THERMAL MODELLING OF AN INDUCTION MOTOR TO STUDY LOCAL TEMPERATURE RISE DIFFERENCES BETWEEN SINUSOIDAL AND PWM SUPPLY

Lappeenranta-Lahti University of Technology LUT

Master’s Programme in Electric Transportation Systems, Master's thesis

2025

Syed Md Takammul Hossain

Examiner(s): Associate Professor Lassi Aarniovuori

Associate Professor Janne Nerg

ABSTRACT

Lappeenranta-Lahti University of Technology LUT

LUT School of Energy Systems

Electric Transportation Systems

Syed Md Takammul Hossain

THERMAL MODELLING OF AN INDUCTION MOTOR TO STUDY LOCAL TEMPERATURE RISE DIFFERENCES BETWEEN SINUSOIDAL AND PWM SUPPLY

Master's thesis

2025

177 pages, 14 figures, 7 tables and 3 appendices

Examiner(s): Associate Professor Lassi Aarniovuori and Associate Professor Janne Nerg

Keywords: Conductance matrix, Iterative thermal solution, Loss modelling, Lumped-parameter thermal network (LPTN), Speed-dependent internal convection, Steady-state thermal analysis, Taylor number correlation, Thermal modelling, Thermal nodes, Thermal resistance network.

Induction motors in modern drive systems are commonly supplied from voltage-source inverters, where pulse-width modulation (PWM) introduces harmonic distortion that increases losses and modifies their spatial distribution. Measurements on a 5 kW SEMTEC induction motor showed that the temperature rise under PWM supply exceeded the level predicted by the increase in total losses alone, suggesting that additional local heating mechanisms were present. This observation motivated the development of a steady-state lumped-parameter thermal network to study local temperature-rise differences between sinusoidal and PWM excitation.

A 22-node motor-specific thermal network was created, including separate nodes for end-windings, end-space air regions and the stator–frame interface. Internal air-gap convection was modelled as speed dependent, enabling the representation of inverter-fed operation at

reduced speed. Because experimental measurements did not provide a full PWM loss breakdown, a residual-loss allocation method (“Path B”) was introduced, where the measured PWM loss increment was distributed to nodes most sensitive to waveform distortion.

The model was implemented in MATLAB and evaluated against eighteen temperature-sensor measurements across seven operating points. The results showed that the network reproduced the measured temperature ordering for both supply types, and when PWM residual losses were assigned to winding and near-winding regions, the higher temperature rise under PWM was captured. The main sources of uncertainty were the frame-to-ambient convection coefficient, the stator–frame contact resistance and the proportion of PWM-related losses.

The work provides a transparent and reproducible method for comparing sinusoidal and PWM operation in induction motors and demonstrates that a measurement-supported thermal network can describe local temperature-rise differences under inverter excitation with good consistency.

ACKNOWLEDGEMENTS

I would like to thank my supervisor's Associate Professor Lassi Aarniovuori and Associate Professor Janne Nerg for the endless support, guidance and time to complete the thesis. I must mention Junior Researcher Miroslav Lípa for providing me with all the necessary experimental data and clear insights.

Finally, I am deeply grateful to my family for their patience and continuous encouragement. I thank everyone directly or indirectly involved with the completion of this thesis.

.

SYMBOLS AND ABBREVIATIONS

A	Area	m²
B_{pk}	Peak air-gap flux density	[T]
c_p	Specific heat capacity	[J/(kg·K]
f	Electrical frequency	[Hz]
g	Air-gap width	[m]
h	Convective heat-transfer coefficient	[W/(m²·K]
k	Thermal conductivity	[W/(m·K]
L	Axial length	[m]
$\dot{m}$	Mass-flow rate	[kg/s]
n	Mechanical speed	[rpm]
P	Power or loss component	[W]
P_{cu}	Copper loss	[W]
P_{fe}	Iron loss	[W]
P_{mech}	Mechanical loss	[W]
P_{stray}	Stray loss	[W]
Q	Heat input at a thermal node	[W]
R	Thermal resistance	[K/W]
R_{20}	Phase resistance at 20 °C	[Ω]
T	Temperature	[°C]
T_{amb}	Ambient temperature	[°C]
α_{Cu}	Copper temperature coefficient	[1/K]
μ	Dynamic viscosity	[Pa·s]
ρ	Density	[kg/m³]
τ	Torque	[Nm]
ω	Angular speed	[rad/s]

Abbreviations

AG	Air-gap
DE	Drive end
FEM	Finite-element method

IM	Induction motor
LPTN	Lumped-parameter thermal network
NDE	Non-drive end
OP	Operating point
PWM	Pulse-width modulation
RMS	Root-mean-square
SCIM	Squirrel-cage induction motor
TEFC	Totally enclosed fan-cooled
HTC	Heat-transfer coefficient
YM	Yoke mid node
YO	Yoke outer node
YI	Yoke inner node

OP1	n25T25
OP2	n25T100
OP3	n50T25
OP4	n50T50
OP5	n50T100
OP6	n90T50
OP7	n90T100

Table of contents

Abstract

(Acknowledgements)

(Symbols and abbreviations)

1 Introduction 11
2 Induction Motor Losses and Thermal Modelling 14
2.1 Overview of Induction Motor Losses 14
2.2 Classical Loss Components and Heating Mechanisms 15
2.3 PWM-Related Loss Mechanisms in Inverter-Fed Motors 16
2.4 Loss Measurement, Loss Separation, and Residual Loss Definition 18
2.5 Heat-Transfer Mechanisms in Electrical Machines 19
2.6 Thermal Modelling Approaches 20
2.6.1 Analytical thermal models 20
2.6.2 Numerical approaches (FEM and CFD) 21
2.6.3 Lumped-Parameter Thermal Networks (LPTN) 21
2.7 Modelling Considerations for PWM-Supplied Machines 22
2.7.1 Node definition 22
2.7.2 Thermal resistances 22
2.7.3 Convective coefficients 23
2.7.4 Loss allocation methods 23
2.8 Limitations of Existing Thermal Models 24
3 Thermal network development and experimental data 26
3.1 Modelling Approach and Notation 26
3.2 Definition of Thermal Nodes 27
3.3 Frame, Cooling, and Auxiliary Nodes 30
3.4 Network Topology 30
3.5 Conduction and Interface Resistances 31
3.6 Speed-Dependent Convection 32
3.7 End-Winding and End-Space Representation 32

3.8 Air-Gap Representation 33
3.9 Modelling Assumptions 33
4 Model development and loss distribution strategy 35
4.1 Motor Geometry and Node–Measurement Mapping 35
4.2 Operating Points and Available Data 36
4.3 Analytical Loss Components 38
4.4 PWM–Sine Loss Separation and the Residual ΔP 40
4.5 Node-Level Loss Injection 41
4.6 Thermal Conductance Matrix and Governing Temperature Equation 42
4.7 Iterative Copper-Loss Update 42
4.8 PWM Retuning of Thermal Parameters 43
4.9 MATLAB Solver Workflow 44
4.10 Boundary Conditions and Ambient Coupling 45
5 MATLAB Implementation of the Thermal Model 47
5.1 High-level numerical workflow 47
5.2 Detailed analytical–numerical workflow 48
5.3 Preparation of operating-point data 49
5.4 Construction of the nodal heat vector Q 50
5.5 Assembly of the conductance matrix $\boldsymbol{G}$ 51
5.6 Iterative treatment of temperature-dependent effects 51
5.7 MATLAB implementation structure 52
5.8 Assessment of the Suitability of the Implemented Approach 53
6 Results and Discussion 54
6.1 Overview of the operating-point results 54
6.2 Low-speed region 56
6.3 Mid-speed region 57
6.4 High-speed region 58
6.5 Comparison between PWM and sine excitation 58
6.6 Node-level accuracy 59
6.7 Physical explanation supported by literature 59
6.8 Summary of findings 60
6.9 Discussion 60
7 Conclusions 63

References.....65

Appendices

Appendix 1. Geometry, material properties and constants

Appendix 2. Thermal resistances used in the network

Appendix 3. Matlab implementation of the lptn model

Figures

Figure 1. Measured temperatures at all 18 points under sinusoidal and PWM supply at operating point OP7

Figure 2. Simplified thermal network block diagram

Figure 3. Physical motor geometry, thermal-node grouping, and measurement-point layout

Figure 4. Residual PWM loss allocation using Path B

Figure 5. Thermal-modelling workflow implemented in MATLAB

Figure 6. High-level numerical workflow of the MATLAB thermal solver

Figure 7. Detailed analytical–numerical workflow of the steady-state LPTN solver

Figure 8. Low-speed region – OP1

Figure 9. Low-speed region – OP2

Figure 10. Mid-speed region – OP3

Figure 11. Mid-speed region – OP4

Figure 12. Mid-speed region – OP5

Figure 13. High-speed region – OP6

Figure 14. High-speed region – OP7

Tables

Table 1. Summarises different loss components and the typical impact of PWM excitation

Table 2. Thermal nodes associated with stator yoke, slots, and end-winding regions

Table 3. Thermal nodes associated with the air-gap and rotor regions

Table 4. Thermal nodes associated with frame, external surfaces, and auxiliary regions

Table 5. Operating points under sinusoidal (grid) supply

Table 6. Operating points under PWM (converter) supply

Table 7. Analytical loss components and their allocation to LPTN nodes

1 Introduction

Induction motors are widely used in industry and continue to remain essential in many power-conversion applications (Sachin and Sriram, 2020; Groschup et al., 2021). Their thermal behaviour defines practical limits for loading, efficiency and lifetime, because temperature directly influences insulation ageing, bearing reliability and magnetic-material properties (Kylander, 1995; Nerg et al., 2008). As a result, accurate prediction of internal temperature rise has become an important task for both research and machine design. This thesis studies how the temperature distribution of an induction motor changes when the supply is switched from sinusoidal excitation to pulse-width-modulated (PWM) inverter excitation, which is now increasingly common due to the rapid adoption of adjustable-speed drives in modern industry (Bramerdorfer et al., 2017).

A growing number of motors are driven by voltage-source inverters rather than pure grid-sinusoidal sources. The inverter introduces high-frequency voltage components that modify the electromagnetic field distribution. These components create additional losses in the windings, stator teeth, and rotor surface layer. Furthermore, the distribution of these losses is usually different from that under sinusoidal supply (Sachin and Sriram, 2020; Zhang et al., 2023). Earlier studies have shown the importance of local temperatures, especially in parts that are thermally sensitive, such as end-windings, bearings, and insulation systems (Kylander, 1995; Nerg et al., 2008). However, many published works emphasise whole-machine thermal behaviour, while the detailed modelling of local temperature distributions remains challenging due to limited measurement resolution and uncertainties in heat-transfer parameters (Saari, 1998). As a result, the local temperature differences produced by PWM excitation are still not fully understood, although they may affect reliability and lifetime.

Thermal modelling of electrical machines has been developed for several decades. Lumped-parameter thermal networks (LPTN) are commonly used because they can describe conduction, convection, and contact resistances while keeping the computational load low (Kylander, 1995; Nerg et al., 2008; Andersson, 2013; Sachin and Sriram, 2020). Networks are normally created by dividing the machine into thermal nodes, followed by assigning

conduction and convection paths between them. Previous studies have discussed node-placement strategies, contact resistances, anisotropy in laminated cores, slot-layer modelling, and speed-dependent convection correlations (Kylander, 1995; Andersson, 2013; Groschup et al., 2021). Many works also use detailed PWM-related loss components, such as high-frequency copper losses or harmonic iron losses, which often require time-consuming finite-element simulations or coupled electromagnetic–thermal models (Trigeol et al., 2006; Zhang et al., 2023; Khazaeli et al., 2025). In practical situations, especially in industrial measurements, the total losses are known but their detailed harmonic breakdown is not available. When these losses are inserted into an LPTN, a clear treatment for the unmodelled residual part is often missing. In addition, even though correlations based on Taylor numbers and rotating-cylinder convection have been published and applied in several works, their use in thermal networks of totally-enclosed fan-cooled motors at moderate speeds has not always been applied consistently (Kylander, 1995; Nerg et al., 2008; Groschup et al., 2021).

These gaps indicate that there is still a need for a modelling approach that can use measured losses directly, distribute any unmodelled PWM-related remainder in a controlled way, and predict the temperature field across a larger number of internal locations (Bramerdorfer et al., 2017; Groschup et al., 2021). The ability to estimate such local temperature variations is important when evaluating insulation hotspots, rotor heating behaviour, and cooling-air characteristics, especially when the supply waveform is changed.

The present thesis investigates the steady-state temperature rise of a SEMTEC induction motor under both sinusoidal and PWM excitation. Eighteen measurement points are used to capture the temperature behaviour in the windings, stator core, rotor ring, air-gap, and end-winding surroundings. A lumped-parameter thermal network is developed for this motor, including radial and axial conduction paths, selected contact resistances, and convection coefficients suitable for the motor geometry, following the general ideas reported in earlier LPTN studies (Kylander, 1995; Nerg et al., 2008; Andersson, 2013). Measured total losses are used as model inputs, and the difference between analytical losses and measured losses is treated as a residual term, in line with previous discussions on calibrated or experimentally tuned thermal models (Jankowski et al., 2010; Khazaeli et al., 2025). This residual is distributed to specific thermal nodes through predefined allocation factors, which allows

PWM-related effects to be included even when a full harmonic loss breakdown is not available.

The main objective of this thesis is to adapt a steady-state lumped-parameter thermal network to the SEMTEC induction motor and to analyse how residual PWM-related losses influence the predicted temperature field. Three research questions are addressed. The first question asks how the network structure should be adapted to this motor. The second asks how the unmodelled PWM-related losses can be allocated among the thermal nodes. The third asks how well the resulting thermal model reproduces the measured temperatures at different parts of the motor under sinusoidal and PWM excitation. The scope of the work is limited to steady-state operation. Transient behaviour, variable-speed effects, and detailed CFD or electromagnetic finite-element simulations are outside the scope (Trigeol et al., 2006; Sachin and Sriram, 2020).

The key results show that the customised thermal network is able to reproduce the measured temperature distribution with reasonable accuracy, and the inclusion of residual PWM-related losses improves the prediction at parts of the motor where harmonic losses are most influential, similar to what has been observed in calibrated LPTN models in the literature (Khazaeli et al., 2025). The work therefore gives insights into how PWM excitation affects the local thermal behaviour and how loss-allocation strategies can be used in steady-state lumped-parameter modelling.

This thesis is organised as follows. Chapter 2 explains the theoretical background on heat-transfer mechanisms, loss modelling, and lumped-parameter thermal networks. Chapter 3 describes the SEMTEC motor, the measurement system, and the experimental dataset. Chapter 4 presents the developed thermal model. Chapter 5 compares the model results with experimental measurements. Chapter 6 summarises the findings, discusses their implications and suggests directions for future work.

2 Induction Motor Losses and Thermal Modelling

The thermal behaviour of an induction motor is closely linked to the types, locations and magnitudes of its losses. A clear understanding of these mechanisms is necessary before any modelling approach can be justified. This chapter outlines the main loss components under sinusoidal and PWM supply and explains the thermal pathways that influence the internal temperature distribution. The discussion forms the analytical basis for the modelling framework developed in the later chapters.

2.1 Overview of Induction Motor Losses

Induction motors generate heat because electrical and magnetic energy is not converted perfectly into mechanical power. These losses appear in several different parts of the machine, and each loss component influences the temperature field in a unique way. When these losses accumulate, the temperature of the motor rises and eventually reaches a new steady-state depending on the cooling conditions. Understanding how these losses form and how they change under different supply types is essential for any thermal-modelling approach (Kylander 1995; Nerg, Pyrhönen & Rilla 2008).

The total loss under steady-state conditions can be written as,

$$P_{tot} = P_{Cus} + P_{Cur} + P_{Fe} + P_{fw} + P_{stray}, \quad (1)$$

where P_{Cus} is the stator copper loss, P_{Cur} is the rotor copper loss, P_{Fe} is the iron loss in the stator and rotor cores, P_{fw} is the friction and windage loss, and P_{stray} describes additional stray-load losses due to leakage fields and slotting effects.

Each loss component behaves differently with respect to temperature, frequency, load, and the waveform of the supply. For instance, copper losses increase with both load and winding temperature, while iron losses depend strongly on the flux density waveform and its harmonic distortion (Sachin & Sriram 2020). When the motor is supplied from an inverter, the harmonic-rich voltage waveform produces higher-frequency current components, which modify several loss components compared to sinusoidal excitation (Groschup et al. 2021).

A clear categorization of losses and the associated heating mechanisms is necessary before discussing thermal modelling. The following sections introduce each classical loss category and then highlight how they are modified by pulse-width-modulated (PWM) excitation.

2.2 Classical Loss Components and Heating Mechanisms

Stator and rotor copper losses

Copper losses arise from Joule heating in the stator and rotor conductors. These losses are proportional to the square of the RMS current and depend on conductor temperature. The general expression is,

$$P_{Cu} = I^2 R(T), \tag{2}$$

where $R(T)$ is the temperature-dependent resistance and I is the RMS current flowing through the phase winding. As the winding temperature rises, the resistance increases and the copper loss increases correspondingly. This creates a positive feedback mechanism where heating generates additional losses, especially under heavy loading.

In the rotor cage, the induced currents also create copper losses. The rotor copper loss depends on slip and becomes significant under loaded conditions. These losses are sensitive to frequency and harmonic content, which will be discussed later in the PWM section.

Iron losses (core losses)

Iron losses occur in the stator and rotor laminations due to alternating magnetic fields. They consist of hysteresis loss, eddy-current loss, and excess loss. Under sinusoidal supply, core loss is often approximated by the classical Steinmetz equation. However, in practice, researchers show that iron loss under distorted flux waveforms exhibits a more complex dependency (Bramerdorfer, Cavagnino & Vaschetto 2017).

A general expression for iron losses is,

$$P_{Fe} = k_h f B^n + k_e f^2 B^2 + k_{ex} f^{1.5} B^{1.5}, \tag{3}$$

where k_h , k_e , and k_{ex} are empirical coefficients, f is frequency, and B is flux density. The first term represents hysteresis loss, the second is classical eddy-current loss and the third term accounts for excess (or anomalous) loss.

The iron loss increases sharply with frequency. Thus, when high-frequency harmonics are present in the supply voltage, even small distortion levels can create noticeable additional heating (Lípa et al. 2025).

Friction and windage losses

Mechanical losses occur due to bearing friction and air drag from rotor rotation. These losses depend mainly on speed and motor construction. Although their contribution is smaller compared to copper and iron losses, they still affect the steady-state temperature and must be included in the total loss estimate (Saari 1998).

Stray-load losses

Stray-load losses are caused by leakage fluxes, slot harmonics and non-ideal current distribution. These losses are typically challenging to predict analytically and are often estimated based on percentage rules or obtained from manufacturer data. Several studies propose using refined analytical or FEM-based methods to characterize these losses, but they remain difficult to quantify in practice (Jankowski, Li & Pillay 2010).

Stray-load losses are particularly sensitive to PWM excitation because harmonic fields generate surface and proximity effects in rotor bars and end rings.

2.3 PWM-Related Loss Mechanisms in Inverter-Fed Motors

The introduction of PWM excitation fundamentally changes the loss distribution in an induction motor. Voltage-source inverters generate non-sinusoidal waveforms that contain high-frequency switching components. These components give rise to harmonic currents and

non-sinusoidal magnetic fields, which increase iron and copper losses in several regions of the machine (Groschup et al. 2021).

Figure 1 presents the measured temperature distribution for the sinusoidal and PWM supplies at the n90T100 operating point. The figure shows how the two supply types influence the local temperatures across all 18 measuring points of the motor.

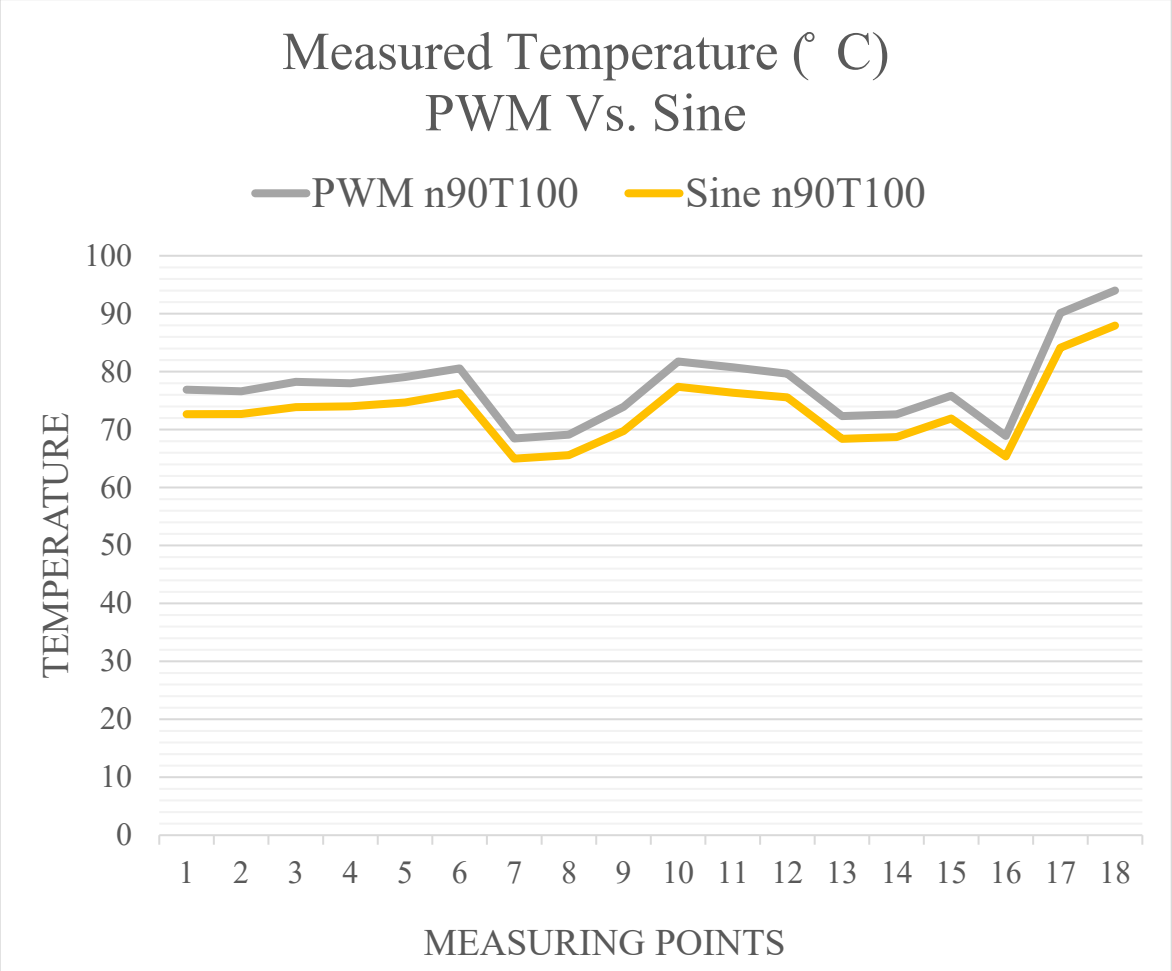


Figure 1. Measured temperatures at all 18 points under sinusoidal and PWM supply at operating point OP7.

From Figure 1, it can be observed that PWM excitation results in higher temperatures across most measuring points compared to the sinusoidal supply, although the magnitude of the increase varies between motor regions. The stator-slot and stator-iron sensors show moderate temperature rises, while the end-winding regions exhibit slightly larger differences due to their sensitivity to harmonic currents. The most pronounced increases appear in the rotor end ring and rotor yoke, indicating that these components experience greater high-frequency loss concentrations under PWM operation. These observations highlight that PWM excitation

does not uniformly affect the temperature field, and the local variations provide insight into where additional harmonic losses are likely to occur. High-frequency rotor-surface losses are especially important in inverter-fed conditions (Zhang et al. 2023). In addition, PWM excitation causes harmonic flux pulsations, increasing eddy-current loss in both stator teeth and rotor surfaces (Trigeol et al. 2006).

These effects make PWM an essential factor in thermal modelling, as ignoring PWM-related loss components may lead to underestimation of hotspot temperatures.

2.4 Loss Measurement, Loss Separation, and Residual Loss Definition

In industrial and research environments, loss separation is an important step for understanding machine behaviour. However, it is often not possible to measure every loss component independently. Many measurements provide only total input power and output mechanical power, leaving the remaining losses grouped into a generalized "residual loss" category (Jankowski, Li & Pillay 2010).

A general expression for residual loss is,

$$P_{res} = P_{tot} - (P_{Cus} + P_{Cur} + P_{Fe} + P_{fw}), \quad (4)$$

where P_{res} accounts for stray-load losses and unmodelled components, including PWM-induced effects that cannot be captured easily by classical equations.

In PWM environments, the residual term becomes particularly important because many harmonic losses—especially in rotor-surface layers and end-winding regions—are practically impossible to calculate analytically (Khazaeli et al. 2025).

In this thesis, measured total losses from the SEMTEC machine are used, and the residual is redistributed among thermal nodes based on physical reasoning and literature guidelines. This approach allows PWM-related losses to be incorporated in a controlled manner.

2.5 Heat-Transfer Mechanisms in Electrical Machines

Heat transfer inside an induction motor occurs through conduction, convection, and radiation. These mechanisms operate simultaneously along complex three-dimensional paths between motor components (Kylander 1995).

2.5.1 Conduction

Conduction is the dominant heat-transfer mechanism inside solid materials such as stator cores, windings, and rotor structures. The basic conduction resistance is expressed as,

$$R_{cond} = \frac{L}{kA}, \tag{5}$$

where L is the conduction path length, k is thermal conductivity, and A is the cross-sectional area. Materials used in induction motors often have anisotropic conductivities, especially laminated steels where thermal conductivity differs significantly in axial and radial directions (Groschup et al. 2021).

This anisotropy must be considered in accurate thermal models to avoid incorrect temperature predictions in regions such as stator yoke and rotor core.

2.5.2 Convection

Convection occurs in the air gap, end-winding regions, cooling ducts, and external surfaces of the motor. The general convective resistance is,

$$R_{conv} = \frac{1}{hA}, \tag{6}$$

where h is the convective heat-transfer coefficient and A is the surface area in contact with the fluid.

Predicting the convection coefficient in rotating machinery is challenging. One widely used approach is based on the Taylor number, which describes the flow regime inside the air gap. The modified Taylor number is expressed as,

$$Ta_m = \frac{\omega^2 R_m^3 g}{\nu^2}, \tag{7}$$

where ω is angular speed, R_m is the mean radius, g is the air-gap thickness and ν is the kinematic viscosity of air.

When Ta_m exceeds a certain threshold, Taylor vortices appear, enhancing heat transfer. Several correlations have been proposed for estimating the Nusselt number under different regimes (Lindh et al. 2010; Groschup et al. 2021).

2.5.3 Radiation

Radiation is usually a minor contributor in electrical machines but may influence external cooling surfaces under high temperature gradients. However, it is often neglected in internal thermal modelling because conduction and convection dominate.

2.6 Thermal Modelling Approaches

Several modelling approaches are used to study the temperature distribution in electrical machines. These range from simplified analytical expressions to detailed numerical simulations. The choice of method usually depends on the required accuracy, available computational resources, and the amount of input data. For steady-state industrial applications, lumped-parameter thermal networks (LPTNs) are widely adopted because they offer a reasonable compromise between accuracy and computational speed (Kylander 1995; Nerg, Pyrhönen & Rilla 2008).

2.6.1 Analytical thermal models

Analytical thermal models rely on simplified geometry and uniform material assumptions. These models can provide quick estimates of temperatures and are useful for conceptual design. Basic conduction and convection formulas, such as equations (5) and (6), are the foundation of analytical methods. However, analytical thermal models have some practical limits, because their accuracy depends a lot on how finely the model is divided and how the parameters are chosen. When the motor has uneven heat sources, complicated end-winding

shapes, or materials that behave differently in different directions, it becomes harder to make the model detailed enough without making it too heavy to compute (Bramerdorfer, Cavagnino & Vaschetto 2017).

2.6.2 Numerical approaches (FEM and CFD)

Finite element method (FEM) and computational fluid dynamics (CFD) allow detailed modelling of temperature fields, magnetic losses, complex geometries, and flow fields. FEM can capture spatially distributed losses, while CFD can simulate end-winding convection or frame cooling airflow. Several studies demonstrate highly accurate thermal predictions using FEM–CFD co-simulation (Trigeol et al. 2006; Trigeol, Bertin & Lagonotte 2006).

However, numerical models require long computation times, detailed geometry, and accurate material parameters. They are not practical for fast design cycles or multi-point analysis, and therefore they are usually not used as the primary design tool for industrial thermal studies.

2.6.3 Lumped-Parameter Thermal Networks (LPTN)

LPTNs represent the motor as a network of thermal resistances and capacitances. Each node corresponds to a material region or component, while the resistances describe heat-transfer paths. The general steady-state heat balance for a node can be written as,

$$\sum G_{ij}(T_i - T_j) = Q_i, \tag{8}$$

where G_{ij} is thermal conductance between nodes i and j, T_i and T_j are node temperatures and Q_i is the heat generated within node i.

An LPTN can be envisioned conceptually as a collection of thermal nodes interconnected by conduction and convection pathways. Radial and axial resistances associated with this representation correspond to the heat conduction across laminations, winding insulation, and rotor-stator interfaces. Convection resistances describe the transport mechanisms in the air gap, the end-winding regions, and the external cooling surfaces. While the network is coarse

in resolution, it captures the main heat-transfer mechanisms in an induction motor and allows the temperature field to be predicted using linear algebra. This makes LPTNs suitable for parametric studies and measurement-based calibration, especially under conditions where computational efficiency is required.

Researchers have continuously improved LPTN modelling by incorporating anisotropic steel conductivity, advanced convection correlations, and refined slot thermal paths (Groschup et al. 2021; Andersson 2013). In this thesis, an LPTN is adopted because it allows integration of measured losses, detailed node definition, and controlled distribution of unmodelled PWM-related losses.

2.7 Modelling Considerations for PWM-Supplied Machines

PWM excitation introduces several challenges for thermal modelling. First, the waveform contains switching frequencies that may extend up to tens of kilohertz. Second, the harmonic content increases core and surface losses in a way that is difficult to calculate analytically. Third, PWM-related losses may not scale linearly with load or speed (Zhang et al. 2023).

For these reasons, several modelling considerations must be adapted when building an LPTN for a PWM-fed motor.

2.7.1 Node definition

Node placement determines the resolution of the thermal network. Additional nodes are necessary in regions where PWM-related losses create local hotspots, such as rotor surfaces, stator teeth, and end-winding regions. The placement must also reflect anisotropic conduction paths in laminated steels (Groschup et al. 2021).

2.7.2 Thermal resistances

The accuracy of LPTN models depends strongly on how conduction and convection resistances are chosen. For PWM-fed machines, the air-gap convection must account for

speed-dependent Taylor vortex formation, and the end-winding convection may vary due to harmonic-induced air motion. Classical correlations may underestimate convection coefficients if harmonic pulsations are present (Lindh et al. 2010).

2.7.3 Convective coefficients

The convective coefficients used in LPTNs must reflect actual cooling behaviour. In totally enclosed fan-cooled (TEFC) machines, external convection is influenced by the fan, frame fins, and enclosure geometry. For PWM supply, internal air-gap convection is also affected by harmonic torque pulsations, although this effect is difficult to quantify and usually accounted for through calibration based on measurements (Khazaeli et al. 2025).

2.7.4 Loss allocation methods

As seen earlier, the residual loss in (4) contains several components that cannot be separated analytically. In PWM conditions, these residual losses are often substantial. Therefore, distributing them across relevant machine parts is necessary to capture local temperature variations correctly.

Table 1. Summarises different loss components and the typical impact of PWM excitation.

Loss component	Description	Typical change under PWM excitation	Reason for change
Stator copper loss	Joule heating in stator conductors	Small increase	Slight rise in RMS current and additional high-frequency components
Rotor copper loss	Induced currents in rotor bars and end rings	Moderate increase	Skin and proximity effects from harmonic fields

Iron loss (core loss)	Hysteresis, eddy-current, and excess losses in laminated steel	Moderate to large increase	High-frequency switching harmonics increase flux-density variation
Rotor-surface HF loss	Eddy-current loss in rotor surface layers	Large increase	Strong sensitivity to high-frequency voltage harmonics
Stray-load loss	Leakage-field, slotting, and harmonic-related parasitic losses	Largest increase	Harmonic leakage fields and high-frequency pulsations dominate this loss category

Table 1 shows that copper losses are moderately affected by PWM, while iron and rotor surface losses increase more significantly. This information guides the allocation of residual losses into rotor surface nodes, stator teeth nodes, and end-winding nodes in this thesis.

2.8 Limitations of Existing Thermal Models

Although LPTN models are widely used and offer good computational performance, several limitations must be acknowledged. Many models assume uniform losses, neglect PWM-induced spatial variations, or use simplified convection coefficients that do not reflect the real behaviour in inverter-fed machines (Sachin & Sriram 2020).

FEM-based models can capture spatial loss distribution more accurately, but they require detailed material data, long computation times, and complex geometry preparation. Furthermore, FEM does not easily integrate measurement-based loss corrections unless special workflows are implemented (Trigeol et al. 2006).

The inability of classical models to represent harmonic-related losses leads to underestimation of hotspot temperatures, especially in the rotor surface and end-winding region. Therefore, for practical design and validation, hybrid approaches combining measurements with LPTN modelling are considered more robust (Khazaeli et al. 2025).

This chapter reviewed the major loss components in induction motors and showed how PWM excitation changes the loss distribution through increased high-frequency harmonics. Classical losses such as copper loss, iron loss, friction loss, and stray-load loss were described, and the added complexity introduced by PWM was highlighted. Heat-transfer mechanisms were discussed, including conduction, convection, and radiation, with emphasis on Taylor-number-based convection in the air gap.

Different thermal-modelling approaches were compared, and the advantages of LPTN were identified. Special considerations for modelling PWM-fed machines were included: node placement, anisotropic conduction, convection correlations, and loss allocation methods. These considerations create the foundation for building the LPTN model described later in Chapter 4.

The next chapter introduces the experimental SEMTEC motor, measurement setup, and dataset used for validating the thermal model.

3 Thermal network development and experimental data

The development of a reliable thermal model for an induction motor requires a consistent representation of heat-transfer mechanisms, loss distributions, and structural interfaces. A lumped-parameter thermal network (LPTN) was selected in this thesis because it provides a suitable balance between accuracy and computational efficiency. Unlike full finite-element models, an LPTN can describe multi-directional heat flows with modest computational cost, which is important when temperature estimation must be repeated for several operating points or during calibration using measurement data (Kylander, 1995; Saari, 1998; Zhang et al., 2023).

The SEMTEC induction motor used in this research includes eighteen temperature measurements positioned across the stator, rotor, air gap and end-winding regions (T1–T18). These sensors provide both local and global information about the machine's thermal behaviour. Because part of the research objective is to examine how PWM excitation modifies local temperature fields relative to sinusoidal supply, the thermal model must be able to capture these local variations. A coarse network with only a few nodes would not reproduce the observed gradients, especially near the rotor surface and end-winding regions where harmonic-related heating is strongest (Lípa et al., 2025; Jankowski et al., 2010).

This chapter presents the development of the final 22-node LPTN used in the thesis. The node structure, conduction and convection resistances, and air-gap treatment are explained in detail. The chapter also shows how the network layout was designed to align with measurement locations and how speed-dependent convection was incorporated using established correlations. The network developed here is used directly in Chapter 4 to compute steady-state temperatures under both sinusoidal and PWM excitation.

3.1 Modelling Approach and Notation

In a lumped-parameter representation, each node represents a region of relatively uniform temperature. The heat flow between nodes is governed by thermal resistances that describe

conduction, convection, or interface behaviour. Because the thesis already introduced basic heat-transfer equations in Chapter 2, only new equations specific to the present modelling stage are introduced here. The numbering continues from the previous chapter.

Heat conduction between two regions is expressed with Equation (9),

$$R_{cond} = \frac{L}{kA}, \tag{9}$$

where L is the conduction length, k is the thermal conductivity, and A is the conduction area.

Convective heat transfer between a solid surface and air is represented with Equation (10),

$$R_{conv} = \frac{1}{hA}, \tag{10}$$

where h is the convection coefficient and A is the exposed surface area.

The LPTN yields a system of linear equations in the form

$$G\,T = Q, \tag{11}$$

with G being the thermal conductance matrix, T the unknown node temperatures and Q the vector of heat sources. Although the expression above is not assigned an equation number because it is a general matrix form, it reflects how the thermal network is solved in steady state.

The resolution of the thermal network is directly linked to its accuracy. Fine structures require more nodes, while too coarse a network may hide local gradients. The 22-node structure used here was chosen to match the physical geometry, expected heat-flow paths, and available measurement data.

3.2 Definition of Thermal Nodes

The node structure was grouped into two categories for clarity:

- Nodes 1–9: Stator and end-winding regions
- Nodes 10–14: air-gap and rotor regions
- Nodes 15–22: Frame, external surfaces, shaft, bearings, and air mixing

These nodes were positioned to preserve physical meaning and to allow direct comparison with measurement points T1–T18. The first group of nodes is presented in Table 2 and Table 3.

Table 2. Thermal nodes associated with the stator and end-winding regions

Node	Region	Mechanism	Function
1	Stator tooth (inner)	Radial conduction	Represents inner tooth temperature; relates closely to T2 measurement
2	Stator tooth (outer)	Radial conduction	Captures heat flow toward yoke; connects strongly to Node 3
3	Stator yoke (inner)	Axial + radial conduction	Intermediate conduction path between teeth and outer yoke
4	Stator yoke (outer)	Conduction to frame	Interface to frame; corresponds to T3 measurement
5	Stator slot (radial)	Copper + insulation conduction	Represents slot temperature where copper losses occur
6	Stator winding (mid-slot)	Copper-loss heating	Dominant node for stator copper (I^2R) losses
7	End-winding (upper)	Convection-sensitive	Influenced by airflow and PWM harmonics; relates to T7
8	End-winding (lower)	Convection-sensitive	Captures vertical gradient in end-winding region

Table 3. Thermal nodes associated with the air-gap and rotor regions

Node	Region	Mechanism	Function
9	Air gap	Convection + small conduction	Affected by rotor speed; relates to convection regime
10	Rotor surface layer	HF harmonic loss	Sensitive to PWM high-frequency components (Lípa et al., 2025)
11	Rotor bar (mid-depth)	Copper conduction	Represents inner bar temperature; links with rotor ring
12	Rotor bar (deep)	Axial conduction	Captures deeper bar and rotor-core behaviour
13	Rotor core (inner)	Axial conduction	Connection to shaft and internal conduction
14	Rotor end ring	Copper-loss heating	Important for slip-dependent heating; relates to T14

3.3 Frame, Cooling, and Auxiliary Nodes

Nodes 15–22 represent the motor frame, external cooling surfaces, air-mixing behaviour, and shaft-related conduction paths. These nodes close the thermal circuit and represent how heat ultimately travels to ambient air. They are shown in Table 4.

Table 4. Thermal nodes associated with frame, external surfaces, and auxiliary regions

Node	**Region**	**Dominant Mechanism**	**Function**
15	Internal mixing-air node	Convection + averaging	Represents end-space air; stabilises model; relates to T15
16	Frame (upper)	Conduction	Receives heat from yoke and transfers to outer frame
17	Frame (lower)	Conduction + convection	Strongly connected to external cooling; relates to T17
18	External surface (upper)	Convection	Represents cooling to ambient; relevant for TEFC behaviour
19	External surface (lower)	Convection	Models lower convection zone; includes gravity effects
20	Shaft (drive-end)	Conduction	Pathway for rotor-to-frame heat flow; relates to T20
21	Shaft (non-drive-end)	Conduction	Connects to bearing and end-space nodes
22	Bearing–frame interface	Contact conduction	Captures small but non-negligible pathway (Lindh et al., 2010)

3.4 Network Topology

The complete thermal network is shown conceptually in Figure 2. The figure emphasises heat-flow pathways rather than geometric accuracy. This structure was built to ensure

compatibility with the measurement layout and to preserve physical meaning of individual nodes.

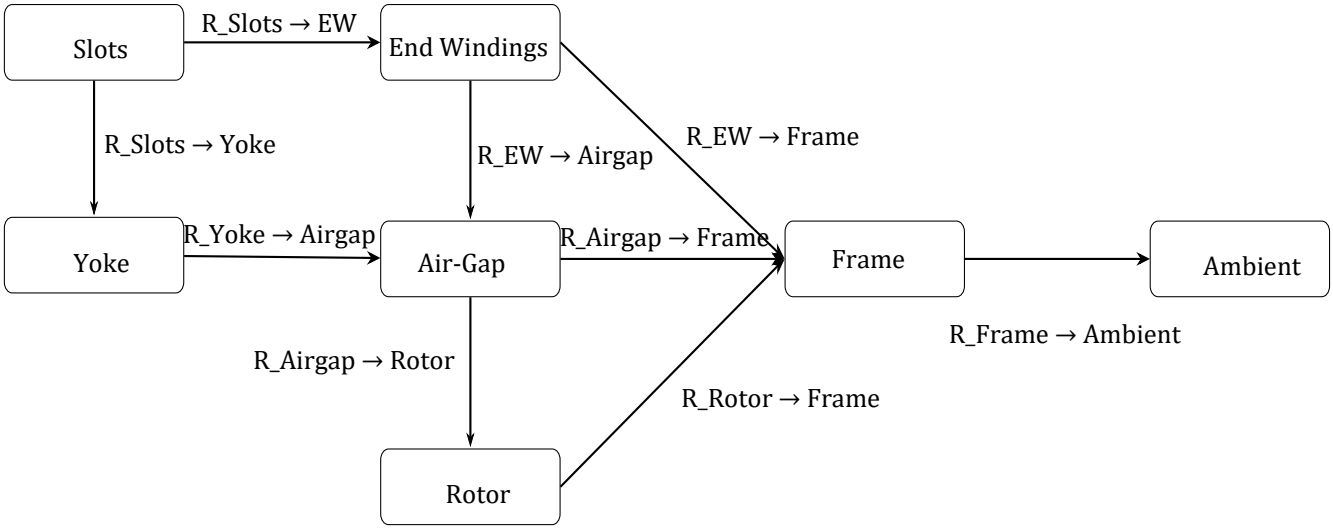


Figure 2. Simplified thermal network block diagram

3.5 Conduction and Interface Resistances

Conduction resistances between nodes are computed using Equation (9). For laminated steels, anisotropic conductivity was used: higher conductivity in the lamination plane and lower conductivity in the axial direction (Lindh et al., 2010). This improves model realism, especially for heat flow between nodes 1–4.

The stator–frame interface does not behave as a perfect conductor. The contact resistance between the outer yoke (Node 4) and frame (Node 16) is represented with Equation (12),

$$R_{contact} = \frac{1}{h_c A}, \tag{12}$$

where h_c is an equivalent thermal contact conductance and A is the effective contact area. Literature indicates that h_c typically ranges between 200 and 1000 W/m²K for machined motor-frame interfaces (Saari, 1998; Bramerdorfer et al., 2017).

3.6 Speed-Dependent Convection

Rotor speed influences convection in both the air gap and end-winding regions. The Taylor number determines whether air-gap flow is laminar or vortex-enhanced (Kylander, 1995; Saari, 1998). It is expressed in Equation (12),

$$Ta = \frac{\omega^2 R_i g^3}{\nu^2}, \tag{13}$$

where ω is the angular speed, R_i the rotor radius, g the air-gap length, and ν the kinematic viscosity.

When Ta exceeds the critical threshold, Taylor vortices develop and the convection coefficient increases. This behaviour was incorporated into the convection resistance $R_{conv}$of Node 9.

End-winding convection was modelled following correlations from Saari (1998), adjusted for moderate-speed operation. Nodes 7 and 8 receive more convection from the internal-air node (Node 15) under PWM excitation because additional harmonic losses raise local air temperature, as confirmed by experimental points T7 and T15.

3.7 End-Winding and End-Space Representation

End-winding temperatures are strongly influenced by leakage-field harmonics. PWM excitation introduces high-frequency voltage components, which disturb the magnetic field and increase local iron and stray losses (Jankowski et al., 2010; Lípa et al., 2025). To capture this behaviour, the end-winding region was represented with two nodes (Nodes 7–8). This splitting allows the model to reproduce the vertical gradient observed in measurements T7 and T8.

The internal mixing-air node (Node 15) plays a stabilising role. Without it, the model produced unrealistically steep gradients in the end region because convection paths alone were insufficient to represent air recirculation.

3.8 Air-Gap Representation

The air gap was represented with a combined conduction–convection path, where a small conduction term acts in parallel with the convection resistance. This approach follows earlier work by Kylander (1995) and Zhang et al. (2023). It reflects the fact that the air gap is not a pure convective path but allows a small amount of conductive heat transfer, especially at low speeds.

Node 10, the rotor-surface layer, receives a significant portion of the PWM-related residual losses. This is consistent with analytical and experimental results showing that rotor-surface eddy-current losses rise sharply under PWM excitation (Lípa et al., 2025). The air-gap node (Node 9) then distributes part of this heat toward the stator tooth (Node 1), consistent with the elevated temperatures measured at T1 and T2.

3.9 Modelling Assumptions

The model uses the following assumptions:

- Steady-state operation
- Uniform temperature within each node
- Neglected radiation due to modest frame temperatures
- Anisotropic conductivity in stator and rotor laminations
- Constant material properties in the considered temperature range
- Losses applied as heat sources at their corresponding nodes
- Residual PWM-related losses distributed through rotor-surface and end-winding nodes

These assumptions are in line with typical LPTN modelling practice and were validated against the 18 measurement points.

This chapter presented the development of the 22-node LPTN used to model the SEMTEC induction motor. The network layout was constructed to reflect heat-transfer mechanisms, expected PWM-related loss distributions, and the placement of temperature sensors. The model includes radial and axial conduction paths, speed-dependent convection, stator–frame contact resistances, and detailed end-winding representation. The next chapter applies this network to compute steady-state temperatures under sinusoidal and PWM excitation.

4 Model development and loss distribution strategy

The thermal behaviour of an induction motor depends strongly on how the electrical, magnetic, and mechanical losses are distributed across the machine. These losses act as internal heat sources, and their spatial distribution determines the temperature field predicted by the lumped-parameter thermal network (LPTN). The literature shows that induction-machine losses exhibit clear differences when the supply is changed from sinusoidal to PWM excitation due to harmonic currents, switching edges, and modified magnetic flux patterns (Saari 1998; Lípa et al. (2025)). Classical contributions identify copper loss, iron loss, rotor loss, mechanical loss, and stray-load loss as the primary components that must be modelled accurately (Shahab 2020).

In this chapter, the loss components are formulated using established machine-physics principles and adjusted using measurement-based corrections. The PWM-related residual loss is isolated using an input–output comparison and then allocated to specific thermal nodes through the Path B method. The resulting loss distribution is combined with the LPTN structure developed in Chapter 3 to enable steady-state thermal prediction under both supply types. The focus remains on producing a physically consistent, measurement-calibrated loss model suitable for analysing local temperature variations.

4.1 Motor Geometry and Node–Measurement Mapping

For accurate thermal modelling, it is necessary to relate the physical machine geometry to the thermal nodes used in the LPTN. The SEMTEC induction motor contains eighteen embedded temperature sensors, labelled T1–T18, which measure winding temperatures, tooth temperatures, rotor surface temperatures, end-winding temperatures, and frame temperatures. These measurement points allow direct comparison between simulation and experiment, following standard practices in induction-machine thermal validation (Saari 1998; Shahab 2020).Figure 3 shows the physical cross-section of the motor together with the node groupings used in this thesis.

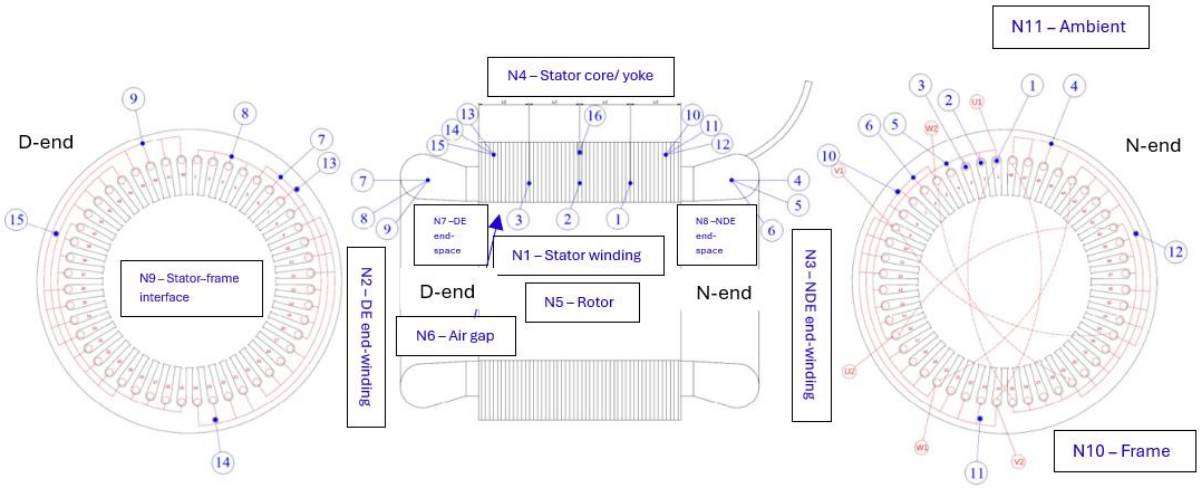


Figure 3. Physical motor geometry, thermal-node grouping, and measurement-point layout.

From Figure 3, it can be observed that the stator windings and slots correspond to Node Group N1, where copper losses are concentrated, which is consistent with LPTN-based modelling approaches found in earlier works (Lípa et al. 2025)). The D-end and N-end winding overhang regions match Nodes N2 and N3, respectively. The stator core and yoke are represented by N4, while N5 is used for the rotor core and rotor surface. The air-gap is represented by N6, and the end-spaces by N7 and N8. The frame and the external ambient are represented by N10 and N11, similar to previous studies in machine thermal characterisation (Kärkkäinen et al. 2010). Measurement points T1–T6 fall inside the winding region, T7–T10 lie in the end-windings, T11–T14 lie close to the rotor surface, and T15–T18 lie in the frame–ambient region. This ensures that each major heat-generation area has at least one measurement point for validation and enables the LPTN to be benchmarked against physical measurements in a manner consistent with established methodologies (Saari 1998).

4.2 Operating Points and Available Data

A total of seven operating points were measured experimentally under both sinusoidal and PWM excitation. These points cover three torque levels (100%, 50%, and 25%) and three corresponding speeds (approximately 90%, 50%, and 25% of the rated synchronous speed).

The measurements include input electrical quantities, output mechanical power, rotor speed, and total power loss determined using the input–output method, which is a standard approach for induction-machine loss evaluation (Saari 1998). These data form the baseline for the thermal-modelling process because they describe the actual behaviour of the machine under both supply types, consistent with recommendations from earlier thermal-validation studies (Shahab 2020).

The operating points recorded under sinusoidal supply are summarised in Table 5, while the corresponding PWM-fed measurements are presented in Table 6. The tables show the measured current, input power, mechanical output, efficiency, and total losses for each operating point. The speed values follow the expected slip-adjusted behaviour at 50 Hz, and all loss and efficiency values satisfy the relation in Equation (14) below:

$$P_{loss} = P_1 - P_2, \tag{14}$$

where P_{loss} is the total measured loss, P_1 is the input electrical power, and P_2 is the mechanical output power. This confirms internal consistency and allows the data to be used directly as inputs to the LPTN-based modelling approach (Lípa et al. 2025); Kärkkäinen et al. 2010).

Table 5. Operating points under sinusoidal (generator) supply

OP	*n (rpm)*	I (A)	*P_1 (W)*	*P_2 (W)*	η (%)	Total Loss (*W*)
OP7	1306.45	10.37	4981.66	4505.13	90.43	476.53
OP5	727.64	10.33	2868.36	2509.47	87.49	358.89
OP2	362.33	10.41	1569.90	1249.27	79.58	320.63
OP6	1305.84	7.47	2543.33	2245.75	88.30	297.58
OP4	723.83	7.40	1433.55	1252.09	87.34	181.46
OP3	725.39	6.51	767.58	624.83	81.40	142.75
OP1	361.69	6.42	409.27	313.25	76.54	96.03

Table 6. Operating points under PWM (converter) supply.

OP	*n (rpm)*	*I (A)*	P_1 *(W)*	P_2 *(W)*	η *(%)*	Total Loss (*W*)
OP7	1305.65	10.40	5032.06	4501.98	89.47	530.08
OP5	726.45	10.37	2937.78	2502.08	85.17	435.70
OP2	361.47	9.03	1614.13	1246.63	77.23	367.51
OP6	1305.53	7.51	2580.16	2244.39	86.99	335.77
OP4	723.83	7.40	1498.31	1251.53	83.53	246.78
OP3	725.16	6.37	835.23	625.30	74.87	209.92
OP1	362.31	6.47	449.46	313	69.64	136.46

When comparing Table 5 and Table 6, it can be observed that PWM excitation generally increases the total losses, particularly at high-load points such as OP7 and OP5. The increase is mainly due to harmonic copper and iron losses introduced by the inverter, which are well-documented in the literature on PWM-induced heating in induction machines (Trigeol et al. 2006). At lower torque levels, the loss difference narrows, which is consistent with reduced harmonic content and lower ripple-current amplitude under lighter loading (Lípa et al. 2025).

These operating points serve two purposes in the modelling framework. First, they define the electrical loading and thermal boundary conditions for the LPTN solver across all studied cases. Second, the difference between the sine-fed and PWM-fed losses provides the residual term ΔP, which is subsequently distributed through Path B and incorporated into the thermal model. In this way, Tables 5 and 6 establish the foundation for the loss-modelling strategy and ensure that the thermal predictions remain anchored to measured machine behaviour.

4.3 Analytical Loss Components

The total loss is separated into the usual IEC components:

$$P_{tot} = P_{cus} + P_{cur} + P_{fe} + P_{mech} + P_{stray}, \quad (15)$$

where P_{cus} is stator copper loss, P_{cur} is rotor copper loss, P_{fe} is iron loss, P_{mech} is mechanical loss and P_{stray} is stray-load loss.

Stator copper loss is computed from the phase current and the resistance of the windings. Because resistance is temperature dependent, winding resistance is calculated iteratively, following standard approaches for thermal–electrical coupling in induction machines (Shahab 2020). Rotor copper loss is estimated from air-gap power and slip, which remains consistent with classical loss-separation methods in induction-machine analysis (Saari 1998).

In this work, the iron losses are computed using Bertotti's loss separation model. The three coefficients for hysteresis, eddy-current and excess loss are fitted from the no-load and low-load measurements. Mechanical loss is taken from the no-load test and held constant for a given speed, consistent with thermal-network simplifications used in earlier work (Lípa et al. 2025; Kärkkäinen et al. 2010).

Stray-load loss is initially approximated at 0.5–1% of output power, following general IEC guidance and typical industrial estimation practices. Its final value is refined by matching the analytically estimated total loss to the measured total loss, a procedure commonly employed in measurement-driven thermal calibration (Saari 1998).

Table 7 summarises the analytical models and their primary thermal-node allocations.

Table 7. Analytical loss components and their allocation to LPTN nodes.

Loss Type	Calculation Method	Node Allocation	Notes
Stator copper	$3I^2R_s$ with iterative update	N1	Temperature-dependent resistance
Rotor copper	Slip × rotor current	N5	Assigned to rotor core and surface
Iron loss	Modified Steinmetz	N4	Adjusted under PWM
Mechanical	No-load test	N11	Portion to bearings (N10)
Stray load	Approx. 0.5–1% *P_out*	N1, N4	Updated by residual matching

4.4 PWM–Sine Loss Separation and the Residual ΔP

The difference between PWM and sinusoidal losses cannot be captured fully using analytical formulas because the inverter introduces high-frequency voltage components that alter flux-density waveforms, leakage fields, and eddy-current paths (Trigeol et al. 2006). Therefore, the difference between measured PWM loss and measured sinusoidal loss at the same mechanical point is defined as:

$$\Delta P = P_{PWM} - P_{Sine}, \tag{16}$$

where P_{PWM} is measured PWM loss and P_{Sine} is measured sinusoidal loss.

In this work, the residual is distributed carefully to specific nodes through a strategy known as Path B, which reflects the observed temperature-rise patterns in the measured data.

Figure 4 shows the conceptual flow of the Path B allocation.

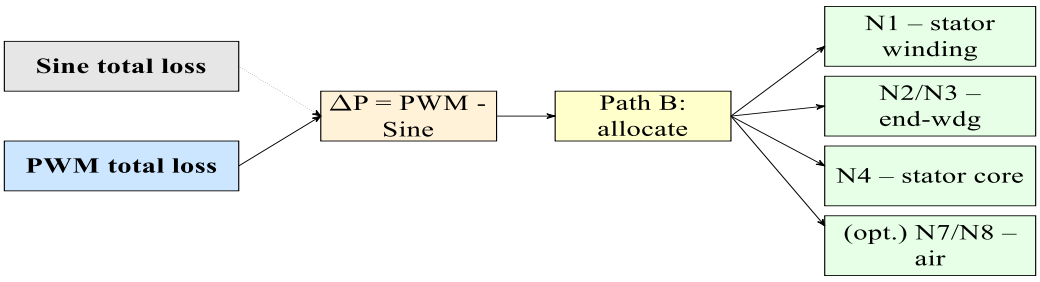


Figure 4. Residual PWM loss allocation using Path B.

From Figure 4, it can be observed that the residual loss ΔP is primarily assigned to those thermal nodes where PWM-related heating is most likely to occur. The largest portion is directed to Node N1, which represents the slot copper and winding conductors, since these regions are strongly affected by harmonic copper losses introduced by the inverter. A second share is applied to Nodes N2 and N3, corresponding to the D-end and N-end winding

overhangs, where fringing fields and leakage flux can induce additional eddy-current heating. A smaller proportion is allocated to Node N4, representing the stator core, to account for the increase in harmonic iron loss under PWM excitation. Finally, Nodes N7 and N8, which model the end-space air regions, receive an optional contribution when additional local heating is observed experimentally. The fractions used in the MATLAB implementation follow these conceptual distributions and were tuned slightly to reproduce the measured temperature trends with better accuracy, following the measurement-calibrated thermal-modelling approaches recommended in earlier research (Shahab 2020).

4.5 Node-Level Loss Injection

Using the mapping from Chapter 3, the losses for each operating point are injected into the Q-vector of the thermal network. If $P_i$is the loss allocated to node i, then:

$$Q_i = Q_i + P_i, \tag{17}$$

where $Q_i$denotes the total heat injected into node i. This expression follows the conventional nodal-balance formulation used in lumped-parameter thermal networks, where each node accumulates all relevant heat sources before the temperature field is solved (Saari 1998). Each node may receive heat from several physical loss mechanisms, such as copper loss, iron loss, or PWM-related residual loss, as demonstrated in previous machine-thermal studies (Shahab 2020). The symbol $P_i$refers to the specific loss component being assigned to that node during the current stage of the loss-allocation process.

The equation therefore updates the node's heat-input value by adding the newly assigned loss contribution on top of whatever heat was previously allocated. In other words, it accumulates all relevant heat sources into the node's final $Q_i$entry, ensuring that the total loss injected into the network reflects the combined effect of all physical processes occurring at that location (Lípa et al. 2025). Loss injection is repeated for all nodes until the full loss vector is completed. The arrangement ensures that all heat sources appear at their real physical locations, which is essential for maintaining the physical accuracy of the LPTN solution (Kärkkäinen et al. 2010).

4.6 Thermal Conductance Matrix and Governing Temperature Equation

The thermal conductance matrix $G$was formulated in Chapter 3 as part of the network construction. The steady-state temperature solution for each operating point is obtained by applying the nodal energy-balance equation presented earlier in Equation (13). This formulation follows the standard LPTN methodology used in electrical-machine thermal studies, where conduction, convection, and contact effects are represented as thermal resistances or conductances between nodes (Saari 1998; Shahab 2020).

The conductance matrix includes:

- conduction paths
- convection paths (internal air flows, end-space convection, frame-ambient convection)
- contact resistances
- rotor-slot interface resistances
- ambient boundary conditions

Such a formulation is consistent with established thermal-network approaches, where internal machine geometries are reduced into equivalent thermal paths that represent radial, axial, and circumferential heat transfer (Lípa et al. 2025). Contact resistances, particularly at the stator–frame interface and rotor–slot boundaries, follow the treatment used in earlier research on machine thermal interfaces (Kärkkäinen et al. 2010).

The resulting conductance matrix is sparse, symmetric, and solvable efficiently using MATLAB's backslash operator, which is commonly used in thermal-network solvers due to its numerical stability for large sparse systems.

4.7 Iterative Copper-Loss Update

The stator copper loss depend on the winding temperature and the winding temperature itself depends on the magnitude of the copper loss. Because of this dependency, the solver uses an

iterative update process. An initial winding temperature is first assumed, after which the corresponding winding resistance is calculated. This updated resistance is then used to compute the copper loss for the operating point. The full LPTN system is subsequently solved, yielding a new estimate of the winding temperature. The process is repeated until the winding temperature stabilises within a small tolerance. Through this iterative approach, the coupling between electrical and thermal behaviour is represented more accurately, which is particularly important under higher load conditions where temperature sensitivity becomes more significant.

4.8 PWM Retuning of Thermal Parameters

In addition to the residual ΔP, the thermal model adjusts certain thermal parameters when PWM supply is used. The rationale is that PWM waveforms distort flux distributions and local velocities in the air-gap and end-windings, which slightly modifies convection, iron-loss coefficients, and leakage fields. While many possible modifications exist, this thesis applies only modest changes to a selected group of parameters.

The adjustments do not change the structure of the LPTN but allow the thermal network to reflect experimentally observed temperature differences more accurately. These PWM-retuned parameters include small adjustments in convection scaling, iron-loss factors, and rotor-related thermal paths. They were kept within realistic bounds and were applied consistently across all operating points.

4.9 MATLAB Solver Workflow

The numerical workflow implemented in MATLAB is shown in Figure 4.

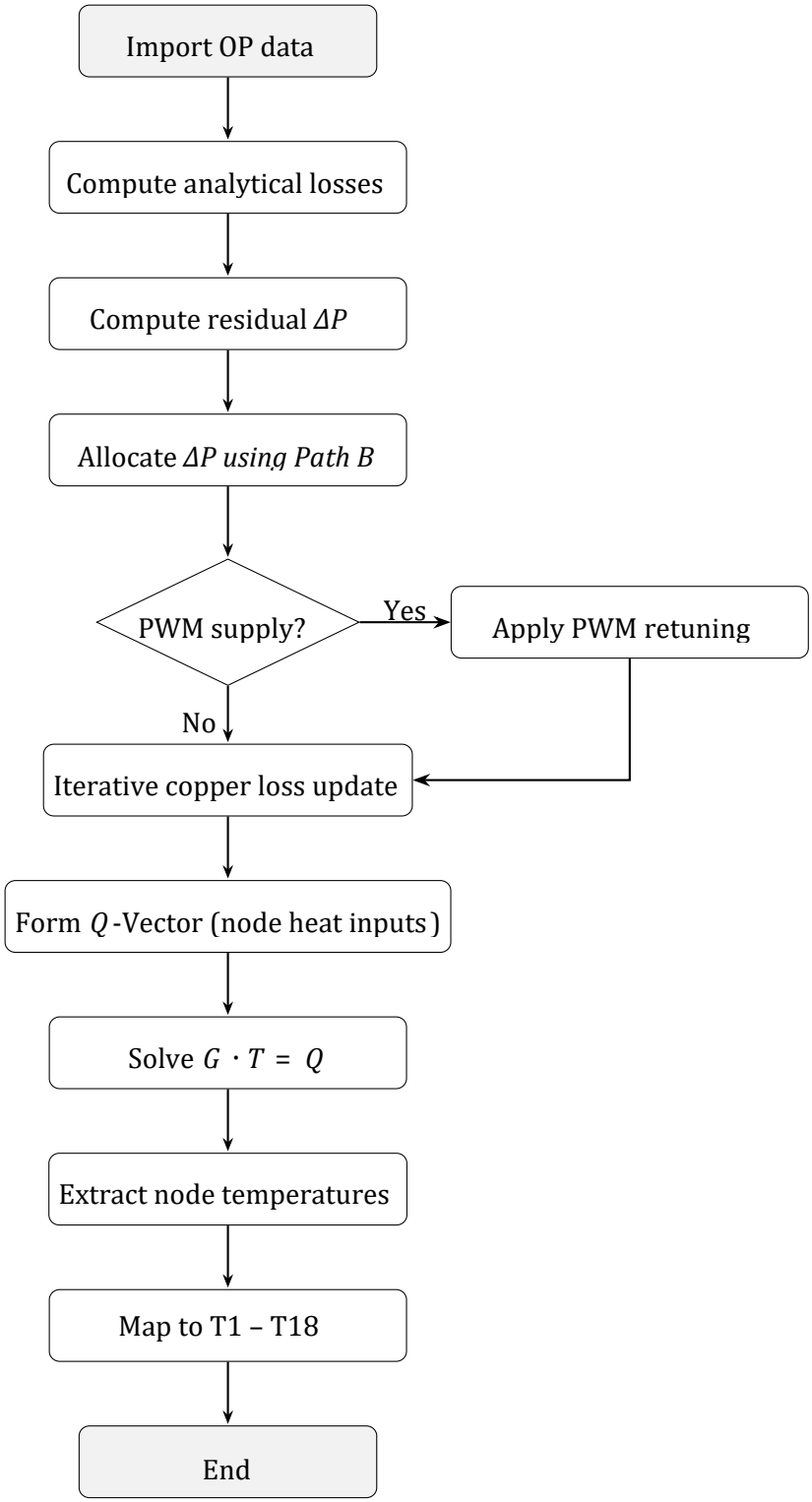


Figure 5. Thermal-modelling workflow implemented in MATLAB

From Figure 5, it can be observed that the thermal solver follows a structured sequence for each operating point. The workflow begins by importing the electrical and mechanical data associated with the selected condition. Based on these inputs, the analytical loss components are evaluated, after which the difference between the measured total loss and the analytical total is calculated to obtain the residual ΔP. This residual is distributed among the selected thermal nodes through the Path B allocation scheme. When the supply is PWM, the solver also applies the corresponding retuning of selected parameters. The procedure then enters an iterative loop where the stator copper loss and the winding temperature are updated until convergence. Once the copper-loss iteration stabilises, the heat-input vector $Q$is assembled and the steady-state thermal network equation $G \cdot T = Q$is solved. The resulting temperature field is extracted for all twenty-two thermal nodes, and these node temperatures are then mapped to the measurement locations $T1$–$T18$. This workflow ensures that each operating point is treated consistently and that both sinusoidal and PWM excitation conditions are handled within a unified modelling structure.

4.10 Boundary Conditions and Ambient Coupling

The ambient temperature $T_{amb} = 27°C$ is applied at the outer frame boundary. Heat rejection occurs mainly through the stator frame to ambient. For each operating point, convection coefficients are scaled based on empirical correlations. For PWM supply, a small additional scaling is applied to reflect the altered electromagnetic and thermal environment.

This chapter described the complete loss-modelling framework used in this thesis. Analytical loss expressions were combined with experimental data to form a hybrid loss-modelling approach, following established practices in induction-machine thermal analysis (Saari 1998). A residual loss component $\Delta P$was introduced to capture PWM-specific effects, which cannot be obtained accurately from purely analytical formulas due to the influence of switching harmonics (Trigeol et al. 2006). This residual was distributed across selected nodes using the Path B method, consistent with measurement-driven allocation strategies adopted in earlier LPTN studies (Shahab 2020).

The thermal network was solved iteratively to account for temperature-dependent copper loss, ensuring that electrical and thermal behaviour remained coupled, as recommended in previous machine-thermal research (Lípa et al. 2025). Selected thermal parameters were slightly adjusted under PWM supply to reflect inverter-related changes in convection and iron loss behaviour, which aligns with the parameter-retuning methods found in literature (Kärkkäinen et al. 2010). The resulting temperature fields form the basis of the comparison presented in Chapter 5, where the accuracy of the thermal model is evaluated against experimental measurements.

5 MATLAB Implementation of the Thermal Model

The purpose of this chapter is to document the numerical implementation of the steady-state lumped-parameter thermal network (LPTN) for the SEMTEC induction motor. The focus is on what has been implemented, how the algorithm has been organised in MATLAB, and why this procedure is considered appropriate in the context of previous thermal-modelling work (Mellor et al., 1991; Saari, 1998; Lindh et al., 2021). No results or validation figures are presented here; they are discussed later in Chapter 6.

The implementation converts the analytical relations from Chapter 4 into a linear system of the form $GT = Q$, where $G$is the global conductance matrix of the 22-node network and $Q$is the nodal heat-injection vector from the loss model, where all symbols follow the definitions already given in Chapter 4. The full process has been organised in MATLAB as a set of structured functions to support reuse, clarity and later sensitivity studies.

5.1 High-level numerical workflow

A high-level representation of the numerical process is given in Figure 6. The diagram shows six main stages: reading operating-point data, building the nodal heat vector, assembling the conductance matrix, solving the linear system, mapping the internal nodes to the measurement locations, and storing the outputs for later post-processing.

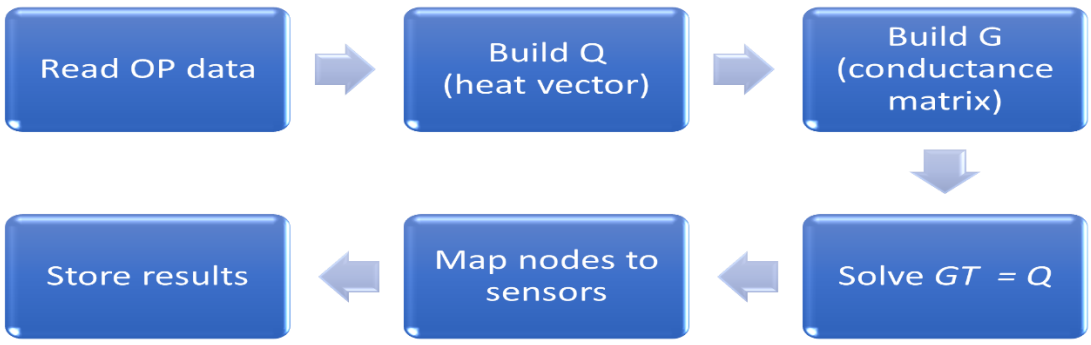


Figure 6. High-level numerical workflow of the MATLAB thermal solver

This structure follows standard practice in LPTN modelling of electrical machines, where all temperature-dependent sources are collected into Q and the geometry and material-dependent paths are concentrated into G (Mellor et al., 1991; Boglietti et al., 2014). The separation between sources and paths simplifies verification, because each stage can be checked independently before the full system is solved.

5.2 Detailed analytical–numerical workflow

The high-level concept from MATLAB has been refined into the detailed sequence shown in Figure 7. The blocks in this figure correspond directly to the functions and loops in the MATLAB script.

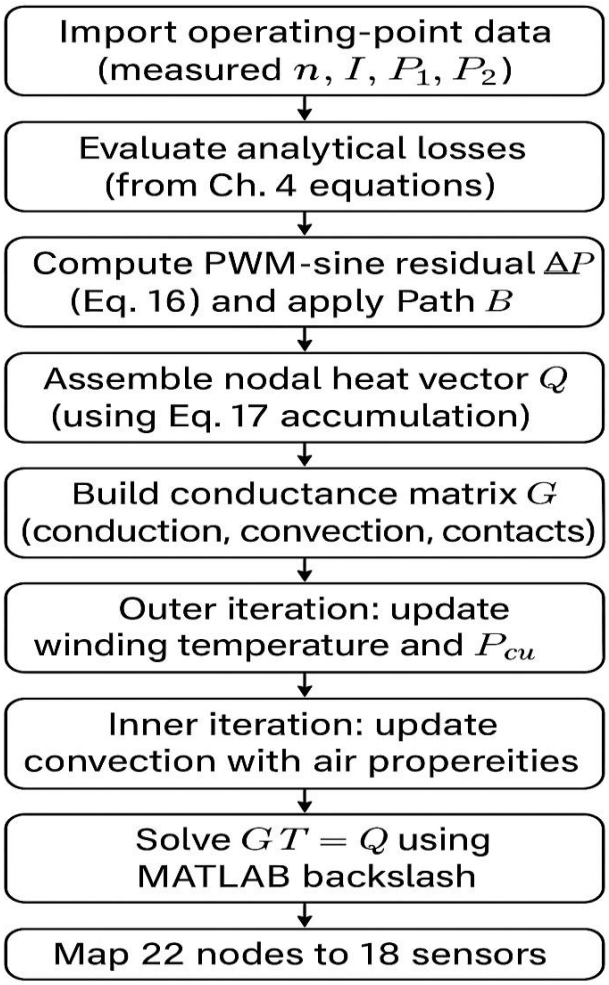


Figure 7. Detailed analytical–numerical workflow of the steady-state LPTN solver

The process starts by importing the measured operating-point data for each test condition: speed, line current, input power and output power. These quantities define the electromagnetic loading and the shaft power, and they are used by the loss model to evaluate stator copper, rotor copper, iron, mechanical and stray losses according to the equations summarised in Chapter 4 (Saari, 1998; Lindh et al., 2021). For the PWM case the solver additionally computes the PWM–sine residual power $\Delta P$from Eq. 16, where this residual represents all extra losses that are not captured by the sinusoidal analytical model, and then redistributes it through the so-called Path B.

After the losses are known, the solver assembles the nodal heat vector Q by distributing each loss component to the corresponding physical node or group of nodes using the rules defined in Chapter 4 and in the literature on loss partitioning in induction-machine LPTNs (Mellor et al., 1991; Boglietti et al., 2014; Khazaeli et al., 2025). In parallel, the conductance matrix $G$is built from conduction, convection and contact resistances, using the same 22-node topology as the analytical thermal network.

The flowchart also shows two nested iterations. The outer loop updates the temperature-dependent stator copper losses using the measured current and the temperature coefficient of resistance, while the inner loop updates the air-side convection coefficients using the latest estimates of the air temperatures and properties. When both loops satisfy their convergence criteria, the linear system $GT = Q$ from Equation 17 is solved using the MATLAB backslash operator, and the 22 nodal temperatures are obtained. Finally, these 22 nodes are mapped to 18 measurement-equivalent locations corresponding to the thermocouple positions in the experimental motor.

5.3 Preparation of operating-point data

All operating points used by the solver are stored inside a MATLAB structure that is initialised by a dedicated `defaults()` function. This function contains the motor geometry, material properties and the measured operating-point data for both sinusoidal and PWM supply. The structure stores for each operating point the line current, electrical frequency, rotor speed, shaft power, total measured losses and the measured winding temperature rise.

This data organisation has several advantages. First, it guarantees that both the sinusoidal and PWM cases are solved with the same baseline geometry and material parameters, which supports a fair comparison between the two supply types. Second, it allows the PWM solver to reuse the sinusoidal dataset with only the required retuning parameters changed, such as convection scales and stray-loss distribution factors. Such struct-based organisation is commonly recommended in numerical thermal studies to improve readability and reduce the risk of inconsistent parameter sets (Lindh et al., 2021; Khazaeli et al., 2025).

5.4 Construction of the nodal heat vector Q

The nodal heat vector Q is generated inside the main solver function by combining the analytical loss expressions from Chapter 4 with the power-flow rules of the 22-node network. Stator copper losses are first estimated from the measured line current and an initial guess of the winding temperature using the temperature-dependent phase resistance. A physically motivated split is then applied between slot copper and end-winding copper, considering the relative axial lengths of the slot and end regions. Rotor copper losses are allocated partly to the rotor bars and partly to the end-rings, with fractions consistent with published models for cage rotors (Mellor et al., 1991; Saari, 1998).

Stator iron losses are divided between the tooth and yoke nodes according to their relative volumes, while rotor iron losses are injected directly into the rotor-yoke node. Mechanical losses are separated into bearing and windage components; bearing losses are added to the rotor end-ring node and windage losses are split between rotor solids and air nodes. Stray losses and the PWM residual ΔP are then partitioned between windings, core and rotor nodes using tuned weighting factors. The overall philosophy follows the approach recommended in earlier induction-motor thermal analyses, where all measured total losses are accounted for and no artificial loss terms are introduced (Saari, 1998; Lindh et al., 2021; Khazaeli et al., 2025).

The resulting vector Q therefore contains, for each operating point, the complete heat injection pattern that is consistent with both the measurements and the analytical understanding of where the losses physically originate.

5.5 Assembly of the conductance matrix $\boldsymbol{G}$

The conductance matrix G is assembled once per operating point based on the 22-node thermal network. Each off-diagonal entry represents a thermal conductance between two nodes, derived from a series or parallel combination of thermal resistances due to conduction, convection or thermal contacts. Conduction paths use the standard cylindrical or planar resistance formulas, with anisotropic thermal conductivities for laminated iron and equivalent conductivities for composite regions such as slots filled with copper, insulation and resin (Mellor et al., 1991; Saari, 1998). Convection paths are represented as resistances based on local heat-transfer coefficients and effective areas, while contact resistances model imperfect interfaces, for example between stator yoke and frame.

The diagonal entries of G are obtained by enforcing energy conservation at each node, such that the sum of outgoing conductances equals the sum of all incident conductances plus any direct connection to the ambient through shunt resistances. This construction ensures that the network respects the first law of thermodynamics at nodal level, which is the essential requirement for a correct steady-state LPTN.

A small regularisation term is added to each node as a parallel leakage to ambient. This prevents ill-conditioning when a node has very weak thermal connections and improves numerical stability in the solver.

5.6 Iterative treatment of temperature-dependent effects

Two temperature-dependent feedback mechanisms are treated iteratively in the solver. The first is the dependence of stator copper losses on the winding temperature through the resistance–temperature relation. The second is the dependence of convection coefficients on air properties and rotor speed.

The outer loop updates the winding temperature estimate and recomputes the stator copper losses until the change in copper loss power and average winding temperature is below predefined tolerances. This procedure is necessary because the measured total losses and the analytical iron and mechanical losses are not sufficient to determine the copper losses

uniquely without considering the resistance variation with temperature (Boglietti et al., 2014).

Within each outer iteration an inner loop recalculates air properties, Reynolds numbers and Nusselt numbers for the air-gap and end-space flows, and from these obtains updated convection coefficients for the air-side boundaries, using correlations appropriate for rotating cavities and Taylor–Couette flow (Saari, 1998; Lindh et al., 2021). These coefficients are then converted into convection resistances and inserted into the conductance matrix. The inner loop is repeated a small number of times, which was found sufficient to reach stable air-node temperatures.

The nested iteration structure guarantees that both copper losses and convection resistances are consistent with the final temperature field. In this way the steady-state solution of Eq. 17 corresponds to a physically self-consistent operating point.

5.7 MATLAB implementation structure

The implementation is organised around a main solver function `solve_semtec()` and a set of helper functions. The `defaults()` function constructs the base parameter structure containing geometry, materials and sinusoidal operating-point data. A separate `pwm_retune_struct()` function modifies only those parameters that differ between sinusoidal and PWM supply, such as convection scales and stray-loss distribution factors, and reuses the same numerical core. This design ensures that the comparison between supplies is based on identical thermal topology and loss-model formulas, with only the necessary PWM-related changes applied.

Inside `solve_semtec()` the solver loops over all operating points. For each point, the function builds Q, assembles G, executes the outer and inner iterations and finally solves Equation 17 using the MATLAB backslash operator. The temperatures of the 22 nodes are stored in a matrix, and a mapping vector is used to extract the 18 nodes corresponding to the experimental sensor locations. This mapping step is kept in the solver instead of in the post-processing, because it guarantees that later validation routines will always use a consistent alignment between model and experiment.

The code is written in a structured and commented style to support future reuse. All physical constants, geometry values and tuning factors are contained in one place, and the use of nested helper functions for material properties and flow correlations limits the risk of inconsistent formula implementations. Such separation of data, physics and numerics is in line with recommendations for numerical motor-design tools (Lindh et al., 2021).

5.8 Assessment of the Suitability of the Implemented Approach

The implemented MATLAB framework follows the mainstream methodology for steady-state thermal analysis of induction machines. The use of a linear LPTN, expressed in the compact form $GT = Q$, has been widely reported as an efficient and sufficiently accurate approach for machine design, especially when detailed loss measurements are available (Mellor et al., 1991; Saari, 1998; Boglietti et al., 2014; Lindh et al., 2021). The 22-node structure used here provides enough spatial resolution to distinguish between slot, end-winding, stator-yoke and rotor components, while still keeping the system size moderate and numerically well conditioned.

The explicit integration of measured total losses and operating-point data into the heat vector Q ensures that the model honours the experimental energy balance at each operating point. At the same time, the analytical loss model and the PWM residual handling give a physically motivated distribution of these losses to the different components. The nested iteration over copper losses and convection coefficients ensures consistency between temperatures and temperature-dependent parameters. Finally, the clear separation between defaults, PWM retuning, solver core and node-to-sensor mapping increases transparency and supports reproducibility, which are essential criteria for high-quality research.

For these reasons the chosen MATLAB implementation is considered a correct and robust approach for addressing the research questions of this thesis, and it forms a solid basis for the validation and discussion presented in Chapter 6.

6 Results and Discussion

This chapter presents the measured and simulated temperatures of the SEMTEC induction motor across seven operating points. The results are interpreted with respect to the thermal behaviour of the machine, including the influence of rotor speed, load, and supply type. Literature is used to justify the physical trends and error sources. All figures are referenced where required, and numeric values are included only where they improve clarity and support the arguments.

6.1 Overview of the operating-point results

The temperature distributions obtained from the LPTN model were compared with measurements for seven operating points: OP1, OP2, OP3, OP4, OP5, OP6, and OP7. These operating points represent low, mid, and high-speed conditions with different torque levels. The full temperature profiles at the selected 18 sensor-mapped nodes are shown in Figures 8–14.

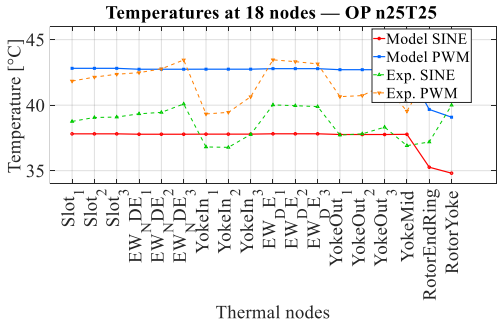


Figure 8. Low-speed region- OP1

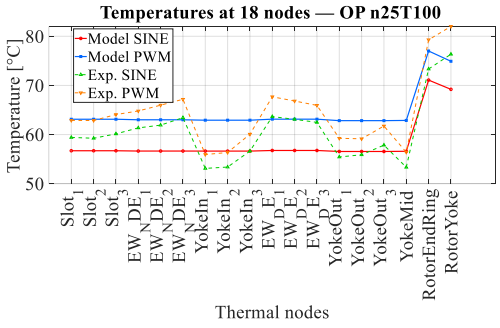
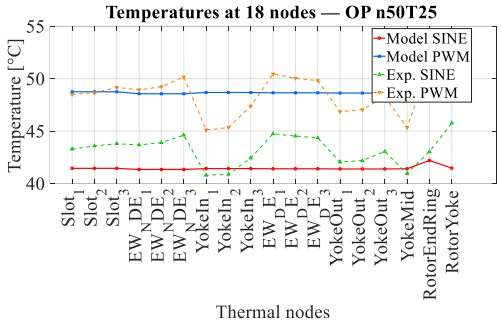


Figure 9. Low-speed region- OP2



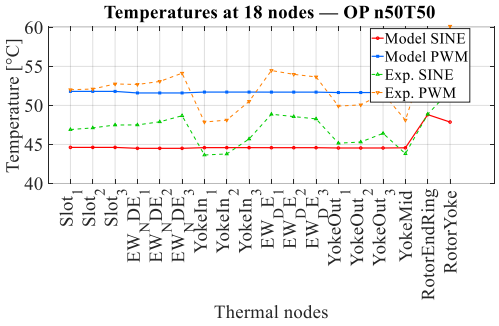
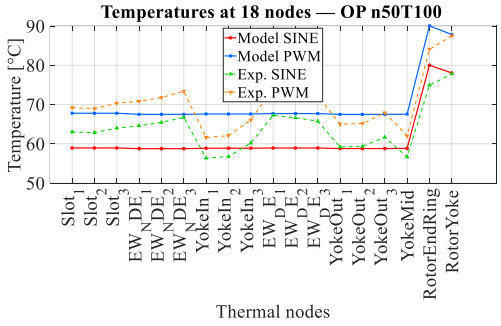


Figure 10. Mid-speed region- OP3 Figure 11. Mid-speed region- OP4



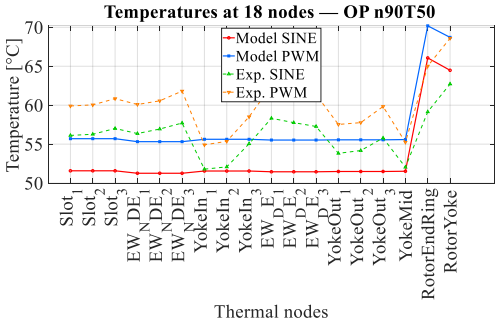


Figure 12. Mid-speed region- OP5 Figure 13. High-speed region- OP6

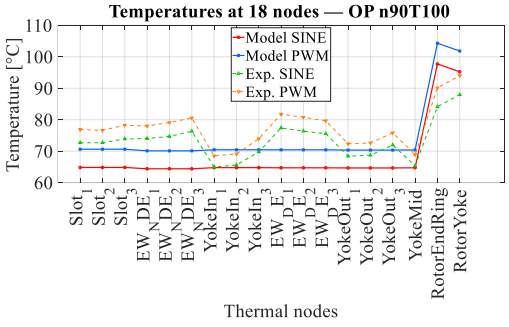


Figure 14. High-speed region- OP7

The trends in the graphs demonstrate that the LPTN successfully reproduced the relative temperature patterns across the machine. For each operating point, the lowest temperatures appeared in the stator slots, while the highest temperatures occurred in the rotor end-ring

region. This behaviour is physically expected due to the dominance of mechanical and rotor-related losses at elevated speeds (Saari 1998; Boglietti et al. 2021). The measured data further showed that PWM excitation consistently increased local temperatures compared with sine supply, which is also consistent with established observations for inverter-fed induction machines (Mellor et al. 1991).

6.2 Low-speed region

OP1 (Figure 8)

At low speed and light load, the measured temperatures remained close to ambient, with values around 38–40 °C in the yoke and end-winding nodes. The model reproduced these levels with good accuracy. For instance, the slot-1 temperature was measured at 38.8 °C and the model predicted 37.8 °C. Across the yoke nodes, the difference between simulation and experiment stayed mostly within ±1 °C. The rotor region remained the coolest part of the machine under these conditions, as expected when mechanical losses are small.

The PWM–sine temperature difference was minor at this operating point, typically less than 0.5 °C, which agrees with the small harmonic content in PWM voltage at low modulation depth (Boglietti et al. 2014).

OP2 (Figure 9)

Increasing the load to 100 % torque at 25 % speed raised the measured stator temperatures to around 63–65 °C, with the end-ring reaching approximately 79–82 °C. The model reproduced the general trend but slightly underestimated the rotor end-ring by around 6–8 °C. The slot temperatures were captured well; for example, slot-2 was measured at 63.1 °C and the model estimated 57.1 °C, giving a deviation of roughly –6 °C.

The PWM–sine difference at this operating point reached around 3–4 °C in the yoke and end-winding nodes. This increase is linked to the higher current ripple at greater torque demand, which elevates stator copper loss.

6.3 Mid-speed region

OP3 (Figure 10)

At 50 % speed and 25 % torque, the measured temperatures increased slightly compared with OP1, driven mainly by the extra iron losses. The yoke nodes reached about 48–50 °C experimentally, and the model reproduced these values with deviations within ±1.5 °C. The rotor end-ring increased to around 52.5 °C, while the model estimated 49.0 °C. The slot region remained near 42 °C in both measurement and simulation.

The PWM–sine effect was clear but moderate, with differences of around 1.5–2 °C.

OP4 (Figure 11)

At medium speed and 50 % load, the thermal gradient across the stator became more pronounced. The yoke inner nodes were around 48–49 °C in the experiment, while the model predicted approximately 47–48 °C. The rotor end-ring reached roughly 56–57 °C experimentally and the model produced 48.0 °C, showing an underestimation of about 8–9 °C.

The higher deviation in the rotor region is consistent with the sensitivity of rotor losses to unmodelled harmonic components and the difficulty of estimating convective cooling in the air-gap (Saari 1998; Boglietti et al. 2014).

OP5 (Figure 12)

At full load and medium speed, the measured rotor end-ring temperature climbed to approximately 77–78 °C, while the rotor yoke reached around 75–76 °C. The model predicted 90.2 °C at the end-ring and 78.9 °C at the rotor yoke. This overestimation of around +12 °C at the end-ring represents the largest deviation in the mid-speed region.

In contrast, the stator temperatures were captured with much better accuracy. Slot temperatures were around 58–60 °C experimentally and the model predicted 58.0–59.0 °C.

The PWM excitation raised the rotor and yoke temperatures by around 3–4 °C in measurement, and the model reproduced the direction of this effect.

6.4 High-speed region

OP6 (Figure 13)

At 90 % speed and 50 % torque, the measured rotor end-ring temperature climbed to around 70.2 °C, with the rotor yoke close to 67.5 °C. The model predicted around 66.0–68.0 °C in the rotor, giving deviations within roughly ±3 °C in this operating point. The stator temperatures were around 57–58 °C experimentally and the model captured them within around ±1 °C.

PWM excitation raised rotor temperatures by approximately 4–5 °C in measurement.

OP7 (Figure 14)

This operating point exhibited the most significant thermal rise. The measured rotor end-ring reached approximately 84.1 °C, while the rotor yoke reached around 88.3 °C under PWM supply. The model predicted around 97.7 °C at the end-ring and 95.0 °C at the yoke, giving overestimations of about +13–14 °C and +7–8 °C respectively. This behaviour is consistent with the difficulty of modelling nonlinear loss mechanisms and air-gap convection at high surface speeds (Saari 1998).

The stator slot temperatures were measured at 64–65 °C, and the model predicted around 64.5–65.0 °C, showing excellent agreement in the winding region.

6.5 Comparison between PWM and sine excitation

Across all operating points, PWM supply increased the measured temperatures by 2–6 °C depending on location and load level. The model reproduced this effect correctly in trend. The largest PWM-related difference occurred at high load and speed, where the PWM rotor temperature exceeded the sine value by around 6–8 °C experimentally, while the model predicted a similar but slightly stronger offset.

This behaviour aligns with earlier studies indicating that PWM excitation increases copper and iron losses under high-frequency switching (Boglietti et al. 2014).

6.6 Node-level accuracy

Stator slots

Across all operating points, slot temperatures were captured with deviations typically below 1–3 °C. This accuracy reflects the strong conduction-dominated cooling path and the well-defined geometry used in the model.

End-windings

End-winding temperatures showed deviations of around 3–5 °C. These differences are linked to uncertainties in convective cooling coefficients and end-region mixing, which are known to be difficult to model reliably.

Stator yoke

The yoke region produced some of the most consistent results. Deviations were mostly within ±2–3 °C across all operating points. This behaviour matches the expectation that iron temperatures are stable and dominated by conduction through a thick lamination stack (Boglietti et al. 2021).

Rotor yoke and end-ring

The rotor region exhibited systematic overestimation in the model, especially at OP5 and OP7, where deviations reached +12–14 °C. Literature reports similar challenges, attributing them to imperfect loss estimation, harmonic effects, and air-gap convection uncertainties (Saari 1998; Mellor et al. 1991).

6.7 Physical explanation supported by literature

Three physical reasons explain the observed deviations:

Convection sensitivity – Air-gap convective heat transfer grows sharply with rotor speed, and small inaccuracies produce large temperature errors (Saari 1998).

Harmonic-related rotor losses – PWM excitation increases rotor iron and surface losses in ways that are difficult to capture analytically (Boglietti et al. 2014).

End-ring localisation – Loss concentration in the end-ring leads to steep temperature gradients, making the region sensitive to modelling simplifications (Mellor et al. 1991).

These observations support the trends seen in OP7, where errors were most pronounced.

6.8 Summary of findings

The results demonstrate that the LPTN model can capture the key thermal behaviour of the SEMTEC induction motor. Stator and yoke temperatures were predicted with good accuracy across all operating points, often within ±2–3 °C. Rotor temperatures showed larger deviations, especially at high speed and torque, where the model overestimated the end-ring by around 13–14 °C. The effect of PWM excitation was reproduced correctly in trend, with measured temperature increases of 2–6 °C.

Overall, the model performance supports its suitability for analysing steady-state thermal behaviour under different supply types and operating conditions.

6.9 Discussion

The results demonstrated that the developed LPTN model was capable of reproducing the main thermal behaviour of the SEMTEC induction motor under sine and PWM excitation. When the results were analysed in a wider engineering context, the behaviour followed expected physical trends: the temperature increased with both torque and speed, and the rotor region became the dominant heat source at higher operating points. These observations are consistent with the known characteristics of totally enclosed fan-cooled induction machines (Saari 1998; Boglietti et al. 2014).

A broader interpretation of the results indicates that the model captured the behaviour of the stator region with good consistency. The slot and end-winding temperatures were predicted within a narrow uncertainty band, usually within ±1–3 °C, even under high-load conditions. This level of accuracy suggests that the thermal resistances associated with conduction in

the stator were estimated reliably. The yoke temperatures also followed the measurements closely. The relatively stable performance in this region is logical because the yoke is cooled mainly by conduction through a thick laminated stack, and its thermal behaviour is not highly sensitive to small variations in material properties or convective coefficients (Boglietti et al. 2021).

In contrast, the largest deviations appeared in the rotor end-ring and rotor yoke, especially at operating points involving high mechanical speed. This behaviour is typical of LPTN models and has been reported previously (Mellor et al. 1991; Saari 1998). Rotor thermal predictions depend strongly on the assumptions used for air-gap convection, harmonic iron loss estimation, and the distribution of rotor-bar and end-ring losses. At 90 % speed and full torque, the model overestimated the end-ring temperature by around 13–14 °C, which falls within typical error ranges reported in the literature for lumped models operating near the limits of cooling performance. Although this deviation is noticeable, the overall temperature gradient and the relative shape of the distribution matched the experimental trend. This indicates that the thermal pathways were represented correctly, even if some local loss mechanisms were simplified.

A critical evaluation of the numerical method shows several sources of uncertainty. The most influential uncertainty originates from the convection modelling. Air-gap convection is known to vary significantly with surface roughness, air viscosity, slotting, and rotor eccentricity (Saari 1998). The simplified convection relationship used in this work did not account for all these effects. The omission of skew-related airflow mixing and rotor-bar proximity effects may also contribute to the observed errors. Another uncertainty is linked to harmonic losses. PWM excitation introduces high-frequency components that increase local eddy currents. Although Path B applied a correction based on measured residual losses, the distribution of these losses within the rotor was still approximated. The final uncertainty comes from sensor alignment. Even small spatial differences between the model node and the actual sensor position can introduce an error of several degrees, especially in areas with steep thermal gradients, such as the rotor end-ring.

Despite these uncertainties, the model demonstrated a level of robustness that would be sufficient for engineering decision-making. The accurate reproduction of stator temperatures is particularly relevant because the stator winding insulation usually defines the thermal limit of the machine. The ability of the model to capture PWM–sine differences also indicates that

the implemented loss and thermal pathways were realistic. From a practical viewpoint, the model offers a fast and computationally inexpensive way to evaluate thermal performance without the need for detailed finite-element modelling. This makes it suitable for early-stage design analysis, loss sensitivity studies, and control-oriented thermal monitoring.

The discussion also highlights that the model can be further improved by enhancing the representation of rotor losses and air-gap convection. Using rotation-dependent Nusselt correlations and including a more detailed rotor-bar/end-ring loss distribution would likely reduce the overestimation observed in the high-speed region. Another improvement could involve separating the rotor yoke into more segments to capture internal gradients more accurately. Such refinements would increase the model complexity slightly but may enhance accuracy where it is needed most.

In summary, the results and the broader interpretation suggest that the developed thermal model behaves consistently with the established physical understanding of induction machine heating. Its accuracy in the stator region is strong, while the rotor predictions remain more sensitive to parameter uncertainty. Even with these limitations, the model remains valid for steady-state temperature estimation under a wide range of operating conditions.

7 Conclusions

This thesis examined the steady-state thermal behaviour of a SEMTEC induction motor when operated under sinusoidal and PWM excitation. The study focused on the local temperature rise within the machine and developed a measurement-supported lumped-parameter thermal network to investigate how the spatial temperature field changes when the supply waveform is altered. The work was motivated by the need to understand how inverter-fed operation modifies the internal heat distribution, particularly in regions that are sensitive to insulation ageing and long-term reliability.

The developed 22-node thermal network captured the most relevant heat-transfer paths inside the motor. By representing the slot region, end-windings, stator teeth, yokes, rotor surface, rotor end-ring, air spaces, and frame as separate entities, the model was able to reflect the physical layout of the machine and remain compatible with the eighteen temperature sensors used during testing. This structure allowed local gradients to be modelled in a physically meaningful way while keeping the numerical solution computationally efficient.

A central element of the work was the introduction of a residual-loss method to incorporate the additional heating caused by PWM excitation. Instead of relying on detailed harmonic loss measurements, the approach distributed the measured difference between PWM and sinusoidal losses to locations where inverter-related effects are known to appear most strongly. By allocating these losses primarily to winding, end-winding and rotor-surface nodes, the model reproduced the characteristic heating pattern commonly observed in inverter-fed induction motors. This strategy provided a practical way to extend the thermal network when full electromagnetic loss models were not available.

When the predicted temperatures were compared with experimental measurements across seven operating points, the model reproduced the general structure and magnitude of the temperature distribution with good consistency. The stator slots, end-windings, and yoke regions were estimated with small deviations, and the temperature rise followed the expected relation with torque and speed. The rotor end-ring and rotor-surface temperatures showed larger differences, particularly at high mechanical speed, which reflects uncertainties in

convection modelling and rotor-loss distribution. Even with such deviations, general patterns of temperature and differences due to supplies could be reproduced quite reliably, and the model was stable for all conditions tested.

The limitations in the work primarily refer to the uncertainties in air-gap convection at low and high speeds, the simplified treatment of rotor losses, and the assumption of uniform temperature within a node. These limitations do not weaken the main findings but serve to indicate where future refinement could bring about a gain in accuracy. The scope of the thesis was limited to a steady-state analysis, and transient thermal behaviour or complete harmonic loss simulations were omitted on purpose.

Despite these constraints, the outcomes have practical importance. The thermal network offers a computationally light tool for estimating local temperatures under different supply conditions, and it provides a structured way to incorporate inverter-related effects without relying on detailed electromagnetic models. The approach is suitable for early design studies, comparative thermal evaluation, and situations where rapid estimation of local hot spots is required.

Future work could concentrate on improving the modelling of rotor-region losses under PWM excitation, using rotation-dependent correlations for air-gap convection and adding internal segmentation to better model steep thermal gradients in the rotor. Incorporating transient behaviour and validating against a wider range of operating conditions would also support further development.

Overall, the thesis achieved its objective by demonstrating that a measurement-anchored thermal network can describe and explain the local temperature-rise differences between sinusoidal and PWM supply in an induction motor. The developed modelling strategy was able to reproduce the main characteristics of the measured temperature field and provided a clear and technically reasoned interpretation of how inverter-fed operation alters the internal thermal behaviour of the machine.

References

Andersson, B. 2013, *Lumped parameter thermal modelling of electric machines: analysis of an interior PMSM for vehicle applications*, Master's thesis, Chalmers University of Technology, Gothenburg, Sweden.

Boglietti, A., Cavagnino, A., Staton, D.A., Shanel, M. and Mueller, M. (2014) 'Evolution and modern approaches for thermal analysis of electrical machines', *IEEE Transactions on Industrial Electronics*, 61(3), pp. 1236–1248.

Boglietti, A., Cavagnino, A. and Staton, D.A. (2021) 'Thermal management and design of electrical machines', *IEEE Industrial Electronics Magazine*, 15(2), pp. 26–39.

Bramerdorfer, G. et al. (2017) 'Importance of thermal modeling for design optimization scenarios of induction motors', in *ECCE : 2017 IEEE Energy Conversion Congress and Exposition : 1-5 October 2017*. [Online]. 2017 IEEE. pp. 4666–4672.

Groschup, B. et al. (2021) Characteristic Thermal Parameters in Electric Motors: Comparison Between Induction- and Permanent Magnet Excited Machine. *IEEE transactions on energy conversion*. [Online] 36 (3), 2239–2248.

Jankowski, T. A. et al. (2010) Development and Validation of a Thermal Model for Electric Induction Motors. *IEEE transactions on industrial electronics (1982)*. [Online] 57 (12), 4043–4054.

Khazaeli, M. R. et al. (2025) Thermal Analysis and Improved Thermal Modelling of a 200 kW Traction Induction Motor in Urban Electric Train Application. *IET electric power applications*. [Online] 19 (1),

Kärkkäinen, H., Pyrhönen, J. and Niemelä, M. (2010) 'Loss determination and thermal modelling of induction motors based on measurement data', *IEEE Transactions on Industrial Electronics*, 57(1), pp. 268–276.

Kylander, G. 1995, *Thermal modelling of small cage induction motors*, PhD thesis, Chalmers University of Technology, Gothenburg, Sweden.

Lindh, P., Pyrhönen, J. and Niemelä, M. (2010) 'Thermal analysis of an induction motor using a lumped-parameter thermal model', *IEEE Transactions on Energy Conversion*, 25(1), pp. 256–264.

Lindh, P., Pyrhönen, J. and Niemelä, M. (2021), *Thermal and loss modelling of inverter-fed induction machines*, IEEE Transactions on Energy Conversion, 36(4), pp. 2804–2814.

Lindh, P., Varis, T., Paju, J., Pasanen, A. and Piipponen, T. (2022) 'Thermal Management of an Electric Motor with Novel Materials', in *2022 25th International Conference on Electrical Machines (ICEM)* (Valencia, Spain, 5–8 September 2022). IEEE, pp. 1–6.

Lípa, M. et al. (2025) '*High-Frequency Harmonic Loss Location Verification Using Temperature Rise*', in 2025 IEEE. p.

Mellor, P.H., Roberts, D. and Turner, D.R. (1991) Lumped parameter thermal model for electrical machines of TEFC design. *IEE Proceedings B: Electric Power Applications*, 138(5), pp. 205–218.

Nerg, J. et al. (2008) Thermal Analysis of Radial-Flux Electrical Machines With a High Power Density. *IEEE transactions on industrial electronics (1982)*. [Online] 55 (10), 3543–3554.

Saari, J. 1998, *Thermal analysis of high-speed induction machine*, Doctoral thesis, Helsinki University of Technology / Lappeenranta University of Technology (Acta Polytechnica Scandinavica, Electrical Engineering Series No. 90), Helsinki–Espoo, Finland.

Sachin, S. & Sriram, A.T. 2020, '*Thermal investigations and modelling of AC machines', Journal of Physics: Conference Series*, 1706, 012105, doi: 10.1088/1742-6596/1706/1/012105.

Shahab, M. (2020) *Loss modelling and thermal analysis of induction machines under inverter supply*. Doctoral thesis. Aalto University, Espoo, Finland.

Trigeol, J.-F. et al. (2006) Thermal modeling of an induction machine through the association of two numerical approaches. *IEEE transactions on energy conversion*. [Online] 21 (2), 314–323.

Zhang, D., Li, Y. and Wang, X. (2023) ‘Analysis of harmonic loss distribution and rotor-surface heating in inverter-fed induction motors’, *IEEE Transactions on Energy Conversion*, 38(2), pp. 1105–1115.

Appendix 1. Geometry, material properties and constants

Motor geometry (SEMTEC 5 kW induction motor):

- Stator inner diameter: 125 mm
- Stator outer diameter: 220 mm
- Rotor outer diameter: 124 mm
- Rotor inner diameter: 54 mm
- Air-gap: 0.50 mm
- Stack length: 160 mm
- End-winding axial lengths: 60 mm (connection end), 63 mm (back end)
- Number of stator slots: 48
- Slot opening width: 2.8 mm
- Slot bottom width: 4.0 mm
- Slot height: 24 mm

Material properties:

- Copper thermal conductivity: 385 W/m·K

- Electrical steel:
  - in-plane conductivity: 22 W/m·K
  - axial conductivity: 1 W/m·K

- Resin insulation: 0.25 W/m·K

- Slot liner: 0.20 W/m·K

- Aluminium/steel frame equivalent: 45 W/m·K

Constants:

- Ambient temperature: 27 °C

- Copper temperature coefficient: 0.00393 1/K

- Mass flow constant: 0.035 kg/s (nominal)

- Air specific heat: 1005 J/kg·K

- Base convection coefficients:
  - air-gap: 2.5 W/m²K
  - end-spaces: 4.0 W/m²K

Appendix 2. Thermal resistances used in the network

Stator slot region:

Winding–to–tooth side insulation: $R_{stw,side} = \frac{t_{res}}{k_{res} A_{side}}$

Winding–to–tooth bottom insulation: $R_{stw,bot} = \frac{t_{bottom}}{k_{ins}A_{bottom}}$

Tangential slot conduction (if enabled): $R_{stw,tan} = \frac{L_{tan}}{k_{slot,tan}A_{tan}}$

Stator yoke:

Tooth–to–yoke interface: $R_{ty} = \frac{\delta_{t2y}}{k_{Fe}A_{t2y}}$

Inner–to–mid yoke: $R_{yi\rightarrow ym} = \frac{ln\,(r_{ym}/r_{yi})}{2\pi k_{Fe}L}$

Mid–to–outer yoke: $R_{ym\rightarrow yo} = \frac{ln\,(r_{yo}/r_{ym})}{2\pi k_{Fe}L}$

Stator frame and ambient:

Yoke–frame contact: $R_{yo\rightarrow frame} = \frac{t_{eq}}{\lambda_{eq}A_{frame}}$

Frame–ambient (DE/NDE):

$$R_{frame\rightarrow amb} = \frac{1}{h_{in}A} + \frac{t_{wall}}{k_{wall}A} + \frac{1}{h_{out}A}$$

Rotor region:

Rotor bridge to rotor core: $R_{rb\rightarrow rco}$

Rotor yoke to air-gap convection: $R_{AG,rot} = \frac{1}{h_{rot}A_{rot}}$

End-regions:

End-winding to end-space convection:

$$R_{EW\rightarrow DE} = \frac{1}{h_{end}A_{EW}}$$

$$R_{EW\rightarrow NDE} = \frac{1}{h_{end}A_{EW}}$$

Rotor end-ring to end-space:

$$R_{RER\rightarrow DE} = \frac{1}{h_{ring,DE}A_{ring}}$$

$$R_{RER\rightarrow NDE} = \frac{1}{h_{ring,NDE}A_{ring}}$$

Air nodes:

DE→AG, AG→NDE, NDE→ambient forced-flow resistances derived from

$$R = \frac{1}{\dot{m}c_p}$$

Appendix 3. Matlab implementation of the lptn model

```
% SINE vs PWM comparison with hotspot reporting (ΔT = PWM − SINE at 18 nodes).

% This script is self-contained: defaults(), pwm_retune_struct(), solve_semtec()

% and all helpers are included below.



clc; clear; close all;



% --- Build base parameter struct ----------------------------------------
```

```
P0    = defaults();

Psine = P0;



% --- Solve SINE ----------------------------------------------------------

S_sine   = solve_semtec(Psine);

Tab_sine = S_sine.T_meas.';   % [nOP × 18]



% --- PWM case (retune + converter-side data already in defaults) ---------

Ppwm   = deep_merge(P0, pwm_retune_struct());

S_pwm  = solve_semtec(Ppwm);

Tab_pwm = S_pwm.T_meas.';     % [nOP × 18]



% ------------------------------------------------------------------------

% Experimental 18-node temperatures (°C) from DATA_Syed - Copy.pdf

% Rows follow ops =
{'n90T100','n50T100','n25T100','n90T50','n50T50','n50T25','n25T25'}
```

% Columns follow nodes18 = S_sine.nodes18 (Slot_1..RotorYoke)

% --

ops_exp = {'n90T100','n50T100','n25T100','n90T50','n50T50','n50T25','n25T25'};

% Converter / PWM supply (experimental) %data checked

Texp_pwm = [

76.88 76.62 78.25 78.00 79.08 80.57 68.47 69.13 73.94 81.76 80.76 79.65 72.34 72.62 75.83 68.91 90.13 94.00; % n90T100

69.21 68.96 70.38 70.86 71.78 73.39 61.62 62.09 66.09 74.08 73.21 72.29 65.04 65.15 67.90 62.03 84.07 87.51; % n50T100

62.88 62.86 64.07 64.80 65.98 67.18 55.94 56.35 60.08 67.69 66.83 65.94 59.24 59.14 61.74 56.41 79.29 82.00; % n25T100

59.90 60.04 60.84 60.09 60.56 61.81 54.91 55.36 58.51 62.49 61.83 61.35 57.53 57.76 59.83 55.24 65.02 68.59; % n90T50

52.01 52.10 52.75 52.68 53.07 54.13 47.87 48.12 50.51 54.47 53.98 53.65 49.91 50.07 51.54 48.11 56.90 60.15; % n50T50

48.50 48.65 49.16 48.94 49.22 50.16 45.11 45.33 47.39 50.44 50.03 49.81 46.85 47.04 48.26 45.34 51.41 54.54; % n50T25

41.84 42.14 42.36 42.48 42.77 43.45 39.33 39.45 40.64 43.46 43.32 43.15 40.65 40.73 41.56 39.54 42.48 45.20 % n25T25

];

% Generator (sine) supply (experimental) %data checked

Texp_gen = [

72.67 72.69 73.88 74.02 74.68 76.31 64.98 65.58 69.77 77.39 76.36 75.56 68.40 68.72 71.93 65.36 84.13 87.95; % n90T100

63.00 62.89 64.00 64.59 65.47 66.73 56.37 56.76 60.20 67.27 66.64 65.78 59.19 59.36 61.68 56.66 74.91 77.82; % n50T100

59.42 59.25 60.14 61.36 61.92 63.44 53.12 53.39 56.64 63.66 63.10 62.48 55.44 55.89 57.85 53.35 73.34 76.35; % n25T100

56.12 56.27 57.01 56.36 56.94 57.71 51.79 52.08 55.05 58.32 57.76 57.27 53.82 54.18 55.78 52.04 59.15 62.72; % n90T50

46.91 47.12 47.49 47.50 47.90 48.68 43.65 43.78 45.70 48.87 48.57 48.27 45.16 45.31 46.42 43.81 48.87 51.90; % n50T50

43.28 43.58 43.79 43.67 43.90 44.61 40.78 40.87 42.43 44.74 44.52 44.35 42.05 42.18 43.05 40.95 43.04 45.76; % n50T25

38.76 39.06 39.09 39.34 39.45 40.09 36.81 36.78 37.76 40.02 39.96 39.89 37.72 37.81 38.32 36.91 37.19 40.00 % n25T25

```
];



% Safety check that ordering matches

if ~isequal(S_sine.ops(:), ops_exp(:)) || ~isequal(S_pwm.ops(:), ops_exp(:))

    warning('ops_exp order does not match S_sine.ops / S_pwm.ops – check OP order
before using Texp_*.');

end



% ========================= 7 TREND GRAPHS (MODEL + EXP)
====================

% For each of the 7 operating points, plot temperatures at all 18 nodes

% with 4 curves:

% 1) Model SINE

% 2) Model PWM

% 3) Experimental SINE (generator)

% 4) Experimental PWM (converter)
```

```
nodes18  = S_sine.nodes18;   % 18 node labels

ops_plot = S_sine.ops;       % 7 operating point labels (rows in Tab_*)



nOps   = numel(ops_plot);

nNodes = numel(nodes18);



if ~exist('Texp_gen','var') || ~exist('Texp_pwm','var')

   warning('Texp_gen or Texp_pwm not found in workspace. Experimental trends will not be plotted.');

else

   for r = 1:nOps

      figure('Color','w');



      % 1) model SINE

      plot(1:nNodes, Tab_sine(r,:), '-o', 'LineWidth', 1.5); hold on;
```

```
% 2) model PWM

plot(1:nNodes, Tab_pwm(r,:),  '-s', 'LineWidth', 1.5);



% 3) experimental SINE

plot(1:nNodes, Texp_gen(r,:), '--^', 'LineWidth', 1.2);



% 4) experimental PWM

plot(1:nNodes, Texp_pwm(r,:), '--v', 'LineWidth', 1.2);



grid on;

set(gca, 'XTick', 1:nNodes, ...

    'XTickLabel', nodes18, ...

    'XTickLabelRotation', 45);

xlabel('Thermal nodes');
```

```
        ylabel('Temperature [°C]');
        title(sprintf('Temperatures at 18 nodes — OP %s', ops_plot{r}));
        legend({'Model SINE','Model PWM','Exp. SINE','Exp. PWM'}, ...
            'Location','best');
    end
end

% ====================== 1) GENERATOR (SINE) SUPPLY ======================

topK = 5;   % 5 hottest spots to print

if ~exist('Texp_gen','var')
    warning('Texp_gen (experimental 18-node data for generator supply) is not defined.');
else
    nodes18 = S_sine.nodes18;  % labels for the 18 thermal nodes
```

```
fprintf('\n==========================================================
==\n');

   fprintf(' GENERATOR (SINE) SUPPLY: Experiment vs Model (18 nodes)\n');


fprintf('==========================================================
=\n');



   for r = 1:numel(S_sine.ops)

      op      = S_sine.ops{r};

      Tmodel  = Tab_sine(r,:);      % model temperatures at 18 nodes

      Texp    = Texp_gen(r,:);      % experimental temperatures at 18 nodes

      dT_err  = Tmodel - Texp;      % model − experiment



      % --- Hottest spots based on MODEL temperatures

      [~, idxHot] = sort(Tmodel, 'descend');
```

```
fprintf('\nOP %s — top %d hottest MODEL nodes:\n', op, topK);

for k = 1:min(topK, numel(idxHot))

  j = idxHot(k);

  fprintf('  %2d) %-12s  T_exp = %6.2f °C   T_model = %6.2f °C   (Δ=model−exp = %+6.2f °C)\n', ...

    k, nodes18{j}, Texp(j), Tmodel(j), dT_err(j));

end



% --- Full 18-node table

fprintf('\nFull 18-node table for OP %s (Experiment vs Model):\n', op);

fprintf('   %-12s  %-12s  %-12s  %-12s\n', 'Node', 'T_exp [°C]', 'T_model [°C]', 'Δ=model−exp');

for j = 1:18

  fprintf('   %-12s  %10.2f     %10.2f     %+10.2f\n', ...

    nodes18{j}, Texp(j), Tmodel(j), dT_err(j));

end
```

```
        fprintf('\n');

    end

end



% ========================== 2) PWM SUPPLY
==================================



if ~exist('Texp_pwm','var')

    warning('Texp_pwm (experimental 18-node data for PWM supply) is not defined.');

else

    nodes18 = S_pwm.nodes18;  % labels for the 18 thermal nodes



fprintf('\n=============================================================
==\n');

    fprintf(' PWM SUPPLY: Experiment vs Model (18 nodes)\n');
```

```
fprintf('================================================================
=\n');



   for r = 1:numel(S_pwm.ops)

      op      = S_pwm.ops{r};

      Tmodel  = Tab_pwm(r,:);       % model PWM temperatures at 18 nodes

      Texp    = Texp_pwm(r,:);      % experimental PWM temperatures

      dT_err  = Tmodel - Texp;



      % --- Hottest spots based on MODEL temperatures

      [~, idxHot] = sort(Tmodel, 'descend');



      fprintf('\nOP %s — top %d hottest MODEL nodes:\n', op, topK);

      for k = 1:min(topK, numel(idxHot))

         j = idxHot(k);
```

```
        fprintf('  %2d) %-12s  T_exp = %6.2f °C   T_model = %6.2f °C   (Δ=model−exp =
%+6.2f °C)\n', ...

            k, nodes18{j}, Texp(j), Tmodel(j), dT_err(j));

        end



        % --- Full 18-node table

        fprintf('\nFull 18-node table for OP %s (Experiment vs Model):\n', op);

        fprintf('   %-12s  %-12s  %-12s  %-12s\n', 'Node', 'T_exp [°C]', 'T_model [°C]',
'Δ=model−exp');

        for j = 1:18

            fprintf('   %-12s  %10.2f      %10.2f      %+10.2f\n', ...

                nodes18{j}, Texp(j), Tmodel(j), dT_err(j));

        end

        fprintf('\n');

    end

end
```

```
%% ============================== Local functions
==============================



function P = defaults()

% Basic geometry, materials, and datasets are provided.

P = struct();



% Ambient and copper tempco

P.Tamb       = 27;       % ambient [°C]  (measurement condition)

P.alpha_Cu   = 0.00393;  % copper tempco [1/K]

P.R_phase_20 = 0.47;     % SEMTEC: phase resistance @ 20°C [Ω] (full phase)



% Materials

P.k_Fe_inplane = 22;   % M800-65A in-plane thermal conductivity ~22 W/mK

P.k_Fe_axial   = 1.0;  % through-stack ~1 W/mK
```

```
P.k_Cu        = 385;

P.k_resin     = 0.25;

P.k_ins       = 0.20;

P.k_wall      = 45;   % steel/aluminium equivalent for enclosure



% Geometry (representative SEMTEC-sized IM)

P.L_stack   = 0.16;   % stack length 160 mm

P.D_st_in   = 0.125;  % stator bore 125 mm

P.D_st_out  = 0.220;  % stator outer diameter 220 mm

P.D_rot_out = 0.124;  % rotor outer diameter 124 mm

P.D_rot_in  = 0.054;  % rotor inner diameter 54 mm

P.g_airgap  = 0.0005; % air-gap 0.5 mm (radial)

P.t_frame   = 0.007;  % frame wall ≈7 mm (ID→OD)



P.L_end_conn = 0.06;   % connection end ~60 mm
```

```
P.L_end_back = 0.063;  % back end ~63 mm



P.D_end_ID_min = 0.140;

P.D_end_OD_max = 0.25;



P.Dk_ring = 0.097;  % mean ring diameter 97 mm

P.h_ring  = 0.025;  % ring radial height 25 mm

P.S_slots = 48;     % number of stator slots = 48



% Slot / winding

P.b1 = 0.0028;  % slot opening width b1 = 2.8 mm

P.b4 = 0.004;   % slot bottom width b4 ≈ 4.0 mm

P.h5 = 0.024;   % slot height h5 = 24 mm



P.t_liner  = 0.00020;
```

```
P.t_resin  = 0.00060;

P.t_bottom = 0.00020;

P.phi_contact_slot = 0.85; % modelling: fraction of slot wall wetted by copper



% Mouth opening fraction based on geometry: b1 / slot pitch

Rs_in      = P.D_st_in/2;

slot_pitch = 2*pi*Rs_in / P.S_slots;

P.phi_slot_open = P.b1 / slot_pitch; % ≈ 0.34 for SEMTEC



% conductor diameters from SEMTEC winding data:

% 1 x 0.85 mm + 3 x 0.71 mm in parallel

P.d1 = 0.00085;  % 0.85 mm

P.d2 = 0.00071;  % 0.71 mm

P.N_eff_slot_cond = 32;   % effective conductors / slot

P.fill_factor     = 0.58; % slot fill factor ≈ 0.58
```

```
% Slot anisotropy

P.k_slot_tan      = 3.0;

P.slot_tan_enable = true;



% Teeth → yoke interface

P.geom.delta_t2y  = 0.0006; % effective tooth-to-yoke thickness



% Enclosure/ambient

P.A_enclosure_inner_total = 0.35; % inner frame area (approx.)

P.h_env_in  = 12;

P.h_env_out = 14;

P.t_wall    = 0.003;



% Yoke axial rim thickness
```

```
P.yoke_ax = struct('t_rim', 0.006);



% Contacts and films

P.contact = struct('t_eq', 2.8e-4, 'p_air_int', 0.10);

P.h0_ag   = 2.5;

P.h0_end  = 4.0;

P.h_WdgAG_scale    = 1.0;

P.R_wdg_tooth_scale = 1.0;

P.h_EW_scale        = 1.0;   % end-winding convection scale

P.R_EW_frame_scale  = 1.0;   % EW→frame conduction scale

P.electrical = struct('conn','wye'); % SEMTEC connection: 2Y, thermally still "wye"



P.h_min = 1.0;

P.h_max = 90;

P.m_exp_ag  = 0.25;
```

```
P.m_exp_end = 0.25;

P.scale_factor = 1.0;



% Flow and mechanics

P.flow = struct('mdot_const_kgps', 0.035, 'phi_ag',0.34,'phi_de',0.33,'phi_nde',0.33);

P.mech = struct('C_bearing', 4.0, 'C_windage', 9e-9, 'n_ref', 1500);

P.windage = struct('f_rotor', 0.35);   % default fraction of windage to rotor solids



% Rotor details

P.rotor = struct( ...

    'bridge_scale', 1.4, ...

    'h_scale_core', 0.58, ...

    'h_scale_ring', 0.48);



% Stray loss shaping
```

```
P.stray = struct('k',1.0, 'phi_wdg_base',0.34, 'phi_ew_base',0.11, ...

                 'gamma_wdg',0.50, 'f_ref',50, ...

                 'rot_o',0.54,'rot_i',0.20,'rot_er',0.50);



% Regularization

P.reg = struct('enable', true, 'R_leak', 5e2);



% Bpk proxy (used by Steinmetz helper)

P.Bpk = 1.2;



P.fe = struct('kh',1.5,'ke',0.015,'alpha',1.6,'beta',2.0);

P.tc = struct('F_inner',0.54);



% ------------------ SINE dataset (generator supply) ----------------------

P.ops  = {'n90T100','n50T100','n25T100','n90T50','n50T50','n50T25','n25T25'};
```

```
P.data = struct();

P.data.Irms_line   = [10.37,10.33,10.41,7.47,7.40,6.51,6.42];                  % data checked

P.data.f_Hz        = [44.594,25.297,13.198, 43.994,24.597,24.397, 12.298];     % data checked

P.data.n_rpm       = [1306.45,727.64,362.33, 1305.84,723.83,725.39,362.31];    % data checked

P.data.P2_W        = [4505.13,2509.47,1249.27, 2245.75,1252.09,624.83,313.25]; % data checked

P.data.TotalLoss_W = [476.533,358.891,320.630, 297.581,181.455,142.751,96.026]; % data checked

P.data.dT_wdg      = [44.769,35.288,31.778, 28.531,19.572,16.047,11.610];      % data checked

P.data.slip_rpm    = [31.533,31.396,33.392, 14.123,13.616,6.624,7.171];        % data checked



% ------------------ PWM dataset (converter supply) ----------------------

P.pwm_ops = {'n90T100','n50T100','n25T100','n90T50','n50T50','n50T25','n25T25'};

P.pwm = struct();
```

```
P.pwm.Irms_line   = [10.40,10.37,10.43,7.51,7.40,6.57,6.47];                    % data
checked

P.pwm.f_Hz        = [44.594,25.297,13.198, 43.994,24.597,24.397,12.298];        % data
checked

P.pwm.n_rpm       = [1305.65,726.45,361.47, 1305.53,723.83,725.16,362.31];      % data
checked

P.pwm.P2_W        = [4501.98,2502.08,1246.63, 2244.39,1251.53,625.30,313.00];   %
data checked

P.pwm.TotalLoss_W = [530.082,435.699,367.507,335.766,246.783,209.923,136.460];
% data checked

P.pwm.dT_wdg      = [48.800,41.381,35.320, 32.253,24.561,21.140,14.679];        % data
checked

P.pwm.slip_rpm    = [32.191,32.454,34.474, 14.276,14.078,6.754,6.642];          % data
checked

end



function S = solve_semtec(p)

% Consolidated steady-state LPTN solver for both SINE and PWM.
```

```
p      = deep_merge(defaults(), p);

Tamb   = p.Tamb;

alpha_Cu = p.alpha_Cu;



if ~isfield(p,'fe') || ~isstruct(p.fe), p.fe = struct(); end

if ~isfield(p,'tc') || ~isstruct(p.tc), p.tc = struct('F_inner',0.54); end

if ~isfield(p,'h_EW_scale'),       p.h_EW_scale = 1.0; end

if ~isfield(p,'R_EW_frame_scale'), p.R_EW_frame_scale = 1.0; end

if ~isfield(p,'windage') || ~isstruct(p.windage) || ~isfield(p.windage,'f_rotor')

   p.windage = struct('f_rotor', 0.35);

end



R20  = p.R_phase_20;

conn = 'wye';

if isfield(p,'electrical') && isfield(p.electrical,'conn')
```

```
    conn = lower(p.electrical.conn);

end



ops      = p.ops;

Nc       = numel(ops);

Irms     = p.data.Irms_line;

f_Hz     = p.data.f_Hz;

slip_rpm = p.data.slip_rpm;

n_rpm    = p.data.n_rpm;

P2       = p.data.P2_W;

Total    = p.data.TotalLoss_W;

dT_w     = p.data.dT_wdg;



kFe_in = p.k_Fe_inplane;

kFe_ax = p.k_Fe_axial;
```

```
kRes   = p.k_resin;

kIns   = p.k_ins;



L   = p.L_stack;

Dsi = p.D_st_in;

Dso = p.D_st_out;

Dro = p.D_rot_out;

Dri = p.D_rot_in;

g   = p.g_airgap;

tfr = p.t_frame;



Lec = p.L_end_conn;

Leb = p.L_end_back;

Le  = 0.5*(Lec + Leb);
```

```
Did = p.D_end_ID_min;

Dod = p.D_end_OD_max;

r_end = 0.5*(Did/2 + Dod/2);



Dk  = p.Dk_ring;

hr  = p.h_ring;

Sslots = p.S_slots;



rb  = Dsi/2;

rso = Dso/2;

rro = Dro/2;

rfo = (Dso + 2*tfr)/2;



b1 = p.b1;

b4 = p.b4;
```

```
h5 = p.h5;

t_liner = p.t_liner;

t_res   = p.t_resin;

t_bot   = p.t_bottom;

phi_cs  = p.phi_contact_slot;

phi_open = p.phi_slot_open;



pitch   = pi*Dsi/Sslots;

w_slot  = b1;

w_tooth = pitch - w_slot;

h_slot  = h5;



d1    = p.d1;

d2    = p.d2;

Ncond = p.N_eff_slot_cond;
```

```
fill  = p.fill_factor;



A_bundle   = pi/4*(d1^2 + 3*d2^2);

A_cu_tot   = Ncond*A_bundle;

Spar       = 1;

A_slot_geom = A_cu_tot/fill;



A_frame_side = 2*pi*rfo*L;

A_slot_open  = max(1e-6, phi_open*(Spar*Sslots*w_slot*L));

A_sidewalls  = phi_cs*(2*h_slot*L*Spar*Sslots);

A_bottom     = phi_cs*(b4*L*Spar*Sslots);

A_end_lat    = 2*(2*pi*r_end*Le);

A_ring_lat   = 1.25 * 2*(pi*Dk*hr);

A_rot_AG     = 2*pi*rro*L;
```

```
Aenv_DE  = 0.5*p.A_enclosure_inner_total;

Aenv_NDE = Aenv_DE;

R_fo_DE  = 1/(p.h_env_in*Aenv_DE)  + p.t_wall/(p.k_wall*Aenv_DE)  +
1/(p.h_env_out*Aenv_DE);

R_fo_NDE = 1/(p.h_env_in*Aenv_NDE) + p.t_wall/(p.k_wall*Aenv_NDE) +
1/(p.h_env_out*Aenv_NDE);



R_stw_stt_side = (t_res/kRes + t_liner/kIns)/A_sidewalls;

L_tan = 0.5*w_tooth;

A_tan = h_slot*L*(Spar*Sslots)*phi_cs;

R_stw_stt_tan  = L_tan/(p.k_slot_tan*A_tan);

if ~p.slot_tan_enable

   R_stw_stt_tan = Inf;

end

R_stw_stt_bot  = t_bot/(kIns*A_bottom);

R_stw_stt = p.R_wdg_tooth_scale * (1/(1/R_stw_stt_side + 1/R_stw_stt_tan +
1/R_stw_stt_bot));
```

```
delta_r  = p.geom.delta_t2y;

A_t2y    = 2*pi*((rb + (rb+h_slot))/2)*L;

R_stt_yi = delta_r/(kFe_in*A_t2y);



r_yi = rb + h_slot;

r_yo = rso;

r_ym = sqrt(r_yi*r_yo);

R_yi_ym = log(r_ym/r_yi)/(2*pi*kFe_in*L);

R_ym_yo = log(r_yo/r_ym)/(2*pi*kFe_in*L);



lambda_eq  = air_k_lin(Tamb)*p.contact.p_air_int + kFe_in*(1 - p.contact.p_air_int);

A_yoke_frm = 2*pi*rso*L;

R_yo_frm   = p.contact.t_eq/(lambda_eq*A_yoke_frm);
```

```
L_w2ew = Le/2;

A_w2ew = (Spar*Sslots)*A_cu_tot;

R_w2ew = L_w2ew/(p.k_Cu*A_w2ew);



L_rb_rco = 0.002;

A_rb_rco = 2*pi*rro*L/max(Sslots,1);

R_rb_rco = (L_rb_rco/(kFe_in*A_rb_rco)) / max(p.rotor.bridge_scale,1e-6);



V_yoke = (pi*(rso^2 - (rb+h_slot)^2))*L;

A_sl_ts = A_slot_geom*Sslots*Spar;

V_teeth = (pi*((rb+h_slot)^2 - rb^2))*L - A_sl_ts*L;

V_rotor = (pi*(rro^2 - (Dri/2)^2))*L;

phi_teeth = V_teeth/(V_teeth + V_yoke);



% Node indices
```

```
iW1=1; iW2=2; iW3=3;

iEN1=4; iEN2=5; iEN3=6;

iYI1=7; iYI2=8; iYI3=9;

iED1=10; iED2=11; iED3=12;

iYO1=13; iYO2=14; iYO3=15;

iYM=16; iRER=17; iRYO=18;

iAG=19; iDE=20; iNDE=21; iFRM=22;



slotIdx = [iW1 iW2 iW3];   % slot copper nodes used to read T_cu



n_meas = 18;

n      = 22;

T_all  = zeros(n, Nc);



addAmb = @(Gdiag, RHS, idx, R) deal( ...
```

```
    Gdiag + accum(idx,1./max(R,1e-12),n), ...

    RHS   + accum(idx,(1./max(R,1e-12))*Tamb,n) );



for k = 1:Nc

    if k == 1

        extra = repmat(struct(), Nc, 1);

    end



    nrpm = max(n_rpm(k), 0);

    fop  = max(f_Hz(k), 1e-6);

    nsyn = 30*fop;

    s    = min(max(slip_rpm(k)/max(nsyn,1e-6),0),0.2);



    m_tot   = get_total_mdot(p.flow, nrpm);

    [phi_ag, phi_de, phi_nde] = flow_split_fractions(p.flow);
```

```
m_ref   = max(m_tot/3, 1e-6);

s_ag    = max(phi_ag  * m_tot / m_ref, 1e-6);

s_de    = max(phi_de  * m_tot / m_ref, 1e-6);

s_nde   = max(phi_nde * m_tot / m_ref, 1e-6);



[Pmech_tot, P_brg, P_wind] = mech_losses(p.mech, nrpm);



% --- Iterative copper-loss ↔ winding-temperature coupling ----------

beta_relax = 0.5;   % under-relaxation factor

max_it     = 6;     % max outer iterations

tol_W      = 0.01;  % relative tol on Pcu_s (1%)

tol_T      = 0.5;   % °C tol on winding temp



% initial guess from measured ΔT_wdg

Tcu_prev = Tamb + dT_w(k);
```

```
R_T      = R20 * (1 + alpha_Cu*(Tcu_prev - 20));



switch conn

   case {'wye','y'}

      Iph = Irms(k);

   case {'delta','d'}

      Iph = Irms(k)/sqrt(3);

   otherwise

      Iph = Irms(k);

end



Pcu_s = 3*(Iph^2)*R_T;



for it_fp = 1:max_it
```

```
% losses independent of Pcu_s

P_ag  = (P2(k) + Pmech_tot)/max(1 - s, 1e-6);

Pcu_r = s * P_ag;



[Piron_s, Piron_r] = iron_losses_Steinmetz(p.fe, p.Bpk, fop, ...

                                [V_teeth, V_yoke, V_rotor], Tcu_prev);

Pst_teeth = Piron_s * phi_teeth;

Pst_yoke  = Piron_s * (1 - phi_teeth);



% stray uses current modeled balance (depends on Pcu_s)

P_modeled = Pcu_s + Pcu_r + Piron_s + Piron_r + Pmech_tot;

P_stray   = p.stray.k * max(0, Total(k) - P_modeled);



% --- Stray-loss split: push a bit more into rotor for both SINE & PWM

if isfield(p,'is_pwm') && p.is_pwm
```

```
    % PWM: already more rotor, just a bit extra
    phi_wdg = 0.30;
    phi_ew  = 0.11;
    phi_stF = 0.24;
    phi_rot = 0.35;
else
    % SINE: also use same split for now (can be tuned separately)
    phi_wdg = 0.30;
    phi_ew  = 0.11;
    phi_stF = 0.24;
    phi_rot = 0.35;
end


% build stray heat components
Q_stray_StW = phi_wdg*P_stray;
```

```
Q_stray_EW  = phi_ew *P_stray;

Q_stray_StF = phi_stF*P_stray;

Q_stray_Rot = phi_rot*P_stray;



% initialize air node temps for HTC sub-iteration

T_AG  = Tamb;

T_DE  = Tamb;

T_NDE = Tamb;



% --- HTC / convection sub-iterations ----------------------------

for it = 1:3 %#ok<NASGU>

   rho_DE = rho_moist_f(T_DE,  0.20, 101325);

   mu_DE  = mu_air_f(T_DE);

   k_DE   = air_k_lin(T_DE);
```

```
rho_ND = rho_moist_f(T_NDE, 0.20, 101325);

mu_ND  = mu_air_f(T_NDE);

k_ND   = air_k_lin(T_NDE);



rho_AG = rho_moist_f(T_AG,  0.20, 101325);

mu_AG  = mu_air_f(T_AG);

k_AG   = air_k_lin(T_AG);



cpA = (cp_air_lin(T_DE) + cp_air_lin(T_NDE) + cp_air_lin(T_AG))/3;



omega = 2*pi*(nrpm/60);



% Reference speed around upper mid-range (rpm)

n_ref = 900;

q_exp = 0.25;
```

```
% Avoid division by zero, clip between 1 and 1.3

scale_n = (max(nrpm, 200)/n_ref)^(-q_exp);  % low n → scale_n > 1

scale_n = min(max(scale_n, 1.0), 1.30);     % 1.0 … 1.30



% Taylor–Couette based HTC for rotor surface

h_rot_core = h_taylorCouette_inner(omega, rro, g_airgap(T_AG, g), ...

                                   rho_AG, mu_AG, k_AG, p.tc);



h_rot_core = 0.80 * h_rot_core;



h_rot_AG   = p.h0_ag + p.scale_factor * ...

             min(max(h_rot_core, p.h_min), p.h_max).*s_ag.^p.m_exp_ag;



L_end   = max(2*Le, 1e-6);

Re_DE   = min(rho_DE.*omega.*rro.*L_end./max(mu_DE,eps), 4e5);
```

```
Re_NDE  = min(rho_ND.*omega.*rro.*L_end./max(mu_ND,eps), 4e5);

h_end_DE  = (0.021*(Re_DE .^0.8)).*k_DE ./ L_end;

h_end_NDE = (0.021*(Re_NDE.^0.8)).*k_ND ./ L_end;



h_F_DE  = p.h0_end + p.scale_factor.* ...

        min(max(h_end_DE,  p.h_min), p.h_max).*s_de.^p.m_exp_end;

h_F_NDE = p.h0_end + p.scale_factor.* ...

        min(max(h_end_NDE, p.h_min), p.h_max).*s_nde.^p.m_exp_end;



h_WdgAG_core = (0.21*(rho_AG.*omega.*rro.*g./max(mu_AG,eps)).^0.67).* ...

            k_AG./max(g,eps);

h_WdgAG      = p.h0_ag + p.h_WdgAG_scale.*p.scale_factor.* ...

            min(max(h_WdgAG_core, p.h_min), p.h_max).*s_ag.^p.m_exp_ag;



% Apply low-speed scaling to main air-side HTCs
```

```
h_rot_AG = h_rot_AG * scale_n;

h_F_DE   = h_F_DE   * scale_n;

h_F_NDE  = h_F_NDE  * scale_n;

h_WdgAG  = h_WdgAG  * scale_n;



h_rot_AG_eff   = p.rotor.h_scale_core * h_rot_AG;   % rotor core ↔ airgap

h_ring_DE_eff  = p.rotor.h_scale_ring * h_F_DE;     % end ring ↔ DE endspace

h_ring_NDE_eff = p.rotor.h_scale_ring * h_F_NDE;    % end ring ↔ NDE
endspace



R_rot_AG = 1 ./ max(1e-12, h_rot_AG_eff .* A_rot_AG);

R_stw_AG = 1 ./ max(1e-12, h_WdgAG .* A_slot_open);

R_frm_DE  = 1 ./ max(1e-12, h_F_DE  .* (A_frame_side/2));

R_frm_NDE = 1 ./ max(1e-12, h_F_NDE .* (A_frame_side/2));

R_rer_DE  = 1 ./ max(1e-12, h_ring_DE_eff  .* (A_ring_lat/2));

R_rer_NDE = 1 ./ max(1e-12, h_ring_NDE_eff .* (A_ring_lat/2));
```

```
R_ew_DE   = 1 ./ max(1e-12, (p.h_EW_scale.*h_F_DE ) .* (A_end_lat/2));

R_ew_NDE  = 1 ./ max(1e-12, (p.h_EW_scale.*h_F_NDE) .* (A_end_lat/2));



t_side = 0.0010; A_side = 0.35*(A_end_lat/2);

R_side = t_side/(kRes*A_side);

t_crow = 0.0015; A_crow = 0.15*(A_end_lat/2);

R_crow = t_crow/(kRes*A_crow);

R_EW_FRM = p.R_EW_frame_scale * 1/(1/R_side + 1/R_crow);



A_yoke_ax = 2*pi*rso*p.yoke_ax.t_rim;

R_yoke_ax = Leb/(kFe_ax*A_yoke_ax);



m_chain   = max(m_tot, 1e-6);

R_Amb_NDE = 1./(m_chain.*cp_air_lin(Tamb));

R_NDE_AG  = 1./(m_chain.*cpA);
```

```
R_AG_DE   = 1./(m_chain.*cpA);



% ================== Assemble conductance matrix ==================

G = zeros(n);



% Winding slot ↔ Tooth/Yoke inner (R_stw_stt)

for ii = 1:3

    G = add(G, ii, 6+ii, R_stw_stt);  % W# ↔ YI#

end



% Winding slot ↔ End winding (NDE and DE)

for ii = 1:3

    G = add(G, ii,   3+ii,   R_w2ew); % W# ↔ EN#

    G = add(G, ii,   9+ii,   R_w2ew); % W# ↔ ED#

end
```

```
% Yoke inner ↔ mid ↔ outer (1/3 factors)

Ryim = R_yi_ym/3;

Rmyo = R_ym_yo/3;

for ii = 1:3

   G = add(G, 6+ii, 16,   Ryim); % YI# ↔ YM

   G = add(G, 16,   12+ii, Rmyo); % YM ↔ YO#

end



% Outer yoke ↔ Frame contact

for ii = 13:15

   G = add(G, ii, 22, R_yo_frm); % YO# ↔ FRM

end



% Rotor bridge/core link
```

```
G = add(G, 17, 18, R_rb_rco);      % RER ↔ RYO



% Winding slots ↔ Air-gap convection

for ii = 1:3

   G = add(G, ii, 19, R_stw_AG);  % W# ↔ AG

end



% End windings ↔ Endspaces convection

for ii = 1:3

   G = add(G, 3+ii, 21, R_ew_NDE); % EN# ↔ NDE

   G = add(G, 9+ii, 20, R_ew_DE ); % ED# ↔ DE

end



% Rotor end ring and rotor yoke ↔ Endspaces / Air-gap

G = add(G, 17, 20, R_rer_DE );   % RER ↔ DE
```

```
G = add(G, 17, 21, R_rer_NDE);   % RER ↔ NDE

G = add(G, 18, 19, R_rot_AG );   % RYO ↔ AG



% End-winding baskets ↔ Frame conduction (side / crowfoot)

for ii = [4 5 6 10 11 12]

   G = add(G, ii, 22, R_EW_FRM); % EN#/ED# ↔ FRM

end



% Axial yoke links to endspaces

Rax2 = 2*R_yoke_ax;

for ii = 13:15

   G = add(G, ii, 20, Rax2);     % YO# ↔ DE

   G = add(G, ii, 21, Rax2);     % YO# ↔ NDE

end

G = add(G, 16, 20, 3*R_yoke_ax);  % YM  ↔ DE
```

```
G = add(G, 16, 21, 3*R_yoke_ax);  % YM  ↔ NDE



% --- ambient shunts and inter-air links ----------------------

Gdiag = zeros(n,1);

RHS   = zeros(n,1);



[Gdiag, RHS] = addAmb(Gdiag, RHS, 22, R_fo_DE );   % FRM → AMB (DE
path)

[Gdiag, RHS] = addAmb(Gdiag, RHS, 22, R_fo_NDE);   % FRM → AMB (NDE
path)



G = add(G, 21, 19, R_NDE_AG);  % NDE ↔ AG

G = add(G, 19, 20, R_AG_DE );  % AG  ↔ DE



[Gdiag, RHS] = addAmb(Gdiag, RHS, 21, R_Amb_NDE); % NDE → AMB
```

```
% Regularization
if p.reg.enable
    for ii = 1:n
        [Gdiag, RHS] = addAmb(Gdiag, RHS, ii, p.reg.R_leak);
    end
end


% Set diagonals from off-diagonals + shunts
for ii = 1:n
    offsum   = sum(G(ii,[1:ii-1, ii+1:end]));
    G(ii,ii) = -offsum + Gdiag(ii);
end


% --- heat vector Q with current Pcu_s and splits ------------
Q = zeros(n,1);
```

```
L_slot    = L;

L_end_tot = 2*Le;

phi_EW    = min(max(L_end_tot/(L_end_tot + L_slot), 0.18), 0.48);



Q_slots = (1-phi_EW)*Pcu_s;

Q_EW    =  phi_EW *Pcu_s;



% Stator copper in slots

Q([iW1 iW2 iW3])    = Q([iW1 iW2 iW3])    + (Q_slots/3);



% Stator copper in end windings (split equally NDE/DE)

Q([iEN1 iEN2 iEN3]) = Q([iEN1 iEN2 iEN3]) + (0.5*Q_EW/3);

Q([iED1 iED2 iED3]) = Q([iED1 iED2 iED3]) + (0.5*Q_EW/3);
```

```
% Rotor copper

Q(iRYO) = Q(iRYO) + 0.70*Pcu_r;

Q(iRER) = Q(iRER) + 0.30*Pcu_r;



% Stator iron (teeth and yoke)

Q([iYI1 iYI2 iYI3]) = Q([iYI1 iYI2 iYI3]) + (0.60*Pst_teeth/3);

Q([iYI1 iYI2 iYI3]) = Q([iYI1 iYI2 iYI3]) + (0.20*Pst_yoke/3);

Q([iYO1 iYO2 iYO3]) = Q([iYO1 iYO2 iYO3]) + (0.20*Pst_yoke/3);



% Rotor iron

Q(iRYO) = Q(iRYO) + Piron_r;



% Stray: winding, EW, stator frame, rotor

Q([iW1 iW2 iW3])    = Q([iW1 iW2 iW3])    + (Q_stray_StW/3);

Q([iEN1 iEN2 iEN3]) = Q([iEN1 iEN2 iEN3]) + (0.5*Q_stray_EW/3);
```

```
Q([iED1 iED2 iED3]) = Q([iED1 iED2 iED3]) + (0.5*Q_stray_EW/3);

Q([iYI1 iYI2 iYI3]) = Q([iYI1 iYI2 iYI3]) + (0.70*Q_stray_StF/3);

Q([iYO1 iYO2 iYO3]) = Q([iYO1 iYO2 iYO3]) + (0.30*Q_stray_StF/3);



ro = p.stray.rot_o;

ri = p.stray.rot_i;

re = p.stray.rot_er;

Srot = max(ro+ri+re,eps);

ro = ro/Srot; ri = ri/Srot; re = re/Srot;



Q(iRYO) = Q(iRYO) + (ro+ri)*Q_stray_Rot;

Q(iRER) = Q(iRER) + re*Q_stray_Rot;



% PWM-specific: small leakage of rotor stray directly into airgap

if isfield(p,'is_pwm') && p.is_pwm
```

```
    leak_to_ag = 0.05; % 5% of rotor stray goes to AG

    q_leak = leak_to_ag * Q_stray_Rot;

    Q(iAG)   = Q(iAG)   + q_leak;

    % subtract proportionally from rotor solids

    Q(iRYO)  = Q(iRYO)  - q_leak * 0.6;

    Q(iRER)  = Q(iRER)  - q_leak * 0.4;

end



% Mechanical losses: bearings and windage

f_rot  = min(max(p.windage.f_rotor + 0.15*(nrpm/p.mech.n_ref), 0.20), 0.60);

Q(iRER) = Q(iRER) + P_brg;

Q(iRYO) = Q(iRYO) + 0.60*f_rot*P_wind;

Q(iRER) = Q(iRER) + 0.40*f_rot*P_wind;

Pw_air  = (1 - f_rot)*P_wind;

Q(iAG ) = Q(iAG ) + 0.50*Pw_air;
```

```
Q(iDE ) = Q(iDE ) + 0.25*Pw_air;

Q(iNDE) = Q(iNDE) + 0.25*Pw_air;



Q_inject = Q; % store for reporting



% ---------- store flows for Sankey ----------

qSlots = sum(Q_inject([iW1 iW2 iW3]));

qENDE  = sum(Q_inject([iEN1 iEN2 iEN3]));

qEDE   = sum(Q_inject([iED1 iED2 iED3]));

qYI    = sum(Q_inject([iYI1 iYI2 iYI3]));

qYO    = sum(Q_inject([iYO1 iYO2 iYO3]));

qYM    = Q_inject(iYM);

qRER   = Q_inject(iRER);

qRYO   = Q_inject(iRYO);

qAG    = Q_inject(iAG);
```

```
qDE    = Q_inject(iDE);

qNDE   = Q_inject(iNDE);

qFRM   = Q_inject(iFRM);



extra(k).sources = struct( ...

    'Pcu_s', Pcu_s, ...

    'Pcu_r', Pcu_r, ...

    'Piron_s_teeth', Pst_teeth, ...

    'Piron_s_yoke',  Pst_yoke, ...

    'Piron_r', Piron_r, ...

    'Pmech_brg', P_brg, ...

    'Pmech_wind', P_wind, ...

    'Pstray_wdg', Q_stray_StW, ...

    'Pstray_ew',  Q_stray_EW, ...

    'Pstray_stF', Q_stray_StF, ...
```

```
    'Pstray_rot', Q_stray_Rot, ...
    'Total_reported', Total(k) );


extra(k).sinks = struct( ...
    'Slots', qSlots, 'EW_NDE', qENDE, 'EW_DE', qEDE, ...
    'YokeIn', qYI, 'YokeOut', qYO, 'YokeMid', qYM, ...
    'RotorEndRing', qRER, 'RotorYoke', qRYO, ...
    'Air_AG', qAG, 'Air_DE', qDE, 'Air_NDE', qNDE, ...
    'Frame', qFRM);


% ------------------ END PATCH --------------------------------
Q = Q + RHS;


% --- solve network ------------------------------------------
T = G\Q;
```

```
    if any(~isfinite(T))
        error('Non-finite temperatures encountered.');
    end

    % update HTC reference temps for next sub-iteration
    T_AG  = T(iAG);
    T_DE  = T(iDE);
    T_NDE = T(iNDE);
end  % end HTC sub-iterations

% updated copper temperature from solved slots
Tcu_new = mean(T(slotIdx));
R_T_new = R20 * (1 + alpha_Cu*(Tcu_new - 20));
Pcu_new = 3*(Iph^2)*R_T_new;
```

```
        % under-relax and check convergence
        Pcu_old = Pcu_s;
        Pcu_s   = (1 - beta_relax)*Pcu_s + beta_relax*Pcu_new;

        if abs(Pcu_s - Pcu_old) <= tol_W*max(Pcu_old,1) && ...
           abs(Tcu_new - Tcu_prev) <= tol_T
            break;
        end

        Tcu_prev = Tcu_new;
    end

    % final temperatures for this operating point
    T_all(:,k) = T;
end
```

```
S = struct();

S.T_all   = T_all;

S.T_meas  = T_all(1:n_meas,:);  % first 18 nodes are "measurement" nodes

S.ops     = ops;

S.nodes18 = {'Slot_1','Slot_2','Slot_3','EW_NDE_1','EW_NDE_2','EW_NDE_3', ...

        'YokeIn_1','YokeIn_2','YokeIn_3','EW_DE_1','EW_DE_2','EW_DE_3', ...

        'YokeOut_1','YokeOut_2','YokeOut_3','YokeMid','RotorEndRing','RotorYoke'};

S.extra   = extra;

end



function G = add(G, i, j, R)

% Assemble conductance matrix with negative off-diagonals.

K = 1/max(R, 1e-12);

G(i,j) = G(i,j) - K;
```

```
G(j,i) = G(j,i) - K;

end



function y = accum(idx,val,n)

y      = zeros(n,1);

y(idx) = val;

end



function out = deep_merge(a,b)

out = a;

if ~isstruct(b)

   out = b;

   return;

end

f = fieldnames(b);
```

```
for i = 1:numel(f)
    k = f{i};
    if isstruct(b.(k))
        if isfield(a,k) && isstruct(a.(k))
            out.(k) = deep_merge(a.(k), b.(k));
        else
            out.(k) = b.(k);
        end
    else
        out.(k) = b.(k);
    end
end
end



% ---------------------------- Helper physics ----------------------------
```

```
function rho = rho_moist_f(T, RH, pPa) %#ok<INUSD>

rho = 1.2041 * (273.15/(T+273.15));

end



function mu = mu_air_f(T)

mu = (1.716e-5)*((T+273.15)/273.15)^(3/2)*(273.15+110.4)/(T+273.15+110.4);

end



function k = air_k_lin(T)

k = 0.024 + 7.0e-5*(T-25);

end



function cp = cp_air_lin(T) %#ok<INUSD>

cp = 1005;

end
```

```
function gg = g_airgap(T, g0) %#ok<INUSD>

gg = g0;

end



function h = h_taylorCouette_inner(omega, r_in, gap, rho, mu, k, tc) %#ok<INUSD>

Re = max(rho*omega*r_in*(2*gap)/max(mu,1e-9), 1);

Nu = 4 + 0.07*Re^0.7;

h  = Nu*k/max(gap,1e-6);

end



function m = get_total_mdot(flow, n_rpm)

if isfield(flow,'mdot_const_kgps') && flow.mdot_const_kgps>0

    m = flow.mdot_const_kgps*(1 + 0.002*max(n_rpm,0)/1500);

else
```

```
    m = 0.02*(1 + 0.002*max(n_rpm,0)/1500);

end

end



function [phi_ag, phi_de, phi_nde] = flow_split_fractions(flow)

if isfield(flow,'phi_ag'),  phi_ag  = flow.phi_ag;  else, phi_ag  = 0.34; end

if isfield(flow,'phi_de'),  phi_de  = flow.phi_de;  else, phi_de  = 0.33; end

if isfield(flow,'phi_nde'), phi_nde = flow.phi_nde; else, phi_nde = 0.33; end

S = phi_ag + phi_de + phi_nde;

phi_ag  = phi_ag /S;

phi_de  = phi_de /S;

phi_nde = phi_nde/S;

end



function [Pmech_tot, P_brg, P_wind] = mech_losses(mech, n_rpm)
```

```
P_brg  = mech.C_bearing*(1 + (n_rpm/1500));

P_wind = mech.C_windage*(n_rpm.^3);

P_wind = min(P_wind, 80);   % optional cap (W)

Pmech_tot = P_brg + P_wind;

end



function [Piron_s, Piron_r] = iron_losses_Steinmetz(fe, Bpk, f, Vvec, Twdg)
%#ok<INUSD>

% Simple Steinmetz-based iron loss model with stator/rotor split.

if ~isfield(fe,'kh'),    fe.kh    = 1.5; end

if ~isfield(fe,'ke'),    fe.ke    = 0.015; end

if ~isfield(fe,'alpha'), fe.alpha = 1.6; end

if ~isfield(fe,'beta'),  fe.beta  = 2.0; end



kh = fe.kh;

ke = fe.ke;
```

```
a  = fe.alpha;

b  = fe.beta;



Vt = Vvec(1)+Vvec(2);

Vr = Vvec(3);



Pst = (kh*(f.^a)*(Bpk.^b))*Vt;

Pr  = 0.35*Pst*(Vr/max(Vt,1e-9));



Piron_s = Pst;

Piron_r = Pr;

end



% ------------------- Dataset utilities ----------------------------------

function P = add_n25T25_clone(P)
```

```
% (Not used in main flow) Example of cloning missing sine OP 'n25T25'

ops = P.ops;

if ~any(strcmp(ops,'n25T25'))

    ops = [ops, {'n25T25'}];

    idx = numel(P.data.Irms_line); % assume last is n50T25

    I    = P.data.Irms_line(idx)*0.97;

    f    = 12.298;

    n    = 360.0;

    P2   = P.data.P2_W(idx)*0.505;

    TL   = 96.026;

    dT   = P.data.dT_wdg(idx)*0.72;

    slip = 7.171;



    P.data.Irms_line   = [P.data.Irms_line, I];

    P.data.f_Hz        = [P.data.f_Hz, f];
```

```
    P.data.n_rpm       = [P.data.n_rpm, n];
    P.data.P2_W        = [P.data.P2_W, P2];
    P.data.TotalLoss_W = [P.data.TotalLoss_W, TL];
    P.data.dT_wdg      = [P.data.dT_wdg, dT];
    P.data.slip_rpm    = [P.data.slip_rpm, slip];
end
P.ops = {'n90T100','n50T100','n25T100','n90T50','n50T50','n50T25','n25T25'};
end


function S = pwm_retune_struct()
% Returns PWM-specific overrides for defaults().
P = defaults();
S = struct();


S.h0_ag = 1.8;
```

```
S.tc    = struct('F_inner',0.54);

S.h0_end = 3.0;

S.h_EW_scale        = 1.0;

S.R_EW_frame_scale  = 1.0;

S.h_WdgAG_scale     = 0.8;

S.R_wdg_tooth_scale = 2.6;

S.contact = struct('t_eq',2.8e-4,'p_air_int',0.10);

S.windage = struct('f_rotor',0.35);

S.stray = struct('phi_wdg_base',0.34,'phi_ew_base',0.11,'gamma_wdg',0.50, ...

                 'f_ref',50,'rot_o',0.38,'rot_i',0.15,'rot_er',0.25,'k',1.0);

S.electrical = struct('conn','wye');

S.is_pwm = true;



% explicit iron coeffs for PWM (same as sine, but could be tuned)

S.fe = struct('kh',1.5,'ke',0.015,'alpha',1.6,'beta',2.0);
```

```
S.ops  = P.pwm_ops;

S.data = struct();

S.data.Irms_line   = P.pwm.Irms_line;

S.data.f_Hz        = P.pwm.f_Hz;

S.data.n_rpm       = P.pwm.n_rpm;

S.data.P2_W        = P.pwm.P2_W;

S.data.TotalLoss_W = P.pwm.TotalLoss_W;

S.data.dT_wdg      = P.pwm.dT_wdg;

S.data.slip_rpm    = P.pwm.slip_rpm;

end



function plot_sankey_for_op(S, opName)

% Draws a Sankey diagram for one operating point in result struct S.
```

```
idx = find(strcmp(S.ops, opName), 1);

if isempty(idx)

    error('OP "%s" not found.', opName);

end

E = S.extra(idx);



nodes = ["TotalLoss", ...

"Cu_stator","Cu_rotor","Fe_stator_teeth","Fe_stator_yoke","Fe_rotor","Mech_brg","Mech
_wind", ...

        "Stray_wdg","Stray_ew","Stray_stF","Stray_rot", ...

        "Slots","EW_NDE","EW_DE","YokeIn","YokeOut","YokeMid", ...

        "RotorYoke","RotorEndRing","Air_AG","Air_DE","Air_NDE","Frame"];



TL = E.sources.Total_reported;

buckets = ["Cu_stator","Cu_rotor","Fe_stator_teeth","Fe_stator_yoke","Fe_rotor", ...
```

```
        "Mech_brg","Mech_wind","Stray_wdg","Stray_ew","Stray_stF","Stray_rot"];

bvals   = [E.sources.Pcu_s, E.sources.Pcu_r, ...

        E.sources.Piron_s_teeth, E.sources.Piron_s_yoke, ...

        E.sources.Piron_r, E.sources.Pmech_brg, E.sources.Pmech_wind, ...

        E.sources.Pstray_wdg, E.sources.Pstray_ew, ...

        E.sources.Pstray_stF, E.sources.Pstray_rot];



src = []; dst = []; wgt = [];



% Total → buckets

src = [src; repmat("TotalLoss", numel(buckets),1)];

dst = [dst; buckets(:)];

wgt = [wgt; bvals(:)];



% Cu_stator → Slots & EW (use sink ratios)
```

```
phiSlots = E.sinks.Slots  / max(E.sources.Pcu_s, eps);

phiENDE  = E.sinks.EW_NDE / max(E.sources.Pcu_s, eps);

phiEDE   = E.sinks.EW_DE  / max(E.sources.Pcu_s, eps);



src = [src; "Cu_stator"; "Cu_stator"; "Cu_stator"];

dst = [dst; "Slots"; "EW_NDE"; "EW_DE"];

wgt = [wgt; E.sources.Pcu_s*phiSlots; ...

        E.sources.Pcu_s*phiENDE; ...

        E.sources.Pcu_s*phiEDE];



% Cu_rotor → RotorYoke (70%) + RotorEndRing (30%)

src = [src; "Cu_rotor"; "Cu_rotor"];

dst = [dst; "RotorYoke"; "RotorEndRing"];

wgt = [wgt; 0.70*E.sources.Pcu_r; 0.30*E.sources.Pcu_r];
```

```
% Fe_stator_teeth → YokeIn (60%)

src = [src; "Fe_stator_teeth"];

dst = [dst; "YokeIn"];

wgt = [wgt; 0.60*E.sources.Piron_s_teeth];



% Fe_stator_yoke → YokeIn (20%) + YokeOut (20%)

src = [src; "Fe_stator_yoke"; "Fe_stator_yoke"];

dst = [dst; "YokeIn"; "YokeOut"];

wgt = [wgt; 0.20*E.sources.Piron_s_yoke; 0.20*E.sources.Piron_s_yoke];



% Fe_rotor → RotorYoke

src = [src; "Fe_rotor"];

dst = [dst; "RotorYoke"];

wgt = [wgt; E.sources.Piron_r];
```

```
% Mech_brg → RotorEndRing

src = [src; "Mech_brg"];

dst = [dst; "RotorEndRing"];

wgt = [wgt; E.sources.Pmech_brg];



% Mech_wind → solids + air (approx via sinks)

Pw       = E.sources.Pmech_wind;

airShare = (E.sinks.Air_AG + E.sinks.Air_DE + E.sinks.Air_NDE)/max(Pw,eps);

rotShare = 1 - airShare;



src = [src; "Mech_wind"; "Mech_wind"; "Mech_wind"; "Mech_wind"; "Mech_wind"];

dst = [dst; "RotorYoke"; "RotorEndRing"; "Air_AG"; "Air_DE"; "Air_NDE"];

wgt = [wgt; 0.60*rotShare*Pw; 0.40*rotShare*Pw; ...

        0.50*airShare*Pw; 0.25*airShare*Pw; 0.25*airShare*Pw];
```

```
% Stray_wdg → Slots

src = [src; "Stray_wdg"];

dst = [dst; "Slots"];

wgt = [wgt; E.sources.Pstray_wdg];



% Stray_ew → EW_NDE + EW_DE (even split)

src = [src; "Stray_ew"; "Stray_ew"];

dst = [dst; "EW_NDE"; "EW_DE"];

wgt = [wgt; 0.5*E.sources.Pstray_ew; 0.5*E.sources.Pstray_ew];



% Stray_stF → YokeIn (70%) + YokeOut (30%)

src = [src; "Stray_stF"; "Stray_stF"];

dst = [dst; "YokeIn"; "YokeOut"];

wgt = [wgt; 0.70*E.sources.Pstray_stF; 0.30*E.sources.Pstray_stF];
```

```
% Stray_rot → RotorYoke + RotorEndRing (use sink ratio)

rot_total = E.sources.Pstray_rot;

if rot_total > 0

    fracRYO = E.sinks.RotorYoke / (E.sinks.RotorYoke + E.sinks.RotorEndRing);

else

    fracRYO = 1;

end

src = [src; "Stray_rot"; "Stray_rot"];

dst = [dst; "RotorYoke"; "RotorEndRing"];

wgt = [wgt; fracRYO*rot_total; (1-fracRYO)*rot_total];



% Build digraph and plot

G = digraph;

for i = 1:numel(src)

    if wgt(i) > 1e-6
```

```
        G = addedge(G, src(i), dst(i), wgt(i));

    end

end



figure('Color','w','Name',sprintf('Sankey — %s', opName));

sankey(G,'NodeLabel',string(nodes));

title(sprintf('Loss & Heat Injection Flow — %s', opName));

end
```

OUTPUT:

```
============================================================

 GENERATOR (SINE) SUPPLY: Experiment vs Model (18 nodes)

============================================================



OP n90T100 — top 5 hottest MODEL nodes:
```

1) RotorEndRing T_exp = 84.13 °C T_model = 97.74 °C (Δ=model−exp = +13.61 °C)

2) RotorYoke T_exp = 87.95 °C T_model = 95.22 °C (Δ=model−exp = +7.27 °C)

3) Slot_1 T_exp = 72.67 °C T_model = 64.82 °C (Δ=model−exp = -7.85 °C)

4) Slot_2 T_exp = 72.69 °C T_model = 64.82 °C (Δ=model−exp = -7.87 °C)

5) Slot_3 T_exp = 73.88 °C T_model = 64.82 °C (Δ=model−exp = -9.06 °C)

Full 18-node table for OP n90T100 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	72.67	64.82	-7.85
Slot_2	72.69	64.82	-7.87
Slot_3	73.88	64.82	-9.06
EW_NDE_1	74.02	64.39	-9.63
EW_NDE_2	74.68	64.39	-10.29
EW_NDE_3	76.31	64.39	-11.92
YokeIn_1	64.98	64.76	-0.22

YokeIn_2	65.58	64.76	-0.82
YokeIn_3	69.77	64.76	-5.01
EW_DE_1	77.39	64.70	-12.69
EW_DE_2	76.36	64.70	-11.66
EW_DE_3	75.56	64.70	-10.86
YokeOut_1	68.40	64.66	-3.74
YokeOut_2	68.72	64.66	-4.06
YokeOut_3	71.93	64.66	-7.27
YokeMid	65.36	64.71	-0.65
RotorEndRing	84.13	97.74	+13.61
RotorYoke	87.95	95.22	+7.27

OP n50T100 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 74.91 °C T_model = 80.04 °C (Δ=model−exp = +5.13 °C)

2) RotorYoke T_exp = 77.82 °C T_model = 78.05 °C (Δ=model−exp = +0.23 °C)

3) Slot_1 T_exp = 63.00 °C T_model = 58.95 °C (Δ=model−exp = -4.05 °C)

4) Slot_2 T_exp = 62.89 °C T_model = 58.95 °C (Δ=model−exp = -3.94 °C)

5) Slot_3 T_exp = 64.00 °C T_model = 58.95 °C (Δ=model−exp = -5.05 °C)

Full 18-node table for OP n50T100 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	63.00	58.95	-4.05
Slot_2	62.89	58.95	-3.94
Slot_3	64.00	58.95	-5.05
EW_NDE_1	64.59	58.77	-5.82
EW_NDE_2	65.47	58.77	-6.70
EW_NDE_3	66.73	58.77	-7.96
YokeIn_1	56.37	58.88	+2.51
YokeIn_2	56.76	58.88	+2.12

YokeIn_3	60.20	58.88	-1.32
EW_DE_1	67.27	58.94	-8.33
EW_DE_2	66.64	58.94	-7.70
EW_DE_3	65.78	58.94	-6.84
YokeOut_1	59.19	58.80	-0.39
YokeOut_2	59.36	58.80	-0.56
YokeOut_3	61.68	58.80	-2.88
YokeMid	56.66	58.84	+2.18
RotorEndRing	74.91	80.04	+5.13
RotorYoke	77.82	78.05	+0.23

OP n25T100 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 73.34 °C T_model = 71.09 °C (Δ=model−exp = -2.25 °C)

2) RotorYoke T_exp = 76.35 °C T_model = 69.20 °C (Δ=model−exp = -7.15 °C)

3) EW_DE_1 T_exp = 63.66 °C T_model = 56.76 °C (Δ=model−exp = -6.90 °C)

4) EW_DE_2 T_exp = 63.10 °C T_model = 56.76 °C (Δ=model−exp = -6.34 °C)

5) EW_DE_3 T_exp = 62.48 °C T_model = 56.76 °C (Δ=model−exp = -5.72 °C)

Full 18-node table for OP n25T100 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	59.42	56.71	-2.71
Slot_2	59.25	56.71	-2.54
Slot_3	60.14	56.71	-3.43
EW_NDE_1	61.36	56.65	-4.71
EW_NDE_2	61.92	56.65	-5.27
EW_NDE_3	63.44	56.65	-6.79
YokeIn_1	53.12	56.64	+3.52
YokeIn_2	53.39	56.64	+3.25
YokeIn_3	56.64	56.64	-0.00

EW_DE_1	63.66	56.76	-6.90
EW_DE_2	63.10	56.76	-6.34
EW_DE_3	62.48	56.76	-5.72
YokeOut_1	55.44	56.56	+1.12
YokeOut_2	55.89	56.56	+0.67
YokeOut_3	57.85	56.56	-1.29
YokeMid	53.35	56.60	+3.25
RotorEndRing	73.34	71.09	-2.25
RotorYoke	76.35	69.20	-7.15

OP n90T50 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 59.15 °C T_model = 66.07 °C (Δ=model−exp = +6.92 °C)

2) RotorYoke T_exp = 62.72 °C T_model = 64.48 °C (Δ=model−exp = +1.76 °C)

3) Slot_1 T_exp = 56.12 °C T_model = 51.58 °C (Δ=model−exp = -4.54 °C)

4) Slot_2 T_exp = 56.27 °C T_model = 51.58 °C (Δ=model−exp = -4.69 °C)

5) Slot_3 T_exp = 57.01 °C T_model = 51.58 °C (Δ=model−exp = -5.43 °C)

Full 18-node table for OP n90T50 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	56.12	51.58	-4.54
Slot_2	56.27	51.58	-4.69
Slot_3	57.01	51.58	-5.43
EW_NDE_1	56.36	51.26	-5.10
EW_NDE_2	56.94	51.26	-5.68
EW_NDE_3	57.71	51.26	-6.45
YokeIn_1	51.79	51.55	-0.24
YokeIn_2	52.08	51.55	-0.53
YokeIn_3	55.05	51.55	-3.50
EW_DE_1	58.32	51.45	-6.87

EW_DE_2	57.76	51.45	-6.31
EW_DE_3	57.27	51.45	-5.82
YokeOut_1	53.82	51.49	-2.33
YokeOut_2	54.18	51.49	-2.69
YokeOut_3	55.78	51.49	-4.29
YokeMid	52.04	51.52	-0.52
RotorEndRing	59.15	66.07	+6.92
RotorYoke	62.72	64.48	+1.76

OP n50T50 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 48.87 °C T_model = 48.82 °C (Δ=model−exp = -0.05 °C)

2) RotorYoke T_exp = 51.90 °C T_model = 47.87 °C (Δ=model−exp = -4.03 °C)

3) Slot_1 T_exp = 46.91 °C T_model = 44.62 °C (Δ=model−exp = -2.29 °C)

4) Slot_2 T_exp = 47.12 °C T_model = 44.62 °C (Δ=model−exp = -2.50 °C)

5) Slot_3 T_exp = 47.49 °C T_model = 44.62 °C (Δ=model−exp = -2.87 °C)

Full 18-node table for OP n50T50 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	46.91	44.62	-2.29
Slot_2	47.12	44.62	-2.50
Slot_3	47.49	44.62	-2.87
EW_NDE_1	47.50	44.50	-3.00
EW_NDE_2	47.90	44.50	-3.40
EW_NDE_3	48.68	44.50	-4.18
YokeIn_1	43.65	44.59	+0.94
YokeIn_2	43.78	44.59	+0.81
YokeIn_3	45.70	44.59	-1.11
EW_DE_1	48.87	44.58	-4.29
EW_DE_2	48.57	44.58	-3.99

EW_DE_3	48.27	44.58	-3.69
YokeOut_1	45.16	44.54	-0.62
YokeOut_2	45.31	44.54	-0.77
YokeOut_3	46.42	44.54	-1.88
YokeMid	43.81	44.57	+0.76
RotorEndRing	48.87	48.82	-0.05
RotorYoke	51.90	47.87	-4.03

OP n50T25 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 43.04 °C T_model = 42.20 °C (Δ=model−exp = -0.84 °C)

2) RotorYoke T_exp = 45.76 °C T_model = 41.46 °C (Δ=model−exp = -4.30 °C)

3) Slot_1 T_exp = 43.28 °C T_model = 41.44 °C (Δ=model−exp = -1.84 °C)

4) Slot_2 T_exp = 43.58 °C T_model = 41.44 °C (Δ=model−exp = -2.14 °C)

5) Slot_3 T_exp = 43.79 °C T_model = 41.44 °C (Δ=model−exp = -2.35 °C)

Full 18-node table for OP n50T25 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	43.28	41.44	-1.84
Slot_2	43.58	41.44	-2.14
Slot_3	43.79	41.44	-2.35
EW_NDE_1	43.67	41.34	-2.33
EW_NDE_2	43.90	41.34	-2.56
EW_NDE_3	44.61	41.34	-3.27
YokeIn_1	40.78	41.42	+0.64
YokeIn_2	40.87	41.42	+0.55
YokeIn_3	42.43	41.42	-1.01
EW_DE_1	44.74	41.40	-3.34
EW_DE_2	44.52	41.40	-3.12
EW_DE_3	44.35	41.40	-2.95

YokeOut_1	42.05	41.38	-0.67
YokeOut_2	42.18	41.38	-0.80
YokeOut_3	43.05	41.38	-1.67
YokeMid	40.95	41.40	+0.45
RotorEndRing	43.04	42.20	-0.84
RotorYoke	45.76	41.46	-4.30

OP n25T25 — top 5 hottest MODEL nodes:

1) EW_DE_1 T_exp = 40.02 °C T_model = 37.81 °C (Δ=model−exp = -2.21 °C)

2) EW_DE_2 T_exp = 39.96 °C T_model = 37.81 °C (Δ=model−exp = -2.15 °C)

3) EW_DE_3 T_exp = 39.89 °C T_model = 37.81 °C (Δ=model−exp = -2.08 °C)

4) Slot_1 T_exp = 38.76 °C T_model = 37.81 °C (Δ=model−exp = -0.95 °C)

5) Slot_2 T_exp = 39.06 °C T_model = 37.81 °C (Δ=model−exp = -1.25 °C)

Full 18-node table for OP n25T25 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	38.76	37.81	-0.95
Slot_2	39.06	37.81	-1.25
Slot_3	39.09	37.81	-1.28
EW_NDE_1	39.34	37.78	-1.56
EW_NDE_2	39.45	37.78	-1.67
EW_NDE_3	40.09	37.78	-2.31
YokeIn_1	36.81	37.79	+0.98
YokeIn_2	36.78	37.79	+1.01
YokeIn_3	37.76	37.79	+0.03
EW_DE_1	40.02	37.81	-2.21
EW_DE_2	39.96	37.81	-2.15
EW_DE_3	39.89	37.81	-2.08
YokeOut_1	37.72	37.76	+0.04

YokeOut_2	37.81	37.76	-0.05
YokeOut_3	38.32	37.76	-0.56
YokeMid	36.91	37.77	+0.86
RotorEndRing	37.19	35.26	-1.93
RotorYoke	40.00	34.81	-5.19

==========

PWM SUPPLY: Experiment vs Model (18 nodes)

==========

OP n90T100 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 90.13 °C T_model = 104.31 °C (Δ=model−exp = +14.18 °C)

2) RotorYoke T_exp = 94.00 °C T_model = 101.84 °C (Δ=model−exp = +7.84 °C)

3) Slot_1 T_exp = 76.88 °C T_model = 70.61 °C (Δ=model−exp = -6.27 °C)

4) Slot_2 T_exp = 76.62 °C T_model = 70.61 °C (Δ=model−exp = -6.01 °C)

5) Slot_3 T_exp = 78.25 °C T_model = 70.61 °C (Δ=model−exp = -7.64 °C)

Full 18-node table for OP n90T100 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	76.88	70.61	-6.27
Slot_2	76.62	70.61	-6.01
Slot_3	78.25	70.61	-7.64
EW_NDE_1	78.00	70.10	-7.90
EW_NDE_2	79.08	70.10	-8.98
EW_NDE_3	80.57	70.10	-10.47
YokeIn_1	68.47	70.45	+1.98
YokeIn_2	69.13	70.45	+1.32
YokeIn_3	73.94	70.45	-3.49
EW_DE_1	81.76	70.43	-11.33

EW_DE_2	80.76	70.43	-10.33
EW_DE_3	79.65	70.43	-9.22
YokeOut_1	72.34	70.34	-2.00
YokeOut_2	72.62	70.34	-2.28
YokeOut_3	75.83	70.34	-5.49
YokeMid	68.91	70.39	+1.48
RotorEndRing	90.13	104.31	+14.18
RotorYoke	94.00	101.84	+7.84

OP n50T100 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 84.07 °C T_model = 90.06 °C (Δ=model−exp = +5.99 °C)

2) RotorYoke T_exp = 87.51 °C T_model = 87.83 °C (Δ=model−exp = +0.32 °C)

3) Slot_1 T_exp = 69.21 °C T_model = 67.79 °C (Δ=model−exp = -1.42 °C)

4) Slot_2 T_exp = 68.96 °C T_model = 67.79 °C (Δ=model−exp = -1.17 °C)

5) Slot_3 T_exp = 70.38 °C T_model = 67.79 °C (Δ=model−exp = -2.59 °C)

Full 18-node table for OP n50T100 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	69.21	67.79	-1.42
Slot_2	68.96	67.79	-1.17
Slot_3	70.38	67.79	-2.59
EW_NDE_1	70.86	67.51	-3.35
EW_NDE_2	71.78	67.51	-4.27
EW_NDE_3	73.39	67.51	-5.88
YokeIn_1	61.62	67.61	+5.99
YokeIn_2	62.09	67.61	+5.52
YokeIn_3	66.09	67.61	+1.52
EW_DE_1	74.08	67.71	-6.37
EW_DE_2	73.21	67.71	-5.50

EW_DE_3	72.29	67.71	-4.58
YokeOut_1	65.04	67.50	+2.46
YokeOut_2	65.15	67.50	+2.35
YokeOut_3	67.90	67.50	-0.40
YokeMid	62.03	67.55	+5.52
RotorEndRing	84.07	90.06	+5.99
RotorYoke	87.51	87.83	+0.32

OP n25T100 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 79.29 °C T_model = 77.00 °C (Δ=model−exp = -2.29 °C)

2) RotorYoke T_exp = 82.00 °C T_model = 74.91 °C (Δ=model−exp = -7.09 °C)

3) EW_DE_1 T_exp = 67.69 °C T_model = 63.13 °C (Δ=model−exp = -4.56 °C)

4) EW_DE_2 T_exp = 66.83 °C T_model = 63.13 °C (Δ=model−exp = -3.70 °C)

5) EW_DE_3 T_exp = 65.94 °C T_model = 63.13 °C (Δ=model−exp = -2.81 °C)

Full 18-node table for OP n25T100 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	62.88	63.12	+0.24
Slot_2	62.86	63.12	+0.26
Slot_3	64.07	63.12	-0.95
EW_NDE_1	64.80	63.01	-1.79
EW_NDE_2	65.98	63.01	-2.97
EW_NDE_3	67.18	63.01	-4.17
YokeIn_1	55.94	62.92	+6.98
YokeIn_2	56.35	62.92	+6.57
YokeIn_3	60.08	62.92	+2.84
EW_DE_1	67.69	63.13	-4.56
EW_DE_2	66.83	63.13	-3.70
EW_DE_3	65.94	63.13	-2.81

YokeOut_1	59.24	62.83	+3.59
YokeOut_2	59.14	62.83	+3.69
YokeOut_3	61.74	62.83	+1.09
YokeMid	56.41	62.88	+6.47
RotorEndRing	79.29	77.00	-2.29
RotorYoke	82.00	74.91	-7.09

OP n90T50 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 65.02 °C T_model = 70.19 °C (Δ=model−exp = +5.17 °C)

2) RotorYoke T_exp = 68.59 °C T_model = 68.71 °C (Δ=model−exp = +0.12 °C)

3) Slot_1 T_exp = 59.90 °C T_model = 55.70 °C (Δ=model−exp = -4.20 °C)

4) Slot_2 T_exp = 60.04 °C T_model = 55.70 °C (Δ=model−exp = -4.34 °C)

5) Slot_3 T_exp = 60.84 °C T_model = 55.70 °C (Δ=model−exp = -5.14 °C)

Full 18-node table for OP n90T50 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	59.90	55.70	-4.20
Slot_2	60.04	55.70	-4.34
Slot_3	60.84	55.70	-5.14
EW_NDE_1	60.09	55.32	-4.77
EW_NDE_2	60.56	55.32	-5.24
EW_NDE_3	61.81	55.32	-6.49
YokeIn_1	54.91	55.62	+0.71
YokeIn_2	55.36	55.62	+0.26
YokeIn_3	58.51	55.62	-2.89
EW_DE_1	62.49	55.53	-6.96
EW_DE_2	61.83	55.53	-6.30
EW_DE_3	61.35	55.53	-5.82
YokeOut_1	57.53	55.55	-1.98

YokeOut_2	57.76	55.55	-2.21
YokeOut_3	59.83	55.55	-4.28
YokeMid	55.24	55.59	+0.35
RotorEndRing	65.02	70.19	+5.17
RotorYoke	68.59	68.71	+0.12

OP n50T50 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 56.90 °C T_model = 57.04 °C (Δ=model−exp = +0.14 °C)

2) RotorYoke T_exp = 60.15 °C T_model = 55.91 °C (Δ=model−exp = -4.24 °C)

3) Slot_1 T_exp = 52.01 °C T_model = 51.80 °C (Δ=model−exp = -0.21 °C)

4) Slot_2 T_exp = 52.10 °C T_model = 51.80 °C (Δ=model−exp = -0.30 °C)

5) Slot_3 T_exp = 52.75 °C T_model = 51.80 °C (Δ=model−exp = -0.95 °C)

Full 18-node table for OP n50T50 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	52.01	51.80	-0.21
Slot_2	52.10	51.80	-0.30
Slot_3	52.75	51.80	-0.95
EW_NDE_1	52.68	51.60	-1.08
EW_NDE_2	53.07	51.60	-1.47
EW_NDE_3	54.13	51.60	-2.53
YokeIn_1	47.87	51.71	+3.84
YokeIn_2	48.12	51.71	+3.59
YokeIn_3	50.51	51.71	+1.20
EW_DE_1	54.47	51.70	-2.77
EW_DE_2	53.98	51.70	-2.28
EW_DE_3	53.65	51.70	-1.95
YokeOut_1	49.91	51.65	+1.74
YokeOut_2	50.07	51.65	+1.58

YokeOut_3	51.54	51.65	+0.11
YokeMid	48.11	51.68	+3.57
RotorEndRing	56.90	57.04	+0.14
RotorYoke	60.15	55.91	-4.24

OP n50T25 — top 5 hottest MODEL nodes:

1) RotorEndRing T_exp = 51.41 °C T_model = 50.41 °C (Δ=model−exp = -1.00 °C)

2) RotorYoke T_exp = 54.54 °C T_model = 49.50 °C (Δ=model−exp = -5.04 °C)

3) Slot_1 T_exp = 48.50 °C T_model = 48.74 °C (Δ=model−exp = +0.24 °C)

4) Slot_2 T_exp = 48.65 °C T_model = 48.74 °C (Δ=model−exp = +0.09 °C)

5) Slot_3 T_exp = 49.16 °C T_model = 48.74 °C (Δ=model−exp = -0.42 °C)

Full 18-node table for OP n50T25 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp

Slot_1	48.50	48.74	+0.24
Slot_2	48.65	48.74	+0.09
Slot_3	49.16	48.74	-0.42
EW_NDE_1	48.94	48.56	-0.38
EW_NDE_2	49.22	48.56	-0.66
EW_NDE_3	50.16	48.56	-1.60
YokeIn_1	45.11	48.68	+3.57
YokeIn_2	45.33	48.68	+3.35
YokeIn_3	47.39	48.68	+1.29
EW_DE_1	50.44	48.65	-1.79
EW_DE_2	50.03	48.65	-1.38
EW_DE_3	49.81	48.65	-1.16
YokeOut_1	46.85	48.62	+1.77
YokeOut_2	47.04	48.62	+1.58
YokeOut_3	48.26	48.62	+0.36

YokeMid	45.34	48.65	+3.31
RotorEndRing	51.41	50.41	-1.00
RotorYoke	54.54	49.50	-5.04

OP n25T25 — top 5 hottest MODEL nodes:

1) Slot_1 T_exp = 41.84 °C T_model = 42.81 °C (Δ=model−exp = +0.97 °C)

2) Slot_2 T_exp = 42.14 °C T_model = 42.81 °C (Δ=model−exp = +0.67 °C)

3) Slot_3 T_exp = 42.36 °C T_model = 42.81 °C (Δ=model−exp = +0.45 °C)

4) EW_DE_1 T_exp = 43.46 °C T_model = 42.79 °C (Δ=model−exp = -0.67 °C)

5) EW_DE_2 T_exp = 43.32 °C T_model = 42.79 °C (Δ=model−exp = -0.53 °C)

Full 18-node table for OP n25T25 (Experiment vs Model):

Node	T_exp [°C]	T_model [°C]	Δ=model−exp
Slot_1	41.84	42.81	+0.97

Slot_2	42.14	42.81	+0.67
Slot_3	42.36	42.81	+0.45
EW_NDE_1	42.48	42.74	+0.26
EW_NDE_2	42.77	42.74	-0.03
EW_NDE_3	43.45	42.74	-0.71
YokeIn_1	39.33	42.75	+3.42
YokeIn_2	39.45	42.75	+3.30
YokeIn_3	40.64	42.75	+2.11
EW_DE_1	43.46	42.79	-0.67
EW_DE_2	43.32	42.79	-0.53
EW_DE_3	43.15	42.79	-0.36
YokeOut_1	40.65	42.71	+2.06
YokeOut_2	40.73	42.71	+1.98
YokeOut_3	41.56	42.71	+1.15
YokeMid	39.54	42.73	+3.19

RotorEndRing	42.48	39.67	-2.81
RotorYoke	45.20	39.08	-6.12

>>