



Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning

Wilhelm Kirchgässner*, Oliver Wallscheid, Joachim Böcker

Paderborn University, Warburger Straße 100, 33098 Paderborn, Germany

ARTICLE INFO

Dataset link: kaggle.com, github.com

Keywords:

Hybrid machine learning
Functional safety
Permanent magnet synchronous motor
System identification
Temperature estimation

ABSTRACT

With electric power systems becoming more compact with higher power density, the relevance of thermal stress and precise real-time-capable model-based thermal monitoring increases. Previous work on thermal modeling by lumped-parameter thermal networks (LPTNs) suffers from mandatory expert knowledge for their design and from uncertainty regarding the required power loss model. In contrast, deep learning-based temperature models cannot be designed with the low amount of model parameters as in a LPTN at equal estimation accuracy. In this work, the thermal neural network (TNN) is introduced, which unifies both, consolidated knowledge in the form of heat-transfer-based LPTNs, and data-driven nonlinear function approximation with supervised machine learning. The TNN approach overcomes the drawbacks of previous paradigms by having physically interpretable states through its state-space representation, is end-to-end differentiable through an automatic differentiation framework, and requires no material, geometry, nor expert knowledge for its design. Experiments on an electric motor data set show that a TNN achieves higher temperature estimation accuracies than previous white-/gray- or black-box models with a mean squared error of 3.18 K^2 and a worst-case error of 5.84 K at 64 model parameters.

1. Introduction

Lumped-parameter thermal networks (LPTNs) denote a system of ordinary differential equations (ODEs) and a popular choice for various thermal modeling applications throughout different industry branches. Most notably, the fact that they can facilitate only few model parameters at a high estimation accuracy renders them popular for real-time temperature estimation tasks in electric power systems, such as multichip power modules (Iachello et al., 2014; Bahman et al., 2016) or electric machines for automotive and automation applications (Wallscheid and Böcker, 2016; Bracikowski et al., 2012; Boglietti et al., 2016; Wöckinger et al., 2020) as well as electric generators within electric power plants (Lim et al., 2008; Rostami et al., 2013). An accurate real-time temperature estimator is often highly desirable whenever a system's performance depends on thermal state information of such parts that are not economically measurable on a sensor base, e.g., the permanent magnets in a synchronous electric machine or semiconductors in power electronic converters. Moreover, even with system temperatures being tracked by sensors, a redundant thermal model becomes more and more relevant against the background of increasing functional safety requirements.

Being a parameter-light model often makes the LPTN approach the only feasible choice, although established, computational heavy, thermal analysis tools like computational fluid dynamics (CFD) and

finite-element modeling (FEM) based on partial differential equations (PDEs) usually give a more thorough and accurate picture of heat flows in any intricate system (Boglietti et al., 2009).

Apart from thermal models, there are also real-time methods in the electric machine domain based on either invasive, high-frequency signal injection (Reigosa et al., 2015, 2019) or precise observers that indirectly assess an electric machine's thermal condition from temperature-dependent electric parameters (Specht et al., 2014). Both methods expose high system parameter sensitivity as well as operation-point-dependent accuracies, that might lead to undesired temperature estimation errors (Wallscheid et al., 2016). What is more, a considerable amount of domain knowledge is required to be incorporated in each system model. A comprehensive review of temperature monitoring methods is given in Wallscheid (2021).

1.1. State-of-the-art thermal modeling

The design of an LPTN of sufficient accuracy likewise necessitates expert knowledge from the application's domain, as well as geometry and material information from manufacturers. These help to identify the parameter variation of thermal resistances and capacitances, which is increasingly abstract and defies physical derivation the more the geometry is approximated with simple shapes (e.g., cylinders or spheres)

* Corresponding author.

E-mail address: kirchgaessner@lea.uni-paderborn.de (W. Kirchgässner).

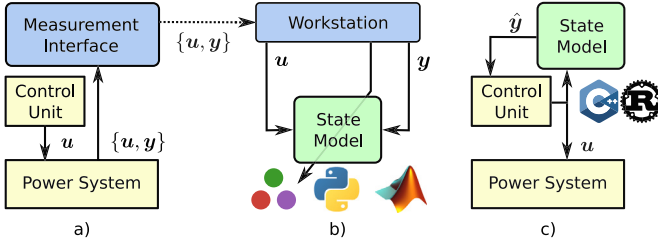


Fig. 1. The data-driven state modeling process is depicted. (a) The power system is either upgraded with measurement equipment or operated at a test bench, and reveals information about the system's input u and measured states y (e.g., interior temperatures). (b) The recorded data set is used to tune a state model on estimating y . (c) The tuned model is deployed on an embedded system and informs the control unit. Relevant programming languages for any phase are indicated, yet not all of them are used in this work, see Section 3.

to further reduce the model complexity and order. What is more, power losses are usually even more difficult to grasp, and significant research effort is put into their modeling (Gedlu et al., 2020; Qi et al., 2014). In order to overcome modeling errors imposed by simplified assumptions over the system, empirical measurement data (for instance, as collected with a test bench) is increasingly utilized and employed, e.g., in the form of look-up tables or for fine-tuning analytically found initial parameter values (Wallscheid and Böcker, 2016; Wöckinger et al., 2020; Gedlu et al., 2020). The idea is depicted in Fig. 1.

Advancing the trend of fitting models on empirical data, machine-learning-based thermal modeling was proposed recently, where artificial neural networks (ANNs) are tuned to estimate temperatures of interest from a parallel stream of readily available sensor data, effectively harnessing statistical correlations between the latent temperatures and measurable quantities (Kirchgässner et al., 2020, 2021b; Lee and Ha, 2020; Zhang et al., 2018). These methods are completely detached from heat transfer theory and rely solely on empirical data. They neither necessitate expert knowledge nor geometry or material information for their design at the same level of accuracy as that of data-driven LPTNs. However, an increased computational demand in real time is required due to additional feature engineering and a substantially increased amount of model parameters (Kirchgässner et al., 2021b). Moreover, learnt parameters of an ANN are not as interpretable as of an LPTN. This implies that other systems, albeit similar to the one observed by the black-box model, would always require, again, empirical data for an appropriate temperature estimation unit. Furthermore, initial conditions cannot be set in black-box models as their internal states are abstract, if there are any at all.

1.2. Lumped-parameter thermal networks

The principles of LPTN design are well-known and, therefore, are only briefly described for illustrative purposes. Based on heat transfer theory, LPTNs are designed to estimate specific spatially separated system components' temperatures by representing heat flows through an equivalent circuit diagram, where parameters stand for thermal characteristics. An appropriate LPTN structure is usually derived from the heat diffusion equation (Bergman et al., 2007):

$$\rho c_p \frac{\partial \vartheta}{\partial t} = p + \nabla \cdot (\lambda \nabla \vartheta), \quad (1)$$

where ρ denotes the mass density, c_p stands for the volume's specific heat at constant pressure, ϑ is the scalar temperature field, p being the thermal energy generation from other energy sources (such as electric or chemical energy), ∇ being the del operator across spatial dimensions, and λ representing the potentially nonconstant thermal conductivity. For steady-state heat transfer, a constant thermal resistance can then be derived analytically from geometry as well as boundary and initial conditions to describe one-dimensional heat flow that depends

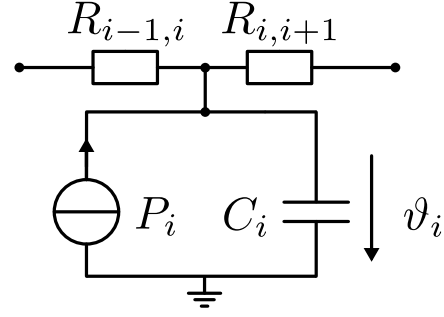


Fig. 2. A typical thermal element consisting of a thermal capacitance, a power loss source, and multiple thermal resistances leading to other thermal elements. ϑ_i denotes the i th element's average temperature.

on the absolute temperature difference, much like current flow depends on potential differences in an equivalent electric circuit. In the multi-dimensional case, converting (1) into a spatial finite-difference equation is often helpful. Fig. 2 shows a thermal element, which is an LPTN's basic building block and represents one particular subdomain in the system. This simplified diagram discretizes a system's state space and converts the governing PDEs into ODEs.

However, the parameters in Fig. 2 are commonly operation-point-dependent rather than being constant for most applications. Moreover, especially convective and radiative heat transfer processes inherit highly nonlinear thermal characteristics making the transient definition of thermal parameters a challenging endeavor (Boglietti et al., 2006; Howey et al., 2012). Usually, thermal parameters are thus denoted in dependence on a scheduling vector $\zeta(t)$. This vector contains not only measurement quantities that are available during operation but also all thermal states themselves, which will be discussed later.

In order to avoid incorporation of geometric specifications upfront, a general-purpose LPTN structure of the following form is considered as starting point for subsequent designs: The i th thermal element in a fully interconnected LPTN of m th order describes the ODE

$$C_i(\zeta(t)) \frac{d\vartheta_i}{dt} = P_i(\zeta(t)) + \sum_{j \in \mathcal{M} \setminus i} \frac{\vartheta_j - \vartheta_i}{R_{i,j}(\zeta(t))} + \sum_{j=1}^n \frac{\bar{\vartheta}_j - \vartheta_i}{R_{i,j}(\zeta(t))}, \quad (2)$$

where C denotes the thermal capacitance, P the power loss, $R_{i,j}$ the bidirectional thermal resistance between the temperatures at node i and j , and ϑ the actual temperature at time t with $\mathcal{M} = \{1, 2, \dots, m\}$. Moreover, temperatures that are measurable during operation (typically ambient and coolant temperatures) can be incorporated as further n ancillary nodes merely consisting of an equivalent thermal source $\bar{\vartheta}$.

It will prove beneficial for the following sections to rearrange (2) into a quasi-linear parameter-varying (quasi-LPV) state-space representation encompassing all nodes in matrix form:

$$\begin{aligned} \frac{dx}{dt} &= A(\zeta(t))x + B(\zeta(t))u(\zeta(t)), \\ y &= Cx, \end{aligned} \quad (3)$$

with

$$\begin{aligned} x &= \vartheta = [\vartheta_1 \dots \vartheta_m]^T, \\ u &= [P_1(\zeta(t)) \dots P_m(\zeta(t)) \quad \bar{\vartheta}_1 \dots \bar{\vartheta}_n]^T, \\ A &\in \mathbb{R}^{m \times m}, \quad B \in \mathbb{R}^{m \times (m+n)}, \quad C \in \mathbb{R}^{q \times m}, \end{aligned}$$

where A , B , and C are the system, input, and output matrix, respectively. The output matrix C is typically a masking operation allowing the system output y to encompass fewer temperatures than are being modeled with $q \leq m$. Identification of $A(\zeta(t))$ and $B(\zeta(t))$ is central to thermal modeling with LPTN structures.

Furthermore, as mentioned in the previous section, determining the dependence of P_i on $\zeta(t)$ is usually a hard task: there are often only few sensor information (e.g., electrical current at the terminals of a

power electronic converter or electric machine) from which a complex nonlinear spatial power loss distribution is to be inferred. Finding this mapping is notoriously challenging in addition to identifying $A(\zeta(t))$ and $B(\zeta(t))$.

1.3. State-space machine learning

Recently, a paradigm shift emerged promoting the combination of general-purpose machine learning, stemming from computer science, and prior scientific knowledge in the form of PDEs commonly appearing in nature, instead of contemplating these as distinct problem sets. Leveraging the potential of automatic differentiation (Baydin et al., 2018), various data-driven approaches were presented for both, discovery and solution of nonlinear PDEs (Rudy et al., 2017; Chen et al., 2018; Raissi et al., 2019; Rackauckas et al., 2020): a prominent idea is the physics-informed ANN (PINN), which adds any PDE of interest to a standard ANN's cost function (Raissi et al., 2019). While a fitted PINN has been shown to respect PDEs with certain boundary and initial conditions, their convergence properties are not fully understood yet, and, more notably, their architecture is unaltered compared to usual ANNs. This makes a PINN as noninterpretable as typical black-box models.

Conversely, neural ordinary differential equations denote initial value problems of the form

$$\frac{dx}{dt} = g_\theta(x, u, t), \quad x(t_0) = x_0 \quad (4)$$

with x denoting a system's state at time $t \in [t_0, t_T]$, u the system's input, and g_θ being a neural network function parameterized with θ (Chen et al., 2018). A differentiable ODE solver replaces a hidden layer, which is potentially repeatedly executed to implicitly learn the latent function g_θ . Although this is a first step to applying an outer structure to an ANN that imitates the nonlinear workings of ODEs, it relies too heavily on the universal approximation theorem (Hornik et al., 1989), and prior scientific knowledge of fundamental mechanisms are not directly incorporated.

This shortcoming is addressed by universal differential equations (UDEs), which extend (4) to

$$\frac{dx}{dt} = f(x, u, t, g_\theta(x, u, t)), \quad x(t_0) = x_0 \quad (5)$$

where f represents arbitrary known mechanisms of the underlying physical problem in which difficult to model parts are approximated by any universal approximator g_θ (Rackauckas et al., 2020). Note, however, that except for some concrete examples with very specific ANN constellations there is no rule-based procedure proposed on how to receive an appropriate UDE given a nonlinear system. Due to the broadly outlined definition of an UDE, most modeling problems still remain to be solved by experts in the corresponding domain.

An early attempt to combine system representations in state space with NNs is denoted by state-space neural networks (SSNNs), which was proposed within the control theory domain over two decades ago (Rivals and Personnaz, 1996). SSNNs were developed especially for general nonlinear, dynamic systems, in order to leverage well-established stability and convergence theorems with the nonlinear modeling capability of ANNs. They describe the difference equation

$$\begin{aligned} \hat{x}[k+1] &= g_\theta(\hat{x}, u, k), \\ \hat{y}[k] &= h_\theta(\hat{x}, u, k), \end{aligned} \quad (6)$$

where g_θ and h_θ are nonlinear functions represented by potentially the same ANN with parameters θ at discrete time k .

Following this, most contributions deriving from SSNNs employ some form of recurrent ANNs (RNNs)

$$g_\theta(\hat{x}, u, k) = \sigma(W_r \hat{x}[k] + W_h u[k] + b), \quad (7)$$

with $W_{[r,h]}$ being the recurrent and forward weight matrices, b being the bias vector, and σ denoting the RNN's activation function (Zamareño and Vega, 1998; Abbas and Werner, 2008). While the computer-science-based movement on machine learning estimation of state derivatives does not restrict the topology of the state estimator to any form (e.g., certain number of layers, standard vs. convolutional ANNs, etc.), SSNN derivatives are often proposed with those settings fixed.

Another attempt to combine pre-existing physical models with ANNs are *structured neural networks* (Garcia et al., 2007; Andre et al., 2013). Here, a standard feed-forward ANN would be customized to have a limited set of interconnections between neurons and layers with carefully selected activation functions to accommodate known physical models. A structured ANN with many different distinct sub-ANNs for different terms of an ODE in a recurrent setting can be regarded as being similar to a TNN, and they share many common advantages over unstructured, black-box ANNs, but a structured ANN was not proposed for heat transfer problems yet.

TNNs are not to be confused with thermodynamics-based NNs (Masi et al., 2021), which were developed in parallel to TNNs and are applicable to constitutive modeling, which restricts itself to thermodynamically closed systems (no heat transfer). In this work, the chosen topology is more sophisticated than (7) and tailored to heat transfer processes. As will be pointed out in the next section, this contribution is motivated by (5) and informed by (2).

1.4. Contribution

In this work, the thermal neural network (TNN) is proposed, which facilitates physically reasoned heat transfer mechanics and can be applied to any system that exposes component temperatures of particular interest. Unifying the concepts of LPTNs and UDEs, a TNN is universally applicable across all relevant engineering domains that deal with heat transfer. In particular, it integrates the following advantages into one modeling approach:

- Physically interpretable model states are featured, which make initial condition adjustment simple.
- TNNs are end-to-end differentiable, which enables fast, parallelized, and distributed training scenarios with automatic differentiation frameworks.
- Neither expert knowledge, nor material or geometry information are required for the design.
- Being completely data-driven, parasitic real-world effects are considered through observational data.
- A quasi-LPV system is identified where unknown relationships between scheduling variables and system matrices are automatically discovered.
- Even relatively small model sizes achieve high estimation accuracies, which facilitates real-time capability.

This article begins with the working principle of a TNN, followed by the demonstration of its modeling capacity through cross-validation on a representative data set. A kernel-based hyperparameter optimization is described, and chances for model reduction are shown. Finally, the TNN's initial condition adjustment and its recovering properties are demonstrated. To assist related work all code is published on GitHub.¹

2. Thermal neural networks

Circumventing the intricate and error-prone design effort that accompanies the identification of the thermal properties' functional dependence on the system's operation point, the following architecture is leveraging the high nonlinear modeling capacity of ANNs. It learns the existing physical relationship between temperatures and other sensor information without geometry or material information — just from

¹ <https://github.com/wkirsnn/thermal-nn>

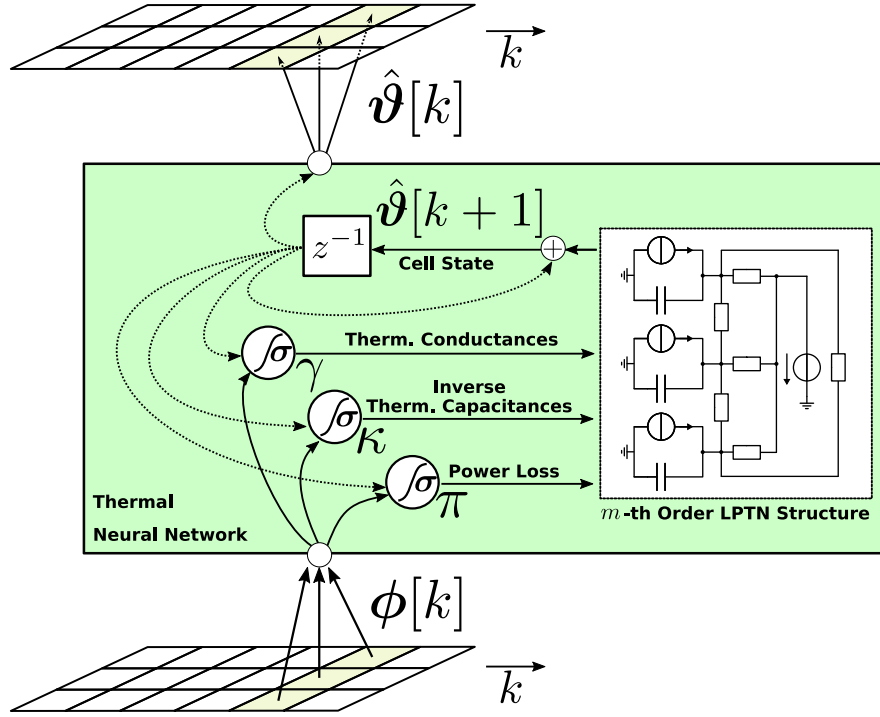


Fig. 3. A TNN considers all input features of the current time step $\phi[k]$ to estimate the next time step's temperatures $\vartheta[k+1]$. This estimate acts as cell information within the TNN to further inform the three function approximators γ , κ , and π . An LPTN structure of third order with an ancillary thermal source is exemplarily depicted.

measurement data and the general structure of an LPTN. Building on the concept of (5), the thermal neural network (TNN) reformulates (2) after a first-order Euler discretization to

$$\begin{aligned} \hat{\vartheta}_i[k+1] = & \hat{\vartheta}_i[k] + T_s \kappa_i[k] \left(\pi_i[k] + \right. \\ & \left. \sum_{j \in \mathcal{M} \setminus i} (\hat{\vartheta}_j[k] - \hat{\vartheta}_i[k]) \gamma_{i,j}[k] + \sum_{j=1}^n (\tilde{\vartheta}_j[k] - \hat{\vartheta}_i[k]) \gamma_{i,j}[k] \right), \end{aligned} \quad (8)$$

with $\hat{\vartheta}_i[k]$ denoting the i th node's normalized temperature estimate at discrete time k , T_s being the sample time, and κ_i , π_i as well as $\gamma_{i,j}$ denoting arbitrary feed-forward ANN outputs dependent on $\zeta[k] = [\tilde{\vartheta}^T \ \hat{\vartheta}^T \ \xi^T]^T$, which in turn consists of the ancillary temperatures $\tilde{\vartheta}[k]$, the temperature estimates $\hat{\vartheta}[k]$, and additional observables $\xi[k] \in \mathbb{R}^o$. More specifically, $\gamma(\zeta)$ approximates all thermal conductances between components of interest, $\pi(\zeta)$ all power losses within these components, and $\kappa(\zeta)$ determines all inverse thermal capacitances. Common observables are, e.g., the current and voltage vector as well as the motor speed in an electric drive train.

Without loss of generality, $\kappa(\zeta)$ can usually be reduced to just end-to-end trainable constants θ_c in the form

$$\kappa = 10^{\theta_c}, \quad \theta_c \in \mathbb{R}^m,$$

whereas

$$\pi : \mathbb{R}^{m+n+o} \rightarrow \mathbb{R}^m,$$

$$\gamma : \mathbb{R}^{m+n+o} \rightarrow \mathbb{R}^{(m+n)(m+n-1)/2}$$

are distinct ANNs. This reduction corresponds to a simplification through the lumped capacitance method, which is feasible in many engineering applications (Bergman et al., 2007). The TNN concept is sketched in Fig. 3.

Hence, except for κ , the general algorithmic shape is described by the multilayer perceptron (MLP) algorithm fed by $\hat{\vartheta}$ and $\phi = [\tilde{\vartheta}^T \ \xi^T]^T$:

$$\begin{aligned} h^{(0)}[k] &= \sigma^{(0)}(\mathbf{W}_r \hat{\vartheta}[k] + \mathbf{W}_h^{(0)} \phi[k] + b^{(0)}), \\ h^{(l)}[k] &= \sigma^{(l)}(\mathbf{W}_h^{(l)} h^{(l-1)}[k] + b^{(l)}), \quad \forall l > 0, \\ g_\theta[k] &= h^{(L-1)}[k], \end{aligned} \quad (9)$$

with $\sigma^{(l)}(\cdot)$ denoting the nonlinear activation function at layer l , and $\theta = \{\mathbf{W}_r, \mathbf{W}_h^{(l)}, b^{(l)} : l \in [0, L-1]\}$ describing the trainable parameters that exist independently for both, π and γ . Note that if one finds the assumptions under the lumped capacitance method to be too simplistic for the task, lifting κ to the MLP form would be an appropriate first measure. Rearranging (8) into matrix form similar to (3) and substituting (κ, π, γ) with (9) will reveal the relationship between the scheduling variables and the quasi-LPV system matrices as modeled by a TNN.

The TNN inherits recurrent connections like a RNN, but its cell consists of three (arbitrary complex) ANNs. The TNN's state is represented by a vector containing the estimated temperatures and the TNN's output for the next time step. In contrast to standard RNN topologies, a TNN's topology can be varied through the three sub-NNs, but it is not meant to be stacked as a whole onto several layers, as this would defeat the resemblance to the ODE structure.

Training remarks

It is well known that training RNNs with the error backpropagation method suffers from the vanishing and exploding gradient problem (Pascanu et al., 2012). The problem defines the phenomena of gradients soon becoming very large or very close to zero the longer the sequence on which gradients are accumulated for the subsequent weight update. This is especially severe for arbitrary long sequences on which thermal models conduct estimates. Early on in machine learning literature, this problem was curbed with the introduction of memory blocks, like the long short-term memory (LSTM) block, that would expand each RNN neuron (Gers and Cummins, 1999). However, the general-purpose topology of an LSTM is in conflict with the physically motivated TNN, such that a TNN has to overcome this hurdle with other methods, e.g., gradient normalization, clipping, and truncated backpropagation through time (TBPTT) (Williams and Peng, 1990).

Physical interpretation

Since no information about the geometry or material of the system is assumed to be known a priori, a fully-connected LPTN denotes the starting point of the TNN design, i.e., \mathbf{A} and \mathbf{B} are dense matrices. For a

Table 1

Test bench parameters.

Motors	device under test	load motor
Type	PMSM	induction machine
Power rating in kW (nom./max.)	22/52	160/210
Voltage rating in V	177	380
Current rating in A (nom./max.)	110/283	293/450
Torque rating in Nm (nom./max.)	110/250	380/510
Pole pair number	8	1
Cooling	water-glycol	forced air
Thermocouples	type K - class 1	-
Inverter	3 × SKiiP 1242GB120-4DW	
Typology	voltage source inverter 2-level, IGBT	
Inverter interlocking time	3.3 μs	
Controller hardware	dSPACE	
Processor board	DS1006MC, 4 cores, 2.8 GHz	

thermal model that is modeling $m+n$ different temperatures, this translates to $(m+n)(m+n-1)/2$ thermal conductances and, thus, a parameter size complexity of $\mathcal{O}((m+n)^2)$. In practice, however, several components within a system are often sufficiently detached such that the heat transfer between them is negligible. Featuring physically interpretable states, a fitted TNN can evidence such prunable connections, that can be removed in a subsequent TNN design iteration. This parameter reduction method is presented in Section 3.3.

Domain knowledge

If domain knowledge is available, this can be incorporated in several ways: for instance, by means of constraining the choice of topology defining parameters like the activation functions of the output layers of γ , κ , and π , or by means of feature engineering or elimination per thermal parameter estimator. Without loss of generality, it can be recommended to apply the ℓ_1 -norm on all the ANNs' outputs, as negative thermal conductances, capacitances, and power losses are rarely physically plausible.

Initial condition adjustment

As is usual for state-space models, the TNN structure asks for an initial temperature estimate. In simulations and training, these initial values can be set with the ground truth values, while in a field application the ambient temperature is a good approximation. Note that this initial conditioning is not possible for black-box models which have completely abstract parameter structures. The ability to set the initial condition is thus to be seen as a major advantage in favor of models representing a dynamic state space. The severeness of a biased initial guess is investigated in Section 3.4.

3. Experiments

In order to demonstrate the TNN concept's efficacy, several experiments are conducted on experimentally obtained data, which was also already utilized in Kirchgässner et al. (2019), Gedlu et al. (2020), Kirchgässner et al. (2020, 2021b). For transparency, we published the data set (Kirchgässner et al., 2021a). This data set represents 185 h of multivariate measurements sampled at 2 Hz from a 52 kW three-phase automotive traction permanent magnet synchronous motor (PMSM) mounted on our test bench, which is torque-controlled while its motor speed is determined by a speed-controlled load motor (210 kW) and fed by a 2-level IGBT inverter. All measurements were recorded by dSPACE analog-digital-converters (ADCs) which have been synchronized with the control task. Table 1 compiles the most important test bench parameters.

The contour plot in Fig. 4 depicts the excitation within the data set, which is characterized by constant and randomized excitation likewise.

The measured permanent magnet surface temperature as averaged across four sensors around a single permanent magnet is assumed to

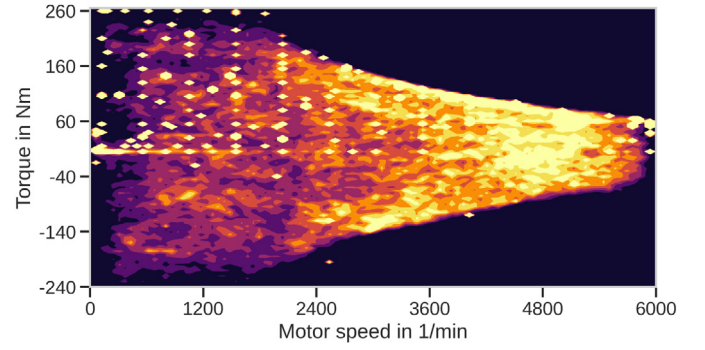


Fig. 4. An interpolated 2D-histogram of visited operation points is shown in the motor-speed-torque-plane. Brighter areas were visited more often.

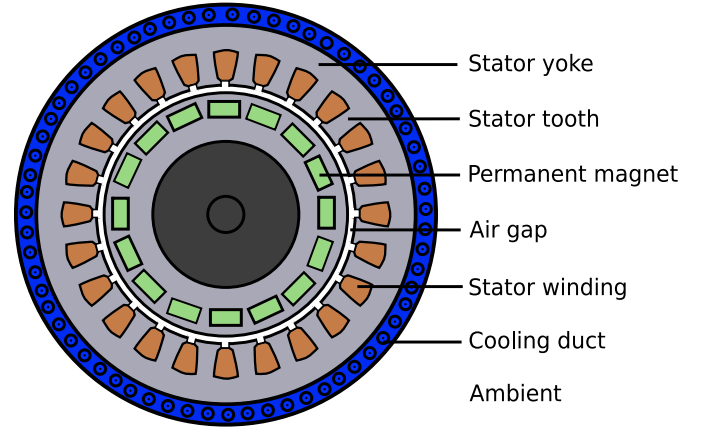


Fig. 5. Sketched cross-section of the PMSM considered for experimental tests. Except for the air gap, all labeled components are measured on a test bench.

be representative for the rotor temperature, and is transmitted with a telemetry unit. For all experiments, sensor information is classified in the following way:

$$\begin{aligned}\xi &= [u_s \quad i_s \quad \omega_{\text{mech}}]^T, \\ \bar{\vartheta} &= [\vartheta_a \quad \vartheta_c]^T, \\ \mathbf{x} = \mathbf{y} &= [\vartheta_{\text{PM}} \quad \vartheta_{\text{SY}} \quad \vartheta_{\text{ST}} \quad \vartheta_{\text{SW}}]^T,\end{aligned}$$

with u_s and i_s being the voltage and current vector norm, respectively; ω_{mech} denoting the mechanical angular frequency; and $\vartheta_{\{\text{a, c, PM, SY, ST, SW}\}}$ representing temperatures at the ambient, coolant, permanent magnet, stator yoke, stator tooth, and stator winding, respectively. An exemplary cross-section of a PMSM is sketched in Fig. 5 highlighting the sensors' geometric arrangement. $\bar{\vartheta}$ is assumed to be the system boundary, i.e., no heat transfer to other environmental temperatures are contemplated. In applications that lack in sensory information for boundary temperatures, adding constant support temperatures, e.g., a very cold and a rather hot one, can be beneficial since a TNN can adjust the corresponding thermal conductances to these ideal sources accordingly.

As is usual for ANN training, all quantities are normalized: i_s is divided by 100 A, u_s by 130 V, ω_{mech} by $2\pi \cdot 6000 \text{ min}^{-1}$, and all temperatures by 100 °C. No further feature engineering is employed, which would be common for pure black-box alternatives where ϕ would effectively grow to over 100 elements (Kirchgässner et al., 2021b). The lack thereof can be seen as a computational advantage in favor of TNNs.

All models are evaluated for their estimation performance by the mean squared error (MSE) between estimated and ground truth trajectory, averaged across the targeted temperatures. Proper cross-validation is ensured and described below. Throughout this work, Tensorflow

Table 2
Hyperparameter search intervals and results.

Symbol	Hyperparameter	Search interval	Optimum
$L_{\{\pi, \gamma\}}$	No. hidden layers in $\{\pi, \gamma\}$	[1, 3]	{3; 3}
$n_{\{\pi, \gamma\}}^{(l)}$	No. hidden units per layer in $\{\pi, \gamma\}$	[2, 128]	{109 \rightarrow 3; 2 \rightarrow 2}
$\sigma_{\{\pi, \gamma\}}^{(l)}$	Activation function per layer in $\{\pi, \gamma\}$	Sigmoid, tanh, linear, ReLU, biased Elu, sinus	{ReLU \rightarrow sigmoid \rightarrow sigmoid; Tanh \rightarrow sinus \rightarrow biased Elu}
$\alpha_{\{\pi, \gamma\}}^{(l)}$	ℓ_2 regularization rate per layer in $\{\pi, \gamma\}$	$[10^{-9}, 10^{-1}]$	{ $10^{-8} \rightarrow 10^{-8}$; $5.5 \cdot 10^{-6} \rightarrow 5.3 \cdot 10^{-8}$ }
$P_{ }$	Dedicated branches in π	[True, False]	False
η	Initial learn rate	$[10^{-5}, 1]$	$11 \cdot 10^{-3}$
\tilde{L}_{TBPTT}	TBPTT length in samples	[8, 2048]	1227
S	Split profiles into subsequences	[True, False]	False
\tilde{L}_S	Subsequence length in hours	[0.5, 4]	–
β	Optimizer	Adam, Nadam, Adamax, SGD, RMSProp	Nadam

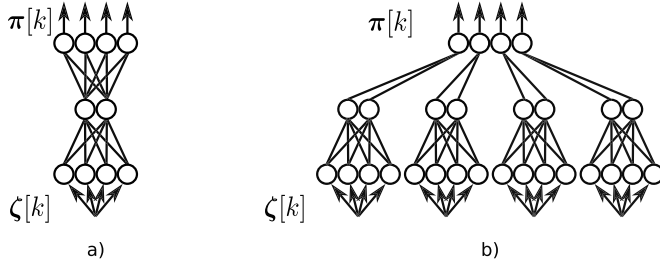


Fig. 6. The HPO is given the opportunity to multiply the amount of model parameters in π (from (a) to (b)) by dedicating single MLP structures to each targeted component. This is a technique that has been proven useful in Kirchgässner et al. (2020). Given the higher amount of thermal conductances compared to targets, this scheme is not contemplated for γ .

2 (Abadi et al., 2016) (v2.3.2) is used to train and investigate TNNs, under Python v3.7. Auxiliary Python packages can be inferred from the GitHub repository. Computations were executed on CPU nodes of the Paderborn Center for Parallel Computing (PC²) with clock rates of 2.45 GHz.

3.1. Automated architecture optimization

Since a TNN only loosely defines the topological extent of its three thermal parameter estimators γ , κ , and π , there is much room for hyperparameter optimization (HPO).

3.1.1. Search intervals

Amongst topological factors, optimization-related hyperparameters are optimized as well. An overview of all hyperparameters, their search intervals, and the found optima, is compiled in Table 2. Generally, interval bounds were chosen to restrict parameter size growth into dimensions that would not be reasonably real-time computable anymore.

The set of activation functions α comprises those common in literature and the biased Elu $\sigma_{\text{biasedElu}}(\cdot) = \sigma_{\text{elu}}(\cdot) + 1$, in order to produce nonnegative output. $P_{||}$ represents a flag that signals the expansion of π into dedicated MLP branches — one per target component (compare Fig. 6). The TBPTT length \tilde{L}_{TBPTT} denotes how many forward passes are accumulated and averaged across for each backward pass during training with error backpropagation. Splitting all measurement profiles into subsequences of a certain length \tilde{L}_S , where each subsequence is treated as an independent record, is indicated by the flag S .

3.1.2. Cross-validation scheme

In this work, the investigated data set consists of several measurement records of different lengths. A common approach is to split those records into different sets dedicated for either training or validating the model. Following scheme is applied: First, no record should be present in both training and testing sets at the same time, in order to preclude exploitation of record-specific anomalies. Second, two so-called folds of mutually-exclusive sets are defined comprising a number of profiles

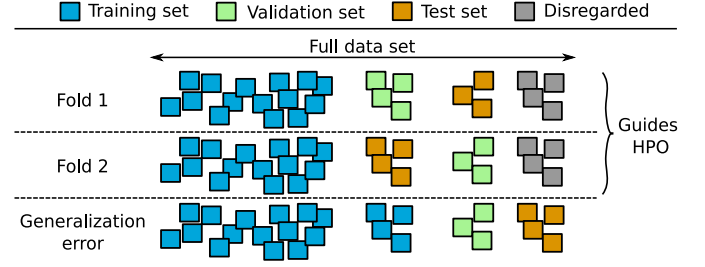


Fig. 7. Cross-validation strategy where each square represents a measurement profile.

that either act as validation or test set depending on the current iteration. Both, validation and test set, are not observed by the model during training. While the validation set is evaluated after each training epoch to apply early stopping, the test set is evaluated only once after training to receive an overall score. The eventual score will be the average across both folds. Last, considering the HPO being another source of overfitting but for hyperparameters, another set of profiles is excluded from training and validation during the HPO, and is called the generalization set, see Fig. 7. Having found optimal hyperparameters, validating a model against this last set will give true generalization errors. Specifically, profile numbers $\mathcal{V}_1 = \{4, 67, 71, 78\}$ (9.4 h), and $\mathcal{V}_2 = \{10, 48, 63\}$ (7.5 h) act as validation sets, while $\mathcal{G} = \{60, 62, 74\}$ (8.9 h) is the held-out generalization set. This cross-validation scheme can be referred to as an overlapping two-fold CV with a held-out set for the evaluation of the hyperparameter optimum's true generalization capability.

Furthermore, training outcome of ANNs is sensitive to their random initialization. The same set of hyperparameters might end training at very different performance values depending on the first weight values being fortunate or not. For this reason, during HPO, each training fold is repeated ten times with different random seeds and the score is then averaged across in order to assess the mean quality of a certain hyperparameter set.

3.1.3. Search results

The HPO's sampling strategy is that of Optuna's tree-structured Parzen estimator (TPE) (Akiba et al., 2019; Bergstra et al., 2011). Hyperparameter sets were independently fetched and evaluated in parallel by 60–80 workers on a high-performance computing cluster. The HPO error trend is shown in Fig. 8. The overall best hyperparameter set is reported in Table 2, and its performance on the generalization set is displayed in Fig. 9.

Comparisons to a hyperparameter-optimized MLP, another optimized 1-D convolutional neural network (CNN), a likewise optimized LPTN, and a simple ordinary least squares (OLS) algorithm can be found in Table 3. All numbers are compiled from an evaluation on the generalization set. Note that for pure black-box algorithms the increased feature entries are to be computed in real-time as well rather than being measured, which, next to the model size, represents an additional computational demand on the embedded system.

Table 3

State-of-the-art thermal modeling performance metrics as evaluated on the generalization set.

	MSE (K ²)	ℓ_∞ (K)	LUT entries	Feature entries	Model size	FLOPs
MLP (Kirchgässner et al., 2021b)	5.58	14.29	0	81	1380	2720
OLS (Kirchgässner et al., 2021b)	4.47	9.85	0	81	328	648
CNN (Kirchgässner et al., 2020)	4.43	15.54	0	81	4916	> 70k
LPTN (Wallscheid and Böcker, 2016)	3.64	7.37	12810	4	34	124
TNN (small)	3.18	5.84	0	5	64	149
TNN (HPO optimum)	2.87	6.02	0	5	1525	2816

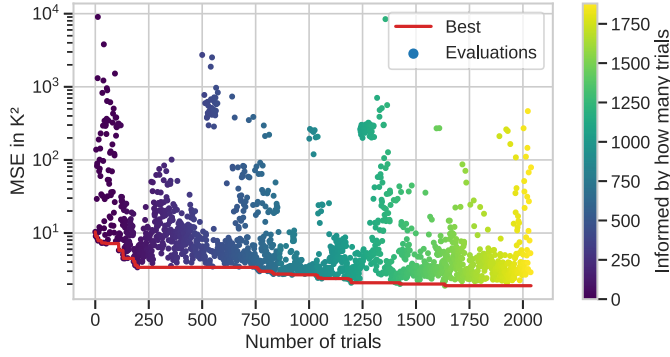


Fig. 8. The HPO error trend over trials on the electric motor dataset is shown. Due to parallel execution, subsequent evaluations are not necessarily strictly-linearly informed by previous evaluations. Each trial denotes the average across ten random seeds on each of two test folds. The best MSE found is 1.9 K^2 .

3.2. Model size Pareto front

The foregoing HPO was guided by estimation accuracy alone, without considering model size. However, in real-world applications, real-time capability and, thus, small, lightweight models play an important role as they are to be deployed prevalently on best-cost hardware. In the following, a random search over a grid of a TNN's topological settings is conducted, keeping other hyperparameters at the optimum, in order to shape an informative Pareto front. This shall give an overview of

the trade-off between model size and estimation accuracy, see Fig. 10. The amount of layers and the number of units per layer for both, γ and π , vary from 1 to 3 and 1 to 128, respectively, and independently from each other. The number of units is varied exponentially in order to bound the combinatorial grid size growth.

It is evident that for very small model sizes the accuracy degrades only insignificantly. Thus, the found hyperparameters reveal a robust guidance on the design of TNNs that are less influenced by the model size. A topology with 2 neurons in an intermediate layer for both thermal parameter approximators, which results in just below 100 model parameters, seems to be a good trade-off between accuracy and model size.

It might appear incoherent that the kernel-based HPO optimum does not touch the Pareto front of the grid search. This has two simple reasons: First, the foregoing HPO had to sample from a substantially larger sample space of 26 hyperparameters, while the grid search sampled from just two, such that the over 2000 candidates could have easily missed better local optima. Second, the TPE sampling method was guided by the averaged estimation accuracy on two test sets, that might expose different peculiarities than the generalization set.

3.3. Thermal conductance sparsification

Having iterated over many TNN design parameters, a vast source for conductance analysis is available. In an attempt to further reduce the model size, together with the fact that not all components within an intricate system necessarily offer direct heat paths to each other, the median activation of each conductance is investigated. Fig. 11 shows an

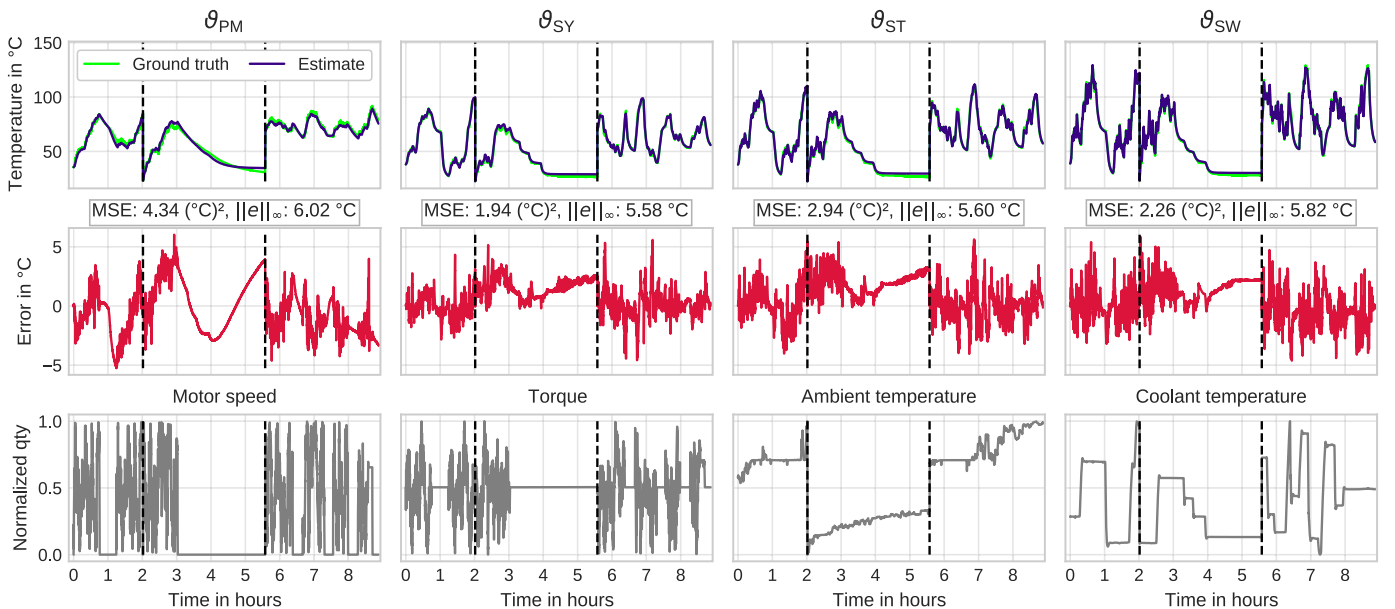


Fig. 9. The performance of the found optimum on the generalization set is depicted. Dashed vertical lines separate independent profiles. Some selected sensor quantities are depicted at the bottom for reference.

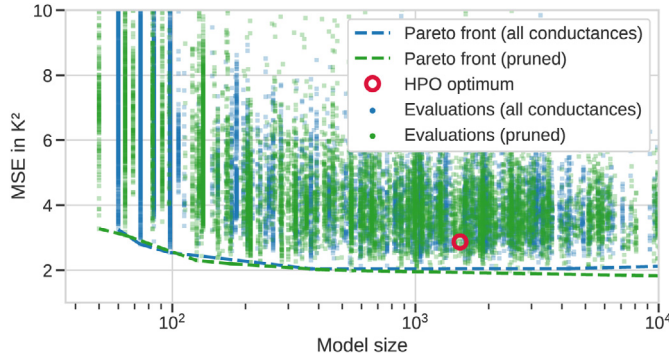


Fig. 10. The MSE on the generalization set over different topological settings is illustrated. Here, $L_{(\pi, \gamma)}$ and $n_{(\pi, \gamma)}^{(i)}$ are independently sampled for both, γ and π .

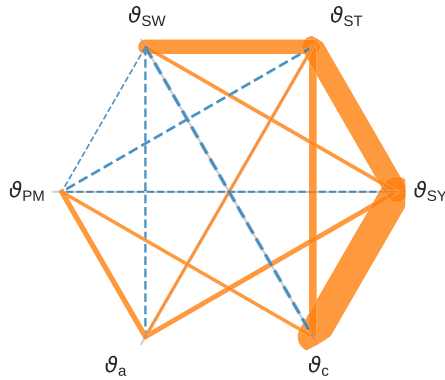


Fig. 11. The median conductance values as being output on random input, averaged over γ networks derived from all successful experiments (i.e., $MSE < 5 K^2$) are visualized. Line thickness scales with conductance values linearly. Conductances under a certain threshold are drawn in blue, dashed lines, and are subject to pruning.

inter-conductance plot between all available temperatures, where line widths denote the median across a subset of all compiled γ networks' median output, when exposed to a random uniform input $\phi \in \mathbb{R}^{n+\sigma}$ with intervals $[0, 1.3]$. Only those models that achieved an average MSE of below $5 K^2$ across the components are considered. Those connections illustrated in blue, dashed lines are candidates for pruning.

It becomes apparent, that the highest thermal conductances exist between the coolant and stator temperatures, peaking between the stator yoke and coolant. This is physically plausible as those two components are spatially attached and form a substantial, convective heat transfer (see Fig. 5). The weakest connections exist for the ambient and rotor temperature to all other, likewise. The marginally significant ambient temperature conductance can be explained by its remote operating area. However, the small conductances around the permanent magnet temperatures are counter-intuitive at first, but become more plausible under the fact that temperatures in the rotor are more power-loss-driven, and, thus, rather determined by internal heat generation than by conductive or convective heat transfer. This hypothesis, if it were not verified in literature yet, is further corroborated by this temperature's small time constant — best observed during cooling phases (refer to Fig. 9).

The conductance analysis represents one of many benefits a TNN exhibits. Without prior knowledge of the geometry or material in the system, the TNN learns the physical thermal relationship between measured nodes solely from data.

With the inferred conductance relevance, those under a certain threshold are pruned thereafter, and another random search over topological settings is conducted and shown in Fig. 10. The minimum

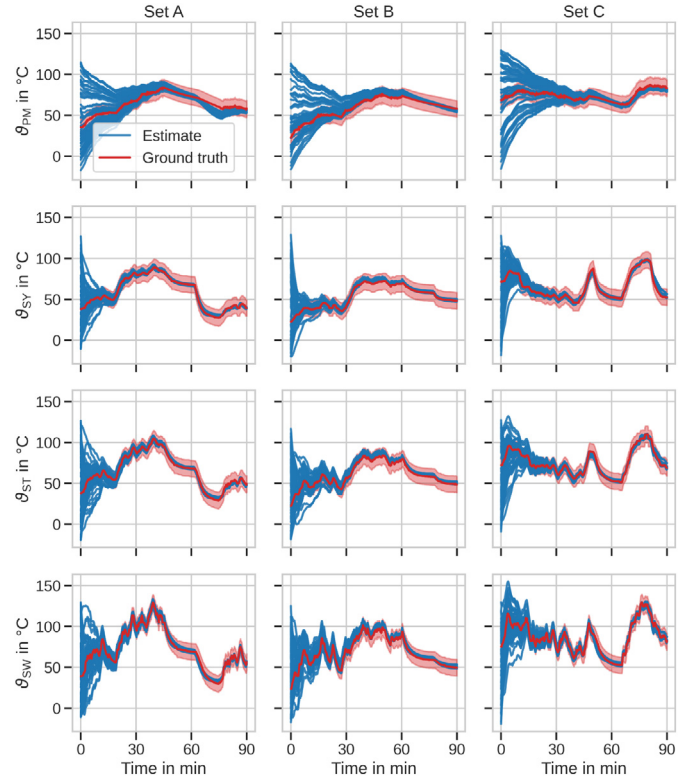


Fig. 12. The impact of misconfigured initial conditions on estimation trajectories is demonstrated. The ground truth trajectory is accompanied by a red-shaded $\pm 10^\circ C$ error band.

amount of model parameters reduces from 60 to 50 due to conductance pruning, which is the case for one hidden layer of one neuron before each of the output layers of γ and π . Note that this can be further reduced through feature selection, which is out of this work's scope. Keeping the model size constant, sparsification also has a beneficial effect on accuracy, because more parameters can be used to model fewer nonlinear functions.

3.4. Detuned initial conditions

Another advantage of TNNs is the possibility to set the initial condition through the cell state, which is not straight forward for usual machine learning methods. Determining the initial condition is trivial in simulations and approximately equal to the ambient temperature in real-world scenarios. However, if this initial guess is inaccurate, how long does it take to recover to a decent estimate?

Fig. 12 exemplifies different scenarios with varying detuned initial estimates. It becomes evident that the worst-case estimate recovers to a $\pm 10^\circ C$ error band (red-shaded area) in under 30 min for the stator temperatures, and in under 45 min for the rotor temperature. The higher the thermal capacitance of a component, the slower the convergence is to be expected. This is acceptable for most applications considering the low risk of a flawed initial condition approximation. Similar observations can be made for pure LPTN models.

Finally, it can be noted that a TNN, being quasi-linear in its state-space representation, lends itself as internal model for state observers (Kalman, Luenberger) as soon as parts of the target temperatures are (virtually) measured in a field application (Wallscheid and Böcker, 2017; Gaona et al., 2020). Standard observer types for linear systems would already suffice while the internal system model is updated by the TNN. Given the external feedback, deviations during estimation and at initialization could be reduced substantially and more rapidly.

4. Conclusion

A hybrid thermal model for real-time temperature estimation, the so-called thermal neural network, is presented in this work, which exhibits the benefits of both worlds — consolidated knowledge on heat transfer phenomena and nonlinear, data-driven function approximation with machine learning tools. Due to a TNN's outer LPTN structure, which lifts the black-box model to a quasi-LPV model, it is applicable everywhere an LPTN would be seen appropriate, but with the additional adaptivity to parasitic real-world effects as observed in measurement data. For the same reason, initial conditions can be set, and model states are physically interpretable, giving rise to conclusions over the thermal characteristics of the measured system. Being a composite of three general-purpose function approximators, a TNN is end-to-end differentiable such that modern automatic differentiation frameworks are employable. Neither expert knowledge nor motor sheet data is necessary to design a TNN, but they can help streamlining its structure. Having these properties combined in one model is unique throughout the thermal modeling literature to the authors' best knowledge.

One disadvantage a TNN shares with common machine learning approaches is the dependence on measurement data, although transferring knowledge from one trained TNN model to another likewise manufactured system is indeed possible. Moreover, data is more and more available even in power system domains nowadays. However, the transfer performance on different motors of the same production line remains unanswered without the corresponding data from a mass-produced motor series being available.

Future investigations shall consider applying the UDE and SSNN principle to other domains of energy conversion where latent states are estimated in real time, e.g., battery systems' state of charge, or induction motors' differential inductances and magnetic flux linkage. Applications to reinforcement learning controllers – especially in power systems – are obvious and promising as well.

CRedit authorship contribution statement

Wilhelm Kirchgässner: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Oliver Wallscheid:** Conceptualization, Investigation, Resources, Data curation, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Joachim Böcker:** Supervision, Resources, Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Wilhelm Kirchgässner reports financial support was provided by German Research Foundation. Wilhelm Kirchgässner reports a relationship with Paderborn University that includes: employment.

Data availability

Data is uploaded at kaggle.com and referenced in the text, same as code at github.com.

Acknowledgments

This work is funded by the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG), Germany under grant BO 2535/15-1. Computing time was provided by the Paderborn Center for Parallel Computing (PC²).

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al., 2016. Tensorflow: A System for Large-Scale Machine Learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation. OSDI 16, pp. 265–283, URL <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>.
- Abbas, H., Werner, H., 2008. Polytopic Quasi-LPV Models Based on Neural State-Space Models and Application to Air Charge Control of a SI Engine. IFAC Proc. Vol. 41 (2), 6466–6471, URL <https://www.sciencedirect.com/science/article/pii/S1474667016399827>.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A Next-Generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
- Andre, D., Nuhic, A., Soczka-Guth, T., Sauer, D.U., 2013. Comparative Study of a Structured Neural Network and an Extended Kalman Filter for State of Health Determination of Lithium-ion Batteries in Hybrid Electric Vehicles. Eng. Appl. Artif. Intell. 26 (3), 951–961. <http://dx.doi.org/10.1016/j.engappai.2012.09.013>, URL <https://www.sciencedirect.com/science/article/pii/S0952197612002448>.
- Bahman, A.S., Ma, K., Ghimire, P., Iannuzzo, F., Blaabjerg, F., 2016. A 3-D-Lumped Thermal Network Model for Long-Term Load Profiles Analysis in High-Power IGBT Modules. IEEE J. Emerg. Sel. Top. Power Electron. 4 (3), 1050–1063. <http://dx.doi.org/10.1109/JESTPE.2016.2531631>, URL <https://ieeexplore.ieee.org/document/7412682>.
- Baydin, A.G., Pearlmutter, B.A., Radul, A., Siskind, J.M., 2018. Automatic Differentiation in Machine Learning: A Survey. J. Mach. Learn. Res. 18, 1–43, [arXiv: 1502.05767](https://arxiv.org/abs/1502.05767).
- Bergman, T.L., Incropera, F.P., DeWitt, D.P., Lavine, A.S., 2007. Fundamentals of Heat and Mass Transfer, 6th ed. John Wiley & Sons.
- Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for Hyper-Parameter Optimization. In: 25th Annual Conference on Neural Information Processing Systems, vol. 24. NIPS 2011.
- Boglietti, A., Carpaneto, E., Cossale, M., Vaschetto, S., 2016. Stator-Winding Thermal Models for Short-Time Thermal Transients: Definition and Validation. IEEE Trans. Ind. Electron. 63 (5), 2713–2721. <http://dx.doi.org/10.1109/TIE.2015.2511170>, URL <https://ieeexplore.ieee.org/document/7362212>.
- Boglietti, A., Cavagnino, A., Parvis, M., Vallan, A., 2006. Evaluation of Radiation Thermal Resistances in Industrial Motors. IEEE Trans. Ind. Appl. 42, N, 688–693.
- Boglietti, A., Cavagnino, A., Staton, D., Shanel, M., Mueller, M., Mejuto, C., 2009. Evolution and Modern Approaches for Thermal Analysis of Electrical Machines. IEEE Trans. Ind. Electron. 56 (3), 871–882. <http://dx.doi.org/10.1109/TIE.2008.2011622>, URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4796880>.
- Bracikowski, N., Hecquet, M., Brochet, P., Shirinskii, S.V., 2012. Multiphysics Modeling of a Permanent Magnet Synchronous Machine by Using Lumped Models. IEEE Trans. Ind. Electron. 59 (6), 2426–2437. <http://dx.doi.org/10.1109/TIE.2011.2169640>, URL <https://ieeexplore.ieee.org/document/6029330>.
- Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K., 2018. Neural Ordinary Differential Equations. In: Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc., pp. 6571–6583, URL <https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf>.
- Gaona, D., Wallscheid, O., Böcker, J., 2020. Improved Fusion of Permanent Magnet Temperature Estimation Techniques for Synchronous Motors Using a Kalman Filter. IEEE Trans. Ind. Electron. 67 (3), 1708–1717. <http://dx.doi.org/10.1109/TIE.2019.2905817>, URL <https://ieeexplore.ieee.org/abstract/document/8672935>.
- Garcia, P., Briz, F., Raca, D., Lorenz, R.D., 2007. Saliency-Tracking-Based Sensorless Control of AC Machines Using Structured Neural Networks. IEEE Trans. Ind. Appl. 43 (1), 77–86. <http://dx.doi.org/10.1109/TIA.2006.887309>, URL <https://ieeexplore.ieee.org/document/4077193>.
- Gedlu, E., Wallscheid, O., Böcker, J., 2020. Permanent Magnet Synchronous Machine Temperature Estimation using Low-Order Lumped-Parameter Thermal Network with Extended Iron Loss Model. In: The 10th International Conference on Power Electronics, Machines and Drives (PEMD 2020). pp. 937–942. <http://dx.doi.org/10.1049/icp.2021.1017>, URL <https://ieeexplore.ieee.org/document/9545383>.
- Gers, F.A., Cummins, F., 1999. Learning to Forget: Continual Prediction with LSTM. In: Ninth International Conference on Artificial Neural Networks, vol. 2. pp. 1–19. <http://dx.doi.org/10.1162/089976600300015015>.
- Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer Feedforward Networks are Universal Approximators. Neural Netw. 2 (5), 359–366, [arXiv:arXiv:1011.1669v3](https://arxiv.org/abs/1011.1669v3), 0893-6080/89.
- Howey, D.A., Childs, P.R.N., Holmes, A.S., 2012. Air-Gap Convection in Rotating Electrical Machines. IEEE Trans. Ind. Electron. 59, N, 1367–1375.
- Iachello, M., De Luca, V., Petrone, G., Testa, N., Fortuna, L., Cammarata, G., Graziani, S., Frasca, M., 2014. Lumped Parameter Modeling for Thermal Characterization of High-Power Modules. IEEE Trans. Components Packaging Manuf. Technol. 4 (10), 1613–1623. <http://dx.doi.org/10.1109/TCPMT.2014.2353695>, URL <https://ieeexplore.ieee.org/document/6902791>.
- Kirchgässner, W., Wallscheid, O., Böcker, J., 2019. Empirical Evaluation of Exponentially Weighted Moving Averages for Simple Linear Thermal Modeling of Permanent Magnet Synchronous Machines. In: Proceedings of the 28th International Symposium on Industrial Electronics. pp. 318–323. <http://dx.doi.org/10.1109/ISIE.2019.8781195>.

- Kirchgässner, W., Wallscheid, O., Böcker, J., 2020. Estimating Electric Motor Temperatures with Deep Residual Machine Learning. *IEEE Trans. Power Electron.* 36 (7), 7480–7488. <http://dx.doi.org/10.1109/TPEL.2020.3045596>.
- Kirchgässner, W., Wallscheid, O., Böcker, J., 2021a. Electric Motor Temperature Dataset. Kaggle, Paderborn, <http://dx.doi.org/10.34740/KAGGLE/DSV/2161054>, URL <https://www.kaggle.com/dsv/2161054>.
- Kirchgässner, W., Wallscheid, O., Böcker, J., 2021b. Data-Driven Permanent Magnet Temperature Estimation in Synchronous Motors with Supervised Machine Learning: A Benchmark. *IEEE Trans. Energy Convers.* 36 (3), 2059–2067. <http://dx.doi.org/10.1109/TEC.2021.3052546>, URL <https://ieeexplore.ieee.org/document/9328339>.
- Lee, J., Ha, J., 2020. Temperature Estimation of PMSM Using a Difference-Estimating Feedforward Neural Network. *IEEE Access* 8, 130855–130865. <http://dx.doi.org/10.1109/ACCESS.2020.3009503>, URL <https://ieeexplore.ieee.org/abstract/document/9141245>.
- Lim, C., Bumby, J., Dominy, R., Ingram, G., Mahkamov, K., Brown, N., Mebarki, A., Shanel, M., 2008. 2-D lumped-parameter thermal modelling of axial flux permanent magnet generators. In: 2008 18th International Conference on Electrical Machines. pp. 1–6. <http://dx.doi.org/10.1109/ICELMACH.2008.4799880>.
- Masi, F., Stefanou, I., Vannucci, P., Maffi-Berthier, V., 2021. Thermodynamics-based Artificial Neural Networks for Constitutive Modeling. *J. Mech. Phys. Solids* 147, 104277, URL <https://www.sciencedirect.com/science/article/pii/S0022509620304841>.
- Pascanu, R., Mikolov, T., Bengio, Y., 2012. Understanding the Exploding Gradient Problem. In: Proceedings of the 30th International Conference on Machine Learning. <http://dx.doi.org/10.1109/72.279181>, arXiv:arXiv:1211.5063v2.
- Qi, F., Schenk, M., De Doncker, R.W., 2014. Discussing Details of Lumped Parameter Thermal Modeling in Electrical Machines. In: 7th IET International Conference on Power Electronics, Machines and Drives. PEMD 2014, pp. 1–6. <http://dx.doi.org/10.1049/cp.2014.0479>, URL <https://ieeexplore.ieee.org/document/6836889>.
- Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., 2020. Universal Differential Equations for Scientific Machine Learning. arXiv:2001.04385.
- Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* 378, 686–707. <http://dx.doi.org/10.1016/j.jcp.2018.10.045>, URL <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- Reigosa, D., Fernández, D., Martínez, M., Guerrero, J.M., Diez, A.B., Briz, F., 2019. Magnet Temperature Estimation in Permanent Magnet Synchronous Machines Using the High Frequency Inductance. *IEEE Trans. Ind. Appl.* 55 (3), 2750–2757. <http://dx.doi.org/10.1109/TIA.2019.2895557>.
- Reigosa, D.D., Fernandez, D., Yoshida, H., Kato, T., Briz, F., 2015. Permanent-Magnet Temperature Estimation in PMSMs Using Pulsating High-Frequency Current Injection. *IEEE Trans. Ind. Appl.* 51 (4), 3159–3168. <http://dx.doi.org/10.1109/TIA.2015.2404922>, URL <https://ieeexplore.ieee.org/document/7046399>.
- Rivals, I., Personnaz, L., 1996. Black-Box Modeling With State-Space Neural Networks. In: Neural Adaptive Control Technology. World Scientific, pp. 237–264, URL https://www.neurones.espci.fr/Articles/_PS/NACT1.pdf.
- Rostami, N., Feyzi, M.R., Pyrhonen, J., Parviainen, A., Niemela, M., 2013. Lumped-parameter thermal model for axial flux permanent magnet machines. *IEEE Trans. Magn.* 49 (3), 1178–1184. <http://dx.doi.org/10.1109/TMAG.2012.2210051>.
- Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N., 2017. Data-Driven Discovery of Partial Differential Equations. *Sci. Adv.* 3 (4), e1602614. <http://dx.doi.org/10.1126/sciadv.1602614>, arXiv:1609.06401.
- Specht, A., Wallscheid, O., Böcker, J., 2014. Determination of Rotor Temperature for an Interior Permanent Magnet Synchronous Machine Using a Precise Flux Observer. In: International Power Electronics Conference. pp. 1501–1507. <http://dx.doi.org/10.1109/IPEC.2014.6869784>.
- Wallscheid, O., 2021. Thermal Monitoring of Electric Motors: State-of-the-Art Review and Future Challenges. *IEEE Open J. Ind. Appl.* 2, 204–223. <http://dx.doi.org/10.1109/OJIA.2021.3091870>, URL <https://ieeexplore.ieee.org/abstract/document/9463739>.
- Wallscheid, O., Böcker, J., 2016. Global Identification of a Low-Order Lumped-Parameter Thermal Network for Permanent Magnet Synchronous Motors. *IEEE Trans. Energy Convers.* 31 (1), 354–365. <http://dx.doi.org/10.1109/TEC.2015.2473673>.
- Wallscheid, O., Böcker, J., 2017. Fusion of Direct and Indirect Temperature Estimation Techniques for Permanent Magnet Synchronous Motors. In: 2017 IEEE International Electric Machines and Drives Conference. IEMDC, pp. 1–8. <http://dx.doi.org/10.1109/IEMDC.2017.8002038>, URL <https://ieeexplore.ieee.org/abstract/document/8002038>.
- Wallscheid, O., Huber, T., Peters, W., Böcker, J., 2016. A Critical Review of Techniques to Determine the Magnet Temperature of Permanent Magnet Synchronous Motors Under Real-Time Conditions. *EPE J.* 26, 1–10. <http://dx.doi.org/10.1080/09398368.2016.1209877>, URL <https://www.tandfonline.com/doi/full/10.1080/09398368.2016.1209877>.
- Williams, R.J., Peng, J., 1990. An Efficient Gradient-Based Algorithm for On-line Training of Recurrent Network Trajectories. *Neural Comput.* 2 (4), 490–501, URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.56.7941&rep=rep1&type=pdf>.
- Wöckinger, D., Bramerdorfer, G., Drexler, S., Vaschetto, S., Cavignino, A., Tenconi, A., Amrhein, W., Jeske, F., 2020. Measurement-Based Optimization of Thermal Networks for Temperature Monitoring of Outer Rotor PM Machines. In: 2020 IEEE Energy Conversion Congress and Exposition. ECCE, pp. 4261–4268. <http://dx.doi.org/10.1109/ECCE44975.2020.9236388>, URL <https://ieeexplore.ieee.org/document/9236388>.
- Zamarreño, J.M., Vega, P., 1998. State Space Neural Network. Properties and Application. *Neural Netw.* 11 (6), 1099–1112. [http://dx.doi.org/10.1016/S0893-6080\(98\)00074-4](http://dx.doi.org/10.1016/S0893-6080(98)00074-4), URL <https://www.sciencedirect.com/science/article/pii/S0893608098000744>.
- Zhang, K., Guliani, A., Ogren-Memik, S., Memik, G., Yoshii, K., Sankaran, R., Beckman, P., 2018. Machine Learning-Based Temperature Prediction for Runtime Thermal Management Across System Components. *IEEE Trans. Parallel Distrib. Syst.* 29 (2), 405–419. <http://dx.doi.org/10.1109/TPDS.2017.2732951>, URL <https://ieeexplore.ieee.org/document/7995115>.