Article

Not peer-reviewed version

Incorporating a Deep Neural Network into Moving Horizon Estimation for Embedded Thermal Torque Derating of an Electric Machine

Alexander Emil Klaus Winkler *

Posted Date: 21 March 2025

doi: 10.20944/preprints202503.1638.v1

Keywords: state estimation; deep learning; moving horizon estimation; nonlinear and optimal automotive control; neural networks; temperature control; electric machine

Article

Incorporating a Deep Neural Network into Moving Horizon Estimation for Embedded Thermal Torque Derating of an Electric Machine

Alexander Winkler [1], Pranav Shah [2], Katrin Baumgärtner [3], Vasu Sharma [1], David Gordon [3] and Jakob Andert [1,*]

[1] Teaching and Research Area Mechatronics in Mobile Propulsion, RWTH Aachen University; Forckenbeckstr. 4, 52074 Aachen, Germany
[2] IMTEK—Department Of Microsystems, University of Freiburg, Georges-Köhler-Allee 103, 79108 Freiburg im Breisgau, Germany
[3] Department of Mechanical Engineering, University of Alberta, 10th Floor, Donadeo Innovation Centre for Engineering, Edmonton, AB T6G 1H9, Canada
* Correspondence: andert_j@mmp.rwth-aachen.de

Abstract: This study introduces a novel state estimation framework that incorporates Deep Neural Networks (DNNs) into Moving Horizon Estimation (MHE), shifting from traditional physics-based models to rapidly developed data-driven techniques. A DNN model with Long Short-Term Memory (LSTM) nodes is trained on synthetic data generated by a high-fidelity thermal model of a Permanent Magnet Synchronous Machine (PMSM), which undergoes thermal derating as part of the torque control strategy in a battery electric vehicle. The MHE is constructed by integrating the trained DNN with a simplified driving dynamics model in a discrete-time formulation, incorporating the LSTM hidden and cell states in the state vector to retain system dynamics. The resulting optimal control problem (OCP) is formulated as a nonlinear program (NLP) and implemented using the `acados` framework. Model-in-the-loop (MiL) simulations demonstrate accurate temperature estimation, even under noisy sensor conditions or failures. Achieving threefold real-time capability on embedded hardware confirms the feasibility of the approach for practical deployment. The primary focus of this study is to assess the feasibility of the MHE framework using a DNN-based plant model instead of focusing on quantitative comparisons of vehicle performance. Overall, this research highlights the potential of DNN-based MHE for real-time, safety-critical applications by combining the strengths of model-based and data-driven methods.

Keywords: state estimation; deep learning; moving horizon estimation; nonlinear and optimal automotive control; neural networks; temperature control; electric machine

1. Introduction & Background

State estimation has been a critical area of research for several decades, playing a fundamental role in various engineering disciplines by providing robust and reliable feedback for control systems [1]. The mathematical foundations of state estimation can be traced back to Carl Gauss in the early 1800s, with further advancements in the 20th century, including Maximum Likelihood Estimation [2] and Linear Minimum Mean-Square Estimation [3]. However, the applicability of these early formulations to real-time systems was limited by available computational resources. The introduction of dynamic state estimation techniques, such as the Kalman Filter and the Luenberger Observer, revolutionized the field by enabling online estimation based on system dynamics and measurement uncertainty [4]. As the focus of both academia and industry moved to non-linear systems, extensions such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) were developed to address non-linearities in state estimation [5,6]. More recently, research has focused on state estimation techniques that can incorporate constraints on states and parameters, like Moving Horizon Estimation (MHE) [7].

The main idea of MHE is to compute a state estimate x by solving an optimization problem over a finite time horizon. At each sample time as soon as a measurement y is made available the horizon is shifted ahead in time in a receding horizon fashion and a new estimate is calculated. Figure 1 illustrates the MHE's operating principle using the receding horizon.

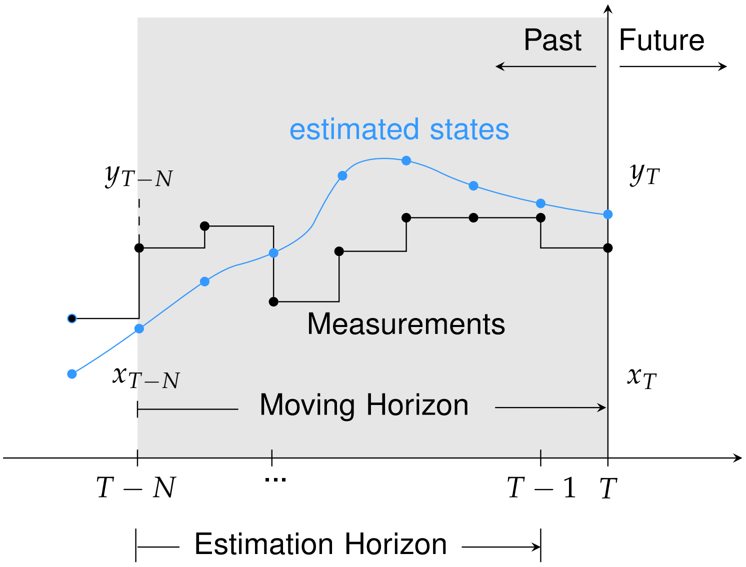


Figure 1. Schematic of moving horizon estimation problem.

Compared to Kalman-based approaches, MHE offers improved handling of constraints, disturbances, and system non-linearities, making it a powerful tool for advanced estimation tasks [8].

A critical aspect of modern state estimation is data assimilation, which integrates sensor measurements with dynamic process models to improve the accuracy of the estimation [9]. Traditional approaches rely on physics-based models, which require detailed system knowledge and intricate mathematical formulations. However, as systems become increasingly complex, the development of precise mathematical models becomes a challenge. Besides, accurate models might be computationally too expensive in particular for real-time applications [10]. These models often involve high-dimensional equations that demand significant computational resources, making them less practical for embedded control systems with stringent latency constraints. To address these challenges, data-driven approaches such as system identification, machine learning, and deep learning have gained prominence [11].

Deep learning architectures, inspired by biological neural networks, progressively build representations of input data while minimizing the need for manual feature engineering [12,13]. The layers in a Deep Neural Network (DNN) consist of neurons that integrate weighted inputs via activation functions. This function introduces non-linearity that is essential for learning complex tasks. While the basic calculations within the neurons remain consistent, the arrangement of the neurons play a crucial role in how the information is processed giving rise to various neural network architectures. Among these, Recurrent Neural Networks (RNNs) are particularly suited for dynamic systems due to their ability to retain temporal information [14]. Long Short-Term Memory (LSTM) networks, a variant of RNNs, are especially effective for time-series modeling, as they mitigate vanishing gradient issues and capture long-term dependencies [15].

The rapid advancement of DNNs has enabled the modeling of complex non-linear relationships without explicit system equations, making them invaluable for state estimation in systems where deriving accurate mathematical models is challenging. DNN-based models provide a scalable alternative by learning system behaviour directly from data, eliminating the need for complex physics-based formulations and enabling efficient real-time deployment. Several studies have successfully integrated DNNs into state estimation frameworks, with applications in pseudo-measurement generation, parameterized EKF formulations, and RNN-based non-linear system identification [16–18]. Notably, the use of LSTM and DNN-based models has also been explored in control applications in previous works, such as model predictive control (MPC) for combustion engines [19], and rapid development toolchains for a low temperature combustion process [20]. However, DNN-based MHE for state estimation remains unexplored. This research addresses that gap by leveraging LSTM-based models to enhance adaptability and robustness in state estimation.

The novelty of this research lies in developing a MHE that integrates a DNN-based plant model — specifically using LSTM architecture — in place of a white- or grey-box model, aiming to improve estimation accuracy while reducing modeling effort and computational complexity.

Building on prior work by [21] that introduced a DNN-based Model Predictive Controller (MPC) for thermal torque derating of a Permanent Magnet Synchronous Machine (PMSM) in a Battery Electric Vehicle (BEV), this research extends the work by integrating a DNN-based state estimator [21,22]. Thermal derating is a protective strategy that limits the torque output of an electric machine to prevent overheating and ensure long-term reliability. The previous study focused on optimizing torque commands while ensuring temperature constraints of the PMSM, using a Model-in-the-Loop (MiL) simulation with a reference velocity trajectory from one lap of Nürburgring Nordschleife.

A high-fidelity 70-node Lumped Parameter Thermal Network (LPTN) model provided by DENSO, the PMSM's manufacturer, has been experimentally validated on test benches and used to determine the motor temperature for feedback to the MPC within the plant. However, in real-world applications, temperature data obtained from sensors is often noisy and inaccurate, highlighting the necessity for robust state estimation. To validate the proposed approach, the DNN-based state estimator is integrated into the existing MiL simulation setup. The estimator processes noisy sensor measurements to reconstruct accurate system states, which are then provided to the MPC for real-time process control. The state estimator is formulated as a MHE problem and is implemented using the acados optimal control framework, ensuring compatibility with embedded systems.

To provide a high-level understanding of the approach and methodology, the graphical abstract in Figure 2 illustrates the key components of the proposed MHE framework, including data generation, DNN training, and its simulative and embedded validation. The central objective of this research is to demonstrate the feasibility of a DNN-based MHE framework. Instead of focusing on quantitative comparisons of vehicle performance under MPC control, the study a qualitative evaluation — assessing how effectively the DNN-based MHE estimates temperature states and how robust it performs in the presence of injected sensor faults.

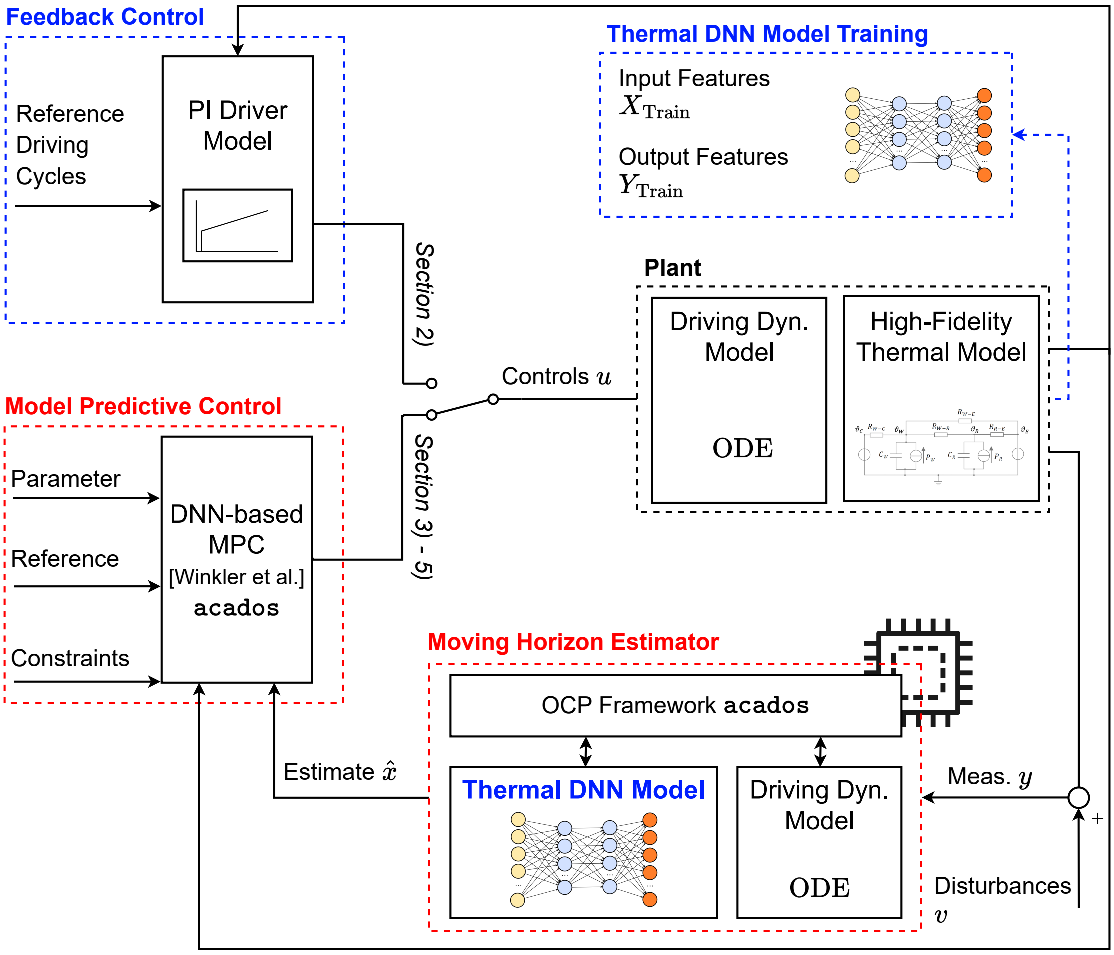


Figure 2. Graphical abstract.

The paper is divided into the data generation process (Section 2), the MHE development (Section 3), and the subsequent simulative and embedded integration for validation (Section 4 and Section 5, respectively) as shown schematically in Figure 2.

This research makes the following key contributions:

1. **Integration of DNN with MHE:** This work represents one of the first attempts to integrate a DNN-based model with MHE, providing a novel framework for state estimation.
2. **Deployment and validation on embedded systems:** The feasibility of implementing a DNN-based MHE framework in real-time is demonstrated. This research is also one of the first to test and validate the deployment of DNN-based state estimation on real-time hardware using the `acados` optimal control framework [23].

The datasets and scripts presented in this work are publicly available [24].

2. Deep Neural Network Modeling

This chapter presents the development of the DNN model used for thermal state prediction. It covers the LSTM architecture, the experimental setup with synthetic data generation, and the training process of the network.

2.1. Long Short-Term Memory Network

LSTMs, one of the most popular variants of RNN, have the ability to maintain a form of memory, enabling them to influence current inputs and outputs based on past sequence information [14]. To regulate the flow of data, LSTMs introduce memory cells and gate units [15]. The core computational unit of the LSTM network is called the memory cell (or simply "cell"), and these networks were primarily designed for sequence modeling while addressing the vanishing gradient problem [14,25].

Each LSTM cell consists of three gates: the forget gate, the input gate and the output gate that regulate information flow using sigmoid (σ) and hyperbolic tangent ($\tanh$) activation functions to maintain stable outputs [14]. The following equations mathematically define the operations within an LSTM cell, where each gate contributes to memory updates and output generation:

$$i_k = \sigma\left(W_{\mathrm{u,i}}^T u_k + W_{\mathrm{h,i}}^T h_{k-1} + b_{\mathrm{i}}\right) , \tag{1a}$$

$$f_k = \sigma\left(W_{\mathrm{u,f}}^T u_k + W_{\mathrm{h,f}}^T h_{k-1} + b_{\mathrm{f}}\right) , \tag{1b}$$

$$g_k = \tanh\left(W_{\mathrm{u,g}}^T u_k + W_{\mathrm{h,g}}^T h_{k-1} + b_{\mathrm{g}}\right) , \tag{1c}$$

$$o_k = \sigma\left(W_{\mathrm{u,o}}^T u_k + W_{\mathrm{h,o}}^T h_{k-1} + b_{\mathrm{o}}\right) , \tag{1d}$$

$$c_k = f_k \odot c_{k-1} + i_k \odot g_k , \tag{1e}$$

$$h_k = o_k \odot \tanh(c_k) , \tag{1f}$$

where $W_{\mathrm{u,(f,g,i,o)}}$ are the weighting matrices applied to the input vector u_k. $W_{\mathrm{h,(f,g,i,o)}}$ are weight matrices of the previous hidden state h_{k-1}. In this equation, $\odot$, is an element-wise multiplication and $b_{(f,g,i,o)}$ are the biases. i_k is the input gate, f_k is the forget gate, g_k is the cell candidate, o_k is the output gate, c_k is the cell state, and h_k is the hidden state. Two activation functions are used in Equation (1) which are given as:

1. $\tanh(z)$ hyperbolic tangent activation function:

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1} , \tag{2}$$

2. $\sigma(z)$ sigmoid activation function:

$$\sigma(z) = \frac{1}{1 + e^{-z}} . \tag{3}$$

These activation functions are used to introduce non-linearity into the otherwise linear layers. Figure 3 shows a visualization of the information flow depicted in Equation (1).

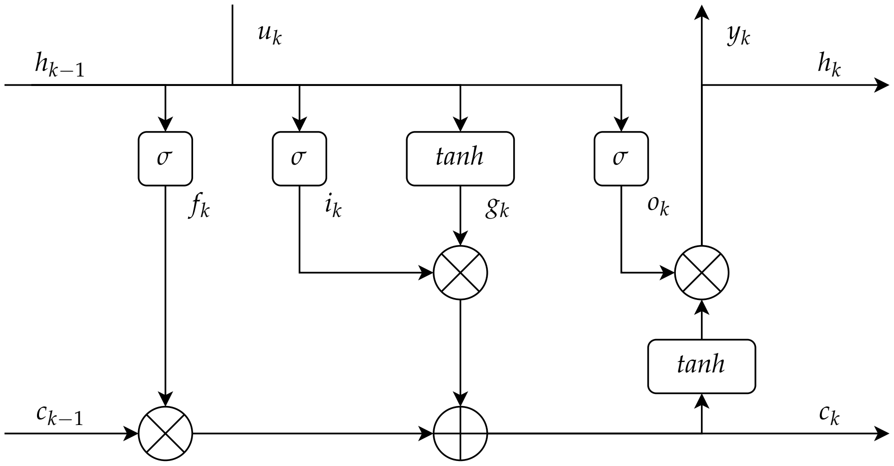


Figure 3. Long Short-Term Memory Unit. $\oplus$ Element-wise Addition, $\otimes$ Element-wise Multiplication, u: Input, y: Output, c: Cell State, h: Hidden State, σ: Sigmoid Function, $tanh$: Hyperbolic Tangent Function.

Due to the temporal dependencies in LSTMs, storing both the cell and hidden states across timesteps increases memory requirements.

2.2. Experimental Setup and Data Generation

The proposed state estimator, akin to the approach of previous works in [21] uses one-dimensional white-box driving dynamics and thermal black-box DNN dynamics to predict the thermal states of a PMSM and generate accurate and robust estimates [21,22]. The DNN, integrated with the state estimator, predicts the temperature gradients of the PMSM to calculate accurate thermal states of the system. For simplification, the PMSM is represented by two thermal masses $(\theta_{\mathrm{w}}, \theta_{\mathrm{r}})$ corresponding to measurable real-world temperatures at the test-bench of the windings and the rotor, respectively.

When combined with the MPC framework developed by [21], the resulting model-in-the-loop (MiL) simulation enables safe thermal derating of a BEV, by effectively handling noisy PMSM temperature measurements while ensuring compliance with thermal constraints.

The performance and accuracy of the DNN models heavily depend on the quality and quantity of training data. To efficiently generate the required data without relying on extensive test-bench experiments, a simulation framework is employed. Within this framework, a proportional-integral (PI) driver controller computes the torque requests to the Electric Machine (EM) and friction brake based on the vehicle velocity v. These include $(T_{\mathrm{EM,acc}}, T_{\mathrm{EM,brk}}, T_{\mathrm{fric,brk}})$, where the torque of the EM (T_{EM}) is split into acceleration and braking components: $T_{\mathrm{EM}} = T_{\mathrm{EM,acc}} + T_{\mathrm{EM,brk}}$. A one-dimensional longitudinal model utilizes these torque values to calculate vehicle velocity v_{veh} utilizing the ordinary differential equation (ODE):

$$\begin{aligned}\dot{v}_{\mathrm{veh}} = m_{\mathrm{veh}}^{-1} \cdot \big((T_{\mathrm{EM,acc}} + T_{\mathrm{EM,brk}} + T_{\mathrm{fric,brk}}) \cdot i_{\mathrm{diff}} \cdot r_{\mathrm{dyn}}^{-1} - m_{\mathrm{veh}} \cdot g \cdot \sin(\phi) \\ - 0.5 \cdot c_{\mathrm{d}} \cdot A_{\mathrm{c}} \cdot \rho \cdot v_{\mathrm{veh}}^2 - m_{\mathrm{veh}} \cdot g \cdot \cos(\phi) \cdot c_{\mathrm{r}} \big) \,,\end{aligned} \tag{4}$$

where $m_{\mathrm{veh}} = 1160$ kg is the vehicle mass (including the driver), $c_{\mathrm{d}} = 0.32$ is the drag coefficient, $c_{\mathrm{r}} = 0.011$ is the rolling friction coefficient, $r_{\mathrm{dyn}} = 0.293$ m is the dynamic tire radius, and $A_{\mathrm{c}} = 2.21\ m^2$ is the car's cross sectional area. The transmission ratio is given by $i_{\mathrm{diff}} = 9.3$, while the maximum vehicle speed is $v_{\mathrm{max}} = 130$ km/h. ϕ depicts an external parameter and input as the road inclination defined by the driving cycle, while ρ and g are the air density and gravitational acceleration, respectively. In this study, a typical minicar BEV is used and its parameters have been validated in previous works by [22].

The relation between the vehicle speed v in Equation (4) and the rotational speed of the electric machine n_{EM} is as follows: $n_{\mathrm{EM}} = (30 \cdot v_{\mathrm{veh}} \cdot i_{\mathrm{diff}})/(\pi \cdot r_{\mathrm{dyn}})$. Thus, the primary operating point of the EM, the torque T_{EM} and the rotational speed n_{EM} serve as the primary inputs to the high-fidelity LPTN model, determining the rotor and winding temperature $(\theta_{\mathrm{w}}, \theta_{\mathrm{r}})$ of the machine, with their derivatives $(\dot{\theta}_w, \dot{\theta}_r)$ being monitored as well. The 70-node high-fidelity LPTN model is provided by the PMSM's manufacturer DENSO and its parameters have been fitted using extensive experimental test-bench data. The full simulation model, implemented in MATLAB/Simulink is depicted in Figure 4 and operates at a sample rate of $100\ Hz$.

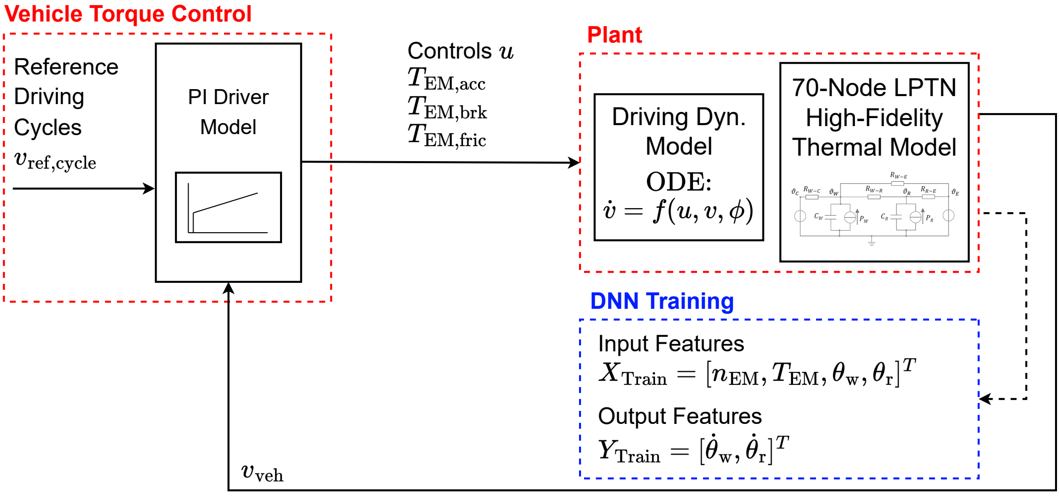


Figure 4. Simulation and model setup for training data generation.

A driving cycle serves as the primary predefined input for the data generation model, providing the reference velocity v_{ref} for the PI controller and the respective road inclination ϕ. To achieve higher and thus safety critical PMSM temperatures, the WLTP Class 3 driving cycle is customized, deliberately influencing the data diatribution. Figure 5 illustrates the velocity profile of the customized cycle and the corresponding temperature response, where only the winding temperature θ_{w} is shown, as the rotor temperature remains below critical levels and is therefore omitted. The figure also highlights that a significant portion of the data lies within the safety-critical temperature range of the machine, between 150 $^{\circ}C$. and 160 $^{\circ}C$, where permanent damage can occur.

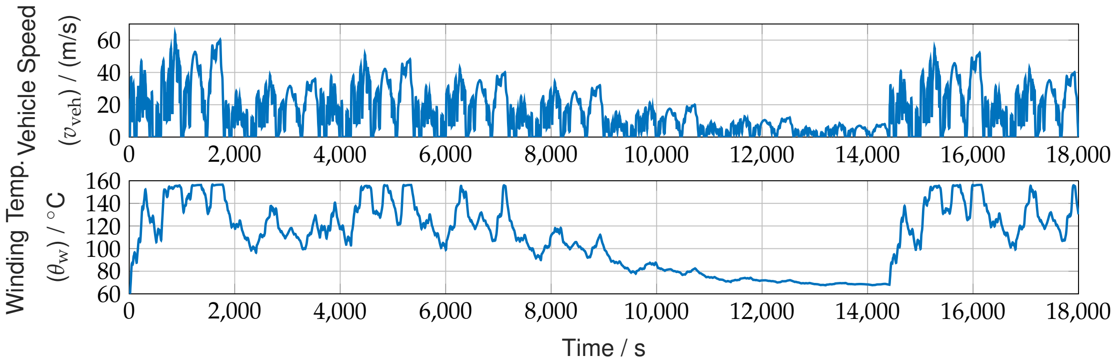


Figure 5. Synthetic data generation for vehicle speed and electric machine winding temperature, utilizing multiple drive cycles.

2.3. Neural Network Training

The network architecture comprises four layers: the input, LSTM, Fully Connected (FC), and output layer. The LSTM layer consists of 8 LSTM cells that compute the temporal dependencies between the inputs. Figure 6 shows the architecture of the DNN used for training.

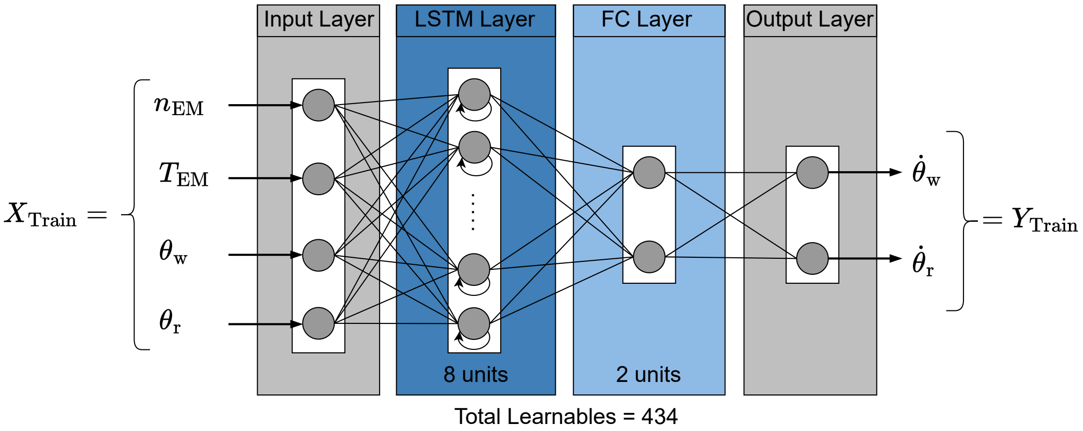


Figure 6. LSTM artificial neural network architecture using an FC layer. LSTM: Long Short-Term Memory, FC: Fully Connected.

Drawing inspiration from LPTNs and the physical intuition behind heat transfer, this work models temperature evolution by predicting temperature change rates rather than absolute temperatures. Subsequently, the input features to the DNN are rotational speed of the EM n_{EM}, torque of the EM T_{EM}, winding temperature θ_{w} and rotor temperature θ_{r} of the EM. The DNN's output features are the gradients $\dot{\theta}_{\text{w}}$ and $\dot{\theta}_{\text{r}}$, at timestep k. This can be summarized as follows in Equation (5):

$$\begin{aligned} X_{\text{Train}} &= \begin{bmatrix} n_{\text{EM}}(k) & T_{\text{EM}}(k) & \theta_{\text{w}}(k) & \theta_{\text{r}}(k) \end{bmatrix}^T \quad \in \mathbb{R}^4 \,, \\ Y_{\text{Train}} &= \begin{bmatrix} \dot{\theta}_{\text{w}}(k) & \dot{\theta}_{\text{r}}(k) \end{bmatrix}^T \quad \in \mathbb{R}^2 \,. \end{aligned} \tag{5}$$

The depth of a DNN directly influences computational complexity, as each additional layer increases the number of learnable parameters and matrix operations. To ensure real-time feasibility, particularly for integration with optimization techniques like MPC and MHE, the DNN's depth and the number of recurrent nodes are constrained. The training data distribution is shown in Figure 7 for the two outputs $\dot{\theta}_{\text{w}}$ and $\dot{\theta}_{\text{r}}$.

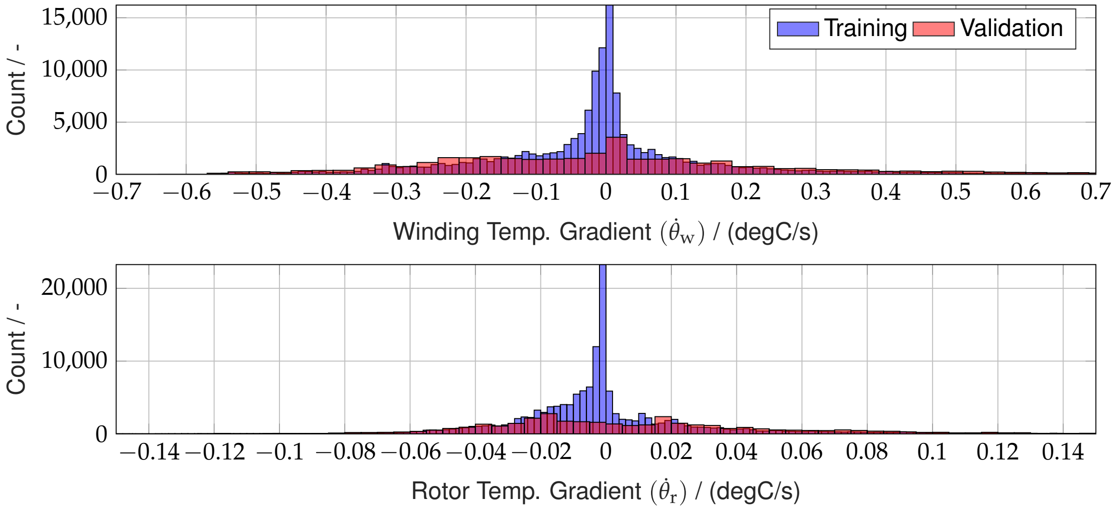


Figure 7. Data distribution of network output gradient of electrical machine winding (upper plot) and rotor (lower plot) temperature. Training and validation dataset (80:20 split). Total data points: 180000.

The training is performed with the training hyperparameters as referenced in Table 1.

Table 1. Training hyperparameter settings

Hyperparameter	**Value**
Max epoch	10000
Optimizer	Adam
Mini-batch size	512
Initial learning rate	0.02
Learn rate schedule	Piecewise drop by 25% every 500 epochs
L2 regularization	0.1
Validation frequency	10

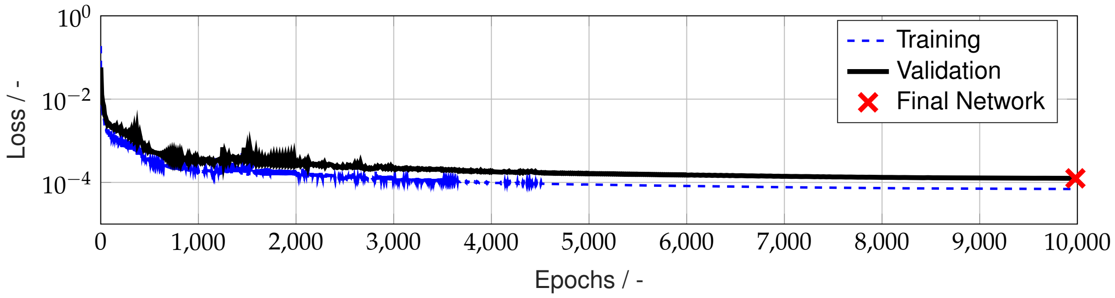


Figure 8. Loss-Epoch plot for the neural network training. The final network is chosen according to the best training loss.

The resulting loss-epoch plot of the training is shown in Figure 8, while the prediction results of the final network on the unseen test dataset are depicted in Figure 9. The unseen test dataset consists of data for a simulated lap on the Nürburgring Nordschleife with its high power requirements, thus posing a major challenge to the electric drivetrain and its thermal constraints. To solve the trade-off between the DNN's depth and thus the computational complexity and the DNN's prediction performance, empirical studies are performed, which are omitted here for brevity. The results in Figure 9 show a good correspondence with the ground truth from the high fidelity model, achieving an RMSE of 0.0373 (degC/s) and 0.0282 (degC/s) and an NRMSE of 2.77% and 9.39% for $\dot{\theta}_{\mathrm{w}}$ and $\dot{\theta}_{\mathrm{r}}$, respectively. Further performance metrics are depicted in Table 2.

Table 2. Deep neural network prediction performance metrics on unseen test dataset. MAE: Mean Absolute Error, MSE: Mean Squared Error, RMSE: Root MSE, NRMSE: Normalized RMSE.

Metric	$\dot{\theta}_{\mathrm{w}}$	$\dot{\theta}_{\mathrm{r}}$
MAE / (degC/s)	0.0288	0.0210
MSE / (degC/s)^2	0.0014	0.0008
RMSE / (degC/s)	0.0373	0.0282
NRMSE / -	2.77%	9.39%

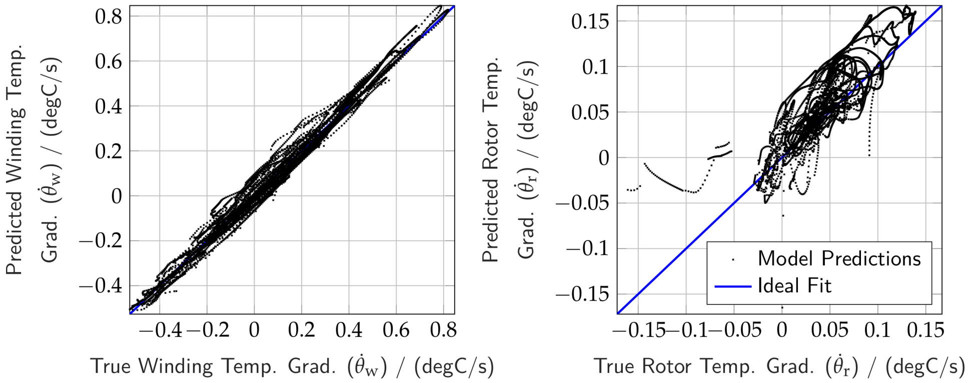


Figure 9. Predicted neural network outputs vs. the actual ground truth data for both network outputs for unseen test dataset, gradients of winding and rotor temperature of the electric machine.

The lower accuracy in predicting $\dot{\theta}_r$ can be attributed to the complexity of the high-fidelity thermal model and the weaker influence of the selected features. To be more precise, the network struggles with lower rotor temperatures due to a lack of training data in that range (see Figure 7). However, since the focus of data generation was on higher, critical temperatures, this limitation is acceptable, and the network is considered sufficiently accurate due to its superior performance on $\dot{\theta}_w$.

The presentation of the DNN predictions in the time-domain of the unseen test dataset of the Nürburgring Nordschleife in Figure 10 underlines the good performance of the chosen network.

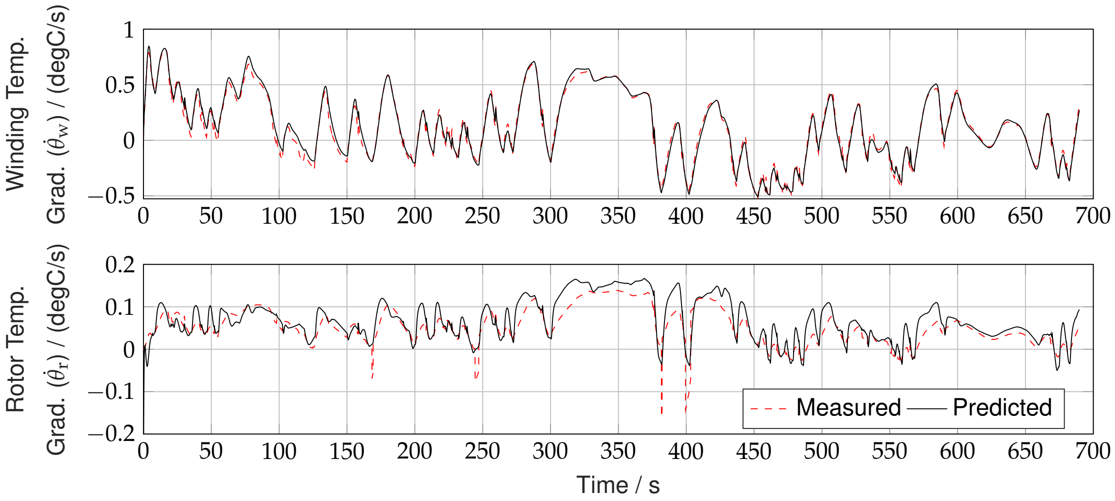


Figure 10. Predicted neural network outputs on the unseen test (Nürburgring Nordschleife) dataset over time domain: gradients of winding and rotor temperature of the electric machine.

Since the neural network predicts temperature gradients, explicit first-order Euler integration with an initial temperature of $\theta_{w,r,0} = 60\ ^{\circ}C$ for both the winding and the rotor is used to visualize the absolute temperatures.

3. Estimator Formulation

MHE is employed to reconstruct system states by solving an optimization problem at each timestep. The optimization problem has a similar structure as an Optimal Control Problem (OCP) such that tailored solvers for OCP-structured problems can be used. Following successful training of the DNN model, the next step involves formulating the estimator and implementing it within the OCP-structure of the `acados` framework. Thereto, both the driving dynamics and the DNN-based thermal model are reformulated in a discrete-time, optimization-compatible form.

3.1. MHE Problem Formulation

The dynamics depicted here are based on a discrete-time, non-linear, time-invariant system [26]:

$$x_{k+1} = \tilde{f}(x_k, w_k), \qquad y_k = g(x_k) + v_k \; , \tag{6}$$

with state $x \in \mathbb{X} \subseteq \mathbb{R}^n$, measurement $y \in \mathbb{Y} \subseteq \mathbb{R}^p$, process disturbance $w \in \mathbb{W} \subseteq \mathbb{R}^g$, measurement disturbance $v \in \mathbb{V} \subseteq \mathbb{R}^p$. The non-linear function $\tilde{f}(x_k, w_k)$ describing the system dynamics and the measurement model $g(x)$ are now developed, bringing the DNN-based thermal model and the one-dimensional driving dynamics model in a common, discrete form.

Firstly, the driving dynamics in Equation (4) can be further summarized and discretized using explicit Euler integration of first order with integration interval δk as:

$$v_{\mathrm{veh},k+1} = v_{\mathrm{veh},k} + f_{\mathrm{DD}}(v_{\mathrm{veh},k}, T_{\mathrm{EM,acc},k}, T_{\mathrm{EM,brk},k}, T_{\mathrm{fric,brk},k}, \phi_k) \cdot \delta k \; , \tag{7}$$

with f_{DD} summarizing the driving dynamics equation.

Equation (1) presents the equations for an LSTM unit. This formulation is transformed into a forward propagating process model by representing operations as equations incorporating stored weights and biases. The LSTM internal memory states - the hidden and cell states (c, h) - are included in the network inputs and outputs due to the unrolling of the recurrent layer. Adding the FC layer of the DNN provides the full dynamics representation of the DNN:

$$\begin{bmatrix} \delta\theta_{\mathrm{w},k+1} & \delta\theta_{\mathrm{r},k+1} & h_{k+1} & c_{k+1} \end{bmatrix}^T = f_{\mathrm{DNN}}(\theta_{\mathrm{w},k}, \theta_{\mathrm{r},k}, h_k, c_k) \; . \tag{8}$$

The function f_{DNN} thus summarizes the complete DNN dynamics.

Finally, the absolute temperature predictions for the next timestep using explicit first-order Euler integration are defined as:

$$\begin{bmatrix} \theta_{\mathrm{w},k+1} & \theta_{\mathrm{r},k+1} \end{bmatrix}^T = \begin{bmatrix} \theta_{\mathrm{w},k} & \theta_{\mathrm{r},k} \end{bmatrix}^T + \begin{bmatrix} \delta\theta_{\mathrm{w},k+1} & \delta\theta_{\mathrm{r},k+1} \end{bmatrix}^T \cdot \delta k \; . \tag{9}$$

Following the dynamics from Equation (7) and Equation (9), the functions within the MHE dynamics in Equation (6) can be further defined as:

$$\tilde{f}(x_k, w_k) = f_{\mathrm{DNN}}(f_{\mathrm{DD}}(x_k),\; x_k) + w_k \; , \tag{10a}$$

$$g(x_k) = f_{\mathrm{DNN}}(x_k) \; . \tag{10b}$$

Furthermore, using these dynamics, the MHE optimization problem including the objective function is defined as:

$$\begin{aligned} \underset{\substack{x_{T-N},\ldots,x_T,\\ w_{T-N},\ldots,w_{T-N}}}{\text{minimize}} \quad & \underbrace{\frac{1}{2}\|x_{T-N} - \bar{x}_{T-N}\|^2_{P_0^{-1}}}_{\alpha_k(x_k)} + \sum_{k=T-N}^{T-1} \frac{1}{2}\|w_k\|^2_{Q^{-1}} + \frac{1}{2}\|g(x_k) - y_k\|^2_{R^{-1}} \\ \text{subject to} \quad & x_{k+1} = \tilde{f}(x_k, w_k), k = T-N, \ldots, T-1, \end{aligned} \tag{11}$$

Here, $x = (x_k, \ldots, x_{k+N+1})$ represents the optimization variables, while $y = (y_k, \ldots, y_N)$ forms the measurement window. The weighting matrices Q and R are positive definite diagonal matrices corresponding to process and measurement variances; P_0 is the weighting matrix of the arrival cost. MHE operates by optimizing on a fixed horizon of N past measurements $[T-N, T]$, where past information outside the estimation window is not directly included in the optimization (see also Fig 1). The arrival cost, $\alpha_k(x_k)$, addresses this by approximately incorporating information from prior states $[0, T-N-1]$.

This concludes the formulation of the MHE and sets the stage for the implementation into an OCP-structured NLP in the framework `acados` in the following subsection.

3.2. Implementation in `acados`

State Estimation is performed by minimizing the variance between system predictions and measurements. To solve the MHE optimization problem, the optimal value of the additive process noise w must be determined to yield the most accurate estimate. The noise w is thus treated explicitly as an optimization variable.

This leads to the optimization problem to be formulated as an OCP, where the process noise w is considered as the controls input. Here, the OCP is structured around four sets of variables: states x, controls u, parameters p, and measurements y, each of which is defined below.

The primary states to be estimated include the winding and rotor temperatures $(\theta_{\mathrm{w}}, \theta_{\mathrm{r}})$, with state evolution governed by the DNN-based thermal model.Given the recurrent nature of an RNN, its hidden and cell states (c_k, h_k) are incorporated into the state vector, resulting in an increased state dimension that depends on the number of LSTM units (here: 8):

$$x_k = \begin{bmatrix} \theta_{\mathrm{w},k} & \theta_{\mathrm{r},k} & c_k & h_k \end{bmatrix}^T , \quad \in \mathbb{R}^{18} . \tag{12}$$

With the process noise w_k defined as an additive term, the state evolution as seen in Equation (10) can be stated as:

$$x_{k+1} = \begin{bmatrix} \theta_{\mathrm{w},k+1} + w_{\theta,\mathrm{w},k} & \theta_{\mathrm{r},k+1} + w_{\theta,\mathrm{r},k} & h_{k+1} & c_{k+1} \end{bmatrix}^T = \tilde{f}(x_k, w_k), \quad \in \mathbb{R}^4 . \tag{13}$$

Further, the control vector can be represented as:

$$u_k = w_k = \begin{bmatrix} w_{\theta,\mathrm{w},k} & w_{\theta,\mathrm{r},k} \end{bmatrix}^T , \qquad w_k \sim \mathcal{N}(0, Q(\alpha)) , \quad \in \mathbb{R}^2 . \tag{14}$$

The process noise follows a Gaussian distribution $\mathcal{N}$ with a variance $Q(\alpha)$, a diagonal matrix derived from the variance of the states. This variance matrix also serves as a weighting matrix in the optimization of state estimation accuracy (see Equation (11)). The MHE model utilizes past control inputs from the MPC, including $T_{\mathrm{EM,acc}}$, $T_{\mathrm{EM,brk}}$, $T_{\mathrm{fric,brk}}$, along with road inclination ϕ, and vehicle velocity v_{veh} as inputs to predict the states. Thus, the parameter vector is defined as:

$$p = \begin{bmatrix} T_{\mathrm{EM,acc},k} & T_{\mathrm{EM,brk},k} & T_{\mathrm{fric,brk},k} & \phi & v_{\mathrm{veh},k} \end{bmatrix}^T , \quad \in \mathbb{R}^5 . \tag{15}$$

The measurements y, as defined in Equation (6), incorporate additive sensor noise to account for real-world inaccuracies such as offset and sensitivity errors. These errors are modeled using additive white Gaussian noise [27], along with a time delay to represent thermal lag in sensor response. Consequently, the noisy measurements $(\theta_{\mathrm{w,meas}}, \theta_{\mathrm{r,meas}})$ serve as inputs to the MHE:

$$y = \begin{bmatrix} \theta_{\mathrm{w,meas}} & \theta_{\mathrm{r,meas}} \end{bmatrix}^T . \tag{16}$$

The term $\alpha_k(x_k)$ from Equation (11) in the `acados` OCP framework encapsulates the information required for the initial node computation and arrival cost. The vectors are defined as:

$$\begin{aligned} x_0 &= \begin{bmatrix} \theta_{\mathrm{w},k} & \theta_{\mathrm{r},k} & w_{\theta,\mathrm{w}} & w_{\theta,\mathrm{r}} & \theta_{\mathrm{w},k} & \theta_{\mathrm{w},k} \end{bmatrix}^T , \\ \overline{x}_0 &= \begin{bmatrix} \theta_{\mathrm{w,meas}} & \theta_{\mathrm{r,meas}} & 0 & 0 & \hat{\theta}_{\mathrm{w},k} & \hat{\theta}_{\mathrm{r},k} \end{bmatrix}^T . \end{aligned} \tag{17}$$

Here, $\theta_{\mathrm{w},k}, \theta_{\mathrm{r},k}, w_{\theta,\mathrm{w},k-1}$, $w_{\theta,\mathrm{r},k-1}$ form the vector for the first node and the remaining terms serve as input for the arrival cost. In the recursive online simulation, the optimizer continuously refines the state trajectory over a moving horizon.

`acados` performs simulation in a forward time manner, progressing from timestep k to $(k+N)$, which corresponds to $(T-N)$ to T as illustrated in Figure 1. The estimated state at the final node $(k+N)$ serves as the current timestep estimate T, while the state at the first node k contributes to the arrival cost for the next OCP iteration with a shifted horizon.

The control variables u_k are optimized to align predictions with available measurements while accounting for uncertainty. A key feature of this approach is that the optimizer autonomously determines the optimal noise added to the state, eliminating the necessity for explicit constraints on controls. Constraints on the physical states $(\underline{\theta}_{\mathrm{w}}, \underline{\theta}_{\mathrm{r}}, \overline{\theta}_{\mathrm{w}}, \overline{\theta}_{\mathrm{r}})$, remain enforced. The various parameters settings, weighting matrices and constraints on the OCP model are detailed in Section 4 along with the results.

4. Simulation Results

The MiL simulation is performed over a single lap of the Nürburgring Nordschleife test dataset, with the reference velocity v_{ref} generated offline. The MPC tracks v_{ref} while adhering to system constraints and determines the control variables, applied to the vehicle without any control disturbances. The resulting torque determines the vehicle velocity through the driving dynamics model, while the corresponding electric machine temperatures are computed using a high-fidelity 70-node LPTN model in the plant. To simulate realistic measurement conditions, a noise is added to the temperature values obtained from the high-fidelity plant model, yielding imprecise sensor readings. The overall simulation framework is illustrated in Figure 11.

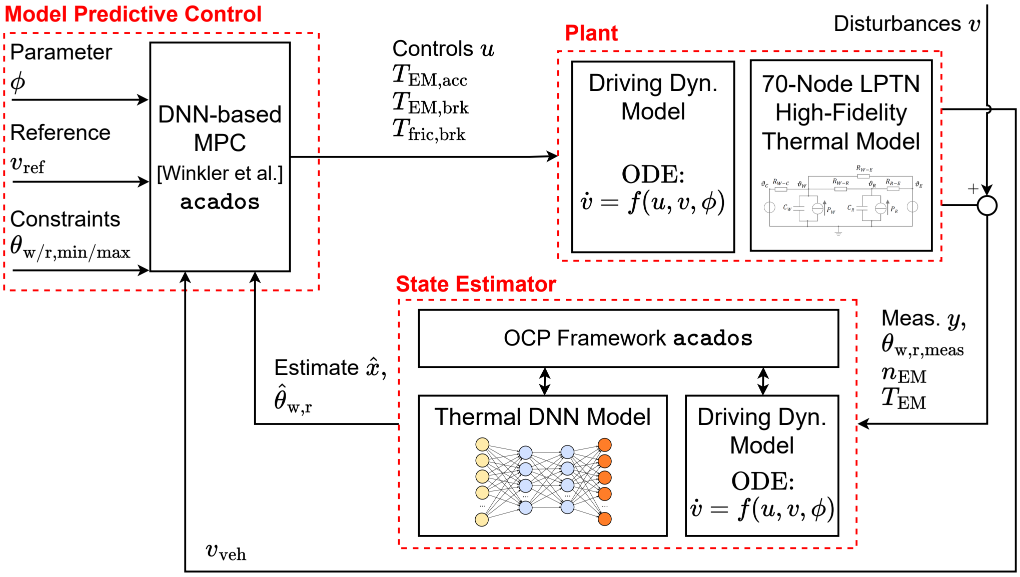


Figure 11. Simulation and model setup for MHE application and validation.

Based on the properties and specifications of commonly used temperature sensors a negative mean of $-1°C$ and variance 0.1 is applied to the added measurement noise thus resulting in $\nu \sim \mathcal{N}(-1, 0.1)$. Additionally, to account for thermal lag a 1.5 second delay is incorporated. The key parameters of the integration of the OCP-structured NLP in the `acados` framework are summarized in Table 3, using a linear least squares cost function. The prediction horizon T and the estimation horizon N are set to 1.5 seconds and 15 shooting nodes, respectively, ensuring a balance between prediction accuracy and computational feasibility. The weighting matrices (Q, R) are tuned based on training session data and modeled sensor noise, with a greater emphasis on arrival cost to ensure effective assimilation of past information into the estimation process.

Sequential Quadratic Programming (SQP) is used to solve the OCP-structured NLP, with a maximum number of SQP iterations of 20. The quadratic subproblems are solved using the High-Performance Interior Point Method (HPIPM) framework developed by [28] which is interfaced via

doi:10.20944/preprints202503.1638.v1

Table 3. Tuning parameters for Optimal Control Problem

Symbol	Parameter	Value
$\underline{\theta}_{\mathrm{w}}, \underline{\theta}_{\mathrm{r}}$	Minimum winding and rotor temperature	$0\ ^{\circ}C$
$\overline{\theta}_{\mathrm{w}}, \overline{\theta}_{\mathrm{r}}$	Maximum winding and rotor temperature	$155\ ^{\circ}C$
P_0	Weighting matrix of the arrival cost	diag(1, 1)
Q	Weighting matrix of mapped states	diag$(0.02, 0.02)$
R	Weighting matrix of controls	diag$(0.7, 0.7)$
N	Estimation horizon	15
δk	Timestep size	100 ms
T	Horizon length	1.5 s

`acados`. To further reduce computational complexity, the full estimation horizon N is condensed from 15 to 5 nodes using a partial condensing routine.

The primary focus of this study is to assess the feasibility of the MHE framework using a DNN-based plant model. Instead of quantitatively comparing vehicle performance under MPC control, the results are evaluated qualitatively to determine the effectiveness of the DNN-based state estimator in reconstructing temperature states. Given that only the winding temperature θ_{w} is susceptible to reaching critical thresholds, the evaluation is centred on monitoring this key metric.

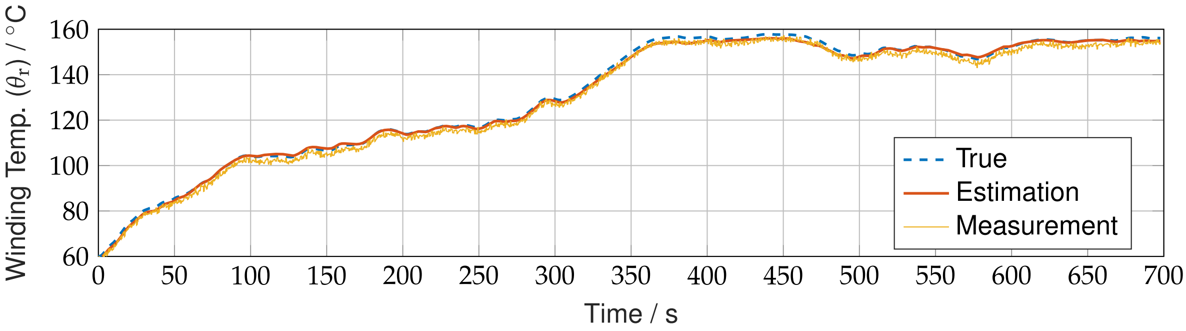


Figure 12. Moving Horizon Estimation utilizing Long Short-Term Memory (LSTM) neural network compared to true and measured value (incl. noise) for whole drive cycle run.

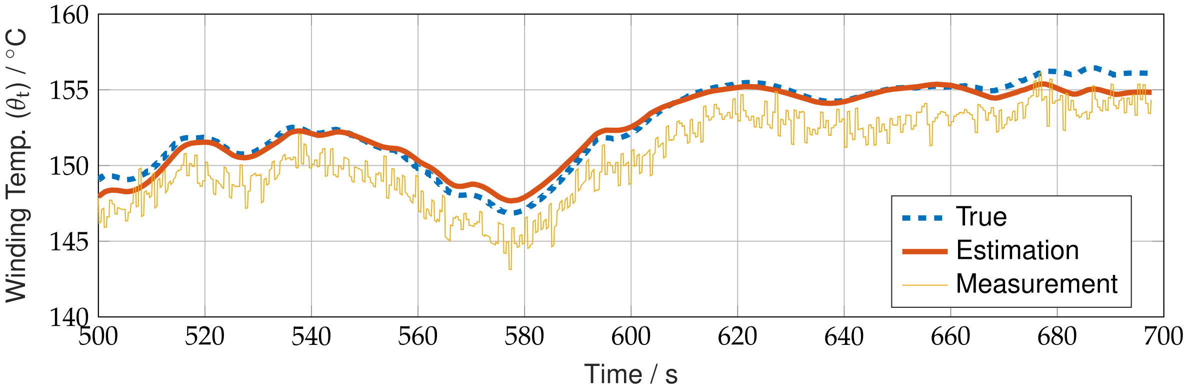
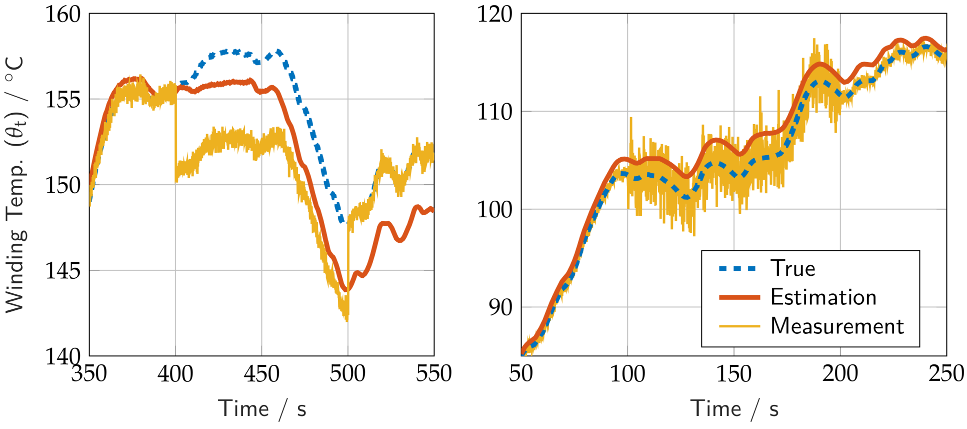


Figure 13. Moving Horizon Estimation utilizing Long Short-Term Memory (LSTM) neural network compared to true and measured value (incl. noise) for sensible temperature range (zoom).

Figure 12 presents the winding temperature estimates generated by the MHE. A closer examination in Figure 13 focuses on the temperature range of $140°C$ to $165°C$, where excessive heating poses a risk of PMSM damage. The estimated values are compared against ideal temperature profiles from the plant model and noisy sensor measurements. Despite deviations in raw sensor readings, the

DNN-based MHE effectively optimizes state estimates, producing values that closely align with the ideal temperature profile.

To further evaluate the robustness of the DNN-based MHE, an experimental scenario is designed by introducing artificial sensor failures, including a high negative offset and increased noise amplitude. As shown in Figure 14, the left plot (a) illustrates the impact of a significant negative offset, while the right plot (b) demonstrates the effect of high noise amplitude.



(a) High negative offset.

(b) High noise amplitude.

Figure 14. Zoom plots of estimator outputs for winding temperature compared to true and measured value (incl. noise) with heavy failure injection for robustness investigation.

Despite these extreme conditions, the MHE maintains stability in its outputs, preventing excessive oscillations or deviations from the true values. The ability of the DNN-based MHE to provide reliable state estimates under severe sensor disturbances underscores its robustness, reinforcing its suitability for safety-critical applications.

5. Embedded Integration

The state estimator undergoes real-time validation following the successful MiL simulations. To assess its feasibility for real-world applications, the MHE's real-time performance on an embedded system is evaluated, ensuring computational efficiency and compatibility with production code and control units.

For this purpose, the simulation is deployed on a SCALEXIO real-time embedded hardware-in-the-loop (HiL) system from dSPACE GmbH. The processing unit features a 3.8 GHz processor with four cores, three of which are dedicated to model computation. The MPC and MHE run as separate instances on two cores, while the third core handles the vehicle model, reference trajectory, and system interfacing. Although this system can be considered as more powerful than traditional processors in automotive engine, vehicle or sensor control units, the MHE problem itself remains computationally demanding due to its high state and control dimensions and horizon length. The necessary cross-compilation of the libraries is executed based on the "embedded workflow", presented by the acados developers [29].

With a solver timestep of 100 ms, the performance of the solver is evaluated based on the corresponding processor calculation time. The solver executes a maximum of 20 SQP iterations per timestep, with a peak solver time of 28 ms and an average of 5.7 ms. Table 4 shows the relevant parameters and results of the real-time testing.

Table 4. Parameter settings and simulation results for embedded real-time testing of deep neural network-based moving horizon estimator.

Parameter	Value
Timestep	100 ms
Horizon length (nodes)	15, condensed to 5
Maximum number of SQP iterations	20
Maximum number of iterations within the QP solver	100
Maximum computation time per timestep	28 ms
Average computation time per timestep	5.7 ms

These findings demonstrate that the DNN-based MHE is roughly 3-fold real-time capable and can be deployed on embedded hardware, marking a significant step toward practical implementation in real-world control applications.

6. Conclusions

This research introduces a novel state estimation framework that integrates DNNs into MHE, replacing conventional physics-based models with data-driven approaches. This innovation enhances adaptability and computational efficiency, making it suitable for real-time applications.

Using extensive synthetically generated data from a high-fidelity thermal model, a DNN featuring LSTM nodes to enhance its temporal prediction performance is trained. The MHE is then formulated by integrating the DNN thermal model with one-dimensional driving dynamics in a discrete form, employing forward propagation for the DNN dynamics. Additionally, the LSTM's hidden and cell states, which capture the long-term dependencies, are incorporated to the MHE's state vector, to preserve the DNN's dynamics. The OCP-structured NLP is then solved using the open-source framework `acados`. Through MiL simulations of thermal derating for a PMSM in a BEV, the framework demonstrated accurate estimation of critical temperatures, even under noisy sensor conditions and artificial sensor failures. Notably, it achieved a 3-fold real-time capability on a real-time computer, confirming its feasibility for embedded systems.

While the framework offers significant advantages, its performance depends on high-quality training data, and its generalization to other systems remains unverified. Additionally, the lack of interpretability in data-driven models may limit adoption in safety-critical applications. Addressing these limitations is key to broader implementation.

Future work will thus focus on enhancing generalization through transfer learning, enabling adaptation to different systems beyond thermal derating without extensive retraining. Additionally, integrating anomaly detection could improve fault resilience by identifying sensor failures and unexpected conditions. Further research may explore self-learning mechanisms, allowing the estimator to dynamically adapt to evolving system dynamics, reducing reliance on pre-collected datasets. In addition, the real-time implementation on embedded systems also opens possibilities for edge computing applications.

Overall, this research highlights the potential of DNN-based MHE for complex, real-time control applications, particularly in scenarios where accurate mathematical models are difficult to obtain or computationally expensive. By bridging the gap between model-based and data-driven approaches, this work paves the way for rapidly developed, adaptive, and computationally efficient state estimation frameworks suitable for next-generation, safety-critical systems.

Author Contributions: A.W.: Writing - Original Draft (lead), Conceptualization, Software, Validation, Visualization, Data Curation; P.S.: Methodology, Investigation, Software (lead), Writing - Original Draft, Data Curation; K.B.: Methodology, Writing - Review and Editing; V.S.: Validation, Writing - Review and Editing, Data Curation; D.G.: Supervision, Validation, Writing - Review and Editing; J.A.: Supervision, Project Administration, Funding Acquisition, Writing - Review and Editing. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The research was performed as part of the Research Group (Forschungsgruppe) FOR 2401 "Optimization based Multiscale Control for Low Temperature Combustion Engines," which is funded by the German Research Association (Deutsche Forschungsgemeinschaft, DFG).

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Simon, D. *Optimal State Estimation - Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons, 2006.
2. Aldrich, J. R.A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science* **1997**, *12*. https://doi.org/10.1214/ss/1030037906.
3. Janacek, G.J. Estimation of the minimum mean square error of prediction. *Biometrika* **1975**, *62*, 175. https://doi.org/10.2307/2334501.
4. Kalman, R.E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering* **1960**, *82*, 35–45.
5. Fujii, K. Extended kalman filter. *Reference Manual* **2013**.
6. Julier, S.; Uhlmann, J. New extension of the Kalman filter to nonlinear systems. *Proceedings of SPIE* **1997**. https://doi.org/10.1117/12.280797.
7. Rao, C.V.; Rawlings, J.B.; Lee, J.H. Constrained linear state estimation — a moving horizon approach. *Automatica* **2001**, *37*, 1619–1628. https://doi.org/10.1016/s0005-1098(01)00115-7.
8. Rawlings, J.B.; Allan, D.A., Moving Horizon Estimation. In *Encyclopedia of Systems and Control*; Springer International Publishing: Cham, 2021; pp. 1352–1358. https://doi.org/10.1007/978-3-030-44184-5_4.
9. Asch, M.; Bocquet, M.; Nodet, M. *Data assimilation: methods, algorithms, and applications*; French National Centre for Scientific Research, 2016.
10. Brunton, S.L.; Kutz, J.N. *Data-Driven Science and Engineering*; Cambridge University Press, 2019. https://doi.org/10.1017/9781108380690.
11. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016. http://www.deeplearningbook.org.
12. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324. https://doi.org/10.1109/5.726791.
13. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer-Verlag: Berlin, Heidelberg, 2006.
14. Sarker, I.H. Deep Learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science* **2021**. https://doi.org/10.1007/s42979-021-00815-1.
15. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, pp. 1735–1780.
16. Bragantini, A.; Baroli, D.; Posada-Moreno, A.F.; Benigni, A. Neural-network-based state estimation: The effect of pseudo- measurements. *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)* **2021**. https://doi.org/10.1109/isie45552.2021.9576442.
17. Suykens, J.a.K.; De Moor, B.L.R.; Vandewalle, J. Nonlinear system identification using neural state space models, applicable to robust control design. *International Journal of Control* **1995**, *62*, 129–152. https://doi.org/10.1080/00207179508921536.
18. Pan, Y.; Sung, S.W.; Lee, J.H. Nonlinear dynamic trend modeling using feedback neural networks and prediction error minimization. *IFAC Proceedings Volumes* **2000**, *33*, 827–832.
19. Norouzi, A.; Shahpouri, S.; Gordon, D.; Winkler, A.; Nuss, E.; Abel, D.; Andert, J.; Shahbakhti, M.; Koch, C.R. Deep learning based model predictive control for compression ignition engines. *Control Engineering Practice* **2022**, *127*, 105299. https://doi.org/10.1016/j.conengprac.2022.105299.
20. Gordon, D.C.; Winkler, A.; Bedei, J.; Schaber, P.; Pischinger, S.; Andert, J.; Koch, C.R. Introducing a Deep Neural Network-Based Model Predictive Control Framework for Rapid Controller Implementation. In Proceedings of the 2024 American Control Conference (ACC), 2024, pp. 5232–5237. https://doi.org/10.23919/ACC60939.2024.10644830.
21. Winkler, A.; Wang, W.; Norouzi, A.; Gordon, D.; Koch, C.; Andert, J. Integrating Recurrent Neural Networks into Model Predictive Control for Thermal Torque Derating of Electric Machines. *IFAC-PapersOnLine* **2023**, *56*, 8254–8259. 22nd IFAC World Congress, https://doi.org/10.1016/j.ifacol.2023.10.1010.

22. Winkler, A.; Frey, J.; Fahrbach, T.; Frison, G.; Scheer, R.; Diehl, M.; Andert, J. Embedded Real-Time Nonlinear Model Predictive Control for the Thermal Torque Derating of an Electric Vehicle. *IFAC-PapersOnLine* **2021**, *54*, 359–364. 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021, https://doi.org/10.1016/j.ifacol.2021.08.570.
23. Verschueren, R.; Frison, G.; Kouzoupis, D.; Frey, J.; van Duijkeren, N.; Zanelli, A.; Novoselnik, B.; Albin, T.; Quirynen, R.; Diehl, M. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation* **2021**. https://doi.org/10.1007/s12532-021-00208-8.
24. Winkler, A. Deep Neural Network Based Moving Horizon Estimation: Data, Models, Scripts (feat. acados), 2025. https://doi.org/10.5281/zenodo.15056783.
25. Brownlee, J. *Long Short-Term Memory Networks With Python Edition v1.0*; Machine Learning Mastery, 2017.
26. Rawlings, J.; Mayne, D.; Diehl, M. *Model Predictive Control: Theory, Computation, and Design*; Nob Hill Publishing, 2017.
27. Haykin, S.; Moher, M. *Communication Systems*; Wiley, 2009.
28. Frison, G.; Diehl, M. HPIPM: a high-performance quadratic programming framework for model predictive control. *IFAC-PapersOnLine* **2020**, *53*, 6563–6569. https://doi.org/10.1016/j.ifacol.2020.12.073.
29. Frey, J.; Hänggi, S.; Winkler, A.; Diehl, M. Embedded Workflow - acados documentation. Available at https://docs.acados.org/embedded_workflow/index.html, accessed March, 5th, 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.