

PSpice-MATLAB Based Evolutionary Algorithm for Automatic Extraction of Parasitics in SiC-MOSFET Test Circuits

MSc. Thesis

Stefan Harmen Scholten & Thomas Broberg Jensen

PED4-1044, Spring 2023

Department of Energy Technology



Copyright © Aalborg University 2023

Software programs utilized in the report.

1. diagrams.net - Creating figures.
2. MATLAB - Calculations, scripting, data processing and plotting.
3. Overleaf - Report writing and formatting.
4. PSpice - Simulating electrical circuits.



Department of Energy Technology
Aalborg University
<http://www.et.aau.dk>

AALBORG UNIVERSITY

STUDENT REPORT

Title:

PSpice-MATLAB Based Evolutionary Algorithm for Automatic Extraction of Parasitics in SiC-MOSFET Test Circuits

Project Period:

Spring 2023

Project Group:

PED4-1044

Participant(s):

Thomas Broberg Jensen
Stefan Harmen Scholten

Supervisor(s):

Pooya Davari
Francesco Iannuzzo
Peng Xue

Copies: Unlimited

Page Numbers: 86

Date of Completion:

June 2, 2023

Abstract:

This thesis presents a PSpice-MATLAB based evolutionary algorithm for automatic extraction of parasitics in SiC-MOSFET test circuits. The model presented in this study focuses on extracting parasitic elements from the data acquired from a DPT. The extraction process combines PSO and PS and the objective function uses WT to identify the frequency content as it evolved in time. To evaluate the fitness of the simulation results, the Euclidean distance was employed as a metric. The simulation results was compared to a reference manufacturer model which displayed errors in terms of overshoot and switching energy. Simulation results demonstrate the effectiveness of the algorithm, showcasing significant improvements in overshoot, EMI, and switching energies compared to the reference model. The proposed model aligns well with experimental data, enhances EMI fitting below 30 MHz, and improves switching energy during turn-on and turn-off.

Abbreviations	Description
DPT	Double-pulse test
EA	Evolutionary algorithm
EMC	Electromagnetic compatibility
FET	Field effect transistor
FFT	Fast Fourier transform
FRA	Frequency response analysis
FT	Fourier transform
GA	Genetic algorithm
GaN	Gallium nitride
MOSFET	Metal-oxide semiconductor field-effect transistor
MRA	Multi-resolution analysis
PS	Pattern search
PSO	Particle swarm optimization
RF	Radio frequency
Si	Silicon
SiC	Silicon carbide
SOA	Safe operating area
STFT	Short-time Fourier transform
TDR	Time domain reflectometry
WBG	Wide bandgap
WT	Wavelet transform

Table 1: Nomenclature of abbreviations throughout the report.

Symbols	Description
ψ	Mother wavelet
C	Capacitance
D	Diode
E	Energy/Voltage source
f	Frequency
G	Voltage controlled current source
g	Gain/Small-signal transconductance
I/i	Current
K	Large-signal transconductance
k	Constant coefficient
L	Inductance
M	Voltage slope coefficient
N	Population
R	Resistance
t	Time
T	Temperature
V/v	Voltage
X	Dependent current source
x	Parameter/variable

Table 2: Nomenclature of symbols throughout the report.

Subscripts	Description
-	Negative
+	Positive
C	Case
Cx	Specific capacitor
D/d	Drain
DC	DC rail
ext	External
fw	Freewheeling
g (PSO)	Social
G/g	Gate
gg	Gate drive
i	Index
int	Internal
iss	Input
J/j/jo	Junction
L	Load
loop	Power loop
Lx	Specific inductor
m	Linearized
Mp	Miller plateau
nm	Normalized
off	Off-state
on	On-state
oss	Output
p	Parallel
p (PSO)	Cognitive
rss	Reverse transfer
Rx	Specific Resistor
s	Series
S/s	Source
sat	Saturation
sw	Switching
th	Threshold
tt	Transient time
v	Mutual inductance

Table 3: Nomenclature of subscripts throughout the report.

Superscript	Description
i	Index
k	Iteration

Table 4: Nomenclature of superscripts throughout the report.

Preface

Aalborg University, June 2, 2023

This MSc. thesis is written by a group of students as part of the Master's programme of Power Electronics and Drives at Aalborg University. The prerequisites for reading the report is a good understanding of power electronics, semiconductors and electrical circuit analysis.

The group would like to thank the main supervisors Pooya Davari and Francesco Ianuzzo for their guidance throughout the project. Additional thanks to Peng Xue for supervision and help with double-pulse tests. A special thanks to Karsten Jørgensen and Sigmundur Gardi from Migatronics for their advice and feedback.

References are provided using the IEEE method. All references are sorted in alphabetical order by surname and are included at the end of the report. Figures, tables and equations are numbered according to the respective chapter. The first figure in Chapter 1 has number 1.1, the next has number 1.2 and so on. Figures without references are made by the project group.

Symbols are distinguished using subscripts, and are clarified in the context of the usage. Capital letters are used for steady-state values and lower case letters for dynamic values.



Thomas Broberg Jensen
<tbye18@et.aau.dk>



Stefan Harmen Scholten
<shsc18@et.aau.dk>

Contents

1	Introduction	1
1.1	State-of-the-art	3
1.2	Problem Statement	5
2	SiC MOSFET Parasitic Circuit Model	7
2.1	Static Characteristics	7
2.2	Dynamic Characteristics	8
2.3	Double-Pulse Testing	9
2.3.1	Inductive Load Switching Waveforms	10
2.4	PSpice Model	12
2.4.1	Output Characteristics	13
2.4.2	Capacitance Characteristics	14
2.4.3	C3M Reference DPT Simulation	15
2.4.4	Switching Energies and EMI	17
2.5	Proposed PSpice Model for Parasitic Extraction	19
3	Parasitic Component Extraction	21
3.1	Parameter Optimization Process	21
3.2	Numerical Search Methods	23
3.2.1	Particle Swarm Optimization	23
3.2.2	PSO Algorithm	24
3.2.3	Pattern Search Algorithm	26
3.3	Objective Function	27
3.4	Time-Frequency Analysis	28
3.5	Wavelets and Multi-Resolution Analysis	30
3.5.1	DPT Time-Frequency Representation	32
3.6	Error Computation	33
4	Results	35
4.1	C3M0060065K Performance	37
4.1.1	Switching Energies and EMI	39
4.1.2	Optimization Algorithm	41
4.2	IMZA65R048M1H Performance	42
4.2.1	Switching Energies and EMI	44
4.2.2	Optimization Algorithm	46
4.3	Overall Comparison	47
5	Discussion	49
6	Conclusion	51
7	Future Work	53
	Bibliography	55

- A DPT Setup** **59**

- B PSpice Code** **61**
 - B.1 Circuit Model 61
 - B.1.1 Misc PSpice 63

- C MatLab code** **65**
 - C.1 Configuration of Optimization 65
 - C.2 Optimization Algorithm 66
 - C.3 PSpice Simulation 68
 - C.4 Double Pulse Test Data 71
 - C.5 Import and Post Processing 73
 - C.6 Fitness Evaluation 75
 - C.7 Save Script 76

- D Optimization Algorithm** **77**

- E Simulation Data** **79**
 - E.1 C3M0060065K 79
 - E.2 IMZA65R048M1H 82
 - E.3 Mean and Standard Deviations 85

1. Introduction

With the increasing demand for renewable energy, power electronic converters has become an integral part of transmission, distribution and storage of energy [1]. The shift towards electrification, especially in the automotive sector, has increased the demand for ultra-compact, energy-efficient and highly reliable power electronic converters [2, 3]. For the previous decades, silicon (Si) based semiconductors have dominated the electronic market and has made up the vast majority of all commercially available semiconductor devices [4]. In recent years, wide bandgap (WBG) devices i.e. gallium nitride (GaN) field-effect transistors (FETs) and silicon carbide (SiC) metal-oxide semiconductor field-effect transistors (MOSFETs), have also emerged due to their favorable material properties [2]. Fig. 1.1 shows the distribution of commonly used semiconductor technologies, with a switching frequency above 1 kHz, with respect to switching frequency and power ratings [5].

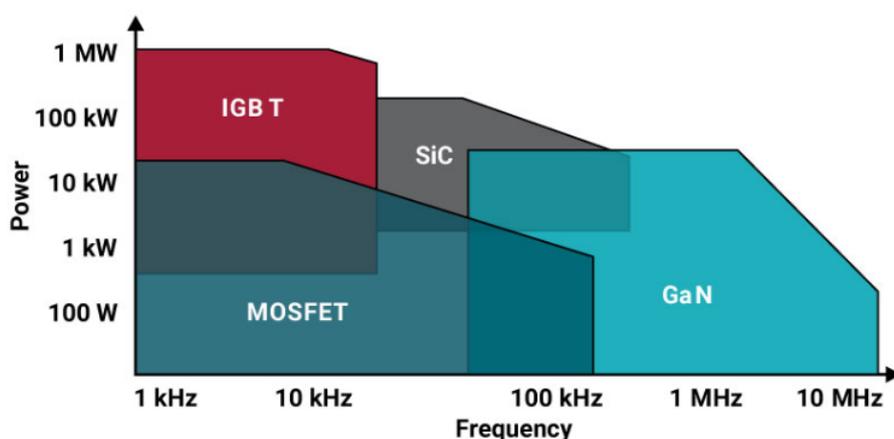


Figure 1.1: Capabilities of commercially available semiconductor devices. Frequency vs. power [5].

In low power applications, with a voltage range below 600 V, FETs are favorable due to their high input impedance and low switching losses. However, when higher blocking voltage is required, the on-state resistance of Si MOSFETs becomes too large and conduction losses significantly increase as a result [4]. Above 600 V, the Si based Insulated-Gate Bipolar Transistor (IGBT) is the semiconductor device of choice. Their popularity is largely due to their high blocking voltage and current-carrying capabilities which enable them to cover a wide area of possible applications in medium-power electronic converters, ranging from a few kilowatt to several megawatts [2].

However, there is a general agreement among researchers that Si technology has matured to a point where further improvements are difficult to achieve [6, 7]. For this reason, WBG devices are regarded as a promising replacement for Si IGBTs in the medium-voltage range. Compared to equivalently rated Si devices, WBG devices have the potential to operate at higher temperatures, with lower conduction losses, faster switching speed and higher switching frequency which reduces the overall system cost and size [6, 8]. However, the faster transients (high di/dt or dv/dt), combined with parasitic inductance in device packages and general printed circuit board (PCB) layout, also introduce unwanted effects

such as switching oscillations, electromagnetic interference (EMI), additional power losses and device stress which can compromise the safe operating area (SOA) and lead to derating of components. These effects can also be observed in e.g. low voltage Si trench MOSFETs, but the more prominent impact in WBG devices causes reliability issues and necessitates derating of components [9]. Over the last decades, SiC devices have evolved from laboratory prototypes to commercial products and are expected to prevail over Si-based devices in the near future [8]. This report will therefore focus on investigating discrete SiC devices.

When electromagnetic signals are unintended, it is called electromagnetic noise and when it causes degradation of equipment performance or malfunction it is called EMI [10, 11]. The allowable EMI levels depend on the frequency range and it is regulated by several governing bodies [12]. An overview of the regulated frequency ranges is shown in Fig. 1.2. The standards cover the frequency range from 0 to 300 GHz, but not all ranges are regulated and some are only regulated for specific products [11, 13].

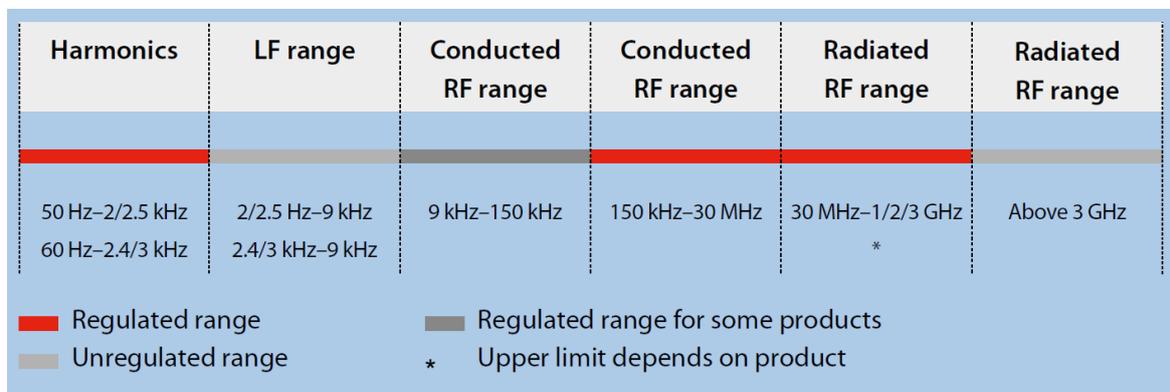


Figure 1.2: Definition of regulated frequency ranges [11].

The low frequency range from 50 Hz to 2.5 kHz regulates the harmonics of the 50 Hz distribution network frequency and typically considers up to 40th or 50th harmonic. The LF range from 2.5 kHz to 9 kHz considers power quality and is not currently regulated [11]. The ranges above 9 kHz is referred to as the radio frequency (RF) range and is divided into conducted and radiated emission [13]. Conducted emission below 30 MHz is transmitted through cables and is often caused by parasitic effects such as series inductance and capacitive effects. Further constraints are therefore imposed on PCB designers when incorporating SiC devices, since filters or shielding is required to comply with electromagnetic compatibility (EMC) standards [8]. However, EMC is often the last step in a design process and it can become very expensive and difficult to handle. For this reason, design engineers should consider EMC design and placement earlier in the design process [11].

In early design stages, computer-aided simulations are utilized to great advantage since it enables engineers to safely study the behavior of complex high-power systems without having to build or operate a physical system [14]. SiC device models are often provided by manufacturers, but these do not always account for non-linearities and parasitics, which are unique to a particular circuit [15]. For this reason, simulation models are often inaccurate and a double pulse test (DPT) is necessary to analyze switching characteristics, which are closely related to parasitics. The DPT switching waveforms can be used to iden-

tify dominant circuit parameters, which would greatly increase the accuracy of simulation models [4]. Direct extraction methods are often used for e.g. MOSFET model parameters which are found in data sheets, but these are only valid at the specified operating point [16]. In many cases, higher accuracy can be achieved by sweeping around the initial parameter values, but this method is often time consuming when numerous parameters are included. For this reason, nature-inspired optimization methods i.e. evolutionary algorithms (EAs) such as genetic algorithms (GAs), can be applied to greatly improve convergence time and accuracy. These algorithms are commonly used to generate high-quality solutions to optimization problems and has become a powerful tool for parameter extraction in active and passive devices [17].

1.1 State-of-the-art

Previous research regarding parasitic extraction and modeling of SiC MOSFETs have focused on three areas, the models used to describe the behavior of the MOSFET, the effect from the parasitic elements on the switching characteristics and the methods used for estimating the parasitic elements.

In [18] MOSFET models are categorized in three main types, tabulated models, compact models and subcircuit or macro models. Manufacturer models are based on either tabulated or macro models, with a varying degree of simulation accuracy [15]. However, these models does not incorporate circuit parameters [15] and as shown in [19], discontinuities in variable parameters can result in convergence errors. Even though compact models do not necessarily have the same level of accuracy, it is easier to relate model parameters to circuit parameters [18].

In [4], an analysis of the effects on the switching waveforms from circuit elements of a double pulse test is done, with the purpose of improving the switching waveforms of SiC MOSFETs in parallel by studying the parasitic elements. In [20] the effect from both circuit and device parameters on the gate voltage is studied. A tool to assist the manual tuning of a spice model to test results is developed, which automates the process of choosing which parameters to tune based on how the designer wants to change characteristics.

Generally, the experimental method chosen for extracting parasitic elements of a MOSFET is either Time Domain Reflectometry (TDR) or a Frequency Response Analysis (FRA) [21]. TDR is based on transmission line theory, which identifies parasitic elements based on delay times of the reflected signals and while accurate, requires expensive equipment [21, 22]. The FRA based methods are done with either one- or two-port measurement, to find the S-parameters of the circuit. However, this method still requires specialized equipment and is applied with floating terminals and often at low voltages. This means that the non-linearity of parameters, such as voltage dependency of capacitors is not estimated accurately [21]. A genetic algorithm has been applied to optimize the parameter extraction from a FRA based test in [17]. However, the model developed is only valid for the saturation region, and the results suffer from the previous mentioned limitations of FRA based methods. Other methods not based on TDR or FRA have been proposed in [21, 23]. [23] proposes a simple method using only standard tools available to electronic engineers, but the method is only valid for estimating the common-source inductance. In [21] a series of tests which enable an analytical isolation of the parasitic elements is proposed, and good correlation between simulation model results and DPT is achieved. The long series of tests

and calculations are inevitably quite time consuming.

In summary, current methods for estimating parasitic circuit elements are based either on arduous manual tuning or a series of specialized tests or tests with expensive and specialized equipment. Though papers such as [20], have proposals for manually estimating parasitic circuit elements through a double pulse test and [17] have used a genetic algorithm to estimate parasitic elements from a FRA based test. These projects show promising results for the methods used, and therefore exploring the possibility of using evolutionary algorithms to automate the process of estimating parasitic elements from simple and accessible tests such as a DPT is interesting. Research into this, could potentially reduce the workload and increase the accessibility of advanced circuit design where parasitic circuit elements are taken into account during the design process.

1.2 Problem Statement

Based on the information presented in Ch. 1, the following problem statement is constructed:

"To what extent can a PSpice-MATLAB based evolutionary algorithm be developed for automatic extraction of parasitics in SiC-MOSFET test circuits using DPT test data?"

5

Research aims: This paper will explore how the process of obtaining the values for parasitic elements on a circuit board can be improved through the use of evolutionary algorithms. A circuit model which can accurately describe the switching behavior also needs to be developed for this purpose.

10

Methods: This will be achieved by developing an optimization algorithm in MATLAB, which will fit a PSPICE circuit model, to match data obtained from a DPT test setup from Keysight Technologies. The results will be verified by testing two devices from different manufactures, namely C3M0060065K and IMZA65R048M1H, in the same circuit setup, and analyzing the correlation of the device and circuit parameters.

15

Scope: The results will be based on tests with two SiC devices in the TO-247-4 package. This package is chosen due to availability and usage. A single package type is chosen, in order to ensure both devices can be tested on the circuit board. This report will neglect temperature dependency of both parameter results. It is assumed this can be neglected because of the short time span of a DPT. Validating the obtained parameters through different methods such as FRA or TDR is not done. Some work will be done on developing an accurate circuit model, but the project will depend on the accuracy of work by done by others for this part. These restriction are made with the time frame of this project in mind and the low amount of available research into the proposed process of automating parasitic extraction from DPT warranting a large focus on developing an efficient algorithm.

20

25

Thesis outline:

To address the given problem problem statement, the thesis is organized as follows.

- 5 - **Chapter 1** discusses the future trends in power electronics and it discusses the advantages SiC devices and the challenges associated with parasitic circuit elements. The importance of computer-aided simulations is then highlighted and state-of-the-art parasitic extraction methods are discussed. The goal of the research paper is then presented.
- 10 - **Chapter 2** describes the MOSFET switching characteristics including output- and capacitance characteristics. The purpose of a DPT is then explained with the aid of simple MOSFET switching waveforms. Overshoot, EMI and switching energies of measured DPT data is then compared to simulation results, using a manufacturer device model. A PSpice circuit model is then developed, based on the default device model, and parasitics are incorporated.
- 15 - **Chapter 3** focuses on the process of extracting parasitic circuit elements. It first presents the optimum design process and then discusses different optimization methods. The iterative processes of PSO and PS algorithms are then explained. It is then discussed how the objective function should be defined and the importance time/frequency analysis is highlighted. An overview of MRA and WT is given before the error computation method is defined.
- 20 - **Chapter 4** presents the results of the proposed model by comparing two devices with the default model and measured data. An overall comparison is then presented and the accuracy and consistency of the fitted parameters is assessed.

2. SiC MOSFET Parasitic Circuit Model

The switching characteristics of power semiconductors are essential when calculating losses, predicting lifetime and designing EMI filters or heat sinks [4]. In the following chapter, the basic switching behavior of a MOSFET is investigated and the purpose of a DPT is explained. Experimental DPT waveforms is then compared to a simulation results obtained using a commercially available device model. An improved simulation model is then developed which is able to replicate effects from external circuit parasitics.

2.1 Static Characteristics

The behavior of a MOSFET can be separated into static and dynamic characteristics. The static characteristics of an N-channel MOSFET is shown in Fig. 2.1. Depending on the gate-source voltage, the MOSFET is operated in the active, cut-off or ohmic region.

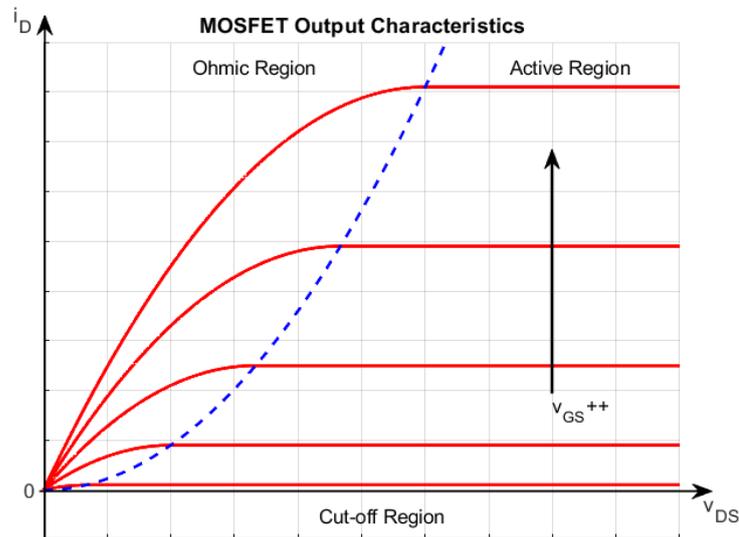


Figure 2.1: Static characteristics of an N-channel MOSFET.

In power electronics converters, the MOSFET is preferably operated in the cut-off or ohmic region where either the drain-source voltage or drain current is low so power losses are reduced [14]. In the three regions, i_D is approximated by the following expressions [14]:

$$i_D(t) = \begin{cases} 0 & \text{Cut-off: } (v_{GS} < V_{th}) \\ K \cdot (v_{GS} - V_{th})^2 & \text{Active: } (v_{GS} > V_{th}) \ \& \ (v_{DS} \geq v_{GS} - V_{th}) \\ K \cdot [2(v_{GS} - V_{th}) \cdot v_{DS} - v_{DS}^2] & \text{Ohmic: } (v_{GS} > V_{th}) \ \& \ (v_{DS} < v_{GS} - V_{th}) \end{cases} \quad (2.1)$$

Where K is the large-signal transconductance, a property of the MOSFETs physical structure that determines how much current it can carry for a given v_{GS} . When the MOSFET is fully off, v_{GS} is below the threshold voltage and it operates in the cut-off region where v_{DS} is high and i_D roughly zero. When it is fully on, it operates in the ohmic region where

v_{DS} is low and the second-order term of Eq. 2.1 can be neglected. The on-state resistance is then approximated as a function of v_{GS} .

$$R_{DS,on} = \frac{v_{DS}}{i_D} \approx \frac{1}{2K \cdot (v_{GS} - V_{th})} \quad (2.2)$$

From Eq. 2.2, it can be seen that higher v_{GS} reduces $R_{DS,on}$ and is therefore desirable to reduce conduction losses and operate the MOSFET efficiently. As the MOSFET turns on or off, it traverses the active region where the product of i_D and V_{DS} is non-zero, resulting in switching losses [14]. For small-signal modeling, a linear approximation of i_D is typically used [14]:

$$i_D \approx g_m \cdot (v_{GS} - V_{th}) \quad (2.3)$$

Where g_m is the transconductance which determines the change in i_D for a given operating point in the active region as a function of v_{GS} . The transitioning between the ohmic and active region is the saturation point and is marked by the dashed line for a given v_{GS} . The saturation voltage can be approximated as $V_{DS,sat} = (V_{GS} - V_{th})$ with $I_{D,sat} = K \cdot V_{DS,sat}^2$ [14].

2.2 Dynamic Characteristics

The dynamic characteristics of a MOSFET is mainly determined by the charging and discharging of internal capacitances [24]. The internal capacitances are shown in Fig. 2.3 and their values are determined by the physical construction of the MOSFET.

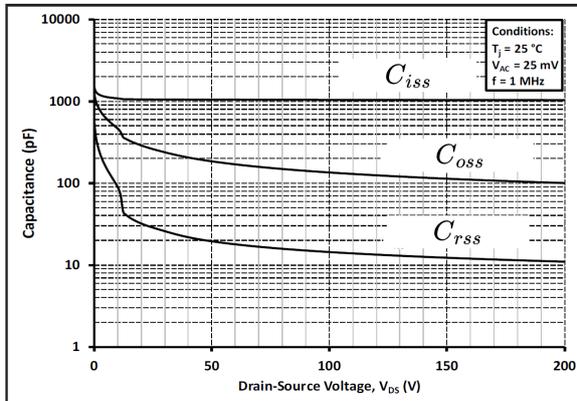


Figure 2.2: Capacitance for different values of drain source voltage, measured at $T_j = 25^\circ\text{C}$ and $v_{GS} = 0$. From C3M0065065K datasheet [25]

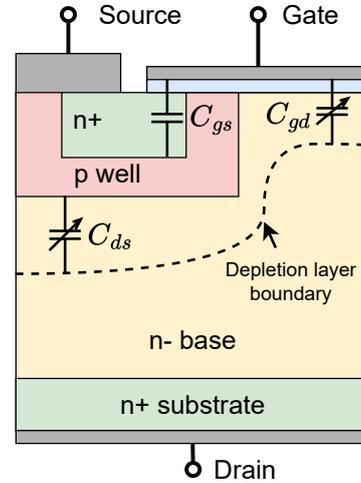


Figure 2.3: MOSFET cross-section with internal capacitors.

In datasheets, the internal capacitances are given as input, output and reverse transfer capacitance and are accompanied by a characteristic curve as seen in Fig. 2.2. The capacitances are a function of the voltage across them and their values can change by up to a factor of 100 from turn on to off [26]. The reverse transfer capacitance or Miller capacitance is equal to the drain-gate capacitance as defined in Eq. 2.4.

$$C_{rss} = C_{GD} \quad (2.4)$$

C_{rss} is a function of the effective base width which changes as the depletion layer changes as shown in Fig. 2.3 [24]. The charging and discharging of C_{GD} determines the drain-source voltage rise- and fall-time as well as the turn-off delay time. The input capacitance is made up of the drain-gate and gate-source capacitors and it must be charged and discharged for the MOSFET to turn on and off [24, 26].

$$C_{iss} = C_{GS} + C_{GD} \quad (2.5)$$

C_{GD} adds some voltage dependency to C_{iss} , but C_{GS} is generally much larger and C_{iss} is therefore mostly independent of voltage. The voltage of this capacitor controls the on/off state of the device [24, 26]. The output capacitance affects the energy stored from the power circuit during switching, and is calculated as in Eq. 2.6.

$$C_{oss} = C_{DS} + C_{GD} \quad (2.6)$$

Both of these capacitances changes with the depletion layer and are therefore dependent on the drain-source voltage. C_{oss} affects switching losses since it is charged during turn-on and discharged in the body-diode during turn-off. It also affects the resonance of the power circuit for soft switching applications [24, 26].

2.3 Double-Pulse Testing

The dynamic characteristics are often analyzed in an inductive load circuit as shown in Fig. 2.4 [26]. It is common to test the inductive switching characteristics using a double-pulse test (DPT).

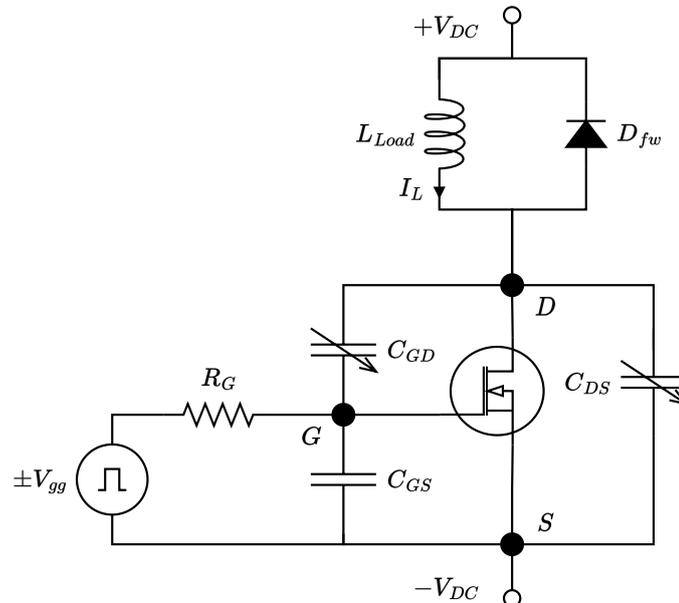


Figure 2.4: Diode-clamped inductive load circuit.

In a DPT, the device under test is turned on/off twice in rapid succession. When the device is on, the current increases at a linear rate through a load inductor which remains

almost constant when the device is turned off. This produces a turn off and turn on event at the same current level, which is configured by controlling the length of the initial pulse. This test is widely used to determine differences in switching characteristics between turn on and turn off, such as rise and fall time or switching losses, since the load conditions for both events are equal [27].

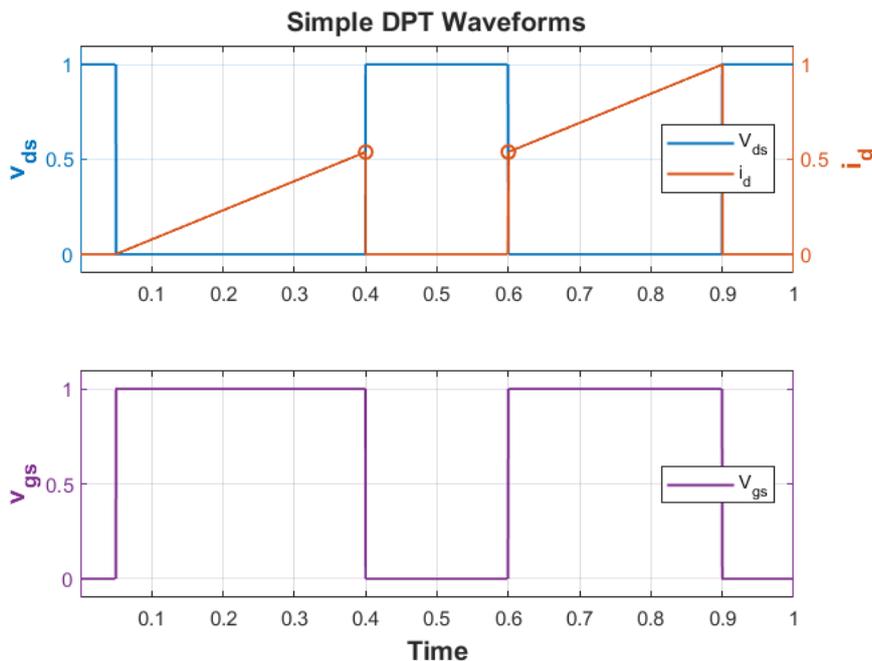


Figure 2.5: Simplified switching behavior in the DPT.

A simplified example of a DPT waveform with normalized axes, is shown in Figure 2.5. It illustrates that the current level at the turn off and turn on events, marked with red circles respectively, is equal for both events and can be controlled by increasing or decreasing the length of the initial gate pulse v_{gs} . The voltage v_{ds} over the device during turn off, is equal to the DC source voltage applied, and is therefore also controllable.

2.3.1 Inductive Load Switching Waveforms

The simplified DPT waveforms are illustrated in Fig. 2.6. The waveforms helps illustrate transients during the switching operation which provides insights into the efficiency and switching behavior. This preliminary analysis can be used to predict when the various circuit parasitics influence the MOSFET device.

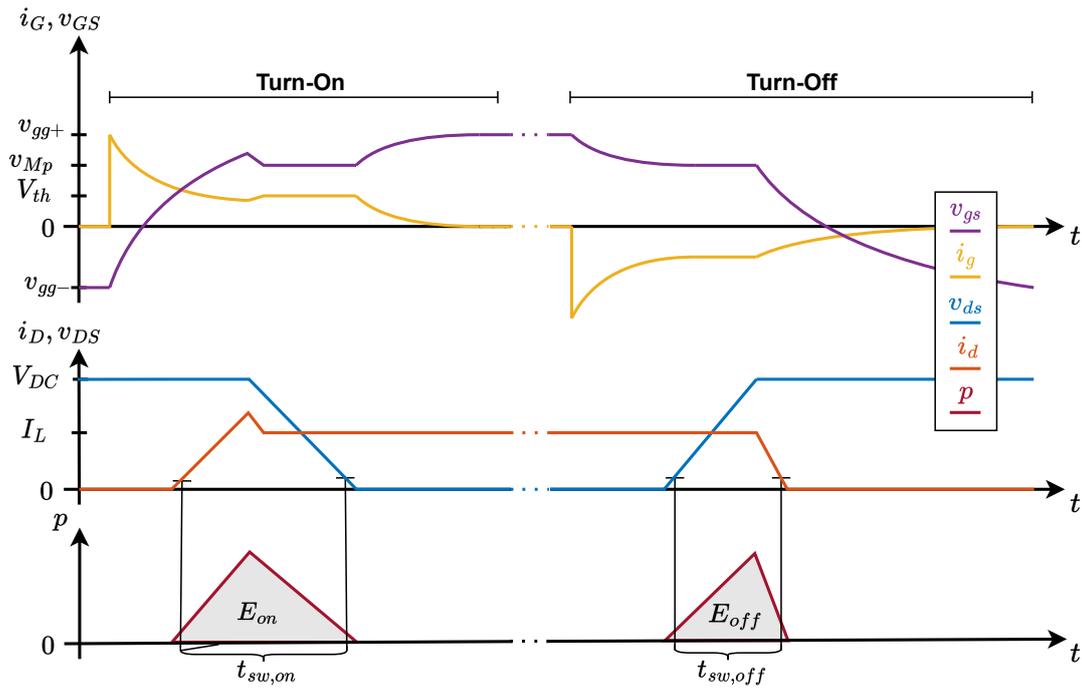


Figure 2.6: Turn-on and turn-off waveforms and definition of switching losses.

The load current I_L is assumed to be constant and is initially freewheeling through the diode. v_{GS} is clamped at negative gate voltage and the MOSFET operates in the cut-off region where $v_{DS} = V_{DS}$ and $i_D = 0$. When a positive gate voltage is applied, v_{GS} starts rising and when $v_{GS} = V_{th}$, the MOSFET starts conducting. At this point, the MOSFET enters the active region where current and voltage are no longer zero and power losses starts increasing. When $i_D = I_L$, the high-side diode starts releasing accumulated charge as reverse recovery which causes an overshoot in i_D . At the peak current, C_{GD} starts discharging which causes v_{DS} to decrease. During this period, the MOSFET transitions towards the ohmic region while v_{GS} is clamped at the Miller plateau voltage. When v_{DS} reaches saturation voltage, the MOSFET enters the ohmic region and V_{GS} starts rising to positive gate voltage. The turn-on process lasts from when the PWM signal is first applied at $v_{GS} = v_{gg-}$ until $v_{GS} = v_{gg+}$ and the MOSFET is fully on. The turn-off process is similar, but reversed and without reverse recovery, since the diode is not conducting any current when the MOSFET is on [14, 26].

Conventionally, turn-on energy is defined as the integral of the product of i_D and v_{DS} during the period where i_D exceeds 5-10% to when v_{DS} drops to 5-10%. Conversely, turn-off energy is defined from when v_{DS} exceeds 5-10% to when i_D drops to 5-10%. The rise- and fall-times are defined as the time it takes for i_D to increase from 10-90% or decrease from 90-10%. These values are often provided in datasheets as they are only advisory since they are only valid at the setup where they were measured [26]. For this reason, circuit specific transient times are vital for design engineers when modeling EMI and power losses.

2.4 PSpice Model

Before testing a device using a DPT, the switching characteristics can be estimated using a simulation model. Manufactures typically provide device model, such as the C3M series model provided by Wolfspeed. A circuit diagram of the lumped-element model is shown in Figure 2.7. This model is made for the TO-247-4 package, but the basic structure is the same for other device packages. The model accounts for the body diode, output characteristics, non-linear capacitors and temperature dependency of the threshold voltage.

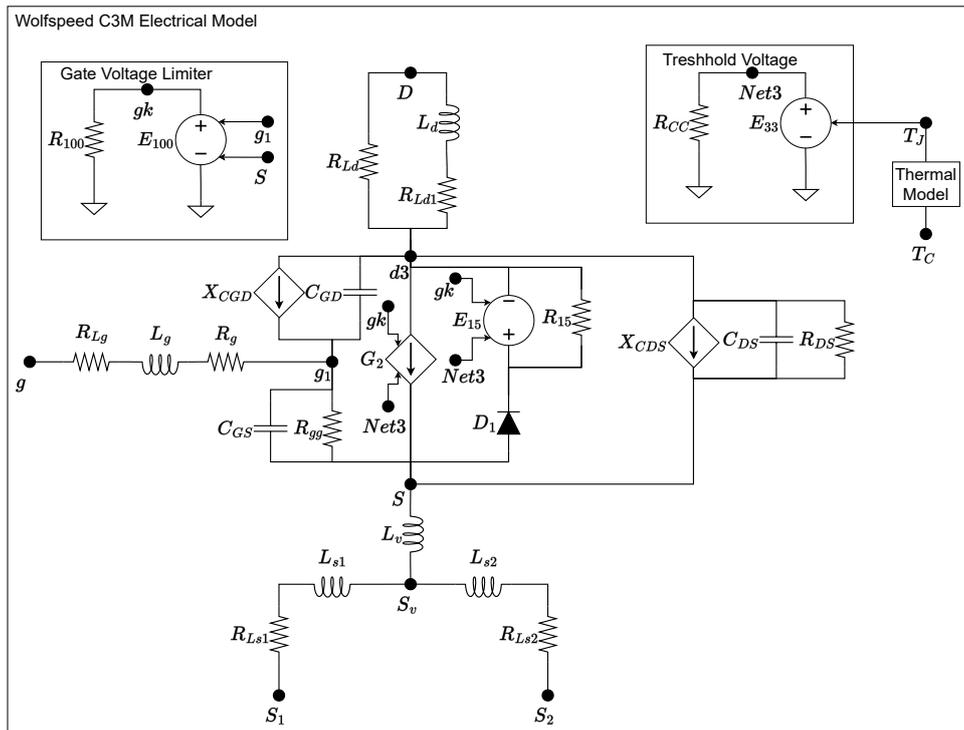


Figure 2.7: Diagram of Wolfspeeds C3M model for the TO-247-4 package. Symbols and subscripts follow the notation used in Wolfspeeds device model [25].

The model is build around a core that models the semiconductor die, with the output characteristics modeled by the current source G2. The internal capacitor are implemented as gate-drain, gate-source and drain-source in contrast to input, output and reverse transfer capacitance as they appear in datasheets. Resistors in parallel with capacitors are set to 0.5 M Ω in order to improve model convergence. The die also includes an internal gate resistance R_g and the body-diode is modeled with a standard spice diode D_1 . Changes in forward voltage from gate voltage and temperature are modeled through the voltage source E_{15} and R_{15} is added for convergence reasons. The package legs g , D , S_1 and S_2 are modeled as inductors with a series resistance. To model high-frequency AC-impedance, a resistor is added in parallel with L_d . The source legs also have a common inductance L_v modeling the inductive influence from the power circuitry on the gate circuitry.

2.4.1 Output Characteristics

The transfer characteristics are determined by the current source G_2 , which follows a variation of the Enz-Krummenacher-Vittoz MOSFET model, which uses a continuous function for calculating the current through different regions [19, 25].

$$i_d = g \cdot (v_{gs} - v_{th}) \cdot \left(\ln\left(1 + e^{(v_{gs} - v_{th})}\right)^2 - \ln\left(1 + e^{(v_{gs} - v_{th} - n \cdot v_{ds})}\right)^2 \right) \cdot (1 + L \cdot v_{ds}) \quad (2.7)$$

The drain current function shown in Eq. 2.7 uses 3 parameters, given as g , n and L , which themselves vary with temperature and gate voltage based on fitted 5th degree polynomials. g is a gain which primarily effects the slope in the ohmic region, n primarily changes the transition point from the ohmic to the active region and L effects the slope in the active region.

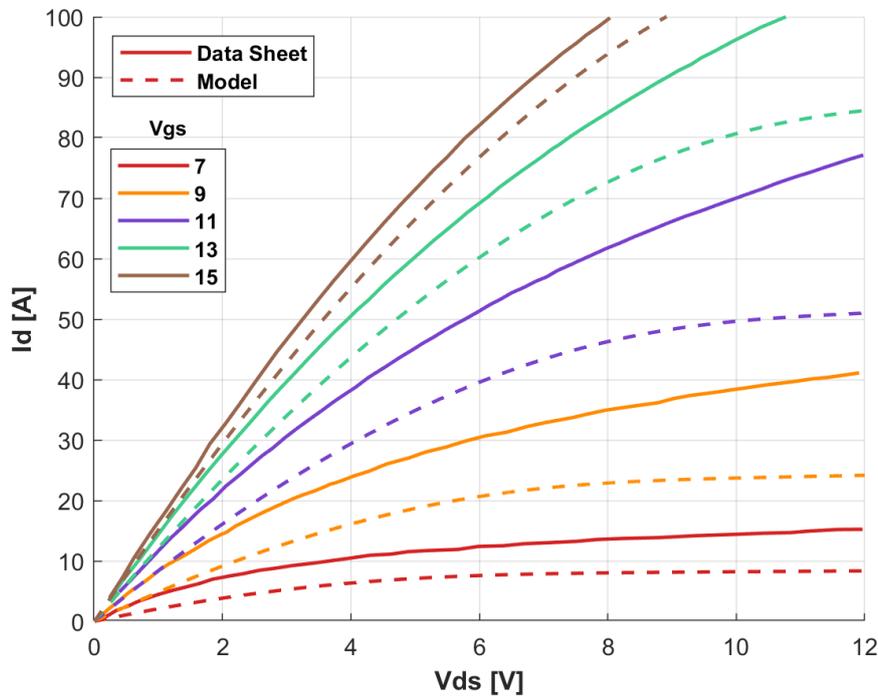


Figure 2.8: Comparison of output characteristics from data sheet and the provided model from Wolfspeed for C3M0060065K at 25 °C.

The static characteristics of a Wolfspeed MOSFET were not measured, so an example model provided by Wolfspeed for C3M0060065K is compared with the datasheet values at 25 °C junction temperature in Figure 2.8. The model correlates best with the datasheet values at $v_{gs} = 15V$, and the correlation reduces as the gate voltage is reduced. This shows that the output characteristics of the C3M model have the same complex behavior expected from a real model where the slope in the active region is not constant. This complexity comes at the cost of utilizing highly customized parameter functions, which increases calculation time significantly and are hard to fit from DPT data for a general model.

2.4.2 Capacitance Characteristics

The non-linear behavior of C_{gd} and C_{ds} is modeled using a dependent current source in parallel with a small capacitor. The capacitors measure gradients while the current sources emulate the change in capacitance as function of voltage. C_{gd} is modeled as function of v_{gd} using Eq. 2.8.

$$i_{C_{gd}} = k_1 \cdot \left((1 + v_{gd}) \cdot \left(1 + \frac{k_a}{2} \cdot \left(1 + \tanh(k_b \cdot v_{gd} - k_c) \right) \right) \right)^{-k_2} \cdot \frac{d}{dt} v_{gd} \quad (2.8)$$

This equation uses five coefficients, and the voltage v_{gd} used to calculate the capacitor value, is limited from 0.1 to 270 volts when used in the equation, to ensure the equation is only applied in a valid range. For voltages outside the range, the capacitor value is assumed to be at the limit. While k_a , k_b and k_c primarily control the shape of the function, k_1 acts as a gain on the capacitance and k_2 primarily controls at what value the capacitance plateaus at high voltage. The equation for C_{ds} is simpler, shown in Eq. 2.9.

$$i_{C_{ds}} = \left(\frac{C_{j0}}{1 + v_{ds} \cdot \frac{1}{V_j}} \right)^M \cdot \frac{d}{dt} v_{ds} \quad (2.9)$$

This equation only uses three coefficients to emulate the varying capacitance and V_{ds} is limited to the same range of 0.1 to 270 volts. C_{j0} is the initial value at 0 V, V_j primarily controls the slope of the function at high voltages and M primarily controls the slope at low voltages. Figure 2.9 compares the capacitance characteristics from the C3M0060065K datasheet with the effective capacitance, calculated by Eq. 2.8 and Eq. 2.9 using the parameter values given in the C3M model.

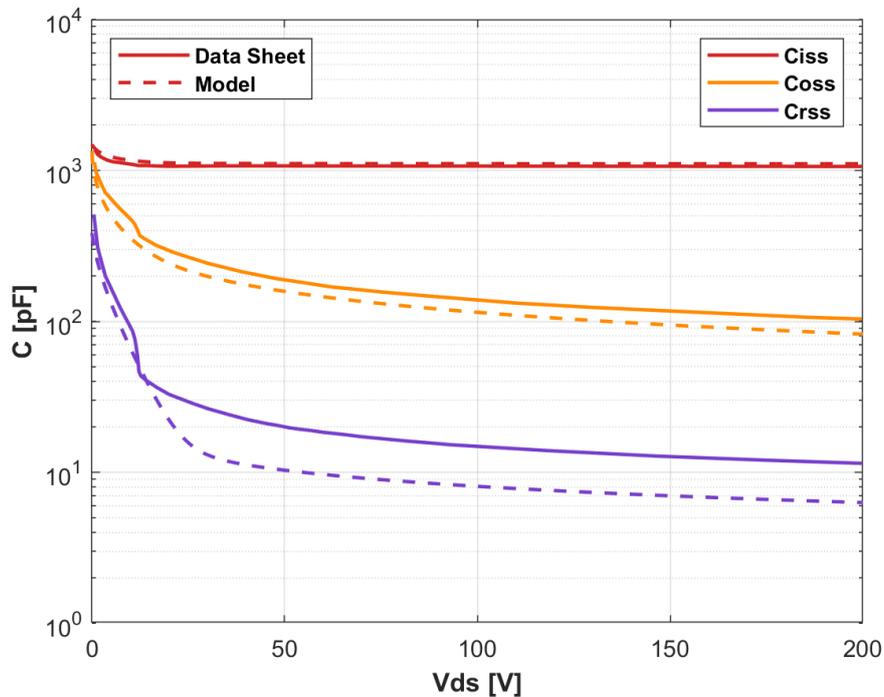


Figure 2.9: Comparison of capacitances from data sheet and the provided model from Wolfspeed for C3M0060065K.

The capacitors in the model correlate with the datasheet, with a difference of at most a few pF. This shows that the equations are good at approximating the tendency of the voltage dependency for the internal capacitors. These equations can therefore also be used for different MOSFET devices where the internal capacitors share similar tendencies, if the parameters are tuned.

5

2.4.3 C3M Reference DPT Simulation

To determine the impact of adding parasitics to the simulation model, a simple DPT simulation model is constructed. The model consists of the C3M model, a load inductor, a DC-link capacitor and shunt resistors for current measurements. The simulated waveforms are compared with experimental data from a DPT conducted on a DPT setup from Keysight, shown in Figure 2.10. The setup for the experiment is elaborated in App. A.

10

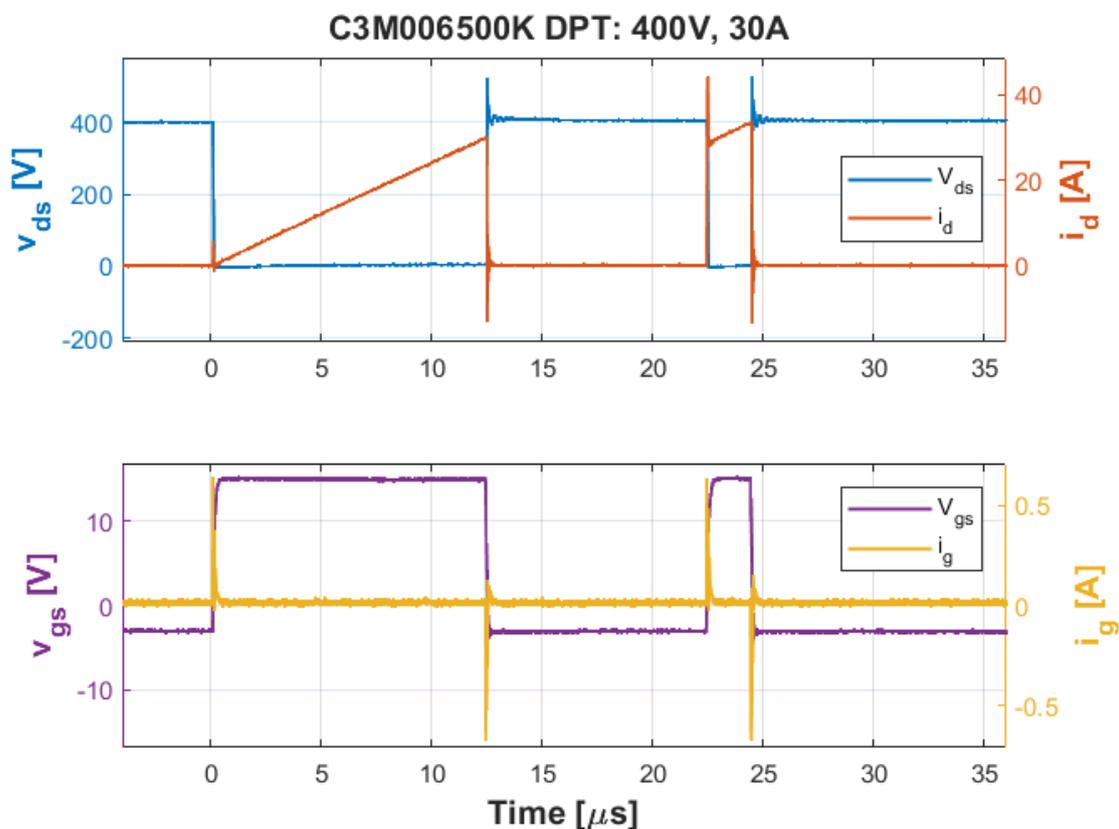


Figure 2.10: Measured DPT switching waveforms for the C3M006500K at $I_L = 30$ A and $V_{ds} = 400$ V.

A diagram of the simple PSpice simulation circuit is shown in Figure 2.11. This reflects a model which is often used for initial calculations of rise and fall times and energy losses when deciding which MOSFET to use in a design. The load inductor and DC-link capacitor are set according to their rated value of $120 \mu\text{H}$ and 115 nF respectively. The test is conducted on the C3M0060065K with V_{dc} set at 400 V and a load current of 30 A at the switching instant. The waveforms of the simulation is compared with the experimental results in Figure 2.12.

15

20

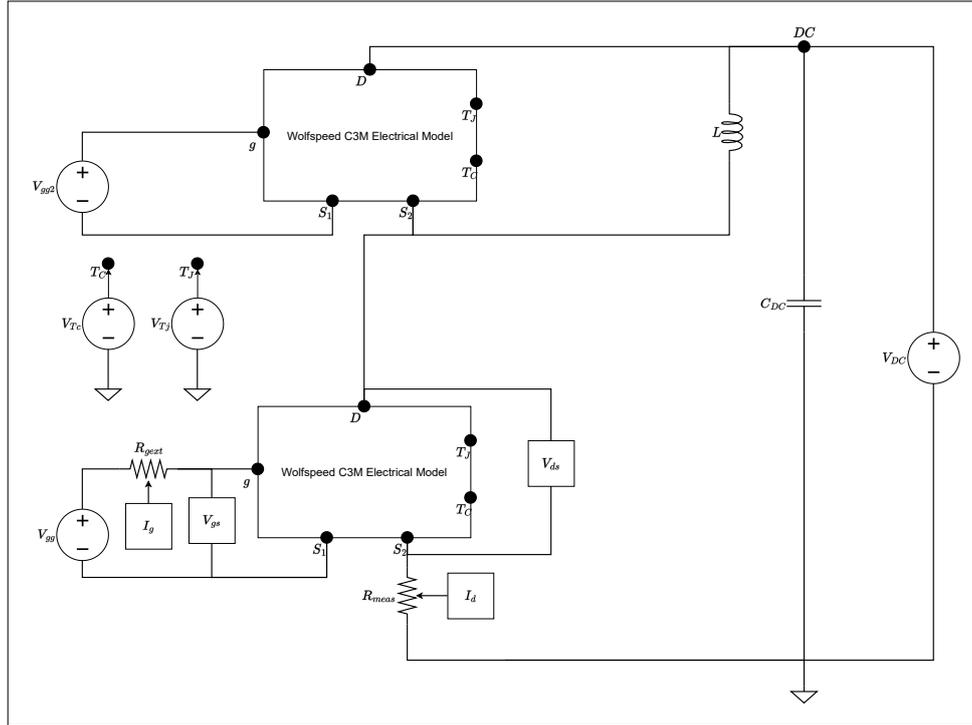


Figure 2.11: A circuit model of a double pulse test without circuit board parasitic elements, utilizing the C3M model provided by Wolfspeed.

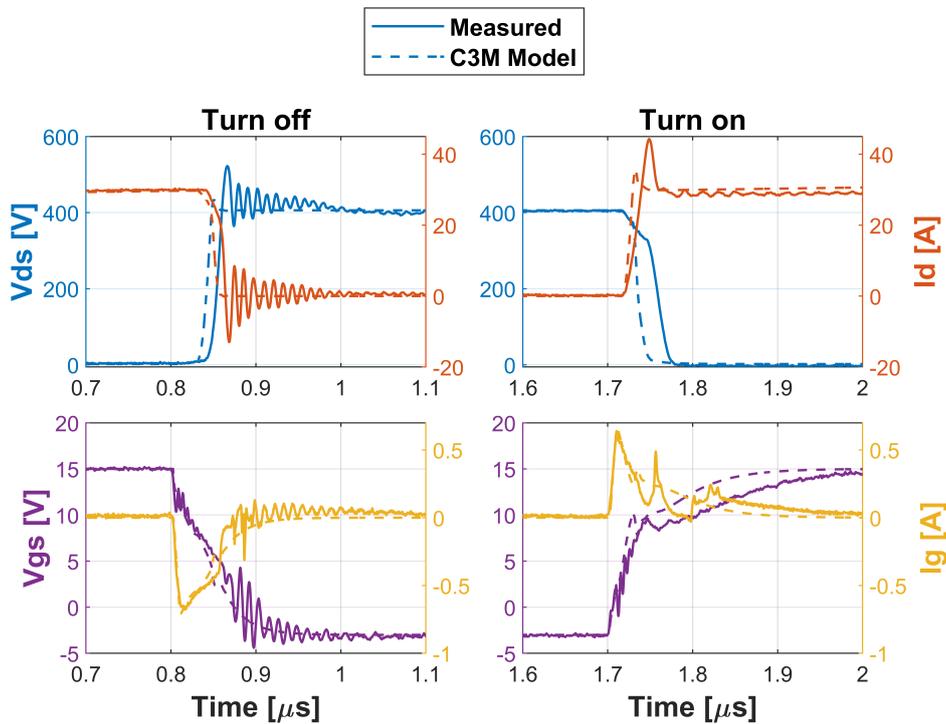


Figure 2.12: Comparison of measured and simulated DPT waveforms at 30 A and 400 V. Results are obtained using the default parameters of the C3M0060065K model with the DPT circuit shown in Fig. 2.11.

In Figure 2.12, v_{ds} rises prematurely by 15 ns during turn off and has an overshoot peak of 434 V compared with the experimental results which reach a peak of 522 V, this is reflected in i_d which falls sooner and faster. The main reason for this can be seen in

v_{gs} , which decreases too fast around $0.85 \mu s$, causing the threshold voltage to be reached sooner in the simulation than observed in the experiment. Since i_g is closely matched up until this point, this can be interpreted as the input capacitance being too low in the simulation. During turn on, i_d rises at the same time for both simulation and experiment, but the slope is steeper for the simulation. This, in combination with a lower overshoot in i_d results in v_{ds} falling sooner in the simulation. The overshoot is controlled by multiple parameters such as parasitic capacitance in the load inductor, the reverse recovery time of the body-diode and the drain inductance. The slope of i_d during turn on, is mainly controlled by the source inductance and the output characteristics of the MOSFET.

2.4.4 Switching Energies and EMI

The consequence of the simulation inaccuracies is that the maximum voltage and current is higher than expected, which means that component ratings will be harder to determine when designing a circuit. The rise and fall time of i_d and v_{ds} also affects the calculated switching losses of the circuit considerably. Since load current have a significant impact on e.g. reverse recovery, the circuit is also simulated for 10 and 20 A. Fig. 2.13 compares the switching losses of the simulation model and measured data for 10, 20 and 30 A.

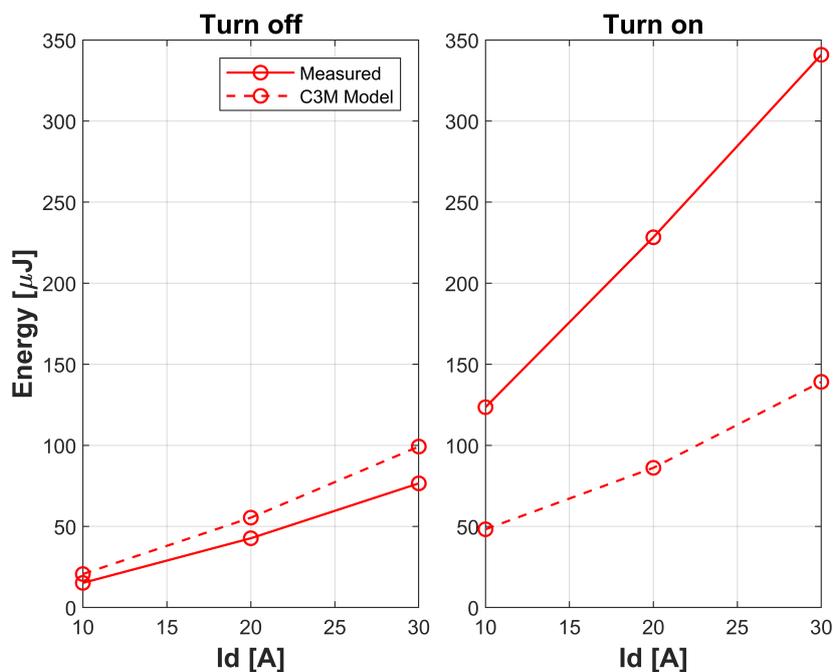


Figure 2.13: Comparison of switching energies during turn on and turn off for experimental results and the simulation model data shown in Fig. 2.12.

The turn off losses of the model follow the upward trend of the measured data but with an error increasing from 10-25%. During turn on, the losses are 60% lower in the simulation for all three operating points. The biggest error can be observed for 30 A, where the model predicts losses of $140 \mu J$ while the experiment shows losses around $345 \mu J$. The frequency spectrum of the simulation is compared with measured data at 30 A and 400 V in Fig. 2.14.

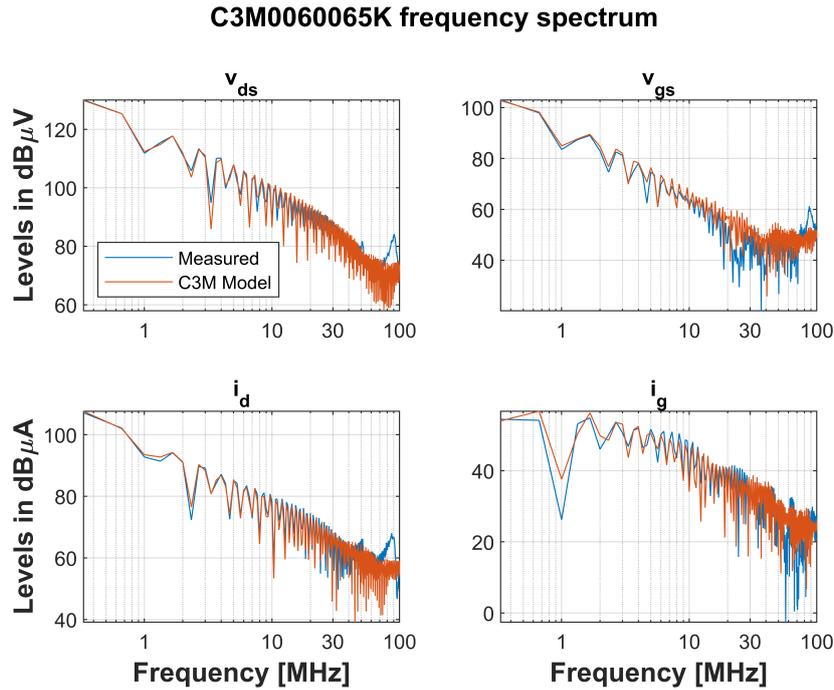


Figure 2.14: Measured and simulated frequency spectrum below 100 MHz, at 30 A and 400 V. Results are obtained using the default parameters of the C3M0060065K model with the DPT circuit shown in Fig. 2.11.

When inspecting v_{ds} and i_d , it can be seen that they fit well up to 30 MHz and only deviates slightly around 50 MHz and 90 MHz where peaks occur. Similarly, i_g only deviates with a few dB μ A up to 100 MHz and v_{gs} fits to around 20 MHz. For modeling conducted EMI, it is sufficient that the power loop signals fits up to 30 MHz to comply with EMC standards. The basic C3M model is therefore sufficient for EMI modeling, but it falls short when switching energy and overshoots are considered.

It is therefore important to account for circuit parasitics in a simulation, when trying to optimize a circuit design. Without an accurate simulation model, changes to a circuit design can have effects which are hard to predict without experimentally testing.

2.5 Proposed PSpice Model for Parasitic Extraction

This section will present a generalized Spice model, based on the C3M model, which can be used to extract parasitic circuit elements. The model is implemented in PSpice and parameters are modified using MatLab. The circuit diagram is shown in Fig. 2.15 and the implementation process is documented in App. B.

5

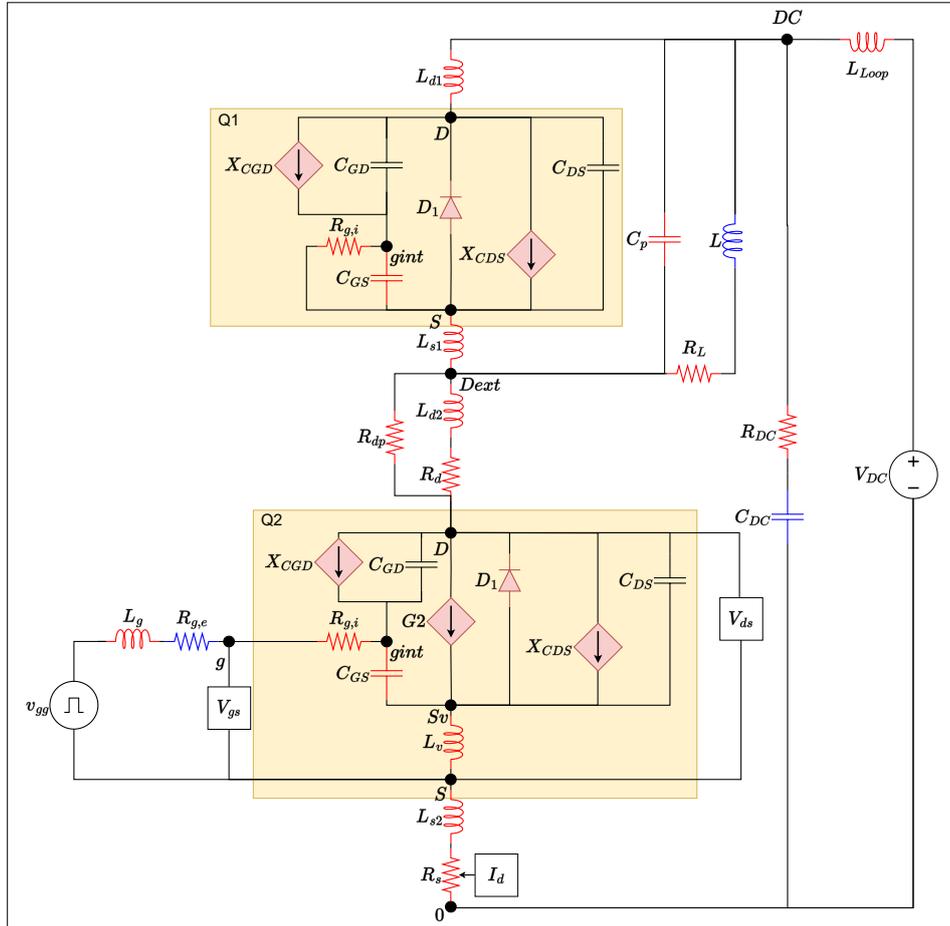


Figure 2.15: Proposed DPT circuit model, based on the C3M MOSFET model from Fig. 2.7. Components marked in blue are known values from the test setup and uncolored are constant values. Components marked in red are parameters which can be optimized.

Due to the complexity and number of parasitics in the model, some simplifications have been made. This is done to reduce the number of optimization variables and to make it more compatible with devices from other manufacturers. First, Q_1 and Q_2 are based on the C3M model and usually the same device in half-bridge configuration. The parameters in the highlighted area are therefore only optimized once and used for both devices. The leg inductances from the TO-247-4 package are combined with corresponding trace inductances since they impact the signal equally. The value of the capacitor C_{gd} is optimized by adjusting k_1 and k_2 from eq. 2.8 and C_{ds} is optimized by adjusting C_{j0} , V_j and M from eq. 2.9. The bodydiode reverse recovery is optimized by adjusting the transient time of the default PSpice diode model. For the highside device the mutual inductance L_v is neglected together with the current source in order to save calculation time, since the voltage from gate to source will be less than the threshold voltage and therefore neither have much effect on the overall system.

10

15

The drain current in the C3M model is modeled through three parameters, which themselves are curve fitted through 5th degree polynomials to temperature and gate voltage. This becomes too complicated to fit through optimization of a single double pulse test for a general model. The simpler approximation of eq. (2.1) on page 7 is therefore used to model the output characteristics. This can be optimized by adjusting the transconductance K and threshold voltage V_{th} as constant values.

The load inductor is modeled with a high frequency model, with an equivalent series resistance R_L and a parasitic capacitor C_p in parallel.

3. Parasitic Component Extraction

The extraction of parasitic circuit elements involves an iterative design process, where multiple trial designs are analyzed until an acceptable design is achieved. This can be manually done by a design engineer based on experience, intuition or trial and error. In the optimum design process, the analysis is automated and trial designs are analyzed and compared to determine which is the best design [28]. This chapter will give an discuss the optimization process and present different optimization methods. An algorithm is then selected which is able to iterate towards a set of parameters which fits the measured waveforms.

3.1 Parameter Optimization Process

In PCB design, the terms "best" and "acceptable" design are relative and based on in-company or industry standards with respect to e.g. efficiency, reliability or cost. In this project, trial designs refer to sets of parameters which is used in the simulation model of Fig. 2.15. In this case, an acceptable design is a set of parameters which is able to model EMI and power losses within the constraints of reasonable values. Thus, the best design is the set of parameters which is most accurately able to represent the physical system. The optimum design process follows a series of steps, presented in Fig. 3.1.

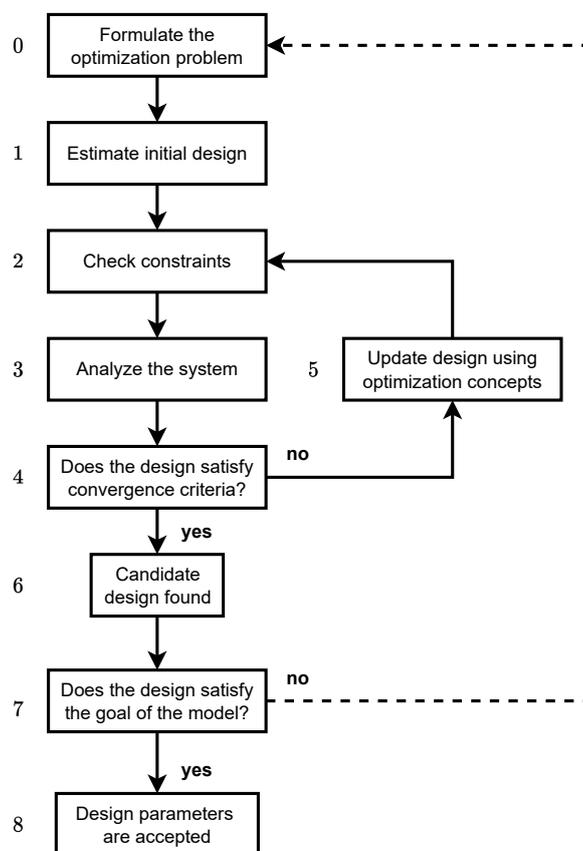


Figure 3.1: Flowchart of optimum design method. Based on [28].

The individual steps depend on the choice of optimization algorithm, data processing and simulation model design. However, the key features of each step in the process are similar and are as follows.

- 5 **0.** In step 0, the optimization problem is partly formulated by the simulation model and partly by defining a fitness function and parameter constraints. The simulation model should incorporate enough parameters to model desired effects and the fitness function should guide the optimization algorithm towards the best solution. Constraints are then imposed to limit parameters to realistic values.
- 10 **1.** The initial design generates a set of parameters which is used for the first iteration. The initial design is usually randomly generated within the design constraints and the number of initial designs can vary from one to several hundred, depending on the optimization algorithm.
- 2.** After a design has been updated, it is checked to see if the values respect the parameter boundaries defined in step 0.
- 15 **3.** Step 3 signifies simulation and necessary data processing such as interpolation, truncation, mapping and signal arithmetic. In this step, the complex simulation waveforms are decomposed to a form which the optimization algorithm can interpret as an improvement or decrease with respect to previous iterations.
- 4.** The waveform is then compared to a set of convergence criteria, which is typically a margin of error or a limit to the maximum number of iterations.
- 20 **5.** If the criteria has not been satisfied, the trial design is updated using the update rules of the selected optimization method. These rules are unique to the optimization algorithm and can also vary significantly based on the type of optimization problem.
- 6.** If the convergence criteria is satisfied, the optimization process stops and the best design is obtained.
- 25 **7.** The best design is then analyzed and it is evaluated if the result satisfies the goal of the model. If the simulation model lack the necessary components to simulate different effects, it has little hope of modeling EMI and power losses. Likewise, it is possible that solutions are found which match the waveform, but have unrealistic values such as 10 mH gate inductance. In both cases, the problem should be reformulated and the process restarted. The dashed line indicate that the simulation model is manually edited to include additional effects or adjust constraints, based on the waveform of the best design.
- 30 **8.** If a satisfactory result is obtained with reasonable values, it is assumed that the optimization process has determined the correct parasitics.
- 35

From Fig. 3.1, it is clear that the difference between manual and optimum design lies in step 5 where the design is updated based on a set of rules. In low dimensions, simpler methods can be used to find optimum designs. However, in higher dimensions the optimization problem becomes difficult to solve and often involves nonlinear dependencies.

40 For this reason, numerical methods are often used since they are able to handle many variables and find optimum in complex problems involving nonlinear equations without the need for analytically solving the optimization problem. This is a difficult task for complex problems and it does not necessarily translate to different test setups.

3.2 Numerical Search Methods

The choice of numerical search method depends on how the problem is formulated and it should be selected based on the following considerations [28].

1. Are the design variables continuous, discrete or integer?
2. Are the problem functions continuous and differentiable?
3. Can the derivatives of all function variables be calculated easily?

If the answer to all three questions is yes, a gradient-based method can be used or if the derivative is difficult to compute, a derivative-free method can approximate derivatives [28]. In this report, the fundamentally nonlinear behavior of the MOSFET makes it difficult to obtain the derivatives since it is governed by complex high dimensional functions. It is therefore impractical to apply gradient-based or derivative-free search methods. If derivatives are unavailable, direct or nature-inspired search methods can be used instead. Direct search methods assume that the function is continuous and differentiable, but derivatives are not used and they only evaluate if a better solution is obtained from one iteration to another. Nature-inspired search methods, or evolutionary algorithms (EA), are based on natural phenomena and can use any problem function in their solution process. Like direct search methods, EAs do not require derivatives in their search process. Their inherently stochastic behavior enables them to search through a large number of possible solutions and they tend to converge to the global minimum, which is not always the case for other methods. They can generally be used to solve all kinds of problems, but the large amount of randomness can increase computation time compared to other methods [28]. In this report, a combination of direct search and evolutionary algorithms are used.

3.2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is an EA that mimics the social behavior of fish schools or bird flocks. In a swarm, each individual makes decisions according to its own knowledge and the knowledge of the group. In the case of a bird flock, each individual will search for food until a source is located by a member of the group. The flock is then drawn to the location of the food, but they will continue their search if a better source is found on the way. This is the basic principles of PSO and it has been successfully used to solve difficult problems such as multi-objective optimization and neural network training etc. Like most EAs, PSO starts with a large population of randomly generated solution within defined parameter boundaries. An optimum solution is searched for by updating the current generation through an iterative process. Although application areas vary, a standard terminology is used to describe the process [28].

- The *swarm* is used to describe all solutions in a population and the *swarm size* N_p is the number of particles in the swarm.
- The *particle* x^i is used to describe an individual in the swarm. In this report, a particle describes a set of simulation parameters.
- The *particle position* describes the coordinates of a particle. This corresponds to the parameter values of a given solution.
- The *particle velocity* v^i is the rate of change of each particle. This determines how much a parameter changes between each iteration.

- The *swarm leader* x_g is the particle with the best solution for a given iteration. Additionally, x_p^i is used to denote the best known location for an individual particle where $x_g = x_p^i$ for the swarm leader. At the point of convergence, the swarm leaders location is considered the global minimum.

5 3.2.2 PSO Algorithm

The PSO algorithm consist of changing the particles velocity, accelerating the particles towards the best known position. Each iteration the velocity is updated according to Eq.[28].

$$v^{i,k+1} = \underbrace{\omega \cdot v^{i,k}}_{\text{Inertia}} + \underbrace{r_p c_p \cdot (x_p^{i,k} - x^{i,k})}_{\text{Cognitive}} + \underbrace{r_g c_g \cdot (x_g^k - x^{i,k})}_{\text{Social}}; \quad i = 1 \text{ to } N_p \quad (3.1)$$

Where k is the current iteration, r is a random number between 0 and 1, ω is inertia and c_g and c_p are social and cognitive coefficients respectively. The social and cognitive terms updates velocity according to the distance between a particles current position, the location of its own previous best and the location of the leader. The coefficient c_g improves convergence and c_p improves exploration and are typically set to $c_p = c_g = 2$ [29]. After velocity is calculated, the positions are updated.

$$x^{i,k+1} = v^{i,k+1} + x^{i,k}; \quad i = 1 \text{ to } N_p \quad (3.2)$$

The boundary conditions are then checked to see if the new positions are within the parameter constraints. The function is then evaluated and each solution is compared to the previous best solutions and updated if necessary.

$$\text{If } f(x^{i,k+1}) \leq f(x^{i,k}) \implies x_p^{i,k+1} = x^{i,k+1}; \quad i = 1 \text{ to } N_p \quad (3.3)$$

If an improvement is made, it is checked if the swarm leader should be updated as well.

$$\text{If } f(x_p^{i,k+1}) \leq f(x_g^k) \implies x_g^{k+1} = x_p^{i,k+1}; \quad i = 1 \text{ to } N_p \quad (3.4)$$

The update process is illustrated in Fig. 3.2 for the first two iterations of a single particle with $\omega = c_p = c_g = 1$. Since each term in Eq. 3.1 produces a vector, they can simply be added together to determine the new position. By multiplying with r_p and r_g , the new particle has a chance of landing anywhere in the highlighted area. If the current position is the individual best, the cognitive term is zero as seen in (a) and (b). If the function value of the new position is worse than the previous, the cognitive term is non-zero as seen in (c) and (d). The inertia term is therefore necessary to maintain momentum as the function value decreases since PSO does not consider the gradient of the function. Through this process, the particles are approaching increasingly better solutions while checking surrounding locations on the way.

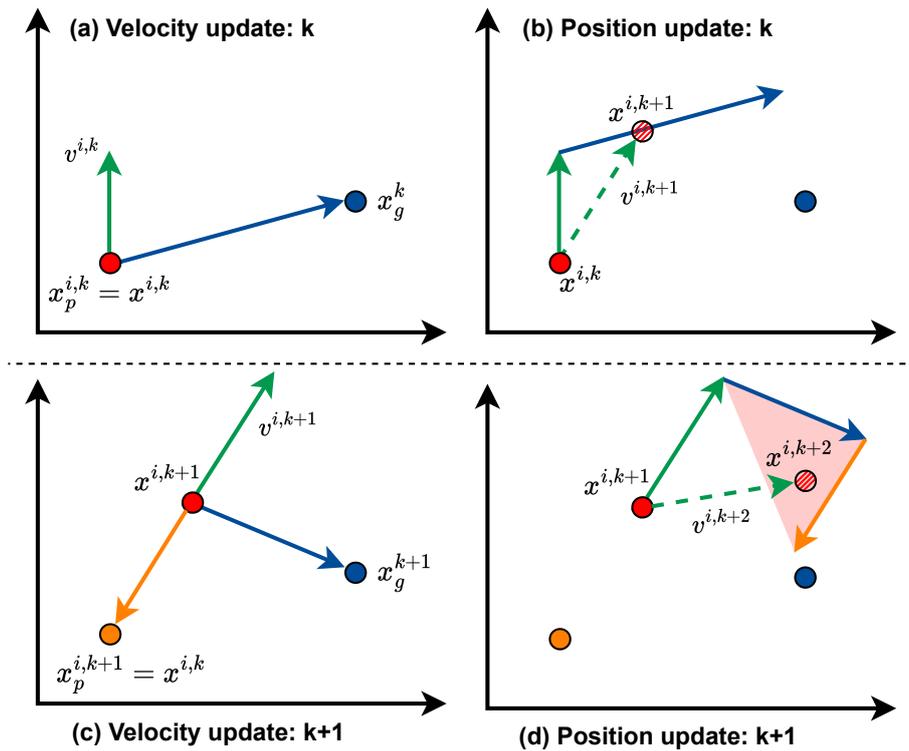


Figure 3.2: Illustration of PSO position and velocity update principles from Eq. 3.1 and Eq. 3.2.

From Fig. 3.2 it is evident that PSO is able to cover a wide area with little knowledge of the objective function. The main disadvantage is evident in the later iterations as the search space narrows. Since the swarm size is constant from start to end, many simulations are executed with very similar parameters in the end. This is not inherently bad, since small adjustments can lead to improvements and they are necessary to find global minimum. However, in each iteration, new solutions are generated pseudo randomly and the search is not very intelligent. At this point improvements can still be made, but the stochastic behavior means that fine tuning parameters is not computationally efficient. A number of PSO variants have been developed which dynamically vary the coefficients to obtain better results but satisfactory results can often be obtained with basic implementation [30]. For this reason, PSO is paired with another direct search method called pattern search (PS).

5

10

3.2.3 Pattern Search Algorithm

PS is a direct search method that starts with an initial solution and uses a sequence of points to approach the global minimum. At each step, a mesh of points is generated around the current best solution, called the base point. The function value of all mesh points are computed in a process called *polling* which is illustrated in Fig. 3.3. If the function value of one of the mesh points is smaller than the base point, the polling is successful and the base point is updated as seen in Fig. 3.3 (a). A set of mesh points is then generated, originating from the new base point. After a successful poll, the distance to the new mesh points are increased, typically by a factor of 2. If the poll is unsuccessful, the distance is halved and the polling process is restarted from the same point as seen in Fig. 3.3 (b). The process is terminated after a series of unsuccessful polls have reduced the distance to the new mesh points below a set threshold [31].

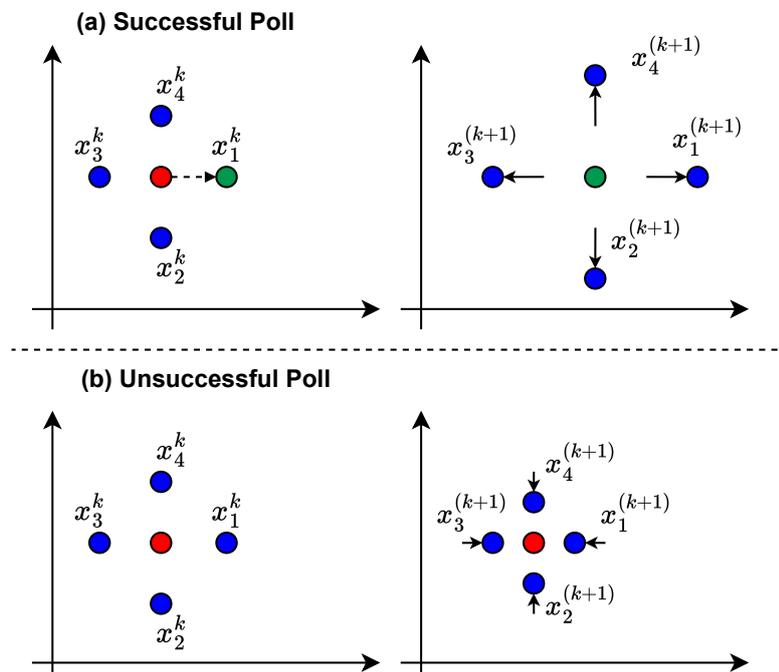


Figure 3.3: Illustration of the PS algorithm polling process in 2-dimensional space.

When initial conditions are good, PS is more efficient at making micro-adjustments than PSO. However, the algorithm requires that a sequence of points exists which can lead the initial solution to the global minimum. This is not guaranteed if a random initial solution is generated in the search space, due to the complexity of the optimization problem. For this reason, PSO is initially used to search for a good initial solution which can be used by PS.

3.3 Objective Function

The objective function is used by the optimization algorithm to evaluate how well the simulation result fits the measured data. It is often based on one or multiple difference metrics and weights are sometimes added to influence the outcome. A good objective function can determine if a set of parameters are closer to the correct solution than other combinations and an even better function will accomplish it in fewer iterations. In this case, the objective function is a multi-variable, non-linear function. Due to the complexity of the optimization problem, the objective function should be selected so it extracts as much information as possible from the switching waveforms. The two main factors that effect the accuracy and convergence of the algorithm is:

1. How the objective function is defined.
2. How the error is calculated.

The first point refer to how the data is processed and analyzed after the waveform is obtained. There are countless ways to transform and analyze data depending on the goal of the analysis [32]. In this case, many parasitics needs to be optimized simultaneously and some effects are only visible in e.g. the power loop or gate signals. It can therefore benefit convergence and accuracy if the signal is transformed and analyzed in a different domain e.g. using Fourier transform. On the other hand, parameters that affect gradients or steady-state may be easier to identify in the time domain.

The second point refer to signal arithmetic and how the error is calculated between the measured and simulated waveform. The simplest operations in signal arithmetic are point-by-point addition, subtraction, multiplication and addition. For instance, in time domain, point-by-point subtraction would compute the difference between two signals at a given point in time. In other cases, the two signals may be of similar shape but with an error in amplitude and point-by-point division can be used to determine the ratio. These operations have different use cases, but the major advantage of point-by-point arithmetic is that the computation time is fast. Other methods, such as local averaging, consider the values of neighboring points and may be more accurate at the cost of longer computation times [33]. Another point which should be considered is that the amplitude of e.g. v_{ds} and v_{gs} is significantly different. One can easily imagine that a small error in high amplitude signals, such as v_{ds} , will have greater influence on the total error than a large in error low amplitude signals such as v_{gs} . For this reason, it may be necessary to apply weights or normalize the signals so that important effects are not overlooked.

The following sections will discuss different analysis methods with the goal of defining a objective function that has the best chance of correctly identifying parasitics from the DPT waveform.

3.4 Time-Frequency Analysis

In time-frequency analysis there is a fundamental uncertainty principle that limits the ability to simultaneously attain high resolution in both time and frequency as shown in Fig. 3.4 (a) and (b). Conventionally, sampled signals are analyzed in either time or frequency domain, depending on which properties are of interest.

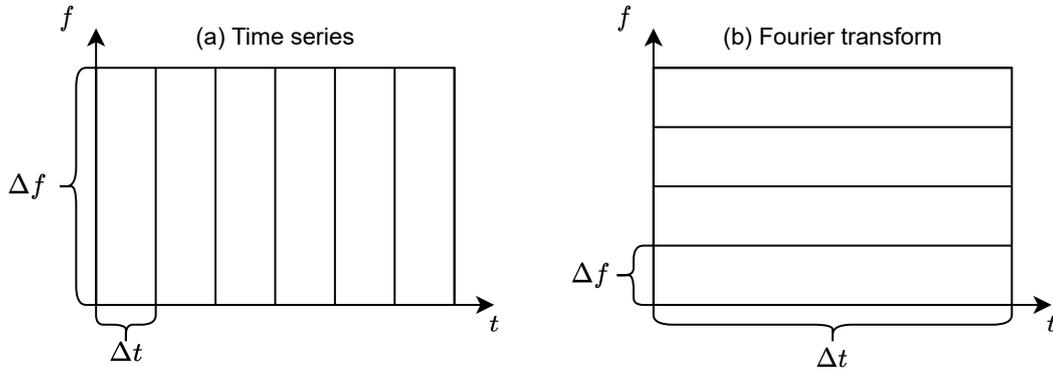


Figure 3.4: Illustration of uncertainty and resolution limitations in time-frequency analysis.

In an ideal case, with infinite instrument bandwidth, a time series perfectly captures the time domain characteristics but contains no information about the frequency content. In power electronics, the Fourier transform (FT) is used to characterize the frequency content over one or more switching cycles to determine EMI and harmonic content. The FT contains detailed information about the frequency content, but it contains no information about its location in time. For this reason, the FT is only truly able to characterize stationary periodic signals, since the stationary signal characteristics are independent of time [32]. Fig. 3.5 compares the time series and frequency spectrum of the C3M0060065K power loop signals, measured at 30 A and 400 V. The time series show that the parasitics affect the MOSFET at different periods in time, resulting in e.g. high frequency oscillations during turn-off and low frequency oscillations in v_{ds} when the device is off. Similarly, the frequency spectrum shows that the majority of the energy is concentrated in the low frequency range. It can also be seen that there are low amplitude peaks occurring at around 90 MHz. It would be reasonable to assume that the peaks are caused by the oscillations during turn-off since these are present in both signals. However, when inspecting the gate loop signals in Fig. 3.6, it is more difficult to pair the many oscillations to a specific resonance peak. This is both due to the lack of filtering in the gate loop, but also an effect of the non-linear capacitors that changes the resonant frequencies depending on v_{ds} .

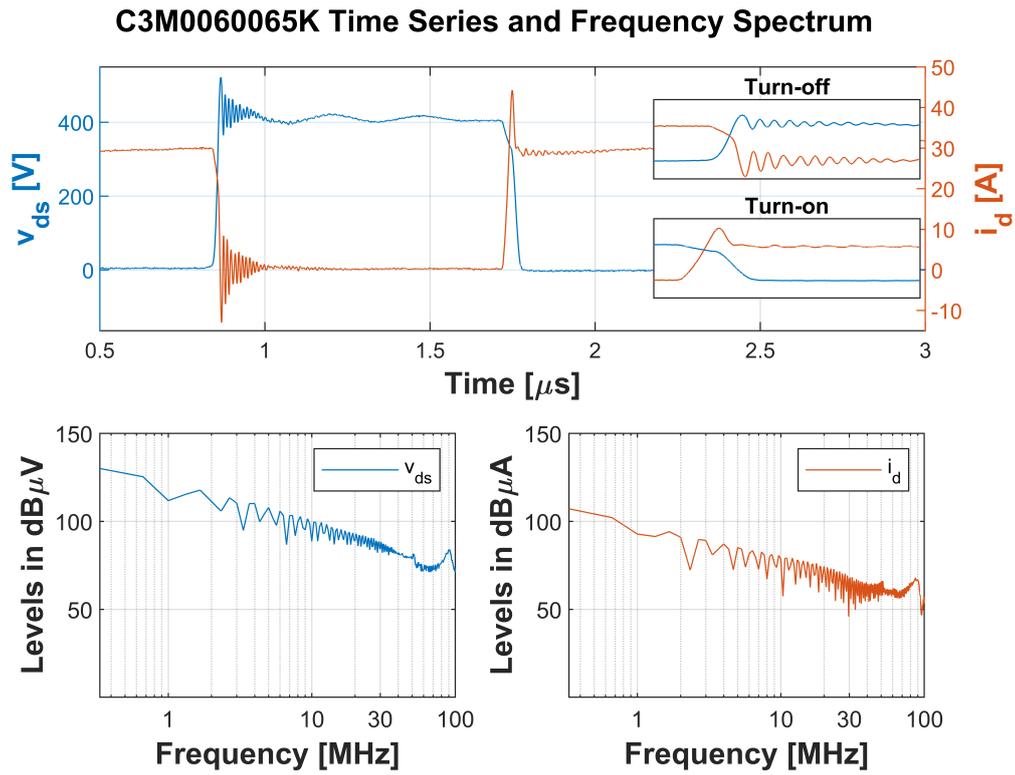


Figure 3.5: Time series and frequency spectrum for v_{ds} and i_d of C3M0060065K measured at 30 A and 400 V.

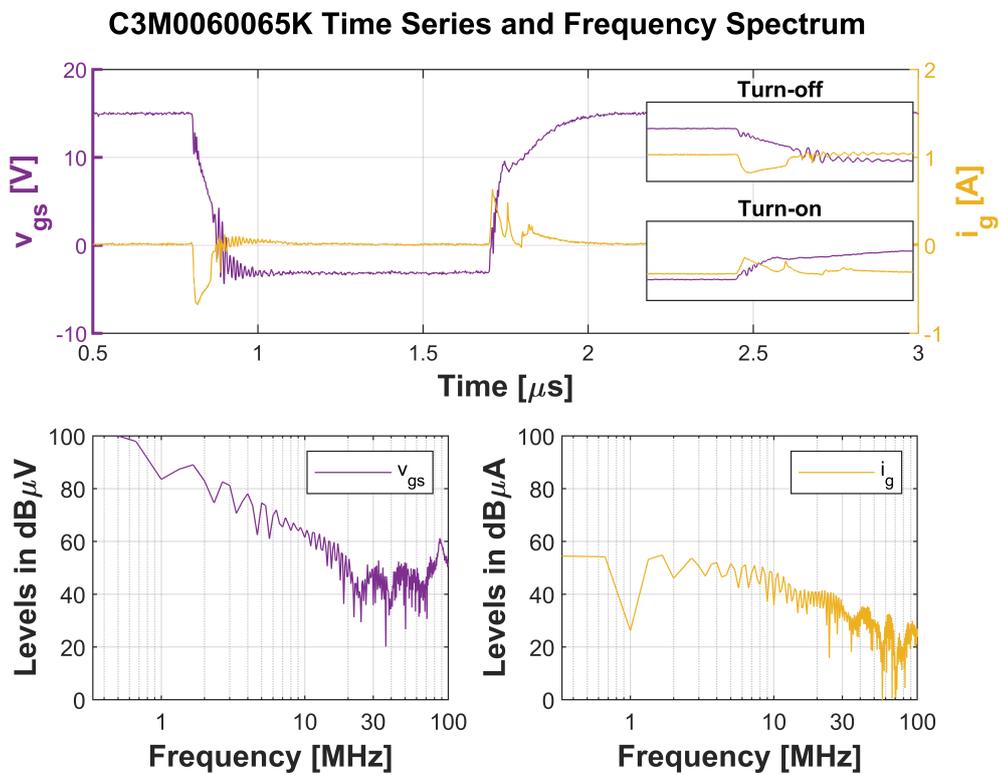


Figure 3.6: Time series and frequency spectrum for v_{gs} and i_g of C3M0060065K measured at 30 A and 400 V.

For these reasons, it is important to characterize the frequency content as it evolves in time in order to identify the parasitics correctly. For this, the short-time FT (STFT) also known as the Gabor transform can be used to obtain the spectrogram of a signal [32]. By computing the FFT in a moving window, it enables the localization of frequency content in time, as shown in Fig. 3.7 (a). A Gaussian function with a fixed window size is typically used and the window size is selected to capture the desired time and frequency content.

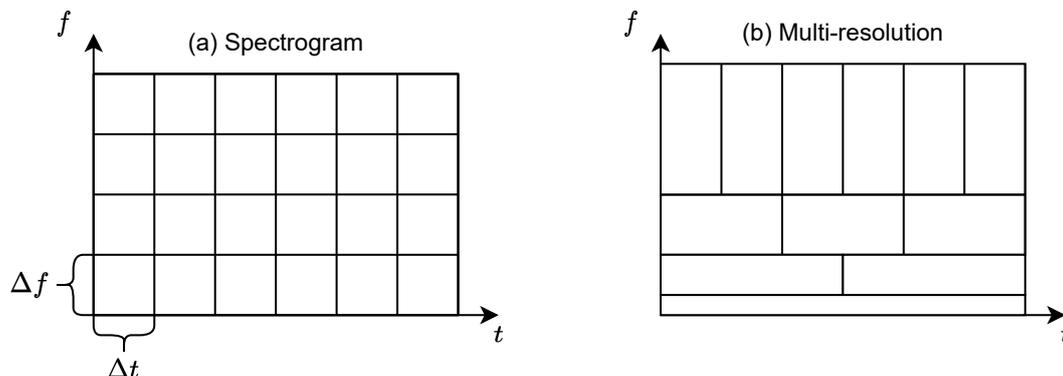


Figure 3.7: Illustration of uncertainty spectrogram and multi-resolution.

However, if the range of the frequency content is large, the STFT suffers from the same resolution problems as the FT, since low- and high-frequency content are captured in the same window. The uncertainty principle can partially be overcome by exploiting a multi-resolution decomposition as shown in Fig. 3.7 (b). Multi-resolution analysis (MRA) refer to breaking up a signal into multiple components, which can reconstruct the original signal when added back together [34]. The benefit of MRA, for MOSFET parameter extraction, is that the switching waveforms naturally decompose into interpretable components like noise, transients and low- and high-frequency oscillations. Like the STFT, MRA allows the localization of frequency content in time, but with with a variable window size. MRA is primarily accomplished using the wavelet transform (WT) which extends the concepts of the FT and has successfully been applied to multi-scale processes such as neuroscience and turbulence [32].

3.5 Wavelets and Multi-Resolution Analysis

The name wavelet means "small wave" and describe oscillating functions which has its energy concentrated in time. Like FT and STFT, WT involves the convolution of two functions where one is the signal of interest and the other is the oscillating function. Unlike sine and cosine waves, wavelets are finite in time which is the characteristic that enable analysis of time-invariant signals [35]. In order for a function to be defined as a wavelet, two criteria must be fulfilled [36]:

$$\int_{-\infty}^{\infty} f(t)dt = 0 \quad \wedge \quad \int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \quad (3.5)$$

The first criterion states that the integral, from negative infinity to infinity, must be equal to zero. This is fulfilled by wavelets as well as sine and cosine waves and is what enables identification of frequency content by convolution. The second criterion states that the function must be square-integrable. This indicate that the energy is finite and is what distinguished wavelets from other oscillating functions [35]. Due to the straightforwardness

of these criteria, numerous wavelet types have been invented [36]. Once a wavelet is constructed, it can then be scaled and translated in time to generate a family of functions as seen in Eq. 3.6[32]:

$$\psi_{a,b}(t) = \psi(t) \cdot \frac{1}{\sqrt{a}} \cdot \left(\frac{t-b}{a} \right) \tag{3.6}$$

Where $\psi(t)$ is known as the mother wavelet, a is a scaling parameter and b translates the function in time. This family of wavelets is what is used to characterize the frequency content in time. In theory, the mother wavelet could be any function as long as it satisfies the two criteria. Nonetheless, many wavelets, such as the "morlet wavelet", are simpler in nature and resemble sine or cosine waves. Fig. 3.8 illustrate how the morlet wavelet is derived from a cosine wave by using a Gaussian window function.

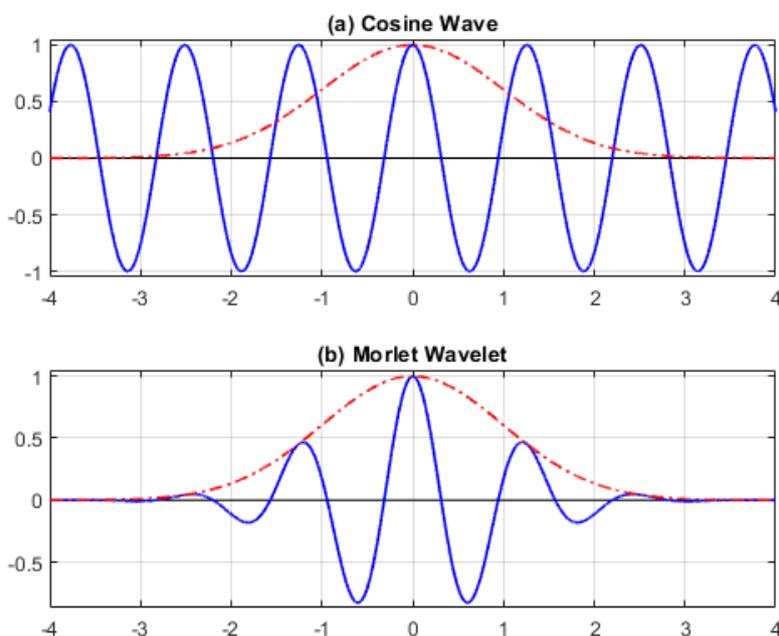


Figure 3.8: Comparison of the morlet wavelet and a cosine wave used in STFT.

From initial inspection, the cosine wave and morlet wavelet appear similar with an amplitude of one at the origin and equal oscillating frequency. The main difference comes from how the window function interacts with the oscillating function. Whereas the STFT computes the FT in a moving window of fixed size, the wavelet has integrated the window function in the mother wavelet. It is therefore easy to deduce that the mother wavelet of this particular wavelet is expressed as:

$$\psi(t) = e^{-t^2/2} \cdot \cos(5t) \tag{3.7}$$

From Eq. 3.7, it is evident that wavelets are simply an extension of concepts used in FT and STFT. In STFT, the window term would be fixed and a sweep is performed to determine the frequency content for each window in time. Here, the only degree of freedom is the width of the window which is fixed for all frequencies as seen in Fig. 3.7 (a). A wider window would increase the frequency resolution, but reduce time resolution since more frequencies are captured over a longer period in time. A window that spans the entire

signal length is equivalent to the FT, since it would include all frequencies contained in the signal. Similarly, a narrow window would have the opposite effect and an infinitely narrow window would have no frequency resolution. This is the main difference that separates the WT from STFT. Since the window is integrated directly in the mother wavelet it is scaled according to Eq. 3.6. For instance, one can imagine that a wavelet, which is stretched in time, would reduce the oscillating frequency, thus increasing frequency resolution. This is what enables localization of both low and high-frequency content.

3.5.1 DPT Time-Frequency Representation

The WT of a signal can be graphically represented in a scalogram which displays the magnitude of wavelet coefficients as a function of both time and frequency. Fig. 3.9 shows the scalogram of the C3M0060065K DPT waveform, measured at 30 A and 400 V. Compared to the frequency spectrum of Fig. 3.5 and Fig. 3.6, the scalogram displays the frequency content as it evolves in time. The bulk of the energy is located at low frequencies in the power loop, where the amplitude is orders of magnitude higher than the gate loop signals. When the transients occur at around 0.9 and 1.9 μs , there is a significant increase in high-frequency content. Here, the 90 MHz frequency peak is now visible in the power loop during turn-off and also in v_{gs} through coupling effects.

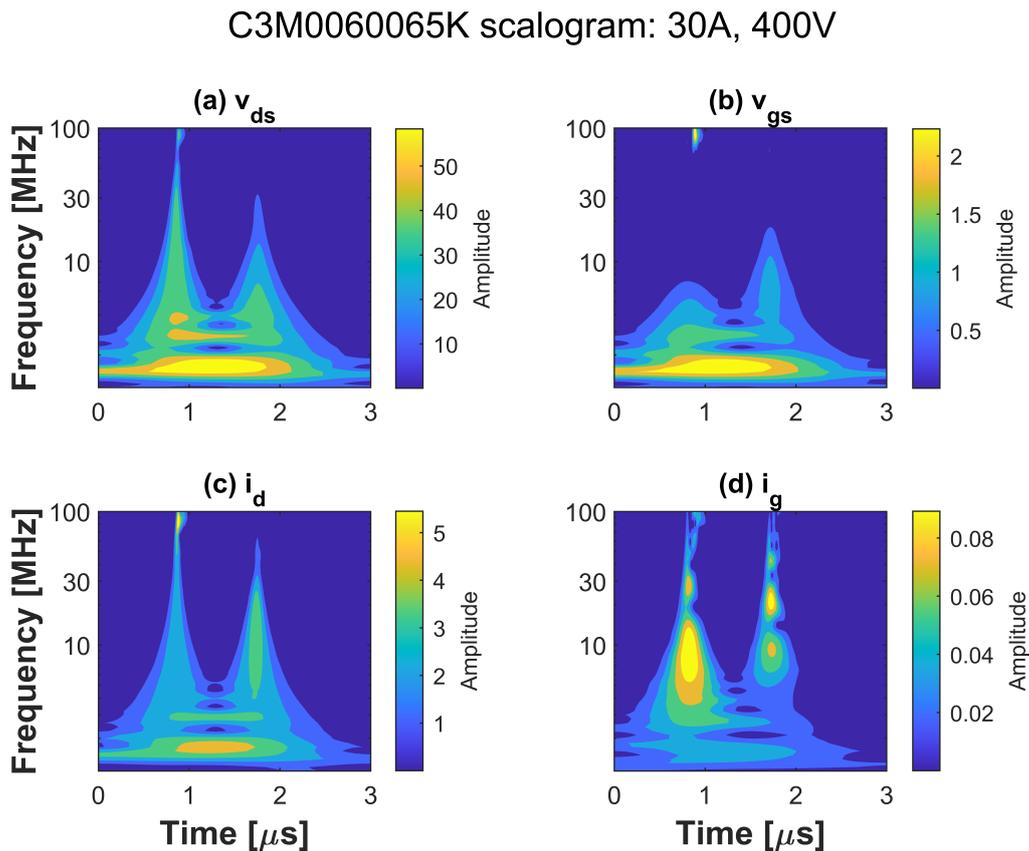


Figure 3.9: Scalogram for C3M0060065K measured at 30 A and 400 V.

It may appear as if the same information is contained in the frequency spectrum time series oscillations. However, the localization in time is key when the optimization algorithm select which set of parameters is closest to the correct solution.

3.6 Error Computation

As previously mentioned, there are many ways of comparing two sets of data where the simplest is point-by-point computations. The point-by-point difference between two functions at the i th point is simply calculated as the absolute value of the difference between the function values:

$$\varepsilon_i = \sqrt{(g_1(t_i) - g_2(t_i))^2} \quad (3.8)$$

The total error ε can then be found as the sum of all the errors as Eq. 3.9.

$$\varepsilon = \frac{1}{n} \cdot \sum_{i=1}^n \varepsilon_i, \quad n = i_{max} \quad (3.9)$$

From Eq. 3.8 it might seem like the simulation waveform with the lowest error in each point, will produce the lowest total error and therefore the best fit. However, this is not necessarily the case, since Eq. 3.8 only computes the difference in amplitude and is indifferent to the error in time or frequency. This problem is illustrated in Fig. 3.10 for two arbitrary 2d waveforms as they evolve in time.

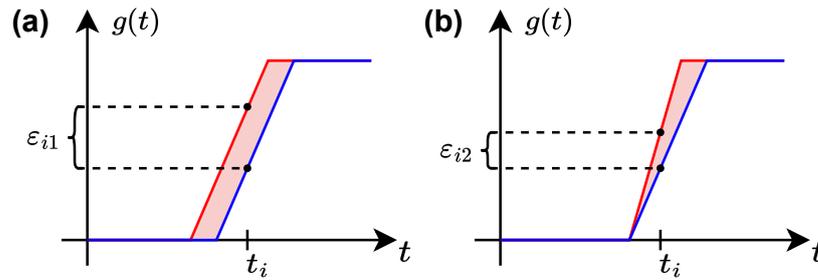


Figure 3.10: Definition of point-by-point error according to Eq. 3.8.

From a practical point of view, figure (a) is a better fit since the slope is correct and it will likely provide a better EMI and loss model. However, by the definition of Eq. 3.10, figure (b) is the best fit since $\varepsilon_{i2} < \varepsilon_{i1}$. If the error function only considered the frequency content, figure (a) chosen but it is also inaccurate due to the displacement in time. Simple point-by-point arithmetic is therefore not sufficient to compute the error as the waveforms evolve in both time and frequency.

For this reason, the Euclidean distance is used in this work to define the error function. The Euclidean distance between two points in the scalogram is computed as Eq. 3.10.

$$d(p_i, p_j) = \sqrt{(t_j - t_i)^2 + (f_j - f_i)^2 + (g(t_j, f_j) - g(t_i, f_i))^2} \quad (3.10)$$

Since the units of time, frequency and amplitude vary from 10^{-6} to 10^6 the frequency term dominates Eq. 3.10. The time and amplitude are therefore normalized between 0-1 as Eq. 3.11.

$$x_{i, nm} = x_i / \max(x) \quad (3.11)$$

The frequency is normalized between 0-1 in log scale as Eq. 3.12. This is necessary to even the frequency content which is distributed over a much larger range than time and amplitude.

$$f_{i,mm} = \log_{10}(f_i) / \log_{10}(\max(f)) \quad (3.12)$$

The distance between the i th simulation point and all points in the measured signal is then calculated and repeated for all points. The error at a given point is defined as the minimum Euclidean distance.

$$\varepsilon_i = \min(d(p_i, p_j)); \quad i = 1 \text{ to } n, \quad j = 1 \text{ to } n \quad (3.13)$$

The total error of each signal is then calculated as in Eq. 3.9. This is computed for v_{ds} , i_d and v_{gs} and the fitness is defined as Eq. 3.14.

$$\varepsilon_{tot} = \varepsilon_{v_{ds}} \cdot \varepsilon_{i_d} \cdot \varepsilon_{v_{gs}} \quad (3.14)$$

The selected objective function is more computationally taxing with $O(n^2)$ in Eq. 3.13 compared to $O(n)$ in Eq. 3.8. Additionally, the scalogram returns an $n \times m$ matrix for each signal which already increases the number of calculations compared to a time-series vector with n elements. Although computation time is low priority, the tools presented in this chapter require several measures to reduce the number of data points. However, the complex nature of the problem requires more complex tools, regardless of computational efficiency. By computing the Euclidean distance in the scalogram, the algorithm is able to correct time displacement and frequency content which greatly increases EMI and loss modeling.

4. Results

To experimentally validate the algorithm, several DPTs were conducted on a Keysight PD1500A Double-Pulse Tester given in App. A. First, tests were conducted on the C3M0060065K device at 400V and 10/20/30A from which the PSpice model is based. A similarly rated device from Infineon, IMZA65R048M1, was then tested at 10/15/20/25A. The experimental waveforms are used to fit the parameters of the circuit model at each current rating. A flowchart of the optimization process is shown in Fig. 4.1 and the MATLAB implementation is documented in App. C.

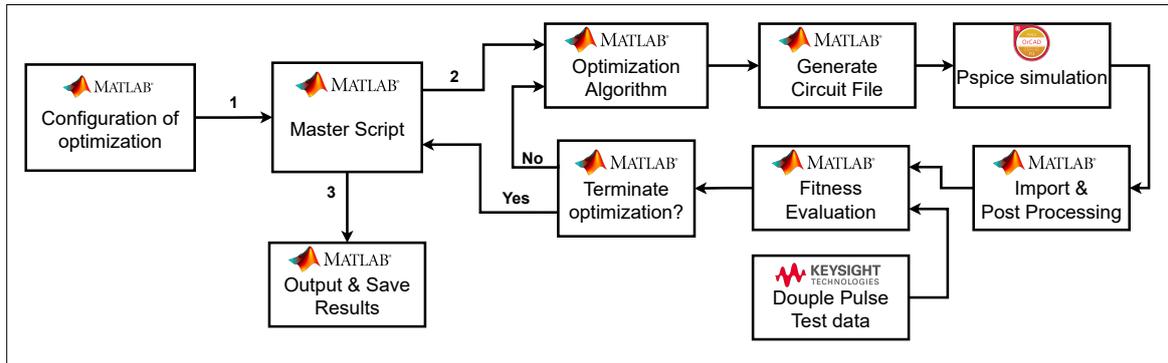


Figure 4.1: MATLAB/PSpice implementation of the automatic parasitic extraction algorithm.

The algorithm is validated by testing two different devices on the same setup. This way, the algorithm should find similar circuit parameters while the device parameters change. This will increase the chance that parasitics are identified correctly. The performance of the model and algorithm are evaluated according to three indicators.

1. Performance and robustness of the simulation model.

The performance of the PSpice model is compared with experimental waveforms and it is evaluated how well it models EMI, overshoot and switching energy. The robustness of the model is tested by simulating all fitted parameters at all current ratings. That is, parameters fitted at e.g. 20/30A are also simulated at 10 A and compared with measured waveforms at 10 A. The model should be robust enough so that it can be fitted at any current rating while still performing well at different operating points.

2. Consistency of the optimization algorithm.

The performance of the optimization algorithm is evaluated with respect to the error function and the consistency of the fitted parameters. The error function is evaluated by comparing the scalogram of the simulated and measured waveforms. For consistency, 5 optimization runs are conducted for each device at three different current ratings. Each run performed 100 iterations of PSO, with a swarm size of 200, followed by PS until convergence. The mean parameter values over the 5 runs are used for comparisons in the following sections.

3. Ability to identify parasitic circuit elements.

The algorithms ability to identify parasitics is evaluated by comparing the mean parameters values for both devices at all current ratings. Since the true values are unknown, the parameter spread across all tests are used as a metric of certainty.

⁵

The following sections compare the simulation results with experimental data at 400 V and 10/20/30A for the C3M0060065K and 10/20/25A for IMZA65R048M1H. Unless stated otherwise, simulation waveforms uses mean values of parameters fitted over 5 runs for a given current rating. The full list of parameters can be found in App. E.

4.1 C3M0060065K Performance

The switching waveforms for the fitted parameters are compared to measured data at 10/20/30A in Fig. 4.2, 4.3 and 4.4 respectively. The gate signals are largely unaffected by load current and the slopes follow the measured data. v_{gs} generally behaves like an RC-filter with an overshoot when the Miller plateau is reached during turn on. 5

When predicting the behavior at 10 A, 20 A and 30 A fits have a noticeable delay during turn off despite very similar v_{gs} and i_g . This suggest that the output characteristics behave less linearly in the active region at low current than the model we used, which can be expected when comparing it with the complexity of the C3M model. The ohmic region is probably modeled more accurately considering the agreement during turn on. 10

The overshoot in i_d is matched better for 20 A and 30 A, suggesting that algorithm evaluates correct solutions better at higher currents.

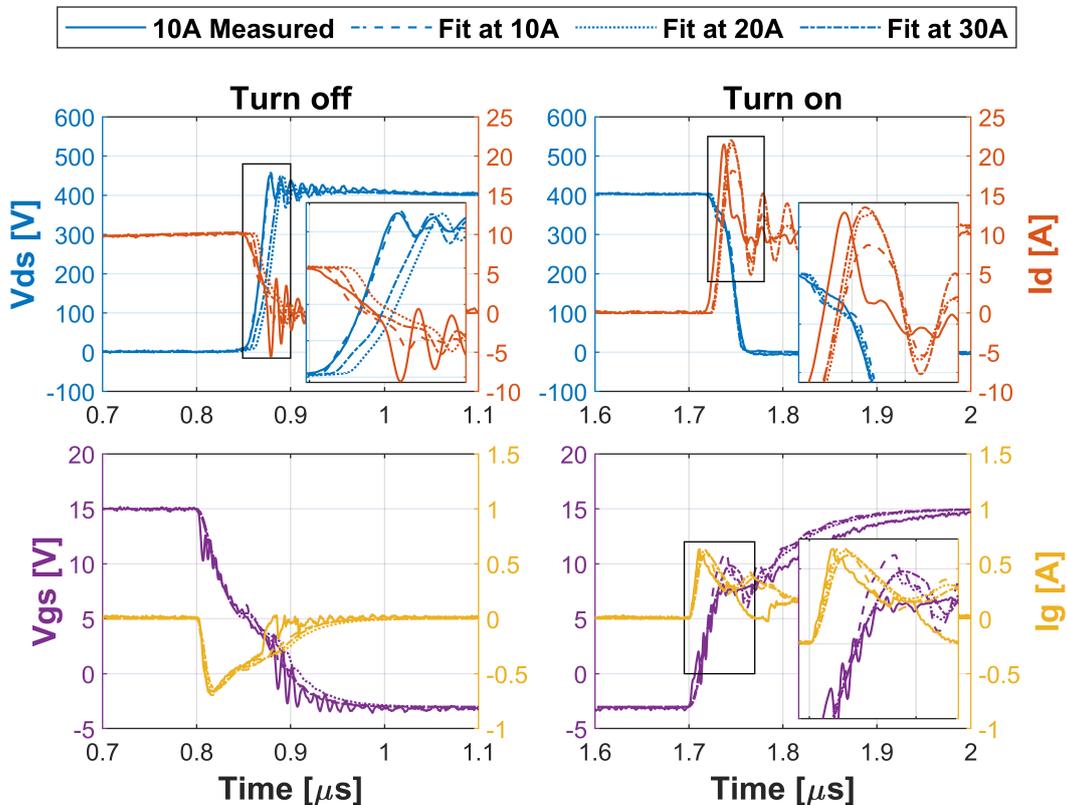


Figure 4.2: Measured and simulated waveforms at 10 A, fitted for C3M0060065K at 10/20/30A.

At the comparison for 20 A and 30 A, v_{ds} starts increasing prematurely during turn off for the 10 A fit. During the voltage rise, i_d drops too low for all fits, compared to measurements, which suggests that capacitance at the high-side MOSFET is fitted incorrectly. 15

However, it is unclear if it is caused by high C_p or the characteristics of C_{ds} at low voltage. During turn on, v_{ds} is noticeably more delayed for the 10 A fit which can be attributed to the displacement of the i_d peak cause by a lower di_d/dt . Again output characteristics of the active region are modeled imprecisely and the optimization tool is having a hard time 20

dealing with it.

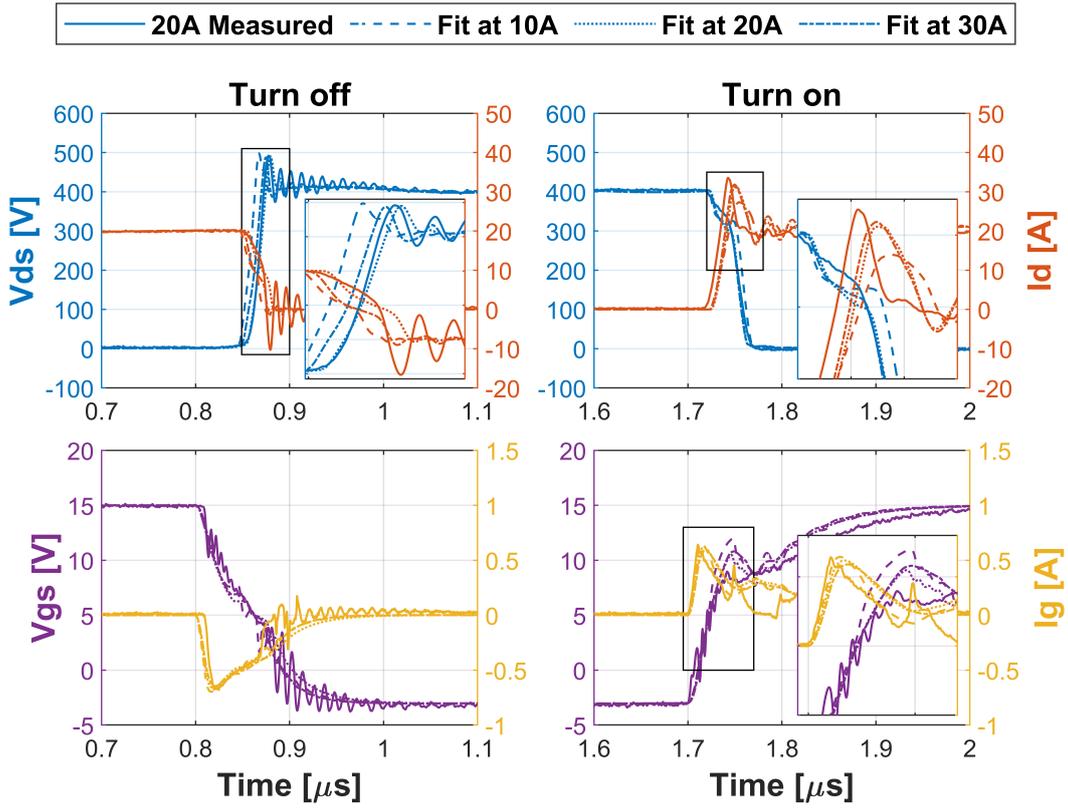


Figure 4.3: Measured and simulated waveforms at 20 A, fitted for C3M0060065K at 10/20/30A.

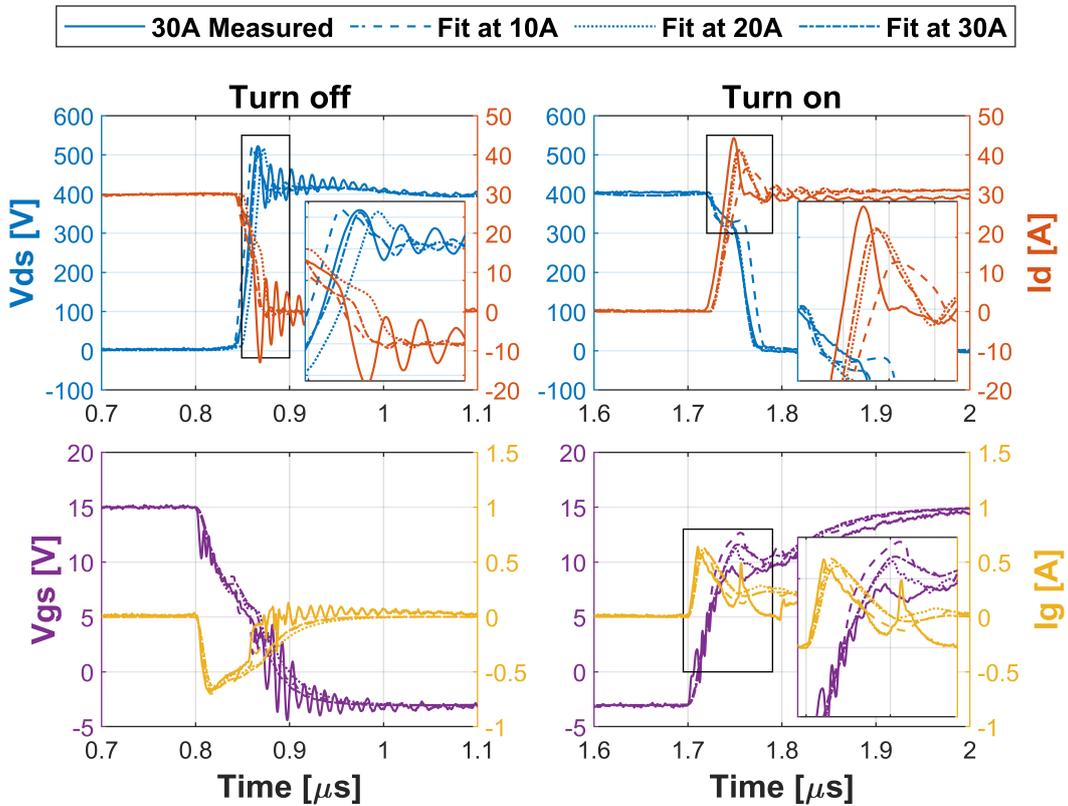


Figure 4.4: Measured and simulated waveforms at 30 A, fitted for C3M0060065K at 10/20/30A.

In general, the slopes are well matched throughout the tests with the gradients being close to the measured signals. The overshoot of v_{ds} is well matched, regardless of at what load current the parameters are fitted. The main difference between the simulations is the displacement in time, where fitting at higher current increases the turn off delay.

4.1.1 Switching Energies and EMI

Fig. 4.5 and Tab. 4.1 show the switching energies, defined as the time integral of $v_{ds} \cdot i_d$ for 10/20/30A load current. The turn on energies follow a similar trend as the measured data with since dv_{ds}/dt and di_d/dt are closely matched by the simulation. The offset is caused by the time delay in i_d which is visible in all tests. The turn off losses are closely matched at low current, but simulation results diverge at higher currents since di_d/dt is lower during turn off at 30 A.

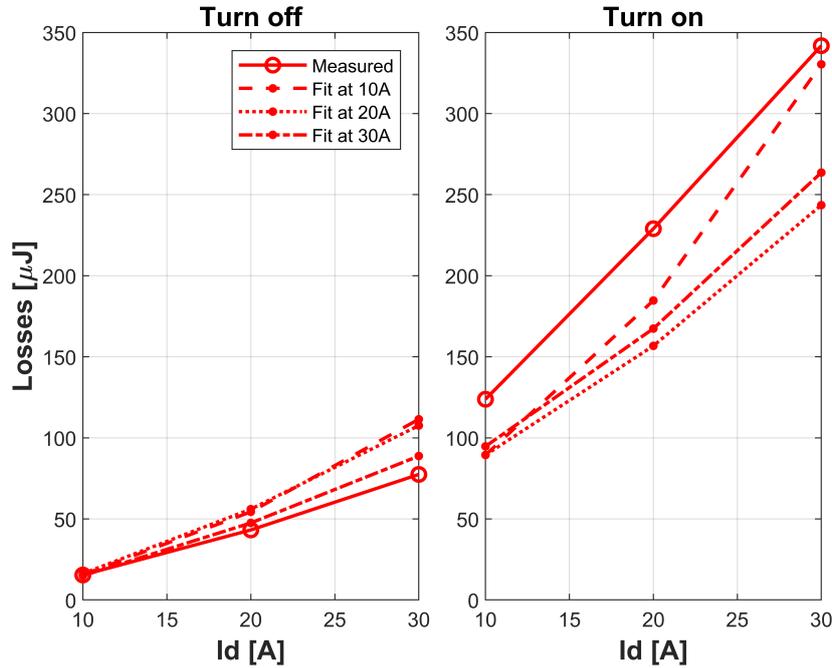


Figure 4.5: Turn on and turn off energy for different loads currents. Parameters fitted for C3M0060065K at 30 A.

C3M0060065K				
Energy [μJ]	Measured	Fit _{10A}	Fit _{20A}	Fit _{30A}
$E_{on,10A}$	124.13	89.42	89.44	95.75
$E_{off,10A}$	15.62	15.56	16.27	12.99
$E_{on,20A}$	229.01	184.69	156.73	167.39
$E_{off,20A}$	43.11	54.37	56.04	47.53
$E_{on,30A}$	342.09	330.37	243.52	263.60
$E_{off,30A}$	78.23	111.47	107.58	88.79

Table 4.1: Measured and simulated switching energies for C3M0060065K. Fitted at 10/20/30A and simulated at 10/20/30A.

The frequency spectrum for 30 A fitted data is compared to the 30 A measured data in Fig. 4.6. Below 30 MHz, the power loop signals closely match and i_d only deviates about 5 dB μ A from 20-30 MHz. Above 30 MHz, both i_d and v_{ds} diverges from the measured signals. However, EMC standards are only valid up to 30 MHz for conducted EMI, and the fit is therefore reasonable in this context. The gate signals diverge much earlier at around 15 MHz for v_{gs} and at 1 MHz for i_g . This is expected since the gate signals are more susceptible to interference and since i_g is not fitted by the algorithm.

C3M0060065K frequency spectrum

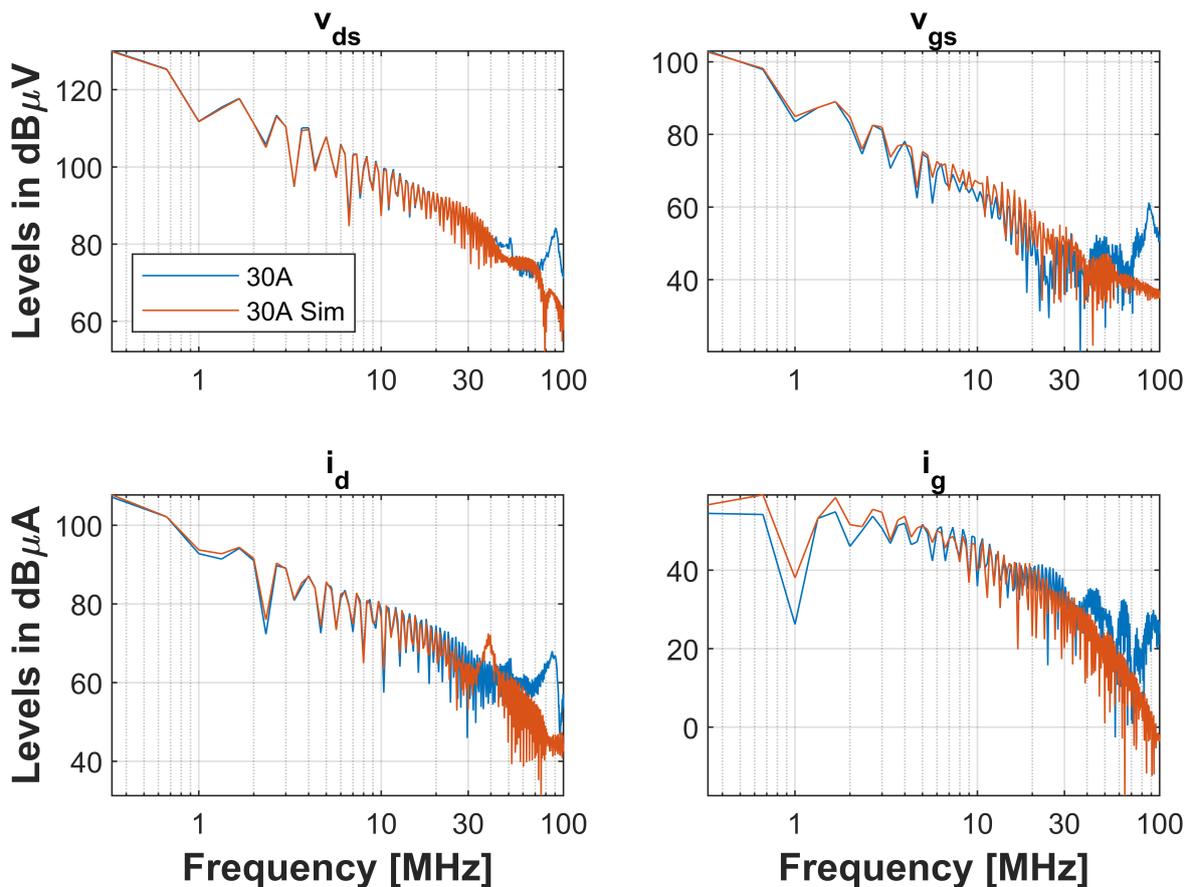


Figure 4.6: Measured and simulated frequency spectrum below 100 MHz, measured and fitted for C3M0060065K at 30 A.

4.1.2 Optimization Algorithm

The parameters used in the simulations are fitted based on the error in v_{ds} , i_d and v_{gs} . In Fig. 4.7, the scalogram, fitted at 30 A, is compared to 30 A measured scalogram. The surface plot show the measured data and the red contour lines show the amplitude of the simulated data. In the power loop, the majority of the frequency content is located below 10 MHz and peaks at around 90 MHz during turn off at $0.9 \mu\text{s}$. It can also be seen that the amplitude at 10-20 MHz varies between turn on and turn off for v_{ds} and i_d with the higher amplitude being where the overshoots occur. Since the frequency content is well matched below 30 MHz, gradients and overshoots are close to the measured data. Above 30 MHz, the function matches less well which is consistent with the lack of oscillations in the simulated waveforms.

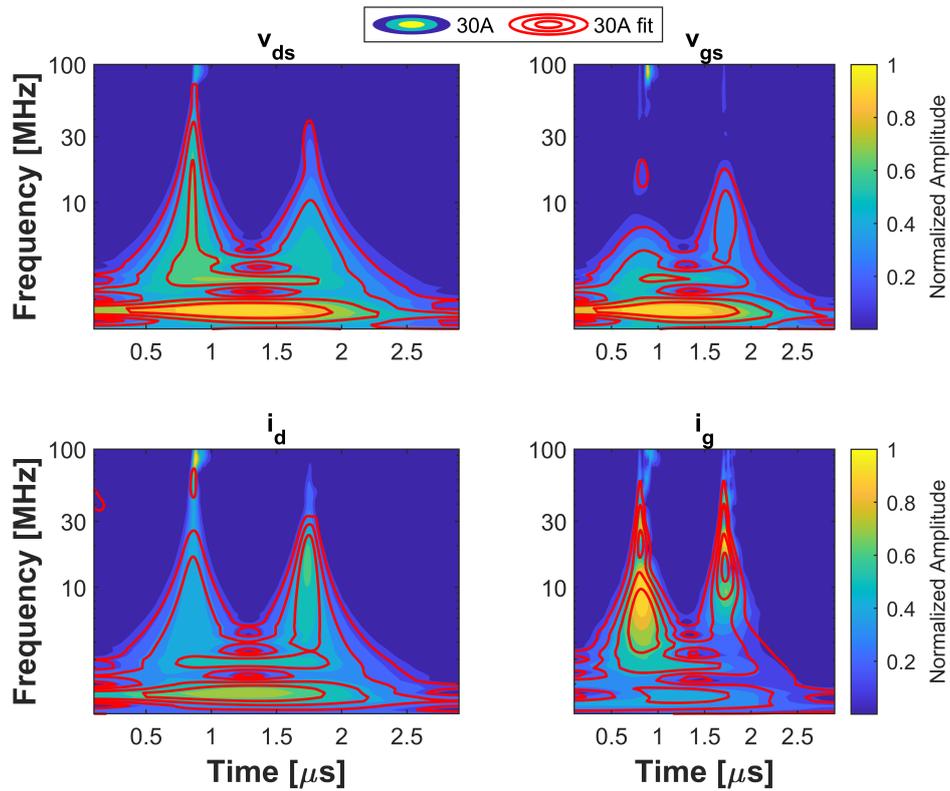


Figure 4.7: Comparison of measured and simulated scalograms for v_{ds} and i_d for C3M0060065K at 30 A.

4.2 IMZA65R048M1H Performance

The IMZA65R048M1H was tested on the same DPT setup as the C3M0060065K and the simulated switching waveforms are compared to measured data at 10/20/25A in Fig. 4.8, 4.9 and 4.10 respectively. v_{gs} follows an exponential trend with a clear Miller plateau around 0.85 and 1.75 μs . The overshoot in v_{gs} , caused by reverse recovery, is more prominent in this data and is better emulated by the simulation. Unlike the C3M0060065K, the gate current overshoot is higher in the simulations during turn off. This is almost exclusively determined by the internal gate resistor, which is likely too low compared to the measurements. Other effects in the gate signals are modeled fairly accurately, considering their complexity.

At 10 A, the 20 A and 25 A fits are almost identical with a slight turn off delay compared to the 10 A fit. The overshoot in v_{ds} is matched for all 3 fits, but the overshoot in i_d is consistently lower, compared to the measurements. During turn on, all 3 signals are delayed, with 30 A being the closest and 10 A being the worst.

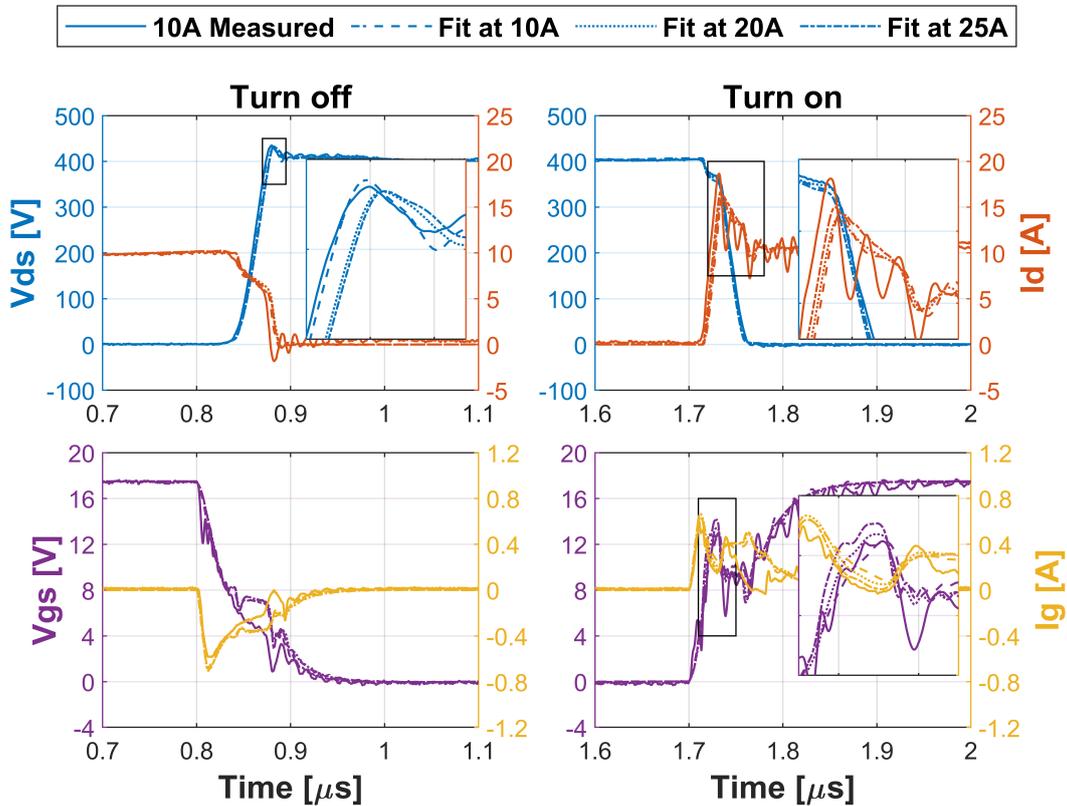


Figure 4.8: Measured and simulated waveforms at 10 A, fitted for IMZA65R048M1H at 10/20/25A.

The 20 A and 25 A comparisons are almost identical, although the turn on delay is more pronounced for the 10 A fit as load current increases. The overshoot in v_{ds} is slightly higher for 10 A at turn off. At turn on, all fits are indistinguishable and the 20 A and 25 A fits are almost perfectly matched for both turn on and turn off. Again, all 3 fits have a slight turn on delay with the 25 A fit being closest to the measurements.

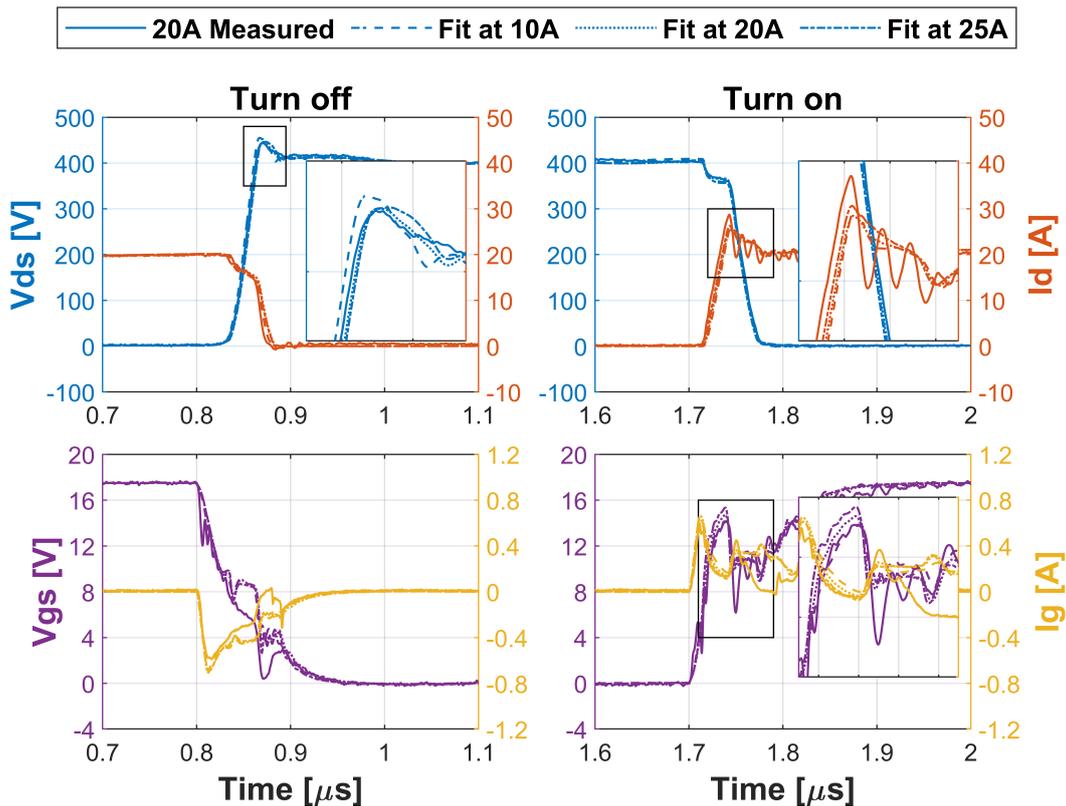


Figure 4.9: Measured and simulated waveforms at 20 A, fitted for IMZA65R048M1H at 10/20/25A.

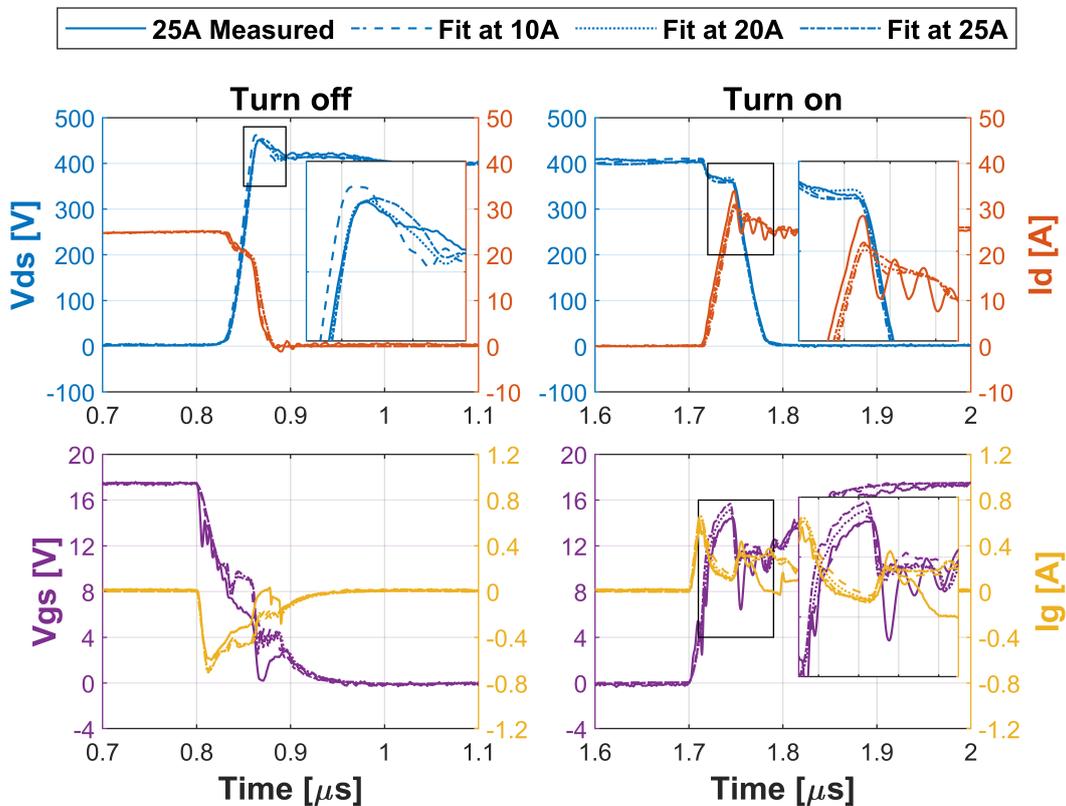


Figure 4.10: Measured and simulated waveforms at 25 A, fitted for IMZA65R048M1H at 10/20/25A.

In general, the IMZA65R048M1H waveforms better resemble ideal switching waveforms with more distinct regions and less oscillations. The fits are generally very consistent between plots and 20 A and 25 A are almost identical. There is noticeably less time displacement in the power loop signals and the overall fits are better, regardless of the current at which the parameters are fitted. For v_{ds} , the slopes and overshoots are matched and the dip caused by di_d/dt is clearly visible during turn on. The i_d turn off waveforms are also improved, compared to C3M0060065K, but there is still a visible turn on delay, albeit smaller. The signals has a better distinction between fall time and drop in current caused by dv_{ds}/dt .

4.2.1 Switching Energies and EMI

Fig. 4.11 and Tab. 4.2 show the switching energies, defined as the time integral of $v_{ds} \cdot i_d$ for 10/20/25A load current. The energies follow a similar trend as the measured data and increase as the load current increases. Similar to C3M0060065K, the offset is caused by the time delay and peak amplitude error in i_d . However, it can clearly be seen that the reduction in time displacement has significantly improved the model. The energies are closely matched for all fits, with the fit at 25 A being the best match at all current ratings.

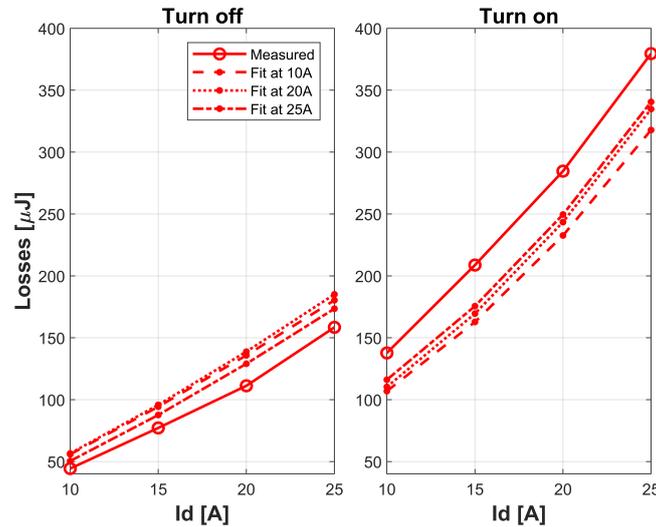


Figure 4.11: Turn on and turn off energy for different loads currents. Parameters fitted for IMZA65R048M1H at 25 A.

IMZA65R048M1H				
Energy [μJ]	Measured	Fit _{10A}	Fit _{20A}	Fit _{25A}
$E_{on,10A}$	137.83	106.88	110.32	116.10
$E_{off,10A}$	44.48	55.83	56.57	50.42
$E_{on,15A}$	208.72	162.78	169.52	175.56
$E_{off,15A}$	77.18	94.30	95.76	87.61
$E_{on,20A}$	284.33	232.63	243.43	249.54
$E_{off,20A}$	110.99	135.81	138.79	129.01
$E_{on,25A}$	379.46	317.81	334.77	340.54
$E_{off,25A}$	158.54	180.32	185.12	173.51

Table 4.2: Measured and simulated switching energies for IMZA65R048M1H. Fitted at 10/20/25A and simulated at 10/20/30A.

The frequency spectrum for 25 A fitted is compared to 25 A measured data in shown in Fig. 4.12. The power loop signals is a close match below 30 MHz with i_d starting to diverge at around 50 MHz while v_{ds} only deviates a few dB μ V in the 30-100 MHz range. The model is therefore more capable of modeling conducted EMI and the significantly fewer oscillations show that the model is also sufficient in the radiated EMI range. Like the C3M0060065K, the gate signals diverge earlier at around 30 MHz for v_{gs} and at 1 MHz for i_g . This is again expected since the gate signals are more susceptible to EMI and since i_g is not fitted by the algorithm.

5

IMZA65R048M1H Frequency Spectrum

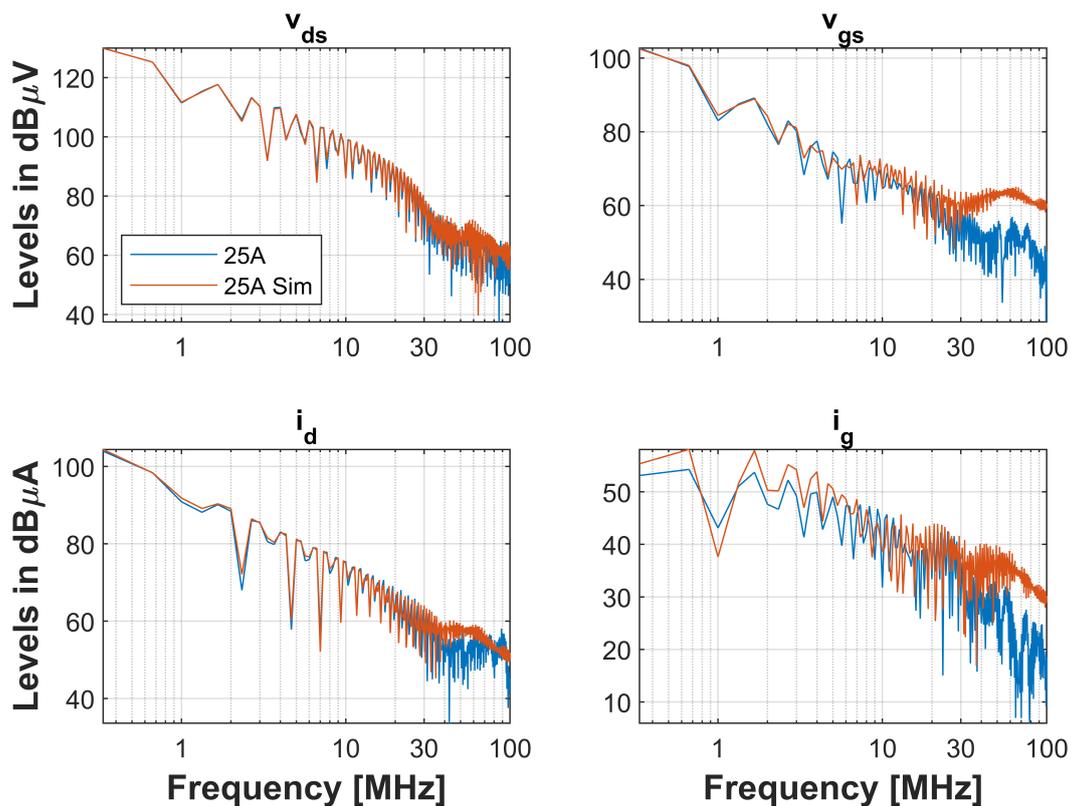


Figure 4.12: Measured and simulated frequency spectrum below 100 MHz at 25 A, fitted for IMZA65R048M1H at 25 A.

4.2.2 Optimization Algorithm

The parameters used in the simulations are also fitted based on the error in v_{ds} , i_d and v_{gs} . In Fig. 4.13, the scalogram, fitted at 25 A, is compared to 25 A measured scalogram. The surface plot show the measured data and the red contour lines show the amplitude of the simulated data. In v_{ds} , the frequency content is exclusively located in the conducted EMI range with no visible peaks above 30 MHz. Here, the contour is almost perfectly matched to the measured signal which explains the fit for v_{ds} . The drain current has exclusively matched the frequency content below 30 MHz, which is why the measurement oscillations are not present in the simulation. In the gate loop, v_{gs} has matched the energy concentrated at lower frequencies, which is why the slope matches. During turn on, at 1.8 μ s, there is a peak at 10-15 MHz which is consistent with the similar peak in i_d caused by reverse recovery.

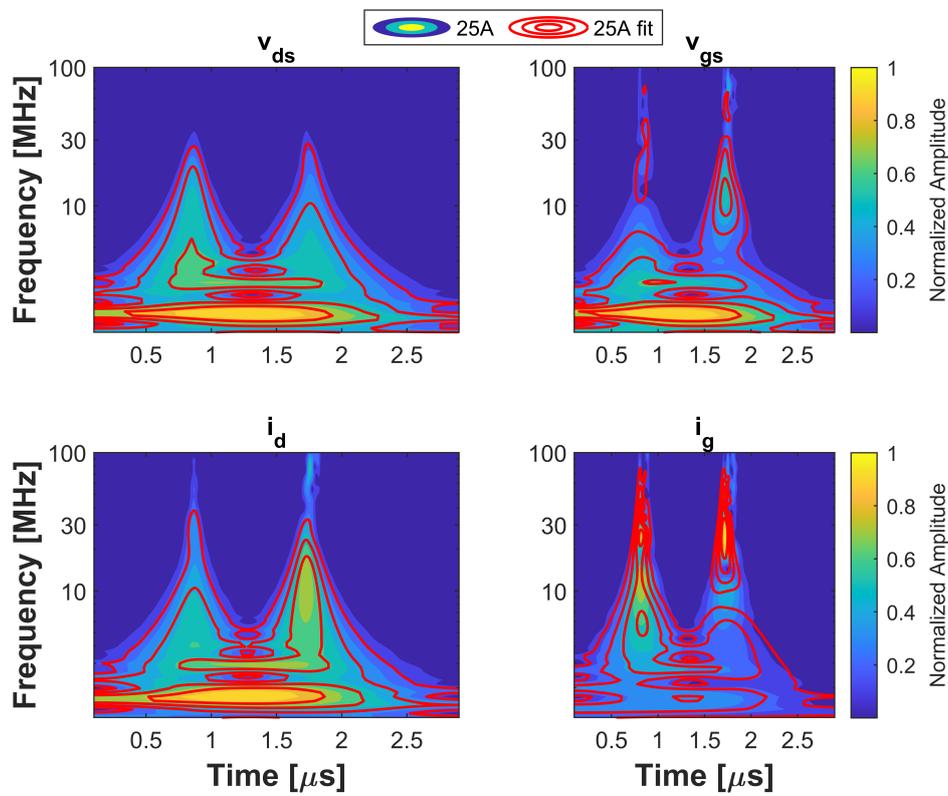


Figure 4.13: Comparison of measured and simulated scalograms for v_{ds} and i_d for IMZA65R048M1H at 25 A.

4.3 Overall Comparison

In order to evaluate the algorithms ability to identify parasitics, the fitted parameters from all tests are compared. The parameters are divided into device and circuit parameters in Tab. 4.3 and Tab. 4.4 respectively. The device parameters consist of parameters found in the C3M PSpice model and are therefore compared against the default model values. These are device specific and they are expected to vary between the two devices. The circuit parameters are generally related to the PCB and should be relatively consistent between tests. When optimizing, the parameters are only allowed to vary within predefined range. The default device parameters are extracted from the C3M PSpice model and the ranges are generally defined as $\pm 50\%$ from the default values. The parameters from C_{jo} to k_2 are curve fitting parameters used to fit the non-linear capacitors. The fitted curves for 30/25A are shown in Fig. 4.14 and Fig. 4.15. C_{gs} and v_{th} are also parameter specific and the fitted values for IMZA65R048M1H are more consistent with the 1100 pF and 4.5 V found in the datasheet.

Param.	Def.	Range		C3M0060065K			IMZA65R048M1H		
		Min	Max	$Mean_{10}$	$Mean_{20}$	$Mean_{30}$	$Mean_{10}$	$Mean_{20}$	$Mean_{25}$
C_{jo} [pF]	957	500	1500	588.45	1046.16	626.63	727.85	668.82	987.62
V_j [V]	2.14	1	3	2.35	1.99	2.01	1.67	2.5	1.33
M [-]	0.54	0.3	0.7	0.57	0.58	0.43	0.65	0.54	0.49
K [-]	-	0.5	1.5	0.74	0.97	0.66	0.6	0.65	0.63
k_1 [e-9]	0.70	0.4	1	1	0.95	0.96	0.45	0.48	0.56
k_2 [-]	0.36	0.2	0.4	0.4	0.38	0.38	0.25	0.25	0.27
C_{gs} [pF]	1100	800	2700	1342.47	1491.17	1319.99	1070.14	1094.61	924.81
V_{th} [V]	2.9	1.5	5.5	3.65	3.22	2.94	4.87	4.73	4.74
R_{gi} [Ω]	3	0.3	4	2.11	2.97	2.85	2.01	0.72	0.79
D_{tt} [ns]	0.5	0.1	15	0.36	0.61	0.25	0.24	0.51	2.84

Table 4.3: Comparison of fitted device parameters, relative to default values, for C3M0060065K and IMZA65R048M1H.

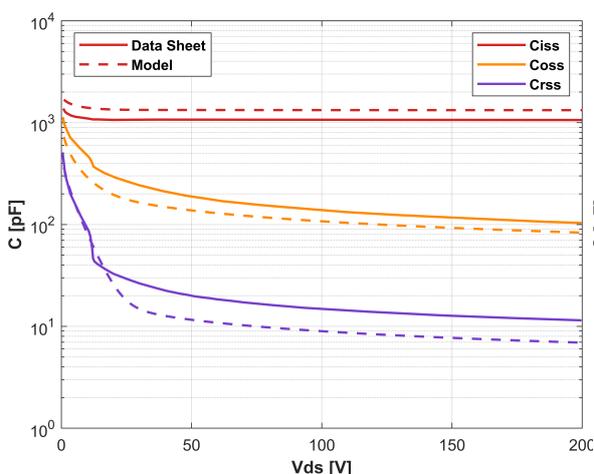


Figure 4.14: C3M0060065K capacitance characteristics fitted at 30 A vs. datasheet.

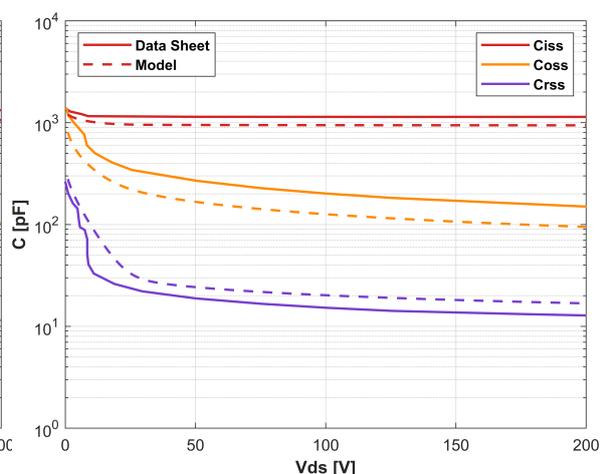


Figure 4.15: IMZA65R048M1H capacitance characteristics fitted at 25 A vs. datasheet.

Since the parasitics are unknown, the circuit parameters are allowed to vary in a wide range as seen in Tab. 4.4. To draw more definitive conclusions, additional data would be desirable, as the standard deviations observed in the results vary, as indicated in App. E.3. However, certain trends can still be observed. Namely L_{loop} , L_{d2} and R_{DC} are fairly consistent across all tests given the large range of possibilities. Other parameters, such as L_{s2} , C_p , R_s and R_d are consistent for each device but not between the two. It is also expected that C_p vary between the two devices since it represents the load inductor parasitic capacitance which can be difficult to distinguish from the effects of the high-side C_{ds} . Also, L_g is significantly higher than what is realistic. This is likely due to i_g not being considered in the error function.

-	Range	C3M0060065K			IMZA65R048M1H		
Param.	Min Max	$Mean_{10}$	$Mean_{20}$	$Mean_{30}$	$Mean_{10}$	$Mean_{20}$	$Mean_{25}$
R_L [$m\Omega$]	1 100	60.6	41.16	38.7	79.21	54.86	71.13
C_p [pF]	25 1000	211.68	269.61	311.68	196.13	162.08	177.46
R_{DC} [$m\Omega$]	1 100	14.96	12.91	13.17	17.47	12.72	15.04
L_{loop} [nH]	5 100	19.14	18.83	18.35	14.67	17.11	16.93
L_{d2} [nH]	1 40	39.92	30.23	28.63	26.03	26.29	31.63
L_{d1} [nH]	1 40	1.64	5.25	1.83	2.02	3.45	2.49
R_d [$m\Omega$]	0.1 100	86.01	72.15	75.35	43.47	58.93	85.15
R_{dp} [Ω]	1 500	21.19	9.43	57.27	83.21	24.84	6.97
L_{s2} [nH]	1 40	23.42	20.81	13.7	9.14	6.49	11.37
L_{s1} [nH]	1 40	2.97	2.28	2.69	4.43	2	2.97
R_s [$m\Omega$]	0.1 10	6.32	6.17	6.61	5.29	3.47	5.95
L_g [nH]	10 150	72.72	101.19	133.56	21.09	31.16	38.93
L_v [nH]	0.01 5	2.25	0.7	0.13	3.24	4.17	4.59

Table 4.4: Comparison of fitted circuit parameters for C3M0060065K and IMZA65R048M1H.

5. Discussion

This project used a number of assumptions and simplifications in order to reach its conclusion within the allotted time frame. As with any work, these give rise to number of possible error factors.

This project chose to simplify some of the complexity which was present in the C3M model from Wolfspeed which this project used as inspiration for its own circuit model. In particular the current dependency of the large signal transconductance was assumed constant, since a single double pulse test provides limited data and the computation time needed to be low for optimization. The temperature dependency of variables such as the threshold voltage or the output characteristics were removed, since optimizing the system for temperature was outside the scope of this project and instead the temperature was assumed a constant room temperature of 25 °C due to the short time-frame of the DPT. For the body diode, a PSpice diode model from the original C3M model was implemented, and only the transition time was left as a variable. A different model where more parameters were adjustable might have given more accurate results.

The effects that were measured in gate current were especially hard for all presented models to follow, and therefore it was chosen not to optimize for this variable since it interfered with the accuracy of other variables, the result of this can be seen in the unrealistic values for gate inductance. These simplifications can be a contributing factor to errors between simulation and measurement, the model was not able to fit frequencies in the range of radiated emissions. For these reasons the optimization tool might have performed differently using another simulation model.

As the model failed to accurately replicate the radiated emissions, the optimization algorithm could have had better performance by focusing the error function solely on the conducted emissions in the range below 30 Mhz. This is substantiated by the better performance when optimizing the waveforms for IMZA65R048M1H compared to C3M0060065K, since it displayed significantly lower radiated emissions in the measured data, resulting in lower interference with the error function.

In order to calculate the chosen error function, the data had to be truncated around the switching periods to reduce the number of calculations in the error function. The sample rate was reduced to 10 ns between switching periods and 1 ns during the switching period. This is however considered appropriate, since the current and voltage probe used for the power circuitry had a bandwidth of 400 and 500 Mhz respectively. However, it must still be considered as a potential error source.

The main results of this project were validated by comparing the difference between the device and circuit parameters for achieved after parameter extraction of tests done on two different devices, being C3M0060065K and IMZA65R048M1H. It was assumed that if the optimization was successful, it would converge on different device parameters, in the neighborhood of their respective data sheet values, while retaining similar circuit board parameters, the optimization was successful. However, since these two devices are still

fairly similar in behavior, the difference in device parameters is comparatively low. Using devices with more significant differences might have been advantageous, though this gives rise to its own problems of comparability, if done using identical models and parameter ranges. Another possibility for verifying the results, would be to use a more established
5 method such as FRA based methods.

Unfortunately this project suffered from an error in data for 2 runs at 10 A for C3M0060065K, and the time frame meant that they could not be redone. The remaining 3 runs where how-
ever still used in the project, though this means that the parameters have a lower chance
10 of being true parameter values. These values therefore have a chance of being closer to the values fitted by algorithm for a specific run instead of the mean parameter value found through multiple iterations. Since each run took more than 12 hours, the number of runs for each data set was limited to 5, but a larger number of runs could have given more con-
clusive data on the spread of parameter values, the chance of outliers effecting the overall
15 mean is larger for smaller datasets. This could be a reason that the turn on losses from the fits done at 10A where closer to the measured losses for C3M0060065K, despite all the other data suggesting that fitting at higher current levels, gave more accurate energy losses at all current levels.

6. Conclusion

This thesis focused on developing a PSpice-MATLAB based evolutionary algorithm for automatic extraction of parasitics in SiC-MOSFET test circuits. The concept improves EMI and switching loss modeling by extracting SiC device and circuit parameters from DPT test data.

5

The research first examined conventional device models and state-of-the-art parameters extraction techniques. MOSFET switching characteristics was investigated and a generalized PSpice model, incorporating parasitic circuit elements, was then developed based on the C3M00650065K SiC model from Wolfspeed. The transfer characteristics of the C3M model was simplified to reduce the number of optimization parameters and to increase compatibility with other devices.

10

The extraction of parameters was achieved by employing a robust optimization algorithm that combined PSO and PS which was necessary due to the complex non-linear nature of MOSFET switching characteristics. The objective function was designed using WT to localize the frequency content as it evolved in time. To compensate for displacement in both time, frequency and amplitude, the error between the measured and simulated signals was defined as the product of the minimum Euclidean distances for i_d , v_{ds} and v_{gs} .

15

20

The proposed model was evaluated using two similarly rated SiC devices at 400 V and 10/20/30A for C3M0060065K and 10/20/25A for IMZA65R048M1H. The results was then compared with the unmodified C3M model with respect to overshoot, EMI and switching energies. The unmodified model was able to model conducted emission well above 30 MHz, but exhibited large error in overshoot and switching energy with up to 60 % error in turn on energy.

25

In the C3M0060065K simulations, the overshoot in v_{ds} aligned well with the experimental data and the slopes were generally matched for both v_{ds} and i_d . The EMI analysis demonstrated a good fit below 30 MHz and the switching energies improved during turn on. The simulation results also showed a displacement in time for i_d and v_{ds} resulting in errors of switching energy for all current ratings. During the IMZA65R048M1H simulations, the behavior of the i_d during turn-off and the drain-to-source voltage v_{ds} during both turn-on and turn-off are close to identical with the experimental data for all fits. The EMI spectrum showed that the conducted EMI fits below 30 MHz. The switching energy is close to the experimental data with the best fit at 25 A and the error mainly stems from i_d overshoot during turn on.

30

35

7. Future Work

Due to the groundwork done in this report, further work can be directed in a number of directions.

More work should be done on the accuracy of the predicted parameters. Validating the method further by comparing it with parameter results from more established methods based on FRA or two-port S-parameter tests is a natural step. 5

Furthermore, the usefulness of utilizing this method in a circuit design process should also be reviewed. How accurate are the predictions from this model? For instance, if this model predicts specific advantages resulting from the reduction of L_d , it prompts the question of whether decreasing the path length of the drain leg will indeed yield the anticipated effect. 10

Further improvement of the optimization process is also possible. This project focused on optimizing along a single work point at a time, due to the time it takes to finish the optimization process on our laptops. However, by either reducing the total number of simulations for convergence or by increasing the available computation power, optimization along multiple work points should be plausible by doing minor tweaks on the fitness function. 15

Another option could be to optimize over more steps. First test data with e.g. a large gate resistor, slowing down the switching period, could be done. This would leave only the effects from the most significant parasitic components. These could then be locked or reduced in range, when optimizing for less significant parameters on test data with lower gate resistors and steeper slopes. 20 25

Bibliography

- [1] M. H. Rashid, *Power Electronics: Devices, Circuits, and Applications*, 4th edition. England: PEARSON, 2014, ISBN: 0-273-76908-1.
- [2] Y. Lobsiger, *Closed-Loop IGBT Gate Drive and Current Balancing Concepts*, eng. ETH-Zürich, 2014. 5
- [3] J. A. Warren *et al.*, “Energy impacts of wide band gap semiconductors in u.s. light-duty electric vehicle fleet”, eng, *Environmental science technology*, vol. 49, pp. 10 294–10 302, 2015.
- [4] H. Li, *Parallel Connection of Silicon Carbide MOSFETs for Multichip Power Modules*, eng. Department of Energy Technology, Aalborg University, 2015. 10
- [5] R. Natarajan, *Automotive gan fets engineered for high frequency and robustness in hev/evs*. [Online]. Available: https://e2e.ti.com/blogs_/b/powerhouse/posts/charge-faster-and-drive-farther-with-gan-based-onboard-charger-in-electric-vehicles, (Accessed: 10-2-2023).
- [6] M. J. Scott *et al.*, “Reflected wave phenomenon in motor drive systems using wide bandgap devices”, eng, in *2014 IEEE WORKSHOP ON WIDE BANDGAP POWER DEVICES AND APPLICATIONS (WIPDA)*, IEEE, NEW YORK: IEEE, 2014, pp. 164–168. 15
- [7] X. She, A. Q. Huang, O. Lucia, and B. Ozpineci, “Review of silicon carbide power devices and their applications”, eng, *IEEE transactions on industrial electronics (1982)*, vol. 64, pp. 8193–8205, 2017. 20
- [8] H. Lee, V. Smet, and R. Tummala, “A review of sic power module packaging technologies: Challenges, advances, and emerging issues”, *IEEE journal of emerging and selected topics in power electronics*, vol. 8, pp. 239–255, 2020.
- [9] T. Liu, T. T. Y. Wong, and Z. J. Shen, “A new characterization technique for extracting parasitic inductances of sic power mosfets in discrete and module packages based on two-port s-parameters measurement”, eng, *IEEE transactions on power electronics*, vol. 33, no. 11, pp. 9819–9833, 2018, ISSN: 0885-8993. 25
- [10] D. Drives, *Facts worth knowing about ac drives*. [Online]. Available: <https://danfoss.ipapercms.dk/Drives/DD/Global/SalesPromotion/FWK/fwk-2019/?page=4>, (Accessed: 23/5-2023). 30
- [11] S. Group, *Basics in emc / emi and power quality*. [Online]. Available: <https://products.schaffner.com/en/guide-to-emc>, (Accessed: 04/05-23).
- [12] T. Instruments, *An overview of radiated emi specifications for power supplies*. [Online]. Available: https://www.ti.com/lit/wp/slyy136/slyy136.pdf?ts=1684764129256&ref_url=https%253A%252F%252Fwww.google.com%252F, (Accessed: 23/5-2023). 35
- [13] P. Davari, F. Blaabjerg, E. Hoene, and F. Zare, “Improving 9-150 khz emi performance of single-phase pfc rectifier”, in *CIPS 2018; 10th International Conference on Integrated Power Electronics Systems*, 2018, pp. 1–6.

- [14] T. K. Gachovska and J. L. Hudgins, "5 - sic and gan power semiconductor devices", in *Power Electronics Handbook (Fourth Edition)*, M. H. Rashid, Ed., Fourth Edition, 2018. DOI: <https://doi.org/10.1016/B978-0-12-811407-0.00005-2>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128114070000052>.
- 5 [15] A. Stefanskyi, Starzak, A. Napieralski, and M. Lobur, "Analysis of spice models for sic mosfet power devices", in *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, 2017, pp. 79–81. DOI: 10.1109/CADSM.2017.7916089.
- [16] Infineon, *Coolmos detailed mosfet behavioral analysis using parameters extracted from models*. [Online]. Available: <https://www.farnell.com/datasheets/2354900.pdf>, (Accessed: 29/5-2023).
- 10 [17] Y. Cao, W. Zhang, J. Fu, N. Liu, Q. Wang, and L. Liu, "Application of the genetic algorithm in mosfet modeling and parameter extraction", in *2018 IEEE 3rd International Conference on Integrated Circuits and Microsystems (ICICM)*, Nov. 2018, pp. 264–267. DOI: 10.1109/ICAM.2018.8596549.
- 15 [18] W. Jouha, A. E. Oualkadi, P. Dherbécourt, E. Joubert, and M. Masmoudi, "Silicon carbide power mosfet model: An accurate parameter extraction method based on the levenberg–marquardt algorithm", *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9130–9133, 2018. DOI: 10.1109/TPEL.2018.2822939.
- 20 [19] H. Hove, O. Spro, G. Guidi, and D. Peftitsis, "Improved sic mosfet spice model to avoid convergence errors", Oct. 2019.
- [20] T. Q. Van Hoang, D. Yassuda-Yamashita, P. Fernandez-Lopez, and F. Lafon, "Spice model extraction for a mosfet based on a parametric simulation and waveform measurement", in *2017 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, 2017, pp. 1–6. DOI: 10.1109/EMCEurope.2017.8094706.
- 25 [21] S. K. Roy and K. Basu, "Measurement of circuit parasitics of sic mosfet in a half-bridge configuration", *IEEE Transactions on Power Electronics*, vol. 37, no. 10, pp. 11 911–11 926, 2022. DOI: 10.1109/TPEL.2022.3176114.
- [22] T. Liu, T. Wong, and Z. Shen, "A new characterization technique for extracting parasitic inductances of sic power mosfets in discrete and module packages based on two-port s-parameters measurement", *IEEE Transactions on Power Electronics*, vol. PP, pp. 1–1, Jan. 2018. DOI: 10.1109/TPEL.2017.2789240.
- 30 [23] K. Umetani, K. Aikawa, and E. Hiraki, "Straightforward measurement method of common source inductance for fast switching semiconductor devices mounted on board", *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8258–8267, 2017. DOI: 10.1109/TIE.2017.2694411.
- 35 [24] S. Linder, *Power semiconductors* (Engineering sciences. Electrical engineering.), eng, Electronic edition. -1st ed. Lausanne, Switzerland: EPFL Press, 2006.
- [25] Wolfspeed, *650 v discrete silicon carbide mosfets*. [Online]. Available: <https://www.wolfspeed.com/products/power/sic-mosfets/650v-silicon-carbide-mosfets/c3m0060065k/>, (Accessed: 1/6-2023).
- 40

- [26] J. Dodge, *Power mosfet tutorial*. [Online]. Available: https://www.microsemi.com/document-portal/doc_view/14692-mosfet-tutorial#:~:text=The%20reverse%20transfer%20capacitance%20is%20equal%20to%20the%20gate%20to%20drain%20capacitance.&text=The%20reverse%20transfer%20capacitance%20C%20often,the%20turn%20off%20delay%20time., (Accessed: 23-2-2023). 5
- [27] T. F. David Levett Ziqing Zheng, *Double pulse testing: The how, what and why*. [Online]. Available: https://www.infineon.com/dgdl/Infineon-Double_pulse_testing-Bodos_power_systems-Article-v01_00-EN.pdf?fileId=5546d46271bf4f920171ee81ad6c4a1f, (Accessed: 25/5-2023).
- [28] J. S. Arora, *Introduction to optimum design*, eng, Fourth edition. Amsterdam: Academic Press, 2017, ISBN: 0-12-800918-7. 10
- [29] J. Kennedy and R. Eberhart, "Particle swarm optimization", eng, in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, IEEE, 1995, 1942–1948 vol.4, ISBN: 9780780327689.
- [30] M. E. H. Pedersen, "Good parameters for particle swarm optimization", 2010. 15
- [31] MathWorks, *Pattern search terminology*. [Online]. Available: <https://www.mathworks.com/help/gads/pattern-search-terminology.html>, (Accessed: 31/05-2023).
- [32] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. DOI: 10.1017/9781108380690. 20
- [33] T. O'Haver, *A Pragmatic Introduction to Signal Processing*. May 2020, ISBN: 9798611266687.
- [34] MathWorks, *Practical introduction to multiresolution analysis*. [Online]. Available: <https://www.mathworks.com/help/wavelet/ug/practical-introduction-to-multiresolution-analysis.html>, (Accessed: 01/06-2023).
- [35] C. Burrus, R. Gopinath, and H. Guo, "Introduction to wavelets and wavelet transform—a primer", *Recherche*, vol. 67, Jan. 1998. 25
- [36] MathWorks, *Get started with wavelet toolbox*. [Online]. Available: https://www.mathworks.com/help/wavelet/getting-started-with-wavelet-toolbox.html?s_tid=CRUX_lftnav, (Accessed: 12/05-2023).
- [37] K. Technologies, *Keysight pd1500a dynamic power device analyzer/ double-pulse tester*. [Online]. Available: <https://www.keysight.com/dk/en/product/PD1500A/dynamic-power-device-analyzer-double-pulse-tester.html>, (Accessed: 25/5-2023). 30

A. DPT Setup

This paper used a Keysight PD1500A DPT setup for the DPTs conducted on the SiC mos-fets C3M0060065K from Wolfspeed and IMZA65R048M1H from Infineon. The test setup consists of optional modules as seen in Fig. A.1.

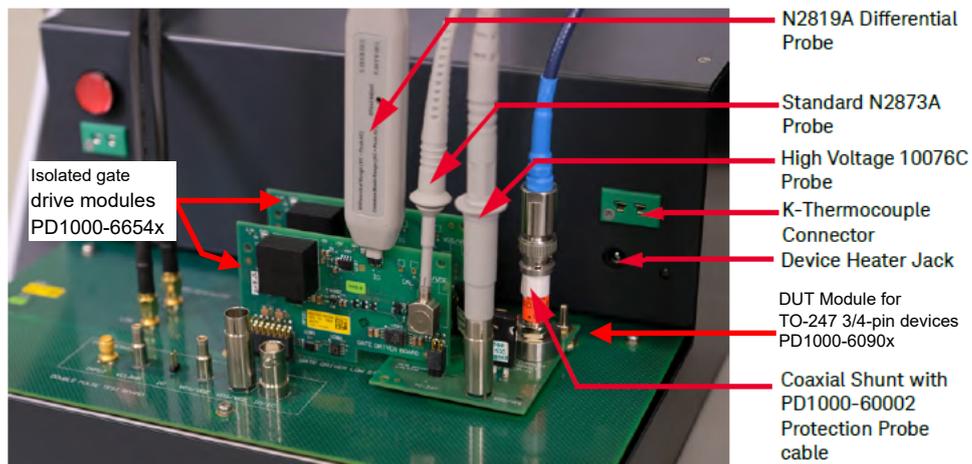


Figure A.1: Oscilloscope connections and test modules in the Keysight PD1500A double pulse test setup. [37]

Figure A.2 shows a simplified circuit schematic for the test board provided with the setup. 5

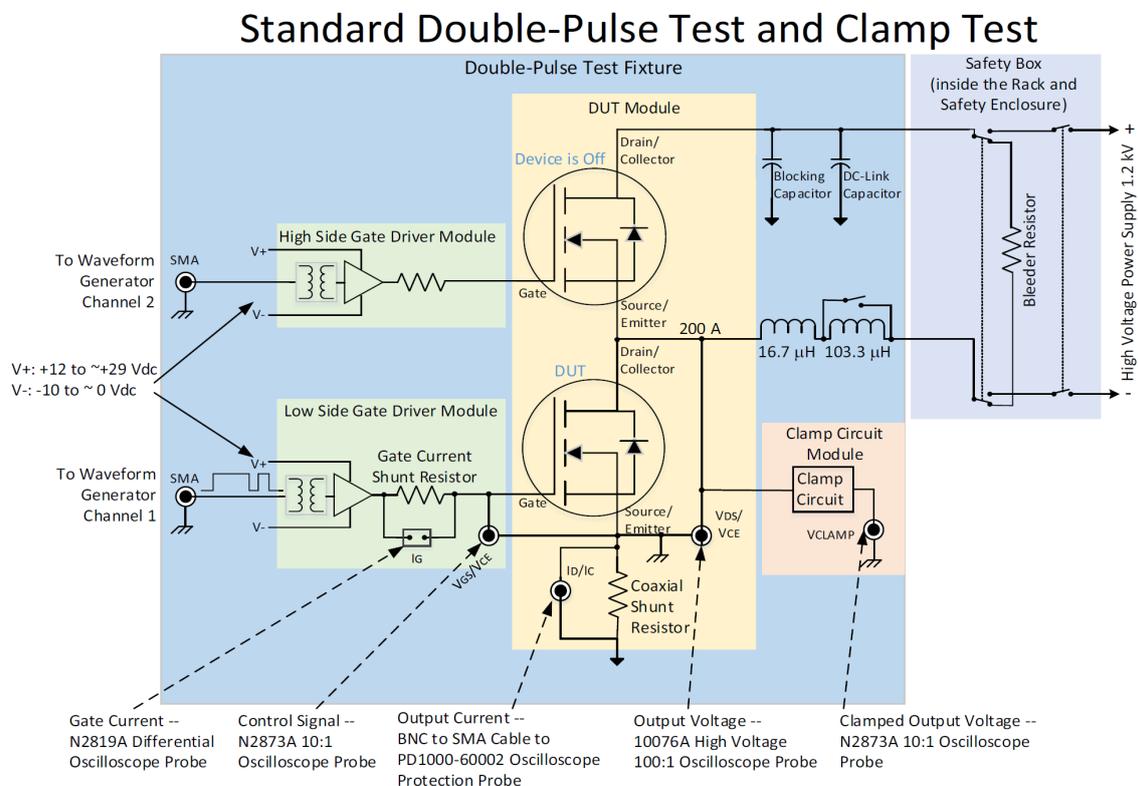


Figure A.2: Simplified circuit schematic of the Keysight PD1500A double pulse test setup. [37]

For the tests on C3M0060065K the gate voltage was set to -3 V and 15 V and for IMZA65R048M1H it was set to 0 V and 18 V in accordance with datasheet recommendations. The values for the circuit board Parts are given in Table A.1.

Parameter	Description	Value
R_g	Gate resistance	$20\ \Omega$
C_{DC}	Blocking Capacitor	$115\ \mu\text{F}$
L	Load inductance	$120\ \mu\text{H}$

Table A.1: Parameter values for the circuit board components used in the experiment.

The data from the tests where saved in the oscilloscope at a rate of 10 Ghz while the bandwidth of the probes ranged between 400 Mhz and 800 Mhz. The Probes used and their bandwidth are given in Table A.2.

Probe	Description	Bandwidth
N2819A	Differential shunt resistor probe for gate current	800 Mhz
N2873	Passive probe for gate voltage	500 Mhz
PD1000-60002 Oscilloscope Protection Probe	Shunt resistor probe for drain current	400 Mhz
10076A High Voltage Probe	High voltage probe for drain to source voltage	500 Mhz

Table A.2: Probes used in the experiment. All data was saved with the oscilloscope at a rate of 10 Ghz.

A screen shot from the oscilloscope, showing the full scale of the captured data is shown in Figure A.3.



Figure A.3: Keysight PD1500A DPT waveforms for C3M0060065K measured at 30 A and 400 V.

B. PSpice Code

This appendix gives a short explanation of how the PSpice model was written through the use of snippets from the PSpice code. In spice code, a dot (.) followed by a statement signals the use of a function, a new line starting with an asterisk (*) is ignored during execution and is used for comments, and starting a new line with a plus (+) signals, means that it should be read as a continuation of the previous line. 5

B.1 Circuit Model

This spice model setup a list of parameters, by using the .PARAM statement. These parameters can then be called in the place of values, for equations and circuit components, by using their name inserted in brackets { }. Most parameters where setup to be changed through matlab by identifying which part of the code should be replaced through the use of a triple dollar sign on both side. A snippet of code from the template, used to generate the circuit file used in Pspice, is shown here: 10

```
.PARAM
* Startup
+ILoad = $$$ILoad$$$
+Vdc = $$$Vdc$$$
+Vg_on = $$$Vg_on$$$
+Vg_off = $$$Vg_off$$$
+Rgnd = 1e6
+Rdecop = 1e-6
*Power circuit
+L = $$$L$$$
+RL = $$$RL$$$
+Cp = $$$Cp$$$
```

Matlab then replaces all the indicated fields, through the use of the WriteTpl function, with a set of parameter values: 15

```
.PARAM
* Startup
+ILoad = 7.5
+Vdc = 400
+Vg_on=15
+Vg_off=-3
+Rgnd = 1e6
+Rdecop = 1e-6
*Power circuit
+L = 0.00012
+RL = 0.092722
+Cp = 2.5003e-11
```

Using these defined parameters, the a circuit to be simulated is then build. An snippet of the code used for simulating the lowside MOSFET is shown here:

```
*Lowside Mosfet
*drain external
Ld2 Dext DLd {Ld2}
Rd DLd D_Q2 {Rd}
Rdp Dext D_Q2 {Rdp}

*chip
XGmos_Q2 D_Q2 Sv_Q2 gint_Q2 Sv_Q2 Gmos PARAMS: K={K} VTH={V_TH}
D1_Q2 Sv_Q2 D_Q2 bodydiode_C3M0060065K
Cds_Q2 D_Q2 Sv_Q2 0.1p
Xcds_Q2 D_Q2 Sv_Q2 cds PARAMS: Cjo={Cjo} Vj={Vj} M={M}
Cgd_Q2 D_Q2 gint_Q2 0.1p
Xcgd_Q2 D_Q2 gint_Q2 cgd PARAMS: k_1={k1} k_2={k2}
Cgs_Q2 gint_Q2 Sv_Q2 {Cgs} [IC=15]
Lv_Q2 Sv_Q2 S_Q2 {Lv}

*Gate internal
Rgi_Q2 g_Q2 gint_Q2 {Rgi}

*Source external
Ls2 S_Q2 SLs {Ls2}
Rs SLs 0 {Rs}
```

- Here a circuit component is defined by the initial letter of the name. PSpice will read a component starting with an L as and inductor, R as a resistor and C as capacitor and D
- 5 as a diode. An X indicates that the device should use a user defined component. Most components will have a structure where first a component name is declared and then the node connections are given. For some components a component value is then declared followed by a model name if the standard model should not be used, and then a set of Parameters to be used in the model can be declared. Initial simulation conditions can be
- 10 set using [IC =]. For capacitors this will be read as a voltage, and for inductors this is read as a current. This was used for reducing simulation time, by reducing the amount of startup oscillations. Components are automatically connected through the names of their nodes. The ground node is always indicated by the use of 0 for a note.
- 15 The drain current was setup as a component, through the use of the .subckt function. A controlled current source defined as Gdrain is used to produce the output characteristics of the MOSFET from eq. (2.1) on page 7, through the use of if statements.

```
.subckt Gmos d s g ks PARAMS: K=0.9 VTH=2.9
```

```
Gdrain d s VALUE = { IF (V(g, ks) < VTH,
+0,
+IF (V(d, s) >= (V(g, ks) - VTH) ,
+K * PWR(V(g, ks) - VTH, 2) ,
+K * (2 * (V(g, ks) - VTH) * V(d, s) - PWR(V(d, s) , 2) ) ) }
.ends Gmos
```

Pspice can run different kinds of analysis, for this project a transient analysis. The .tran statement runs a transient analysis, for the purpose of this model, only the 2nd and 4th statement are needed. They determine the total simulation time and the maximum step size respectively. A total simulation time of $3\ \mu\text{s}$ and maximum step size of $0.5\ \text{ns}$ were chosen. This was based on a balance between allowing enough time during startup and between switching periods to allow oscillations to die down, while also minimizing simulation time and having a sufficient simulation resolution for accuracy. Especially simulation time was a big factor, since the optimization algorithm has to run tens of thousands of simulations.

```
.tran 0 3e-06 0 0.5e-9
.probe/CSDF V([D_Q2],[S_Q2]) I(Rs) V([g_Q2],[S_Q2]) I(Rge_Q2)
```

The .probe/CSDF statement, results in pspice writing the resulting solution from the simulation at each timestep as a .txt matrix for each voltage or current that is called. In this example it is v_{ds} , i_d , v_{gs} and i_g respectively.

B.1.1 Misc PSpice

This function gives PSpice effects how the PSpice simulation converges at each time step, and how to handle failures in converges. This increases the range of parameters which result in successful convergences.

```
.OPTIONS ADVCONV
.AUTOCONVERGE ITL1 = 150 ITL2 = 20 ITL4 = 10 RELTOL = 0.001
+ ABSTOL = 1e-012 VNTOL = 1e-006 PIVTOL = 1e-013
```

Every node in the Pspice model was grounded through a large resistor of $1\ \text{M}\Omega$. This increases both the convergence speed and the range of parameters which result in successful convergence.

```
*Gnd list
RDCagnd DCa 0 {Rgnd}
RDCgnd DC 0 {Rgnd}
RDCbgnd DCb 0 {Rgnd}
RDCLgnd DCL 0 {Rgnd}
```


C. MatLab code

This appendix documents the MATLAB implementation presented in 4.1. The *Master Script* executes the various scripts and functions.

C.1 Configuration of Optimization

The master script used to execute the optimization had a list of options which could be set through the matlab command window. This shows a snippet of how the options changed certain default parameters based on which device was simulated. 5

```
Setup.Select = 0;
% Execute the loop until a device is selected
while Setup.Select == 0
    % Prompt the user to select a device and store the input
    Setup.Device = input("Select device (cree/inf): ", "s");
    % Evaluate the selected device
    switch Setup.Device
        case "cree"
            Setup.DeviceName = 'C3M0060065K';
            Setup.nom.Vgs_on = 15;
            Setup.nom.Vgs_off = -3;
            Setup.nom.Cdc=1.15e-7;
            Setup.nom.L=120e-6;
            Setup.nom.Rge=20;

            Setup.PTable=ParamTable_Generel();
            Setup.Select = 1; % Set the flag to exit the loop
        case "inf"
            Setup.DeviceName = 'IMZA65R048M1H';
            Setup.nom.Vgs_on = 17.5;
            Setup.nom.Vgs_off = 0;
            Setup.nom.Cdc=1.15e-7;
            Setup.nom.L=120e-6;
            Setup.nom.Rge=20;

            Setup.PTable=ParamTable_Generel();
            Setup.Select = 1; % Set the flag to exit the loop
    end
end
```

C.2 Optimization Algorithm

The settings used for optimization were selected based on user inputs through the command window.

```

%% PS Settings
if Setup.PS_flag == 1
    PSoptions = optimoptions('patternsearch', ...
        'InitialMeshSize', 50, 'StepTolerance', 1e-3, 'MaxMeshSize', ...
        50, 'PlotFcn', {@psplotbestf, @CustomPSPlotBestIdVd}, ...
        'Display', 'iter');
end

%% Hybrid Settings
if Setup.Hybrid_Flag == 1
    PSOptions = optimoptions('particleswarm', 'SwarmSize', ...
        Setup.SwarmSize, 'MaxIterations', Setup.MaxIter, 'PlotFcn', ...
        {@pswplotbestf, @CustomPSOPlotBestIdVd}, 'Display', 'iter', ...
        'HybridFcn', {'patternsearch', PSoptions});
end

```

Based on which optimization was chosen, the script would run either a PS, PSO or a hybrid
5 optimization of the problem. The error function was selected during setup.

```

%% PSO hybrid algorithm

```

```
a = CenterPointInit;

% Assigning the number of variables to 'nvars' based on the
% length of Optimization variables
nvars=length(OptIndex);
% Initializing empty matrices A, b, Aeq, beq
A = [];
b = [];
Aeq = [];
beq = [];
% Setting lower bound 'lb' as an array of zeros and
% upper bound 'ub' as an array of 100s
lb = zeros(1,length(OptIndex));
ub = 100*ones(1,length(OptIndex));

% Initializing nonlcon as an empty array for nonlinear constraints
nonlcon = [];

if Setup.Hybrid_Flag==1
sol = particleswarm(fun,nvars,lb,ub,PSOptions);
% Using hybrid pso/ps optimization algorithm with provided options
elseif Setup.PS_flag==1
sol = patternsearch(fun,a,A,b,Aeq,beq,lb,ub,nonlcon,PSoptions);
% Using patternsearch optimization algorithm with provided options
else
sol = particleswarm(fun,nvars,lb,ub,PSOptions);
% Using particleswarm optimization algorithm with provided options
end
```

C.3 PSpice Simulation

In order to run Pspice from matlab, it is executed as a command line simulation. This is a version of Pspice that can be executed from third party software. The simulation is executed through the system command, which executes the .cir File defined by filename in Pspice. The WriteTpl command generates the circuit file to be simulated based on the template and the parameter solution to be simulated.

```
% Set the root directory for the PSPICE command-line tool
% (Change the directory as per your system)
RootDirect = 'C:\Cadence\SPB_17.2\tools\bin\psp_cmd.exe_';

% Specify the file name for the circuit model
FileName = 'General_Model_DPT';

% Find parameter names and replace values in the cir file
WriteTpl(FileName, PNames, SolValues);

% Run the simulation with echo enabled to display the output
system([RootDirect convertStringsToChars(FileName) '.cir'] ,...
'-echo');
```

The matlab function used to generate the circuit file from the template file, by replacing the fields indicated by the parameter names with the parameter values.

```
function [b] = WriteTpl(filename, ParamNames, ParamValues)
```

```

filechar=convertStringsToChars(filename);
strTpl = [filechar '.tpl'];
strCir = [filechar '.cir'];

% Read the template file
TemplateLibFile = fileread(strTpl);
LibFile = TemplateLibFile;

% Replace parameter placeholders with their corresponding
% values
for ParameterIndex = 1:length(ParamValues)
    LibFile = strrep(LibFile ,...
    ['$$$'
    convertStringsToChars(ParamNames(ParameterIndex))
    '$$$'], ...
    num2str(ParamValues(ParameterIndex)));
end

% Write the updated values to the .cir file
tempsum=0; % Counter for the number of attempts to open the % file
LibFileID=-1;
while LibFileID == -1
    LibFileID = fopen(strCir , 'w');
    if LibFileID ~= -1
        break; % File opened successfully , exit the loop
    else
        pause(0.5); % Pause for 0.5 seconds before retrying
        tempsum = tempsum+1; % Increment the attempt counter
    end
    if tempsum == 4
        % If the file fails to open after 4 attempts ,
        % throw an error
        error('Error_with_writing')
    end
    end
    fwrite(LibFileID , LibFile , 'char');
    fclose(LibFileID);

    % Return success flag
    b=1;
end

```

A shortened version of the function used to call the parameter table, where each row has a parameter name, a default value, an option of either using the default value or optimizing the parameter and lastly a range where the parameter is optimized within.

```
function ParamTable = ParamTable_Generel()
ParamTable={"ILoad",8.5,"def",[28,32];...
            "Vdc",400,"def",[395,410];...
            "L",53e-6,"def",[46,60]*10^-6;...
            "Vg_on",15,"def",[13,20];...
            "Vg_off",-3,"def",[-4,0.1];...
            "RL",50e-3,"opt",[1,100]*10^-3;...
            "Cp",50e-12,"opt",[25,1000]*10^-12;...
            };

end
```

C.4 Double Pulse Test Data

This function imports the data output from the PSpice simulation into a vector in Matlab.

```

function [time,vds,id,vgs,ig] = ImpDat(FileName)
% Open the file for reading
OutFID = fopen([convertStringsToChars(FileName) '.csd'],'r');
% Initialize variables
k=0;
%continue until reaching the end of the file
while ~feof(OutFID)
    % Read the file line by line
    Line = fgetl(OutFID);
    % Check if the line starts with '#C', indicating data.
    if ~strncmp(Line,'#C',2)
        % If not, continue to the next line
        continue
    else
        % If yes, extract the data from the line
        k = k+1;
        % Extract time
        temp = sscanf(Line(4:end),'%f');
        time(k) = temp(1);
        % Extract vds, id, vgs, ig
        Line = fgetl(OutFID);
        temp = sscanf(Line,'%f:%f');
        vds(k) = temp(1);
        id(k) = temp(3);
        vgs(k) = temp(5);
        ig(k) = temp(7);
    end
end
fclose(OutFID);
end

```

Reading measurement data and then creating a truncated time vector and interpolate measurement data according to it:

```

% Read data from a table and convert it to an array

```

```

Data = table2array(readtable(mFile,'NumHeaderLines',1));
% Extract data from the array and transpose it
timedat = Data(:,1)';
vgsdat = Data(:,2)';
igdat = Data(:,3)';
vdsdat = Data(:,4)';
iddat = Data(:,5)';

timejumpsmall=1e-9; % Set the small time jump value
timejumplarge=10e-9; % Set the large time jump value
i=1;
time(1)=0.5e-6; % Initialize the time array with the initial time
while time<2.3e-6 % Loop continues until turn on is finished
    i=i+1;
    % Check if the time is within turn on period
    if and(time(i-1) > 1.685e-6,time(i-1) < 1.9e-6)
        time(i)=time(i-1)+timejumpsmall;
    % Check if the time is within turn off period
    elseif and(time(i-1) > 7.85e-7,time(i-1) < 1.2e-6)
        time(i)=time(i-1)+timejumpsmall;
    else
        time(i)=time(i-1)+timejumplarge;
    end
end
end
% Interpolate measurement data based on the time scale.
vds=interp1(timedat,vdsdat,time);
vgs=interp1(timedat,vgsdat,time);
ig=interp1(timedat,igdat,time);
id=interp1(timedat,iddat,time);

```

C.5 Import and Post Processing

The simulation data which was imported from PSpice, was interpolated to have the same truncated time base as the measured data.

```
%Import data
[timeopt , vdsopt , idopt , vgsopt , igopt ]=ImpDat( FileName );
VdsSync=interp1( timeopt , vdsopt , time );
VgsSync=interp1( timeopt , vgsopt , time );
IgSync=interp1( timeopt , igopt , time );
IdSync=interp1( timeopt , idopt , time );
```

To reduce the number of elements in the matrix, produced by the continuous wavelet matrix, a 1-D discrete wavelet transformation is used to compress the data. This is a common technique that is used when transmitting signals, filtering noise or compressing image files such as JPEGs [32]. The signal is deconstructed using the `wavedec()` function from MatLab. The input signal is deconstructed into approximation and detail coefficients by convolving the signal with a series of low- and high-pass filters. The filter coefficients depends on the wavelet type and Daubechies #4 is a typical wavelet choice for signals with mixed frequency content [36]. A compressed version of the deconstructed signal is then reconstructed using the `appcoef()` function where $RecLvl < DecLvl$. The x- and y-axis of the reconstructed signal are scaled and they are therefore scaled back with respect to the original signal. The compressed signal is then transformed using the `cwt()` function with the Morlet wavelet as explained in Sec. 3.5.

```
function [sVgsRec , stRec , sVgs_cwt , sf_cwt] = ...
```

```
WT_func_time(time ,VgsSync ,wType ,DecLvl ,RecLvl );

[XRec ,tRec ,X_cwt ,f_cwt] = WT_func_time(time ,X ,...
wType ,DecLvl ,RecLvl)

% Transpose vector if dimensions are wrong
if size(X,1) > size(X,2)
    X = X';
end

[c ,l] = wavedec(X,DecLvl ,wType);

% Levels of details in the reconstruction
XRec1 = appcoef(c ,l ,wType ,RecLvl);

% normalize x-axis
XRec = XRec1/max(XRec1)*max(X);
Sample = (0:length(XRec)-1)';

% Scale time axis
tRec = Sample/max(Sample)*max(time);
TsRec = tRec(2)-tRec(1);

% Compute cwt
[X_cwt ,f_cwt] = cwt(XRec ,'amor' ,(1/TsRec)); % 'amor' = Morlet

end
```

C.6 Fitness Evaluation

The 3d matrix of time, frequency and amplitude was transformed into a vector of 3d points for both simulation and measured data. The euclidian distance from every point in the simulation to every point in the measured data is then calculated, and the error between eg. the measured and simulated drain-source voltage was calculated as the average of the minimum distances from every point of the simulation to the measured data. 5

```
% Creating a normalized 3D vector for  
% simulated drain voltage 'psVds'  
psVds=p3dvector(log10(sf_cwt)/log10(max(sf_cwt)),...  
stRec/max(stRec),abs(sVds_cwt)/max(max(abs(Vds_cwt))));  
  
% Calculating the Euclidean distance between  
% measured 'pVds' and simulated 'psVds'  
dist3dVds=pdist2(pVds,psVds,'euclidean');  
  
% Calculating the mean of the minimum distances  
% in 'dist3dVds' and assigning it to 'vderror'  
vderror=mean(min(dist3dVds));
```

C.7 Save Script

When the optimization terminates, the final solution is saved as a file together with plots showing the progress of the optimization algorithm and the options which were chosen during setup.

```
SaveFlag = 0;
ii = 0;
while SaveFlag == 0
    SaveBestSolution = [num2str(ii) '_SaveValue.mat'];
    SaveSetupFile = [num2str(ii) '_Setup.mat'];
    SaveFigFile = [num2str(ii) '_OptAlgorithm'];

    SolutionSaveDir = [FolderName '\ ' SaveBestSolution];
    SetupSaveDir = [FolderName '\ ' SaveSetupFile];
    FigSaveDir = [FolderName '\ ' SaveFigFile];

    if isfile(SolutionSaveDir) || isfile(SetupSaveDir) || ...
       isfile(FigSaveDir)
        ii = ii + 1;
    else
        save(SolutionSaveDir, 'SaveValue');
        save(SetupSaveDir, 'Setup');
        FigList = findobj(allchild(0), 'flat', 'Type', ...
            'figure');
        for n = 1:length(FigList)
            savefig(FigList(n), [FigSaveDir num2str(n) '.fig']);
        end
        SaveFlag = 1;
    end
end
```

D. Optimization Algorithm

This appendix shows how the optimization algorithm iterates towards the best function values. Each run performs 100 iterations of PSO with a swarm size of 200. The best function value from PSO is then followed by PS until convergence as seen in Fig. D.1 and Fig. D.2. The first and last iteration of the Matlab output log is shown in Fig D.3.

5

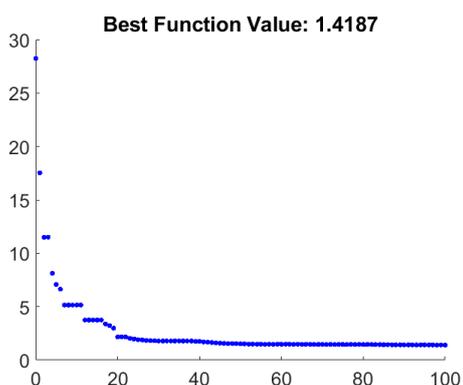


Figure D.1: PSO best function value vs. iteration for IMZA65R048M1H 25 A run1.

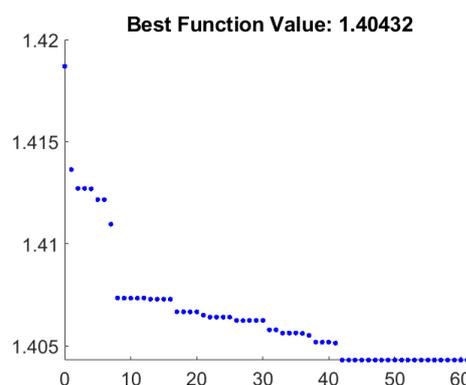


Figure D.2: PS best function value vs. iteration for IMZA65R048M1H 25 A run1. Initial values are the parameters found in iteration #100 of Fig. D.1.

Iteration	f-count	Best f(x)	Mean f(x)	Stall Iterations
0	200	28.26	NaN	0
1	400	17.53	661.3	0
2	600	11.51	134.8	0
98	19800	1.42	1.757	1
99	20000	1.419	1.971	0
100	20200	1.419	1.681	0

Optimization ended: number of iterations exceeded OPTIONS.MaxIterations. Switching to the hybrid optimization algorithm (PATTERNSEARCH).

Iter	Func-count	f(x)	MeshSize	Method
0	1	1.4187	50	
1	2	1.41364	50	Successful Poll
2	10	1.41272	50	Successful Poll
3	33	1.41272	25	Refine Mesh
61	1919	1.40432	1.49e-06	Refine Mesh
62	1963	1.40432	7.451e-07	Refine Mesh

Optimization terminated: mesh size less than options.MeshTolerance. PATTERNSEARCH ended.

Figure D.3: Matlab output log of first and last iteration for PSO and PS for IMZA65R048M1H 25 A run1.

E. Simulation Data

This appendix contain the full list of parameters, minimum function values, mean and standard deviation of all 6 runs starting with C3M0060065K 10/20/30A followed by IMZA65R048M1H 10/20/25A.

E.1 C3M0060065K

5

Due to an error in the MatLab code, run 4 and 5 of the 10A optimization run in Tab. E.1 was fitted with fixed random number generator seed and the runs are therefore identical. The optimization algorithm did therefore not work correctly and mean and standard deviation for 10A are taken from run 1-3 exclusively.

C3M0060065K 10A								
Parameter		$Mean_{10}$	Std_{10}	Run1	Run2	Run3	Run4	Run5
R_L	[$m\Omega$]	60.6	29.84	79.09	26.18	76.53	92.72	92.72
C_p	[pF]	211.68	32.76	249.08	197.86	188.09	25	25
R_{DC}	[$m\Omega$]	14.96	3.9	10.87	15.39	18.64	11.21	11.21
L_{loop}	[nH]	19.14	0.96	18.12	20.01	19.28	19.15	19.15
L_{d2}	[nH]	39.92	0.08	39.96	39.83	39.98	26.37	26.37
L_{d1}	[nH]	1.64	1.06	1.05	1	2.87	5.89	5.89
R_d	[$m\Omega$]	86.01	8.54	89.96	91.86	76.21	99.69	99.69
R_{dp}	[Ω]	21.19	20.45	4.96	44.16	14.45	1.3	1.3
L_{s2}	[nH]	23.42	9.41	29.86	12.63	27.78	15	15
L_{s1}	[nH]	2.97	1.93	1.13	4.98	2.81	1.24	1.24
R_s	[$m\Omega$]	6.32	4.69	9.75	0.98	8.23	6.56	6.56
L_g	[nH]	72.72	67.02	149.95	38.38	29.84	148.88	148.88
L_v	[nH]	2.25	1.49	3.96	1.22	1.56	3.35	3.35
C_{jo}	[pF]	588.45	127.29	508.29	735.23	521.85	839.87	839.87
V_j	[V]	2.35	0.47	2.52	1.82	2.72	1.19	1.19
M	[—]	0.57	0.06	0.64	0.57	0.51	0.67	0.67
K	[—]	0.74	0.09	0.74	0.65	0.83	1.39	1.39
k_1	[e-9]	1	0	1	1	0.99	0.99	0.99
k_2	[—]	0.4	0	0.4	0.39	0.4	0.36	0.36
C_{gs}	[pF]	1342.47	239.15	1066.33	1479.7	1481.39	1039.21	1039.21
V_{th}	[V]	3.65	0.31	3.89	3.3	3.76	4.82	4.82
R_{gi}	[Ω]	2.11	1.33	3.62	1.11	1.6	3.97	3.97
D_{tt}	[ns]	0.36	0.2	0.13	0.47	0.47	9.92	9.92
f_{min}	[—]	1.6961	0.33205	1.3288	1.9749	1.7846	3.1545	3.1545

Table E.1: C3M0060065K parameters fitted at 10 A over 5 runs.

C3M0060065K 20A								
Parameter	$Mean_{20}$	Std_{20}	Run1	Run2	Run3	Run4	Run5	
R_L [$m\Omega$]	41.16	40.17	17.8	12.33	99.2	67.32	9.15	
C_p [pF]	269.61	64.42	336.79	230.84	280.73	319.43	180.28	
R_{DC} [$m\Omega$]	12.91	4.07	14.97	8.75	11.66	10.23	18.91	
L_{loop} [nH]	18.83	0.76	18.64	19.92	19.01	18.77	17.8	
L_{d2} [nH]	30.23	10.22	37.52	16.63	35.05	22.17	39.78	
L_{d1} [nH]	5.25	4.11	3.56	11.77	1.02	6.41	3.5	
R_d [$m\Omega$]	72.15	39.91	88.17	82.46	92.24	96.5	1.36	
R_{dp} [Ω]	9.43	6.17	9.46	11.58	17.9	7.21	1	
L_{s2} [nH]	20.81	11.89	7.88	16.2	13.89	29.13	36.95	
L_{s1} [nH]	2.28	1.7	3.45	1.14	4.72	1.11	1	
R_s [$m\Omega$]	6.17	3.26	0.64	8.11	5.99	7.37	8.75	
L_g [nH]	101.19	49.04	130.74	69.49	138.93	30.1	136.71	
L_v [nH]	0.7	0.54	1.2	0.18	1.36	0.44	0.31	
C_{jo} [pF]	1046.16	318.94	542.54	1277.46	913.31	1252.73	1244.78	
V_j [V]	1.99	0.62	1.99	1.98	2.26	1	2.7	
M [$-$]	0.58	0.04	0.56	0.56	0.62	0.63	0.54	
K [$-$]	0.97	0.27	0.86	0.57	1.27	1.17	0.99	
k_1 [$e-9$]	0.95	0.09	1	0.97	0.99	1	0.8	
k_2 [$-$]	0.38	0.03	0.38	0.4	0.37	0.4	0.34	
C_{gs} [pF]	1491.17	122.76	1275.62	1544.75	1528.41	1523.66	1583.41	
V_{th} [V]	3.22	0.82	3.38	1.93	3.93	3.86	3.01	
R_{gi} [Ω]	2.97	0.9	3.99	3.01	1.64	3.55	2.68	
D_{tt} [ns]	0.61	0.37	0.92	0.82	0.87	0.32	0.1	
f_{min} [$-$]	2.5925	0.41924	2.1151	3.0529	2.5661	2.2527	2.9759	

Table E.2: C3M0060065K parameters fitted at 20 A over 5 runs.

C3M0060065K 30A								
Parameter		$Mean_{30}$	Std_{30}	Run1	Run2	Run3	Run4	Run5
R_L	[$m\Omega$]	38.7	29.03	35.49	82.7	32.84	1.48	40.97
C_p	[pF]	311.68	44.62	381.72	262.49	313.81	286.6	313.8
R_{DC}	[$m\Omega$]	13.17	3.45	14.94	10.57	11.77	18.37	10.21
L_{loop}	[nH]	18.35	0.33	18.52	17.96	18.4	18.78	18.11
L_{d2}	[nH]	28.63	6.49	27.54	39.8	23.42	24.86	27.53
L_{d1}	[nH]	1.83	0.86	1.02	1.46	3.09	1.26	2.34
R_d	[$m\Omega$]	75.35	17.8	97.3	61.1	53.66	85.08	79.59
R_{dp}	[Ω]	57.27	51.06	6.18	75.39	77.82	122.61	4.37
L_{s2}	[nH]	13.7	7.6	16.63	1.03	19.72	12.46	18.64
L_{s1}	[nH]	2.69	1.74	1.37	4.13	1.82	4.98	1.17
R_s	[$m\Omega$]	6.61	3.76	9.92	6.92	0.17	8.36	7.67
L_g	[nH]	133.56	29.87	149.83	148.95	139.93	148.47	80.61
L_v	[nH]	0.13	0.2	0.06	0.03	0.49	0.05	0.03
C_{jo}	[pF]	626.63	177.34	501.32	718.47	897.6	504.01	511.75
V_j	[V]	2.01	0.78	1.18	1.34	2.98	2.59	1.97
M	[$-$]	0.43	0.15	0.56	0.3	0.61	0.38	0.3
K	[$-$]	0.66	0.18	0.57	0.5	0.96	0.62	0.63
k_1	[$e-9$]	0.96	0.07	0.99	1	0.99	0.83	1
k_2	[$-$]	0.38	0.02	0.39	0.4	0.37	0.35	0.38
C_{gs}	[pF]	1319.99	107.54	1209.52	1209.59	1432.97	1414.47	1333.39
V_{th}	[V]	2.94	0.7	2.88	2.43	4.15	2.47	2.76
R_{gi}	[Ω]	2.85	1.24	3.99	3.94	0.97	2.47	2.86
D_{tt}	[ns]	0.25	0.09	0.19	0.15	0.31	0.37	0.23
f_{min}	[$-$]	2.7097	0.10746	2.6335	2.6973	2.5838	2.791	2.843

Table E.3: C3M0060065K parameters fitted at 30 A over 5 runs.

E.2 IMZA65R048M1H

IMZA65R048M1H 10A								
Parameter		$Mean_{10}$	Std_{10}	Run1	Run2	Run3	Run4	Run5
R_L	[$m\Omega$]	79.21	28.39	88.56	98.32	81.79	29.91	97.46
C_p	[pF]	196.13	47.72	216.73	222.76	207.88	111.43	221.87
R_{DC}	[$m\Omega$]	17.47	10.73	1	29.04	17.01	24.69	15.62
L_{loop}	[nH]	14.67	1.37	12.53	15.93	14.81	15.77	14.34
L_{d2}	[nH]	26.03	11.38	27.37	6.99	32.32	36.68	26.78
L_{d1}	[nH]	2.02	1.09	1	1	3.36	2.92	1.8
R_d	[$m\Omega$]	43.47	32.96	53.43	18.58	49.42	5.83	90.09
R_{dp}	[Ω]	83.21	101.11	154.87	14.24	225.96	5.79	15.18
L_{s2}	[nH]	9.14	10.2	4.94	27.03	1.36	5.69	6.7
L_{s1}	[nH]	4.43	1.49	5.11	5.98	5.36	2.6	3.08
R_s	[$m\Omega$]	5.29	3.56	0.39	8.07	7.48	7.9	2.59
L_g	[nH]	21.09	15.06	10	22.37	46.38	16.69	10
L_v	[nH]	3.24	1.29	3.94	1.85	3.96	1.87	4.6
C_{jo}	[pF]	727.85	420.09	502.41	505.51	542.83	1475.06	613.44
V_j	[V]	1.67	0.78	1	2.11	1.33	1.06	2.83
M	[$-$]	0.65	0.1	0.7	0.7	0.7	0.48	0.68
K	[$-$]	0.6	0.16	0.87	0.5	0.5	0.52	0.59
k_1	[$e-9$]	0.45	0.06	0.46	0.4	0.4	0.44	0.55
k_2	[$-$]	0.25	0.02	0.24	0.23	0.24	0.25	0.28
C_{gs}	[pF]	1070.14	178.46	1147.59	1285.34	1016.35	1098.48	802.96
V_{th}	[V]	4.87	0.51	5.49	4.43	4.36	4.76	5.32
R_{gi}	[Ω]	2.01	1.26	0.75	0.73	3.11	2.08	3.38
D_{tt}	[ns]	0.24	0.06	0.22	0.14	0.26	0.29	0.28
f_{min}	[$-$]	1.188	0.097416	1.175	1.0998	1.114	1.208	1.3434

Table E.4: IMZA65R048M1H parameters fitted at 10 A over 5 runs.

IMZA65R048M1H 20A								
Parameter		$Mean_{20}$	Std_{20}	Run1	Run2	Run3	Run4	Run5
R_L	[$m\Omega$]	54.86	38.25	99.94	20.66	9.55	73.59	70.56
C_p	[pF]	162.08	53.35	96.26	175.93	234.82	125.69	177.71
R_{DC}	[$m\Omega$]	12.72	5.19	10.25	5.65	12.15	18.13	17.4
L_{loop}	[nH]	17.11	0.38	16.71	16.74	17.16	17.61	17.32
L_{d2}	[nH]	26.29	8.02	18.34	39.2	26.34	26.5	21.08
L_{d1}	[nH]	3.45	2.58	3.79	1.48	7.73	1.5	2.73
R_d	[$m\Omega$]	58.93	31.05	40.76	18.15	64.98	70.74	100
R_{dp}	[Ω]	24.84	11.72	21.39	7.14	32.26	25.55	37.84
L_{s2}	[nH]	6.49	6.6	16.03	1.01	1.31	3.48	10.62
L_{s1}	[nH]	2	1.59	1.39	1.81	1	4.78	1.01
R_s	[$m\Omega$]	3.47	3.89	0.24	4.12	9.69	3.1	0.18
L_g	[nH]	31.16	14.7	18.66	18.2	25.68	42.57	50.66
L_v	[nH]	4.17	0.95	2.63	4.99	3.93	4.54	4.77
C_{jo}	[pF]	668.82	100.29	744.95	744.97	512.98	623.98	717.23
V_j	[V]	2.5	0.84	2.98	2.91	1.03	3	2.55
M	[$-$]	0.54	0.16	0.3	0.64	0.57	0.48	0.7
K	[$-$]	0.65	0.17	0.5	0.58	0.55	0.93	0.71
k_1	[$e-9$]	0.48	0.11	0.41	0.68	0.45	0.41	0.47
k_2	[$-$]	0.25	0.02	0.24	0.29	0.25	0.24	0.25
C_{gs}	[pF]	1094.61	151.46	1293.73	877.07	1155.04	1062.24	1084.98
V_{th}	[V]	4.73	0.58	4.03	4.96	4.28	5.49	4.92
R_{gi}	[Ω]	0.72	0.7	0.39	0.32	0.64	1.95	0.3
D_{tt}	[ns]	0.51	0.74	0.34	1.81	0.11	0.18	0.11
f_{min}	[$-$]	1.5222	0.1091	1.3745	1.6238	1.5052	1.635	1.4727

Table E.5: IMZA65R048M1H parameters fitted at 20 A over 5 runs.

IMZA65R048M1H 25A								
Parameter	$Mean_{25}$	Std_{25}	Run1	Run2	Run3	Run4	Run5	
R_L [m Ω]	71.13	40.75	75.13	1	80.29	99.52	99.72	
C_p [pF]	177.46	59.55	138.8	201.53	213.12	94.08	239.8	
R_{DC} [m Ω]	15.04	2.48	15.01	16.25	11.02	15.27	17.68	
L_{loop} [nH]	16.93	0.42	16.37	17.46	16.98	17.16	16.68	
L_{d2} [nH]	31.63	7.65	23.71	39.97	39.51	29.16	25.8	
L_{d1} [nH]	2.49	1.53	3.46	2.19	1.2	4.59	1	
R_d [m Ω]	85.15	20.33	100	95.87	99.98	76.81	53.09	
R_{dp} [Ω]	6.97	7.81	19.3	1	1.01	3.58	9.97	
L_{s2} [nH]	11.37	10.02	1.02	21.87	20.85	1.67	11.44	
L_{s1} [nH]	2.97	3.46	9.12	1.01	1.92	1.79	1.02	
R_s [m Ω]	5.95	2.37	8.42	2.91	8.32	4.94	5.17	
L_g [nH]	38.93	29.38	10.07	50.15	29.33	84.43	20.7	
L_v [nH]	4.59	0.42	4.21	4.83	4.83	5	4.06	
C_{jo} [pF]	987.62	379.11	1245.19	1052.96	1425.69	713.96	500.3	
V_j [V]	1.33	0.52	1.36	2.21	1	1.05	1.01	
M [–]	0.49	0.17	0.4	0.69	0.66	0.3	0.41	
K [–]	0.63	0.15	0.5	0.88	0.62	0.65	0.5	
k_1 [e-9]	0.56	0.07	0.66	0.48	0.59	0.53	0.53	
k_2 [–]	0.27	0.02	0.29	0.25	0.28	0.26	0.26	
C_{gs} [pF]	924.81	104.09	907.17	1017.31	801.71	853.66	1044.22	
V_{th} [V]	4.74	0.56	4.42	5.5	4.93	4.84	4.02	
R_{gi} [Ω]	0.79	0.65	0.3	0.33	1.88	0.8	0.64	
D_{tt} [ns]	2.84	1.82	0.29	1.8	3.92	4.93	3.25	
f_{min} [–]	1.365	0.065579	1.4043	1.2654	1.3376	1.3853	1.4326	

Table E.6: IMZA65R048M1H parameters fitted at 25 A over 5 runs.

E.3 Mean and Standard Deviations

-		Range		C3M0060065K			IMZA65R048M1H		
		Min	Max	<i>Std</i> ₁₀	<i>Std</i> ₂₀	<i>Std</i> ₃₀	<i>Std</i> ₁₀	<i>Std</i> ₂₀	<i>Std</i> ₂₅
R_L	[$m\Omega$]	1	100	29.84	40.17	29.03	28.39	38.25	40.75
C_p	[pF]	25	1000	32.76	64.42	44.62	47.72	53.35	59.55
R_{DC}	[$m\Omega$]	1	100	3.9	4.07	3.45	10.73	5.19	2.48
L_{loop}	[nH]	5	100	0.96	0.76	0.33	1.37	0.38	0.42
L_{d2}	[nH]	1	40	0.08	10.22	6.49	11.38	8.02	7.65
L_{d1}	[nH]	1	40	1.06	4.11	0.86	1.09	2.58	1.53
R_d	[$m\Omega$]	0.1	100	8.54	39.91	17.8	32.96	31.05	20.33
R_{dp}	[Ω]	1	500	20.45	6.17	51.06	101.11	11.72	7.81
L_{s2}	[nH]	1	40	9.41	11.89	7.6	10.2	6.6	10.02
L_{s1}	[nH]	1	40	1.93	1.7	1.74	1.49	1.59	3.46
R_s	[$m\Omega$]	0.1	10	4.69	3.26	3.76	3.56	3.89	2.37
L_g	[nH]	10	150	67.02	49.04	29.87	15.06	14.7	29.38
L_v	[nH]	0.01	5	1.49	0.54	0.2	1.29	0.95	0.42
C_{jo}	[pF]	500	1500	127.29	318.94	177.34	420.09	100.29	379.11
V_j	[V]	1	3	0.47	0.62	0.78	0.78	0.84	0.52
M	[–]	0.3	0.7	0.06	0.04	0.15	0.1	0.16	0.17
K	[–]	0.5	1.5	0.09	0.27	0.18	0.16	0.17	0.15
k_1	[e-9]	0.4	1	0	0.09	0.07	0.06	0.11	0.07
k_2	[–]	0.2	0.4	0	0.03	0.02	0.02	0.02	0.02
C_{gs}	[pF]	800	2700	239.15	122.76	107.54	178.46	151.46	104.09
V_{th}	[V]	1.5	5.5	0.31	0.82	0.7	0.51	0.58	0.56
R_{gi}	[Ω]	0.3	4	1.33	0.9	1.24	1.26	0.7	0.65
D_{tt}	[ns]	0.1	15	0.2	0.37	0.09	0.06	0.74	1.82

Table E.7: Standard deviation for fitted parameters for C3M0060065K and IMZA65R048M1H.

-		Range		C3M0060065K			IMZA65R048M1H		
		Min	Max	<i>Mean</i> ₁₀	<i>Mean</i> ₂₀	<i>Mean</i> ₃₀	<i>Mean</i> ₁₀	<i>Mean</i> ₂₀	<i>Mean</i> ₂₅
R_L	[$m\Omega$]	1	100	60.6	41.16	38.7	79.21	54.86	71.13
C_p	[pF]	25	1000	211.68	269.61	311.68	196.13	162.08	177.46
R_{DC}	[$m\Omega$]	1	100	14.96	12.91	13.17	17.47	12.72	15.04
L_{loop}	[nH]	5	100	19.14	18.83	18.35	14.67	17.11	16.93
L_{d2}	[nH]	1	40	39.92	30.23	28.63	26.03	26.29	31.63
L_{d1}	[nH]	1	40	1.64	5.25	1.83	2.02	3.45	2.49
R_d	[$m\Omega$]	0.1	100	86.01	72.15	75.35	43.47	58.93	85.15
R_{dp}	[Ω]	1	500	21.19	9.43	57.27	83.21	24.84	6.97
L_{s2}	[nH]	1	40	23.42	20.81	13.7	9.14	6.49	11.37
L_{s1}	[nH]	1	40	2.97	2.28	2.69	4.43	2	2.97
R_s	[$m\Omega$]	0.1	10	6.32	6.17	6.61	5.29	3.47	5.95
L_g	[nH]	10	150	72.72	101.19	133.56	21.09	31.16	38.93
L_v	[nH]	0.01	5	2.25	0.7	0.13	3.24	4.17	4.59

Table E.8: Comparison of fitted circuit parameters for C3M0060065K and IMZA65R048M1H.

-	-	Range		C3M0060065K			IMZA65R048M1H		
Param.	Def.	Min	Max	$Mean_{10}$	$Mean_{20}$	$Mean_{30}$	$Mean_{10}$	$Mean_{20}$	$Mean_{25}$
C_{jo} [pF]	957	500	1500	588.45	1046.16	626.63	727.85	668.82	987.62
V_j [V]	2.14	1	3	2.35	1.99	2.01	1.67	2.5	1.33
M [-]	0.54	0.3	0.7	0.57	0.58	0.43	0.65	0.54	0.49
K [-]	-	0.5	1.5	0.74	0.97	0.66	0.6	0.65	0.63
k_1 [e-9]	0.70	0.4	1	1	0.95	0.96	0.45	0.48	0.56
k_2 [-]	0.36	0.2	0.4	0.4	0.38	0.38	0.25	0.25	0.27
C_{gs} [pF]	1100	800	2700	1342.47	1491.17	1319.99	1070.14	1094.61	924.81
V_{th} [V]	2.9	1.5	5.5	3.65	3.22	2.94	4.87	4.73	4.74
R_{gi} [Ω]	3	0.3	4	2.11	2.97	2.85	2.01	0.72	0.79
D_{tt} [ns]	0.5	0.1	15	0.36	0.61	0.25	0.24	0.51	2.84

Table E.9: Comparison of fitted device parameters, relative to default values, for C3M0060065K and IMZA65R048M1H.