Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2021

Towards automatic and systematic compressor model selection based on measurement data

Henrik Holmberg

Master of Science Thesis in Electrical Engineering

Towards automatic and systematic compressor model selection based on measurement data

Henrik Holmberg

LiTH-ISY-EX--21/5426--SE

Supervisor: **Viktor Leek**
ISY, Linköpings universitet

Examiner: **Lars Eriksson**
ISY, Linköpings universitet

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Centrifugal compressors are mechanical devices that move and compress gas and are used in a wide range of applications. To get as much efficiency out of the compressors as possible, high demands are put on having good compressor models. One method of compressor modeling is the curve fitting method, where empirical and mathematical models are used to parameterize relationships between different compressor characteristics. The models come in a vast variety of formats, complexity and difficulty of applying them. With the variety of compressors and models to select from, it is important to select an appropriate model. A tool for automatic parameterization, evaluation and selection of compressor model would be very useful.

A model library is proposed in the thesis that utilizes the Total Least Squares method for parameterizing and evaluating the models. By initializing the model library with measurement data, called a compressor map, the compressor is broken down into smaller components in a hierarchy, based on physical expressions and characteristics. The individual compressor models are defined and parameterized on top of the base model, overriding and replacing the base model feature. Total Least Squares offers a countermeasure to avoid large model prediction errors associated with the nonlinear behavior of the compressor. To utilize the model library further, a simulation interface is proposed, so the selected compressor model can easily be applied in an engine and vehicle simulation environment.

The model library is evaluated using three different compressor maps and shows promising results, where the TLS method seem to better capture bad visual curve-fitted models, compared to other simpler methods. Parameterization of the models are done automatically without any manual user input needed. The simulation interface is tested by replacing the compressor block of an engine simulation model, where the results show that the model component hierarchy can be integrated in a simulation environment.

Sammanfattning

Centrifugalkompressorer är mekaniska apparater som flyttar och komprimerar gas och används inom ett brett användningsområde. För att få ut så mycket effektivitet som möjligt ur kompressorerna, ställs det höga krav på bra kompressormodeller. En kompressormodellsmetod är *curve-fitting*, eller kurvanpassning, där empiriska och matematiska modeller används för att parameterisera samband mellan olika kompressorkaraktärer. Modellerna kommer i en bred utsträckning av olika format, komplexiteter och svårighetsgrader. Med alla dessa variationer på kompressorer och modeller, är det viktigt att välja en lämplig modell. Ett verktyg för automatisk parameterisering, utvärdering och val av kompressormodell vore av stort intresse.

I denna rapport föreslås ett modellbibliotek, där metoden Total Least Squares utnyttjas för parameterisering och utvärdering av modeller. Genom att initiera modellbiblioteket med mätdata, kallat kompressormap, bryts kompressorn ner i mindre komponenter i en hierarki baserat på dess fysiska uttryck och egenskaper. De individuella kompressormodellerna defineras, parameteriseras ovanpå basmodellen och sätter sig över specifika komponenter. Total Least Squares erbjuder en lösning för att undvika stora modellfel hos kompressorns olinjära beteende. För att utnyttja modellbiblioteket vidare föreslås ett gränssnitt för simulering, för att möjliggöra användning av den valda modellen i en simuleringsmiljö.

Modellbiblioteket utvärderas med hjälp av tre olika kompressormappar och visar lovande resultat, där TLS-metoden verkar bättre fånga de visuellt dåligt anpassade modellerna, till skillnad mot enklare metoder. Parameteriseringen av modellerna görs automatiskt utan behov av extra manuell input. Gränssnittet för simulering testas genom att byta ut och ersätta kompressorblocket i simuleringsmodellen för en förbränningsmotor, där resultaten visar att komponenthierarkin för modellerna kan implementeras i en simuleringsmiljö.

Acknowledgments

I would like to thank my supervisor Viktor Leek and examiner Lars Eriksson for all the help and guidance through the project.

Norrköping, Juni 2021
Henrik Holmberg

Contents

Notation

Abbreviations

Abbreviation	Meaning
ICE	Internal Combustion Engine
LSQ	Least Squares
TLS	Total Least Squares
SpL	Speed Line
ZSL	Zero Slope Line
ChL	Choke Line

Symbols

Symbol	Description
N_c	Compressor rotational speed [RPM]
$\tilde{N}_c$	Corrected compressor rotational speed [RPM]
$N_{c,n}$	Normalized compressor rotational speed [RPM]
$\dot{m}_c$	Compressor mass flow [kg/s]
$\tilde{m}_c$	Corrected compressor mass flow [kg/s]
$\dot{m}_{ZSL}$	Compressor mass flow zero slope line [kg/s]
$\dot{m}_{ChL}$	Compressor mass flow choke line [kg/s]
Π_c	Compressor pressure ratio [-]
Π_{ZSL}	Compressor pressure ratio zero slope line [-]
Π_{ChL}	Compressor pressure ratio choke line [-]
$T_{c,ref}$	Compressor map reference temperature [K]
T_{01}	Compressor inlet temperature [K]
T_{02}	Compressor outlet temperature [K]
$p_{c,ref}$	Compressor map reference pressure [Pa]
p_{01}	Compressor inlet pressure [Pa]
p_{02}	Compressor outlet pressure [Pa]
d_c	Compressor impeller diameter [m]
U_c	Compressor impeller tip speed [m/s]
ω_c	Compressor angular speed [rad/s]
P_c	Compressor power [W]
Ψ	Head coefficient [-]
Φ	Flow coefficient [-]
Γ_c	Torque coefficient [-]
Δh_{is}	Compressor isentropic enthropy [J/K]
Δh_{act}	Compressor isentropic actual enthropy [J/K]
η_c	Compressor isentropic efficiency [-]
Ma	Compressor mach number [-]
c_p	Constant pressure specific heat capacity [J/(kg K)]
γ	Heat capacity ratio [-]
ρ	Density [Kg/m ^{3}]
R	Gas constant [J/(mol $\cdot$ K)]
k_{loss}	Compressor impeller friction and losses [-]

1

Introduction

Centrifugal compressors are used in a so wide range of applications, from refrigeration systems to vehicle and jet engines [1]. Compressors are mechanical devices that move and compress gas, and are similar to pumps which moves incompressible fluids. Compressors are widely used in refrigeration systems to compress air and vapors, so called air-cycle and vapor-cycle, in the compression phase. In vehicles using internal combustion engines (ICE), it is common to use one or more centrifugal compressors to increase the intake air density through forced induction and downsizing the engine volume. The compressor can be driven by either a belt from the engine, which is called a supercharger, or by a gas turbine driven by the engine exhaust gases and connected through a shaft, called a turbocharger. Turbochargers are a great way to improve engine power, emissions and efficiency [2] since they use the otherwise wasted energy in the engine exhausts. Jet engines, like a turbocharger, consist of a gas turbine driven compressor that delivers compressed air to a combustion chamber, where the air is mixed with fuel. The air-fuel mixture is ignited and the hot pressurized air are fed into the turbine which in turn drives the compressor [3]. Turbocharger compressors are also a good choice for use in fuel cell technology where the cells require constant supplies of pressurized air because of their high-power density [4].

Getting as much efficiency out of the compressors as possible, puts high demand on having good compressor models. The compressor models however, are complex, with a wide area of use, and the characteristics and normal operational regions of the compressors may differ a lot. This can lead to a challenge in finding a fitting model for a compressor, since a particular model might fit well in one application but less well in another. Measured data often only covers the normal operational region where the efficiency is the highest, while in a real-world usage,

the outer regions are also used to a certain extend. For this reason, the models need to be able to extrapolate outside of the operational region [5].

1.1 Motivation

There are different types of compressor model methodologies, such as, the look-up table method, curve fitting method, neural network method and theoretical model method [2]. The look-up table method uses an appropriate interpolation algorithm to interpolate the measurement data points in a mesh that for example describes the compressor mass flow as a discrete function of the pressure ratio and rotational speed. The method can be simple and accurate within the span of the measurement data mesh, but lacks the extrapolation outside the mesh, as well as being a non-continuous function. The curve fitting methods use empirical or mathematical models to parameterize relationships between different characteristic compressor expressions and comes in a vast variety of formats, complexity and difficulty of applying them. Some models describe the relationship between the compressor mass flow rate, pressure ratio, isentropic efficiency and rotational speed which are available in the measured data. Other models instead use the relationship between different dimensionless coefficients derived from expressions describing the compressor's characteristics. Compared to the look-up method, the curve-fitted models have the strength of continuous functions that naturally can span and extrapolate outside of the normal operational region, given that the model has taken the outside regions into account. Like the curve-fitting method, the neural network method uses a neural network to fit a model to a compressor characteristic relationship. The challenge lies in how to structure the neural network with number of nodes, choice of algorithm etc, to avoid under- or over-fitting the model to the data. Lastly, the theoretical method is aimed at improving the previously mentioned methods in that they can lack in extrapolation performance outside of the measured data. Instead of using measurement data to fit a model, the theoretical models instead try to predict the working compressor characteristics by modeling the physical properties of the compressor such as geometric dimensions, the velocity triangle, fluid dynamics, friction losses and so on. The models can be categorized by their dimensions as 1-D, 2-D and 3-D, where complexity and computation time increase for higher number of dimensions.

The curve-fitting method has the advantage over the look-up table method with its possibility of using mathematical functions to model outside the normal operational region. The physical bases of some curve-fitted models can be an extra strength compared to the neural network methods, which only use relationships in the data. The theoretical method doesn't need measurement data, but the very complex models makes the curve-fitted method more suitable, because of its simpler models. Also, according to [2] the curve-fitting method is the most commonly used method among studies.

1.1.1 Curve-fitted models

As mentioned, the curve-fitted compressor models comes in a vast variety and different fundamental ideas.

Malkhede et al. [6] models the compressor mass flow rate as a polynomial function of the pressure ratio and rotational speed directly from measured data. Karlsen [7] use an exponential function to model the relationship between the dimensionless flow coefficient and the head coefficient and mach number, as well as modeling the isentropic efficiency as a fifth order polynimial function of the mass flow rate and pressure ratio. Guan Cong et al. [8] also models the relationship between dimensionless flow coefficient and the head coefficient and mach number as an expression based on the Jensen model [9]. The model is divided into different zones by compressor rotational speed and includes a model for isentropic efficiency as a second order polynomial function of the torque coefficient.

Hadef et al. [10, 11] models the actual isentropic enthalpy through a simplified expression of the physical velocity characteristics triangle of the compressor impeller, which uses the compressor mass flow rate and rotational speed, which was later further improved in the Ellipse model [12, 13]. The actual isentropic enthalpy can then be used to calculate the compressor isentropic efficiency. Like the Hadef et al. model, Zeng Tao [14] also utilizes the velocity triangle and Euler's equations for turbomachinery to model the compressor power by, again, using the compressor mass flow rate and rotational speed. Müller et al. [15] also use the velocity triangle to propose two models by modeling the isentropic enthalpy, where the first model utilizes the so called Zero Slope Line of the mass flow rate.

The Ellipse compressor mass flow model [12, 13, 16] uses an elliptic function, which returns functions to calculate, either compressor mass flow rate from pressure ratio and rotational speed, or the other way around with pressure ratio from mass flow rate and rotational speed. The model also uses specific models for the outer regions outside normal operational use.

The study [2] tests and evaluates a large number of compressor models [6–8, 10–17] for use with large-scale marine turbocharger compressors. The study cover models for both compressor mass flow rate and isentropic efficiency. Evaluation of the models have been carried out using methods of model prediction errors, with some surprising results as seen in Figure 1.1 as an example. Here the predicted *Müller II* model to the left visually looks worse than the *Ellipse* model to the right, while almost all statistic results in [2] claim the opposite. The evaluation methods compare the model predictions to the measurement data, which could be sensitive to large errors in the nonlinear characteristics of the compressor. One solution to avoid these errors, is to use the Total Least Squared method which also takes model input error into consideration, resulting in a two dimensional error that could be more suitable for nonlinear functions. More about this issue in Section 2.6.

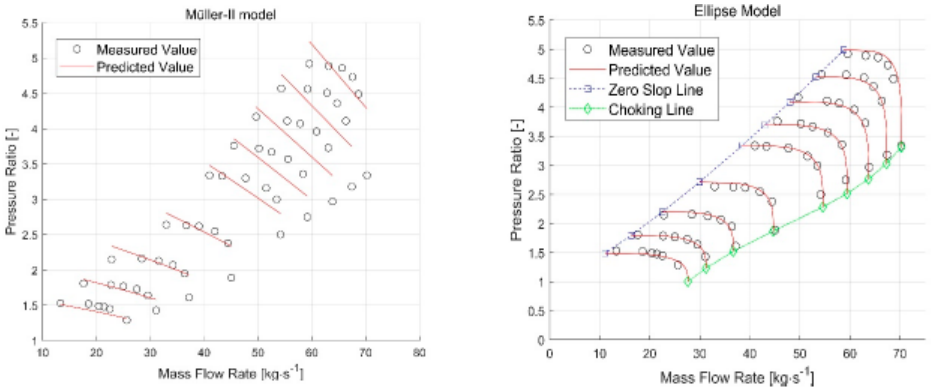


Figure 1.1: *A surprising result in [2] is that the Muller II Model is stated to have better performance than the Ellipse Model.*

1.1.2 Model parameterization interface

Model parameterization can be done with the help of various program tools and function. For example, the MATLAB functions `lsqnonlin` and `lsqcurvefit` are Least Square solvers made for nonlinear problems.

LiU CPgui [18] is a MATLAB toolbox used to parameterize the Ellipse compressor models [12, 13, 16] for a given compressor map. The toolbox uses a graphical user interface (GUI) to parameterize a compressor mass flow rate and isentropic efficiency model, by going through several steps of initiation and curve fitting. As an extra module, the toolbox also comes with a Simulink library block that can be used to implement the parameterized models in a simulation environment. The downside to the toolbox is that it is strictly only for the Ellipse models, with no simple framework to allow extending it to other models, and simple express functionality to avoid going through all the parameterization steps.

1.2 Purpose

As mentioned, there are a wide range of applications for centrifugal compressors, and at the same time, an abundance of different compressor models of high variety. With the importance of selecting appropriate compressor models for a given compressor and application computer based analysis, tools could be of very good use. The purpose of this thesis is to develop a compressor model library. The aim is to use the library as a tool to help evaluate different compressor models on compressor maps and select the best model candidate. Another goal is to develop a simulation interface where the library and the selected model can be implemented in a simulation environment. A user-friendly and automatic approach should be followed for the library interface where manual input and intervention are kept at a minimum. Model implementation should also be open for future ex-

tensions, additions and experiments.

Different compressor models are to be evaluated with the compressor model library as a tool. This thesis will focus on the curve-fitted empirical and mathematical modeling methods as mentioned in Section 1.1.1. The models will be presented with more in detail in Chapter 2. The explored model methods vary in difficulty and complexity, leading to varying accuracy and computational time. Measured compressor data are available through a database of a few hundred turbocharger compressors of a wide range of sizes and applications. A small selection of measurement data of three turbocharger compressors will be used for testing and evaluation covering the range from large marine use, to smaller car engines, with the aim of covering the most important range of applications that have a general use of compressors.

For model evaluation the *Total Least Square* method will be used as a countermeasure to avoid large nonlinear model errors. By evaluating the models with TLS as an additional tool for model evaluation, the results of the study [2] are to be repeated and evaluated with the aim to confirm if the results seem reasonable or not.

The purpose is divided into the following three questions:

1. Can a compressor model library with automatic parameterization be implemented?
2. Is Total Least Squares a viable evaluation method for compressor models and can the method be used to reevaluate the results in the study [2]?
3. Can the parameterized library models be implemented into a simulation interface?

2

Theory

In this chapter, the theory used in the thesis is presented. The chapter starts with some basic information about the compressor in the form of a turbocharger, compressor maps and how they are obtained. The generic compressor models are presented, followed by the derived compressor models to be used in the library. Lastly the Total Least Squares method used for model evaluation is presented.

2.1 Turbocharger compressor

As mentioned, the compressor is a mechanical device that move and compress gas. One example of a compressor application is the turbocharger compressor, which will be used as an example. A simple schematic of a turbocharger can be seen in Figure 2.1. The turbocharger consists of the centrifugal compressor and a radial turbine connected with a shaft seated in a bearing housing. The turbine is driven by leftover energy in the exhaust from the combustion engine [19] which can be seen as the enthalpy loss $\Delta h_{\text{loss}} = c_{\text{p}}(T_{04} - T_{01})$, where T_{03} and T_{04} is the inlet and outlet temperature of the turbine. Air enters the compressor inlet with mass flow $\dot{m}_{\text{c}}$, temperature T_{01} and pressure p_{01}. The air is compressed and fed into the engine with temperature T_{02} and pressure p_{02}. The engine exhaust gasses of temperature T_{03} and pressure p_{03} enters and spins up the turbine which in turn spins the compressor shaft with angular speed ω_{c} and exits the turbine with temperature T_{04} and pressure p_{04}.

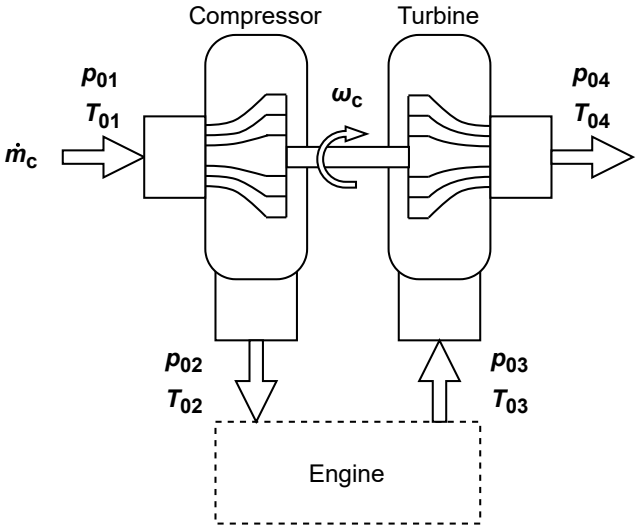


Figure 2.1: *Turbocharger compressor drawing.*

2.2 Compressor map

The characteristics of the compressor is usually represented by maps as seen in Figure 2.2. In the case of the compressor its map consists of a plot with the mass (or alternatively volumetric) flow rate as x-axis and pressure ratio the y-axis. Measured data points of specific compressor rotational speeds are plotted in Speed Lines (SpL). The map is mainly divided into three main regions. The operating region is the area of the highest efficiency, where the compressor normally operates and where the data points are mostly measured. The surge region is a region of instability where the mass flow is lower, and the pressure ratio is dropping off. In the Choke region, the pressure ratio goes towards dropping under a value of 1, while the flow stops increasing. The regions are divided by the *Zero Slope Line* where the derivative of the pressure ratio is zero, and the *Choke Line* where the derivative goes towards negative infinity [19]. The data points are measured at specific compressor rotational speeds, or *Speed Lines*, and the map data often uses the corrected values

$$\bar{N}_c = N_c \frac{1}{\sqrt{T_{01}/T_{c,ref}}} \tag{2.1a}$$

$$\bar{m}_c = \dot{m}_c \frac{\sqrt{T_{01}/T_{c,ref}}}{p_{01}/p_{c,ref}} \tag{2.1b}$$

where $T_{c,ref}$ and $p_{c,ref}$ are the reference temperature and pressure used in the map while T_{01} and T_{02} are the compressor inlet temperature and pressure [12]. The corrected values is a way of normalizing the inlet air temperature and pressure to not depend on differences in the ambient environment such as time of the day, seasons, elevation etc.

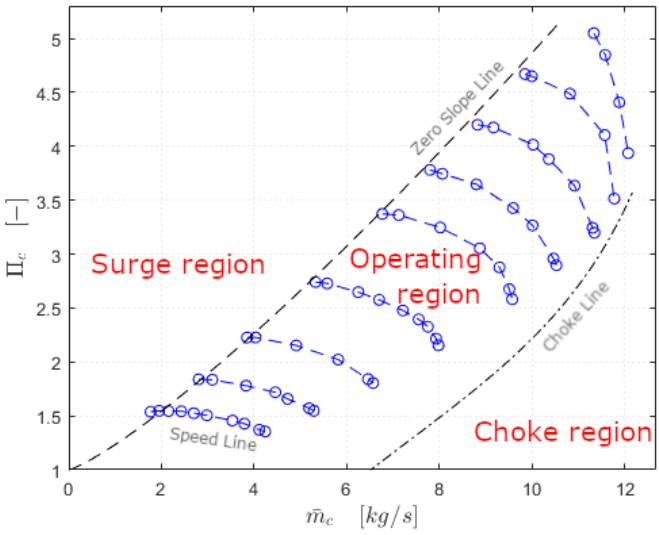


Figure 2.2: *Compressor flow map.*

2.3 Compressor basics

The power of the compressor shaft compresses the inlet air of temperature T_{01} and pressure p_{01} into higher outlet temperature T_{02} and pressure p_{02} which for a steady flow gives the expression

$$P_c = \dot{m}_c c_p (T_{02} - T_{01}) \tag{2.2}$$

where c_p is a constant value for the specific heat transfer coefficient in the compressor. By using the ideal gas law, the ideal outlet temperature $T_{02,ideal}$ for an isentropic process is

$$\frac{T_{02,ideal}}{T_{01}} = \frac{p_{02}}{p_{01}}^{\frac{\gamma-1}{\gamma}} \tag{2.3}$$

which in turn gives the ideal shaft power $P_{c,ideal}$ needed by the compressor

$$P_{c,ideal} = \dot{m}_c \Delta h_{is} = \dot{m}_c c_p T_{01} \left[\left(\frac{p_{02}}{p_{01}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \tag{2.4}$$

$$\eta_c = \frac{\Delta h_{is}}{\Delta h_{act}} = \frac{\left(\frac{p_{02}}{p_{01}} \right)^{\frac{\gamma-1}{\gamma}} - 1}{\frac{T_{02}}{T_{01}} - 1} \tag{2.5}$$

Compressors mainly use the pressure ratio Π_c as the difference between the inlet and outlet pressure as $\Pi_c = \frac{p_{02}}{p_{01}}$ to describe how the inlet gas is pressurized. To

calculate the outlet temperature and pressure the expression in (2.5) and pressure ratio is used which gives the two following expressions

$$T_{02} = T_{01}\left[\frac{1}{\eta_c}\left(\Pi^{\frac{\gamma-1}{\gamma}} - 1\right) + 1\right] \tag{2.6}$$

$$p_{02} = \Pi_c \cdot p_{01} \tag{2.7}$$

2.3.1 Dimensionless numbers

Since the centrifugal compressor is like a pump the affinity laws for fluid dynamics of pumps can be used. The head coefficient Ψ and the flow coefficient Φ are dimensionless coefficients that shows the energy transfer and air flow through the compressor by impeller tip speed and can be used to describe the characteristics of the compressor. An alternative to the normal compressor inlet temperature T_{01} and pressure p_{01} is the mach number Ma which shows the fluid velocity relative to the speed of sound and is also dimensionless. These dimensionless numbers are defined as

$$\Psi = \frac{\Delta h_{is}}{\frac{1}{2}U_c^2} = \frac{c_p T_{01}\left(\Pi^{\frac{\gamma-1}{\gamma}} - 1\right)}{\frac{1}{2}U_c^2} \tag{2.8a}$$

$$\Phi = \frac{\dot{m}_c}{\frac{\pi}{4}\rho d_c^2 U_c} \tag{2.8b}$$

$$Ma = \frac{U_c}{\sqrt{\gamma R T_{01}}} \tag{2.8c}$$

$$U_c = \frac{\pi d_c N_c}{60} \tag{2.8d}$$

where U_c is the compressor impeller tip speed.

2.4 Compressor flow models

In this section, the compressor flow models used in this thesis are presented. The compressor flow models are divided into two function categories as seen in (2.9) and (2.10). A special case is the Ellipse model in Section 2.4.1 that supports both functions.

$$\dot{m}_c = f(\Pi_c, N_c) \tag{2.9}$$

$$\Pi_c = f(\dot{m}_c, N_c) \tag{2.10}$$

The models are defined for either corrected or non-corrected values in their respective studies. To avoid a cluttered notation, all models presented here are expressed using only non-corrected values for the compressor rotational speed and mass flow rate.

2.4.1 Ellipse model

The Ellipse model [12, 13, 16] divides the model into the surge, operating and choke zones. The zones are divided by the modeled Zero Slope Line (ZSL) using (2.12c) (2.12d) and the Choke Line modeled by (2.12a) (2.12b). The operating zone between the ZSL and ChL ($\dot{m}_{\mathrm{ZSL}} \leq \dot{m}_{\mathrm{c}} < \dot{m}_{\mathrm{ChL}}$) is modeled as an implicit elliptic function

$$\left(\frac{\dot{m}_{\mathrm{c}} - \dot{m}_{\mathrm{ZSL}}}{\dot{m}_{\mathrm{ChL}} - \dot{m}_{\mathrm{ZSL}}}\right)^{CUR} + \left(\frac{\Pi_{\mathrm{c}} - \Pi_{\mathrm{ChL}}}{\Pi_{\mathrm{ZSL}} - \Pi_{\mathrm{ChL}}}\right)^{CUR} = 1 \tag{2.11}$$

where the parameters are defined as

$$\dot{m}_{\mathrm{ChL}}(N_{\mathrm{c,n}}) = \dot{m}_{\mathrm{c,max}}\left(C_{1,1} + C_{1,2}\arctan(C_{1,3}N_{\mathrm{c,n}} - C_{1,4})\right) \tag{2.12a}$$

$$\Pi_{\mathrm{ChL}}(N_{\mathrm{c,n}}) = \Pi_{\mathrm{c,max}}\left(C_{2,1} + C_{2,2}N_{\mathrm{c,n}}^{C_{2,3}}\right) \tag{2.12b}$$

$$\dot{m}_{\mathrm{ZSL}}(N_{\mathrm{c,n}}) = \dot{m}_{\mathrm{c,max}}\left(C_{3,1}N_{\mathrm{c,n}}^{C_{3,2}}\right) \tag{2.12c}$$

$$\Pi_{\mathrm{ZSL}} = 1 + \left(1 - \Pi_{\mathrm{c,max}}\right)C_{4,1}N_{\mathrm{c,n}}^{C_{4,2}} \tag{2.12d}$$

$$CUR = C_{5,1} + C_{5,2}N_{\mathrm{c,n}}^{C_{5,3}} \tag{2.12e}$$

These functions use normalized compressor rotational speeds

$$N_{\mathrm{c,n}} = \frac{N_{\mathrm{c}}}{N_{\mathrm{c,max}}} \tag{2.13}$$

The implicit (2.11) can be solved for either mass flow $\dot{m}_{\mathrm{c}}$ or pressure ratio Π_{c} using the explicit expressions

$$\dot{m}_{\mathrm{c}} = \dot{m}_{\mathrm{ZSL}} + (\dot{m}_{\mathrm{ChL}} - \dot{m}_{\mathrm{ZSL}})\left[1 - \left(\frac{\Pi_{\mathrm{c}} - \Pi_{\mathrm{ChL}}}{\Pi_{\mathrm{ZSL}} - \Pi_{\mathrm{ChL}}}\right)^{CUR}\right]^{\frac{1}{CUR}} \tag{2.14a}$$

$$\Pi_{\mathrm{c}} = \Pi_{\mathrm{ChL}} + (\Pi_{\mathrm{ZSL}} - \Pi_{\mathrm{ChL}})\left[1 - \left(\frac{\dot{m}_{\mathrm{c}} - \dot{m}_{\mathrm{ZSL}}}{\dot{m}_{\mathrm{ChL}} - \dot{m}_{\mathrm{ZSL}}}\right)^{CUR}\right]^{\frac{1}{CUR}} \tag{2.14b}$$

2.4.2 Guan Cong model

Guan Cong et al. [8] proposed using the Jensen et al. [9] model that models the head coefficient as a function of the flow coefficient Φ and the Mach number Ma with the expression

$$\Psi = \frac{k_1 + k_2 Ma + k_3\Phi + k_4 Ma\Phi}{k_5 + k_6 Ma - \Phi} \tag{2.15}$$

which can give the flow coefficient as a function of the head coefficient and the Mach number

$$\Phi = \frac{k_1 + k_2 Ma + k_3 \Psi + k_4 Ma \Psi}{k_5 + k_6 Ma - \Psi} \tag{2.16}$$

The map is divided into zones, which is each Speed Line. The model is then parameterized for each zone.

2.4.3 Zero Slope Line Method

Kolmanovsky et al. [17] propose modeling the flow coefficient Φ as an exponential function of pressure ratio Π_c and compressor rotational speed N_c. The model called *Zero Slope Line Method* (ZSLM) utilizes the Zero Slope Line by modeling it as the polynomials

$$\Phi_{ZSL} = k_1 N_c + k_2 N_c^2 \tag{2.17}$$

$$\Pi_{ZSL} = k_3 + k_4 \dot{m}_{ZSL}^2 \tag{2.18}$$

The area to the right of the Zero Slope Line $\Pi_c < \Pi_{ZSL}$ is then represented by the exponential expression

$$\frac{\Phi}{\Phi_{ZSL}} = 1 + \alpha \left[1 - e^{k_5 \left(\frac{\Pi_c}{\Pi_{ZSL}} - 1 \right)} \right] \tag{2.19}$$

while the left side is linearly extended using the expression

$$\frac{\Phi}{\Phi_{ZSL}} = 1 - \alpha k_5 \left(\frac{\Pi_c}{\Pi_{ZSL}} - 1 \right) \tag{2.20}$$

where the parameter α is modeled as

$$\alpha = k_6 e^{-k_7 N_c} \tag{2.21}$$

2.4.4 Karlsen

Karlsen [7] propose modeling the flow coefficient Φ as an exponential function of the head coefficient Ψ and the Mach number Ma. The equations are similar to the Kolmanovsky model in Section 2.4.3 where the first model called the *Exponential Method* has the expression

$$\Phi = a + \left(1 - e^{\Psi^c + b} \right) \tag{2.22}$$

where

$$a = a_0 + a_1 Ma + a_2 Ma^2 + a_3 Ma^3 \tag{2.23a}$$

$$b = b_0 + b_1 Ma + b_2 Ma^2 \tag{2.23b}$$

$$c = c_0 + c_1 Ma + c_2 Ma^2 \tag{2.23c}$$

The second model is called the *Exponential Threshold Method* which scales the head coefficient Ψ by a threshold Ψ_{th} using the expression

$$\Phi = \alpha \left[1 - e^{\beta\left(\frac{\Psi}{\Psi_{\text{th}}} - 1\right)} \right] \tag{2.24}$$

where

$$\alpha = \alpha_0 + \alpha_1 Ma + \alpha_2 Ma^2 + \alpha_3 Ma^3 \tag{2.25a}$$

$$\beta = \beta_0 + \beta_1 Ma + \beta_2 Ma^2 \tag{2.25b}$$

$$\Psi_{\text{th}} = c_0 + c_1 Ma + c_2 Ma^2 + c_3 Ma^3 + c_4 Ma^4 \tag{2.25c}$$

2.4.5 Müller models

Müller et al. [15] proposes two models based on the compressor isentropic enthalpy as seen in (2.27) and (2.31). The models can then be used to calculate the compressor pressure ratio Π_{c} using the expression

$$\Pi_{\text{c}} = \left(\frac{\Delta h_{\text{is}}}{c_{\text{p}} T_{01}} + 1 \right)^{\frac{\gamma}{\gamma - 1}} \tag{2.26}$$

The first model uses simplified physical impeller models to model the isentropic enthalpy Δh_{is} with the help of the mass flow at zero slope line $\dot{m}_{\text{ZSL}}$ that gives the expression

$$\Delta h_{\text{is}} = U_{\text{c}}^2 \left[A \left(\frac{\dot{m}_{\text{c}}}{U_{\text{c}}} \right)^2 + B \left(\frac{\dot{m}_{\text{c}}}{U_{\text{c}}} \right) + C \right] \tag{2.27}$$

where

$$A = A_2 U_{\text{c}}^2 + A_1 U_{\text{c}} + A_0 \tag{2.28}$$

$$B = \frac{2A\dot{m}_{\text{ZSL}}}{U_{\text{c}}}, \quad \dot{m}_{\text{ZSL}} = m_2 U_{\text{c}}^2 + m_1 U_{\text{c}} + m_0 \tag{2.29}$$

$$C = C_2 U_{\text{c}}^2 + C_1 U_{\text{c}} + C_0 \tag{2.30}$$

The second model is in a much more simplified form which utilizes that the dominant dependence of Δh_{is} is the square of the compressor rotational speed and that the mass flow rate dependence is mostly independent of the compressor rotational speed for low to medium compressor rotational speeds.

$$\Delta h_{\text{is}} = s_1 {U_{\text{c}}}^2 + s_2 \dot{m}_{\text{c}}^2 + s_3 \dot{m}_{\text{c}} + s_4 \tag{2.31}$$

2.4.6 Malkhede model

Malkhede et al. [6] used a compressor flow map to curve fit a three-dimensional polynomial for the compressor flow as a function of the compressor pressure ratio.

$$\dot{m}_c = a_1 + a_2 N_c + a_3 N_c^2 + a_4 N_c^3 + \frac{a_5}{\Pi_c} + \frac{a_6}{\Pi_c^2} + \frac{a_6}{\Pi_c^3} + a_8 \frac{N_c}{\Pi_c} + a_9 \frac{N_c}{\Pi_c^2} + a_{10} \frac{N_c^2}{\Pi_c} \quad (2.32)$$

2.5 Compressor efficiency models

In this section are the compressor efficiency models used in this thesis.

2.5.1 Guan Cong model

The Guan Cong efficiency model, just like its mass flow model in Section 2.4.2, divides the map into zones. It is proposed [8] to model the linear relationship between the non-dimensional torque constant Γ_c and flow coefficient Φ. The torque coefficient is defined as

$$\Gamma_c = \frac{\frac{\dot{m}_c c_p T_{01}}{\omega_c \eta_c} \left(\Pi^{\frac{\gamma-1}{\gamma}} - 1 \right)}{\rho \pi d_c^3 U_c^2} \quad (2.33)$$

where $\omega_c = 2/d_c * N_c$, and modeled as the expression

$$\Gamma_c = k_1 \Phi + k_2 \quad (2.34)$$

Linear interpolation is used to approximate parameter $k_j, j = 1, 2$ in a specific zone according to

$$k_j = k_{j,lower} + \frac{(k_{j,higher} - k_{j,lower})}{(N_{higher} - N_{lower})} (N_c - N_{lower}) \quad (2.35)$$

where *lower* and *higher* implies the neighboring speed lines below and above than the current rotational speed N_c. Since the calculated isentropic efficiency may differ from the actual values, it is proposed to use a correction. The maximum isentropic efficiency is held consistent with the actual values using the following expression

$$\eta_{c,corr} = \eta_{c,non-corr} \frac{\eta_{c,max}(N_c)}{\eta_{c,non-corr,max}(N_c)} \quad (2.36)$$

where $\eta_{c,max}(N_c)$ is the maximum compressor isentropic efficiency for the compressor rotational speed N_c and $\eta_{c,non-corr,max}(N_c)$ is the maximum uncorrected isentropic efficiency for the rotational speed N_c calculated from the predicted model.

2.5.2 Karlsen

Karlsen [7] models the isentropic efficiency as a fifth order polynomial function of the mass flow rate $\dot{m}_c$ and pressure ratio Π_c with a binominal expansion form as seen in (2.37).

$$\begin{aligned}\eta_c = {} & p_{00} \\ & + p_{10}\dot{m}_c + p_{01}\Pi_c \\ & + p_{20}\dot{m}_c^2 + p_{11}\dot{m}_c\Pi_c + p_{02}\Pi_c^2 \\ & + p_{30}\dot{m}_c^3 + p_{21}\dot{m}_c^2\Pi_c + p_{12}\dot{m}_c\Pi_c^2 + p_{03}\Pi_c^3 \\ & + p_{40}\dot{m}_c^4 + p_{31}\dot{m}_c^3\Pi_c + p_{22}\dot{m}_c^2\Pi_c^2 + p_{13}\dot{m}_c\Pi_c^3 + p_{04}\Pi_c^4 \\ & + p_{50}\dot{m}_c^5 + p_{41}\dot{m}_c^4\Pi_c + p_{32}\dot{m}_c^3\Pi_c^2 + p_{23}\dot{m}_c^2\Pi_c^3 + p_{14}\dot{m}_c\Pi_c^4 + p_{05}\Pi_c^5\end{aligned} \tag{2.37}$$

2.5.3 Hadef et al.

The Hadef et al. model [10, 11] uses the velocity characteristics triangle of a centrifugal compressor with Euler's equation to get a physical model expression

$$\Delta h_{act} = U_c^2 - \frac{U_c \cot \beta_{out}}{\rho_{out} A_{out}} \dot{m}_c \tag{2.38}$$

for the actual isentropic enthalpy Δh_{act} where β_{out}, ρ_{out} and A_{out} is the outlet angle, air density and area. In [11] the expression is proven to have a linear relationship between the mass flow rate and isentropic enthalpy as

$$\Delta h_{act} = b - a\dot{m}_c \tag{2.39}$$

where a and b are functions of the compressor rotational speed ($a = b = f(N_c)$) which finally gives the expression

$$\Delta h_{act} = \left(b_1 N_c + b_2 N_c^2\right) - \left(a_1 N_c + a_2 N_c^2\right)\dot{m}_c \tag{2.40}$$

The isentropic efficiency can finally be calculated using the (2.5).

2.5.4 Zeng Tao model

The Zeng Tao model [14] uses the Euler equations for turbomachinery to model the compressor power by only three parameters. By defining $C_{Power} = P_c/\dot{m}_c^3$ and $C_{Speed} = \omega_c/\dot{m}_c$ gives the equation

$$C_{Power} = \epsilon_1 (C_{Speed})^2 - \epsilon_2 (C_{Speed}) + \epsilon_3 \tag{2.41}$$

which can then in turn be rewritten as

$$P_c = \dot{m}_c^3 \left[\epsilon_1 \left(\frac{\omega_c}{\dot{m}_c}\right)^2 - \epsilon_2 \left(\frac{\omega_c}{\dot{m}_c}\right) + \epsilon_3\right] \tag{2.42}$$

Using the (2.2) and (2.5) the actual compressor power is given as

$$P_c = \dot{m}_c \Delta h_{act} = \frac{\dot{m}_c \Delta h_{is}}{\eta_c} \tag{2.43}$$

2.5.5 Ellipse model

To compliment the Ellipse compressor mass flow model in Section 2.4.1 an isentropic efficiency model is proposed [12, 13] based on the model in Section 2.5.3. The model is extended into the expression

$$\Delta h_{\text{act}} = (1 + k_{\text{loss}}(N_{\text{c}}, \dot{m}_{\text{c}}))\left(b(N_{\text{c,n}}) - a(N_{\text{c,n}})\dot{m}_{\text{c}}\right) \tag{2.44}$$

where k_{loss}, $b(N_{\text{c,n}})$ and $a(N_{\text{c,n}})$ is defined as

$$b(N_{\text{c,n}}) = \Delta h_{\text{act,max}}\left(C_{\text{b,1}}N_{\text{c,n}}^2 + C_{\text{b,2}}N_{\text{c,n}}^3\right) \tag{2.45a}$$

$$a(N_{\text{c,n}}) = \frac{\Delta h_{\text{act,max}}}{\dot{m}_{\text{c,max}}}\frac{C_{\text{a,1}}N_{\text{c,n}}}{\left(1 + C_{\text{a,2}}N_{\text{c,n}}^2\right)^{C_{\text{a,3}}}} \tag{2.45b}$$

$$k_{\text{loss}}(N_{\text{c}}, \dot{m}_{\text{c}}) = \frac{C_{\text{loss}}\rho d_c^3 \pi N_{\text{c}}}{60\dot{m}_{\text{c}}} \tag{2.45c}$$

and the compressor rotational speed N_{c} is normalized as

$$N_{\text{c,n}} = \frac{N_{\text{c}}}{N_{\text{c,max}}} \tag{2.46}$$

Here the k_{loss} is the compressor impeller friction and other losses such as flow recirculation at low flows and leakage. The function $a(N_{\text{c,n}})$ has been extended to take density change into consideration.

2.6 Total least squares

Total least squares (*TLS*) [20] is a method for finding an approximation of a problem by minimizing the errors to a orthogonal point of the solution. In the case of this thesis it is used to curve fit a model to measured data. Figure 2.3 shows an example of when solutions using total least squares (*TLS*) and least squares (*LSQ*) are compared. The normal least squares finds the solution by minimizing the error on the y-axis as in (2.47), while total least squares also uses an error on the input. This can avoid huge errors at function slopes as illustrated at the right in the left-hand figure which can occur for compressor map shapes. In (2.48) the total sum of the TLS where ϵ is used for normalizing both x and y-axes. The term ϵ is used for normalization between x and y.

$$LSQ = \sum_{i=1}^{N}(y_i' - y_i)^2 \tag{2.47}$$

$$TLS = \sum_{i=1}^{N}\left((y_i' - y_i)^2 + \epsilon(x_i' - x_i)^2\right), \quad \epsilon = \left(\frac{mean(y)}{mean(x)}\right)^2 \tag{2.48}$$

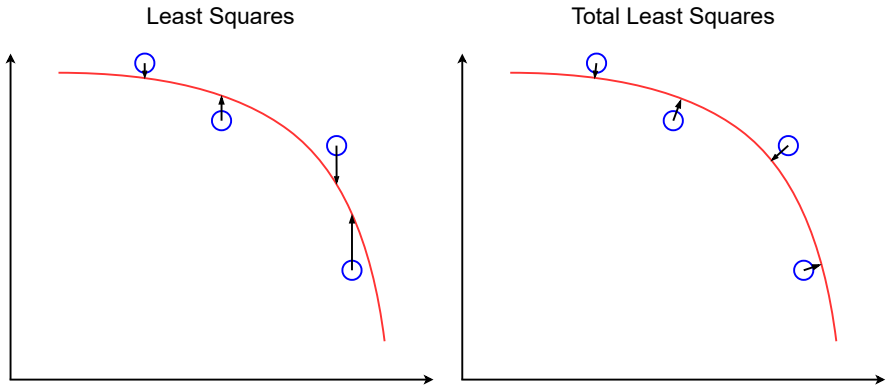


Figure 2.3: *Comparison example between Least Squares and Total Least Squares error.*

2.7 Simulation

There are several available applications for a simulation solution. In this Section a few examples are mentioned.

2.7.1 Simulink

Simulink [21] is a MATLAB-based programing environment using mainly a causal graphical block diagram interface. Causal implying that the signal flows one way, from input to output. It is possible to generate C source code for use in real-time applications. LiuCPGUI [18] mentioned in Section 1.1.2 is a MATLAB toolbox used specifically for parameterizing the Ellipse compressor models in Sections 2.4.1 and 2.5.5. The toolbox comes with a Simulink library containing a Ellipse model compressor block to be used in simulations.

2.7.2 Modelica

Modelica [22] is a object-oriented language for modeling physical systems that have many tool vendors that implements the standard. It can handle large, complex heterogenous systems by connecting blocks for adding friction, inertia or an electric resistor as an example. A big difference compared to Simulink is that Modelica supports acausal modeling. Vehicle Propulsion Library VehProLib [5, 23] is a package where complete vehicle model from driver input to wheel output can be modeled. The model also includes a turbocharger compressor model.

2.7.3 GT-Power

GT-Power is an advanced industry standard simulation software "used by all major engine manufacturers and vehicle OEMs" [24]. It is used to predict engine

performance quantities with the possibility to extend outside of basic models into physical models like in-cylinder, fluid dynamics etc. Gt-Power can also co-simulate directly with Simulink among other programming languages. In [25] GT-Power was used to create both a hot gas stand model and an engine model.

3
Method

In this chapter the implementation method of the proposed model library is presented. The model library implementation consists of three parts. First, the main part is the compressor map which was introduced in Section 3.1. Using the compressor map, the library is initialized, and everything generally needed for model implementation is set up. The second part is the compressor models that are presented in Section 3.2 and constitutes the base model structure for which all parameterized models are built on as well as the parameterization and evaluation methods used by the compressor models. The third and last part is the simulation interface implementation which in presented in Section 3.3.

The choice of programming languages for the library implementation is MATLAB along with the Simulink toolbox as the simulation interface. This means that the library and simulation interface can be kept in the same programming environment. Also the *LiU CPgui* [18] as mentioned in Section 1.1.2 uses the same environment which helped as inspiration for the implementations.

3.1 Compressor map and library initialization

The model library is structured around a given compressor map for which parameterized models can be applied. When initializing the library, the given map is loaded and all expressions and values needed for parameterizing models have to be calculated in preparation. The main data from the compressor map is the corrected rotational speed $\bar{N}_{\mathrm{c}}$, corrected mass flow rate $\bar{m}_{\mathrm{c}}$ and pressure ratio Π_{c} and isentropic efficiency η_{c}. Additionally, the map also includes the compressor impeller diameter d_c as well as the reference temperature $T_{\mathrm{c,ref}}$ and reference pressure $p_{\mathrm{c,ref}}$ used while measuring the map. In some cases the map may also include the compressor inlet temperature T_{01} and pressure p_{01}. If these are not

given, they are set as their individual reference values.

The corrected values are uncorrected using the relationships in (2.1a) and (2.1b). Notice that if the inlet temperature and pressure are set as their reference values, the uncorrected values will simply be the same as the corrected values. With everything set up to this point, the library has the main input data as the compressor rotational speed N_c, mass flow rate $\dot{m}_c$ and pressure ratio Π_c and isentropic efficiency η_c for which all other data will be derived.

3.1.1 Experimental data

To test and evaluate the compressor models over a wider range of application, three compressor maps are used from a database consisting of several hundred compressor maps. The chosen turbocharger compressor maps are used in marine, heavy duty and automobile use to cover a wider range of application. Information about the compressor and its map can be seen in Table 3.1. For marine use, the turbocharger *TCA55* by *MAN Energy Solutions* has been chosen which weigh over 3 tonnes and can deliver just over 10 kW of power. For heavy duty, the Garret turbocharger model *GAR C117* has been chosen. Lastly for automotive applications, the Garret turbocharger model *GAR GT14* has been chosen, which, for example is used in the Kia Stinger 3.3L engine with twin turbochargers.

The *GAR GT14* is the only compressor map that includes specific inlet temperature and pressure.

Model	**TCA55**	**Garret C117**	**Garret GT14**
Data points	62	49	91
Impeller diameter [m]	0.561	0.082	0.041
Maximum mass flow [m^3/s]	12.07	0.3252	0.1017
Maximum pressure ratio [-]	5.05	3.30	2.98
Maximum speed [RPM]	18,724	116,980	230,050

Table 3.1: *Test compressor map data.*

3.1.2 Compressor expressions and dimensionless numbers

The individual compressor models use a variety of different data based on expressions and dimensionless numbers. This data is derived using the already set up compressor rotational speed N_c, mass flow rate $\dot{m}_c$ and pressure ratio Π_c and isentropic efficiency η_c in order to extend the map data. The final extended map data is shown in Table 3.2. Before looking at the compressor expressions, one last preparation is needed to set up the properties of the inlet gas which are the constant pressure specific heat capacity c_p, heat capacity ratio γ, density ρ and

gas constant R. These values are as default set for air as

$$
\begin{aligned}
c_{\mathrm{p}} &= 1005\ [\mathrm{J/(kg\ K)}] \\
\gamma &= 1.4\ [\text{-}] \\
\rho &= 1.23\ [\mathrm{Kg/m^{\,3}}] \\
R &= c_{\mathrm{p}} \left(\frac{\gamma - 1}{\gamma} \right)\ [\mathrm{J/(mol\ \cdot K)}]
\end{aligned}
$$

With the map base data at hand, looking at the physical expressions of the compressor in Section 2.3, the isentropic enthalpy Δh_{is} and isentropic actual enthalpy Δh_{act} can be derived from (2.5) with the use of the isentropic efficiency from the map. Derived from the same equation of the efficiency, the outlet temperature T_{02} and pressure p_{02} is defined in (2.6) and (2.7) and can now also be acquired. With this data at hand, the expression for compressor power P_{c} in (2.2) at the beginning of the section can now also be calculated.

From the dimensionless numbers in Section 2.3.1, are the head coefficient Ψ, flow coefficient Φ and mach number Ma as seen in (2.8a), (2.8b) and (2.8c). The compressor impeller tip speed seen in (2.8d), is a physical expression of the rotational speed N_{c} used in the dimensionless numbers. Similar to the impeller tip speed, the compressor angular speed, measured in radians, is often used as $\omega_{\mathrm{c}} = \frac{2\pi}{60} N_{\mathrm{c}}$.

3.2 Compressor models

Compressor models are, as seen in Sections 2.4 and 2.5, modeled for different relationships between physical compressor expressions and dimensionless numbers. To account for all types of models and the ability to add models with new approaches, a component hierarchy is proposed where the main compressor functions can be divided into smaller components linked in a chain. This approach will also structure the library in a suitable form to be implemented into a block based simulation interface which can be seen later in Section 3.3. The functions will use the base compressor map data as inputs. The isentropic efficiency data will however not be used since it hos got its own component function. Also, the compressor map reference values $T_{\mathrm{c,ref}}$ and $p_{\mathrm{c,ref}}$, inlet temperature T_{01} and pressure p_{01} and impeller diameter d_c will be seen as trivial inputs and ignored in the listed functions.

3.2.1 Model basics and structure

The compressor models need a good defined base from which each individual compressor model can be parameterized. Like Section 3.1 where the compressor expressions and dimensionless numbers are used the extended map data, the

Cathegory	Component	Symbol
Compressor map	Rotational speed	N_c
	Mass flow rate	$\dot{m}_c$
	Pressure ratio	Π_c
	Isentropic efficiency	η_c
	Map reference temperature	$T_{c,ref}$
	Map reference pressure	$p_{c,ref}$
	Inlet temperature (optional)	T_{01}
	Inlet pressure (optional)	p_{01}
	Impeller diameter	d_c
Inlet gas properties	Constant pressure specific heat capacity	c_p
	Heat capacity ratio	γ
	Density	ρ
	Gas constant	R
Dimensionless numbers	Head coefficient	Ψ
	Flow coefficient	Φ
	Mach number	Ma
Physical expression	Angular speed	ω_c
	Impeller tip speed	U_c
	Power	P_c
	Isentropic enthalphy	Δh_{is}
	Isentropic actual enthalphy	Δh_{act}
	Outlet temperature	T_{02}
	Outlet pressure	p_{02}

Table 3.2: *Extended compressor map data.*

same equations can be used to break down the base model structure into smaller function components. In a similar fashion the physical expressions gives the function components $P_c(\dot{m}_c, \Delta h_{is})$ for compressor power, $\eta_c(\Delta h_{is}, \Delta h_{act})$ for isentropic efficiency, $\Delta h_{is}(\Pi_c)$ for isentropic enthalphy and $\Delta h_{act}(T_{01}, T_{02})$ for isentropic actual enthropy with their respective input data.

From the dimensionless numbers in (2.8a), (2.8b) and (2.8c) we get the functions for the head coefficient $\Psi = f(\Delta h_{is}, U_c)$, flow coefficient $\Phi = f(\dot{m}_c, U_c)$ and mach number $Ma = f(U_c)$. The impeller tip speed in (2.8d) gives the function $U_c = f(N_c)$ and similar for the angular speed as $\omega_c = f(N_c)$. Higher up in the hierarchy functions for the mass flow rate and pressure ratio is needed and by again using (2.8a) and (2.8b) we get the functions for mass flow rate $\dot{m}_c = f(\Phi, U_c)$ and pressure ratio $\Pi_c = f(\Delta h_{is})$.

One might notice that naturally it would be impossible to recursively call the functions in a hierarchy without having two functions call on each other, creating an infinite loop. This is where the parameterized model functions come in.

Each model are built on a base model class including all the functions as seen in Table 3.3. It is then up to the individual compressor models to override the functions for which are implemented and parameterized. As an example, the two Karlsen mass flow models in Section 2.4.4 models the flow coefficient Φ as functions of the flow coefficient Ψ and mach number Ma, as $\hat{\Phi} = f(Psi, Ma)$. The Karlsen models are of function type (2.9) that calculates mass flow $\dot{m}_c$ from pressure ratio Π_c and rotational speed N_c. Following the compressor flow model backwards from output to input with the help of the functions in Table 3.3, the head function (2.9) can be broken down into the function hierarchy as seen in (3.1).

$$\dot{m}_c = f(\Phi, U_c) \Rightarrow \begin{cases} \hat{\Phi} = f(\Psi, Ma) \Rightarrow \begin{cases} \Psi = f(\Pi_c) \\ Ma = f(U_c) \end{cases} \Rightarrow U_c = f(N_c) \\ U_c = f(N_c) \end{cases} \tag{3.1}$$

Component	Function
Power	$P_c = f(\dot{m}_c, \Delta h_{is})$
Isentropic efficiency	$\eta_c = f(\Delta h_{is}, \Delta h_{act})$
Head coefficient	$\Psi = f(\Delta h_{is}, U_c)$
Flow coefficient	$\Phi = f(\dot{m}_c, U_c)$
Mach number	$Ma = f(U_c)$
Pressure ratio	$\Pi_c = f(\Delta h_{is})$
Mass flow rate	$\dot{m}_c = f(\Phi, U_c)$
Outlet temperature	$T_{02} = f(\Pi_c, \eta_c)$
Outlet pressure	$p_{02} = f(\Pi_c)$
Isentropic enthalpy	$\Delta h_{is} = f(\Pi_c)$
Isentropic actual enthalpy	$\Delta h_{act} = f(T_{01}, T_{02})$
Angular speed	$\omega_c = f(N_c)$
Impeller tip speed	$U_c = f(N_c)$

Table 3.3: *Compressor model functions.*

3.2.2 Model parameterization

A general solution for parameterizing the compressor models in Sections 2.4 and 2.5 are needed. For user friendliness it is preferred to be as automated as possible and work for any given model. From the MATLAB Optimization Toolbox the function `lsqnonlin` is used to solve the parameterization problems. It solves the curve fitting problem for the minimal solution of x through non-linear least-squares of the form

$$\min_x \|f(x)\|_2^2 = \min_x \left(f_1(x)^2 + f_2(x)^2 + ... + f_n(x)^2 \right) \tag{3.2a}$$

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \tag{3.2b}$$

where $f(x)$ is a function that returns the values $f_i(x), i \in [1, n]$ as a vector in the form and also supports the use of upper and lower bounds.

As previously defined in (2.47), parameterizing using the LSQ method redefines the error function in (3.2b) as simply calculating the error as the difference $y' - y$ between the model prediction and the measured values. Using the measured output data as $y = y_{\text{meas}}$ and the prediction function as $y' = y_{\text{pred}}(param, x_{\text{meas}})$ where $param$ is the parameters to be optimized and x_{meas} are the measured input data gives the LSQ error function

$$f_{LSQ}(x) = \begin{bmatrix} y'_1 - y_1 \\ y'_2 - y_2 \\ \vdots \\ y'_n - y_n \end{bmatrix} = \begin{bmatrix} y_{\text{pred,1}}(param, x_{\text{meas,1}}) - y_{\text{meas,1}} \\ y_{\text{pred,2}}(param, x_{\text{meas,2}}) - y_{\text{meas,2}} \\ \vdots \\ y_{\text{pred,n}}(param, x_{\text{meas,n}}) - y_{\text{meas,n}} \end{bmatrix} \tag{3.3}$$

Parameterizing using the Total Least Squares method requires, unlike for LSQ, to handle an extra error on the function input, called Δx, where $x' = x + \Delta x$ is the new input value. By minimizing the errors for both input and output errors $y' - y$ with the input errors Δx as parameters, the closest orthogonal point in the model function can be obtained. The TLS error sum as seen in (2.48) normalizes the x and y values by their mean values to give a good normalization between different models. In the case of the parameterization error this can be simplified by dividing each dimension by its max value, which gives the new TLS error sum

$$TLS_{param} = \sum_{i=1}^{n} \left[\left(\frac{y'_i - y_i}{y_{max}} \right)^2 + \left(\frac{x'_i - x_i}{x_{max}} \right)^2 \right] \tag{3.4}$$

The prediction function values are defined as $y' = y_{\text{pred}}([param, \Delta x], x_{\text{meas}} + \Delta x)$ where the extra term $\Delta x = x' - x$ are added compared to the LSQ method. Extending (3.3) with the TLS parameterization error sum in (3.4) now gives the TLS error function

$$f_{TLS}(x) = \begin{bmatrix} (y'_1 - y_1)/y_{max} \\ (y'_2 - y_2)/y_{max} \\ \vdots \\ (y'_n - y_n)/y_{max} \\ (x'_1 - x_1)/x_{max} \\ (x'_2 - x_2)/x_{max} \\ \vdots \\ (x'_n - x_n)/x_{max} \end{bmatrix} = \begin{bmatrix} (y_{\text{pred,1}}([param, \Delta x], x_{\text{meas,1}} + \Delta x_1) - y_{\text{meas,1}})/y_{max} \\ (y_{\text{pred,2}}([param, \Delta x], x_{\text{meas,2}} + \Delta x_2) - y_{\text{meas,2}})/y_{max} \\ \vdots \\ (y_{\text{pred,n}}([param, \Delta x], x_{\text{meas,n}} + \Delta x_n) - y_{\text{meas,n}})/y_{max} \\ \Delta x_1/x_{max} \\ \Delta x_2/x_{max} \\ \vdots \\ \Delta x_n/x_{max} \end{bmatrix} \tag{3.5}$$

Parameter initialization

Initial parameters can be very important for good curve fitting. This is especially important when using the TLS method, since the optimization problem also includes the input errors which can lead to the parameterization method not converging. Meanwhile the LSQ method often works just fine by using simple initial parameters such as zeros.
With a user-friendly approach, the parameterization should be kept as automated and independent to compressor maps of specific characteristics as possible. By starting with a very simple parameter initialization, for example zeros as mentioned, the LSQ method can be used to obtain roughly optimized initial parameters. The new optimized initial parameters should now be close to the best solution and the TLS parameterization can now act as fine tuning parameters for the curve fitting, with more weight on optimizing the input errors.

Algorithm

The algorithm of the parameterizing method from initial parameters to optimized model parameters are presented in Figure 3.1 as a flow chart. The different steps are as followed:

1. **Start**
   The parameterization method starts.

2. **Parameter initialization**
   Set default initial parameters. For example, zeros.

3. **Optimize initial parameters?**
   If yes, continue. If no, go to the TLS method at 5.

4. **Parameterize using LSQ**
   Optimize the initial parameters by parameterizing using the LSQ method. The process continues to optimize the parameters until a minimizing solution is found or if it does not converge. The optimized initial parameters are obtained from the output parameters.

5. **Use TLS method?**
   If yes, continue. If no, go to the end of the algorithm at 7.

6. **Parameterize using TLS**
   Optimize the model parameters and input error parameters using the TLS method. The process continues to optimize the parameters until a minimized solution is found or if it does not converge. The optimized model parameters are obtained from the output parameters. The optimized input error parameters are discarded.

7. **End**
   The parameterization method ends. The optimized model parameters are the output values from the method.

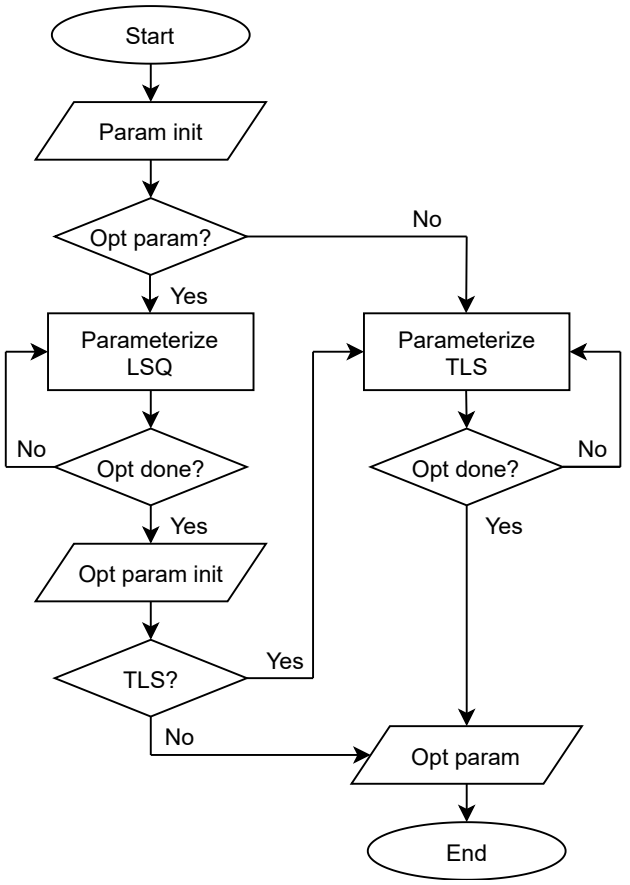


Figure 3.1: *Flow chart of parameterizing method algorithm.*

3.2.3 Model evaluation

In this thesis the same model evaluation methods as in study [2] is used as a base measure for replicability purposes. The methods for evaluating model agreement consist of the coefficient of determination R^2, corrected coefficient of determination R_C^2, relative deviation RD mean absolute deviation MAD and percentage of data points within the error bounds PEB for ±5 and ±10 percentage. The definitions as in [1, 3] are

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_{\text{meas,i}} - y_{\text{pred,i}}\right)^2}{\sum_{i=1}^{n} \left(y_{\text{meas,i}} - y_{\text{mean}}\right)^2} \tag{3.6}$$

$$R_c^2 = 1 - \frac{(1 - R^2)(n-1)}{n - m - 1} \tag{3.7}$$

$$RD_i = \frac{y_{\text{pred,i}} - y_{\text{meas,i}}}{y_{\text{meas,i}}} \tag{3.8}$$

$$MAD = \frac{1}{n}\sum_{i=1}^{n} |RD_i| \tag{3.9}$$

$$PEB_{\pm k\%} = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1, & if\ |RD_i| \leq k\% \\ 0, & otherwise \end{cases} \tag{3.10}$$

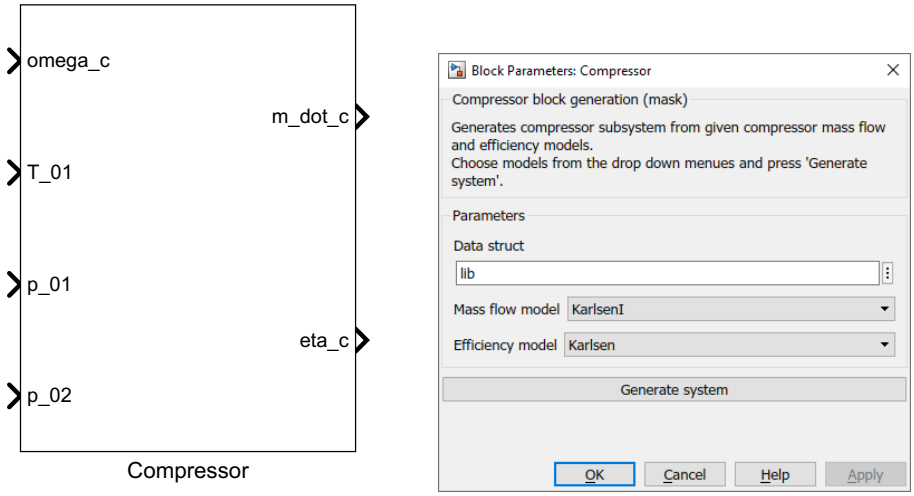
where i is the measured data point index, RD_i the relative deviation, $y_{\text{meas,i}}$ the measured data point, $y_{\text{pred,i}}$ the predicted model value, n is the number of measured data points and m the number of predictors (model parameters). These methods only calculate errors of the predicted model output value. In Section 2.5.5 the difference between the LSQ and TLS methods is mentioned. In this case the methods shown in (3.7), (3.9) and (3.10) works in the same fashion as LSQ, which could potentially lead to great errors.

In addition to the methods above, the Total Least Square method which is presented in (2.48), is used as an alternative measure for model evaluation. Just as in the model parameterization in Section 3.2.2, the problem in (3.2a) is solved using the TLS parameterization function in (3.5), but this time by using the already obtained model parameters and only parameterizing the model input errors Δx_i. With the parameterized model input errors Δx_i the TLS sum in (2.48) can be modified using the derived TLS expressions in (3.5) which gives the final TLS error sum

$$TLS = \sum_{i=1}^{N} \left((y_{pred,i} - y_{meas,i})^2 + \epsilon(\Delta x_i)^2 \right), \quad \epsilon = \left(\frac{mean(x_{meas})}{mean(y_{meas})} \right)^2 \tag{3.11}$$

3.3 Simulation interface

With the function component hierarchy approach of the model library structured as in Section 3.2.1 in mind, the block based environment of the Simulink toolbox of MATLAB was proposed. By assigning each component functionality to a block, they can be connected in a chain from input to output. However, with all models' differences and the aim for user friendliness, all blocks should be generated automatically. Simulink allows programmatically manipulation of its environment, making it possible to create a subsystem for a specified model. In Figure 3.2 the simulation interface can be seen with the compressor block and menu for its subsystem generation. The block and menu are done automatically by running a script. The block inputs are set as compressor angular speed ω_c, inlet temperature T_{01} and inlet/outlet pressure p_{01}, p_{02}. The outputs are set as compressor mass flow rate $\dot{m}_c$ and isentropic efficiency η_c. The subsystem generation menu is made by creating a block mask. By choosing mass flow and efficiency models in the menu and pressing *Generate system* the automatic process of creating the



(a) Simulink compressor block. *(b) Subsystem generator menu.*

Figure 3.2: *Model library simulation interface.*

subsystem is run, which is illustrated as the flow chart in Figure 3.3. Initially the input and output signal names of the compressor subsystem are stored in lists as sources and targets, with connecting lines between blocks in mind. A while loop is run if the target list isn't empty.

For every target entry, a MATLAB function block is created. The function code for said component name and active model is read and inserted in the new block and its input/output signal names are stored in the target/source lists, flipped since inputs will act as line targets and vice versa. Finally, any block source/target signals are connected if possible. The component for the efficiency function η_c is prioritized for last, as it might use other model function than the mass flow model.

When the target list is empty the system should be complete, and the loop stops.

3.3.1 Algorithm

The algorithm of the compressor model system generation algorithm is presented as a flow chart in Figure 3.3. The steps of the algorithm is as followed:

1. **Start**
   The compressor model system generation is started for the given mass flow and isentropic efficiency models.

2. **Find compressor block signal I/O**
   The inputs and outputs of the main compressor block is found and added to the list of sources and targets for block connections.

3. **Signal target list empty?**
   Is the list of available signal targets empty? If yes, go to 7.

4. **Add signal block**
   The first entry of the signal target list is popped. A MATLAB function block is created and the function code for the current signal component is added to the block.

5. **Find signal block inputs**
   The inputs of the added block is added as to the signal target list.

6. **Connect block**
   Connect the block output to its target. Connect block inputs signal (targets) if signal sources are available.

7. **End**
   Compressor model system generation is completed.

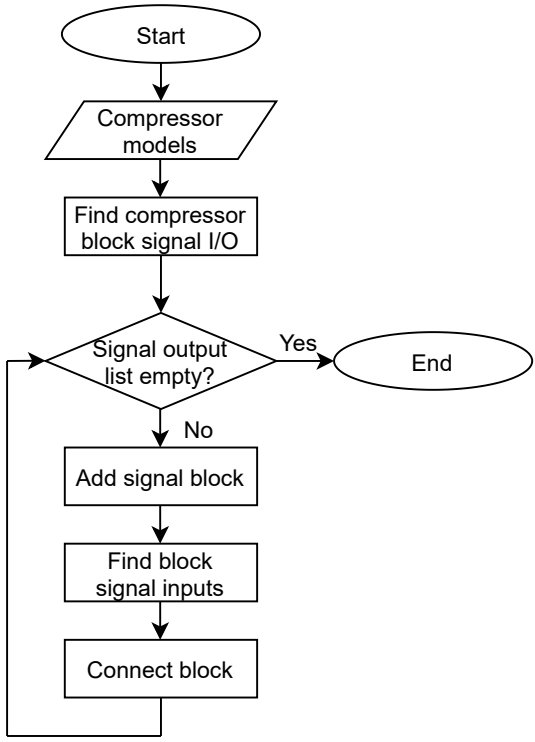


Figure 3.3: *Simulink compressor subsystem generation flow chart.*

4

Results

In this chapter the results from evaluating the implemented models are presented. The models in Sections 2.4 and 2.5 are implemented and
The *Ellipse* model was never fully implemented in the library due to the importance and dependence on the initial guess in the tuning process for the fitting. Instead the *LiU CPgui* [18] was used with the chosen compressor maps to extract plots and optimized parameters due to the support for adjusting the initial guess that it provides to the user. The functionality from the toolbox generated the model parameters, then the model was evaluated inside the model library developed in this thesis. Every parameterized model has been run with default settings without any manual adjustments.

4.1 Compressor mass flow model results

Below are the results of the compressor mass flow models used for each compressor map. They are evaluated using graphical plots and the methods mentioned in Section 3.2.3. The R_c^2 score for the *Guan Cong* model has been ignored since the number of parameters are greater than the number of measured data points, resulting in R_c^2 values over 1.

4.1.1 TCA55

Model evaluation of compressor mass flow using the *TCA55* compressor map can be seen in Figures 4.1 and 4.2 and Table 4.1.

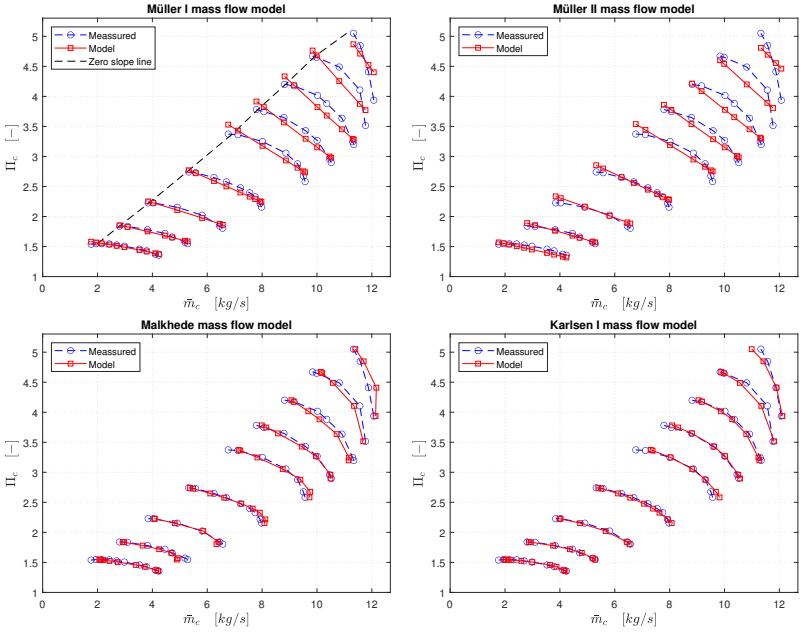


Figure 4.1: *TCA55 compressor map mass flow models.*

Model	Param	TLS [-]	Rc2 [-]	MAD [%]	$PEB_{5\%}$ [%]	$PEB_{10\%}$ [%]
Müller I	9	0.8022	0.9861	2.5567	88.7097	98.3871
Müller II	4	1.0184	0.9839	3.0274	85.4839	98.3871
Malkhede	10	0.6331	0.9956	2.9347	85.4839	96.7742
Karlsen I	10	0.2066	0.9967	2.2551	91.9355	96.7742
Karlsen II	12	0.1565	0.9972	2.2664	88.7097	96.7742
Guan Cong	72	0.0109	-	1.539	95.1613	96.7742
ZSLM	8	1.4967	0.9844	5.2655	62.9032	87.0968
Ellipse	15	0.0249	0.9831	1.6347	90.3226	95.1613

Table 4.1: *TCA55 compressor map mass flow model evaluation.*

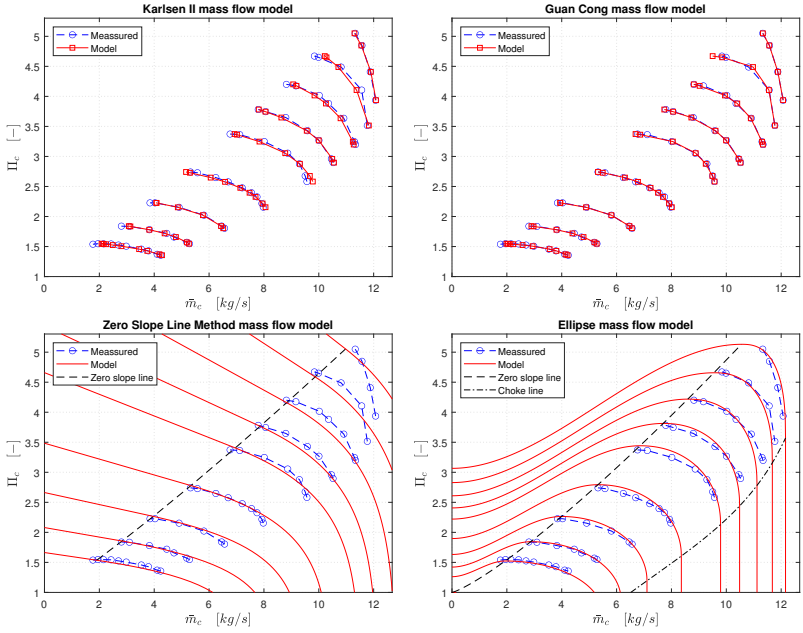


Figure 4.2: *TCA55 compressor map mass flow models.*

4.1.2 Garret C117

Model evaluation of compressor mass flow using the *TCA55* compressor map can be seen in Figures 4.3 and 4.4 and Table 4.2.

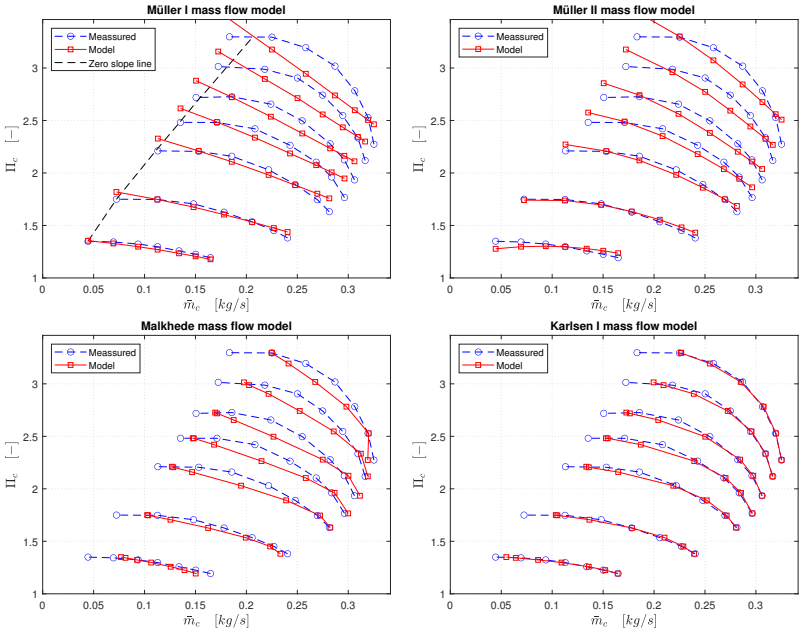


Figure 4.3: *GAR C117 compressor map mass flow models.*

Model	Param	TLS [-]	Rc2 [-]	MAD [%]	$PEB_{5\%}$ [%]	$PEB_{10\%}$ [%]
Müller I	9	0.604	0.9568	3.5857	71.4286	97.9592
Müller II	4	0.3864	0.9755	2.9164	79.5918	97.9592
Malkhede	10	0.0091	0.9184	9.555	40.8163	65.3061
Karlsen I	10	0.0015	0.964	5.6091	65.3061	79.5918
Karlsen II	12	0.0015	0.9646	5.6912	67.3469	81.6327
Guan Cong	56	0.0003	-	2.5169	83.6735	91.8367
ZSLM	7	0.0028	0.9498	8.6128	42.8571	73.4694
Ellipse	15	0.0037	0.9994	0.4571	100	100

Table 4.2: *GAR C117 compressor map mass flow model evaluation.*

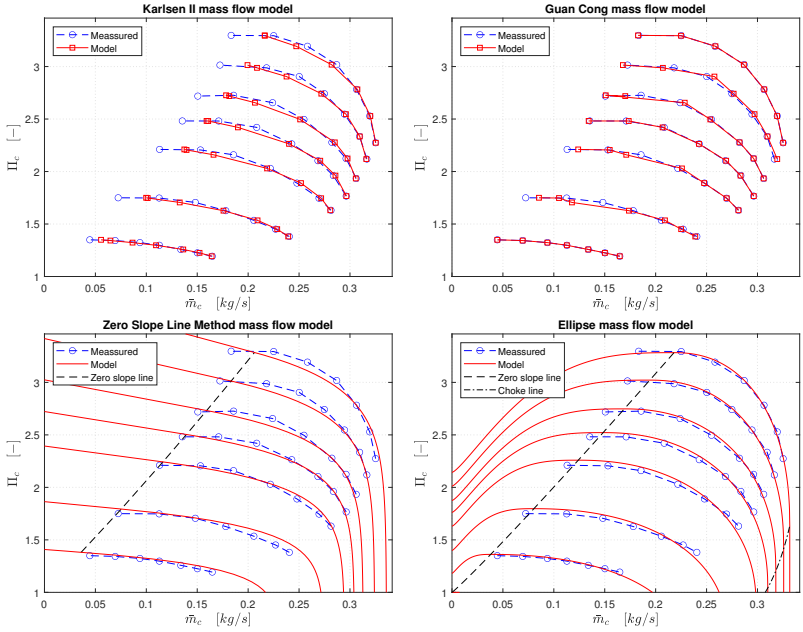


Figure 4.4: *GAR C117 compressor map mass flow models.*

4.1.3 Garret GT14

Model evaluation of compressor mass flow using the *TCA55* compressor map can be seen in Figures 4.5 and 4.6 and Table 4.3. The *Guan Cong* model has a singularity so its scores has been ignored.

Model	Param	TLS [-]	Rc2 [-]	MAD [%]	$PEB_{5\%}$ [%]	$PEB_{10\%}$ [%]
Müller I	9	3.7947	0.8838	7.5941	53.8462	73.6264
Müller II	4	3.3558	0.9032	7.1177	53.8462	73.6264
Malkhede	10	0.0093	0.9169	18.2155	18.6813	40.6593
Karlsen I	10	0.0023	0.9587	8.4008	52.7473	73.6264
Karlsen II	12	0.0024	0.9591	8.0592	53.8462	74.7253
Guan Cong	112	-	-	-	-	-
ZSLM	7	0.0183	0.8309	43.3159	18.6813	41.7582
Ellipse	15	0.0181	0.9239	3.1445	94.5055	95.6044

Table 4.3: *GAR GT14 compressor map mass flow model evaluation.*

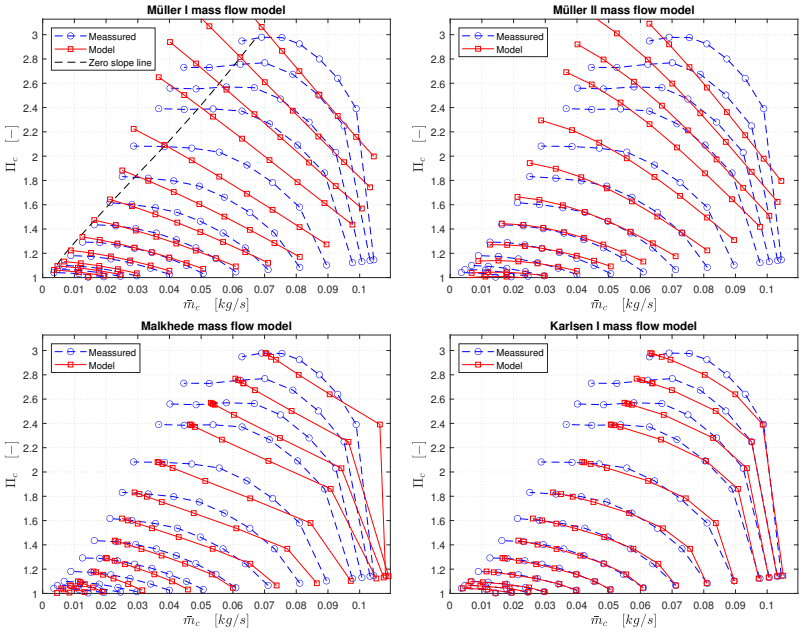


Figure 4.5: *GAR GT14 compressor map mass flow models.*

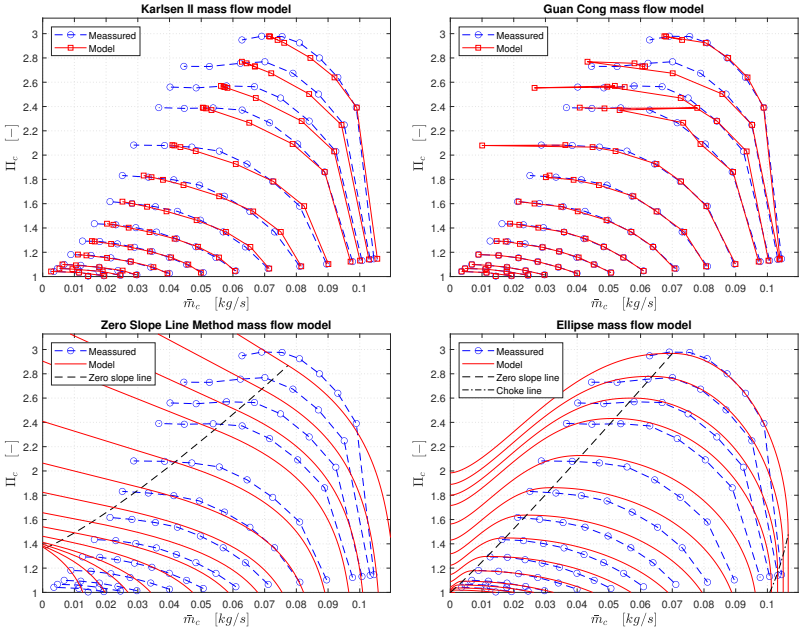


Figure 4.6: *GAR GT14 compressor map mass flow models.*

4.2 Compressor isentropic efficiency model results

Below are the results of the compressor isentropic efficiency models used for each compressor map. The models are evaluated using graphical plots and the methods mentioned in Section 3.2.3.

4.2.1 TCA55

Model evaluation of compressor isentropic efficiency using the *TCA55* compressor map can be seen in Figure 4.7 and Table 4.4.

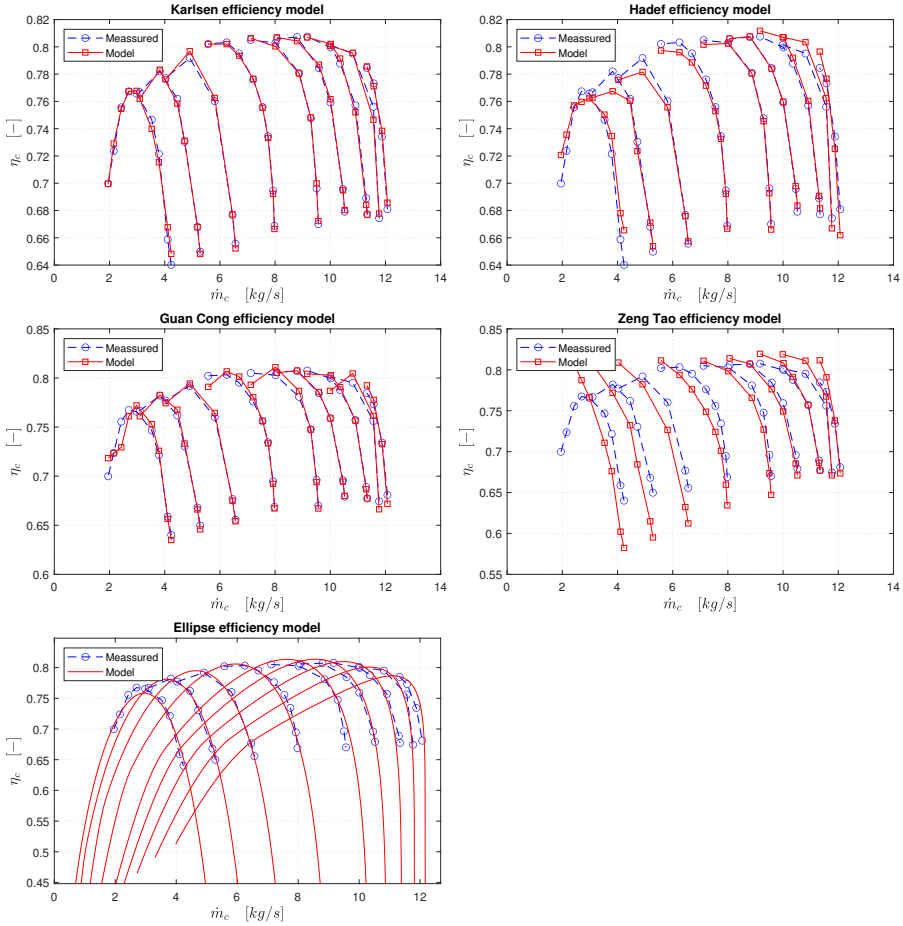


Figure 4.7: *TCA55 compressor map isentropic efficiency models.*

Model	Param	TLS [-]	Rc2 [-]	MAD [%]	$PEB_{5\%}$ [%]	$PEB_{10\%}$ [%]
Karlsen	21	0.0006	0.9925	0.3525	100	100
Hadef	4	0.0034	0.9736	0.7919	100	100
Guan Cong	18	0.0024	0.9742	0.6174	100	100
Zeng Tao	3	0.0585	0.559	3.4145	74.0741	96.2963
Ellipse	6	0.0014	0.9888	0.481	100	100

Table 4.4: *TCA55 compressor map isentropic efficiency model evaluation.*

4.2.2 GAR C117

Model evaluation of compressor isentropic efficiency using the *Garret C117* compressor map can be seen in Figure 4.8 and Table 4.5. The Guan Cong model has an asymptotical behavior, and its evaluation scores has been ignored.

Model	Param	TLS [-]	Rc2 [-]	MAD [%]	$PEB_{5\%}$ [%]	$PEB_{10\%}$ [%]
Karlsen	21	0.0003	0.9971	0.2642	100	100
Hadef	4	0.0099	0.935	1.3787	97.9592	97.9592
Guan Cong	14	-	-	-	-	-
Zeng Tao	3	0.0091	0.9411	1.4534	97.9592	100
Ellipse	6	0.0052	0.9607	1.318	97.9592	100

Table 4.5: *GAR C117 compressor map isentropic efficiency model evaluation.*

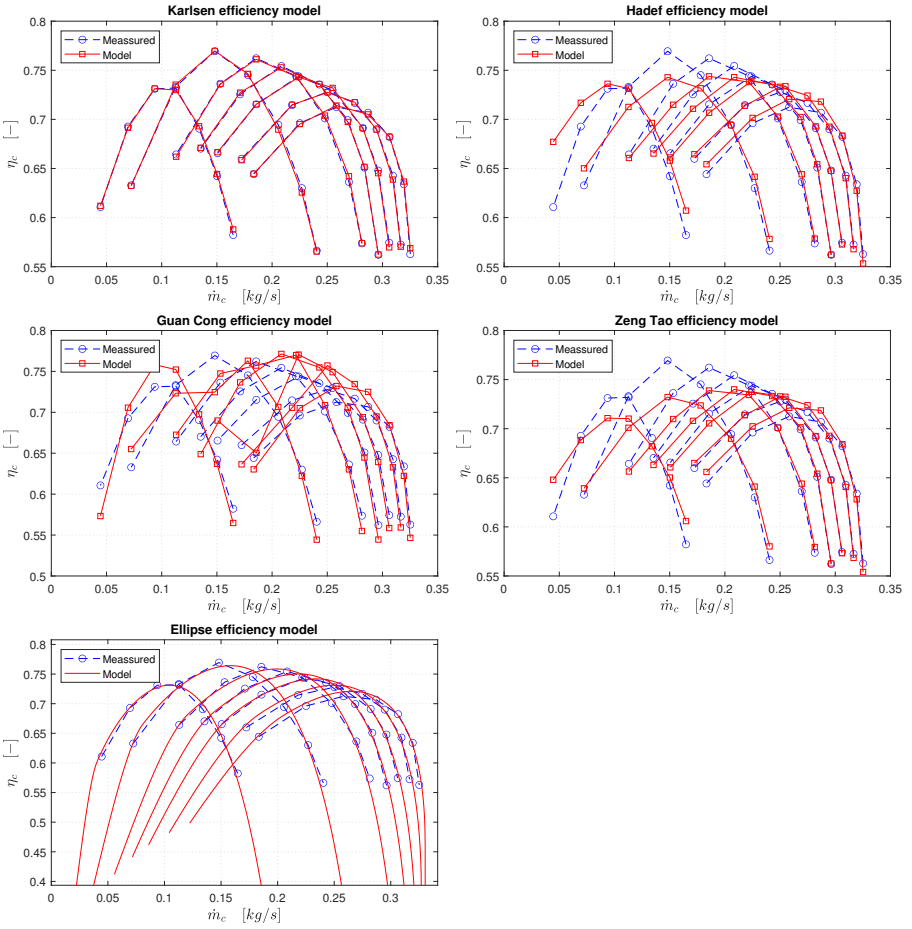


Figure 4.8: *GAR C117 compressor map isentropic efficiency models.*

4.2.3 GAR GT14

Model evaluation of compressor isentropic efficiency using the *Garret GT14* compressor map can be seen in Figure 4.8 and Table 4.5. The Guan Cong model has an asymptotical behavior, and its evaluation scores has been ignored.

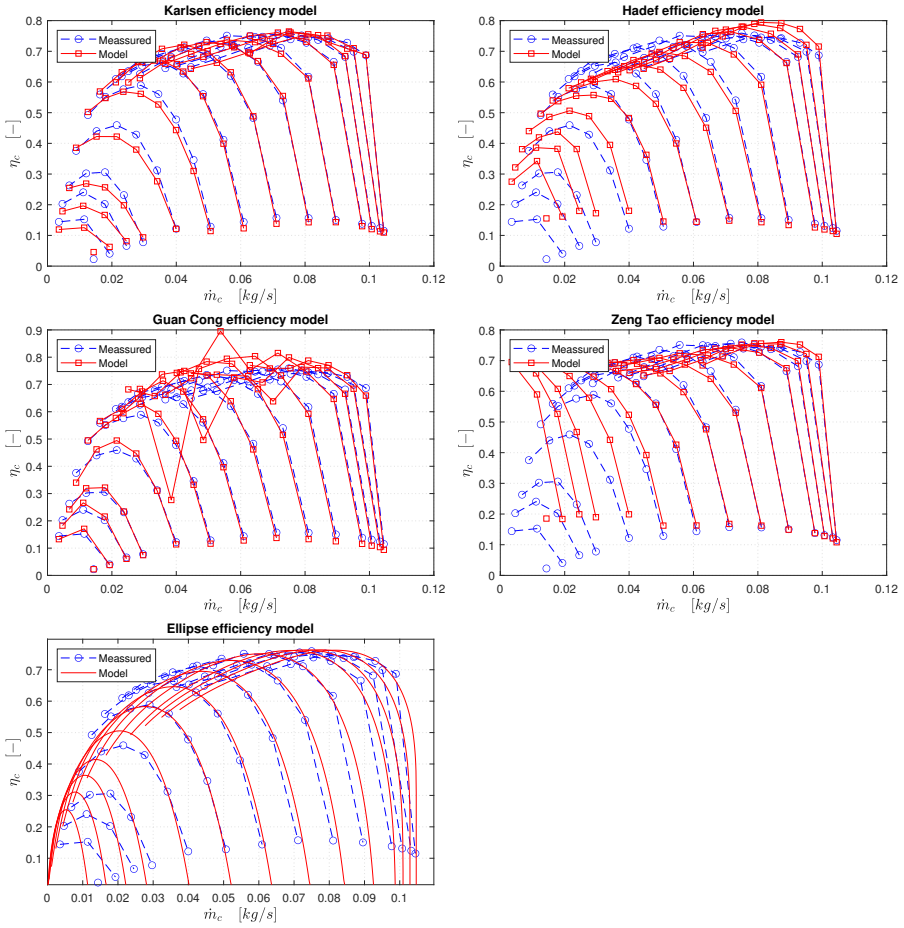


Figure 4.9: *GAR GT14 compressor map isentropic efficiency models.*

Model	Param	TLS [-]	Rc2 [-]	MAD [%]	$PEB_{5\%}$ [%]	$PEB_{10\%}$ [%]
Karlsen	21	0.0306	0.992	6.1419	71.4286	80.2198
Hadef	4	0.3243	0.9316	24.6869	45.0549	76.9231
Guan Cong	28	-	-	-	-	-
Zeng Tao	3	1.9959	0.5837	42.401	60.4396	69.2308
Ellipse	6	0.4172	0.7428	36.574	53.8462	64.8352

Table 4.6: *GAR GT14 compressor map isentropic efficiency model evaluation.*

5
Discussion

In this chapter the implementation of the model library and results will be discussed.

5.1 Model library implementation

The class object structure of the models proved to be a simple system, where the parameterized models derive from the base model class and overrides the modeled components. Being a derived sub class also allow for specialized functions and variables to be stored and used along with great flexibility, without causing any problems with other models. Implementing new models using the function hierarchy of the base model class proved to be very successful, where a not to advanced model could implemented within an hour. The Guan Cong is problematic to parameterize, using 8 parameters for every speed line, which led to the minimization problem having lots of local minima. A simple fix was to start parameterizing the lowest speed line with the simplest, most linear characteristics, then use its optimized model parameters as initial parameters for the next speed line, and so on, giving more stable solutions. This was not a perfect solution, but did give a higher success rate of parameterizing the models. The Müller I mass flow model has a very linear behavior similar to the Müller II model, however looking at the model definition in section 2.4.5 and results from the study [15] it was expected to have a more nonlinear behavior.

The results in chapter 4 were acquired by running the model library and parameterizing of the implemented models using default settings, with the only exception being the input for different compressor maps. The computation of this process took a few seconds using a fairly new *AMD Ryzen 7 3700X* processor. The Ellipse model was never fully implemented due to the importance and

dependence on the initial guess in the tuning process for the fitting. Instead the *LiU CPgui* mentioned in section 1.1.2 was used to extract the model parameters and functions to be plotted and evaluated using the same methods as the model library.

5.2 Compressor mass flow rate models

The models in figures 4.1 and 4.2 have an overall good fit of the TCA55 map, with the linear behavior of the Müller models fit worse as the compressor map gets increased curved at higher rotational speeds. The ZSLM model has an even characteristic, which seem to have an ok fit at first glance but has a very hard time capturing the characteristics where the speed lines drop off into the choke region. The Guan Cong models has the absolute best fit, followed by the Ellipse and both Karlsen models. Looking at the evaluation scores for the TCA55 map in table 4.1 the Guan Cong model has the best overall scores, especially in the TLS score. However, the model has as many as 72 parameters. The Ellipse model has the next best TLS score and among the best in the rest of the scores. The ZSLM model has weak scores which is reflected in the visual fit in figure 4.2. Interestingly the Müller models has the best $PEB_{\pm 10}$ score while all other scores and the visual fit is among the worst.

For the GAR C117 map, the visual fit in figures 4.3 and 4.4 are again worst for the Müller models, but also a similar bad fit for the Malkhede model. The Guan Cong and Ellipse models has the best visual fits, followed by the Karlsen and ZSLM models. The Guan Cong model does however show some weird behavior of smaller "knees" in some speed lines. The ZSLM model manages to capture the characteristics really well this time with again an even pattern, with the small exception for the edge of the lower speed lines, which could be caused by the lack of data points in that region. In the evaluation scores in table 4.2the Ellipse model has the by far best overall scores. The Karlsen models has a better TLS score, but lacks in all other scores. The Müller models again shows a really high $PEB_{\pm 10}$ score, while having the worst TLS score. Guan Cong has the absolute best TLS score but uses 56 parameters.

The GAR GT14 map is by far the hardest to get a good model parameterization as can be seen visually in figures 4.5 and 4.6. The Guan Cong model shows an singular behavior on higher speed lines and its scores has been ignored. This behavior might have something to do with the last measured data point of the higher speed lines dropping off very fast, causing problems with the parameterization. The Müller models has the worst fit where their linear form cannot match the very nonlinear characteristics of the GAR GT14 map. The ZSLM, Ellipse and Malkhede models The *ZSLM* model does not get a good curve fit and the first speed lines of the ZSLM model even converge into a single point. The Karlsen models have the best fit and models the drop towards the choke region very good and only lacks a little at the highest pressure ratio values. Looking at the evalua-

tion scores in table 4.3 the Ellipse model has the overall best scores, especially in the MAD, $PEB_{\pm 5}$ and $PEB_{\pm 10}$ scores, even though it visually has some problems capturing the difference in speed line characteristics. Except the Ellipse model, the Karlsen models has the best scores, and has the absolute best TLS score. Interestingly the scores for the Karlsen and Müller models are very similar, except for TLS, where the Karlsen models have a significant better score, hinting at the strength of using TLS.

5.3 Compressor isentropic efficiency models

The curve fitting of the TCA55 map as seen in figure 4.7 looks good overall, with the exception of the Zeng Tao model which goes off track at lower compressor mass flow rates. The evaluation scores in table 4.4 has the $PEB_{\pm 5}$ and $PEB_{\pm 10}$ scores of 100 for all models except Zeng Tao, which seem reasonable. The other models all have good scores, with the Karlsen and Ellipse models as the best. The Karlsen model does however have a much more complex model of 21 parameters.

As seen in figure 4.8 the Karlsen and Ellipse models manage to parameterize the GAR C117 map really well. The Guan Cong model has a couple of speed lines which looks to have an singular behavior which is not good. The Hadef and Zeng Tao models has a hard time getting a good general fit for the difference in characteristics between the speed lines. The evaluation scores in table 4.5 reflects the visual results really good. Here the Karlsen and Ellipse models has the best scores and Guan Cong with its singular behavior has the worst. The Hadef and Zeng Tao models still get decent scores, and by using only 3 and 4 parameters, they could be a great choice for compressor maps with the right characteristics, befitting to the models.

The GAR GT14 compressor map seen in figure 4.8 has a very special characteristic where it has a sudden drop in efficiency at lower mass flow rates which has proved to be a challenge to parameterize. The Guan Cong model handles these characteristics the best since it parameterizes the speed lines independently, but unfortunately a few speed lines gets an singular behavior. The Karlsen model has the best curve fitting followed by the Ellipse and Hadef models with an overall good fit. In table 4.5 the evaluation scores again reflects the visual evaluation pretty good. The Karlsen model has the best evaluation scores but with 21 parameters. The Zang Tao model has a surprising result where the predicted model efficiency increases at lower mass flow, while the measured data decreases. It is speculated if the extra low mass flow of the map cause problems with division of the model equation (2.42) and/or floating point errors for values near zero.

5.4 Simulation interface

The simulation interface was to be tested and evaluated in an engine model used in the LiU course *Modelling and Control of Engines and Drivelines*, where the

objectives was to implement and verify turbocharger models. The compressor block in Figure 3.2 replaced the existing compressor block. The choice of generated system was once again the example of the Karlsen mass flow models for which function hierarchy can be seen in (3.1). The Karlsen mass flow model has its own parameterized function for the flow coefficient as $\hat{\Phi} = f(Psi, Ma)$. The Karlsen model was further extended by its isentropic efficiency model from Section 2.4.4 as the function $\hat{\eta_c} = f(\dot{m}_c, \Pi_c)$.

The simulation interface uses the same implemented functions straight from the model library and encountered problems where the sudden changes of integrator start up in the simulation could put these functions in situations where impossible values where calculated, such as negative efficiency and pressure. Saturations were therefore needed to be put in the generated system to avoid bad negative and complex signal values. The problem with the library functions in the simulation environments was fixed with temporary saturation fixes, and was able to execute the simulation successfully. The results using the temporary fixes did however show weird behavior, and was not deemed good enough to be presented.

6

Conclusions

In this chapter the conclusion of the thesis is presented, as well as proposing future work. The model library is implemented with an automatic user-friendly approach in mind. The physical properties and behavior of the compressor is broken down into components functions in a hierarchy, where each component function is responsible of calculating its own part recursively. Each model can be parameterized and built with the component function hierarchy as a base where the parameterized components replace the base feature. The Total Least Squares method proved to be a great tool for model parameterization, where the naturally nonlinear behavior of the compressor can be easier captured, by adding one or more model input errors, increasing the dimensions of the model error. The method is however more dependent of good initial parameters compared to a more normal Least Squares method. The problem was countered by using simple initial parameters and utilizing a LSQ solver to optimize the parameters in preparation for the TLS method, as mentioned in section 3.2.2.

The TLS method proved to be an interesting method for evaluating the performance of the compressor models. In table 4.3 the evaluation scores of the Müller and Karlsen models are very similar, with the exception for the huge difference in the TLS score. This result seem much more reasonable when looking at the visual results in figures 4.5 and 4.6. The Ellipse and Karlsen mass flow models have the best overall evaluation results. The Ellipse mass flow model does however have the strength of specific models for the surge and choke regions outside of the ordinary operational region. The Guan Cong models could really be seen as a double edge sword. Since the models for mass flow has 8 parameters per speed line, it will have a great total number of parameters which would need a compressor map with many measured data to get a good fit. Thanks to the parameters, the model has a very volatile behavior where it might get a good fit

as in figure 4.2, or an asymptotical behavior as in figure 4.6, where the function goes towards infinity. A big problem with the mass flow model is that parameterizing a single speed line with 8 parameters will have lots of different solutions, and thus are very dependent on good initial parameters. The efficiency model also shows problems with parameterization in figures 4.8 and 4.9, while having 2 parameters per speed line. The Ellipse and Karlsen isentropic efficiency models also have the best performance, where the Karlsen model are slightly better, especially for the GAR GT14 compressor map. The model does however have 21 model parameters and might have the risk of overfitting the model. The Hadef and Zeng Tao isentropic efficiency models prove that they can be a good choice with their few 4 and 3 model parameters, provided they are used with compressor maps that suit the models behavior.

With the results of the Müller and Ellipse models in thesis as an example, the study [2] and its surprising results seen in figure 1.1 of said models, seem highly unlikely and unreliable. In the case of the study, the evaluation scores which points at the Müller II model having a better curve fit than the Ellipse model, while at the same time having the shown visual results, should be enough to question if the results were correct. The results of this thesis does though have some interesting results where the Müller models get decent evaluation scores, as in table 4.2 except for the TLS score, while having a horrible curve fit in figure 4.3. This might indicate that the R_c^2, MAD and PEB evaluation methods can be flawed for evaluating models like Müller. At the same time there are also cases where the TLS method seem to be a little flawed, as in table 4.1 where the TLS score of the Karlsen models are a lot lower than the Ellipse model, while they seem to have a slightly better curve fit in figures 4.1 and 4.2. With all the results in mind the best method of evaluating models seem to be a combined overview of all the evaluation scores, as well as the visual plots. Interestingly the Müller models has the best $PEB_{\pm 10}$ scores in table 4.1, while all other scores and the visual fit in figure 4.1 is among the worst.

The simulation interface is implemented as a stand-alone Simulink block module along the model library. The interface uses the model library and the parameterized models to generate a compressor system from the component function hierarchy. By using built in programmatically functionalities in Simulink, the system and blocks can be added and modified automatically through code. The compressor model, in this example Karlsen, can be generated as the hierarchy seen in equation (3.1) with the extension of the isentropic efficiency model in Section 2.4.4. The simulation interface does however lack any upper- or lower bounds of its signals, and temporary saturations were needed to be put in manually to get the simulation to execute. This problem needs to be addressed before any proper results can be achieved.

The three questions from section 1.2 are revisited and answered:

1. **Can a compressor model library with automatic parameterization be implemented?**

Yes. The implemented model library handles compressor models of both mass flow and isentropic efficiency in a wide variety. The compressor models can be parameterized automatically without any manual input with only a very few exceptions as the problematic Guan Cong model. The model library follows a user-friendly approach allowing for flexible implementations of new models.

2. **Is Total Least Squares a viable evaluation method for compressor models and can the method be used to reevaluate the results in the study [2]?**
Yes. TLS is firstly proven to be a great method of calculating model errors of nonlinear functions for parameterization, where an error of the function input is considered. Secondly the TLS method is shown to be a great alternative for model evaluation where normal model prediction errors in one dimension might be flawed. This was shown for the study [2] where the visually worst model fit of the Müller II model as in figure 1.1 could get a decent evaluation score, while TLS score in this thesis points at a more realistic result. The results of the study seem very unreliable and should need a reevaluation, with a better evaluation method, such as TLS.

3. **Can the parameterized library models be implemented into a simulation interface?**
Yes. While the evaluation of the simulation was hastily put together and with the extra help of signal saturation, the results still shows that the system generated from the parameterized compressor models of the model library, can be run in a simulation environment.

6.1 Future work

Below are some suggestion of future work:

- The Ellipse model isn't fully implemented in the library due to the importance and dependence on the initial guess in the tuning process for the fitting. With the very good results of the model, it would be very good to fully implement it.

- The scores of the Ellipse model depend on the initial guess of the parameters, and in the present work the initial guesses that are provided out of the box from the LiU CPgui. Another extension would be to perform manual adjustments to the initial guesses using the support that the GUI from the library provides for a user, to see if the state of the art tool provides any added value.

- Seeing how volatile it is parameterizing the Guan Cong models, it might be interesting to investigate possibilities of using constrains of the model output. For example that the model compressor mass flow output $\dot{m}_c$ has to strictly grow or be equal as the pressure ratio falls in the operational region, simply put as $\bar{m}_{c,i} \leq \bar{m}_{c,j}$ *when* $\Pi_{c,i} > \Pi_{c,j}$. Another solution might be to

interpolate the data points of the compressor map to get more data for the parameterization.

- The simulation interface was only tested briefly because of lack of time and also showed that some extra steps need to be takes, such as avoiding bad negative complex signal values. As of right now, the interface is only implemented for models calculating mass flow $\dot{m}_c$ from pressure ratios Π_c as in equation (2.9) which was the simpler case, and should be implemented for models that are the other way around. The final step should be trying out the simulation interface for simulation of a real-time turbocharged engine.

- The measurement data in this thesis was chosen from a database of hundreds of turbocharger compressors maps. To get a more general evaluation, testing should be carried out for the complete set of maps. As mentioned in the introduction chapter 1 there are also lots of other applications where different types of compressor characteristics are used. Therefore, testing the models using measured data maps from different applications would be interesting. Since the models varies a lot in number of parameters, testing the computational time needed for parameterization might be interesting. However, the probably most interesting would be the computation of the already parameterized models to evaluate how well they could handle real-time simulation of high update rates.

- In the evaluation results in chapter 4 all plots shows the predicted model values, with exception for the ZSLM and Ellipse models where the continuous function is shown. The reason being that the models have specific modeled surge regions. With extrapolation being an important topic of compressor models, it would be interesting to investigate the outer regions of all the presented models.

Bibliography

[1] X. Fang, W. Chen, Z. Zhou, and Y. Xu, "Empirical models for efficiency and mass flow rate of centrifugal compressors," *International Journal of Refridgeration*, vol. 41, pp. 190–199, 2014. doi: 10.1016/j.ijrefrig.2014.03.005.

[2] H. Shen, C. Zhang, J. Zhang, B. Yang, and B. Jia, "Applicable and comparative research of compressor mass flow rate and isentropic efficiency empirical models to marine large-scale compressor," *Energies*, vol. 13, no. 1, p. 47, 2019. doi: 10.3390/en13010047.

[3] B. Yang, X. Fang, L. Zhang, F. Zhuang, M. Bi, C. Chen, G. Lic, and X. Wang, "Applicability of empirical models of isentropic efficiency and mass flow rate of dynamic compressors to jet engines," *Progress in Aerospace Science*, vol. 106, pp. 32–42, 2019. doi: 10.1016/j.paerosci.2019.01.007.

[4] M. Antivachis, F. Dietz, C. Zwyssig, D. Bortis, and J. W. Kolar1, "Novel high-speed turbo compressor with integrated inverter for fuel cell air supply," *Boosting and Air Handling Technologies for Advanced Power Systems*, 2021. doi: 10.3389/fmech.2020.612301.

[5] L. Eriksson, V. Nezhadali, , and C. Andersson, "Compressor flow extrapolation and library design for the modelica vehicle propulsion library - vehprolib," tech. rep., Linkoping University, 2016. doi: 10.4271/2016-01-1037.

[6] D. N. Malkhede, B. Seth, and H. C. Dhariwal, "Mean value model and control of a marine turbocharged diesel engine," tech. rep., Department of Mech. Engg., Govt. College of Engineering, 2005. doi: 10.4271/2005-01-3889.

[7] A. T. Karlsen, "On modeling of a ship propulsion system for control purposes," Master's thesis, Norwegian University of Science and Technology, 2012. doi: 11250/260722.

[8] C. Guan, G. Theotokatos, P. Zhou, and H. Chen, "Computational investigation of a large containership propulsion engine operation at slow steaming conditions," *Applied Energy*, vol. 130, pp. 370–383, 2014. doi: 10.1016/j.apenergy.2014.05.063.

[9] J.-P. Jensen, A. Kristensen, S. Sorenson, N. Houbak, and E. Hendricks, "Mean value modeling of a small turbocharged diesel engine," tech. rep., Lab. for Energetics Technical University of Denmark, 1991. doi: 10.4271/910070.

[10] G. Martin, V. Talon, P. Higelin, A. Charlet, and C. Caillol, "Implementing turbomachinery physics into data map-based turbocharger models," *SAE International Journal of Engines*, vol. 2, no. 1, pp. 211–229, 2009. doi: 10.4271/2009-01-0310.

[11] J. E. Hadef, G. Colin, Y. Chamaillard, and V. Talon, "Physical-based algorithms for interpolation and extrapolation of turbocharger data maps," *SAE International Journal of Engines*, vol. 5, no. 2, pp. 363–378, 2012. doi: 10.4271/2012-01-0434.

[12] X. Llamas and L. Eriksson, "Control-oriented compressor model with adiabatic efficiency extrapolation," *SAE International Journal of Engines*, vol. 10, no. 4, pp. 1903–1916, 2017. doi: 10.4271/2017-01-1032.

[13] X. Llamas and L. Eriksson, "Parameterizing compact and extensible compressor models using orthogonal distance minimization," *Journal of Engineering for Gas Turbines and Power*, vol. 139, no. 1, p. 012601, 2016. doi: 10.1115/1.4034152.

[14] T. Zeng, D. Upadhyay, and G. Zhu, "A reduced complexity model for the compressor power of an automotive turbocharger," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 6, p. 061018, 2018. doi: 10.1115/1.4039285.

[15] M. Müller, E. Hendricks, and S. C. Sorenson, "Mean value modelling of turbocharged spark ignition engines," tech. rep., Technical University of Denmark, 1998. doi: 10.4271/980784.

[16] O. Leufvén and L. Eriksson, "A surge and choke capable compressor flow model—validation and extrapolation capability," *Control Engineering Practice*, vol. 21, no. 12, pp. 1871–1883, 2013. doi: 10.1016/j.conengprac.2013.07.005.

[17] P. Moraal and I. Kolmanovsky, "Turbocharger modeling for automotive control applications," tech. rep., 1999. doi: 10.4271/1999-01-0908.

[18] X. Llamas and L. Eriksson, "Liu cpgui: A toolbox for parameterizing compressor models," tech. rep., Vehicular Systems, Department of Electrical Engineering Linkoping University, 2018. `https://www.vehicular.isy.liu.se/en/Software/LiU_CPgui`.

[19] L. Eriksson and L. Nielsen, *Modeling and Control of Engines and Drivelines*. John Wiley and Sons Inc, 2014.

[20] J. Nocedal and S. J. Wright, "Numerical optimization, 2nd ed.," pp. 265–268, 2006.

[21] "Simulink documentation." https://se.mathworks.com/help/simulink/index.html.

[22] M. Association, *Modelica® - A Unified Object-Oriented Language for Systems Modeling*.

[23] C. Andersson, "Design of the modelica library vehprolib with non-ideal gas models in engines," Master's thesis, Linköping University, 2015. LiTH-ISY-EX–15/4888–SE.

[24] L. Gamma Technologies, "Gt-power engine simulation software." https://www.gtisoft.com/gt-suite-applications/propulsion-systems/gt-power-engine-simulation-software/.

[25] O. VÍTEK, J. MACEK, and M. POLÁŠEK, "New approach to turbocharger optimization using 1-d simulation tools," tech. rep., Czech Technical University in Prague Josef Božek Research Center, 2006. doi: 10.4271/2006-01-0438.