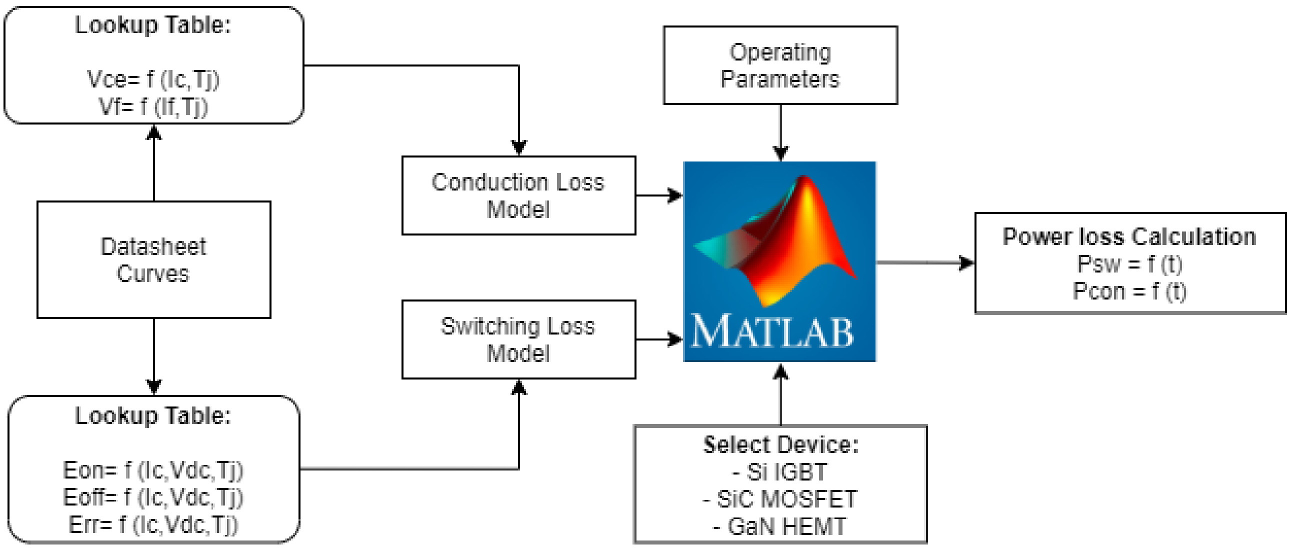
Semiconductors Power Losses in a Three-phase Inverter using MATLAB Application Designer

Master's Thesis



Group PED4 - 1042
Department of Energy Technology
Aalborg University

Written By:
Vardis Kartsonakis

AALBORG UNIVERSITY
STUDENT REPORT

School of Engineering and Science (SES)
Energy Engineering
Pontoppidanstræde 111
9220 Aalborg Ø, Denmark
http://www.ses.aau.dk/

Title:
Semiconductors Power Losses in a Three-phase Inverter using MATLAB Application Designer

10th Semester Project

Master's Thesis

Project period:

February 2021 - June 2021

Project group:

PED4 - 1042

Participants:

Vardis Kartsonakis

Supervisors:
Prof. Francesco Iannuzzo

Abstract:

As the renewable energy sources penetration is rapidly increasing, new emerging applications such as EVs, require higher efficiency. Wide Bandgap based inverters can better meet these requirements compared to conventional Si-based inverters. Higher switching applications, leads to higher power losses due to the switching behavior of the power devices. Therefore, several commercial simulation tools have been established to accurately estimate the power losses of an inverter and improve its performance. The goal of this project is to design an application capable of estimating the power losses of a three-phase, hard-switched inverter using various power semiconductor devices.
The structure of the application as well as the process of power loss calculation using the datasheet curves is described in detail for each semiconductor device. After defining the estimation procedure, the application was built using MATLAB Application Designer. It was designed to be flexible for the end user and also editable so that the user can adjust all the respective parameters.
Finally, the purpose of this project is accomplished by simulation results which are compared with other available simulation tools. Therefore, the application created is able to provide a good approximation of the power losses in a two-level inverter topology.

Circulation: Uploaded digitally
Page count: 77
Finished: May 28, 2021

By signing this document, each member of the group confirms participation on equal terms in the process of writing the project. Thus, each member of the group is responsible for the all contents in the project.

Preface

This thesis report is written by the *PED4 - 1042* group in the Department of Energy Technology at Aalborg University. The main objective of this thesis is to estimate the power dissipation of a three phase hard switched inverter based on different power semiconductor devices such as: Si IGBT, SiC MOSFET and GaN HEMT. To perform this analysis, a power loss application has been developed on MATLAB Application Designer for fast estimation of power dissipation based on curve fitting, lookup tables and the interpolation method. To validate the functionality of the application, a comparative analysis was performed with simulation tools/software from other manufacturers. Power modules from manufacturers were selected and tested with the three-phase inverter topology and the simulation results obtained were compared.

I would like to thank my supervisor Prof. Francesco Iannuzzo for his guidance and support during this semester. From our meetings, I have gained valuable knowledge and inspiration to carry out this thesis. I would also like to thank from the bottom of my heart my family, my girlfriend and friends for their support during these two years of my MSc as I could not have done it without them.

The following software has been used to:

- **Overleaf** - Write the report.
- **Mendeley** - Sort and share the bibliography.
- **MATLAB** - For general calculations.
- **MATLAB Application Designer** - Environment for building Applications in MATLAB
- **Inkscape** - Edit/Create figures.
- **draw.io** - Edit/Create figures.

Reader's Guide:

On page ix, a table of contents is given. When viewing this report as a PDF, hyperlinks in the table of contents will allow fast navigation to the desired sections.
Page vii displays a nomenclature listing the abbreviations, as well as the variables and their respective units used in this report.

The bibliography on page 41 presents the literature used in this report. The references are given in the following format:

[Author][Title](Institution)(ISBN)[Year](URL)(Date Accessed)

Where fields in square brackets are mandatory, while regular brackets are only relevant for certain formats, i.e books or web pages. The bibliography entries are sorted after their appearance in the text.

Summary

This thesis report focuses on the design of a power loss simulation tool that will be able to estimate the power losses of a three-phase hard switched inverter based on the datasheet characteristics. This simulation tool is based on MATLAB Application Designer and was designed to be flexible for the end user, since its parameters can be edited and the obtained results can be saved.

Chapter 1 provides an introduction to the power losses of a three-phase inverter, along with available manufacturer simulation tools that provide fast power loss estimation, to ensure the maximum performance of the selected topology. In addition, to the introduction the objectives along with the limitations are included based on the scope of the project.

In Chapter 2 MATLAB Application Designer is introduced. First, the main advantages of the application were described and its operation was explained with a simple example. Then, the structure of the application was analyzed in terms of power dissipation estimation.

After defining the power loss structure in Chapter 1, a thorough analysis of the power dissipation estimation for each power semiconductor device is performed in Chapter 3. Based on the power module datasheets, the IGBT conduction losses were approximated using lookup tables and the interpolation method, and similarly the switching losses were approximated using lookup tables and the curve fitting method. A similar calculation process has been carried out with the SiC MOSFET. On the other hand, the GaN HEMT datasheets do not provide an energy loss curve for the switching loss calculation. Therefore, an approximation was made based on the datasheet values provided. The above procedure was developed in MATLAB Application Designer so that the user can set the desired parameters and build the characteristics of the selected power device.

Using the procedure described above in Chapter 3, the Matlab Application was created for each of the selected devices. In order to test its functionality, a comparative analysis regarding the approximated power dissipation of 2-level inverter was performed in Chapter 4. First, a particular IGBT device from Semikron was selected. The required datasheet curves were selected and lookup tables for both IGBT and anti-parallel diode were created to obtain the approximated switching and conduction losses. Then the obtained losses were compared with those of a Simulink model, a PSIM model and with those of Semisel. It was found that the application created in M.A.D gives a good approximation of the power losses for the 2-level inverter topology.

Based on the research question stated in Chapter 1, Chapter 5 describes the achieved objectives as well as some suggestions that can be accomplished in further studies.

Nomenclature

Acronyms

AC	Alternating Current
DC	Direct Current
EV	Electric Vehicles
GaN	Gallium Nitride
HEMT	High Electron Mobility Transistor
IGBT	Insulated-Gate Bipolar Transistor
K	Second-order Polynomials
M.A.D.	MATLAB Application Designer
MAPP	MATLAB Application
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
N	Number of Samples
PF	Power Factor
rc	Reverse Conduction
RMS	Root Mean Square
Si	Silicon
SiC	Silicon Carbide
SPWM	Sinusoidal Pulse-Width-Modulation
VSI	Voltage Source Inverter
WBG	Wide Bandgap

Variables

Symbol	**Description**	**Unit**
ϕ	Power Factor Angle	[degrees]
D	Duty Cycle	[-]
E_{off}	Turn-off Energy	[mJ]
E_{on}	Turn-on Energy	[mJ]
E_{rr}	Reverse Recovery Energy	[mJ]
f_{nom}	Nominal-Frequency	[Hz]
f_{sw}	Switching-Frequency	[kHz]
I_c	Collector Current	[A]
I_{ds}	Drain-Source Current	[A]
I_d	Drain Current	[A]
I_f	Forward Current	[A]
I_m	AC Current Amplitude	[A]
I_{out}	AC Output RMS Current	[A]
I_{sd}	Source-to-Drain Current	[A]
m_a	Modulation Index	[-]
P_{Diode}^{avg}	Diode Average Losses	[W]
P_{IGBT}^{avg}	IGBT Average Losses	[W]
P_{Diode}^{con}	Diode Conduction Losses	[W]
P_{IGBT}^{con}	IGBT Conduction Losses	[W]
P_{Diode}^{sw}	Diode Switching Losses	[W]
P_{IGBT}^{sw}	IGBT Switching Losses	[W]
P_{con}	Conduction Losses	[W]
P_{sw}	Switching Losses	[W]

Q_{rr}	Reverse Recovery Charge	[nC]
Rds_{on}	On-state Resistance	[Ω]
T	Fundamental Period	[sec]
T_j	Junction Temperature	[°C]
t_{off}	turn-off time	[nsec]
t_{on}	turn-on time	[nsec]
T_{sw}	Switching Period	[msec]
Tj_{data}	Datasheet Test Junction Temperature	[°C]
V_{con}	Modulated Signal Amplitude	[-]
V_{data}	Datasheet Test Voltage	[V]
V_{dc}	DC-link Voltage	[V]
V_{ds}	Drain-Source Voltage	[V]
V_f	Forward Voltage	[V]
V_{GE}	Gate-to-Emitter Voltage	[V]
V_{GS}	Gate-to-Source Voltage	[V]
V_{out}	AC Output RMS Voltage	[V]
V_{sd}	Source-to-Drain Voltage	[V]
V_t	Carrier Signal Amplitude	[-]

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Due to environmental concerns, a continuous conversion of electrical power generation from conventional fossil fuel sources to renewable energy sources has been taking place in recent decades. As a result, the importance of power electronics is rapidly increasing. The electrical conversion process from DC to AC is achieved with the use of power semiconductor devices. The most common inverter topology uses silicon (Si) Insulated Gate Bipolar Transistors (IGBTs) and is shown in Figure 1.1:

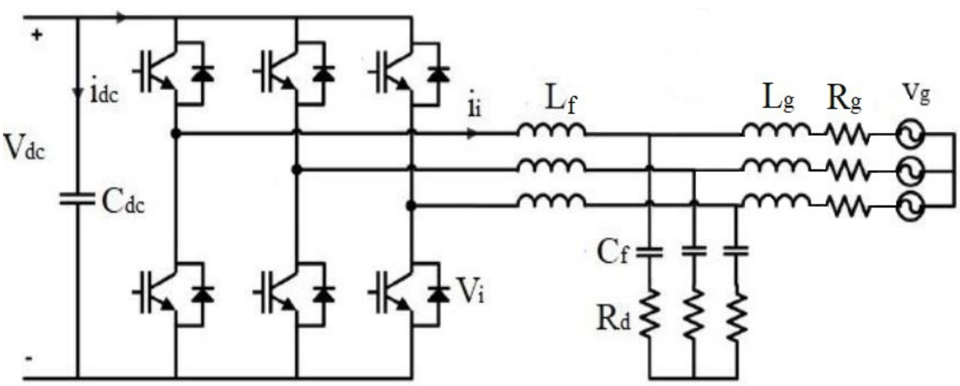


Figure 1.1: Grid-connected inverter with IGBT power switches [1].

While this type of semiconductors has proven to be very efficient over the years, the new emerging applications such as Electric Vehicles (EVs), aerospace, etc. require higher efficiency, higher operating voltage, and operation at higher switching frequencies and temperatures. These requirements can be met with the use of Wide Bandgap (WBG) power semiconductors, which can improve the overall performance of the inverter [2]. The switching behavior of the power devices generates power losses that are an essential part of the power converter structure, and in addition, the applications with higher switching frequencies will contribute to further increase the power losses [3]. As a result, applications that can estimate the power losses of the power semiconductor devices become an important factor to improve the efficiency of the inverter structure [3]. Power losses are divided into two main categories: the conduction and switching losses. The conduction losses occur when the power semiconductor is in the on-state and conducting current, and the switching losses occur during the transition between the turn-on and turn-off switching states. The described power losses can be seen more analytically in the following Figure/example 1.2 of *Mitsubishi* ^{TM} [4]:

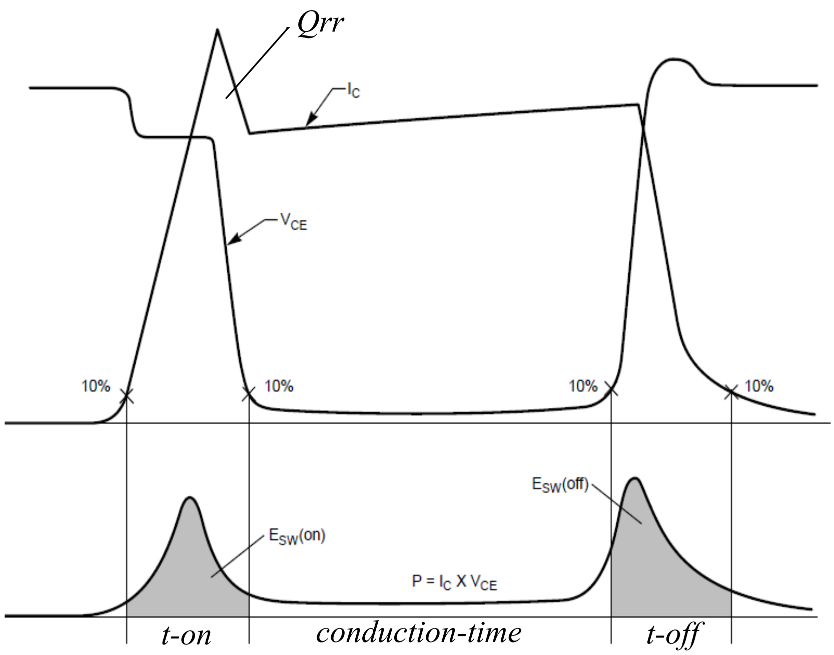


Figure 1.2: Switching Losses (Grey areas) and Conduction Losses (conduction time) of an IGBT with anti-parallel diode [4].

1.2 Silicon (Si) and Wide Bandgap (WBG) Semiconductors

In recent decades, Si semiconductors such as Insulated Gate Bipolar Transistor (IGBT) and Metal Oxide Field-Effect Transistors (MOSFET) have been widely used in both medium and low voltage applications [5]. Emerging applications such as automotive, aerospace and demand for higher power density have enabled the use of WBG power semiconductor devices. WBG devices such as Silicon Carbide (SiC), Gallium Nitride (GaN) have the main advantage of operating at a higher switching speed with lower switching losses [6]. The following table 1.1 shows the basic characteristics of Si and WBG semiconductors:

Properties	**Si**	**SiC**	**GaN**
Bandgap(eV)	1.1	3.26	3.45
Electron Mobility(cm^2/Vs)	1500	1000	1250
Hole Mobility(cm^2/Vs)	600	115	850
Saturated Drift Velocity(cm/s)	$1 \cdot 10^7$	$2 \cdot 10^7$	$2.2 \cdot 10^7$
Critical field (V/cm)	$3 \cdot 10^5$	$2.2 \cdot 10^6$	$2 \cdot 10^6$
Thermal Conductivity($W/cm - K$)	1.5	4.9	1.3

Table 1.1: WBG and Si materials properties [7].

From the above table, the most important factors in choosing between semiconductor materials are electron drift velocity, higher breakdown field and thermal conductivity. The higher breakdown field of WBG devices compared to Si shows that they can be used in high voltage applications. SiC material with higher drift velocity enables operation in higher switching frequencies and hence in high power density applications. The higher thermal conductivity also results in more efficient heat transport and thus a lower junction temperature. Last but not least, the approximately tenfold lower on-state resistance leads to higher efficiency due to reduced conduction losses [5–7].

1.3 Manufacturer Simulation Tools

There are simulation tools provided by manufacturers that allow the estimation of power losses as well as junction temperature of the power devices used in different inverter topologies. First, the user has to select a particular topology and then fill in the relevant parameters according to the particular application. According to the given parameters, suitable power modules are proposed and finally the power losses are calculated. Some of the most popular simulation tools are $SemiSel - V5^{\text{TM}}$ from Semikron, $CIPM^{\text{ TM}}$ from Infineon, IGBT simulator software from Fuji, and the UnitedSiC simulator [8–11] follows:

1.3.1 Semikron's Tool

The Semikron tool offers the advantage of being able to choose between different power converter configurations, as shown in Figure 1.3 in comparison to the Infineon and Fuji simulation tools. In addition, $SemiSel - V5^{\text{ TM}}$ takes into account the need for a heatsink to protect the power devices from overheating and improve the performance of the chosen topology, as shown in Figure 1.5 [8]. Lastly the obtained losses are analytically shown after the simulations, as in Figures 1.7, 1.8:

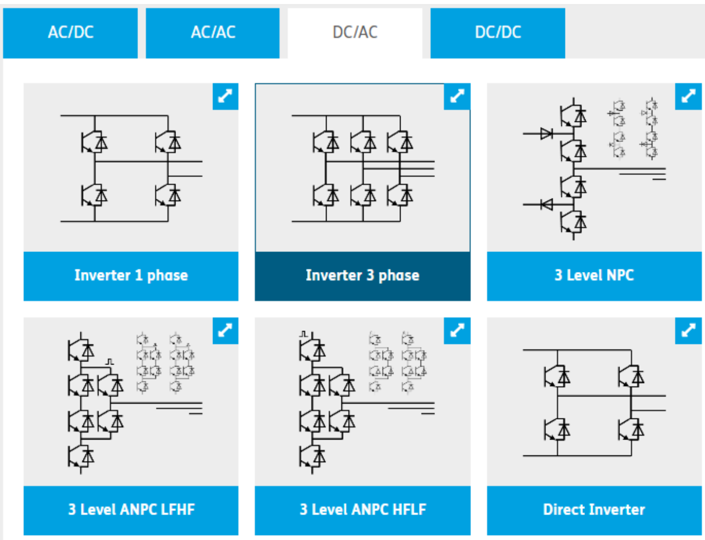


Figure 1.3: Converter Topology Selection [8].

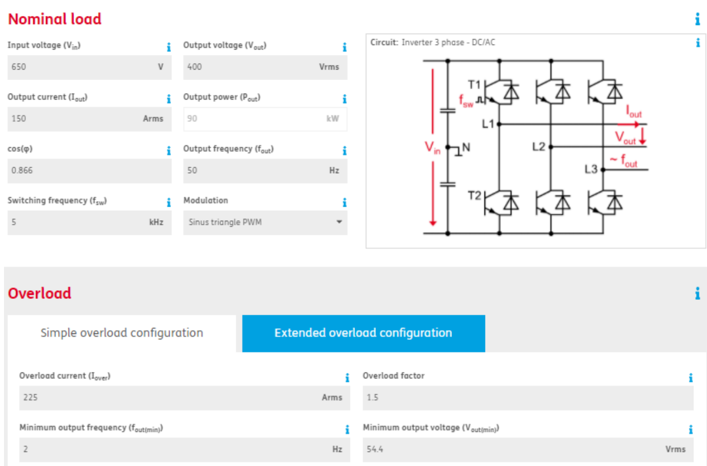


Figure 1.4: Input parameters for nominal & overload converter operation [8].

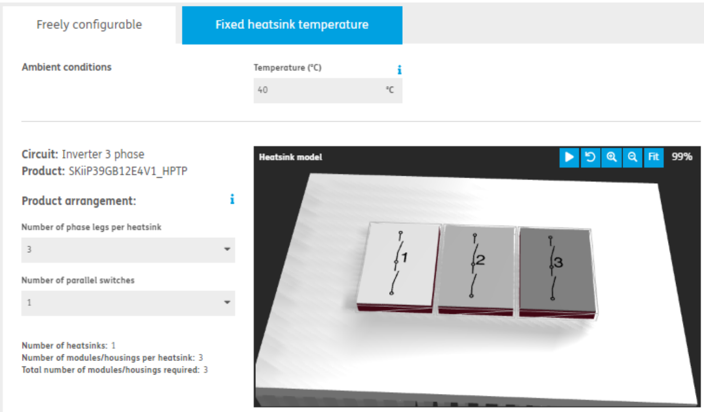


Figure 1.5: Heatsink and junction temperature selection [8].

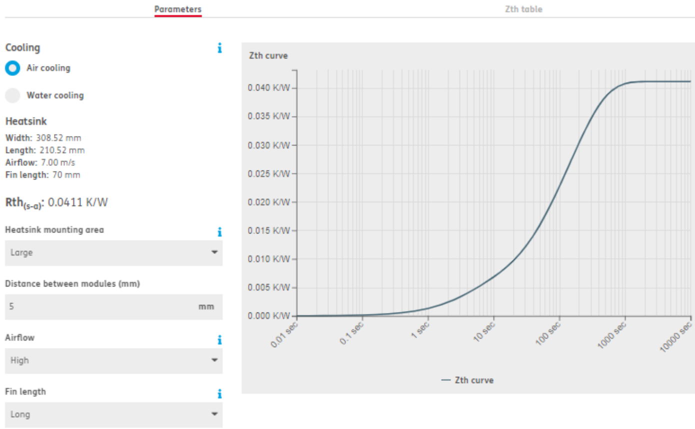


Figure 1.6: Thermal Impedance curve and heatsink characteristics [8].

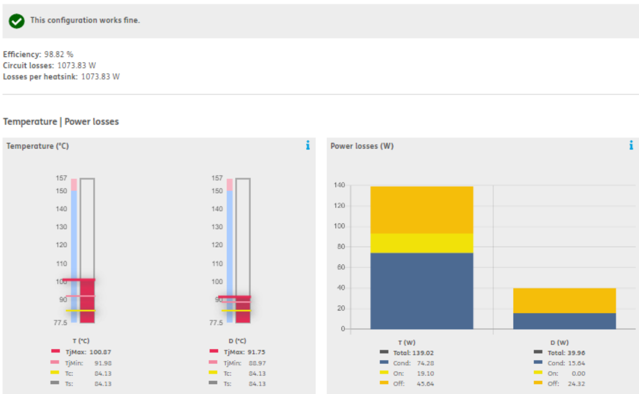


Figure 1.7: Calculated efficiency and total power losses, along with temperature variation [8].

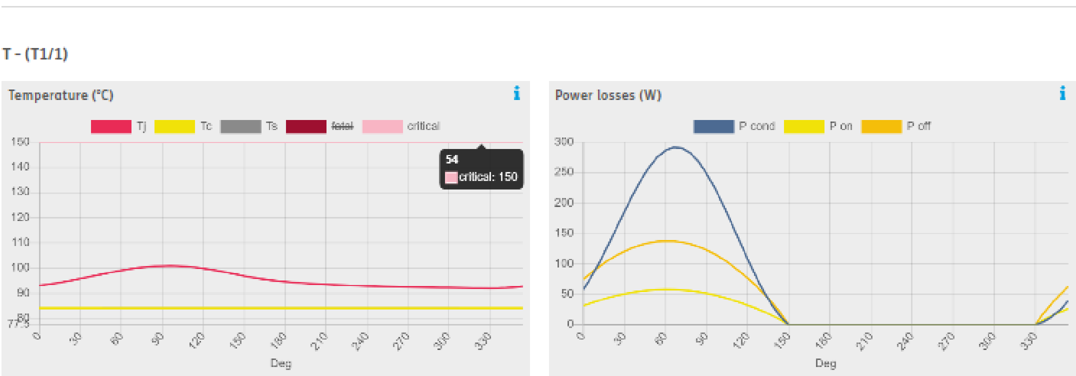


Figure 1.8: Total power losses and temperature variation for one inverter-leg [8].

1.3.2 Infineon's *CIPM*

Infineon's tool has similar characteristics to Semikron's. However, its main drawback is the limited choice of converter topologies, along with the test conditions for each one. *CIPM* ™ simulation tool is shown in the next figures [9]:

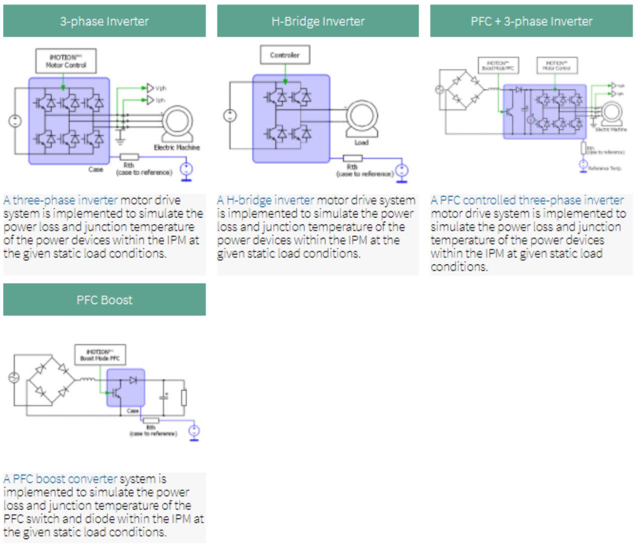


Figure 1.9: Converter application selection [9].

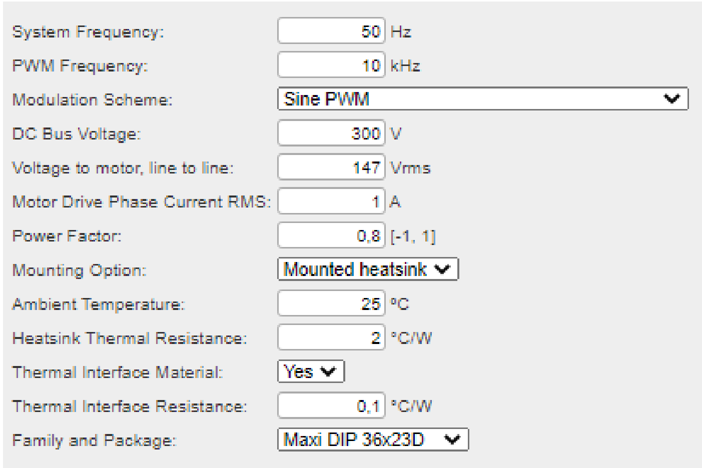


Figure 1.10: Inverter input parameters for nominal load [9].

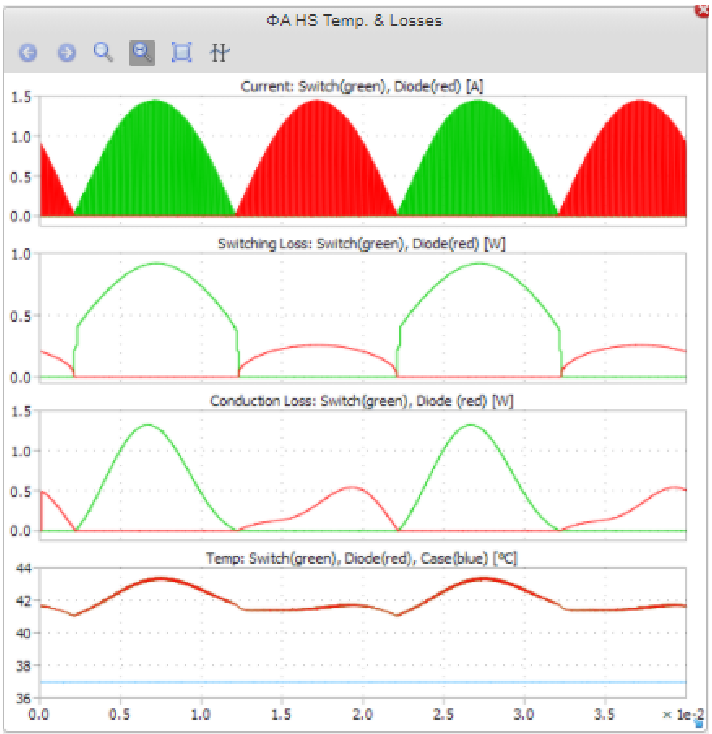


Figure 1.11: High-side power device losses [9].

Inverter Losses				
	IGBT Device	**Total**	**Efficiency**	
IGBTs	IKQ75N120CS6	4.741 W		
Diodes	IKQ75N120CS6	2.699 W		
Inverter	IKQ75N120CS6	7.440 W	97.55 %	

Phase A High Side Device Losses and Maximum Junction Temperatures				
	IGBT Device	**Switching**	**Conduction**	**Device maximum junction temperature**
IGBT	IKQ75N120CS6	0.5631 W	0.2270 W	101.0 ºC
Diode	IKQ75N120CS6	0.3830 W	0.06676 W	101.1 ºC

Phase A Low Side Device Losses				
	IGBT Device	**Switching**	**Conduction**	
IGBT	IKQ75N120CS6	0.5631 W	0.2270 W	
Diode	IKQ75N120CS6	0.3830 W	0.06676 W	

Figure 1.12: Inverter calculated power losses [9].

1.3.3 UnitedSiC Simulator

Compared to the other manufacturer tools, the UnitedSiC simulator uses only WBG devices, more specifically SiC FETs, for its simulations. It also offers a variety of topologies similar to Semikron's. Its main advantage is the fast calculation of power losses, but without providing a corresponding graph.

Input Parameters	
Power rating [VA]	600
Load power factor (-1 = rectifier, 1 = inverter)	0.5
Number of phases	3
Line to neutral RMS voltage [V]	230
DC link voltage [V]	650
Switching frequency [Hz]	8000
Input inductor ripple current [%]	0
Number of parallel FETs	3
Rdson type	Maximum
Rthjc type	Maximum
Heatsink temperature [°C]	55
Rthcs (isolator pad) [K/W]	0,6

Figure 1.13: Selected parameters on 2-level VSI topology [11].

Calculated Parameters	
Line-to-line voltage [V]	398,4
RMS current per phase [A]	0,9
Modulation factor	1
FET Rdson adjusted to temperature [Ohm]	0,0502
Losses per FET and Temperature	
Forward conduction [W]	0
Reverse conduction [W]	0
Turn-on switching [W]	0,9
Turn-off switching [W]	0,1
Total [W]	1
Junction temperature [°C]	56
Total Losses and Efficiency	
Per phase [W]	6,3
Total [W]	18,9
Semiconductor efficiency [%]	96,85

Figure 1.14: Calculated losses and parameters [11].

1.3.4 Fuji IGBT Simulator

Fuji software offers for accurate power dissipation calculations for various inverter topologies. It provides product information with the basic characteristics along with the respective topology as shown in Figure 1.15. The corresponding device datasheet can also be downloaded directly from the application. Finally, it provides an analytical loss calculation for the selected topology as well as a sweep for each parameter.

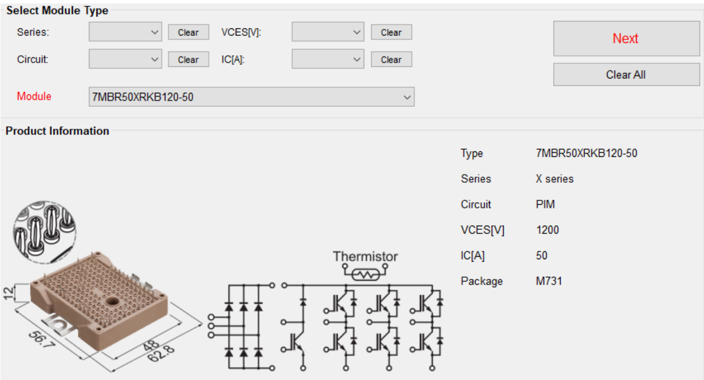


Figure 1.15: Module selection along with its main characteristics [10].

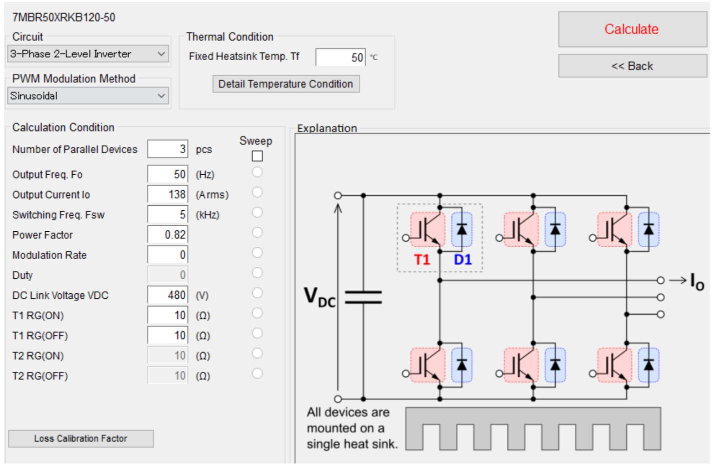


Figure 1.16: Input parameters [10].

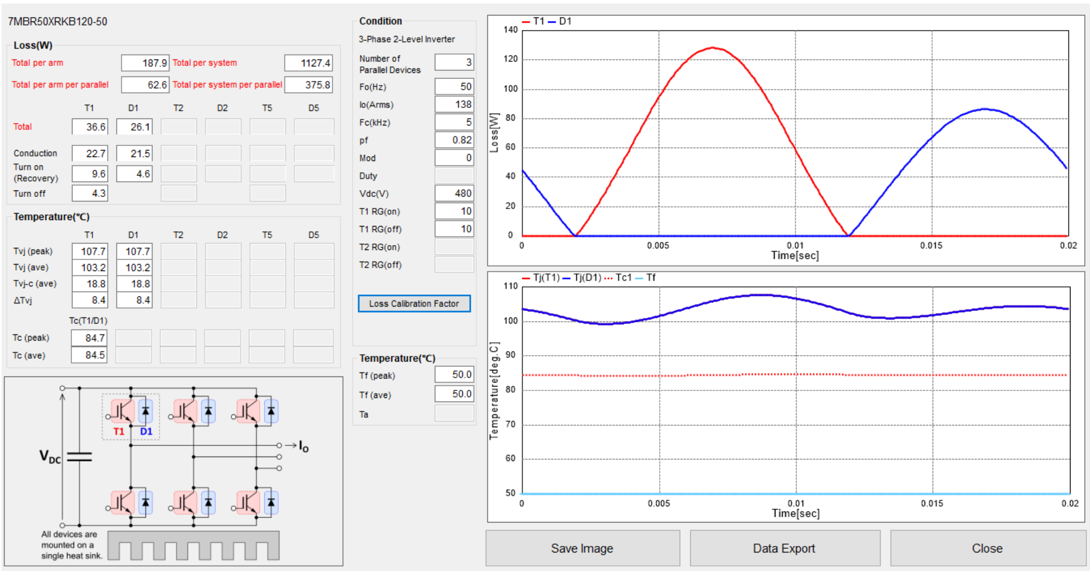


Figure 1.17: Inverter calculated losses [9].

1.4 Research Question

Based on the analysis of the above sections the following research question is proposed:

"How an application can be designed to estimate the power losses of a three-phase inverter structure using different power modules?

The analysis which will be carried out in the following chapters will be guided by this question. First, the environment of the application as well as its structure is presented. This gives an insight into how the application will be built. Secondly, the power dissipation estimation process for each semiconductor device is described and the final application is created. Finally, the designed application is validated based on a comparative analysis with other manufacturer simulation tools.

1.5 Objectives and Limitations

1.5.1 Objectives

- Define the process to calculate the power losses for Si IGBT, SiC MOSFET and GaN HEMT power devices in a three-phase hard switched inverter based on the manufacturer's datasheet curves.
- Using MATLAB application designer, build an application to simplify the calculation process. First define the application interface and then the code structure for power dissipation calculation.
- Design the application to be flexible, editable and easy to use for the end user.
- Using the created application calculate the power dissipation of a 2-level inverter and compare the obtained results with a commercially available simulation tool to test its effectiveness.

1.5.2 Limitations

- Due to time constraints, no harmonics were included in the analysis of the project.
- Determining the power dissipation using the manufacturer's datasheet curves, scaled linearly with voltage and junction temperature, gives approximate and not accurate results.
- Although Matlab Application Designer is a flexible program, there were some problems with the software itself. The software bugs affected the functionality of the integrated components and thus the whole simulation process.
- Due to the Covid-19 restrictions, no laboratory work was included, to validate the effectiveness of the created application, with real-time applications.
- Due to the limited commercially available simulation tools related to WBG devices, which are manufacturer-dependent and created only for specific applications, no comparative analysis was performed in this project for either GaN or SiC power devices with the created application and the commercially available simulation tools.

1.6 Thesis Outline

This thesis report is organized as follows:

Chapter 2 introduces the MATLAB Application Designer environment. It also introduces the structure of the application which shows the process required to calculate the power dissipation based on a selected power semiconductor.

Chapter 3 is focused on the analysis of the process of power dissipation calculation for each power semiconductor module. This process serves as the basis for the final design of the application.

In **Chapter 4**, the performance of the developed application is tested. First, the input parameters are selected along with a particular power semiconductor. Then, a comparative analysis is performed between the obtained results and the results of the other simulation tools to determine the accuracy of the application in terms of power dissipation calculation.

Lastly in **Chapter 5** the conclusion and some further ideas that can be implemented for future research are presented.

Chapter 2

MATLAB Application

This chapter aims to present the steps required to build an application that estimates the power losses of a three-phase inverter structure. First, the interface of MATLAB Application Designer (M.A.D.) will be introduced and then the structure of the application will be shown.

2.1 MATLAB Application Designer Framework

As mentioned in Chapter 1, power dissipation is an important factor to consider in applications with high switching frequencies, such as pulse-width modulated inverters. There are several manufacturing simulation tools available from Semikron, Infineon, Fuji, UnitedSiC etc., which estimate power dissipation for different inverter topologies with given input parameters.

MATLAB Application Designer (M.A.D.) is used to build such an application for this project. With a wide range of pre-built components and tools such as graphs, buttons, knobs, switches, etc., the process of application design and code generation is simplified [12]. More specifically, this application consists of two main tabs: the $DesignView$ tab, where the application format is defined/customized using various components, and the $CodeView$ tab, where the function of each component is defined. An example is shown in the following Figure 2.1 [12] :

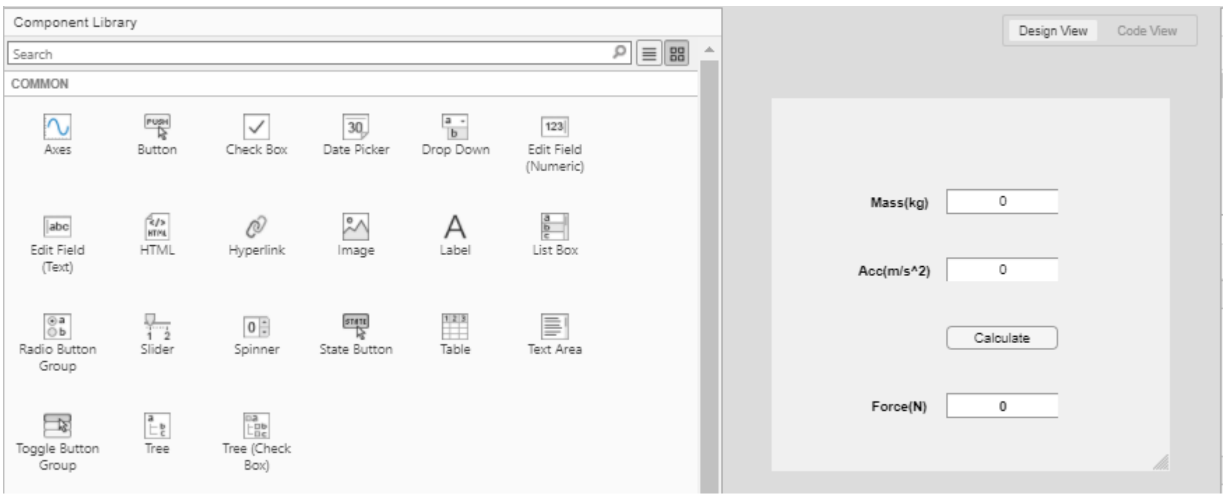


Figure 2.1: Design View Tab with four selected components from the Components Library (on the left) [12].

When a component is placed on the $DesignView$ tab, the corresponding variables are automatically created on the $CodeView$ tab. Similarly, when a component needs to be assigned a specific function, the corresponding function is automatically generated, and therefore the user only needs to develop the specific code, as in the next Figure 2.2:

Design View Code View

```
classdef b_Calculator_2 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        ForceNEditField         matlab.ui.control.NumericEditField
        ForceNEditFieldLabel    matlab.ui.control.Label
        CalculateButton         matlab.ui.control.Button
        Accms2EditField         matlab.ui.control.NumericEditField
        Accms2EditFieldLabel    matlab.ui.control.Label
        MasskgEditField         matlab.ui.control.NumericEditField
        MasskgEditFieldLabel    matlab.ui.control.Label
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: CalculateButton
        function CalculateButtonPushed(app, event)
           mass=app.MasskgEditField.Value; % value that is inside the text-box
           acc=app.Accms2EditField.Value; % similar as the above line

           % Force calculation
           Force = mass * acc;
           app.ForceNEditField.Value=Force; % we want to "GET" the calculated value

        end
    end
```

Figure 2.2: The Code View Tab in which the specific code for each component is developed. In this example a simple calculator application based on the 2^{nd} Newton's Law is created [12].

This automated process generates multiple chunks that can easily be used for multiple functions simultaneously. These features are the main advantages of M.A.D., as they simplify the code generation process for complex applications and provide an overview of the final format of the application. After these tabs are successfully configured, an application is created and can function as it supposed to. The final result of the above example can be seen in Figure 2.3 [12]:

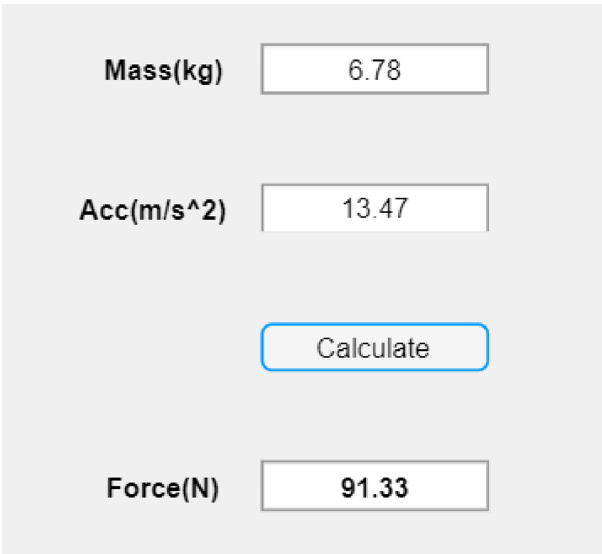


Figure 2.3: Final application output based on the defined code. According to the 2^{nd} Newton's Law $\Sigma F = m \cdot a$ and thus the calculated result proves that the application functions correctly [12].

2.2 Application Structure

In this section, the method for calculating the power losses of a power module in a three-phase voltage source inverter (VSI) configuration based on M.A.D. is described.

The method is divided into different steps so that the user is able to configure both switching and conduction losses based on the datasheet curves. First, the user is asked to enter the operating parameters of the inverter topology and then choose between three different power devices: Si IGBT, SiC MOSFET and GaN HEMT (High Electron Mobility Transistor). Specific datasheet parameters are required for each selected device to estimate the conduction and the switching losses. For example, for a selected IGBT device, the turn-on energy E_{on}, turn-off energy E_{off} and diode reverse recovery energy E_{rr} are required as a function of collector current I_c, off-state blocking voltage V_{dc} as well as the junction temperature T_j to approximate the switching losses. Similarly, the on-state voltage drop V_{ce} as a function of the collector current I_c and the diode forward voltage drop V_f as a function of the forward current I_f and the junction temperature T_j [13].

Based on the energy versus current datasheet curve, the user must select different data points to build a lookup table. This process is necessary for a second order polynomial approximation and thus for the determination of the switching energy losses. The lookup table is configured by the user based on the selected datasheet. Based on the given data, the corresponding curves are generated in M.A.D.. The obtained curves have to be compared with the original curves to get accurate approximations. Similar procedure is followed for the conduction losses, where different data points are selected from the respective datasheet curves to generate a lookup table. Then, using the interpolation method leads to an accurate approximation of the datasheet curves. Since the collector current is assumed to be sinusoidal, both switching losses and conduction losses over time are calculated, as well as the average losses [14].

The procedure described above serves as a guide for calculating the power dissipation in M.A.D., allowing the user to insert different parameters based on the selected manufacturer's datasheet. The application structure used for this project with an IGBT power module as an example is shown in Figure 2.4. The procedure for calculating the power dissipation for the other power devices follows the same structure, but with different operating parameters from the respective datasheets [13].

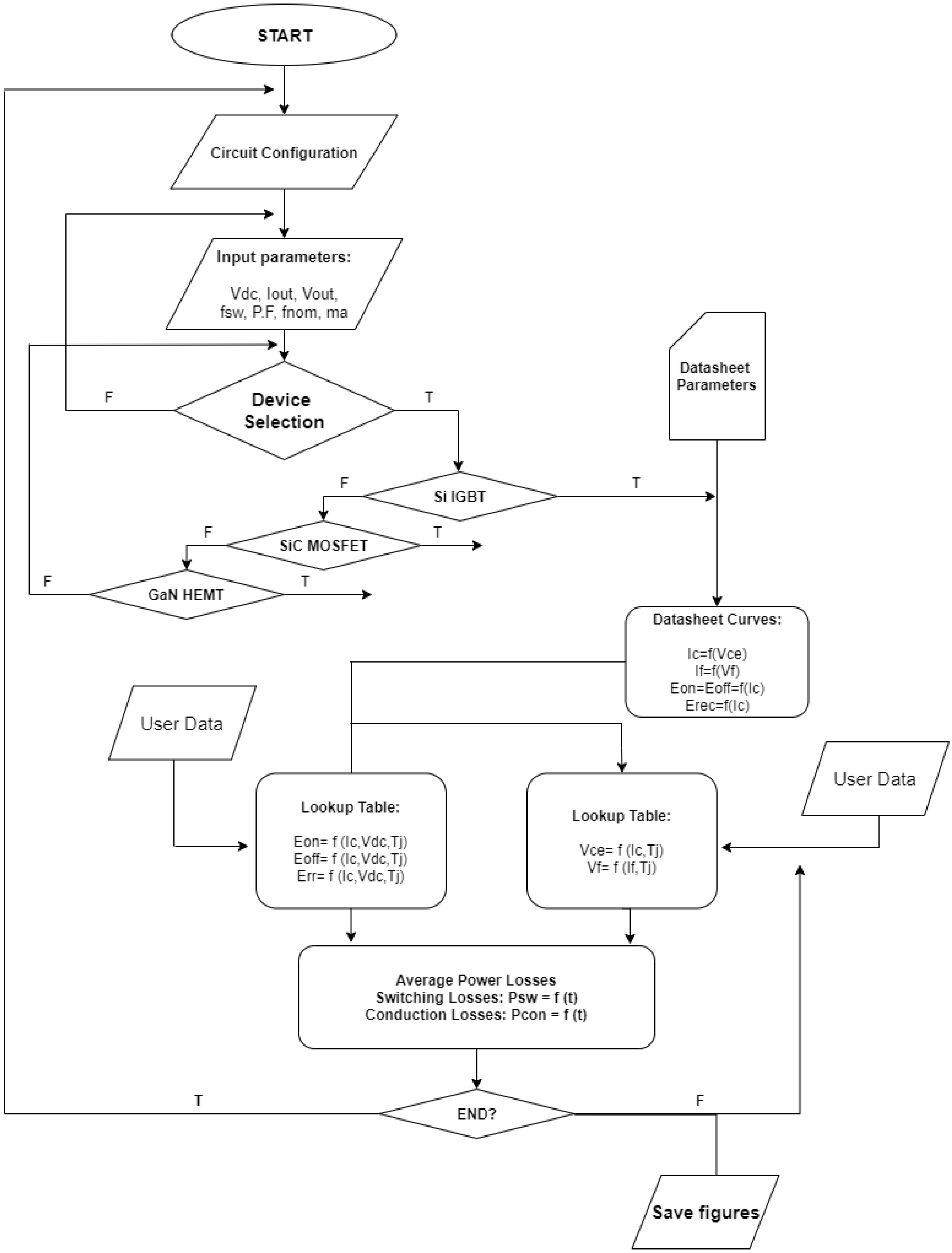


Figure 2.4: Application structure analysis for the power loss calculation [13].

Chapter 3

Application Design

In this chapter, the basic parameters for estimating power dissipation are presented. Using the datasheet parameters, the conduction and switching model for Si IGBT, SiC MOSFET and GaN HEMT is derived. Based on these derivations, the power loss calculation process is defined and the power loss application is created on MATLAB Application Designer.

3.1 Modulation Strategy

A standard sinusoidal pulse width modulation (SPWM) switching scheme is used to control the output voltage and frequency of the inverter. The generation of a sinusoidal output voltage waveform is achieved by comparing a triangular waveform and a sinusoidal control signal at the desired frequency. The amplitude of the triangular waveform (or carrier signal) (V_t) is kept constant and its frequency represents the switching frequency (f_{sw}) at which the switches of the inverter are switched. The control signal (V_{con}) has a frequency which represents the fundamental frequency of the output voltage [15]. If the amplitude of the control signal is larger compared to the carrier wave, the output is high (pulse is 1), otherwise, the output is low (pulse is 0) [15]. This procedure is shown in Figure 3.1:

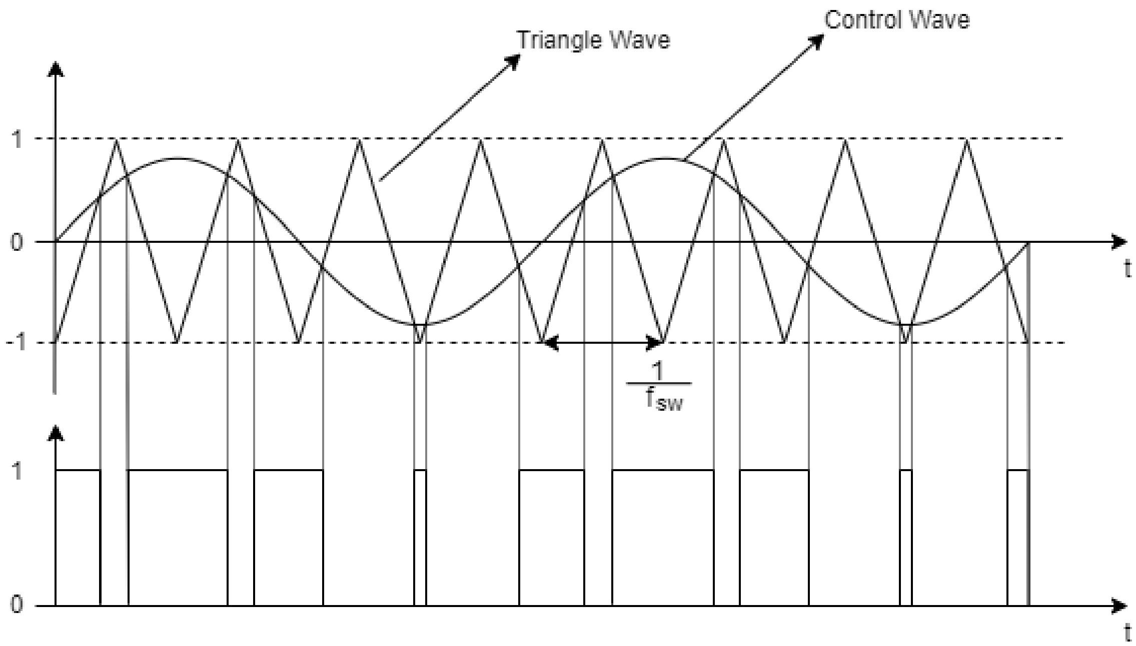


Figure 3.1: Sinusoidal Pulse Width Modulation [15].

The amplitude modulation ratio or modulation index m_a is defined as [15]:

$$m_a = \frac{\hat{V}_{con}}{\hat{V}_t} \tag{3.1}$$

Where $\hat{V}_{con}$ and $\hat{V}_t$ are the amplitude of the control signal and the amplitude of the triangular signal, respectively. In this project, a three-phase inverter is used. Therefore, three voltage control signals (shifted by 120 degrees) are compared with the triangular wave to generate three switching pulses for the three different phases.

3.2 Power loss estimation process

In this section, the three-phase inverter losses for each of the selected power semiconductor modules are analyzed as mentioned in Section 2.2. For each of the following power modules, a specific datasheet is selected to show the step-by-step process for the power loss calculation. This process is then implemented in M.A.D., in such a manner so that the user can process the data and calculate the losses of any manufacturer's module for this topology.

3.2.1 IGBT Power Module

Conduction Losses

The conduction losses are calculated during the on-state condition of the active device and conducting current. Thus, the conduction loss of an IGBT is given by Equation 3.2 and for the diode by Equation 3.3 [16]:

$$P_{igbt} = I_c \cdot V_{ce} \cdot D \tag{3.2}$$

where I_c is the collector current, V_{ce} is the collector-emitter voltage which is a function of I_c as well as a function of junction temperature T_j and D is the duty cycle.

$$P_{diode} = I_f \cdot V_f \cdot (1 - D) \tag{3.3}$$

where I_f and V_f is the forward current and forward voltage respectively. The forward voltage is a function of forward current I_f as well as a function of junction temperature T_j.

The current through the IGBT (I_c) and the diode (I_f) is assumed to be sinusoidal without any harmonics and is described by:

$$i(t) = I_m \cdot sin(2 \cdot \pi \cdot f \cdot t - \phi) \tag{3.4}$$

where I_m is the current amplitude, f is the fundamental frequency, t is a time vector and ϕ is the power factor angle.

The V_{ce} and the V_f can be approximated based on the IGBT datasheet curves ($I_c \;\; vs \;\; V_{ce}$) and ($I_f \;\; vs \;\; V_f$) respectively using data interpolation method with lookup tables [17]. The use of standard equations as in [18–20] cannot be used as they provide less accurate results, in different operating conditions [21]. For the following power dissipation calculation process, the power module FS50R12KT4 from Infineon is used [22]. From the datasheet curves ($I_c \;\; vs \;\; V_{ce}$) and ($I_f \;\; vs \;\; V_f$), it is observed that the voltages for both IGBT and diode are affected by the current as well as the junction temperature T_j [23] and must be taken into account in the respective calculation.

First, for both current and voltage, five different points are extracted from the corresponding datasheet curves and a lookup table is created as shown in Tables 3.1, 3.2.

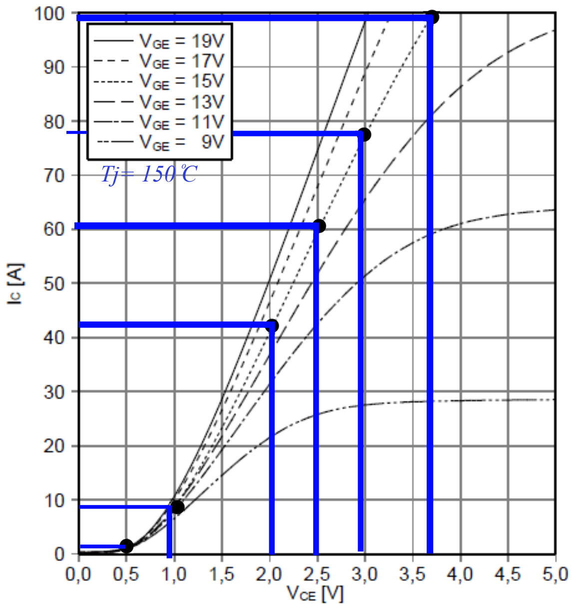


Figure 3.2: Selected data-points to determine the interpolated values of V_{ce} at junction temperature T_j=150°C and gate voltage V_{GE}=15V [22].

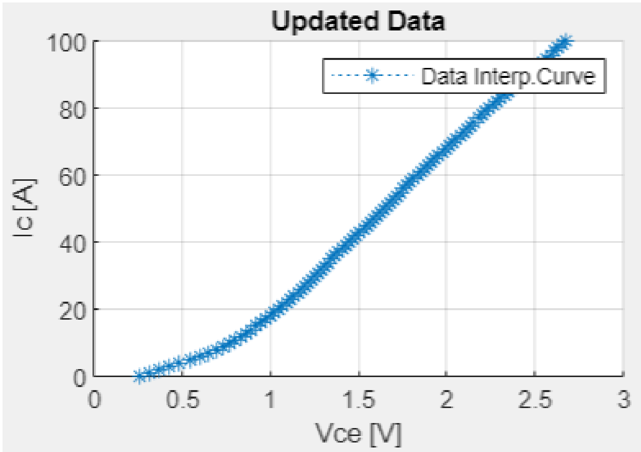


Figure 3.3: Obtained interpolated IGBT curve in M.A.D., based on the created lookup table and linearly scaled with the junction temperature.

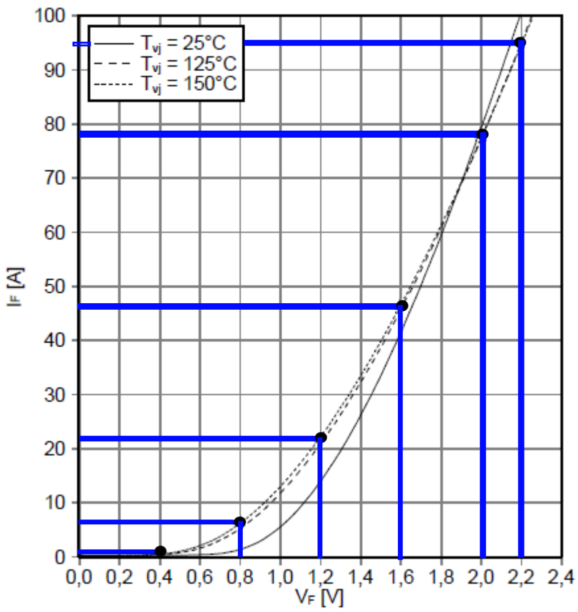


Figure 3.4: Selected data-points to determine the interpolated values of V_f at T_j=150°C [22].

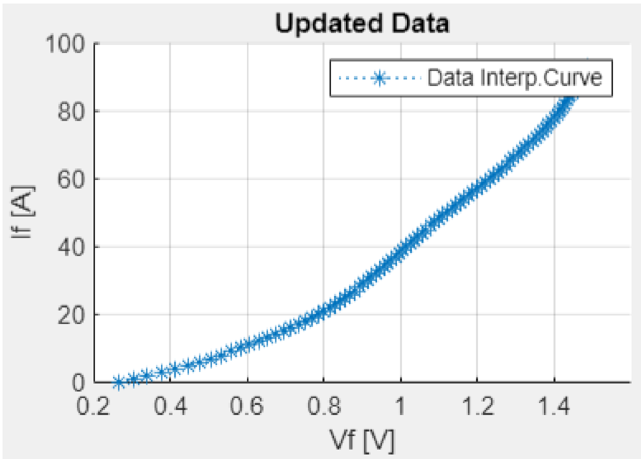


Figure 3.5: Obtained interpolated diode curve in M.A.D., based on the created lookup table and linearly scaled with the junction temperature.

Current, I_c **(A)**	**Voltage** V_{ce} **(V)**
2	0.5
9	1
41	2
60	2.5
78	3
100	3.65

Table 3.1: Selected data points based on the IGBT datasheet curve.

Current, I_f **(A)**	**Voltage** V_f **(V)**
0	0.4
8	0.8
21	1.2
45	1.6
78	2.1
93	2.23

Table 3.2: Selected data points based on the diode datasheet curve.

Using the "*interp*" command, along with the lookup tables, the polynomials for both V_{ce} and V_f are obtained for a given current range. Thus, the corresponding interpolated curves linearly scaled by the junction temperature are shown in Figure 3.4, Figure 3.3 and are described by Equations 3.5, 3.6 [17]:

$$\texttt{V\_\{ce\}} = \texttt{Tj/Tdata} \cdot \texttt{interp(Ic,Vce,Ic,"pchip",extrap);} \tag{3.5}$$

$$\texttt{V\_\{f\}} = \texttt{Tj/Tdata} \cdot \texttt{interp(If,Vf,If,"pchip",extrap);} \tag{3.6}$$

where T_j is the junction temperature and T_{data} is the datasheet specified junction temperature. I_c, I_f, V_{ce} and V_f, represent the data from the lookup table. The $'pchip'$ command, is a specific interpolation method and $'extrap'$ is the extrapolation strategy [17].

Compared to the manufacturer datasheet curves, it is observed that the generated curves show a good approximation to the original ones, but due to the linear scaling of junction temperature it is observed that there is some difference to the resulted curve.

As the current is assumed to be sinusoidal, the interpolated values of V_{ce} and the V_f need to be re-evaluated. Therefore, Equations 3.5, 3.6 are modified to the following Equations [17]:

$$\texttt{V(t)\_\{ce\}} = \texttt{Tj/Tdata} \cdot \texttt{interp(Ic,Vce,i,"pchip",extrap);} \tag{3.7}$$

$$\texttt{V(t)\_\{f\}} = \texttt{Tj/Tdata} \cdot \texttt{interp(If,Vf,i,"pchip",extrap);} \tag{3.8}$$

Based on Equations 3.4,3.7,3.8 the power losses over time are estimated as:

For the IGBT:

$$P_{IGBT}^{con}(t) = V_{ce}(t) \cdot i(t) \cdot D \tag{3.9}$$

and for the diode:

$$P_{diode}^{con}(t) = V_f(t) \cdot i(t) \cdot (1 - D) \tag{3.10}$$

The average conduction losses over one switching period are obtained from the following Equations: [18, 20]:

$$P_{igbt}^{avg} = \frac{1}{T_{sw}} \cdot \int_0^{T_{sw}} P_{igbt}^{con}(t)\, dt \tag{3.11}$$

$$P_{diode}^{avg} = \frac{1}{T_{sw}} \cdot \int_0^{T_{sw}} P_{diode}^{con}(t)\, dt \tag{3.12}$$

Switching Losses

Another type of power losses are the switching losses, that occur during the turn-on and turn-off switching transitions. For the IGBT the switching losses are calculated by Equation 3.13 [20, 24]:

$$P_{igbt}^{sw} = (E_{on} + E_{off}) \cdot f_{sw} \tag{3.13}$$

and for the anti-parallel diode by equation 3.14 [20, 24]:

$$P_{diode}^{sw} = E_{rr} \cdot f_{sw} \tag{3.14}$$

E_{on} and E_{off} are the energy losses during the turn-on and turn-off of the power device respectively, E_{rr} is the reverse recovery energy of the diode, and f_{sw} is the switching frequency. The energy losses are calculated based on a second-order polynomial approximation using the energy versus collector current curves with the worst-case junction temperature of 150°C [25]. The calculation procedure similarly to the conduction losses, starts by selecting different data points from the manufacturer datasheet curves, as shown in Table 3.3 and Figure 3.6 and Figure 3.8.

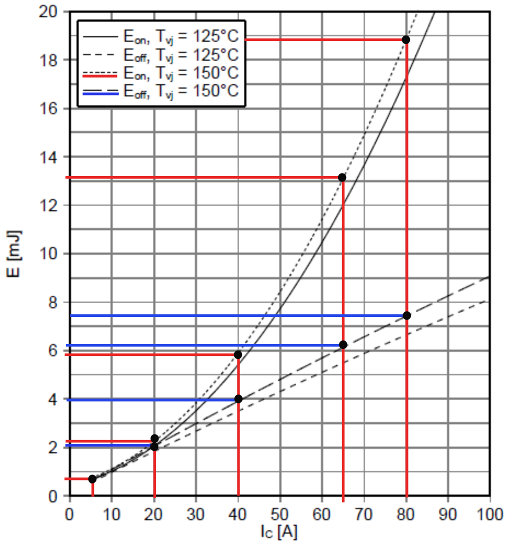


Figure 3.6: IGBT switching loss approximation based on datasheet, at junction temperature T_j=150°C and $V_{ce} = 600V$ [22].

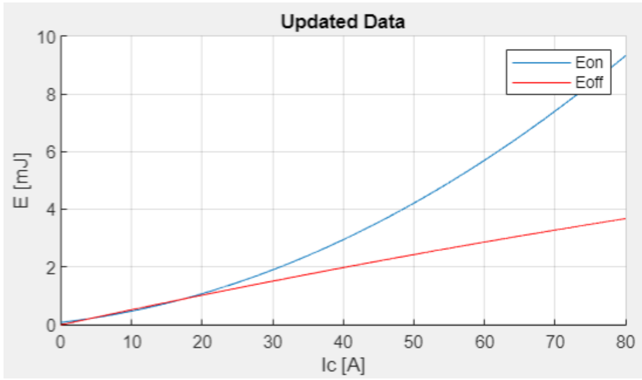


Figure 3.7: IGBT switching loss approximation on M.A.D, scaled by junction temperature and the off-state blocking voltage.

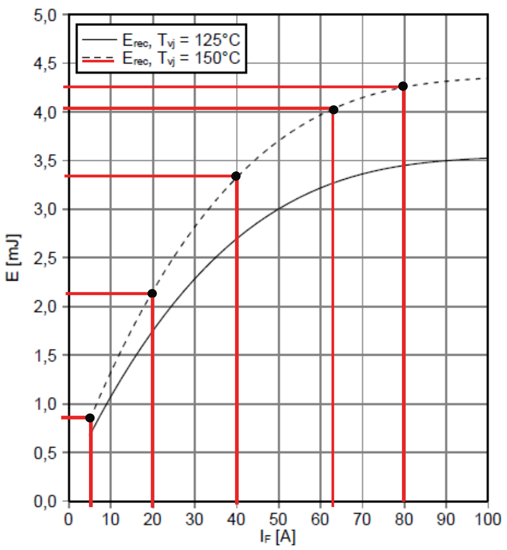


Figure 3.8: Diode switching loss approximation based on datasheet, at junction temperature T_j=150°C and $V_{ce} = 600V$ [22].

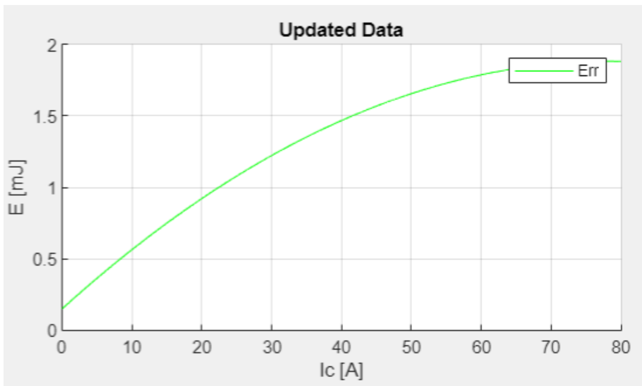


Figure 3.9: Diode switching loss approximation on M.A.D, scaled by junction temperature and the off-state blocking voltage.

Curves	Current,I_c(A)	Energy (mJ)
E_{on}	[5, 20, 40, 65, 80]	[0.5, 2.2, 5.9, 13.5, 19]
E_{off}	[5, 20, 40, 65, 80]	[0.5, 2.1, 4, 6.3, 7.5]
E_{rr}	[5, 20, 40, 65, 80]	[0.8, 2.1, 3.3, 4.1, 4.25]

Table 3.3: Selected E_{on}, E_{off}, E_{rr} points on the datasheet curve.

Using the data from the above Table and the *polyfit* function from MATLAB [26] the approximated quadratic polynomial is obtained as shown in Equation 3.15:

$$K1 = aI_c^2 + bI_c + c \tag{3.15}$$

This equation only shows the general form of the polynomial. The variables a, b, c are automatically change based on the inserted values in the lookup table. Thus, after obtaining the resulting polynomial, it must be validated to a specific range using MATLAB's *Polyval* function [27]. Since the switching energy curves are specified for a given circuit voltage and junction temperature, the switching energy curves must be linearly scaled by the ratio of the DC-link voltage V_{dc} which represents the blocking voltage during the off-state of the IGBT, and the applied circuit voltage V_{data} as specified in the datasheet [14, 28]. Accordingly, the switching curves must be scaled based on the given junction temperature, as shown in Equation 3.16.

$$\texttt{E\_\{on\},E\_\{off\},E\_\{rr\}} = \texttt{(Tj/Tdata)} \cdot \texttt{(Vdc/Vdata)} \cdot \texttt{polyval(K1, Ic)} \tag{3.16}$$

Specifying the desired DC-link voltage as well as the collector current and junction temperature, the approximated polynomial curves are plotted as shown in Figures 3.7, 3.9. To ensure that the approximation is accurate, it should be validated with respect to the datasheet curves [14]. This validation is performed by M.A.D., where the user specifies the desired DC-link voltage as well as the collector current and junction temperature. From Figures 3.7, 3.9 it is observed that they show a good approximation compared to the original curves (Figure 3.6,3.8).

As the the current through the IGBT and diode is assumed to be sinusoidal, using Equation (3.4), Equation (3.16) is modified to:

$$\texttt{E(t)\_\{on\},E(t)\_\{off\},E(t)\_\{rr\}} = \texttt{(Tj/Tdata)} \cdot \texttt{(Vdc/Vdata)} \cdot \texttt{polyval(K1, i)} \tag{3.17}$$

Therefore, the total switching losses are approximated by Equation 3.18 for the IGBT and Equation 3.19:

$$P(t)_{igbt}^{sw} = (E(t)_{on} + E(t)_{off}) \cdot f_{sw} \tag{3.18}$$

$$P(t)_{diode}^{sw} = E(t)_{rr} \cdot f_{sw} \tag{3.19}$$

As all the parameters are obtained the total average switching energy loss over one switching period is calculated as in Equation 3.20 [28]:

$$P(t)_{avg}^{sw} = \frac{1}{T_{sw}} \cdot \sum_{0}^{N-1} \cdot (E(t)_{on} + E(t)_{off}) \tag{3.20}$$

$$P(t)_{avg}^{sw} = \frac{1}{T_{sw}} \cdot \sum_{0}^{N-1} \cdot E(t)_{rr} \tag{3.21}$$

where T_{sw} is the switching period, N is number of samples per fundamental period, which can be described by $N = \frac{f_{sw}}{f}$. f_{sw} represents the switching frequency and f the fundamental frequency. For example if the $f_{sw} = 10kHz$ and $f = 50Hz$ then the number of samples per fundamental period is N=200.

3.2.2 SiC MOSFET Power Module

As Si power devices have reached their physical limits, WBG devices such as SiC MOSFET have many advantages over Si, such as higher thermal conductivity, higher voltage operation, and higher switching speed. These characteristics enable higher efficiency and power density of the power converters [5]. For the following power dissipation estimation, the SiC power module from Wolfspeed (CCS020M12CM2) with an anti-parallel Schottky diode is used [29].

Conduction Losses

Similar to the IGBT, the MOSFET conduction losses are estimated during the on-state of the power device and conducting current. Equation 3.22 estimates the conduction losses of a MOSFET, where I_d is the drain current, Rds_{on} is the on-state resistance and D is the duty cycle. Equation 3.23 estimates the conduction loss of the MOSFET's anti-parallel Schottky diode, where I_{sd} is the source-to-drain current and V_{sd} is the source-to-drain voltage [30].

$$P_{mos} = I_d^2 \cdot Rds_{on} \cdot D \tag{3.22}$$

$$P_{diode}^{mos} = I_{sd} \cdot V_{sd} \cdot (1 - D) \tag{3.23}$$

The source-to-drain voltage V_{sd} as well as the on-state resistance Rds_{on} are a function of the source-drain current I_{sd} and the drain current I_d, respectively, as well as the junction temperature T_j.

The current through the MOSFET I_d and the anti-parallel diode is assumed to be sinusoidal and is described by:

$$i(t) = I_m \cdot sin(2 \cdot \pi \cdot f \cdot t - \phi) \tag{3.24}$$

Based on the selected module datasheet, Rds_{on} can be estimated using the ($Rds_{on}\ vs\ I_{ds}$) datasheet curve, using polynomial curve fitting with a lookup table [26]. As the on-state resistance is affected by temperature it needs to be included in the respective estimation [31]. Thus, selecting different data points from the respective curve as shown in Table 3.4, Figure 3.10 and using the $polyfit$ function, the fitted quadratic polynomial is obtained similar to Equation 3.15 as :

$$K2 = aI_{ds}^2 + bI_{ds} + c \tag{3.25}$$

The resulted polynomial needs to evaluated for a given range using the $Polyval$ function [27]. As the on-state resistance is specified for a given junction temperature, the corresponding curves must be scaled according to the given junction temperature as in the following Equation:

$$\texttt{Rds-on} = \texttt{(Tj/Tdata)} \cdot \texttt{polyval(K2,Ids)} \tag{3.26}$$

To determine the voltage V_{sd} for the anti-parallel diode of the MOSFET the (I_{sd} vs V_{sd}) curve from the respective datasheet is used. For the estimation of V_{sd} same process is followed as the IGBT anti-parallel diode, using interpolation method with lookup table. The selected data points, along with the lookup table is shown in Figure 3.12 and in Table 3.5. The interpolated values of V_{sd} are obtained from the following equation:

$$\texttt{V\_\{sd\}} = \texttt{Tj/(Tdata)} \cdot \texttt{interp(Isd,Vsd,Isd,"pchip",extrap);} \tag{3.27}$$

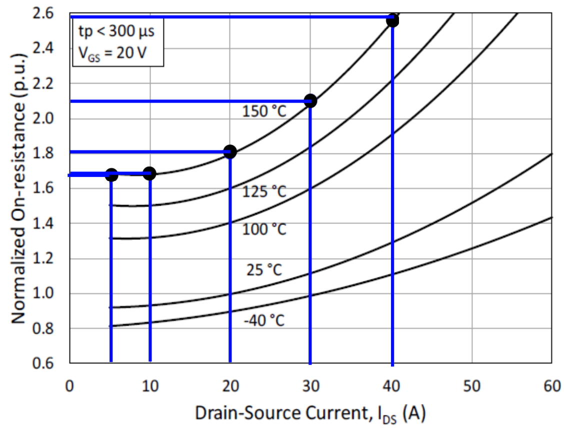


Figure 3.10: Rds_{on} approximation based on datasheet curve, at junction temperature T_j=150°C and at $V_{GS} = 20V$ [29].

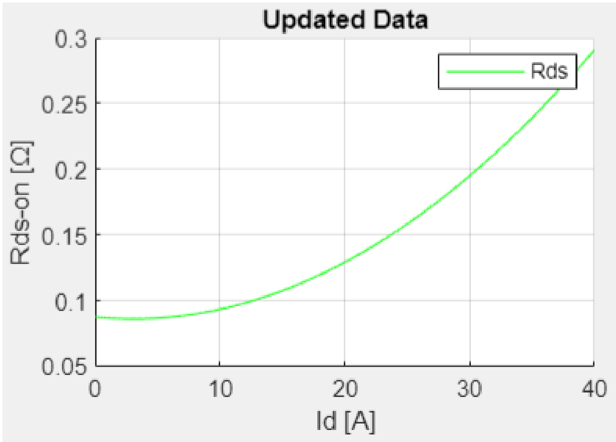


Figure 3.11: Verified Rds_{on} curve approximation, based on the created lookup table scaled by the junction temperature.

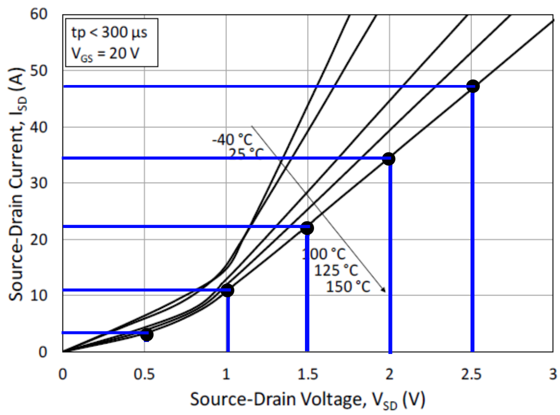


Figure 3.12: Selected data-points to determine the interpolated values of V_{sd} at junction temperature T_j=150°C and at $V_{GS} = 20V$ [29].

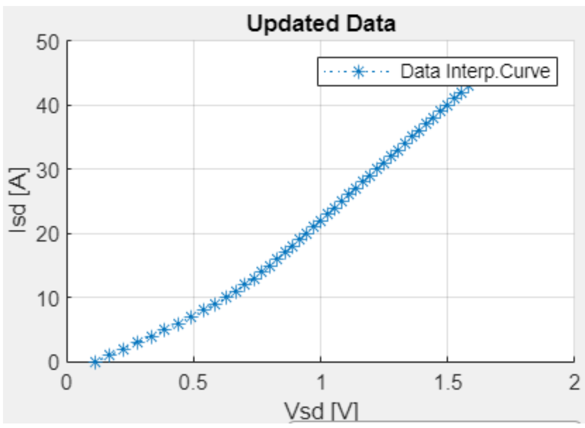


Figure 3.13: Verified Schottky Diode interpolated curve, based on the respective lookup table and scaled by the junction temperature.

Current, I_d**(A)**	$Rds-on(\Omega)$
5	0.14
10	0.142
20	0.145
30	0.35
40	0.416

Table 3.4: Selected data points based on the datasheet curve for the SiC MOSFET

Current, I_{sd}**(A)**	**Voltage** V_{sd}**(V)**
4	0.5
11	1
22	1.5
34	2
46	2.5

Table 3.5: Selected data points based on the datasheet curve for the Schottky Diode

The obtained polynomial, as well as the interpolated voltage value for a given range needs to be validated, to ensure that the approximations are accurate enough compared to the original curves. Thus the selected points, were inserted to M.A.D. and the corresponding curves are shown in Figure 3.11 and Figure 3.13. Observing the resulted curves, they both show a similar trend compared to the original ones but with some difference due to linear scaling of junction temperature as shown in Figure 3.10, 3.12.

As the current is assumed to be sinusoidal, using Equation 3.24, Equations 3.26 and 3.27 are modified to:

$$\texttt{Rds-on(t)} = \texttt{Tj/(Tdata)} \cdot \texttt{polyval(K2,i)} \tag{3.28}$$

$$\texttt{V(t)\_\{sd\}} = \texttt{Tj/(Tdata)} \cdot \texttt{interp(Isd,Vsd,i,"pchip",extrap);} \tag{3.29}$$

Based on Equations (3.24),(3.28),(3.29) the power losses over time are estimated as:

For the MOSFET:

$$P^{con}_{MOSFET}(t) = i^2(t) \cdot Rds_{on}(t) \cdot D \tag{3.30}$$

and for the Schottky diode:

$$P^{con}_{diode}(t) = V_{sd}(t) \cdot i(t) \cdot (1-D) \tag{3.31}$$

The average conduction losses over one switching period are obtained from the following Equations: [30]:

$$P^{avg}_{MOSFET} = \frac{1}{T_{sw}} \cdot \int_0^{T_{sw}} P^{con}_{MOSFET}(t)\, dt \tag{3.32}$$

$$P^{avg}_{diode} = \frac{1}{T_{sw}} \cdot \int_0^{T_{sw}} P^{con}_{diode}(t)\, dt \tag{3.33}$$

Switching Losses

The approximation of the switching losses of the MOSFET follows the same approximation procedure as for the IGBT and will not be analyzed further. Using the datasheet Energy curves (Figure 3.14), a lookup table is created based on the extracted data points from the respective Energy curves as shown in Figure 3.14 and Table 3.6.

Curves	Current,I_d(A)	Energy (mJ)
E_{on}	[10, 20, 25, 35, 40]	[0.051, 0.056, 0.065, 0.08, 0.09]
E_{off}	[10, 20, 25, 35, 40]	[0.0495, 0.048, 0.048, 0.049, 0.0505]
E_{rr}	[10, 20, 25, 35, 40]	[0.16, 0.16, 0.185, 0.21, 0.225]

Table 3.6: Selected E_{on}, E_{off}, E_{rr} points on the SiC datasheet curve.

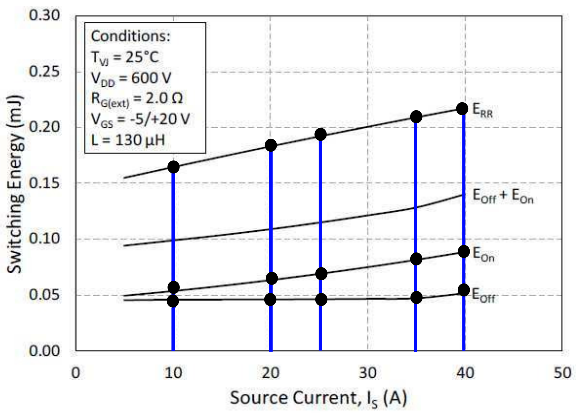


Figure 3.14: MOSFET & anti-parallel Schottky diode switching losses approximation, based on the datasheet curve at a junction temperature of 25°C and V_{DD}=600V [29].

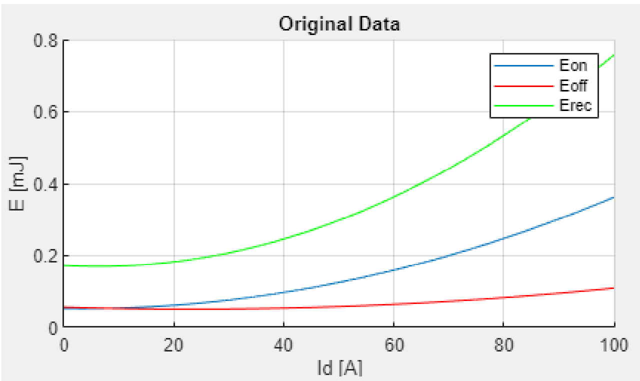


Figure 3.15: Verified switching loss approximation for both MOSFET and the anti-parallel Schottky diode in M.A.D.

Using the data from the above Table and the *polyfit* function from MATLAB [26] the approximated quadratic polynomial is obtained as shown in Equation 3.34:

$$K3 = aI_d^2 + bI_d + c \tag{3.34}$$

Evaluating the obtained polynomial at the specified current range using the *Polyval* function the switching energies can be approximated by Equation 3.35 :

$$\texttt{E\_\{on\},E\_\{off\},E\_\{rr\}} = \texttt{Tj/(Tdata)} \cdot \texttt{Vdc/(Vdata)} \cdot \texttt{polyval(K3, Id)} \tag{3.35}$$

The approximated polynomial curves are shown in Figure 3.15. Comparing the obtained switching curves, with the respective original datasheet curves shows a good approximation, with some small differences as the resulting curves are linearly scaled by the junction temperature and off-state blocking voltage. As the the current through the MOSFET and diode is assumed to be sinusoidal, using Equation 3.24, Equation 3.16 is modified to:

$$\texttt{E(t)\_\{on\},E(t)\_\{off\},E(t)\_\{rr\}} = \texttt{Tj/(Tdata)} \cdot \texttt{Vdc/(Vdata)} \cdot \texttt{polyval(K3, i)} \tag{3.36}$$

Therefore, the total switching losses are approximated by Equation 3.37 for the MOSFET and Equation 3.38 for the diode:

$$P(t)^{sw}_{MOSFET} = (E(t)_{on} + E(t)_{off}) \cdot f_{sw} \tag{3.37}$$

$$P(t)^{sw}_{MOSFET} = E(t)_{rr} \cdot f_{sw} \tag{3.38}$$

As all the parameters are obtained the total average switching energy loss over one switching period is calculated [28] as in Equation 3.39:

$$P(t)^{sw}_{avg} = \frac{1}{T_{sw}} \cdot \sum_{0}^{N-1} \cdot (E(t)_{on} + E(t)_{off}) \tag{3.39}$$

$$P(t)^{sw}_{avg} = \frac{1}{T_{sw}} \cdot \sum_{0}^{N-1} \cdot E(t)_{rr} \tag{3.40}$$

where N is the number of samples per fundamental period as defined in section 3.2.1.

3.2.3 GaN HEMT

Similar to SiC devices, GaN devices show similar characteristics in terms of high frequency operation, high thermal conductivity and high blocking voltage capability. Due to the maturity of SiC devices, along with their higher thermal conductivity, they are preferred for high power applications [5]. For the following power dissipation estimation, a GaN E-HEMT transistor from GaN-Systems is used [32].

GaN Conduction Losses

Similar to the MOSFET, the conduction losses of the GaN HEMT can be estimated from the datasheet curves. Its forward characteristic is similar to that of a MOSFET, but the reverse conduction characteristic is different because there is no anti-parallel body diode connected to the GaN HEMT. The reverse conduction path is formed by the GaN device itself [32, 33].

Forward Conduction loss:

$$P_{GaN} = I_d^2 \cdot Rds_{on} \cdot D \tag{3.41}$$

Reverse Conduction loss:

$$P^{GaN}_{reverse} = I_{sd} \cdot V_{sd} \cdot (1 - D) \tag{3.42}$$

Similar to the MOSFET, the source-to-drain voltage V_{sd} and the on-state resistance Rds_{on} are a function of the source-to-drain current I_{sd} and the drain current I_d, respectively, and the junction temperature T_j.

To determine the on-state resistance, polynomial curve fitting was performed at the maximum junction temperature specified in the manufacturer's datasheet. Thus, the points were extracted as shown in Figure 3.16, the corresponding Table 3.7 and using the *Polyfit* function, the following polynomial was obtained:

$$K4 = aI_{ds}^2 + bI_{ds} + c \tag{3.43}$$

Using the *Polyval* function the resulted polynomial was evaluated for a given current range. As the on-state resistance curve is specified for a given junction temperature at the datasheet curves, it was linearly scaled by the junction temperature as in the following Equation 3.44:

$$\texttt{Rds-on} = \texttt{Tj/(Tdata)} \cdot \texttt{polyval(K4,Ids)} \tag{3.44}$$

To determine the V_{sd} for the reverse conduction of the GaN the (I_{sd} vs V_{sd}) curve from the respective datasheet is used. As the gate voltage greatly affect the GaN reverse conduction characteristic the $V_{GE} = 6V$ curve was selected to obtained the highest operating efficiency [32], at the highest junction temperature. Similarly, to the IGBT and MOSFET the interpolation method is used with a lookup table. The selected data points, along with the lookup table is shown in Figure 3.18 and in Table 3.8. The interpolated values of V_{sd} are obtained from the following equation:

$$\texttt{V\_\{sd\}} = \texttt{Tj/(Tdata)} \cdot \texttt{interp(Isd,Vsd,Isd,"pchip",extrap);} \tag{3.45}$$

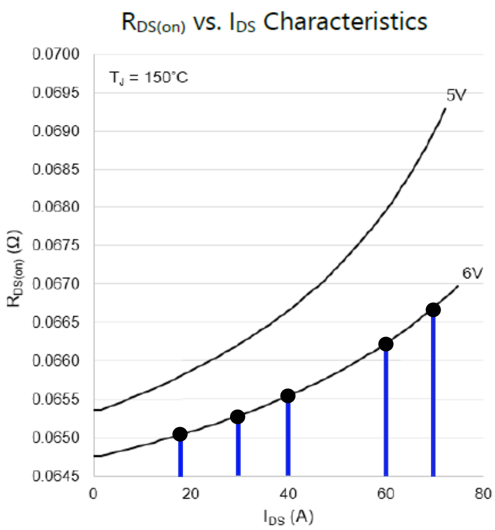


Figure 3.16: Rds_{on} approximation based on datasheet curve, at T_j=150C and $V_{GE} = 6V$ [32].

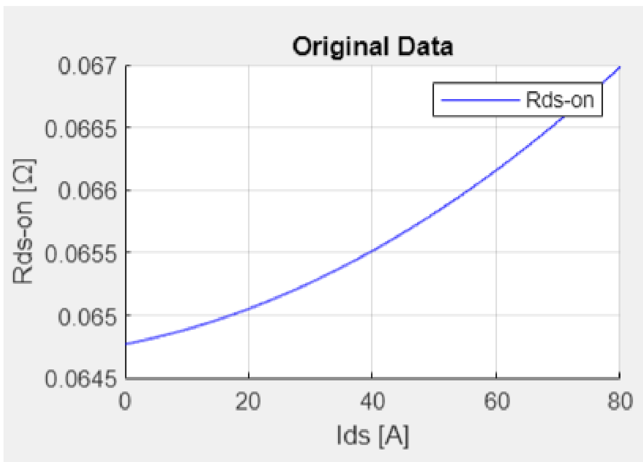


Figure 3.17: Verified Rds_{on} curve approximation, based on the created lookup table scaled by the junction temperature.

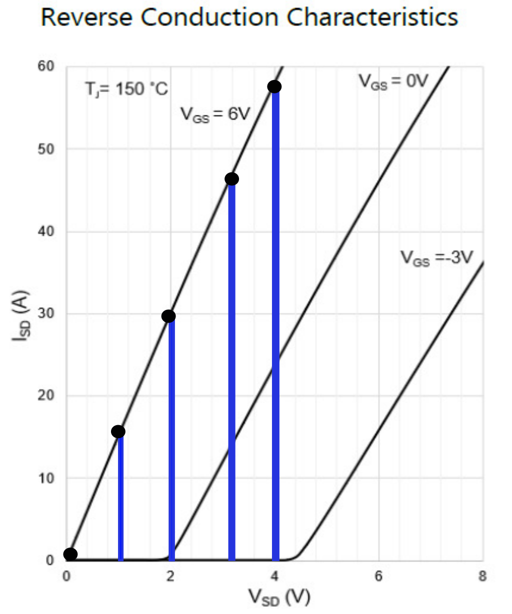


Figure 3.18: Reverse conduction approximation based on the datasheet curve, at T_j=150C and $V_{GE} = 6V$ [32]

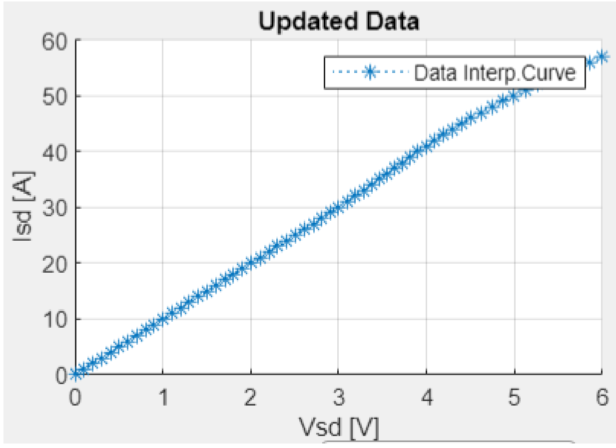


Figure 3.19: Verified reverse condcuction interpolated curve, based on the respective lookup table and scaled by the junction temperature

Current, I_{ds} **(A)**	$Rds - on(\Omega)$
18	0.065
30	0.0653
40	0.0655
60	0.0661
78	0.669

Table 3.7: Selected data points based on the datasheet curve.

Current, I_{sd} **(A)**	**Voltage** V_{sd} **(V)**
0	0
15	1
30	2
46	3
57	4

Table 3.8: Selected data points based on the datasheet curve for the reverse conduction

Comparing both generated curves (see Figures 3.17, 3.19) with the original ones, it can be observed that the used approximation is accurate enough.The small differences observed are due to the linear scaling that was used.

As the current through the GaN is assumed to be sinusoidal, using Equation 3.24, Equations 3.44 and 3.45 are modified to:

$$\texttt{Rds-on(t)} = \texttt{Tj/(Tdata)} \cdot \texttt{polyval(K4,i)} \tag{3.46}$$

$$\texttt{V(t)\_\{sd\}} = \texttt{Tj/(Tdata)} \cdot \texttt{interp(Isd,Vsd,i,"pchip",extrap);} \tag{3.47}$$

Thus using Equations 3.24, 3.46, 3.47, the power losses are estimated as:

For the GaN:

$$P_{GaN}^{con}(t) = i^2(t) \cdot Rds_{on}(t) \cdot D \tag{3.48}$$

and for the reverse conduction:

$$P_{rc}^{con}(t) = V_{sd}(t) \cdot i(t) \cdot (1 - D) \tag{3.49}$$

The average conduction losses over one switching period are obtained from the following Equations [30]:

$$P_{GaN}^{avg} = \frac{1}{T_{sw}} \cdot \int_0^{T_{sw}} P_{GaN}^{con}(t)\, dt \tag{3.50}$$

$$P_{rc}^{avg} = \frac{1}{T_{sw}} \cdot \int_0^{T_{sw}} P_{rc}^{con}(t)\, dt \tag{3.51}$$

GaN Switching Losses

As the manufacturer datasheets do not provide, Energy versus drain current curves the switching energy approximation, is based on a linear relationship [28] as shown in Equation 3.52. It should be noted, that since the GaN does not have a body diode there is no reverse recovery energy loss [32]:

$$\texttt{E(t)\_\{on\},E(t)\_\{off\}} = \texttt{E\_\{data\}} \cdot \texttt{(Tj/Tdata)} \cdot \texttt{(Vdc/Vdata)} \cdot \texttt{(i/Idata)} \tag{3.52}$$

Where E_{data} is the specified datasheet turn-on Energy or turn-off energy value and I_{data} is the specified test current. The total switching losses are approximated by Equation 3.53:

$$P(t)_{GaN}^{sw} = (E(t)_{on} + E(t)_{off}) \cdot f_{sw} \tag{3.53}$$

As all the parameters are obtained the total average switching energy loss over one switching period is calculated [28] as in Equation 3.54:

$$P(t)_{avg}^{sw} = \frac{1}{T_{sw}} \cdot \sum_{0}^{N-1} \cdot (E(t)_{on} + E(t)_{off}) \tag{3.54}$$

where N is the number of samples per fundamental period as defined in section 3.2.1.

3.3 Final Application

After defining the process of power dissipation calculation for each of the power semiconductors in the previous sections and following the application structure defined in Chapter 2, the final application was created in MATLAB Application Designer, as shown in the Appendix B. In Figure B.1, the "Welcome tab" gives an overview of the steps that need to be taken to determine the power dissipation. In Figure B.2, the "Circuit Configuration Tab", the user is asked to enter the operating parameters of the inverter and in Figure B.3 the "Device Selection Tab", the user is asked to select the appropriate power device for the simulations. In the next two steps shown in the Figures B.4, B.5, the user manually creates the corresponding switching and conduction models based on the datasheet curves. By extracting some points from the manufacturer's datasheet curves, the user manually creates a lookup table that provides the corresponding curves, linearly scaled by temperature and voltage. Finally, the corresponding loss curves for both the transistor and the diode over time are obtained in the "Losses Tab", as shown in Figure B.6. Last but not least, the user has the possibility to save the resulting curves or to change the input parameters and observe how the power losses change.

3.4 Summary

Having analyzed the power loss calculation process, for the IGBT module, MOSFET module and the GaN HEMT based on the datasheet curves, in Sections 3.2.1, 3.2.2, 3.2.3 respectively, the final application was designed in MATLAB Application Designer and its functionality will be validated in Chapter 4.

Chapter 4

Results

After defining the application structure in Chapter 2 and calculating the power dissipation in Chapter 3, this chapter provides a comparative analysis of the power dissipation of three-phase IGBT-based inverter obtained using the created application in MATLAB and simulation tools from other manufacturers.

4.1 Si IGBT power-module

In order to verify the performance of the created simulation tool and the methodology used to estimate the power dissipation, a 2-level Si-based IGBT inverter, made with SEMIX151GD066HD from Semikron [34] was used. In [14], the inverter is used in Electric Vehicles to drive the electric motor. The operating parameters used are given in the following Table 4.1:

Parameter	Symbol	Value	Unit
DC-link Voltage	V_{dc}	450	V
Output Current	I_{rms}	138	A
Power Factor	PF	0.82	-
Carrier Frequency	f_{sw}	8	kHz
Modulation Index	m_a	0.85	-

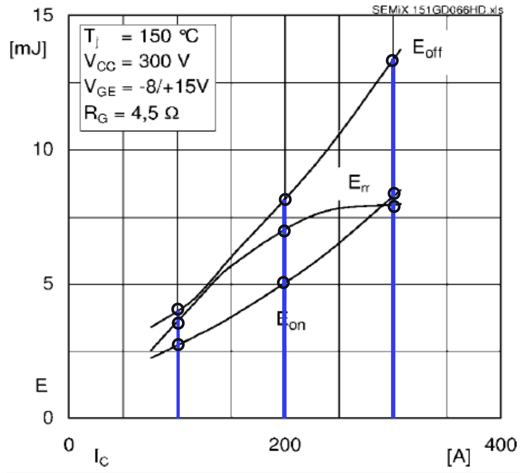
Table 4.1: Operating Parameters of the three-phase inverter [14, 35]

First, these parameters are inserted in M.A.D. and the switching and conduction curves are generated manually using the given datasheet. For the switching curves, a lookup table is created by selecting different points on the energy curve from the respective datasheet as shown in Figure 4.1b. The extracted points are shown in Table 4.2.

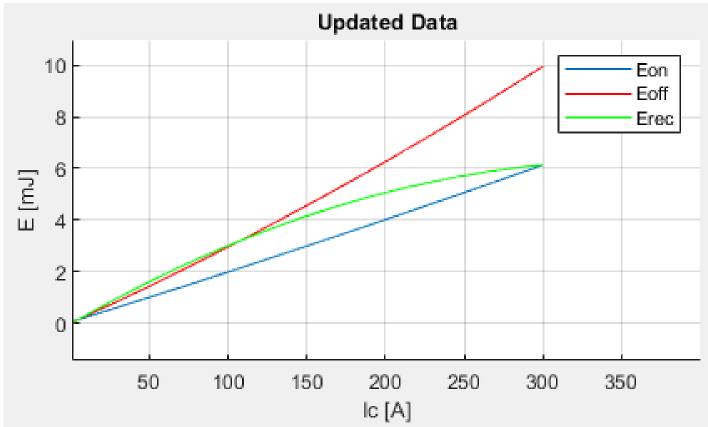
Curves	Current,I_c(A)	Energy (mJ)
E_{on}	[100,200,300]	[3, 5, 8.3]
E_{off}	[100,200,300]	[4, 8.3, 13.3]
E_{rr}	[100,200,300]	[3.75, 7, 8.1]

Table 4.2: Lookup table based on SEMIX151GD066HD Energy Curves.

Taking into consideration the off-state blocking voltage (V_{data}=300 V), the specified junction temperature (T_{data}=150 C °) and using second-order polynomial approximation as in Equation 3.17 yields:



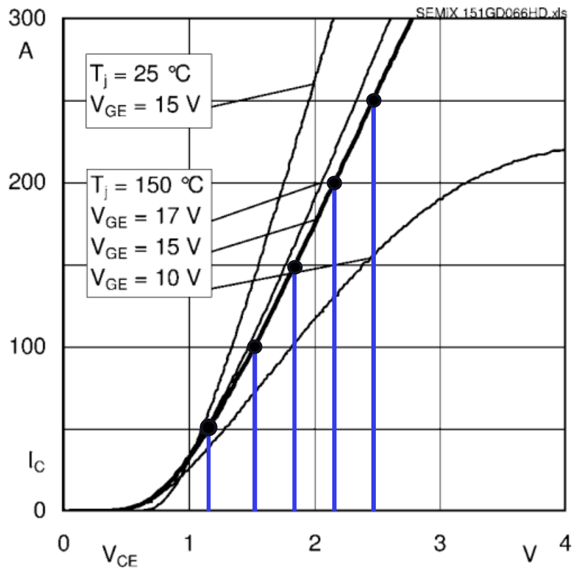
(a) IGBT & Diode switching loss approximation, based on the datasheet curves at junction temperature of 150°C and V_{cc}=300V [34]



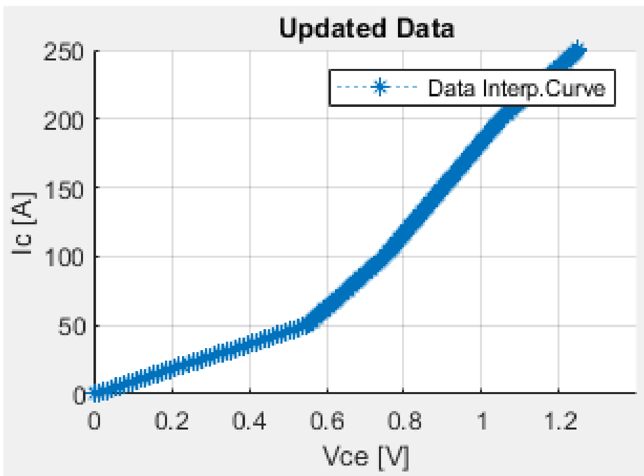
(b) Generated turn-on and turn-off Energy curves on M.A.D. based on a lookup-table scaled by the specified off-state blocking voltage

Figure 4.1: IGBT Switching model build-up.

Comparing Figure 4.1a and Figure 4.1b, it is observed that the approximated curves scaled by the junction temperature and the off-state blocking voltage are accurate enough compared to the datasheet ones. The next step is to create the conduction curves for both IGBT & anti-parallel diode in the app, using the interpolation method with lookup tables, as described in Section 3.2.1. For both IGBT & diode, the worst case temperature of 150°C is selected. So, different data points are selected from the given datasheet curves (I_c vs V_{ce}) and (I_f vs V_f) curve as shown in Figure 4.2b and Figure 4.3b. The extracted data points are given in Tables 4.3, 4.4. Using the extracted points and applying MATLAB's interpolation method, the following curves are obtained for both IGBT & diode.

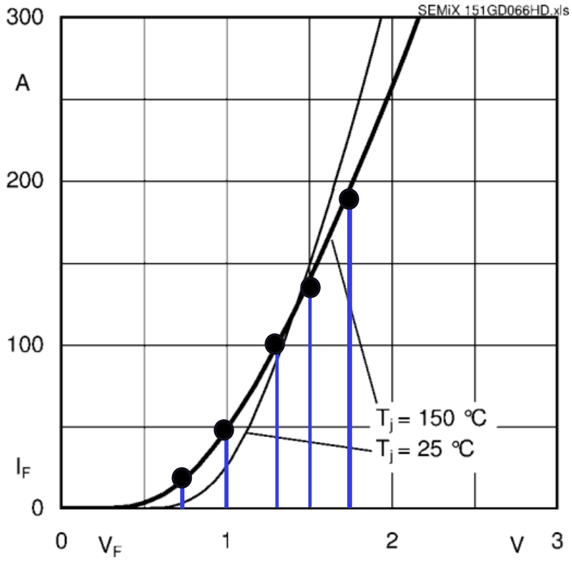


(a) IGBT conduction loss approximation, using the worst-case junction temperature and gate voltage of $V_{GE} = 15V$ [34]

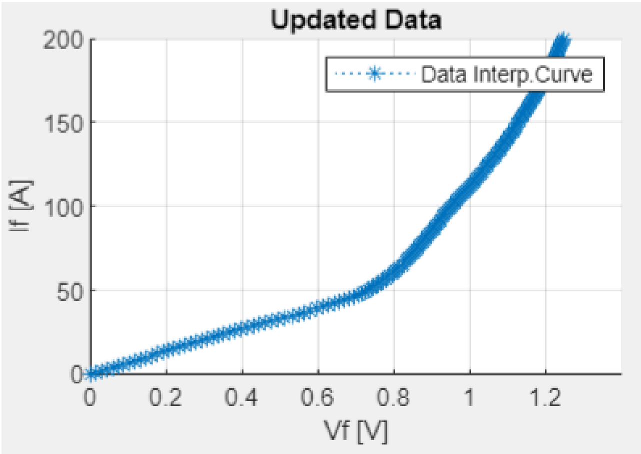


(b) IGBT interpolated curve based on the lookup table.

Figure 4.2: IGBT Conduction model build-up



(a) Diode conduction loss approximation, using the worst-case junction temperature curve [34]



(b) Diode interpolated curve based on the lookup table.

Figure 4.3: Diode Conduction model build-up.

Current,I_c(A)	**Voltage V_{ce}(V)**
0	0
50	1.1
100	1.5
150	1.8
200	2.1
250	2.4

Table 4.3: Selected data points based on the datasheet curve for IGBT.

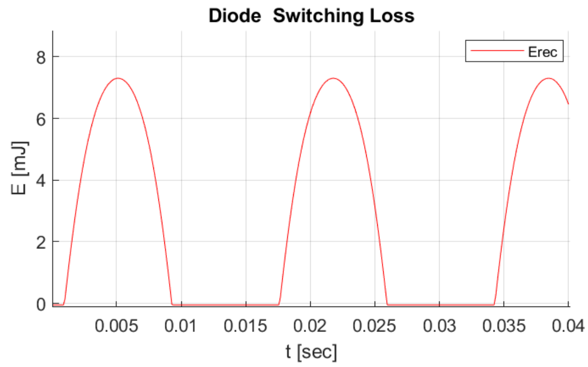
Current,I_f(A)	**Voltage V_f(V)**
0	0
25	0.5
50	1
100	1.3
140	1.5
200	1.7

Table 4.4: Selected data points based on the datasheet curve for diode.

Comparing Figure 4.2a and Figure 4.2b, as well as Figure 4.3a and Figure 4.3b, it can be observed that the approximated curves have a similar trend, but the obtained results are different from those of the datasheet ones. This is because both interpolated values of V_f and V_{ce} are scaled based on the given junction temperature as shown in Equations 3.5, 3.6.

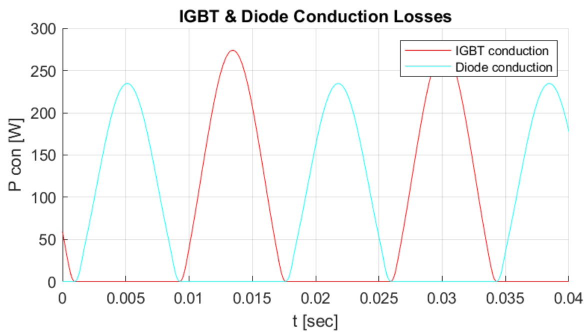
Thus, the corresponding polynomials for the energy and conduction curves are automatically generated, stored in the app, and used for the switching losses along with the conduction losses calculations over time. The obtained losses according to Equations 3.9, 3.10, 3.18, 3.19 are shown in the following Figures:

(a) Simulated IGBT turn-on and turn off energy losses.



(b) Simulated Diode reverse recovery energy losses.

Figure 4.4: Obtained IGBT & anti-parallel diode energy losses.



(a) IGBT (red) and anti-parallel Diode (blue) simulated conduction losses.

(b) IGBT (red) and anti-parallel Diode (blue) simulated switching losses.

Figure 4.5: Obtained IGBT & anti-parallel diode, conduction & switching losses.

The step-by-step power loss calculation procedure in M.A.D. is shown in Appendix B. Based on the created switching and conduction models, the obtained average power losses according to Equations 3.11, 3.12, 3.20, are given in Table 4.5 and compared with:

- a MATLAB -Simulink loss model [14]
- Semikron's simulation tool Semisel [8]
- a PSIM inverter model [35]

Simulation Tool	Power Losses (W)	IGBT conduction losses	IGBT switching losses	Diode conduction losses	Diode switching losses	Total Power losses	Power losses Difference in %
This work	Single Switch	64.37	31.88	63.43	19.95	179.63	-
This work	Complete Inverter	386.22	191.28	380.58	119.7	1077.8	0 %
Simulink Model	Single Switch	83.8	52.0	16.7	26.2	182.7	-
Simulink Model	Complete Inverter	502.8	312.2	124.6	157.1	1096.4	1.68 %
PSIM model	Single Switch	81.0	50.7	20.1	26.6	178.4	-
PSIM model	Complete Inverter	485.8	304.0	120.6	159.6	1070.1	0.68 %
Semisel	Single Switch	81.0	51.0	20.0	16.0	168.0	-
Semisel	Complete Inverter	486.0	306.0	120.0	96.0	1008.0	6.92 %

Table 4.5: Simulated average power losses from the different simulation tools [14]

In order to validate the results obtained with M.A.D. for the IGBT-based inverter from the above Table 4.5 in comparison with the other simulation tools, it is sufficient to show the relative error between the approximated power dissipation. Following the calculation process described in Chapter 3, the obtained results show a good approximation compared to other simulation tools with a maximum difference of 6.92% with Semisel, 1.68% with the Simulink model and only 0.68% with the PSIM model. Therefore:

- From the results obtained, it appears that the application created is working satisfactorily.
- The differences between the individual values between conduction losses and switching losses result from the linear scaling of the datasheet curves based on the off-state blocking voltage and junction temperature, which is not the case for other simulation tools as shown in [14, 35, 36].

A similar validation procedure could not be performed for the SiC MOSFET and GaN HEMT power devices. Although there are some commercially available simulation tools, such as SpeedFit [37] from Wolfspeed for SiC modules and the GaN Systems Circuit simulator tool for GaN HEMTs [38], there are no available articles in the literature that compare their calculated power losses in a 2-level inverter topology with the aforementioned tools, as in the case of [14]. Most of these experiments are performed in laboratories, using different parameters, specific topologies and extensive calculations, as shown for example in [39, 40].

Chapter 5

Discussion and Future Work

5.1 Discussion

The goal of this project was to develop an application that can estimate the power losses of a three-phase, hard-switched inverter. First, an introduction to the Si-based inverter was given, as well as to the Wide Bandgap devices that can improve its efficiency and performance. Then, the types of power losses that occur due to the switching behavior of the power devices were described, as well as some commercially-available simulation tools from the manufacturers that are capable of accurately estimating the power losses.

Using these simulation tools as a guide for estimating power dissipation, MATLAB Application Designer was selected to create such an application. First, the selected application benefits were described and then the structure of the application was determined. This structure was used as a basis to build a step-by-step procedure for power dissipation calculation. Using the manufacturer's datasheet curves, the user manually creates a lookup table, and then based on polynomial curve fitting and interpolation methods, the respective curves were manually obtained.

Based on the process of power dissipation estimation the application was created in MATLAB Application Designer as described in Chapter 3, and a comparative analysis was performed in Chapter 4 to validate its effectiveness. First, an IGBT-based inverter was selected and its operating parameters were defined. The curves from the manufacturer's datasheet were manually obtained for both switching and conduction losses to obtain the corresponding polynomials. Then, for a given sinusoidal current, the losses were determined over a fundamental period. Compared to the other available simulation tools, the application created in MATLAB Application Designer provides good approximate results.

In summary, the application created in MATLAB Application Designer can accurately approximate with a relative error of less than 7%, the power losses of a three-phase inverter based on the manufacturer's data sheet curves.

5.2 Future Work

Due to the limited period of time, some possible ideas that could be explored, using the created application are described in this section as follows:

- Application Flexibility - In this project, the power dissipation calculation process is based on a selected power device datasheet curve into which the parameters are manually inserted in M.A.D. Another approach that can be implemented is to create a "global" library of power devices from all manufacturers. Based on the circuit input parameters, the application can automatically recommend specific part numbers for the particular topology. This approach is already used by commercially available simulation tools such as Semisel, Infineon and Fuji.

- Harmonics - As the simulation tool was designed without taking into consideration the current harmonics, it would be interesting to study how they can affect the efficiency of the inverter.

- Loss calculation - The use of linear scaling in power loss estimation can lead to some estimation errors compared to other simulation tools, as shown in Chapter 4. To make the calculation process more accurate, a temperature dependent polynomial and a voltage dependent polynomial could be introduced for the power dissipation calculation instead. Thus, following the same procedure as described in Chapter 3, the user has the possibility to generate the corresponding polynomials using lookup tables.

Bibliography

[1] Pedro Rodr. Grid Converters for Photovoltaic and Photovoltaic and. 2011. ISBN 9780470057513.

[2] P. Sanjeev and Shailendra Jain. Analysis of conduction and switching losses in two level inverter for low power applications. 2013 Annual IEEE India Conference, INDICON 2013, (December 2013), 2013. doi: 10.1109/INDCON.2013.6725881.

[3] Sachin Bhide and Taehyun Shim. Inverter loss estimation method for real time applications. 2011 IEEE Vehicle Power and Propulsion Conference, VPPC 2011, (April), 2011. doi: 10.1109/VPPC.2011.6043243.

[4] Intelligent Power. Semiconductors et al. - 1998 - General considerations for igbt and intelligent power modules 3.0. (3), 1998.

[5] Jose Millan, Philippe Godignon, Xavier Perpina, Amador Perez-Tomas, and Jose Rebollo. A survey of wide bandgap power semiconductor devices. IEEE Transactions on Power Electronics, 29(5):2155–2163, 2014. ISSN 08858993. doi: 10.1109/TPEL.2013.2268900.

[6] Shan Yin, King Jet Tseng, Rejeki Simanjorang, and Pengfei Tu. Experimental Comparison of High-Speed Gate Driver Design for 1.2-kV/120-A Si IGBT and SiC MOSFET Modules. IET Power Electronics, 10(9):979–986, 2017. ISSN 17554543. doi: 10.1049/iet-pel.2016.0668.

[7] Heena Jain, Sapna Rajawat, and Puja Agrawal. Comparision of wide band gap semiconductors for power electronics applications. 2008 International Conference of Recent Advances in Microwave Theory and Applications, MICROWAVE 2008, pages 878–881, 2008. doi: 10.1109/AMTA.2008.4763184.

[8] SemiSel V5. URL `https://semisel.semikron.com/{#}/home`.

[9] Infineon Tools - Infineon Technologies. URL `https://www.infineon.com/cms/en/tools/landing/ipm.html`.

[10] Power Semiconductors - Fuji IGBT Simulator | Fuji Electric. URL `https://www.fujielectric.com/products/semiconductor/model/igbt/simulation/`.

[11] UnitedSiC FET-Jet Calculator. URL `https://info.unitedsic.com/fet-jet`.

[12] MATLAB App Designer - MATLAB. URL `https://www.mathworks.com/products/matlab/app-designer.html`.

[13] Stig Munk-Nielsen, Lucian N. Tutelea, and Ulrik Jaeger. Simulation with ideal switch models combined with measured loss data provides a good estimate of power loss. Conference Record - IAS Annual Meeting (IEEE Industry Applications Society), 5:2915–2922, 2000. ISSN 01972618. doi: 10.1109/IAS.2000.882580.

[14] Josép Pou, Daniel Osorno, Jordi Zaragoza, Salvador Ceballos, and Carles Jaen. Power losses calculation methodology to evaluate inverter efficiency in electrical vehicles. 2011 7th International Conference-Workshop Compatibility and Power Electronics, CPE 2011 - Conference Proceedings, pages 404–409, 2011. doi: 10.1109/CPE.2011.5942269.

[15] Ned Mohan, Tore M. Undeland, and William P. Robbins. Power electronics. John Wiley & Sons, Inc, 3rd editio edition, 2002. ISBN 9780471226932. doi: 10.3103/S1068371208100015.

[16] CALCULATION OF MAJOR IGBT OPERATING Infineon. Power, pages 1–24.

[17] 1-D data interpolation (table lookup) - MATLAB interp1 - MathWorks Nordic. URL `https://se.mathworks.com/help/matlab/ref/interp1.html`.

[18] L. K. Mestha and P. D. Evans. Analysis of On-State Losses in PWM Inverters. IEE Proceedings B: Electric Power Applications, 136(4):189–195, 1989. ISSN 01437038. doi: 10.1049/ip-b.1989.0025.

[19] F. Casanellas. Losses in PWM inverters using IGBTs. pages 235 – 239, 1994. doi: 10.1049/ip-epa:19941349.

[20] Dušan Graovac and Marco Pürschel. IGBT Power Losses Calculation Using the Data-Sheet Parameters. Infineon Technologies AG, (January):17, 2009. URL `http://www.element14.com/community/servlet/JiveServlet/download/20553-1-3493/IGBTPowerLossesCalculationusingtheDataSheetParameters.pdf`.

[21] Yanming Xu, Carl Ngai Man Ho, Avishek Ghosh, and Dharshana Muthumuni. A datasheet-based behavioral model of SiC MOSFET for power loss prediction in electromagnetic transient simulation. Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC, 2019-March:521–526, 2019. doi: 10.1109/APEC.2019.8721881.

[22] Vorläufige Daten. FP50R12KT4 Technische Information / Technical Information FP50R12KT4. 2013.

[23] STMicroelectronics. Application note IGBT datasheet tutorial. (September):1–35, 2014. URL `https://www.st.com/resource/en/application_note/dm00122161-igbt-datasheet-tutorial-stmicroelectronics.pdf`.

[24] SEMIKRON. Application Manual Power Semiconductors. 2015. ISBN 9783938843666. URL `http://ieeexplore.ieee.org/document/5560803/`.

[25] DYNEX Power through Innovation. Calculating power losses in an IGBT module Application Note AN6156. DYNEX Semiconductor, 44(September):1–14, 2014.

[26] Polynomial curve fitting - MATLAB polyfit - MathWorks Nordic, . URL `https://se.mathworks.com/help/matlab/ref/polyfit.html`.

[27] Polynomial evaluation - MATLAB polyval - MathWorks Nordic, . URL `https://se.mathworks.com/help/matlab/ref/polyval.html`.

[28] Seyyed Abbas Saremi Hasari, Ahmad Salemnia, and Mohsen Hamzeh. Applicable method for average switching loss calculation in power electronic converters. Journal of Power Electronics, 17(4):1097–1108, 2017. ISSN 20934718. doi: 10.6113/JPE.2017.17.4.1097.

[29] Regen Drive. CCS020M12CM2. pages 1–12, 2020.

[30] Dušan Graovac and Marco Pürschel. M O S F ET P o w e r L o s s e s Calculation Using the Data- Sheet Parameters. (July):1–23, 2006.

[31] Infineon Technologies Ag. Infineon OptiMOS TM Power MOSFET Datasheet. (March):1–30, 2012.

[32] GanSystems. GS66508B Bottom-side cooled 650 V E-mode GaN transistor Datasheet Features GS66508B Bottom-side cooled 650 V E-mode GaN transistor Datasheet Absolute Maximum Ratings (T case = 25 ° C except as noted) Parameter Thermal Characteristics (Typical values un. pages 1–16, 2020.

[33] Baochao Wang, Shili Dong, Shanlin Jiang, Chun He, Jianhui Hu, Hui Ye, and Xuezhen Ding. A comparative study on the switching performance of GaN and Si power devices for bipolar complementary modulated converter legs. Energies, 12(6), 2019. ISSN 19961073. doi: 10.3390/en12061146.

[34] Absolute Maximum Ratings. SEMiX151GD066HDs SEMiX151GD066HDs. pages 1–5, 2010.

[35] Psim Software. PSIM Software IGBT Loss Calculation Using the Thermal Module. (March):1–7, 2007.

[36] Jaso Noh. Thermal Calculation and SEMISEL Simulation on IGBT Modules on IGBT Modules SEMISEL Simulation on IGBT Modules.

[37] SiC and GaN Solutions SpeedFit Design Simulator | Wolfspeed | Wolfspeed. URL https://www.wolfspeed.com/speedfit.

[38] Three-phase Traction Inverter | GaN Systems. URL https://gansystems.com/design-center/three-phase-traction-inverter/.

[39] Kazuki Ohta, Yasuki Kanazawal, Hirotaka Akatuka, Shota Hori, Shinji Doki, Hiroshi Tadano, and Koji Shiozaki. Variable switching frequency control for efficiency improvement of motor drive system by using GaN three phase inverter. Proceedings of the IEEE International Conference on Industrial Technology, 2020-February:119–123, 2020. doi: 10.1109/ICIT45562.2020.9067266.

[40] Alessandro Acquaviva, Artem Rodionov, Anton Kersten, Torbjorn Thiringer, and Yujing Liu. Analytical Conduction Loss Calculation of a MOSFET Three-Phase Inverter Accounting for the Reverse Conduction and the Blanking Time. IEEE Transactions on Industrial Electronics, 68(8): 6682–6691, 2021. ISSN 15579948. doi: 10.1109/TIE.2020.3003586.

Appendix A

Code Structure

In order to create the power loss application in MATLAB Application Designer, several steps had to be created to obtain accurate results and allow the end user to build each power semiconductor device individually. The following sections define the IGBT code structure for both the switching and conduction models, as well as the results obtained. Similar process, was followed for the SiC MOSFET and the GaN HEMT.

A.1 Sinusoidal Pulse Width Modulation

```
t=0:0.0000001:0.02; %time vector
% sawtooth wave
Xsw=sawtooth(2*pi*(app.fswEditField.Value)*1000*t,1/2);
% phase A sine-wave
Xa = (app.maEditField_2.Value)*sin(2*pi*(app.fnomEditField.Value)*t);
% phase B sine-wave
Xb = (app.maEditField_2.Value)*sin(2*pi*(app.fnomEditField.Value)*t-(2*pi/3));
% phase C sine-wave
Xc = (app.maEditField_2.Value)*sin(2*pi*(app.fnomEditField.Value)*t-(4*pi/3));
%get the pwm
AmpPwm=1; % pwm amplitude
pwm1= AmpPwm * (Xa>=Xsw); % Comparison
pwm2= AmpPwm * (Xb>=Xsw); % Comparison
pwm3= AmpPwm * (Xc>=Xsw); % Comparison

% Plots
plot(app.UIAxes5,t,Xsw);
legend(app.UIAxes5,'Carrier wave');

plot(app.UIAxes5_2,t,Xa,'k',t,Xb,'r',t,Xc,'m');
legend(app.UIAxes5_2,'Phase A','Phase B','Phase C');

plot(app.UIAxes5_3,t,pwm1,'k',t,pwm2,'r',t,pwm3,'m');
legend(app.UIAxes5_3,'PWM A','PWM B','PWM C');
```

A.2 Switching & Conduction Loss model build-up

For the switching loss calculation, a table is created based on the energy curves of the respective datasheet. Five data-points are selected and then are inserted to the application in order to acquire the respective datasheet curves.

A.2.1 Switching loss model build-up "Original Data"

```
% Read table array from file
Read = readtable("Mapp.xlsx");
vars = {'Ic','Eon','Eoff','Erec'};

% Select a subset of the table array
Read = Read(1:5,vars);

% Add data to the Table UI Component
app.UITable.Data = Read;

% Plot the original data
x1 = app.UITable.Data.Ic;
y1 = app.UITable.Data.Eon;
z1 = app.UITable.Data.Eoff;
z2 = app.UITable.Data.Erec;

p1=polyfit(x1,y1,2); % find the polynomial
p2=polyfit(x1,z1,2); % find the polynomial
p3=polyfit(x1,z2,2); % find the polynomial

I1_new=(0:0.001:100); % range of Current values
y1_new=1.083.*polyval(p1,I1_new); % Evaluate the polynomial at the specified range
y2_new=1.083.*polyval(p2,I1_new); % Vdc/Vdata = 650/600 = 1.083
y3_new=1.083.*polyval(p3,I1_new);

plot(app.UIAxes6,I1_new,y1_new,'-',I1_new,y2_new,'r',I1_new,y3_new,'g');
legend(app.UIAxes6,'Eon','Eoff','Erec');

% Plot the updated data
updateplot(app);
```

A.2.2 Conduction model build-up "Original Data"

```
%% IGBT conduction %%
% Read table array from file
Read11 = readtable("IGBT_cond.xlsx");
vars11 = {'Ic','Vce'};

% Select a subset of the table array
Read11 = Read11(1:6,vars11);

% Add data to the Table UI Component
app.UITable3.Data = Read11;
% Plot the original data
```

```
    y11 = app.UITable3.Data.Ic;
    x11 = app.UITable3.Data.Vce;
    yi=(0:1:100);
    xi = interp1(y11,x11,yi);% Evaluate the Vce value at the specified current range.

    plot(app.UIAxes13,x11,y11,xi,yi,':*');
    legend(app.UIAxes13,'Datasheet Curve','Interpolated Curve');

    % Plot the updated data
    updateplot3(app);

    %% Diode Conduction %%
    % Read table array from file
    Read111 = readtable("Diode_Igbt.xlsx");
    vars111 = {'If','Vf'};

    % Select a subset of the table array
    Read111 = Read111(1:6,vars111);

    % Add data to the Table UI Component
    app.UITable3_2.Data = Read111;
    % Plot the original data
    y111 = app.UITable3_2.Data.If;
    x111 = app.UITable3_2.Data.Vf;
    yi2 = (0:1:100);
    xi2 = interp1(y111,x111,yi2); %Evaluate the Vf value at the specified current range  with ir

    plot(app.UIAxes13_3,x111,y111,xi2,yi2,':*');
    legend(app.UIAxes13_3,'Datasheet Curve','Interpolated Curve');
    %% Plot the updated data
    updateplot4(app);
```

With the above commands, an example is created in the application interface (shown as "Original Data" in the M.A.D.) for the user, showing how the table must be filled in order to obtain the turn-on and turn-off Energy datasheet curves as well as the respective curves for the conduction loss approximation. Following the same procedure the user is able to create his own table in the application based on the device respective datasheet. The below commands are created to give the user the ability to be able to set the values from the respective datasheet. This process is shown in Figure B.4 and Figure B.5.

A.2.3 Switching loss model "Updated Data"

```
% IGBT Switching model
  function updateplot(app)
    % Get Table UI component data
    Read = app.UITable.DisplayData;
```

```
    % Read modified data
    Ic = Read.Ic;
    Eon = Read.Eon;
    Eoff = Read.Eoff;
    Erec = Read.Erec;

    % find the polynomial
    p2=polyfit(Ic,Eon,2);
    p3=polyfit(Ic,Eoff,2);
    p4=polyfit(Ic,Erec,2);

    %new values Vdata,Vdc,Temp
    Vdata=app.VdataEditField.Value;
    Vdc=app.VdcEditField.Value;
    Tdata=app.TempEditField_3.Value;
    Tnew=app.Temp_newEditField_4.Value;

    % range of Current values
    I2_new=(0:0.01:max(Ic));

    % Evaluate the polyn. at the specified range
    y2_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p2,I2_new);
    y3_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p3,I2_new);
    y4_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p4,I2_new);

    %Plot_new
    plot(app.UIAxes6_2,I2_new,y2_new,'-',I2_new,y3_new,'r',I2_new,y4_new,'g');
    legend(app.UIAxes6_2,'Eon','Eoff','Erec');
  end
```

A.2.4 Conduction loss model Updated

```
    % IGBT
    function updateplot3(app)
        % Read table array from file
        Read11 = app.UITable3.DisplayData;

        % Read modified data
        xVce = Read11.Vce;
        yIc = Read11.Ic;

        %new values Tj,Tdata
        Tdata3=app.TempEditField_3.Value;
        Tnew3=app.Temp_newEditField_4.Value;
```

```
        %calculation
        yIc_new=(0:1:max(yIc));%x(x(:,1)
        xVce_new = (Tnew3/Tdata3).*interp1(yIc,xVce,yIc_new,"pchip","extrap");
        plot(app.UIAxes13_2,xVce_new,yIc_new,':*');
        legend(app.UIAxes13_2,'Data Interp.Curve');
    end

    % IGBT DIODE
    function updateplot4(app)
        Read111 = app.UITable3_2.DisplayData;

        % Read modified data
        xVf = Read111.Vf;
        yIf = Read111.If;

        %new values Tj,Tdata
        Tdata4=app.TempEditField_3.Value;
        Tnew4=app.Temp_newEditField_4.Value;

        yIf_new=(0:1:max(yIf));
        xVf_new = (Tnew4/Tdata4).*interp1(yIf,xVf,yIf_new,"pchip","extrap");
        plot(app.UIAxes13_4,xVf_new,yIf_new,':*');
        legend(app.UIAxes13_4,'Data Interp.Curve');
        end
```

A.3 Obtaining Final Results

Since the user has defined the topology parameters, the switching and conduction model with the press of a button, the respective power losses are estimated. The below code structure, shows the process for the power loss calculation. The obtained results are shown in Figure B.6.

A.3.1 IGBT Power Losses Calculation

```
    function ResultsButton_2Pushed(app, event)
        app.TabGroup.SelectedTab=app.IGBTLossesTab;

        t=linspace(0,0.04,400000);
        fsw=app.fswEditField.Value;
        K=1/(fsw*1000);
        t1=linspace(0,0.04,0.04/K);
        ma=app.maEditField_2.Value;
        f=app.fnomEditField.Value;
        Irms=app.IoutEditField.Value;
        cos=app.pfEditField.Value;
        phi= acosd(cos);
```

```
Vdata=app.VdataEditField.Value;
Vdc=app.VdcEditField.Value;
Tdata=app.TempEditField_3.Value;
Tnew=app.Temp_newEditField_4.Value;

%sawtooth wave
Xsw=sawtooth(2*pi*fsw*1000.*t,1/2);
% phase A sine-wave
Xa = ma*sin(2*pi*f.*t);

%Pwm amplitude +Comparison+ Duty Cycle
AmpPwm=1;
pwm1= AmpPwm * (Xa>=Xsw); % Comparison

d=dutycycle(pwm1);

% Sinusoidal Current
i1 = (Irms*sqrt(2))*sin((2*pi*f.*t1)-phi);
i1(i1<0)=0;

i1d= -(Irms*sqrt(2))*sin((2*pi*f.*t1)-phi);
i1d(i1d<0)=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% IGBT Conduction Losses %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% IGBT transistor %%
Read11 = app.UITable3.DisplayData;
% Read modified data
xVce = Read11.Vce;
yIc = Read11.Ic;

xVce_new = (Tnew/Tdata).*interp1(yIc,xVce,i1,"pchip","extrap");
Pigbt=i1.*xVce_new.*(max(d))';

%%%%%%% IGBT Diode %%%%%%%%%
Read111 = app.UITable3_2.DisplayData;
% Read modified data
xVf = Read111.Vf;
yIf = Read111.If;

xVf_new = (Tnew/Tdata).*interp1(yIf,xVf,i1d,"pchip","extrap");
Pdiode = i1d.*xVf_new.*(max(1-d))';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% IGBT Switching losses %%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Get Table UI component data
Read = app.UITable.DisplayData;

% Read modified data
x2 = Read.Ic;
y2 = Read.Eon;
y3 = Read.Eoff;
y4 = Read.Erec;

% find the polynomial
p2=polyfit(x2,y2,2);
p3=polyfit(x2,y3,2);
p4=polyfit(x2,y4,2);

% Evaluate the polynomial at the specified range
y2_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p2,i1); % Eon
y3_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p3,i1); % Eoff
y4_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p4,i1d); % Erec

Pswi = (y2_new+y3_new).*fsw;
Pswd = y4_new.*fsw;
Psw=(y2_new+ y3_new+ y4_new).*fsw;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%IGBT PLOTS Conduction +Switching %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

plot(app.UIAxes8,t1,Pigbt,'r',t1,Pdiode,'c')
legend(app.UIAxes8,'IGBT conduction','Diode conduction');

plot(app.UIAxes7,t1,y2_new,'r',t1,y3_new,'b',t1,y2_new+y3_new,'g');
legend(app.UIAxes7,'Eon','Eoff','Etot');

plot(app.UIAxes7_2,t1,y4_new,'r');
legend(app.UIAxes7_2,'Erec');

plot(app.UIAxes7_3,t1,Pswi,'r',t1,Pswd,'b');
legend(app.UIAxes7_3,'IGBT','Diode');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% IGBT 'Extra' Calculations %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% max-min
Pc_all=(max(Pdiode)+max(Pigbt));
```

```
        Conmax=max(Pc_all);
        app.ConductionLossMaxEditField.Value=Conmax;
        Conmin=min(Pdiode)+min(Pigbt);
        app.ConductionLossMinEditField.Value=Conmin;
        Swmax=max(Psw);
        app.SwitchingLossMaxEditField.Value=Swmax;
        Swmin=min(Psw);
        app.SwitchingLossMinEditField.Value=Swmin;

        % Peak to peak
        Conpeak=peak2peak(Pdiode+Pigbt);
        app.ConductionlossPeaktoPeakEditField.Value=Conpeak;
        Swpeak=peak2peak(Psw);
        app.SwitchingLossPeaktoPeakEditField.Value=Swpeak;

%%%%%%%%%%%%%%%%%%%%%% Avg of one period %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        t2=linspace(0,0.02,200000);
        K2=1/(fsw*1000);
        t3=linspace(0,0.02,0.02/K2);

        %sawtooth wave
        Xsw2=sawtooth(2*pi*fsw*1000.*t2,1/2);
        % phase A sine-wave
        Xa2 = ma*sin(2*pi*f.*t2);

        %Pwm amplitude +Comparison+ Duty Cycle
        pwm2= AmpPwm * (Xa2>=Xsw2); % Comparison

        d2=dutycycle(pwm2);

        % Sinusoidal Current
        i11 = (Irms*sqrt(2))*sin((2*pi*f.*t3)-phi);
        i11(i11<0)=0;

        i11d= -(Irms*sqrt(2))*sin((2*pi*f.*t3)-phi);
        i11d(i11d<0)=0;

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%% IGBT Conduction Losses %%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %%% IGBT
        xVce_new2 = (Tnew/Tdata).*interp1(yIc,xVce,i11,"pchip","extrap");
        Pigbt2=i11.*xVce_new2.*max(d2)';
        Piavg=mean(Pigbt2);
```

```
%%%%%%% IGBT Diode %%%%%%%%
xVf_new2 = (Tnew/Tdata).*interp1(yIf,xVf,i11d,"pchip","extrap");
Pdiode2 = i11d.*xVf_new2.*max(1-d2)';
Pdavg= mean(Pdiode2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% IGBT Switching losses %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Evaluate the polynomial at the specified range
y5_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p2,i11); % Eon
y6_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p3,i11); % Eoff
y7_new = (Tnew/Tdata).*(Vdc/Vdata).*polyval(p4,i11d); % Erec

Pswi2 = (y5_new+y6_new).*fsw;
Pswd2 = y7_new.*fsw;

%%%%%% Average Switching  %%%%%%%%%
PswE = fsw.*mean(y5_new+y6_new);
Pswd = fsw.*mean(y7_new);
Psw_avg= PswE+Pswd;
Psw_avg_tot=6*Psw_avg;
app.EditField.Value=Psw_avg_tot; % total %2

Psw_IGBT=PswE;
Psw_Diode=Pswd;
app.AvgLossesEditField_17.Value=Psw_IGBT; %per switch %1
app.AvgLossesEditField_4.Value=Psw_Diode; %per switch %1

%%%%% Average conduction %%%%%%%%%%%
Pcon_IGBT=Piavg;
Pcon_Diode=Pdavg;
app.AvgLossesEditField_18.Value=Pcon_IGBT; %per switch %3
app.AvgLossesEditField_10.Value=Pcon_Diode; %per switch %3

Pc_avg=Piavg+Pdavg;
Pc_avg_tot=6*Pc_avg;
app.AvgLossesEditField_11.Value=Pc_avg_tot; % total %4

%%%%%%%%%%%EXTRA %%%%%%%%%%%%%
% Total losses displayed
Pc_all2=(max(Pdiode2)+max(Pigbt2));
app.ConductionLossesperswitchEditField.Value=Pc_all2;
Pc_total=6*Pc_all2;
app.TotalConductionLossesEditField.Value=Pc_total;

Psw_all=(max(Pswi2)+max(Pswd2));
```

```
        app.SwitchingLossesperswitchEditField.Value=Psw_all;
        Psw_tot = 6*Psw_all;
        app.TotalSwitchingLossesEditField.Value=Psw_tot;

    end
```

As the "Original Data" follows the same procedure, for the SiC MOSFET and the GaN only the code that estimates the power losses for both devices will be shown.

A.3.2 SiC Power Losses Calculation

```
    function ResultsButton_3Pushed(app, event)

        app.TabGroup.SelectedTab=app.SiCMOSLossesTab;
        ma=app.maEditField_2.Value;
        f=app.fnomEditField.Value;
        t=linspace(0,0.04,400000);
        fsw=app.fswEditField.Value;
        K=1/(fsw*1000);
        t1=linspace(0,0.04,0.04/K);

        Irms=app.IoutEditField.Value;
        cos=app.pfEditField.Value;
        phi= acosd(cos);
        Vdata2=app.VdataEditField_2.Value;
        Vdc2=app.VdcEditField.Value;
        Tdata2=app.TempEditField_4.Value;
        Tnew2=app.Temp_newEditField_5.Value;

        %sawtooth wave
        Xsw=sawtooth(2*pi*fsw*1000.*t,1/2);
        % phase A sine-wave
        Xa = ma*sin(2*pi*f.*t);

        %Pwm amplitude +Comparison+ Duty Cycle
        AmpPwm=1;
        pwm1= AmpPwm * (Xa>=Xsw); % Comparison
        d=dutycycle(pwm1);

        i1 = (Irms*sqrt(2))*sin((2*pi*f.*t1)-phi);
        i1(i1<0)=0;

        i1d= -(Irms*sqrt(2))*sin((2*pi*f.*t1)-phi);
        i1d(i1d<0)=0;

        %%%% SiC Conduction %%%%
```

```
%%%% MOSFET Transistor %%%%
Read15 = app.UITable4.DisplayData;

% Read modified data
Id = Read15.Id;
Rds = Read15.Rds;

% find the polynomial
p15=polyfit(Id,Rds,2);
% Evaluate the polynomial at the specified range
Rds_15 = (Tnew2/Tdata2).*polyval(p15,i1);
Pmos=(i1.^2).*Rds_15.*(max(d))';

%%%% MOSFET diode %%%%
Read12 = app.UITable3_3.DisplayData;

% Read modified data
xVsd = Read12.Vsd;
yIsd = Read12.Isd;
xVsd_new = (Tnew2/Tdata2).*interp1(yIsd,xVsd,i1d,"pchip","extrap");
Pdio=i1d.*xVsd_new.*(max(1-d))';

%%%% SiC-Switching losses %%%%

% Get Table UI component data
Read1 = app.UITable2.DisplayData;

% Read modified data
x5 = Read1.Id;
y5 = Read1.Eon1;
y6 = Read1.Eoff1;
y7 = Read1.Erec1;

% find the polynomial
p5=polyfit(x5,y5,2);
p6=polyfit(x5,y6,2);
p7=polyfit(x5,y7,2);

% Evaluate the polynomial at the specified range
y5_new = (Tnew2/Tdata2).*(Vdc2/Vdata2).*polyval(p5,i1); % Eon
y6_new = (Tnew2/Tdata2).*(Vdc2/Vdata2).*polyval(p6,i1); % Eoff
y7_new = (Tnew2/Tdata2).*(Vdc2/Vdata2).*polyval(p7,i1d); % Erec

Pswi = (y5_new+y6_new).*fsw;
Pswd = y7_new.*fsw;
```

```
        %Plot_new
        plot(app.UIAxes10,t1,y5_new,'r-',t1,y6_new,'b-',t1,y5_new+y6_new,'g-');
        legend(app.UIAxes10,'Eon','Eoff','Etot');

        plot(app.UIAxes11,t1,y7_new,'r-');
        legend(app.UIAxes11,'Erec');

        plot(app.UIAxes10_5,t1,Pswi,'g-',t1,Pswd,'r');
        legend(app.UIAxes10_5,'MOSFET','Diode');

        plot(app.UIAxes10_4,t1,Pmos,'r-',t1,Pdio,'b-');
        legend(app.UIAxes10_4,'MOSFET','Diode');

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%% SIC 'Extra' Calculations %%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        % max-min
        Pc_all=(max(Pmos)+max(Pdio));

        Conmax=max(Pc_all);
        app.ConductionLossMaxEditField_2.Value=Conmax;
        Conmin=min(Pdio)+min(Pmos);
        app.ConductionLossMinEditField_2.Value=Conmin;
        Swmax=max(Pswi)+max(Pswd);
        app.SwitchingLossMaxEditField_2.Value=Swmax;
        Swmin=min(Pswi)+min(Pswd);
        app.SwitchingLossMinEditField_2.Value=Swmin;

        % Peak to peak
        Conpeak=peak2peak(Pdio+Pmos);
        app.ConductionlossPeaktoPeakEditField_2.Value=Conpeak;
        Swpeak=peak2peak(Pswi)+peak2peak(Pswd);
        app.SwitchingLossPeaktoPeakEditField_2.Value=Swpeak;

%%%%%%%%%%%%%%%%%%%%%% Avg of one period %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        t2=linspace(0,0.02,200000);
        K2=1/(fsw*1000);
        t3=linspace(0,0.02,0.02/K2);

        %sawtooth wave
        Xsw2=sawtooth(2*pi*fsw*1000.*t2,1/2);
        % phase A sine-wave
        Xa2 = ma*sin(2*pi*f.*t2);

        %Pwm amplitude +Comparison+ Duty Cycle
```

```
pwm2= AmpPwm * (Xa2>=Xsw2); % Comparison

d2=dutycycle(pwm2);

% Sinusoidal Current
i11 = (Irms*sqrt(2))*sin((2*pi*f.*t3)-phi);
i11(i11<0)=0;

i11d= -(Irms*sqrt(2))*sin((2*pi*f.*t3)-phi);
i11d(i11d<0)=0;

%%%%% MOSFET Transistor %%%%%

% Evaluate the polynomial at the specified range
Rds_2 = (Tnew2/Tdata2).*polyval(p15,i11);
Pmos2=(i11.^2).*Rds_2.*(max(d2))';
Pmos_avg=mean(Pmos2);

%%%%% MOSFET diode %%%%%%
xVsd2_new = (Tnew2/Tdata2).*interp1(yIsd,xVsd,i11d,"pchip","extrap");

Pdiode2=i11d.*xVsd2_new.*(max(1-d2))';
Pdavg= mean(Pdiode2);

%%%%% SiC-Switching losses %%%%

% Evaluate the polyn. at the specified range
y8_new = (Tnew2/Tdata2).*(Vdc2/Vdata2).*polyval(p5,i11); % Eon
y9_new = (Tnew2/Tdata2).*(Vdc2/Vdata2).*polyval(p6,i11); % Eoff
y10_new = (Tnew2/Tdata2).*(Vdc2/Vdata2).*polyval(p7,i11d); % Erec

Pswi2 = (y8_new+y9_new).*fsw;
Pswd2 = y10_new.*fsw;

PswE = fsw.*mean(y8_new+y9_new);
Pswd = fsw.*mean(y10_new);
Psw_avg= PswE+Pswd;
Psw_avg_tot=6*Psw_avg;

app.AvgLossesEditField_16.Value=Psw_avg_tot; % total

Psw_MOS=PswE;
Psw_dio=Pswd;
app.AvgLossesEditField_6.Value=Psw_MOS; % switch
app.AvgLossesEditField_19.Value=Psw_dio; % switch

%%%%% Average conduction %%%%%%
```

```
    Pc_avg=Pmos_avg+Pdavg;
    Pc_avg_tot=6*Pc_avg;
    app.AvgLossesEditField_9.Value=Pc_avg_tot; % total

    Pcon_MOS=Pmos_avg;
    Pcon_dio=Pdavg;
    app.AvgLossesEditField_20.Value=Pcon_MOS; %per switch
    app.AvgLossesEditField_8.Value=Pcon_dio; %per switch

    %%%%%%%%%%EXTRA %%%%%%%%%%%%
    % Total losses displayed
    Pc_all2=(max(Pmos2)+max(Pdiode2));
    app.ConductionLossesperswitchEditField_2.Value=Pc_all2;
    Pc_total=6*Pc_all2;
    app.TotalConductionLossesEditField_2.Value=Pc_total;

    Psw_all=(max(Pswi2)+max(Pswd2));
    app.SwitchingLossesperswitchEditField_2.Value=Psw_all;
    Psw_tot = 6*Psw_all;
    app.TotalSwitchingLossesEditField_2.Value=Psw_tot;

end
```

A.3.3 GaN Power Losses Calculation

```
function ResultsButton_3Pushed(app, event)

    app.TabGroup.SelectedTab=app.SiCMOSLossesTab;
    ma=app.maEditField_2.Value;
    f=app.fnomEditField.Value;
    t=linspace(0,0.04,400000);
    fsw=app.fswEditField.Value;
    K=1/(fsw*1000);
    t1=linspace(0,0.04,0.04/K);
    Irms=app.IoutEditField.Value;
    cos=app.pfEditField.Value;
    phi= acosd(cos);
    Vdata2=app.VdataEditField_2.Value;
    Vdc2=app.VdcEditField.Value;
    Tdata2=app.TempEditField_4.Value;
    Tnew2=app.Temp_newEditField_5.Value;

    %sawtooth wave
    Xsw=sawtooth(2*pi*fsw*1000.*t,1/2);
    % phase A sine-wave
```

```
Xa = ma*sin(2*pi*f.*t);

%Pwm amplitude +Comparison+ Duty Cycle
AmpPwm=1;
pwm1= AmpPwm * (Xa>=Xsw); % Comparison
d=dutycycle(pwm1);

i1 = (Irms*sqrt(2))*sin((2*pi*f.*t1)-phi);
i1(i1<0)=0;

i1d= -(Irms*sqrt(2))*sin((2*pi*f.*t1)-phi);
i1d(i1d<0)=0;

%%%%%%%% GaN Switching %%%%%%%%%%%

Eon=Eon_data*10e-3.*(Vdata3/Vdc3).*(Tnew3/Tdata3).*(i1./Ids);
Eoff=Eoff_data*10e-3.*(Vdata3/Vdc3).*(Tnew3/Tdata3).*(i1./Ids);
Pswg=(Eon+Eoff).*fsw;

%%%%%%%% GaN Reverse Conduction %%%%%%%%

Read13 = app.UITable3_4.DisplayData;
% Read modified data
xVsd_GaN = Read13.Vsd_GaN;
yIsd_GaN = Read13.Isd_GaN;
xVsd_new2 = (Tdata3/Tnew3).*interp1(yIsd_GaN,xVsd_GaN,i1d,"pchip","extrap");
Prev=i1d.*xVsd_new2.*(max(1-d))';

%%%%%%%% GaN Rds_on %%%%%%%%%

Read14 = app.UITable3_5.DisplayData;
% Read modified data
Ids = Read14.Ids;
Rds = Read14.Rds;

% find the polynomial
p14=polyfit(Ids,Rds,2);

% Evaluate the polynomial at the specified range
Rds_new = (Tdata3/Tnew3).*polyval(p14,i1);
Pgan=(i1.^2).*Rds_new.*(max(d))';

%%%%%%%%%%%%%%%% PLOTS %%%%%%%%%%%%%%%%%%%%%%%

plot(app.UIAxes10_2,t1,Eon,'r',t1,Eoff,'g');
legend(app.UIAxes10_2,'Eon','Eoff');
```

```
        plot(app.UIAxes10_6,t1,Pgan,'r-',t1,Prev,'b-');
        legend(app.UIAxes10_6,'GaN','GaN Reverse');

        plot(app.UIAxes10_3,t1,Pswg);
        legend(app.UIAxes10_2,'Switching Loss');

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%% GaN 'Extra' Calculations %%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % max-min
        Pc_all=(max(Prev)+max(Pgan));

        Conmax=max(Pc_all);
        app.ConductionLossMaxEditField_3.Value=Conmax;
        Conmin=min(Prev)+min(Pgan);
        app.ConductionLossMinEditField_3.Value=Conmin;
        Swmax=max(Pswg);
        app.SwitchingLossMaxEditField_2.Value=Swmax;
        Swmin=min(Pswg);
        app.SwitchingLossMinEditField_2.Value=Swmin;

        % Peak to peak
        Conpeak=peak2peak(Prev+Pgan);
        app.ConductionlossPeaktoPeakEditField_3.Value=Conpeak;
        Swpeak=peak2peak(Pswg);
        app.SwitchingLossPeaktoPeakEditField_3.Value=Swpeak;

%%%%%%%%%%% Average for one period %%%%%%%%%%%%%%%%%%%%%%%%

        t2=linspace(0,0.02,200000);
        K2=1/(fsw*1000);
        t3=linspace(0,0.02,0.02/K2);

        %sawtooth wave
        Xsw2=sawtooth(2*pi*fsw*1000.*t2,1/2);
        % phase A sine-wave
        Xa2 = ma*sin(2*pi*f.*t2);

        %Pwm amplitude +Comparison+ Duty Cycle
        pwm2= AmpPwm * (Xa2>=Xsw2); % Comparison

        d2=dutycycle(pwm2);

        % Sinusoidal Current
        i11 = (Irms*sqrt(2))*sin((2*pi*f.*t3)-phi);
        i11(i11<0)=0;
```

```
    i11d= -(Irms*sqrt(2))*sin((2*pi*f.*t3)-phi);
    i11d(i11d<0)=0;

    %%%%%%% GaN Reverse Conduction %%%%%%%

    xVsd_new3 = (Tdata3/Tnew3).*interp1(yIsd_GaN,xVsd_GaN,i11d,"pchip","extrap") Prev_new=i1
    Prev_newavg=mean(Prev_new);

    %%%%%%%% GaN Rds_on %%%%%%%%%

    % Evaluate the polynomial at the specified range
    Rds_new2 = (Tdata3/Tnew3).*polyval(p14,i11);
    Pgan2=(i11.^2).*Rds_new2.*(max(d2))';
    Pgan2_avg=mean(Pgan2);

    %%%%% Avg Conduction %%%%
    Pc_avg=(Pgan2_avg + Prev_newavg);
    Pc_avg_tot=6*Pc_avg;
    app.AvgLossesEditField_15.Value=Pc_avg_tot; % total
    Pcon_GaN=Pgan2_avg;
    Pcon_Rev=Prev_newavg;
    app.AvgLossesEditField_21.Value=Pcon_GaN; %per switch
    app.AvgLossesEditField_14.Value=Pcon_Rev; %per switch

    %%%%% Avg Switching %%%%
    Eon2 = Eon_data.*(Vdata3/Vdc3).*(Tnew3/Tdata3).*(i11./Ids);
    Eoff2= Eoff_data.*(Vdata3/Vdc3).*(Tnew3/Tdata3).*(i11./Ids);
    Pswi2 = (Eon2+Eoff2).*fsw;
    Psw_avg = max(fsw.*mean(Eon2+Eoff2));
    Psw_avg_tot=6*Psw_avg;
    app.EditField_4.Value=Psw_avg; % switch
    app.EditField_3.Value=Psw_avg_tot; % total

    %%%%%%%%%%EXTRA %%%%%%%%%%%

    % Total losses displayed
    Pc_all2=max(Prev_new)+max(Pgan2);
    app.ConductionLossesperswitchEditField_3.Value=Pc_all2;
    Pc_total=6*Pc_all2;
    app.TotalConductionLossesEditField_3.Value=Pc_total;

    Psw_all2=max(Pswi2);
    app.SwitchingLossesperswitchEditField_3.Value=max(Psw_all2);
    Psw_tot2 = 6*Psw_all2;
    app.TotalSwitchingLossesEditField_3.Value=max(Psw_tot2);

end
```

Appendix B

MATLAB Designed Application

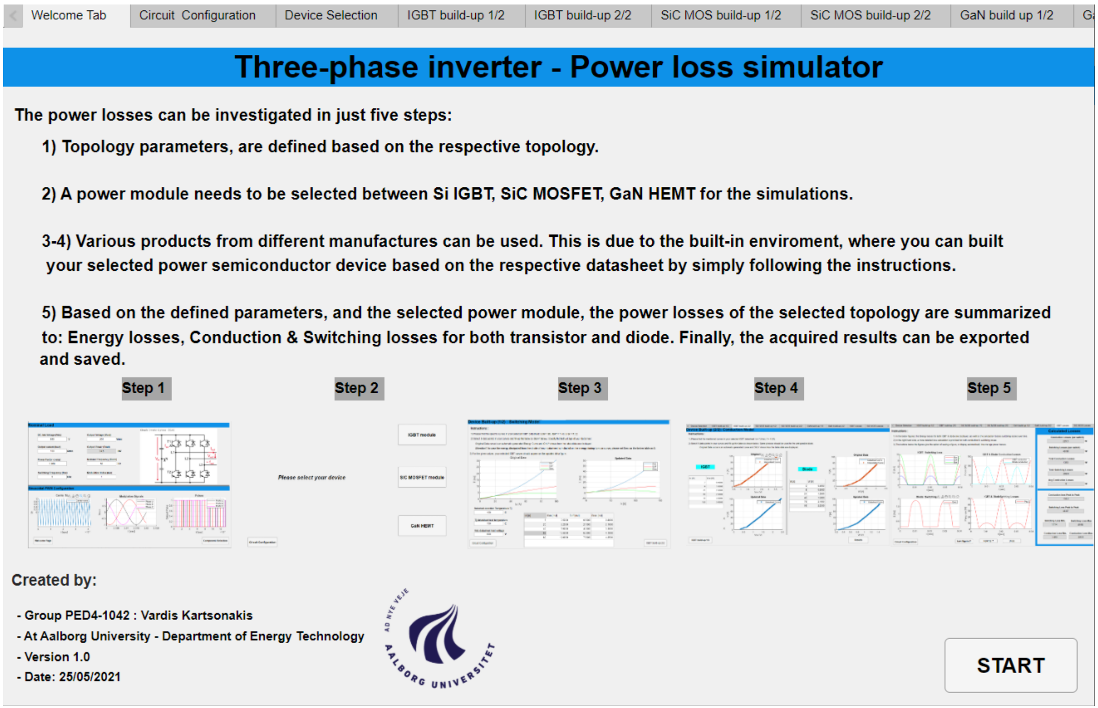


Figure B.1: Welcome Tab of the created app. along with an overview of steps that will be followed for the power loss derivation.

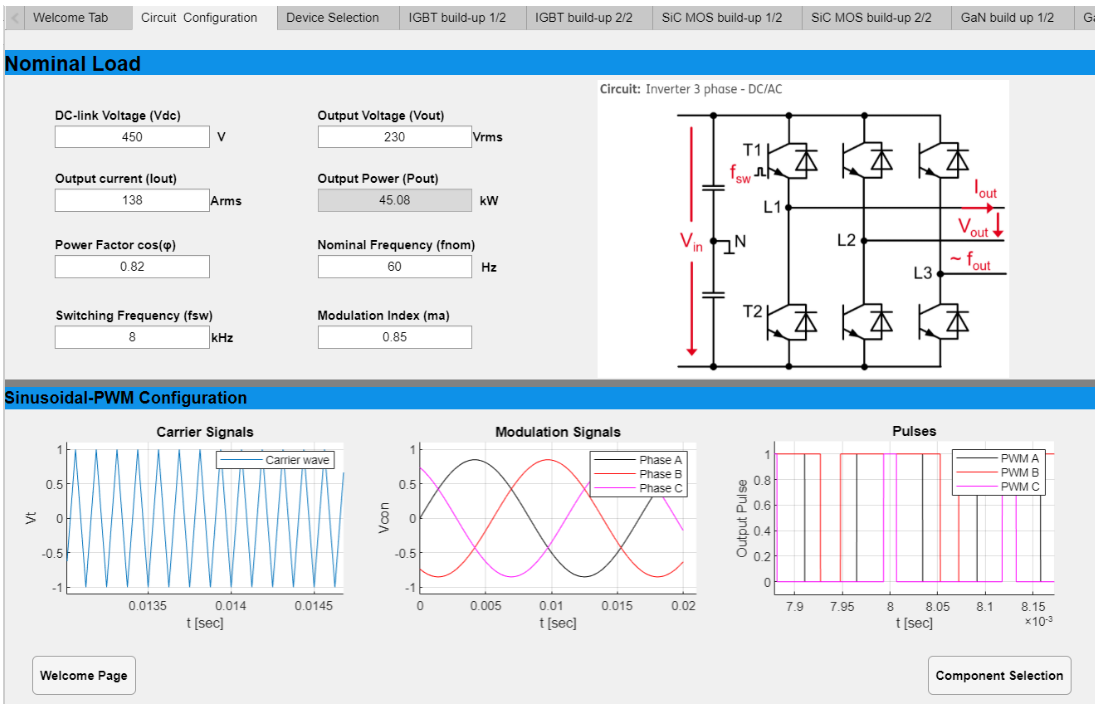


Figure B.2: Three-phase inverter input parameters and regulated Sinusoidal Pulse Width Modulation based on the modulation index and the switching frequency

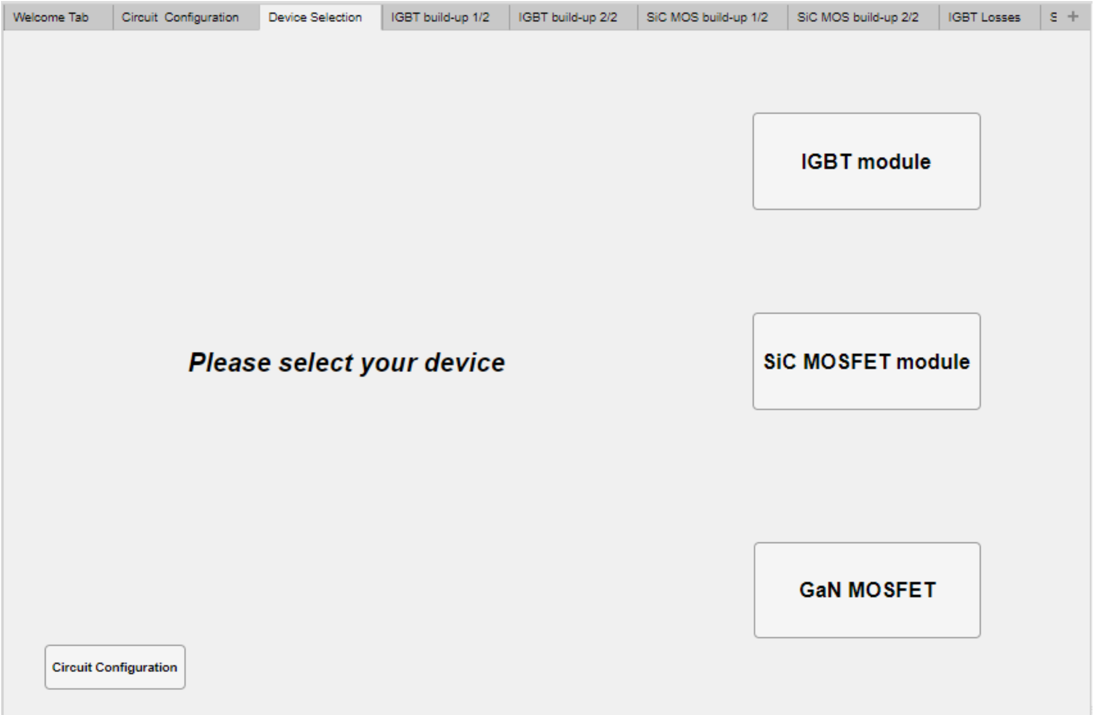


Figure B.3: Device Selection after the input parameter configuration

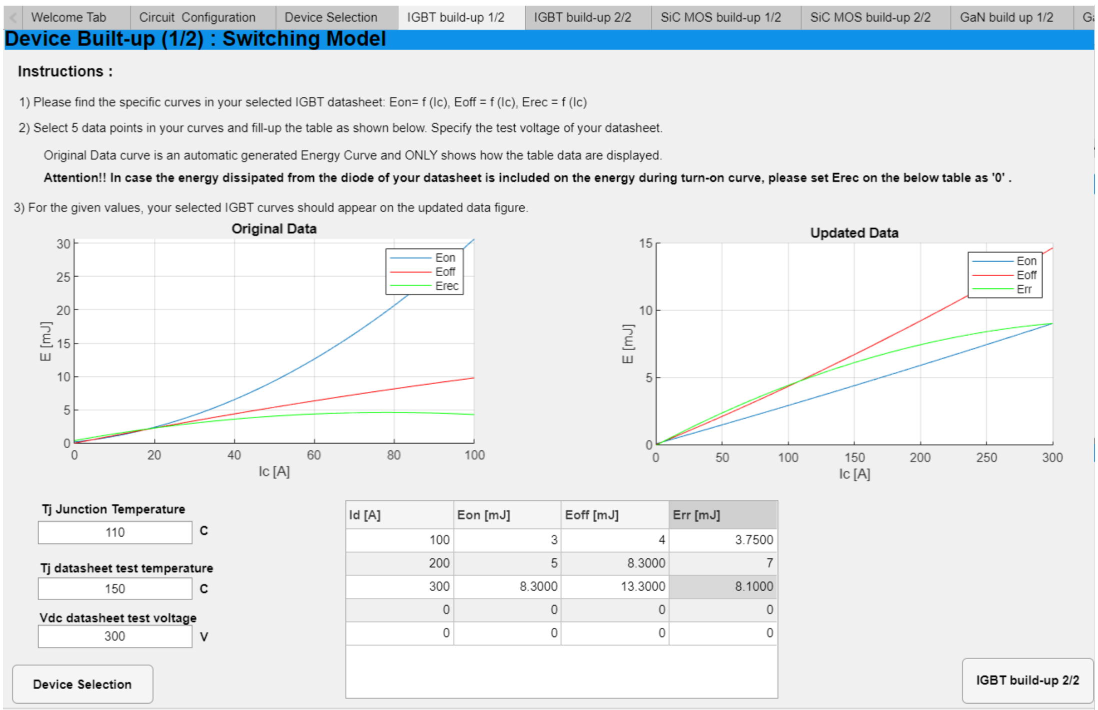


Figure B.4: Switching loss model build-up based on the selected data-points from the datasheet Energy Curves displayed on the lookup table. Using polynomial curve fitting the Energy curves are shown on the "Updated Data" plot scaled by the junction temperature and voltage.

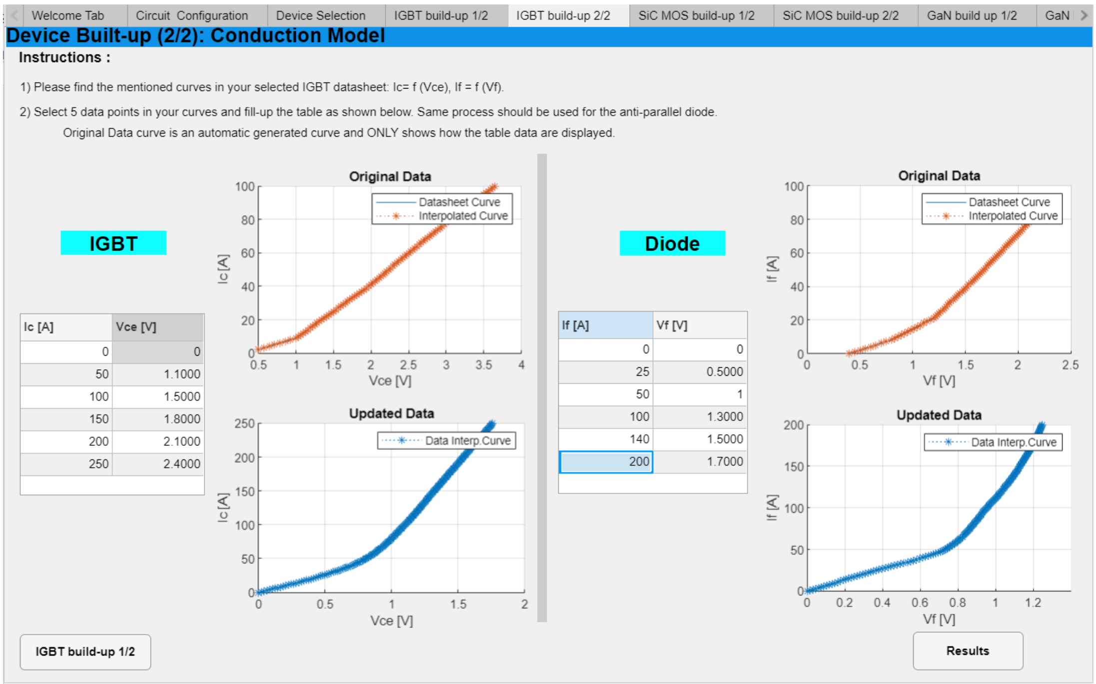


Figure B.5: Conduction loss model build-up based on the datasheet curves, using interpolation method and lookup table.

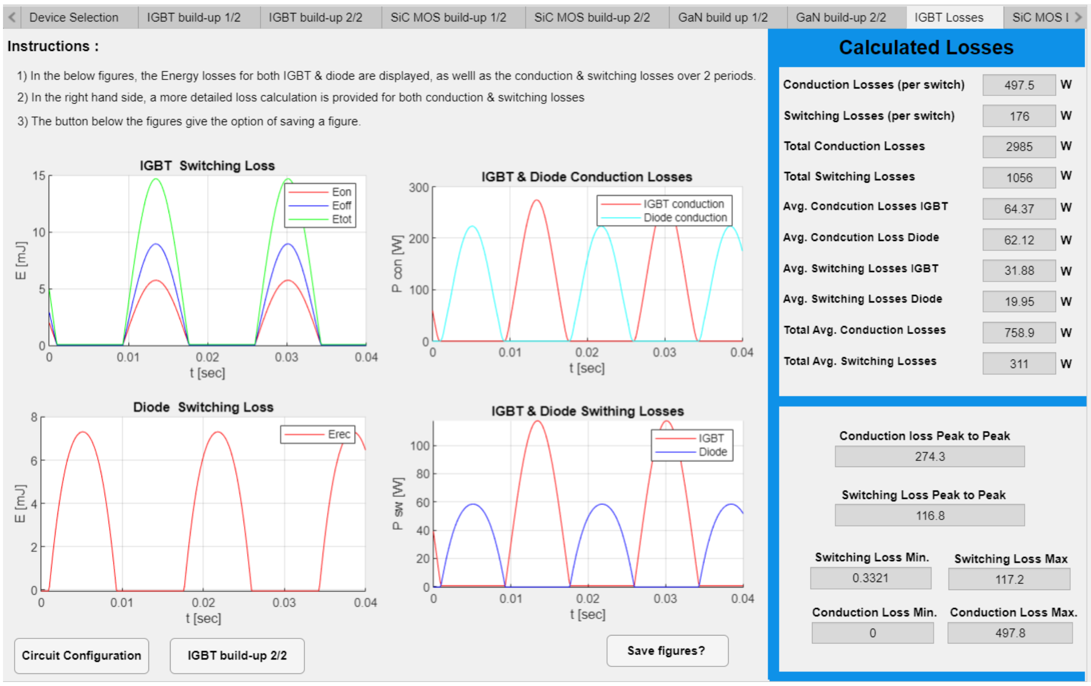


Figure B.6: Obtained power losses based on the switching and conduction models created on the previous steps along with a summary of calculated losses for both IGBT & Diode