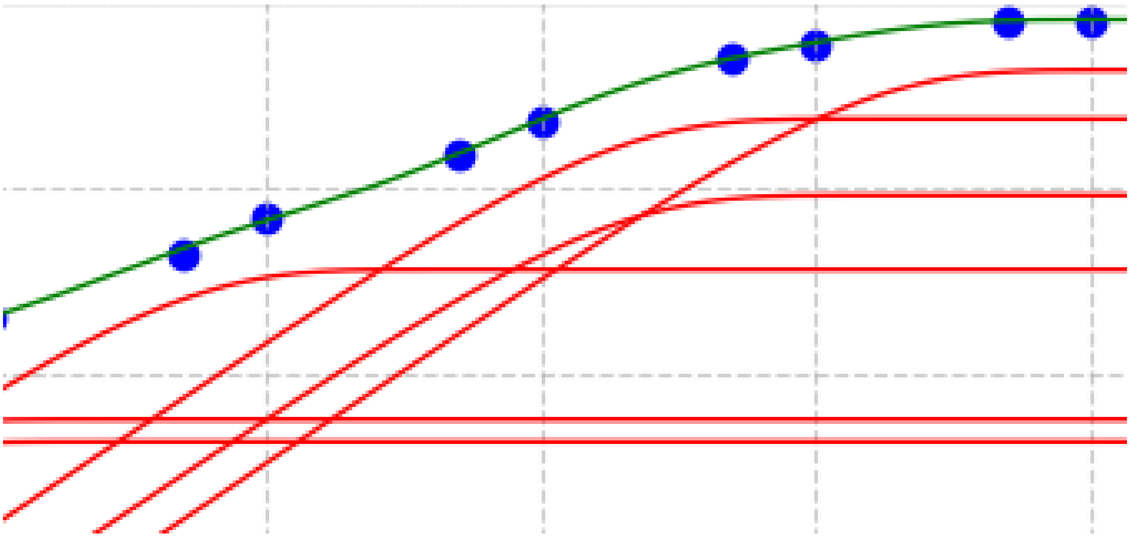


Simplified Heat Simulation of PCB Circuitry

An evaluation of electrothermal simulation in SPICE

Thesis in the bachelor program Electrical Engineering

Rasmus Feltzing

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg 2025
www.chalmers.se

THESIS 2025

Simplified Heat Simulation of PCB Circuitry

An evaluation of electrothermal simulation in SPICE

Rasmus Feltzing

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg 2025

Simplified Heat Simulation of PCB Circuitry
An evaluation of electrothermal simulation using SPICE
Rasmus Feltzing

© Rasmus Feltzing, 2025.

Supervisor: Tomas Karlsson, SEM AB
Examiner: Thomas Hammarström, Department of Electrical Engineering

Thesis 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover Photo: Partial fraction thermal model of a MOSFET.

Written in LaTeX
Gothenburg 2025

Simplified Heat Simulation of PCB Circuitry
An evaluation of electrothermal simulation using SPICE
Rasmus Feltzing
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The project aimed to develop a procedure of simplified heat simulation of printed circuit boards (PCB). The procedure made use of electrothermal analogies to perform heat analysis in spice (Simulation Program with Integrated Circuit Emphasis) software. The procedure included curve fitting algorithms to generate thermally analogous networks of electrical components from thermal impedance graphs provided in semiconductor datasheets as well as programs to convert the parameters into spice compatible format. The project succeeded in developing the procedure but the resulting generated simulation model required manual intervention to fall within 10% deviation from an existing model. The issues are related to the conversion between the foster and cauer topologies.

Key Words: HEAT, SIMULATION, SPICE, LTSPICE, CIRCUIT, THEORY, ELECTROTHERMAL, TRANSISTOR, CAUER, DIODE.

Acknowledgements

I would like to thank Tomas Karlsson for his expertise in electronics and support throughout the whole project. I would also like to thank Åsa Svensson for making this project a reality and for connecting me to the right people inside SEM AB when needed.

Rasmus Feltzing, Gothenburg, May 2025

Contents

List of Figures

1

Introduction

1.1 Background

Svenska Elektromagneter AB, SEM AB, is looking for a method to perform heat analysis of their circuit boards where changing components is easier than in conventional physics simulators. Conventional heat analysis software such as Comsol Multiphysics usually requires knowledge of heat transfer theory as well as knowledge of the program itself which can be time consuming to learn. Another problem is that building a model inside Comsol Multiphysics and similar programs can be time consuming itself even when you have learnt the software. Since the hardware at SEM AB is designed for relatively high temperatures and electrically sensitive applications a combined electro-thermal method of simulation is desired.

1.2 Objective

The objective of this thesis is to devise a method for simulating heat effects on printed circuit boards, PCB. The method is meant to be accessible to electrical engineers and therefore abstract away as much thermodynamics as possible making it easy to use for electrical engineers. The focus will be on transistors and diodes. The method will be based on data found in a datasheet and no thermal tests will be required from the engineer and tools used will aim to be similar to that of the tools used by an electrical engineer such as spice software.

1.3 Limitations

Limitations of the development of this method are the 400 hours time span of a thesis on bachelors level at Chalmers University of Technology. Limitations on the test cases will be on only one diode and one transistor and no tests on physical equipment will be done due to the time limit. The accuracy goal of the method will be limited to 10% deviation from the datasheet and existing models. This number is derived from the 10% intervals between the discrete values of resistors following the E12 series which usually are available in a lab. Provided the linear correlation between resistance and resistive power-loss $P = \frac{I^2}{R}$ this ensures a baseline for usefulness of the method in a regular lab environment while respecting the time limit of the project. The simulation procedure will be limited by availability of thermal behavior in the datasheet of a chosen component.

2

Theoretical Background

2.1 Electro-Thermal Analogies

Heat is energy transferred to the motion of atoms and molecules. Electrical current similarily is the motion of electrical charge carriers such as electrons. The analogies are based on the motion of different but comparable particles. You can therefore model a thermal system much the same way you model an electrical system of circuits and much of the same theories can be used as those presented in appendix A.

2.1.1 Thermal Charge and Thermal Capacitance

Thermal charge can be compared to electric charge which can be transferred and stored in an object. The amount of charges determines the state of charge of the object [1]. The SI unit for thermal charge, also called entropy, is $JK-1$ which also is the unit for heat capacity [2]. Heat capacity can be likened to thermal capacitance and an object can be charged with heat through entropy. The amount of heat capacity determines how much energy is needed to raise the objects temperature such as the amount of electrical capacitance of a component determines how much energy is needed to raise the voltage over the component. An objects thermal capacitance is proportional to it's mass.

2.1.2 Thermal Current

The concept of thermal charges relates only to entropy and heat capacitance in the micro scale. When talking about an analogue to electrical current the perspective must be on the macro scale and the flow of energy, heat flow. This is also the connection between the thermal and electrical models. The electrical power is what gives the conversion from electrical energy to thermal energy. The electrical power consumed in the electrical component is the flow in the thermal circuit and can therefore be modeled as a thermal current.

2.1.3 Thermal Voltage and Potential

The thermal equivalence of electrical voltage is temperature since in a thermal system heat flows from high to low temperature much like electrical current flows from high to low potential. When looking at a system for thermal analysis you assume isolation and constant ambient temperature analogous to when defining an electrical ground/signal return. Thermal potential at a node is the temperature difference between the node and ambient.

2.1.4 Thermal Resistance

Thermal resistance is how much the path through an object or the junction between two objects resists heatflow and therefore how much energy is needed for heat flow to occur. The SI unit for thermal resistance is KW^{-1}. The thermal resistance therefore is the ratio between temperature and heat flow the same way electrical resistance is the ratio between voltage and current. If P_D is the power dissipated in the electrical system, R_θ the thermal resistance and T_x is the temperature at node x then the ratio can be derived as follows.
$T_1 - T_2 \propto P_D$
$T_1 - T_2 = P_D \times R_\theta$
$R_\theta = \frac{T_1 - T_2}{P_D}$
$R_\theta = \frac{\Delta T}{P_D}$
$R_\theta = \frac{\Delta T}{P_D} \Leftrightarrow R_\theta = \frac{\Delta V}{I} (Electrical)$

2.2 Transistor Thermal Model

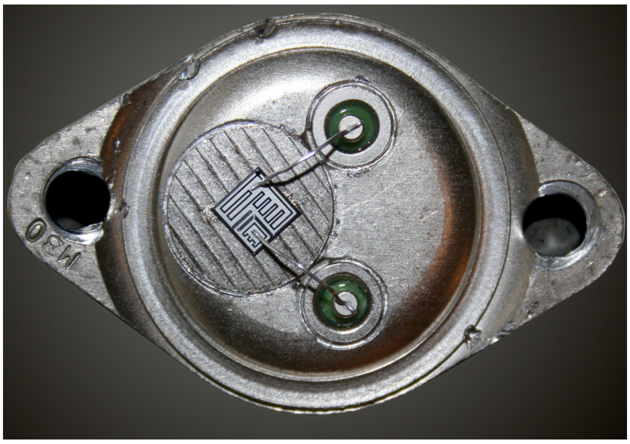


Figure 2.1: Insides of a MJ1000 Darlington Transistor.

Transistors are built around a silicon wafer die which can be seen in figure 2.1. The die is thermally connected to the case and electrically connected to the pins.

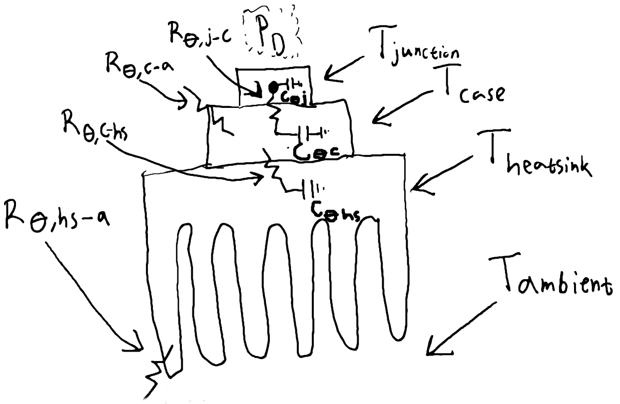


Figure 2.2: Resistor-Capacitor model of a transistor with a heatsink.

Considering the electro-thermal analogies a thermal circuit of a transistor with a heatsink attached can be modeled as in figure 2.2. The two thermal resistances $R_{\theta c-hs} R_{\theta j-c}$ represent the conductive heat transfer between the case and the heatsink and the junction to the case respectively. The two thermal resistances $R_{\theta c-a}$ $R_{\theta hs-a}$ represent both the conductive and convective heat transfer between the ambient, heatsink and case. Radiation will however not be covered in this project. The four nodal temperatures $T_{junction} T_{case} T_{heatsink} T_{ambient}$ represents the temperature of the respective sections of the model at a given time. Lastly the three thermal capacitances $C_{\theta junction} C_{\theta case} C_{\theta heatsink}$ represents the heat capacity of the respective objects. This can be applied to diodes as well. In this project the heatsink will be represented by the PCB and it's ground plane. The amount of momentary energy being converted to heat, distributed power(P_D), in the die will be calculated as $P_D = V_{DS(sat)} \cdot I_D$.

2.3 Cauer and Foster synthesis

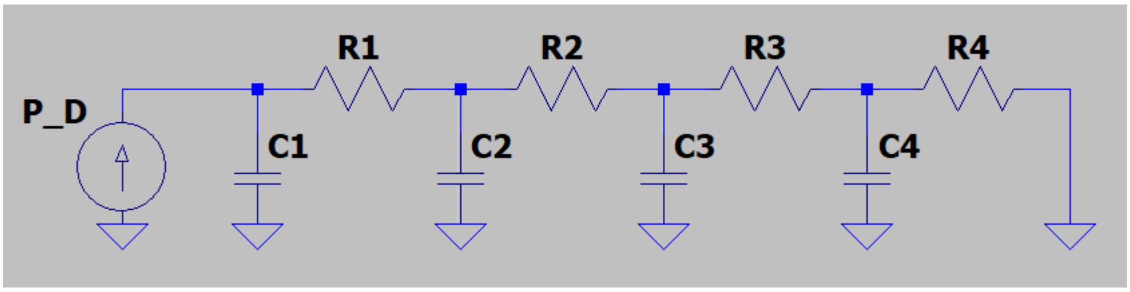


Figure 2.3: A cauer chain of form 1 with 4 RC pairs.

A more structured circuit diagram of the system described in figure 2.2 is a cauer chain as seen in figure 2.3. A cauer network is a network of RC-pairs that mathematically is expanded into a continued fraction which is not easily calculated but has a physical meaning.

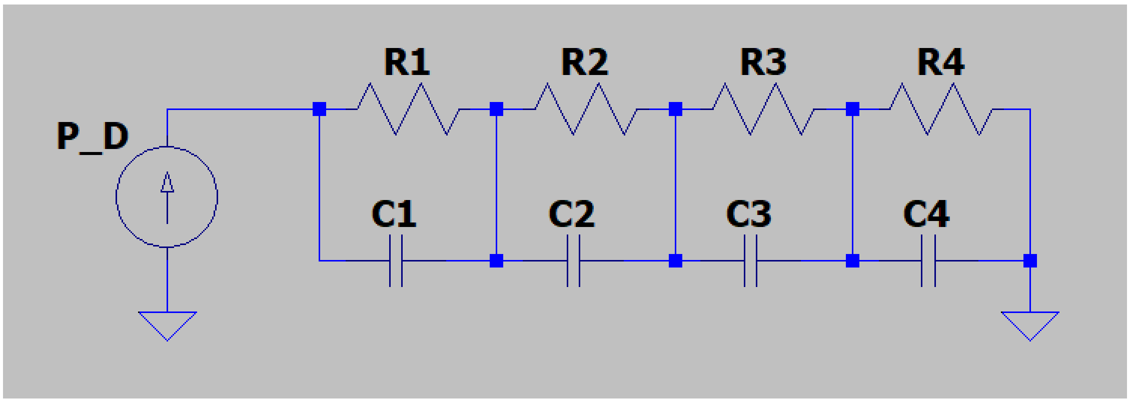


Figure 2.4: A foster chain of form 1 with 4 RC pairs.

A more easily synthesized model is the foster system which is a network of RC-pairs like seen in figure 2.4. A foster network is expanded into partial fractions and the impedance of the foster chain can be mathematically expressed as a partial fraction model $Z_\Theta(t) = \sum_{i=1}^{n} r_i(1-e^{-\frac{t}{\tau_i}})$ where $\tau_i = r_i c_i$ which means $c_i = \frac{\tau_i}{r_i}$. A model function can therefore be made easily as a foster topology. Therefore a Foster network is synthesized and afterwards converted into a cauer which more accurately represents a physical thermal system. Synthesis from foster to cauer can be done by writing the foster chain in the form $Z_{in}(s) = \frac{s(R_iR_{i+1}Ci+1+R_iR_{i+1}C_i)+(R_i+R_{i+1})}{1+s(R_{i+1}C_{i+1})+s^2(R_iR_{i+1}C_iC_{i+1})}$ and factorising the expression until it matches the cauer format $\frac{T_i(s)}{P_i(s)} = Z_{\Theta i}(s) = \frac{1}{sC_i+\frac{1}{R_i+Z_{\Theta i+1}(s)}}$ [3].

2.4 Nonlinear Curve fitting

To match the foster type model function to the datapoints from the datasheetn nonlinear curve fitting was used. It is a process used to find the best fitting curve for a set of data. To find this curve a model function will be used with distinct parameters that the process will change in order to find an optimal solution. The optimal solution will be defined as the the parameters of the foster circuit, the thermal resistances and capacitances, that lead to the least sum of residuals squared. The residuals are the difference between the value of the datapoints and the value of the function at the same point in time [4]. The algorithm used for this fitting is the Levenberg-Marquardt algorithm which is an interpolation between the method of steepest-descent and the Gauss-Newton method [4]. Further description of this algorithm is not in the scope of this report since this project will use a precompiled Levenberg-Marquardt algorithm packaged in SciPy.

2.5 Python

"Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms."[5]. Python features modules installable by the pip module. The module used for math is numpy. The module used for the Non-linear curve fitting is SciPy. The module used for plotting is matplotlib. Jupyter Notebook is a module that provides a notebook interface for your code that you can write three types of enries in: markdown, python and raw. The markdown can be combined with a latex input to write good instructions for your code and provide formulas as documentation inside your program. Python the script language. Raw format was not be used in this project.

2.6 SOAtherm

A tool and library was found integrated into LTspice called SOAtherm which stands for Safe Operating Area thermal. The library had thermo-electrical models for specific NMOS transistors [6] as well as accompanying models for PCB and Heat Sinks [7] included in their library. The transistors featured in SOAtherm are mostly from Infineon [8] with a few inclusions of Nexperia and International Rectifier which is a part of Infineon [9]. Nexperia also offer their own thermo-electrical spice models [10] which are proprietary and encrypted. The PCB models and heatsink feature sufficient settings. The models used were of the topology cauer network form 1 and a region called spirito was considered in their models which is omitted in this project in order to narrow the scope [11]. The spice models where accompanied with symbols to use in LTspice graphical interface and the same symbol with the addition of a port for ambient temperature was used in this project.

3

Method

The methods used in this project consisted of development of the software used in the simulation procedure, development of a test case as well as a literature study to support the validity of the procedure.

3.1 Development of the Simulation Procedure

The simulation procedure is an expansion upon the library SOAtherm in LTspice [6] with accompanying python scripts that were used for the curve fitting. The components that were used for reference were the transistor PSMN3R4-30BLE from Nexperia and the diode SM6TY from ST. The datasheet of both components featured a thermal impedance graph. The transistor was already featured in SOAtherm and therefore had a model to compare with. The diode was chosen to include different type of component with a PN junction in contrary to the N-channel MOSFET.

The simulation procedure was chosen to have the following structure:
1. Find a thermal impedance graph in the components datasheet
2. Choose datapoints that capture the shape of the curve
3. Insert datapoints in a curve fitting program
4. Extract parameters from the curve fit function
5. Convert parameters into a spice model
6. Use the thermal model in your electrical simulations

The simulation method therefore needed the following parts:
- A method for choosing datapoints
- A function that describes a chain of thermally analoguos electrical components
- A program for fitting parameters of the function to a set of datapoints
- A program for extracting the parameters
- A program for converting the parameters to thermal parameters
- A program for formatting the parameters and inserting them into a spice model
- Component specific symbols to graphically connect the parameters to the component in the spice simulation

3.2 Verification

Verification of the accuracy of the model was done in 2 ways. The first way was to see if the thermal impedance graph of the generated model matched with the thermal impedance graph in the datasheet. This applies to both the transistor and the diode. The second way is to compare the pulse-response between the generated model and a preexisting model in LTspice. This will only apply to the transistor due to time limitations.

3.3 Literature Studies

The literature used in this report was found in three ways. Firstly, scientific literature concerning the electro-thermal analogies, algorithms and other theoretical background was found using the chalmers library database, IEEE and scopus. Secondly, the information concerning the software was found on the maintainers websites as for instance analog.com for LTspice. Thirdly a few course books used during my studies was used for basic knowledge of electrical engineering.

4

Results

The results described in this chapter can be devided into 2 parts. Firstly into the developed simulation procedure and secondly into verifcation of it.

4.1 Simulation Procedure Overview

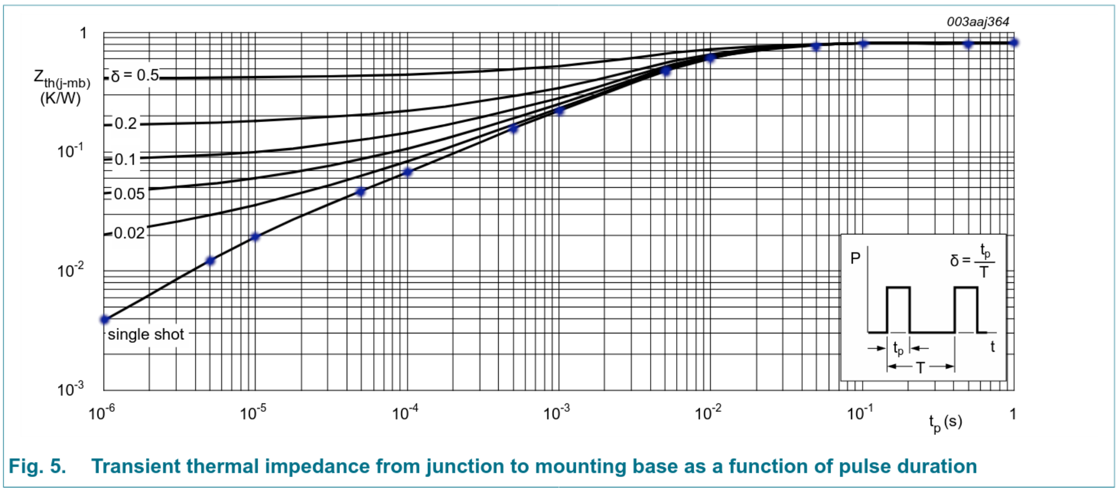
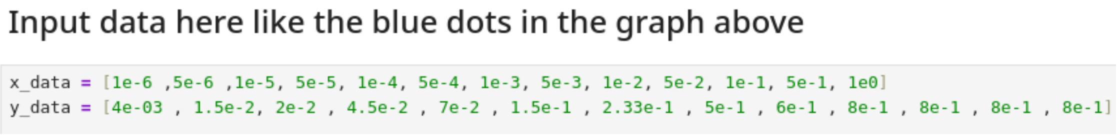


Fig. 5. Transient thermal impedance from junction to mounting base as a function of pulse duration

Figure 4.1: Transient thermal impedance,$Z_{th(j-mb)}$, from junction to mounting base as a function of pulse duration with chosen data points for curve matching as blue dots [12]

As stated in the method the procedure starts by finding the thermal impedance graph for the transistor PSMN3R4-30BLE as seen in figure 4.1. 13 datapoints were chosen and marked in blue on the graph for the single shot response in accordance with the low duty cycles used in the appliances at SEM AB.



Input data here like the blue dots in the graph above

```
x_data = [1e-6 ,5e-6 ,1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1, 1e0]
y_data = [4e-03 , 1.5e-2, 2e-2 , 4.5e-2 , 7e-2 , 1.5e-1 , 2.33e-1 , 5e-1 , 6e-1 , 8e-1 , 8e-1 , 8e-1 , 8e-1]
```

Figure 4.2: Interface for inserting data into the program inside the Jupyter Notebook

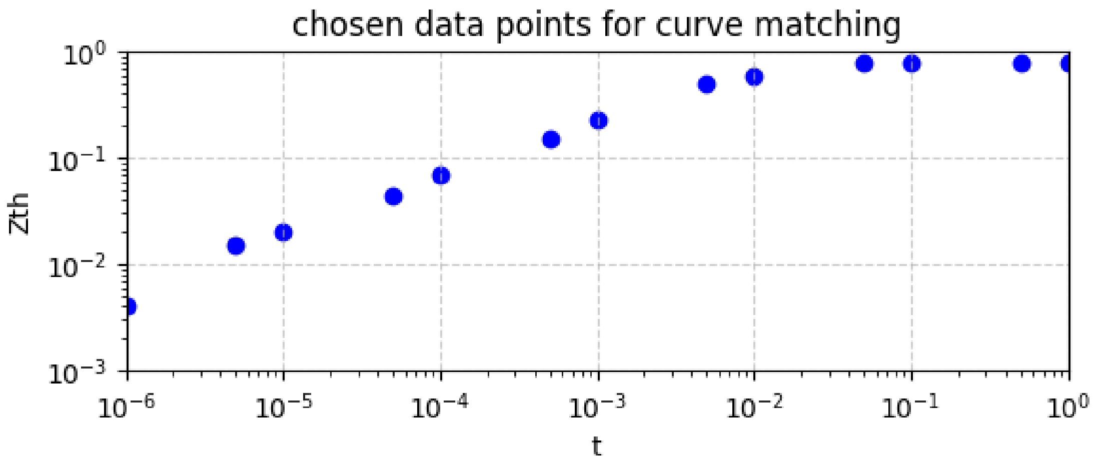


Figure 4.3: Data points from figure 4.1 plotted in the Jupyter Notebook

The data is inserted into the program by editing the arrays as seen in figure 4.2. The datapoints are then plotted as seen in figure 4.3 to confirm that the input was done correctly.

```
[0.00261825 0.00755469 0.0217983  0.06289682 0.18148247 0.52364948]
```

Figure 4.4: Parameter Ri.

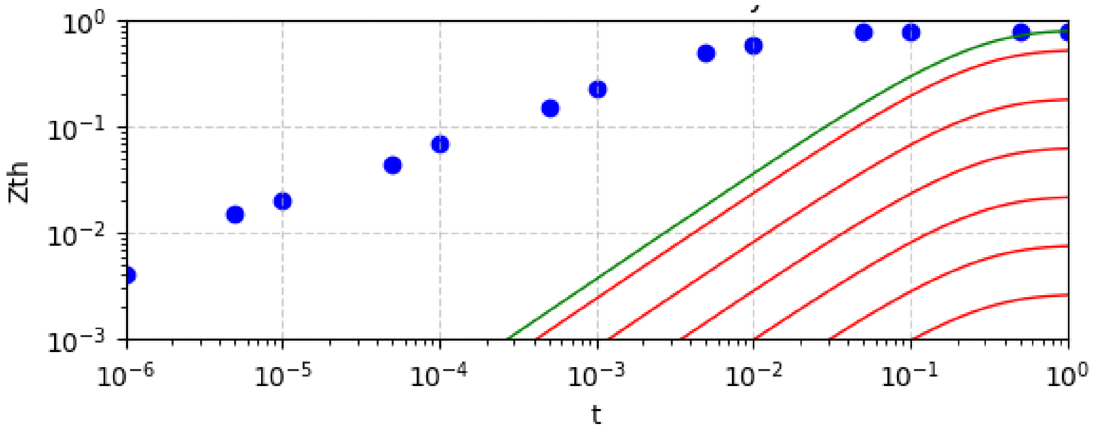


Figure 4.5: Parameter vector Ri seen in figure 4.4 chosen to match steady state value of function. $\frac{1}{\tau_i} = 4.6$. Red graphs are the individual model functions and the green graph is the sum.

The simulation procedure requires guess-work to be able to converge to a solution like most nonlinear curve fitting. The final point should coincide with the last datapoint since the function is assumed steady state after that point therefore $\Sigma_{i=1}^{n} R_i = endvalue = 0.8$. Since the function is almost linear in logarithmic space then a good guess of the R values is a logarithmic distribution between the first value $min = 0.004 = 4 * 10^{-3}$, the $max = 0.8 = 8 * 10^{-1}$ with $n = 6$ RC-pairs. To normalize the sum of the values to the max value the following algorithm is used:

$normalized(i) = distribution(i) * \frac{max}{\Sigma_{i=1}^{n} R_i}$. This gives the parameters in figure 4.4 which produces the graphs in figure 4.5.

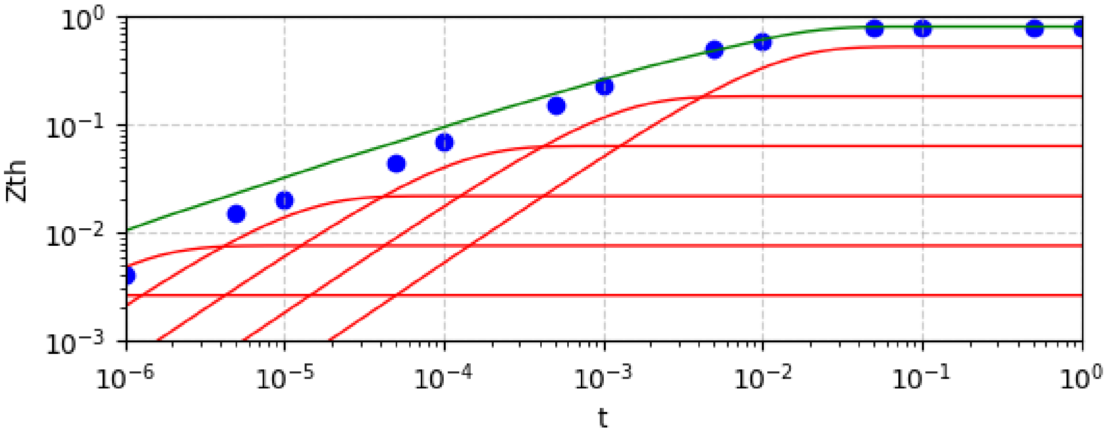
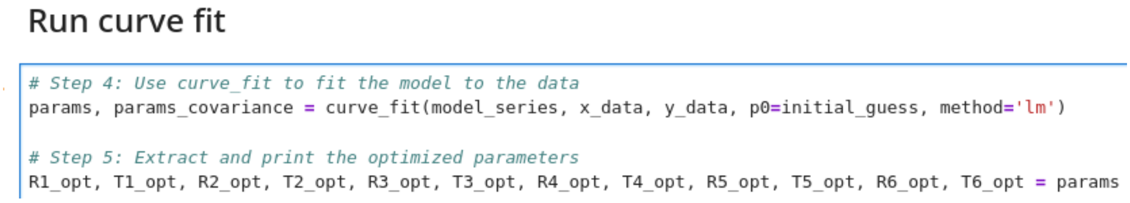


Figure 4.6: Parameter vector Ti chosen to match curvature of function.

If the time constants are distributed the same way but inversely since $(1 - e^{-t \cdot T_i})$ The lowest value is paired with the largest $\frac{1}{\tau_i}$. It can be observed in figure 4.5 that the function stops being linear in logarithmic space at $t = 10^{-2}$ therefore our first $\tau_1 = 10^{-2}$ scaling by a factor of 10 up until $\tau_6 = 10^{-7}$. The result can be seen in figure 4.6. If the function was not linear in logarithmic space then using previous analysis and moving the individual functions by modifying r_i and τ_i.

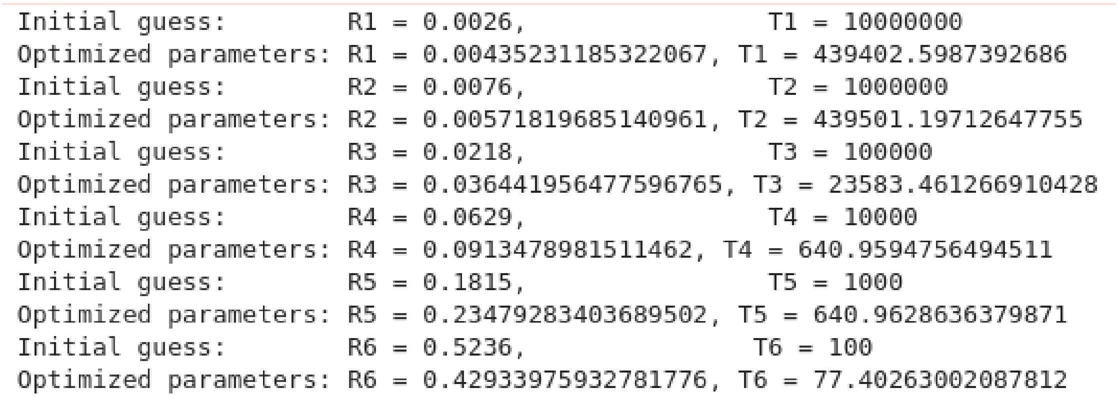


Run curve fit

```
# Step 4: Use curve_fit to fit the model to the data
params, params_covariance = curve_fit(model_series, x_data, y_data, p0=initial_guess, method='lm')

# Step 5: Extract and print the optimized parameters
R1_opt, T1_opt, R2_opt, T2_opt, R3_opt, T3_opt, R4_opt, T4_opt, R5_opt, T5_opt, R6_opt, T6_opt = params
```

Figure 4.7: Command used for curve fitting parameters to data points.



```
Initial guess:        R1 = 0.0026,                T1 = 10000000
Optimized parameters: R1 = 0.00435231185322067, T1 = 439402.5987392686
Initial guess:        R2 = 0.0076,                T2 = 1000000
Optimized parameters: R2 = 0.00571819685140961, T2 = 439501.19712647755
Initial guess:        R3 = 0.0218,                T3 = 100000
Optimized parameters: R3 = 0.036441956477596765, T3 = 23583.461266910428
Initial guess:        R4 = 0.0629,                T4 = 10000
Optimized parameters: R4 = 0.0913478981511462, T4 = 640.9594756494511
Initial guess:        R5 = 0.1815,                T5 = 1000
Optimized parameters: R5 = 0.23479283403689502, T5 = 640.9628636379871
Initial guess:        R6 = 0.5236,               T6 = 100
Optimized parameters: R6 = 0.42933975932781776, T6 = 77.40263002087812
```

Figure 4.8: Terminal output showing intial parameters compared to optimized parameters where $Ti = \frac{1}{\tau_i}$.

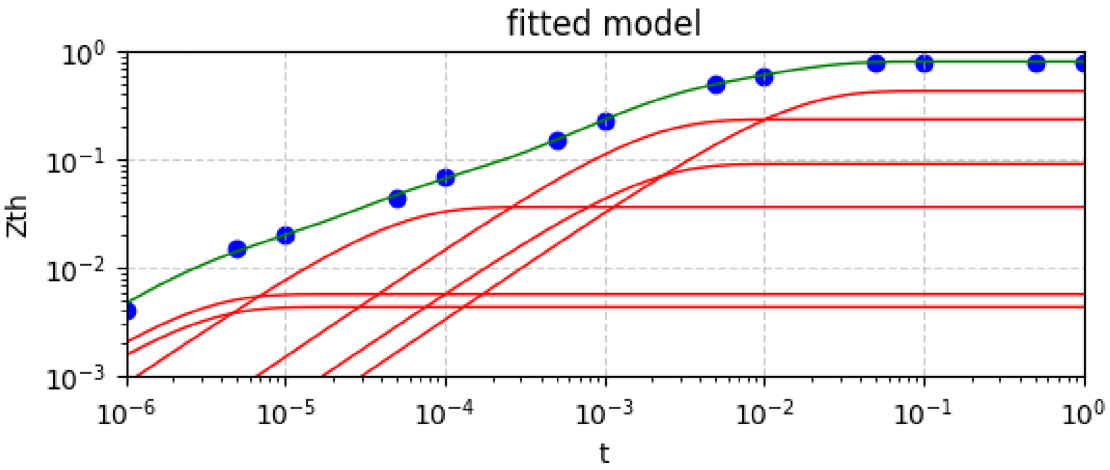
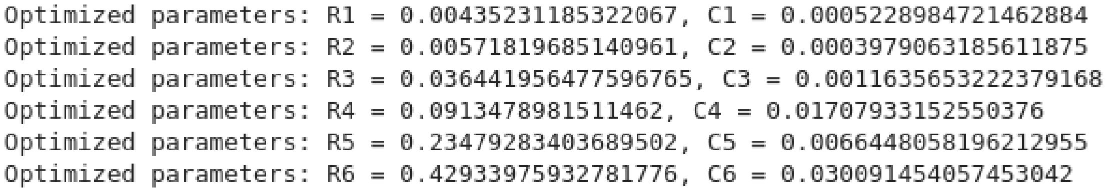


Figure 4.9: Parameter vector Ti chosen to match curvature of function.

Now using the Levenberg-Marquardt algorithm from SciPy to fit the model function more precisely to the datapoints. The commands in figure 4.7 is run to extract the curve fitted parameters and the covariance matrix for the parameters. Detailed syntax for arguments are found in the SciPy documentation [13]. The parameters are extracted to individual variables which are printed out in the notebook terminal, fig 4.8 as well as plotted through the partial fraction model as seen in figure 4.9 where the red curves are the individual RC-pairs and the green curve is the sum which now more closely fits the data points and resembles the original graph provided in the datasheet, figure 4.1.



```
Optimized parameters: R1 = 0.00435231185322067, C1 = 0.0005228984721462884
Optimized parameters: R2 = 0.00571819685140961, C2 = 0.0003979063185611875
Optimized parameters: R3 = 0.036441956477596765, C3 = 0.0011635653222379168
Optimized parameters: R4 = 0.0913478981511462, C4 = 0.01707933152550376
Optimized parameters: R5 = 0.23479283403689502, C5 = 0.0066448058196212955
Optimized parameters: R6 = 0.42933975932781776, C6 = 0.030091454057453042
```

Figure 4.10: Terminal output of the parameters converted to their foster tank RC-pairs.

Since $C_i = \frac{\tau_i}{R_i}$ and $\tau_i = \frac{1}{T_i}$ then $C_i = \frac{1}{T_i R_i}$ which gives us the pairs of resistances and capacitances as seen in figure 4.10. Conversion to cauer is then done by calling a conversion script found on github [14]. The parameters are converted into a spice model by calling the formatting function developed for this project [15].

4.2 Verifying the Generated Models

As per the method chapter verification was done by firstly checking if the model converged to the datapoints and matched the thermal impedance graph in the datasheet. The Transistors generated model converged well as seen in the section above.

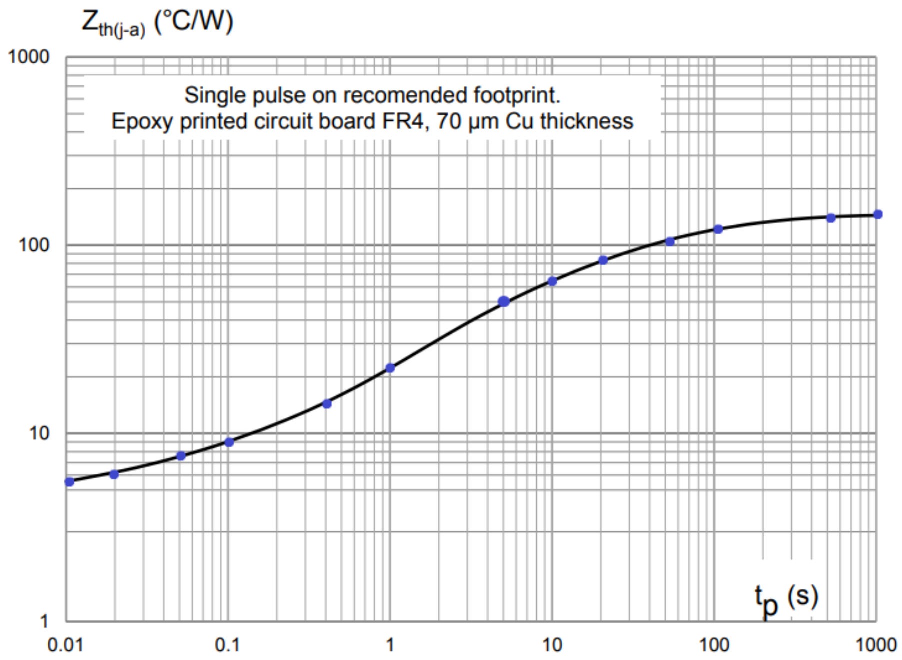


Figure 4.11: Datapoints marked out on the thermal impedance graph from the datasheet of the diode SM6TY.

The model generation procedure is repeated on a SM6TY diode as further testing. The datapoints are chosen from the thermal impedance graph the same way as for the transistor as seen in figure 4.11.

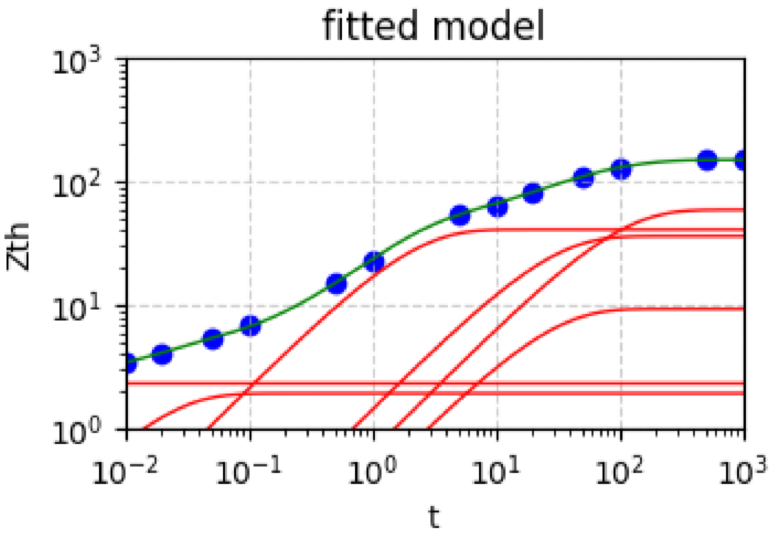
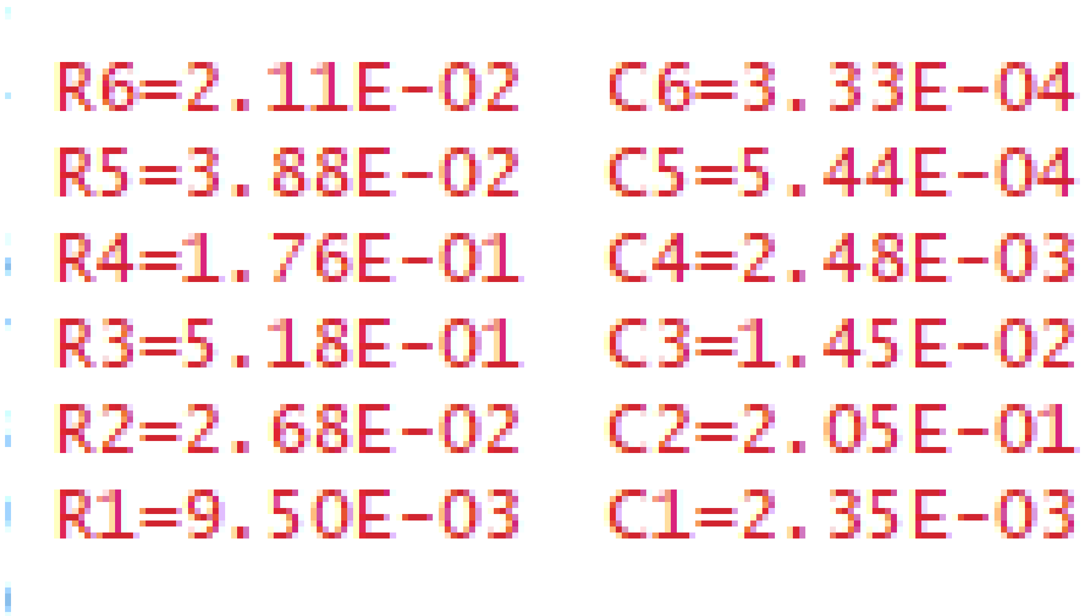


Figure 4.12: curve fit data of the diode SM6TY.

The resulting model of the diode SM6TY converged sufficiently as seen in figure 4.12. The model shows less accuracy in the convergence than the transistor which could have been prevented by a better first guess.

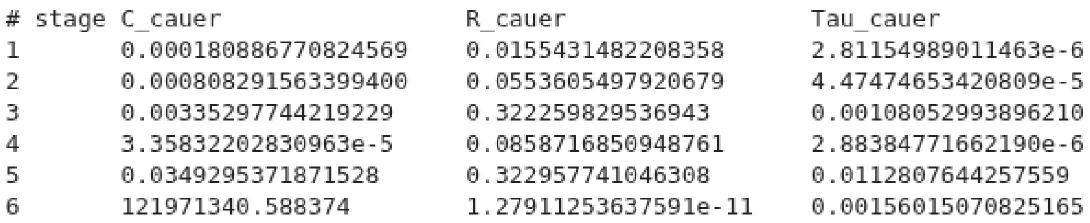
Comparison with SOAtherm

Comparison with SOAtherm was only done for the transistor.



R6=2.11E-02 C6=3.33E-04
R5=3.88E-02 C5=5.44E-04
R4=1.76E-01 C4=2.48E-03
R3=5.18E-01 C3=1.45E-02
R2=2.68E-02 C2=2.05E-01
R1=9.50E-03 C1=2.35E-03

Figure 4.13: Cauer network parameters from the SOAtherm model of the transistor PSMN3R430BLE.



```
# stage C_cauer                R_cauer                Tau_cauer
1       0.000180886770824569   0.0155431482208358     2.81154989011463e-6
2       0.000808291563399400   0.0553605497920679     4.47474653420809e-5
3       0.00335297744219229    0.322259829536943      0.00108052993896210
4       3.35832202830963e-5    0.0858716850948761     2.88384771662190e-6
5       0.0349295371871528     0.322957741046308      0.0112807644257559
6       121971340.588374       1.27911253637591e-11   0.00156015070825165
```

Figure 4.14: Cauer network parameters from the generated model of the transistor PSMN3R430BLE.

At first inspection the value of C6 in the results shown in figure 4.14 seems abnormally high compared to C6 in SOAtherm seen in figure 4.13. If the discrepancy from the SOAtherm model is further analysed the τ_6 in figure 4.14 also does not match its C_6. This is likely due to a bug in the library from github for the foster to cauer conversion [14].

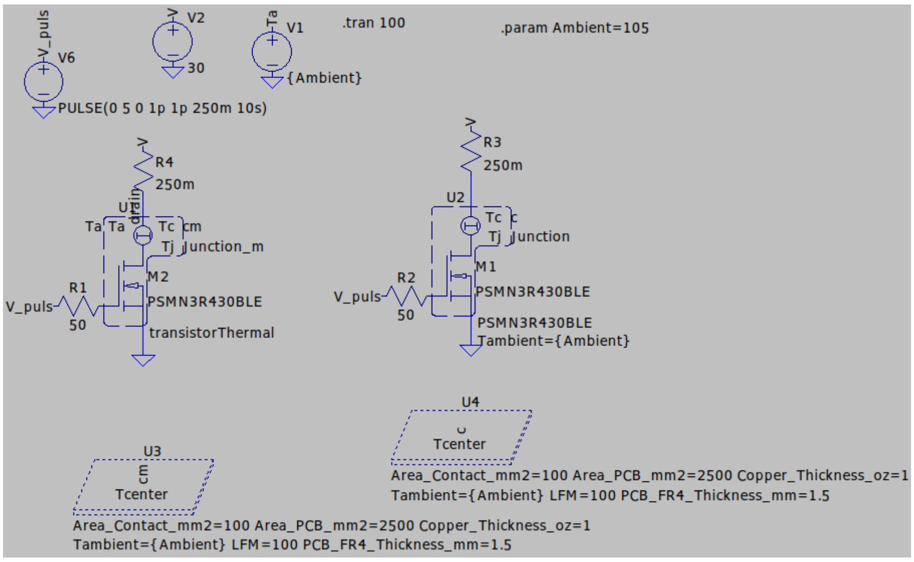


Figure 4.15: Ciruit for comparing the response between the generated spice model and SOAtherm of a signal with 5V gate-amplitude, 0.025 duty cycle, an ambient temperature of 105 °C and a peak drain current of approximately 120A.

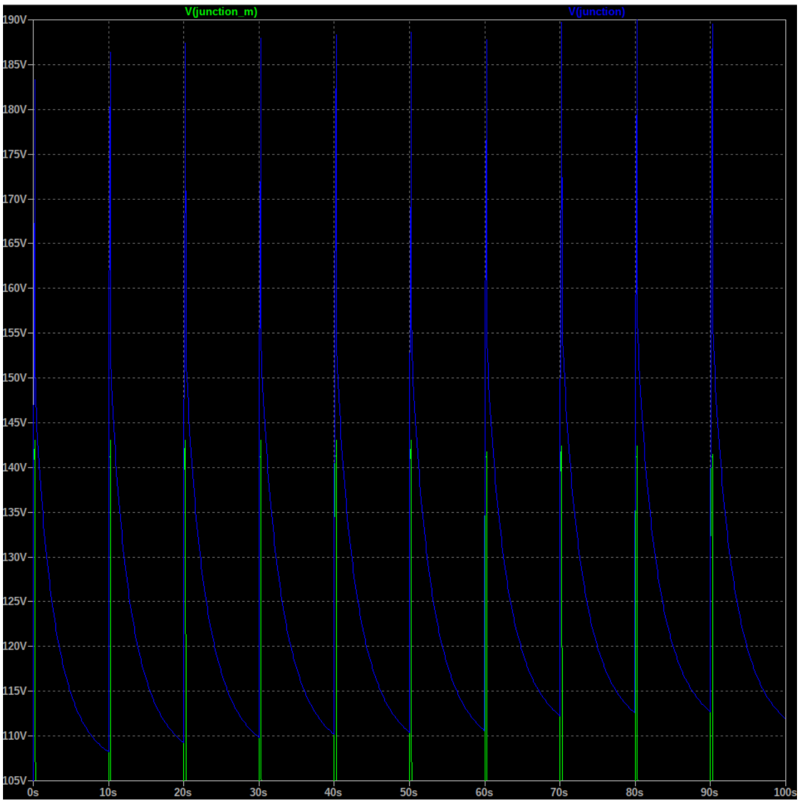


Figure 4.16: SOAtherm models response in blue and generated models response in green. Observe that the generated models signal quickly drops back to ambient temperature after the peak as opposed to the exponential decay of SOAtherms model.

The pulse-response testing was done using the circuit seen in figure 4.15. The ambient temperature was set to 105 $^{\circ}C$ which is a common temperature in automotive applications at SEM. The testing signal has a duty cycle of 0.025 to mimic test cases at SEM. A drain current of $\frac{30V}{250m\Omega} = 120A$ was chosen from the datasheets maximum ratings [12]. The model does not show a similar pulse response in figure 4.16 due to the large C_6. The model was observed to have a maximum deviation at $t \approx 80s$.

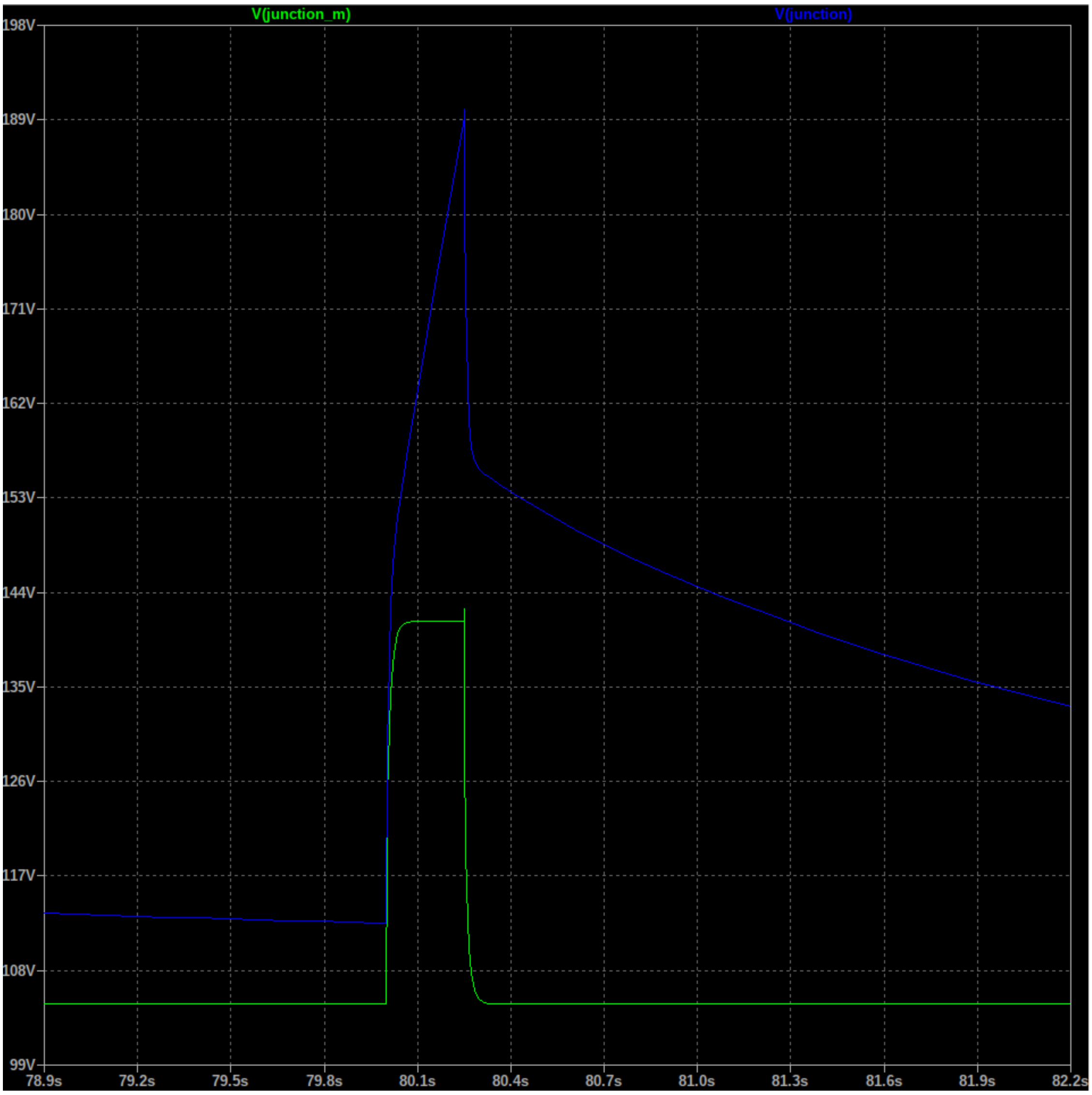


Figure 4.17: SOAtherm models response in blue and generated models response in green. Results zoomed in around the area of most deviance where it can more easily be observed that the generated model absorbs the heat energy too quickly compared to SOAtherms model.

Zooming in around $t = 80s$ as in figure 4.17 it can be observed that the model quickly returns to ambient temperature after the pulse as if it had a lot of mass to absorb the energy i.e. a large thermal capacitance. At $t \approx 80.3s$ a $\Delta V = 157V - 105V = 52V$ between the models. The total voltage span is $190V - 105V = 85V$ which gives a deviance of $\frac{52V}{85V} \approx 61\%$.

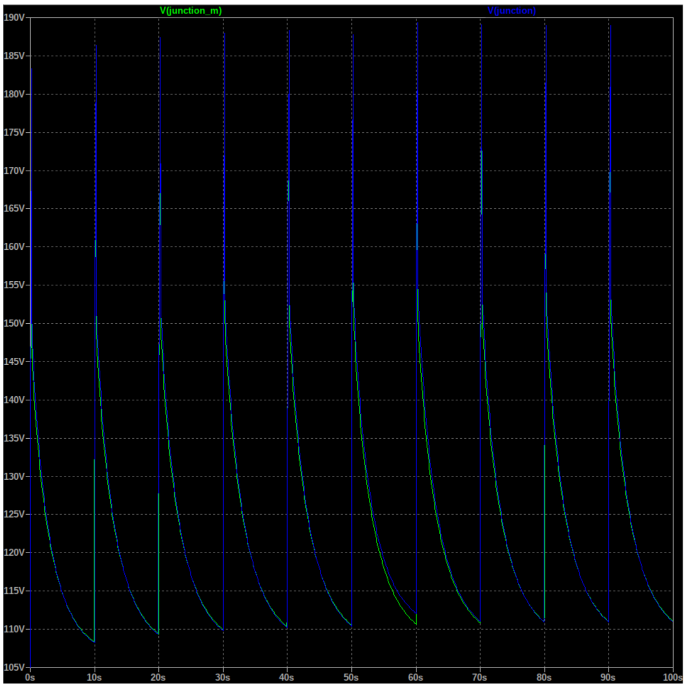


Figure 4.18: SOAtherm models response in blue and generated models response in green. Adjusted C_6 to lower the heat absorbtion and more closely match the SOAtherm model.

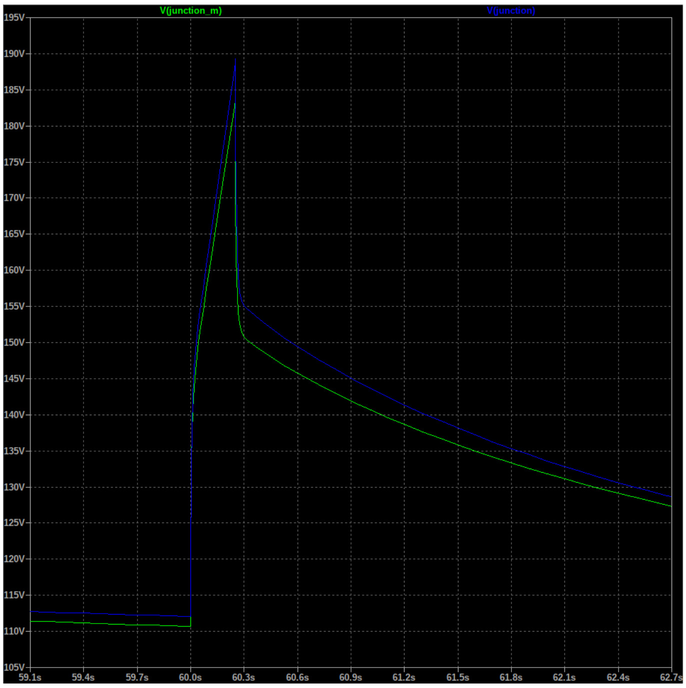


Figure 4.19: SOAtherm models response in blue and generated models response in green. Adjusted C_6. Results zoomed in around the area of most deviance. The generated model now closely follows the shape of SOAtherms model isolating the cause of deviance to be the foster to cauer conversion.

Manually adjusting C_6 to a value producing the the least amount of deviance resulted in $C_6 = 0.2$ seen in figure 4.18. The models now resemble each other and feature a similar step response. The model was observed to deviate most at $t \approx 60s$.
Zooming in at around $t = 60s$ as seen in figure 4.19 it can be observed that the voltage peak at $t \approx 60.25s$ a $\Delta V = 188V - 183V = 5V$ between the models. The total voltage span is $198V - 111V = 87V$ which gives a deviance of $\frac{5V}{87V} \approx 5.7\%$.

5

Sustainable Development

Regarding sustanable development this project is highly relevant. In order to increase the lifespan of a product and reducing electronic waste one of the key parameters to control is its operating temperature according to the Arrhenius equations. To optimize for lower temperature the best first step is to simulate the temperature and optimize the circuit with the best possible components, thermal wise, before ordering and testing them. Thermal simulations are also important to reduce waste during development due to components getting destroyed in testing.

6

Conclusion

6.1 Problems

The conversion from foster to cauer introduced errors into the component parameters most prominently C_6 that was observed to be abnormally large. This resulted in quick absorbtion of the heat inside the model which made the thermal results deceiving. The maximum deviance from the SOAtherm model was 61% but when adjusting C_6 the bit instead became 5.7%. Since the model showed conversion in the fosters mathematical model this points at the conversion to cauer having the highest effect on the result. To make this procedure viable a new program for converting foster models to cauer models needs to be developed using the same python version as the rest of the project.

The generation for a partial fraction model, foster network, of the diode SM6TY converged and the model resembled the thermal impedance graph however more testing needs to be done like with the transistor.

6.2 Future Developement

Python Scripts

More code could be hidden and abstracted away for it to be more usable by an electrical engineer. The goal to make it easy to use was accomplished to a degree but it is now also easy to break.

Guess-work

The guess-work assumed a close to linear relation in logarithmic space as the transistor almost showed. This proved a problem for the diode which had to adjust one of it's parameters afterwards in a non-intuitive way making graphs not similar to the transistor harder to model. Since the goal was to make it easy to use the guess-work part of the Jupyter Notebook could in the future provide guesses for a variety of common curves that can be displayed and chosen. Another method could be to device a method for intuitively place the exponentials under any local maxima or minima to provide local curvature to the partial fraction series.

Interface

Make interface more friendly by using deephaven input_tables in the jupyter notebook to get away from direct input to the code and making the user interface more graphical.

Datapoint Selection

Increasing the amount of data points and automating the data point collection with AI might be a possibility for the future. Increasing the amount of data points reduces the degrees of freedom and gives a more exact shape of the graph to the function.

Future Tests

This model could benefit to be tested through pulsing a physical PSMN3R4-30BLE transistor and measuring the thermal response with thermal camera and/or thermocouples connected to a logger and comparing this to the same signal pattern run through the generated spicemodel. Another future test is to design a simple dc-dc converter and build both a physical device as well as a spice model and compare the measured thermal data the same way as the previously mentioned test.

Network Synthesis of More Components

The foster to cauer conversion needs to be adjusted since it resulted in the most deviance in the results.

Future projects could explore other types of network synthesis such as Brune or Bott-Duffin to expand the usability of this method to other components. This could also be expanded towards behaviors such as temperature dependent inductance, capacitance and resistance to simulate not only the temperature but also other parameters dependent on temperature.

Bibliography

[1] G. Wu and A. Y. Wu, “A new perspective of how to understand entropy in thermodynamics,” *Phys. Educ.*, vol. 55, no. 1, p. 015 005, Nov. 2019. [Online]. Available: `https://iopscience.iop.org/article/10.1088/1361-6552/ab4de6`.

[2] P. A. Heckert, “Heat capacity,” in *Salem Press Encyclopedia of Science*. 2024.

[3] A. M. Aliyu, “On-board health monitoring of power modules in inverters driving induction motors,” Doctorate, University of Nottingham, Oct. 2016.

[4] D. W. Marquardt, “An algorithm for least-aquares estimation of nonlinear parameters,” *J. Soc. Indust. Appl. Math*, vol. 11, no. 2, Jun. 1963.

[5] P. S. Foundation, *The python tutorial*, 2025. Accessed: Apr. 30, 2025. [Online]. Available: `https://docs.python.org/3/tutorial/index.html`.

[6] D. Eddleman and G. Alonso, *Ltspice: Modeling safe operating area behavior of n-channel mosfets*, Analog Devices, Inc., 2016. Accessed: Feb. 25, 2025. [Online]. Available: `https://www.analog.com/en/resources/technical-articles/ltspice-modeling-safe-operating-area-behavior-of-n-channel-mosfets.html`.

[7] D. Eddleman, *Ltspice: Soatherm support for pcb and heat sink thermal models*, Analog Devices, Inc., 2014. Accessed: Feb. 25, 2025. [Online]. Available: `https://www.analog.com/en/resources/technical-articles/ltspice-soatherm-support-for-pcb-and-heat-sink-thermal-models.html`.

[8] F. Stueckler, G. Noebauer, and K. Bueyuektas, *Introduction to infineon’s simulation models power mosfets*, Infineon Technologies Austria AG, 2014. Accessed: Feb. 25, 2025. [Online]. Available: `https://www.infineon.com/dgdl/Infineon-ApplicationNote_PowerMOSFET_SimulationModels-ApplicationNotes-v01_00-EN.pdf?fileId=5546d46250cc1fdf0151588db5ef1b18`.

[9] B. Hops, “Infineon technologies ag successfully acquires international rectifier,” *Business & Financial Press*, 2015. [Online]. Available: `https://www.infineon.com/cms/en/about-infineon/press/press-releases/2015/INFXX201501-020.html`.

[10] A. Velcescu, *Ian50012 - nexperia precision electrothermal models for power mosfets*, Nexperia, 2023. Accessed: Feb. 25, 2025. [Online]. Available: `https://www.nexperia.com/applications/interactive-app-notes/IAN50012_Nexperia_Precision_Electrothermal_models_for_Power_MOSFETs`.

[11] I. T. AG, *Transient thermal measurements and thermal equivalent circuit models*, 2020. Accessed: May 13, 2025. [Online]. Available: `https://www.infineon.com/dgdl/Infineon-AN2015_10_Thermal_equivalent_circuit_models-AN-v01_00-EN.pdf?fileId=db3a30431a5c32f2011aa65358394dd2`.

[12] *Datasheet*, PSMN3R4-30BLE, Nexperia, Oct. 2012. [Online]. Available: `https://assets.nexperia.com/documents/data-sheet/PSMN3R4-30BLE.pdf`.

[13] T. S. community, *Scipy.optimize curve_fit*, 2025. Accessed: May 5, 2025. [Online]. Available: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html`.

[14] T. Hara, *Fostercauer_cauerfoster*, `https://github.com/thara3/FosterCauer_CauerFoster`, 2021.

[15] R. Feltzing, *Fostercauer_cauerfoster*, `https://github.com/LordMusse/FosterCauer_CauerFoster`, 2025.

[16] B. Karlström, *Kretsanalys*, 3rd ed. Studentliteratur AB, 2022, ISBN: 9789144166001.

A

Appendix 1

A.1 Electrical Circuit Theory

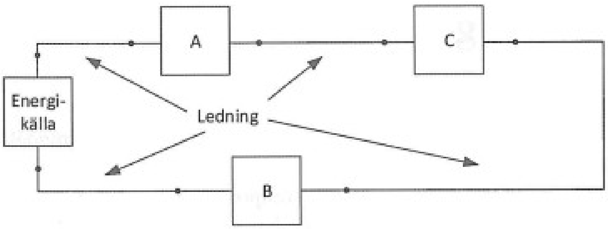


Figure A.1: Basic model of an electrical circuit [16].

Like work is energy transferred to the motion of objects electrical power is energy transferred to the motion of electrical charges like electrons. In this report electrical circuits will refer to a model of an electrical system that is built upon discrete components that are said to be ideal [16]. The circuit consists of these components connected with perfect conducting wires where one of the components is the energy source as seen in fig A.1. To be able to use this model the system needs to be much smaller than the wavelength of the signals the system is exposed to and the system must not be located inside an external electrical or magnetic field. Between and connecting the components are whats called nodes. Each point across an uninterupted wire is of the same node.

Electrical Charge

The electrical charge q is the physical quantity that is at the core of electrical phenomena and it is measured in the unit coulomb (C) [16]. This carries the energy of the system and two of the main charge carriers are electron and electron holes.

Electrical Current

Electrical current i is the net motion of charges which means it is a vector valued physical quantity with a direction and a magnitude and is defined as the amount of charges traveling between 2 nodes during a given time interval [16].

Electrical Voltage

Electrical voltage u between 2 nodes is defined as the exchange of energy that happend as one charge travels between them [16].

Electrical Potential

Electrical potential v_p of a node is defined as the electrical voltage between the node and the circuits defined reference earth where the potential energy is defined as 0 [16].

Electrical Power

Electrical power is defined as the amount of energy delivered per unit time [16].

Electrical Resistance

Electrical resistance is according to Ohms Law the rate of voltage per unit current [16]. This means that electrical resistance of an element is its ability to impede the amount of charges you can push throgh with a given energy potential.

Electrical Capacitance

Electrical capacitance is a the ability of a component to store energy in an electrical field it is defined as the amount of charges per unit voltage [16].

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS