University of
South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2021
Electrical Power Engineering

Hybrid Machine Learning and Mechanistic Thermal Model of Synchronous Generator

Prakash Dhakal

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

Number of pages: 60

Keywords: Machine learning, Thermal model, Linear regression, Nonlinear regression.

Student:	Prakash Dhakal
Supervisor:	Bernt Lie, Co-supervisor:Madhusudhan Pandey, Thomas Øyvang
External partner:	Skagerak Kraft, Ingunn Granstrøm

University of South-Eastern Norway

www.usn.no

Summary:

Overheating of synchronous generators may result in shortened generator lifespan, thus strict constraints are imposed on their operation. A dynamic model of the generator temperature may leave good monitoring of the generator condition, and also more flexible operation. Within the past, the mix of a thermal model of an air-cooled generator with better control has been considered to assist ride-through problems: By using a model-based online monitoring , the temperature development in certain locations within the synchronous generator were kept in restraint. Additionally, exploiting the generator’s full thermal capacity led to improved performance [1]. Further work has considered various improved thermal generator models, together with model fitting and state estimation [2]. Now also, the studies thus far have used normal, counter-current heat exchanger models with constant Stanton numbers, which authorize for an analytic, explicit heat exchanger models description. In [2], a heat exchanger model with temperature-dependent heat capacities was considered. The result's a two-point boundary value problem that's several thousand-fold slower to resolve than with a relentless Stanton number. To hurry up the solution, a nonlinear regression model was trained off-line to suit the solution of the boundary value problem. However, what is missing in [3] is that the possibility to think about heat exchangers with varying heat transfer coefficients. The thermal model of an air-cooled synchronous generator created in [4] and enlarged in [3] with a more realistic temperature-dependent condition was continued during this thesis, with variable heat transfer coefficients, to lower the time it takes a heat exchanger model to resolve a temperature-dependent problem. A hybrid model was created with estimated parameters from a data-driven model for a spread of scenarios, and also the performance was compared to the numeric solution, which was around 220 times quicker.

Preface

This thesis was completed in January 2020 as part of the FMH606 course at the University of Southeastern Norway (USN) in the Department of Electrical Engineering, IT, and Cybernetics, under the supervision of Professor Bernt Lie and co-supervisor Madhusudhan Pandey.

I would want to express my heartfelt thanks to my supervisor and co-supervisor for assisting me in overcoming obstacles by providing helpful recommendations, direction, and comments throughout the project. I would want to express my gratitude to my family for their unwavering support and affection.

The possibilities of a hybrid mechanistic-empirical thermal model of an air-cooled synchronous generator is investigated in this thesis. The goal is to use linear and nonlinear regression to shorten the simulation duration of the heat exchanger sub-nonlinear model's two-point boundary value issue.

Porsgrunn, May 19, 2021

Prakash Dhakal

Content

Nomenclature

Symbol	**Explanation**
R^2	The Coefficient of Determination
RMSE	Root-Mean Square Error
T_i^j(j=h,c,N,A. i=a,w,t,s)	The temperature of species j obtained by:
	A = Analytic solution
	N = Numeric solution
	h = Hot
	c = Cold
	a = Air
	w = water
	t = tube
	s = shell

List of figures

List of tables

1 Introduction

1.1 Background

The power factor of the system plays a key role in the production of electricity. According to [4], the power factor in European hydropower generation is limited to [0.85,0.95] and for the Norwegian hydropower system, it is below 0.86. High power factor results in to increase in current and therefore, synchronous generators are overheated. To protect these Hydro generators maturing of temperature should be under dominance and thermal capacity should be properly taken into consideration. A closed-loop heat exchanger model has been developed in [1] for cooling heated air generated inside the generator.

1.2 Previous Work

It was suggested in [1], the thermal model of an air-cooled synchronous generator, which was investigated in [4]and[2]. In [3], a heat exchanger model with temperature-dependent heat capacities was considered. The result is a two-point boundary value problem that is several thousand times slower to solve than with a constant Stanton number. To speed up the solution, a nonlinear regression model was trained off-line to fit the solution of the boundary value problem. However, what is missing in [3] is the possibility to consider heat exchangers with varying heat transfer coefficients.

1.3 Scope and outline

In this thesis, an air-cooled synchronous generator's thermal model developed in [4] and expanded in [5] with a more realistic temperature-dependent situation is continued, with different heat transfer coefficients to reduce the time it takes for a heat exchanger to solve a problem that is temperature-dependent. For a range of scenarios, a hybrid model is developed with estimated parameters from a data-driven model, and the speed is compared to numeric solutions.

The model of an air-cooled synchronous generator is briefly discussed in Chapter 2. Following that, the construction of a counter-current model is demonstrated. Finally, Julia is used to comparing the analytic and numerical solutions for the ideal and non-ideal cases.

The linear and nonlinear regression of the counter-current heat exchanger model is implemented in Julia in Chapter 3.

The result and discussion are shown in chapter 5 along with the conclusion and future work is planned in chapter 6.

1.4 Theory

1.4.1 Synchronous Generator

It is an electrical machine that converts mechanical power into AC e power at a specific voltage and frequency. It always runs at a relentless speed called synchronous speed. It has multiple applications in generation, transmission, and distribution and is generally employed in nuclear, thermal, and hydropower systems for generating voltage[6].

1.4.1.1 Losses in synchronous generator

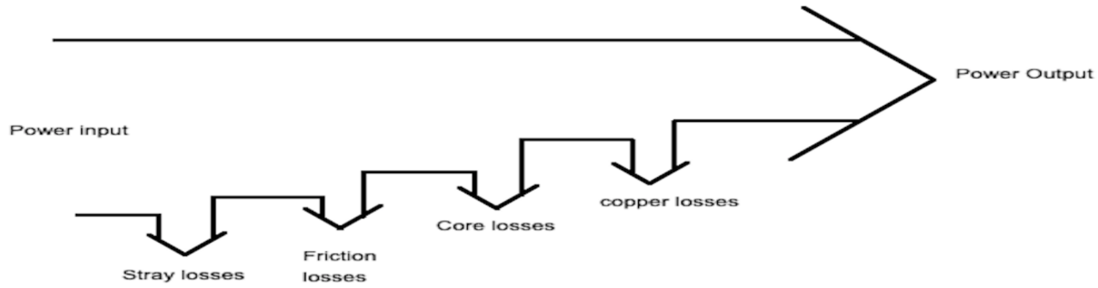

Figure 1-1:losses in Synchronous generator

Friction losses are related to the force it takes to beat drag related to rotating parts. Generally, these losses are because of the friction of bushings ,bearings which are proportional to the rotor speed. Iron losses are also called core losses and are associated losses within the magnetic path and are divided into hysteresis loss and Eddy current loss due to changing polarity of the flux within the core. Ohmic or losses are because of this current flowing into the conductor. They are adequate the resistance of the trial where the current flows multiplied by the square of the current Stray losses are normally termed as losses that do not correlate to the explained losses above. These losses are the main causes for heat generation inside the generator. So, to avoid this heat should be properly dispatched with a good heat exchanger[5].

1.4.1.2 Heat exchanger

They are the system want to transfer heat between the fluids. They are both using heating and cooling processes. The fluids are separated by a solid wall which preventing mixing. They are mostly used in Power stations, chemical plants, air conditioning, petroleum refineries, etc. Among such a lot of different types in our thesis, we are coping with shell and tube heat exchangers.

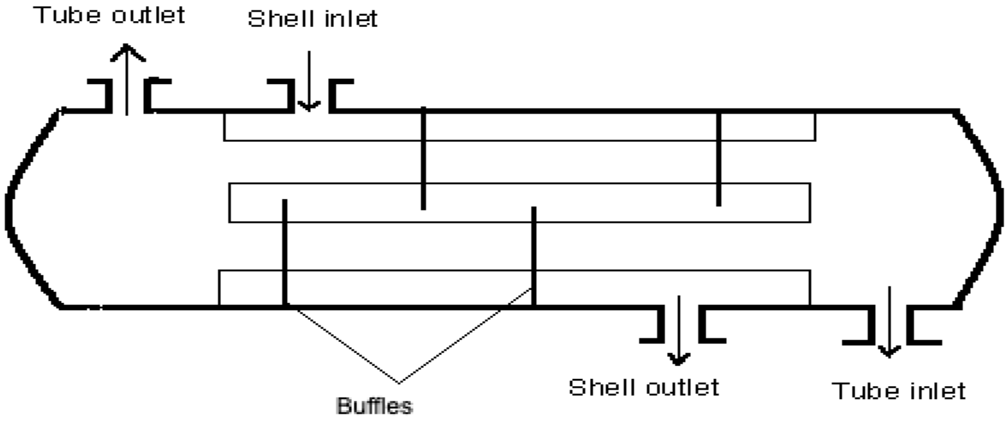

Figure 1-2:Shell and tube heat exchanger

It consists of a shell with several tubes inside it. One fluid runs inside the tubes and another fluid flows through the shell to transfer heat between two fluids. The gathering of tubes is named a tube bundle and may have several tubes. Baffles are used for steering the flow through the shell side; therefore, the fluid does not take a shortcut path leaving the system[7].

2 Thermal Model of an Air -cooled Synchronous Generator with ideal and Non-ideal Heat Exchanger Model

A thermal model for a fully enclosed air-cooled synchronous generator has been created by [1]. [4] used principles and notations from [1] to create a similar model with a more general design and a more powerful heat exchanger. The thermal model of the air-cooled synchronous generator is shown in Figure 2-1 in action. A fan blows cold air from the heat exchanger into the rotor/stator air gap. Heat flow from the rotor, air gap windage, and bearing friction heat the air. Air is then pushed into the iron cores and is then heated by the heat flow from the iron cores. The heated air is now stored and moved through the heat exchanger at the stator's outlet. The heated air is then cooled to the desired temperature in the heat exchanger using continuous cold-water circulation before being fed back into the air gap in a continuous operation.

At temperature $T_{\mathrm{w}}^{\mathrm{c}}$, the heat exchanger is fed with cold water at a mass flow rate of $\dot{m}_{\mathrm{w}}$. At the stator outlet and heat exchanger entry, the air mass flow rate is $\dot{m}_{a}$, and the temperature is $T_{\mathrm{a}}^{\mathrm{h}}$. Due to rotor field current, I_{f}.rotor copper heat source $\dot{Q}_{\mathrm{r}}^{\sigma}$ is created and similarly stator copper heat source, $\dot{Q}_{\mathrm{s}}^{\sigma}$due to stator current I_{t}.Heat generated due to friction in stator/rotor air gap is $\dot{Q}_{\mathrm{F}}^{\sigma}$ and $\dot{Q}_{\mathrm{Fe}}^{\sigma}$, is stator iron heat source. $\dot{m}_{a}$, $\dot{m}_{\mathrm{w}}$,T_{w}^{c}, $\dot{Q}_{\mathrm{Fe}}^{\sigma}$, $\dot{Q}_{\mathrm{F}}^{\sigma}$, I_{Fe}, I_{F} highly influences the thermal operation of air-cooled synchronous generator . T_{r}, T_{s}, and T_{Fe}, are rotor, stator and iron core temperatures respectively.

Figure 2-2 depicts the functional diagram of an air-cooled synchronous generator, which shows the inputs and outputs

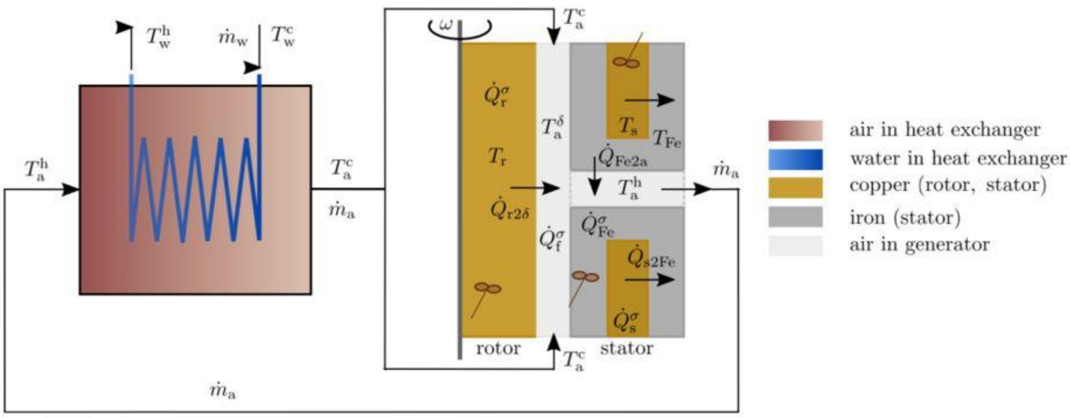

Figure 2-1: Thermal operation of an air-cooled synchronous generator [2].

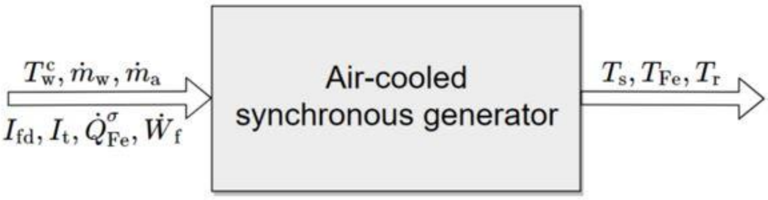

Figure 2-2:Functional diagram of an air-cooled synchronous generator

2.1 Overview of Counter-Current Heat Exchanger Model

Different heat exchanger model is there. Among them, shell and tube heat exchangers are described in [4] which involves the flow of water inside the tube and flow of air in the shell. Depending on the type of flow three models are developed i.e., cross-current, co-current, and counter-current heat exchangers. But in our work, we mainly focus on the counter-current heat exchanger model.

The total mass balance can be written as:

$$\frac{dm}{dt} = \dot{m}_i - \dot{m}_e = 0 \qquad \text{2-1}$$

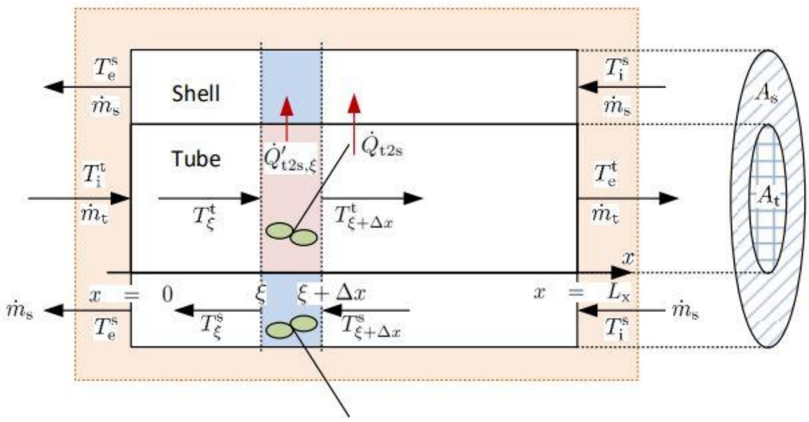

Figure 2-3:distributed model of a counter-current heat exchanger [6].

For steady state mass balance for tube side is

$$\dot{m}_i^t = \dot{m}_e^t \quad 2\text{-}2$$

Similarly, the mass balance for the shell side is.

$$\dot{m}_i^s = \dot{m}_e^s \quad 2\text{-}3$$

When friction and volume work is neglected, the energy balance becomes.

$$\frac{dU}{dt} = \dot{H}_\iota - \dot{H}_e + Q \quad 2\text{-}4$$

The perfect mixing in the volume defined by $x \in [\,\xi, \xi + \triangle x\,]$.The flow rates for the tube side can be expressed as[2]:

$$\dot{H}_\iota^t = \dot{m}_t \widehat{H}_\xi^t \quad 2\text{-}5$$

$$\dot{H}_e^t = \dot{m}_t \widehat{H}_{\xi+\Delta x}^t \quad 2\text{-}6$$

Heat flow is denoted as:

$$\dot{Q} = -\dot{Q}_\xi^{t2s} \quad 2\text{-}7$$

For an interval, heat flow is expressed as:

$$\dot{Q}_{\xi+\Delta x}^{t2s} = \dot{Q}'_{t2s,\xi+\Delta x}\Delta x \quad 2\text{-}8$$

And the internal energy as

$$U_{\xi+\Delta x}^t = \triangle\, x \rho_t A_t \widehat{U}_{\xi+\Delta x}^t \quad 2\text{-}9$$

Which leads to:

$$\frac{d}{dx}(\triangle x \rho_t A_t \widehat{U}_{\xi+\Delta x}^t = \dot{m}_t \widehat{H}_\xi^t - \dot{m}_t \widehat{H}_{\xi+\Delta x}^t - \dot{Q}'_{t2s,\xi+\Delta x}\Delta x \quad 2\text{-}10$$

By letting $\triangle x \rightarrow 0$ and generalizing ξ, the previous equations changes to:

$$\rho_t A_t \frac{\partial \widehat{U}_t}{\partial t} = -\dot{m}_t \frac{\partial \widehat{H}_t}{\partial t} - \dot{Q}'_{t2s} \quad 2\text{-}11$$

So, here

$$\dot{Q}'_{t2s} = U\wp(T_t - T_s) \qquad \text{2-12}$$

With $\partial\hat{U} \approx \hat{c}_v\, \partial T$ and $\partial\hat{H} \approx \hat{c}_p\, \partial T$, which results in.

$$\rho_t A_t \hat{c}_{v,t} \frac{\partial T_t}{\partial t} = \dot{m}_t \hat{c}_{p,t} \frac{\partial T_t}{\partial t} - U\wp(T_t - T_s) \qquad \text{2-13}$$

Similarly, for shell side

$$\rho_s A_s \hat{c}_{v,s} \frac{\partial T_s}{\partial t} = -\dot{m}_t \hat{c}_{p,s} \frac{\partial T_s}{\partial t} - U\wp(T_t - T_s) \qquad \text{2-14}$$

for steady state $\frac{\partial T}{\partial t} = 0$ so PDEs reduces to ordinary differential equations i.e.

$$\frac{dT_t}{dx} = -\frac{U\wp}{\dot{m}_t \hat{c}_{p,t}}(T_t - T_s) \qquad \text{2-15}$$

$$\frac{dT_s}{dx} = -\frac{U\wp}{\dot{m}_s \hat{c}_{p,s}}(T_t - T_s) \qquad \text{2-16}$$

Linear, space invariant boundary value problem is formed if $\frac{U\wp}{\dot{m}\hat{c}_p}$ is constant with respect to x

$$\frac{d}{dx}\begin{pmatrix} T_t \\ T_s \end{pmatrix} = \begin{pmatrix} -\frac{U\wp}{\dot{m}_t \hat{c}_{p,t}} & \frac{U\wp}{\dot{m}_t \hat{c}_{p,t}} \\ -\frac{U\wp}{\dot{m}_s \hat{c}_{p,s}} & \frac{U\wp}{\dot{m}_s \hat{c}_{p,s}} \end{pmatrix} \begin{pmatrix} T_t \\ T_s \end{pmatrix}$$

The solution of this model in [2] is found to be

$$T_w(x) = \frac{1}{\alpha_t - \alpha_s}\left[\alpha_t e^{(\alpha_s - \alpha_t)x} - \alpha_s\right)T_i^t + (\alpha_t - \alpha_t e^{(\alpha_s - \alpha_t)x})T_e^s] \qquad \text{2-17}$$

$$T_a(x) = \frac{1}{\alpha_t - \alpha_s}\left[\alpha_t e^{(\alpha_s - \alpha_t)x} - \alpha_s\right)T_i^t + (\alpha_t - \alpha_s e^{(\alpha_s - \alpha_t)x})T_e^s] \qquad \text{2-18}$$

Here,

$$\alpha_t \triangleq \frac{U\wp}{\dot{m}_t \hat{c}_{p,t}} \qquad \text{2-19}$$

$$\alpha_s \triangleq \frac{U\wp}{\dot{m}_s\hat{c}_{p,s}} \qquad 2\text{-}20$$

The explicit expression for effilent temperatures $T_t\,(x = L_x) = T_i^s$ and $T_s\,(x = 0) = T_i^t$ by utilizing Stanton number are:

$$T_{\mathrm{e}}^{\mathrm{t}} = \frac{N_{\mathrm{St}}^{\mathrm{t}}(1-\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})T_{\mathrm{i}}^{\mathrm{s}} + (N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}})\exp\,(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})T_{\mathrm{i}}^{\mathrm{t}}}{N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}}\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})} \qquad 2\text{-}21$$

$$T_{\mathrm{e}}^{s} = \frac{(N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}})T_{\mathrm{i}}^{\mathrm{s}}+N_{\mathrm{St}}^{\mathrm{t}}(1-\exp\,(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})T_{\mathrm{i}}^{\mathrm{t}}}{N_{\mathrm{St}}^{\mathrm{t}}-N_{\mathrm{St}}^{\mathrm{s}}\exp(N_{\mathrm{St}}^{\mathrm{s}}-N_{\mathrm{St}}^{\mathrm{t}})} \qquad 2\text{-}22$$

This analytic expression applies only when and is independent of x. A numerical solution is required if either of the total heat transfer coefficient U, the perimeter, or the individual heat capacities vary with x. The non-ideal case of temperature-dependent heat capacity, in which and are no longer independent of x, is of interest in this work. This creates a nonlinear two-point boundary value problem with a costly numerical solution. An alternative approach that does not require an iterative process, namely a strategy that blends a data-driven model with a mechanistic model, is of interest.

Finally, in the thermal model of an air-cooled synchronous generator, the air is circulating in the shell and water is flowing within the tube.

2.2 Analytic and Numeric Solution

When the thermal model of an air-cooled synchronous generator from [2] is extended with a more practical heat exchanger model, such as one 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, resulting in a nonlinear two-point boundary value problem. In this section, the analytic solution of the ideal heat exchanger model and the numeric solution of the non-ideal model are compared[7].

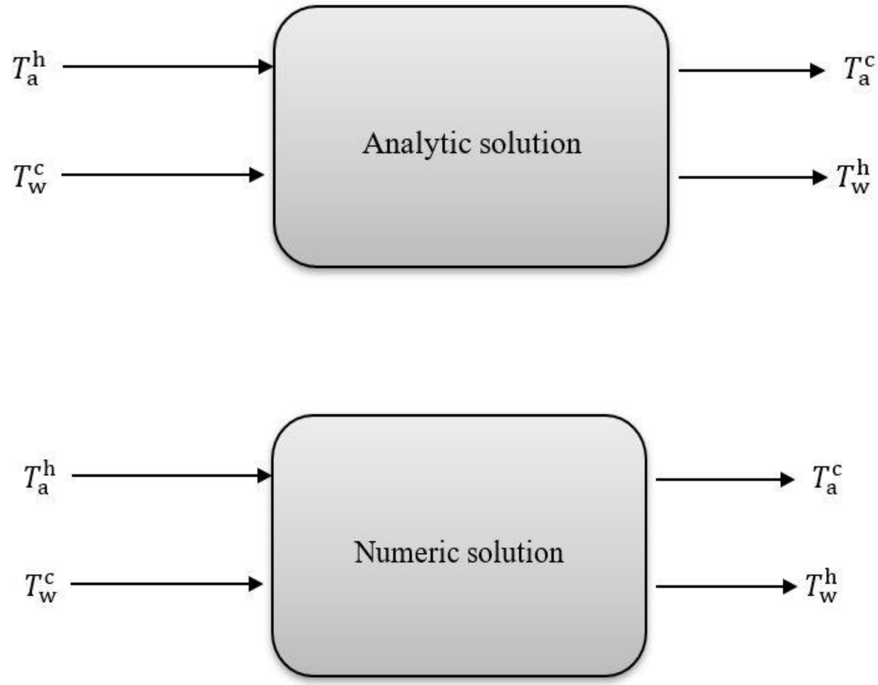

Figure 2-4: Analytic and Numeric Solution

2.2.1 Comparing Analytic and Numeric Solution in Julia for Ideal case

The primary aim of applying the numeric solver for the ideal case of temperature independence in the real heat capacities of air and water is to see if the analytic solution suits the ideal case numeric solution.Table 2-1 shows a comparison of the analytic and numeric solutions of the ideal heat exchanger model.

Model	Median time	Mean time
The analytic solution	22.601 μs μs	23.118 μs
The numeric solution	4.665 ms	5.877 ms

Table 2-1: Benchmark results for the ideal heat exchanger model.

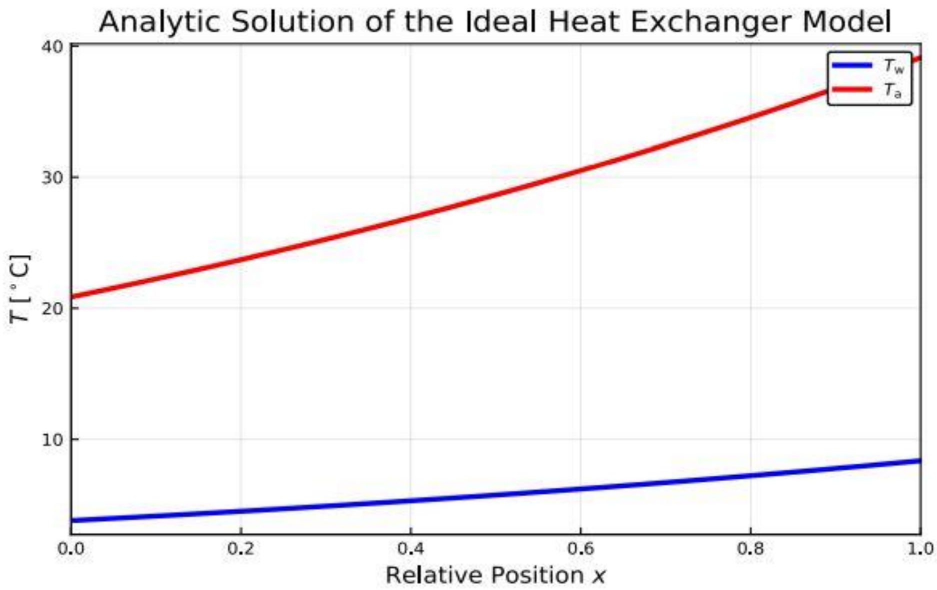

Figure 2-5:Analytic solution of the ideal heat exchanger model.

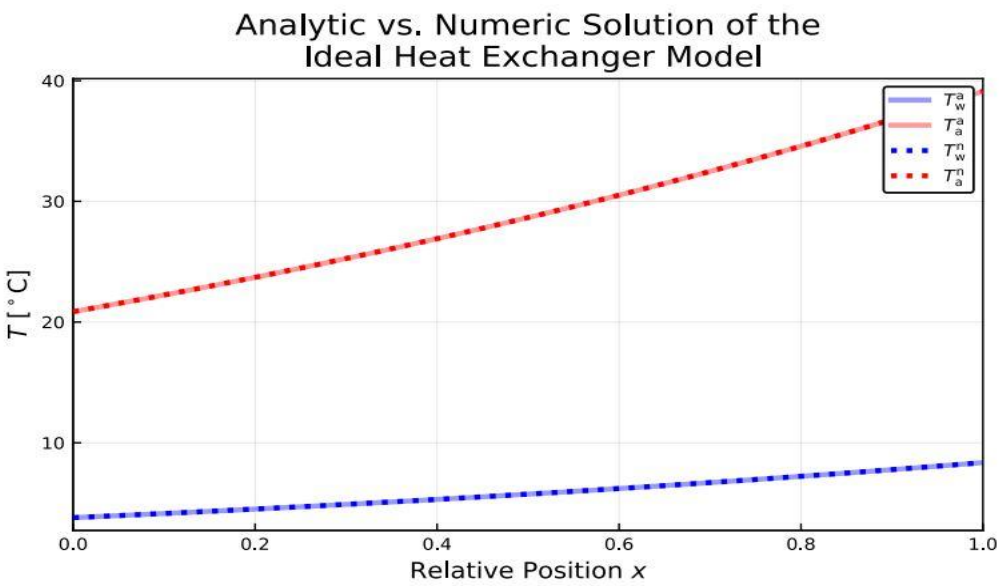

Figure 2-6:Analytic Vs Numeric Solution of Ideal Heat exchanger Model

Since the solutions are identical, it is safe to move on to the next step, which is to find a solution for the non-ideal case of temperature dependence in the basic heat capacities of air and water. Comparing the mean time between the two Solutions it seems Analytic Expressions are evaluated 200 times faster than numeric Solutions.

2.2.2 Comparing Analytic and Numeric Solution in Julia for non-ideal case

Figure 2-8 depicts a contrast of the analytic and numerical solutions while .Comparing the processing times of the two numeric solutions is also interesting. summarizes the benchmark findings, which reveals that the non-ideal case of temperature dependence is nearly eight times slower than the ideal case.

Model	Median time	Mean time
The numeric solution when $\hat{c}_p$	4.665 ms	5.877 ms
The numeric solution when $\hat{c}_p(t)$	39.947 ms	38.786 ms

Table 2-2:Benchmark results for the non-ideal heat exchanger model.

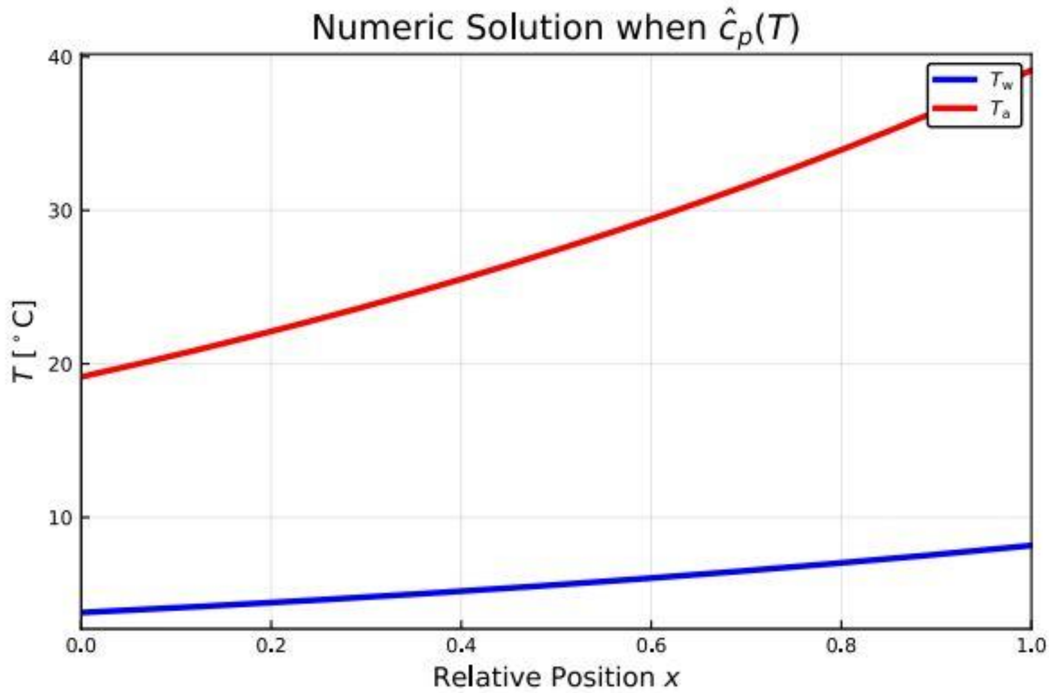

Figure 2-7: Numeric solution of the non-ideal heat exchanger model when $\hat{c}_p$(T).

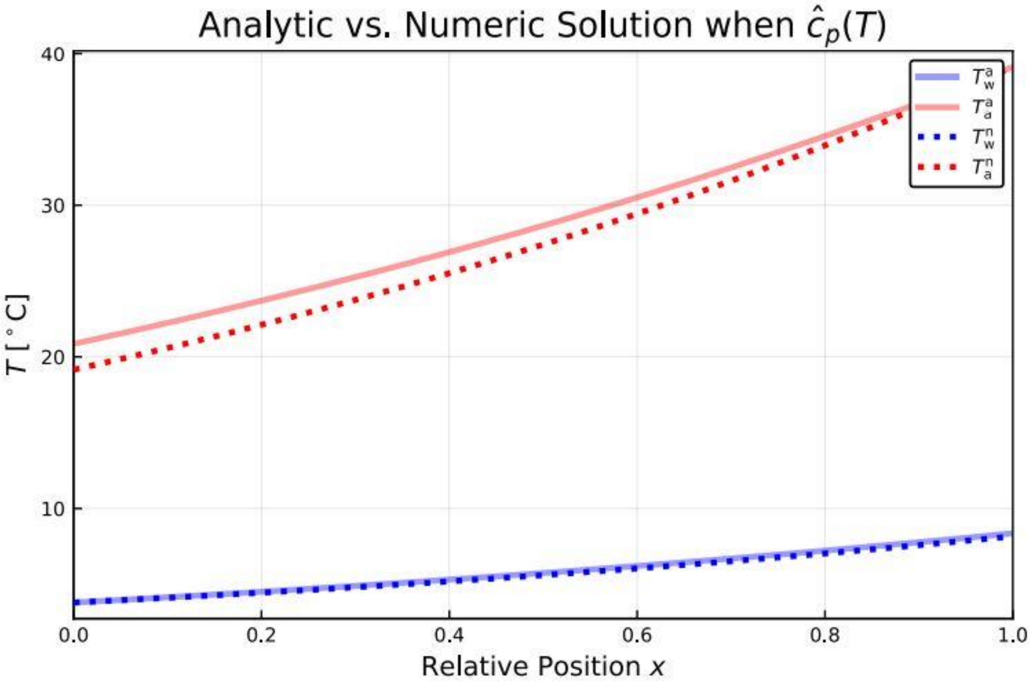

Figure 2-8: Analytic solution vs. numeric solution of non-ideal heat exchanger model

3 Heat Exchanger Regression Model

3.1 Overview or regression analysis

The theory of how a response variable is influenced by one or more predictors is known as regression analysis. Typically, dependency is believed to be by the mean, and the mean or regression function is used to describe how the response's mean is affected by the predictors. While nonparametric regression methods with few assumptions can be useful for summarizing certain regression problems, using parametric regression models with a few plausible assumptions allows for elegant and clear results. This article discusses a significant class of linear regression models. These models are widely used in practice and can provide a variety of useful outcomes. Easy and multiple regression are addressed, as well as the specific types of equations, analysis of estimates and criteria, research ideas, including group comparisons, and diagnostic techniques.

The first of the two goals of regression analysis will be the subject of this paper. The aim is to use a regression model to predict the performance temperatures of air and water from the non-ideal heat exchanger model. Linear regression approaches, such as those used in chemometrics, and nonlinear regression methods, such as those used in neural networks, are also considered.

3.1.1 Linear regression

A linear strategy to modeling the connection between a scalar answer and one or more explanatory factors is known as linear regression. Simple linear regression is used when there is only one variable and multiple linear regression is used when there is more than one.[8]

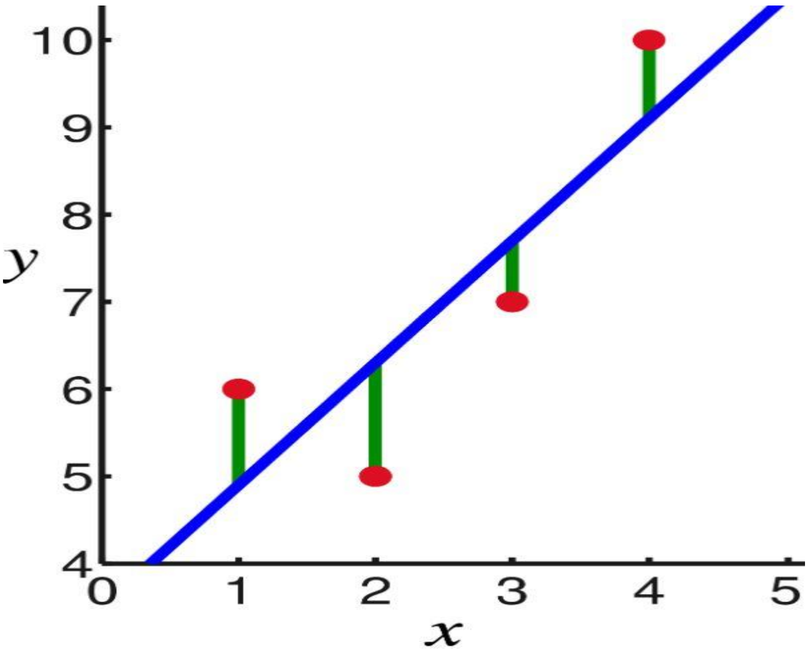

Figure 3-1 shows relationship between dependent variables(y) and independent variables(x)[9]

The linear regression modules assume that the connection between dependent variables y and n-vector regressors x is linear for a data set $\{y_{i,} x_{i1,\dots.} x_{ip}\}_{i=1}^{n}\}$ of n unit.

A disturbance term, often known as an error ε, is an unobserved random variable that adds "noise" to the linear connection between the dependent variable and the regressors. Then the model is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x i_p + \varepsilon_i = X_i^T \beta \qquad 3\text{-}1$$

These are also written in matrix form as

$$y = X\beta + \varepsilon \qquad 3\text{-}2$$

Where

$$y = \begin{pmatrix} y_i \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$X = \begin{pmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

Here,

y is the regressand or dependent variables, which is a vector of observed values.

X is the matrix of row-vector x_i of n-dimensional known as independent variables or regressors.

β is a dimensional parameter vector in which β_0 represents the intercept and the elements represent the regression coefficient.

ε_i is the error term, which is a vector of values ε_i

To fit a linear model to a given data set, the regression coefficients β must be estimated in such a way that the error is minimized. Generally,ordinary least square method is used.

3.1.2 Ordinary least squares(OSL)

For estimating the unknown parameters in a linear regression model, ordinary least squares (OLS) is a sort of linear least-squares approach. By decreasing the sum of the squares of the differences between the dependent variable in the provided data and those estimated by the linear function of the independent variable, OLS finds the parameters of a linear function of a collection of explanatory variables.[10]

Consider a system

$$\sum_{j=1}^{p} X_{ij} B_j = y_i, (i = 1,2 \dots, n), \qquad 3\text{-}3$$

Of n equation in o unknown coefficients. In matrix form it can be written as:

$$y = X\,\beta \qquad 3\text{-}4$$

where

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix},$$

$$y = \begin{pmatrix} y_i \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \qquad \text{and } \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}$$

Because such systems do not have a precise solution, the objective is to identify β that fits the equation in the context of solving the problem[11].

$$\widehat{\beta} = \text{arg min } S(\beta) \qquad 3\text{-}5$$

Where

$$S(\beta) = \sum_{j=1}^{p} \left| y_i - \sum_{j=1}^{p} x_{ij}\,\beta_j \right|^2 = ||y - X\,\beta||^2 \qquad 3\text{-}6$$

If the p columns of the matrix X are linearly independent, then this minimization issue has a single solution, which may be found by solving the normal equations.

$$(X^T X)\widehat{\beta} = X^T y \qquad 3\text{-}7$$

Where

$X^T X$ is the normal matrix

$X^T y$ is the moment matrix and

$\widehat{\beta}$ is the coefficient vector expressed as

$$\widehat{\beta} = (X^T X)^{-1} X^T y \qquad 3\text{-}8$$

3.1.2.1 The coefficient of determination(R^2)

Here are some figures to compare related ordinary least squares (OLS) estimated regression models. One metric that explains how well a model matches a series of findings is the coefficient of determination. The coefficient of determination quantifies how much of the uncertainty in the regression and can be predicted using the regressors. R^2stands for relative goodness-of-fit and is defined as follows [8]:

$$R^2 = \frac{\text{Explained sum of squares}}{\text{Total sum of squares}} = \frac{\text{ESS}}{\text{TSS}} \qquad 3\text{-}9$$

For a zero-intercept multivariate regression model, total sum of square is given by:

$$\text{TSS} = \sum_{i=1}^{n} y_i^2 \qquad 3\text{-}10$$

And in matrix form as:

$$\text{TSS}=\text{Y}^\text{T}\text{Y} \qquad 3\text{-}11$$

When model contained an intercept, it needs to be mean-corrected.

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \qquad 3\text{-}12$$

Here, $\bar{y}$ is the mean.

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad 3\text{-}13$$

In matrix notation TSS can be expressed as:

$$\mathrm{TSS} = \mathrm{Y}^{\mathrm{T}}\mathrm{Y} - n\bar{y}^2 \qquad \text{3-14}$$

And TSS need to be partition as:

$$\mathrm{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \qquad \text{3-15}$$

$$= \sum_{i=1}^{n}(y_i - \hat{y}_1 + \hat{y}_1 - \bar{y})^2$$

$$= \sum_{i=1}^{n}(y_i - \hat{y}_1)^2 + \sum_{i=1}^{n}(\hat{y}_1 - \bar{y})^2 + 2\sum_{i=1}^{n}(y_i - \hat{y}_1)(\hat{y}_1 - \bar{y})$$

$$= \sum_{i=1}^{n}(y_i - \hat{y}_1)^2 + \sum_{i=1}^{n}(\hat{y}_1 - \bar{y})^2 + 2\sum_{i=1}^{n}(y_i - \hat{y}_1)e_i$$

Since sum of residual is zero the last term in the previous expression also becomes zero

$$\mathrm{TSS} = \sum_{i=1}^{n}(\hat{y}_1 - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_1)^2 \qquad \text{3-16}$$

Here, the first term in the previous expression is called mean-corrected explained sum of squares.

$$\mathrm{ESS} = \sum_{i=1}^{n}(\hat{y}_1 - \bar{y})^2 \qquad \text{3-17}$$

In matrix form:

$$\mathrm{ESS} = \hat{\beta}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}\mathrm{X}\hat{\beta} - n\bar{y}^2 \qquad \text{3-18}$$

The residual sum of squares can be expressed as:

$$\mathrm{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_1)^2 \qquad \text{3-19}$$

In matrix form:

$$\sum e_i^2 = e^{\mathrm{T}} e \tag{3-20}$$

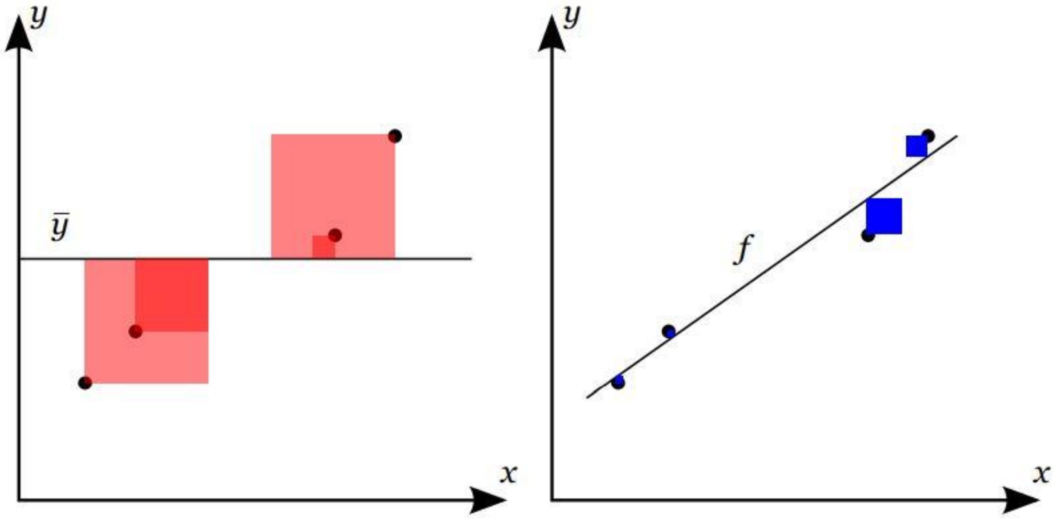

Figure 3-2: coefficient of determination[9]

For a model with an intercept, the coefficient of determination can be expressed as:

$$\mathrm{R}^2 = \frac{\hat{\beta}^{\mathrm{T}} \mathrm{X}^{\mathrm{T}} \mathrm{X} \hat{\beta} - n\bar{y}^2}{\mathrm{Y}^{\mathrm{T}} \mathrm{Y} - n\bar{y}^2} \tag{3-21}$$

and for zero intercept:

$$\mathrm{R}^2 = \frac{\hat{\beta}^{\mathrm{T}} \mathrm{X}^{\mathrm{T}} \mathrm{X} \hat{\beta}}{\mathrm{Y}^{\mathrm{T}} \mathrm{Y}} \tag{3-22}$$

In terms of RSS coefficient of determination can be expressed as:

$$\mathrm{R}^2 = 1 - \frac{\text{Residual sum of squares}}{\text{Total sum of squares}} \tag{3-23}$$

$$= 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

$$= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_1)^2}{\sum_{i=1}^{n}(\hat{y}_1 - \bar{y})^2}$$

In Figure 3-2the coefficient of determination is:

$$\mathrm{R}^2 = 1 - \frac{\sum \text{Area in blue}}{\sum \text{Area in red}} \qquad \text{3-24}$$

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. Where k is the regression parameters, and n is the number of observations [12]

The model with an intercept is expressed as:

$$\mathrm{R}^2 = 1 - \frac{e^{\mathrm{T}}e}{\mathrm{Y}^{\mathrm{T}}\mathrm{Y} - n\bar{y}^2} \qquad \text{3-25}$$

And also, for zero intercept:

$$\mathrm{R}^2 = 1 - \frac{e^{\mathrm{T}}e}{\mathrm{Y}^{\mathrm{T}}\mathrm{Y}} \qquad \text{3-26}$$

A useful property of R^2 is that it usually varies from 0 to 1, with one indicating optimal fit and zero indicating little change over the so-called mean model, making measuring the model's goodness-of-fit more intuitive. On the other hand, a well-known property of R^2 is that it grows every time a new regressor is applied to the model, even though the model's fit does not change. Then, to make R^2 an unbiased estimator, it must be modified to use the model's degrees of freedom(df). R^2 is adjusted for the number of regressors in relation to the number of measurements using degrees of freedom[13].

$\overline{\mathrm{R}}^2$ is the modified R^2 , and it is written as:

$$\overline{\mathrm{R}}^2 = 1 - \frac{\frac{\mathrm{RSS}}{\mathrm{df_{res}}}}{\frac{\mathrm{TSS}}{\mathrm{df_{tot}}}} \tag{3-27}$$

$$= 1 - (1 - \mathrm{R}^2)\frac{n-1}{n-k} \tag{3-28}$$

When number of observations are smaller than number of regression parameters than $\overline{\mathrm{R}}^2$ can take negative values.

3.1.2.2 The standard error of regression

If the key goal of the regression model is estimation, then one metric, the root mean square error(RMSE), takes precedence over the others. It's called the standard error of the regression or the Standard Error of the Estimate when the degrees of freedom for the residuals are taken into account[14].

RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\text{Mean Square Error}} \tag{3-29}$$

$$= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_1)^2}{n}}$$

$$= \sqrt{\frac{\mathrm{RSS}}{n}}$$

The number of observations is denoted by the letter n. The standard error of the regression is the standard deviation of the residuals, and is defined as:

$$= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_1)^2}{n-k}} \tag{3-30}$$

The number of parameters in the model is given by k. The disparity between RMSE and regression standard error would be negligible if the number of observations is much greater than the number of coefficients in the regression model. The standard error of the regression is useful because it has the same unit as the regressand and, unlike R^2 or $\overline{\mathrm{R}}^2$, it is an absolute metric of goodness-of-fit, with a lower value indicating a better fit[15].

3.2 Linear Regression of Counter Current Heat Exchanger model

To repeat what was mentioned at the outset of this chapter, one of the goals of this project is to reduce the time it takes to solve the non-ideal heat exchanger model (temperature dependence in the specific heat capacities of water and air). Then, in this section, an explicit data-driven model for the non-ideal heat exchanger model is constructed using linear regression and expressed as a correction expression to the ideal heat exchanger model and with the estimated parameters from this model hybrid model is developed to speed up the solution time.

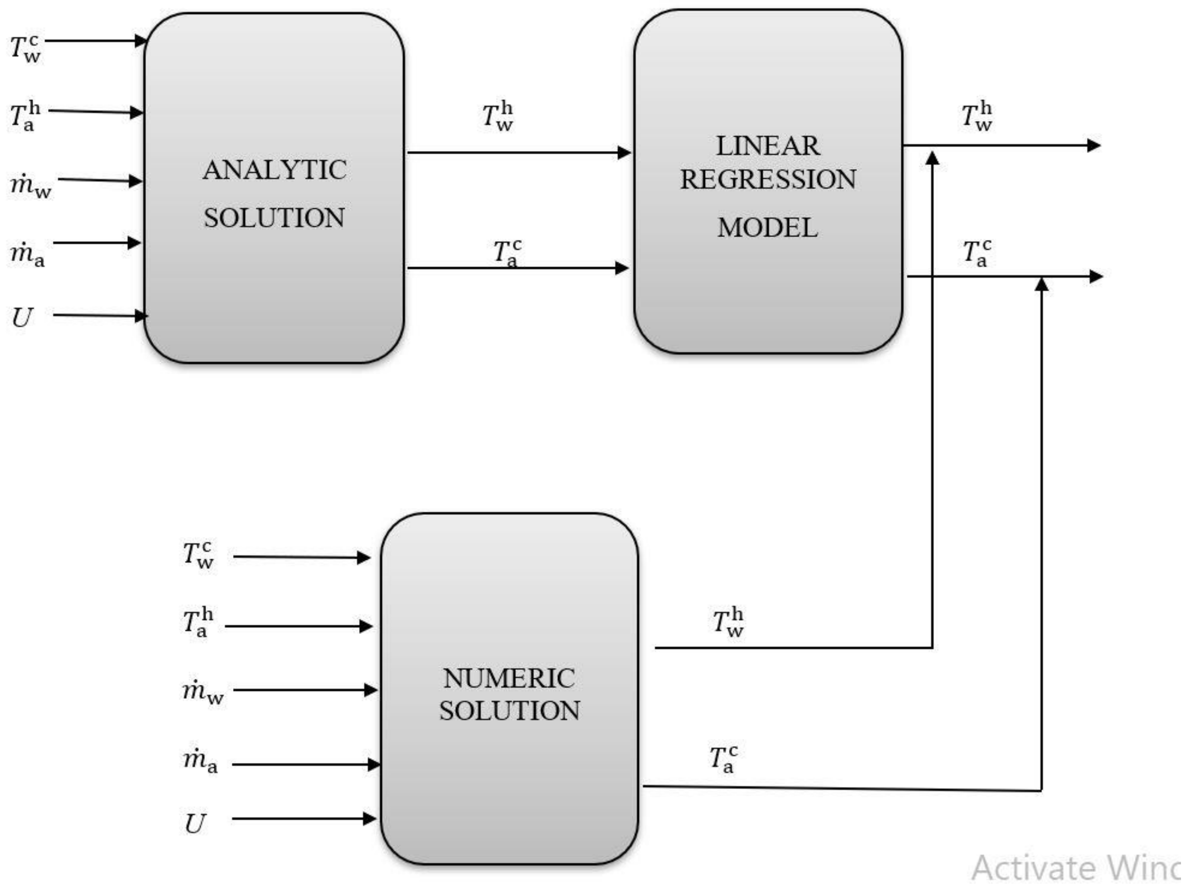

Figure 3-3: Explicit data-driven model using linear regression.

3.2.1 Linear Regression in Julia implementation

```julia
#Analytic model
function hex_a(inputs,par,x)
# If x is an Array/tuple, the function can generate the temperature profile across x.
    Twc, Tah, mdw, mda, UAx= inputs
    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  # Difference in Stanton numbers, -
    #
    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
sol_analytic = map(x -> hex_a(u_hex, par_hex_a, x), xspan_a) |> vec2vec
```

The function (hex_a) in the above code takes the inputs, the parameters, and the interval x as argument, and returns temperature of air and water whose values are displayed by sol_analytic.

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

In the above code, since heat capacity are no longer constant parameters, we can see in line 10 and 11,specific heat capacities are functions of temperatures.

First, the ideal and non-ideal heat exchanger models are solved for several conditions ($T_{\mathrm{w}}^{\mathrm{c}}, T_{\mathrm{a}}^{\mathrm{h}}, \dot{m}_{\mathrm{w}}, \dot{m}_{\mathrm{a}}, U$) to generate data matrices of analytic and numeric solutions, respectively. Then the parameter beta(β)is estimated by linear regression on the data matrices.

```
# Experimental ranges
n_Tw = 4
n_Ta = 4
n_mdw = 4
n_mda = 4
n_UAx = 4
#
r_Twc = range(4.,30.,length=n_Tw)
r_Tah = range(40.,100.,length=n_Ta)
r_mdw = range(20,200,length=n_mdw)
r_mda = range(20,200,length=n_mda)
r_UAx = range(20,200,length=n_UAx)
x_grid = [[Twc,Tah,mdw,mda,UAx] for Twc in r_Twc, Tah in r_Tah,mdw in r_mdw,mda in r_mda,UAx i
ngrid = length(x_grid)
XAgrid = Matrix{Float64}(undef,5,ngrid)
TNgrid = Matrix{Float64}(undef,2,ngrid)
 for i in 1:ngrid
    Twc,Tah,mdw,mda,UAx = x_grid[i]
    u_hex = [Twc Tah mdw mda UAx ]
    par_hex_a = [ chpw chpa]
    par_hex_n_Cp_T_dep = [ cp_w cp_a]
    sol_analytic = map(dx -> hex_a(u_hex, par_hex_a, dx), xspan_a)
    XAgrid[:,i] .= [sol_analytic[end][1],sol_analytic[1][end],mda,mdw,UAx]
    sol_numeric = hex_n_Cp_T_dep(u_hex, par_hex_n_Cp_T_dep, xspan_n)
    TNgrid[:,i] .= [sol_numeric[end][1],sol_numeric[1][end]]
end
X= [phi_m(Xscale[:,i];n=norder) for i in 1:size(Xscale,2)]  |> x -> reduce(hcat,x);
Y= TNgrid
beta=Y/X
```

3.2.2 Fitting the data-driven model

To test the regression model's ability to forecast the future, and to find the model that best describes the data, Root means square error(RMSE) was determined for multiple orders.

```
#calculating error
nmax = 12
e1 = zeros(nmax)
for j in 1:nmax
    Phi = [phi_m(Xscale[:,i];n=j) for i in 1:size(Xscale,2)]  |> x -> reduce(hcat,x)
    println(cond(Phi))
    beta= Yscale/Phi
    E1= Yscale-beta*Phi
    e1[j] = norm(E1)/sqrt(length(Yscale))
end
```

In the above code, error up to the 12th order was determined and nmax denotes the maximum number of orders to be generated, whereas e1 denotes the RMSE to be generated for all orders to achieve the best fit.

```
# j = [1 to 12 orders]
function TimeCalc(j)
    Phi = [phi_m(Xscale[:,i];n=j) for i in 1:size(Xscale,2)]  |> x -> reduce(hcat,x)
    beta= Yscale/Phi
    E1= Yscale-beta*Phi
    RMSE = norm(E1)/sqrt(length(Yscale))
end
t1 = @benchmark TimeCalc(1)
t2 = @benchmark TimeCalc(2)
t3 = @benchmark TimeCalc(3)
t4 = @benchmark TimeCalc(4)
t5 = @benchmark TimeCalc(5)
t6 = @benchmark TimeCalc(6)
t7 = @benchmark TimeCalc(7)
t8 = @benchmark TimeCalc(8)
t9 = @benchmark TimeCalc(9)
t10 = @benchmark TimeCalc(10)
t11 = @benchmark TimeCalc(11)
t12 = @benchmark TimeCalc(12)
```

The above code's function (TimeCalc) is used to calculate time for RMSE generated up to 12th order. Benchmark Tool’s main macro is @benchmark, which is used to see the total time taken to complete a process in Julia.

3.2.2.1 Simulation results

ORDER	SAMPLE	TIME				RMSE
		Minimum Time	Median Time	Mean Time	Maximum Time	
1	5280	396.200 µs	571.400 µs	939.725 µs	27.096 ms	0.0425398638142452
2	4990	396.200 µs	595.500 µs	994.118 µs	18.939 ms	0.028367246996547312
3	573	4.384 ms	6.447 ms	8.712 ms	30.308 ms	0.01610987759101991
4	177	18.337 ms	28.689 ms	28.290 ms	46.347 ms	0.009232005001767048
5	128	28.319 ms	37.853 ms	39.091 ms	77.413 ms	0.005053013888943954
6	60	62.311 ms	80.537 ms	83.536 ms	141.195 ms	8.720849925285297e-15
7	31	123.103 ms	148.956 ms	162.952 ms	411.333 ms	3.980227263007599e-15
8	21	202.854 ms	213.236 ms	239.858 ms	361.964 ms	1.2888472448938928e-14
9	15	272.091 ms	322.650 ms	338.899 ms	510.504 ms	1.1073878713191111e-13
10	10	428.683 ms	444.786 ms	507.211 ms	740.337 ms	2.478575600507276e-13
11	7	674.968 ms	715.101 ms	753.483 ms	875.286 ms	1.1531462754468309e-12
12	5	1.034 s	1.097 s	1.148 s	1.325 s	1.3036504819103911e-12

Table 3-2: Validation results

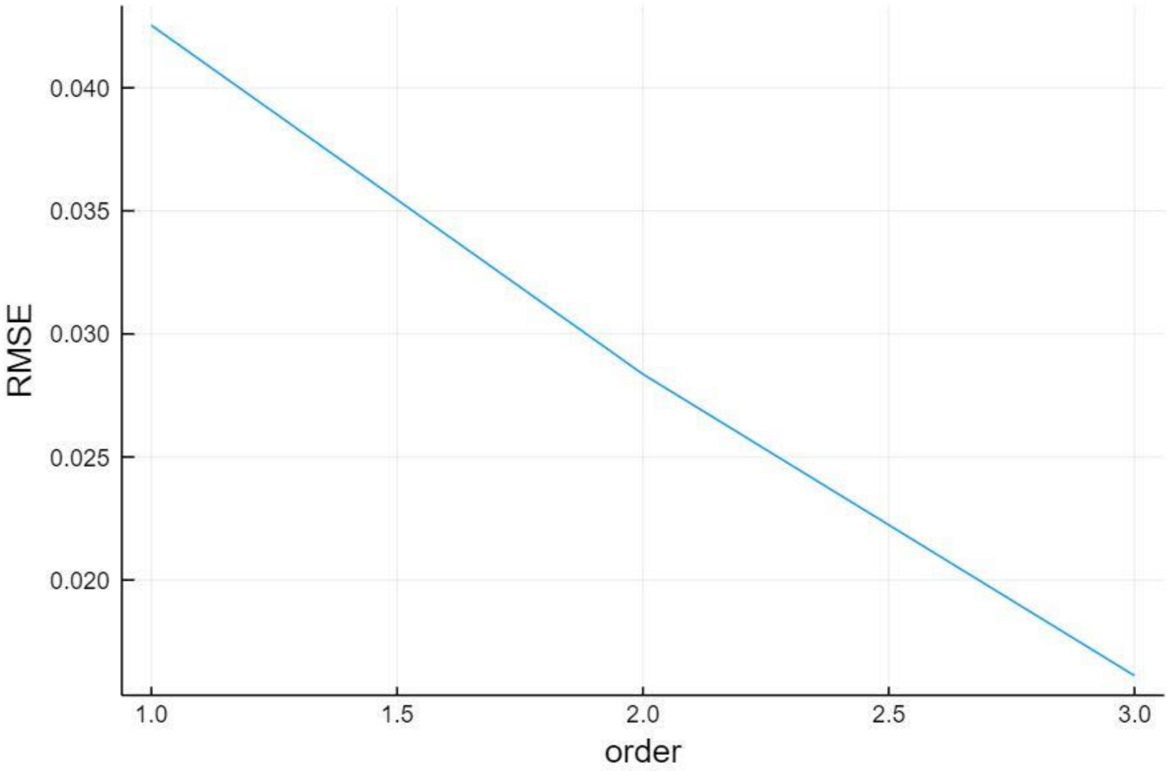

Figure 3-4:RMSE vs 3rd model order

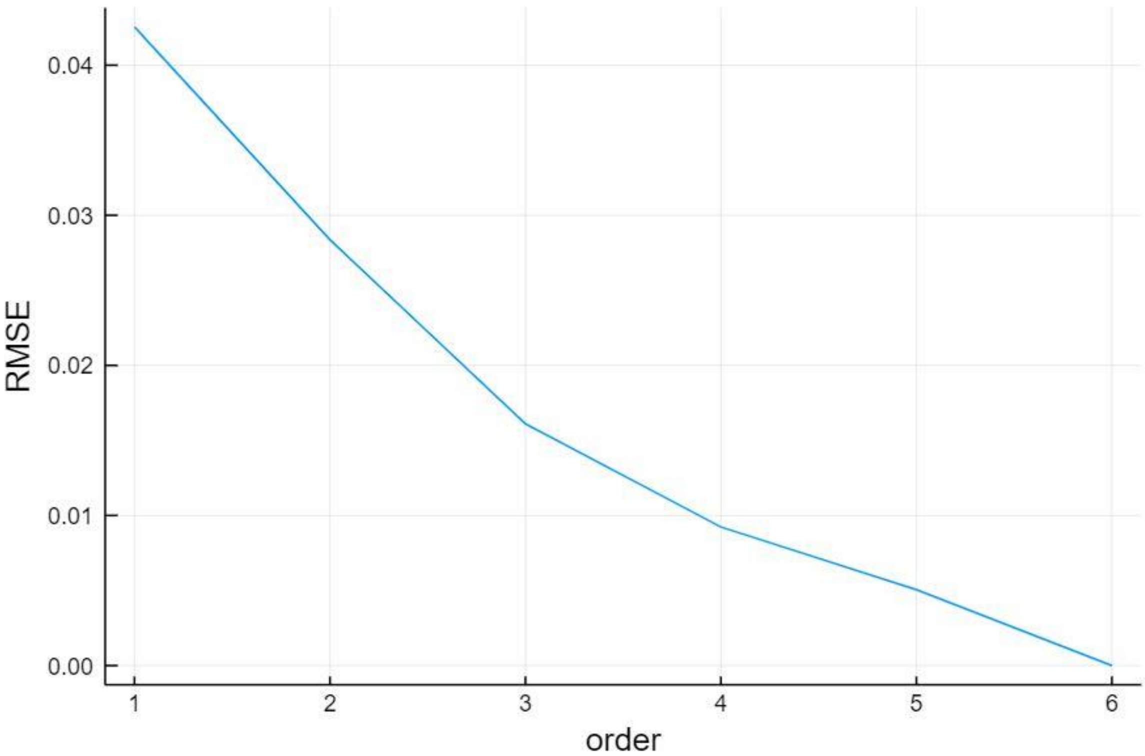

Figure 3-5:RMSE vs 6th model order

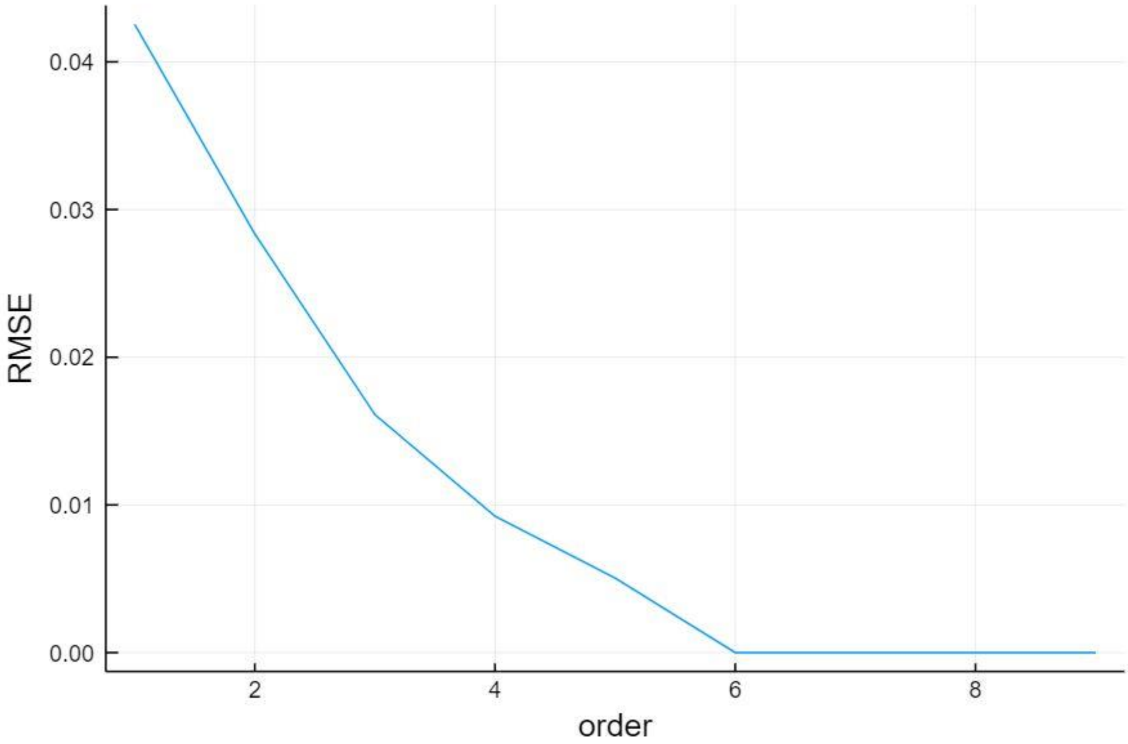

Figure 3-6:RMSE vs 9th model order

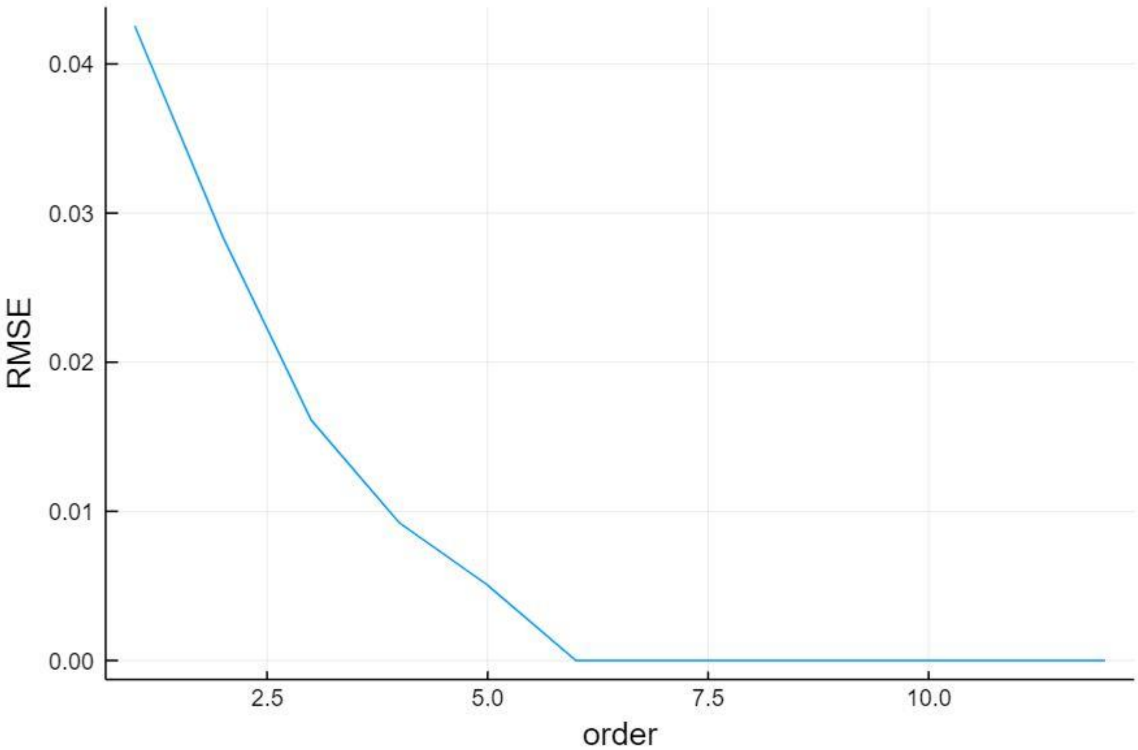

Figure 3-7:RMSE vs 12^{th} model order

3.2.3 Running the hybrid model(mechanistic + data-driven/empirical/machine learning model) using linear regression

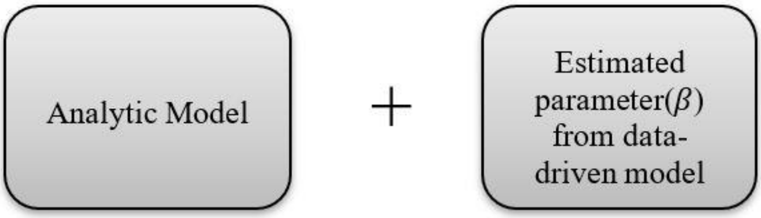

Figure 3-8:Hybrid model

In this section, the estimated parameters(β) from the data-driven model are combined with an analytic model termed as a hybrid model, and Julia is used to execute it, as seen below.

```
#combining analytic model with regression model
function timecalc(k)
    function hex_a(inputs,par,beta)
    Twc, Tah, mdw, mda,UAx = inputs
    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  # Difference in Stanton numbers, -
    #
    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
    Phi = [phi_m(Xscale[:,i];n=norder) for i in 1:size(Xscale,2)]  |> x -> reduce(hcat,x)
    Ys=beta*Phi
    return [Ys[1] Ys[end]]
    end
end
@benchmark timecalc(k)
```

Using function(Timecalc) the time required to run the model is seen. Benchmark Tool's main macro is @benchmark, which is used to see the total time taken to complete a process in Julia.

ORDER	MINIMUM TIME	MEDIAN TIME	MEAN TIME	MAXIMUM TIME
1	3.425 µs	3.538 µs	4.908 µs	582.275 µs
2	7.525 µs	7.825 µs	9.474 µs	844.000 µs
3	18.600 µs	19.900 µs	20.507 µs	3.329 ms
4	50.699 µs	51.600 µs	53.132 µs	6.143 ms
5	99.599 µs	100.200 µs	102.354 µs	6.353 ms
6	180.299 µs	181.100 µs	183.113 µs	4.389 ms
7	178.001 µs	188.200 µs	191.017 µs	6.233 ms
8	265.201 µs	272.900 µs	280.980 µs	3.718 ms
9	435.900 µs	446.401 µs	469.712 µs	6.513 ms
10	648.301 µs	673.000 µs	715.323 µs	6.478 ms
11	932.900 µs	954.399 µs	989.128 µs	7.744 ms
12	1.269 ms	1.294 ms	1.331 ms	16.591 ms

Table 3-3: Benchmark results non-ideal heat exchanger with linear regression

3.3 Non-linear Regression

Nonlinear regression is a mathematical model that uses a generated line to match an equation to a set of data. Nonlinear regression, unlike linear regression, which uses a straight-line equation, reveals correlation across a curve, rendering it nonlinear in the parameter.

A simple nonlinear regression model is expressed as[16]:

$$Y = f(x, \beta) + \varepsilon \qquad \text{3-31}$$

Where:

x is a vector of P predictors

β is a vector of K predictors

f is the known regression function and

ε is the error term

Apart from nonlinear regression models, multivariate adaptive regression splines (MARS), classification and regression trees (CART), and prediction pursuit regression (PPR) have all been implemented into chemometrics, and they have gained less consideration than linear statistical methods and neural networks for empirical modeling. A popular context that highlights the similarities and differences between analytical modeling approaches such as CART and PPR can be used to gain insight into the relationship between them. The paradigm is built on the idea that all empirical modeling approaches can be expressed as a weighted sum of basis functions. Both empirical modeling approaches can be interpreted as a weighted sum of basis functions, according to the framework as[17]:

$$\hat{y}_k = \sum_{m=1}^{M} \beta_{mk} \emptyset_m(\varphi_m(\alpha; X) \qquad \text{3-32}$$

Where the kth expected performance is denoted by $\hat{y}_k$, β_{mk} is the output weight, α as matrix function parameters, X are the inputs and input transformation denoted by $\emptyset_m$ and.As implemented in chemometrics, the neural network is implemented for empirical modeling in our task.

3.3.1 Neural networks

In nonlinear approximation and pattern recognition, neural networks have been extensively used. Neural networks can be thought of as a nonlinear input and output paradigm when used for forecasting[18].

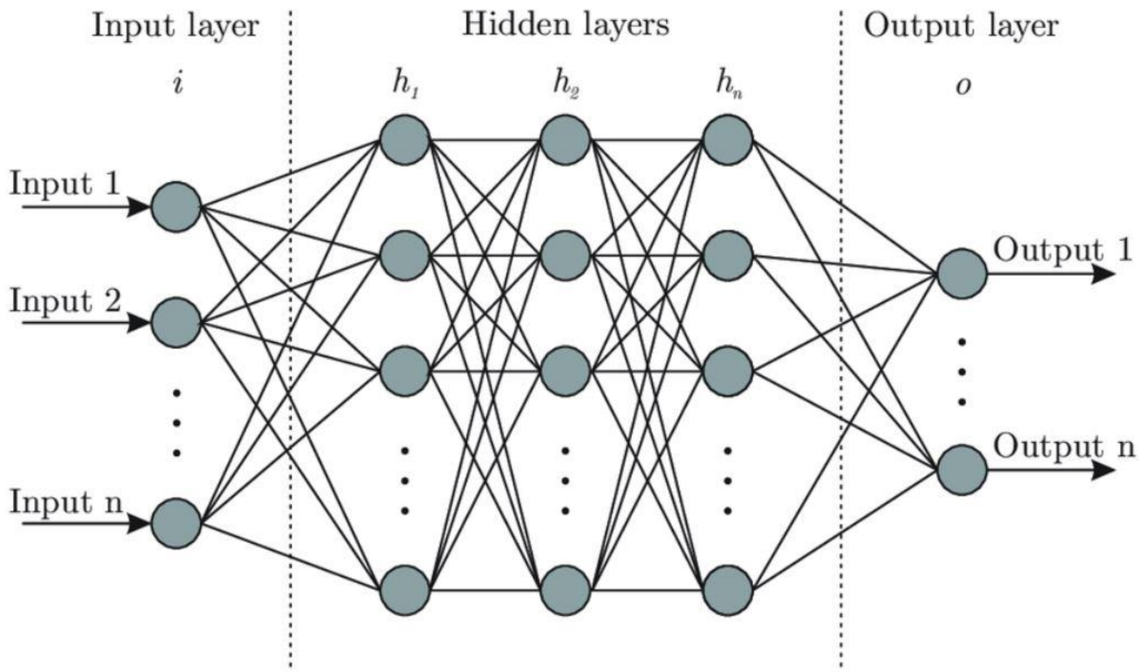

Figure 3-9:multi-layer neural network[19]

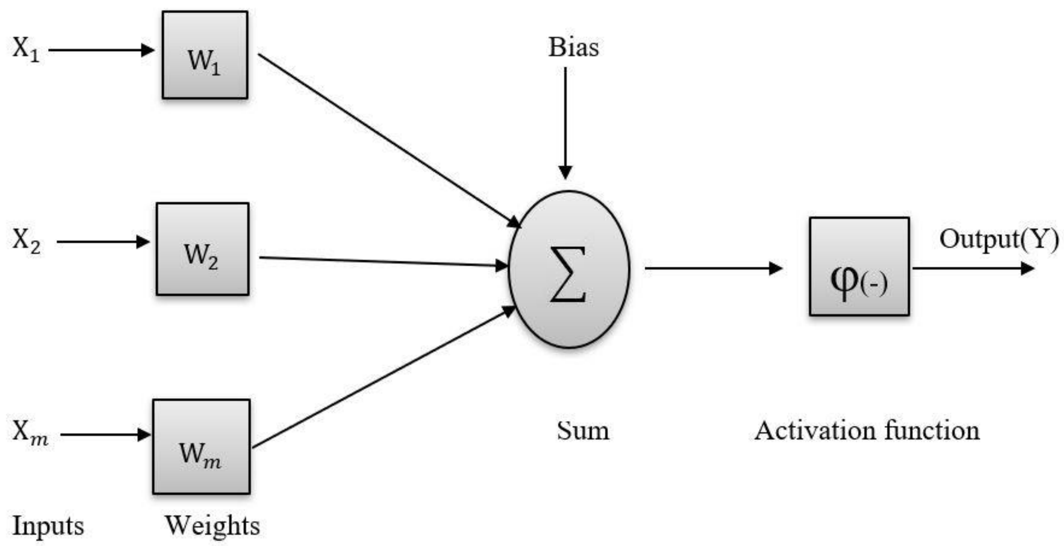

Figure 3-10:simple mechanism[16]

Neural networks are made from layers of neurons. The main processing units of the system are the neurons. In the input layer it receives the input, the output layer estimates our final output. In between exist the hidden layers which perform most of the computations required by our network. When the input is fed to each neuron of the first layer neurons of one layer are interconnected to neurons of the next layer through channels. Each of those channels is assigned a numerical value which is known as weight. The inputs are multiplied by the weights, and the sum is given to the hidden layer neurons as input. Each of those neurons is related to a numerical value which is known as bias, which is then added to the input sum. This value is then tried and true a function called the activation function.[20]

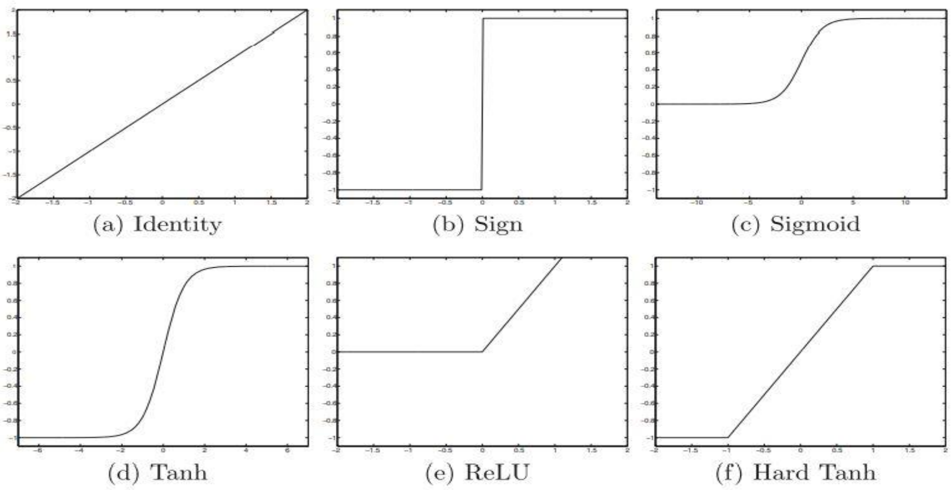

Figure 3-11:Various activation function[21]

The output of the activation function decides if the neuron will get activated or not. An activated neuron transmits information to the neurons of the following layer over the channels. This is often called forward propagation within the output layer the neuron with the very best value fires and determines the output the values are probable. Initially, when output is predicted and it is a wrong prediction, it is not the end because the model needs to be trained. At this training process together with the input, the network result is fed to that the expected output and compared against the output to understand the error. In prediction, the magnitude of the error suggests if our predicted values are bigger or smaller than expected results. Here indicates the direction and magnitude of change to reduce the error. This information is transferred backward through our network which is known as backpropagation. Now supported by this information, the weights are adjusted. This cycle of forwarding propagation and backpropagation is iteratively performed with number of inputs. This process goes on until our weights are assigned such that the network can predict the output correctly [18].

3.3.2 Non-linear Regression of Counter Current Heat Exchanger model

Given the success of an explicit data-driven approach for the non-ideal heat exchanger model using linear regression, it is still worth considering using non-linear regression for the optimal match. So, an explicit data-driven model for the non-ideal heat exchanger model is built in this section. To shorten the solution time, a correction expression to the optimal heat exchanger model is used.

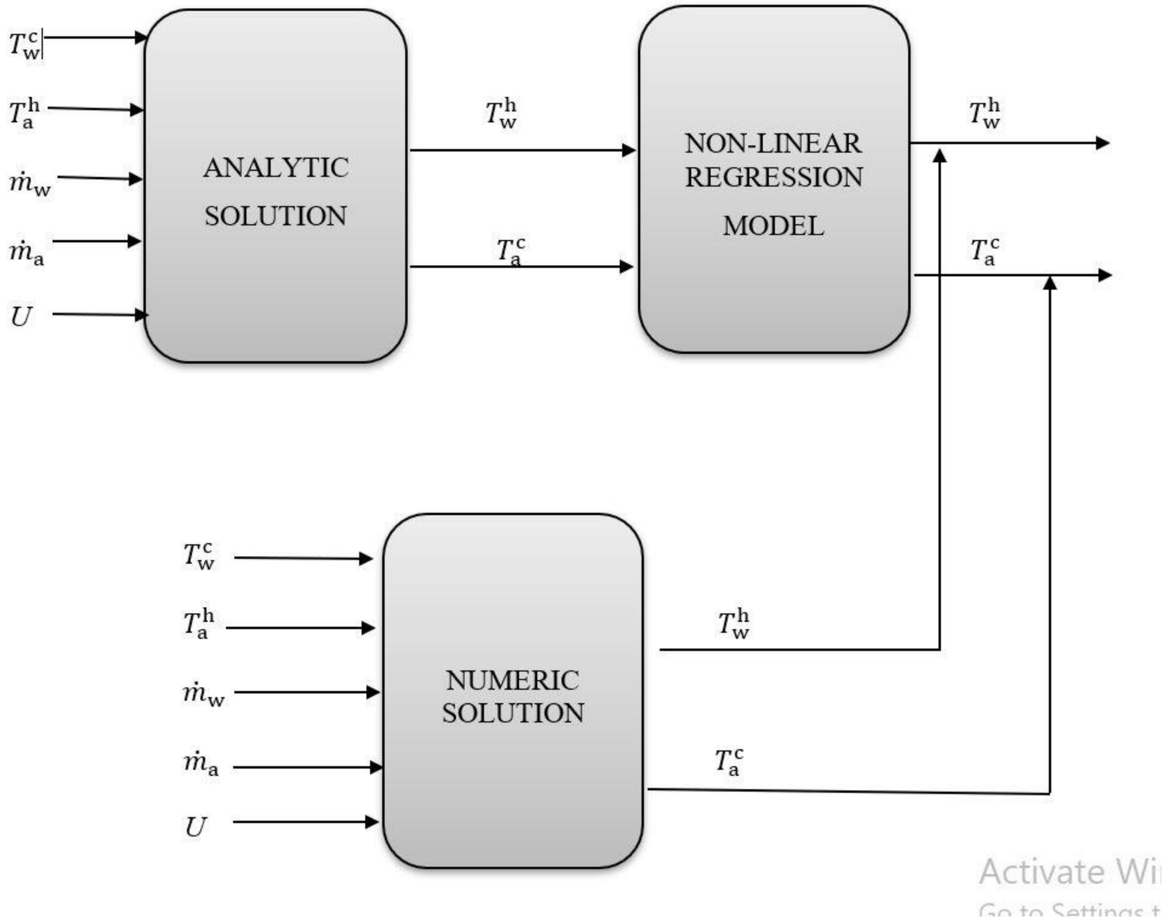

Figure 3-12:explicit data driven model using non-linear regression.

3.3.3 Non-linear regression in Julia implementation

```
# Normalization function
function normalize(X)
    Xmin = minimum(X,dims=2)
    Xmax = maximum(X,dims=2)
    Xnorm = (X .- Xmin)./(Xmax-Xmin)
    denormalize = (Xnorm,Xmin,Xmax) -> Xnorm.*(Xmax-Xmin).+Xmin
    return Xnorm, Xmin,Xmax,denormalize
end
# no of epochs
nE = 50000
# no of internal node
nz = 2
# defining model
md = Chain(Dense(5,nz,tanh),Dense(nz,2))
# Initial mapping
Yd_0= md(Xd)
#Defining loss function
loss(x, y) = mean((md(x).-y).^2)
#specyfying optimization method
opt = ADAM(0.01, (0.99, 0.999))
# Extracting parameters from the model
par = Flux.params(md)
# Training over nE epochs
for i in 1:nE
    Flux.train!(loss,par,data,opt)
end
# Final mapping
Yd_nE = md(Xd)
# Calculating error
RMSE = norm(Yd_nE-Yd)^2/sqrt(length(Yd))
```

The function(normalize) is used to normalize the value of X in the code above. As previously said, epochs are chosen, as well as a node with the activation function tanh for the first layer and identity for the second layer. The model is eventually conditioned, and the error is measured.

The below code's function (TimeCalc) is used to calculate time for RMSE generated up to the15th node. Benchmark Tool's main macro is @benchmark, which is used to see the total time taken to complete a process in Julia.

```
# time calculation
function TimeCalc(nz)
    #nz=node
    # defining model
    md = Chain(Dense(5,nz,tanh),Dense(nz,2))
    # Initial mapping
    Yd_0= md(Xd)
    #Defining loss function
    loss(x, y) = mean((md(x).-y).^2)
    #specyfying optimization method
    opt = ADAM(0.01, (0.99, 0.999))
    # Extracting parameters from the model
    par =Flux.params(md)
    # Training over nE epochs
    for i in 1:nE
        Flux.train!(loss,par,data,opt)
    end
    # Final mapping
    Yd_nE = md(Xd)
    # Calculating error
    RMSE = norm(Yd_nE-Yd)^2/sqrt(length(Yd))
end
t = @benchmark TimeCalc(nz)
```

3.3.4 Training the data-driven model

In this section, the model is trained with multiple epochs in Julia which can be seen in section 3.3.3 and the results obtained are summarized.

3.3.4.1 Simulation results

NODE	MEAN TIME	RMSE
1	71.494 s	0.5548718311388271
2	75.703 s	0.003169573105621138
3	121.190 s	0.0014953212724717116
4	139.010 s	0.0013872511470593093
5	105.891 s	0.000922273658861804
6	157.149 s	0.0008863095123224582
7	159.512 s	0.0006584271926556707
8	167.705 s	0.000690118770099446
9	180.997 s	0.0006312129929776083
10	146.797 s	0.0006590233056936986
11	146.218 s	0.0006040803974399049
12	141.263 s	0.000574280213920806
13	142.986 s	0.0005705601620618436
14	160.483 s	0.0005402155956200594
15	150.124 s	0.0005423131502707424

Table 3-4:validation results with 50000 epochs

Epochs=50

Node=20

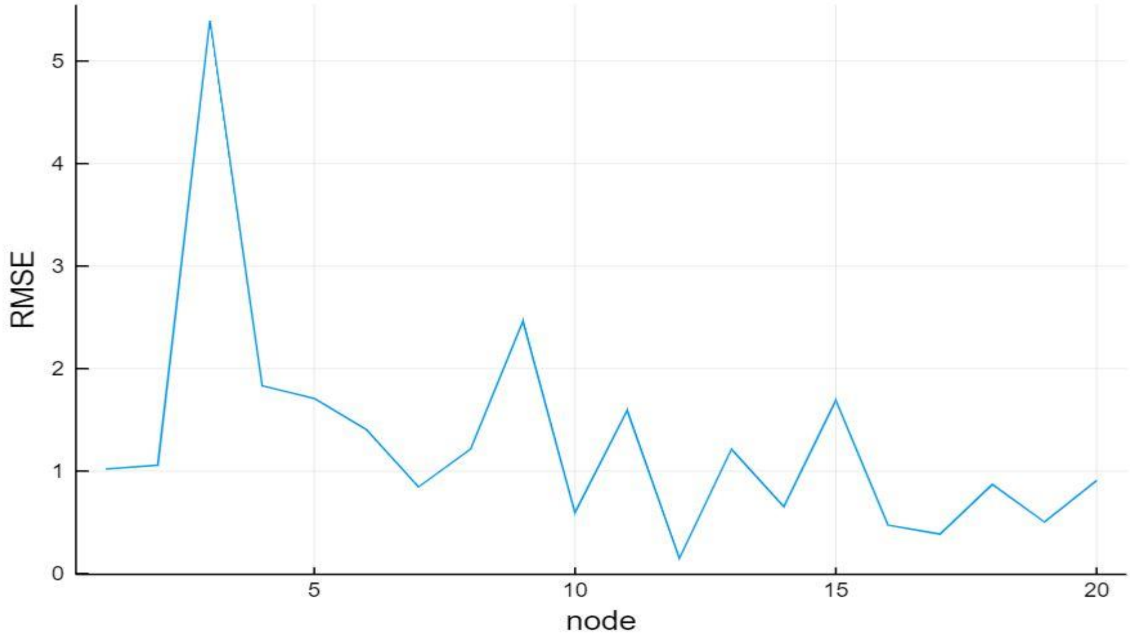

Figure 3-13:RMSE vs node with 500 epochs

Epochs=500

Node=20

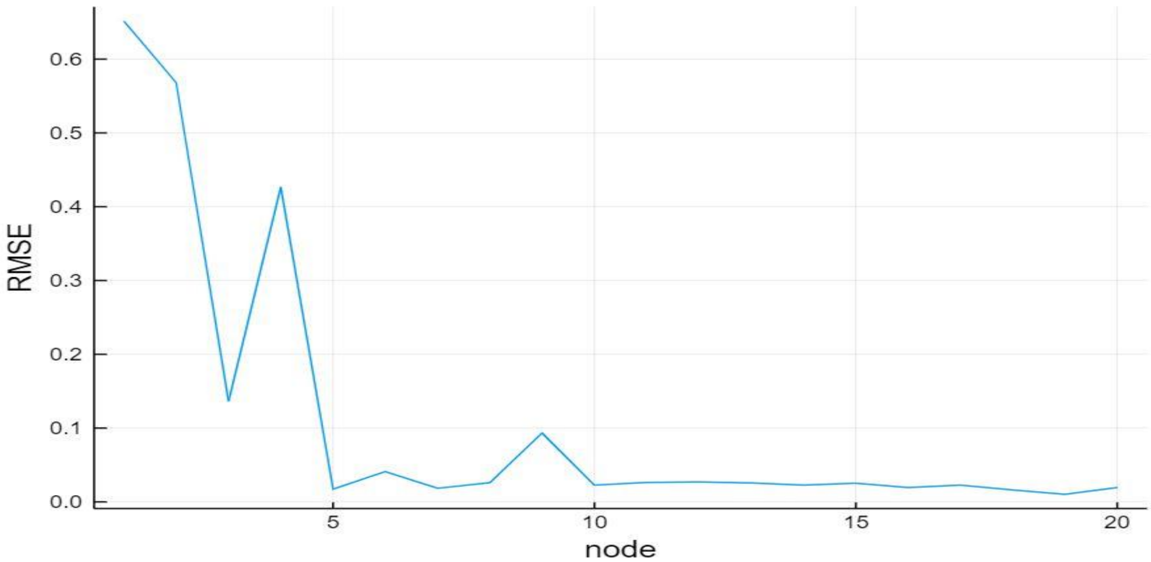

Figure 3-14:RMSE vs node with 5000 epochs

Epochs=50000

Node=15

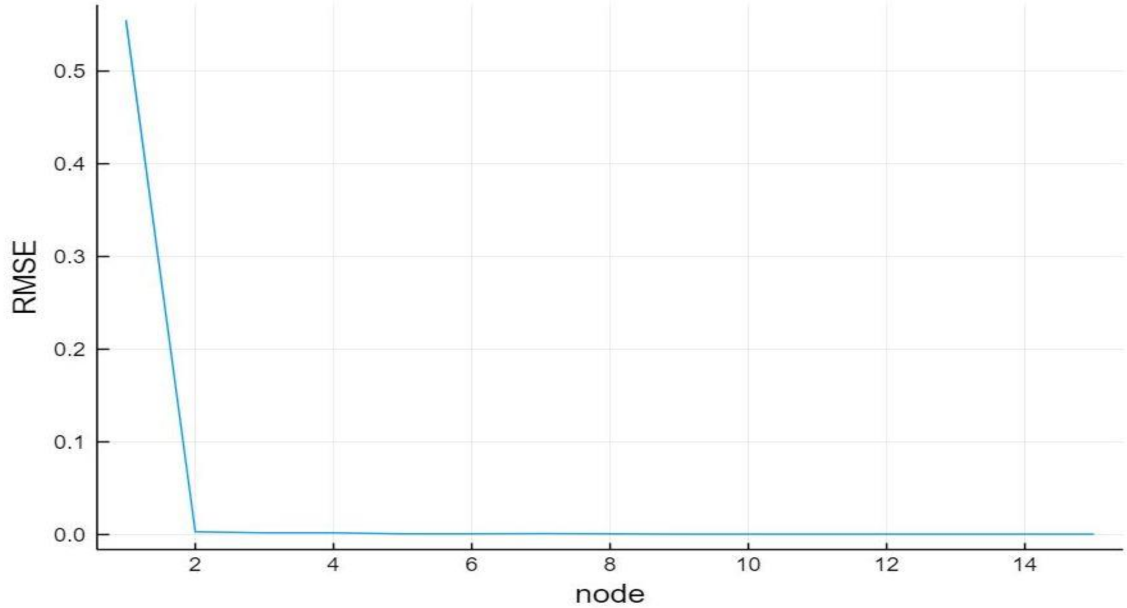

Figure 3-15:RMSE vs node with 50000 epochs

Epochs=50000

Node=15

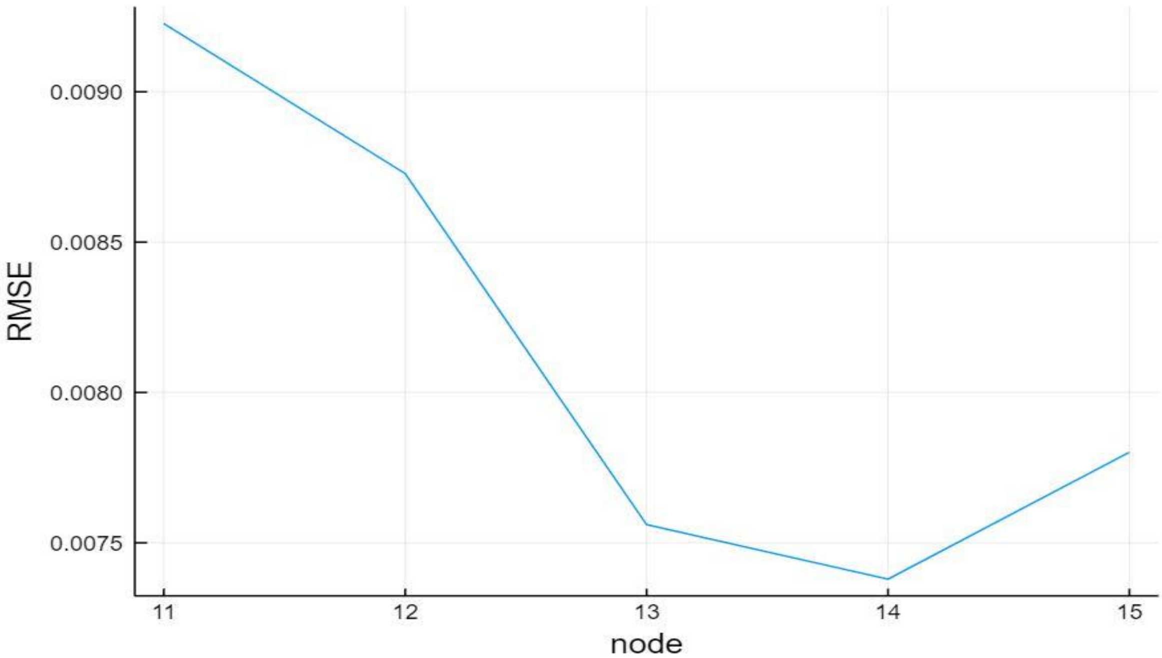

Figure 3-16:RMSE vs node with 50000 epochs

3.3.5 Running the hybrid(mechanistic + data-driven/empirical/machine learning model)

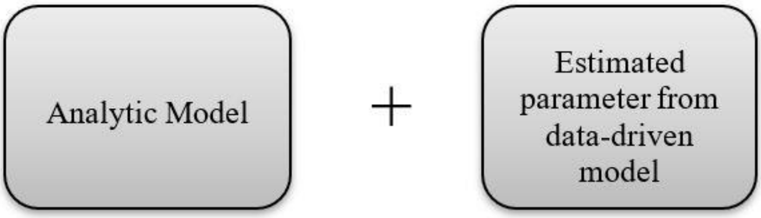

Figure 3-17:Hybrid model

The estimated parameters from data-driven model are combined with analytic model which is termed as hybrid model Figure 3-17 and Julia is used to execute it, as seen below.

```
#combining analytic model with non-regression model
function timecalc(k)
    function hex_a(inputs,par,md)
    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  # Difference in Stanton numbers, -
    #
    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
    Yd_nE = md(Xd)
    return [ Yd_nE[1]  Yd_nE[end]]
    end
end
@benchmark timecalc(k)
```

Using function(Timecalc) the time required to run the model is seen. Benchmark Tools main macro is @benchmark, which is used to see the total time taken to complete a process in Julia.

NODE	MINIMUM TIME	MEDIAN TIME	MEAN TIME	MAXIMUM TIME
1	49.300 µs	50.200 µs	57.175 µs	3.313 ms
2	74.099 µs	75.401 µs	83.200 µs	3.109 ms
3	143.499 µs	145.500 µs	157.641 µs	3.867 ms
4	156.899 µs	159.801 µs	177.033 µs	6.551 ms
5	198.099 µs	201.200 µs	221.701 µs	5.961 ms
6	216.700 µs	219.601 µs	241.105 µs	6.778 ms
7	900.899 µs	902.999 µs	930.397 µs	6.934 ms
8	285.599 µs	524.100 µs	535.524 µs	55.527 ms
9	317.600 µs	319.901 µs	344.208 µs	5.108 ms
10	383.001 µs	385.900 µs	414.426 µs	5.190 ms
11	401.200 µs	403.700 µs	434.663 µs	9.285 ms
12	448.200 µs	450.900 µs	481.260 µs	5.612 ms
13	467.000 µs	469.700 µs	498.906 µs	6.961 ms
14	1.685 ms	1.697 ms	1.732 ms	7.670 ms
15	499.101 µs	502.000 µs	537.954 µs	6.135 ms

Table 3-5:Benchmark results non-ideal heat exchanger with non-linear regression

4 Results and Discussion

This chapter summarizes the conclusions of the previous chapters' research and addresses the most important findings. The relation between the analytic and numeric solutions of the counter-current heat exchanger models is discussed first. Second, the counter-current heat exchanger model's regression findings are analyzed and clarified. Third, the hybrid models' execution speeds and the numeric solution of the non-ideal heat exchanger model are contrasted.

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

An overview of the thermal model of an air-cooled synchronous generator was given in Chapter 2, along with ideal and non-ideal heat exchanger models. Also addressed was the effect of temperature dependency in the basic heat capacities of air and water on the heat exchanger sub-model solution. It was specifically addressed how, based on the assumption of temperature dependency in the specific heat capacities, a linear/nonlinear two-point boundary value problem might arise.

Model	Median time	Mean time
The numeric solution when $\hat{c}_p$	4.665 ms	5.877 ms
The numeric solution when $\hat{c}_p(t)$	39.947 ms	38.786 ms
The Analytic solution	22.601 µs	23.118 µs

Table 4-1: Summary of the benchmark results of Chapter 2.

Table 4.1 shows that the numeric solution when specific heat capacities are temperature dependent is 4 to 5 times slower than the numeric solution when specific heat capacities are not temperature dependent. In the situation of temperature independence in the specific heat capacity of water and air , Table 4.1 reveals that the analytic solution is around 200 to 300 times quicker than the numeric approach. This supports the necessity for an explicit data-driven model to reduce the heat exchanger model's solution time which is done in chapter 3.

4.2 Regression of the Counter-Current Heat Exchanger Model

One of the primary goals of this project is to shorten the time it takes to solve the non-ideal heat exchanger model (the case of temperature dependency in the specific heat capacity of air and water), as described in Chapter 3. Specifically, linear, and nonlinear regression was used to build explicit data-driven models for the non-ideal heat exchanger model, which were then represented as a correct expression for the ideal heat exchanger model.

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

order	Median Time	Mean Time	RMSE
3rd	6.447 ms	8.712 ms	0.01610987759101991
6th	80.537 ms	83.536 ms	8.720849925285297e-15
9th	322.650 ms	338.899 ms	1.1073878713191111e-13
12th	1.097 s	1.148 s	1.3036504819103911e-12

Table 4-2:Computational time along with RMSE different order

Table 4-2 displays the summarized time i.e., minimum time, median time, mean time, and maximum time calculated with benchmark methods, as well as the RMSE calculated for each order to determine the right suit. It can also be seen in Figure 3-4,Figure 3-5 , Figure 3-7 and Figure 3-7 which graphically depicts the relationship between RMSE and different order. i.e., 3rd, 6th, and 12th respectively.

We can see that the time required to compute error increases as the order increases, while the RMSE decreases up to the sixth order and then steadily increases after that. In 6th order, the optimal match for an explicit data-driven model can be seen.

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

Epochs=50000

node	Mean Time	RMSE
1st	71.494 s	0.5548718311388271
5th	105.891 s	0.000922273658861804
10th	146.797 s	0.0006590233056936986

14^{th}	160.483 s	0.0005423131502707424

Table 4-3:Computational time along with RMSE for different node

Table 4-3 displays the summarized time calculated with benchmark methods, as well as the RMSE calculated for each node to determine the right suit. Figure 3-13 shows that with 50 epochs, the error is higher. If the number of epochs increases, the error decreases, as seen in Figure 3-14. Figure 3-15 shows a good match by using different epochs for a total of 50000 epochs. We can see that the time required to compute error increases as the node increases. The 14th node, the optimal match for an explicit data-driven model can be seen with minimum RMSE.

4.2.3 Comparison of the Execution Speed of the data-driven models

We can see the least error at 6th order for linear regression and 14th node for nonlinear regression, as mentioned in 3.2.2 and 3.3.4, which is then used to develop a hybrid model.

Model	Mean Time	Median Time
Data-driven(linear regression model)	80.573ms	83.536ms
Data-driven(non-linear regression model)	160.483 s	164.378ms

Table 4-4:benchmark results for data-driven model

As indicated in Table 4-4, it appears that linear regression yielded the lowest error when compared to nonlinear regression with less computing time for these data-driven models.

4.2.4 Comparison of the Execution Speed of the Hybrid model (mechanistic + data-driven/empirical/machine learning model) and the Numeric Non-Ideal Heat Exchanger Models

In 3.2.3 and 3.3.5 the hybrid and numeric non-ideal heat exchanger models were compared in terms of execution speed.

model	Mean Time	Median Time
hybrid(linear regression model)	181.100 µs	183.113 µs
hybrid(non-linear regression model)	1.697 ms	1.732 ms
The numeric solution when $\hat{c}_p(t)$	39.947 ms	38.786 ms

The Analytic Solution	22.601 μs	23.118 μs

Table 4-5:Benchmark result for hybrid model

To build the hybrid model, correction factors from data-driven models were employed, and the computing time after solving the hybrid model is displayed in Table 4-5. We can observe that the solution time has improved significantly and is now around 220 times quicker than Numeric Non-ideal heat exchanger models.

5 Conclusion

This thesis provides an overview of the thermal model of an air-cooled synchronous generator provided by [1] and explored by [2], as well as a discussion of the heat exchanger sub-probable model's extension to the scenario of temperature dependency in the specific heat capacities of air and water. Furthermore, explicit data-driven models were constructed using linear and nonlinear regression for a range of situations and stated as a correction expression to the ideal heat exchanger model to speed up the solution time of the non-ideal heat exchanger sub-model. Furthermore, the numeric solution of the nonlinear two-point boundary value issue was compared to the hybrid(mechanistic + data-driven/empirical/machine learning model)models in terms of execution speed and the hybrid models were shown to have a quicker execution time.

6 Future work

Following things can be done.

1. When solving the generator's dynamic thermal model, a hybrid model may be employed to see the computing time and remove the nonlinear two-point boundary value problem.
2. More experimental data may be used to assess the hybrid model's computational time.

References

[1] T. Øyvang, *Enhanced power capability of generator units for increased operational security*, vol. no. 21. Porsgrunn: University of South-Eastern Norway, Faculty of Technology, Natural Sciences and Maritime Studies, 2018.

[2] B. Lie, “FM 1015 Modeling of Dynamic Systems,” p. 16, 2018.

[3] K. Aleikish, “Hybrid Machine Learning and Mechanistic Thermal Model of Synchronous Generator,” p. 122.

[4] M. Pandey, “Model Fitting and State Estimation for Thermal Model of Synchronous Generator,” p. 119.

[5] D. S. I. Shahl, “II. Synchronous Generators,” p. 62.

[6] B. Lie, “Modeling of Dynamic Systems,” p. 666.

[7] K. Aleikish, M. Pandey, T. Øyvang, and B. Lie, “Hybrid Model for Fast Solution of Thermal Synchronous Generator With Heat Exchanger,” Mar. 2021, pp. 91–98. doi: 10.3384/ecp2017691.

[8] ad, “Assumptions of Classical Linear Regression Models (CLRM),” *Economic Theory Blog*, Apr. 01, 2015. https://economictheoryblog.com/2015/04/01/ols_assumptions/ (accessed May 16, 2021).

[9] “Linear regression,” *Wikipedia*. May 11, 2021. Accessed: May 16, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Linear_regression&oldid=1022686593

[10] “forelesn.kap5.pdf.” Accessed: May 16, 2021. [Online]. Available: https://www.mn.uio.no/ifi/english/services/knowledge/cs/forelesn.kap5.pdf

[11] “OLS.pdf.” Accessed: May 16, 2021. [Online]. Available: https://www.massey.ac.nz/~bjmoyle/mu/OLS.pdf

[12] “Coefficient-of-Determination.pdf | Coefficient Of Determination | Errors And Residuals,” *Scribd*. https://www.scribd.com/document/378919230/Coefficient-of-Determination-pdf (accessed May 16, 2021).

[13] “Coefficient of determination,” *Wikipedia*. May 03, 2021. Accessed: May 16, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Coefficient_of_determination&oldid=1021197177

[14] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” *Geosci. Model Dev.*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmd-7-1247-2014.

[15] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?– Arguments against avoiding RMSE in the literature,” *Geoscientific Model Development*, vol. 7, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmd-7-1247-2014.

[16] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-94463-0.

[17] "IBM Docs," Mar. 22, 2021. https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/spss-statistics/25.0.0?topic=regression-nonlinear (accessed May 16, 2021).

[18] M. Mijwil, A. Esen, and A. Alsaadi, "Overview of Neural Networks," vol. 1, p. 2, Apr. 2019.

[19] Y. E. Shao and S.-C. Lin, "Using a Time Delay Neural Network Approach to Diagnose the Out-of-Control Signals for a Multivariate Normal Process with Variance Shifts," *Mathematics*, vol. 7, no. 10, Art. no. 10, Oct. 2019, doi: 10.3390/math7100959.

[20] V. Cheung and K. Cannons, "An Introduction to Neural Networks," *Neural Networks*, p. 45.

[21] "Figure 2. Commonly used neural network activation functions (a) Binary...," *ResearchGate*. https://www.researchgate.net/figure/Commonly-used-neural-network-activation-functions-a-Binary-threshold-b-Bipolar_fig1_236268473 (accessed May 16, 2021).

Appendices A

Masters 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: Prof. Bernt Lie (main supervisor), co-supervisors: Madhusudhan Pandey, (PhD student) and Assoc. Prof. Thomas Øyvang

External partner: Skagerak Kraft (Ingunn Granstrøm)

Task background:

A thermal model of a synchronous generator was proposed in Øyvang (2018), with a reformulation studied in a group project in course FM1015 Modelling of Dynamic Systems, Lie (2018). The model from Lie (2018) was further studied in an MSc thesis in 2019 (Pandey *et al.*, 2019), in a subsequent summer job (Pandey, 2019), and in an MSc thesis in 2020 (Aleikish, 2020; Aleikish *et al.* 2020). The purpose of such a model is to allow for monitoring/control of the generator temperature, and thereby consider relaxed constraints on the power factor in the operation of generators.

Based on available experimental data from Skagerak, Pandey *et al.* (2019) studied model fitting and state estimation of the generator based on the model in Lie (2018). In general, the included heat exchanger sub model requires the numeric solution of a two-point boundary value problem for each time-step in the ODE solver; this is very time consuming. However, under some conditions (e.g., constant heat capacity), the heat-exchanger model can be solved analytically. In Pandey et al. (2019), constant heat capacity in the heat exchanger was assumed.

To prepare for a more general study involving model fitting with a temperature dependent heat-exchanger model, Aleikish (2020) studied the possibility to develop a hybrid heat exchanger model consisting of the analytic model with a regression/machine learning modification to capture the nonlinear effects of the temperature dependence. Such a fitting lead to excellent model fitting for the heat-exchanger model and a dramatic reduction in computation time compared to numerically solving the nonlinear two-point boundary value problem.

However, the initial study by Aleikish (2020) is based on known heat-exchanger parameters such as heat capacity coefficients, heat transfer coefficients, etc. To allow for tuning the physical parameters in the heat exchanger, it is necessary to further improve on the heat-exchanger model. One possibility available in the eco-system of the Julia programming language is to use "universal differential equations", which combines physics-based models (e.g., assuming an analytic, approximate heat-exchanger model) with neural networks.

References:

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

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*, pp. 190—197. Published by Linköping University Electronic Press, Series: Linköping Electronic Conference Proceedings No. 170, ISBN: 978-91-7929-897-5, eISSN: 1650-3740, doi: https://doi.org/10.3384/ecp20170190.
Pandey, Madhusudhan (2019). Unpublished results from summer job, University of SouthEastern Norway.
Aleikish, Khaled (2020). *Hybrid Machine Learning and Mechanistic Thermal Model of Synchronous Generator*. MSc thesis, University of South-Eastern Norway, 2020.
Aleikish, Khaled, Øyvang, Thomas, and Lie, Bernt (2020). "Hybrid model for fast solution of thermal synchronous generator with heat exchanger". *Proceedings of SIMS 2020*, September 2020. To be published in Linköping University Electronic Press.

Task description:

The following tasks are relevant:

1. Give a short overview of the given thermal model of a synchronous generator, with special focus on the ideal and nonideal 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. Identify (dimensionless?) parameters in the heat exchanger model as well as operating condition inputs. Carry out experiments (= simulations) on the non-ideal heat exchanger model to generate data that will subsequently be used for model fitting. In the experiments, make sure that you vary dimensionless parameters (e.g., by simulating several different fluids, i.e., with different combinations of heat capacities, heat transfer coefficient, etc.).
3. Develop a hybrid model of the counter current heat exchanger consisting of an analytic solution of the ideal model, as well as a data-driven/machine learning correction term. The correction term should include the effect of varying operating conditions as well as (dimensionless?) parameters in an extension of the work of Aleikish (2020). The hybrid extension can be based on linear or nonlinear regression (e.g., neural network).
4. Test the combined hybrid heat exchanger model with the thermal synchronous generator model, and assess the sensitivity of the solution wrt. uncertain model parameters.
5. Based on available experimental data from the operation of a synchronous generator, use your new model and tune parameters in the model to achieve improved model fit. Compare your results to those of Pandey et al. (2019).
6. Report the work in the Master's Thesis, and possibly in a suitable conference/journal paper, e.g., SIMS EUROSIM 2021.

Student category: The tasks can be solved by EPE, IIA, PT, EET students, but EPE students will be given priority. (The ideas under study are valid and useful within all scientific fields).

The task is suitable for online students (not present at the campus): Yes

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.

Supervision:

One-hour weekly supervision meetings are offered (on campus or via MS Teams), as well as feedback on partial reports every 3 weeks, and help with formulating a scientific paper. The last month of the work, the candidate is expected to work independently. In total, this surpasses the 15-20 hours of supervision that the candidate is entitled to.

Signatures:

Supervisor (date and signature):

Student (write clearly in all capitalized letters):

Student (date and signature):

Appendices B

Julia code used in this thesis are linked below:

[counter current]
[linear regression]
[nonlinear regression]
[Time comparison]