PEARL: IGBT and Diode

Anirudh Katoch

December 5, 2025

1 Introduction

The ongoing transition of modern power systems into converter-dominated systems has made the reliability of power electronic converters critical for grid resilience. In solar PV, inverters cause over 60% of system failures, resulting in an estimated $2.5 billion in annual losses [1] and in wind turbines, converters are the primary source of downtime [2]. Collectively, these data support a clear conclusion that power electronics are the weakest link in the energy transition, and their relatively short 10–12 year lifetimes present a primary challenge to long-term grid planning [1, 3].

The dominant failure mechanisms in high-power semiconductor modules are not stochastic but rather deterministic wear-out processes driven primarily by thermo-mechanical fatigue [4]. Field data indicate that temperature loading accounts for over 55% of failures in power electronic systems [5]. These failures result from repetitive thermal cycling, which is caused by fluctuating electrical power profiles and environmental conditions, and progressively degrades interconnects and solder joints over time [4].

At the same time, inverter-based resources (IBRs) are increasingly required to deliver grid-critical services such as voltage regulation, reactive power support and fault ride-through [6]. These additional operational demands intensify electrical and thermal stresses, accelerating device aging and compounding existing reliability challenges [7–9]. With growing market participation/competition and substantial investments in IBRs [10], ensuring longer asset lifetimes is becoming economically critical. Consequently, accurate and real-time reliability estimation has become essential not only to predict converter wear-out but also to enable lifetime-aware control and maintenance strategies that safeguard both asset longevity and grid stability [11].

1.1 Identified Research Gap

This reliability challenge has created a tooling gap, as shown by the current landscape of analysis software summarized in Table 1. Traditional approaches to reliability estimation for power electronics often rely on finite-element modeling (FEM) or accelerated aging tests [12], which although precise are computationally expensive for long-duration mission profiles and challenging to adapt to different inverter designs. On the other hand, manufacturer tools like the Infineon Mission Profile Tool or Semikron lifetime models are closed-source and specific to certain devices, limiting their use in broader field applications. Therefore, there is a clear need for a standard, open-source and reproducible tool that can connect physics models with practical, long-term reliability assessments.

Table 1: Comparison of available tools for power-electronics lifetime / RUL / electro-thermal analysis.

Tool	Type	Main capabilities	Strengths	Weaknesses / watchouts
PEARL	Open-source / Python	Electrical → losses → thermal → RUL + design optimization	Integrated Python framework; ideal for research	Accuracy depends on empirical parameters; not FEM replacement
CeRULEo	Open-source / Python	Train/evaluate RUL ML models	Good for ML benchmarking	No device-to-mission physics chain
ProgPy / NASA libs	Open-source / Python	PHM / prognostics components	Useful for RUL algorithm validation	Not power-electronics specific
Infineon IPOSIM	Manufacturer-proprietary	Device-specific lifetime estimation	Easy; vendor-calibrated tools	Closed-source; limited to Infineon parts
Semikron / SemiSel	Manufacturer-proprietary	Loss/thermal calculators; lifetime curves	Manufacturer-recommended curves	No system-level RUL integration
PLECS (Plexim)	Commercial (simulation)	Electrical–thermal co-simulation	Well-integrated; widely used	Requires custom RUL integration
ANSYS (Icepak)	Commercial (FEM)	3D CFD/thermal + physics-of-failure	High-fidelity FEM, thermo-mechanics	Heavy, costly; not real-time capable
COMSOL	Commercial (FEM)	3D multiphysics thermo-mechanical	Excellent for hotspot/stress validation	Slow; unsuitable for real-time RUL
MathWorks RUL	Commercial (toolbox)	Data-driven RUL estimation; Simulink integration	Rich ML + hybrid modelling features	License cost; requires physics integration

1.2 Repository Contribution

This paper introduces PEARL, an open-source, physics-based electro-thermal reliability tool for power converters. The primary contribution is the unified modeling of the electrical, thermal and lifetime domains of all power electronics, enabling robust analysis under dynamic, real-world mission profiles for the first time in a standardized, accessible platform. This documention only cover IGBT and Diode analysis.

2 Methodology

The developed lifetime model is primarily grounded in four key references: [7], [13], [14], and the lifetime accumulation framework presented in [15]. Among these, [14] redefined the widely used empirical LESIT-based power cycling lifetime model, which provides the mathematical foundation for estimating the number of thermal cycles to failure, N_f, as a function of various design-dependent constants and thermally induced stress parameters. A small representation of the methodology can be seen in the Figure 1.

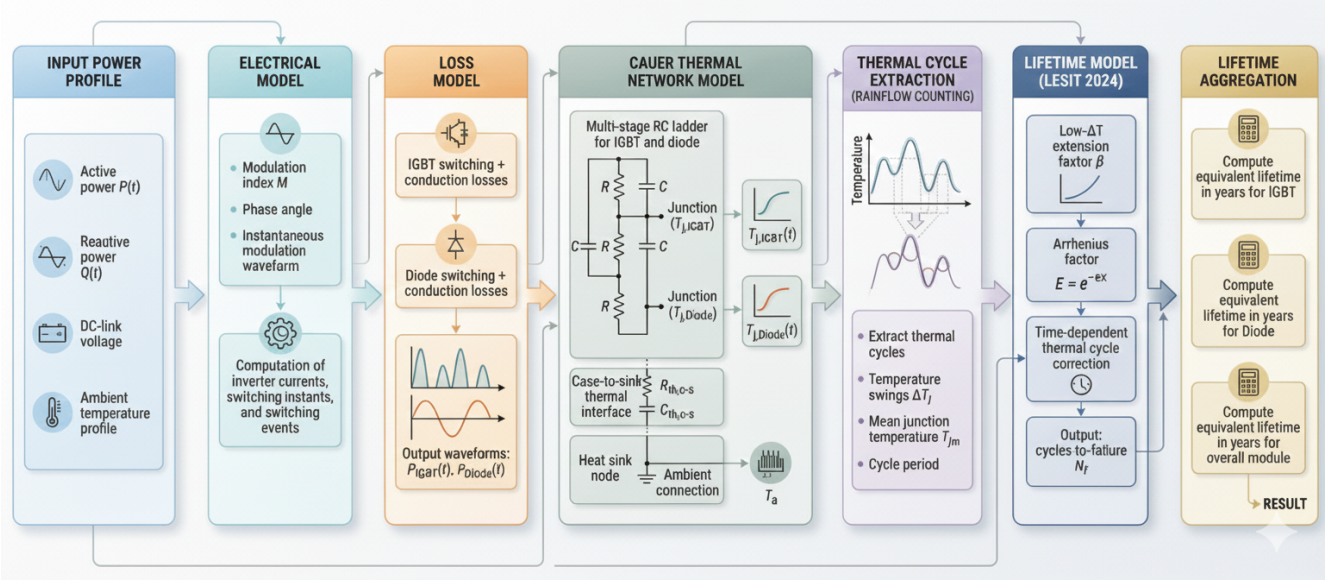

Figure 1: Methodology.

2.1 Lifetime Model

The more recent document [13] extends this already redefined LESIT model in a new power cycle model formulation by incorporating updated experimental results, improved modelling of temperature dependent degradation mechanisms and new correction factors relevant to modern IGBT and diode technologies. Since it represents the state of the art understanding as of 2024, this latest model is adopted in our work in preference to the earlier formulation presented in [14]. The resulting expression for N_f, adapted from [13], is shown in Equation 1.

$$N_f = A_0 \, A_1^{\exp\left(-\frac{\Delta T_j - T_0}{\lambda}\right)} (\Delta T)^{\alpha - \exp\left(-\frac{\Delta T_j - T_0}{\lambda}\right)} \exp\left(\frac{E_a}{k_B T_{jm}}\right) \left(\frac{C + t_{on}^{\gamma}}{C + 2^{\gamma}}\right) k_{thickness} \tag{1}$$

Table 2: Parameter for power cycle model

Parameter	Value	Explanation
A_0	2.9×10^9	Technology Coefficient
A_1	60	Factor of Low ΔT Extension
T_0 [K]	40	Initial Temperature for Low ΔT Extension
λ [K]	17	Drop Constant of Low ΔT Extension
α	−4.3	Coffin–Manson Exponent
E_a [J]	4.50×10^{-20}	Activation Energy
k_B [J/K]	1.38×10^{-23}	Boltzmann Constant
C	1	Time Coefficient
γ	−0.75	Time Exponent
$k_{\text{thickness}}$	1 | 0.65 | 0.5 | 0.33	Chip Thickness Factor (see Sec. 2.5)

The medium junction temperature T_{jm} represents the mean junction temperature during a power-cycling event and is defined as $\frac{T_{j,\max}+T_{j,\min}}{2}$. The junction temperature swing ΔT_{j}, given by $T_{j,\max} - T_{j,\min}$, is the primary thermal stress amplitude acting on the device. The parameter t_{on} denotes the pulse duration or heating time of each cycle, which directly determines the development of the temperature swing. Further details on the definitions and their roles in the power cycling model can be found in [13].

Furthermore the cumulative lifetime of the device over a mission profile is then obtained using Miner's linear damage rule [15]:

$$LC = \sum_i \frac{n_i}{N_{f,i}} \tag{2}$$

where n_i represents the number of thermal cycles experienced at a particular stress condition i and $N_{f,i}$ is the number of cycles to failure predicted by (1) under that same condition. The device is considered to reach its end of life when $LC = 1$, indicating full life consumption.

2.2 Power Losses Model

Paper [7] provides the theoretical and computational framework for determining the electrical and thermal parameters required to evaluate (1). One basically needs the switching and conduction losses of both the IGBT and diode which are calculated as functions of the instantaneous current flowing through each device for a given inverter configuration. The model takes into account the electrical inputs and a comprehensive range of device-specific parameters obtained directly from the respective IGBT and diode datasheets. These parameters are used to compute the instantaneous conduction and switching power losses, which collectively define the total device power dissipation under different loading and reactive power conditions.

2.3 Thermal Model

The electro–thermal behaviour of the IGBT and diode is modelled using a multi-stage Cauer-type RC network that represents the heat flow path from the semiconductor junctions to the ambient environment as can be seen in Figure 2. The thermal network consists of four coupled subsystems that is the IGBT junction ladder, the diode junction ladder, the case to sink interface (thermal paste), and the heatsink. All networks are thermally linked through the case node, thereby enabling simultaneous simulation of both

devices under arbitrary and time-varying power losses. (Note: The resistor and capacitor values shown in the figure are illustrative only. Depending on the chosen size of the RC network, the number of nodes can be adjusted accordingly. In this figure it is assumed that the IGBT and diode branches are modelled with five thermal nodes each, while the thermal paste layer and the heat sink are each represented by a single node.)

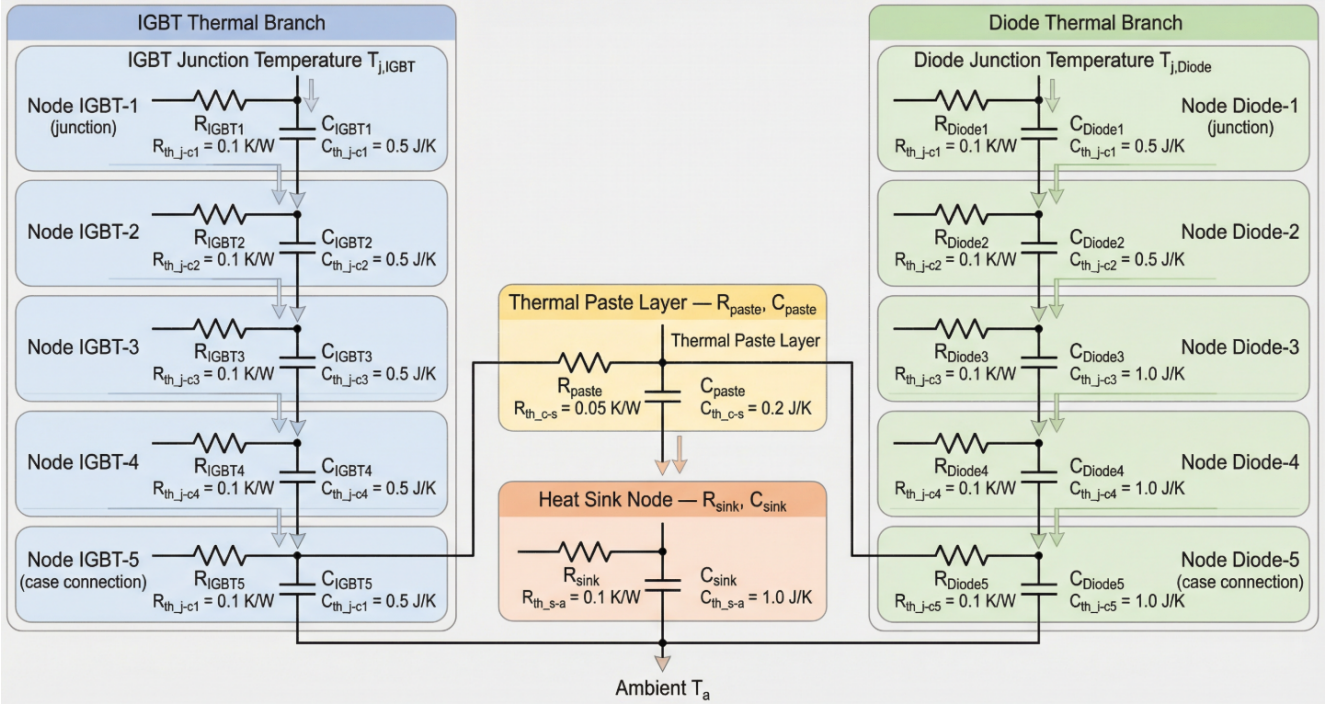

Figure 2: Thermal model methodology.

The Cauer parameters for the IGBT and diode ladders (thermal resistances r and capacitances C for each stage) are obtained directly from the device datasheet Foster curves and converted to an equivalent Cauer representation. This preserves the dynamic heating behaviour across a broad spectrum of time constants, ranging from microseconds to several seconds.

At each simulation time step, the instantaneous power losses of the IGBT and diode computed from the electrical model are injected into the first nodes of their respective RC ladders. The thermal state evolution of the entire network is then obtained by numerically solving the coupled differential equations

$$\frac{dT}{dt} = f(T, P_{\mathrm{IGBT}}(t), P_{\mathrm{Diode}}(t), T_{\mathrm{env}}(t)),$$

where T contains the junction, case, thermal interface, and sink node temperatures. The function $f(\cdot)$ is defined by the heat-balance law for each node, accounting for conduction between adjacent RC elements and heat rejection to the ambient environment. A piecewise-constant profile sampler is used so that the power-loss and ambient-temperature signals may vary at millisecond resolution, while the solver integrates smoothly between updates.

The outputs of the thermal model are the junction temperature trajectories of the IGBT and diode, $T_{j,\mathrm{IGBT}}(t)$ and $T_{j,\mathrm{Diode}}(t)$. These temperature profiles serve as inputs to the rainflow cycle extraction stage, where thermal cycles (ΔT_j, T_{jm}, and cycle period) are identified for subsequent lifetime estimation.

3 Model architecture

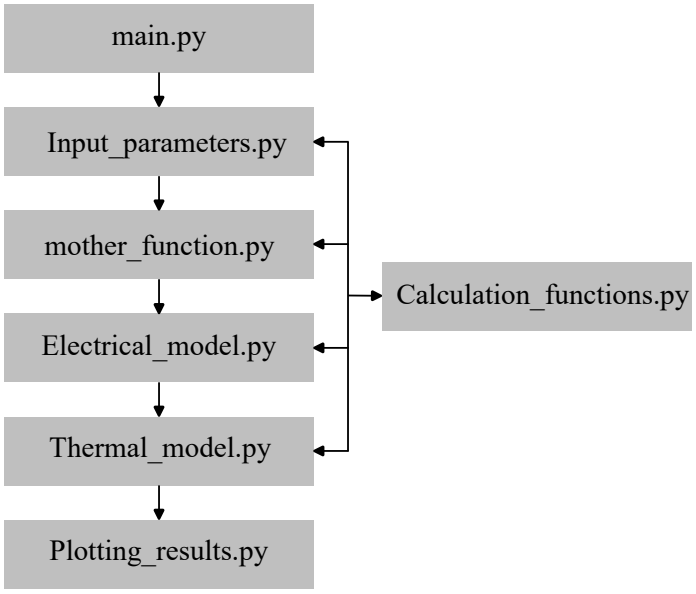

Figure 3: Model architect.

The complete computational framework of the developed model is illustrated in Fig. 3. The flowchart summarizes the interaction among the main Python scripts. The process begins with the definition of the inverter, device, IGBT, Diode and thermal parameters and progresses through electrical loss computation, thermal modelling and lifetime estimation, before final visualization.Each component of the flowchart corresponds directly to a functional module in the codebase. A description of the python scripts can be seen in Table 3.

Table 3: Core modules and their functionality

Module	Description	Inputs	Outputs
Input_parameters.py	Defines all simulation constants and user-tunable parameters.	Electrical and thermal inputs.	Provides global variables.
Calculation_functions.py	Implements all models: electrical loss, thermal equations, lifetime functions etc.	Instant aneous operating points.	Power losses, temperature values etc.
Electrical_model.py	Simulates electrical inputs to calculare power losses.	Electrical variables.	Power Losses.
Thermal_model.py	Simulates dynamic thermal response using Cauer model.	Power loss time series.	Junction temperature series.
mother_function.py	Central orchestrator that calls the individual computation functions in correct sequence.	Mission profile and parameter sets.	Structured numerical results.
main.py	Manage initialization, chunked runs and runtime logging.	User-specified settings.	Folder of results and statistics.
Plotting_results.py	Generates standardized figures(losses, temperature, CDFs etc).	Processed dataFrames.	22 visual figures.

Detailed Description of Python Modules

A comprehensive overview of each Python module that forms the foundation of the developed electro–thermal lifetime assessment framework. Each module in the repository has been designed with a specific purpose ranging from input parameter configuration and mission profile generation to electrical loss computation, thermal modelling, lifetime estimation, and result visualization. The following subsections describe these modules in detail, outlining their objectives, key functions, data flow and interaction with other components of the framework.

3.1 Input_parameters.py

This module initializes and stores all input parameters required by the model. It defines the operating conditions, mission profile, thermal and electrical device parameters, lifetime constants, and control flags for saving and plotting.

3.1.1 Lifetime model parameters

Table 4 summarizes all constants and coefficients which go as an input in the Equation 1.

Table 4: Lifetime model coefficients and constants used in the LESIT-based power cycle model.

Variable	Type	Unit	Description
A0	float	–	Technology coefficient defining the overall scale of the power-cycling lifetime model. Fitted to Semikron–Danfoss test data.
A1	float	–	Scaling factor for the low–ΔT_j enhancement term. Increases predicted cycle count at small temperature swings.
T0_K	float	K	Threshold temperature for the low–ΔT_j behaviour; determines when the exponential correction begins to affect the model.
lambda_K	float	K	Slope parameter controlling how rapidly the low–ΔT_j lifetime enhancement decays with increasing temperature swing.
alpha	float	–	Coffin–Manson exponent describing fatigue sensitivity to ΔT_j. A negative exponent reflects reduced lifetime under larger temperature swings.
Ea_J	float	J	Activation energy of the dominant degradation mechanism (solder fatigue or bond-wire damage), used in the Arrhenius temperature dependence.
kB_J_per_K	float	J/K	Boltzmann constant appearing in the Arrhenius expression.
C	float	–	Time-dependent scaling constant used to normalise the heating-time correction term $(C + t_{\text{on}}^{\gamma})/(C + 2^{\gamma})$.
gamma	float	–	Exponent governing the effect of the heating duration t_{on} on lifetime. Shorter cycle times increase N_f for $\gamma < 0$.
k_thickness	dict	–	Chip-thickness correction factor accounting for mechanical stiffness effects. Values follow Semikron–Danfoss classification: 1.0 for $\leq$1200 V IGBT, 0.65 for 1700 V IGBT/CAL diode, 0.50 for rectifier chips, 0.33 for SiC devices.

3.1.2 Thermal Model Parameters

Table 5: Thermal model parameters used in the junction to ambient Cauer network.

Variable	Type	Unit	Description
T_env	array	K	Ambient temperature profile supplied to the thermal solver. May vary with time depending on operating conditions.
r_I	array	K/W	Thermal resistances of the IGBT junction to case path, extracted from manufacturer datasheet.
tau_I	array	s	Thermal time constants associated with the IGBT thermal resistances, extracted from manufacturer datasheet.
cap_I	array	J/K	Thermal capacitances of the IGBT RC ladder.
r_D	array	K/W	Thermal resistances of the diode junction to case, extracted from datasheet.
tau_D	array	s	Thermal time constants of the diode RC ladder, extracted from datasheet.
cap_D	array	J/K	Thermal capacitances of the diode layers.
r_paste	array	K/W	Thermal resistance of the case to sink interface (thermal paste), extracted from datasheet.
tau_paste	array	s	Thermal time constant of the paste layer, extracted from datasheet.
cap_paste	array	J/K	Thermal capacitance of the paste interface.
r_sink	array	K/W	Heatsink thermal resistance to ambient, extracted from datasheet.
tau_sink	array	s	Thermal time constant of the heatsink, extracted from datasheet.
cap_sink	array	J/K	Thermal capacitance of the heatsink.

3.1.3 IGBT and Diode related parameters and constraints

This section defines the key electrical and operational limits for the semiconductor devices (IGBT and diode) used within the electro–thermal lifetime framework. The defined constraints originate from the device datasheets and manufacturer specifications, and they represent the maximum permissible electrical, thermal, and temporal operating limits for the switching components.

Switching Device Limits

The switching limit parameters specify the maximum allowable voltage, current, and temperature for both the IGBT and diode, as summarized in Table 6. These values are used by the simulation routines to flag or terminate any operating conditions exceeding safe limits.

Table 6: Package maximum electrical and thermal limits used for device derating and safety checks.

Variable	Type	Unit	Description
max_V_CE	float	V	Maximum allowable collector–emitter voltage for the IGBT module. Used to ensure safe operating limits are not exceeded.
max_IGBT _current	float	A	Rated maximum continuous current for the IGBT chip. The simulation checks that peak and RMS currents remain below this limit.
max_IGBT _temperature	float	K	Maximum permissible IGBT junction temperature. Exceeding this limit results in immediate device failure.
max_Diode _current	float	A	Rated maximum continuous current for the free-wheel diode. Ensures the diode is not overstressed during conduction intervals.
max_Diode _temperature	float	K	Maximum permissible diode junction temperature. Used as a thermal safety limit during mission-profile evaluation.

Switching Loss Parameters

Switching losses occur during device turn-on and turn-off transitions due to simultaneous voltage and current overlap. The model defines the switching frequency and effective transition times based on the manufacturer's datasheet. These parameters are temperature dependent but for the simplicity of this model the values are taken at constant temperature. The relevant parameters are shown in Table 7.

Table 7: Switching-loss parameters for the IGBT and diode.

Variable	Type	Unit	Description
f_sw	float	Hz	Inverter switching frequency. Defines the number of turn-on and turn-off transitions per second, determining total switching-loss power.
t_on	float	s	Effective IGBT turn-on time, approximated as $t_{d(on)}+t_r$. Used to compute turn-on switching energy. Assumed constant at 25°C per datasheet.
t_off	float	s	Effective IGBT turn-off time, approximated as $t_{d(off)}+t_f$. Used to compute turn-off switching energy. Assumed constant at 25°C per datasheet.
I_ref	float	A	Reference test current for diode reverse-recovery characterization. Used in normalizing reverse-recovery losses.
V_ref	float	V	Reference test voltage for diode reverse-recovery characterization. Multiplied with Q_{rr} to estimate reverse-recovery energy.
Err_D	float	J	Reverse-recovery energy per diode switching event, computed as $E_{rr} = Q_{rr} \cdot V_{\text{ref}}$. Contributes to total diode switching losses.

Conduction Loss Parameters

The conduction losses model is based on a simplified linear representation of the I–V characteristics of the semiconductor devices. Each device is described by a threshold (knee) voltage and an effective on-state resistance, as given in Table 8. The parameter values are current and temperature dependent but for the simplicity constant temperature and current is assumed.

Table 8: Conduction-loss parameters for the IGBT and diode.

Variable	Type	Unit	Description
R_IGBT	float	Ω	Effective dynamic on-resistance of the IGBT, obtained by linear fitting of the $V_{CE(sat)}$ vs. current characteristic.
V_0_IGBT	float	V	Effective knee voltage of the IGBT conduction characteristic, extracted from linear regression of datasheet curves.
R_D	float	Ω	Effective dynamic resistance of the diode forward conduction characteristic, obtained from linear fitting of V_F vs. current data.
V_0_D	float	V	Effective forward knee voltage of the diode, representing its threshold voltage before significant conduction occurs.

General simulation parameters

Table 9: General simulation parameters controlling numerical resolution, chunking, and plotting options.

Variable	Type	Unit	Description
dt	float	s	Simulation time-step size used for evaluating electrical signals and loss profiles. Determines temporal resolution of the model.
chunk_seconds	int	s	Duration of each simulation segment (chunk). Chunking reduces RAM usage by processing long mission profiles in manageable intervals.
Plotting_electrical_flag	bool	–	Enables or disables plotting of electrical waveforms (currents, voltages, modulation signals).
Plotting_lifetime_flag	bool	–	Enables or disables plotting of lifetime results, including cycle counts and N_f distributions.
Plotting_electrical_loss_flag	bool	–	Enables or disables plotting of conduction and switching loss waveforms.
Plotting_thermal_flag	bool	–	Enables or disables plotting of thermal responses, such as junction, case, and heatsink temperatures.
Plotting_Monte_Carlo_flag	bool	–	Enables Monte Carlo visualization when uncertainty analyses or parameter distributions are simulated.
T0_init	float / None	K	Initial temperature state for the thermal model. Set to None for the first chunk, subsequent chunks use the final temperature from the previous segment.
Cauer_model_accuracy	float	–	Accuracy tolerance for the implicit ODE solver used in the Cauer thermal network. A value of 10^{-3} offers the optimal trade-off between numerical stability and computational effort.

Inverter configuration parameters

Table 10: Electrical input and inverter configuration parameters used in the electro–thermal simulation.

Variable	Type	Unit	Description
P	array	W	Active power profile at the inverter AC side (mission-profile input).
Q	array	VAr	Reactive power profile at the inverter AC side (mission-profile input).Positive for capacitive and negative for inductive
Vs	array	V	Phase RMS AC-side voltage of the inverter (line-to-neutral).
V_dc	array	V	DC-link voltage applied to the inverter. Also used to check V_{CE} limits.
M	float	–	Inverter modulation index. Here fixed to $M = 1$ (no overmodulation modelled).
inverter_phases	int	–	Number of inverter phases: 1 for single-phase, 3 for three-phase operation.
modulation_scheme	str	–	PWM strategy for a three-phase inverter: "svm" (space vector / 3rd-harmonic injection) or "spwm" (sinusoidal PWM).
single_phase_inverter_topology	str	–	Single-phase bridge configuration: "full" or "half". Not used when inverter_phases = 3.
N_parallel	int	–	Number of identical semiconductor switches connected in parallel per inverter leg. Used to scale current per device.
Is	array	A	Phase RMS current on the AC side of the inverter (output current per phase).
phi	array	rad	Phase angle between phase voltage and current. Positive for capacitive, negative for inductive operation.
pf	array	–	Instantaneous power factor.
S	array	VA	Apparent power profile.
f	float	Hz	Fundamental grid frequency.
omega	float	rad/s	Angular grid frequency.

In principle, the full inverter simulation is not required for the electro–thermal model. The core electrical inputs needed by the model are the phase RMS voltage V_s, DC-link voltage V_dc, modulation index M, phase current Is, phase angle phi, power factor pf, grid frequency f, and angular frequency omega. If these quantities are available directly from measurements or a higher-level system model, they can be supplied to the electro–thermal model without using the internal inverter front-end. The function compute_power_flow in Input_parameters.py is only a convenience layer that converts active/reactive power mission profiles (P, Q) and basic inverter settings into the required set of electrical inputs.

3.2 Calculation_functions.py

The Calculation_functions_file.py module which has a class Calculation_functions_class within which we have many functions serves as the computational core of the electro–thermal and reliability modeling framework. It provides a structured collection of static methods that implement the analytical and numerical equations governing inverter operation, semiconductor device losses, thermal stress cycling and lifetime estimation.Below most of the important functions are explained along with the physics behind those function.

3.2.1 compute_power_flow

The function compute_power_flow converts a mission profile defined in terms of active and reactive power (P, Q) into the electrical quantities required by the electro–thermal model, while accounting for the inverter configuration and the number of parallel devices. First, the total active and reactive power are scaled to a per-switch basis as

$$P_{\mathrm{sw}} = \frac{P}{N_{\mathrm{parallel}}}, \qquad Q_{\mathrm{sw}} = \frac{Q}{N_{\mathrm{parallel}}}.$$

From these, the apparent power is obtained as

$$S = \sqrt{P_{\mathrm{sw}}^2 + Q_{\mathrm{sw}}^2}.$$

Depending on the topology and modulation, the function determines the theoretical maximum RMS phase voltage Vs_theoretical that the inverter can generate from the DC-link voltage V_dc and modulation index M:

$$\texttt{Vs\_theoretical} = \begin{cases} \dfrac{M\,V_{\mathrm{dc}}}{\sqrt{2}}, & \text{single-phase full-bridge,} \\ \dfrac{M\,V_{\mathrm{dc}}}{2\sqrt{2}}, & \text{single-phase half-bridge,} \\ \dfrac{M\,V_{\mathrm{dc}}}{\sqrt{6}}, & \text{three-phase, svm,} \\ \dfrac{M\,V_{\mathrm{dc}}}{2\sqrt{2}}, & \text{three-phase, spwm.} \end{cases} \tag{3}$$

If no AC-side voltage profile Vs is provided, it is set equal to Vs_theoretical. Otherwise, the function enforces Vs ≤ Vs_theoretical, raising an error if this condition is violated (overmodulation or inconsistent input).
The power factor is computed robustly by distinguishing special cases:

$$pf = \begin{cases} 0, & P_{\mathrm{sw}} = 0,\ Q_{\mathrm{sw}} \neq 0, \\ \mathrm{sign}(P_{\mathrm{sw}}), & P_{\mathrm{sw}} \neq 0,\ Q_{\mathrm{sw}} = 0, \\ \mathrm{sign}(Q_{\mathrm{sw}})\,\dfrac{|P_{\mathrm{sw}}|}{S}, & P_{\mathrm{sw}} \neq 0,\ Q_{\mathrm{sw}} \neq 0, \end{cases}$$

so that purely reactive operation yields $pf = 0$, purely active operation tends to $|pf| = 1$, and the sign of pf encodes inductive ($pf < 0$) or capacitive ($pf > 0$) behaviour.
The RMS current per phase is then obtained from the apparent power and phase voltage. For single-phase operation,

$$I_s = \frac{S}{V_s},$$

while for three-phase operation S is interpreted as total three-phase apparent power and V_s as phase RMS voltage, giving

$$I_s = \frac{S}{3\,V_s}.$$

The phase angle between voltage and current is computed from the power factor. For nonzero power factor ($pf \neq 0$),

$$\phi = \mathrm{sign}(pf)\ \arccos(|pf|),$$

so that inductive loads ($pf < 0$) yield a negative angle and capacitive loads ($pf > 0$) a positive angle. For purely reactive operation ($pf = 0$), the function sets

$$\phi = \begin{cases} +\dfrac{\pi}{2}, & Q_{\text{sw}} > 0, \\ -\dfrac{\pi}{2}, & Q_{\text{sw}} < 0, \\ 0, & S = 0, \end{cases}$$

which covers the special case of zero apparent power.

3.2.2 check_max_package_current_limit

The function check_max_package_current_limit verifies that the device-level RMS currents of the IGBT and diode do not exceed their package ratings over the entire mission profile. From the AC-side phase current trajectory Is, the maximum RMS phase current is first determined as

$$I_{s,\text{rms}} = \max_t \{I_s(t)\}.$$

Using modulation-dependent current sharing relations between the AC phase current and the individual semiconductor currents in a three-phase inverter leg [7], the equivalent RMS currents in the IGBT and diode are estimated as

$$I_{\text{IGBT,rms}} = \frac{1}{2} I_{s,\text{rms}} \sqrt{1 + \frac{8M}{3\pi}}, \tag{4}$$

$$I_{\text{Diode,rms}} = \frac{1}{2} I_{s,\text{rms}} \sqrt{1 - \frac{8M}{3\pi}}, \tag{5}$$

where M is the modulation index. These values are then compared against the maximum allowable RMS currents max_IGBT_current and max_diode_current from the device datasheet. If either

$$I_{\text{IGBT,rms}} > \texttt{max\_IGBT\_current} \quad \text{or} \quad I_{\text{Diode,rms}} > \texttt{max\_diode\_current},$$

the function raises a ValueError, indicating that the specified mission profile exceeds the safe operating limits of the package.

3.2.3 check_vce

The function check_vce ensures that the DC-link voltage applied to the inverter does not exceed the maximum permissible collector–emitter voltage rating of the IGBT module. Since the blocking voltage stress of the IGBT is directly determined by the DC-link level, the function treats

$$V_{CE} = V_{\text{dc}}$$

and checks whether any value in the supplied DC-voltage profile violates the datasheet limit max_V_CE. For all time samples, the condition

$$V_{CE}(t) \leq \texttt{max\_V\_CE}$$

must hold. If any value exceeds this threshold, the function raises a ValueError, reporting the maximum encountered violation and instructing the user to reduce the DC-link voltage or update the device rating. The function otherwise returns the validated collector–emitter voltage array.

3.2.4 Instantaneous_modulation

This function computes the instantaneous modulation signal (`m`) used to describe the inverter's switching behavior over time [7]. The modulation index (`M`) defines the amplitude of the sinusoidal reference waveform relative to the DC-link voltage, while the term (`omega` $\cdot$ `t` + `phi`) represents its instantaneous phase. The instantaneous modulation function is expressed as:

$$m = \frac{M \sin(omega \cdot t + phi) + 1}{2} \tag{6}$$

This formulation normalizes the modulation waveform between 0 and 1, which is suitable for pulse-width modulation (PWM) schemes. The modulation signal thus defines the instantaneous duty ratio of the inverter's switching devices, directly influencing the output voltage waveform and harmonic content.

3.2.5 IGBT_and_diode_current

This function computes the instantaneous current through the IGBT (`is_I`) and its antiparallel diode (`is_D`) within a single inverter leg [7]. The calculation uses the inverter's RMS output current (`Is`), instantaneous modulation signal (`m`), and grid angular frequency (`omega`) to determine the current waveform over time. The instantaneous base current is first defined as:

$$base = \sqrt{2}\, Is \sin(omega \cdot t)\, m \tag{7}$$

Since power electronic switches conduct unidirectionally, the IGBT current corresponds to the positive portion of the waveform, while the diode current corresponds to the negative portion. These are extracted using:

$$is\_I = \max(base, 0), \quad is\_D = \max(-base, 0) \tag{8}$$

This ensures that the IGBT carries current only during its conduction interval (positive half-cycles), while the diode conducts during the complementary negative intervals, accurately reflecting the switching leg's operation in a half-bridge configuration.

3.2.6 Switching_losses

This function computes instantaneous switching power losses for the IGBT (`P_sw_I`) and the antiparallel diode (`P_sw_D`) [7]. The IGBT loss is modeled from turn-on/turn-off overlap energy, proportional to `V_dc`, device current, and switching transition times `t_on`, `t_off`. The diode loss is based on reverse-recovery energy `Err_D` scaled to the operating `V_dc` and current. Energies per event are multiplied by the switching frequency `f_sw` to obtain power, and negative values are clipped to zero.

$$P\_sw\_I = \left[\frac{\sqrt{2}}{2\pi} V\_dc\, is\_I(t)\, t\_on + \frac{\sqrt{2}}{2\pi} V\_dc\, is\_I(t)\, t\_off \right] f\_sw \tag{9}$$

$$P\_sw\_D = \left[\frac{\sqrt{2}}{\pi} \frac{is\_D(t)\, V\_dc}{I\_ref\, V\_ref} \right] Err\_D\, f\_sw \tag{10}$$

3.2.7 Conduction_losses

This function calculates the instantaneous conduction losses of the inverter's IGBT (`P_con_I`) and diode (`P_con_D`) based on their effective on-resistance and knee voltage [7]. The model accounts for both resistive and threshold voltage contributions, as well as modulation index (`M`) and power factor (`pf`) effects, which influence the average conduction interval of each device. Negative values are clipped to zero to ensure physical validity.

$$P\_con\_I = \frac{is\_I(t)^2}{4} R\_IGBT + \frac{is\_I(t)}{\sqrt{2\pi}} V\_0\_IGBT + \left[\frac{is\_I(t)^2}{4} \frac{8\,M}{3\pi} R\_IGBT + \frac{is\_I(t)}{\sqrt{2\pi}} \frac{\pi\,M}{4} V\_0\_IGBT \right] |pf| \tag{11}$$

$$P\_con\_D = \frac{is\_D(t)^2}{4} R\_D + \frac{is\_D(t)}{\sqrt{2\pi}} V\_0\_D - \left[\frac{is\_D(t)^2}{4} \frac{8\,M}{3\pi} R\_D + \frac{is\_D(t)}{\sqrt{2\pi}} \frac{\pi\,M}{4} V\_0\_D \right] |pf| \tag{12}$$

3.2.8 `rainflow_algorithm`

The function `rainflow_algorithm` extracts thermal cycles from a junction-temperature time series using the standard ASTM rainflow counting procedure. Given a sampled temperature profile $T(t)$ with time resolution `dt`, the algorithm identifies all closed temperature cycles and returns their amplitudes, mean temperatures, heating durations, and cycle counts. The rainflow extraction yields, for each cycle, the range ΔT, mean temperature T_{mean}, cycle count (typically 0.5 for half cycles and 1 for full cycles), and the start and end sample indices of that cycle. The cycle parameters are assembled as

$$\Delta T = T_{\max} - T_{\min}, \qquad T_{\text{mean}} = \frac{T_{\max} + T_{\min}}{2}.$$

The heating duration of each thermal cycle is computed from the distance between the start and end indices:

$$t_{\text{on}} = |\,\texttt{end} - \texttt{start}\,| \times \texttt{dt},$$

which provides an effective thermal cycle period consistent with the power cycle lifetime model.

3.2.9 `cycles_to_failure_lesit`

The function `cycles_to_failure_lesit` implements the LESIT-based power cycling lifetime model as given in (1), using the parameter set from [13]. For each extracted thermal cycle, it takes as input the junction temperature swing ΔT_j (`deltaT`), the medium junction temperature T_{jm} (`Tmean`), and the heating time t_{on} (`thermal_cycle_period`), and returns the corresponding number of cycles to failure N_f. All operations are performed in a fully vectorized manner using `numexpr` for efficiency, so that `cycles_to_failure_lesit` returns an array `Nf` with the cycles-to-failure estimate for each thermal cycle identified by the rainflow algorithm, consistent with the power cycle model in (1).

3.2.10 `miners_rule`

The function `miners_rule` applies Miner's linear damage accumulation rule to convert the per-cycle fatigue predictions of the power cycle model into an overall device lifetime estimate. Given the cycles-to-failure values N_f computed for each thermal cycle and the corresponding cycle counts (typically 0.5

or 1), the accumulated damage is computed as

$$D = \sum_i \frac{n_i}{N_{f,i}},$$

where n_i is the counted number of occurrences of cycle i. Invalid or non-physical entries ($N_f \leq 0$ or non-finite values) are removed prior to evaluation. The equivalent number of full cycles to failure is then

$$N_{f,\text{eq}} = \frac{1}{D},$$

which represents the number of identical cycles that would cause failure under Miner's rule.
The total mission duration is obtained from the length of the current waveform `Is`, allowing conversion of $N_{f,\text{eq}}$ into an expected lifetime in years:

$$\text{lifetime}_{\text{years}} = \frac{N_{f,\text{eq}}\, T_{\text{mission}}}{T_{\text{year}}}, \quad T_{\text{year}} = 365 \times 24 \times 3600.$$

For monte carlo reliability assessment which will be discussed later , the function also evaluates the yearly life consumption

$$LC_{\text{year}} = \frac{1}{\text{lifetime}_{\text{years}}},$$

and computes the corresponding target number of cycles per year under equivalent loading assumption:

$$N_{f,\text{target}} = \frac{T_{\text{year}} \cdot 50}{LC_{\text{year}}}.$$

3.2.11 `check_igbt_diode_temp_limits`

The function `check_igbt_diode_temp_limits` performs a safety validation of the simulated junction temperatures of the IGBT and diode. For each device, the maximum junction temperature over the mission profile is compared against its allowable datasheet limit. The function evaluates

$$\max(T_{j,\text{IGBT}}) \leq T_{j,\text{max}}^{\text{IGBT}}, \qquad \max(T_{j,\text{Diode}}) \leq T_{j,\text{max}}^{\text{Diode}},$$

where the limits $T_{j,\text{max}}^{\text{IGBT}}$ and $T_{j,\text{max}}^{\text{Diode}}$ correspond to the maximum permissible junction temperatures. If either device exceeds its thermal limit, the function raises a `ValueError` and reports the maximum simulated

3.2.12 `normal_distributio_MC`

The function `normal_distributio_MC` generates Monte Carlo samples for both the thermal stress parameters and the power cycle model coefficients by assuming normally distributed uncertainties around their nominal values. Each variable x is sampled as

$$x_i \sim \mathcal{N}\left(x,\ \sigma_x^2\right), \qquad \sigma_x = \texttt{normal\_distribution} \times |x|,$$

so that the coefficient of variation is constant and equal to the user-specified relative deviation. This provides a consistent way to account for measurement uncertainty, mission-profile variability, and parameter dispersion across semiconductor devices.
Three types of parameters are sampled:

- **Thermal stress parameters:** The function generates normally distributed samples for the mean junction temperature $T_{j,\text{mean}}$, the temperature swing ΔT_j, and the heating time t_{on}, producing perturbed stress cycles representative of real operating conditions.
- **Lifetime-model parameters:** The LESIT parameters ($A_0, A_1, T_0, \lambda, \alpha, C, \gamma, k_{\text{thickness}}$) are sampled to capture device-to-device variability. The activation energy E_a and Boltzmann constant k_B remain fixed, since their uncertainties are negligible compared to the other parameters.
- **Output structure:** All sampled arrays are returned as a dictionary, allowing direct propagation of both stress and parameter uncertainty into the lifetime prediction model.

This function is used to perform Monte Carlo lifetime estimation by repeatedly evaluating the cycles-to-failure model with randomized stress and parameter sets, thus producing a statistical distribution of device lifetime.

3.2.13 `residual_deltaT_MC`

The function `residual_deltaT_MC` evaluates the mismatch between the cycles-to-failure predicted by the LESIT power-cycle model and a prescribed target value $N_{f,\text{target}}$. Using the same expression as in (1), it computes

$$N_f = N_f\big(\Delta T_j,\ T_{\text{jm}},\ t_{\text{on}},\ A_0,\ A_1,\ T_0,\ \lambda,\ \alpha,\ E_a,\ k_B,\ C,\ \gamma,\ k_{\text{thickness}}\big)$$

based on the given stress parameters (ΔT_j, T_{jm}, t_{on}) and lifetime-model coefficients. The low-ΔT_j extension, Arrhenius temperature factor, time-dependence scaling, and chip-thickness term are all applied consistently with [13].
The function returns the residual

$$r(\Delta T_j) = N_f - N_{f,\text{target}},$$

which quantifies whether a proposed temperature swing ΔT_j results in a lifetime greater or smaller than the target fatigue strength.

Use in Monte Carlo lifetime inversion

In the Monte Carlo reliability analysis, `residual_deltaT_MC` is used together with Brent's root-finding method to solve for the equivalent temperature swing ΔT_j that yields a desired target lifetime $N_{f,\text{target}}$. For each Monte Carlo sample, the heating time t_{on} and mean junction temperature T_{jm} are known (derived from sampled stress conditions), but ΔT_j must be inferred. The scalar equation

$$r(\Delta T_j) = 0$$

is solved numerically over a physically meaningful interval (e.g. 1–150 K):

$$\Delta T_{j,\text{MC}} = \texttt{brentq}(\texttt{residual\_deltaT\_MC},\ 1.0,\ 150.0,\ \texttt{args} = (N_{f,\text{target}}, T_{\text{jm}}, t_{\text{on}}, \ldots)).$$

This procedure yields a consistent stress triplet ($\Delta T_{j,\text{MC}}, T_{j,\text{mean,MC}}, t_{\text{on,MC}}$) for each Monte Carlo realization, enabling the probabilistic evaluation of device lifetime.

3.3 Electrical_model.py

The file `Electrical_model.py` provides the high-level interface for computing the instantaneous power losses of the IGBT and diode. It builds directly on the primitive functions already introduced in `Calculation_functions.py` (see Section 3.2), namely `Instantaneous_modulation`, `IGBT_and_diode_current`, `Switching_losses`, and `Conduction_losses`. No new physical models are introduced here. This module simply orchestrates and accelerates their use.
The core routine is `_compute_IGBT_and_Diode_power_losses_cached`, which, for a given set of electrical inputs $(I_s, \phi, V_{\mathrm{dc}}, pf, M, \omega)$ and device parameters, generates a time-resolved loss profile over one second with time step `dt`. The computation proceeds as follows:

1. A time vector $t \in [0, 1]$ s is constructed with sampling interval `dt`.
2. The instantaneous modulation waveform $m(t)$ is obtained via
   `Instantaneous_modulation(M,` ω, t, ϕ).
3. Using $m(t)$ and the phase current magnitude I_s, the instantaneous IGBT and diode currents $i_{\mathrm{IGBT}}(t)$ and $i_{\mathrm{Diode}}(t)$ are computed by `IGBT_and_diode_current`.
4. For each switching event, IGBT and diode switching losses $P_{\mathrm{sw,IGBT}}(t)$ and $P_{\mathrm{sw,Diode}}(t)$ are evaluated by `Switching_losses`, based on the DC-link voltage, current waveforms, switching times $(t_{\mathrm{on}}, t_{\mathrm{off}})$, and the reverse-recovery energy parameters.
5. Conduction losses $P_{\mathrm{con,IGBT}}(t)$ and $P_{\mathrm{con,Diode}}(t)$ are computed via `Conduction_losses`, using the linearized on-state models $(R_{\mathrm{IGBT}}, V_{0,\mathrm{IGBT}})$ and $(R_D, V_{0,D})$.
6. The total IGBT and diode power-loss waveforms are then formed as

   $$P_{\mathrm{IGBT}}(t) = \max\{P_{\mathrm{sw,IGBT}}(t)+P_{\mathrm{con,IGBT}}(t),\ 0\}, \qquad P_{\mathrm{Diode}}(t) = \max\{P_{\mathrm{sw,Diode}}(t)+P_{\mathrm{con,Diode}}(t),\ 0\},$$

   and returned together with the intermediate current and loss components.

To avoid recomputing identical 1-second templates for repeated combinations of $(I_s, \phi, V_{\mathrm{dc}}, pf, M, \omega)$, the internal function is decorated with `@lru_cache`. The public wrapper `compute_IGBT_and_Diode_power_losses` first quantizes all scalar inputs to 10 decimal places before passing them to the cached function. This quantization stabilizes cache keys and significantly reduces computational cost when the same operating point occurs multiple times in the mission profile, while keeping the underlying physical models exactly as defined in `Calculation_functions.py`.

3.4 Thermal_model.py

The file `Thermal_model.py` implements the coupled electro–thermal simulation of the IGBT and diode using a Cauer-type RC network. The thermal network represents the complete heat-flow path from each junction to the ambient and consists of four parts:

- an IGBT ladder (junction → case) defined by `r_I`, `cap_I`,
- a diode ladder (junction → case) defined by `r_D`, `cap_D`,
- a case/paste node linking both devices to the heatsink, defined by `r_paste`, `cap_paste`,

- a heatsink node connected to ambient via `r_sink`, `cap_sink`.

The loss profiles of the IGBT and diode, `P_I` and `P_D`, and the ambient temperature profile `T_env` are supplied as time-discrete signals with resolution `dt`. Internally, a piecewise-constant sampler maps the continuous solver time t to the nearest profile index, so that

$$P_{\mathrm{IGBT}}(t) = P_{\mathrm{I}}[k], \quad P_{\mathrm{Diode}}(t) = P_{\mathrm{D}}[k], \quad T_{\mathrm{amb}}(t) = T_{\mathrm{env}}[k], \qquad k = \left\lfloor \frac{t}{\mathtt{dt}} \right\rfloor.$$

The state vector T of the ODE contains the temperatures of all thermal nodes:

$$T = \left[T_{\mathrm{IGBT},1}, \ldots, T_{\mathrm{IGBT},n_I},\ T_{\mathrm{Diode},1}, \ldots, T_{\mathrm{Diode},n_D},\ T_{\mathrm{case/paste},1}, \ldots,\ T_{\mathrm{sink},1}, \ldots \right]^{\mathsf{T}},$$

with indexing chosen such that the first `n_i` entries correspond to the IGBT ladder, the next `n_d` to the diode ladder, followed by the case/paste node(s), and finally the sink node(s). For each node, the heat balance is written in the standard Cauer form

$$C_k \frac{dT_k}{dt} = P_k(t) + \sum_j \frac{T_j - T_k}{R_{kj}},$$

where C_k is the thermal capacitance of node k, R_{kj} are the connecting thermal resistances, and $P_k(t)$ is the local heat input (non-zero only at the IGBT and diode junction nodes). This yields a linear ODE system

$$\frac{dT}{dt} = f(T, t)$$

which is integrated over the mission duration using `scipy.integrate.solve_ivp` with user-selectable method (`"BDF"`, `"Radau"`, `"RK45"`, etc.) and tolerances `rtol`, `atol`. The initial condition `T0_init` is either set to the ambient temperature (first chunk) or reused from the end of the previous chunk to ensure continuity across long mission profiles.
`simulate_igbt_diode_cauer` implements the complete Cauer-type thermal network and computes the temperature evolution of all nodes. At each RHS evaluation, the function constructs a new derivative vector `dTdt` and uses the time step `dt` directly for profile sampling of the power losses and ambient temperature. The function returns the full time vector together with the temperature trajectories of every thermal node. In post-processing, the first nodes of the IGBT and diode ladders are interpreted as the junction temperatures, while the case/paste and heatsink nodes represent the package and sink temperatures required for thermal-limit checking and lifetime calculation.

3.5 `Plotting_results.py`

The file `Plotting_results.py` contains a set of post-processing routines that generate figures for electrical quantities, thermal responses, lifetime metrics, and Monte Carlo results. Each function saves a fixed set of figures to the directory `Figures_dir`.

3.5.1 `Plotting_electrical`

The function `Plotting_electrical` plots the mission-profile electrical quantities that serve as inputs to the electro–thermal model.

- **Figure 1**: Apparent power S, active power P, and reactive power Q versus time.

- **Figure 2**: RMS phase voltage V_s and RMS phase current I_s versus time.
- **Figure 3**: Power factor pf and phase angle ϕ versus time.
- **Figure 4**: DC-link voltage V_{dc} versus time.
- **Figure 5**: Ambient temperature T_{env} (in °C) versus time.

3.5.2 `Plotting_lifetime`

The function `Plotting_lifetime` visualises the rainflow-based lifetime quantities of the IGBT and diode as extracted from the LESIT model.

- **Figure 6**: Log-scale plots of cycles-to-failure N_f for diode and IGBT versus cycle index.
- **Figure 7**: Temperature swing ΔT per cycle for diode and IGBT.
- **Figure 8**: Mean junction temperature T_{mean} (in °C) per cycle for diode and IGBT.
- **Figure 9**: Thermal cycle period (heating time) for each cycle, limited to 0.2 s for clarity.
- **Figure 10**: Rainflow cycle count (0.5 or 1) per cycle index for diode and IGBT.
- **Figure 11**: Bar chart comparing equivalent cycles-to-failure $N_{f,\mathrm{eq}}$ for IGBT and diode using twin y-axes.
- **Figure 12**:Bar chart comparing equivalent lifetimes in years for IGBT and diode using twin y-axes.
- **Figure 12a**: Single bar showing the switch lifetime, defined as the minimum of IGBT and diode lifetime.

3.5.3 `Plotting_electrical_loss`

The function `Plotting_electrical_loss` visualises the detailed electrical losses and device currents at the switching time scale.

- **Figure 13**: Total IGBT and diode power losses $P_{\mathrm{I}}(t)$ and $P_{\mathrm{D}}(t)$ over time.
- **Figure 14**: Instantaneous IGBT and diode currents $i_{\mathrm{IGBT}}(t)$ and $i_{\mathrm{Diode}}(t)$.
- **Figure 15**: IGBT and diode switching-loss components $P_{\mathrm{sw,IGBT}}(t)$ and $P_{\mathrm{sw,Diode}}(t)$.
- **Figure 16**: IGBT and diode conduction-loss components $P_{\mathrm{con,IGBT}}(t)$ and $P_{\mathrm{con,Diode}}(t)$.

3.5.4 `Plotting_thermal`

The function `Plotting_thermal` plots the time-domain thermal response of the Cauer network.

- **Figure 17**: IGBT and diode junction temperatures $T_{j,\mathrm{IGBT}}(t)$, $T_{j,\mathrm{Diode}}(t)$ in °C.
- **Figure 18**: Case/paste node temperature and heatsink temperature in °C, showing the thermal path to ambient.

3.5.5 Plotting_Monte_Carlo

The function Plotting_Monte_Carlo visualises the probabilistic lifetime results obtained from the Monte Carlo analysis, including fitted Weibull distributions.

- **Figure 19**: Histogram of Monte Carlo IGBT lifetimes (in years) with overlaid fitted Weibull PDF.
- **Figure 20**: Histogram of Monte Carlo diode lifetimes with fitted Weibull PDF.
- **Figure 21**: Histogram of switch lifetime, defined as the element-wise minimum of IGBT and diode lifetimes, with a Weibull PDF fitted to the resulting distribution.

3.6 mother_function.py

The routine mother_function(P, Q, T_env) is the top-level orchestration function that executes the complete electro–thermal lifetime evaluation workflow for a given active/reactive power mission profile and ambient temperature trajectory. All physical models and numerical routines defined in Input_parameters.py, Calculation_functions.py, Electrical_model.py, Thermal_model.py, and Plotting_results.py are invoked from this function in a structured sequence:

1. **Parameter initialisation and directory setup:** The mission profiles P, Q, and T_env are passed to Input_parameters_class, which computes all derived electrical, thermal, and lifetime-model parameters (Section 3.1). Simulation options (time step dt, chunk length chunk_seconds, plotting flags, maximum currents, voltages, temperatures, and lifetime limits) are read from params, and the required folder structure for saving intermediate and final results is created via create_simulation_folders().

2. **Chunking of long mission profiles:** To reduce memory usage, the full mission profile is divided into chunks of length chunk_seconds. For each chunk, the corresponding slices of Is, phi, V_dc, pf, and T_env are extracted and processed independently, while thermal state continuity is preserved through T0_init.

3. **Electrical loss calculation (per chunk):** For every second within a chunk, compute_IGBT_and_Diode_power_losses (Section 3.3) is called with the instantaneous operating point $(I_s, \phi, V_{\mathrm{dc}}, pf, M, \omega)$. This yields 1-second templates of IGBT and diode switching and conduction losses, which are concatenated into full chunk-length waveforms (P_I, P_D) along with device currents and individual loss components. The electrical-loss data for each chunk are written to disk as Parquet files.

4. **Thermal simulation (per chunk):** The time-resolved power losses P_I, P_D and the ambient temperature profile are passed to simulate_igbt_diode_cauer (Section 3.4). This produces the temperatures of all thermal nodes, from which the IGBT and diode junction temperatures, case temperature, and heatsink temperature are extracted. If the Cauer solver accuracy is sufficiently tight, junction temperatures are checked against their maximum allowable limits using check_igbt_diode_temp_limits. The final thermal state of the chunk is stored in T0_init and used as the initial condition for the next chunk. Thermal results are saved as Parquet files.

5. **Rainflow counting and LESIT lifetime (per chunk):** The junction temperature profiles of each chunk are processed by rainflow_algorithm to obtain thermal cycles $(\Delta T_j, T_{\mathrm{mean}}, t_{\mathrm{on}}, \mathrm{count})$.

For each device (IGBT and diode), the LESIT power cycling model `cycles_to_failure_lesit` (Section 1) is applied to compute the cycles-to-failure N_f per rainflow cycle. These per-chunk lifetime datasets are written to separate directories for IGBT and diode.

6. **Global lifetime aggregation with Miner's rule:** After all chunks are processed, the chunk-wise lifetime data are reassembled using `read_datafames`. Miner's rule `miners_rule` is then applied to the full set of cycles to obtain equivalent cycles-to-failure $N_{f,\mathrm{eq}}$ and the corresponding lifetime in years for the IGBT and diode. The lifetimes are clipped to user-defined maximum values (`IGBT_max_life`, `Diode_max_life`) and stored in final summary files. Optional plotting of lifetime-related quantities is handled by `Plotting_lifetime`.

7. **Electrical, thermal, and loss visualisation:** Depending on the plotting flags, the mission-profile electrical quantities are saved and visualised by `Plotting_electrical`, the detailed power-loss and current waveforms by `Plotting_electrical_loss`, and the thermal responses (junction, case, and heatsink temperatures) by `Plotting_thermal`.

8. **Monte Carlo lifetime assessment:** A simplified equivalent thermal-stress point for each device is derived from the average junction temperature and an assumed thermal cycle period $t_{\mathrm{on}} = 1/f$. The corresponding ΔT_j that yields the target lifetime $N_{f,\mathrm{target}}$ is obtained by inverting the LESIT model with `residual_deltaT_MC` and Brent's method. Around this equivalent operating point, `normal_distributio_MC` generates normally distributed samples of thermal stress and model parameters. For each Monte Carlo sample, `cycles_to_failure_lesit` is evaluated, and the resulting lifetime distributions for the IGBT, diode, and the combined switch (minimum lifetime) are computed and stored. If enabled, `Plotting_Monte_Carlo` produces Weibull-based visualisations of these lifetime distributions.

9. **Execution-time reporting:** Finally, `mother_function` reports the total wall-clock execution time, providing a simple benchmark for the overall computational cost of the simulation.

In summary, `mother_function` acts as the central driver that links the electrical, thermal, and lifetime sub-models, manages chunking and I/O, and optionally generates all figures required for analysis and documentation of the electro–thermal lifetime behaviour of the IGBT/diode switch.

Important Notes

- **Temporal resolution of mission profiles.** The mission-profile inputs `P`, `Q`, and `T_env` are provided with a resolution of 1 s. All derived electrical quantities such as `Vs` and `V_dc` are therefore also defined at a 1 s resolution on the system level. Within each second, the detailed sinusoidal waveforms used for loss calculation are reconstructed internally using the time step `dt` (see below).

- **Choice of `Cauer_model_accuracy`.** The parameter `Cauer_model_accuracy` in `Input_parameters.py` controls the relative tolerance `rtol` of the implicit ODE solver for the Cauer thermal network. Its suitable value depends on the smallest thermal time constants present in the RC ladder. For very small time constants (e.g. of the order of 10^{-6} s), a tolerance of about 10^{-3} was found to provide a good compromise between numerical accuracy and computation time. Larger tolerances (e.g. above 2×10^{-3}) start to introduce noticeable errors in the junction-temperature trajectories, whereas smaller tolerances increase runtime significantly. In practice, this parameter should be tuned by trial-and-error for a given device and RC network. Smaller the value higher will be computation time.

- **Choice of electrical time step `dt`.** The time step `dt` in `Input_parameters.py` defines the number of samples per second used to construct the electrical sinusoidal waveforms for loss calculations. At 50 Hz, one electrical period is 0.02 s. A choice of `dt` = 0.001 s yields 20 points per electrical cycle, which provides a sufficiently accurate representation of the sinusoidal waveform with limited numerical error. A coarser step such as `dt` = 0.002 s (10 points per cycle) may still work but can introduce additional approximation error. Reducing `dt` increases accuracy but also increases the number of data points, which raises computational cost and especially RAM usage. Hence, `dt` should be selected as a compromise between accuracy and available computational resources.

- **Chunked simulation using `chunk_seconds`.** The full mission profile is not simulated in a single run. Instead, it is divided into segments of length `chunk_seconds`. For each chunk, the electrical, thermal, and lifetime calculations are carried out, and the resulting quantities are immediately written to disk as Parquet files. At the end of a chunk, only the final thermal state is retained (as `T0_init`) and used as the initial condition for the next chunk; all large arrays from the previous chunk are deleted from memory. This chunk-based strategy is essential to keep RAM usage manageable, since the combined electrical, thermal, and lifetime variables make a single full-sweep simulation impractical for long mission profiles.

- **Handling very small temperature swings (ΔT_j).** The power-cycling lifetime models [13] exhibit a steep increase in the predicted cycle count N_f when the junction temperature swing becomes very small. As discussed in Section 2.4 of the Semikron Application Note [13], this "low-ΔT_j behaviour" can lead to extremely large and often impractical lifetime predictions for $\Delta T_j < 30$ K, because the models are extrapolated far outside the region where accelerated tests are performed.

  To avoid unrealistic inflation of the predicted lifetime, a lower bound (cap) should be applied to the temperature swing ΔT_j when evaluating power-cycling capability. Likewise, if the resulting number of cycles or the lifetime obtained after applying Miner's rule exceeds a practical upper limit, the value is capped accordingly. This prevents numerical artefacts in the lifetime estimation and reflects the fact that very small thermal swings do not meaningfully contribute to device ageing over typical operating lifetimes. When such capping is applied, the device can be regarded as surviving its full expected lifetime as specified by the manufacturer.

4 Case Study

To demonstrate the functionality of the proposed electro–thermal lifetime model, a case study was performed. The objective of this study is to evaluate how different power factors influence the lifetime of the semiconductor switch (IGBT and diode). A mission profile of constant electrical loading over a 15-minute interval was selected. In practice, the model simulates this 15-minute segment and then extrapolates the resulting thermo–mechanical degradation to estimate how long the switch would survive if it were to operate continuously under these repeated 15-minute batches.

The apparent power was fixed at the rated value of the inverter, while the active and reactive power components were adjusted to generate the desired power factors. The ambient temperature was held constant at 25°C, and both the DC-side voltage and AC-side phase voltage were also kept constant for all simulations.

The semiconductor device considered in this study is the Infineon IKW50N60H3 IGBT together with its corresponding free-wheel diode. The thermal interface material used between the device and the heatsink is Thermal Grizzly Kryonaut. The cooling system is represented by a passive heatsink model, Cooling Innovations 3-121208U.

All required electrical, thermal, and lifetime-model input parameters for these components have already been detailed in Section 3.1. The full repository contains the exact numerical values used in the simulation; therefore, individual parameter values are not repeated here.

4.1 Results and Discussions

Figure 4 shows the average current of the IGBT and diode as a function of power factor. As the power factor deviates from unity toward either inductive or capacitive operation, the IGBT current decreases, whereas the diode current increases. The IGBT carries most of the load current across all operating conditions, while the diode conducts less frequently and exhibits a lower average current. The symmetry between inductive and capacitive cases indicates that the magnitude of reactive power, rather than its sign, governs the current behavior.

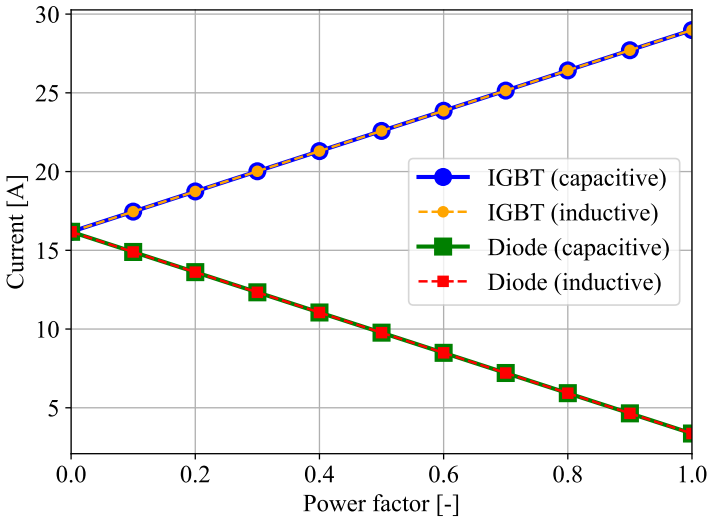

Figure 4: Variation of average current across different power factors.

Figure 4 shows that the current differences between inductive and capacitive operation are visually minimal. To clarify this, Fig. 5 plots the difference as (inductive – capacitive). The results indicate that the IGBT carries slightly higher current under inductive conditions, while the diode carries slightly higher current under capacitive conditions. However, these variations are negligible and have no meaningful impact on interpreting device behaviour. Although such small deviations can accumulate in the thermal model, they are too minor to draw any operational conclusions regarding inductive versus capacitive loading.

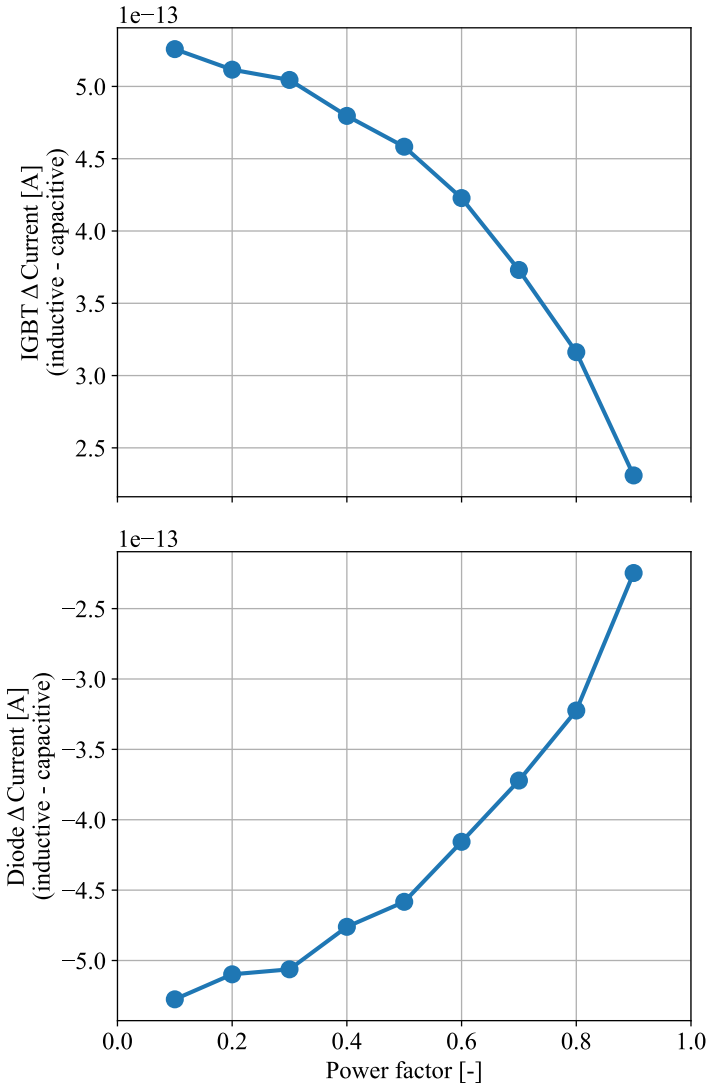

Figure 5: Variation of average current in IGBT and Diode across different power factors.

The conduction power losses of the IGBT and diode for varying power factors are shown in Fig. 6. As the power factor shifts between inductive and capacitive operation, both devices exhibit distinct loss characteristics: IGBT conduction losses increase as the power factor magnitude decreases, whereas the diode shows lower yet similarly dependent behavior. These results demonstrate that reactive power exchange significantly affects device loading even when the apparent power is fixed.

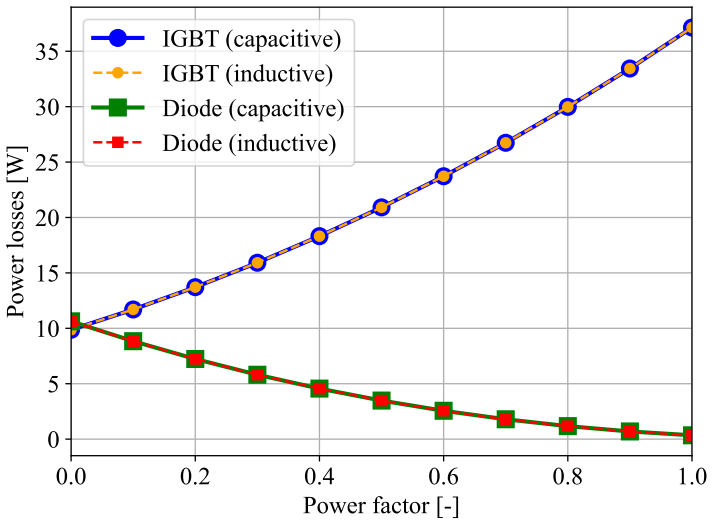

Figure 6: Variation of average conduction power losses across different power factors.

Figure 7 shows the switching power losses of the IGBT and diode as a function of power factor. The IGBT dominates the total switching losses under both inductive and capacitive loading, while the diode contribution remains minor. For the IGBT, switching losses decrease as the power factor magnitude decreases, whereas diode losses increase. This behaviour underscores the sensitivity of switching losses to operating conditions, even when the apparent power is constant.

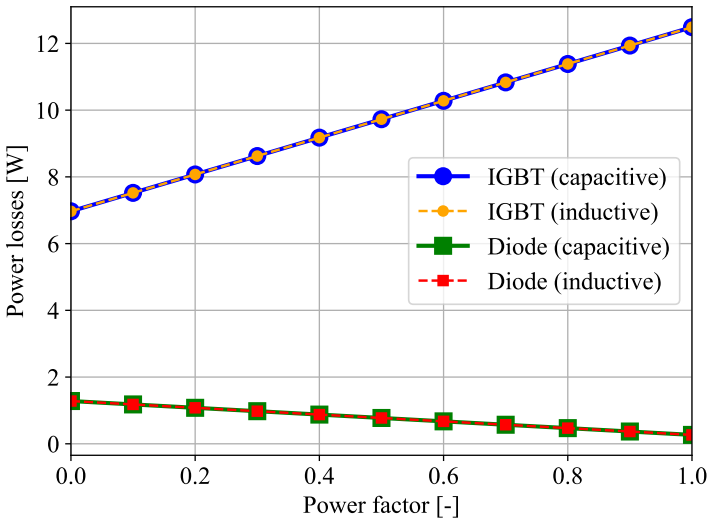

Figure 7: Variation of average switching power losses across different power factors.

Figure 8 shows the total power losses defined as the sum of switching and conduction losses of the IGBT and diode across the full range of power factors. The IGBT dominates the overall losses under both inductive and capacitive conditions, whereas the diode contributes comparatively little. As the power factor magnitude decreases, total IGBT losses decrease while diode losses increase, indicating that higher reactive-power demand imposes greater electrical and thermal stress on the diode. These trends

highlight the importance of power factor dependent loading in evaluating device reliability and thermal performance.

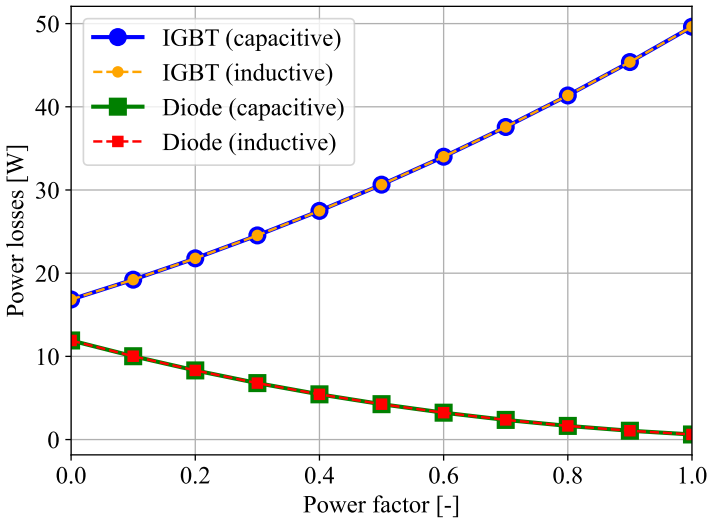

Figure 8: Variation of average total power losses across different power factors.

As shown in Figure 8, decreasing the power factor reduces IGBT losses while increasing diode losses, suggesting a corresponding decrease in IGBT junction temperature and increase in diode temperature. Although Figure 9 exhibits this general behavior, the temperature trends are markedly less pronounced than the loss curves alone imply. This attenuation stems from the strong thermal coupling between the IGBT and diode, which share a common thermal interface material and heatsink, as represented by the RC network in Figure 2. Because both devices dissipate heat through the same thermal path, the temperature of one influences the other, consistent with the zeroth law of thermodynamics. Furthermore, the IGBT typically the dominant heat source exerts a greater influence on the module's thermal state. Although the diode's higher junction-to-case thermal resistance ($R^{\mathrm{D}}_{\mathrm{th},j\text{-}c} \approx 1.05$ K/W) makes it more sensitive to its own losses than the IGBT ($R^{\mathrm{IGBT}}_{\mathrm{th},j\text{-}c} \approx 0.45$ K/W), the shared thermal network is primarily heated by the IGBT. Consequently, as the power factor decreases, the diode experiences competing effects: its rising losses tend to raise its temperature, while reduced IGBT losses cool the common thermal path. The resulting junction temperatures therefore reflect a superposition of these interactions, producing the smoother and less divergent trends observed in Figure 9.

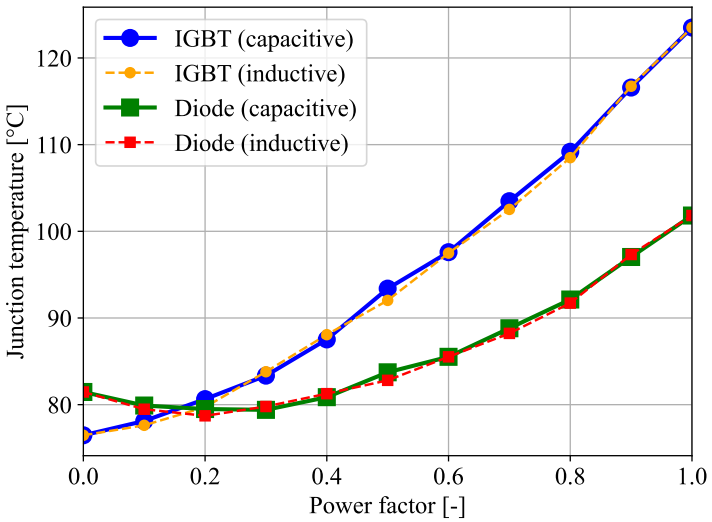

Figure 9: Variation of average junction temperature across different power factors.

Figure 10 shows the average thermal-paste and heatsink temperatures versus power factor. Both increase monotonically as the power factor approaches unity, consistent with higher total semiconductor losses at high–power–factor operation. The pad and sink exhibit nearly identical trends under inductive and capacitive conditions, indicating that the magnitude of reactive power, rather than its sign, governs thermal loading along the shared cooling path. As common thermal nodes for both the IGBT and diode, these elements capture the cumulative heating from both devices. Consequently, their monotonic temperature rise with power factor aligns with the combined loss behavior reported earlier, confirming that the external thermal structure responds primarily to total module heat dissipation rather than device-specific variations.

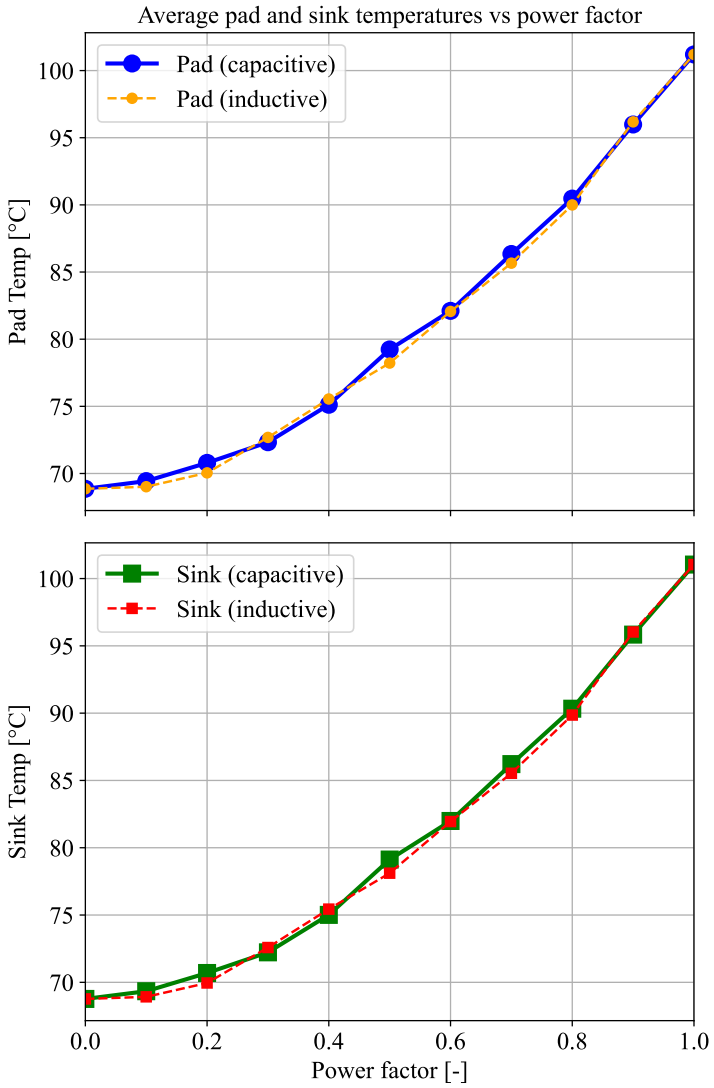

Figure 10: Variation of average thermal paste and heat sink temperature across different power factors.

Figure 11 shows that the junction-temperature swing ΔT_j of both the IGBT and diode closely follows the total power losses in Figure 8. For the IGBT, decreasing power factor reduces device losses and thereby ΔT_j. The diode exhibits the inverse trend, consistent with its opposite loss behavior. Overall, both figures demonstrate a direct relationship between power dissipation and thermal-cycling amplitude: lower losses yield smaller temperature excursions, and higher losses result in larger thermal swings.

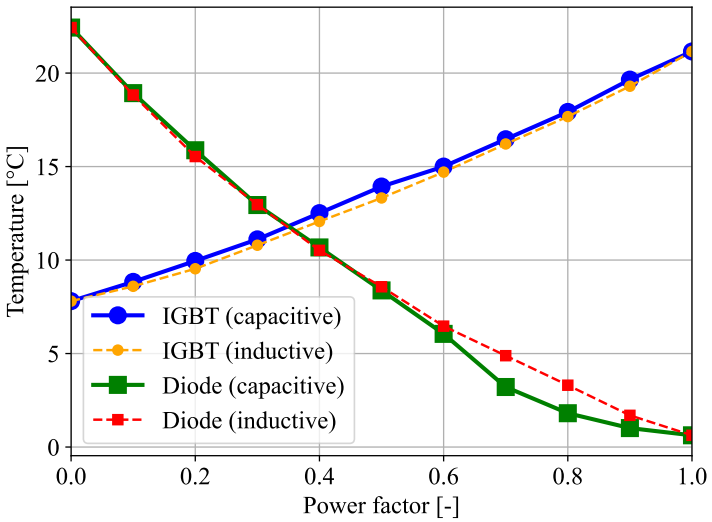

Figure 11: Variation of delta t temperature across different power factors.

Figure 12 shows the predicted IGBT and diode lifetimes versus power factor, obtained from the power-cycling model. IGBT lifetime increases markedly as power factor decreases, reflecting reduced total losses and the smaller junction-temperature swings reported earlier. At low power factors, the resulting ΔT_j becomes very small, producing extremely large cycle-to-failure estimates. As noted in the Important Notes section, such low-ΔT_j conditions can numerically inflate lifetime predictions; thus, values exceeding practical limits (e.g., 20 years) are capped. At higher power factors, where losses and temperature swings are larger, the predicted lifetime decreases accordingly.

Overall, the figure confirms the expected inverse relationship between thermal stress and device lifetime, though interpretation is complicated by the coupled thermal behaviour of the IGBT and diode. At low power factors, the diode exhibits elevated average junction temperatures and large ΔT_j, resulting in reduced predicted lifetime. At high power factors, the behaviour is more nuanced: despite the diode's small ΔT_j, the IGBT operates with substantially higher junction temperatures and temperature swings. Because both devices share a common thermal interface and heatsink, the IGBT's increased losses raise the module's base temperature, thereby increasing the diode's average junction temperature even under modest self-heating. As a result, the diode's lifetime decreases despite its lower ΔT_j at high power factor.

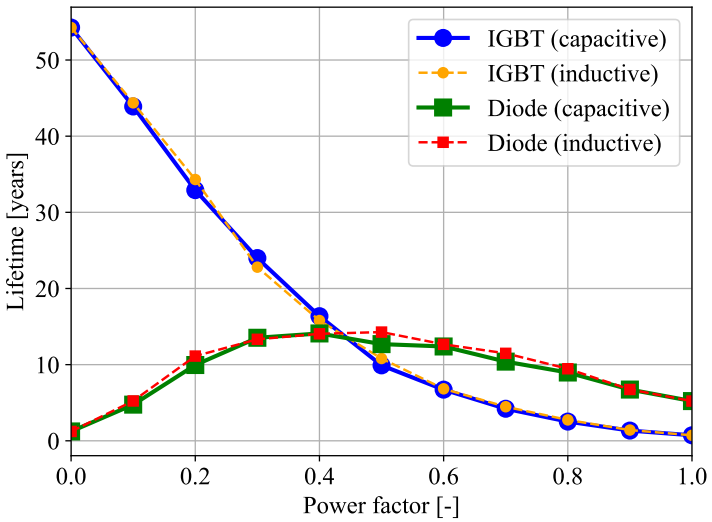

Figure 12: Variation of IGBT and diode lifetime across different power factors.

Figure 13 shows the lifetime of the switching unit, treating the IGBT and its anti-parallel diode as an inseparable pair. In practice, failure of either device causes failure of the entire switch; hence, the switch lifetime is governed by the weaker component. As indicated in Figure 12, the dominant ageing mechanism depends on the power factor: the IGBT limits lifetime in some operating regions, whereas the diode is limiting in others.

The switch-lifetime curve in Fig. 13 represents this interaction by selecting, at each operating point, the minimum lifetime of the two devices. The predicted lifetime drops sharply at high power factors, where the IGBT encounters elevated junction temperatures and larger thermal swings that accelerate wear-out. At low power factors, the diode may incur relatively higher losses, yet the reduced overall thermal stress substantially extends the switch lifetime. Thus, the figure provides an integrated view of how power-factor-dependent thermal loading influences system-level reliability and identifies operating regions most susceptible to premature switch failure.

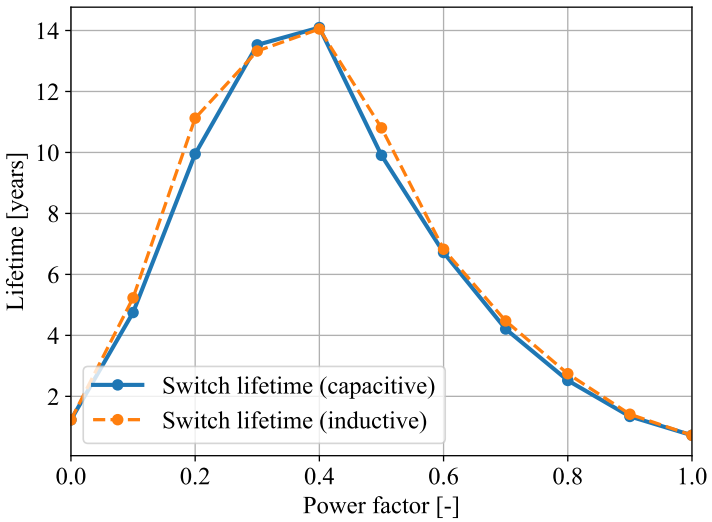

Figure 13: Variation of switch lifetime across different power factors.

In conclusion, thermal stress remains the primary driver of degradation in both the IGBT and diode, with device materials, the thermal interface, and the heat sink further shaping overall lifetime. The coupled thermo-dynamic behaviour of the two devices also influences their aging. As shown in Fig. 13, the Infineon IKW50N60H3 attains its longest lifetime near a power factor of 0.4, whereas most IBRs operate around pf = 0.9. This highlights the importance of enhanced cooling strategies and thermally informed device design to ensure reliable operation under practical grid conditions.

Bibliography

[1] D. J. Friedman, P. L. Hacke, M. Al-Jassim, S. Ovaitt, and S. Shoemaker, "2024 photovoltaic inverter reliability workshop summary report proceedings," tech. rep., National Renewable Energy Laboratory (NREL), 2024.

[2] K. Fischer, J. Besold, M. Lange, and J. Wenske, "Reliability of power converters in wind turbines: A comprehensive field-data analysis," *Energies*, 2019.

[3] M. Owen-Bellini, T. Barnes, K. Boyce, J. Braid, D. Brearly, E. Butler, K. Davis, M. Deceglie, C. Deline, T. Erion-Lorico, R. Flottemesch, A. Gabor, P. Hacke, C. Hansen, H. Hieslmair, W. Hobbs, A. Jain, C. Libby, M. Mikofski, G. Oreski, J. Previtali, I. Repins, L. Schelhas, C. Sillerud, N. de Vries, and K. Whitfield, eds., *2024 PV Reliability Presentation Proceedings*, National Renewable Energy Laboratory (NREL), 2024.

[4] H. Wang, M. Liserre, F. Blaabjerg, P. de Place Rimmen, J. B. Jacobsen, T. Kvisgaard, and J. Landkildehus, "Transitioning to physics-of-failure as a reliability driver in power electronics," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2014.

[5] H. Wang and F. Blaabjerg, "Power electronics reliability: State of the art and outlook," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2021.

[6] P. Pant, T. Hamacher, and V. S. Perić, "Ancillary services in germany: Present, future and the role of battery storage systems," *TechRxiv*, 2024. DOI: 10.36227/techrxiv.171177336.66870344/v2.

[7] Y. Liu, L. M. Tolbert, P. Kritprajun, J. Dong, L. Zhu, J. Hambrick, K. P. Schneider, and K. Prabakar, "Aging effect analysis of pv inverter semiconductors for ancillary services support," *IEEE Open Journal of Industry Applications*, 2020.

[8] J. M. S. Callegari, J. L. Domingues, and J. L. D. Silva, "Lifetime evaluation of three-phase multi-functional photovoltaic inverters," *Electric Power Systems Research*.

[9] Q. Chai, D. C. Yu, Y. Du, and J. Li, "Operational reliability assessment of photovoltaic inverters considering a voltage/VAR control (VVC) function," *Electric Power Systems Research*, 2021.

[10] P. Pant and F.-M. Belz, "Energy market liberalization and the emergence of new energy ventures in germany," *Energy Policy*, vol. 208, p. 114882, 2026.

[11] P. Pant, V. Terzija, T. Hamacher, and V. S. Peric, "Grid agnostic droop control strategy for damping restoration and optimal reactive power-sharing," *IEEE Open Journal of Power Electronics*, 2025.

[12] K. Mainka, M. Thoben, and O. Schilling, "Counting methods for lifetime calculation of power modules," *Power Electronics Europe*, 2011.

[13] A. Wintrich, "Power cycle model for igbt product lines," tech. rep., Semikron Danfoss International GmbH, Nuremberg, Germany. Approved by Dr. Uwe Scheuermann.

[14] U. Scheuermann, R. Schmidt, and P. Newman, "Power cycling testing with different load pulse durations," in *Proceedings of the International Conference on Power Electronics, Machines and Drives (PEMD)*, pp. 1–6, IEEE, 2014.

[15] A. Sangwongwanich, Y. Yang, D. Sera, and F. Blaabjerg, "Lifetime evaluation of grid-connected pv inverters considering panel degradation rates and installation sites," *IEEE Transactions on Power Electronics*, vol. 33, pp. 1225–1236, February 2018.