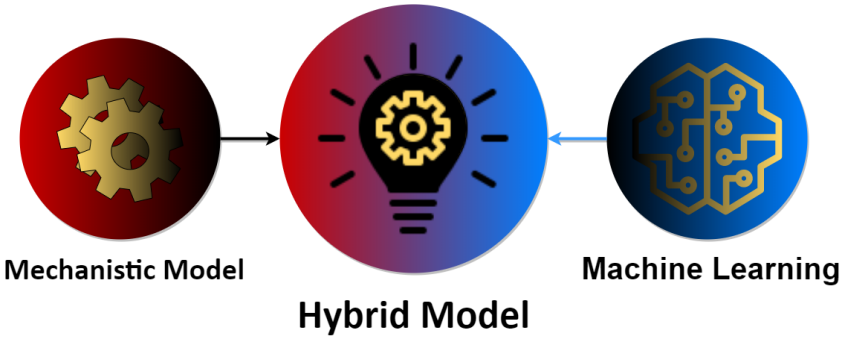
University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2020
Electrical Power Engineering

Hybrid Machine Learning and Mechanistic Thermal Model of Synchronous Generator



Khaled Aleikish

Faculty of Technology, Natural Sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis 2020
Title: *Hybrid Machine Learning and Mechanistic Thermal Model of Synchronous Generator*
Pages: *122*
Keywords: *Thermal model, Machine learning, Linear regression, Nonlinear regression.*

Student: *Khaled Aleikish*
Supervisor: *Bernt Lie, co-supervisor Madhusudhan Pandey*
External partner: *Skagerak Kraft, Ingunn Granstrøm*
Availability: *Open*

Summary:
Strict constraints are imposed on the power factor in the operation of the synchronous machines for electricity production, which are in place to protect the synchronous generators from overheating due to the increase in currents from a high power factor. However, by exploiting the total thermal capacity of the synchronous generator and by keeping temperature development in key points under control, it is possible to relax the constraints on the power factor. A thermal model of an air-cooled synchronous generator was proposed in Øyvang (2018) and studied in Lie (2018a). A possible extension of this model is the case of temperature dependence in the heat capacities of air and water in the counter-current heat exchanger. To handle this case, it is necessary to solve a nonlinear two-point boundary value problem numerically for each time step when solving the thermal synchronous generator model, which is relatively costly for online use. In this thesis work, to speed up the solution time of the non-ideal heat exchanger sub-model, explicit data-driven models were developed using linear and nonlinear regression for a variety of conditions and expressed as a correction expression to the ideal heat exchanger model. Moreover, the execution speed of the numeric solution of the nonlinear two-point boundary value problem was compared with that of the explicit data-driven models. Also, the experimental data of Åbjøra was compared with the predictions of the mechanistic model, and it was observed that even when extending the model from Lie (2018) with a more realistic heat exchanger model with temperature dependence in the specific heat capacities of air and water, it was not possible to get a good fit to the available experimental data. To get improved model fit, one possibility is to model the system as good as possible with a mechanistic model and then add a dynamic regression model to describe the difference between the mechanistic model and the experimental data. In summary: a hybrid mechanistic-empirical model is a hot research topic. The promise is to get a good model fit, with efficient model simulation.

Preface

In January 2020, this thesis was undertaken as a part of the course FMH606 in the Department of Electrical Engineering, IT, and Cybernetics of the University of South-Eastern Norway (USN), under the supervision of Professor Bernt Lie, and co-supervisor Madhusudhan Pandey. I offer my profound gratitude for the support and assistance of these supervisors. My special thanks go to my family members and my friends for the encouragement and support during my studies.
This thesis explores the potential of a hybrid mechanistic-empirical thermal model of an air-cooled synchronous generator. Mainly, the aim is to reduce the simulation time of the nonlinear two-point boundary value problem of the heat exchanger sub-model using linear and nonlinear regression.

Porsgrunn, May 15, 2020

Khaled Aleikish

Contents

List of Figures

List of Tables

Nomenclature

Symbol	Explanation
R^2	The Coefficient of Determination
$\bar{\mathrm{R}}^2$	The adjusted Coefficient of Determination
$T_{\mathrm{j}}^{\mathrm{X}}$(X =A, N, H, R, or M. j = a, w, etc.)	The temperature of species j obtained by: A = Analytic solution N = Numeric solution H = Hybrid solution R = Regression M = Measurements/Experimental data

1 Introduction

1.1 Background

Strict constraints are imposed on the power factor in the operation of the synchronous machines for electricity production. For example, in M. Pandey (2019a), it was reported that the power factor is constrained to the range $[0.85, 0.95]$ in the European hydropower generation, and below 0.86 in the Norwegian hydropower generation. Furthermore, the strict constraints are in place to protect the synchronous generators from overheating, which can occur due to the increase in currents from a high power factor. However, by exploiting the total thermal capacity of the synchronous generator and by keeping temperature development in key points under control, it is possible to relax the constraints on the power factor (Lie, 2018a; Øyvang, 2018).

1.2 Previous Work

A thermal model of an air-cooled synchronous generator was proposed in Øyvang (2018), and studied in Lie (2018a), M. Pandey (2019a), and M. Pandey (2019b). In M. Pandey (2019a), the numeric solution of the two-point boundary value problem of the heat exchanger sub-model proved to be costly for online use. Specifically, the simulation time of models 3b and 4b with temperature dependence in the specific heat capacities of air and water in the heat exchanger sub-model could be improved. Moreover, in Lie (2019b), ideas relevant to data-driven models and machine learning are described. Particularly, a simple introduction is given to machine learning and the machine learning package Flux for the modern computer science language Julia.

1.3 Scope and Outline

In this thesis work, the thermal model of an air-cooled synchronous generator that was studied in Lie (2018a), is extended with the more realistic case of temperature dependence in the specific heat capacities of air and water. However, the aim is not to demonstrate the impact of temperature dependence in the specific heat capacities on the thermal model,

but to speed up the solution time of heat exchanger sub-model when the temperature dependence is considered. Furthermore, explicit data-driven models are developed using linear and nonlinear regression for a variety of conditions, and their execution speeds are compared with that of the numeric solution of the nonlinear two-point boundary value problem. Finally, the experimental data of Åbjøra is compared with the predictions of the mechanistic model. However, due to time constraints, a data-driven model is not fitted to Åbjøra experimental data.[1]

In Chapter 2, first, an overview of the thermal model of an air-cooled synchronous generator with an ideal heat exchanger model is presented. Second, an overview of the counter-current heat exchanger model development is given. Third, empirical expressions in absolute temperature (T) are introduced to describe the temperature dependence in the specific heat capacities of air and water in the heat exchanger sub-model. Fourth, the analytic solution of the ideal heat exchanger model and the numeric solution of the non-ideal model are compared. Finally, the thermal model of an air-cooled synchronous generator is implemented in Julia with ideal and non-ideal heat exchanger models and compared.

In Chapter 3, first, an overview of regression analysis is given. Next, cross-validation is briefly described. Then, linear and nonlinear regression of the counter-current heat exchanger model is implemented in the Julia language. Finally, the experimental data of Åbjøra is compared with the predictions of the mechanistic model.

Chapter 4 will be for results and discussion. Also, Chapter 5 gives the concluding remarks, and Chapter 6 describes possible future work.

Appendix A contains the tasks description of this thesis work, and Appendix B contains the code listing in Julia language.

[1] Task 5 in Appendix A was not carried out in this thesis work.

2 Overview of the Thermal Model of an Air-Cooled Synchronous Generator, With Ideal and Non-Ideal Heat Exchanger Model

In this chapter, first, an overview of the thermal model of an air-cooled synchronous generator with an ideal heat exchanger model is given, which was developed as part of a group project in course FM1015 Modelling of Dynamic Systems (Lie, 2018a). Next, an overview of the counter-current heat exchanger model development is presented, and an efficient explicit/analytic expression is obtained for the case of temperature independence in heat capacity and/or heat transfer of air/water (the case of an ideal heat exchanger model) (Lie, 2019a). Then, for the case of temperature dependence in heat capacity and/or heat transfer of air/water (the case of a non-ideal heat exchanger model), empirical expressions in absolute temperature ($\boldsymbol{T}$) are introduced to describe the specific heat capacity (Lie, 2019a; Murphy, 2020). After that, the analytic solution of the ideal heat exchanger model and the numeric solution of the non-ideal model are compared. Finally, the thermal model of an air-cooled synchronous generator is implemented in Julia with ideal and non-ideal heat exchanger models and compared.

2.1 Overview of the Thermal Model of an Air-Cooled Synchronous Generator

In a recent group project in course FM1015 Modelling of Dynamic Systems, Lie (2018a), a dynamic model was proposed to study the temperature evolution in the synchronous generator. The main ideas were taken from **OyvangThomas2019OMTP**, but the model development followed the structure of the course to give experience in formulating and solving dynamic models of systems. This section is an overview of the model development and a summary of the key assumptions that were introduced in Lie (2018b). First, the thermal operation of an air-cooled synchronous generator is described, and the model objectives are defined. Next, the model is developed in a structure similar to that of course FM1015 Modelling of Dynamic Systems. At last, a model summary is presented.

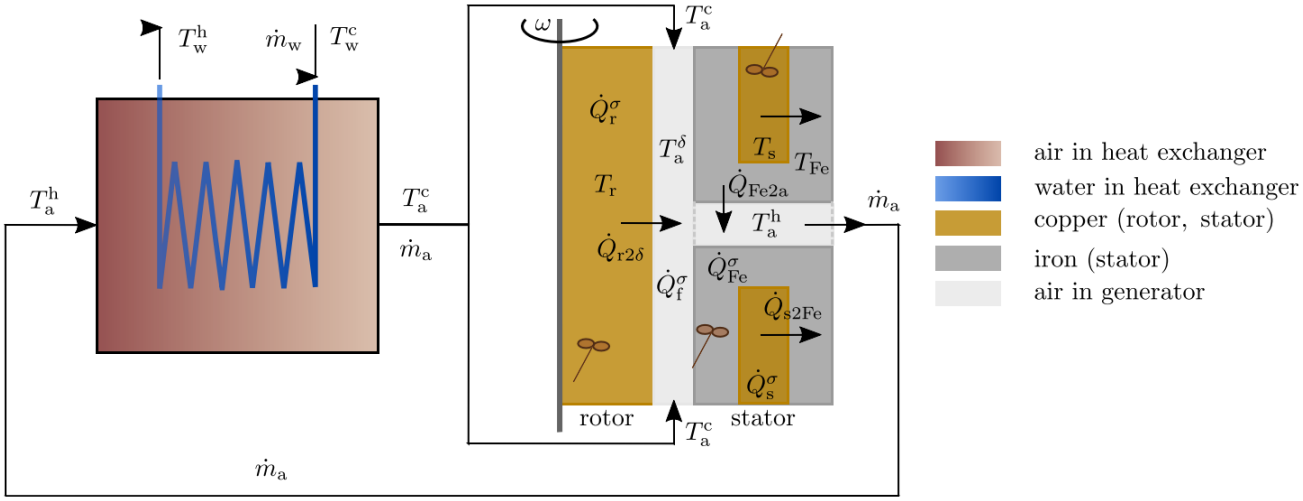


Figure 2.1: Thermal operation of an air-cooled synchronous generator (Lie, 2018a).

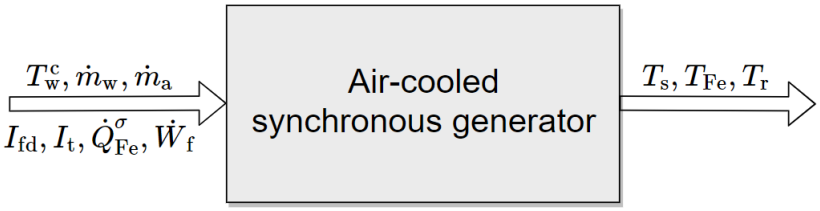


Figure 2.2: Functional diagram of an air-cooled synchronous generator.

2.1.1 Model objectives

An overview of the thermal operation of an air-cooled synchronous generator is shown in Fig. 2.1. First, the cold air coming out of the heat exchanger is blown by a fan into the gap between the rotor and the stator, and is heat by heat flow from the rotor ($\dot{Q}_{r2\delta}$) and by windage and bearing friction ($\dot{Q}_f^\sigma$). Then, the air flows through the iron windings surrounding the copper wires in the stator and is heated by the heat transfer from the iron ($\dot{Q}_{Fe2a}$). At last, the air flows through the heat exchanger and is cooled down by cold water before returning to the air gap between the rotor and the stator.

Based on the project task description and the functional diagram in Fig. 2.2, a DAE model of the inputs u and outputs y can be developed for the synchronous generator. The model inputs are:

$$u = (T_w^c, \dot{m}_w, \dot{m}_a, I_{fd}, I_t, \dot{Q}_{Fe}^\sigma, \dot{W}_f),$$

and the model outputs are:

$$y = (T_s, T_{Fe}, T_r).$$

Symbol	Description	Unit
	Inputs u	
$T_{\mathrm{w}}^{\mathrm{c}}$	Influent water temperature	°C
$\dot{m}_{\mathrm{w}}$	Water mass flow rate	kg/s
$\dot{m}_{\mathrm{a}}$	Air mass flow rate	kg/s
I_{fd}	Rotor field current	A
I_{t}	Stator terminal current	A
$\dot{Q}_{\mathrm{Fe}}^{\sigma}$	Stator iron generated heat	kW
$\dot{W}_{\mathrm{f}}$	Friction work rate in the air gap	kW
	Outputs y	
T_{s}	Stator copper temperature	°C
T_{r}	Rotor temperature	°C
T_{Fe}	Stator iron temperature	°C

Table 2.1: Description of the inputs and outputs of an air-cooled synchronous generator model.

The inputs and outputs of the model are described in Table 2.1 (Lie, 2018b).

2.1.2 Model development

The model development is divided into three steps:

- Step 1: Introduce the relevant balance laws.
- Step 2: Relate the quantities in the balance laws to the inputs and outputs.
- Step 3: Manipulate the model into a suitable form (DAE or ODE) for the computer language that will be used to describe/solve the model.

2.1.2.1 Step 1: Introduce the relevant balance laws

With the model objective specified, the next step is to introduce the necessary balance laws. Because the model involves flow of mass (air, but also water in the heat exchanger sub-model), the mass balance is relevant to model development. The total mass balance is given in Lie (2019a) as:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \dot{m}_{\mathrm{i}} - \dot{m}_{\mathrm{e}}. \tag{2.1}$$

Since the dynamics in the air is much faster than that of the metals, the model development can be simplified by neglecting the dynamics/inertia in the air volumes. Then it is assumed that the air volumes have constant density and no accumulation of mass. This is a key assumption because the mass balance for the air volumes reduces to $\dot{m}_{\mathrm{i}} = \dot{m}_{\mathrm{e}}$ under steady-state conditions, i.e., the model is simplified by reducing the number of differential

equations. Also, since there is no mass flow in the metals, their masses remain constant, and their enthalpy flow rates $\dot{H} \equiv 0$. Since there is no change in mass for the air or the solids, the mass balance is no longer relevant for model development.

However, since the development of the temperature in the system is of interest, the energy balance is relevant. The thermal energy balance is given in Lie (2019a) as:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \dot{H}_\mathrm{i} - \dot{H}_\mathrm{e} - \dot{W}_\mathrm{f} - \dot{W}_\mathrm{v} + \dot{Q}. \tag{2.2}$$

The thermal energy balance is simplified by assuming the volume of each subsystem is constant, so $\dot{W}_\mathrm{v} = p\frac{\mathrm{d}V}{\mathrm{d}t}$ is zero. Moreover, the mechanical power produced from the friction work in the air volumes is included into the total heat flow $\dot{Q}_\mathrm{total}$, so $\dot{W}_\mathrm{f} = 0$. For the metal volumes, there is no friction. Then, for metals, the thermal energy balance is reduced to (Lie, 2018b):

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \dot{H}_\mathrm{i} - \dot{H}_\mathrm{e} + \dot{Q}. \tag{2.3}$$

2.1.2.2 Step 2: Relate the quantities in the balance laws to the inputs and outputs

First, the quantities on the left-hand side in Eq. 2.3 are described for the air volumes and the metal parts. Since the air volumes have fast dynamics and are assumed to be in steady-state, $\frac{\mathrm{d}U}{\mathrm{d}t}$ can be set to zero in their thermal energy balance. In addition, as mentioned in the task description, it is assumed that the heat conduction in the metal parts is much larger than the heat transport across the metal boundaries. Then the metal temperatures can be assumed to be homogeneous within each metal part, which is indicated by the impeller symbol in Fig. 2.1. A homogeneous system is composed of the same kind of elements and have a common property throughout. The specific volume for a homogeneous system is defined in Lie (2019a) as:

$$\hat{V} \triangleq \frac{V}{m}. \tag{2.4}$$

Then the volume can be expressed as $V = m\hat{V}$ in the following expression for the internal energy U in the metals (Lie, 2018b):

$$U = H - pV. \tag{2.5}$$

Here, the pressure p is the atmospheric pressure p_a as mentioned in the project description. and the enthalpy can be given in terms of specific enthalpy (Lie, 2018b):

$$H = m\hat{H}. \tag{2.6}$$

To describe the specific enthalpy $\hat{H}$, the coefficient of thermal expansion α_p is required, which is defined in Lie (2019a) as:

$$\alpha_p \triangleq \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_p. \tag{2.7}$$

Also, in Lie (2019a), the differential in specific enthalpy is given by:

$$\mathrm{d}\hat{H} = \hat{c}_p \mathrm{dT} + \hat{V}(1-\alpha_p T)\mathrm{d}p, \tag{2.8}$$

and the coefficient of thermal expansion for an ideal gas (air in this model) is given as:

$$\alpha_p = \frac{1}{T}. \tag{2.9}$$

Moreover, by inserting the expression of Eq. 2.9 in Eq. 2.8, then the specific enthalpy $\hat{H}$ of air can be described by:

$$\mathrm{d}\hat{H} = \hat{c}_p \mathrm{d}T \Rightarrow \hat{H} = \hat{H}^\circ + \int_{T^\circ}^{T} \hat{c}_p dT. \tag{2.10}$$

For an ideal solid, $\alpha_p = 0$ as mentioned in Lie (2019a), and the specific enthalpy $\hat{H}$ for the metals is expressed as:

$$\hat{H} = \hat{H}^\circ + \int_{T^\circ}^{T} \hat{c}_p dT + \hat{V}(p-p^\circ). \tag{2.11}$$

Typically, the standard state pressure $p^\circ = 1\,\mathrm{atm}$, and as mentioned before, the pressure in the system is also the atmospheric pressure, then the term $\hat{V}(p-p^\circ)$ in Eq. 2.11 can be set to zero, and the specific enthalpy of the metals is expressed the same as for the air:

$$\hat{H} = \hat{H}^\circ + \int_{T^\circ}^{T} \hat{c}_p dT. \tag{2.12}$$

As described in Lie (2019), if the heat capacity $\hat{c}_p$ is considered temperature independent, then Eq. 2.12 for the specific enthalpy can be written in the form:

$$\hat{H}(T,p) = \hat{H}(T^\circ,p^\circ) + \hat{c}_p(T-T^\circ). \tag{2.13}$$

The temperature T° is taken as $25^\circ C$, and $\hat{H}(T^\circ,p^\circ)$ is set to zero (Lie, 2018b). Next, the quantities on the right-hand side in Eq. 2.3 are described for the air volumes and the metal parts. The enthalpy flow rate $\dot{H}$ is given in Lie (2019a) as:

$$\dot{H} = \dot{m}\hat{H}, \tag{2.14}$$

here, the specific enthalpy is described by Eq. 2.13 since the heat capacities of air, water and metals are assumed to be constant (Lie, 2018a). Finally, the flow rates $\dot{Q}$ in the thermal energy balance are expressed by Eqs. 2.18, 2.19, 2.25, 2.30,2.35, 2.36, and Eq. 2.46.

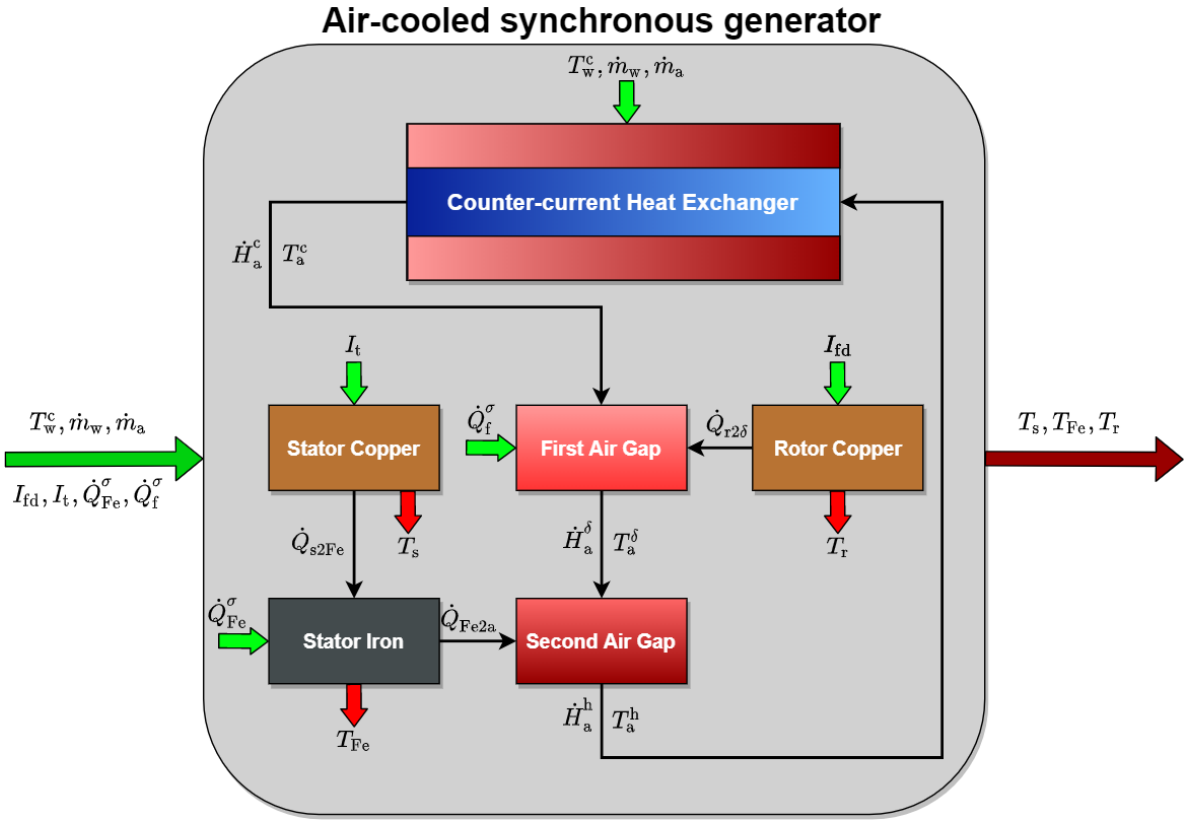


Figure 2.3: Functional diagram of the synchronous generator model with the subsystems.

2.1.2.3 Step 3: Manipulate the model into a suitable form

The synchronous generator model can be divided into five subsystems: first air gap, second air gap, rotor copper, stator copper, and stator iron. In addition, there is the heat exchanger sub-model. Figure 2.3 shows a detailed functional diagram of the synchronous generator model with the subsystems. In this step in the model development, the model equations are expressed for each subsystem (Lie, 2018b). Table 2.2 describes the parameters, the algebraic variables, and the differential variables used in the thermal model of an air-cooled synchronous generator (Lie, 2018b).

First air gap equations:

$$\frac{\mathrm{d}U_a^\delta}{\mathrm{d}t} = \dot{H}_a^c - \dot{H}_a^\delta + \dot{Q}_{r2\delta} + \dot{Q}_f^\sigma \approx 0 \tag{2.15}$$

$$\dot{H}_a^c = \dot{m}_a \hat{H}_a^c \tag{2.16}$$

$$\hat{H}_a^c = \hat{H}_a^\circ + \hat{c}_{p,a}(T_a^c - T_a^\circ) \tag{2.17}$$

$$\dot{Q}_{\mathrm{f}}^{\sigma} = 0.8\dot{W}_{\mathrm{f}} \tag{2.18}$$

$$\dot{Q}_{\mathrm{r}2\delta} = \mathscr{U} A_{\mathrm{r}2\delta}(T_{\mathrm{r}} - T_{\mathrm{a}}^{\delta}) \tag{2.19}$$

$$\dot{H}_{\mathrm{a}}^{\delta} = \dot{m}_{\mathrm{a}} \hat{H}_{\mathrm{a}}^{\delta} \tag{2.20}$$

$$\hat{H}_{\mathrm{a}}^{\delta} = \hat{H}_{\mathrm{a}}^{\circ} + \hat{c}_{p,\mathrm{a}}(T_{\mathrm{a}}^{\delta} - T_{\mathrm{a}}^{\circ}). \tag{2.21}$$

Second air gap equations:

$$\frac{\mathrm{d}U_{\mathrm{a}}^{\mathrm{h}}}{\mathrm{d}t} = \dot{H}_{\mathrm{a}}^{\delta} - \dot{H}_{\mathrm{a}}^{\mathrm{h}} + \dot{Q}_{\mathrm{Fe}2\mathrm{a}} \approx 0 \tag{2.22}$$

$$\dot{H}_{\mathrm{a}}^{\mathrm{h}} = \dot{m}_{\mathrm{a}} \hat{H}_{\mathrm{a}}^{\mathrm{h}} \tag{2.23}$$

$$\hat{H}_{\mathrm{a}}^{\mathrm{h}} = \hat{H}_{\mathrm{a}}^{\circ} + \hat{c}_{p,\mathrm{a}}(T_{\mathrm{a}}^{\mathrm{h}} - T_{\mathrm{a}}^{\circ}) \tag{2.24}$$

$$\dot{Q}_{\mathrm{Fe}2\mathrm{a}} = \mathscr{U} A_{\mathrm{Fe}2\mathrm{a}}(T_{\mathrm{Fe}} - T_{\mathrm{a}}^{\mathrm{h}}). \tag{2.25}$$

Rotor copper equations:

$$\frac{\mathrm{d}U_{\mathrm{r}}}{\mathrm{d}t} = \dot{Q}_{\mathrm{r}}^{\sigma} - \dot{Q}_{\mathrm{r}2\delta} \tag{2.26}$$

$$U_{\mathrm{r}} = H_{\mathrm{r}} - p_{\mathrm{a}} V_{\mathrm{r}} \tag{2.27}$$

$$H_{\mathrm{r}} = m_{\mathrm{r}} \hat{H}_{\mathrm{r}} \tag{2.28}$$

$$\hat{H}_{\mathrm{r}} = \hat{H}_{\mathrm{Cu}}^{\circ} + \hat{c}_{p,\mathrm{Cu}}(T_{\mathrm{r}} - T_{\mathrm{Cu}}^{\circ}) \tag{2.29}$$

$$\dot{Q}_{\mathrm{r}}^{\sigma} = 1.1 R_{\mathrm{r}} I_{\mathrm{fd}}^{2}. \tag{2.30}$$

Stator copper equations:

$$\frac{\mathrm{d}U_{\mathrm{s}}}{\mathrm{d}t} = \dot{Q}_{\mathrm{s}}^{\sigma} - \dot{Q}_{\mathrm{s}2\mathrm{Fe}} \tag{2.31}$$

$$U_{\mathrm{s}} = H_{\mathrm{s}} - p_{\mathrm{a}} V_{\mathrm{s}} \tag{2.32}$$

$$H_{\mathrm{s}} = m_{\mathrm{s}} \hat{H}_{\mathrm{s}} \tag{2.33}$$

$$\hat{H}_{\mathrm{s}} = \hat{H}_{\mathrm{Cu}}^{\circ} + \hat{c}_{p,\mathrm{Cu}}(T_{\mathrm{s}} - T_{\mathrm{Cu}}^{\circ}) \tag{2.34}$$

$$\dot{Q}_{\mathrm{s}}^{\sigma} = 3 R_{\mathrm{s}} I_{\mathrm{t}}^{2} \tag{2.35}$$

$$\dot{Q}_{\mathrm{s}2\mathrm{Fe}} = \mathscr{U} A_{\mathrm{s}2\mathrm{Fe}}(T_{\mathrm{s}} - T_{\mathrm{Fe}}). \tag{2.36}$$

Stator iron equations:

$$\frac{\mathrm{d}U_{\mathrm{Fe}}}{\mathrm{d}t} = \dot{Q}_{\mathrm{Fe}}^{\sigma} - \dot{Q}_{\mathrm{Fe}2\mathrm{a}} + \dot{Q}_{\mathrm{s}2\mathrm{Fe}} \tag{2.37}$$

$$U_{\mathrm{Fe}} = H_{\mathrm{Fe}} - p_{\mathrm{a}} V_{\mathrm{Fe}} \tag{2.38}$$

$$H_{\mathrm{Fe}} = m_{\mathrm{Fe}} \hat{H}_{\mathrm{Fe}} \tag{2.39}$$

$$\hat{H}_{\mathrm{Fe}} = \hat{H}^{\circ}_{\mathrm{Fe}} + \hat{c}_{p,\mathrm{Fe}}(T_{\mathrm{Fe}} - T^{\circ}_{\mathrm{Fe}}). \tag{2.40}$$

Finally, the heat exchanger explicit equations taken from the task description:

$$T^{\mathrm{h}}_{\mathrm{w}} = \frac{N^{\mathrm{w}}_{\mathrm{St}}(1-\exp(-N^{\triangle}_{\mathrm{St}}))T^{\mathrm{h}}_{\mathrm{a}} + N^{\triangle}_{\mathrm{St}}\exp(-N^{\triangle}_{\mathrm{St}})T^{\mathrm{c}}_{\mathrm{w}}}{N^{\mathrm{w}}_{\mathrm{St}} - N^{\mathrm{a}}_{\mathrm{St}}\exp(-N^{\triangle}_{\mathrm{St}})} \tag{2.41}$$

$$T^{\mathrm{c}}_{\mathrm{a}} = \frac{N^{\triangle}_{\mathrm{St}}T^{\mathrm{h}}_{\mathrm{a}} + N^{\mathrm{a}}_{\mathrm{St}}(1-\exp(-N^{\triangle}_{\mathrm{St}}))T^{\mathrm{c}}_{\mathrm{w}}}{N^{\mathrm{w}}_{\mathrm{St}} - N^{\mathrm{a}}_{\mathrm{St}}\exp(-N^{\triangle}_{\mathrm{St}})}, \tag{2.42}$$

here,

$$N^{\triangle}_{\mathrm{St}} = N^{\mathrm{w}}_{\mathrm{St}} - N^{\mathrm{a}}_{\mathrm{St}}, \tag{2.43}$$

and,

$$N^{\mathrm{w}}_{\mathrm{St}} = \frac{\mathscr{U}A_{\mathrm{x}}}{\hat{c}_{p,\mathrm{w}}\dot{m}_{\mathrm{w}}} \tag{2.44}$$

$$N^{\mathrm{a}}_{\mathrm{St}} = \frac{\mathscr{U}A_{\mathrm{x}}}{\hat{c}_{p,\mathrm{a}}\dot{m}_{\mathrm{a}}}. \tag{2.45}$$

Here, $\mathscr{U}A_{\mathrm{x}} = \frac{1}{\frac{1}{hA_{\mathrm{ax}}}+\frac{1}{hA_{\mathrm{wx}}}}$. Optionally, the heat transfer from water to air can also be calculated from:

$$\dot{Q}_{\mathrm{w2a}} = \frac{\exp(-N^{\triangle}_{\mathrm{St}})-1}{\frac{1}{\hat{c}_{p,\mathrm{a}}\dot{m}_{\mathrm{a}}}\exp(-N^{\triangle}_{\mathrm{St}}) - \frac{1}{\hat{c}_{p,\mathrm{w}}\dot{m}_{\mathrm{w}}}}(T^{\mathrm{c}}_{\mathrm{w}} - T^{\mathrm{h}}_{\mathrm{a}}), \tag{2.46}$$

here, it is important to highlight that It is not necessary to compute $\dot{Q}_{\mathrm{w2a}}$ or $T^{\mathrm{h}}_{\mathrm{w}}$ to solve the thermal model of an air-cooled synchronous generator. However, computing these quantities give insight into the thermal operation of the model. Also, it is important to highlight that $T^{\mathrm{h}}_{\mathrm{w}}$, $T^{\mathrm{c}}_{\mathrm{a}}$, and $\dot{Q}_{\mathrm{w2a}}$ are functions of the specific heat capacities of air and water ($\hat{c}_{p,\mathrm{a}}$ and $\hat{c}_{p,\mathrm{w}}$) . In Lie (2018a), $\hat{c}_{p,\mathrm{a}}$ and $\hat{c}_{p,\mathrm{w}}$ were assumed to be constant, but for a more realistic heat exchanger model, it is possible to extend the model in Lie (2018a), with the case of temperature dependence in heat capacity and/or heat transfer of air/water.

Symbol	Description	Unit
	Differential variables x	
U_{r}	Internal energy of rotor	kJ
U_{s}	Internal energy of stator copper	kJ
U_{Fe}	Internal energy of stator iron	kJ
	Algebraic variables z	

Table 2.2: Description of the parameters, the algebraic variables, and the differential variables used in the thermal model of an air-cooled synchronous generator.

Symbol	Description	Unit
T_r	Rotor temperature	°C
T_s	Stator copper temperature	°C
T_Fe	Stator iron temperature	°C
T_w^h	Hot water temperature	°C
T_a^h	Hot air temperature	°C
T_a^c	Cold air temperature	°C
T_a^δ	Air gap temperature	°C
$\dot{H}_\mathrm{a}^\delta$	Enthalpy flow rate of air in the gap	kJ/s
$\dot{H}_\mathrm{a}^\mathrm{c}$	Enthalpy flow rate of cold air	kJ/s
$\dot{H}_\mathrm{a}^\mathrm{h}$	Enthalpy flow rate of hot air	kJ/s
H_r	Enthalpy of rotor	kJ
H_s	Enthalpy of stator copper	kJ
H_Fe	Enthalpy of stator iron	kJ
$\hat{H}_\mathrm{a}^\delta$	Specific enthalpy of air in the gap	kJ/kg
$\hat{H}_\mathrm{a}^\mathrm{c}$	Specific enthalpy of cold air	kJ/kg
$\hat{H}_\mathrm{a}^\mathrm{h}$	Specific enthalpy of hot air	kJ/kg
$\hat{H}_\mathrm{r}$	Specific enthalpy of rotor	kJ/kg
$\hat{H}_\mathrm{s}$	Specific enthalpy of stator copper	kJ/kg
$\hat{H}_\mathrm{Fe}$	Specific enthalpy of stator iron	kJ/kg
N_St^a	Stanton number of air	–
N_St^w	Stanton number of water	–
$N_\mathrm{St}^\triangle$	Difference in Stanton numbers	–
$\dot{Q}_\mathrm{r}^\sigma$	Heat source in the rotor copper	kW
$\dot{Q}_\mathrm{s}^\sigma$	Heat source in the stator copper	kW
$\dot{Q}_\mathrm{Fe}^\sigma$	Heat source in the stator iron	kW
$\dot{Q}_{\mathrm{r}2\delta}$	Heat loss from rotor to air gap	kW
$\dot{Q}_\mathrm{s2Fe}$	Heat loss from stator copper to iron	kW
$\dot{Q}_\mathrm{Fe2a}$	Heat loss from stator iron to air	kW
$\dot{Q}_\mathrm{w2a}$	Heat transfer from water to air	kW
$\dot{Q}_\mathrm{f}^\sigma$	Friction heating rate in the air gap	kW
	Parameters $\boldsymbol{\theta}$	
$\hat{c}_{p,\mathrm{a}}$	Specific heat capacity, air	kJ/kg/K
$\hat{c}_{p,\mathrm{w}}$	Specific heat capacity, water	kJ/kg/K
$\hat{c}_{p,\mathrm{Cu}}$	Specific heat capacity, copper	kJ/kg/K
$\hat{c}_{p,\mathrm{Fe}}$	Specific heat capacity, iron	kJ/kg/K
p_a	Atmospheric pressure	Pa
$\mathscr{U}A_\mathrm{x}$	Heat transfer, air to water	kW/K
hA_ax	Heat transfer, air side of the heat exchanger	kW/K

Table 2.2: Description of the parameters, the algebraic variables, and the differential variables used in the thermal model of an air-cooled synchronous generator.

Symbol	Description	Unit
hA_{wx}	Heat transfer, water side of the heat exchanger	kW/K
$\mathscr{U}A_{\mathrm{r2}\delta}$	Heat transfer, rotor to air gap	kW/K
$\mathscr{U}A_{\mathrm{Fe2a}}$	Heat transfer, stator iron to air	kW/K
$\mathscr{U}A_{\mathrm{s2Fe}}$	Heat transfer, stator copper to iron	kW/K
$\hat{H}^{\circ}_{\mathrm{a}}$	Reference specific enthalpy of air	kJ/kg
$\hat{H}^{\circ}_{\mathrm{Cu}}$	Reference specific enthalpy of copper	kJ/kg
$\hat{H}^{\circ}_{\mathrm{Fe}}$	Reference specific enthalpy of iron	kJ/kg
T°_{a}	Reference temperature of air	°C
T°_{Cu}	Reference temperature of copper	°C
T°_{Fe}	Reference temperature of iron	°C
m_{r}	Mass of rotor	kg
m_{s}	Mass of stator copper	kg
m_{Fe}	Mass of stator iron	kg
R_{r}	Rotor copper ohmic resistance	kΩ
R_{s}	Stator copper ohmic resistance	kΩ

Table 2.2: Description of the parameters, the algebraic variables, and the differential variables used in the thermal model of an air-cooled synchronous generator.

2.1.3 Model summary.

The model can be expressed in the standard DAE form, which is a suitable form for the Modelica computer language:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x, z, u; \theta)$$

$$0 = g(x, z, u; \theta)$$

$$y = h(x, z, u; \theta).$$

The inputs (u) and the outputs (y) are described in Table 2.1.Also, the parameters (θ), the algebraic variables (z), and the differential variables (x) are described in Table 2.2. The model consists of 32 independent equations (Eqs. 2.15 − 2.46), and 32 unknowns (dim x + dim z), which makes it a well-posed model that does not require any further re-formulation before implementation in computer (Lie, 2018b).

2.2 Overview of the Counter-Current Heat Exchanger Model Development

The explicit expressions of $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$ in Eqs. 2.41 and 2.42 are only valid for an ideal heat exchanger model. If the specific heat capacities of air and water depended on temperature, then a nonlinear two-point boundary value problem is formed, which require a numerical method to solve. In this section, an overview of the counter-current heat exchanger model development is presented to demonstrate how a linear/nonlinear two-point boundary value problem may form depending on the assumption of temperature dependence in the specific heat capacities of air and water.

Thermal models of distributed heat exchangers are developed in Modeling of Dynamic Systems lecture notes (Lie, 2019a). The focus of the model development is the tube-and-shell type of heat exchangers. The tube-and-shell is a common type of heat exchangers where one fluid flows inside a tube, and the other fluid flows outside of the tube, between the tube and the shell. Models are developed for the three primary forms of flow configurations in Fig. 2.4: (1) cross-current heat exchangers, (2) parallel flow or co-current heat exchangers, (3) counter-current heat exchangers. During this work, the focus will be on the counter-current flow arrangement.

In Lie (2019a), the model development starts by introducing the required balance laws:

1. The total mass balance as in Eq. 2.1:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \dot{m}_{\mathrm{i}} - \dot{m}_{\mathrm{e}}.$$

2. The thermal energy balance as in Eq. 2.2:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \dot{H}_{\mathrm{i}} - \dot{H}_{\mathrm{e}} - \dot{W}_{\mathrm{f}} - \dot{W}_{\mathrm{v}} + \dot{Q}.$$

Next, some simplifying assumptions are introduced. The first assumption is the assumption of steady-state conditions for the mass balance for both fluids in the heat exchanger. Under steady-state conditions, the left-hand side of the mass balance equation $\frac{\mathrm{d}m}{\mathrm{d}t}$ becomes zero, which leads to a constant mass flow rate at the inlet and outlet of both shell side and tube side of the counter-current heat exchanger. That is, the mass balance for the tube side reduces to:

$$\dot{m}_{\mathrm{i}}^{\mathrm{t}} = \dot{m}_{\mathrm{e}}^{\mathrm{t}}. \tag{2.47}$$

Similarly, the mass balance for the shell side reduces to:

$$\dot{m}_{\mathrm{i}}^{\mathrm{s}} = \dot{m}_{\mathrm{e}}^{\mathrm{s}}. \tag{2.48}$$

The second simplifying assumption relates to the work terms in the thermal energy balance. The friction term $\dot{W}_{\mathrm{f}}$ is neglected. Also, the volume of both sides, the tube side,

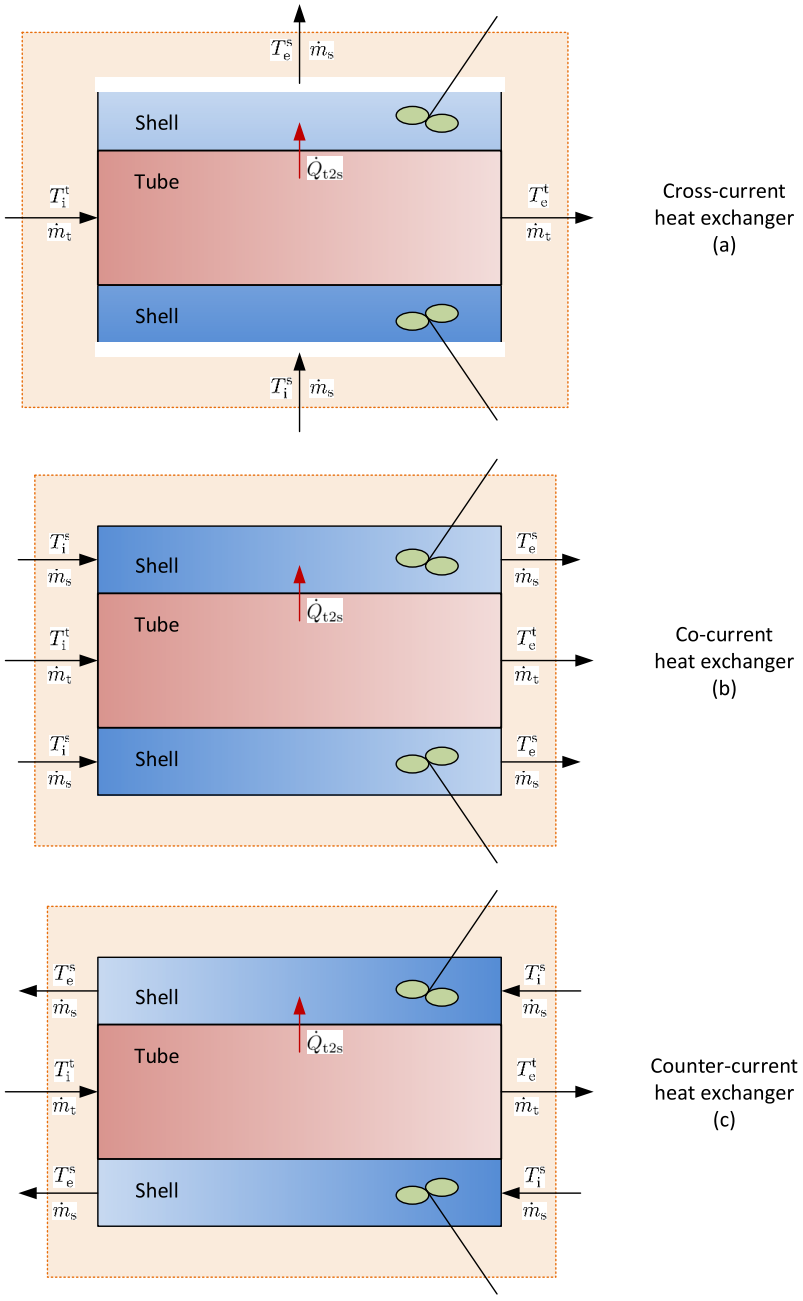


Figure 2.4: Tube-and-shell heat exchangers, with cross-current (a), co-current (b), and counter-current (c) flow configurations (Lie, 2019a).

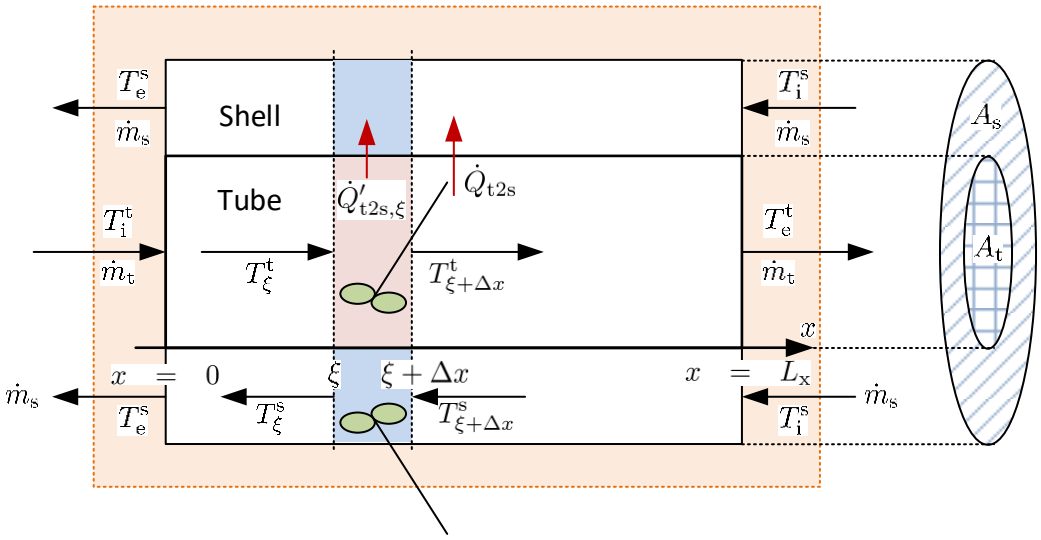


Figure 2.5: Distributed model of a counter-current heat exchanger (Lie, 2019a).

and the shell side can be assumed constant. Therefore the volume work $\dot{W}_v = p\frac{dV}{dt}$ can be set to zero. In other words, the volume work is also neglected. The reduced thermal energy balance for both fluids in the counter-current heat exchanger:

$$\frac{dU}{dt} = \dot{H}_i - \dot{H}_e + \dot{Q}. \tag{2.49}$$

Figure 2.5 shows the model of the counter-current heat exchanger, where the model is treated as a large number of connected homogeneous systems. Then, the third assumption is the perfect mixing in the volume defined by $x \in [\xi, \xi + \triangle x]$. The influent and effluent enthalpy flow rates for the tube side can be expressed as (Lie, 2019a):

$$\dot{H}_i^t = \dot{m}_t \hat{H}_\xi^t \tag{2.50}$$

$$\dot{H}_e^t = \dot{m}_t \hat{H}_{\xi+\triangle x}^t. \tag{2.51}$$

The heat flow $\dot{Q}$ from the tube side to the shell side is denoted as:

$$\dot{Q} = -\dot{Q}_\xi^{t2s}, \tag{2.52}$$

Here, the heat diffusion in the x-direction is neglected. For the interval/segment defined by $x \in [\xi, \xi + \triangle x]$, the heat flow is expressed as:

$$\dot{Q}_{\xi+\triangle x}^{t2s} = \dot{Q}'_{t2s,\xi+\triangle x} \triangle x. \tag{2.53}$$

The internal energy U for each segment is expressed as:

$$U_{\xi+\triangle x}^t = \triangle x \rho_t A_t \hat{U}_{\xi+\triangle x}^t. \tag{2.54}$$

Then, the thermal energy balance for the tube side becomes:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\triangle x \rho_\mathrm{t} A_\mathrm{t} \hat{U}^\mathrm{t}_{\xi+\triangle x}) = \dot{m}_\mathrm{t} \hat{H}^\mathrm{t}_\xi - \dot{m}_\mathrm{t} \hat{H}^\mathrm{t}_{\xi+\triangle x} - \dot{Q}'_{\mathrm{t2s},\xi+\triangle x} \triangle x. \tag{2.55}$$

In the previous expression, ξ is valid for any position $x \in [0, L_x]$, where L_x is the length of the heat exchanger. By generalizing ξ to any position and letting $\triangle x \to 0$, the previous formulation reduces to:

$$\rho_\mathrm{t} A_\mathrm{t} \frac{\partial \hat{U}_\mathrm{t}}{\partial t} = -\dot{m}_\mathrm{t} \frac{\partial \hat{H}_\mathrm{t}}{\partial x} - \dot{Q}'_{\mathrm{t2s}}, \tag{2.56}$$

Here, the overall heat transfer across the layers of the solid material between the two fluids is given as:

$$\dot{Q}'_{\mathrm{t2s}} = \mathscr{U} \wp (T_\mathrm{t} - T_\mathrm{s}). \tag{2.57}$$

With negligible pressure drop and constant density in the fluids, the internal energy and specific enthalpy are approximated as:

$$\partial \hat{U} \approx \hat{c}_v \partial T \tag{2.58}$$

$$\partial \hat{H} \approx \hat{c}_p \partial T, \tag{2.59}$$

which leads to the formulation of a Partial Differential Equation for the tube side:

$$\rho_\mathrm{t} A_\mathrm{t} \hat{c}_{v,\mathrm{t}} \frac{\partial T_\mathrm{t}}{\partial t} = -\dot{m}_\mathrm{t} \hat{c}_{p,\mathrm{t}} \frac{\partial T_\mathrm{t}}{\partial x} - \mathscr{U} \wp (T_\mathrm{t} - T_\mathrm{s}). \tag{2.60}$$

Similarly, the shell side PDE is formed:

$$\rho_\mathrm{s} A_\mathrm{s} \hat{c}_{v,\mathrm{s}} \frac{\partial T_\mathrm{s}}{\partial t} = \dot{m}_\mathrm{s} \hat{c}_{p,\mathrm{s}} \frac{\partial T_\mathrm{s}}{\partial x} + \mathscr{U} \wp (T_\mathrm{t} - T_\mathrm{s}). \tag{2.61}$$

In distributed systems such as this model of the heat exchanger, the dependent variables (temperatures of tube and shell side in this case) are functions of time and one more spatial variable. Since the dynamics of the heat exchanger is faster than the dynamics of the synchronous generator, the energy balance for both fluids can be assumed to be in steady-state, i.e., $\frac{\partial T}{\partial t} = 0$ and both PDEs reduce to ordinary differential equations. The resulting two ODEs form the steady-state model:

$$\frac{\mathrm{d}T_\mathrm{t}}{\mathrm{d}x} = -\frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{t}} \dot{m}_\mathrm{t}} (T_\mathrm{t} - T_\mathrm{s}) \tag{2.62}$$

$$\frac{\mathrm{d}T_\mathrm{s}}{\mathrm{d}x} = -\frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{s}} \dot{m}_\mathrm{s}} (T_\mathrm{t} - T_\mathrm{s}). \tag{2.63}$$

If $\frac{\mathscr{U} \wp}{\hat{c}_p \dot{m}}$ is constant with respect to x, then a linear, space invariant boundary value problem is formed (an ideal heat exchanger model is formed):

$$\frac{\mathrm{d}}{\mathrm{d}x} \begin{pmatrix} T_\mathrm{t} \\ T_\mathrm{s} \end{pmatrix} = \begin{pmatrix} -\frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{t}} \dot{m}_\mathrm{t}} & \frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{t}} \dot{m}_\mathrm{t}} \\ -\frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{s}} \dot{m}_\mathrm{s}} & \frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{s}} \dot{m}_\mathrm{s}} \end{pmatrix} \begin{pmatrix} T_\mathrm{t} \\ T_\mathrm{s} \end{pmatrix}$$

The solution of this model for the known inputs $T_{\mathrm{i}}^{\mathrm{t}}$ and $T_{\mathrm{i}}^{\mathrm{s}}$ is discussed in Problem 8.2 in Lie (2019a), and is found to be

$$T_{\mathrm{t}}(x) = \frac{1}{\alpha_{\mathrm{t}} - \alpha_{\mathrm{s}}} \left[(\alpha_{\mathrm{t}} e^{(\alpha_{\mathrm{s}} - \alpha_{\mathrm{t}})x} - \alpha_{\mathrm{s}}) T_{\mathrm{i}}^{\mathrm{t}} + (\alpha_{\mathrm{t}} - \alpha_{\mathrm{t}} e^{(\alpha_{\mathrm{s}} - \alpha_{\mathrm{t}})x}) T_{\mathrm{e}}^{\mathrm{s}} \right] \tag{2.64}$$

$$T_{\mathrm{s}}(x) = \frac{1}{\alpha_{\mathrm{t}} - \alpha_{\mathrm{s}}} \left[(\alpha_{\mathrm{t}} e^{(\alpha_{\mathrm{s}} - \alpha_{\mathrm{t}})x} - \alpha_{\mathrm{s}}) T_{\mathrm{i}}^{\mathrm{t}} + (\alpha_{\mathrm{t}} - \alpha_{\mathrm{s}} e^{(\alpha_{\mathrm{s}} - \alpha_{\mathrm{t}})x}) T_{\mathrm{e}}^{\mathrm{s}} \right]. \tag{2.65}$$

Here,

$$\alpha_{\mathrm{t}} \triangleq \frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{t}} \dot{m}_{\mathrm{t}}} \tag{2.66}$$

$$\alpha_{\mathrm{s}} \triangleq \frac{\mathscr{U} \wp}{\hat{c}_{p,\mathrm{s}} \dot{m}_{\mathrm{s}}}. \tag{2.67}$$

An implicit expressions for the effluent temperatures $T_{\mathrm{e}}^{\mathrm{t}}$ and $T_{\mathrm{e}}^{\mathrm{s}}$ can be obtained, but an explicit form is more efficient. To formulate the explicit expressions for the effluent temperatures, the definition of Stanton number is utilized. The Stanton number N_{St} is defined as:

$$N_{\mathrm{St}} \triangleq \frac{N_{\mathrm{Nu}}}{N_{\mathrm{Re}} N_{\mathrm{Pr}}} = \frac{\frac{hL}{k}}{\frac{\rho v D}{\mu} \frac{\mu \hat{c}_p}{k}} = \frac{hL}{\rho v D \hat{c}_p} = \frac{hLD}{\rho v D^2 \hat{c}_p} = \frac{hA}{\dot{m} \hat{c}_p}. \tag{2.68}$$

At $x = L_x$ in Eqs. 2.64 and 2.65, α_{t} and α_{s} can be expressed by the respective Stanton number N_{St}:

$$\alpha_{\mathrm{t}} L_x = N_{\mathrm{St}}^{\mathrm{t}}, \tag{2.69}$$

and,

$$\alpha_{\mathrm{s}} L_x = N_{\mathrm{St}}^{\mathrm{s}}. \tag{2.70}$$

Here, $\wp L_x = A_x$ is the heat transfer area. The explicit expressions for the effluent temperatures $T_{\mathrm{t}}(x = 0) = T_{\mathrm{i}}^{\mathrm{t}}$, and $T_{\mathrm{s}}(x = L_x) = T_{\mathrm{i}}^{\mathrm{s}}$ are:

$$T_{\mathrm{e}}^{\mathrm{t}} = \frac{N_{\mathrm{St}}^{\mathrm{t}} (1 - \exp(N_{\mathrm{St}}^{\mathrm{s}} - N_{\mathrm{St}}^{\mathrm{t}})) T_{\mathrm{i}}^{\mathrm{s}} + (N_{\mathrm{St}}^{\mathrm{t}} - N_{\mathrm{St}}^{\mathrm{s}}) \exp(N_{\mathrm{St}}^{\mathrm{s}} - N_{\mathrm{St}}^{\mathrm{t}}) T_{\mathrm{i}}^{\mathrm{t}}}{N_{\mathrm{St}}^{\mathrm{t}} - N_{\mathrm{St}}^{\mathrm{s}} \exp(N_{\mathrm{St}}^{\mathrm{s}} - N_{\mathrm{St}}^{\mathrm{t}})} \tag{2.71}$$

$$T_{\mathrm{e}}^{\mathrm{s}} = \frac{(N_{\mathrm{St}}^{\mathrm{t}} - N_{\mathrm{St}}^{\mathrm{s}}) T_{\mathrm{i}}^{\mathrm{s}} + N_{\mathrm{St}}^{\mathrm{s}} (1 - \exp(N_{\mathrm{St}}^{\mathrm{s}} - N_{\mathrm{St}}^{\mathrm{t}})) T_{\mathrm{i}}^{\mathrm{t}}}{N_{\mathrm{St}}^{\mathrm{t}} - N_{\mathrm{St}}^{\mathrm{s}} \exp(N_{\mathrm{St}}^{\mathrm{s}} - N_{\mathrm{St}}^{\mathrm{t}})}. \tag{2.72}$$

This analytic expression is only valid for the case of α_{t} and α_{s} being independent of x. If any of the overall heat transfer coefficient $\mathscr{U}$, the perimeter $\wp$, or the specific heat capacities $\hat{c}_p$ varies with x, then a numerical solution is necessary. In this work, it is of interest to study the non-ideal case of temperature-dependent heat capacity $\hat{c}_p$, where α_{t} and α_{s} are no longer independent of x. This leads to a nonlinear two-point boundary value problem, where the numerical solution is computationally expensive. It is of interest to investigate an alternative strategy that does not involve an iterative method, namely, a strategy that combines a data-driven model with a mechanistic model.

As a final point, in the thermal model of an air-cooled synchronous generator, $T_{\mathrm{e}}^{\mathrm{t}}$ is the temperature of hot water ($T_{\mathrm{w}}^{\mathrm{h}}$) coming out of the heat exchanger in Eq. 2.41, and $T_{\mathrm{e}}^{\mathrm{s}}$ is the temperature of the cold air ($T_{\mathrm{a}}^{\mathrm{c}}$) coming into the air-cooled synchronous generator model in Eq. 2.42, i.e., the air is flowing in the shell and the water is flowing inside the tube.

2.3 Temperature Dependence of the Specific Heat Capacity

In Eq. 2.12 for the specific enthalpy, $\hat{c}_p$ is considered temperature independent. If $\hat{c}_p$ depends on temperature, then the specific enthalpy would be expressed as:

$$\hat{H} = \hat{H}^{\circ} + \int_{T^{\circ}}^{T} \hat{c}_p(T)dT, \tag{2.73}$$

here, the specific heat capacity is a function of the temperature, unlike Eq. 2.12, where it is a scalar quantity. In Lie (2019a), it is mentioned that the molar heat capacity ($\tilde{c}_p$) is often expressed as a polynomial in absolute temperature (T). In addition, Murphy (2020), state that the molar heat capacity can either be expressed as a polynomial in T or as an empirical power series in T, depending on which function fits the measured data better. In Poling, Prausnitz, and O'Connell (2001), as cited in Lie (2019a), the dimensionless molar heat capacity ($\frac{\tilde{c}_p}{R}$) is given as:

$$\frac{\tilde{c}_p}{R} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4, \tag{2.74}$$

here, R is the ideal gas constant. Also, in Mcbride (1993), the dimensionless molar heat capacity ($\frac{\tilde{c}_p^{\circ}}{R}$) at a standard state of ($298.15\,\mathrm{K}, 1\,bar$) is given as:

$$\frac{\tilde{c}_p^{\circ}}{R} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4, \tag{2.75}$$

Alternatively, in McBride (2002), the dimensionless molar heat capacity ($\frac{\tilde{c}_p^{\circ}}{R}$) at a standard state of ($298.15\,K, 1\,bar$) is given as:

$$\frac{\tilde{c}_p^{\circ}}{R} = a_1 T^{-2} + a_2 T^{-1} + a_3 + a_4 T + a_5 T^2 + a_6 T^3 + a_7 T^4. \tag{2.76}$$

In Eqs. 2.74, 2.75, and 2.76, $a_i (i = 1, \ldots, 7)$ are called the temperature coefficients, and they differ from one reference to another. Apart from this, the molar heat capacity can be converted to the specific heat capacity by multiplying the $\tilde{c}_p$ by the molar mass (also known as the molecular weight). Finally, In Bergman et al. (2011), the thermophysical properties of air at atmospheric pressure is given over the temperature range

of $(100-3000\,\mathrm{K})$, and the thermophysical properties of saturated water is given over the temperature range of $(273.15-647.3\,\mathrm{K})$. In this work, the experimental data from Bergman et al. (2011), is fitted over a relevant temperature range, and compared with the power series of McBride (2002).

Furthermore, in this work, the specific heat capacities in the expression for $\dot{Q}_{\mathrm{w2a}}$ in Eq.2.46 will be kept constant since it is not necessary to compute $\dot{Q}_{\mathrm{w2a}}$ to solve the thermal model of the generator.

2.4 Analytic vs. Numeric Solution of the Counter-current Heat Exchanger Model

In Section 2.2, an analytical solution (Eqs 2.64 and 2.65) was obtained to the linear, space invariant boundary value problem of an ideal heat exchanger model (Lie, 2019a). However, when extending the thermal model of an air-cooled synchronous generator from Lie (2018a) with a more realistic heat exchanger model, specifically, a model with temperature dependence in heat capacity and/or heat transfer of air/water, the specific heat capacity is no longer a scalar quantity, but a function of temperature, which leads to a nonlinear two-point boundary value problem. In Section 2.3, the temperature dependence in the specific heat capacity was discussed, and it was found that the $\hat{c}_p(T)$ can either be expressed as a polynomial in T or as an empirical power series in T (Lie, 2019a; Murphy, 2020). In this section, the analytic solution of the ideal heat exchanger model and the numeric solution of the non-ideal model are compared.[1]

2.4.1 The analytic solution (exact solution)

For an ideal heat exchanger model (for the case of temperature independence in heat capacity and/or heat transfer of air/water), a linear, space invariant boundary value problem was formed, to which the analytical solution was expressed in Eqs 2.64 and 2.65 as:

$$T_{\mathrm{t}}(x) = \frac{1}{\alpha_{\mathrm{t}}-\alpha_{\mathrm{s}}}\left[(\alpha_{\mathrm{t}}e^{(\alpha_{\mathrm{s}}-\alpha_{\mathrm{t}})x}-\alpha_{\mathrm{s}})T_{\mathrm{i}}^{\mathrm{t}}+(\alpha_{\mathrm{t}}-\alpha_{\mathrm{t}}e^{(\alpha_{\mathrm{s}}-\alpha_{\mathrm{t}})x})T_{\mathrm{e}}^{\mathrm{s}}\right]$$

$$T_{\mathrm{s}}(x) = \frac{1}{\alpha_{\mathrm{t}}-\alpha_{\mathrm{s}}}\left[(\alpha_{\mathrm{t}}e^{(\alpha_{\mathrm{s}}-\alpha_{\mathrm{t}})x}-\alpha_{\mathrm{s}})T_{\mathrm{i}}^{\mathrm{t}}+(\alpha_{\mathrm{t}}-\alpha_{\mathrm{s}}e^{(\alpha_{\mathrm{s}}-\alpha_{\mathrm{t}})x})T_{\mathrm{e}}^{\mathrm{s}}\right].$$

[1] All the code presented in Section 2.4 are part of the Jupyter Notebook "Counter-current Heat Exchanger Models" in Appendix B.1.

Then, by utilizing the definition of Stanton number, an explicit expressions for the effluent temperatures were obtained in Eqs 2.71 and 2.72 as:

$$T_{\mathrm{e}}^{\mathrm{t}} = \frac{N_{\mathrm{St}}^{\mathrm{t}}(1-\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}}))T_{\mathrm{i}}^{\mathrm{s}} + (N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}})\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})T_{\mathrm{i}}^{\mathrm{t}}}{N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}}\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})}$$

$$T_{\mathrm{e}}^{\mathrm{s}} = \frac{(N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}})T_{\mathrm{i}}^{\mathrm{s}} + N_{\mathrm{St}}^{\mathrm{s}}(1-\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}}))T_{\mathrm{i}}^{\mathrm{t}}}{N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}}\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})}.$$

For air and water, the Stanton numbers are given in Eqs. 2.44 and 2.45 as:

$$N_{\mathrm{St}}^{\mathrm{w}} = \frac{\mathscr{U}A_{\mathrm{x}}}{\hat{c}_{p,\mathrm{w}}\dot{m}_{\mathrm{w}}}$$

$$N_{\mathrm{St}}^{\mathrm{a}} = \frac{\mathscr{U}A_{\mathrm{x}}}{\hat{c}_{p,\mathrm{a}}\dot{m}_{\mathrm{a}}}.$$

When solving the dynamic thermal model of the generator using the DAE solvers of Julia or OpenModelica, for each time step, the previous expressions for $T_{\mathrm{e}}^{\mathrm{t}}$ and $T_{\mathrm{e}}^{\mathrm{s}}$ ($T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$) are evaluated relatively fast compared to the numeric solution of the heat exchanger model, where a numerical method is applied to solve the two-point boundary value problem with respect to x, where $x \in [0, L_x]$. In addition, as mentioned in Section 2.1, it is not necessary to compute $T_{\mathrm{w}}^{\mathrm{h}}$ to solve the thermal model of the generator.

2.4.2 Implementation of the analytic model in Julia

By substituting the explicit expression for $T_{\mathrm{e}}^{\mathrm{s}}$ in the expressions for $T_{\mathrm{t}}(x)$ and $T_{\mathrm{s}}(x)$, it is possible to evaluate the temperatures of air and water for the interval $x \in [0, L_x]$, where L_x is the length of the heat exchanger. In the Julia code below,[2] the expressions for $T_{\mathrm{e}}^{\mathrm{s}}$, $T_{\mathrm{t}}(x)$, and $T_{\mathrm{s}}(x)$ are wrapped in a function to make the comparison between the analytic and numeric solutions simpler (the expressions are computed in the lines $9-11$).

[2]The package minted with the syntax highlighting library Pygments was used to print out this code snippet (Poore, 2020).

Symbol	Description	Value	Unit
	Inputs u		
$T_{\mathrm{w}}^{\mathrm{c}}$	Cold water temperature	3.8	°C
$T_{\mathrm{a}}^{\mathrm{h}}$	Hot air temperature	39.12	°C
$\dot{m}_{\mathrm{w}}$	Water mass flow rate	53.9	kg/s
$\dot{m}_{\mathrm{a}}$	Air mass flow rate	49.2	kg/s
	Parameters θ		
$\hat{c}_{p,\mathrm{a}}$	Specific heat capacity, air	1.15	kJ/kg/K
$\hat{c}_{p,\mathrm{w}}$	Specific heat capacity, water	4.2	kJ/kg/K
$\mathscr{U}A_{\mathrm{x}}$	Heat transfer, air to water	44.46	kW/K
hA_{ax}	Heat transfer, air side of the heat exchanger	55.6	kW/K
hA_{wx}	Heat transfer, water side of the heat exchanger	222	kW/K

Table 2.3: Operating conditions for the ideal heat exchanger model.

```
function hex_a(inputs,par,x)
    Twc, Tah, mdw, mda = inputs
    UAx, chpw, chpa = par
    # Stanton numbers for air and water.
    NSta = UAx/chpa/mda # Stanton number for air, -
    NStw = UAx/chpw/mdw # Stanton number for water, -
    NStd = NStw - NSta
    #
    Tac = (NStd*Tah + NSta*(1-exp(-NStd))*Twc)/(NStw-NSta*exp(-NStd))
    Tw(x) = ((NStw*exp(-NStd*x)-NSta)*Twc+(NStw - NStw*exp(-NStd*x))*Tac)/NStd
    Ta(x) = ((NSta*exp(-NStd*x)-NSta)*Twc+(NStw - NSta*exp(-NStd*x))*Tac)/NStd
    return [Tw(x), Ta(x)]
end
```

Here, the function (**hex_a**) takes the inputs, the parameters, and the interval x as arguments (The main reason to pass them as arguments is to avoid the problems associated with global variables), and return the temperature of air and water along the heat exchanger length.

Figure 2.6 shows the simulation result for the temperature profile along the length of ideal heat exchanger model, where the model operating conditions are specified in Table 2.3 (Lie, 2018b).

2.4.3 The numeric solution (approximate solution)

A steady-state model of a counter-current heat exchanger, in general, leads to a two-point boundary value problem, which can be solved numerically. In this work, the numerical solution is obtained by utilizing the boundary value problem (BVP) solvers available in

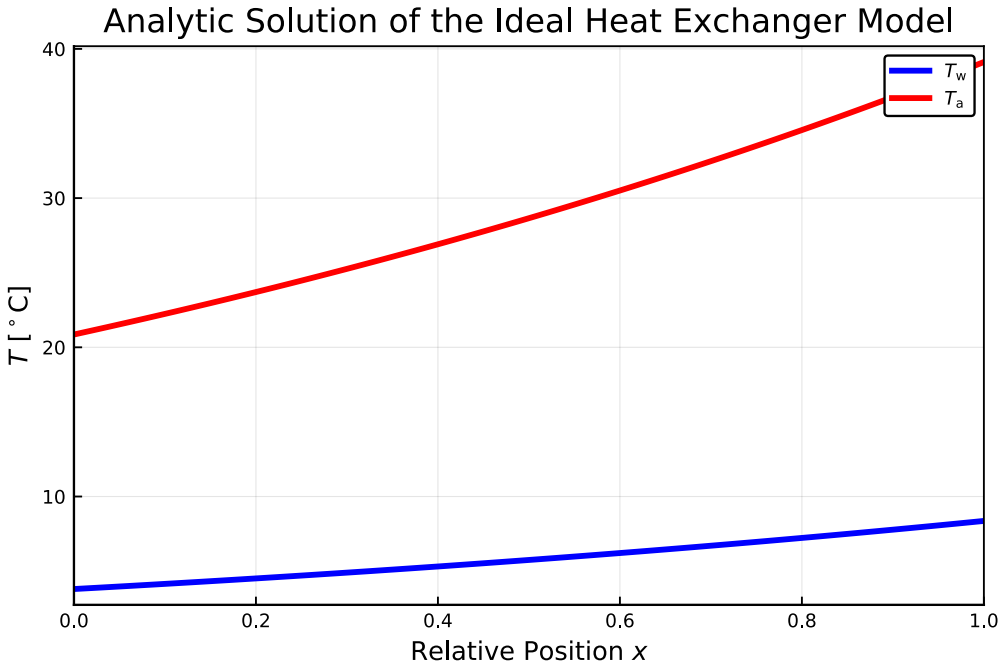


Figure 2.6: Analytic solution of the ideal heat exchanger model.

DifferentialEquations.jl Julia package (Rackauckas and Nie, 2017). Alternatively, the component package BoundaryValueDiffEq.jl of the DifferentialEquations ecosystem can be used. In addition, there are several methods that can be used in the solver, these methods are categorized based on the problem type:

1. BVProblem, for general boundary conditions.
    - Shooting: in this method, the problem is treated as an initial value problem where the initial conditions are varied until the boundary conditions are met.
    - GeneralMIRK4: is described in the documentation as a 4th order collocation method.
2. TwoPointBVProblem, for boundary conditions that are defined at the beginning and the end of the integration interval.
    - MIRK4: is similar to GeneralMIRK4, a 4th order collocation method, but it is only compatible with TwoPointBVProblem problem type.

In this work, the execution speed of each of the three methods was compared using the package BenchmarkTools.jl, and based on the benchmark results shown in Table 2.4, the method Shooting achieved the fastest execution speed, and thus will be used in the comparison with the analytic solution or in the comparison with the hybrid model (The combination of the mechanistic and the data-driven models).

Method	Median time (ms)	Mean time (ms)
Shooting	20.064	21.741
GeneralMIRK4	846.158	849.103
MIRK4	867.653	873.557

Table 2.4: Benchmark results for the numeric solution of a non-ideal heat exchanger model. To make the comparison fair, the step size was fixed to the value of 0.01 (an optional argument **dtmax** was passed to the solver for the Shooting method). Otherwise, the Shooting method would perform 20 times better.

2.4.3.1 Model implementation of the ideal case of temperature independence in the specific heat capacities of air and water in the heat exchanger

The model of the ideal case of temperature independence in the specific heat capacities of air and water is implemented in the code below.

```
1  function hex_n_Cp_T_indep(inputs,par,x)
2      # The function for the ODE.
3      function hex_i!(dy,y,par,x)
4          # Interior of heat exchanger with temperature independent heat capacity
5          mdw, mda = par[3],par[4]
6          UAx, cp_w, cp_a = par[5],par[6],par[7]
7          #
8          Tw,Ta = y
9          dy[1] = -UAx/(mdw*cp_w)*(Tw - Ta)
10         dy[2] = -UAx/(mda*cp_a)*(Tw - Ta)
11     end
12     #
13     # Boundaries of heat exchanger with temperature independent heat capacity
14     function hex_b!(residual, y, par, x)
15         Twc,Tah = par[1],par[2]
16         residual[1] = y[1][1]-Twc        # y[1] is the beginning of the spatial span
17         residual[2] = y[end][2]-Tah        # y[end] is the ending of the spatial
               ↪ span
18     end
19     #
20     u0 = [0.0, 0.0] # The initial condition.
21     prob_hex = BVProblem(hex_i!, hex_b!, u0, x, [inputs  par])
22     sol_hex = solve(prob_hex, Shooting(Vern7()), dtmax=0.01) # Three solvers are
           ↪ available. Shooting is the fastest.
23     return sol_hex # dtmax: Maximum dt for adaptive timestepping.
24 end
```

Here, the main function (**hex__n__Cp__T__indep**) consists of the initial condition (line 20), the two-point boundary value problem definition (line 21), the solver (line 22), and two sub-functions:

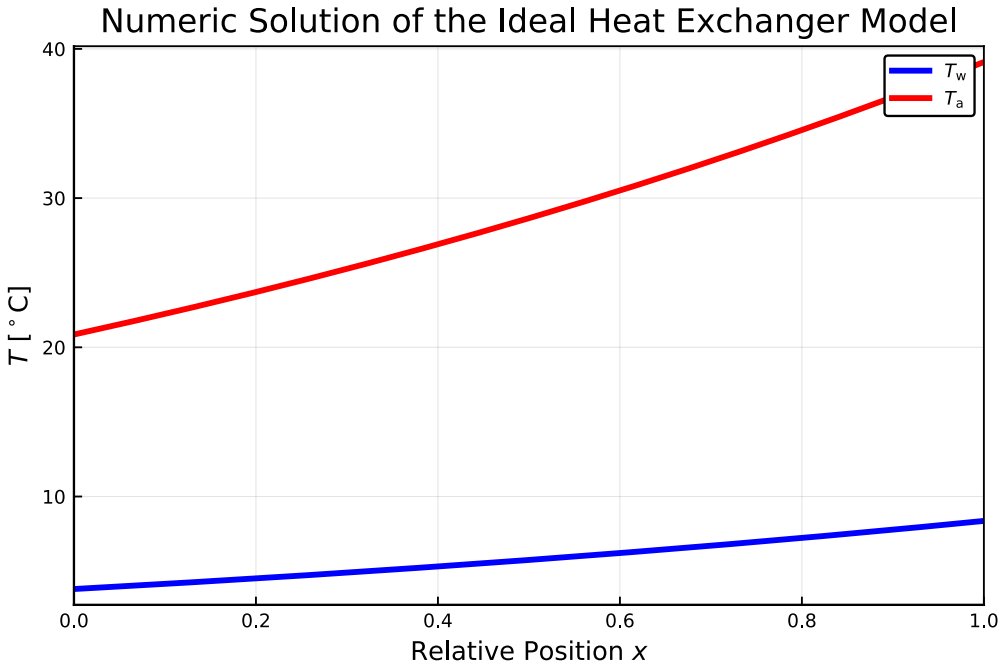


Figure 2.7: Numeric solution of the ideal heat exchanger model.

1. **hex_i!**: The function for the ODE. The parameters of the heat exchanger ($\hat{c}_{p,\mathrm{a}}$, $\hat{c}_{p,\mathrm{w}}$, and $\mathscr{U}A_{\mathrm{x}}$) are passed to this function, in addition to the inputs and $\dot{m}_{\mathrm{a}}$ and $\dot{m}_{\mathrm{w}}$.
2. **hex_b!**: The boundary condition function. This function takes the boundary conditions $T_{\mathrm{w}}^{\mathrm{c}}$ and $T_{\mathrm{a}}^{\mathrm{h}}$.

In addition, the main function takes the inputs, the parameters, and the interval x as arguments and returns the temperature profile along the heat exchanger length in a similar manner to the analytic model. Also, depending on which type of problem was chosen (BVProblem or TwoPointBVProblem) in the line 21, the appropriate method should be passed to the solver in line 22. It is important to add that, the third optional argument in the solver **dtmax** is the maximum step size for adaptive time-stepping, but in the context of the heat exchanger problem, it refers to spatial-stepping instead. By calling the main function, it is possible to obtain the temperature profile along the heat exchanger length. Figure 2.7 shows the Numeric solution of the ideal heat exchanger model.

2.4.3.2 Comparison between the analytic solution and the numeric solution of the ideal case of temperature independence in the specific heat capacities of air and water

The main goal for implementing the numeric solver of the ideal case of temperature independence in the specific heat capacities of air and water is to check and see if the

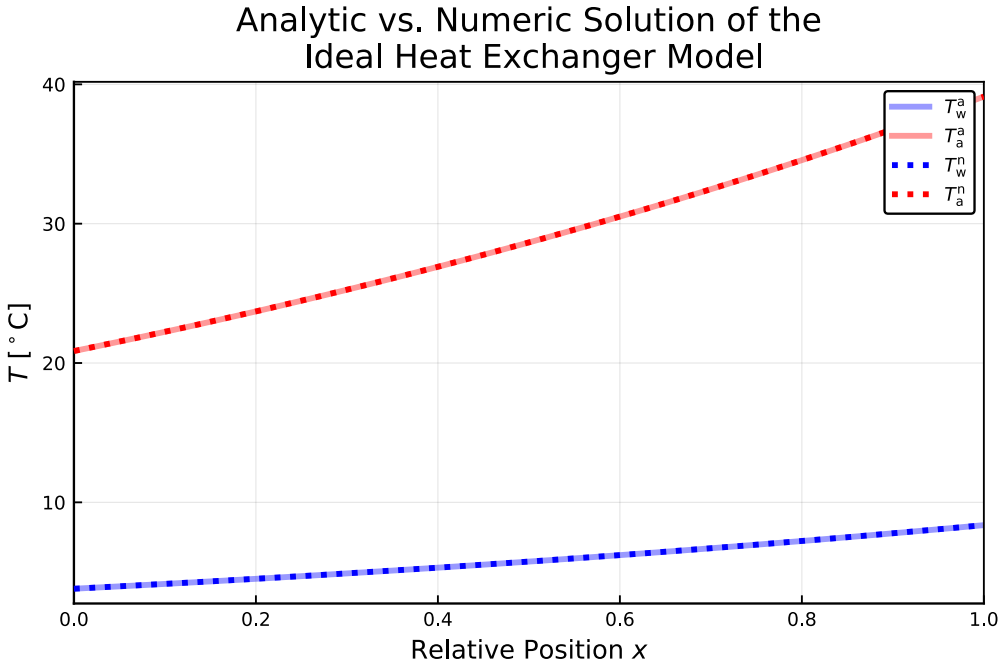


Figure 2.8: Analytic vs. numeric solution of the ideal heat exchanger model.

Model	Median time (ms)	Mean time (ms)
The analytic solution	0.009	0.01063
The numeric solution	4.487	5.310

Table 2.5: Benchmark results for the ideal heat exchanger model.

analytic solution in 2.6 matches the numeric solution of the ideal case. Figure 2.8 shows a comparison between the analytic and the numeric solutions of the ideal heat exchanger model, and since the solutions match, then it is safe to proceed to the next step, and obtain the solution for the non-ideal case of temperature dependence in the specific heat capacities of air and water.

Before proceeding to the non-ideal heat exchanger model, it is of interest to compare the computation time between the two solutions in Fig. 2.8. The benchmark results are summarized in Table 2.5. Moreover, the benchmark results indicate that when solving the dynamic thermal model of the generator, for each time step, the numeric model obtain a solution in a mean time of $5.310\,\text{ms}$, while the analytic expressions are evaluated in a mean time of $10.630\,\mu\text{s}$. Besides, when solving the dynamic thermal model of the generator using the analytical solution of the heat exchanger model, there is no need to evaluate the temperature along the heat exchanger length, and the explicit expressions for the effluent temperatures that were obtained in Eqs 2.71 and 2.72 should be used instead, i.e., the analytic solution is much faster.

2.4.3.3 Model implementation of the non-ideal case of temperature dependence in the specific heat capacities of air and water in the heat exchanger

The model of the non-ideal case of temperature dependence in the specific heat capacities of air and water is implemented in the code below. The model implementation is the same as the previous code for the ideal case of temperature independence in the specific heat capacities of air and water except for, of course, the specific heat capacities are no longer constant parameters, which can be observed in the lines 9 and 10 in the code where the specific heat capacities are functions of temperature.

```
1  function hex_n_Cp_T_dep(inputs,par,x)
2      # The function for the ODE.
3      function hex_i!(dy,y,par,x)
4          # Interior of heat exchanger with temperature dependent heat capacity
5          mdw, mda = par[3],par[4]
6          UAx, cp_w, cp_a = par[5],par[6],par[7]
7          #
8          Tw,Ta = y
9          dy[1] = -UAx/(mdw*cp_w(Tw))*(Tw - Ta)
10         dy[2] = -UAx/(mda*cp_a(Ta))*(Tw - Ta)
11     end
12     #
13     # Boundaries of heat exchanger with temperature dependent heat capacity
14     function hex_b!(residual, y, par, x)
15         Twc,Tah = par[1],par[2]
16         residual[1] = y[1][1]-Twc        # y[1] is the beginning of the spatial span
17         residual[2] = y[end][2]-Tah        # y[end] is the ending of the spatial
           ↪ span
18     end
19     #
20     u0 = [0.0, 0.0] # The initial condition.
21     prob_hex = BVProblem(hex_i!, hex_b!, u0, x, [inputs  par])
22     sol_hex = solve(prob_hex, Shooting(Vern7()), dtmax=0.01) # Three solvers are
       ↪ available. Shooting is the fastest.
23     return sol_hex # dtmax: Maximum dt for adaptive timestepping.
24 end
```

At the beginning of this work, the temperature dependence in the specific heat capacities was implemented using the expressions for the dimensionless molar heat capacity ($\frac{\tilde{c}_p^\circ}{R}$) in Mcbride (1993), and McBride (2002), but later on, it was implemented by fitting a polynomial to the experimental data in Bergman et al. (2011), since the coefficients in Mcbride (1993) and McBride (2002) were obtained by fitting a polynomial/power-series over a large interval that is irrelevant to this work (The power series in McBride (2002), is valid over the ranges of 273.15 K to 373.15 K for water, and 200 K to 1000 K for air, which is irrelevant to this work). In Julia, it is possible to construct a polynomial of degree n

that goes through the points specified by x and y of the experimental data (x,y) using **polyfit** of the Julia package Polynomials.jl, as shown below.

```
using Polynomials
#
# Data for water and air from Incropera et al (temperature and specific heat capacities)
T_w_I = [273.15, 275.0, 280.0, 285.0, 290.0, 295.0, 300.0, 305.0, 310.0, 315.0, 320.0, 325.0, 330.0, 335.0, 340.0, 345.0, 350.0, 355.0, 360.0, 365.0, 370.0].-273.15
cp_w_I = [4.217,4.211,4.198,4.189,4.184,4.181,4.179,4.178,4.178,4.179,4.18 ,4.182,4.184,4.186,4.188,4.191,4.195,4.199,4.203,4.209,4.214]
T_a_I = [100.0, 150.0, 200.0, 250.0, 300.0, 350.0, 400.0].-273.15
cp_a_I = [1.032, 1.012, 1.007, 1.006, 1.007, 1.009, 1.014]
#
# polyfit(x, y, n)
#
polynomial_w = polyfit(T_w_I, cp_w_I, 7) # Cp is expressed as a polynomial in temperature for water.
polynomial_a = polyfit(T_a_I, cp_a_I, 5) # Cp is expressed as a polynomial in temperature for air.
```

Also, the following function was used to select the proper degree for the polynomial without overfitting the data, for example, Fig. 2.9 shows the output of the function for the water experimental data in Bergman et al. (2011), where it can be seen that a 7th order polynomial fits the data well.

```
# This function is used to select the proper polynomial degree
function findbestfit(y,x)
    RMSE =  Array{Float64}(undef, length(y)-1);
    for i in 1:(length(y)-1)
        p=polyfit(x, y,i)
        RMSE[i]=sqrt(sum((y-p(x)).^2) / (length(y))) # Root Mean Square Error
        # println("Degree ", i," with RMSE: ", RMSE[i])
    end
    plot(1:(length(y)-1),RMSE,ylabel="RMSE",xlabel="Degree of Polynomial")
end
# Example: findbestfit(cp_w_I,T_w_I)
```

The temperature dependence in the specific heat capacities was implemented using the dimensionless molar heat capacity in McBride (2002) in the following code. To convert the dimensionless molar heat capacities to specific heat capacities, the expressions were multiplied by the universal gas constant and divided over the molar mass in lines 9 and 10.

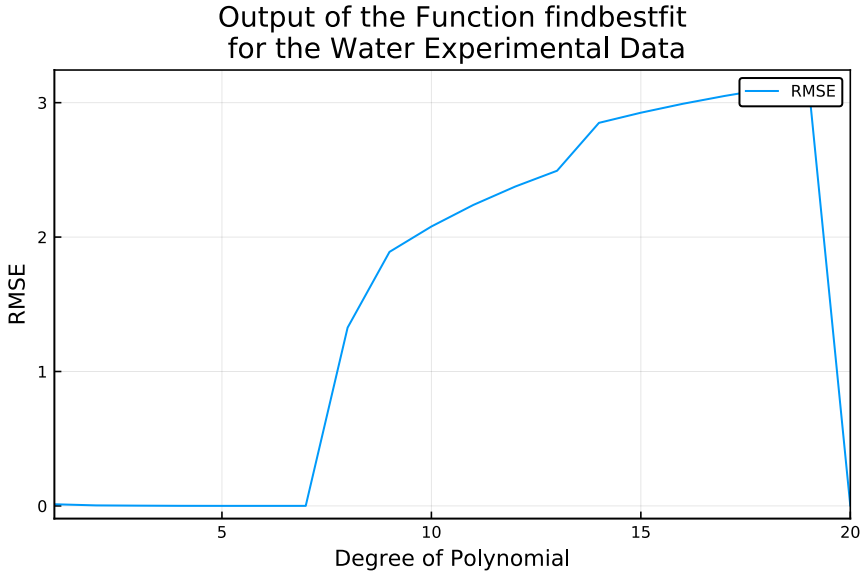


Figure 2.9: Output of the function **findbestfit** for the water experimental data. The function **findbestfit** use the RMSE as a measure to help select the degree of the polynomial, while avoiding overfitting the data.

```
# H2O (L)
c_w = [1.326371304e+09, -2.448295388e+07, 1.879428776e+05, -7.678995050e+02,
 ↪  1.761556813e+00, -2.151167128e-03, 1.092570813e-06]
# AIR CALCULATED FROM INGREDIENTS %N2=78.084 %O2=20.9476 %AR=0.9365 %CO2=0.0319
c_a = [1.009950160e+04, -1.968275610e+02, 5.009155110e+00, -5.761013730e-03,
 ↪  1.066859930e-05, -7.940297970e-09, 2.185231910e-12]
const Mw = 18.01528 # Molecular weight of Water, g/mol
const Ma = 28.96518 # Molecular weight of Air, g/mol ## AIR CALCULATED FROM
 ↪  INGREDIENTS %N2=78.084 %O2=20.9476 %AR=0.9365 %CO2=0.0319
const R = 8.31446261815324 # The Universal gas constant, J*K-1*mol-1
#
Cp_w_nasa(x) = (R/Mw)*(c_w[1]*(x+273.15)^-2 + c_w[2]*(x+273.15)^-1 + c_w[3] +
 ↪  c_w[4]*(x+273.15) + c_w[5]*(x+273.15)^2 + c_w[6]*(x+273.15)^3 +
 ↪  c_w[7]*(x+273.15)^4) # Valid for 273.15 - 373.15 K
Cp_a_nasa(x) = (R/Ma)*(c_a[1]*(x+273.15)^-2 + c_a[2]*(x+273.15)^-1 + c_a[3] +
 ↪  c_a[4]*(x+273.15) + c_a[5]*(x+273.15)^2 + c_a[6]*(x+273.15)^3 +
 ↪  c_a[7]*(x+273.15)^4) # Valid for 200 - 1000 K
```

Figure 2.10 shows a comparison between the polynomials constructed from the experimental data of Bergman et al. (2011) and the specific heat capacities functions obtained using the dimensionless molar heat capacity in McBride (2002). Furthermore, it can be seen in Fig.2.10 that the polynomials constructed from Bergman et al. (2011) fit the experimental data better than the specific heat capacities functions obtained using the dimensionless molar heat capacity in McBride (2002)

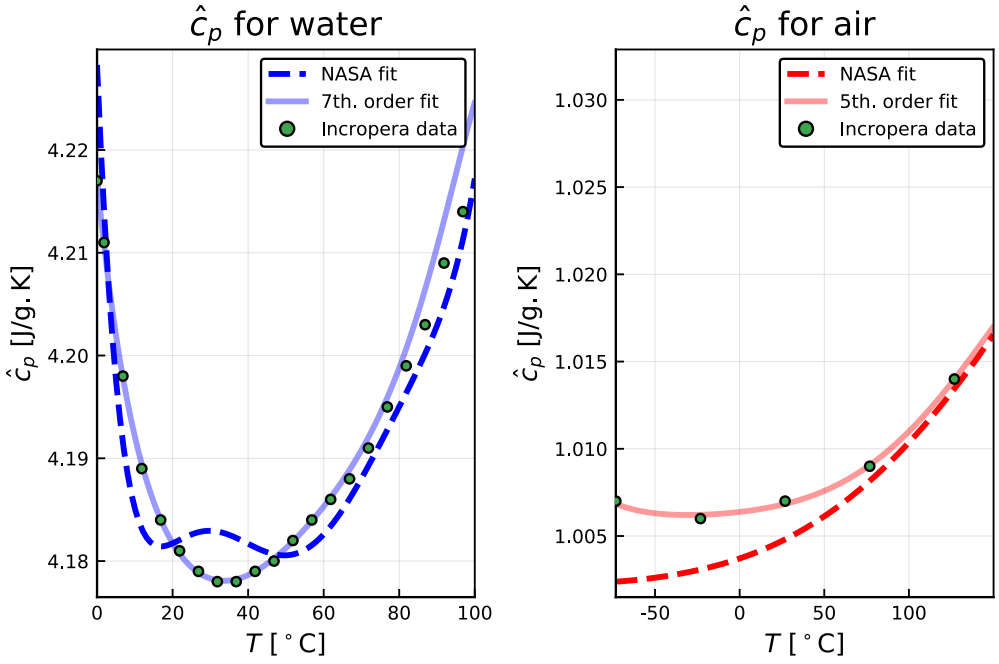


Figure 2.10: Comparison between the polynomials constructed from the experimental data of Bergman et al. (2011) and the specific heat capacities functions obtained using the dimensionless molar heat capacity in McBride (2002).

During this work, the supervisor, wrote the following functions that fit the experimental data of Bergman et al. (2011), these functions achieve a good correlation while using the lowest polynomial order possible. In addition, the coefficients in the functions are scaled as close as possible to one.

```
# BL heat capacities
function cp_a(x) # For air.
    cp_a0 = 1.007
    y = x+273.15
    y0 = 300
    return cp_a0*(299.94 +
     ↪  1.008*(y-y0)+1.2e-4*(y-y0)^2+5.9e-7*(y-y0)^3+2.56e-9*(y-y0)^4)/y
end
#
function cp_w(x) For water.
    cp_w0 = 4.179
    y = x+273.15
    y0 = 300
    return cp_w0*(0.9998-6.6e-5*(y-y0) +
     ↪  6.39e-6*(y-y0)^2-1.145e-7*(y-y0)^3+8.9e-10*(y-y0)^4)
end
```

Moreover, Fig. 2.11 shows a comparison between the supervisor functions and the func-

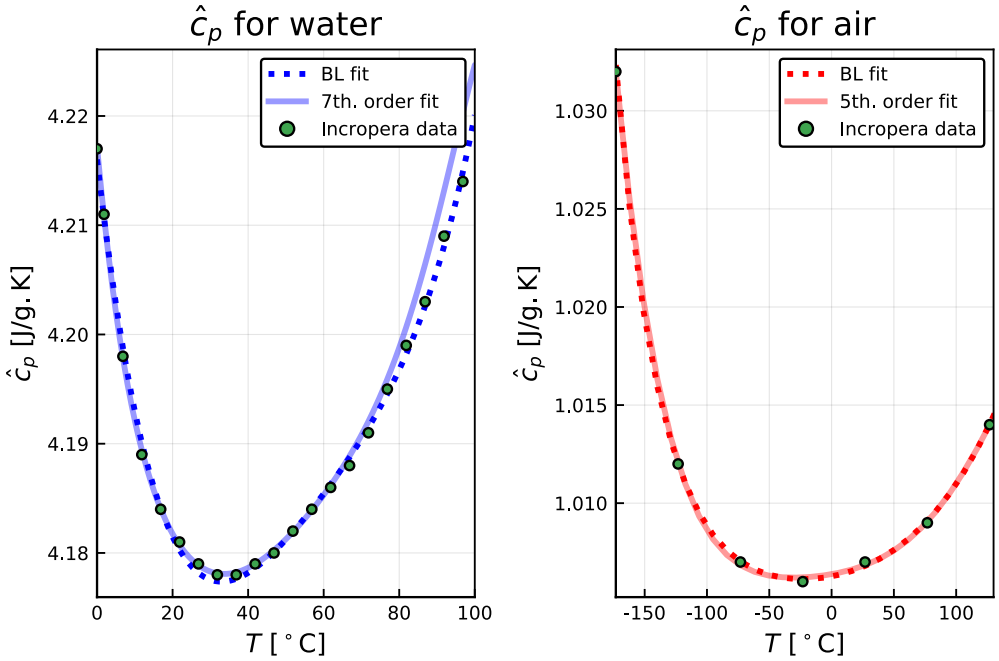


Figure 2.11: Comparison between the polynomials constructed using **polyfit** and the functions constructed by the supervisor (BL fit) for the experimental data in Bergman et al. (2011).

tions obtained using **polyfit**.

Finally, the supervisor functions were used to obtain the numeric solution of the non-ideal case of temperature dependence in the specific heat capacities of air and water for the heat exchanger model. Figure 2.12 shows the numeric Solution of the non-ideal heat exchanger model when $\hat{c}_p(T)$.

2.4.3.4 Comparison between the analytic solution and the numeric solution of the non-ideal case of temperature dependence in the specific heat capacities of air and water

Figure 2.13 shows a comparison between the analytic solution and the numeric solution when $\hat{c}_p(T)$.

Also, it is of interest to compare the computation time between the two numeric solutions. The benchmark results are summarized in Table 2.6, which shows the non-ideal case of temperature dependence almost five times slower than the ideal case.

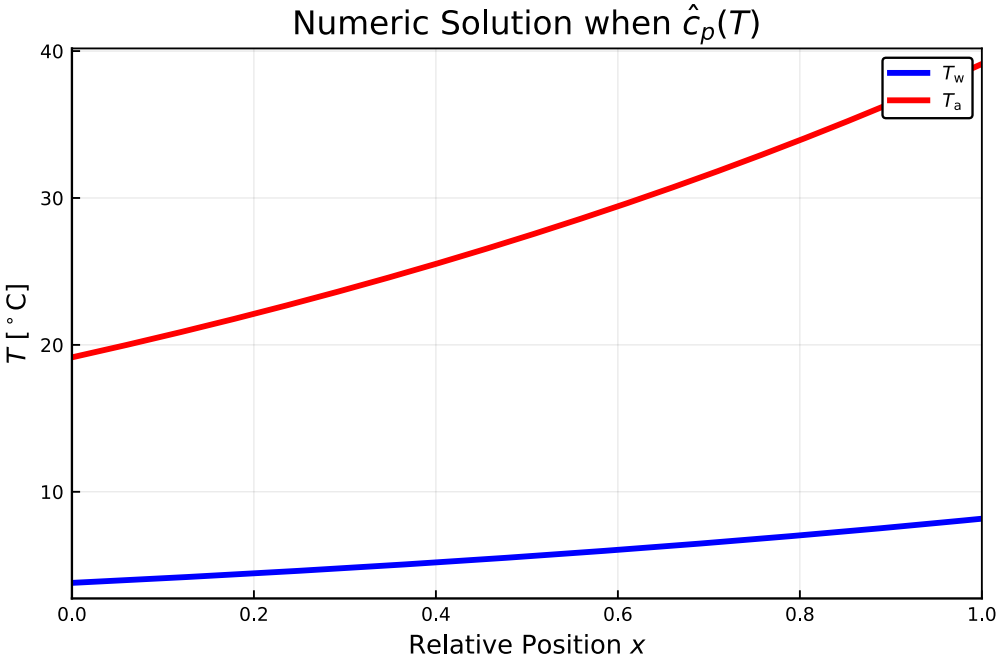


Figure 2.12: Numeric solution of the non-ideal heat exchanger model when $\hat{c}_p(T)$.

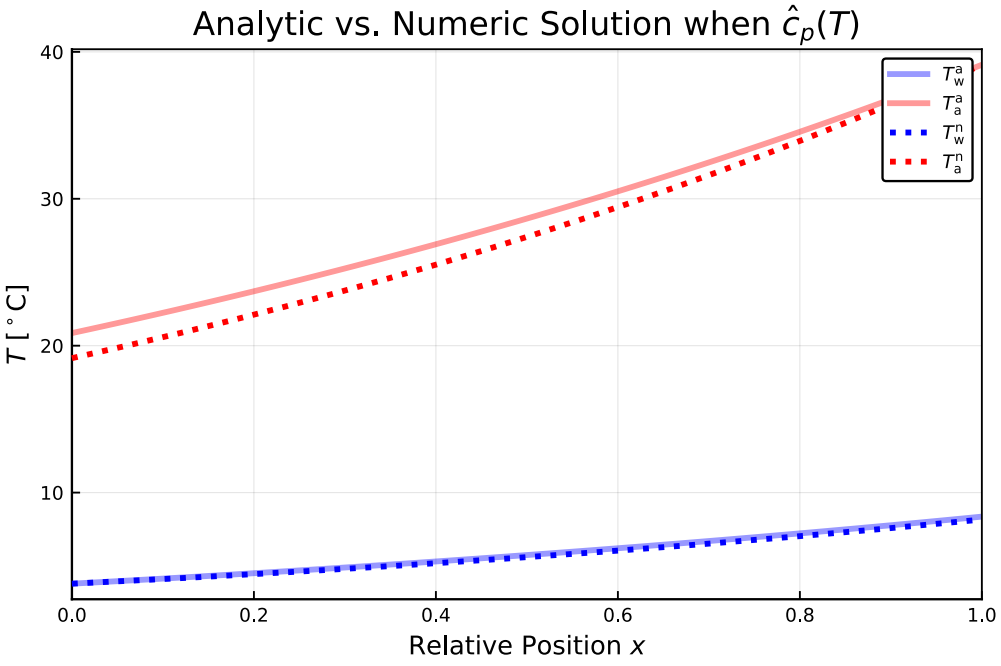


Figure 2.13: Analytic solution vs. numeric solution when $\hat{c}_p(T)$.

Model	Median time (ms)	Mean time (ms)
The numeric solution when $\hat{c}_p$	3.990	4.798
The numeric solution when $\hat{c}_p(T)$	19.061	20.630

Table 2.6: Benchmark results for the non-ideal heat exchanger model.

2.5 Implementation of the Thermal Model of an Air-Cooled Synchronous Generator in Julia with Ideal and Non-Ideal Heat Exchanger Models

The DAE model of the air-cooled synchronous generator is implemented in Appendix B.2 as an ODE function called **DAE_Model_Synchronous_Generator**. Furthermore, the ODE function is part of the **DAEProblem** definition which is solved by a DAE solver (**IDA**) from the Julia package Sundials.jl. In addition, the previous heat exchanger models where implemented in the lines $98-121$ of the ODE function as shown in the code below.

```
98  ## Heat exchanger temperatures
99  u_hex = [Twc Tah mdw mda]
100 if f == hex_a
101     xspan_a = collect(0:0.01:1)  # step size for the analytic model is specified
        ↪ here.
102     par_hex_a = [UAx chpw chpa]
103     sol_analytic = map(x -> hex_a(u_hex, par_hex_a, x), xspan_a) |> vec2vec
104     out[30] = - Twh + sol_analytic[1][end]
105     out[31] = - Tac + sol_analytic[end][1]
106 elseif f ==  hex_n_Cp_T_dep
107     xspan_n = (0.0, 1.0)
108     par_hex_n_Cp_T_dep = [UAx cp_w cp_a] # Here, cp_w and cp_a are functions.
109     sol_numeric_Cp_T_dep = hex_n_Cp_T_dep(u_hex, par_hex_n_Cp_T_dep, xspan_n);
110     out[30] = - Twh + sol_numeric_Cp_T_dep.u[end][1]
111     out[31] = - Tac + sol_numeric_Cp_T_dep.u[1][end]
112 elseif f ==  hex_n_Cp_T_indep
113     xspan_n = (0.0, 1.0)
114     par_hex_n_Cp_T_indep = [UAx chpw chpa] # Here, chpw and chpa are constants.
115     sol_numeric_Cp_T_indep = hex_n_Cp_T_indep(u_hex, par_hex_n_Cp_T_indep, xspan_n);
116     out[30] = - Twh +sol_numeric_Cp_T_indep.u[end][1]
117     out[31] = - Tac + sol_numeric_Cp_T_indep.u[1][end]
118 else
119     InterruptException()
120     println("Invalid Heat-Exchanger function name")
121 end
```

In this code, it is important to highlight that the heat exchanger model is solved for each time step in the DAE solver to obtain $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$. Next the following models are

Model	Median time (ms)	Mean time (ms)
Model 1	5.387	6.604
Model 2	2428	2398
Model 3	9796	9796

Table 2.7: Benchmark results for the thermal model of an air-cooled synchronous generator.

compared:

1. Model 1: Air-cooled synchronous generator with the analytic heat exchanger model.
2. Model 2: Air-cooled synchronous generator with the numeric ideal heat exchanger model.
3. Model 3: Air-cooled synchronous generator with the numeric non-ideal heat exchanger model.

In model 1, the explicit expressions of Eqs 2.71 and 2.72 are used, whereas in model 2 and model 3, the thermal model of an air-cooled synchronous generator have to be solved in two stages, in the first stage, the two-point boundary value problem of the heat exchanger model is solved numerically (using one of the BVP solvers mentioned earlier), and in the second stage, the DAE model of the air-cooled synchronous generator is solved using the DAE solver (IDA).

Then, the generator model outputs y (the generator metal temperatures) are selected for comparison, and as expected,[3] Fig. 2.14 shows no difference between the metal temperatures of model 1 and model 2 (since the numeric solution of the ideal heat exchanger is the same as the analytic), on the other hand, the temperature dependence in the specific heat capacities in model 3 decreased the metal temperatures about 3.3°C as shown in Fig. 2.15

Finally, Table 2.7 summarizes the execution speed:

[3]There should not be any difference in the generator metal temperatures since analytic and the numeric solution of the ideal heat exchanger model matched in Fig. 2.8.

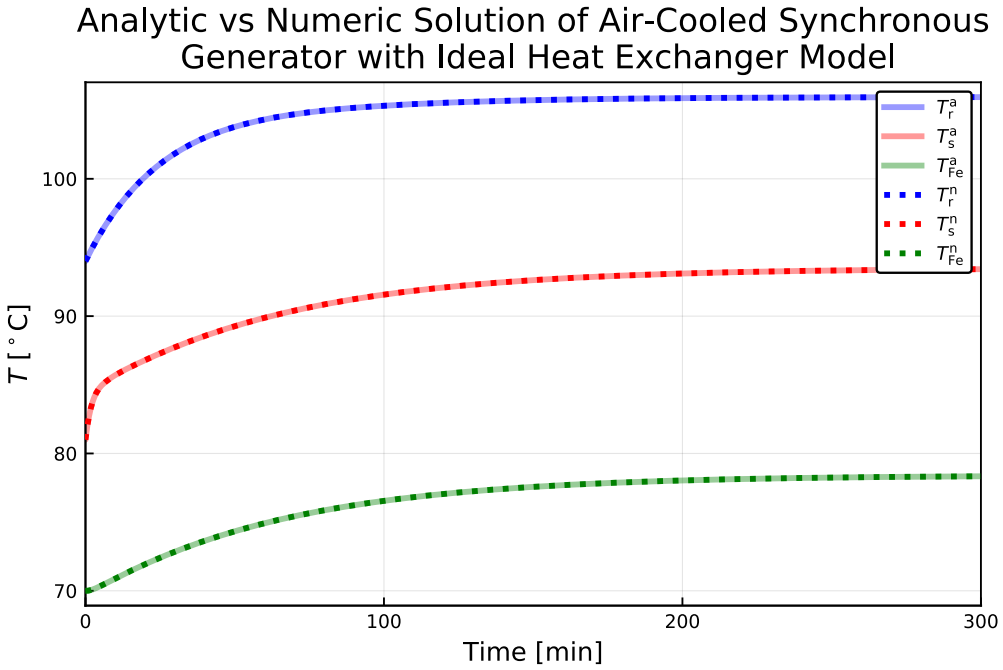


Figure 2.14: Analytic vs. numeric solution of the thermal model of an air-cooled synchronous generator with an ideal heat exchanger model.

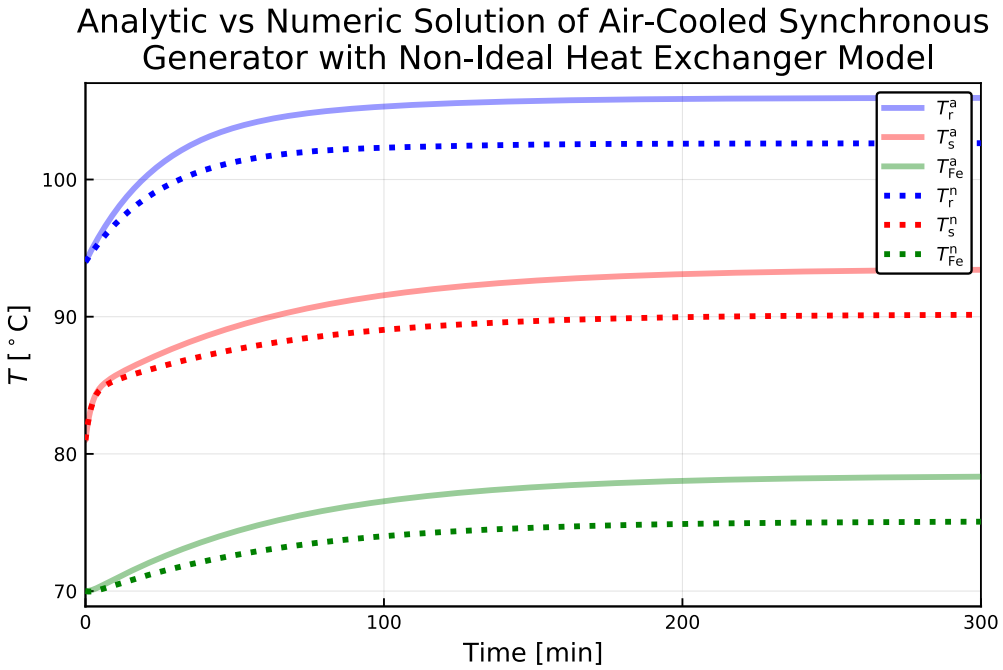


Figure 2.15: Analytic vs. numeric solution of the thermal model of an air-cooled synchronous generator with a non-ideal heat exchanger model.

3 Counter-Current Heat Exchanger Regression Model

In Chapter 2, the benchmark results in Table 2.5 and Table 2.6 for the heat exchanger model, and the benchmark results in Table 2.7 for the thermal model of an air-cooled synchronous generator show that the numeric solution of the linear two-point boundary value problem is slower than analytic solution for the ideal heat exchanger model. More importantly, the benchmark results show a very long simulation time for the numeric solution of the nonlinear two-point boundary value problem for the case of temperature dependence in the specific heat capacities of air and water. In this work, one of the objectives is to speed up the solution time through the following strategy:

- First, both the ideal and the non-ideal heat exchanger models are solved many times off-line for a variety of conditions ($T_{\mathrm{w}}^{\mathrm{c}}$, $T_{\mathrm{a}}^{\mathrm{h}}$, $\dot{m}_{\mathrm{w}}$, and $\dot{m}_{\mathrm{a}}$), and the results ($T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$) are stored in a data matrix.

- Second, a regression model is fitted to the data to form explicit expressions of $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$ for the case of temperature dependence in the specific heat capacities of air and water. In addition, two types of regression models are considered:

    1. Linear regression methods as in Chemometrics.

    2. Nonlinear regression methods such as in Neural Networks.

- Finally, the thermal model of the synchronous generator is solved using a hybrid heat exchanger model that combines the mechanistic model with the regression model, and the benchmark results are compared to that of Table 2.7.

3.1 Overview of Regression Analysis

Regression analysis is a set of mathematical methods used to model the relationship between a variable of interest, called the *regressand*, and a set of predictor variables called

the *regressors*. It is used widely and intensively in almost all scientific disciplines and has become an integral part of quantitative research (Gujarati, 2019).[1]

Regression analysis is mainly used for two distinct purposes, prediction, where it overlaps with the field of machine learning, and hypothesis testing. If the aim is prediction, then regression analysis is used to *estimate* the parameters of a predictive model to an observed data set of values of the regressand and the regressor variables. Only after developing the predictive model, also called the regression model, a prediction of the regressand for additional values of the regressor variables is possible. In practice, all the analysis is based on sample data because one rarely observes the true population of interest of some phenomenon. If the aim is hypothesis testing, then regression analysis is used to draw conclusions about how the population of regressand relates to the population of the regressor based on the sample data (Gujarati, 2019).[1]

In this work, the focus will be on the first of the two aims of regression analysis. The goal is to predict the output temperatures of air and water, of the non-ideal heat exchanger model using a regression model. Two different types of regression models are considered; linear regression methods as in Chemometrics, and nonlinear regression methods such as in Neural Networks.

3.1.1 Linear regression

Given a data set $\{y_\mathrm{i}, x_{1i}, x_{2i}, \ldots, x_{ki}\}_{i=1}^{n}$ of n observations, a generic linear regression model can be expressed as (Gujarati, 2019; Chatterjee and Simonoff, 2013):[1]

$$y_\mathrm{i} = \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \ldots + \beta_k x_{ki} + \varepsilon_i, \qquad i = 1, 2, 3, \ldots, n, \tag{3.1}$$

here, y is the regressand; alternative names are dependent variable, response, explained variable, and endogenous variable. $x_1, x_2, \ldots, x_k$ are called the regressor variables; other names that are used interchangeably are independent variable, explanatory variable, predictor, covariates, and exogenous variable. In this work, the neutral terms regressand and regressor are used. $\beta_1, \beta_2, \ldots, \beta_k$ are the *population* regression coefficients or regression parameters. ε is the stochastic error term, also called the disturbance term. The disturbance term ε accounts for all additional factors or uncontrolled influences which affect the regressand y_i other than the regressors $x_1, x_2, \ldots, x_k$. The subscript i refers to the observation in the data set (Gujarati, 2019).[1]

It is important to mention that the term linear in linear regression means that the regressand is a linear function of the regression parameters; it does not have to be a linear

[1] Wikipedia contributors (Mar. 2020g). *Regression analysis — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Regression_analysis&oldid=944367056. [Online; accessed 12-April-2020]

function of the regressors variables. It is equally important to understand that the values of the regressand and the regressors are known from the data set; they are the observed values. However, the values of the population regression coefficients are unknown. The goal in linear regression is to estimate these coefficients using one of many available estimation methods. One of the common estimation techniques is the method of ordinary least squares (**OLS**), also known as Linear least squares (Gujarati, 2019; Chatterjee and Simonoff, 2013).[2]

In this work, matrix notation is used to simplify the representation of calculations that are made in linear regression, and to make use of the Array data structure in Julia language. Equation 3.1 represents a system of equations that can be stacked together and written in matrix notation as (Gujarati, 2019):

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \tag{3.2}$$

here, $\mathbf{y}$ and ε are $n \times 1$ column vectors, and β is an $k \times 1$ column vector, while $\mathbf{X}$ is the regressor matrix of size $n \times k$. Alternative names for $\mathbf{X}$ are the data matrix, the design matrix and the model matrix.[3] The columns of the data matrix are not necessarily the regressors but can be any functions of them, such as cross-product or quadratic terms (Marquardt, 1980). Equation 3.2 can be written out fully as (Gujarati, 2019; Helwig, 2017):

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{pmatrix} x_{11} & x_{21} & x_{31} & \dots & x_{k1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & x_{3n} & \dots & x_{kn} \end{pmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}. \tag{3.3}$$

The regression in 3.3 is known as Multiple Linear Regression if the number of regressors $k > 1$. Otherwise, if $k = 1$, it is known as a simple linear regression. If the number of the regressand $m > 1$, then the regression is known as Multivariate Linear Regression, and the model is called a general linear model (Helwig, 2017).[4] In Multivariate Linear Regression,

[2] Wikipedia contributors (Apr. 2020e). *Linear regression — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Linear_regression&oldid=948596126. [Online; accessed 12-April-2020]

[3] Wikipedia contributors (Jan. 2020c). *Design matrix — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Design_matrix&oldid=937659051. [Online; accessed 12-April-2020]

[4] Wikipedia contributors (Dec. 2019b). *General linear model — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=General_linear_model&oldid=932027631. [Online; accessed 13-April-2020]

Eq. 3.3 takes the form (Helwig, 2017):

$$\begin{pmatrix} y_{11} & \dots & y_{m1} \\ \vdots & \ddots & \vdots \\ y_{1n} & \dots & y_{mn} \end{pmatrix} = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & \dots & x_{kn} \end{pmatrix} \begin{pmatrix} \beta_{11} & \dots & \beta_{m1} \\ \vdots & \ddots & \vdots \\ \beta_{1k} & \dots & \beta_{mk} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} & \dots & \varepsilon_{m1} \\ \vdots & \ddots & \vdots \\ \varepsilon_{1n} & \dots & \varepsilon_{mn} \end{pmatrix}, \tag{3.4}$$

here, $\mathbf{Y}$ and $\boldsymbol{\varepsilon}$ are $n \times m$ matrices, and $\boldsymbol{\beta}$ is a matrix of size $k \times m$, Usually, a constant regressor that takes the value of 1 is introduced to each observation in the data set. The constant regressor coefficient β_1 is called the intercept. The other regression parameters $\beta_2, \dots, \beta_k$, are known as the partial regression coefficients because they relate the change in the regressand when the regressor associated with it change while all other regressors are held constant (Gujarati, 2019; Chatterjee and Simonoff, 2013).[2]

By including a constant regressor in the data matrix, Eq. 3.4 is expressed as:

$$\begin{pmatrix} y_{11} & \dots & y_{m1} \\ \vdots & \ddots & \vdots \\ y_{1n} & \dots & y_{mn} \end{pmatrix} = \begin{pmatrix} 1 & x_{21} & \dots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{2n} & \dots & x_{kn} \end{pmatrix} \begin{pmatrix} \beta_{11} & \dots & \beta_{m1} \\ \vdots & \ddots & \vdots \\ \beta_{1k} & \dots & \beta_{mk} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} & \dots & \varepsilon_{m1} \\ \vdots & \ddots & \vdots \\ \varepsilon_{1n} & \dots & \varepsilon_{mn} \end{pmatrix}. \tag{3.5}$$

The model in Eq. 3.5 is based on the population of interest that is the focus of some statistical analysis. Therefore, it is known as the population regression model. However, as mentioned before, one rarely observes the true population of interest, and the estimation is based on sample data. Using the sample data, an estimate $\hat{\boldsymbol{\beta}}$ of the population regression coefficients can be made (Gujarati, 2019).

3.1.1.1 **Ordinary least squares** (OSL)

To estimate the regression coefficients using the method of ordinary least squares, one can begin with the matrix form of the multiple linear regression model (Gujarati, 2019):

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + e. \tag{3.6}$$

The disturbance term $\boldsymbol{\varepsilon}$ is replaced by e, a vector of residuals. Also, the regression parameters $\boldsymbol{\beta}$ is replaced by $\hat{\boldsymbol{\beta}}$, a vector of estimators of $\boldsymbol{\beta}$. The residual is the sample equivalent of the disturbance term (Gujarati, 2019). The disturbance term is the difference between the actual values of the regressand and the population mean. On the other hand, the residual is the difference between the actual values of the regressand and the sample mean.[5] Using Eq. 3.6 the residual can be expressed as:

$$e = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}. \tag{3.7}$$

[5]Wikipedia contributors (Apr. 2020d). *Errors and residuals — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Errors_and_residuals&oldid=949672133. [Online; accessed 13-April-2020]

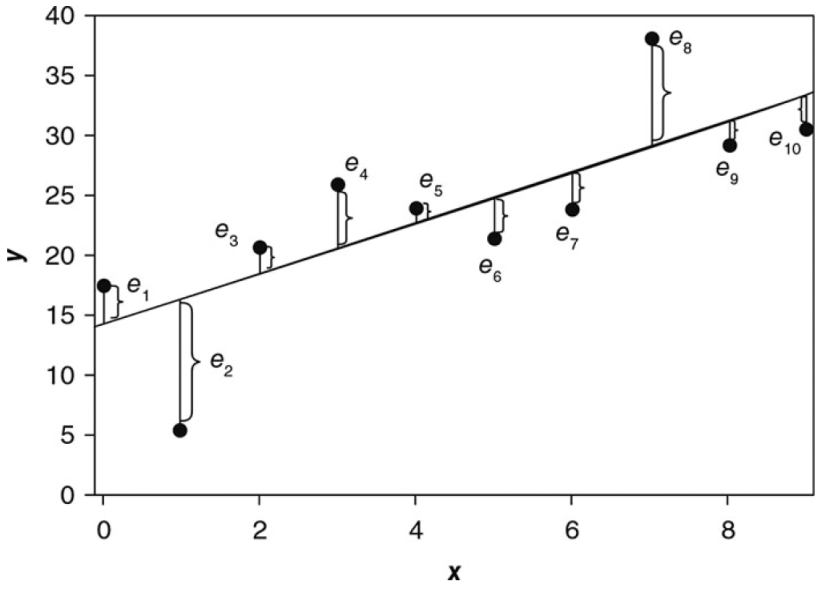


Figure 3.1: Hypothetical scattergram (Gujarati, 2019).

Figure 3.1 shows an example of a hypothetical Scattergram where the residual is the vertical distance between the data points (y_i, the actual values) and the straight line. The straight line in Fig. 3.1 is the ordinary least-squares regression line, which is an estimate of the true population regression line. Also, it can be seen that residual for some of the observations have a negative value while others have a positive value. The aim is to find an estimate of the regression coefficients that make the sum of residuals as small as possible, ideally zero. In ordinary least squares, one minimizes the sum of squared residuals (SSR) instead of the sum of residuals to avoid problems associated with the sign of the residuals.[6] In this work, the term residual sum of squares (RSS) will be used instead of SSR. In the hypothetical Scattergram, ordinary least squares found the best-fitting line, which is the case for the simple linear regression. If there were two regressors, ordinary least squares would have found a regression plane or a regression surface (Gujarati, 2019; Chatterjee and Simonoff, 2013).

The residual sum of squares can be minimized by taking the derivative with respect to the unknown $\hat{\beta}$ and setting the resulting equations to zero (Gujarati, 2019). First, the

[6]The sum of squared residuals is also called the sum of squared estimate of errors (SSE).

residual sum of squares is expressed in matrix notation:

$$\begin{aligned}\sum e_i^2 &= e^\intercal e \\ &= \left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)^\intercal \left(\mathbf{y} - \mathbf{X}\hat{\beta}\right) \\ &= \left(\mathbf{y}^\intercal - \hat{\beta}^\intercal \mathbf{X}^\intercal\right) \left(\mathbf{y} - \mathbf{X}\hat{\beta}\right) \\ &= \mathbf{y}^\intercal\mathbf{y} - 2\hat{\beta}^\intercal\mathbf{X}^\intercal\mathbf{y} + \hat{\beta}^\intercal\mathbf{X}^\intercal\mathbf{X}\hat{\beta},\end{aligned} \tag{3.8}$$

where $\mathbf{y}^\intercal\mathbf{X}\hat{\beta} = \hat{\beta}^\intercal\mathbf{X}^\intercal\mathbf{y}$ by matrix transpose properties. Second, taking the derivative of Eq. 3.8 with respect to $\hat{\beta}$:

$$\frac{\partial e^\intercal e}{\partial \hat{\beta}} = -2\mathbf{X}^\intercal\mathbf{y} + 2\mathbf{X}^\intercal\mathbf{X}\hat{\beta}. \tag{3.9}$$

Next, setting the derivative to zero:

$$0 = -2\mathbf{X}^\intercal\mathbf{y} + 2\mathbf{X}^\intercal\mathbf{X}\hat{\beta}. \tag{3.10}$$

Then, move the term with $\hat{\beta}$ to the left hand side:

$$\mathbf{X}^\intercal\mathbf{X}\hat{\beta} = \mathbf{X}^\intercal\mathbf{y}. \tag{3.11}$$

Note that $\mathbf{X}^\intercal\mathbf{X}$ is a square matrix. If the inverse of $\mathbf{X}^\intercal\mathbf{X}$ exists, then it is possible to premultiply the expression with the inverse $(\mathbf{X}^\intercal\mathbf{X})^{-1}$ to obtain an expression for $\hat{\beta}$:

$$(\mathbf{X}^\intercal\mathbf{X})^{-1}(\mathbf{X}^\intercal\mathbf{X})\hat{\beta} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y}, \tag{3.12}$$

here $(\mathbf{X}^\intercal\mathbf{X})^{-1}(\mathbf{X}^\intercal\mathbf{X}) = \mathbf{I}$, which is the identity matrix, and $I\hat{\beta} = \hat{\beta}$. Finally, the regression coefficients are given by (Gujarati, 2019; Chatterjee and Simonoff, 2013):

$$\hat{\beta} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y}. \tag{3.13}$$

Its worth mentioning that the inverse of $\mathbf{X}^\intercal\mathbf{X}$ exists only if the number of the observations n is larger than the number of the regressors k, which is when the matrix $\mathbf{X}$ is of full column rank . After obtaining an estimate of the population regression coefficients, the regression function is expressed as (Gujarati, 2019):

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}, \tag{3.14}$$

here, $\hat{\mathbf{y}}$is called the predicted value of $\mathbf{y}$; it is an estimation of the population mean response for a given value of the regressand (Gujarati, 2019; Kiernan, 2018).

3.1.1.2 The coefficient of determination (R^2)

There are several statistics to compare different regression models that use OLS estimation. The coefficient of determination is one statistic that describes how well the model fits a set of observations.[7] Specifically, the coefficient of determination measures the percentage of the variance in the regressand that is predictable from the regressors; R^2 is a relative measure of the goodness-of-fit of the model and is defined as (Gujarati, 2019):[8]

$$R^2 = \frac{\text{Explained sum of squares}}{\text{Total sum of squares}} = \frac{\text{ESS}}{\text{TSS}}. \tag{3.15}$$

The total sum of squares (TSS) for a zero-intercept multivariate regression model is given by:[9]

$$\text{TSS} = \sum_{i=1}^{n} y_i^2, \tag{3.16}$$

or in matrix notation as:

$$\text{TSS} = \mathbf{Y}^\intercal \mathbf{Y}. \tag{3.17}$$

However, if the model contained an intercept, then the total sum of squares needs to be mean-corrected (Gujarati, 2019; Chatterjee and Simonoff, 2013; Helwig, 2017):

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2, \tag{3.18}$$

where, $\bar{y}$ is the mean of the regressand observations:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i. \tag{3.19}$$

In matrix notation, Eq. 3.18 is expressed as (Gujarati, 2019):

$$\text{TSS} = \mathbf{Y}^\intercal \mathbf{Y} - n\bar{y}^2. \tag{3.20}$$

[7] The coefficient of determination should not be confused with the coefficient of multiple correlation, which is the square root of R^2.

[8] Wikipedia contributors (Mar. 2020a). *Coefficient of determination — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Coefficient_of_determination&oldid=946053744. [Online; accessed 13-April-2020]

[9] In some literature, the symbol SS_{tot} is used instead of TSS.

It is possible to partition **TSS** in Eq. 3.18 as (Helwig, 2017):[10]

$$\begin{aligned}\text{TSS} &= \sum_{i=1}^{n}(y_i-\bar{y})^2 \\ &= \sum_{i=1}^{n}(y_i-\hat{y}_i+\hat{y}_i-\bar{y})^2 \\ &= \sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2+2\sum_{i=1}^{n}(y_i-\hat{y}_i)(\hat{y}_i-\bar{y}) \\ &= \sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2+2\sum_{i=1}^{n}(\hat{y}_i-\bar{y})e_i.\end{aligned} \tag{3.21}$$

The last term in the previous expression is zero because one of the properties of **OSL** is that the sum of the residuals is zero (Gujarati, 2019).[10] Therefore,

$$\text{TSS} = \sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2+\sum_{i=1}^{n}(y_i-\hat{y}_i)^2. \tag{3.22}$$

The first term on the right-hand side in the previous expression is the mean-corrected explained sum of squares (Gujarati, 2019; Chatterjee and Simonoff, 2013):[11]

$$\text{ESS} = \sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2. \tag{3.23}$$

In matrix notation, **ESS** is expressed as (Gujarati, 2019):

$$\text{ESS} = \hat{\boldsymbol{\beta}}^{\intercal}\mathbf{X}^{\intercal}\mathbf{X}\hat{\boldsymbol{\beta}}-n\bar{y}^2. \tag{3.24}$$

For regression through the origin $\text{ESS} = \sum_{i=1}^{n}\hat{y}_i^2$, or in matrix notation, $\text{ESS} = \hat{\boldsymbol{\beta}}^{\intercal}\mathbf{X}^{\intercal}\mathbf{X}\hat{\boldsymbol{\beta}}$. The last term in Eq. 3.22 is the residual sum of squares (**RSS**) (Gujarati, 2019; Chatterjee and Simonoff, 2013):[12]

$$\text{RSS} = \sum_{i=1}^{n}(y_i-\hat{y}_i)^2. \tag{3.25}$$

In matrix notation, the residual sum of squares can be expressed by Eq. 3.8:

$$\sum e_i^2 = e^{\intercal}e. \tag{3.26}$$

[10] Wikipedia contributors (Nov. 2019a). *Explained sum of squares — Wikipedia, The Free Encyclopedia.* https://en.wikipedia.org/w/index.php?title=Explained_sum_of_squares&oldid=925963298. [Online; accessed 13-April-2020]

[11] Explained sum of squares is also called the regression sum of squares (SS_{reg}).

[12] In some literature, the symbol SS_{res} is used instead of **RSS**.

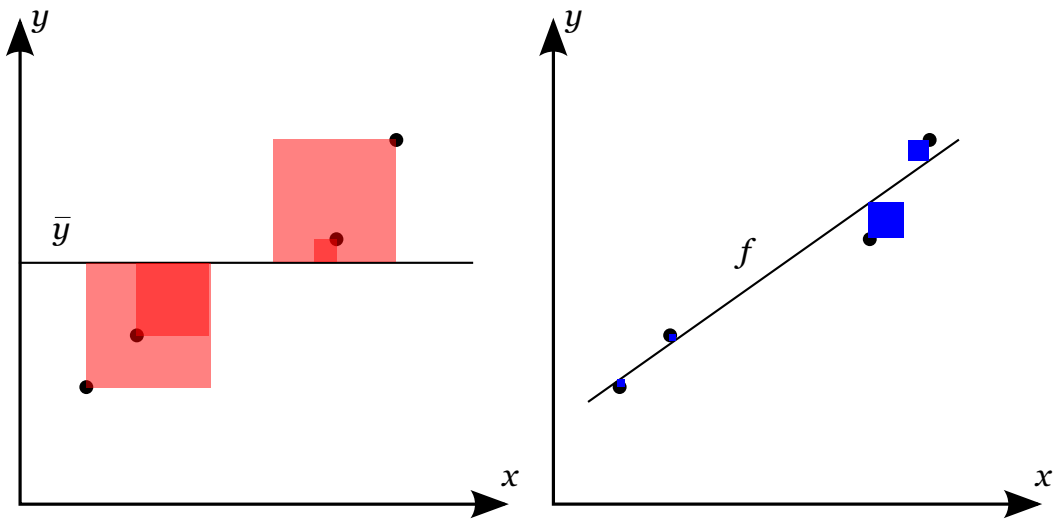


Figure 3.2: A visual interpretation of the coefficient of determination.[8]

Accordingly, the coefficient of determination for a model with an intercept can be expressed as (Gujarati, 2019):[8]

$$\mathrm{R}^2 = \frac{\hat{\boldsymbol{\beta}}^\intercal \mathbf{X}^\intercal \mathbf{X} \hat{\boldsymbol{\beta}} - n\bar{y}^2}{\mathbf{Y}^\intercal \mathbf{Y} - n\bar{y}^2}, \tag{3.27}$$

and for a zero-intercept model as:

$$\mathrm{R}^2 = \frac{\hat{\boldsymbol{\beta}}^\intercal \mathbf{X}^\intercal \mathbf{X} \hat{\boldsymbol{\beta}}}{\mathbf{Y}^\intercal \mathbf{Y}}. \tag{3.28}$$

Alternatively, the coefficient of determination can be expressed in terms of RSS as (Gujarati, 2019; Chatterjee and Simonoff, 2013):[8]

$$\begin{aligned} \mathrm{R}^2 &= 1 - \frac{\text{Residual sum of squares}}{\text{Total sum of squares}} \\ &= 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} \\ &= 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}. \end{aligned} \tag{3.29}$$

Figure 3.2 shows a visual interpretation of R^2. In the figure, the blue area on the right is equal $(y_i - \hat{y}_i)^2$, and the red area on the left is $(y_i - \bar{y})^2$ for some observations, i.e., the coefficient of determination is:

$$\mathrm{R}^2 = 1 - \frac{\sum \text{Area in blue}}{\sum \text{Area in red}}. \tag{3.30}$$

Sum of squares (SS)	The degrees of freedom (df)
$\text{ESS} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	$\text{df}_{\text{reg}} = k$
$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$\text{df}_{\text{res}} = n - k$
$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$	$\text{df}_{\text{tot}} = n - 1$

Table 3.1: Sum of squares and their corresponding degrees of freedom. k is the number regression parameters, and n is the number of observations (Gujarati, 2019; Chatterjee and Simonoff, 2013; Helwig, 2017; S. Pandey and Bright, 2008; Nau, 2014).[8]

Similarly, R^2 for a model with an intercept is expressed as:

$$\text{R}^2 = 1 - \frac{e^\intercal e}{\mathbf{Y}^\intercal \mathbf{Y} - n\bar{y}^2}, \tag{3.31}$$

and for a zero-intercept model as:

$$\text{R}^2 = 1 - \frac{e^\intercal e}{\mathbf{Y}^\intercal \mathbf{Y}}. \tag{3.32}$$

A helpful property of R^2 is that normally it ranges 0 to 1, where a value of one indicates a perfect fit, and a value of zero indicates no improvement over the so called mean model, which makes assessing the goodness-of-fit of the model more intuitive.[13] On the other hand, a well-known property of R^2 is that it increases every time additional regressor is added to the model despite no improvement in the model's fit. Then, R^2 must be adjusted to incorporates the model's degrees of freedom (df) to make it an unbiased estimator. The degrees of freedom adjust R^2 for the number of regressors relative to the number of observations (Gujarati, 2019; Chatterjee and Simonoff, 2013).[8]

Table 3.1 summarizes the previously mentioned Sum of squares and their corresponding degrees of freedom.[14]

The adjusted R^2 is denoted $\bar{\text{R}}^2$ and is expressed as (Gujarati, 2019):

$$\begin{aligned}
\bar{\text{R}}^2 &= 1 - \frac{\frac{\text{RSS}}{\text{df}_{\text{res}}}}{\frac{\text{TSS}}{\text{df}_{\text{tot}}}} \\
&= 1 - \frac{\text{RSS}}{\text{TSS}} \times \frac{\text{df}_{\text{tot}}}{\text{df}_{\text{res}}} \\
&= 1 - (1 - \text{R}^2)\frac{\text{df}_{\text{tot}}}{\text{df}_{\text{res}}} \\
&= 1 - (1 - \text{R}^2)\frac{n-1}{n-k}.
\end{aligned} \tag{3.33}$$

[13] In a zero-intercept model (regression through the origin), the last term in Eq. 3.21 can take a negative value. Then R^2 will take a negative value, which means that the mean $\bar{y}$ (which is a horizontal line) describes the data better than the regression model.

[14] In Helwig (2017), The degrees of freedom in Multivariate Linear Regression are multiplied by the number of regressands m.

It is worth mentioning that $\bar{R}^2$can take a negative value if the number of regression parameters are larger than the number of observations.

3.1.1.3 The standard error of the regression

If the main aim of the regression model is prediction, then there is one statistic that takes precedence over the others, namely the root mean square error (RMSE) (Grace-Martin, 2008). When adjusted for the degrees of freedom for the residuals, then it is called the standard error of the regression or the Standard Error of the Estimate.[15] RMSE is defined as:[16,17,18]

$$\begin{aligned} \text{RMSE} &= \sqrt{\text{Mean Square Error}} \\ &= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \\ &= \sqrt{\frac{\text{RSS}}{n}}, \end{aligned} \tag{3.34}$$

where n is the number of observation. On the other hand, the standard error of the regression is the standard deviation of the residuals, and is defined as:[15,19,18]

$$\text{s} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-k}} \quad , \tag{3.35}$$

here, k is the number of parameters in the model. If the number of observations was much larger than the number of coefficients in the regression model, then the difference between RMSE and the standard error of the regression will be small. A helpful property of the standard error of the regression is that it have same unit of the regressand, and unlike R^2 or $\bar{R}^2$ it is an absolute measure of goodness-of-fit, where a lower value indicates a better fit (Grace-Martin, 2008).

[15] Wikipedia contributors (Apr. 2020f). *Ordinary least squares — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Ordinary_least_squares&oldid=948717245. [Online; accessed 13-April-2020]

[16] Wikipedia contributors (Feb. 2020h). *Root-mean-square deviation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Root-mean-square_deviation&oldid=941256353. [Online; accessed 13-April-2020]

[17] RMSE is also known as the root mean square deviation (RMSD).

[18] NobleProg, ed. (June 2014). *Standard Error of the Estimate - Training Material*. https://training-course-material.com/training/Standard_Error_of_the_Estimate. [Online; accessed 13-April-2020]

[19] The standard error of the regression is also called the residual standard error.

3.2 Cross-Validation

Cross-validation, also called out-of-sample testing, is any of the various model validation procedures in which the model's predictive ability is tested for accuracy. In cross-validation, the sample data is partitioned into complementary subsets called the training dataset and the validation dataset. The training dataset is used to train/build or estimate the model parameters, and the validation dataset is used to evaluate the model's predictive performance. For the evaluation purpose it is common to use single-parameter indicators of goodness-of-fit, such as R^2 and **RMSE**. The process of partitioning, training, and evaluation is considered a single run or one round of cross-validation. In most methods of cross-validation, multiple runs are usually performed to reduce the variability of the validation results, the results of multiple runs are typically averaged together. The goal in cross-validation is to obtain a consistent accuracy for the training and validation data sets, which is an indication that the model will continue to be accurate when applied to an entirely new set of data. In addition, cross-validation is used to detect issues such as model overfitting. Figure 3.3 shows how that total error varies as the model complexity increase.[20] Cross-validation methods can be classified into two types according to the way the original sample is divided (Allen, 2017):[21,22]

1. Exhaustive cross-validation: as the name implies, these methods considers all possible ways to divide the original sample.
2. Non-exhaustive cross-validation: the methods that do not consider all possible ways to divide the original sample.

One of the simplest methods is the Holdout method: it is a non-exhaustive cross-validation method, where the data is split randomly into two datasets; a training set to estimate the model's parameters and test set to assess the performance of the model. The training dataset is typically larger than the test set. Also, the Holdout method involves a single run, but it can be repeated multiple times to improve its strength as a validation method.[22] The Holdout method is relatively easy to program in Julia, but for more complex methods, the package MLBase.jl can be used, it contains the following cross-validation schemes:

1. Exhaustive methods:
    - Leave-one-out cross-validation (**LOOCV**).

[20] DataVedas (Mar. 2018). *REGULARIZED REGRESSION ALGORITHMS | Data Vedas*. en-US. https://www.datavedas.com/regularized-regression-algorithms/. [Online; accessed 11-April-2020]

[21] Wikipedia contributors (Aug. 2019c). *Regression validation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Regression_validation&oldid=913207226. [Online; accessed 13-April-2020]

[22] Wikipedia contributors (Apr. 2020b). *Cross-validation (statistics) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)&oldid=948965871. [Online; accessed 13-April-2020]

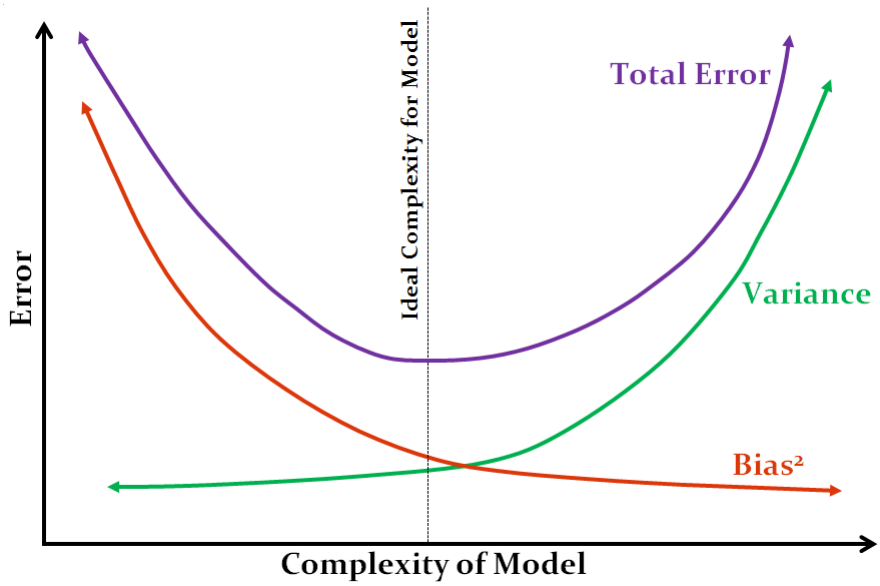


Figure 3.3: Model overfitting.[20]

2. Non-exhaustive methods:

   - k-fold cross-validation (k-fold).
   - Repeated random sub-sampling validation, also known as Monte Carlo cross-validation.

3.3 Linear Regression of the Counter-Current Heat Exchanger Model

To reiterate what was mentioned at the beginning of this chapter, one of the objectives in this work is to speed up the solution time of the non-ideal heat exchanger model (the case of temperature dependence in the specific heat capacities of air and water). Then, in this section, to speed up the solution time, an explicit data-driven model for the non-ideal heat exchanger model is developed by linear regression, which is expressed as a correction expression to the ideal heat exchanger model. Also, when solving the dynamic thermal model of the generator, the explicit data-driven model (which is fast) is used instead of solving a nonlinear two-point boundary value problem (which is slow) in each time step of the DAE solver.

3.3.1 Implementation of linear regression in Julia

Based on the information presented in Section 3.1, linear regression of the of the counter-current heat exchanger model is implemented in Appendix B.3. In this work, first, the ideal and the non-ideal heat exchanger models are solved for a variety of conditions ($T_{\mathrm{w}}^{\mathrm{c}}$, $T_{\mathrm{a}}^{\mathrm{h}}$, $\dot{m}_{\mathrm{w}}$, and $\dot{m}_{\mathrm{a}}$) to generate data matrices of the analytic (a data matrix of $T_{\mathrm{w}}^{\mathrm{h,A}}$ and $T_{\mathrm{a}}^{\mathrm{c,A}}$) and the numeric (a data matrix of $T_{\mathrm{w}}^{\mathrm{h,N}}$ and $T_{\mathrm{a}}^{\mathrm{c,N}}$) solutions, respectively.[23] Then, linear regression is carried out on the data matrices to estimate the parameters ($\hat{\boldsymbol{\beta}}$) of the regression model. Next, once the parameters are known, prediction of $T_{\mathrm{w}}^{\mathrm{h,R}}$ and $T_{\mathrm{a}}^{\mathrm{c,R}}$ is performed using Eq. 3.14.[24] The following code illustrate:

```
####### Generating the dataset #######
# Experimental ranges
n_Tw = 20
n_Ta = 20
ngrid = n_Tw*n_Ta
#
r_Twc = range(4.,30.,length=n_Tw)
r_Tah = range(40.,100.,length=n_Ta)
#
T_grid = [[Twc,Tah] for Twc in r_Twc, Tah in r_Tah]
TTAgrid_c1 = Matrix{Float64}(undef,2,ngrid)
TTNgrid_c1 = Matrix{Float64}(undef,2,ngrid);
####### Solving model for all inputs #######
for i in 1:ngrid
    Twc,Tah = T_grid[i]
    u_hex = [Twc Tah mdw mda]
    sol_analytic = map(dx -> hex_a(u_hex, par_hex_a, dx), xspan_a)
    TTAgrid_c1[:,i] .= [sol_analytic[end][1],sol_analytic[1][end]]
    sol_numeric = hex_n(u_hex, par_hex_n, xspan_n)
    TTNgrid_c1[:,i] .= [sol_numeric[end][1],sol_numeric[1][end]]
end
####### Estimating the Parameters of the Model #######
X = (phi_m.(mat2vec(TTAgrid_c1);n=1) |> x -> reduce(hcat,x))'
Y = TTNgrid_c1';
beta = inv(X'X)X'Y
#
# However, in practice, the backslash operator \ is used in the calculation of beta
# since not all matrices are well defined with full-rank.
beta = (X'X)\X'Y
# or
beta = X\Y
####### Predicting new values #######
ŷ = X*beta;
```

[23] In this work, the design matrix is a polynomial of the regressors $T_{\mathrm{w}}^{\mathrm{h,A}}$ and $T_{\mathrm{a}}^{\mathrm{c,A}}$.
[24] In this work, the superscript R is used for the predicted values of $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$ from the regression model.

3.3.2 The best-fit model and regression validation

To assess the predictive ability of the regression model, several datasets under the following conditions were generated:

- Case 1: Constant mass flow rates ($\dot{m}_w$ and $\dot{m}_a$), and variable influent temperatures (T_w^c and T_a^h).
- Case 2: Constant influent temperatures (T_w^c and T_a^h), and variable mass flow rates ($\dot{m}_w$ and $\dot{m}_a$).
  - Case A: Mass flow rates are generated in a high range, including the values used in case 1.
  - Case B: Mass flow rates are generated to a lower range than case (2.A). Here, the aim is to evaluate the impact of low mass flow rates on the goodness-of-fit.
- Case 3: Variable influent temperatures (T_w^c and T_a^h) and mass flow rates ($\dot{m}_w$ and $\dot{m}_a$).
  - Case A: Influent temperatures are generated in the same range as case 1, and the mass flow rates are generated in the same range as case (2.A).
  - Case B: The datasets are generated in a wider range as in the case of (3.C). However, in this case, the mass flow rates are also regressors. The aim, in this case, is to evaluate the impact of mass flow rates as regressors on the goodness-of-fit and to compare the goodness of fit with nonlinear regression later on.
  - Case C: This case is the same as case (3.A), except the data is generated for a wider range. The goal here is to compare the goodness of fit with nonlinear regression later on.

More details about each case can be found in Appendix B.3. Moreover, to find the model that best describes the data, two functions were made in Julia to ease the process. The first function (**goodness_of_fit**) was made to find the polynomial order of the design matrix that gives the highest adjusted R-squared and the least standard error of the regression. The second function (**multiple_Holdout**) runs the holdout method multiple times and returns an average of adjusted R-squared, an average of the standard error of the regression, and an average of RMSE(the root-mean-square error will be used in comparison with the nonlinear models). The second function aims to validate the regression models by repeated random sampling of the datasets.[25]

[25]Both **goodness_of_fit** and **multiple_Holdout** generate the polynomial of the design matrix using **phi_m**, a function made by the supervisor.

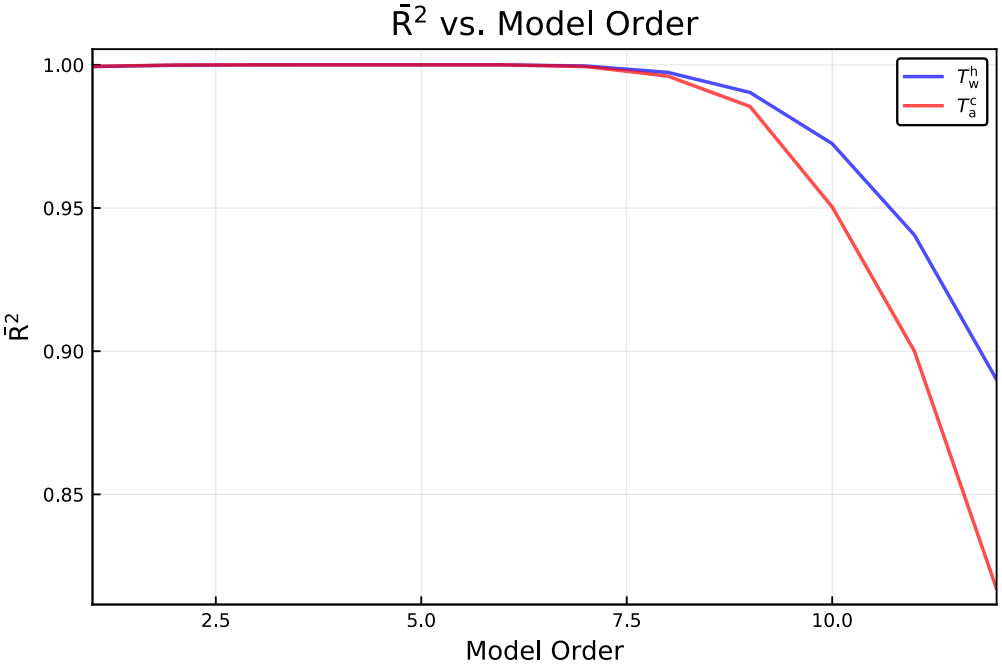


Figure 3.4: Case 1: $\bar{R}^2$ vs. model order.

3.3.2.1 Simulation results of case 1

In this case, the impact of influent temperatures (T_w^c and T_a^h) on the goodness-of-fit is evaluated. First, the best-fit polynomial order of the design matrix is found using the **goodness_of_fit** function. Figure 3.4 and Fig. 3.5 show the impact of increasing the polynomial order of the design matrix on the adjusted R-squared ($\bar{R}^2$), and on the standard error of the regression, respectively.

Based on Fig. 3.4 and Fig. 3.5, the following models were chosen to demonstrate the goodness of fit:

- Model 1: The design matrix has a **1st** order polynomial.
- Model 2: The design matrix has a **6th** order polynomial.
- Model 3: The design matrix has a **12th** order polynomial.

Next, Fig. 3.6 shows a comparison between the regression models and the numerical solution of the nonlinear two-point boundary value problem for models 1, 2, and 3. In addition, Fig. 3.7, Fig. 3.8, and Fig. 3.9 show the residuals, and the errors in T_w^h and T_a^c(regressed vs. analytic model) for models 1, 2, and 3, respectively. Finally, the

Figure 3.5: Case 1: The standard error of the regression vs. model order.

Model order	Avg. s, °C		Avg. RMSE, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.01797396	0.0190537	0.00450191	0.00505826	0.99938061	0.99944294
6th	$8.062e-5$	$9.369e-5$	$9.7e-7$	$1.96e-6$	0.99999987	0.99999979
12th	0.41295796	0.59963816	1.39820202	2.9637627	0.80846771	0.67603817

Table 3.2: Case 1: Validation results.

function **multiple_Holdout** is used to validate the three models. Table 3.2 summarizes the validation results.[26]

3.3.2.2 Simulation results of case 2.A

In this case, the impact of mass flow rates ($\dot{m}_w$ and $\dot{m}_a$) on the goodness-of-fit is evaluated. Similarly to case 1, first, the best-fit polynomial order of the design matrix is found using the **goodness_of_fit** function. Figure 3.10 and Fig. 3.11 show the impact of increasing the polynomial order of the design matrix on the adjusted R-squared ($\bar{R}^2$), and on the standard error of the regression, respectively.

[26] In Table 3.2, the results were rounded to the 8th digit after the decimal point. Also, as mentioned in Section 3.1, s is the symbol for the standard error of the regression.

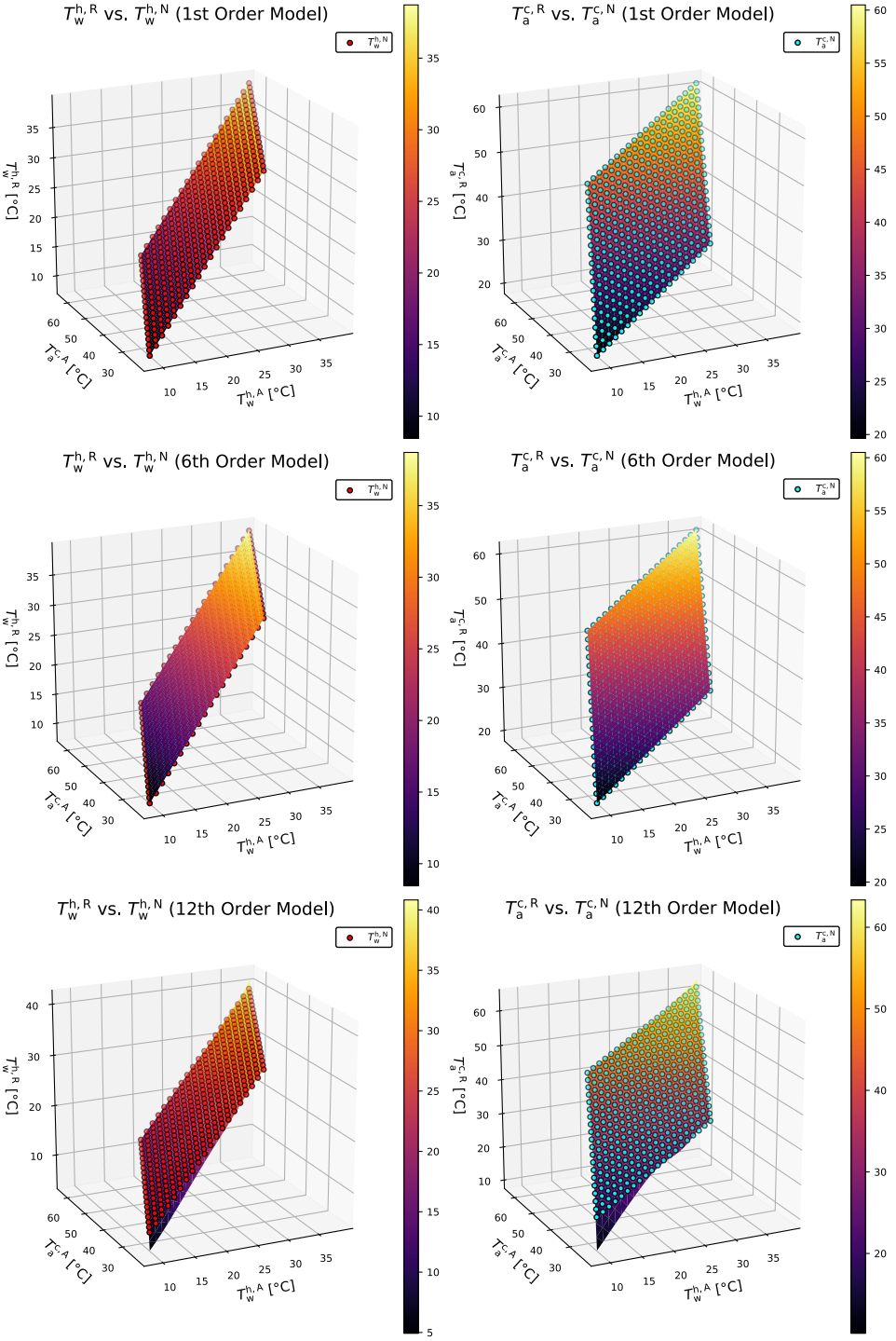


Figure 3.6: Case 1: Comparison between the regression models (the surfaces in the figure) and the numerical solution of the nonlinear two-point boundary value problem (the data points) for a 1st order, a 6th order, and a 12th order polynomial regression models.

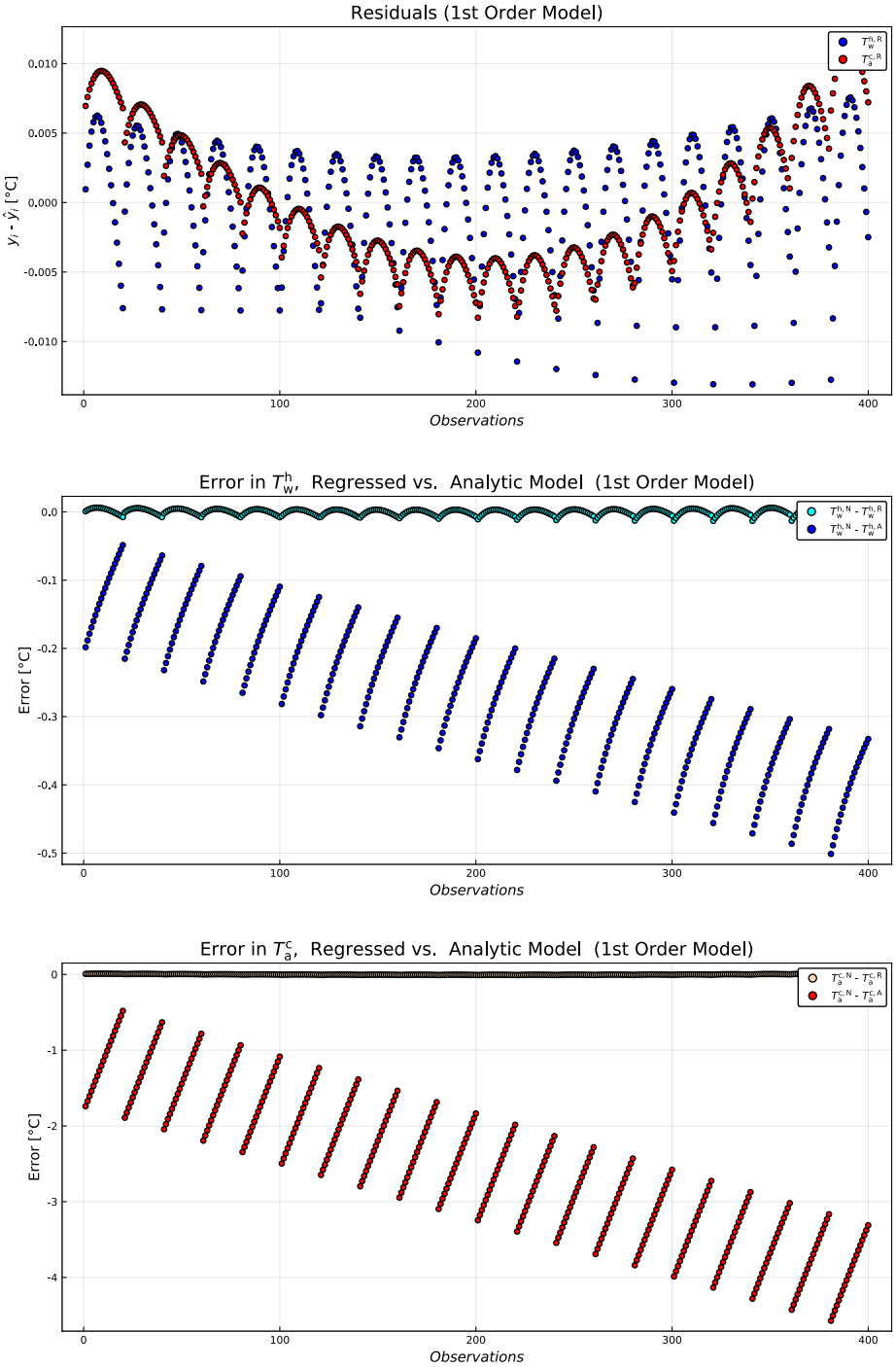


Figure 3.7: Case 1: The residuals, and the errors in $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$, regressed vs. analytic model (1st order model).

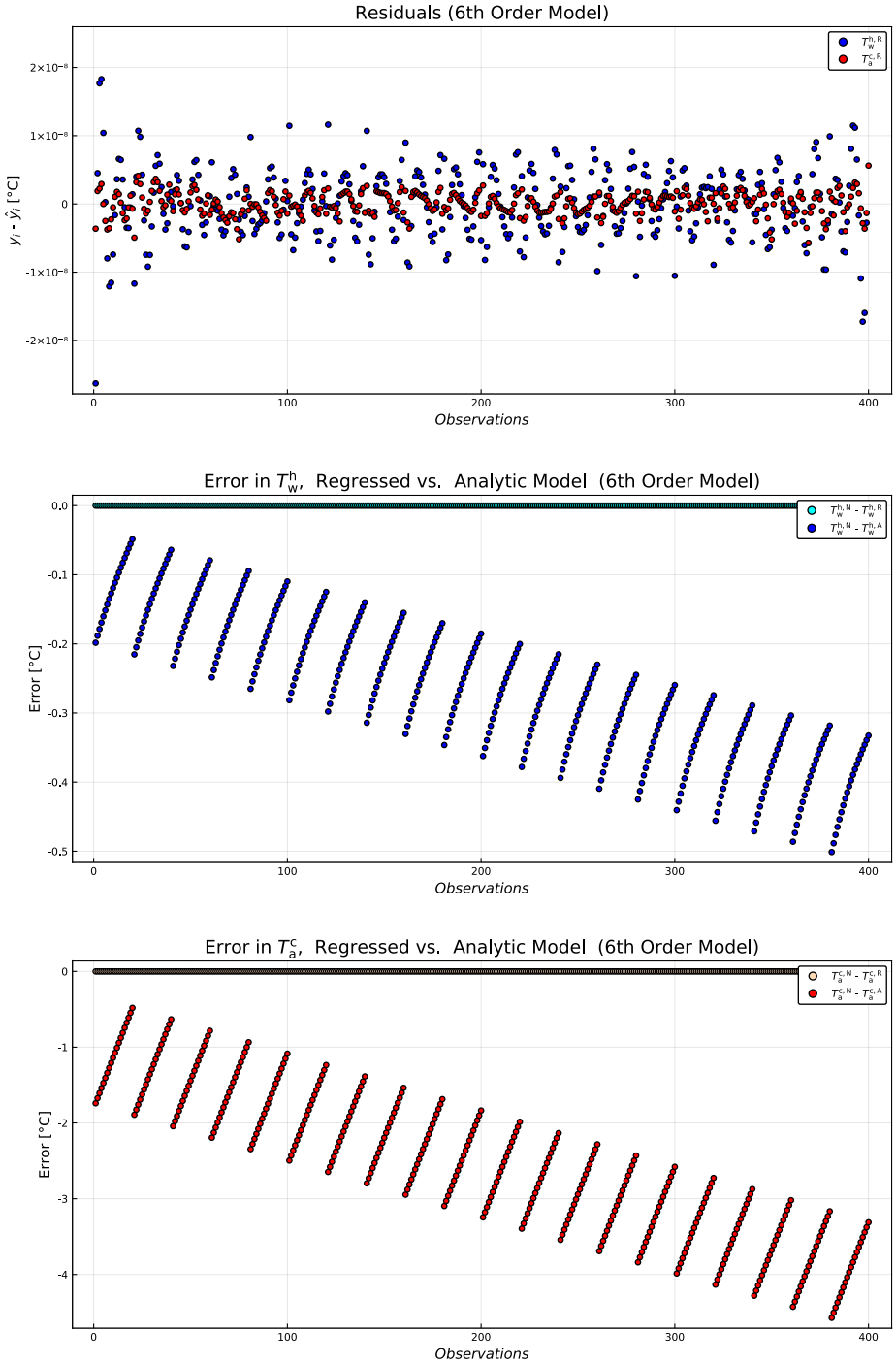


Figure 3.8: Case 1: The residuals, and the errors in $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$, regressed vs. analytic model (6th order model).

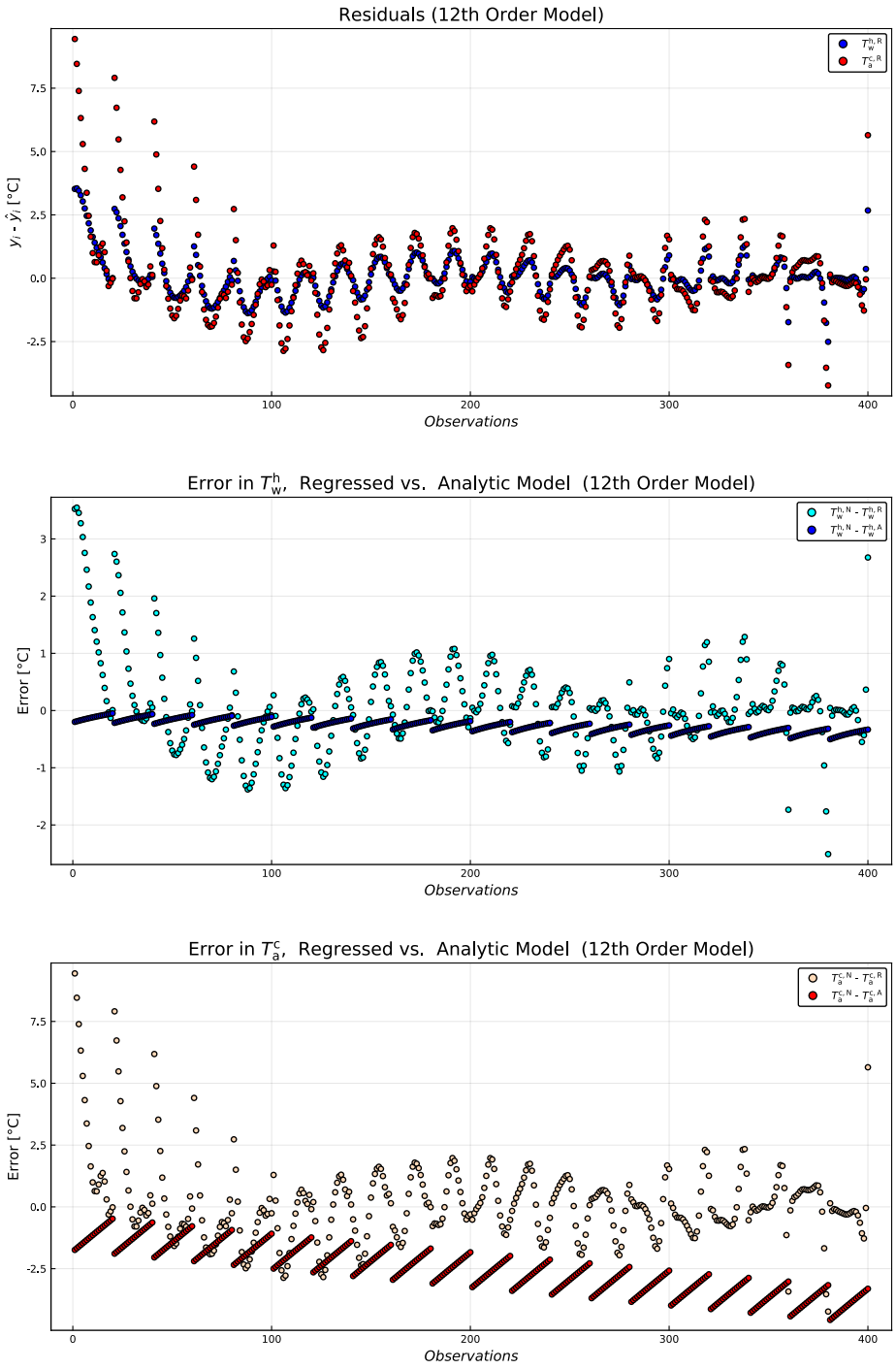


Figure 3.9: Case 1: The residuals, and the errors in $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$, regressed vs. analytic model (12th order model).

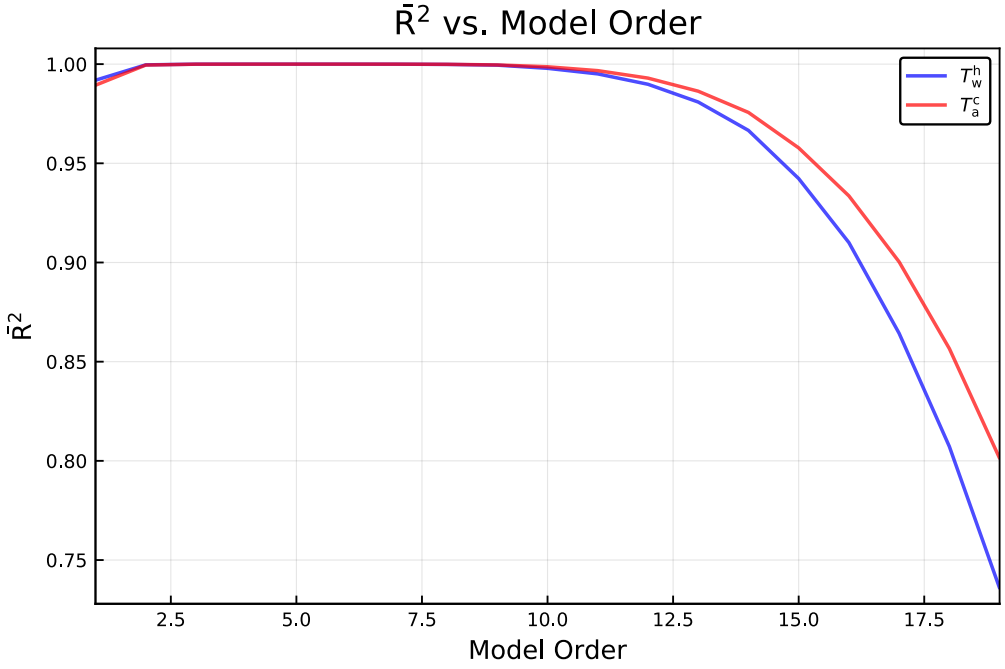


Figure 3.10: Case 2: $\bar{R}^2$ vs. model order.

Figure 3.11: Case 2: The standard error of the regression vs. model order.

Model order	Avg. s, °C		Avg. RMSE, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.0257428	0.0503602	0.00923701	0.03534026	0.99172604	0.98918441
6th	$4.705e-5$	$8.981e-5$	$7.0e-8$	$1.9e-7$	0.99999994	0.99999994
18th	0.64079578	0.94432409	0.296523	0.64326974	0.73520221	0.80353184

Table 3.3: Case 2: Validation results.

Based on Fig. 3.10 and Fig. 3.11, the following models were chosen to demonstrate the goodness of fit:

- Model 1: The design matrix has a 1st order polynomial.
- Model 2: The design matrix has a 6th order polynomial.
- Model 3: The design matrix has a 18th order polynomial.

Next, Fig. 3.12 shows a comparison between the regression models and the numerical solution of the nonlinear two-point boundary value problem for models 1, 2, and 3. In addition, Fig. 3.13, Fig. 3.14, and Fig. 3.15 show the residuals, and the errors in T_w^h and T_a^c(regressed vs. analytic model) for models 1, 2, and 3, respectively. Finally, the function **multiple_Holdout** is used to validate the three models. Table 3.3 summarizes the validation results.

3.3.2.3 Simulation results of case 2.B

In this case, the mass flow rates ($\dot{m}_w$ and $\dot{m}_a$) were taken down to a lower range to observe the impact on the goodness-of-fit. In a similar manner to the previous cases, first, the best-fit polynomial order of the design matrix is found using the **goodness_of_fit** function. Figure 3.16 and Fig. 3.17 show the impact of increasing the polynomial order of the design matrix on the adjusted R-squared ($\bar{R}^2$), and on the standard error of the regression, respectively.

Based on Fig. 3.16 and Fig. 3.17, the following models were chosen to demonstrate the goodness of fit:

- Model 1: The design matrix has a 1st order polynomial.
- Model 2: The design matrix has a 6th order polynomial.
- Model 3: The design matrix has a 18th order polynomial.

Next, Fig. 3.18 shows a comparison between the regression models and the numerical solution of the nonlinear two-point boundary value problem for models 1, 2, and 3. Finally, the function **multiple_Holdout** is used to validate the three models. Table 3.4 summarizes the validation results.

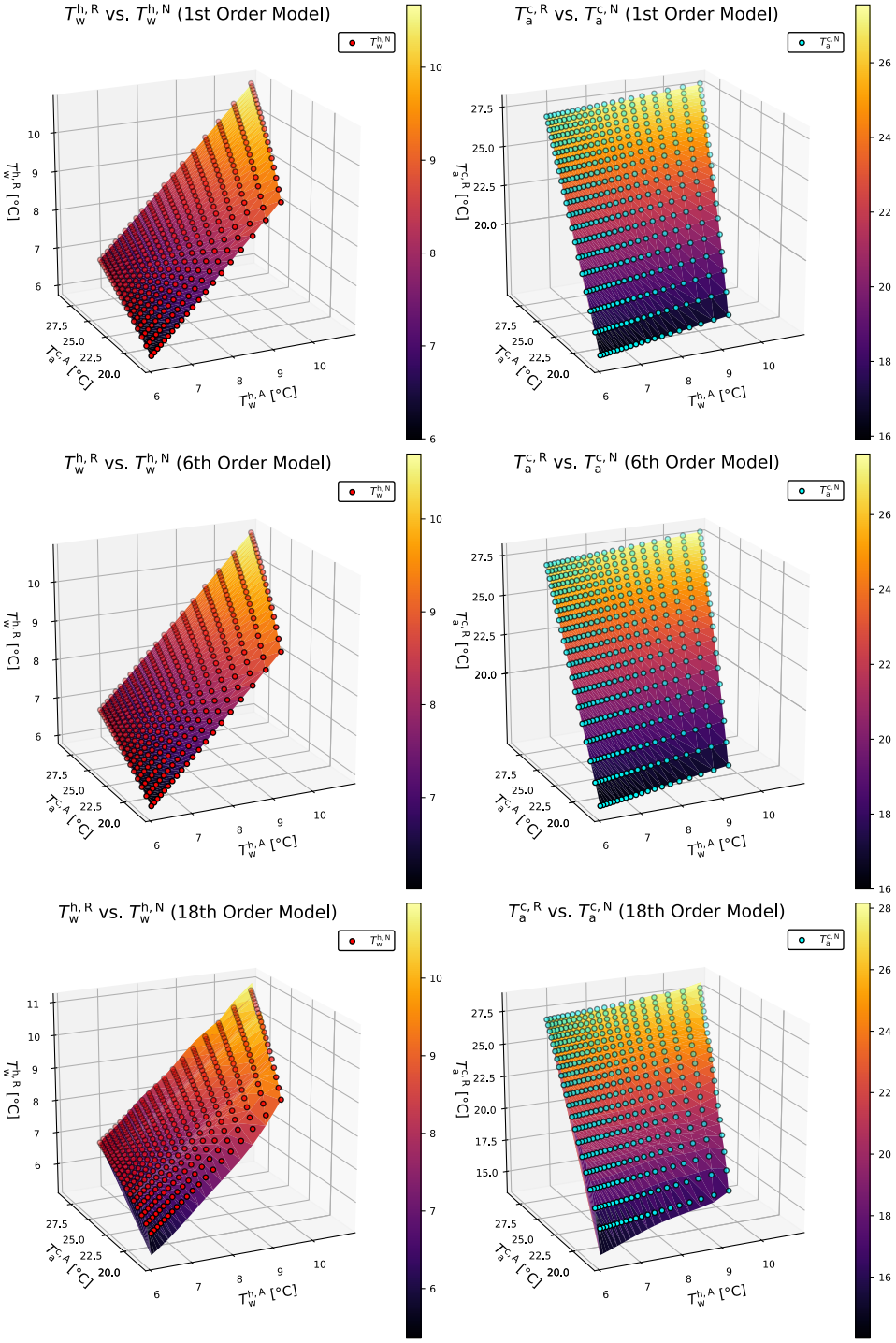


Figure 3.12: Case 2: Comparison between the regression models (the surfaces in the figure) and the numerical solution of the nonlinear two-point boundary value problem (the data points) for a 1st order, a 6th order, and an 18th order polynomial regression models.

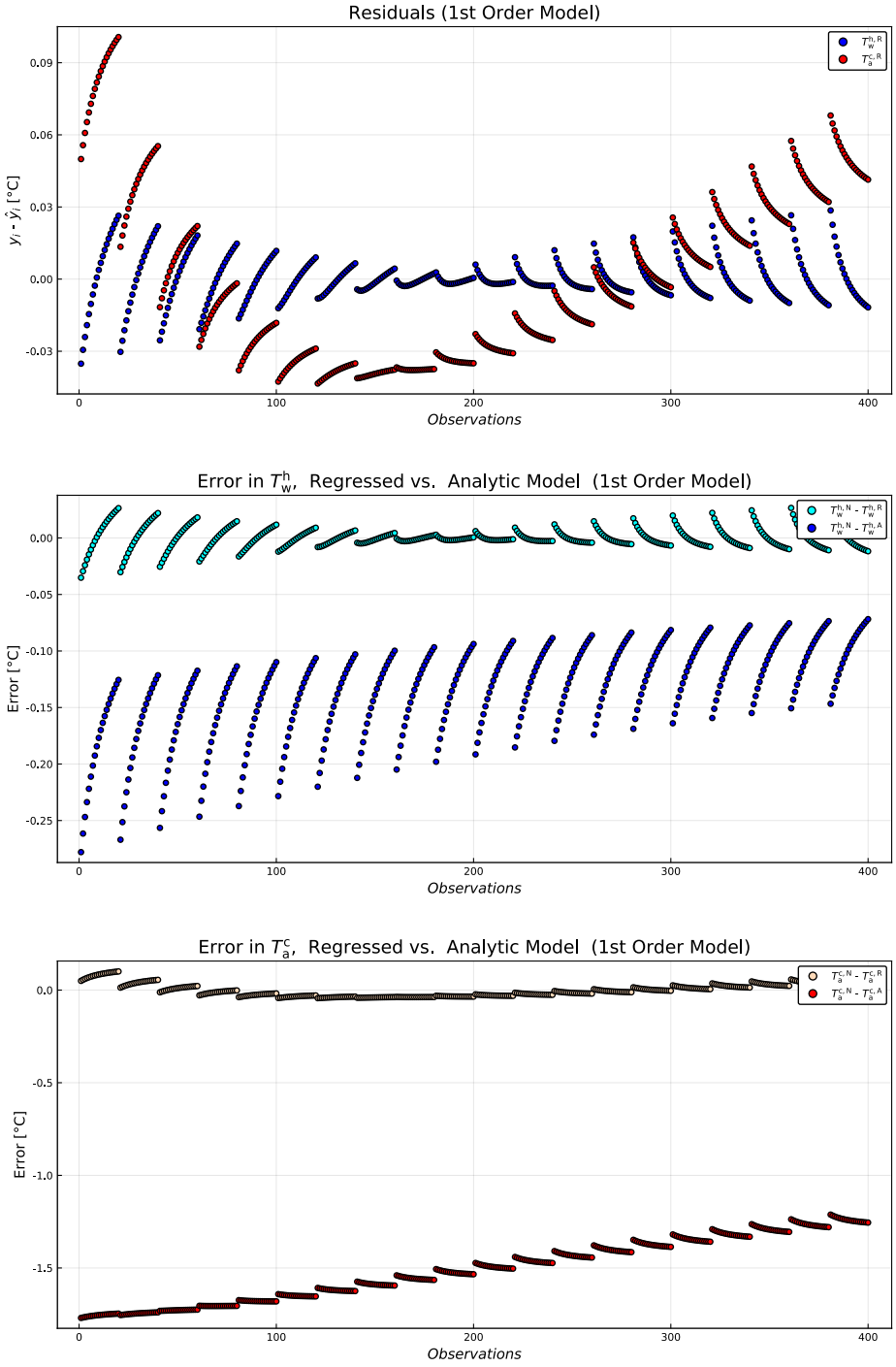


Figure 3.13: Case 2: The residuals, and the errors in $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$, regressed vs. analytic model (1st order model).

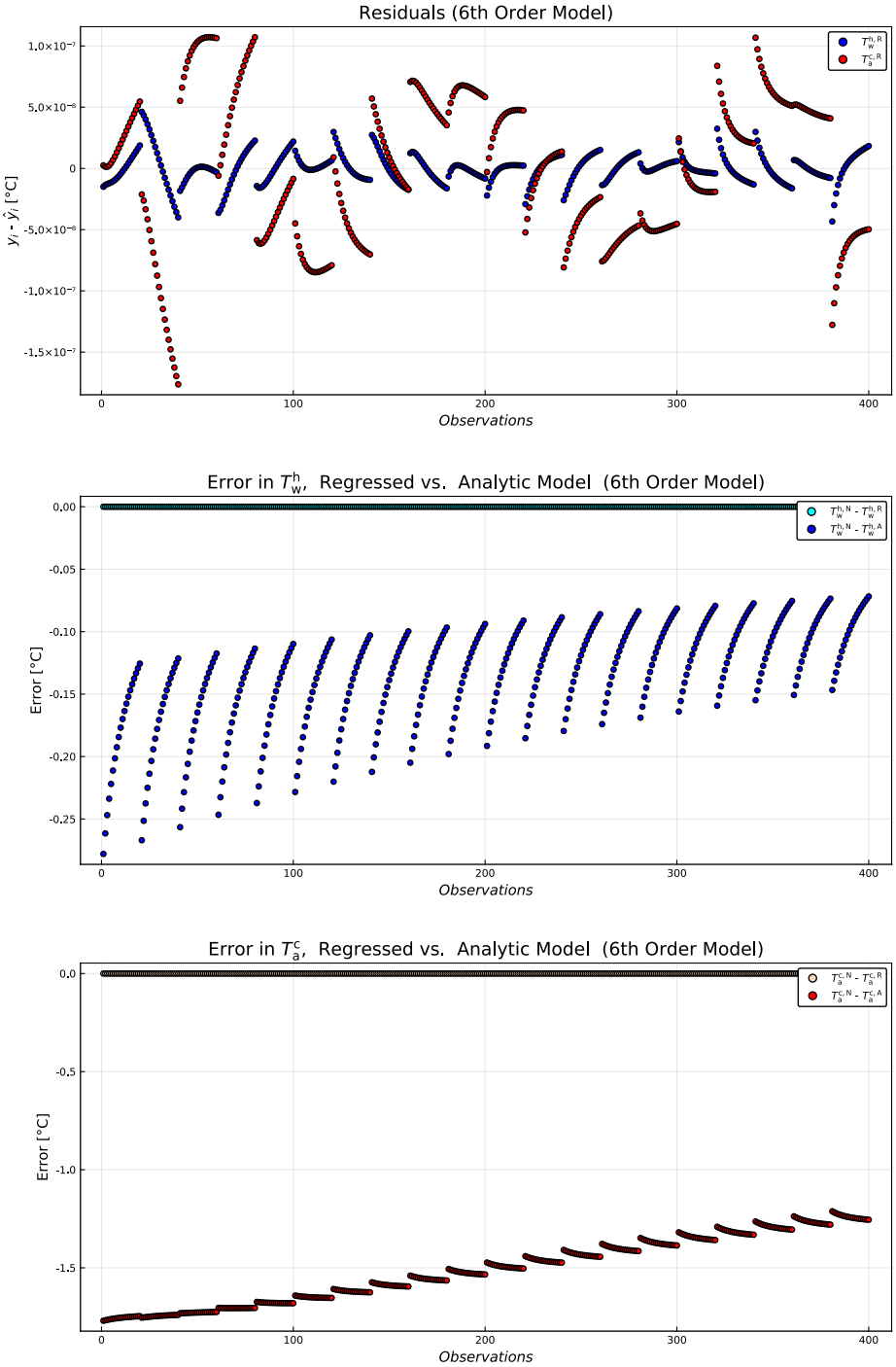


Figure 3.14: Case 2: The residuals, and the errors in T_w^h and T_a^c, regressed vs. analytic model (6th order model).

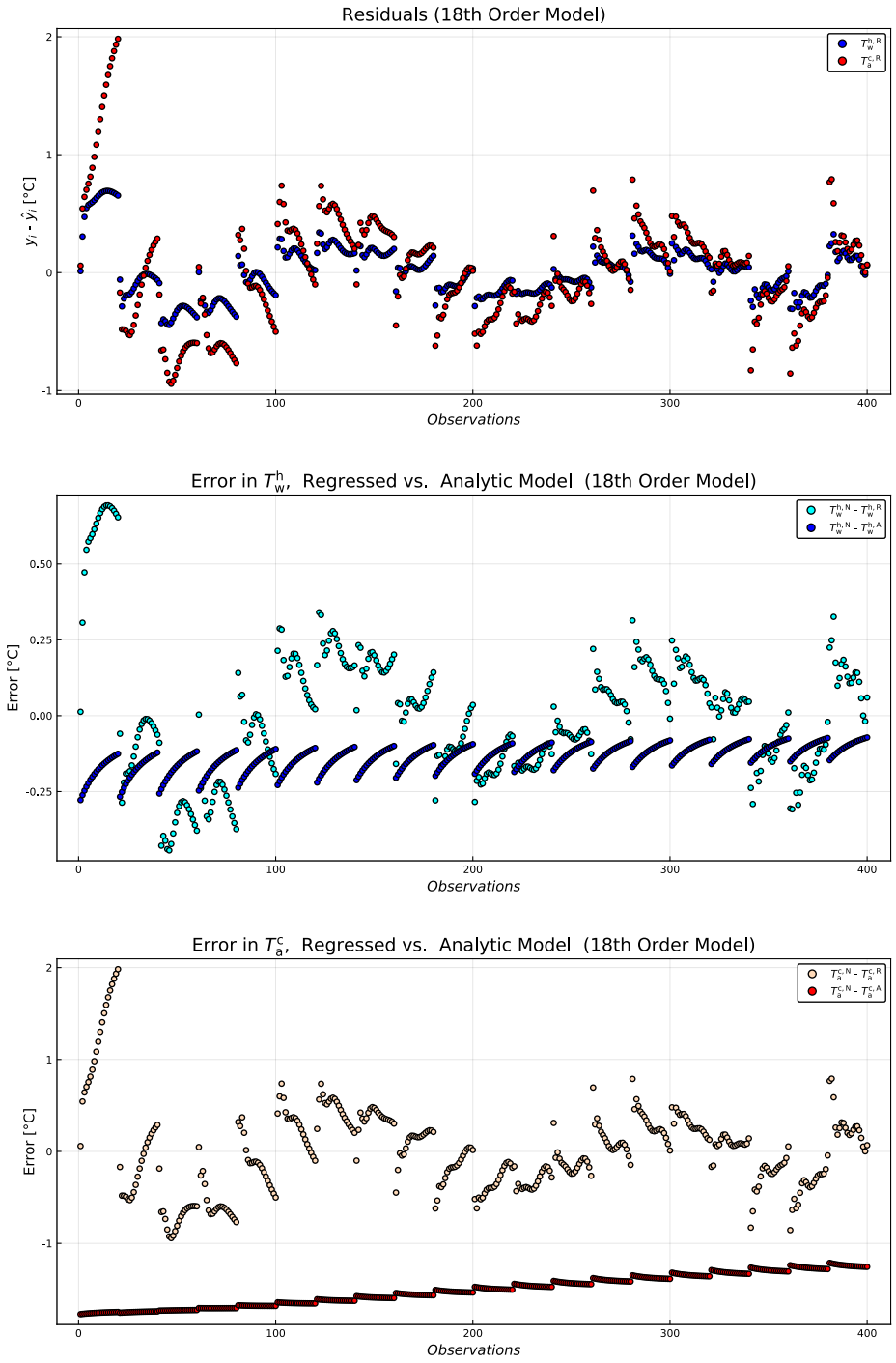


Figure 3.15: Case 2: The residuals, and the errors in $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$, regressed vs. analytic model (18th order model).

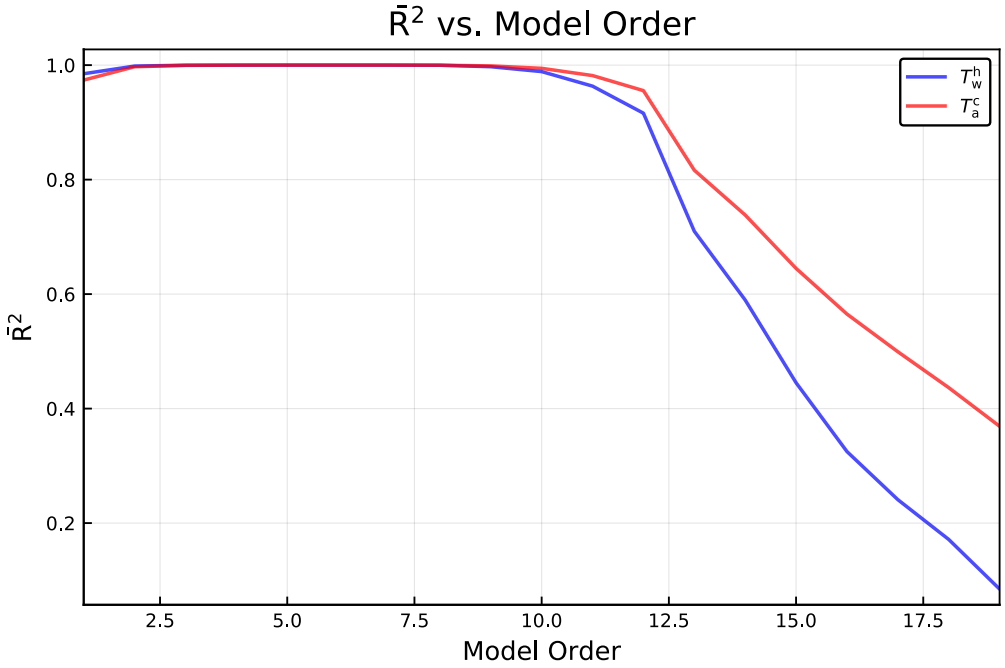


Figure 3.16: Case 2.B: $\bar{R}^2$ vs. model order.

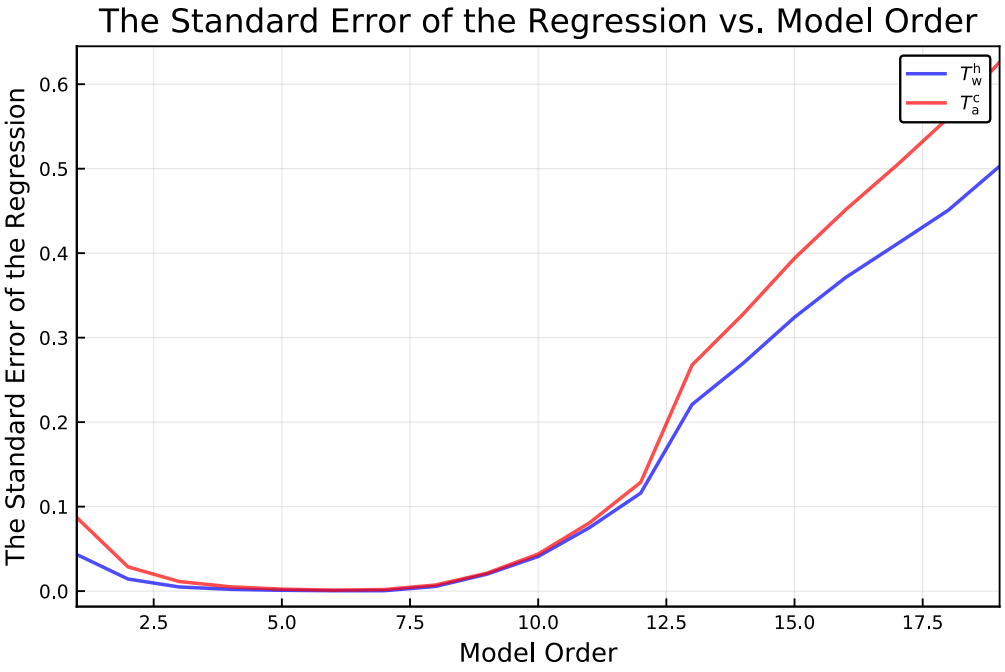


Figure 3.17: Case 2.B: The standard error of the regression vs. model order.

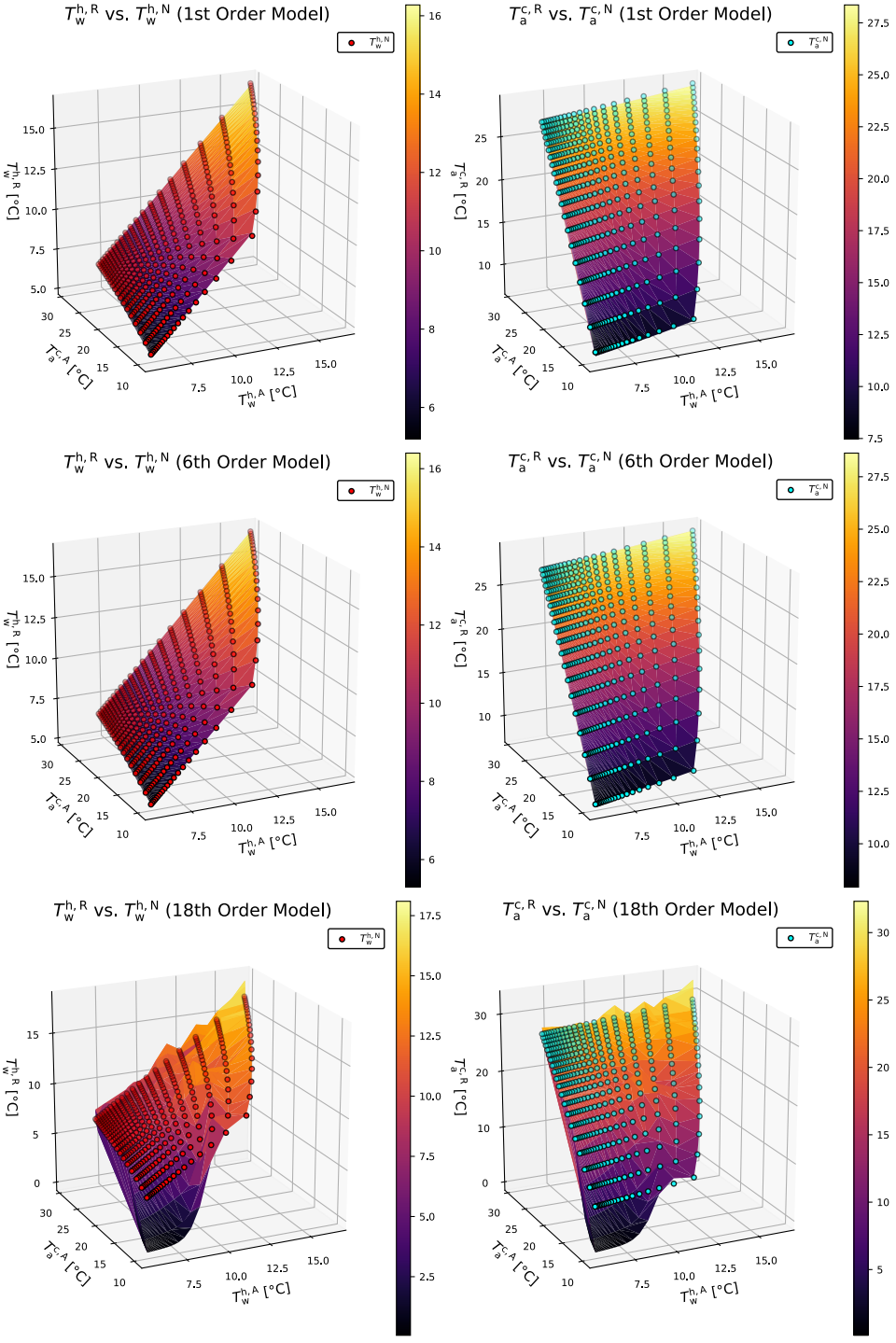


Figure 3.18: Case 2.B: Comparison between the regression models (the surfaces in the figure) and the numerical solution of the nonlinear two-point boundary value problem (the data points) for a 1st order, a 6th order, and an 18th order polynomial regression models.

Model order	Avg. s, °C		Avg. RMSE, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.05227211	0.10468139	0.0381016	0.15271355	0.98457848	0.97338327
6th	0.00070054	0.00151362	$6.16e-6$	$2.806e-5$	0.99999751	0.99999512
18th	2.86256012	3.58890931	8.46572521	12.65635149	−1.07405676	−0.57264847

Table 3.4: Case 2.B: Validation results.

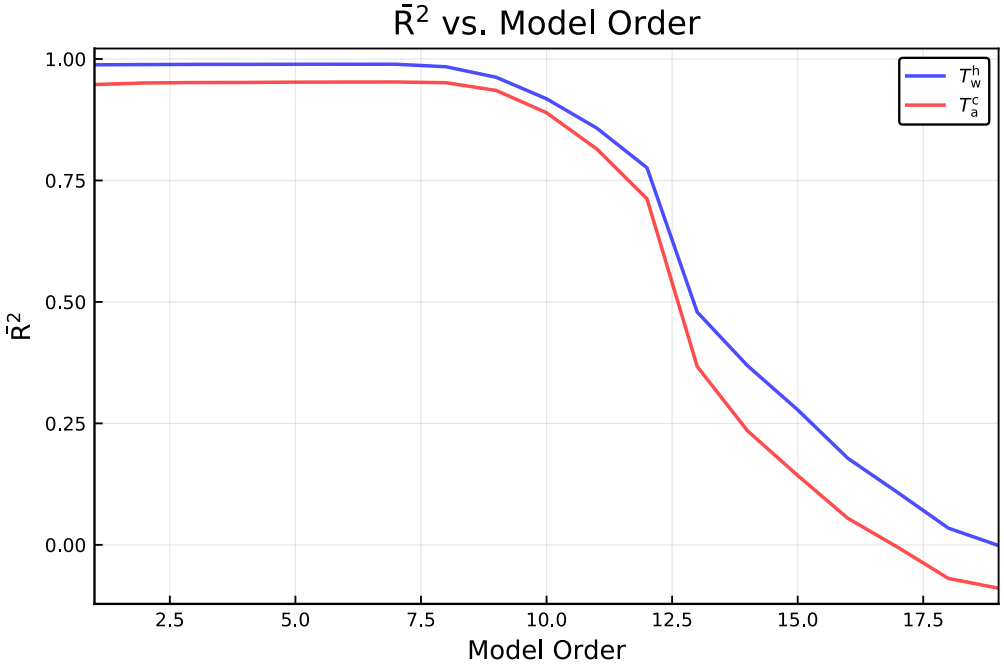


Figure 3.19: Case 3.A: $\bar{R}^2$ vs. model order.

3.3.2.4 Simulation results of case 3.A

This case is the most comprehensive yet, that is because the dataset is generated on a wider range than previously. In a similar manner to the previous cases, first, the best-fit polynomial order of the design matrix is found using the **goodness_of_fit** function. Figure 3.19 and Fig. 3.20 show the impact of increasing the polynomial order of the design matrix on the adjusted R-squared ($\bar{R}^2$), and on the standard error of the regression, respectively.

Based on Fig. 3.19 and Fig. 3.20, the following models were chosen to demonstrate the goodness of fit:

- Model 1: The design matrix has a 1st order polynomial.
- Model 2: The design matrix has a 6th order polynomial.

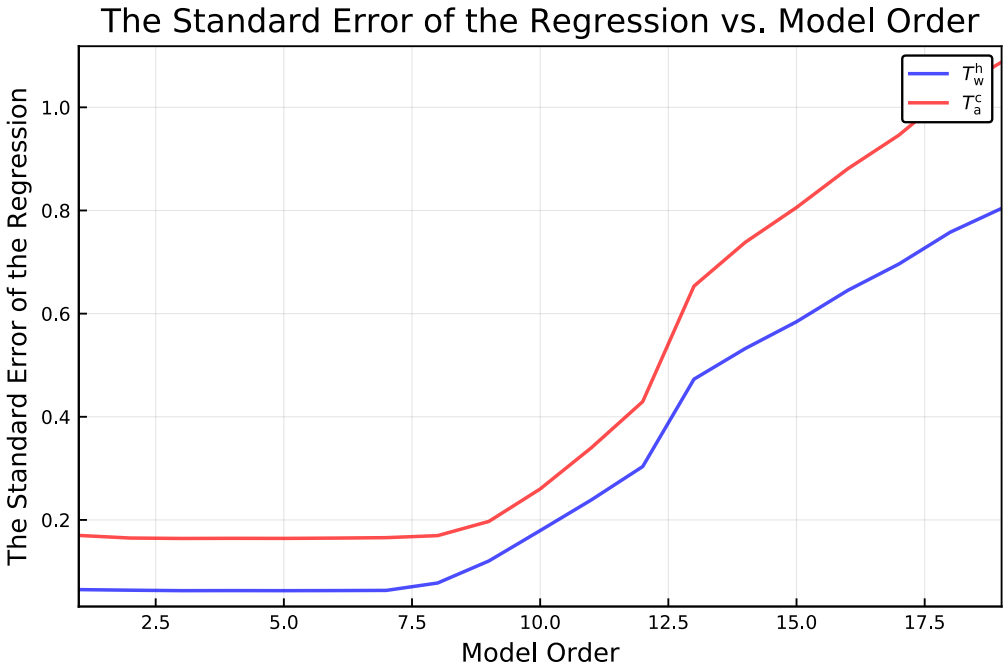


Figure 3.20: Case 3.A: The standard error of the regression vs. model order.

Model order	Avg. s, °C		Avg. RMSE, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.0775944	0.20320863	0.10555031	0.72372327	0.98797304	0.94702488
6th	0.07866278	0.20599263	0.09975618	0.68380548	0.98862523	0.94991916
18th	2.82202043	3.55549681	71.74191069	111.37147539	−4.03230806	−4.15195954

Table 3.5: Case 3.A: Validation results.

- Model 3: The design matrix has a **18th** order polynomial.

Next, Fig. 3.21 shows a comparison between the regression models and the numerical solution of the nonlinear two-point boundary value problem for models 1, 2, and 3. In addition, Fig. 3.22, Fig. 3.23, and Fig. 3.24 show the residuals, and the errors in T_w^h and T_a^c(regressed vs. analytic model) for models 1, 2, and 3, respectively. Finally, the function **multiple_Holdout** is used to validate the three models. Table 3.5 summarizes the validation results.

3.3.2.5 Simulation results of case 3.B

In this case, the impact of adding the mass flow rates as regressors on the goodness-of-fit is examined. In a similar manner to the previous cases, first, the best-fit polynomial order of the design matrix is found using the **goodness_of_fit** function. Figure 3.25

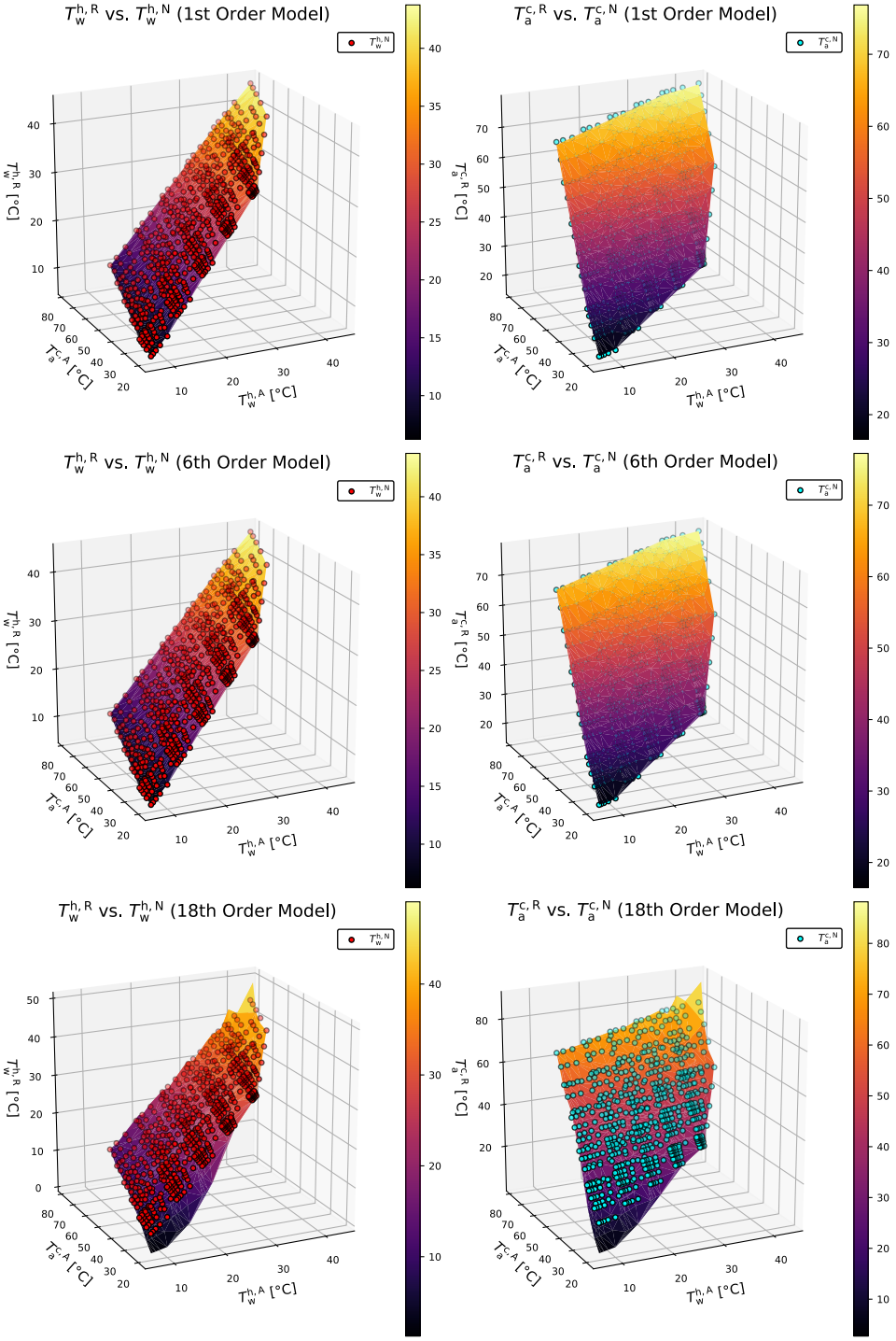


Figure 3.21: Case 3.A: Comparison between the regression models (the surfaces in the figure) and the numerical solution of the nonlinear two-point boundary value problem (the data points) for a 1st order, a 6th order, and an 18th order polynomial regression models.

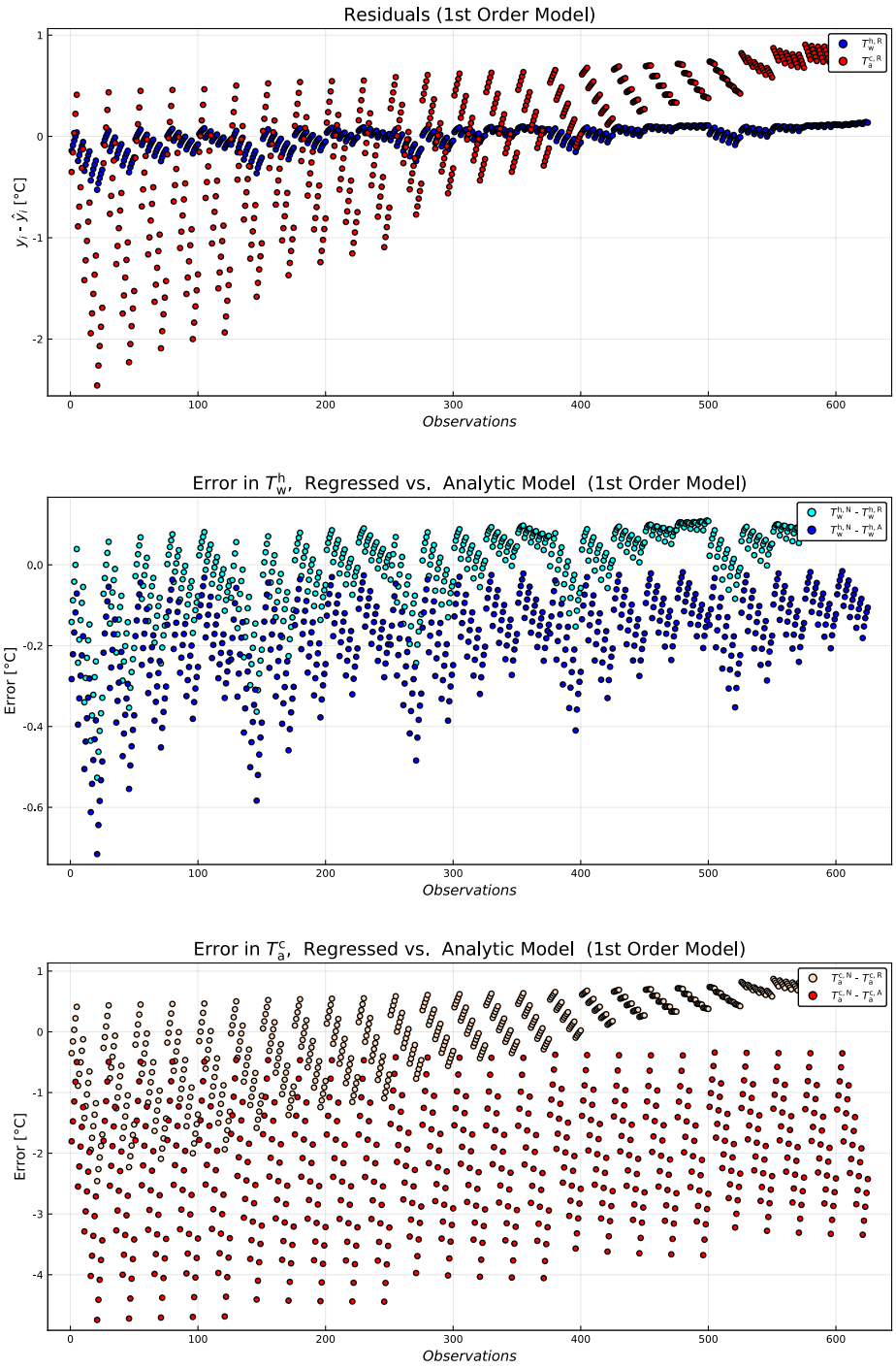


Figure 3.22: Case 3.A: The residuals, and the errors in T_w^h and T_a^c, regressed vs. analytic model (1st order model).

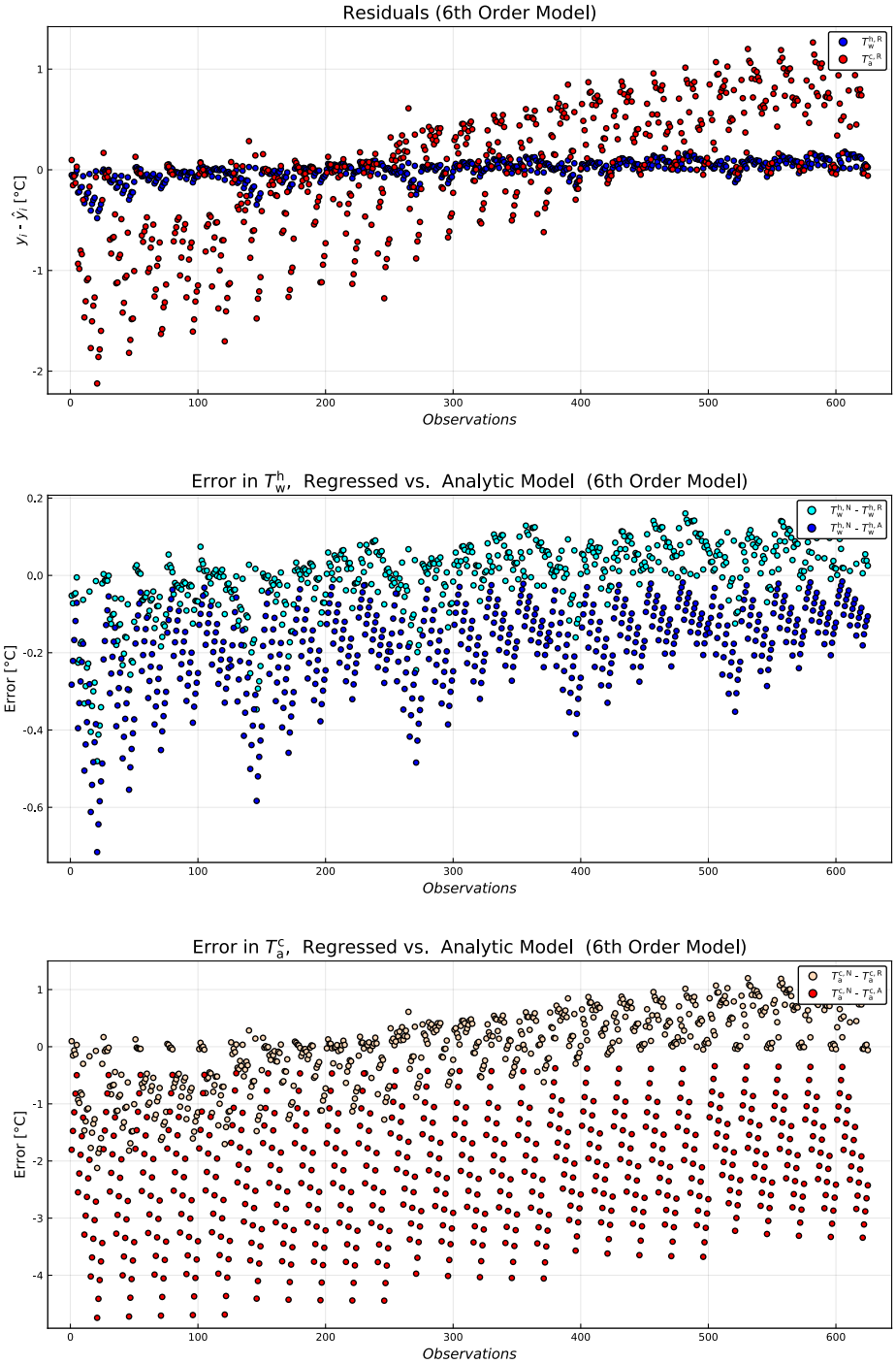


Figure 3.23: Case 3.A: The residuals, and the errors in T_w^h and T_a^c, regressed vs. analytic model (**6th** order model).

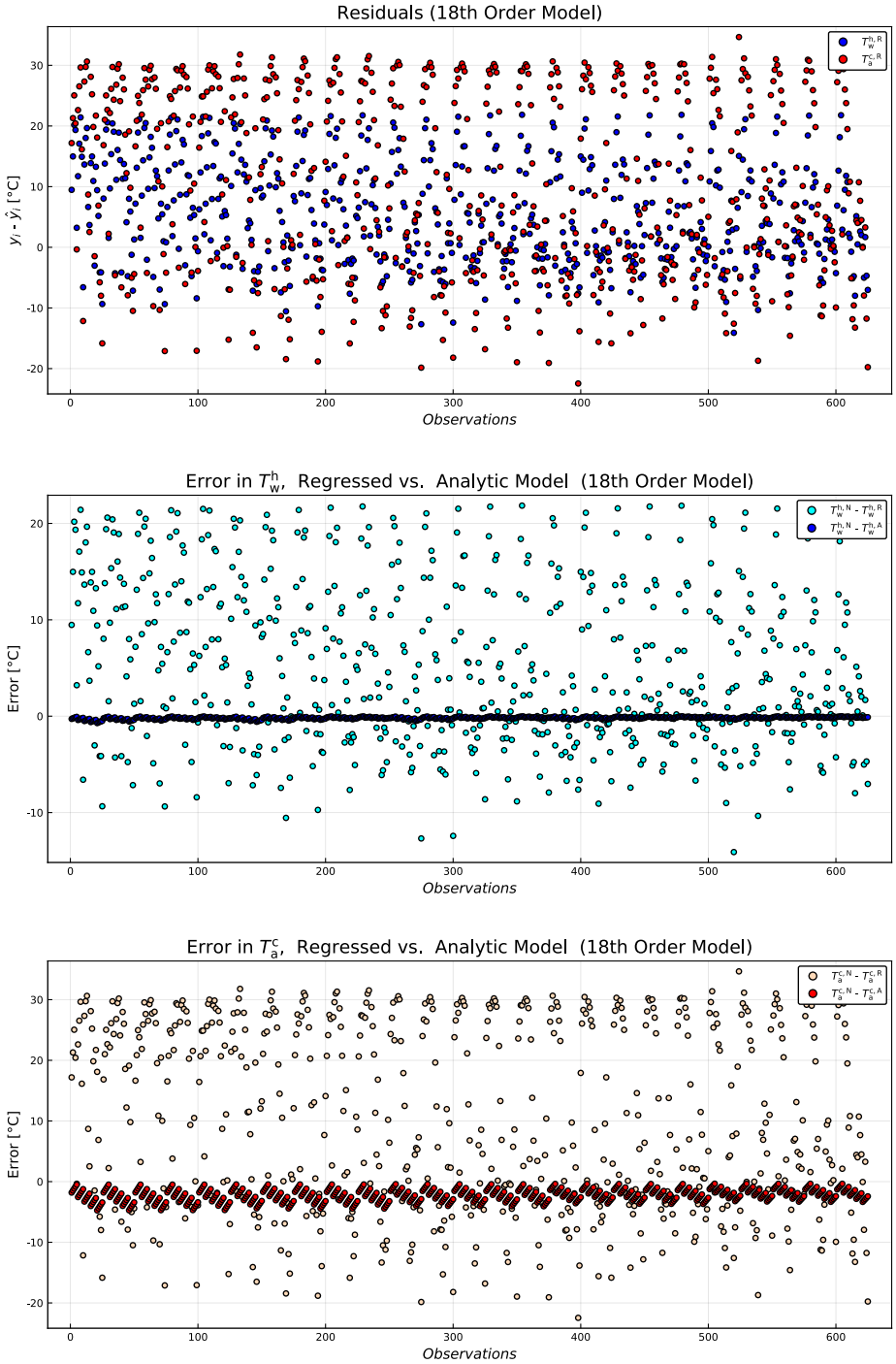


Figure 3.24: Case 3.A: The residuals, and the errors in $T_{\mathrm{w}}^{\mathrm{h}}$ and $T_{\mathrm{a}}^{\mathrm{c}}$, regressed vs. analytic model (18th order model).

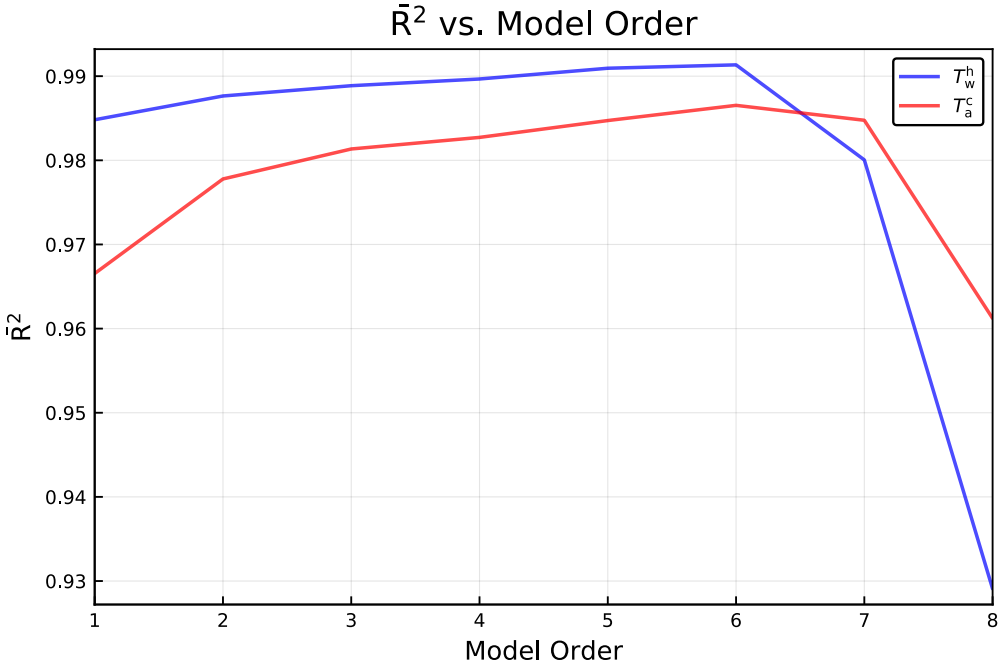


Figure 3.25: Case 3.B: $\bar{R}^2$ vs. model order.

Model order	Avg. s, °C		Avg. RMSE, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.05293457	0.10581149	0.19796433	0.79091089	0.98482044	0.96653181
6th	0.04148825	0.06945538	0.11662211	0.32683733	0.99105622	0.98617308

Table 3.6: Case 3.B: Validation results.

and Fig. 3.26 show the impact of increasing the polynomial order of the design matrix on the adjusted R-squared ($\bar{R}^2$), and on the standard error of the regression, respectively.

Based on Fig. 3.25 and Fig. 3.26, the following models were chosen to demonstrate the goodness of fit:

- Model 1: The design matrix has a **1st** order polynomial.
- Model 2: The design matrix has a **6th** order polynomial.

Next, Fig. 3.27 shows a comparison between the regression models and the numerical solution of the nonlinear two-point boundary value problem for the two models. Finally, the function **multiple_Holdout** is used to validate the two models. Table 3.6 summarizes the validation results.

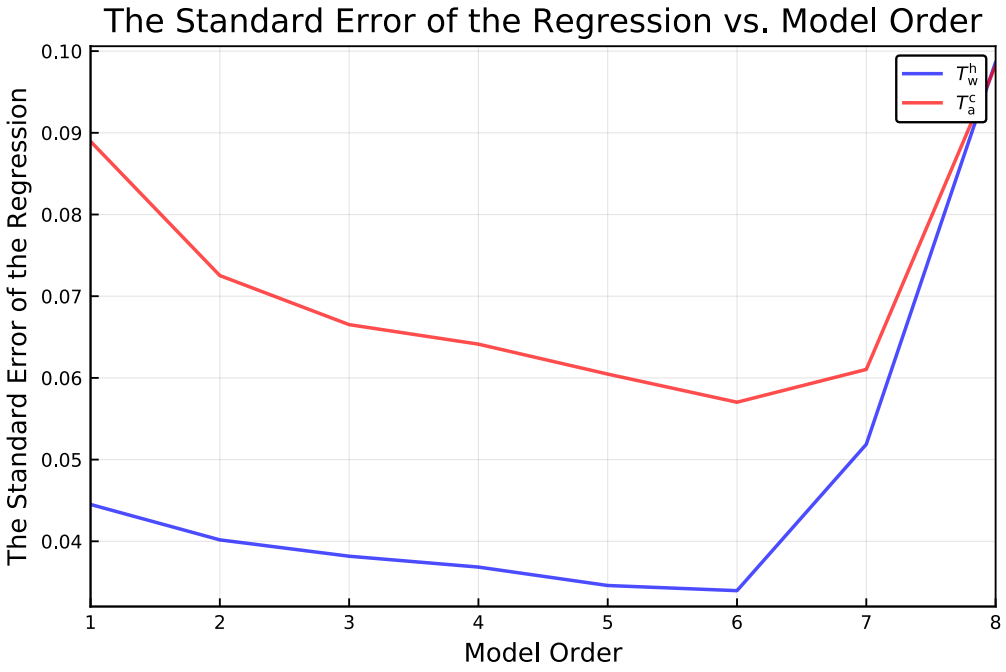


Figure 3.26: Case 3.B: The standard error of the regression vs. model order.

Model order	Avg. s, °C		Avg. RMSE, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.05941832	0.12447671	0.24952088	1.09498773	0.98087158	0.9536708
6th	0.05634448	0.11860076	0.22325409	0.98907569	0.98288539	0.95815315

Table 3.7: Case 3.C: Validation results.

3.3.2.6 Simulation results of case 3.C

In a similar manner to the previous cases, first, the best-fit polynomial order of the design matrix is found using the **goodness_of_fit** function. Figure 3.28 and Fig. 3.29 show the impact of increasing the polynomial order of the design matrix on the adjusted R-squared ($\bar{R}^2$), and on the standard error of the regression, respectively. Then, the function **multiple_Holdout** is used to validate the two models. Table 3.7 summarizes the validation results.

3.4 Nonlinear Regression

In this work, the nonlinear regression of the counter-current heat exchanger model is implemented in Julia in Appendix B.4 using the package Flux.jl (Innes et al., 2018).

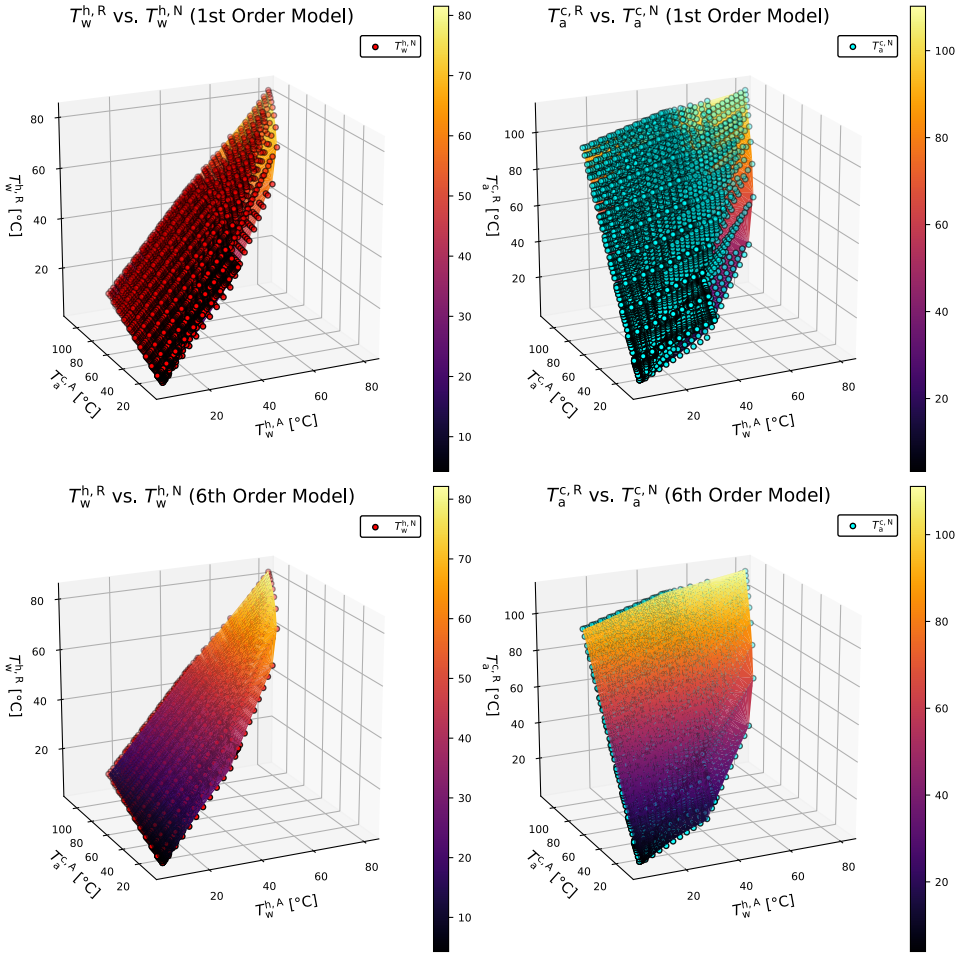


Figure 3.27: Case 3.B: Comparison between the regression models (the surfaces in the figure) and the numerical solution of the nonlinear two-point boundary value problem (the data points) for a 1st order and a 6th order polynomial regression models.

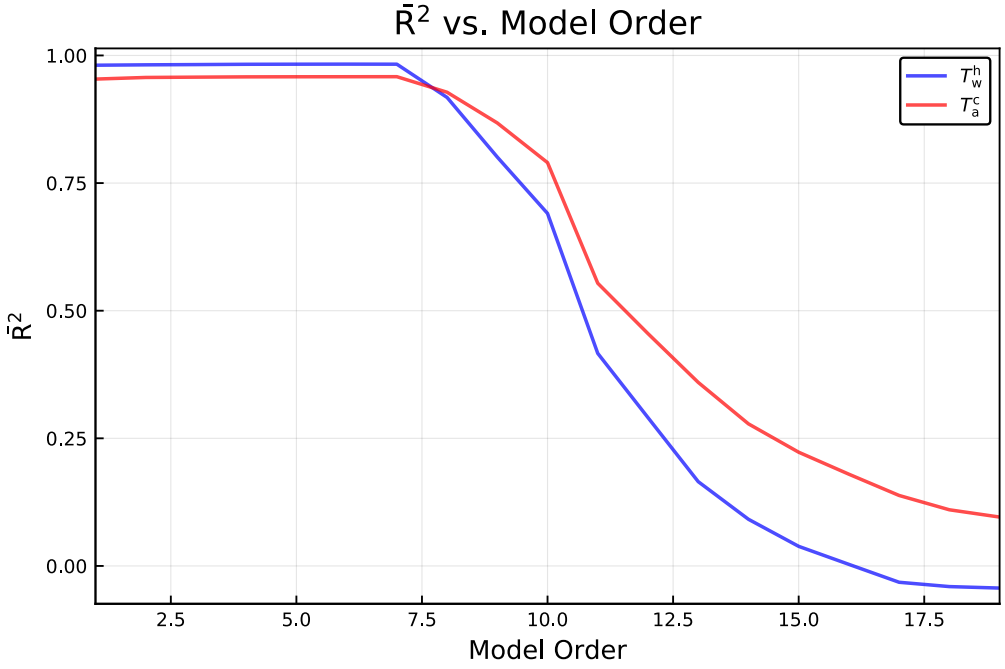


Figure 3.28: Case 3.C: $\bar{R}^2$ vs. model order.

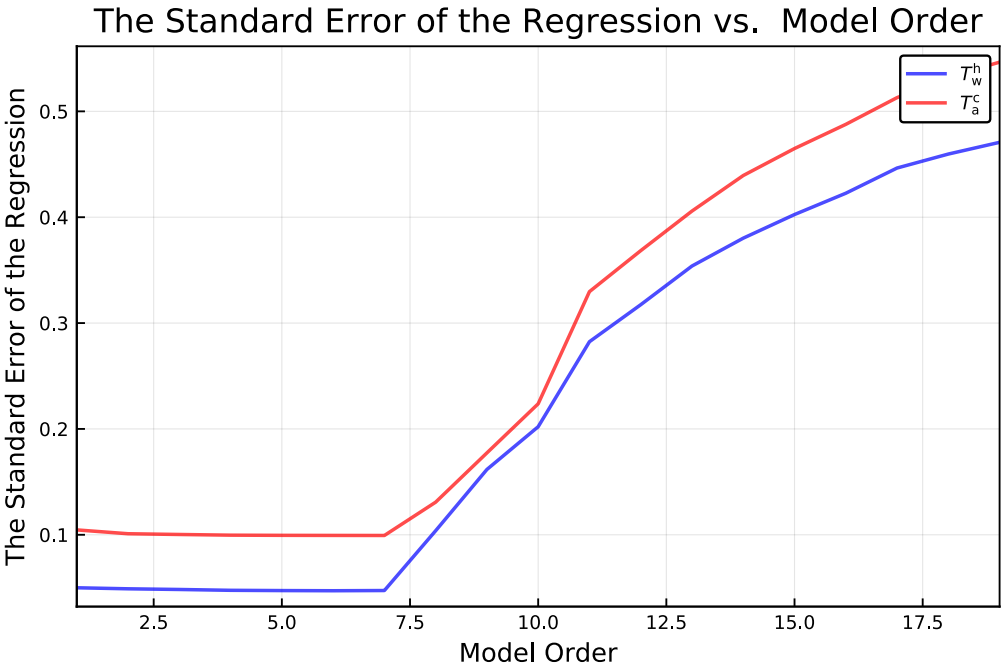


Figure 3.29: Case 3.C: The standard error of the regression vs. model order.

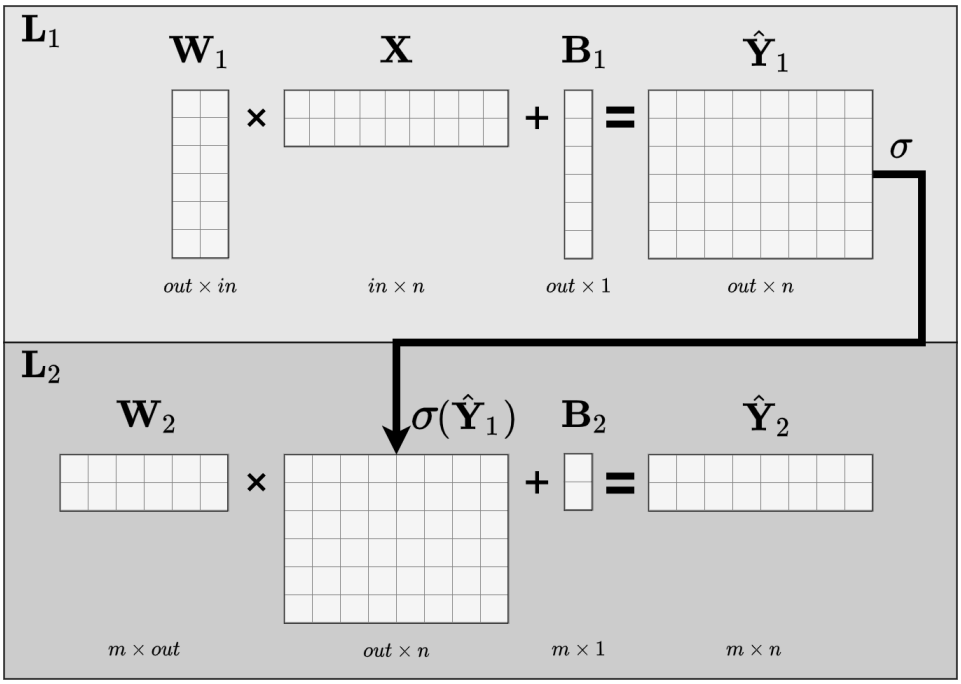


Figure 3.30: Matrix form of a feedforward neural network with two dense layers and an activation function in between them.

Moreover, the nonlinear mapping between the analytic solution of the ideal counter-current heat exchanger model and the numeric solution of the non-ideal heat exchanger model (the case of temperature dependence in the specific heat capacities of air and water) is achieved using the logistic (also known as sigmoid) activation function ($\boldsymbol{\sigma}$), which is introduced between two linear layers in the classical Feedforward Neural Network (**FNN**). Furthermore, the **FNN** is implemented using the description in Lie (2019b) and Flux documentation.

In this work, the nonlinear regression model is composed of two dense layers with the non-linearity ($\boldsymbol{\sigma}$) between them as illustrated in Fig. 3.30. Also, note that the activation function of the second layer is the identity function ($\boldsymbol{\sigma}(\hat{\mathbf{Y}}_2) = \hat{\mathbf{Y}}_2$).

In a similar manner to linear regression, to select the dimension (**out**) of the layers in Fig. 3.30 that gives the best fit, the function **multiple_Holdout** was created to run the multiple holdout method. In addition, the datasets were generated under the same conditions of case 3 of linear regression. The results of the multiple holdout method are presented in Fig. 3.31. It is important to add that the datasets are normalized before model training begins. Normalization is performed to bring the values of the regressors to a common scale, where each regressor influences the result according to its importance as a predictor. The following formula is used to normalize the data:[27]

[27]Swetha Lakshmanan (May 17, 2019). *How, When and Why Should You Normalize/ Standardize/*

Model	Median time (ms)	Mean time (ms)
Hybrid solution (Linear regression)	0.0264	0.035998
Hybrid solution (Nonlinear regression)	0.00123	0.001502
Numeric solution	19.316	21.540

Table 3.8: Benchmark results: Hybrid vs. numeric solution of the non-ideal heat exchanger model.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{3.36}$$

According to Lakshmanan (2019), the approach to normalization in Eq. 3.36 is sensitive to outliers. For example, Fig. 3.32 shows the data, the data after normalization, and the results of the nonlinear regression for a model trained using the dataset of case (3.C), respectively.

3.5 Comparison of the Execution Speed of the Hybrid and the Numeric Non-Ideal Heat Exchanger Models

In Section 2.4, the execution speed of the non-ideal heat exchanger model with temperature dependence in the specific heat capacities of air and water was compared to the execution speed of the ideal heat exchanger model. In addition, the execution speed of the thermal model of an air-cooled synchronous generator was compared in Table 2.7 for the heat exchanger models. In this section, the execution speeds of the data-driven models of the heat exchanger, which were developed in Section 3.3 and Section 3.4, are compared to the execution speed of the non-ideal heat exchanger model (temperature dependence in the specific heat capacities requires solving a nonlinear two-point boundary value problem numerically).

The comparison of the execution speeds is implemented in Appendix B.5, where the explicit data-driven models are expressed as a correction expression to the ideal heat exchanger model. Furthermore, the combination of the mechanistic model of the heat exchanger and the data-driven models forms the hybrid heat exchanger model. Moreover, Table 3.8 shows a comparison of the execution speeds of the hybrid and the numeric solution of the non-ideal heat exchanger model. Also, Table 3.9 shows a comparison of the execution speeds of the hybrid and the numeric solution of the thermal model of an air-cooled synchronous generator with the non-ideal heat exchanger model.

Rescale Your Data? Medium. https://medium.com/@swethalakshmanan14/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff. [Online; accessed 09-May-2020]

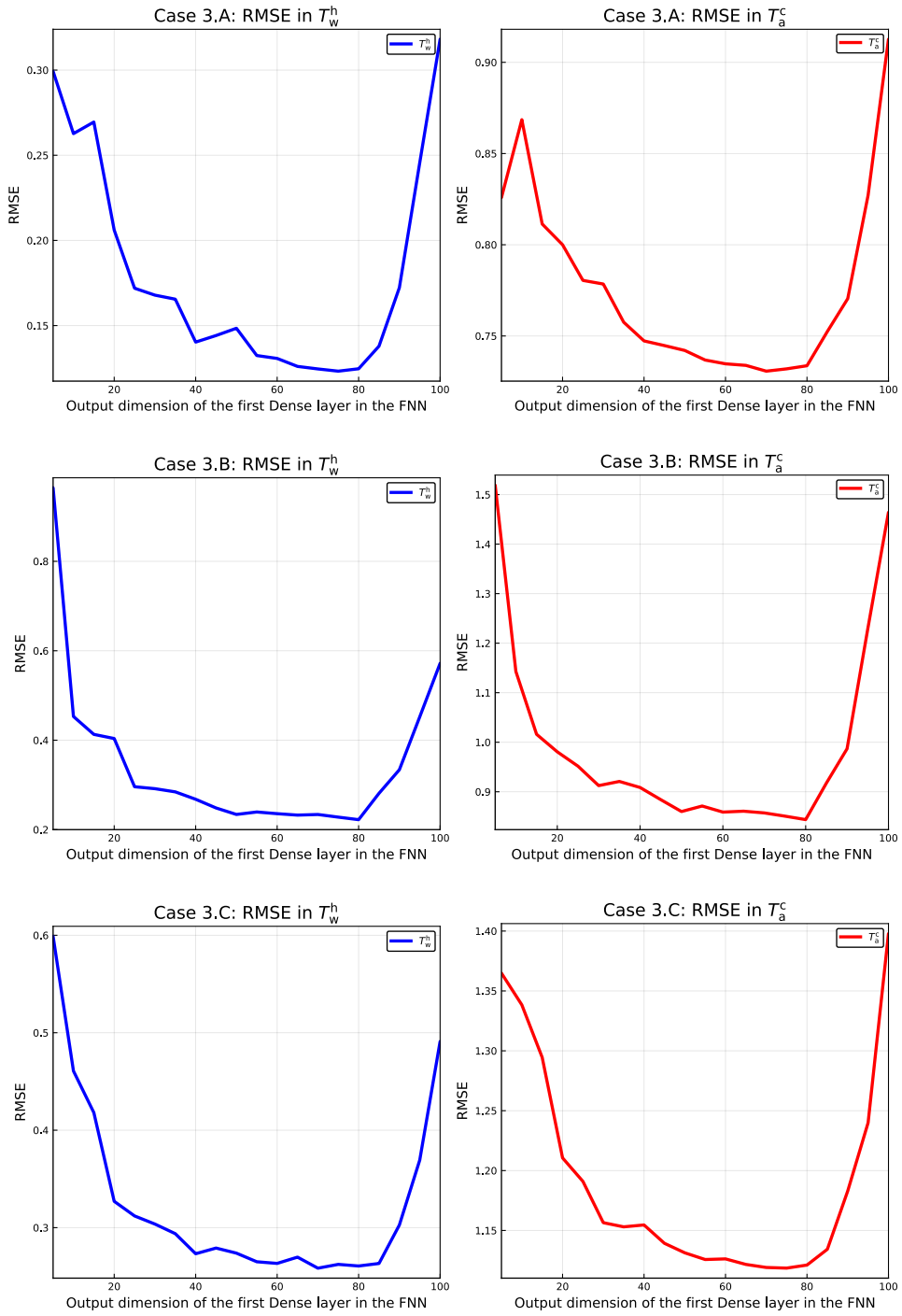


Figure 3.31: Nonlinear regression validation results. The Y-axis is the average **RMSE** for three models, each trained on a random sample for 10000 epoch.

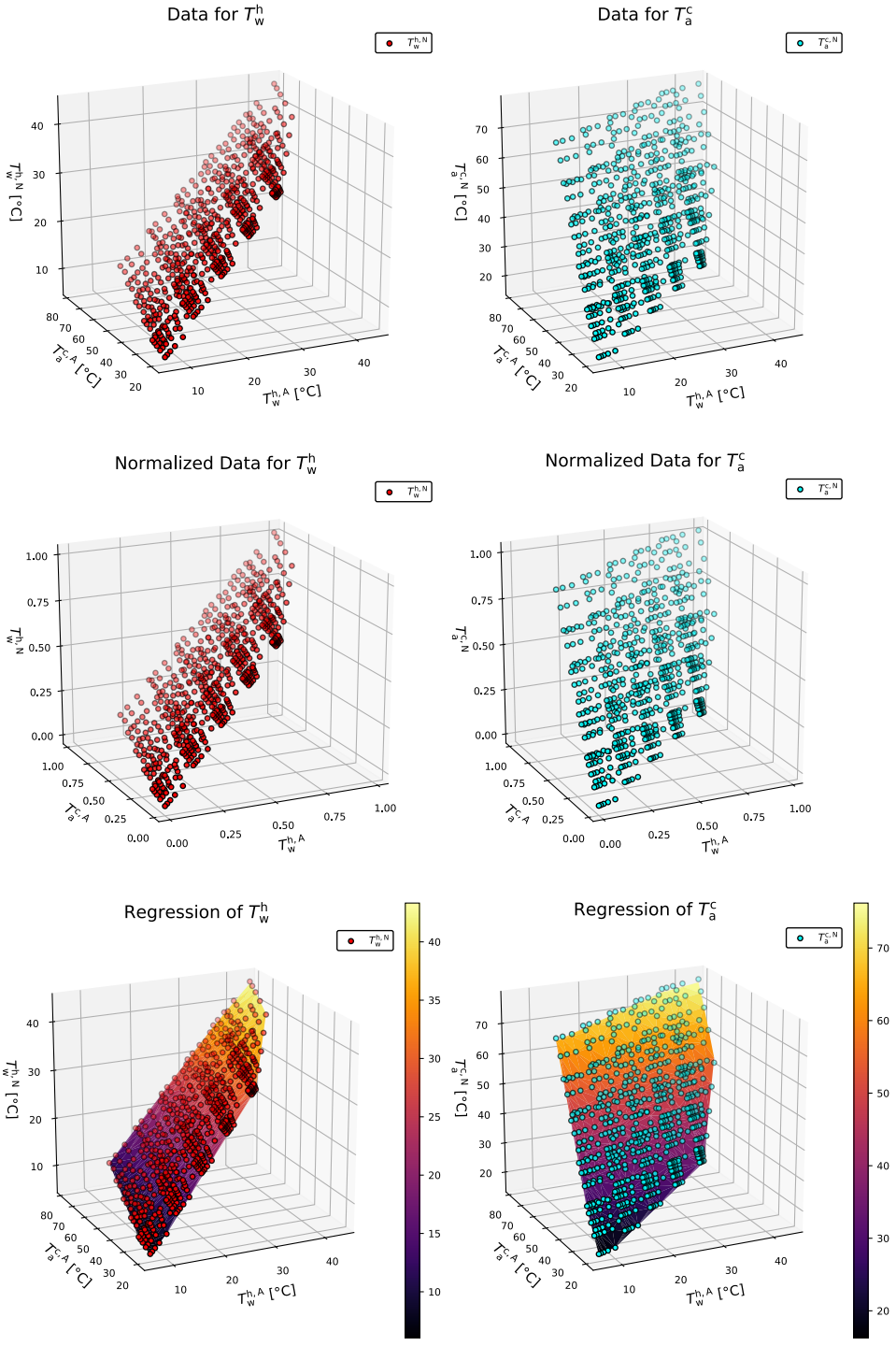


Figure 3.32: Example of nonlinear regression using normalized data.

Model	Median time (s)	Mean time (s)
Hybrid solution (Linear regression)	0.016633	0.019648
Hybrid solution (Nonlinear regression)	0.002216	0.002676
Numeric solution	9.673	9.673

Table 3.9: Benchmark results: Hybrid vs. numeric solution of the thermal model of an air-cooled synchronous generator with the non-ideal heat exchanger model.

Model	Median time (ms)	Mean time (ms)
Hybrid solution (Linear regression)	0.0008535	0.001466
Hybrid solution (Nonlinear regression)	0.00125	0.001469
Numeric solution	19.316	21.540

Table 3.10: Benchmark results for case 4: Hybrid vs. numeric solution of the non-ideal heat exchanger model.

In addition to Table 3.8 and Table 3.9, Fig. 3.33 compares the metal temperatures of the data-driven models with the metal temperatures obtained using the numeric solver for the thermal model of an air-cooled synchronous generator with temperature dependence in the specific heat capacities of water and air.

Finally, new models were created to improve the previous results using a new dataset (case 4) that was generated in Appendix B.3 and Appendix B.4. The new dataset cover a wide range of possible value of $\dot{m}_\mathrm{w}$, $\dot{m}_\mathrm{a}$, T_w^c, and T_a^h, but improves the goodness-of-fit. The reasoning behind this new dataset is discussed in Chapter 4. Table 3.10, Table 3.11, and Fig. 3.34 show the improved results.

3.6 Comparison of the Mechanistic Model Predictions and Åbjøra Experimental Data

In Øyvang (2018), a *heat-run* test was performed for the 103 MVA hydro-generator located at Åbjøra in Norway, and measurements of currents, voltage, power, and temperatures were taken for 600 min from a *cold-start*. Also, the *cold-run* lasted up to 53 min, after

Model	Median time (s)	Mean time (s)
Hybrid solution (Linear regression)	0.002052	0.002497
Hybrid solution (Nonlinear regression)	0.002312	0.002745
Numeric solution	9.673	9.673

Table 3.11: Benchmark results for case 4: Hybrid vs. numeric solution of the thermal model of an air-cooled synchronous generator with the non-ideal heat exchanger model.

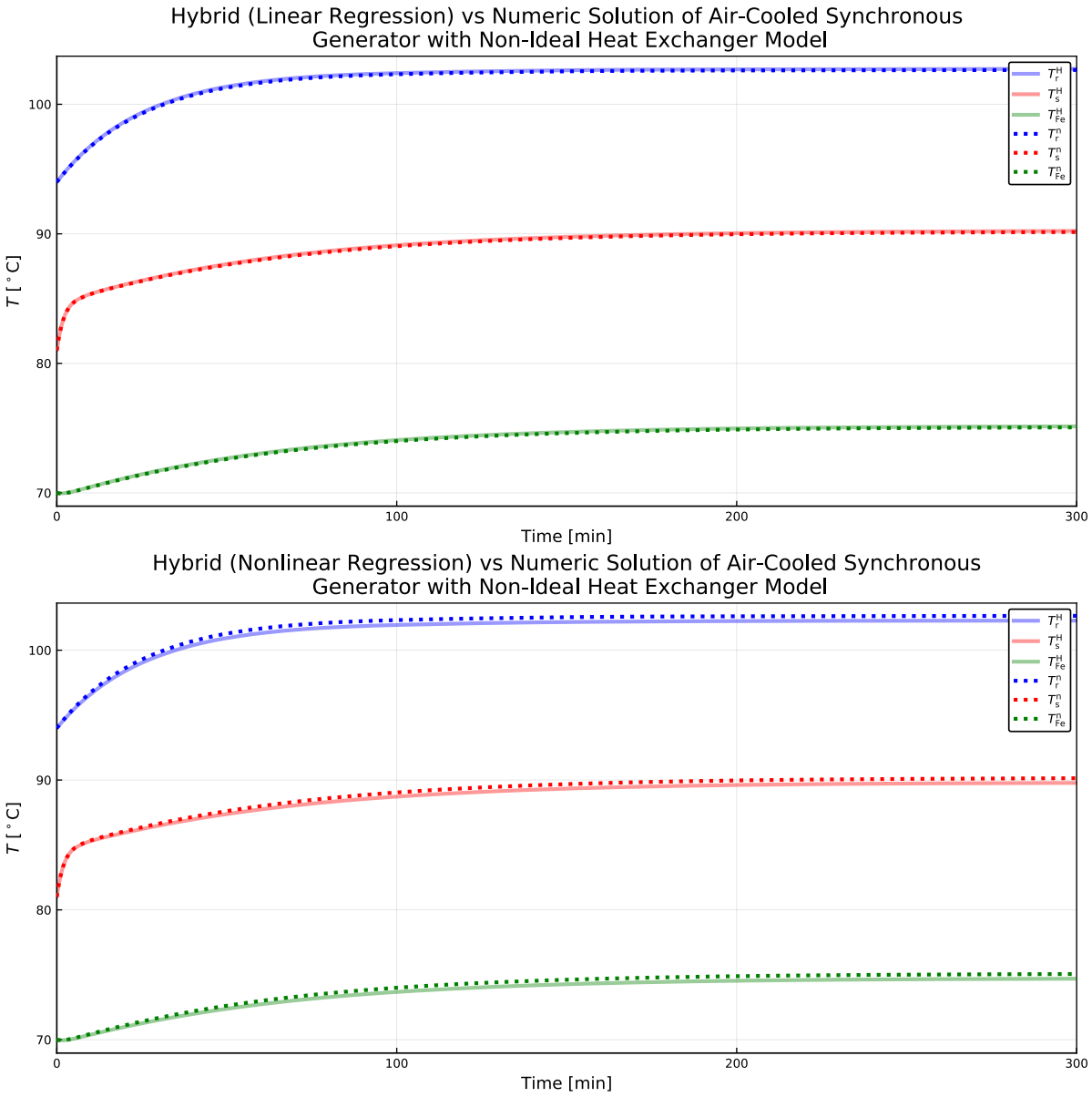


Figure 3.33: Hybrid solution using linear regression (upper figure), and the hybrid solution using nonlinear regression (lower figure) of the thermal model of an air-cooled synchronous generator.

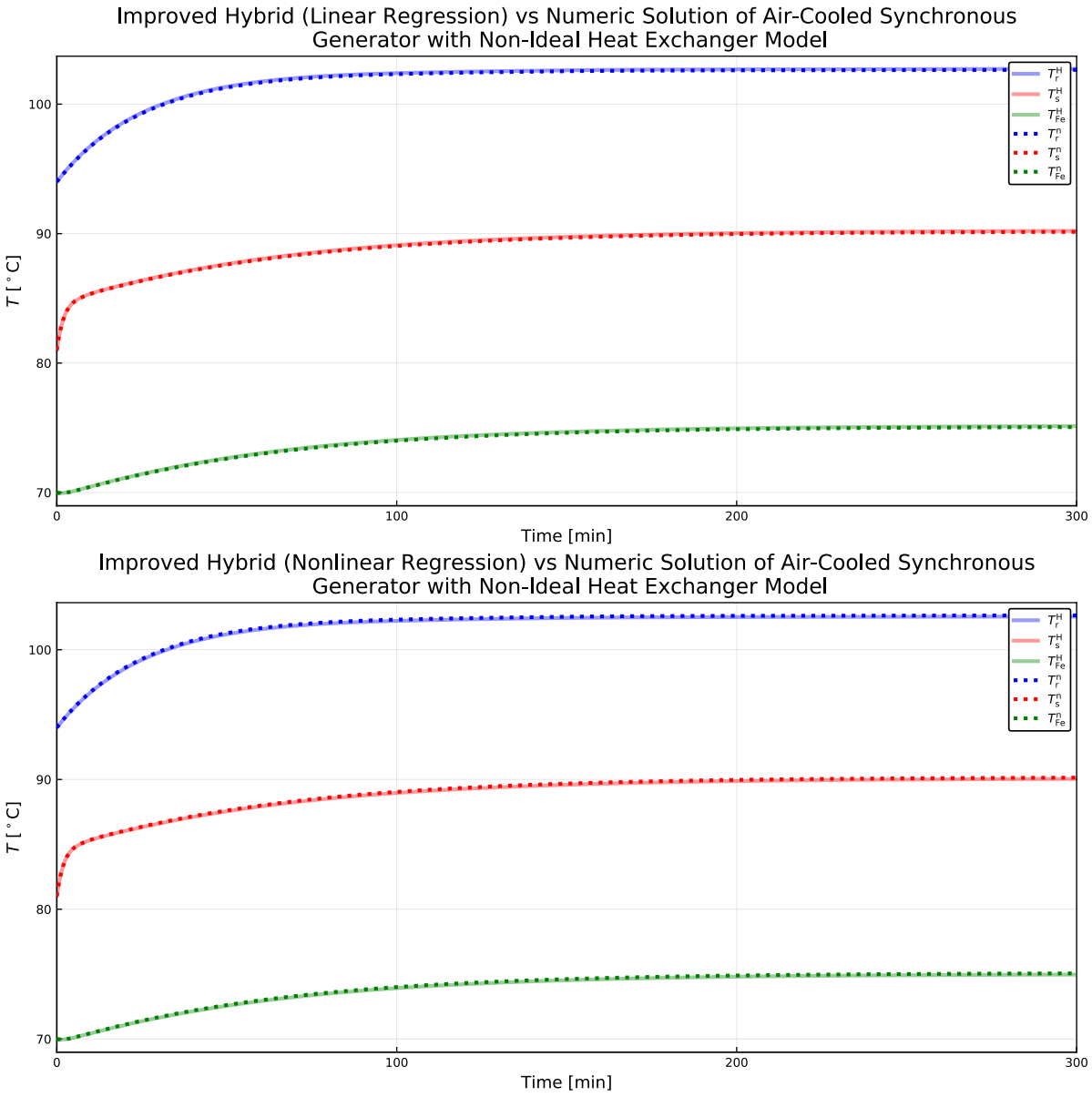


Figure 3.34: Hybrid solution using linear regression (upper figure), and the hybrid solution using nonlinear regression (lower figure) of the thermal model of an air-cooled synchronous generator for case 4.

Symbol	Description	Unit
I_{fd}	Rotor field current	A
I_{t}	Stator terminal current	A
$T_{\mathrm{w}}^{\mathrm{c}}$	Cold water temperature	°C
$T_{\mathrm{w}}^{\mathrm{h}}$	Hot water temperature	°C
$T_{\mathrm{a}}^{\mathrm{h}}$	Hot air temperature	°C
$T_{\mathrm{a}}^{\mathrm{c}}$	Cold air temperature	°C
T_{s}	Temperature of stator copper	°C
T_{Fe}	Temperature of stator iron	°C

Table 3.12: Quantities relevant to the comparison of the mechanistic model predictions and Åbjøra experimental data.

which the field current was increased. In addition, in M. Pandey (2019a), several thermal models of an air-cooled synchronous generator were developed, and the predictions of the thermal models were compared with the experimental data of Øyvang (2018).

In this work, in a similar manner to M. Pandey (2019a), the experimental data of the heat-run test is compared with the predictions of the mechanistic model. However, due to time constraints, parameter optimization is not carried out. The comparison of the mechanistic model predictions and Åbjøra experimental data is implemented in Appendix B.6, and the relevant quantities to the comparison are described in Table 3.12. Figure 3.35 shows the comparison of the mechanistic model predictions and Åbjøra experimental data. Moreover, in Fig. 3.35, the hybrid thermal model of an air-cooled synchronous generator (linear regression using case 4 datasets) is compared with the experimental data of Åbjøra to highlight the impact of the temperature dependence in the specific heat capacities of air and water in the heat exchanger model.

Finally, to describe the error in the mechanistic model (the error between the predictions and the experimental data), the data used in the development of an empirical/data-driven/machine learning model must be informative, that is the data should contain relevant information to the model, and specifically to the model objectives (Lie, 2019a).

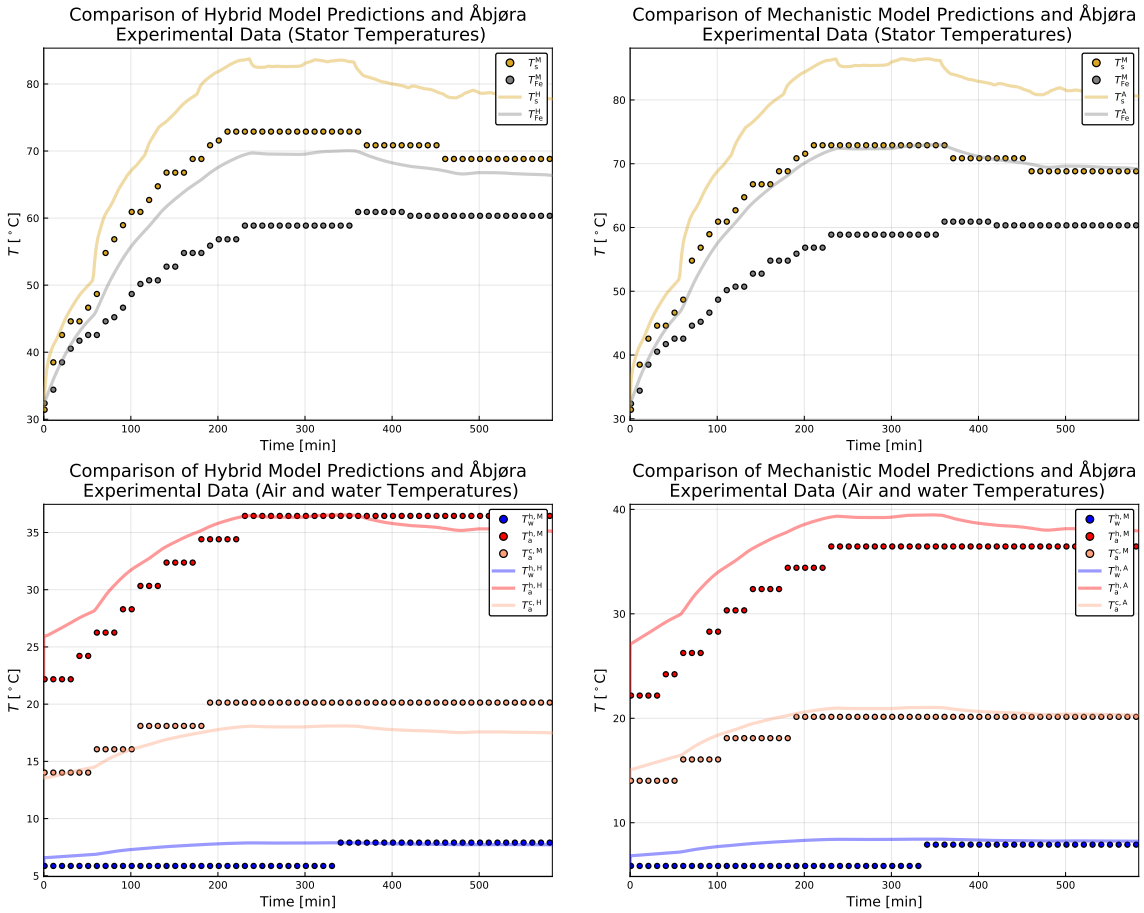


Figure 3.35: Comparison of the hybrid model predictions (When $\hat{c}_p(T)$) and Åbjøra experimental data (Figures on the left). Comparison of the mechanistic model predictions and Åbjøra experimental data (Figures on the right).

4 Results and Discussion

This chapter summarizes the results of the work carried out in earlier chapters and discusses key findings. First, the results of the comparison between the analytic and the numeric solution of the counter-current heat exchanger models are presented. Second, the results of the regression of the counter-current heat exchanger model are examined and explained. Third, the execution speeds of the data-driven models and the numeric solution of the non-ideal heat exchanger model are analyzed and compared. Finally, the comparison between the predictions of the mechanistic model and Åbjøra experimental data are presented.

4.1 Analytic vs. Numeric Solution of the Counter-current Heat Exchanger Model

In Chapter 2, an overview of the thermal model of an air-cooled synchronous generator with ideal and non-ideal heat exchanger models was given. Also, the impact of the temperature dependence in the specific heat capacities of air and water on the solution of the heat exchanger sub-model was discussed. In particular, it was discussed how a linear/nonlinear two-point boundary value problem may form depending on the assumption of temperature dependence in the specific heat capacities. Moreover, in Appendix B.2, the solution of the thermal model of an air-cooled synchronous generator with ideal and non-ideal heat exchanger models was implemented, and it was highlighted how the linear/nonlinear two-point boundary value problem of the heat exchanger sub-model is solved for each time step in the dynamic model.

In addition, as mentioned in the task background in Appendix A, the numerical solution of the linear/nonlinear two-point boundary value problem is relatively costly for online use, which was verified in the benchmark results of Table 2.4, Table 2.5, Table 2.6, and Table 2.7. For convenience, Table 4.1 summarizes the benchmark results that was presented in the previous tables. It can be seen in Table 4.1 that the numeric solution when the specific heat capacities depend on temperature is about 4 to 5 times slower than the numeric solution when the specific heat capacities does not depend on temperature. Also, Table 4.1 shows that the analytic solution is about 350 to 450 times faster than the numeric solution for the case of temperature independence in the specific heat capacities

Benchmark results of the **BVP** solvers available in Julia language when $\hat{c}_p(T)$ in the heat exchanger sub-model		
Method	Median time (ms)	Mean time (ms)
Shooting	20.064	21.741
GeneralMIRK4	846.158	849.103
MIRK4	867.653	873.557
Benchmark results of the heat exchanger sub-model		
Model	Median time (ms)	Mean time (ms)
The analytic solution	0.009	0.01063
The numeric solution when $\hat{c}_p$	3.990	4.798
The numeric solution when $\hat{c}_p(T)$	19.061	20.630
Benchmark results of the thermal model of an air-cooled synchronous generator		
Model	Median time (ms)	Mean time (ms)
The analytic solution	5.387	6.604
The numeric solution when $\hat{c}_p$	2428	2398
The numeric solution when $\hat{c}_p(T)$	9796	9796

Table 4.1: Summary of the benchmark results of Chapter 2.

of air and water. Overall, this confirms the need for an explicit data-driven model to speed up not only the solution time of the heat exchanger sub-model when solving the dynamic thermal model of the generator for online use, but also to reduce the simulation time of state estimation reported in M. Pandey (2019a) of model 3b and model 4b and online parameters optimization of the thermal models in M. Pandey (2019b).

Furthermore, Fig. 2.13 shows that the cold air temperature ($T_{\mathrm{a}}^{\mathrm{c}}$) is affected more than the hot water temperature ($T_{\mathrm{w}}^{\mathrm{h}}$) by the temperature dependence in the specific heat capacities, which is reflected in the temperatures of the synchronous generator metals in Fig. 2.15 by a decrease of about $3.3\,^{\circ}\mathrm{C}$.

Finally, it is important to add that the temperature dependence in the specific heat capacities was implemented by fitting a polynomial to the experimental data in Bergman et al. (2011) over the interval $(0, 100)\,^{\circ}\mathrm{C}$ for water, and over the interval $(-173.15, 126.85)\,^{\circ}\mathrm{C}$ for air. As a consequence, the numerical solution of the non-ideal heat exchanger model is only valid when the model is solved over temperatures in these intervals, and possibly for temperatures in the immediate vicinity.

4.2 Regression of the Counter-Current Heat Exchanger Model

In this work, one of the main tasks is to speed up the solution time of the non-ideal heat exchanger model (the case of temperature dependence in the specific heat capacities of air and water), which was carried out in Chapter 3. Particularly, explicit data-driven models for the non-ideal heat exchanger model were developed by linear and nonlinear regression, which were expressed as a correction expression to the ideal heat exchanger model. Also, when solving the dynamic thermal model of the generator, the explicit data-driven models (which are fast) were used instead of solving the nonlinear two-point boundary value problem (which is slow) in each time step of the DAE solver.

4.2.1 Results and discussion of the linear regression of the counter-current heat exchanger model

In Appendix B.3, linear regression of the heat exchanger model was implemented, and several datasets were generated under a variety of conditions to study the predictive ability of the regression model. First, the results of cases 1 and 2 will be discussed.

4.2.1.1 Datasets of cases 1 and 2

By a brief look at Figs. 3.4, 3.5, 3.10, 3.11, 3.16, and Fig. 3.17 one can tell that a 6th order polynomial in the design matrix of the model gives the best measures of goodness-of-fit for both cases 1 and 2, where the goodness-of-fit of the model is measured by the standard error of the regression and the adjusted coefficient of determination. Moreover, the figures show model overfitting, which was briefly described in Section 3.2 and verified in the validation results of Section 3.3.

In addition, a 1st order, a 6th order, and a 12th order models were chosen to visualize the goodness-of-fit of case 1 in Fig. 3.6, and a 1st order, a 6th order, and an 18th models were chosen to to visualize the goodness-of-fit of case 2.A in Fig. 3.12 and of case 2.B in Fig. 3.18. Mainly, the aim in Fig. 3.6, Fig. 3.12, and Fig. 3.18 is to inspect if the regression surface overlaps or not with the data points. If the surface overlaps, then the standard error of the regression is low. Similarly, $\bar{R}^2$ is high when the regression surface overlaps with the data points.

Alternatively, one can analyze the residuals, and the errors in T_w^h and T_a^c for the regression models by inspecting Figs. 3.7, 3.8, 3.9, 3.13, 3.14, and Fig. 3.15. However, all the models discussed in this section so far were based on the regression of the complete dataset, which is not a good indicator of the models' predictive accuracy. Furthermore, to obtain a better assessment of the models' predictive ability, cross-validation of the datasets was carried out. Specifically, the multiple holdout method was used in this work.

Case 1				
Model order	Avg. s, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.01797396	0.0190537	0.99938061	0.99944294
6th	$8.062e-5$	$9.369e-5$	0.99999987	0.99999979
Case 2.A				
Model order	Avg. s, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.0257428	0.0503602	0.99172604	0.98918441
6th	$4.705e-5$	$8.981e-5$	0.99999994	0.99999994
Case 2.B				
Model order	Avg. s, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.05227211	0.10468139	0.98457848	0.97338327
6th	0.00070054	0.00151362	0.99999751	0.99999512

Table 4.2: Summary of the validation results of cases 1 and 2.

To reemphasize what was mentioned in Chapter 3, the holdout method normally involves a single run, but it can be repeated multiple times to improve its strength as a validation method. Furthermore, Table 4.2 summarizes the relevant validation results of cases 1 and 2 that was presented in Section 3.3 and confirms that a 6th order polynomial in the design matrix of the model gives the best measures of goodness-of-fit for both case 1 and case 2. Also, Table 4.2 shows that T_w^h has a lower standard error of the regression compared to T_a^c. Moreover, comparing the results of case 2.A with those of case 2.B , it can be seen that the standard error of the regression is higher for case 2.B, which means that increasing the dataset range to include lower values of $\dot{m}_w$ and $\dot{m}_a$ have a negative impact on the linear regression. Finally, based on the results of Table 4.2 and the previous figures, a 1st or a 2nd order regression model may be sufficient as a correction expression to the ideal heat exchanger model.

4.2.1.2 Datasets of case 3

In the datasets of case 3, both the influent temperatures and the mass flow rates were varied to further analyze the linear regression of the counter-current heat exchanger model.

Table 4.3 summarizes the relevant validation results of case 3, which was presented in Section 3.3. First, comparing the results of case 3 with those of cases 1 and 2, it can be observed that the standard error of the regression is higher for case 3. Next, comparing the results of case 3.C with case 3.B, it can be observed that including the mass flow rates as regressors in case 3.B decreases the standard error of the regression. Then, comparing

Case 3.A				
Model order	Avg. s, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.0775944	0.20320863	0.98797304	0.94702488
6th	0.07866278	0.20599263	0.98862523	0.94991916
Case 3.B				
Model order	Avg. s, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.05293457	0.10581149	0.98482044	0.96653181
6th	0.04148825	0.06945538	0.99105622	0.98617308
Case 3.C				
Model order	Avg. s, °C		Avg. $\bar{R}^2$, –	
	T_w^h	T_a^c	T_w^h	T_a^c
1st	0.05941832	0.12447671	0.98087158	0.9536708
6th	0.05634448	0.11860076	0.98288539	0.95815315

Table 4.3: Summary of the validation results of case 3.

the standard error of the regression of T_w^h to that of T_a^c, it can be observed that T_w^h has a lower standard error of the regression compared to T_a^c, which is consistent with the results of Table 4.2. Similarly, based on the results of Table 4.3 and the figures of case 3 in Section 3.3, one can observe that a 6th order model fits the generated datasets best. Finally, based on the regression results of cases 1, 2, and 3, the selection of the dataset range and the regressors is clearly essential in linear regression.

4.2.2 Results and discussion of the nonlinear regression of the counter-current heat exchanger model

In this work, due to time constraints and the nature of machine learning, nonlinear regression of the counter-current heat exchanger model was not investigated thoroughly as linear regression. However, important results were obtained for a two-layer FNN.

First, for numeric reasons, as mentioned in Lie (2019b), the datasets were normalized as in Fig. 3.32. Next, validation tests were carried out for the datasets of case 3, and the results of the validation were summarized in Fig. 3.31, where RMSE was chosen as the measure of goodness-of-fit. Also, Fig. 3.31 shows that the best-fit dimension **(out)** of the two-layer FNN in Fig. 3.30 is somewhere between 70 and 80. In addition, the figure shows slightly higher RMSE values to those reported in linear regression for the datasets of case 3.[1] However, in this work, the two-layer FNN models were only trained for 10000

[1]The results (RMSE values) of the validation tests for the nonlinear regression of the counter-current heat exchanger model are available in Appendix B.4.

Hybrid vs. numeric solution of the non-ideal heat exchanger model		
Model	Median time (ms)	Mean time (ms)
Hybrid solution (Linear regression)	0.0264	0.035998
Hybrid solution (Nonlinear regression)	0.00123	0.001502
Numeric solution	19.316	21.540
Hybrid vs. numeric solution of the thermal model of an air-cooled synchronous generator with the non-ideal heat exchanger model		
Model	Median time (s)	Mean time (s)
Hybrid solution (Linear regression)	0.016633	0.019648
Hybrid solution (Nonlinear regression)	0.002216	0.002676
Numeric solution	9.673	9.673

Table 4.4: Summary of the benchmark results of case 3.B.

epochs during the validation test, which may explain this finding, but further analysis is required.

4.3 Execution Speed of the Data-Driven Models and the Numeric Non-Ideal Heat Exchanger Model

The comparison of the execution speeds of the hybrid and the numeric non-ideal heat exchanger models was carried out in Appendix B.5 and presented in Section 3.5.

First, the models which are based on the dataset of case **3.B** were used for comparison. However, based on the results of Table 3.8 and Table 3.9, which are summarized in Table 4.4, a new dataset (case **4**) was created to fix the inconsistency between the execution speeds of the hybrid solutions and to improve goodness-of-fit.

Moreover, the source of the inconsistency was the implementation of the linear regression model in Appendix B.5. Specifically, the use of the function **phi_m** to generate the polynomial of the design matrix, which was verified by using a low order model that does not require the use of the function. Also, the new dataset achieved a better **RMSE** for the nonlinear regression model, which can be verified by examining Figs. 3.33 and 3.34. Finally, Table 4.5 summarizes the execution speed of the new dataset and shows a faster execution speed for the data-driven models compared to the numeric solutions.

Hybrid vs. numeric solution of the non-ideal heat exchanger model		
Model	Median time (ms)	Mean time (ms)
Hybrid solution (Linear regression)	0.0008535	0.001466
Hybrid solution (Nonlinear regression)	0.00125	0.001469
Numeric solution	19.316	21.540
Hybrid vs. numeric solution of the thermal model of an air-cooled synchronous generator with the non-ideal heat exchanger model		
Model	Median time (s)	Mean time (s)
Hybrid solution (Linear regression)	0.002052	0.002497
Hybrid solution (Nonlinear regression)	0.002312	0.002745
Numeric solution	9.673	9.673

Table 4.5: Summary of the benchmark results of case 4.

4.4 Predictions of the Mechanistic Model and Åbjøra Experimental Data

In Section 3.6, the experimental data of Åbjøra was compared with the predictions of the mechanistic model. Also, it was observed in Fig. 3.35 that even when extending the model from Lie (2018a) with a more realistic heat exchanger model with temperature dependence in the specific heat capacities of air and water, it was not possible to get a good fit to the available experimental data. Moreover, due to time constraints, an empirical model was not fitted to the experimental data of Åbjøra, which opens the possibility for future work.

5 Conclusion

In this thesis, an overview of the thermal model of an air-cooled synchronous generator that was proposed in Øyvang (2018), and studied in Lie (2018a) was given, and the possible extension of the heat exchanger sub-model with the case of temperature dependence in the specific heat capacities of air and water was discussed. Furthermore, to speed up the solution time of the non-ideal heat exchanger sub-model, explicit data-driven models were developed using linear and nonlinear regression for a variety of conditions and expressed as a correction expression to the ideal heat exchanger model. Moreover, the execution speed of the numeric solution of the nonlinear two-point boundary value problem was compared with that of the explicit data-driven models, and the results showed a faster execution speed for the data-driven models. In addition, the experimental data of Åbjøra was compared with the predictions of the mechanistic model, and it was observed that even when extending the model from Lie (2018a) with a more realistic heat exchanger model with temperature dependence in the specific heat capacities of air and water, it was not possible to get a good fit to the available experimental data.

6 Future Work

A hybrid mechanistic-empirical model is a hot research topic that applies to many disciplines. However, to be specific to the topics presented in this work:

1. Reduction of the simulation time of the state estimation reported in M. Pandey (2019a) of model 3b and model 4b.
2. Fitting an empirical model to the experimental data of Åbjøra.

References

Allen, Mike (Apr. 2017). "The SAGE Encyclopedia of Communication Research Methods." In: Number Of Volumes: 4 Place: Thousand Oaks, California. DOI: 10.4135/9781483381411. URL: https://methods.sagepub.com/reference/the-sage-encyclopedia-of-communication-research-methods.

Bergman, Theodore L. et al. (2011). *Fundamentals of Heat and Mass Transfer*. Wiley. ISBN: 978-0-470-50197-9. URL: https://books.google.no/books?id=vvyIoXEywMoC.

Chatterjee, S. and J.S. Simonoff (2013). *Handbook of Regression Analysis*. Wiley Handbooks in Applied Statistics. Wiley. ISBN: 9781118532836. URL: https://books.google.no/books?id=X95obhB6RQcC.

DataVedas (Mar. 2018). *REGULARIZED REGRESSION ALGORITHMS | Data Vedas*. en-US. https://www.datavedas.com/regularized-regression-algorithms/. [Online; accessed 11-April-2020].

Grace-Martin, Karen (Dec. 2008). *Assessing the Fit of Regression Models*. en-US. [Online; accessed 13-April-2020]. URL: https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/.

Gujarati, Damodar N. (2019). *Linear Regression: A Mathematical Introduction*. [Online; accessed 15-April-2020]. 2455 Teller Road, Thousand Oaks California 91320: SAGE Publications, Inc. ISBN: 978-1-5443-3657-2 978-1-07-180257-1. DOI: 10.4135/9781071802571. URL: https://methods.sagepub.com/book/linear-regression.

Helwig, Nathaniel (Jan. 2017). *Multivariate Linear Regression*. en.

Innes, Michael et al. (2018). "Fashionable Modelling with Flux." In: *CoRR* abs/1811.01457. arXiv: 1811.01457. URL: http://arxiv.org/abs/1811.01457.

Kiernan, Diane (Apr. 2018). *7.3: Population Model*. en. [Online; accessed 13-April-2020]. URL: https://stats.libretexts.org/Bookshelves/Applied_Statistics/Book%3A_Natural_Resources_Biometrics_(Kiernan)/07%3A_Correlation_and_Simple_Linear_Regression/7.03%3A_Population_Model.

Lakshmanan, Swetha (May 17, 2019). *How, When and Why Should You Normalize/ Standardize/ Rescale Your Data?* Medium. https://medium.com/@swethalakshmanan14/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff. [Online; accessed 09-May-2020].

Lie, Bernt (Nov. 2018a). *Project, FM1015 Modelling of Dynamic Systems*. Tech. rep. University of South-Eastern Norway.

– (Nov. 2018b). *Solution, Project, FM1015 Modelling of Dynamic Systems*. Tech. rep. University of South-Eastern Norway.

Lie, Bernt (Aug. 2019a). *Modeling of Dynamic Systems*. University of South-Eastern Norway.
– (Jan. 2019b). "Surrogate and Hybrid Models for Control." In: pp. 1–8.
Marquardt, Donald W. (1980). "A Critique of Some Ridge Regression Methods: Comment." In: *Journal of the American Statistical Association* 75.369, pp. 87–91. ISSN: 01621459. URL: www.jstor.org/stable/2287388.
Mcbride, Bonnie J. Gordon (Oct. 1993). *Coefficients for calculating thermodynamic and transport properties of individual species*. Tech. rep. URL: https://ntrs.nasa.gov/search.jsp?R=19940013151 (visited on 02/15/2020).
McBride, Bonnie J. Zehe (Sept. 2002). *NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species*. Tech. rep. URL: https://ntrs.nasa.gov/search.jsp?R=20020085330 (visited on 02/15/2020).
Murphy, Kathleen E. (Mar. 2020). *Thermodynamics Problem Solving in Physical Chemistry: Study Guide and Map*. en. Google-Books-ID: fabYDwAAQBAJ. CRC Press. ISBN: 978-1-00-003028-0.
Nau, Robert (2014). *Review of basic statistics and the simplest forecasting model: the sample mean*.
NobleProg, ed. (June 2014). *Standard Error of the Estimate - Training Material*. https://training-course-material.com/training/Standard_Error_of_the_Estimate. [Online; accessed 13-April-2020].
Øyvang, Thomas (2018). "Enhanced Power Capability of Generator Units for Increased Operational Security." PhD thesis. University of South-Eastern Norway, Faculty of Technology, Natural Sciences and Maritime Sciences.
Pandey, Madhusudhan (2019a). "Model Fitting and State Estimation for Thermal Model of Synchronous Generator." MA thesis. University of South-Eastern Norway, Faculty of Technology, Natural Sciences and Maritime Sciences.
– (2019b). "Unpublished results from summer job."
Pandey, Shanta and Charlotte Bright (June 2008). "What Are Degrees of Freedom?" In: *Social Work Research* 32. DOI: 10.1093/swr/32.2.119.
Poling, Bruce E., John M. Prausnitz, and John P. O'Connell (2001). *Properties of Gases and Liquids, Fifth Edition*. eng. New York: McGraw-Hill Education. URL: https://www.accessengineeringlibrary.com/content/book/9780070116825.
Poore, Geoffrey (May 7, 2020). *gpoore/minted*. [Online; accessed 8-May-2020]. URL: https://github.com/gpoore/minted.
Rackauckas, C and Q Nie (2017). *DifferentialEquations.jl - A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia*. DOI: 10.5334/jors.151.
Wikipedia contributors (Nov. 2019a). *Explained sum of squares — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Explained_sum_of_squares&oldid=925963298. [Online; accessed 13-April-2020].
– (Dec. 2019b). *General linear model — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=General_linear_model&oldid=932027631. [Online; accessed 13-April-2020].

– (Aug. 2019c). *Regression validation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Regression_validation&oldid=913207226. [Online; accessed 13-April-2020].
– (Mar. 2020a). *Coefficient of determination — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Coefficient_of_determination&oldid=946053744. [Online; accessed 13-April-2020].
– (Apr. 2020b). *Cross-validation (statistics) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)&oldid=948965871. [Online; accessed 13-April-2020].
– (Jan. 2020c). *Design matrix — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Design_matrix&oldid=937659051. [Online; accessed 12-April-2020].
– (Apr. 2020d). *Errors and residuals — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Errors_and_residuals&oldid=949672133. [Online; accessed 13-April-2020].
– (Apr. 2020e). *Linear regression — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Linear_regression&oldid=948596126. [Online; accessed 12-April-2020].
– (Apr. 2020f). *Ordinary least squares — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Ordinary_least_squares&oldid=948717245. [Online; accessed 13-April-2020].
– (Mar. 2020g). *Regression analysis — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Regression_analysis&oldid=944367056. [Online; accessed 12-April-2020].
– (Feb. 2020h). *Root-mean-square deviation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Root-mean-square_deviation&oldid=941256353. [Online; accessed 13-April-2020].

Appendix A

Master’s Thesis Task Description

University of South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Hybrid Machine Learning and Mechanistic Thermal Model of Synchronous Generator

USN supervisor: Bernt Lie, professor, co-supervisors Madhusudhan Pandey, PhD student, and Thomas Øyvang, associate professor

External partner: Skagerak Kraft (Ingunn Granstrøm)

Task background:
A thermal model of a synchronous generator was proposed in Øyvang (2018), and studied in a group project in course FM1015 Modelling of Dynamic Systems, Lie (2018). The model from Lie (2018) was further studied in a MSc thesis in 2019 (Pandey *et al*., 2019), and in a subsequent summer job (Pandey, 2019). The purpose of such a model is to allow for monitoring/control of the generator temperature, and thereby open up for relaxed constraints on the power factor in the operation of generators.

A possible extension of the model in Lie (2018), is the case of temperature dependence in heat capacity and/or heat transfer of air/water in a counter-current heat exchanger. A steady state model of a counter-current heat exchanger in general leads to a two-point boundary value problem, which can be solved numerically; numerical solution is relatively costly for on-line use. With temperature independence in heat capacity/heat transfer, it is possible to find an efficient explicit/analytic expression as in Lie (2018). When we cannot find an analytic expression, it is still possible to solve the heat exchanger problem numerically for each time step in the remaining model, but this leads to very long simulation time. To speed up the solution time, it is of interest to consider the following strategy:

- The heat exchanger model is solved many times off-line for a variety of conditions (inlet temperature of water and air, flow rates, thermal dependencies, etc.), and the results are stored in a data matrix.
- A regression model is fitted to the data leading to an explicit expression relating influent temperature of water/air to effluent temperature of water/air.
- When solving the dynamic thermal model of the generator, for each time step: instead of solving the heat exchanger model numerically as a two-point boundary value problem – which is slow, we solve it by the regression model – which is fast.
- Experience with similar problems indicate that this strategy should give several decades faster solution time.

Different types of regression models can be considered, e.g., (i) linear regression methods as in Chemometrics, and (ii) nonlinear regression methods such as in Neural Networks.

Even when extending the model from Lie (2018) with a more realistic heat exchanger model, it will not be possible to get perfect fit of the mechanistic model to available experimental data as used by Øyvang (2018) and Pandey (Pandey *et al*., 2019, Pandey, 2019). To get improved model fit, one possibility is to model the system as good as possible with a mechanistic model, and then add a dynamic regression model to describe the difference between the mechanistic model and the experimental data, see, e.g., Lie (2019). Related

ideas are available in package DiffEqFlux.jl as part of the DifferentialEquations.jl package for computer language Julia.

In summary: combination of mechanistic and data-driven/empirical/machine learning models is a hot research topic. The promise is to get good model fit, with efficient model simulation. Furthermore, a mechanistic model can be built and studied without available experimental data, while a machine learning model/improved fit can only be built after experimental data become available. The results from this thesis has very wide applicability in every field of science.

The modern computer science language *Julia* has class leading differential equations solvers, optimization code, plotting facilities, and easy-to-use machine learning tools. Julia takes the best from MATLAB, Python, and R, and is free. In this project, the work should be carried out using Julia.

References:

Lie, Bernt (2018). *Group project task, course FM1015 Modelling of Dynamic Systems*. University of South-Eastern Norway.

Lie, Bernt (2019). "Surrogate and Hybrid Models for Control". *Proceedings of SIMS 2019*, August 2019. To be published in Linköping University Electronic Press.

Øyvang, Thomas (2018). *Enhanced power capability of generator units for increased operational security*. Ph.D.-thesis, University of South-Eastern Norway.

Pandey, Madhusudhan, Øyvang, Thomas, and Lie, Bernt (2019). "State Estimation of a Thermal Model of Air-cooled Synchronous Generator". *Proceedings of SIMS 2019*, August 2019. To be published in Linköping University Electronic Press.

Pandey, Madhusudhan (2019). Unpublished results from summer job, University of South-Eastern Norway.

Task description:

The following tasks are relevant:

1. Give an overview of a thermal model of a synchronous generator, with ideal and non-ideal heat exchanger model. Discuss how the model can be solved with a non-ideal heat exchanger sub-model by using a two-stage strategy (two-point boundary value problem to be solved for each time step in the dynamic model).
2. Develop an explicit data-driven model for the non-ideal heat exchanger model. This should be expressed as a correction expression to the ideal heat exchanger model. The correction factor will be a function of inlet temperatures, flow rates, temperature dependence of heat capacities, etc.
3. Compare the execution speed of the two-stage strategy (point 1 above), with that of the use of the data-driven model.
4. Use available experimental data, and compare the predictions from the mechanistic model with those of the experiments. Next, discuss principles of fitting an empirical/data-driven/machine learning model to describe the error in the model. In particular, discuss the needs of data to fit an empirical model.
5. If there is time, try to fit an empirical model to describe the mechanistic model deviation, and discuss the results.
6. Report the work in the Master's Thesis, and possibly in a suitable conference/journal paper, e.g., SIMS 2020.

Student category: EPE, IIA, PT, EET (EPE students will have more appreciation of the actual model of the generator, but the ideas are valid and useful within all scientific fields).

Practical arrangements:
A working place for the candidate will be offered at University of South-Eastern Norway, Campus Porsgrunn; candidates can choose to sit elsewhere. Some information about generator data, etc. will be provided by Skagerak Kraft, e.g., in relation with the Grunnåi power plant.

Supervision:
Weekly supervision meetings are offered on Mondays (up to 1 hour per week) until Week 15, April 6, 2020; for net based students and Industry Master students, meetings can take place via Skype/Teams.

The candidate is expected to hand in partial reports every three weeks, and will receive feedback on these. The last month (after Easter), the candidate is expected to work independently. If the candidate chooses to write a paper based on the work, supervision on the paper writing will be given until early June 2020. Observe: it will not be possible to get feedback on the thesis itself from the principal supervisor after Week 15. Because of this, use the opportunity to get feedback on partial reports prior to Week 15.

Signatures:

Supervisor (date and signature):

Student (write clearly in all capitalized letters):

Student (date and signature):

Appendix B

Code listing

In this work, all the code is implemented in Julia using Jupyter graphical notebook and is available in the following GitHub repository:

https://github.com/KhaledAleikish/MasterThesis2020/tree/master/Jupyter_Notebooks

B.1 Counter-current Heat Exchanger Models

Jupyter notebook 1

B.2 Thermal Model of an Air-Cooled Synchronous Generator, With Ideal and Non-Ideal Heat Exchanger Model

Jupyter notebook 2

B.3 Linear Regression of the Counter-Current Heat Exchanger Model

Jupyter notebook 3

B.4 Nonlinear Regression of the Counter-Current Heat Exchanger Model

Jupyter notebook 4

B.5 Comparison of the Execution Speed of the Hybrid Non-Ideal Heat Exchanger Model and the Numeric Solver

Jupyter notebook 5

B.6 Comparison of the Predictions From the Mechanistic Model With Those of the Experiments

Jupyter notebook 6