A Neural Network Approach to Thermal Gray-Box Modeling

Master's thesis in Complex Adaptive Systems

Nils Backlund

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

A Neural Network Approach to Thermal Gray-Box Modeling

Nils Backlund

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Electric Power Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

A Neural Network Approach to Thermal Gray-Box Modeling
Nils Backlund

Supervisors:
Alejandro Marquez Ruiz, ASML
Georgios Boutselis, ASML
Konstantinos Papangelou, ASML

Examiner:
Torbjörn Thiringer, Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Division of Electric Power Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

A Neural Network Approach to Thermal Gray-Box Modeling
Nils Backlund
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Accurate thermal models are important in precision systems where temperature variations can affect performance, stability, and control. This report explores a gray-box modeling approach for thermal systems, where a lumped thermal network is combined with neural-network parameterizations for selected difficult-to-model model components. The aim is to retain the simplicity and physical interpretability of lumped thermal models while using neural networks to learn complex and operating-dependent relations from data.

The proposed method uses a lumped thermal model, selects difficult-to-model and sensitive parameters using physical insight and sensitivity analysis, and represents these parameters using constant, linear, or neural-network functions. The approach is evaluated on a simulated thermal cooling system containing a Peltier element, heat pipe and heat exchangers with fluid-flow.

The results show that learning the heat-pipe parameters reduces the prediction error substantially, with the constant-parameter model resulting in 4.2 and 3.1 times larger mean errors compared to the linear and neural-network heat-pipe models, respectively. The linear and neural-network heat-pipe parameterizations perform similarly, differing by only 11 mK, suggesting that neural networks can be used even if simpler models are sufficient. Furthermore, replacing the analytical Peltier equation with a neural network gives similar in-distribution performance, but weaker generalization. When evaluated 40% outside the training range, the Peltier neural-network models increase in error by approximately 7 times, compared with about 3–4 times for the models that retain the analytical Peltier equation.

Overall, the results support a constrained gray-box approach where known thermal physics is preserved and data-driven functions are applied only to uncertain, sensitive, or difficult-to-model components. This provides a compact and interpretable model structure that is relevant for applications such as state estimation, fault detection, digital twins, and model predictive control.

Keywords: Thermal Neural Networks, Thermal Modeling, Lumped Mass Models, System Identification

Acknowledgements

I would like to thank my supervisor, Alejandro Marquez Ruiz, for his guidance, ideas, and continuous support throughout this project.

I would also like to thank Konstantinos Papangelou and Georgios Boutselis for their support at ASML.

Nils Backlund, Eindhoven, May 2026

Use of Generative AI

Generative AI has been used throughout the project, with the primary tool being Microsoft M365 Copilot Chat (OpenAI GPT models). The tools were used for code assistance, summarizing literature, improving language, spelling control, and formulation suggestions.

All AI-generated content has been critically reviewed, verified, and in most cases rewritten by the author. The author takes full responsibility for the final content, results and conclusions. No results, analyses or claims are based solely on AI-generated output without validation against primary sources.

Contents

1 Introduction **1**

2 Thermal Modeling **3**
2.1 Heat Transfer 3
2.2 Thermal Lumped Mass Models 4
2.2.1 Network Formulation 4
2.2.2 Thermal Resistance Estimation 4
2.2.3 External Heat Inputs and Outputs 5
2.3 State Space Formulation 6
2.4 Specific Heat-Transfer Components 6
2.4.1 Heat Pipes 6
2.4.2 Peltier Effect Cooler 7
2.4.3 Fluid Flow Heat Exchange 7
2.5 Relation to Finite Element Modeling 7

3 Data-Driven Modeling Approaches **9**
3.1 Data for Dynamic Modeling 9
3.2 Parameter Sensitivity and Identifiability 11
3.2.1 Local Sensitivity Analysis 11
3.2.2 Parameter Identifiability 11
3.2.3 Data Distribution and Generalization 12
3.3 Classical System Identification 12
3.4 Physics Guided Neural Models 12
3.4.1 Feedforward Neural Networks 12
3.4.2 Recurrent Neural Networks 13
3.4.3 Trainable State-Space Models 14
3.4.4 Physics-Informed Neural Networks 14
3.4.5 Thermal Neural Networks 14

4 Constrained Thermal Neural Networks **17**
4.1 Nominal Lumped Thermal Model 17
4.2 Experiment Design and Data Collection 18
4.3 Sensitivity Based Parameter Selection 18
4.4 Parameter selection 19
4.5 Parameterization of Unknown Parameters 19

4.6 Evaluation Procedure 20
4.7 Application to Simple Thermal System 20

5 Case-Study System **23**
5.1 Physical System Description 23
5.2 Lumped Model Construction 24
5.3 Generated Dataset 24
5.4 Selected Learned Parameters 25
5.5 Model Variants and Experiment Design 25
5.6 Results 26
5.6.1 Data Excitation 26
5.6.2 Sensitivity-Based Parameter Selection 26
5.6.3 Model Performance 27
5.6.4 Out-of-Distribution Model Performance 28

6 Discussion **31**
6.1 Parameter Selection and Identifiability 31
6.2 Model Performance and Generalization 32
6.2.1 Learning the Heat-Pipe Parameters 32
6.2.2 Replacing Known Physical Relations 32
6.2.3 Out of Distribution Performance 33
6.2.4 Parameter Compensation 33
6.3 Interpretability and Practical Use 34
6.4 Applications 35
6.4.1 Fault Detection 36
6.4.2 Digital Twin 36
6.4.3 Model Predictive Control 37
6.5 Ethics and Sustainability 38
6.5.1 Sustainability Benefits 38
6.5.2 Ethical Risks and Responsible Use 39
6.6 Limitations 39
6.7 Recommendations and Future Work 40

7 Conclusion **41**

Bibliography **43**

1

Introduction

Thermal systems are common in engineering applications, and accurate thermal models are important in complex precision machines, where temperature variation can impact performance, stability and control. These systems can be modeled by, for example, the lumped element method. Such models can range from simple with limited accuracy, to complex models with difficult parameter estimations.

Thermal systems are important in precision systems such as ASML lithography machines, where thermal variations can affect imaging, and good understanding of the thermal systems are therefore necessary. The ASML lithography machines are governed by the critical dimension (CD), defined as

$$CD = k_1 \frac{\lambda}{\mathrm{NA}} \tag{1.1}$$

where λ is the wavelength, NA is the numerical aperture and k_1 is a process-dependent coefficient. As lithography systems become more advanced, understanding thermal effects becomes more important. Exposure light and other heat loads can lead to temperature gradients, which cause structural deformation and optical distortion. Accurate and compact thermal models are therefore useful for understanding, estimating, and controlling temperature-dependent behavior in such systems [36].

The goal of this master's thesis project is to develop simple, interpretable, and accurate thermal models by combining traditional thermal modeling and neural networks to create a gray-box model. This will be done by selecting difficult-to-model parameters in lumped thermal network to be parameterized by neural networks. More accurate and interpretable thermal models can strengthen fault detection, digital twins for capturing internal states, and improved control systems with model predictive control (MPC).

2

Thermal Modeling

A thermal system can be defined as a bounded region of matter whose temperature evolution is of interest [13]. This chapter introduces the heat-transfer mechanisms governing energy exchange in such systems, followed by the lumped-mass modeling framework used in this work. The lumped formulation represents the system as thermal nodes connected by resistive heat-transfer paths, which can then be written as a state-space model. Finally, the relation to finite element modeling is discussed as a higher-resolution discretization method and as background for estimating effective thermal parameters.

2.1 Heat Transfer

There are four main heat transfer mechanisms: conduction, convection, radiation, and advection. Conduction describes heat transfer through a solid or stationary medium due to a temperature gradient, convection describes heat transfer between a surface and a moving fluid, and radiation the transfer of energy through electromagnetic emission. Finally, advection transports heat with the bulk motion of a fluid or gas.

The heat flow for conduction and convection can be calculated as

$$\begin{aligned} Q_{cond} &= \frac{kA_{cond}\Delta T}{d} \\ Q_{conv} &= hA_{conv}\Delta T. \end{aligned} \tag{2.1}$$

Here k is the thermal conductivity, h is the convective heat-transfer coefficient, d is the conduction length, and A_{cond} and A_{conv} are effective heat-transfer areas. The thermal resistance can be defined as $R = \Delta T/Q$ and would be $R_{convection} = 1/hA_{conv}$ and $R_{conduction} = d/kA_{cond}$ for convection and conduction respectively [20]. Furthermore, radiation from surface 1 to surface 2 can be written as

$$Q_{1\to 2} = \sigma A_1 F_{1\to 2}(T_1^4 - T_2^4) \tag{2.2}$$

for ideal surfaces assuming they are parallell. Around a nominal temperature T_{nom} this can be linearized using Taylor expansions to $(T_1^4 - T_2^4) \approx 4T_{nom}^3(T_1 - T_2)$ if the temperature difference $T_1 - T_2$ is small compared to the absolute temperatures involved [20]. For larger temperature differences, the effective radiative conductance depends on T_1 and T_2 and therefore varies with time.

2.2 Thermal Lumped Mass Models

Lumped mass models, or lumped element models, are commonly used to model systems in electronics, mechanics, audio and thermodynamics where a finite number of discrete nodes are identified. In thermal systems, each node is assumed to have a uniform temperature, and to exchange heat between nodes through thermal resistances or conductances. This results in a system of ordinary differential equations, with each equation describing the temperature change of a thermal node [37].

2.2.1 Network Formulation

A thermal node represents a part of the system whose internal temperature gradients are neglected. Node i stores thermal energy through its heat capacity $C_{p,i} = m_i c_{p,i}$ and exchanges heat with neighboring nodes, the environment, and external heat sources.

Applying conservation of energy for element i connected to n nodes gives

$$C_{p,i}\frac{dT_i}{dt} = \sum_{j=1,i\neq j}^{n} \frac{T_j - T_i}{R_{ij}} + \frac{T_a - T_i}{R_{ia}} + Q_i \tag{2.3}$$

where T_i is the node temperature, R_{ij} is the thermal resistance between nodes i and j, R_{ia} is the resistance between node i and the ambient temperature T_a, and Q_i is the external heat load to node i, including impacts from unmeasured inputs.

The first term in (2.3) describes heat flow between neighboring nodes. The second term is convection to the environment, the heat escaping to ambient, which also can be written as $h_i A_i (T_a - T_i)$ with $R_{ia} = 1/(h_i A_i)$ when convection dominates.

For all nodes, the system can be written in vector form as

$$C_{th}\dot{T} = GT + g_a T_a + Q \tag{2.4}$$

where $C_{th} = \text{diag}(C_1, \ldots, C_n)$ is a diagonal matrix in $\mathbb{R}^{n\times n}$, $T = [T_1, \ldots, T_n]$, $g_a = [1/R_{1a}, \ldots, 1/R_n a]$, and $Q = [Q_1, \ldots, Q_n]$ are all vectors in $\mathbb{R}^n$. Furthermore $G \in \mathbb{R}^{n\times n}$ is the conduction matrix where the off-diagonal elements are $1/R_{ij} \quad i \neq j$ and the diagonal of row i is $-\sum_{j=1}^{n} \frac{1}{R_{ij}} - \frac{1}{R_{ia}}$. The ambient temperature could also be treated as a state rather than an input to include it in the G and T matrix.

2.2.2 Thermal Resistance Estimation

Thermal resistance between two elements can be estimated in many different ways depending on physical information. For simple geometries and well defined paths they can be calculated directly with (2.1) giving $R_{cond} = d/(kA)$ and $R_{conv} = 1/(hA)$. Here, k is the material thermal conductivity, while h is an effective convective heat-transfer coefficient. In practice, k is obtained from material data, while h is commonly estimated from empirical Nusselt-number correlations based on geometry, flow regime, and fluid properties [20]. These estimates depend on operating conditions and are therefore interpreted as approximate effective values rather than exact constants [20].

However, this is not always the case in lumped mass models where the effective resistance between two elements may be represented by several combined heat transfer mechanisms. For example, two bodies that are not directly connected will transfer heat through a combination of different heat transfer mechanisms [8]. This creates a more complex model and can be dependent on materials, contact conditions, geometry and other unknown parameters [10]. When these are difficult to estimate analytically, numerical approaches can be a method forward. Simulating the thermal system using tools and techniques like finite element method (FEM) can help calculate the thermal resistance by estimating the temperature difference per power input [33].

Thermal resistances may also vary with other external parameters or temperature which can make them nonlinear. Convection can change with flow conditions, temperatures and material properties, and radiation is nonlinear in temperature [27]. Constant resistance networks can therefore lose physical meaning when these variations exist, especially in transient operations. In general, every thermal resistance can be modeled as

$$R_{ij} = f_{R_{ij}}(T, T_a, u) \tag{2.5}$$

where u contains other external inputs.

2.2.3 External Heat Inputs and Outputs

In addition to heat transfer between two nodes, there can also be additional heat flows Q_i into or out of an element. These heat flows can describe internal heat generation, active heating or cooling, or heat exchange with elements that are not explicitly modeled as nodes. For example, a flowing fluid in a pipe may remove or supply heat to a solid component without being represented as its own thermal state. In that case, the heat exchange can be included as an external heat-flow term acting on the connected node [27, 20].

The heat flow Q_i, or the parameters used to calculate it, can depend on other physical properties. Both cases are represented here by η_i, written as

$$\eta_i = f_{\eta_i}(T, T_a, u). \tag{2.6}$$

where η_i denotes a general heat-flow related quantity.

In general we can write (2.4) as

$$C_{th}\dot{T} = G(\theta)T + g_a(\theta)T_a + Q(\theta) \tag{2.7}$$

with

$$\theta = [R_{12}, \ldots, R_{ij}, \ldots, R_{n-1,n},\ \eta_1, \ldots, \eta_i, \ldots, \eta_n]. \tag{2.8}$$

From here on θ_i refers to the i-th element of θ and can be considered a function of physical properties.

2.3 State Space Formulation

The lumped mass model can be written in state space form by using the node temperatures as states. The representation for the thermal network from (2.4) could be rewritten as

$$\begin{aligned} C_{th}\dot{T}(t) &= GT(t) + B_u u(t) \\ y(t) &= C_y T(t) \end{aligned} \tag{2.9}$$

with the $T(t)$ as the state vector, $y(t)$ is the measured output, and C_y maps states to the measured temperatures. $u(t)$ would be the external inputs such as external heat loads, other input variables or ambient temperature if not viewed as an element, and $B_u = [g_a, I]$ represents its input matrix.

Since C_{th} is invertible, (2.9) also be rewritten in the standard state-space form

$$\begin{aligned} \dot{T}(t) &= AT(t) + Bu(t) \\ y(t) &= C_y T(t) \end{aligned} \tag{2.10}$$

where $A = C_{th}^{-1}G$ and $B = C_{th}^{-1}B_u$. For a constant resistance thermal network, A and B are constant. If they vary based on operating conditions and physical properties, we have a linear parameter-varying system (LPV)

$$\begin{aligned} \dot{T}(t) &= A(\theta)T(t) + B(\theta)u(t) \\ y(t) &= C_y T(t). \end{aligned} \tag{2.11}$$

2.4 Specific Heat-Transfer Components

The case-study system in this report contains components with heat-transfer behavior that is not fully captured by generic thermal resistances. This section introduces the heat pipe, Peltier-effect cooler, and fluid-flow heat exchange, focusing on which relations are modeled explicitly and which are treated as uncertain effective parameters.

2.4.1 Heat Pipes

Heat pipes are known for having exceptional heat transfer efficiency and are widely used to dissipate high heat fluxes, for example in electronic systems [25]. A heat pipe is a sealed device containing a two-phase working fluid that transports heat through evaporation, vapor flow, and condensation. Depending on the configuration, liquid return to the evaporator can be driven by gravity, rotational effects, or more commonly by capillary forces within a wick structure [25].

Modeling of heat pipes has traditionally relied on thermal resistance or conductance-based models. While computationally efficient, these approaches have limited accuracy and do not fully capture the complex two-phase flow behavior and phase-change processes of the working fluid. More advanced numerical and data-driven approaches, such as CFD-based simulations and artificial neural networks, have shown improved predictive capability, but the underlying physics and fluid dynamics involved are not always represented [25]. The heat pipe can therefore be treated as an uncertain resistance.

2.4.2 Peltier Effect Cooler

The Peltier-effect cooler is a thermoelectric cooling device using p- and n-type materials to induce a heat flow between its two sides when electric current is applied. The standard formulation consists of a Seebeck/Peltier contribution, a Joule-heating contribution, and a conductive contribution between the hot and cold sides [16]. As discussed previously, the conductive contribution can be implemented as a thermal resistance between the two elements. The thermoelectric element can then be written as

$$Q_c = -f_S T_c I + \frac{1}{2} f_{Re} I^2, \tag{2.12}$$

for the cold side, and

$$Q_h = f_S T_h I + \frac{1}{2} f_{Re} I^2, \tag{2.13}$$

for the hot side. Here, positive Q denotes heat added to the corresponding thermal node, f_S is the effective Seebeck/Peltier coefficient, f_{Re} is the effective electrical-resistance coefficient, and I is the applied current [16]. The conductive heat transfer between the two sides is then captured by modeling the Peltier element as two thermal nodes connected by an internal thermal resistance.

2.4.3 Fluid Flow Heat Exchange

A common heat-loss mechanism is the transfer of heat from a pipe containing a flowing fluid running through it, which is common in both heat exchangers and pipes. This process can be summarized in three steps: convection from the fluid to the wall, conduction through the wall, and convection from the wall to the surrounding fluid. All these effects can be combined into an effective conductance UA_{eff} [35].

For a fluid with constant mass flow rate flowing through a tube with approximately uniform wall temperature T_{node}, the heat transferred to or from the fluid is obtained from the enthalpy change of the stream. It is then possible to calculate the segment relation $T_{\text{out}} = aT_{\text{in}} + (1-a)T_{\text{node}}$, with $a = \exp\left(-\frac{UA_{\text{eff}}}{\dot{m}c_p}\right)$ [35]. The corresponding heat exchanged by the fluid is then computed from the enthalpy change as

$$Q_{\text{fluid}} = \dot{m}c_p \left(T_{\text{out}} - T_{\text{in}}\right). \tag{2.14}$$

2.5 Relation to Finite Element Modeling

Finite element modeling is a numerical method for solving heat-transfer problems by dividing a continuous physical domain into smaller elements. In thermal analysis, the temperature field is approximated over these elements, and the heat-transfer problem is described in equations for steady-state problems or ordinary differential equations for transient problems [26]. This makes FEM suitable for resolving spatial temperature gradients in complex geometries and boundary conditions.

Both FEM and lumped-mass models discretize a continuous thermal system into a finite number of regions. The main difference is the level of detail. A lumped mass model assumes uniform temperature in each of n identified elements, and as n increases, the model moves towards a more finely discretized, FEM-like representation of the thermal system. However, this also increases the number of states and constrained parameters, which can make gray-box modeling impractical for large n.

3

Data-Driven Modeling Approaches

This chapter introduces data-driven modeling concepts relevant for dynamic thermal systems. The focus is on how measured data can be used, how informative data affects parameter estimation, and how sensitivity and identifiability help determine which parameters in a lumped thermal model can be learned from data. Finally, classical system-identification and neural-network-based modeling approaches are introduced to later be compared to the approach discussed in the next chapter.

3.1 Data for Dynamic Modeling

Data-driven modeling uses measured input and output data to estimate or improve dynamic models. For thermal systems, this is useful when parts of the system are difficult to model from first principles. However, the usefulness of a data-driven model depends strongly on the information content of the data, the operating conditions covered during training, and the degree to which the chosen model structure preserves relevant physical behavior [22].

Open and closed loop systems are two configurations to generate data. An open loop configuration is controlled by adjusting the inputs without using measurements of the output, which makes it more sensitive to disturbances and modeling errors [4, 5]. In closed loop operation, the measured output is compared to a setpoint and is fed back to a controler that regulates the system input as shown in Figure 3.1 [4, 5].

Closed loop systems are common in industrial settings to regulate an output within specified limits. This creates input data whose variation depends on disturbances and setpoints [4, 14]. Open-loop systems are simpler and are often used in testing or system-identification experiments, where designed excitation signals such as step, random-step, PRBS, or sinusoidal inputs can be applied to the system [5, 1].

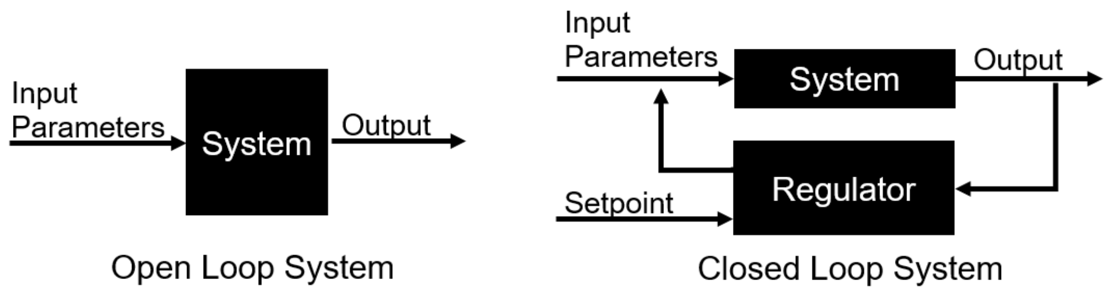

Figure 3.1: Schematic of open and closed looped systems.

The quality of a dataset does not only depend on its length, but also on how well the input excites the system dynamics. Steady-state data can be useful for estimating nominal operating points, but it contains little transient data, so identifying dynamic thermal behavior is therefore difficult.

Experiment design is important in system identification because the input should make the relevant dynamics observable in the measured outputs [7]. This is related to the persistence of excitation, where the input signal must contain enough independent temporal variation to produce different dynamic effects. For discrete-time systems, persistence of excitation can be analyzed through rank conditions on Hankel matrices constructed from the input data [2]. A step input is only persistently exciting of low order, while pseudo-random inputs can be persistently exciting of higher order [11]. Pseudo-random step data is therefore used in this work because it gives repeated transient responses over a wider operating range than steady-state data or a single step test.

Other approaches also exist for experiment design. [22] formulates a design of experiments (DoE) for nonlinear system identification using set membership methods, where the uncertainty of the identified model is used to locate informative regions of the regressor space. The input sequence must be designed to move the system toward these uncertain regions and collect informative measurements in dynamic systems. Their method aims to reduce the radius of information, corresponding to the worst-case model error [22].

It is also possible to reduce output correlation by designing the input signals in specific ways. Häggblom highlights that uncorrelated inputs may still produce correlated outputs in MIMO system identification, which is unfavorable for identifiability [17]. The method starts from a persistently exciting input signal and transforms it into an input sequence that aims to make the sampled output covariance matrix diagonal. In this way, the excitation not only excites the system sufficiently, but also reduces correlation between the measured outputs and thereby reduces directionality in the data.

3.2 Parameter Sensitivity and Identifiability

Before selecting parameters to be estimated from data, assessing whether they influence the measured outputs and whether their effects can be distinguished from other parameters, is useful. Sensitivity analysis gives a local measure of parameter influence, while identifiability describes whether the available data is sufficient to estimate parameters uniquely or reliably.

An optimization can also be made to identify a baseline for difficult-to-model parameters in θ based on the collected data series $y_{meas}(t)$. This can be done by solving the optimization

$$\theta_{ref} = \arg\min_{\theta} \|y_{meas}(t) - y(t;\theta)\|_2 \tag{3.1}$$

where well known parameters in θ are kept fixed. More informative data here can lead to a better estimation of θ_{ref}.

3.2.1 Local Sensitivity Analysis

For a local sensitivity write (2.4) as $\dot{T} = f(T, u, \theta)$, where θ are the model parameters. The local sensitivity of parameter θ_i can be defined as $S_i(t) = \partial T/\partial\theta_i$. Differentiating the state equation with respect to θ_i gives the sensitivity dynamics

$$\dot{S}_i = \frac{\partial f}{\partial T} S_i + \frac{\partial f}{\partial \theta_i}, \tag{3.2}$$

with $S_i(0) = 0$ if the initial condition does not depend on θ_i [34]. At steady state, let the state $T^\star$ satisfies $0 = f(T^\star, u, \theta)$, differentiating this state equation gives

$$0 = \frac{\partial f}{\partial T}\frac{\partial T^\star}{\partial \theta_i} + \frac{\partial f}{\partial \theta_i}, \tag{3.3}$$

and the local steady-state sensitivity for parameter θ_i becomes

$$\frac{\partial T^\star}{\partial \theta_i} = -\left(\frac{\partial f}{\partial T}\right)^{-1} \frac{\partial f}{\partial \theta_i}. \tag{3.4}$$

This sensitivity is used to quantify how strongly the state changes with respect to small variations in parameter θ_i, and can furthermore be used for parameter estimation and control [34].

3.2.2 Parameter Identifiability

Parameter identifiability describes whether model parameters can be estimated reliably from the available input-output data. Even physically meaningful parameters may be difficult to identify if they have little influence on the measured outputs, or if their effect can be reproduced by changes in other parameters [31]. Sensitivity analysis therefore gives a local indication of practical identifiability where low sensitivity makes a parameter difficult to observe.

3.2.3 Data Distribution and Generalization

A data-driven model is only directly supported by the operating conditions represented in the training data. If the validation data covers similar input amplitudes, temperature ranges, and operating regimes, the evaluation mainly tests interpolation within the collected data. If the validation data instead contains conditions that were poorly represented during training, the evaluation becomes a extrapolation problem, where the test distribution differs from the training distribution [24]. This is especially relevant for thermal systems where heat-transfer behavior can vary with temperature, flow conditions, and other operating-dependent effects [27]. This could as an example help understand if the learned model stays physically stable compared to data.

3.3 Classical System Identification

Classical system identification uses input-output data to estimate system dynamics by fitting a model that minimizes prediction error. This can include transfer-function models, time-series models, and nonlinear input-output models such as Volterra series. Transfer-function models describe the input-output relation in the frequency domain and are widely used in control applications [28]. Time-series models such as ARX and ARMAX describe the output using past input and output data, where ARMAX additionally includes a moving-average disturbance model [28]. Furthermore, Volterra series extend input-output modeling to nonlinear systems with memory, but the number of coefficients in the kernels increases rapidly with model order and memory length, which can make the model strongly parameter-dependent [12]. These methods are useful for prediction, but their parameters are not necessarily directly connected to physical quantities. They are therefore less suitable for this work, where physical interpretation is important. This motivates the gray-box approach used in this thesis.

3.4 Physics Guided Neural Models

Dynamic systems can be identified using methods with different levels of physical structure. Classical system-identification methods estimate parameters within a chosen model structure and can preserve physical interpretability when the structure is physically motivated. Black-box models learn input-output relations more directly and can take many different forms, but are usually harder to interpret physically.

3.4.1 Feedforward Neural Networks

A feedforward neural network is a function approximator built from layers of connected neurons. The input layer receives the variables used by the model, such as temperatures, inputs, or other measured quantities. These values are passed through one or more hidden layers, where each layer applies trainable weights, and biases, that can be optimized based on data, and a nonlinear activation function. The output layer then maps the final hidden representation to the predicted quantity, such as a heat flow, thermal resistance, or temperature correction.

For a neural network with a single hidden layer, this can be written as

$$\hat{y} = W_2\sigma(W_1 x + b_1) + b_2, \tag{3.5}$$

where x is the input vector, W_1 and W_2 are weight matrices, b_1 and b_2 are bias vectors, and $\sigma(\cdot)$ is the activation function. The activation function introduces nonlinearity, allowing the network to represent nonlinear relations that cannot be captured by a purely linear model [15].

This can be expanded to n layers creating a multilayer perceptron (MLP) by letting the output of the previous layer be the input for the next. This would give the recursive formula

$$h_i = \sigma(W_i h_{i-1} + b_i) \tag{3.6}$$

where $h_0 = x$ is the input vector, $h_1, \dots h_{n-1}$ are the states of hidden layer neurons and h_n is the output layer. A schematic of this can be seen in Figure 3.2. All parameters W_i and b_i, $i - 1, ..., n$ can be collected into ϕ.

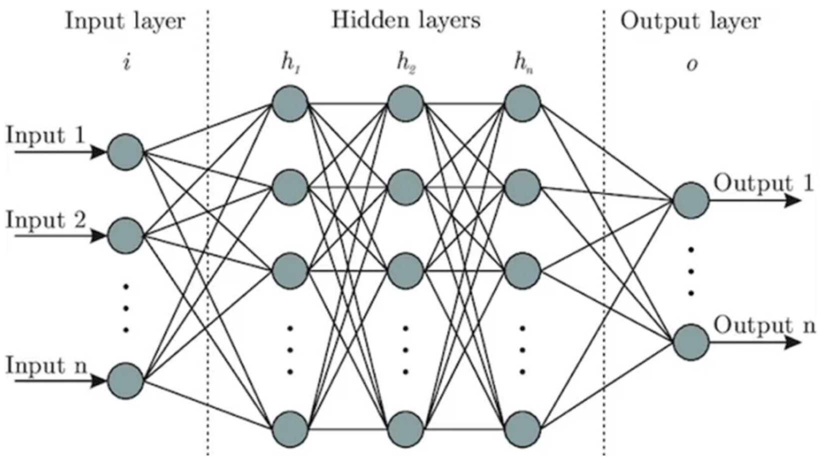

Figure 3.2: Schematic of a simple feed forward neural network.

3.4.2 Recurrent Neural Networks

Another common time series prediction technique is recurrent neural networks (RNN) or similar approaches like long short-term memory (LSTM) or gated recurrent unit (GRU). These networks can also update internal states, with LSTM and GRU adding additional gating mechanisms for these. A comparison of a single cell of each is shown in Figure 3.3, and these cells are chained together to predict series through time.

These models can learn input-output relations directly from data, but their hidden states are generally not tied to any physical quantities. This makes them less interpretable compared to both state space models and lumped networks. They generally require larger datasets to train on as well, compared to other methods presented in this report.

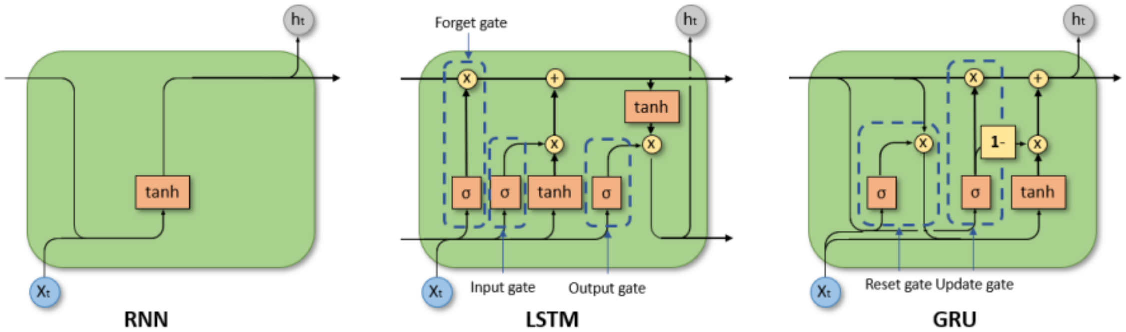

Figure 3.3: Schematic on the three common types of recurrent neural networks

3.4.3 Trainable State-Space Models

A common modeling technique in state space models is treating the matrices A, B and C from (2.10) as trainable parameters [18]. This gives a flexible dynamic model, but the learned states and matrices do not necessarily correspond to physical relations. This model has the same recursive structure as a recurrent model, where the next state depends on the previous state and current input, after discretazation [18].

3.4.4 Physics-Informed Neural Networks

Another approach is using neural networks to solve physics-based equations is by the use of physics informed neural networks (PINN) [32]. PINNs include knowledge of governing equations into the training objective by penalizing the PDE residual Ginven a neural approximation $v_\theta(t,x)$, the residual is defined as

$$r_\theta(t,x) = \partial_t v_\theta(t,x) + \mathcal{N}[v_\theta(t,x)]\,, \tag{3.7}$$

and the model is trained by minimizing both data and residual errors. PINNs can reduce reliance on labeled data, but knowledge of governing equation is required[32]. The learned solution remains an implicit function represented by a neural network, limiting direct physical interpretability.

3.4.5 Thermal Neural Networks

Thermal Neural Networks (TNN) combine lumped mass models with neural-network based parameter estimation [23]. The method keeps the thermal state space structure which increases interpretability of the temperature states, while neural networks estimate the conductances, power losses, and inverse thermal capacitances from temperatures and additional measured inputs [23]. The estimated values are then used in the thermal equations and the system is propagated through time, as shown in Figure 3.4. This makes the model closely related to trainable state-space models.

The TNN utilizes a single general neural network for all parameters in each of the conductances, power losses, and inverse thermal capacitances. This makes the method useful if they are unknown and difficult to model, but it does not make it possible to apply additional constraints to the system.

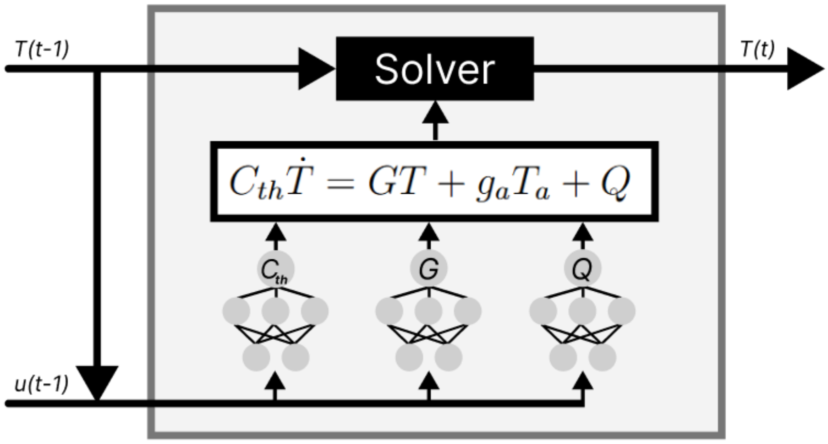

Figure 3.4: Thermal neural network flow for a single timestep. The previous state temperatures $T(t-1)$ and measured inputs $u(t-1)$ are passed through neural networks to predict C_{th}, G and Q. These are inserted into the thermal state equation which is solved to get the next temperature state $T(t)$.

4

Constrained Thermal Neural Networks

This chapter presents the general gray-box modeling method and aims to combine the classical modeling techniques with data-driven approaches to construct a gray-box model. It will utilize physical domain knowledge while applying data-driven functions to describe parameters that are nonlinear, or difficult to calculate analytically.

The proposed workflow consists of six main steps:

1. **Lumped thermal model creation:** A lumped thermal model is constructed from the physical system.
2. **Experiment design and data collection:** The data used for training the network should be relevant for the use case.
3. **Sensitivity analysis:** Parameters are ranked based on sensitivity, where the most sensitive parameters affect the output the most.
4. **Parameter selection:** Parameters that are likely to vary with operating conditions are selected based on the sensitivity analysis and physical insight.
5. **Parameter optimization:** The selected parameters are parameterized as optimizable neural networks.
6. **Model evaluation:** The resulting model is trained on transient input-output data and evaluated on separate data sets.

A simplified representation of the model architecture is shown in Figure 4.1.

4.1 Nominal Lumped Thermal Model

A lumped mass model is constructed and the selected system is split into thermal nodes with resistances connecting elements. Other known physical constraints are also added, like additional heat flows in the system. Parameters that can be calculated from known geometry and material properties are kept fixed, while difficult-to-model and complex heat transfer paths are treated as candidates for estimation. The parameters from the model are formulated into the matrices G, C_{th} and Q, which are included in θ, with the state temperatures T and mapping C_y to the measured temperatures y according to (2.9). The difficult-to-model parameters of θ are then calculated as explained in Section 2.2.2 to estimate θ_{ref}

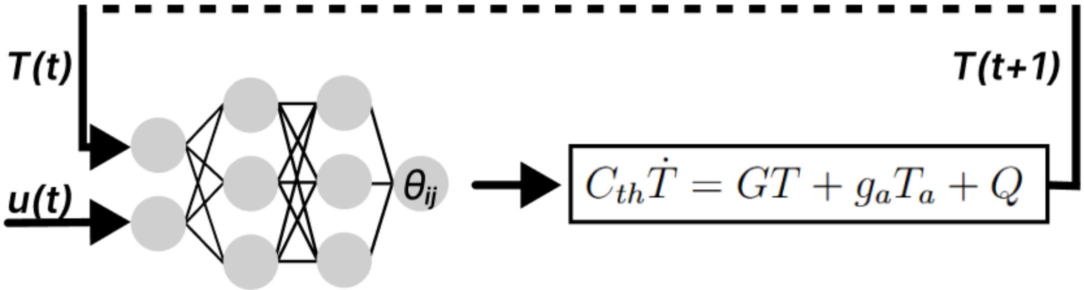

Figure 4.1: Schematic of the proposed model, specific parameters θ_{ij} of G and Q are parameterized by a neural network, where previous states T and other inputs u are inputs. The dashed line is meant to represent a recurrency with the output temperature being the input temperature for the next time step.

4.2 Experiment Design and Data Collection

Data collected from the system can be used for different purposes. Steady state data can be used to estimate parameter values for the sensitivity analysis, transient data to optimize the parameters in the model, and separate validation data to evaluate the model performance.

Since controlled industrial data often is concentrated near steady state operations, additional excitation could be needed to identify dynamical behavior. In this work, pseudo random step inputs are used to generate multiple transient responses over operating ranges. The excitation of each dataset is compared using the singular values of the Hankel matrix based on the input sequence, also called Hankel condition metric, where more evenly distributed singular values are considered to be more informative. Evaluation of the final model performance made using separate validation data sets.

4.3 Sensitivity Based Parameter Selection

The sensitivity analysis is done in the following steps:

1. Estimate he steady state internal temperatures $T^\star$ based on the initial parameter vector θ_{ref}.
2. Perturb one candidate parameter at a time using a multiplicative factor α.
3. Recompute the steady-state output for each perturbation.
4. Use the change in temperature perturbation with α to calculate the sensitivity measure.

First, the steady state internal temperatures $T^\star$ and the nominal parameter vector $\theta^\star$ is estimated by solving

$$T^\star = \arg\min_T \|f(T, \theta_{ref}, u)\|_2^2 \tag{4.1}$$

for a constant input vector u.

Each parameter of $\theta^\star$ is perturbed separately. For parameter i, the perturbed parameter can be written as

$$\theta^{(i)}(\alpha) = [\theta_1^\star, \dots, \alpha\theta_i^\star, \dots, \theta_n^\star]^T, \qquad \alpha \in [1-\delta, 1+\delta]. \tag{4.2}$$

For each value of α, the perturbed steady-state temperature is calculated from

$$T^{(i)}(\alpha) = \arg\min_T ||f(T, \theta^{(i)}(\alpha), u)||_2^2 \tag{4.3}$$

and the output temperature is computed as

$$y^{(i)}(\alpha) = C_y T^{(i)}(\alpha). \tag{4.4}$$

If $y^{(i)}(\alpha)$ contains multiple temperatures, the sensitivity can be computed for each output separately or summarized using the average sensitivity.

The sensitivity of parameter i is then approximated using (3.4) with θ_i replaced by $\alpha\theta_i^\star$. This gives the per parameter sensitivity measure

$$S_i \approx \left| \frac{\Delta y^{(i)}(\alpha)}{\Delta\alpha} \right| \tag{4.5}$$

In practice, S_i is estimated as the absolute slope from a linear regression of $y^{(i)}(\alpha)$ with respect to α. The model parameters can thereafter be ranked depending on their sensitivity.

4.4 Parameter selection

The sensitivity analysis, combined with physical insight, is used to indicate which parameters should be kept constant and which should be parameterized. Parameters with little influence on the measured temperatures are kept constant, since estimating them is unlikely to significantly improve the output. Parameters with high sensitivity are further assessed using domain knowledge to determine whether they are expected to vary with states or external inputs. These parameters are selected for parameterization by neural networks or other functions.

4.5 Parameterization of Unknown Parameters

The selected parameters are represented as learnable parameters, as a predefined function, or as a neural network function, where they can be a function of states and other inputs. For a neural network MLP parameterization, parameter i at timestep k is written as

$$\theta_{i,k} = f_{\mathrm{NN},i}(T_k, u_k \mid \phi_i) \tag{4.6}$$

where $\theta_{i,k}$ denotes the value of parameter i at timestep k, T_k is the state vector and u_k is the complete input vector. The vector ϕ_i are the weight and bias terms for each layer in the MLP of neural network function $f_{\mathrm{NN},i}$ where each layer follows from (3.6).

The model parameters are trained in discrete time by rolling out the model over finite windows. For each timestep, the current parameter vector θ_k is used in Euler discretization of state-space model (2.10)

$$\begin{aligned} \dot{T}_k &= A_k(\theta_k)T_k + B_k(\theta_k)u_k, \\ T_{k+1} &= T_k + \Delta t \dot{T}_k, \\ \hat{y}_k &= C_y T_k, \end{aligned} \tag{4.7}$$

where Δt is the timestep between time k and $k+1$.

The trainable quantities are optimized by minimizing the prediction error over the rollout window,

$$\Theta^\star = \arg\min_{\Theta} \sum_{k=1}^{N} \|y_k - \hat{y}_k\|_2^2 + \lambda \max_k \|y_k - \hat{y}_k\|_\infty + \|\theta - \theta_{ref}\|_\gamma . \tag{4.8}$$

Here, Θ contains all trainable quantities, including the neural-network weights ϕ_i for each parameterized model parameter. The scalar λ weights the maximum prediction-error penalty over the rollout window, while the matrix γ selects and weights which physical parameters are regularized toward their reference values. The vector θ_{ref} contains reference parameter values, for example values obtained from the nominal steady-state fit or from physical estimates.

During training, the model can either be corrected using the measured states/outputs or be rolled out recurrently over the full sequence. The initial state is calculated using the steady state solution, and the training is done using gradient descent based optimization.

4.6 Evaluation Procedure

The models are evaluated on a separate in-distribution validation set using the mean absolute error (MAE) and max absolute error,

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{k=1}^{N} ||y_k - \hat{y}_k||_1, \\ e_{\max} &= \max_k ||y_k - \hat{y}_k||_1. \end{aligned} \tag{4.9}$$

The main comparison between models will be performed using a validation dataset with the same distribution as the original function, but additional out of distribution tests can also be made to understand the performance outside the training distribution.

4.7 Application to Simple Thermal System

A simple illustration of the method can be created on the small two body system with schematics shown in Figure 4.2 from TCLabs [3]. It uses two heaters providing a positive heat flow for its two elements, and it exchanges energy with the environment and each other. Its two governing equations are

$$
\begin{aligned}
\dot{T}_1 &= \frac{1}{C_1}\left(\frac{T_2 - T_1}{R_{12}} + \frac{T_a - T_1}{R_{1a}} + Q_1\right), \\
\dot{T}_2 &= \frac{1}{C_2}\left(\frac{T_1 - T_2}{R_{12}} + \frac{T_a - T_2}{R_{2a}} + Q_2\right)
\end{aligned}
\tag{4.10}
$$

which can be used for the system. From domain perspective, the resistance R12 is difficult to calculate since the two elements are next to each other with a space in between. The heater elements have input currents I_1 and I_2 as input and the mapping from current to heat flow is also unknown. If the output temperatures T_1 and T_2 are sensitive to these parameters, they can be considered good candidates to be formulated as

$$
\begin{aligned}
R_{12,k} &= f_{NN}(T_{1,k}, T_{2,k} \mid \phi_{R12}) \\
Q_{1,k} &= f_{NN}(I_{1,k}, T_{1,k} \mid \phi_{Q1}) \\
Q_{2,k} &= f_{NN}(I_{2,k}, T_{2,k} \mid \phi_{Q2})
\end{aligned}
\tag{4.11}
$$

where f_{NN} is a MLP as shown in Figure 3.2 consisting of 2 hidden layers with 5 neurons in each. This size is selected to produce an overparameterization since the constrained structure of the rest of the model prohibits severe overfitting. The parameters for these networks can then be optimized using (4.8) based on its collected data.

The data selected here is pseudo random steps changing I_1 and I_2, and the analysis of the optimized functions shows that its parameters predicted becomes constant for R_{12} and linear with I_1 and I_2 respectively. This performance is the same as original thermal neural network model optimized with constant parameters.

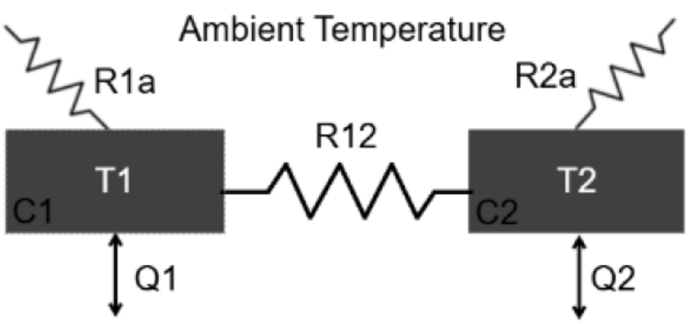

Figure 4.2: The arduino board from TCLabs with its two heaters and elements, and it formulated as a simple two body system in a lumped mass format, with thermal resistances between each other and to the environment.

5

Case-Study System

The presented method is applied to a simulated thermal cooling system. This section describes the physical system, the generated dataset and the model variants used for evaluation

5.1 Physical System Description

The method is applied to a thermal cooling system representative of industrial cooling applications. Thermoelectric cooling systems using Peltier elements are common in both research and industry. Connecting Peltier elements to heat pipes is a common technique to mitigate Peltier limitations, like limiting reverse heat flux, and heat pipes also provide faster heat transfer between elements which is useful for cooling [6, 30]. The system used in this report takes heavy inspiration from such configurations, with slight modifications to make it suitable for evaluating the proposed gray-box modeling method.

The system consists of a Peltier element used as a thermoelectric cooling device, with its hot side connected to a heat sink. The cold side is connected directly to a heat exchanger, which cools a fluid flowing through it. This heat exchanger is also connected through a heat pipe to a second heat exchanger, which cools another fluid. The output temperature is considered to be the temperature of the second heat exchanger, and it is also known to be affected by an additional parameter u_1. The system is shown in Figure 5.1. It is used because it contains both well-described heat-transfer components and more difficult-to-model components with more complex thermal behavior.

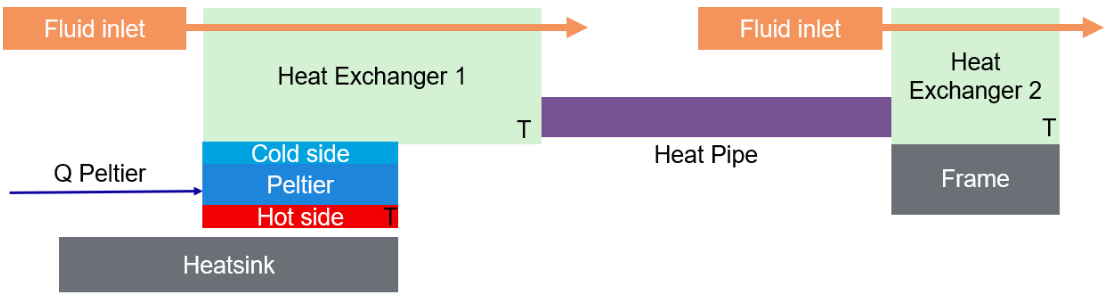

Figure 5.1: System used in case-study with relevant components

5.2 Lumped Model Construction

This system is converted into a lumped-mass model as shown in Figure 5.2, where the main components in colors, the fluid is an exception, and connections between components in black. Each main component i is considered to be a thermal capacitances C_i with internal temperature T_i. Heat-transfer paths between connected components i and j are modeled as thermal resistances R_{ij}. Each component is also affected by the ambient temperature T_a through a resistance $R_{i,a}$. Additionally, the input current to the Peltier element controls the heat flows Q_h and Q_c for the hot and cold sides, respectively, according to (2.12). Heat exchange between each heat exchanger and its respective fluid is modeled according to (2.14). External heat flows are considered for each node as Q_i. This can be formulated as (2.4) with

$$G = \begin{bmatrix} -\sum_{j\neq 1} \frac{1}{R_{1j}} & \frac{1}{R_{12}} & \cdots & \frac{1}{R_{17}} \\ \frac{1}{R_{12}} & -\sum_{j\neq 2} \frac{1}{R_{2j}} & \cdots & \frac{1}{R_{27}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{R_{17}} & \frac{1}{R_{27}} & \cdots & -\sum_{j\neq 7} \frac{1}{R_{7j}} \end{bmatrix} \tag{5.1}$$

which is a symmetric matrix with its structure determined by the thermal connections, and

$$C_{\text{th}} = \begin{bmatrix} C_1 \\ \vdots \\ C_7 \end{bmatrix}, \quad T = \begin{bmatrix} T_1 \\ \vdots \\ T_7 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 \\ \vdots \\ Q_7 \end{bmatrix}, \quad g_a T_a = \begin{bmatrix} \frac{1}{R_{1,a}} T_a \\ \vdots \\ \frac{1}{R_{7,a}} T_a \end{bmatrix}. \tag{5.2}$$

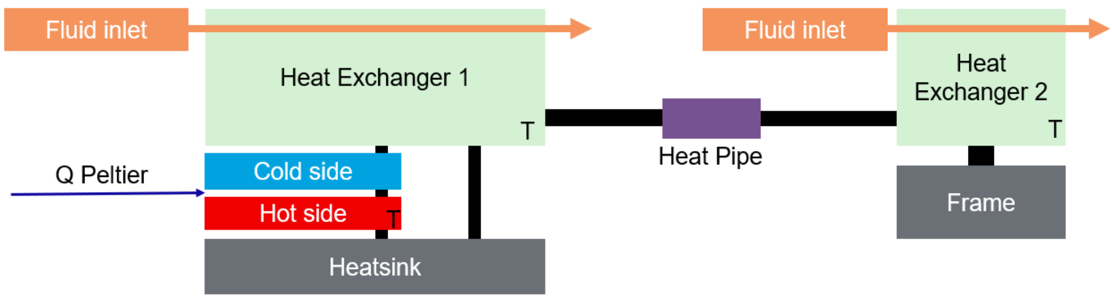

Figure 5.2: Lumped mass model of the case study. Black connections represent thermal resistances and colored blocks represent thermal capacitances.

5.3 Generated Dataset

The case study uses simulated data for a controlled evaluation setting. This makes it possible to define the underlying parameter relations while still testing whether the proposed method can recover useful parts of the system. To preserve the physical structure while introducing more complex behavior, elements from the system in

Figure 5.2 are selected to change with time. The Peltier will be formulated as (2.12) with the applied current as input, and the thermal resistances connected to the heat pipe will be linearly dependent on their connected node temperatures as well as the external input u_1.

The constructed system is simulated using pseudo-random step input sequences for the current I and u_1 bounded between selected values. This is done separately for the training and validation datasets, and the construction of the out of distribution dataset is made by sampling outside the bounds for I and u_1 up to a selected percentage. The resulting temperature state trajectories are saved as input-output data for parameter optimization and evaluation. The dataset is intentionally kept limited to reflect the limited amount of high quality testing data often available in real world systems.

5.4 Selected Learned Parameters

The learnable parameters are based on physical insight and the sensitivity analysis described in Section 4.3. Each difficult-to-model parameter is perturbed multiplicatively according to (4.2), using $\alpha \in [0.8, 1.2]$. Heat-transfer paths with well described geometry and simple analytical descriptions are kept fixed. The heat-pipe thermal resistances are difficult to model accurately using physical expressions and are therefore treated as components that are difficult to model. These resistances are parameterized as constant values, linear functions or neural netowrks.

Furthermore, the Peltier element heat flow is treated in two ways, either with the known analytical form (2.12) and its coefficients retrained, or as a neural network. This allows the study to compare strongly physics guided representation with a more flexible data-driven representation.

5.5 Model Variants and Experiment Design

The model variants can be divided into the two groups with different Peltier estimation approaches. In the first group, the heat pipe resistances changes between having constant, linear, and neural-network parameterizations. This tests whether the heat pipe behavior can be captured without replacing the known Peltier physics. The second group tests whether the data-driven model can recover the nonlinear Peltier heat flow with neural network. The heat pipe is kept constant in one case and parameterized as a neural network in the second. The second case utilizes the neural network function for parameterized Peltier from the first case to simplify training.

The experiments compare how the different parameterizations affect the prediction accuracy and generalization. Each model is trained on the same training dataset and evaluated on a separate in-distribution validation set. There will also be tests made with test sets out of distributionto explore how this affects the models. The model variants are summarized in Table 5.1.

Table 5.1: Overview of the evaluated model variants.

Model	Peltier heat flow	Heat-pipe resistance
M1	Analytical model, trainable coefficients	Constant
M2	Analytical model, trainable coefficients	Linear function
M3	Analytical model, trainable coefficients	Neural network
M4	Neural network	Constant
M5	Fixed neural-network Peltier model	Neural network

The neural networks have been chosen to be multilayer perceptrons with two hidden layers with 8 neurons each, with a sigmoid function as activation function. The inputs for the heat pipe parameterization is the temperature of each heat exchanger, the state temperature of the heat pipe, and the extra input u_1. Its outputs are the values of the resistances to ambient and heat exchanger 1 and 2.

The Peltier parametersiation has the temperature of its hot and cold side, along with the input current, as inputs, and the two heat loads as outputs. This gives the two models 139 and 122 prameters respectively. Alternative neural-network architectures were tested, but the selected small architecture provided similar performance while requiring fewer trainable parameters.

5.6 Results

This chapter presents the results of the case study following the workflow in the method. First the excitation quality of the generated data is evaluated. Second, a sensitivity analysis is used to help the selection of learnable parameters. Finally, the model variants are compared using in- and out-of-distribution test sets.

5.6.1 Data Excitation

The pseudo-random step data gives lower Hankel condition metric of compared to the step test and steady state data using the same window size. This is the dataset used for model training and validation in the following sections.

5.6.2 Sensitivity-Based Parameter Selection

The local steady-state sensitivity analysis is used to rank selected model parameters by their influence on the outlet temperature, here defined as the temperature at Heat Exchanger 2. Figure 5.3 shows the resulting sensitivity ranking. The largest sensitivities are found for Peltier-related parameters, including both the heat-flow terms and the coefficients in (2.12). Several resistances around the heat exchangers and heat pipe also appear among the most sensitive parameters.

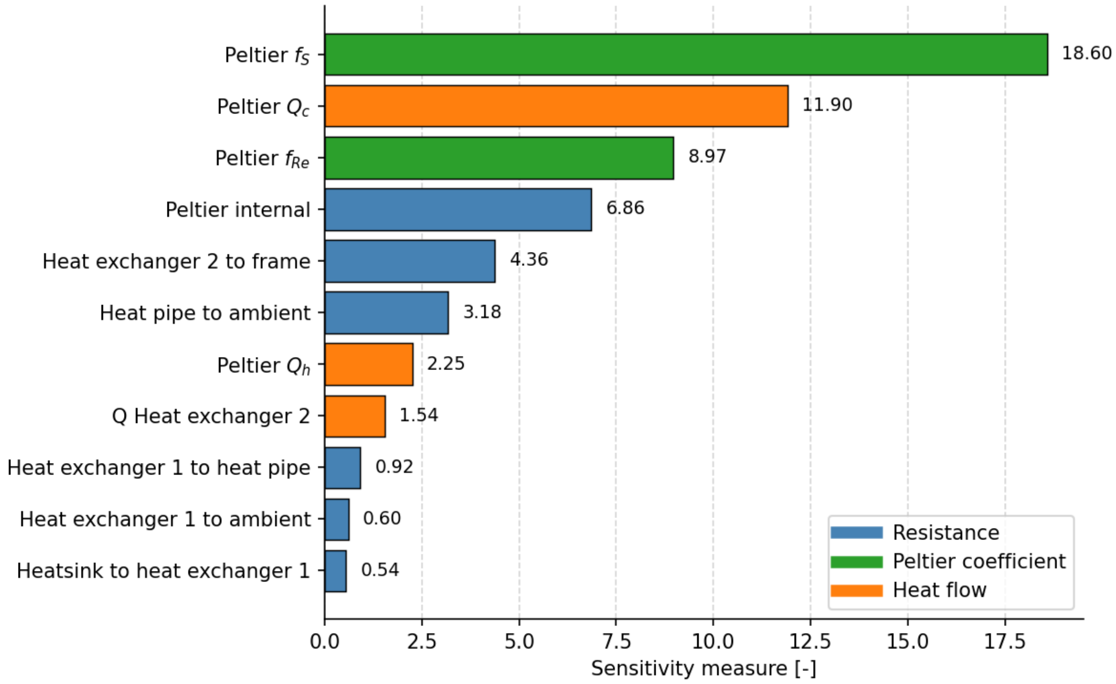

Figure 5.3: Local steady-state sensitivity ranking of selected thermal resistances and Peltier-related parameters with respect to the temperature at Heat Exchanger 2.

5.6.3 Model Performance

The models are first evaluated on an in-distribution validation sequence. Figure 5.4 shows the propagation of temperature states and errors through time for all model variants.

Model M1–M3 use analytical Peltier heat flow equation and differ only in the heat pipe resistance parameterization. As shown in Figure 5.5, both linear and neural network heat pipe parameterizations reduce the prediction error compared to the constant resistance model, with M2 having the smallest MAE.

Model M4 and M5 are also compared against the other models in Figure 5.5. The neural Peltier models show similar maximum error to the constant parameter baseline, but M5 gives a lower mean absolute error than both M4 and M1.

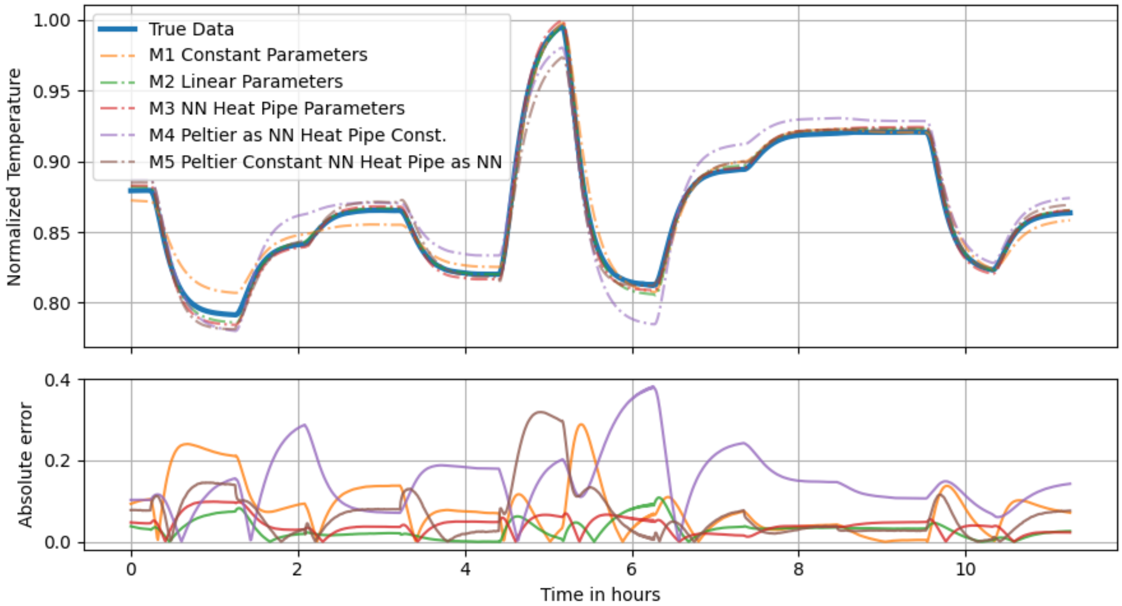

Figure 5.4: State propagation and error throughout time for each model

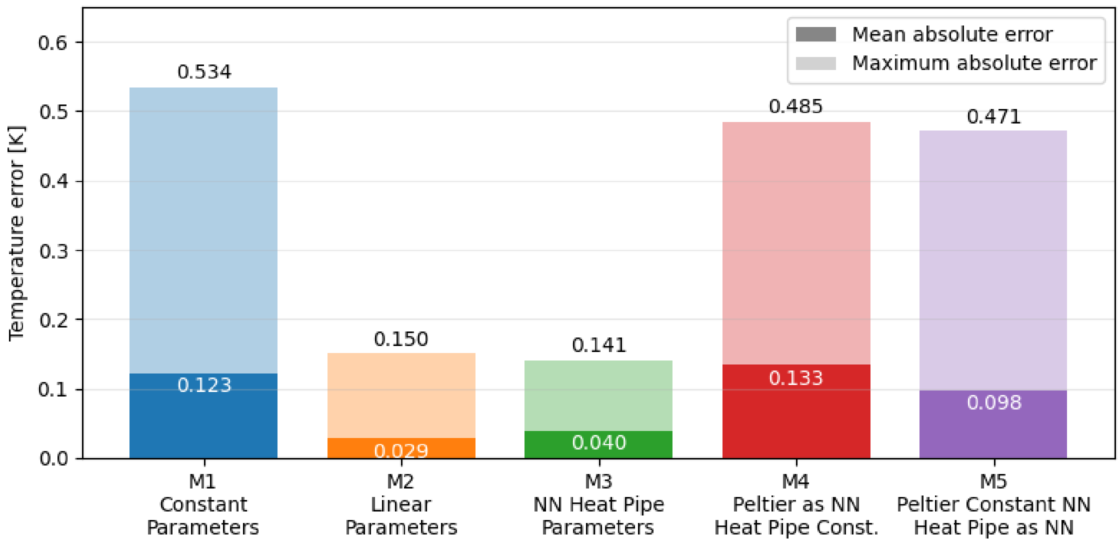

Figure 5.5: Prediction error for the physics-based Peltier models with constant, linear, and neural-network heat-pipe resistance models

5.6.4 Out-of-Distribution Model Performance

When the models are evaluated on input ranges outside the training distribution, all models show an increase of prediction error. The largest increase in error is observed for models using a neural network representation for the Peltier element, M4 and M5. Models using the analytical Peltier equation, M1–M3 show smaller error growth outside the training range.

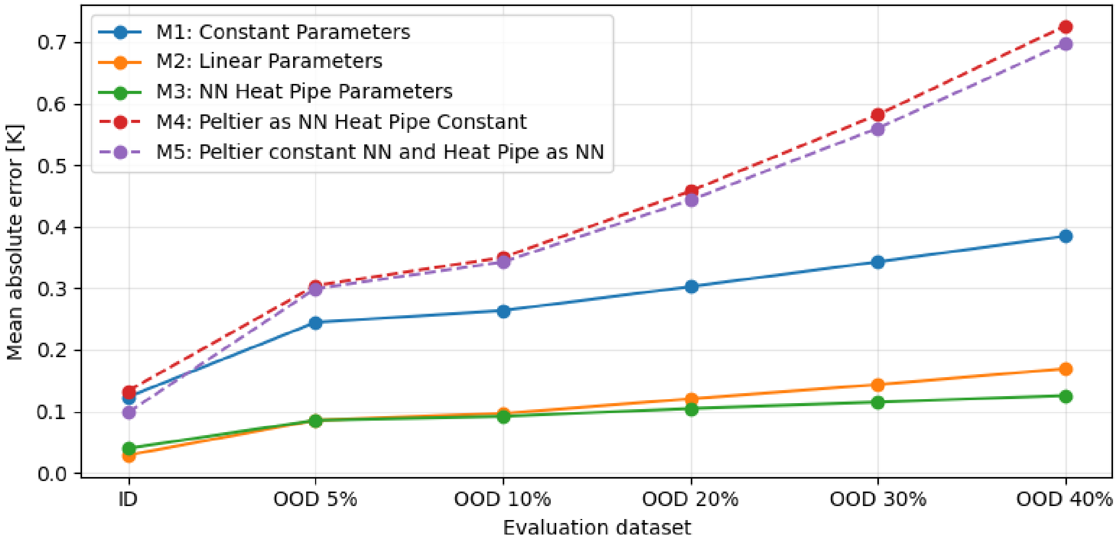

Figure 5.6: Mean absolute error for out of distribution data, averaged over 10 evaluation datasets. M1–M3 show a smaller increase in error compared to M4 and M5 as the distribution shifts away from the training distribution.

6

Discussion

This chapter discusses the results of the case study in relation to the proposed gray-box modeling approach. The discussion focuses on how parameter selection, model structure and training-data coverage affect prediction performance and generalization behavior. Finally, the practical usefulness, limitations, and possible extensions of the approach are discussed.

6.1 Parameter Selection and Identifiability

The method uses transient pseudo-random step data for training because adaptive parameters must be exposed through changes in the measured temperature response. As discussed in the method, steady-state data is mainly useful for estimating nominal parameter values, while transient data is needed to evaluate how inputs affect the dynamics. The data-excitation analysis in the results therefore supports the use of pseudo-random training and validation sequences, but should only be interpreted as an indication of input richness. Comparisons can also be made with data produced with uncorrelated outputs or by exciting difficult-to-model components more. This will be discussed further in future work.

The sensitivity analysis shows which parameters have the strongest influence on the outlet temperature, but the ranking should be combined with domain knowledge when selecting candidates for adaptive parameterization. A parameter can have high sensitivity, but if its physical relation is well known, it can be kept physics-based. Similarly, if an uncertain parameter has low sensitivity, there is little benefit in making it adaptive. Sensitivity should also not be interpreted as identifiability, since different parameters can affect the same measured output in similar ways. The analysis is local and steady-state, so nonlinear interactions, transient effects, and dependencies between parameters are not captured.

In the case study, both the Peltier heat-flow terms and the coefficients in (2.12) have a strong influence on the output. However, the Peltier heat-flow relation is well described by (2.12), so replacing the full relation with a neural network introduces additional degrees of freedom that may be unnecessary or nonphysical. For this reason, the analytical Peltier relation is retained in some experiments and compared with experiments where it is replaced by a neural network.

In contrast, the thermal resistances around the heat pipe are difficult to describe accurately from geometry and material properties alone. These resistances are therefore treated as uncertain components and parameterized using neural networks with relevant inputs. The remaining sensitive components have sufficiently well described geometry and are therefore kept constant or physics-based.

6.2 Model Performance and Generalization

The in-distribution results show how well each parameterization fits data from the same operating range as the training data. The out-of-distribution results indicate how the learned relations depend on the training range, and whether the model remains meaningful outside it. The discussion points are summarized in Table 6.1.

6.2.1 Learning the Heat-Pipe Parameters

Model M1 gives 4.2 and 3.1 times larger prediction errors than M2 and M3, respectively. This is likely because M1 uses constant heat-pipe resistances, which cannot represent the generated linear heat-pipe relation. In contrast, M2 and M3 use adaptive parameterizations, allowing them to fit this relation and achieve much lower errors.

Model M2 and M3 also have relatively similar performance, with M3 having only 11 mK larger error than M2. This suggests that the generated heat-pipe relation is sufficiently simple to be captured by a linear parameterization, which is expected since it is a linear relationship. From a modeling perspective, this is useful because linear models have fewer parameters, are less likely to overfit, and are easier to implement and train. However, their similar performance also shows that neural networks can estimate simple relations. A more nonlinear heat-pipe relation would likely increase the advantage of the neural-network model.

Furthermore, the learned parameter functions should not be expected to recover the exact functions used to generate the data. The optimization problem is non-convex, the dataset is limited and several parameter combinations can produce similar output trajectories. Even though low prediction error might be achieved, it does not guarantee parameter values to be physically meaningful, which is explored further in Section 6.2.4.

6.2.2 Replacing Known Physical Relations

Keeping the heat pipe resistances constant, replacing the analytical Peltier equation with a neural network gives only a 10 mK larger mean absolute error, while reducing the maximum error. This suggests that the neural network can approximate the Peltier relations within the training distribution.

When a neural-network heat pipe parameterization is added in addition to the Peltier neural-network parameterization, M5 improves by 26 % compared with M4. However, M5 still performs 2.5 times worse than M3, where the analytical Peltier equation is retained. A possible reason is that the neural-network Peltier parameterization introduces more degrees of freedom and is less constrained than the analytical equation. This can make the optimization problem harder and allow parameter combinations that fit the training distribution without representing physical relations.

Another possible explanation is that the neural-network Peltier model may partly compensate for effects that are later assigned to the heat-pipe parameterization in M5. If the Peltier network is first fitted with constant heat-pipe parameters, its learned relation may not remain optimal after the heat-pipe model is made adaptive. The results therefore suggest that known physical equations should be used when available, while neural networks still provide strong performance for select components.

6.2.3 Out of Distribution Performance

All models show increased prediction error outside the training range. The largest degradation comes from model M4 and M5 where the Peltier heat flow is represented by a neural network. Their errors increase by approximately 7 times when evaluated on data 40 % outside the training distribution. This suggests that the neural network mainly learns the heat-flow relation for the training distribution and extrapolates less reliably outside it. For example, even though M5 performs better than M1 in distribution, it shows larger deviations outside the learned range.

The models that retain the analytical Peltier equation show smaller error growth outside the training range, which indicates that preserving known physical structure improves extrapolation. M2 and M3 perform similarly out of distribution, with the error of M3 increasing by around 3 times compared with around 4 times for M2. This suggests that learning the uncertain heat pipe relation is useful, and that the neural-network parameterization can approximate the generated relation well.

6.2.4 Parameter Compensation

A limitation of the learned parameter approach is that different parameters may compensate for each other. A parameter might have high impact in a sensitivity analysis, and could strongly reduce the prediction error, even if it does not represent the physical source for the error. For example a heat load added near the outlet sensor could compensate for errors caused by other parts of the network, as demonstrated in Figure 6.1.

This issue is especially relevant for neural-network parameterization because of their ability to represent many different mappings. A neural network can minimize the output error by learning a compensating mapping, even if that mapping does not correspond to the physical parameter it is intended to represent. Models M4 and M5 in Figure 5.6 illustrate this risk, since they perform well in the training distribution but their error grows outside it.

The risk of parameter compensation can be reduced by enforcing stronger model restrictions. This includes carefully selecting which parameters are parameterized, limiting the inputs provided to each learned function, and keeping the known physical relations fixed whenever available.

Figure 6.1: Simple thermal network with two nodes, T_1 and T_2, connected through a thermal resistance. Node with temperature T_2 also has an additional heat flow Q. If Q is modeled as a neural network and received information such as T_1, it may compensate for incorrect values of R, reducing the output error without representing the true physical cause.

Table 6.1: Summary of the main discussion points from the model comparisons.

No.	Main discussion point
1	**Observation:** M1 gives 4.2 and 3.1 times larger prediction errors than M2 and M3. **Interpretation:** Input-dependent parameterizations improve accuracy.
2	**Observation:** M2 and M3 differ by only 11 mK. **Interpretation:** The neural-network parameterization can approximate the simple linear heat-pipe relation similarly to the linear one.
3	**Observation:** Replacing the analytical Peltier equation with a neural network gives only 10 mK larger mean error. **Interpretation:** A neural network can approximate the Peltier relation within the training range.
4	**Observation:** M5 improves by 26% compared with M4, but performs 2.5 times worse than M3. **Interpretation:** Learning uncertain components helps, but accurate known physics should be retained.
5	**Observation:** M4 and M5 errors increase by approximately 7 times outside the training range. **Interpretation:** Replacing known physics with neural networks reduces extrapolation reliability.

6.3 Interpretability and Practical Use

A main advantage of this proposed gray-box modeling approach is that it retains the underlying physical thermal network. Instead of learning the full input-output behavior from data alone, the number of relations that must be learned from data is reduced. This also makes the model more suitable when only limited transient data is available.

The interpretability and inspectability of the model is another important advantage. Since the model structure is based on physical relations, the internal states can be linked to meaningful quantities such as component temperatures, thermal resistances and heat flow terms. This makes it possible to understand internal behavior of the model while evaluating the final output prediction. It is also possible to check whether the model behaves consistently with the expected physics. This is important because low prediction error does not necessarily imply physically meaningful learned parameters, as discussed in the previous section.

The approach is closely related to Thermal Neural Networks, where neural networks are used inside a lumped thermal model to estimate thermal parameters. The main difference is that this work has a more selective parameterization. Known relations, such as the Peltier heat-flow equation, are kept explicit, while neural-network functions are applied to uncertain components, such as the heat-pipe resistances. This makes the learned parts easier to inspect and reduces the risk that neural networks absorb effects from known physical parameter relations.

Compared with black-box time-series models, such as RNNs or LSTMs, the proposed model provides less flexibility but stronger physical interpretation. They could act similarly to the previously discussed additional heat flow to the outlet temperature. Trainable state-space models can provide internal states, but unless additional constraints are imposed, these states and parameters do not necessarily correspond to physical temperatures or thermal couplings. Training such models on limited datasets can also be challenging.

Physics-informed neural networks can also improve data efficiency and physical plausibility by introducing soft physics-based constraints during training. The approach used in this work differs by embedding the thermal structure directly into the model, rather than only penalizing nonphysical behavior through the loss function. This makes the resulting model easier to interpret as a thermal network.

The practical use of the proposed type of model is therefore strongest in applications where both prediction accuracy and physical consistency are important. Examples include state estimation, disturbance estimation, and model-based control, where internal temperature estimates or physically meaningful parameters can be useful in addition to the predicted output.

6.4 Applications

Beyond simulation, an accurate gray-box thermal model can be useful for estimation, control and more. First, the model can help with state estimation, reconstructing unmeasurable sensors. Second, it can be used for disturbance and parameter estimation to take unknown heat loads, modeling error and changing operating conditions into account. Third, the model can be used to predict future system states. It is possible to predict future system states, and design control systems around these predictions, such as model predictive control.

6.4.1 Fault Detection

The gray-box model could be used for fault detection by comparing measured temperatures with model predictions. A persistent residual between the predicted and measured temperature may indicate a sensor fault, an unexpected heat load, or a change in the thermal behavior of the system.

Furthermore, the physical interpretability makes it possible to understand when the model itself breaks down. It is possible to track the changes in the predicted values from neural networks, and if they become nonphysical, it is an indication that the model is outside its operating range.

6.4.2 Digital Twin

A possible application of the gray-box thermal model is more accurate state estimation for digital twins. A Kalman filter combines a dynamic model with noisy sensor measurements to recursively estimate the internal state of a system [21]. For the thermal models considered in this work, this means that measured temperatures with unknown and unmeasured disturbances can be used to estimate unmeasured thermal node temperatures. This is useful when some temperatures are difficult to measure directly, or when a sensor becomes unavailable or unreliable.

For a discrete-time linear state-space model with measurement noise, (2.10) can be written as

$$\begin{aligned} T_{k+1} &= A_k T_k + B_k u_k + w_k, \\ y_k &= C T_k + v_k, \end{aligned} \tag{6.1}$$

for a discrete timestep k, where w_k and v_k are process and measurement noise. The Kalman gain can be calculated with the proposed model for a specific timestep k using

$$K_k = P_{k|k-1} C^\top \left(C P_{k|k-1} C^\top + R \right)^{-1}, \tag{6.2}$$

where

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^\top + Q. \tag{6.3}$$

An additional step can then be added in the prediction step, where the predicted state can be described as

$$\hat{T}_{k|k} = \hat{T}_{k|k-1} + K_k \left(y_k - C \hat{T}_{k|k-1} \right), \tag{6.4}$$

where K_k is the Kalman gain. The gain determines how strongly the estimate should be corrected by the measurement and is computed from the predicted state covariance and the measurement noise covariance [21].

In the context of the proposed thermal model, the state vector T_k would contain the thermal node temperatures, while y_k would contain the available sensor measurements. If one sensor is unavailable, the corresponding row can be removed from the measurement vector and measurement matrix, allowing the remaining sensors to correct the full thermal state estimate. This makes it possible to estimate temperatures at unmeasured or faulty sensor locations using the thermal coupling encoded in the model.

This principle is illustrated using the arduino setup shown in Figure 4.2. Sensor 2 is treated as unavailable and is therefore predicted from the model, while sensor 1 is the available measurement in the Kalman filter. The result is compared to a model without kalman filter in Figure 6.2, which shows that the kalman filter is able to respond to the large outside unmeasured disturbance applied at approximately 100 seconds using the remaining temperature sensor. Although the Kalman filter is not fully tuned, the example shows how gray-box models can be used for virtual sensing and disturbance compensation in a digital twin setting.

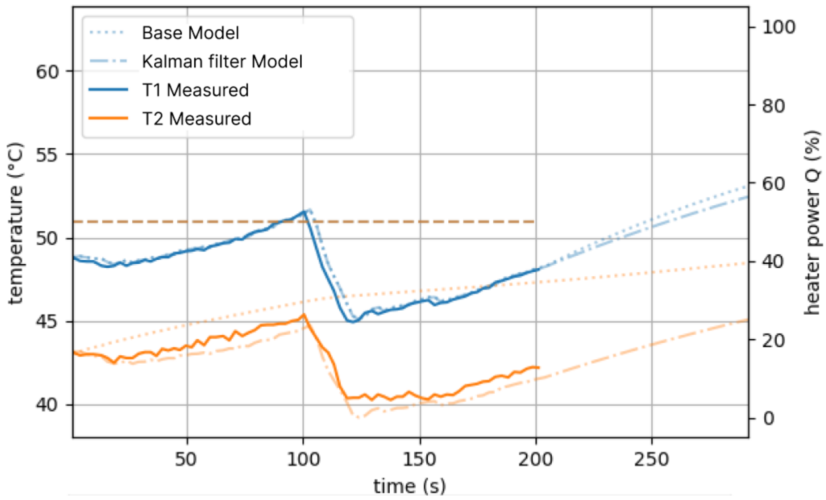

Figure 6.2: Comparison between model predictions with and without a Kalman filter, including the measured temperatures. An external cooling disturbance is introduced at approximately 100 seconds. The dashed lines, constant at 50 on the right axis, shows the heater power.

A moving horizon estimator (MHE) could also be applied in a similar way to the Kalman filter. Instead of updating a single state on the most recent data, the MHE estimates the current state by solving an optimization problem over a finite window of recent input-output data. This can be applied by making it predict the deviation from the calculated model and is perhaps more suitable for more nonlinear disturbances.

6.4.3 Model Predictive Control

Model predictive control (MPC) is a model-based control method where an optimization problem is solved at each sampling instant over a finite prediction horizon. The controller predicts the future system behavior, computes an input sequence that optimizes a chosen cost function while satisfying constraints, and applies only the first input before repeating the procedure at the next time step [29]. An example for a simple system can be seen in Figure 6.3.

The gray-box model proposed in this work is suitable for a simple MPC. Its internal states and physical interpretation is useful for additional constraints on e.g. predicted temperatures, heat flows, or actuator inputs. Furthermore, if a twice differentiable activation function is used in the neural networks, the model remains differentiable, and gradients can propagate through the model. This makes the model compatible with gradient based nonlinear optimization methods, which are commonly used in MPC formulations [9]. Non-smooth activation or switching functions can still work but may require other optimization methods.

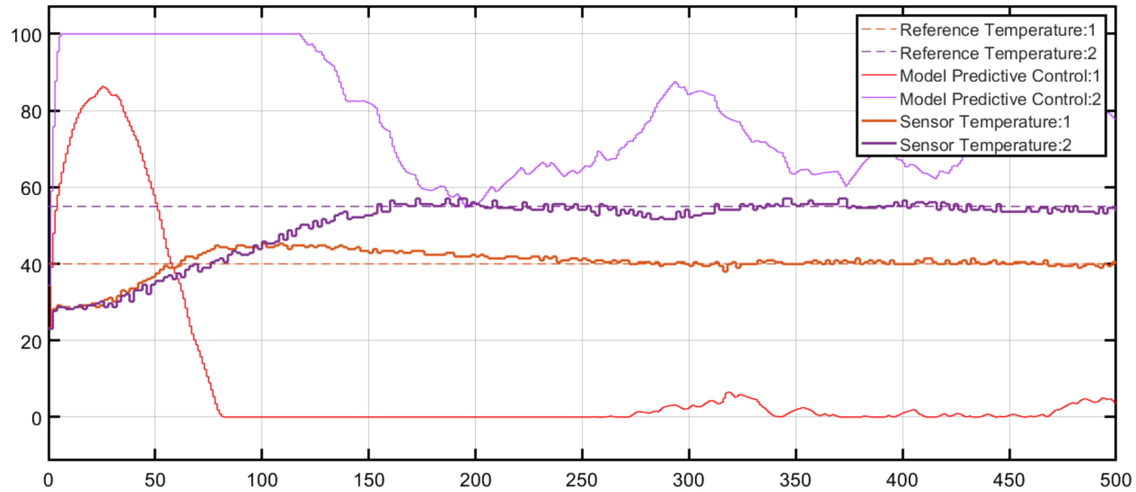

Figure 6.3: MPC applied to a simple thermal system consisting of two heaters and two heat sinks.

6.5 Ethics and Sustainability

The proposed modeling approach have both ethical and environmental implications when used in practical systems.

6.5.1 Sustainability Benefits

An advantage of more accurate thermal prediction is that it can support more energy-efficient operation. Better control can reduce unnecessary heating and cooling, while keeping operating temperatures closer to their desired range. This may also reduce thermal stress on components and improve their lifetime.

The approach can also support digital twins and fault detection, by identifying abnormal thermal behavior early. Early fault detection can reduce unnecessary replacement of components by supporting condition based maintenance instead of fixed-interval maintenance. Detecting and correcting faults early may also help extend components lifetime.

Furthermore, gray-box models may require less data than full black-box models because known physical relations are retained. This can reduce experimental effort and make model development more resource-efficient. In practical settings, this also means that test systems can be used for other experiments or operations instead of being occupied by long data-collection runs.

6.5.2 Ethical Risks and Responsible Use

Even though the models can produce low prediction errors during training, this does not guarantee physical correctness of learned parameters. Neural networks can hide nonphysical effects and different parameters can compensate for each other. This makes it important not to interpret low output error as proof of physical validity.

Overconfidence in the models may lead to unsafe decisions, especially if they are used for control, fault detection, or digital-twin applications. This risk increases when the system operates outside the training distribution, where prediction errors may grow and learned relations may become unreliable. Assumptions and limitations of the model should be clearly communicated.

6.6 Limitations

One limitation of this case study is that it is based on simulated data. This makes all the underlying parameters controlled, but does not include the additional challenge of real sensor noise, installation effects, and imperfect geometry. The method has been applied to the simple system in Figure 4.2, but there has not been any industrial validation yet.

Another limitation is that the generated heat pipe relation is linear, which is not as complex as its real world nonlinear relations. This causes there to be no large advantage of nonlinear neural networks and the results mainly show that adaptive parameterization is useful, but they do not clearly demonstrate that a neural-network parameterization is necessary.

Furthermore, the model variants tested were limited. There is no linear correlation for the Peltier and no parameterizations using hidden states like RNN and LSTM even though they are possible to implement. A comparison to a full black-box baseline could also be useful, although such models may require more data to train and perform reliably.

There is also some optimization sensitivity in a system like this. Fitting depends on initialization, learning rate, batching and selected trainable parameters. Some models required more tuning effort than others to obtain a stable fit. Neural models might require initialization strategies, such as initializing from a fitted linear model.

Finally, a low output error does not necessarily prove learned parameters are physically correct. This was discussed using out of distribution performance in this report, but might not be possible in an actual system. Different parameters can compensate for others and limited number of sensors make this worse.

6.7 Recommendations and Future Work

A recommendation from this work is to keep the model structure as simple as possible while retaining relevant physical relations. Known physical equations should not be replaced by neural networks unless they clearly are not represented. The results indicate that linear parameterization can be sufficient, but neural networks also result in similar performance. Low sensitive parameters can stay fixed, as well as well known parameters to reduce parameter compensation. In addition, transients and sufficiently exciting data should be used.

Future work could further explore data utilization. It is shown that the model can achieve good performance with a limited amount of data, but reducing the required testing time without significantly degrading performance would be useful in an industrial setting. This could be explored using input-design methods such as those proposed in [17] and [22]. The resulting models could then be evaluated as a function of test length to quantify the trade-off between model performance and testing time.

Furthermore, more comparisons between models could be made. A linear correlation for the Peltier element could clarify whether a simple correction is enough or if neural networks help with performance. Comparisons with black box models like RNN, LSTM or trainable state space models will also clarify the usefulness of this gray-box modeling. History-dependent parameterizations could be evaluated for systems with hysteresis or delayed thermal effects.

The training of networks and parameter selection could also be improved. Possible improvements include using a fitted linear model as an initialization for the neural network, exploration of other optimization algorithms, adding regularization terms, and improving input and parameter scaling and normalization. The parameter selection could also be extended beyond the simple sensitivity analysis by, for example, using Fisher-information based criteria to select parameters that are both influential and identifiable from available measurements.

7
Conclusion

This report explores a gray-box modeling approach for thermal systems, where selected components of a lumped thermal network are parameterized by neural networks. This is done with the goal of developing simple, interpretable, and accurate models.

The results show that linear and neural network parameterizations of the heat pipe perform similarly with average prediction errors of 0.15 K and 0.14 K, respectively. This indicates that neural networks can approximate relatively simple relationships. When the heat pipe is modeled, the physical Peltier model achieved an average error of 41% compared with the neural-network implementation. This indicates that established physical relations should be preserved when available and found to be accurate, while data-driven approaches are most effective for difficult-to-model, sensitive, or difficult-to-model components.

The accuracy also decreases for all models outside the training distribution. When evaluated 40 % outside the original operating range, the errors of the Peltier neural network parameterization models increased by approximately 7 times. In comparison, the constant-parameter model increased by around 3.5 times, while the final two tested models increased by approximately 3–4 times. This shows limitations in generalization, especially when known physical relations are replaced by neural networks.

Overall, this demonstrates that neural network parameterizations can capture unknown or unmodeled relationships, achieving performance comparable to known physical dynamics within the training distribution. The models are interpretable, although with some limitations, and can simplify the knowledge required about geometry.

Bibliography

[1] *2.160 Identification, Estimation, and Learning: Lecture 19*. MIT OpenCourseWare. 2006. URL: https://ocw.mit.edu/courses/2-160-identification-estimation-and-learning-spring-2006/resources/lecture_19/ (visited on 03/30/2026).

[2] Mohammad Alsalti, Victor G. Lopez, and Matthias A. Müller. "On the Design of Persistently Exciting Inputs for Data-driven Control of Linear and Nonlinear Systems". In: *arXiv preprint arXiv:2303.08707* (2023). URL: https://arxiv.org/pdf/2303.08707 (visited on 04/30/2026).

[3] APMonitor. *Temperature Control Lab*. Accessed 2026-05-07. URL: https://apmonitor.com/pdc/index.php/Main/ArduinoTemperatureControl.

[4] Nikolay Atanasov. *ECE171A: Linear Control System Theory – Lecture 2: Feedback Control Principles*. URL: https://natanaso.github.io/ece171a/ref/ECE171A_2_FeedbackPrinciples.pdf (visited on 03/30/2026).

[5] Alexandre Bayen. *EE C128 / ME C134 – Feedback Control Systems: Lecture – Chapter 1 – Introduction*. Sept. 10, 2013. URL: https://bayen.berkeley.edu/sites/default/files/ee_c128_chapter_1_0.pdf (visited on 03/30/2026).

[6] R. Best, U. Hingst, and H. Kreeb. "Cooling Systems Based on Peltier Elements and Heat Pipes for Space Application". In: *AIAA Thermophysics Conference*. 1980. DOI: 10.2514/6.1980-1464.

[7] Xavier Bombois et al. "Optimal Experiment Design for Open and Closed-loop System Identification". In: *Communications in Information and Systems* 11.3 (2011), pp. 197–224. DOI: 10.4310/CIS.2011.v11.n3.a1. URL: https://link.intlpress.com/JDetail/1805791097244827649 (visited on 04/30/2026).

[8] Yunus A. Çengel and Afshin J. Ghajar. *Heat and Mass Transfer: Fundamentals and Applications*. 6th ed. McGraw-Hill Science Engineering, 2020.

[9] COIN-OR. *Ipopt Documentation*. Accessed 2026-05-04. URL: https://coin-or.github.io/Ipopt/.

[10] J. Culham, M. Yovanovich, and Gerry Schneider. "Review of Thermal Joint Resistance Models for Nonconforming Rough Surfaces". In: *Applied Mechanics Reviews - APPL MECH REV* 59 (Jan. 2006). DOI: 10.1115/1.2110231.

[11] Munther A. Dahleh. *System Identification: Input Design, Persistence of Excitation, Pseudo-random Sequences*. MIT OpenCourseWare, 6.435 System Identification, lecture notes. Massachusetts Institute of Technology. URL: https:

//ocw.mit.edu/courses/6-435-system-identification-spring-2005/ffdd1299a755458f8b990bf3f8de8d20_lec4_6_435.pdf (visited on 04/30/2026).

[12] Alberto Dalla Libera, Ruggero Carli, and Gianluigi Pillonetto. "Kernel-based methods for Volterra series identification". In: *Automatica* 129 (2021), p. 109686. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2021.109686. URL: https://www.sciencedirect.com/science/article/pii/S0005109821002065.

[13] Fiveable Content Team. *Thermal Systems Definition - Thermodynamics I Key Term*. Last updated September 2025. Sept. 2025. URL: https://fiveable.me/key-terms/thermodynamics-i/thermal-systems (visited on 02/16/2026).

[14] Zhiqiang Gao. "On the centrality of disturbance rejection in automatic control". In: *ISA Transactions* 53.4 (July 1, 2014), pp. 850–857. DOI: https://doi.org/10.1016/j.isatra.2013.09.012. URL: https://www.sciencedirect.com/science/article/pii/S0019057813001559.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[16] Tamer Guclu and Erdem Cuce. "Thermoelectric Coolers (TECs): From Theory to Practice". In: *Journal of Electronic Materials* 48.1 (2019), pp. 211–230. DOI: 10.1007/s11664-018-6753-0.

[17] Kurt E. Häggblom. "Easy Ways to Design Inputs to obtain Uncorrelated Outputs in MIMO System Identification". In: *IFAC-PapersOnLine* 51.15 (2018). 18th IFAC Symposium on System Identification SYSID 2018, pp. 227–232. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.09.139. URL: https://www.sciencedirect.com/science/article/pii/S2405896318318007.

[18] IBM. *What Are State Space Models?* IBM Think Topics. IBM. n.d. URL: https://www.ibm.com/think/topics/state-space-model (visited on 02/23/2026).

[19] Frank P. Incropera et al. *Fundamentals of Heat and Mass Transfer*. 6th ed. John Wiley & Sons, 2007. ISBN: 978-0-471-45728-2.

[20] Frank P. Incropera et al. *Fundamentals of Heat and Mass Transfer*. 6th ed. John Wiley & Sons, 2007.

[21] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45. DOI: 10.1115/1.3662552.

[22] Milad Karimshoushtari and Carlo Novara. "Design of experiments for nonlinear system identification: A set membership approach". In: *Automatica* 119 (2020), p. 109036. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2020.109036. URL: https://www.sciencedirect.com/science/article/pii/S000510982030234X.

[23] Wilhelm Kirchgässner, Oliver Wallscheid, and Joachim Böcker. "Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning". In: *Engineering Applications of Artificial Intelligence* 117 (2023), p. 105537. DOI: 10.1016/j.engappai.2022.105537. URL: https://www.sciencedirect.com/science/article/pii/S0952197622005279.

[24] Pang Wei Koh et al. "WILDS: A Benchmark of in-the-Wild Distribution Shifts". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR, 2021, pp. 5637–5664.

[25] Ji-Su Lee, Seok-Ho Rhi, and Sun-Kook Kim. "Modeling and simulation of heat pipes: review". In: *Journal of Mechanical Science and Technology* (2024). URL: https://link.springer.com/article/10.1007/s12206-024-0437-x.

[26] Roland W. Lewis et al. *The Finite Element Method in Heat Transfer Analysis*. John Wiley & Sons, 1996.

[27] J. H. Lienhard V and J. H. Lienhard IV. *A Heat Transfer Textbook*. 6th. Version 6.00. Cambridge, MA: Phlogiston Press, Apr. 2024. 810 pp. URL: https://ahtt.mit.edu.

[28] Lennart Ljung. *System Identification: Theory for the User*. 2nd ed. Prentice Hall, 1999.

[29] Jacob Mattingley, Yang Wang, and Stephen Boyd. "Code Generation for Receding Horizon Control". In: *2010 IEEE International Symposium on Computer-Aided Control System Design*. 2010.

[30] K. K. Nandini and Muralidhara. "Peltier Based Cabinet Cooling System Using Heat Pipe and Liquid Based Heat Sink". In: *Proceedings of the National Conference on Challenges in Research & Technology in the Coming Decades (CRT 2013)*. IET, 2013, pp. 1–5. DOI: 10.1049/cp.2013.2536.

[31] Martijn van Noort et al. "Navigating the Landscape of Parameter Identifiability Methods: A Workflow Recommendation for Model Development". In: *CPT: Pharmacometrics & Systems Pharmacology* 13.7 (2024), pp. 1170–1179. DOI: 10.1002/psp4.13148.

[32] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[33] Steven L. Rickman. *Introduction to Numerical Methods in Heat Transfer*. Presentation. Cleveland, OH: NASA Technical Reports Server, 2024. URL: https://ntrs.nasa.gov/citations/20200006182 (visited on 05/11/2026).

[34] Viktoria Savatorova. "Exploring Parameter Sensitivity Analysis in Mathematical Modeling with Ordinary Differential Equations". In: *CODEE Journal* 16.1 (2023). DOI: 10.5642/codee.CZKZ5996. URL: https://scholarship.claremont.edu/codee/vol16/iss1/4/ (visited on 04/22/2026).

[35] Z. S. Spakovszky, E. M. Greitzer, and I. A. Waitz. *Convective Heat Transfer*. 16.Unified: Thermodynamics and Propulsion, lecture notes. Massachusetts Institute of Technology. URL: https://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node131.html (visited on 04/30/2026).

[36] Marc van de Wal and Dennis Heck. *Thermal Control Challenges in ASML Wafer Scanners*. Accessed 2026-05-07. 2021. URL: https://bigscience.nl/assets/Documents/Presentations/ILO-net-Symposium-07-10-2021/Marc_van_de_Wal_D001079592-01-PSA_Thermal_challenges_in_ASML_wafer_scanners_ILO_Net_Event_October_07_2021_LR.pdf.

[37] O. C. Zienkiewicz and Y. K. Cheung. *The Finite Element in Structural and Continuum Mechanics*. p. 148. London: McGraw-Hill Publishing Company Ltd., 1967.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY