 2.1 leading from the orange box to the biggest blue box.

2.4.3. Stage 3: calculation of relative ship's speed

The last stage in the iteration is to update the relative ship's speed, using:

$$V_S' = V_G \pm V_C \tag{2.11}$$

With this updated V_S' and thus new (P, V_S') pairs, an updated regression curve can be fitted, using the same methods as stated before. This is represented by the two arrows leading from the biggest blue box back to the blue box, where formula 2.8 is written. Stages 2 and 3 will be repeated until the error-function 2.12 is converged.
The error function is given by:

$$\sum_i (P(V_S')_i - P_{id,i})^2 \tag{2.12}$$

where

$$\begin{aligned} P(V_S')_i &: \text{ approximated power for Run } i \text{ in } [kW], \\ P_{id,i} &: \text{ measured power for Run } i \text{ in } [kW] \end{aligned}$$

2.4.4. Ambiguity "Iterative" Method

In Toki's explanation of the "Iterative" Method [7], there still remains a lot of ambiguity in the "Iterative" Method. The three main points that are open for interpretation are:

- Stopping criterion
- Initial conditions
- Method of non-linear fitting

Stopping criterion

In order for the "Iterative" Method to stop, the error term 2.12 has to be converged. Note that this is not necessarily the same as minimized, because in the case of minimization the function can be minimized for the wrong (P, V_S) points. Since the existence of a limit is yet to be proven, the assumption of this existence has to be made. If the sequence of error terms $\{E_n(\boldsymbol{\beta})\}_{n\in\mathbb{N}}$ converges, then this sequence is also Cauchy (well known fact from the Theory of Real Analysis). Therefore the following stopping criterion is taken in this report:
If two consequent are sufficiently close enough, say 10^{-4}, then the loop stops and convergence is reached. Or mathematically stated in pseudo code:

```
Stop when:
```

$$|E_n(\boldsymbol{\beta}) - E_{n-1}(\boldsymbol{\beta})| < 10^{-4}$$

The value 10^{-4} is chosen, to be reasonably small, but also large enough to be above computer precision. Note that an approximation of the real error is used for the fitting, but not the real error. Obviously this is not possible, once the method is applied on non-academic data sets. However it is important to keep in mind that there are other error terms possible to find convergence. Different choices of error terms will not be dealt with in this analysis, but this is an idea for further research. Obviously the Error term depends on the input data and is not always of the same order of magnitude. It is however important to realise that this stopping criterion is based on the difference between subsequent iteration results. Thus the order of magnitude does not play a role in this matter.

Initial conditions

For every non-linear least squares fitting an initial parameter estimation is needed, regardless of the method used. Therefore this is not only an aspect of the "Iterative" Method, but also of the "Direct" Method (section 2.5). However, if the "Iterative" Method is used, initial (P, V_S) points have to be used. As explained in subsection 2.4.1, the "Mean of means" approximation is taken as these initial points. Note that this is certainly not the only possibility. In this report other initial points are not analysed, but it could be an interesting follow-up research to find even more judicious guesses for the initial (P, V_S) points.

Methods of non-linear fitting

The non-linear fitting of the functions can be performed with various non-linear least squares fitting methods. In this report only the Trust Region Reflective and the Levenberg-Marquardt Method are analyzed, since these are the most frequently used. An important difference between the two methods is that the Trust Region Reflective allows the setting of bounds for the parameters. The performance of both is described in chapter 5.

2.5. "Direct" Method

The "Direct" Method is a more mathematically intuitive method to assess the water speed of ships. Analogously to the "Iterative" Method it assumes the validity of formulas 2.7 and 2.8. However it is chosen to use dimensionless time in formula 2.7, i.e. the next formula is used:

$$V_C(\tau) = A\cos(2\pi\tau) + B\sin(2\pi\tau) + C\tau + D \tag{2.13}$$

where

$$V_C(\tau):\ \text{current speed on time } \tau \text{ in } [kts],$$
$$\tau : \text{dimensionless time, i.e. } \tau = \frac{t}{T_C}$$

where the period of the dominant tidal constituent, called "the principal lunar semidiurnal period", is given by $T_C = 12$ hours, 25 minutes and 12 seconds. A, B, C and D are unknown constants. The reason for this is explained in chapter 3.

In contrast to the "Iterative" Method, the "Direct" Method combines both formulas to find an expression for the power, dependent on both dimensionless time, ground speed and the unknown constants. To accomplish this, expressions 2.11 and 2.13 are substituted in formula 2.8. This results in the following expression for the delivered power, in terms of the ground speed, dimensionless time and the unknown constants:

$$P(V_G,\tau) = a + b\,[V_G \pm (A\cos(2\pi\tau) + B\sin(2\pi\tau) + C\tau + D)]^q \tag{2.14}$$

where:

$$\begin{aligned} V_G &: \text{ ground speed of the ship in } \left[\frac{m}{s}\right], \\ \tau &: \text{ dimensionless time, i.e. } \tau = \frac{t}{T_C} \end{aligned}$$

and $a, b, q, A, B, C, D \in \mathbb{R}$ unknown constants. Of course the ±-sign depends on if the Run is carried up- or downstream. For n Double Runs, the following system is obtained:

$$\boldsymbol{P}(\boldsymbol{V_G},\boldsymbol{\tau}) = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ \vdots \\ \vdots \\ \vdots \\ P_{2n} \end{pmatrix} = \begin{pmatrix} a + b\left[V_{G,1} + (A\cos(2\pi\tau_1) + B\sin(2\pi\tau_1) + C\tau_1 + D)\right]^q \\ a + b\left[V_{G,2} + (A\cos(2\pi\tau_2) + B\sin(2\pi\tau_2) + C\tau_2 + D)\right]^q \\ a + b\left[V_{G,3} + (A\cos(2\pi\tau_3) + B\sin(2\pi\tau_3) + C\tau_3 + D)\right]^q \\ a + b\left[V_{G,4} + (A\cos(2\pi\tau_4) + B\sin(2\pi\tau_4) + C\tau_4 + D)\right]^q \\ \vdots \\ \vdots \\ \vdots \\ a + b\left[V_{G,2n} + (A\cos(2\pi\tau_{2n}) + B\sin(2\pi\tau_{2n}) + C\tau_{2n} + D)\right]^q \end{pmatrix} \tag{2.15}$$

Subsequently this vector function will be fitted to the data points, using non-linear least squares method. Note that for this method, Double Runs need not necessarily to be performed with the same power setting. Furthermore this method can also be applied in the case that there detached single Runs have been carried out. In other words, the method also works in case an odd number of Runs is carried out.

In contrast with the "Iterative" Method, the solution is already found after one process of non-linear least squares fitting. Therefore the "Direct" Method is expected to be more flexible and faster than the "Iterative" Method. It is however of great importance that the initial guess is chosen reasonably close to the true value, since the parameter vector that needs to be fitted contains seven elements, instead of the three elements in the "Iterative" Method. Fortunately the extra four constants that need to be fitted are linear in the current function, so the problem is not as hard as it seems. The performance between the "Iterative" Method and the "Direct" Method is analysed in section 5.4.

3

New improvement of the "Iterative" Method

3.1. IMPROVEMENTS OF THE "ITERATIVE" METHOD

Despite the frequent use of the "Iterative" Method to determine the ship's speed at sea trials, there still exist some flaws in the algorithm. Analysis by various authors has already yielded some improvements. The first important improvement, found by my colleague Floris Buwalda [8], is the normalization of the current function. As we have seen before, the current function is given by

$$V_C(t) = A\cos\left(\frac{2\pi}{T_c}t\right) + B\sin\left(\frac{2\pi}{T_c}t\right) + Ct + D \tag{3.1}$$

according to equation 3.1 constants A, B and D have dimensions $[\frac{L}{T}]$, where constant C is of dimension $\left[\frac{L}{T^2}\right]$ (in fundamental terms). Evidently, to have a dimensionless number in the sine and cosine function, time t has to be chosen in the same time unit as the semidiuarnal period T_c is expressed. Thus constants A, B do not depend on the time unit used. On the other hand, constant C does depend on the time unit chosen. For example, if T_c is chosen in minutes instead of hours, t would have to be 60 times as big as well. In order for the current function to remain the same, constant C would have to be 60 times as small.

This dependency of constant C on the time unit chosen can result in problems, since the same linear least squares method is used to fit the four constants. C can be of a completely different order of magnitude than the other constants. Using the non-dimensional variable $\tau = \frac{t}{T_C}$ can prevent this from happening. Moreover, with this adaption, the Jacobian will also become independent of the time unit. This will improve the accuracy of the least squares fitting. In the remaining of this report, the new current function will be used:

$$V_C(\tau) = A\cos(2\pi\tau) + B\sin(2\pi\tau) + C\tau + D \tag{3.2}$$

where

$$V_C(\tau):\ \text{current speed on time } \tau \text{ in } [kts],$$
$$\tau: \text{dimensionless time, i.e. } \tau = \frac{t}{T_C}$$

4

Numerical Methods

4.1. INTRODUCTION

The "Iterative" Method, as described in [4], and the "Direct" Method (2.5) are mostly based on the fitting of data points to the assumed functions, to retrieve the unknown constants. Therefore (non-linear) least squares fitting is an important aspect, if not the most important aspect of both methods. In this chapter, the most commonly used numerical methods will be explained, regarding non-linear least squares fitting. Later on some of these non-linear least squares fitting methods are implemented in order to make simulations.

4.2. NON-LINEAR LEAST SQUARES FITTING

Two variables, say x and y, are subject to the relation $y = f(x, \boldsymbol{\beta})$ where $\boldsymbol{\beta} \in \mathbb{R}^m$ is a parameter vector and $f(x, \boldsymbol{\beta})$ is the nonlinear function that needs to be fitted. Given a set of data pairs $\{x_i, y_i\}$, the goal is to find a $\boldsymbol{\beta}$ that minimizes the sum of squares of the residuals. The same definition for the error term will be used as in [9]:

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{n}\left[y_i - f(x_i, \boldsymbol{\beta})\right]^2 \tag{4.1}$$

Note that the scalar in front of the error function does not influence the solution of the problem, as long as the scalar is positive. The scalar is chosen to be $\frac{1}{2}$, since this will cancel out if we look at the gradient of the error function. If a residual function is introduced, the error term can also be expressed as:

$$\begin{aligned} E(\boldsymbol{\beta}) &= \frac{1}{2}\sum_{i=1}^{n}(e_i(\boldsymbol{\beta}))^2 \\ &= \frac{1}{2}\boldsymbol{e}(\boldsymbol{\beta})^T\boldsymbol{e}(\boldsymbol{\beta}) \end{aligned} \tag{4.2}$$

where:

$$\begin{aligned} (\boldsymbol{e}(\boldsymbol{\beta}))_i &= e_i(\boldsymbol{\beta}) \\ &= y_i - f(x_i, \boldsymbol{\beta}) \end{aligned}$$

The non-linear minimization problem that needs to be solved is:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \quad E(\boldsymbol{\beta}) \tag{4.3}$$

or equivalently:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \quad \boldsymbol{e}(\boldsymbol{\beta})^T\boldsymbol{e}(\boldsymbol{\beta}) \tag{4.4}$$

For the minimum of this unconstrained minimization problem, the gradient at the minimum of this error function needs to be equal to zero. Thus this results in an system of equations (as many equations as the dimension of $\boldsymbol{\beta}$) of the form:

$$\begin{aligned}\frac{\partial E(\boldsymbol{\beta})}{\partial \beta_j} &= \sum_{i=1}^{n} e_i(\boldsymbol{\beta}) \frac{e_i(\boldsymbol{\beta})}{\partial \beta_j} \\ &= 0 \quad \forall j = 1, \ldots, m \end{aligned} \tag{4.5}$$

or equivalently:

$$\nabla E(\boldsymbol{\beta}) = J(\boldsymbol{\beta})^T \boldsymbol{e}(\boldsymbol{\beta}) = \boldsymbol{0} \tag{4.6}$$

where $J \in \mathbb{R}^{n \times m}$ is the Jacobian of $\boldsymbol{e}(\boldsymbol{\beta})$. Non-linear least squares fitting methods try to solve this system of non-linear equations. First the default non-linear fitting method of MATLAB will be explained, the Trust Region Reflective. This method will later be used for simulations. Subsequently two other common numerical methods will be clarified, namely the Gradient Descent Method and the Gauss-Newton Method ([10]). However both these methods have their up- and downsides. Luckily they can be combined in a smart way, which results in a new numerical method: the Levenberg-Marquardt method. This method will also be used for implementation.

4.3. TRUST REGION REFLECTIVE

According to [11] a lot of methods are based on the concept of trust regions. This is the idea that the function $f(\boldsymbol{x})$ that needs to be minimized, can be approximated with a much simpler function $q(\boldsymbol{x})$ in a certain neighborhood N around the current approximation $\boldsymbol{x}$ of the minimizer. This neighborhood is the so-called trust region. The updated value of the approximation is found by the following trust-region subproblem ([12]):

$$\min_{\boldsymbol{s} \in \mathbb{N}} q(\boldsymbol{s}) \tag{4.7}$$

In the case that:

$$f(\boldsymbol{x} + \boldsymbol{s}) < f(\boldsymbol{x})$$

the current point $\boldsymbol{x}$ is updated to be $\boldsymbol{x} + \boldsymbol{s}$. If this is not the case, the current point remains the same and the trust region will be reduced. Subsequently the step will be repeated until a minimizing step is found or $\boldsymbol{s} \approx \boldsymbol{0}$. Of course this approach raises some questions, like how to pick and compute the approximation q, defined in $\boldsymbol{x}$, how to choose and modify the trust region and how to solve the subproblem in an accurate way. Matlab uses Trust Region Reflective as the default algorithm to solve non-linear least squares method. In this standard trust-region method, the approximation q is defined by the quadratic version of the Taylor serie of f at $\boldsymbol{x}$, i.e. the subproblem is stated as:

$$\min \left\{ \frac{1}{2} s^T H(\boldsymbol{x}_k) s + s^T \nabla f(\boldsymbol{x}_k) \text{ such that } \|Ds\|_2 \leq \Delta \right\} \tag{4.8}$$

with $H(\boldsymbol{x}_k)$ and $\nabla f(\boldsymbol{x}_k)$ respectively the Hessian and gradient of function $f(\boldsymbol{x})$, at current point $\boldsymbol{x}_k$, $\Delta \in \mathbb{R}_{>0}$ and D a diagonal scaling matrix. Luckily there are very reliable algorithms to solve this minimization problem ([11]).

In pseudo code the Trust Region Reflective can be stated as:

1. Formulate the subproblem: $\min\{\frac{1}{2}s^T H(\boldsymbol{x}_k)s + s^T\nabla f(\boldsymbol{x}_k)$ such that $\|Ds\|_2 \le \Delta\}$
2. Solve the subproblem to determine the trial step s
3. If $f(\boldsymbol{x_k} + s) < f(\boldsymbol{x}_k)$, let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s$
4. Adjust Δ if needed

These steps will be repeated until convergence has been reached.

A more fundamental numerical method is the Gradient Descent Method. This method will now be described.

4.4. GRADIENT DESCENT METHOD

The Levenberg-Marquardt Method makes use of both the Gradient Descent Method and the Gauss-Newton Method. First these algorithms will be explained and considered what their up- and downsides are. Subsequently the idea behind the Levenberg-Marquardt Method will be substantiated.
Line search strategies are frequently used in non-linear minimization problems ([13]). The main idea is to produce a descending sequence $\{x_k\}_{k\in\mathbb{N}}$, given by:

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.9}$$

where

$$\begin{aligned} x_k(\in \mathbb{R}^n): &\text{ approximation of value after } k \text{ iterations} \\ \alpha_k(\in \mathbb{R}_{\ge 0}): &\text{ step size at iteration } k \\ p_k(\in \mathbb{R}^n): &\text{ search direction at iteration } k \end{aligned}$$

Note that $\alpha_k = 0$ if and only if x_k is optimal. Descent methods are used for minimization, which means that the sequence is produced such that:

$$F(x_{k+1}) < F(x_k) \tag{4.10}$$

and an equality if and only if x_k is optimal. In this case, F is the function that is to be minimized.

The gradient descent method is a line search method, which uses the most intuitive search direction, namely $-\nabla f_k$. It is a logical choice to choose the search direction along which f decreases most rapidly. The claim is that $-\nabla f_k$ is indeed the steepest descent. By the use of Taylor's theorem, we can write the value of the objective function at iteration k as:

$$F(x_k + \alpha_k p_k) = F(x_k) + \alpha_k p_k^T \nabla F_k + \frac{1}{2}\alpha_k^2 p_k^T \nabla^2 F(x_k + t p_k) p_k, \quad t \in (0, \alpha_k) \tag{4.11}$$

Hence, the unit search direction p_k with the most rapid decrease has to satisfy the following problem:

$$\min_{p_k} p_k^T \nabla F_k, \qquad \text{subject to } \|p_k\| = 1 \tag{4.12}$$

Clearly:

$$\begin{aligned} p_k^T F_k &= \|p_k^T\| \, \|\nabla F_k\| \cos(\theta) \\ &= \|\nabla F_k\| \cos(\theta) \end{aligned}$$

where θ is the angle between p_k and ∇f_k. This value is obviously minimized for:

$$\cos(\theta) = -1 \iff \theta = \pi \iff p_k = -\frac{\nabla F_k}{\|\nabla F_k\|}$$

Thus the claim was valid. In the case of the Error function, the steepest direction method would result in the following recursive formula:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha_k \frac{\nabla E(\boldsymbol{\beta}_k)}{\|\nabla E(\boldsymbol{\beta}_k)\|} \tag{4.13}$$

There are various methods to determine the step size α_k for every iteration, for example with the Wolfe conditions. However this will not be covered in this section.
The Gradient Descent method is recognized as a reliable algorithm for finding a local minimum of simple objective functions. However the method can turn out to be very slow. A method that is quite fast in general, is the Gauss-Newton method.

4.5. THE GAUSS-NEWTON METHOD

According to [14], the simplest method for non-linear least squares fitting is the Gauss-Newton method. This method can be seen as a modified Newton's method with line search. Provided that $\boldsymbol{e}(\boldsymbol{\beta}) : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ has continuous second partial derivatives, the Taylor expansion around a fixed $\boldsymbol{\beta}$ is:

$$\begin{aligned} \boldsymbol{e}(\boldsymbol{\beta} + \boldsymbol{h}) &= \boldsymbol{e}(\boldsymbol{\beta}) + J(\boldsymbol{\beta})\boldsymbol{h} + O(\|\boldsymbol{h}\|^2) \\ &\approx \boldsymbol{e}(\boldsymbol{\beta}) + J(\boldsymbol{\beta})\boldsymbol{h} \\ &\equiv \boldsymbol{l}(\boldsymbol{h}) \end{aligned} \tag{4.14}$$

where $J(\boldsymbol{\beta}) \in \mathbb{R}^{n \times m}$ is the Jacobian of $\boldsymbol{e}(\boldsymbol{\beta})$. Plugging this in definition 4.1 gives:

$$\begin{aligned}
E(\boldsymbol{\beta}+\boldsymbol{h}) &= \frac{1}{2}\boldsymbol{e}(\boldsymbol{\beta}+\boldsymbol{h})^T\boldsymbol{e}(\boldsymbol{\beta}+\boldsymbol{h}) \\
&\approx \frac{1}{2}\boldsymbol{l}(\boldsymbol{h})^T\boldsymbol{l}(\boldsymbol{h}) \\
&= \frac{1}{2}(\boldsymbol{e}(\boldsymbol{\beta})+J(\boldsymbol{\beta})\boldsymbol{h})^T(\boldsymbol{e}(\boldsymbol{\beta})+J(\boldsymbol{\beta})\boldsymbol{h}) \\
&= \frac{1}{2}\boldsymbol{e}(\boldsymbol{\beta})^T\boldsymbol{e}(\boldsymbol{\beta})+\boldsymbol{e}(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h}+\frac{1}{2}\boldsymbol{h}^TJ(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h} \\
&= E(\boldsymbol{\beta})+\boldsymbol{e}(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h}+\frac{1}{2}\boldsymbol{h}^TJ(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h} \\
&\equiv L(\boldsymbol{h}) \qquad (4.15)
\end{aligned}$$

The so-called Gauss-Newton step tries to find a $\boldsymbol{h}_*$ that minimizes $L(\boldsymbol{h})$:

$$\boldsymbol{h}_* = \min_{\boldsymbol{h}} L(\boldsymbol{h}) \qquad (4.16)$$

From the definition in it is easily seen that:

$$\boldsymbol{L}'(\boldsymbol{h}) = J(\boldsymbol{\beta})^T\boldsymbol{e}(\boldsymbol{\beta})+J(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h} \qquad (4.17)$$

$$\boldsymbol{L}''(\boldsymbol{h}) = J(\boldsymbol{\beta})^TJ(\boldsymbol{\beta}) \qquad (4.18)$$

Note that $L''(\boldsymbol{h})$ is independent of $\boldsymbol{h}$. Furthermore it is a symmetric matrix. If J has full rank, matrix $\boldsymbol{L}''(\boldsymbol{h})$ is positive definite. This means that there exists an unique minimizer for $L(\boldsymbol{h})$. For this minimizer, it holds that:

$$\boldsymbol{L}'(\boldsymbol{h}_*) = \boldsymbol{0} \qquad (4.19)$$

and thus this minimizer can be obtained by solving:

$$J(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h}_* = -J(\boldsymbol{\beta})^T\boldsymbol{e} \qquad (4.20)$$

This solution is a descent direction for $E(\boldsymbol{\beta})$, since:

$$\begin{aligned}
\boldsymbol{h}_*^T\nabla E(\boldsymbol{\beta}) &= \boldsymbol{h}_*^TJ(\boldsymbol{\beta})^T\boldsymbol{e}(\boldsymbol{\beta}) \\
&= -\boldsymbol{h}_*^TJ(\boldsymbol{\beta})^TJ(\boldsymbol{\beta})\boldsymbol{h}_* \\
&= -(J(\boldsymbol{\beta})\boldsymbol{h}_*)^TJ(\boldsymbol{\beta})\boldsymbol{h}_* \\
&< 0 \qquad (4.21)
\end{aligned}$$

The iterative step in the Newton-Gauss algorithm can be described in pseudo code as:

```
Solve: $J(\boldsymbol{\beta}_k)^TJ(\boldsymbol{\beta}_k)\boldsymbol{h}_k = -J(\boldsymbol{\beta}_k)^T\boldsymbol{e}(\boldsymbol{\beta}_k)$
Update: $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha_k\boldsymbol{h}_k$
```

According to [9] the method converges, provided that:

- $\{\boldsymbol{\beta}|E(\boldsymbol{\beta}) \leq E(\boldsymbol{\beta}_0)\}$ is bounded.
- Jacobian $J(\boldsymbol{\beta})$ has full rank in all steps.

As stated before, both the Gradient Descent Method and the Gauss-Newton Method have their pros and cons. To maximize the efficiency the best parts of the methods are used for a new method, the Levenberg-Marquardt method.

4.6. THE LEVENBERG-MARQUARDT METHOD

The method of gradient descent is considered to be a very reliable algorithm, since its convergence can be guaranteed under a lot of circumstances ([15]). For example if every step size is chosen through a line search that satisfies the Wolfe conditions and convexity of F. However the algorithm can turn out to be very slow. On the contrary, the Newton-Gauss method converges much faster in general, because of the convenient approximations for the Hessians and objective function. However the Newton-Gauss method has a smaller trust region and only converges under strict assumptions. In the Levenberg method a dampening factor $\lambda \geq 0$ is introduced ([16]), which varies the parameter updates between the gradient descent and the Gauss-Newton update ([17], [18]). Mathematically stated, the update follows from the system :

$$(J^T J + \lambda I)\boldsymbol{h}_{LM} = -J^T \boldsymbol{e} \tag{4.22}$$

Small values of λ coincide in a Gauss-Newton update, since:

$$-J^T \boldsymbol{e} = (J^T J + \lambda I)\boldsymbol{h}_{LM} \approx J^T J \boldsymbol{h}_{LM} \tag{4.23}$$

This is usually used if current iterate value is close to the solution, because of the convergence rate of the Gauss-Newton method. For large values of λ, the update corresponds with a small step in the steepest descent direction, since:

$$-J^T \boldsymbol{e} = (J^T J + \lambda I)\boldsymbol{h})_{LM} \approx \lambda I \boldsymbol{h}_{LM} \tag{4.24}$$

$$\Rightarrow \boldsymbol{h}_{LM} \approx -\frac{1}{\lambda} J^T \boldsymbol{e} = -\frac{1}{\lambda}\nabla E(\boldsymbol{\beta}) \tag{4.25}$$

Large values of λ are usually chosen at the start of the iteration process, if the current iterate is still far from the actual solution. With every update of the Levenberg algorithm, λ will be updated too. If the value of the error function decreases, the update is cancelled and λ is decreased for a higher convergence rate. On the other hand, if the approximation of the solution increases, the update is cancelled and λ is increased such that the iteration will again decrease. Often least squares problems are poorly scaled [14]. Some parameters can be of order 10^4, while orders can be of order 10^{-6}. Marquardt notices this problem and modified the Levenberg algorithm ([19]). This final modification in its whole is called the Levenberg-Marquardt method and the update of this algorithm is given by:

$$\left(J^T J + \lambda D_k^2\right)\boldsymbol{h}_{LM} = -J^T \boldsymbol{e} \tag{4.26}$$

where the scaling matrix D_k can change per iteration. D_k has to be a diagonal matrix with positive diagonal entries. Seber and Wild [20] proposed to take D_k such that

$$D_k^2 = \text{diag}(J_k^T J_k) \tag{4.27}$$

The argumentation behind this idea is that the algorithm will become invariant under diagonal scaling of the elements of $\boldsymbol{\beta}$. The iterative step in the Levenberg-Marquardt algorithm can be described in pseudo code as:

$$\begin{aligned}&\text{Solve: } \left(J_k^T J_k + \lambda \text{diag}(J_k^T J_k)\right)\boldsymbol{h}_k = -J_k^T \boldsymbol{e}(\boldsymbol{\beta}_k)\\ &\text{Update: } \boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \boldsymbol{h}_k\end{aligned}$$

4.7. MEASUREMENT ERRORS

In a similar fashion, the same analysis can be done, with an additional weighting matrix W, that takes measurements errors into account. Typically W is a diagonal matrix with $W_{ii} = \frac{1}{\sigma_{y_i}^2}$, where σ_{y_i} is the measurement error for the measurement $y(x_i, \boldsymbol{\beta})$. This will result in the following pseudo codes:
Gradient descent:

$$\begin{aligned}&\text{Calculate: } \nabla E(\boldsymbol{\beta}_k) = J(\boldsymbol{\beta}_k)^T W \boldsymbol{e}(\boldsymbol{\beta}_k)\\ &\text{Update: } \boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha_k \frac{\nabla E(\boldsymbol{\beta}_k}{\|\nabla E(\boldsymbol{\beta}_k\|}\end{aligned}$$

Gauss-Newton:

$$\begin{aligned}&\text{Solve: } J(\boldsymbol{\beta}_k)^T W J(\boldsymbol{\beta}_k)\boldsymbol{h}_k = -J(\boldsymbol{\beta}_k)^T W \boldsymbol{e}(\boldsymbol{\beta}_k)\\ &\text{Update: } \boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha_k \boldsymbol{h}_k\end{aligned}$$

Levenberg-Marquardt:

$$\begin{aligned}&\text{Solve: } \left(J_k^T W J_k + \lambda \text{diag}(J_k^T W J_k)\right)\boldsymbol{h}_k = -J_k^T \boldsymbol{e} W(\boldsymbol{\beta}_k)\\ &\text{Update: } \boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \boldsymbol{h}_k\end{aligned}$$

5
Results

5.1. DATA

For the simulations made, the data of various model tests are used. The motivation behind this is to only analyse the performance of the current correction methods under the presence of measurement noise. Model tests are performed in a towing tank, without any additional wave- or wind resistance. Furthermore the water speed is a known value in a model test. In order to simulate the current correction methods, an academic example of a current function is used. This current function is chosen to be positive at all times, such that downstream- and upstream Runs alternate. The parameters are chosen to be:

$$\begin{aligned} A &= 1.5 \\ B &= 1.3 \\ C &= 0.8 \\ D &= 1.5 \end{aligned}$$

For every simulation that is carried out (simulation 1.1 up to and including simulation 3.2), the associated data sets can be found in appendix C. In this appendix, the used (non-dimensional) time vectors, power vectors and ground speed vectors are given.

5.2. "ITERATIVE" METHOD VS "DIRECT" METHOD

In this section, the performances of the "Iterative" Method and the "Direct" Method will be analyzed, through six partly academic simulations, with varying parameters. The differences between these simulations are chosen mainly to observe the influence of the differences in delivered power (both ’small’ and ’big’ ships), the number of Runs, the measurement noise on specific variables and the interval of the water speed. Every simulation is carried out 1000 times. The average of the errors are compared between different methods. The following abbreviations and errors are used:

	Definition
MN	Measurement Noise
IM-TRR	"Iterative" Method, using Trust Region Reflective
IM-LM	"Iterative" Method, using Levenberg Marquardt
DM-TRR	"Direct" Method, using Trust Region Reflective
DM-LM	"Direct" Method, using Levenberg Marquardt
E_P	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(P(V_{S,i})-P_i)^2}$
E_{V_C}	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(V_C(\tau_i)-V_{C,i})^2}$
E1	E_P of IM-TRR
E2	E_P of IM-LM
E3	E_P of DM-TRR
E4	E_P of DM-LM
E5	E_{V_C} of IM-TRR
E6	E_{V_C} of IM-LM
E7	E_{V_C} of DM-TRR
E8	E_{V_C} of DM-LM
$\overline{E*}$	mean (out of 1000 runs) of error $E*$ where $* \in \{1,2,...,8\}$

5.3. MEASUREMENT NOISE

In order to evaluate the difference between the methods, normally distributed measurement noise was added to the data discussed in 5.1. The mean of the normally distributed noise is always taken as 0. The standard deviation depends on the parameter. The standard deviation is taken as one-third of the maximum error, such that 99.73 % of the measurement errors produced, lie within the interval of maximum errors.

The following maximum errors are used:

- The desired maximum error in the measurement of power is 25 kW (according to Naval Architect Hans Huisman)

- A maximum error of 0.05 kts is taken for the measurement of ground speed [8]

- The maximum measurement error of the time is taken to be 36 seconds [8]

To check which noise gives the most trouble, the simulations are also carried out with only noise on respectively the ground speed, the time and the delivered power. This resulted in the following tables:

	With MN	Only MN for V_G	Only MN for τ	Only MN for P
$\overline{E1}$	27.2683	26.6745	15.8345	16.2969
$\overline{E2}$	27.2751	26.6683	15.8249	16.2975
$\overline{E3}$	37.6322	36.8216	15.6210	16.7799
$\overline{E4}$	37.6517	36.8216	15.6341	16.7910
$\overline{E5}$	0.0106	0.0102	0.0013	0.0028
$\overline{E6}$	0.0106	0.0102	0.0012	0.0028
$\overline{E7}$	0.0117	0.0113	0.0015	0.0028
$\overline{E8}$	0.0118	0.0113	0.0015	0.0028

Table 5.1: Effect of noise, simulation 1.1

	With MN	Only MN for V_G	Only MN for τ	Only MN for P
$\overline{E1}$	75.1424	74.9381	72.1807	72.3705
$\overline{E2}$	75.5081	75.3119	72.5957	72.7814
$\overline{E3}$	63.5869	62.8880	56.2183	56.5816
$\overline{E4}$	63.5869	62.8901	56.2183	56.5824
$\overline{E5}$	0.0621	0.0611	0.0611	0.0610
$\overline{E6}$	0.0621	0.0620	0.0620	0.0619
$\overline{E7}$	0.0221	0.0220	0.0180	0.0184
$\overline{E8}$	0.0221	0.0220	0.0180	0.0184

Table 5.2: Effect of noise, simulation 1.2

	With MN	Only MN for V_G	Only MN for τ	Only MN for P
$\overline{E1}$	120.7689	118.2249	72.3666	78.2625
$\overline{E2}$	132.7752	130.1251	76.5222	71.2028
$\overline{E3}$	133.3464	132.5885	69.3641	68.8340
$\overline{E4}$	133.0717	132.4448	69.0242	68.5866
$\overline{E5}$	0.0119	0.0115	0.0022	0.0028
$\overline{E6}$	0.0125	0.0130	0.0026	0.0017
$\overline{E7}$	0.0118	0.0115	0.0034	0.0033
$\overline{E8}$	0.0118	0.0115	0.0033	0.0033

Table 5.3: Effect of noise, simulation 2.1

	With MN	Only MN for V_G	Only MN for τ	Only MN for P
$\overline{E1}$	122.9398	126.6331	89.6081	89.1764
$\overline{E2}$	127.6453	128.7517	90.4859	90.5551
$\overline{E3}$	124.5025	126.8945	67.8445	67.5090
$\overline{E4}$	124.5743	126.7790	67.6222	67.3098
$\overline{E5}$	0.0150	0.0148	0.0095	0.0095
$\overline{E6}$	0.0144	0.0138	0.0090	0.0088
$\overline{E7}$	0.0142	0.0140	0.0078	0.0078
$\overline{E8}$	0.0143	0.0140	0.0079	0.0079

Table 5.4: Effect of noise, simulation 2.2

	With MN	Only MN for V_G	Only MN for τ	Only MN for P
$\overline{E1}$	195.6527	195.2012	191.9206	192.9281
$\overline{E2}$	222.9617	238.4654	235.4622	236.9145
$\overline{E3}$	41.5872	40.6229	36.2987	36.9153
$\overline{E4}$	41.5835	40.5794	36.2604	38.8818
$\overline{E5}$	0.5439	0.5442	0.5409	0.5420
$\overline{E6}$	0.4252	0.4239	0.4253	0.4260
$\overline{E7}$	0.0333	0.0314	0.0302	0.0307
$\overline{E8}$	0.0330	0.0309	0.0297	0.0303

Table 5.5: Effect of noise, simulation 3.1

	With MN	Only MN for V_G	Only MN for τ	Only MN for P
$\overline{E1}$	187.9131	187.7091	188.3313	188.2271
$\overline{E2}$	198.6361	197.7476	201.7287	192.0345
$\overline{E3}$	58.9172	58.4425	56.3291	56.7464
$\overline{E4}$	58.9212	58.4435	56.3270	57.7441
$\overline{E5}$	0.6131	0.6130	0.6125	0.6129
$\overline{E6}$	0.6484	0.6574	0.6759	0.6642
$\overline{E7}$	0.0695	0.0680	0.0673	0.0678
$\overline{E8}$	0.0694	0.0679	0.0672	0.0677

Table 5.6: Effect of noise, simulation 3.2

From the simulations, it clearly comes out that for the performed simulations, the measurement noise for variable V_G has by far the most effect on the total outcome. That is, it has the biggest negative impact on the results in average. Since the time- and ground speed vectors remain in the same order of magnitude, the effects for each simulation are more or less the same. However the effect of measurement noise for the delivered power can vary largely in between simulations. Logically this noise makes very little difference for simulation 2.1 and 2.2, since the delivered power is much bigger than in the other simulations. Evidently the noise will be a smaller percentage of the delivered power in this simulation and thus have less influence on the fitting.

5.4. RESULTS

The results from the simulations are given by:

	Without MN	With MN	8 Runs (without last DR)	8 Runs (without middle DR)
$\overline{E1}$	15.5096	27.2683	38.1573	22.2606
$\overline{E2}$	15.5016	27.2751	37.0640	22.3006
$\overline{E3}$	15.3047	37.6322	39.0031	36.4072
$\overline{E4}$	15.3047	37.6517	38.9998	36.4155
$\overline{E5}$	0.0005	0.0106	0.0156	0.0117
$\overline{E6}$	0.0004	0.0106	0.0153	0.0118
$\overline{E7}$	0.0011	0.0117	0.0145	0.0133
$\overline{E8}$	0.0011	0.0118	0.0145	0.0133

Table 5.7: Results simulation 1.1

	Without MN	With MN	8 Runs (without last DR)	8 Runs (without middle DR)
$\overline{E1}$	72.1232	75.1424	47.2946	66.0671
$\overline{E2}$	72.5363	75.5081	47.1464	65.9531
$\overline{E3}$	56.1472	63.5869	43.2975	58.8061
$\overline{E4}$	56.1472	63.5869	43.2972	58.8058
$\overline{E5}$	0.0610	0.0612	0.0349	0.0367
$\overline{E6}$	0.0620	0.0621	0.0345	0.0362
$\overline{E7}$	0.0179	0.0221	0.0207	0.0357
$\overline{E8}$	0.0179	0.0221	0.0207	0.0358

Table 5.8: Results simulation 1.2

	Without MN	With MN	8 Runs (without first DR)	8 Runs (without middle DR)
$\overline{E1}$	71.3158	120.7689	173.3588	166.3249
$\overline{E2}$	71.2384	132.7752	139.6555	126.9443
$\overline{E3}$	68.3178	133.3464	144.3827	153.0499
$\overline{E4}$	69.2502	133.0717	144.4060	152.8960
$\overline{E5}$	0.0018	0.0119	0.0174	0.0167
$\overline{E6}$	0.0017	0.0125	0.0143	0.0121
$\overline{E7}$	0.0032	0.0118	0.0142	0.0137
$\overline{E8}$	0.0032	0.0118	0.0142	0.0137

Table 5.9: Results simulation 2.1

	Without MN	With MN	8 Runs (without first DR)	8 Runs (without middle DR)
$\overline{E1}$	88.9418	122.9398	126.9183	139.7789
$\overline{E2}$	89.9528	127.6453	141.0659	138.5970
$\overline{E3}$	66.9680	124.5025	125.1196	146.2348
$\overline{E4}$	66.9300	124.5743	125.3697	146.3506
$\overline{E5}$	0.0095	0.0150	0.0157	0.0177
$\overline{E6}$	0.0087	0.0144	0.0160	0.0171
$\overline{E7}$	0.0077	0.0142	0.0159	0.0161
$\overline{E8}$	0.0078	0.0143	0.0160	0.0162

Table 5.10: Results simulation 2.2

	Without MN	With MN	8 Runs (without first DR)	8 Runs (without middle DR)
$\overline{E1}$	192.0725	195.6527	178.5741	162.1266
$\overline{E2}$	189.0745	222.9617	38.9607	247.3323
$\overline{E3}$	36.2841	41.5872	28.4146	36.9910
$\overline{E4}$	36.2263	41.5835	25.6604	36.9726
$\overline{E5}$	0.5411	0.5439	0.2149	0.5686
$\overline{E6}$	0.1987	0.4252	0.0282	0.2354
$\overline{E7}$	0.0303	0.0333	0.0222	0.0610
$\overline{E8}$	0.0296	0.0330	0.0219	0.0611

Table 5.11: Results simulation 3.1

	Without MN	With MN	8 Runs (without first DR)	8 Runs (without middle DR)
$\overline{E1}$	192.0725	195.2012	178.5741	162.1266
$\overline{E2}$	140.2041	198.6361	87.5697	234.5306
$\overline{E3}$	56.3148	58.9172	35.5060	59.3848
$\overline{E4}$	56.3148	58.9212	24.6920	60.0274
$\overline{E5}$	0.6127	0.6131	0.7938	0.5906
$\overline{E6}$	0.5317	0.6484	0.1240	0.8609
$\overline{E7}$	0.0672	0.0695	0.0696	0.0810
$\overline{E8}$	0.0672	0.0694	0.0576	0.0814

Table 5.12: Results simulation 3.2

5.4.1. GENERAL OBSERVATIONS

Note that the most important error for the application of the methods is E_{V_C}, since the goal is to approximate the current as good as possible. In comparing the different methods, this error will implicitly play a bigger role.
Apart from the actual simulation results, the simulations with less data (8 Runs) took way longer. This resulted in long run times for the simulations, since 1000 Runs were executed for every data set combination. As stated before in this report, the problem

needs to be well defined and this is usually achieved by adding more Runs. This is probably the reason that speed trials with less (Double) Runs can result in more calculation time. Furthermore simulation 3.1 and simulation 3.2 took significantly longer than the other simulations. For some reason this data set is hard to cope with for the methods.

5.4.2. Interpretation simulation results

If there is a local minimum that minimizes the error term E_P both the Trust Region Reflective and the Levenberg-Marquardt algorithm should yield the same answer. To obtain the exact same answer is however impossible, since there will always be errors made on noise level. Furthermore, specifically in the "Iterative" Method, the rate of convergence does not have to be the same, but can be fitting method- dependent. Since the value of the stopping criterion is set on 10^{-4}, a slower convergence will lead to a premature stop of the iterative loop. This is because the steps could be so small, that they become smaller than the stopping criterion. Fortunately, no major differences were seen in the results by adjusting the stopping criterion. The results are more or less the same. The calculation time however increases drastically for a smaller stopping criterion value, so that is why a stopping criterion of 10^{-4} is preferred. In that way the method is both fast and accurate enough.

A quick observation shows us that in general the error terms turn out higher for the simulations with non-coinciding points ('impure' Double Runs), i.e. simulations 1.1, 2.1 and 3.1. This was expected, since less points need to be fitted, using 'pure' Double Runs. The fitting can be made way more precise. It can be concluded from these simulations that it is always a good idea to do a minimum of two Runs for the contract speed, of course provided that there are enough Runs performed on different power settings, but this is already concluded in ISO [4]. The closer the delivered powers for these Runs, the better the fitting result.

It is difficult to say something about the difference between the speed trials with 10 Runs and the speed trials with 8 Runs. This strongly depends on the input data. For example the regular 10 Run speed trial performance is better for simulation 2.1 and 2.2. In other cases either 8 Runs (without middle DR) and 8 Runs (without first/last DR) perform better. Of course in the case of 8 Runs, fitting can be better or worse for the used points, but the error for the left out Double Run is certainly worse. It is thus important to perform Double Runs for at least all the contract speeds, in order to get a good result.

The most remarkable aspect of the simulation results is that the "Direct" Method performs better in almost all the simulations. Furthermore in the only simulation where the "Iterative" Method performs better (simulation 1.1), the difference is not that big. Of course this conclusion is only drawn based on these specific academic simulations, but it sure looks promising.

Another interesting aspect of the "Direct" Method is the calculation time. The "Direct" Method only needs to perform a non-linear fitting once. The "Iterative" Method on the other hand contains a loop, in which two fittings take place: a linear- and a non-linear fitting. This means that if the method stops after 100 iterations, 100 non-linear fittings are carried out. Obviously this results in a significantly larger calculation time than that of the "Direct" Method.

As can be seen in the results of simulations 3.1 and 3.2, the "Iterative" Method gives a poor estimate parameter vector. The reason for this is unclear. The only difference that can be found between the data sets is that the power settings of simulation 3.1 and 3.2 are a lot smaller than the remainder of the simulations. However theoretically there is no clue to assume this should result in a misfit. Additionally the "Iterative" Method does not even give the same estimation for the Trust Region Reflective - and Levenberg-Marquardt algorithm. Maybe the "Iterative" Method provides another local minimum (but not global) as a result. This will be analysed in section 5.4.4. It could also be the case that something went wrong in the non-linear fitting process. In some cases of measurement noise, this could be fixed by tweaking the parameters of the Matlab function. For example it sometimes lowering the Function Tolerance, Optimality Tolerance and/or Step Tolerance from 10^{-6} to 10^{-8} or even smaller. Unfortunately this results in non-proportionate computation times, that could be up to a few hours per simulation. Naturally simulating this is close to impossible time wise, especially given that proper convergence for all noise additions is still not assured with these tweaked parameters. It can be concluded that the "Iterative" Method is simply not working the way it is desired for this specific data set. The "Direct" Method on the other hand, proves itself to be a very robust algorithm. All the results seem to be quite accurate and no major differences can be seen by changing the non-linear fitting method. Consequently it looks like a trustworthy method, that can either be used to provide the curve fitting needed, or provide an initial estimate for the "Iterative" Method.

5.4.3. LIMITING PARAMETERS

The parameters found for the fitting are usually not realistic. For example the exponential parameter q is expected to be somewhere around 3. However, in simulation 3.2 (without measurement noise) this parameter is estimated to be around 8.7. Even though this is not a realistic value for the behaviour of the ship resistance, for all $0 \leq V_S < \infty$, it can be the best approximation for a small interval of the positive real axis. Furthermore since we are only interested in the fitting of the current speed in the end, the values do not have to be realistic in order to get a realistic approximation of the current speed.

Yet, this only works when this unrealistic value gives indeed the best approximation for the P-V relation on this interval. The Trust-Region Reflective gives us the freedom to set bounds for the to-be-fitted parameters. In this fashion we can observe the effect of setting bounds for q. This will be exclusively for q, since this is the only parameter where we have an idea about the order of magnitude. Namely we can not say a lot more about a and b, apart from that they both should be positive. Parameters A, B, C and D are even more difficult to get a hold on. As lower bound for q, 2.5 is chosen. The upper bound will be set on 4. An example simulation with bounds for q is given by table 5.13.

	With bounds	Without bounds
E1	255.6652	187.9131
E3	164.3271	58.9172
E5	0.2063	0.6131
E7	0.2242	0.0695

Table 5.13: Simulation 3.2 (10 Runs), with and without bounds

As can be seen in table 5.13, E_{V_C} becomes smaller for the "Iterative" Method, by setting bounds. However this is not a fair comparison, since the fitting without bounds was already considered to be wrong. On top of that, the errors are still way larger than desired. If we look at the error terms for the "Direct" Method, they become remarkably larger. Whereas this initial fitting was already very accurate, setting bounds only worsens the performance of the "Direct" Method in this case. If the same bounds are applied on the other simulations, they all confirm this statement. Thus it is important to keep in mind that unrealistic values of parameters do not necessarily imply bad performance of the method. As seen here, they even perform better than using the method with bounds.

5.4.4. INITIAL VALUE

Of course the fitting of the curves heavily depends on the initial estimate of the parameter vector that is used for the fitting. This initial estimate can luckily be chosen in a smart way. For example, it is known that the parameter a of the P-V relation 2.8 should be larger than 0 (it represents the stationary resistance), but smaller than the value of the smallest power setting. The variables b and q should be positive and somewhere in the order of magnitude 10^0, which we know from fundamental estimations in the calculation of ship resistance. For the "Direct" Method, estimates for the parameters of the current function are also needed. Constant C is usually close to 0 (otherwise the current would have a large trend). Constant D is the value that can be seen as the equilibrium position. The sinusoid that represents the current curve oscillates around this value. This parameter can thus be taken somewhere around half of the difference from two Runs in the same Double Run. It is a lot harder to get a grip on parameters A and B, so it is of great importance to choose these parameters in a neutral way. In the simulations all these values are chosen to be 1. This is the thought process that lies behind the choices of initial parameter vector, that can be found under the figures for each simulation.

The initial estimates of the simulations seem to work out nicely, except for simulations 3.1 and 3.2. The "Iterative" Method in particular fails here. It could be that the initial estimate plays a role in this failure. To check this, distinctive estimates are used in the case that there is no measurement noise. Note that C and D are kept the same, since there is just been argued that these are reasonably easy to estimate. The results of these simulations can be found in tables 5.14 and 5.15.

The other distinctive initial estimates do not seem to have an effect on the results for simulation 3.1. The "Iterative" Method still gives bad results, but the "Direct" Method did not budge, except for the case that all five variables are taken distinctive of the ones used in the simulations (the last two columns).

$\boldsymbol{x}_0$	$\begin{bmatrix}10\\2\\3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\2\\3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}10\\0.2\\0.3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\0.2\\0.3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}10\\2\\3\\10\\10\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\2\\3\\10\\10\\1\\1\end{bmatrix}$	$\begin{bmatrix}10\\0.2\\0.3\\10\\10\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\0.2\\0.3\\10\\10\\1\\1\end{bmatrix}$
E1	209.7860	251.1788	197.5166	218.8581	209.7860	251.1788	197.5166	218.8581
E2	160.8347	282.7447	204.5678	272.2534	160.8347	282.7447	204.5678	272.2534
E3	36.2263	36.2299	36.2278	36.6347	36.2790	38.3074	179.6543	286.0984
E4	36.2263	36.2263	36.2263	36.2263	36.2263	36.2263	1272.9	1272.9
E5	0.5589	0.3704	0.5463	0.3080	0.5589	0.3704	0.5463	0.3080
E6	0.3253	0.7561	0.7282	0.3310	0.3253	0.7561	0.7282	0.3310
E7	0.0296	0.0295	0.0294	0.0315	0.0331	0.0312	0.4412	0.6963
E8	0.0296	0.0296	0.0296	0.0296	0.0296	0.0296	7067.8	7154.4

Table 5.14: Results simulation 3.1 with various initial estimates

$\boldsymbol{x}_0$	$\begin{bmatrix}10\\2\\3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\2\\3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}10\\0.2\\0.3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\0.2\\0.3\\1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix}10\\2\\3\\10\\10\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\2\\3\\10\\10\\1\\1\end{bmatrix}$	$\begin{bmatrix}10\\0.2\\0.3\\10\\10\\1\\1\end{bmatrix}$	$\begin{bmatrix}1000\\0.2\\0.3\\10\\10\\1\\1\end{bmatrix}$
E1	188.3137	188.3298	188.1081	188.3972	188.3137	188.3298	188.1081	188.3972
E2	222.6484	228.8363	116.3035	277.4651	222.6484	228.8363	116.3035	277.4561
E3	56.3149	56.3148	56.3164	56.3452	56.3148	56.3148	56.3164	56.3148
E4	56.3148	56.3148	56.3148	56.3148	56.3148	56.3148	1159.5	1159.5
E5	0.6129	0.6128	0.6120	0.6124	0.6129	0.6128	0.6120	0.6124
E6	0.6762	0.7044	0.5010	0.5028	0.6762	0.7044	0.5010	0.5028
E7	0.0671	0.0672	0.0668	0.0685	0.0672	0.0672	0.0672	0.0672
E8	0.0672	0.0672	0.0672	0.0672	0.0672	0.0672	795.89 + 66.090i	805.61 + 98.404i

Table 5.15: Results simulation 3.2 with various initial estimates

For simulation 3.2, not a lot of differences can be seen either. The "Iterative" Method, using the Levenberg-Marquardt method behaves better when $a = 10$, $b = 0.2$ and $q = 0.3$. However once $a = 1000$ it behaves far worse. All the other initial values still give similar bad results as the one used for the simulations.
The "Direct" Method behaves quite similar for all the different initial parameter vectors. Only in the case that all five variables are chosen differently, the "Direct" Method using Levenberg-Marquardt fails (it even takes complex values).

It can be concluded that, if the initial estimate of the parameter vector is reasonably close, the methods should converge to the right value (that is, if no other causes for wrong convergence or divergence are present). Additionally, the Trust Region Reflective Method seems to be more robust, i.e. it is more likely to converge to the right value in the case that the initial value is chosen far from the actual value. It is thus recommended to use the Trust Region Reflective. However note that this is not yet 'hard' proof, since it is only shown to be the case for this example.

5.4.5. ERROR DISTRIBUTION

The error distributions of the measured errors E1 up to E8 usually follow more or less the same distribution, if the fitting gives a proper result. The distributions generally look similar to a normal distribution or part of a normal distribution. This is an expected result, since the noise was generated with the use of a normal distribution. In order to give an idea of the distributions, an example will be shown of simulation 1.2, using 10 Runs with measurement noise, in appendix D.

5.4.6. EXAMPLES OF FITTING

Two visualise the outcome of the simulations, two examples of fittings will be shown in figures here, simulation 2.2 and simulation 3.2. Notice that the fitting for simulation 2.2 more or less works in all cases, but the fitting for simulation 3.2 fails for the "Iterative" Method.

SIMULATION 2.2

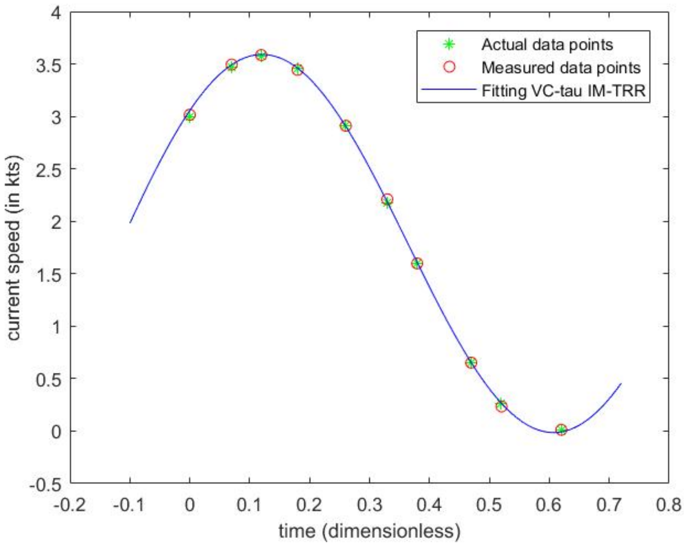

Figure 5.1: Current fitting IM-TRR of simulation 2.2

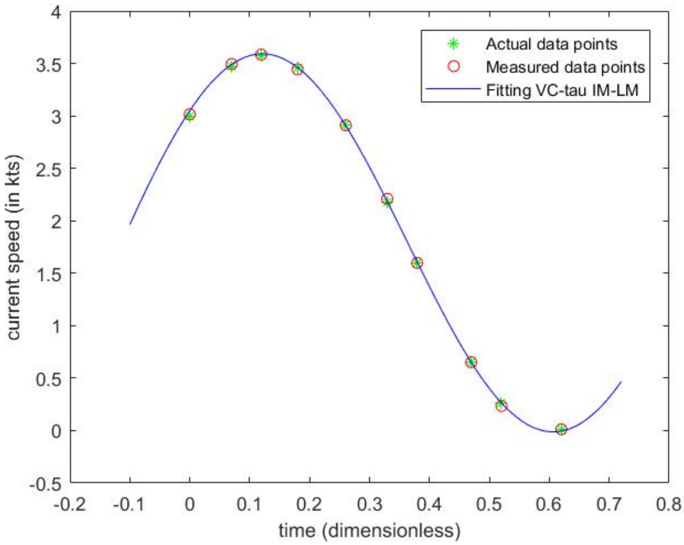

Figure 5.2: Current fitting IM-LM of simulation 2.2

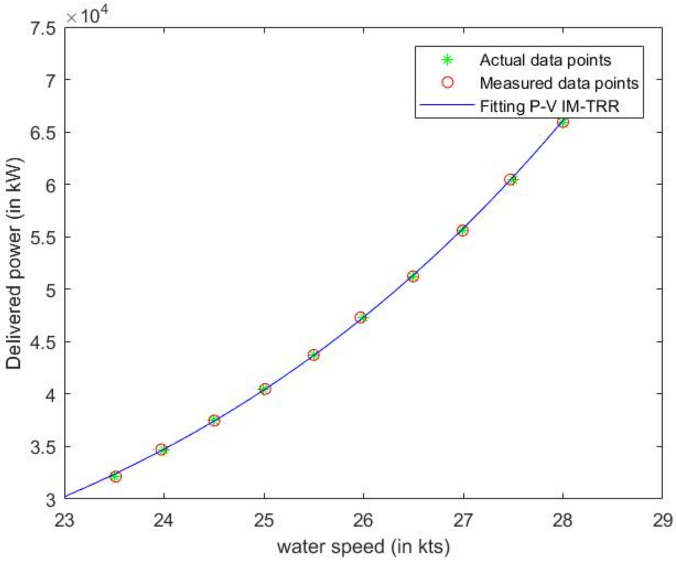

Figure 5.3: P-V fitting IM-TRR of simulation 2.2

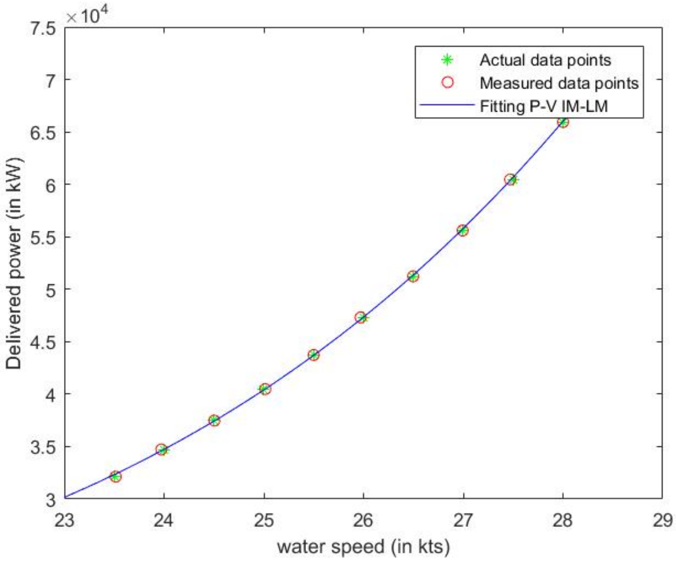

Figure 5.4: P-V fitting IM-LM of simulation 2.2

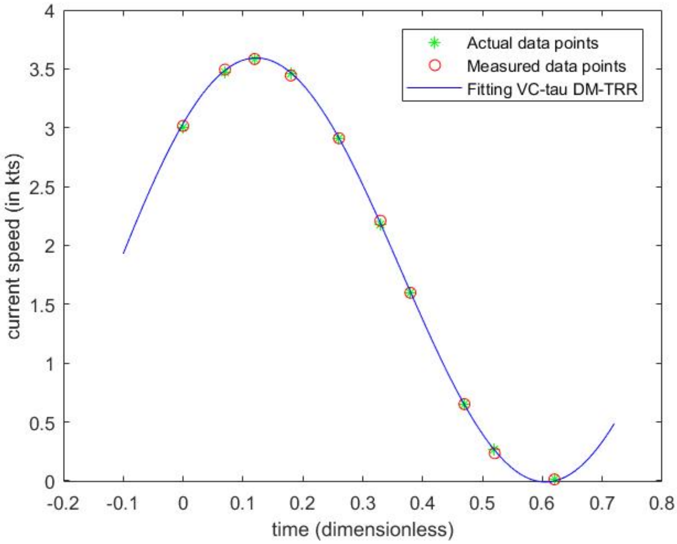

Figure 5.5: Current fitting DM-TRR of simulation 2.2

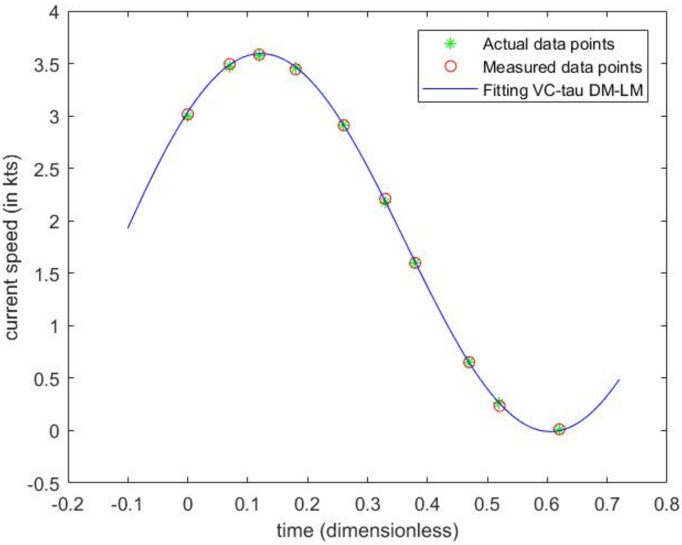

Figure 5.6: Current fitting DM-LM of simulation 2.2

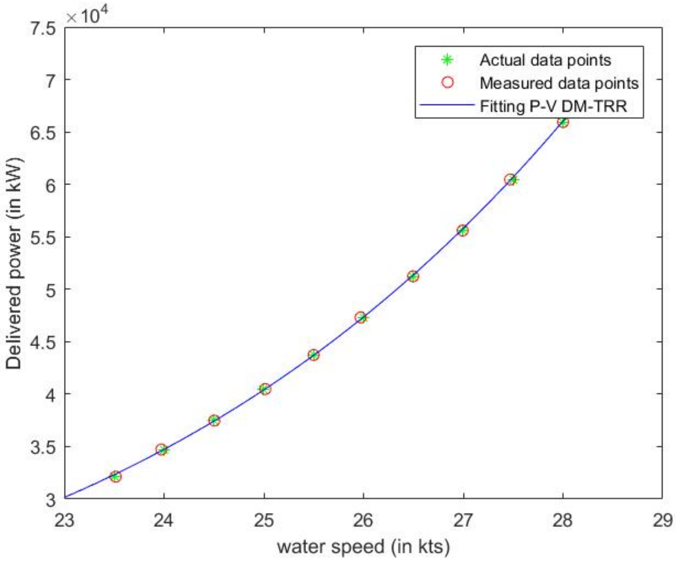

Figure 5.7: P-V fitting DM-TRR of simulation 2.2

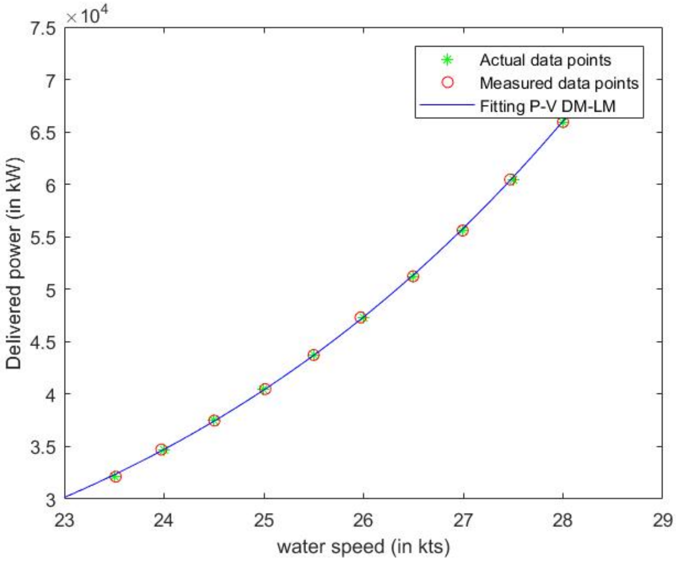

Figure 5.8: P-V fitting DM-LM of simulation 2.2

SIMULATION 3.2

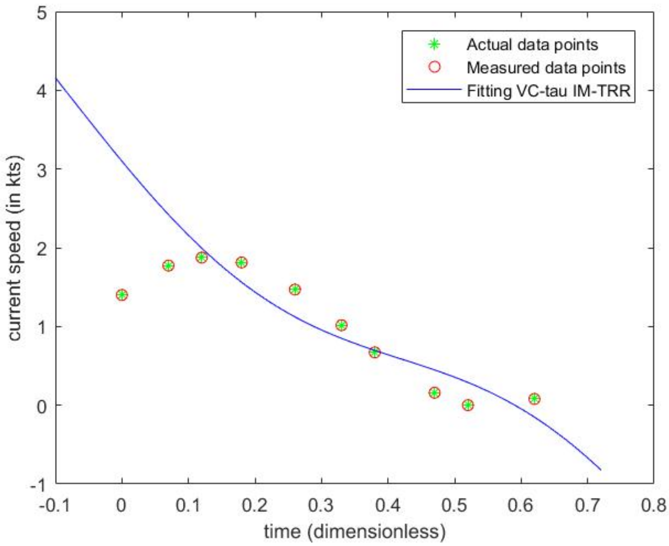

Figure 5.9: Current fitting IM-TRR of simulation 3.2

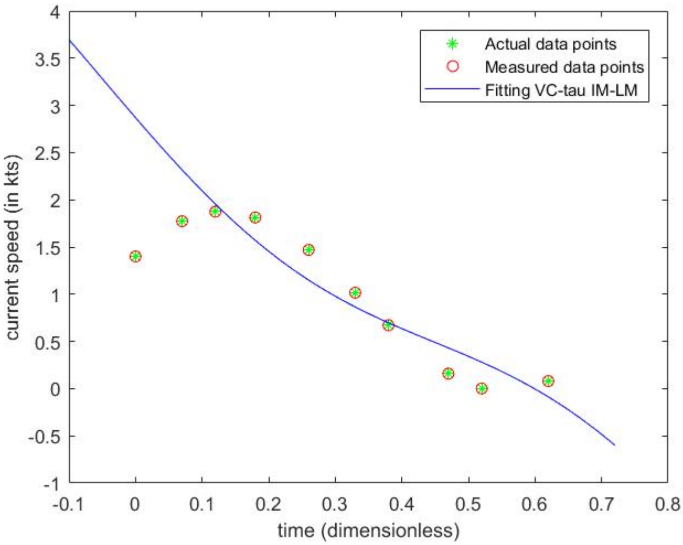

Figure 5.10: Current fitting IM-LM of simulation 3.2

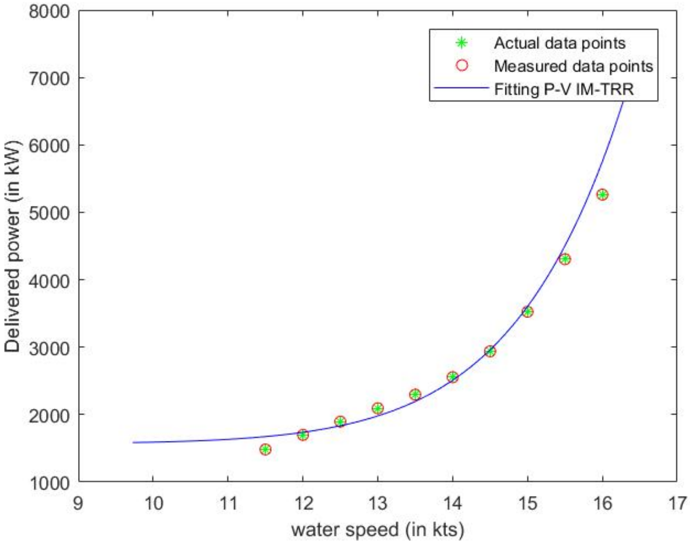

Figure 5.11: P-V fitting IM-TRR of simulation 3.2

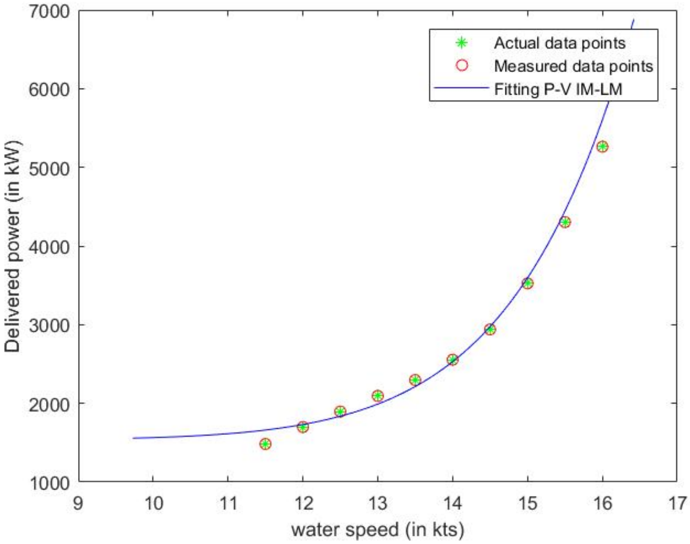

Figure 5.12: P-V fitting IM-LM of simulation 3.2

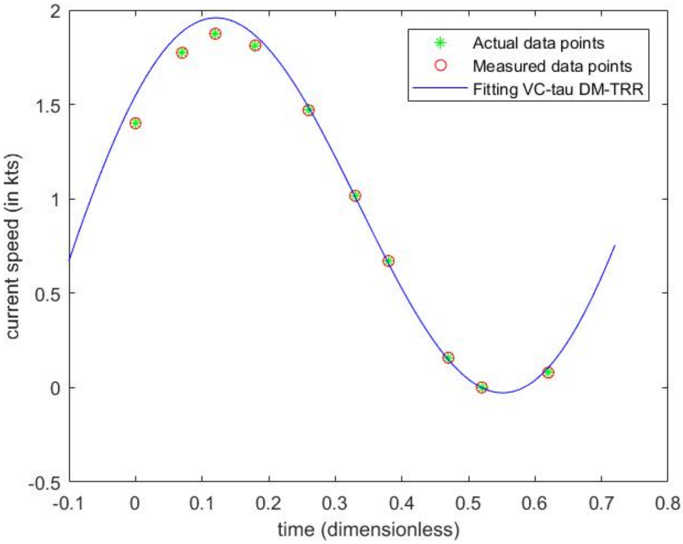

Figure 5.13: Current fitting DM-TRR of simulation 3.2

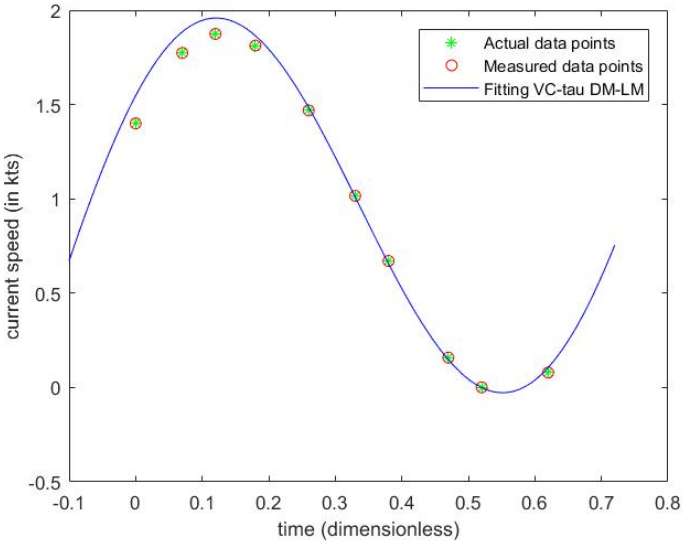

Figure 5.14: Current fitting DM-LM of simulation 3.2

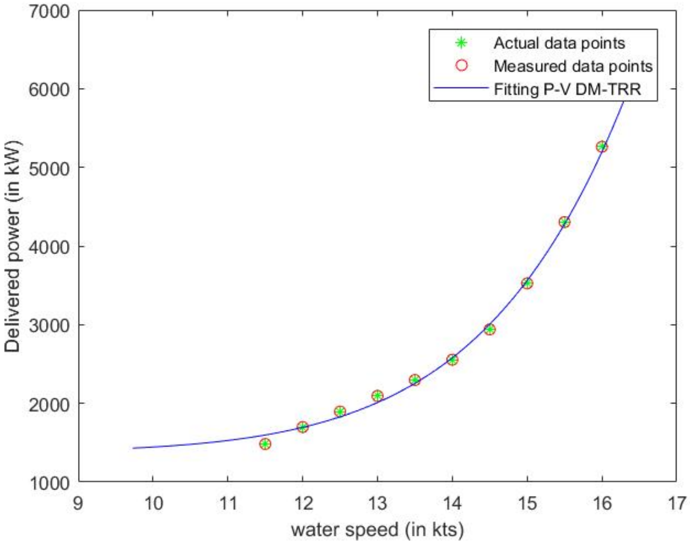

Figure 5.15: P-V fitting DM-TRR of simulation 3.2

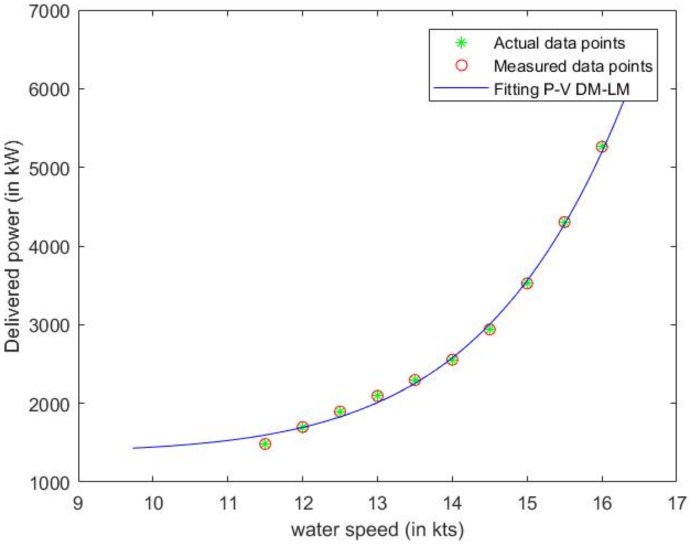

Figure 5.16: P-V fitting DM-LM of simulation 3.2

6
Conclusion

The main purpose of this research is to analyse current correction methods under specific parameter combinations, curve fitting algorithms and exposed to different data sets, that are derived from actual data retained in speed trials. The research question is thus formulated as:

> What is the best possible current correction method for speed trial analysis?

In order to answer this question, it is parsed into sub questions:

- What is the cause of wrong convergence/divergence of the "Iterative" Method?
- What can be said about the reproducibility of both methods?
- Which algorithm can best be used for non-linear fitting?
- Which of the two methods has the best performance?

Unfortunately the answer of the first question was not found. The "Iterative" Method converged for simulations 1.1 up to simulation 2.2, but failed for simulation 3.1 and 3.2. The only difference was that the order of magnitude of the power vector was smaller than those of other simulations. Nonetheless there is not any theoretical or empirical proof that this should result in a misfit.

The reproducibilty of the "Direct" Method and the "Iterative" Method comes down to a few aspects. To start the parameters for non-linear fitting (f.e. Optimality Tolerance, Max Iterations, ...) and the initial estimation of the solution. In 5.4.4 an example is given, where various initial estimations are used as input for the methods. It can be seen here that, as long as the initial estimation is not to far off the true value, both Methods are quite robust. It can be concluded that reproducibility is retained if the initial parameter vector is chosen wisely. The parameters for the non-linear fitting on the other hand, do provide a reproducibility problem. In the simulations it was observed that in some cases the "Iterative" Method would give an incorrect answer, if the parameters were too small. This gives a problem for reproducibility when there are no guidelines, regarding this matter. Further research is needed in order to generalise this. The "Direct" Method always converged in the performed simulations, but it is not unlikely that there will be reproducibility problems for this method as well, regarding the non-linear fitting parameters. The "Iterative" Method has an extra aspect that can cause reproducibilty problems, namely the stopping criterion. In the simulations performed in this research, this value did not have a great effect on the solution. However it is still unknown if this holds in general. Also further research is needed for this.

Regarding the algorithms used for non-linear fitting, both the Trust Region Reflective and the Levenberg-Marquardt algorithm usually provided the same local minimum. Yet in 5.4.4 it was observed that, taking an initial value that is far off from the true value, the Levenberg-Marquardt method failed, while the Trust Region Reflective did not. The Trust Region Reflective seems to be a bit more robust. It is thus recommended to use the Trust Region Reflective, since both methods converge to the same value anyway in good circumstances.

In the simulations performed, based on the chosen data sets, the "Direct" Method was more stable than the "Iterative" Method, since it provided a good fit for simulations 3.1 and 3.2, while the "Iterative" Method failed. From a mathematical point of view, this makes sense. If a local minimum is found for the "Direct" Method, the solution automatically has to be a local minimum for the current correction problem. This is simply how the "Direct" Method is defined. The "Iterative" Method on the other hand, does not have this guarantee. There is no theoretical proof that a local minimum of the "Iterative" Method is equivalent to the solution of the current correction problem. It just turns out to be this way in some cases, where the problem is well defined and holds to certain requirements, that are still unknown to us.

Taken everything together, it can be concluded that the "Direct" Method provides a more (mathematically) trustworthy and robust approach of the current correction. Of course this conclusion is based on a few academic examples and needs more testing to verify this statement.

7

DISCUSSION

As stated in chapter 2, both methods ("Iterative" Method and "Direct" Method) assume:

- The relation between power and water speed of a ship can be approximated as $P(V_S) = a + bV_S^q$
- The current speed can be approximated as $V_C(t) = A\cos(\frac{2\pi}{T_C}t) + B\sin(\frac{2\pi}{T_C}t) + Ct + D$

The accuracy of the methods strongly depends on the correctness of this assumptions. It is important to realize that once these approximations are proven to be the most accurate, current correction methods are also optimized. Fundamental research for this is needed. Especially an important question regarding this matter is in which time frame these approximations are valid.

Even though the research was carried out under generally accepted assumptions and with partly real data from model tests, both methods have not been compared to data of actual speed trials yet. The results of this research are thus valid in a controlled space, where a lot of data is known quite precisely. In order to extend the research, a thorough analysis of both methods has to be made, using data from actual speed trials.

A first step would be to observe both methods under various current functions, since only one current function has been used in this research. Important to mention is that this specific current function resulted in bad fittings for some data, but very precise fittings for other data. This shows that the problem primarily depends on the input data, that is collected in the speed trials or model tests. However, in order to make general conclusions, more current functions have to be examined.

Once both methods pass this test, they can be applied to actual speed trial data, i.e. with a completely unknown current curve. An accessory issue in this matter is that various corrections have to be implemented too, for example the draft correction.
Of course it is also important to test the current corrections method for more different data sets, since six is still a reasonably small number.
Another interesting development is bisection, a numerical method that is now used in STAIMO. Unfortunately I found out too late about this to really dig into the theory, but of what I heard at MARIN it sure sounds promising. This could be an interesting subject for further research, regarding the improvement of the "Iterative" Method.

Last but not least, I propose that further research also experiments with various initial values. The non-linear fitting tends to a local minimum, so it is possible that it sometimes converges to the wrong local minimum, that is not the global minimum. Since the results in this report hint that the "Direct" Method is more robust, this could be used as the initial value for the "Iterative" Method. Unfortunately I did not have the time to take a closer look at this idea.

Appendices

A

Proofs

Theorem 1. *For a approximation of the current speed by a n^{th} order polynomial, the "Mean of means" approximation is given by:*

$$V_{MoM} = \frac{1}{2^{n+1}} \sum_{i=1}^{n+2} \binom{n+1}{i-1} V_i \tag{A.1}$$

Proof. The statement will be proven by induction.
Induction basis ($n = 0$)
For $n = 0$, it holds that:

$$\begin{aligned}\frac{1}{2^{n+1}} \sum_{i=1}^{n+2} \binom{n+1}{i-1} V_i &= \frac{1}{2^1} \sum_{i=1}^{2} \binom{1}{i-1} V_i \\ &= \frac{\binom{1}{0} V_1 + \binom{1}{1} V_2}{2} \\ &= \frac{V_1 + V_2}{2} \\ &= V_{MoM,0}\end{aligned}$$

Induction hypothesis
The following notation will be used:

$$\begin{aligned}V_{MoM,i} &: \text{ the "Mean of means" value for a } i^{\text{th}} \text{ order polynomial approximation} \\ V_{MoM,i \ldots j} &: \text{ the "Mean of means" value of measurements } V_i \text{ up to and including } V_j\end{aligned}$$

Assume that there exists an $k \in \mathbb{N}$ such that:

$$V_{MoM,k} = \frac{1}{2^{k+1}} \sum_{i=1}^{k+2} \binom{k+1}{i-1} V_i$$

Then:

$$\begin{aligned}\frac{1}{2^{k+2}} \sum_{i=1}^{k+3} \binom{k+2}{i-1} V_i &= \frac{1}{2} \frac{V_1 + V_{k+3} + \sum_{i=2}^{k+2} \binom{k+2}{i-1} V_i}{2^{k+1}} \\ &= \frac{1}{2} \frac{V_1 + V_{k+3} + \sum_{i=2}^{k+2} \binom{k+1}{i-1} V_i + \sum_{i=2}^{k+2} \binom{k+1}{i-2} V_i}{2^{k+1}}\end{aligned}$$

by Pascal's rule. This can be rewritten as:

$$\begin{aligned}
\frac{1}{2}\frac{V_1 + V_{k+3} + \sum_{i=2}^{k+2}\binom{k+1}{i-1}V_i + \sum_{i=2}^{k+2}\binom{k+1}{i-2}V_i}{2^{k+1}} &= \frac{1}{2}\frac{\sum_{i=1}^{k+2}\binom{k+1}{i-1}V_i + \sum_{i=2}^{k+3}\binom{k+1}{i-2}V_i}{2^{k+1}} \\
&= \frac{1}{2}\frac{\sum_{i=1}^{k+2}\binom{k+1}{i-1}V_i + \sum_{i=1}^{k+2}\binom{k+1}{i-1}V_{i+1}}{2^{k+1}} \\
&= \frac{V_{MoM,1\dots k} + V_{MoM,2\dots k+1}}{2} \\
&= V_{MoM,k+1}
\end{aligned}$$

By induction, it is proven that:

$$V_{MoM} = \frac{1}{2^{n+1}}\sum_{i=1}^{n+2}\binom{n+1}{i-1}V_i$$

□

Theorem 2. $\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}i^j = 0 \quad \forall n \in \mathbb{N}, j = 0,\dots,n$

Proof. The statement will be proven by induction.
Induction basis ($n = 0$)

For $n = 0$, it holds that:

$$\begin{aligned}
\sum_{i=1}^{n+2}\binom{n+1}{i-1}(-1)^{i+1}i^j &= \sum_{i=1}^{2}\binom{1}{i-1}(-1)^{i+1}i^j \qquad , j = 0 \\
&= \sum_{i=1}^{2}\binom{1}{i-1}(-1)^{i+1} \\
&= \binom{1}{0}(-1)^2 + \binom{1}{1}(-1)^3 \\
&= 0
\end{aligned}$$

Induction hypothesis

Assume that there exists an $k \in \mathbb{N}$ such that:

$$\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^j = 0 \qquad , j = 0,\dots,k$$

Then:

$$\begin{aligned}
\sum_{i=1}^{k+3}\binom{k+2}{i-1}(-1)^{i+1}i^j &= 1+(-1)^{k+4}(k+3)^j+\sum_{i=2}^{k+2}\binom{k+2}{i-1}(-1)^{i+1}i^j && ,j=0,\dots,k\\
&= 1+(-1)^{k+4}(k+3)^j\\
&\quad+\sum_{i=2}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^j+\sum_{i=2}^{k+2}\binom{k+1}{i-2}(-1)^{i+1}i^j && ,j=0,\dots,k\\
&= \sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^j+\sum_{i=2}^{k+3}\binom{k+1}{i-2}(-1)^{i+1}i^j && ,j=0,\dots,k
\end{aligned}$$

where the second equality follows from Pascal's rule.
By the induction hypothesis, the first term vanishes, such that:

$$\begin{aligned}
\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^j+\sum_{i=2}^{k+3}\binom{k+1}{i-2}(-1)^{i+1}i^j &= \sum_{i=2}^{k+3}\binom{k+1}{i-2}(-1)^{i+1}i^j && ,j=0,\dots,k\\
&= \sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+2}(i+1)^j && ,j=0,\dots,k\\
&= -\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}(i+1)^j && ,j=0,\dots,k
\end{aligned}$$

Now $(i+1)^j$ can be written by the binomial identity, which results in:

$$\begin{aligned}
-\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}(i+1)^j &= -\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}\sum_{m=0}^{j}\binom{j}{m}i^{j-m} && ,j=0,\dots,k\\
&= -\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}\sum_{m=1}^{j}\binom{j}{m}i^{j-m}-\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^j && ,j=0,\dots,k\\
&= -\sum_{m=1}^{j}\binom{j}{m}\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^{j-m}-\sum_{i=1}^{k+2}\binom{k+1}{i-1}(-1)^{i+1}i^j && ,j=0,\dots,k\\
&= 0
\end{aligned}$$

The last step follows by the hypothesis induction (two times). Note that $j-m\in\{0,\dots,k\}$.

The statement also has to hold for the case that $j=k+1$. Fortunately it does, since:

$$\begin{aligned}
\sum_{i=1}^{k+3}\binom{k+2}{i-1}(-1)^{i+1}i^{k+1} &= \sum_{i=1}^{m+2}\binom{m+1}{i-1}(-1)^{i+1}i^m\\
&= 0
\end{aligned}$$

by the induction hypothesis with $j = m$. It can be concluded that:

$$\sum_{i=1}^{k+3} \binom{k+2}{i-1} (-1)^{i+1} i^j = 0 \quad \forall n \in \mathbb{N}, j = 0, \ldots, k+1$$

Thus theorem 2 is true by induction. □

B

Matlab implementation

```
%% Input data
clear all
close all

% % time of tidal current (in s)
% TC = 12*3600+25*60+12;
% % virtual exact dimensionless times
% Texact = [0, 0.07, 0.12, 0.18, 0.26, 0.33, 0.38, 0.47, 0.52, 0.62];
% % virtual exact waterspeeds
% VSexact = [24, 24, 25, 25, 26, 26, 27, 27, 28, 28];
% % virtual exact ground speeds
% VGexact = [27, 20.5332,28.5794, 21.5411, 28.9112,
% 23.8194, 28.6005, 26.3538, 28.2649, 27.9874];
% % virtual exact current speed
% VCexact = abs(VGexact - VSexact);
% % noise levels
% n1 = 0.05/3; %sigma of noise ground speed
% n2 = 36/(3*TC); %sigma of noise time
% n3 = 25/3; % sigma of noise power
% % virtual exact corrected powers
% Pcor = [34713, 34713, 40478, 40478, 47305, 47305, 55599, 55599, 65932, 65932];

% time of tidal current (in s)
TC = 12*3600+25*60+12;
% virtual exact dimensionless times
Texact = [0, 0.07, 0.12, 0.18, 0.26, 0.33, 0.38, 0.47, 0.52, 0.62];
% virtual exact waterspeeds
VSexact = [23.5 24, 24.5, 25, 25.5, 26, 26.5, 27, 27.5, 28];
% virtual exact ground speeds
VGexact = [26.5, 20.5332, 28.0794, 21.5411, 28.4112, 23.8194, 28.1005,
26.3538, 27.7649, 27.9874];
% virtual exact current speed
VCexact = abs(VGexact - VSexact);
% noise levels
n1 = 0.05/3; %sigma of noise ground speed
n2 = 36/(3*TC); %sigma of noise time
n3 = 25/3; % sigma of noise power
% virtual exact corrected powers
Pcor = [32141, 34713, 37483, 40478, 43737, 47305, 51238, 55599, 60466, 65932];

% virtual exact parameter vector
% beta = [?, ?, ?, 1.5, 1.3, 0.8, 1.5] ;
% initial guess of parameters P-V relation
x0step1 = [1000, 2, 3];

options1 = optimoptions('lsqnonlin', 'Algorithm', 'trust-region-reflective',
'FunctionTolerance', 1e-6, 'MaxFunctionEvaluations', 100000, 'MaxIterations',
100000, 'OptimalityTolerance', 1e-6, 'StepTolerance', 1e-6);
options2 = optimoptions('lsqnonlin', 'Algorithm', 'levenberg-marquardt',
'FunctionTolerance', 1e-6, 'MaxFunctionEvaluations', 100000, 'MaxIterations',
100000, 'OptimalityTolerance', 1e-6, 'StepTolerance', 1e-6);

% bounds
% lbim = [0, 0, 2.5];
% ubim = [1000000, 1000000, 4];
% lbdm = [0, 0, 2.5, -1000000, -1000000, -1000000, -1000000];
% ubdm = [1000000, 1000000, 4, 1000000, 1000000, 1000000, 1000000];
```

```

%% Simulation matrices
k = 1;
VCerror_imtrr = [];
VCerror_imlm = [];
VCerror_dirtrr = [];
VCerror_dirlm = [];

Est_imtrr = [];
Est_imlm = [];
Est_dirtrr = [];
Est_dirlm = [];

while k < 2
%% Noise
% % virtual measured ground speeds (with noise)
% VGnoise = [normrnd(27, n1), normrnd(20.5332, n1),normrnd(28.5794, n1),
% normrnd(21.5411, n1), normrnd(28.9112, n1), normrnd(23.8194, n1),
% normrnd(28.6005, n1), normrnd(26.3538, n1), normrnd(28.2649, n1),
% normrnd(27.9874, n1)];
% VGdifference = VGexact - VGnoise;
% % virtual measured dimensionless time (with noise)
% Tnoise = [normrnd(0, n2), normrnd(0.07, n2), normrnd(0.12, n2),
% normrnd(0.18, n2), normrnd(0.26, n2), normrnd(0.33, n2),
% normrnd(0.38, n2), normrnd(0.47, n2), normrnd(0.52, n2), normrnd(0.62, n2)];
% Tdifference = Texact - Tnoise;
% % virtual measured powers (with noise)
% Pcornoise = [normrnd(34713, n3), normrnd(34713, n3), normrnd(40478, n3),
% normrnd(40478, n3), normrnd(47305, n3), normrnd(47305, n3),
% normrnd(55599, n3), normrnd(55599, n3), normrnd(65932, n3), normrnd(65932, n3)];
% Pdifference = Pcor - Pcornoise;

% virtual measured ground speeds (with noise)
VGnoise = [normrnd(26.5, n1), normrnd(20.5332, n1), normrnd(28.0794, n1),
normrnd(21.5411, n1), normrnd(28.4112, n1), normrnd(23.8194, n1),
normrnd(28.1005, n1), normrnd(26.3538, n1), normrnd(27.7649, n1), normrnd(27.9874, n1)];
VGdifference = VGexact - VGnoise;
% virtual measured dimensionless time (with noise)
Tnoise = [normrnd(0, n2), normrnd(0.07, n2), normrnd(0.12, n2), normrnd(0.18, n2),
normrnd(0.26, n2), normrnd(0.33, n2), normrnd(0.38, n2), normrnd(0.47, n2),
normrnd(0.52, n2), normrnd(0.62, n2)];
Tdifference = Texact - Tnoise;
% virtual measured powers (with noise)
Pcornoise = [normrnd(32141, n3), normrnd(34713, n3), normrnd(37483, n3),
normrnd(40478, n3), normrnd(43737, n3), normrnd(47305, n3), normrnd(51238, n3),
normrnd(55599, n3), normrnd(60466, n3), normrnd(65932, n3)];
Pdifference = Pcor - Pcornoise;

%% Iterative Method
% Creation of helpful vectors

% number of runs
n_runs = length(Texact);
% vectors for plotting the functions
vs = linspace(min(VSexact) - 0.5,max(VSexact) + 0.5,100);
tau = linspace(min(Tnoise) - 0.1, max(Tnoise) + 0.1,100);

```

```
% vectors for initial fitting iterative method
for i=1:1:n_runs*0.5
    Pstep1(i) = (Pcornoise(2*i-1) + Pcornoise(2*i))/2;
    VGstep1(i) = (VGnoise(2*i-1) + VGnoise(2*i))/2;
end

% function initial fitting iterative method
fun1 = @(B)B(1)+B(2)*VGstep1.^B(3) - Pstep1;

% initial fitting with Trust Region Reflective
[x,resnorm,residual,exitflag,output] = lsqnonlin(fun1,x0step1,[],[],options1);
% initial fitting with Levenberg Marquardt
[y,resnorm,residual,exitflag,output] = lsqnonlin(fun1,x0step1,[],[],options2);

% creation of matrices that contain parameter estimations and standard errors for
% every iteration

% parameter matrices, for Trust Region Reflective and Levenberg-Marquardt
Ptrr = [];
Plm = [];
Ptrrlin = [];
Plmlin = [];
% error matrices, for Trust Region Reflective and Levenberg-Marquardt
% initial error
Etrr = [sqrt(sum(Pcornoise.^2)/n_runs)];
Elm = [sqrt(sum(Pcornoise.^2)/n_runs)];

% iteration variable
j1=1;
j2=1;

%% first iteration

% addition of new parameter vector estimations
Ptrr = [Ptrr; x];
Plm = [Plm; y];

% calculation of water speeds for first iteration
VS1 = VS(Ptrr(j1,:), Pcornoise);
VS2 = VS(Plm(j1,:), Pcornoise);

% calculation of current speeds for first iteration
VC1 = abs(VGnoise - VS1);
VC2 = abs(VGnoise - VS2);

% current function
fun2 = @(D)D(1).*cos(2*pi*Tnoise) + D(2).*sin(2*pi*Tnoise) + D(3).*Tnoise + D(4);

% linear least squares fitting current function to data
C = [cos(2*pi*Tnoise).' sin(2*pi*Tnoise).' Tnoise.' ones(n_runs, 1)];
xlin = lsqlin(C, VC1);
ylin = lsqlin(C, VC2);

%update linear part
Ptrrlin = [Ptrrlin; xlin.'];
Plmlin = [Plmlin; ylin.'];

```

```
% calculation of new current speeds with found parameter vector
VC_1 = fun2(xlin);
VC_2 = fun2(ylin);

% calculation of new water speeds
for i=1:1:n_runs
    VS_1(i) = VGnoise(i) + (-1)^(i)*VC_1(i);
    VS_2(i) = VGnoise(i) + (-1)^(i)*VC_2(i);
end

% new P-V relation that needs fitting
fun3 = @(B)B(1)+B(2)*VS_1.^B(3) - Pcornoise;
fun4 = @(B)B(1)+B(2)*VS_2.^B(3) - Pcornoise;

% non-linear least squares fitting
[x,resnorm,residual,exitflag,output] = lsqnonlin(fun3,Ptrr(j1,:),[],[],options1);
[y,resnorm,residual,exitflag,output] = lsqnonlin(fun4,Plm(j1,:),[],[],options2);

% addition of new standard errors
Etrr = [Etrr;sqrt(sum(fun3(x).^2)/n_runs)];
Elm = [Etrr;sqrt(sum(fun4(y).^2)/n_runs)];
% update parameter matrices
Ptrr = [Ptrr; x];
Plm = [Plm; y];

% update iteration variable
j1 = j1+1;
j2 = j2+1;

%% Iteration process TRR
% stopping criterion: if consequent steps of both methods are close enough,
% the iteration process stops
while abs(Etrr(j1-1,:) - Etrr(j1,:)) > 10^(-4)

% update water speeds
VS1 = VS(Ptrr(j1,:), Pcornoise);

% update current speeds
VC1 = abs(VGnoise - VS1);

% linear least squares fitting
xlin = (C.'*C)\(C.'*VC1.');
Ptrrlin = [Ptrrlin; xlin.'];

% update current speeds
VC_1 = fun2(xlin);

% update water speeds
for i=1:1:n_runs
    VS_1(i) = VGnoise(i) + (-1)^(i)*VC_1(i);
end

% non-linear least squares fitting
[x,resnorm,residual,exitflag,output] = lsqnonlin(fun3,Ptrr(j1,:),[],[],options1);

% update parameter matrices
Ptrr = [Ptrr; x];
```

```
% update error matrix
Etrr = [Etrr;sqrt(sum(fun3(x).^2)/n_runs)];

% update iteration variable
j1 = j1+1;
end

% select best minimizer of last two iterations
if Etrr(j1-1,:) < Etrr(j1,:)
    x = Ptrr(j1-1,:);
    xlin = Ptrrlin(j1-1,:);
end

%% Iteration process LM
% stopping criterion: if consequent steps of both methods are close enough,
% the iteration process stops
while abs(Elm(j2-1,:) - Elm(j2,:)) > 10^(-4)

% update parameter matrices
Plm = [Plm; y];

% update water speeds
VS2 = VS(Plm(j2,:), Pcornoise);

% update current speeds
VC2 = abs(VGnoise - VS2);

% linear least squares fitting
ylin = (C.'*C)\(C.'*VC2.');
Plmlin = [Plmlin; ylin.'];

% update current speeds
VC_2 = fun2(ylin);

% update water speeds
for i=1:1:n_runs
    VS_2(i) = VGnoise(i) + (-1)^(i)*VC_2(i);
end

% non-linear least squares fitting
[y,resnorm,residual,exitflag,output] = lsqnonlin(fun4,Plm(j2,:),[],[],options2);

% update error matrices
Elm = [Etrr;sqrt(sum(fun4(y).^2)/n_runs)];

% update iteration variable
j2 = j2+1;
end

% select best minimizer of last two iterations
if Elm(j2-1,:) < Elm(j2,:)
    y = Plm(j2-1,:);
    ylin = Plmlin(j2-1,:);
end

%% Analysis of results

```

```
% current function
fun5 = @(B)B(1)*cos(2*pi*tau) + B(2)*sin(2*pi*tau) + B(3)*tau + B(4);

% current difference function
fun6 = @(B)B(1)*cos(2*pi*Texact) + B(2)*sin(2*pi*Texact) + B(3)*Texact + B(4) - VCexact;

% calculation of real standard errors
fun7 = @(B)B(1)+B(2)*VSexact.^B(3) - Pcor;
Eim_trr = sqrt(sum(fun7(x).^2)/n_runs);
Eim_lm = sqrt(sum(fun7(y).^2)/n_runs);
EVCim_trr = sqrt(sum(fun6(xlin).^2)/n_runs);
EVCim_lm = sqrt(sum(fun6(ylin).^2)/n_runs);

% P-V function
fun8 = @(B) B(1) + B(2)*vs.^B(3);

% final P-V formula plots (plotted against data points without noise)
for i=1:1:n_runs
    VSnoise(i) = VGnoise(i) + (-1)^(i)*VCexact(i);
end

% update exact matrices
VCerror_imtrr = [VCerror_imtrr; EVCim_trr];
VCerror_imlm = [VCerror_imlm; EVCim_lm];
Est_imtrr = [Est_imtrr; Eim_trr];
Est_imlm = [Est_imlm; Eim_lm];


%% Direct Method

% initial guess parameter vector
x0 = [500,2,3,1,1,1,1];

% direction vector of current
for i=1:n_runs
    c(i) = (-1).^(i);
end

% total P-VG, tau relation with and without noise
fun9 = @(B)B(1)+B(2)*(VGnoise + c.*(B(4)*cos(2*pi*Tnoise) + B(5)*sin(2*pi*Texact)
+ B(6)*Texact + B(7))).^B(3) - Pcornoise;
fun10 = @(B)B(1)+B(2)*(VGexact + c.*(B(4)*cos(2*pi*Texact) + B(5)*sin(2*pi*Texact)
+ B(6)*Texact + B(7))).^B(3) - Pcor;

% non-linear fitting with Trust Region Reflective and Levenberg-Marquardt
[xdir,resnorm,residual,exitflag,output] = lsqnonlin(fun9,x0,[],[],options1);
[ydir,resnorm,residual,exitflag,output] = lsqnonlin(fun9,x0,[],[],options2);

% needed entries of parameter vector
xdirpv = xdir(1:3);
ydirpv = ydir(1:3);
xdircf = xdir(4:7);
ydircf = ydir(4:7);

% calculation of real errors
Edm_trr = sqrt(sum(fun10(xdir).^2)/n_runs);
Edm_lm = sqrt(sum(fun10(ydir).^2)/n_runs);
```

```
EVCdm_trr = sqrt(sum(fun6(xdircf).^2)/n_runs);
EVCdm_lm = sqrt(sum(fun6(ydircf).^2)/n_runs);

VCerror_dirtrr = [VCerror_dirtrr; EVCdm_trr];
VCerror_dirlm = [VCerror_dirlm; EVCdm_lm];
Est_dirtrr = [Est_dirtrr; Edm_trr];
Est_dirlm = [Est_dirlm; Edm_lm];

k = k + 1
end

% plots (optional)
VCnoise = abs(VGnoise - VSexact);
figure(1)
plot(Texact,VCexact, 'g*', Tnoise, VCnoise, 'ro',tau,fun5(xlin),'b-')
xlabel('time (dimensionless)')
ylabel('current speed (in kts)')
legend('Actual data points', 'Measured data points','Fitting VC-tau IM-TRR')
figure(2)
plot(Texact,VCexact, 'g*', Tnoise, VCnoise, 'ro',tau,fun5(ylin),'b-')
xlabel('time (dimensionless)')
ylabel('current speed (in kts)')
legend('Actual data points', 'Measured data points','Fitting VC-tau IM-LM')
figure(3)
plot(VSexact,Pcor, 'g*', VSnoise, Pcornoise, 'ro',vs,fun8(x),'b-')
xlabel('water speed (in kts)')
ylabel('Delivered power (in kW)')
legend('Actual data points', 'Measured data points', 'Fitting P-V IM-TRR')
figure(4)
plot(VSexact,Pcor, 'g*', VSnoise, Pcornoise, 'ro',vs,fun8(y),'b-')
xlabel('water speed (in kts)')
ylabel('Delivered power (in kW)')
legend('Actual data points', 'Measured data points', 'Fitting P-V IM-LM')
figure(5)
plot(Texact,VCexact, 'g*', Tnoise, VCnoise, 'ro',tau,fun5(xdircf),'b-')
xlabel('time (dimensionless)')
ylabel('current speed (in kts)')
legend('Actual data points', 'Measured data points','Fitting VC-tau DM-TRR')
figure(6)
plot(Texact,VCexact, 'g*', Tnoise, VCnoise, 'ro',tau,fun5(ydircf),'b-')
xlabel('time (dimensionless)')
ylabel('current speed (in kts)')
legend('Actual data points', 'Measured data points','Fitting VC-tau DM-LM')
figure(7)
plot(VSexact,Pcor, 'g*', VSnoise, Pcornoise, 'ro',vs,fun8(xdirpv),'b-')
xlabel('water speed (in kts)')
ylabel('Delivered power (in kW)')
legend('Actual data points', 'Measured data points', 'Fitting P-V DM-TRR')
figure(8)
plot(VSexact,Pcor, 'g*', VSnoise, Pcornoise, 'ro',vs,fun8(ydirpv),'b-')
xlabel('water speed (in kts)')
ylabel('Delivered power (in kW)')
legend('Actual data points', 'Measured data points', 'Fitting P-V DM-LM')

% calculation of means
E1 = mean2(Est_imtrr);
E2 = mean2(Est_imlm);
```

```
E3 = mean2(Est_dirtrr);
E4 = mean2(Est_dirlm);
E5 = mean2(VCerror_imtrr);
E6 = mean2(VCerror_imlm);
E7 = mean2(VCerror_dirtrr);
E8 = mean2(VCerror_dirlm);

%% Used functions
function  V = VS(x, P)
    V = ((P-x(1))/x(2)).^(1/x(3));
end
```

C

SIMULATION DATA

C.1. DATA AND RESULTS

Six tuples of $(\boldsymbol{P}, \boldsymbol{V_S})$ vectors were used for simulations. These combinations are taken from existing model test data. Both $\boldsymbol{P}$ and V_S vectors are 10-dimensional, i.e. cover a speed trial of 10 Runs. In order to observe the effect between a speed trial with 8 - and 10 Runs, simulations are also made where two Runs are taken away.

For all simulations the same current function and non-dimensional time vector was used:

$$\boldsymbol{\beta}_1 = \begin{pmatrix} 1.5 \\ 1.3 \\ 0.8 \\ 1.5 \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} 0 \\ 0.07 \\ 0.12 \\ 0.18 \\ 0.26 \\ 0.33 \\ 0.38 \\ 0.47 \\ 0.52 \\ 0.62 \end{pmatrix}$$

C.2. SIMULATION 1

Based on the model tests of training case [21].

	P_i (kW)	$V_{S,i}$ (kts)
1	5225	14
2	5225	14
3	6728	15
4	6728	15
5	8716	16
6	8716	16
7	11503	17
8	11503	17
9	11503	17
10	11503	17

Table C.1: Data tuples simulation 1.1

	P_i (kW)	$V_{S,i}$ (kts)
1	2313	11
2	3035	12
3	4018	13
4	5225	14
5	6728	15
6	8716	16
7	9987	16.5
8	9987	16.5
9	11503	17
10	11503	17

Table C.2: Data tuples simulation 1.2

C.3. SIMULATION 2

Based on the model tests of training case [22].

	P_i (kW)	$V_{S,i}$ (kts)
1	34713	24
2	34713	24
3	40478	25
4	40478	25
5	47305	26
6	47305	26
7	55599	27
8	55599	27
9	65932	28
10	65932	28

Table C.3: Data tuples simulation 2.1

	P_i (kW)	$V_{S,i}$ (kts)
1	32141	23.5
2	34713	24
3	37483	24.5
4	40478	25
5	43737	25.5
6	47305	26
7	51238	26.5
8	55599	27
9	60466	27.5
10	65932	28

Table C.4: Data tuples simulation 2.2

C.4. SIMULATION 3

Based on the model tests of training case [23].

	P_i (kW)	$V_{S,i}$ (kts)
1	1699	12
2	1699	12
3	2094	13
4	2094	13
5	2554	14
6	2554	14
7	3525	15
8	3525	15
9	5262	16
10	5262	16

Table C.5: Data tuples simulation 3.1

	P_i (kW)	$V_{S,i}$ (kts)
1	1483	11.5
2	1699	12
3	1896	12.5
4	2094	13
5	2296	13.5
6	2554	14
7	2939	14.5
8	3525	15
9	4304	15.5
10	5262	16

Table C.6: Data tuples simulation 3.2

D

Error distribution

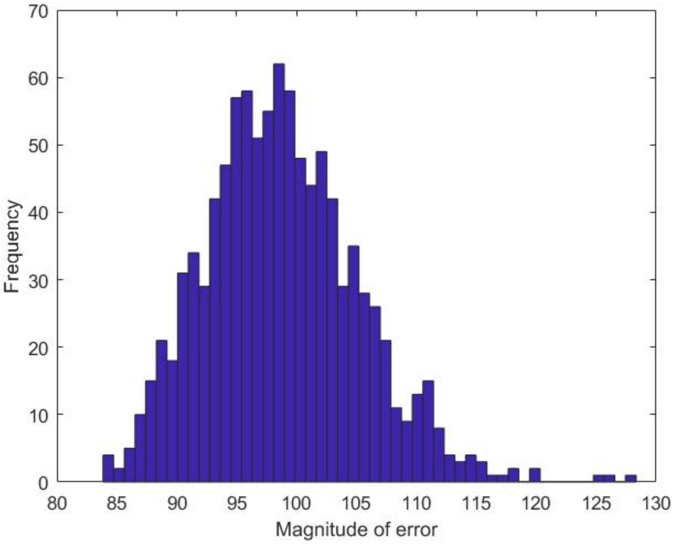

Figure D.1: Error distribution E1

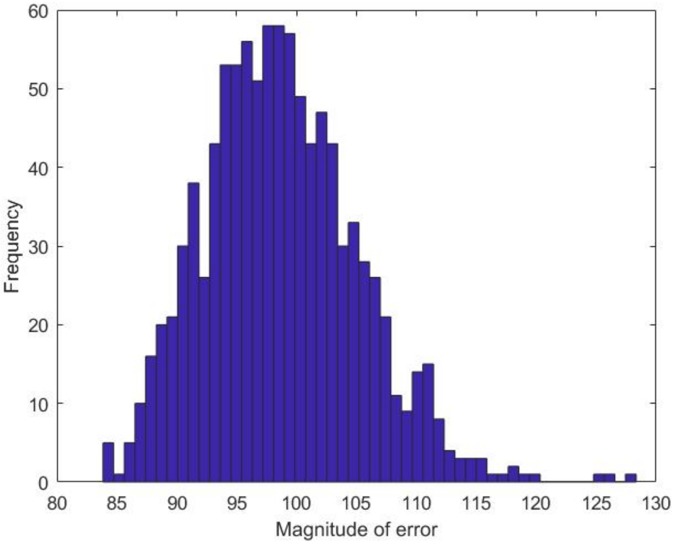

Figure D.2: Error distribution E2

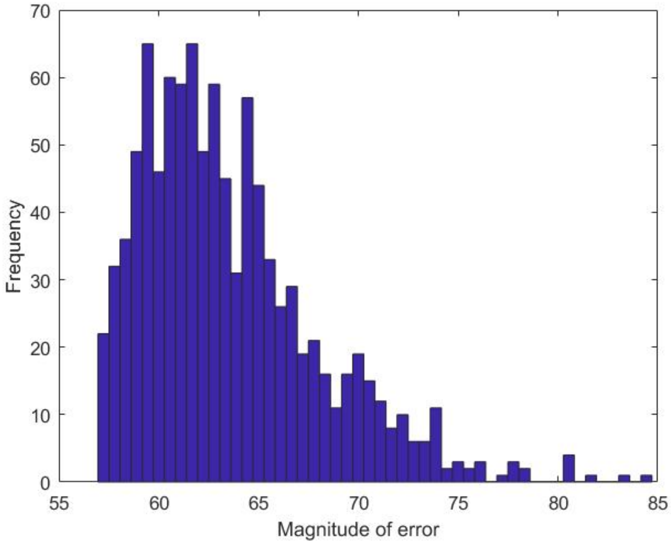

Figure D.3: Error distribution E3

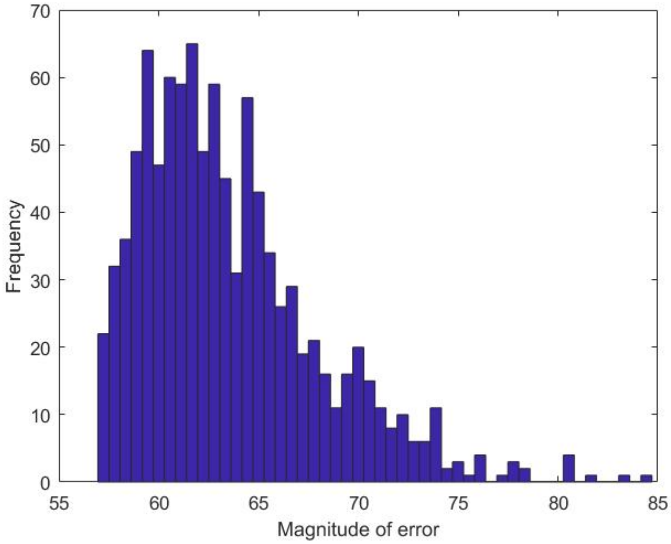

Figure D.4: Error distribution E4

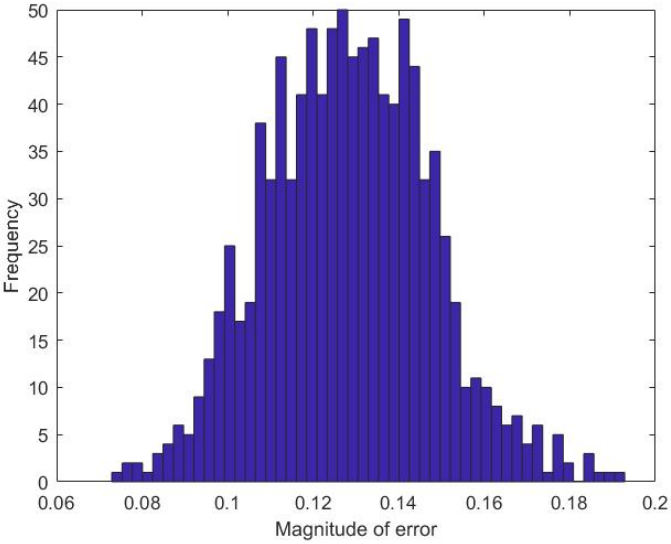

Figure D.5: Error distribution E5

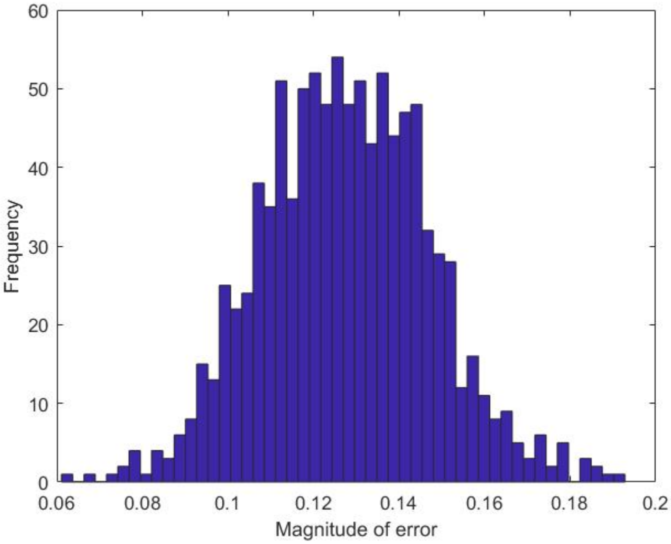

Figure D.6: Error distribution E6

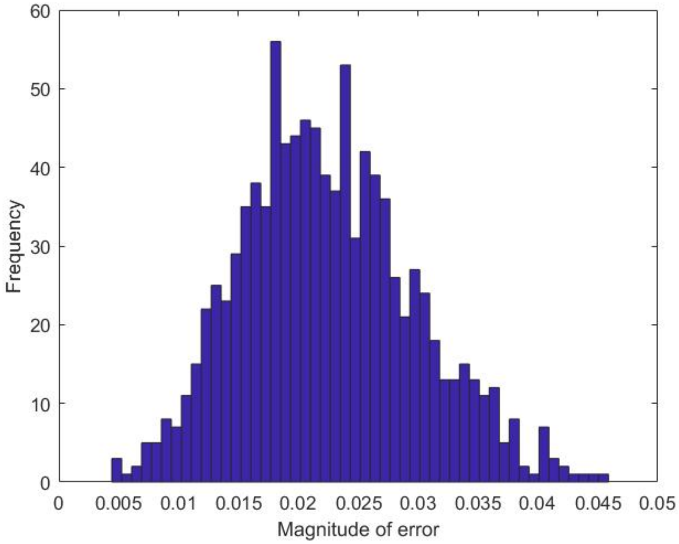

Figure D.7: Error distribution E7

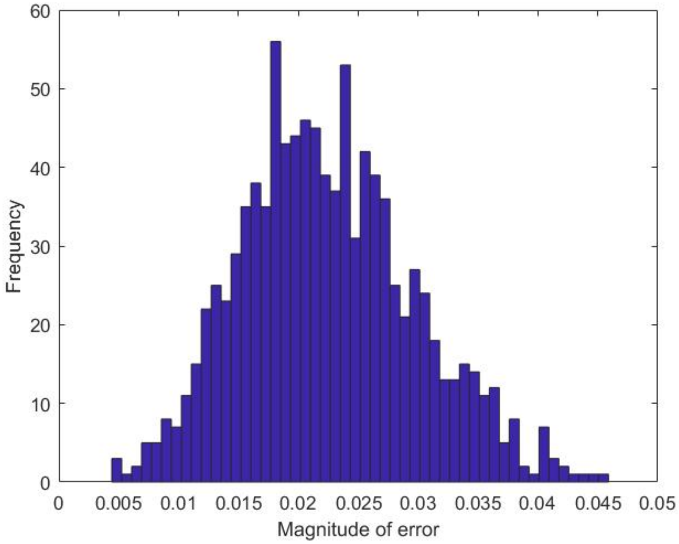

Figure D.8: Error distribution E8

E

Visit MARIN

On the 30th of June 2017, I paid a visit to the MARIN, the Maritime Research Institute Netherlands, located in Wageningen. This internationally renown institute has various experimental facilities like model basins with wave- and current generators, various simulators and in the near future an unique two phase laboratry becomes availables tank to analyse the behaviour of gas-liquid mixtures. Furthermore they also provide software tools for hydrodynamic design and analysis to the maritime industry. One of these tools is the shareware STAIMO [24]. STAIMO is a software tool which analyses the data from Speed/Power Trials through the "Iterative" Method or the "Mean of means" Method. Needless to say, MARIN also benefits from a more stable method, since it encounters the same problems that are stumbled upon in the industry. Martin van Hees and Henk van den Boom are gradually improving STAIMO. One of the issues that needs to be dealth with is related to the "Iterative" Method. During my visit to MARIN, I had a meeting with both of them, to exchange ideas and ask questions.

In this meeting I found out that in STAIMO, both the "Iterative" Method and the "Mean of means" Method are solely used to approximate the current function. The parameters of the P-V relation are irrelevant, once the parameters of the current function are obtained. With this current function, the (P, V_S) points can be calculated from the (P, V_G) points. For new ships, MARIN or other model basis should perform model tests if the sea trial loading condition does not correspond with the contract loading condition. Since these tests are performed in nearly perfect circumstances (no wind, no waves, known current), the P-V curve of the actual ship (in ideal weather conditions) would have to look like the curve found in the model test. The shape of the curve found in the model test will therefore be fitted to the calculated (P, V_S) found through the analysis. This final curve will be used to determine whether the ship satisfies the contract speeds or not. Of course the (P, V_S) points first have to be corrected, since the trials are basically never performed in ideal weather conditions. In this report, ideal weather conditions are assumed to merely analyse the difference between various methods. However it is important to realize that corrections always have to be made.

The "Mean of means" Method has a similar approach. For each Double Run the mean of the measured powers and the measured ground speeds are taken. These tupels are used as new (P, V_S) points. Again, the curve found in the model test will be fitted to these points.

In the new STAIMO software, the "Mean of means" solution is used as the initial guess for the "Iterative" Method, which seems to result in better results. Furthermore bisection is used for parameter q in the P-V relation. MARIN claims that this makes the "Iterative" Method more stable and thus reliable. I hope that with my analysis I can help the people at MARIN (and other Research Institutes) to view the problem from a more mathematical perspective and in that way contribute to improving the "Iterative" Method or provide an alternative method.

REFERENCES

[1] J.D. van Manen and P. van Oossanen. *Principles of Naval Architecture, Volume II: Resistance, Propulsion and Vibration.* The Society of Naval Architects and Marine Engineers, 1988.

[2] E.C. Tupper. *Introduction to Naval Architecture.* Butterworth Heinemann, 1996.

[3] J.J. Stoker. Waves caused by a moving pressure point. kelvin's theory of the wave pattern created by a moving ship. In *Water Waves: the Mathematical Theory with Applications*, chapter 8. John Wiley Sons, Inc.

[4] Iso15016: Guidelines for the assessment of speed and power performance by analysis of speed trial data. Technical report, ISO and ITTC, 2015.

[5] G. Strasser, K. Takagai, and S. Werner. A verification of the ittc/iso speed/power trials analysis. *Journal of Marine Science and Technology*, 20:2–13, 2015.

[6] K. Takagi. New standard for speed trial analysis iso 15016:2015. `http://www.jstra.jp/html/PDF/ws20150722_6.pdf`, 2015.

[7] N. Toki. New procedure for the analysis of speed trial results, with special attention to the correction of tidal current effect. *Journal of Marine Science and Technology*, 21:1–22, 2016.

[8] Floris Buwalda. Analysis of methods for determining ship speed during a sea trial, 2016.

[9] K. Madsen, H.B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems (2nd edition), April 2004.

[10] A. Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[11] MATLAB. Equation solving algorithms. `http://nl.mathworks.com/help/optim/ug/equation-solving-algorithms.html#f51887`, 2017.

[12] J.J. Moré and D.C. Sorensen. Computing a trust region step. *Journal of the Society for Industrial and Applied Mathematics*, 4(3):553–572, 1983.

[13] X. Chenguang and A.K. Usman. Directed-distributed gradient descent. `http://www.eecs.tufts.edu/~cxi01/paper/2015D-DGD_Conf.pdf`, 2015.

[14] J. Nocedal and S.J. Wright. *Numerical optimization (2nd edition).* Springer, 2006.

[15] M.I.A. Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar. Technical report, Institute of Computer Science Hellas, 2005.

[16] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[17] H.P. Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems, March 2017.

[18] A. Ranganathan. The levenberg-marquardt algorithm. http://www.ananth.in/docs/lmtut.pdf, 2004.

[19] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[20] G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. John Wiley Sons, 1989.

[21] Analysis of speed/power trial data (sea trials m.v. bulker). Technical report, MARIN.

[22] Analysis of speed/power trial data (sea trials m.v. container). Technical report, MARIN.

[23] Analysis of speed/power trial data (sea trials m.v. supplier). Technical report, MARIN.

[24] STA and MARIN. Staimo. http://www.staimo.org/, 2014–2017.