Thermal Runaway Mitigation through Electrothermal Constraints Mapping for MCPM Layout Optimization

Quang Le[a], Md Maksudul Hossain[a], Tristan Evans[a], Yarui Peng[b], H. Alan Mantooth[a]

[a] Electrical Engineering Department, [b] Computer Science and Computer Engineering Department

University of Arkansas, Fayetteville, AR, USA

qmle@uark.edu, mantooth@uark.edu

Abstract— **Along with the developments in power electronic packaging technology, many studies on design automation for MCPMs layout further push the design limits for their power density and compactness. Among these studies, PowerSynth has shown the complete design flow for MCPMs, which offers a multi-objective layout optimization algorithm and reduced-order models for electrical parasitic extraction and thermal evaluation. While these models are accurate, there is no connection between the electrical parasitic and device temperature during the layout optimization process. Hence, the multi-objective optimization algorithm optimizes these objectives separately without insights into their impacts on the reliability and performance of the wide bandgap (WBG) device. This limitation can lead to a layout solution with undesirable performance compared to the WBG device's safe operation area (SOA). Therefore, this work incorporates the WBG physics-based device knowledge into the power loss calculation for a more accurate electro-thermal prediction in PowerSynth. A better decision can then be made on the most suitable thermal management system.**

Keywords—Electrothermal Design, MCPM, PowerSynth, Machine Learning, Neural Network

I. INTRODUCTION

The traditional design process for Multichip Power Module (MCPM) layout is very manual, tedious, and time-consuming, involving many different analyses and simulation tools. Hence, MCPM layout optimization and design automation have become trending research topics in the power electronics community in the past few years. These studies have developed design automation methods to accelerate further and push the limits for a more compact, reliable, and efficient MCPM design. Among these studies, PowerSynth [1]–[3] has the most developed design flow, which has been validated many times through experiments and measurements. In the last few versions, the combination of constraint-aware layout generation algorithm, reduced-order modeling toolbox, and optimization algorithm allows the tool to search for an ample layout solution space with optimized layouts in both electrical and thermal aspects [2], [4], [5]. However, the tool lacks insights into the Wide Bandgap (WBG) devices' physics [6], which plays a crucial goal in the overall performance and reliability of the MCPM design.

This material is based on work supported by The National Science Foundation under Grant No. EEC-1449548. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science

As has been shown in many studies [7][8], although WBG devices such as SiC can handle much higher junction temperatures than their Si counterpart, the thermal run-away problem is still an issue if the thermal dissipation system is designed poorly. As shown in Fig. 1, the power loss (dashed line) of a SiC device is temperature dependent. A steady state can be reached with a good cooling system when the device's power loss equals the cooling capability. Normally, this steady state is defined by the user by fixing the upper limit for device junction temperature from which a good thermal management system is designed to meet the heat dissipation requirements. In the worst scenario, the device will eventually run into thermal run-away issues due to the exponential increase of WBG device power loss versus temperature. Before this event, the solder attach and the aluminum metallization melts and the circuit fails. In either case, improper cooling system quickly leads to the failure of the whole system. Furthermore, according to [9], this power loss value also depends on the electrical parameters such as parasitic inductance, device’s internal parasitic, gate resistance, etc. Therefore, an accurate estimation on device power loss and its correlation with electrical-thermal parameters during optimization is crucial for the performance and reliability of the MCPM design.

In the MCPM design, especially design automation tool such as PowerSynth. Trade-offs between the electrical parasitic and thermal performance are analyzed through built-in model [4] and Application Programing Interface (API) [5]. The results from these models can serve as inputs for the power loss calculation to ensure the thermal reliability MPCM design during the dynamic performance. In this work, to further improve the computational efficiency of and robustness during the layout optimization, a Feed Forward Neural Network (FFNN) regression model has been trained to capture the relationship among parasitic inductances and device temperature. The FFNN model is reused during layout optimization to accurately mitigate the thermal-runaway issue during the dynamic operation of the devices. This model ensures a fast evaluation of the power loss during the layout optimization while maintaining less than 5% error.

II. METHODOLOGY

There are currently many theoretical models in the literature for accurate prediction of the power loss of the device during the dynamic operation. In this paper, the model in [9] has been reimplemented and combined with the physics-based device

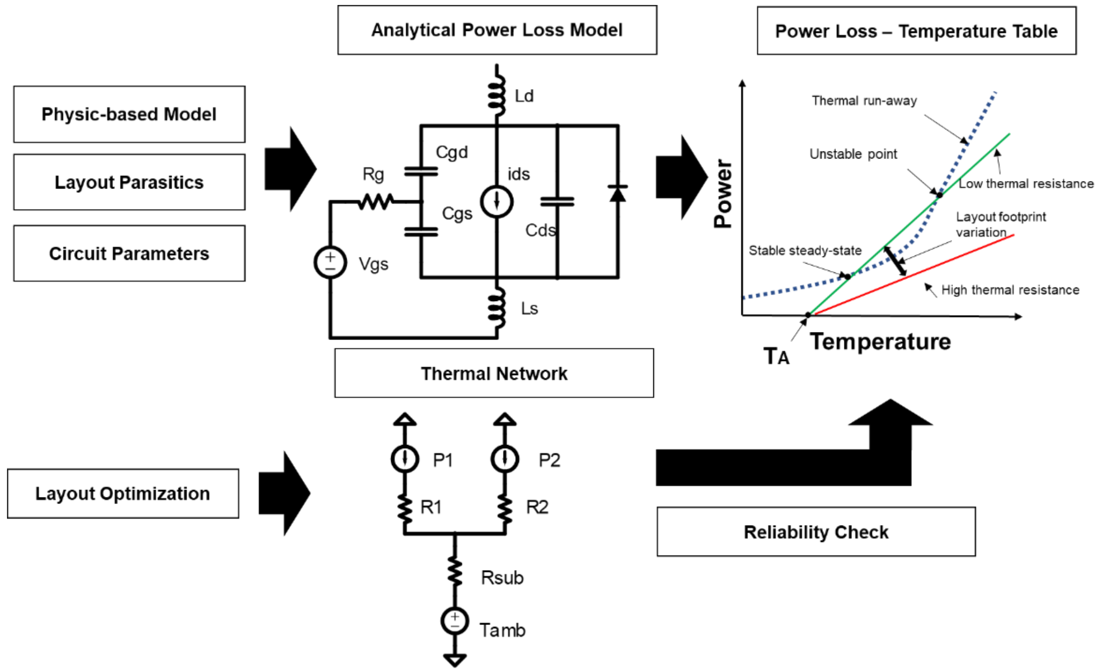


Fig. 1 Thermal Reliability Check Procedure

model [6][10]. Both models are reimplemented in the Python language to have a better interface with the PowerSynth MCPM design tool. To ensure the functionality and accuracy of the models, the models are validated versus previously provided measurement data and information from the datasheet.

A. *Implementation of SiC device model*

The device structure and corresponding equivalent circuit representation is presented in Fig. 2. To explain the device characteristics in terms of the physical attributes, the total drain-source voltage may be divided into 3 parts. The voltage emanating from the parasitic source/substrate and contact resistance, R_s. Next, the voltage resulting from the Ohmic drift resistance in the n- epitaxial layer. Finally, the voltage dropped across the inversion channel is V_{dnrsnr} and is a function of gate-source voltage and is the core of device operation. The channel resistance is non-linear which is a function of both drain-source, V_{dnrsnr} and gate-source voltage, V_{gsnr}. Depending on the drain and gate bias, the drain-source current can be divided into 2 regions namely linear and saturation region. The linear region ($V_{gs} > V_{th}$ and $0 < V_{ds} < V_{dssat}$) current is expressed by equation (1):

$$I_{mos_x} = \boldsymbol{kf_x} \cdot \boldsymbol{kp_x} \cdot (v_{gsi} - \boldsymbol{vt_x}) v_{disi} - \boldsymbol{pvf}^{yx-1} \cdot \frac{v_{disi}^{yx} v_{gsi}^{\frac{2-yx}{yx}}}{1 + \boldsymbol{theta_x}(v_{gsi} - \boldsymbol{vt_x})} \quad (1)$$

Here:

$$yx = \frac{kf_x}{kf_x - \frac{pvf}{2}} \quad (2)$$

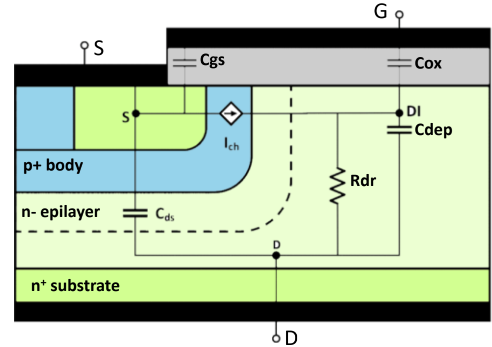


Fig. 2 SiC Power MOSFET devices structure with corresponding parasitic elements

In the above equations, x denotes the low and high region (x=low, high) which arises from the gradual de-trapping of the trap states near the conduction band. ***theta*** accounts for the vertical filed dependent mobility reduction while the pinch-off parameter ***pvf*** represents the gradual transition between linear and saturation region, a characteristic of SiC power MSOFETs. ***kf*** and ***kp*** control the transconductance of the saturation and linear region respectively. It should be noted that the identical equation is used for low and high current components in order for decoupling the parameter influence on each other. The distinguished threshold parameters, ***vt*** are to represent the different effective MOS capacitance.

The saturation region current ($V_{ds} > V_{dssat}$) is expressed by equation (3):

$$I_{mossat_x} = \frac{\frac{1}{2} kp_x (v_{gsi} - \boldsymbol{vt_x})^2}{1 + \boldsymbol{theta_x}(v_{gsi} - \boldsymbol{vt_x})} \quad (3)$$

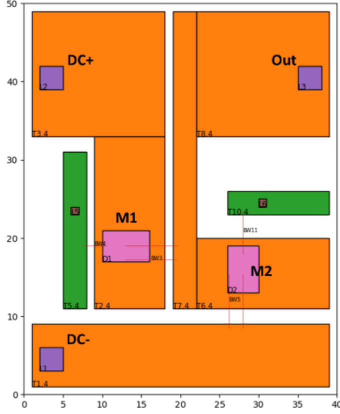


Fig. 3 The MCPM layout for this study

The model includes temperature scalable equations for the temperature dependent parameters such as transconductance, threshold voltage, pinch-off voltage parameter, vertical field mobility reduction parameter etc. based on [11].

Apart from the static characteristics, the dynamic behavior has been captured with the intrinsic inter-electrode capacitance formulations. Three major dynamic components of the power MOSFETs are gate-drain, gate-source, and drain-source capacitances. They are related to the conventional datasheet provided characteristics in the following manner:

$$C_{iss} = C_{gs} + C_{gd} \quad (4)$$

$$C_{oss} = C_{gd} + C_{ds} \quad (5)$$

$$C_{rss} = C_{gd} \quad (6)$$

For this case, the C-V characteristic is considered as an average value in the voltage range between 0-600 V as the input for the power loss model in [9]. Once the parameters of the model are fitted against the datasheet, the channel current I_{mos} and the drift resistance equations are combined in the differential iterative solver using Python SciPy package. Then, the solver is setup to solve the Kirchhoff Voltage Law (KVL) problem in equation (7) to find the correct voltage drop over the drift-resistance:

$$\text{Solve } (v_{dnrsnr}) \text{ where:} \quad (7)$$
$$V_{ds} = \left(r_{drift} + r_s\right) \times I_{mos} + v_{dnrsnr}$$

here, r_{drift} is a function of v_{dnrsnr} and v_{gs}, I_{mos} is the function of v_{gs}

Using this setup, the device characteristics can be directly calculated through the physics-based equation without the help from circuit simulation. With the implementation of the device model in Python, for each different temperature, the temperature-dependent characteristics such as threshold voltage V_{th} or transconductance gfs can be quickly and accurately calculated. These values serve as inputs for the analytical switching loss model for different switching loss calculation versus temperature.

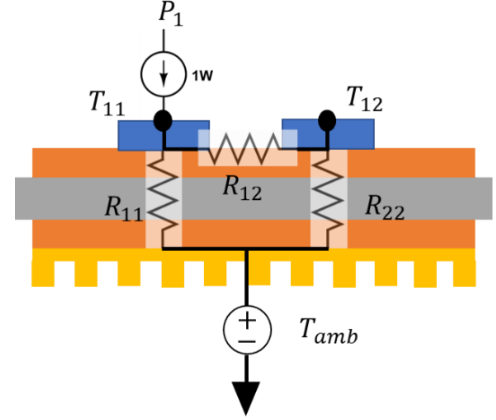


Fig. 4 Thermal network extraction procedure for an MCPM with 2 devices

B. Thermal Netlist Extraction

In this paper, the layout for a half-bridge MCPM (Fig. 3) will be used for the PowerSynth layout optimization study. For each of the layout solutions from the solution space, ParaPower-PowerSynth API [5] has been used to extract the thermal resistance result from each device. This API can compute the steady state temperatures of different MCPM layout elements such as device, trace, substate material, and so on. A 1W heat loss is applied sequentially to top surface of the devices in the setup to find the thermal resistance of the devices in the layout. A heat convection coefficient is also applied to the backside of the baseplate in the ParaPower simulation. The thermal resistance of each device considering the coupling resistance from the device on the opposite switching position can be extracted by calculating the temperature difference between the device top surface and the ambient temperature (T_{amb}) using the equation below.

$$R_{TH} = \frac{T_{device} - T_{amb}}{P_{loss} = 1W} \quad (8)$$

Using equation (8), a 2×2 thermal resistance matrix can be extracted from the layout. Here, R_{11} and R_{22} are the self-thermal resistance of each device. R_{12} is the coupling resistance calculated using the equation (8) while the power is applied to one device and measured at the other. This results in the thermal network in Fig. 4.

C. The Succesive Approximation Method

Since the power loss of a MOSFET is a temperature-dependent value. It is almost impossible to accurately evaluate the steady state value of the circuit without circuit simulation. Several works have demonstrated an iterative method to solve this problem namely the successive approximation method. This successive approximation method (Fig. 5) iteratively updates the temperature and power loss value where the device's temperature difference of the consecutive iterations is calculated. When the temperature difference is smaller than a tolerance value (e.g 0.5 °C) the iterative process is reached, and

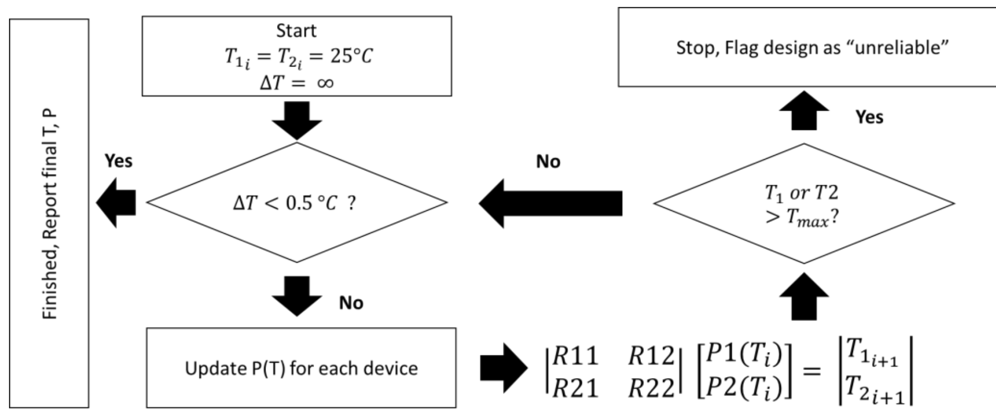


Fig. 5 The Successive Approximation Method

the final steady state temperature is reported. In this work, the steady state junction temperautre is set below 220 C for each device. This is because at this temperature, even when the thermal runaway event does not occur, the solder attach of the device has been melted. This method has been used in [12] and [13]. In [12], a very simple model for power loss evaluation has been used to optimize the runtime. However, this model does not take into account the parasitic parameters. Because of this, the energy loss in [12] is a fixed value for every parasitic and temperature combination. The work in [13] first performed the Finite Element Analysis (FEA) to extract the thermal network. From here, iterative circuit simulations have been done to find the final steady state temperature of the circuit. The only drawback of this method are the time-consuming circuit simulations.

D. Feed Forward Neural Network (FFNN) Regression for Swtiching Loss Modeling.

While the model in [9] is quite fast, performing a thermal sweep for each different layout parasitic configuration while calculating power loss is quite computationally expensive. Furthermore, since the Python implementation of the analytical power loss requires a differential equation solver to solve for turn-on and turn-off periods it is sometimes a bit slow to reach the convergence. Moreover, Python requires overhead interpretation of the code which also slows down the analytical computation. Thus, directly placing the analytical power loss calculation into the layout optimization loop is not preferred. Here, a feed-forward Artificial Neural Network (ANN) regression model using the Scikit-learn machine learning package can be used instead. This model is guaranteed to be faster thanks to its optimized implementation from the Scikit-learn library. The Python SALib library is first used to generate a set of 800 input parasitic parameter combinations for the power loss calculation. This set of parameters includes parasitic inductance values of the gate, drain, and source for each device (Lg, Ld, and Ls). The parameters are randomized in the range of 1-10nH, 1-20nH, and 1-20nH for Lg, Ld, and Ls, respectively. For each variation of Lg, Ld, and Ls a temperature sweep with 50 data points is performed between 25°C and 200°C to evaluate the switching losses versus temperature dependent.

In this calculation, the load current is set to 50 A, and the DC-DC voltage is set to 600V. In the future, the current can be considered as an input for the model. However, the relationship between current and switching loss is quite linear. Also, the circuit parameters are usually defined prior to the layout optimization. Hence, the circuit parameters such as voltage and current are set to be constants now. There are 4000 data points to train the neural network, and 400 data points are randomly taken to test the accuracy of the trained model. The results show less than 5% of error between the simulated and FFNN implementation. It is worth sharing that, during the training process of the model, 40 Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz are run parallelly on a Linux server. The number of epochs for the FFNN is set to 500 to train the most accurate FFNN model. The total time for data collection, model training, and model validation using the multiprocessing evaluation is 160 seconds. It would take 6400 seconds or about 1.8 hours on a single CPU computer. The total time for 4400 evaluations using the FFNN model is 17ms, with less than 5% error. Hence this model is very suitable for the optimization process.

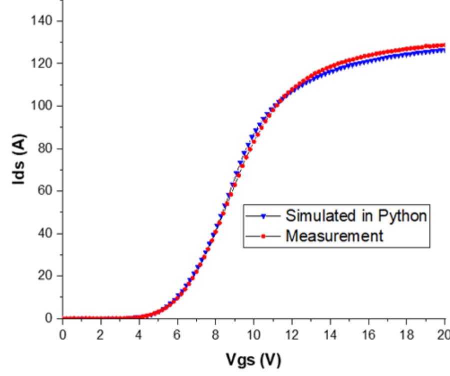
III. Modeling Validation

A. Python Physics-based Device.

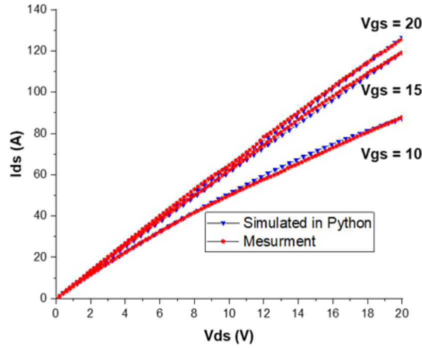
While the accuracy, correctness and functionality of the device model in [6][10] have been proven many times. In this work, for the first time, this model is implemented in Python. Fig. 5 shows the comparison for the device fitting results versus measurement of a C2M0025120D MOSFET from CREE at room temperature. In this study, the same device model has been fitted versus the datasheet for the bare die CPM2-1200-0040B MOSFET from CREE. The model is fitted at various temperatures from the datasheet. Hence, the thermal dependency can be captured in the equations and parameters of the device.

B. FFNN versus Power Loss Model.

To verify the accuracy of the FFNN model, three different combinations of Lg, Ld, and Ls have been randomly selected as input. The total switching loss for each selected parasitic combination is then swept between 25°C to 200°C. The same swept has been done for the analytical loss model. For this



(a)



(b)

Fig. 6 Python implementation vs Measurement (a) Id-Vg (b) Id-Vd

comparison, the switching frequency is set to 10kHz and the power loss for both the analytical and FFNN models are compared in Fig. 7. The results have shown very good fit between the FFNN model and the analytical model. However, the FFNN model is much faster and more preferred for the optimization process. The thermal dependent conduction loss can also be obtained from the SiC MOSFET model due to the change of Rds_{On} versus temperature. Hence, the total power loss for each temperature can be calculated by:

$$P_{tot}(T) = P_{cond}(T) + Esw(T) * f_{sw} \quad (9)$$

IV. Optimization Study and Results

For this study, a layout optimization for a half-bridge layout with the footprint size of (40x50 mm^2) and one device per switching position as shown in Fig. 3 has been performed using PowerSynth. In the first optimization study, a power loss of 65W has been applied to each device where the convection coefficient value is set to be 500 (W/m^2 .K). This 65W value has been chosen since it is closed to the conduction loss value at 25°C of the MOSFET. The maximum temperature and DC-DC loop inductance are used for the optimization target. Fig. 8 illustrates the solution space for 500 solutions, in this case, the loop inductance results are ranging from 10nH to 30 nH for this layout. The temperature results vary from 146 °C to 152 °C. Since the tool has no information about the power loss and

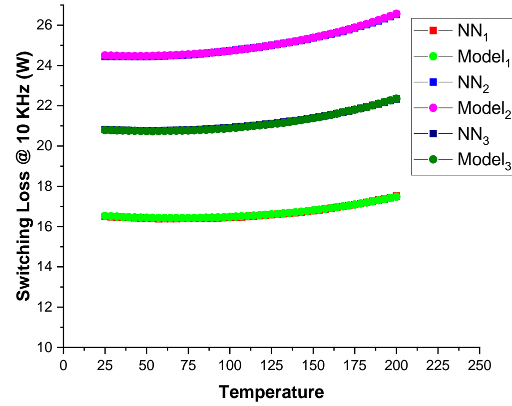


Fig. 7 Curve fitting results from FFNN vs Analytical model

temperature dependency, all of these layout solutions are marked as valid (green).

Once these layouts are obtained, both thermal and electrical netlists are extracted for each device. These netlists serve as input for the FFNN power loss model to quickly find the power loss and temperature-dependent curve. The successive approximation method is applied to each layout solution where the new maximum temperature results are updated. If the process in Fig. 5 does not converge, or the temperature of a device is higher than the maximum temperature set at 220 °C, the process stops. The layout is then flagged as an invalid layout. Because the FFNN-based power loss model is fast and accurate, the average time it takes to run the successive approximation for 500 layouts is about 5 ms on a single core of Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz. Therefore, performing a frequency sweep and seeing its impacts on the layout solutions is possible. Fig 9 shows the updated Pareto-frontier for various switching frequencies from 10kHz to 20kHz. As seen from these results, as opposed to the steady state solutions in Fig. 8, the layout solutions with higher loop-inductance tend to have a higher maximum temperature. This is simply because switching losses increases with higher parasitic values. It is worth noting that many solutions have been invalidated during the process described in Fig. 5. Table 1 below shows the number of invalid solutions for each frequency value.

The total number of iterations has been collected during the iterative evaluation of the successive approximation method. The total number of iterations for this experiment is 18489. Even though the analytical model is fast, this would take up to 7.5 hours to complete. This is mainly due to the overhead interpretation time of the Python language used to implement this model. Conversely, the total time taken for the FFNN model inside the successive approximation method is only 12.5s on the same machine, thanks to the optimized implementation of the method from the Scikit-learn library.

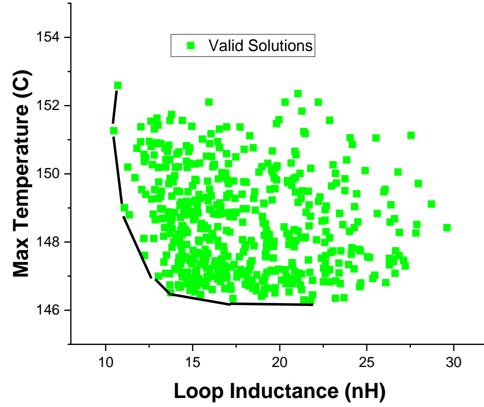


Fig. 8 Initial solution space from PowerSynth

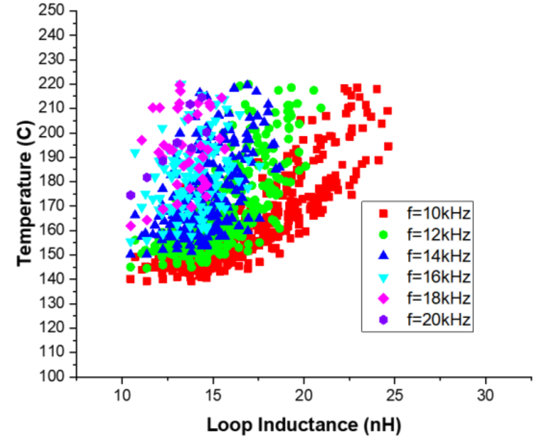


Fig. 9 Frequency sweep impact on the solution space

Table 1 Number of Invalid Solutions for Each Switching Speed

Frequency (kHz)	# Invalid Solutions / 500
10	72
12	186
14	287
16	388
18	460
20	492

V. Conclusion and Future Work

This paper has developed a new approach to performing electrothermal co-simulation. This method allows the designer to verify the functionality of the circuit, where a true performance trade-off between electrical and thermal domains can be achieved. In future work, an FFNN model can be built based on circuit simulation data to capture the unbalance switching losses among parallel devices. This FFNN model shows benefit thanks to it optimized implementation from the Python Scikit learn library.

Acknowledgment

The author would like to thank Dr. Andrea Stratta from the University of Nottingham who visited the University of Arkansas as a postdoctoral researcher during the Fall of 2021. He is a good friend who gives us a lot of advice and initially shares some ideas on this research.

References

[1] T. M. Evans *et al.*, “PowerSynth: A Power Module Layout Generation Tool,” *IEEE Trans. Power Electron.*, vol. 34, no. 6, pp. 5063–5078, 2019.

[2] I. Al Razi *et al.*, “PowerSynth Design Automation Flow for Hierarchical and Heterogeneous 2.5-D Multichip Power Modules,” *IEEE Trans. Power Electron.*, vol. 36, no. 8, pp. 8919–8933, 2021.

[3] Q. Le *et al.*, “PowerSynth Integrated CAD flow for High Density Power Modules,” in *Design Methodologies for Power Electronics (DMC)*, 2021.

[4] Q. Le *et al.*, “Fast and Accurate Parasitic Extraction in Multichip Power Module Design Automation Considering Eddy-Current Losses,” *IEEE J. Emerg. Sel. Top. Power Electron.*, p. 1, 2022.

[5] T. M. Evans *et al.*, “Electronic Design Automation (EDA) Tools and Considerations for Electro-Thermo-Mechanical Co-Design of High Voltage Power Modules,” in *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020, pp. 5046–5052.

[6] M. Mudholkar *et al.*, “Datasheet driven silicon carbide power MOSFET model,” *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2220–2228, 2014.

[7] K. Sheng, “Maximum Junction Temperatures of SiC Power Devices,” *IEEE Trans. Electron Devices*, vol. 56, no. 2, pp. 337–342, 2009.

[8] W. Zhou, X. Zhong, and K. Sheng, “High temperature stability evaluation of SiC MOSFETs,” in *2014 IEEE 26th International Symposium on Power Semiconductor Devices & IC’s (ISPSD)*, 2014, pp. 305–308.

[9] D. Christen and J. Biela, “Analytical Switching Loss Modeling Based on Datasheet Parameters for mosfets in a Half-Bridge,” *IEEE Trans. Power Electron.*, vol. 34, no. 4, pp. 3700–3710, 2019.

[10] M. M. Hossain *et al.*, “An Improved Physics-based LTSpice Compact Electro-Thermal Model for a SiC Power MOSFET with Experimental Validation,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 1011–1016.

[11] M. R. Ahmed, R. Todd, and A. J. Forsyth, “Predicting SiC MOSFET Behavior under Hard-Switching, Soft-Switching, and False Turn-On Conditions,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 9001–9011, 2017.

[12] Q. Le *et al.*, “Fast transient thermal and power dissipation modeling for multi-chip power modules: A preliminary assessment of different electro-thermal evaluation methods,” in *Control and Modeling for Power Electronics (COMPEL), 2016 IEEE 17th Workshop on*, 2016, pp. 1–6.

[13] Y. Nakamura *et al.*, “Electrothermal Cosimulation for Predicting the Power Loss and Temperature of SiC MOSFET Dies Assembled in a Power Module,” *IEEE Trans. Power Electron.*, vol. 35, no. 3, pp. 2950–2958, 2020.