Sindre Bjørbekk Kongerød

Modeling of SiC MOSFETs and Parameter Fitting Using a Genetic Algorithm

Master's thesis in Energy and Environmental Engineering

Supervisor: Assoc. Prof. Dimosthenis Peftitsis

June 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Kunnskap for en bedre verden

Sindre Bjørbekk Kongerød

Modeling of SiC MOSFETs and Parameter Fitting Using a Genetic Algorithm

Master’s thesis in Energy and Environmental Engineering
Supervisor: Assoc. Prof. Dimosthenis Peftitsis
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Norwegian University of
Science and Technology

Acknowledgement

In this segment I would like to acknowledge those who have helped me in my work. Associate Professor Dimosthenis Peftitsis and Ph.D Candidate Ole Christian Spro have been my supervisors and their guidance and support has been vital in my work. I thank Dimosthenis for his motivation and positivity, encouraging me in my work. Ole Christian has been my day-to-day supervisor and has helped me with difficulties I have had. His guidance has always challenged me to think and given me pointers which have helped me understand rather than simply giving me the easiest answer. This has helped me grow as a person. I thank Ole Christian for using many hours of his time to guide and motivate me in my work.

Furthermore, my appreciation also goes to Ph.D Candidate Daniel Alexander Philipps for discussing different topics with me and giving me feedback. This has helped me improve my work.

Lastly, I would like to thank my family and friends for believing in me and encouraging me. This helped me overcome difficult times and ultimately finish my studies!

Abstrakt

Denne masteroppgaven demonstrerer bruken av en genetisk algoritme til å optimalisere parametere i simuleringsmodeller for å nøyaktig representere målte statiske egenskaper til en SiC MOSFET ved romtemperatur. Ulike SiC MOSFET modeller funnet i litteraturen er undersøkt og klassifisert. To modeller fra produsenter er valgt for optimaliseringsprosessen; en forbedret modell basert på Cree-modellen og modellen fra ROHM. En genetisk algoritme implementeres og brukes for å tilpasse de to modellene til eksperimentelle måledata. De statiske egenskapene - også kjent som IV-kurver eller IV-karakteristikk - for en kommersiell SiC MOSFET (C3M0075120D) ble målt ved bruk av en kommersiell curve tracer.

To metoder for å oppnå karakteristikkene ved bruk av curve traceren blir undersøkt og diskutert. Blant testinnstillingene på maskinen har det vist seg at pulsbredden til påført spenning eller strømpuls påvirker målingene sterkt. Den indre temperaturen i enheten som måles blir undersøkt gjennom simuleringer mens den påfører samme puls som den opplever under karakteriseringen. Ofte antas den indre temperaturen å være tilnærmet lik den ytre temperaturen for korte testpulser. Ved å bruke produsentens termiske modell av enheten, ble det funnet at den indre temperaturen øker til nærmere 65^oC for flere av karakteriseringspunktene under en fullstendig curve tracer test. Følgelig er den indre temperaturen større enn den ytre temperaturen når målingene blir utført, og denne feilen kan overføres til modelleringen av den samme enheten. Disse termiske undersøkelsene blir verifisert ved termiske beregninger ved bruk av den termiske impedansen fra enhetens datablad.

I dette arbeidet blir temperaturøkningen under eksperimentell karakterisering tatt med i beregningen av optimaliseringsalgoritmen. Dette oppnås ved å sette den indre temperaturen til verdiene oppnådd fra termiske beregninger på hvert målepunkt i simuleringene. Den genetiske algoritmen justerer modellparametrene slik at de passer til måledataene. Begge modellene oppnår parameterverdier som gir en samlet tilpassing av måledataene. Imidlertid har ROHM-modellen bedre passform i alle kjøringene av algoritmen.

Abstract

This master thesis demonstrates the use of a genetic algorithm to optimize parameters of simulation models to accurately represent the measured static characteristics of a SiC MOSFET at room temperature. Various models in the literature for simulating SiC MOSFETs are investigated and classified. Two models from device manufacturers are selected for the optimization process; an improved model based on the Cree model and the model from ROHM. A genetic algorithm is implemented and used to fit the two models to experimental measurement data. The static characteristics – also known as IV-curves – of a commercial SiC MOSFET (C3M0075120D) were measured using a commercial curve tracer.

Two methods for obtaining the characteristics when using the curve tracer are investigated and discussed. Among the test settings of the machine, the pulse width of the applied voltage or current pulse has been found to strongly influence the measurements. The junction temperature of the device-under-test is investigated through simulations while applying the same pulse as it experiences during characterization. Often, the junction temperature is assumed to be approximately equal to the case temperature for short test pulses. By using the manufacturer provided thermal model of the device, it is found that the junction temperature increases up to approximately 65°C for several of the characterization points during a complete curve tracer test. Hence, the junction temperature is greater than the case temperature when the measurements are taken, and this error can carry over to the modelling of that same device. These thermal investigations are verified by thermal calculations using the thermal impedance from device datasheet.

In this work, the temperature increase during experimental characterization is taken into account by the optimization algorithm. This is accomplished by setting the junction temperature to the values obtained from thermal calculations at each measurement point in the simulations. The genetic algorithm adjusts the model parameters to fit the measurement data. Both models obtain parameter values that provide an overall fit of the measurement data. However, the ROHM model is found to have better fit in all runs of the algorithm.

Contents

Acknowledgement **i**

Abstrakt **ii**

Abstract **iii**

List of Abbreviations **vi**

1 Introduction **1**
1.1 Scope of the Thesis . . . 1
1.2 Outline . . . 1

2 Semiconductor Devices **2**
2.1 The Physics of Semiconductor Devices . . . 2
2.1.1 Doped Semiconductors . . . 3
2.1.2 Current Flow . . . 4
2.1.3 pn-junction . . . 5
2.2 Semiconductor Device Structures and Characteristics . . . 6
2.2.1 Bipolar Junction Transistor . . . 6
2.2.2 Metal-Oxide-Semiconductor Field-Effect Transistor . . . 7
2.2.3 Insulated Gate Bipolar Transistors . . . 9
2.2.4 Capacitor Model of a MOSFET . . . 10
2.2.5 Transconductance and IV-Characteristics . . . 11
2.3 Switching Events of MOSFETs . . . 13
2.3.1 Turn-on Waveforms . . . 13
2.3.2 Turn-off Waveforms . . . 15
2.4 Si and SiC Comparison . . . 16

3 Modeling of SiC MOSFETs **18**
3.1 What Makes a Good MOSFET Model? . . . 18
3.2 SiC MOSFET Models . . . 18
3.2.1 Established Models . . . 20
3.2.2 Charge-Based Model . . . 20
3.2.3 Non-segmented Model . . . 21
3.2.4 Electro-Thermal Model . . . 23
3.2.5 Curtice-Ettenberg Based Model . . . 24
3.3 Manufacturer Models . . . 26
3.3.1 Cree Wolfspeed: C3M0075120K SPICE Model . . . 26
3.3.2 STMicroelectronics: SCT30N120_V3 SPICE Model . . . 29
3.3.3 ROHM: SCT2080KE SPICE Model . . . 32
3.4 Universal Model . . . 35

4 Method of Data Extraction and Optimization **36**
4.1 Curve Tracer . . . 36
4.1.1 Load Line . . . 38
4.1.2 Comparison of Measurement Methods . . . 39
4.2 Parameter Extraction and Optimization . . . 46
4.2.1 Evolutionary Algorithms . . . 46
4.2.2 Genetic Algorithm . . . 47
4.2.3 Implementation of the GA . . . 48

5 Results **59**
5.1 Fitting of IV-curves . . . 59
5.2 Evaluation of the Algorithm . . . 61

6 Conclusions **68**
6.1 Further Work . . . 68

Appendices **70**

A IV-characteristics Simulation Circuit **70**

B CV-characteristics Simulation Circuits **71**
B.1 Ciss Simulation Circuit . . . 71
B.2 Coss Simulation Circuit . . . 72
B.3 Crss Simulation Circuit . . . 72

C Junction Temperature Simulation Circuit **73**

D Verification of Temperature Simulations **74**

E Setting Junction Temperature LTSpice **76**

References **77**

List of Abbreviations

AC	Alternating Current
ACO	Ant Colony Optimization
BJT	Bipolar Junction Transistor
BSIM	Berkeley Short-channel IGFET Model
BVS	Behavioral Voltage Source
CMOS	Complementary Metal–Oxide–Semiconductor
CT	Curve Tracer
DC	Direct Current
EA	Evolutionary Algorithm
EKV	Enz-Krummenacher-Vittoz
FET	Field-Effect Transistor
GA	Genetic Algorithm
HV	High Voltage
IGBT	Insulated-Gate Bipolar Transistor
JFET	Junction-Field-Effect Transistor
LV	Low Voltage
MOSFET	Metal-Oxide-Semoconductor Field-Effect-Transistor
NSGA	Non-dominated Sorting Genetic Algorithm
OOP	Object-Oriented Programming
PSO	Particle Swarm Optimization
RC	Resistor-Capacitor
RMSE	Root Mean Square Error
SPICE	Simulation Program with Integrated Circuit Emphasis
VCCS	Voltage-Controlled Current Source
VCVS	Voltage-Controlled Voltage Source
WBG	Wide Bandgap

1 Introduction

Modeling of real systems and components is a major contributor to technological innovation and developments by abstracting ideas, systems or processes into a manageable and understandable set of rules or equations. This set of rules or equations can, in many cases, be used to solve or optimize problems, achieving desired outcomes. Models of real components are used in many design processes to find potential flaws of a system before implementing a prototype version. Furthermore, the use of simulation tools as part of a product design process is becoming increasingly important in the power electronic industry. By using simulated circuits, the design space can be explored in less time and at lower cost compared to prototype based design processes. Accurate circuit simulations can also be used in automated design and optimization processes to reduce workload of engineers or explore design spaces that are not directly accessible to design engineers. E.g. investigate operational parameters on junction temperature of a semiconductor component, which is otherwise impossible to measure in practice. Additionally, simulation tools can be used for troubleshooting circuits and converters that are not behaving as expected. Moreover, new power semiconductors made from wide band gap materials open new possibilities for designing power electronic converters, such as increased power density and efficiency at lower system costs. This is suitable for a wide range of applications like industry, marine applications and utilization of renewable energy. However, to fully explore the potential of wide band gap materials, advanced circuitry and accurate models may be required. Accurate models can be used in simulation of new power converters that takes advantage of the wide band gap materials.

1.1 Scope of the Thesis

The thesis is written as part of the energy and environment master of technology program at NTNU. It is a continuation of a semester project from previous semester [1]. However, the focus has shifted from advanced gate drivers to modeling of SiC MOSFETs. The objectives are to present different MOSFET modeling approaches, $I_D V_{DS}$-characterization of MOSFETs and optimization of models using a genetic algorithm. $I_D V_{DS}$-characterizations using a curve tracer are performed to obtain measurement data that is used in the genetic algorithm. Changes on junction temperature during curve tracer measurements are investigated. The measurements are limited to one device (C3M0075120D), to a single case temperature (25^oC) and static characteristics. Furthermore, the genetic algorithm is made in the Python programming language and simulations of certain models are performed in LTspice as part of the optimization process by the genetic algorithm.

1.2 Outline

Firstly, theory of semiconductor devices is presented in section 2. Section 2 covers semiconductor physics and examples of different devices with focus on SiC MOSFETs. This will serve as a foundation to understanding certain elements of SiC MOSFET modeling. Secondly, different SiC MOSFET modeling approaches are presented in section 3. A range of different models found in the literature are presented and classified. Additionally, three manufacturer models are discussed and reviewed. Thirdly, section 4 covers the method of data extraction and optimization. The equipment will be introduced and two different methods of performing the characterization is discussed. Effects on the junction temperature during the measurements are investigated. Furthermore, the genetic algorithm will be explored in detail. Finally, results from several runs of the genetic algorithm are presented, discussed and concluded in section 5 and 6.

2 Semiconductor Devices

Semiconductor devices are widely used in electronics and communication technologies. These devices utilize the properties of semiconductor materials. Today, silicon (Si) is the dominating material used in semiconductor devices in power electronics. However, silicon based semiconductor devices are reaching their theoretical limit. Therefore, efforts in both research and development have targeted new and better semiconductor materials. This may allow the development of new components with characteristics better suited for various applications. Higher efficiency, higher switching frequency and higher operating temperature are the most important characteristics in the development of new power electronic devices. Throughout the following subsections, knowledge about semiconductor devices is presented. Even though the presented theory is based on silicon semiconductors, the same concepts of carriers, doping and device structure are applicable to semiconductor devices based on other materials such as silicon carbide (SiC), which is the selected semiconductor material for this thesis.

2.1 The Physics of Semiconductor Devices

The characteristics of semiconductor materials range between conductor and insulator materials. An electric field applied to a material will cause a current to flow if there are free charge carriers [2]. The density of free charge carriers varies greatly from material to material. Most metals have a large density, e.g. copper has a density in the order of 10^{23} per cm^3, while other materials such as quartz have densities less than 10^3 per cm^3. Semiconductors have a density somewhere in between that of conductors and insulators (10^8-10^{19} per cm^3) [2]. The density in semiconductors can be changed by orders of magnitude by introducing impurities as will be discussed in section 2.1.1.

Silicon (Si) is the dominating semiconductor material. Si with atom number 14 in the periodic table, has four valence electrons. Valence electrons are the electrons located at the outer shell of an atom. These are the electrons that participate in the formation of a chemical bond and may be gained or lost. Due to having 4 valence electrons, Si atoms will lay in an array in a single crystal.

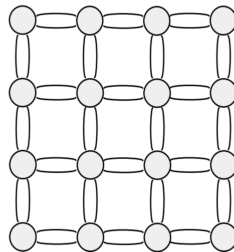
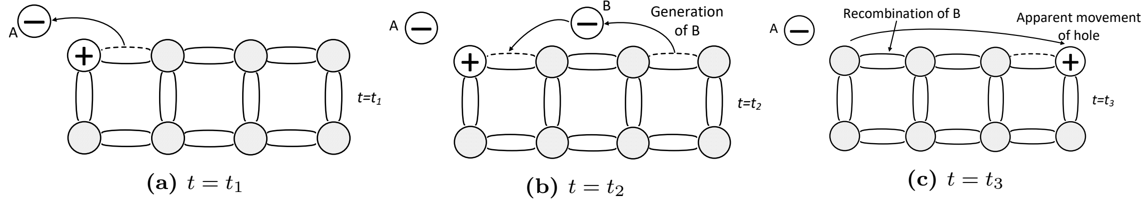


Figure 2.1: Silicon in array, taken from [2]

Each Si atom is bound to four other Si atoms by covalent electron bonds. Covalent binding means that the electrons are shared between the two atoms. These bonds may be broken by a process called thermal ionization. This means that the random thermal motion around the atom's equilibrium position may break the bonds. A free electron and a positive charge, called hole, is created when the bond is broken. Consequently, allowing the electron to move and recombine at another hole location or simply recombine at the same place. Figure 2.2 illustrates the process of thermal ionization; generation and movement of holes and free electrons in pure silicon material.



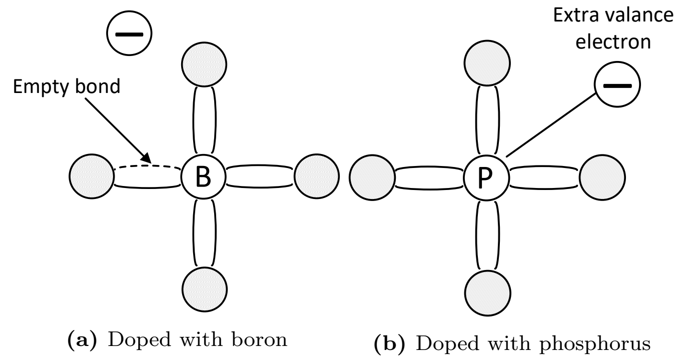
(a) $t = t_1$ **(b)** $t = t_2$ **(c)** $t = t_3$

Figure 2.2: Process of thermal ionization and hole movement ($t_1 < t_2 < t_3$), taken from [2]

At time t_1 in figure 2.2a an electron, named A, breaks the bond and becomes a free electron inside the material. Some time later, t_2, another electron, named B, may be attracted to the positive charge and breaks free from its bond somewhere else in the material, illustrated by figure 2.2b. As electron B is attracted towards the positive hole generated by electron A earlier, it may take that place and become bound again at time t_3. Figure 2.2c shows the recombination of electron B and illustrates the apparent movement of the positive hole generated by thermal ionization of electron A. The process of thermal ionization will increase with temperature. Higher temperature means greater thermal motion, subsequently leading to higher chances of breaking the bonds. Regardless, pure Si will always generate an equal amount of free electrons and holes at any time as a result of thermal ionization.

2.1.1 Doped Semiconductors

Adding atoms of other elements to the Si array may change the thermal equilibrium density of electrons. Boron (B) or phosphorus (P) can be introduced to create excess electrons or holes. Boron has 3 valence electrons. That means in Si doped with B, the B tends to accept a free electron and completes the last bond with Si, figure 2.3a. Hence, free electrons from thermal ionization get taken by B, leaving positive holes free to move in the material. This type of silicon doping is called p-type. In a p-type Si material there are more holes than electrons. In this case, holes are the majority carriers and electrons minority carriers.



(a) Doped with boron **(b)** Doped with phosphorus

Figure 2.3: Illustration of doped silicon material, taken from [2]

The opposite occurs when introducing phosphorus. P has 5 valence electrons, meaning that in the Si array there will be additional electrons not in bonds with the Si atoms, figure 2.3b.

Furthermore, the P atom is easily ionized, subsequently releasing the fifth valence electron. This electron is donated to the Si array and creates more free electrons than holes. Hence, a negatively doped Si material, called n-type, results. In an n-type Si material the electrons are the majority carriers while the holes are the minority carriers.

2.1.2 Current Flow

Two mechanisms control the movement of the free carriers; drift and diffusion. Drift happens when an electric field is applied to the material. In the presence of an electric field, free carriers are accelerated. Thus, moving them parallel to the field as shown in figure 2.4.

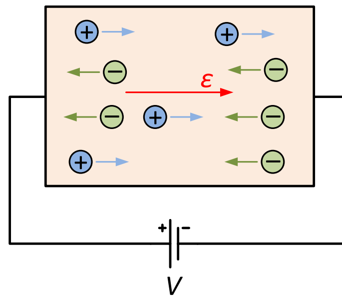


Figure 2.4: Free carrier movement when applied to an electric field [3]

Positive and negative carriers will move parallel to the field, but in opposite directions. The amount of drift current depends on the size of the electric field, the amount of free carriers and the mobility of the carriers in the material.

Diffusion on the other hand, is a result of spatial density variations of the free carriers. Figure 2.5 shows the cause of diffusion current.

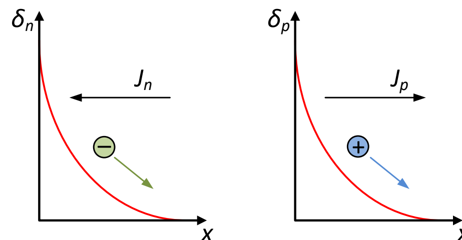


Figure 2.5: Free carrier movement due to spatial density variations [3]

δn and δp represent negative and positive free carrier density respectively. The x-axis represents different positions in the material. As the figure shows, when the spatial carrier density varies inside the material, a movement of carriers from the high density area to the lower density area within the material occurs. Thus, the diffusion current density will be the sum of the electron current density J_n and the hole current density J_p.

Different semiconductor devices capitalize on these effects differently. Some devices utilize the diffusion effect while others use the drift effect. For example, BJTs utilize the diffusion effect while MOSFETs are based on the drift effect.

2.1.3 pn-junction

A pn-juntion is formed by doping one part of the semiconductor material n-type and doping the other side to a p-type. In a pn-junction there will be an inherent electric field at the border between the layers, called junction, as shown in figure 2.6.

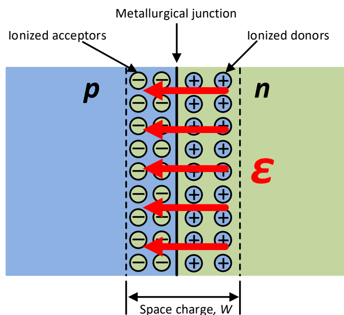


Figure 2.6: Space charge in a pn-junction [3]

The field originates as a result of majority carriers on both sides diffusing over to the lower density side. On each side of the junction, a small region has received majority carriers from the other side. I.e. electrons from the n-type layer diffuse over to the p-type layer and holes from p-type layer diffuse over to the n-type layer. These two regions create a space charge, also called depletion layer. Furthermore, the generated electric field will set up a drift current, pushing the free carriers back to their respective layers. An equilibrium is reached when the diffusion current equals the drift current.

An external voltage source can change the generated field. When the applied voltage is positive on the p-region it will counter the space charge and the pn-junction is called forward biased. By increasing the applied voltage greater than that of the space charge, current will flow through the pn-junction. If the applied voltage is negative on the p-region, the space charge will increase. In such a state the pn-junction is said to be reverse biased. As a result, the pn-junction will block the majority of the current while in this state. However, a small current is still present, called leakage current. Leakage current is a result of minority carriers diffusing into and pushed across the field due to small variations in the minority densities which are caused by the applied voltage. In addition, a small contribution to the leakage current is made by thermal ionization occurring in the space charge. If the space charge region becomes too large as a result of increasing the applied voltage, the pn-junction will start to conduct. Electrons with high kinetic energy collides with atoms in the material and can break a covalent bond, creating another free electron and a hole. This process is called impact ionization [2]. Additionally, the newly freed electron may gain sufficient kinetic energy by the applied field to break other covalent bonds. Consequently, if this process cascades into a chain reaction, a large current is produced and will quickly destroy the device. The voltage level that makes this happen is called the breakdown voltage of the device. Pn-junctions enable more advanced semiconductor devices. Semiconductor devices are built up by several layers with various doping levels and types to achieve different properties.

2.2 Semiconductor Device Structures and Characteristics

This subsection will briefly discuss bipolar junction transistors (BJT), metal-oxide-semiconductor field-effect transistors (MOSFET) and insulated bipolar gate transistors (IGBT). However, since the focus of this thesis is on MOSFET devices, more detailed explanations and figures are given for the MOSFET structure.

2.2.1 Bipolar Junction Transistor

Bipolar junction transistors (BJT) have a four layer structure of alternating doping levels and types as shown in figure 2.7. The doping level and thickness of the layers have a significant effect on the characteristics of the device [2].

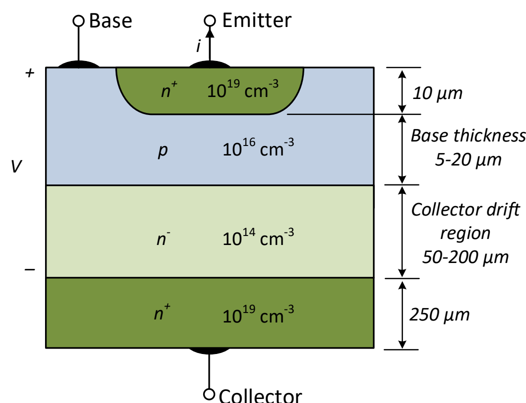


Figure 2.7: Structure of a BJT [3]

A BJT uses the diffusion effect for conducting current from collector to emitter. By injecting a constant base-emitter current, holes are introduced in the p-layer. This will introduce a change in the carrier densities in the layers, giving rise to a diffusion current from collector to emitter. In on-state, it requires a constant base-emitter current to be able to keep up the diffusion current. It takes some time to change the carrier densities in the materials, resulting in longer switching time. However, the conduction losses are low due to a small on-state resistance. In summary, the BJT is a current controlled device, where the base-emitter current is controlling the collector-emitter current flow.

2.2.2 Metal-Oxide-Semiconductor Field-Effect Transistor

Metal-oxide-semiconductor field-effect transistors (MOSFETs) have a four layer structure of alternating doping levels and types. For example, an n-channel MOSFET has an $n^+pn^-n^+$ structure as shown in figure 2.8.

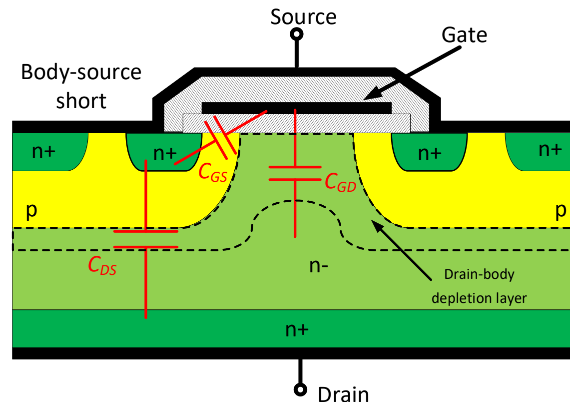
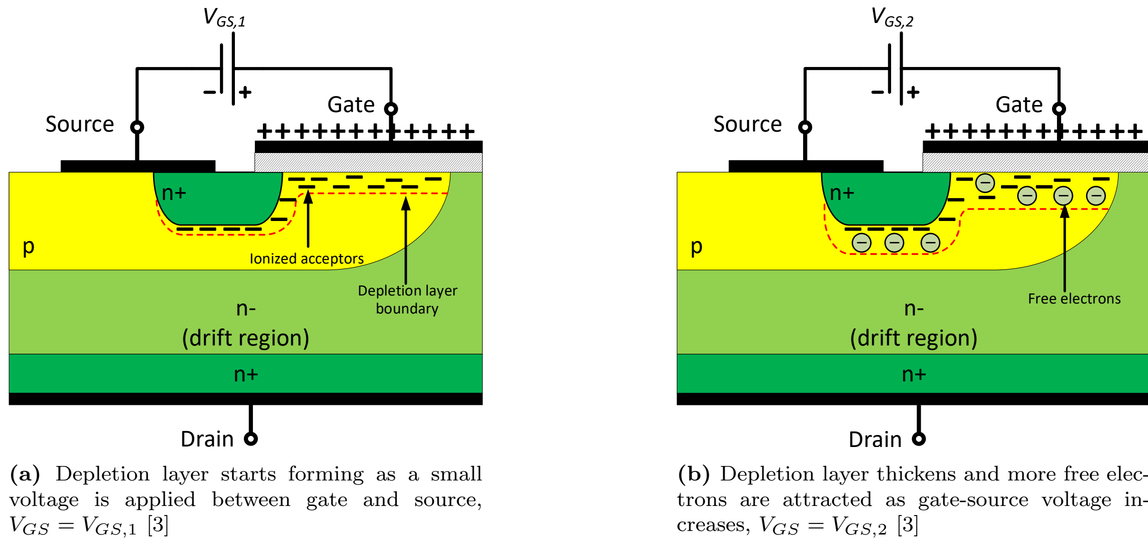


Figure 2.8: MOSFET structure, taken from [3]

The end n^+-layers at source and drain are approximately the same in terms of doping level. The n^+-layer located at source is encapsulated in a p-type middle layer, often termed body [2]. The n^--layer is called the drift region and has a lower doping level compared to the n^+-layers, hence the signs. This layer determines the breakdown voltage of the device. For a p-channel MOSFET the structure is the same with opposite doping profile, i.e. $p^+np^-p^+$.

When a voltage is applied to the gate, the carriers in the semiconductor material will be affected by the field. Carriers will move and create a depletion layer at the boundary of gate-oxide and p-layer. Figure 2.9 illustrates this process.



(a) Depletion layer starts forming as a small voltage is applied between gate and source, $V_{GS} = V_{GS,1}$ [3]

(b) Depletion layer thickens and more free electrons are attracted as gate-source voltage increases, $V_{GS} = V_{GS,2}$ [3]

Figure 2.9: Development of depletion layer as gate-source voltage increases, $V_{GS,1} < V_{GS,2}$

As a small positive gate-source voltage is applied, electrons will get attracted to the positive charge on the gate. Between the gate and the silicon material there is an oxide layer blocking the charge from passing through. Majority carriers in the p-layer (holes) will be repelled from the border between the gate and p-layer, figure 2.9a. I.e. electrons will occupy the holes and create negatively charged acceptors. Thus, a depletion layer is formed. When the gate-source voltage increases further, the depletion layer thickens to provide additional negative charge. In addition to repelling more holes, free electrons are also attracted to the depletion layer, figure 2.9b. Moreover, as this process continues, more free electrons are attracted and eventually the negative charge density becomes greater than the positive hole density in the p-layer. Consequently, a so called inversion layer forms at the border between the gate-oxide layer and p-layer. Since the electrons become the majority carriers in the inversion layer, the p-layer close to the border is effectively inverted to an n-layer, hence the name inversion layer. The voltage level at which point the inversion layer emerges is called the threshold voltage. Figure 2.10 illustrates the inversion layer.

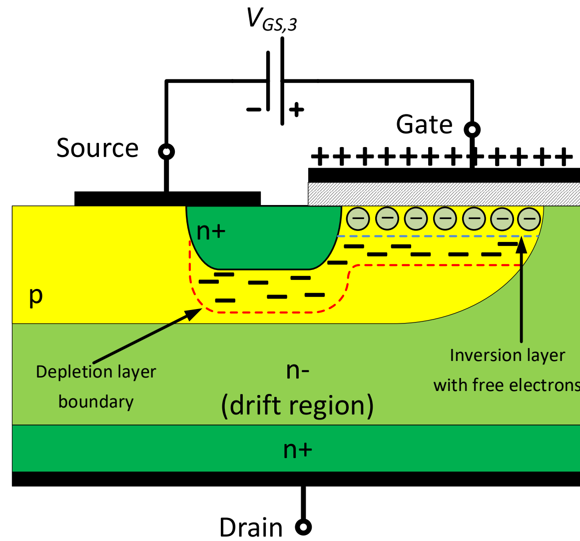


Figure 2.10: Inversion layer has been created, $V_{GS} = V_{GS,3}$ [3]

The inversion layer contains free electrons that are free to move, i.e. highly conducting. This opens a channel through the p-layer between the n-layers. Thus making it possible for the current to flow from drain to source.

After opening the channel, the gate voltage is increased further to enlarge the inversion layer, allowing higher current to pass through and making the on-state resistance of the MOSFET smaller. In conclusion, the MOSFET is a voltage controlled device, where the gate-source voltage controls the drain-source current flow.

2.2.3 Insulated Gate Bipolar Transistors

Insulated gate bipolar transistors (IGBTs) are a result of combining the best characteristics of both BJTs (low conduction losses) and MOSFETs (low switching losses). The IGBT consists of five alternating layers of different doping levels and types as shown in figure 2.11.

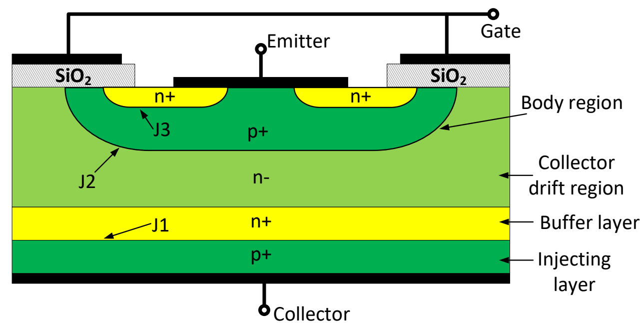


Figure 2.11: Structure of an IGBT [3]

The IGBT is structurally similar to that of the MOSFET shown earlier. It is designed as a

MOSFET with an injecting layer on its collector side. This layer provides conductivity modulation of the collector-drift region. This means that holes from the lower p^+-layer will get injected into the drift region, making the resistance in the drift region decrease. Consequently, the on-state resistance is decreased. The performance of an IGBT is therefore in between that of a MOSFET and a BJT. It is slower than a MOSFET because, as for the BJT, the injected charge carriers in the drift region must be removed before it is completely turned off. The IGBT is also a voltage controlled device.

2.2.4 Capacitor Model of a MOSFET

To describe the switching characteristics of MOSFETs, the capacitor model shown in figure 2.12 is commonly used.

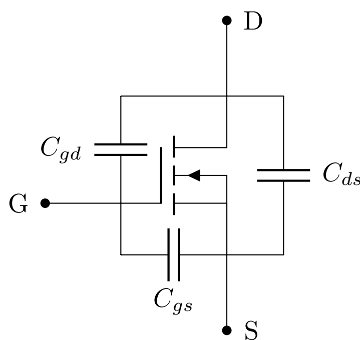


Figure 2.12: Capacitor model of MOSFET

C_{gd}, C_{gs} and C_{ds} represent the internal parasitic capacitances of the MOSFET. They are dependent on the structure and material of the semiconductor device [2]. For the gate-source and gate-drain capacitances, the insulator material is an important factor. In addition to this, the depletion layer contributes to these capacitances. Hence, they are dependent on the voltage. Furthermore, the drain-source capacitance is dependent on the size of the drift region. Due to these effects, manufacturers present measurements of these capacitances in datasheets like in equation 2.1 and figure 2.13.

$$\begin{aligned} C_{iss} &= C_{gs} + C_{gd} \\ C_{oss} &= C_{ds} + C_{gd} \\ C_{rss} &= C_{gd} \end{aligned} \tag{2.1}$$

It is observed in figure 2.13 how the capacitances vary with the drain-source voltage.

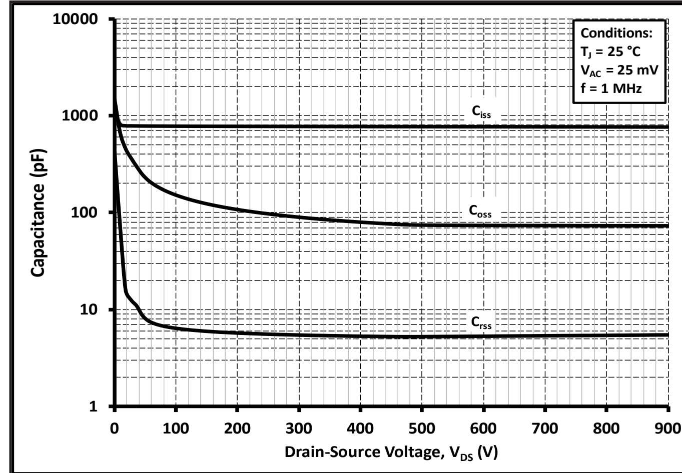


Figure 2.13: MOSFET capacitances as function of drain-source voltage. Taken from C3M0065090D datasheet [4]

Modeling the switching waveform is challenging when considering the parasitic capacitors in the MOSFET since C_{gd} is dependent on the drain-source voltage as shown in figure 2.13. Advanced models are used to investigate the switching event in simulation environments such as SPICE based simulation software.

2.2.5 Transconductance and IV-Characteristics

Devices such as MOSFETs, BJTs and IGBTs are three terminal devices where the input (gate signal) controls the output (drain-source or collector-emitter) current. As discussed in previous sections, the drain-source current of a MOSFET is dependent on the gate-source voltage. The relationship between the gate voltage magnitude and the resulting drain current is commonly presented in a transfer curve. In a transfer curve, the drain current is plotted as a function of gate-source voltage. As the gate-source voltage (input) increases, the drain current (output) increases. From the transfer curve, the threshold voltage can be found as the voltage where the current starts to increase from zero. The transconductance is the derivative of the transfer curve with respect to gate-source voltage and is defined in equation 2.2.

$$g_m = \frac{\partial I_D}{\partial V_{GS}} \tag{2.2}$$

From this definition, transconductance is the ratio of how much change is seen on the output by a change on the input. Figure 2.14 shows the transconductance of a SiC MOSFET using the simulation model provided by the manufacturer. The figure also shows how the transconductance changes with temperature.

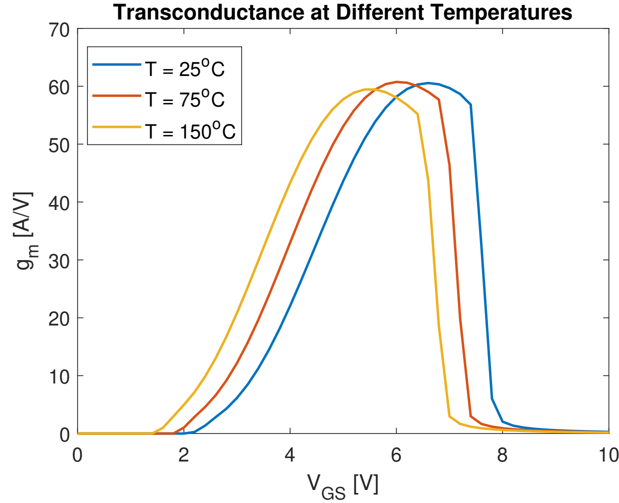


Figure 2.14: Transconductance at different temperatures from simulations using model for CPM3-0900-0010A provided by Wolfspeed

From the figure it is observed that as the temperature increases, the threshold voltage is reduced. Higher temperature will increase the thermal ionization in the material, creating more holes and free electrons. This means that the inversion layer will be established faster and the MOSFET starts to conduct earlier for higher temperatures compared to low temperatures. Consequently, this may have an impact on the design and driving of the MOSFET.

In figure 2.15, I_D is shown as a function of V_{DS} at different V_{GS} values for an n-channel MOSFET.

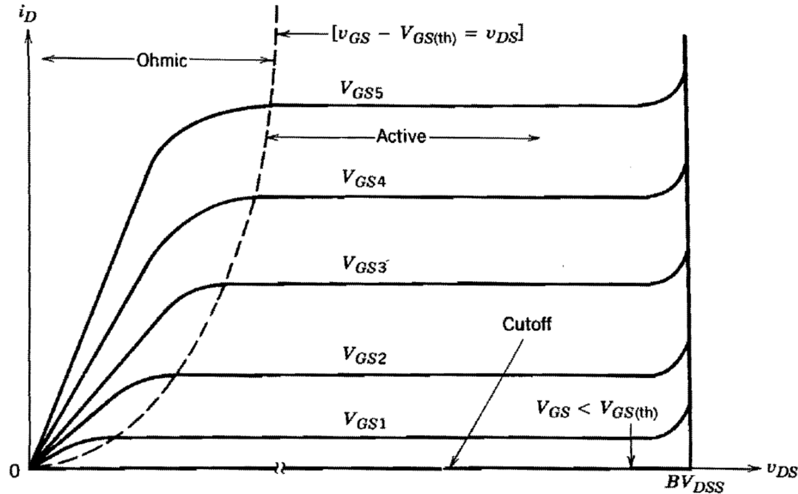


Figure 2.15: $I_D V_{DS}$-characteristics from [2], $V_{GS5} > V_{GS4}$ etc.

Conventionally, the characteristics of a MOSFET have been divided into three regions; the cutoff, active and ohmic region [2]. When the gate-source voltage is below the threshold voltage, the MOSFET is in the cutoff region. No current is flowing through the device and it has a drain-source voltage equal to the applied voltage. In the active region, the drain current is only dependent of the gate-source voltage. A higher gate-source voltage allows for larger current through the MOSFET. The device consumes a lot of power in the active region because the drain-source voltage is approximately equal to the applied voltage. When the gate-source voltage is increased to a large value, the MOSFET will eventually enter the ohmic region. The power consumption is much lower in this region compared to the active region. The goal when switching a MOSFET is to move from the cutoff region into the ohmic region and vice versa.

2.3 Switching Events of MOSFETs

In this subsection, the turn-on and turn-off switching events of a MOSFET will be investigated. A simple resistor in series with the gate voltage source is connected to the gate of the MOSFET to explain the switching events. Thus, creating an RC-circuit from the gate to source loop. This RC-circuit will determine the shape of the gate-source voltage. Figure 2.16 shows the circuit used to explain the switching event.

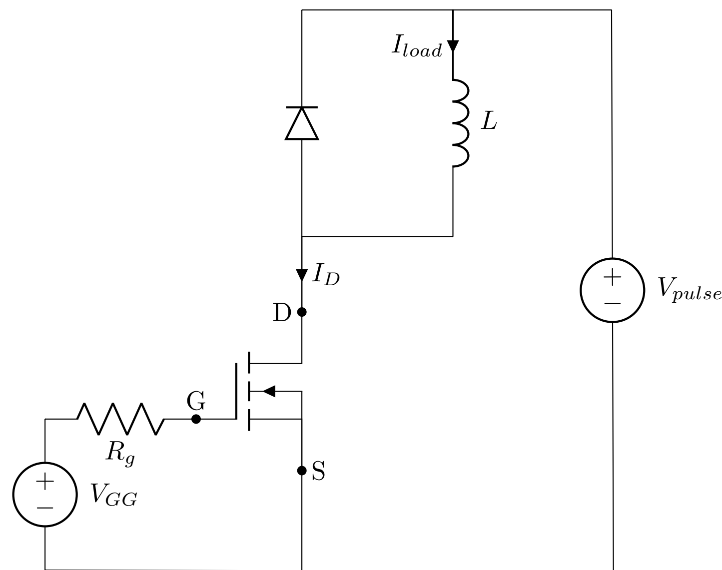


Figure 2.16: Inductive switching circuit

2.3.1 Turn-on Waveforms

In order to turn-on a MOSFET, a voltage must be applied to the gate. This voltage depends on the type of MOSFET. An n-channel MOSFET will be used in this example. N-channel MOSFETs require positive applied gate voltage to charge the gate capacitors, turning the MOSFET on. Figure 2.17 shows the turn-on waveform divided into 4 time intervals.

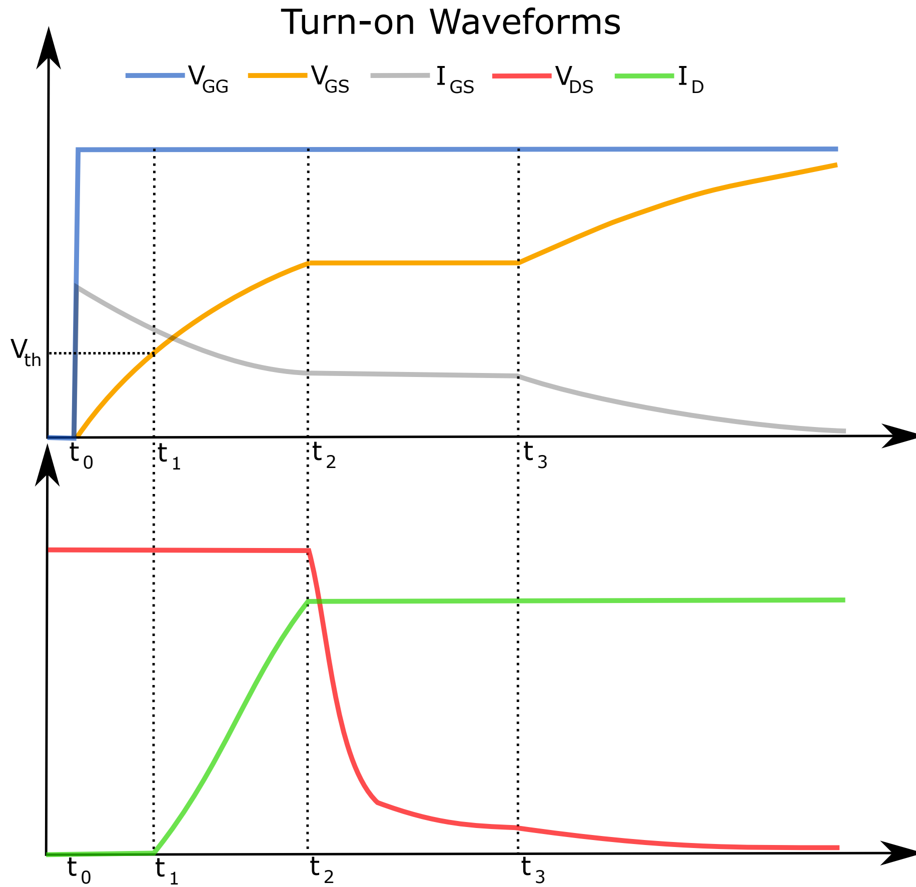


Figure 2.17: Turn-on event

The first interval, $\boldsymbol{t_0}$ to $\boldsymbol{t_1}$, is called the turn-on delay interval. Here, the gate-source voltage is charged from zero to the threshold voltage. During this interval, the depletion layer is changed by the applied gate voltage and the inversion layer is formed at $\boldsymbol{t_1}$. In this interval, the MOSFET is not conducting, i.e. it is in the cutoff region of figure 2.15. At $\boldsymbol{t_1}$, the threshold voltage of the MOSFET is reached. At this point, current starts flowing through the MOSFET. From $\boldsymbol{t_1}$ to $\boldsymbol{t_2}$, the current through the MOSFET increases until it reaches the load current. The rate of change of the current is given by the transconductance curve and equation 2.2. The MOSFET is now in the active region. When the drain current reaches the load current, it is seen from the IV-characteristics that the gate-source voltage can not increase until the drain-source voltage has decreased significantly. Hence, the drain-source voltage will change. $\boldsymbol{t_2}$ to $\boldsymbol{t_3}$ represents the time interval of the voltage commutation. The drain voltage drops to the on-state voltage drop given by the IV-characteristics curve (zero in the ideal case). This time interval is also called the miller plateau. During the miller plateau, the capacitor between gate and drain will change and become larger, making dV/dt smaller for the last part of the voltage commutation. In this interval, most of the current goes through the gate-drain capacitor, keeping the gate-source voltage close to constant, hence the name miller "plateau". Once the miller plateau ends, the MOSFET has been moved along the constant V_{GS}

lines in figure 2.15 into the ohmic region. Finally, the last stage of the waveform is from the end of the miller plateau to when the gate voltage has reached the reference voltage, V_{GG}, applied in the drive circuit. This last part reduces the on-state resistance of the MOSFET, decreasing the conduction losses.

2.3.2 Turn-off Waveforms

N-channel MOSFETs are turned off by reducing the gate voltage to zero or a negative value. Figure 2.18 shows the turn-off event. The turn-off event has the same intervals found in the turn-on event, though they happen in reverse order. I.e. the gate-source voltage goes from a high V_{GG} value to a low V_{GG} value.

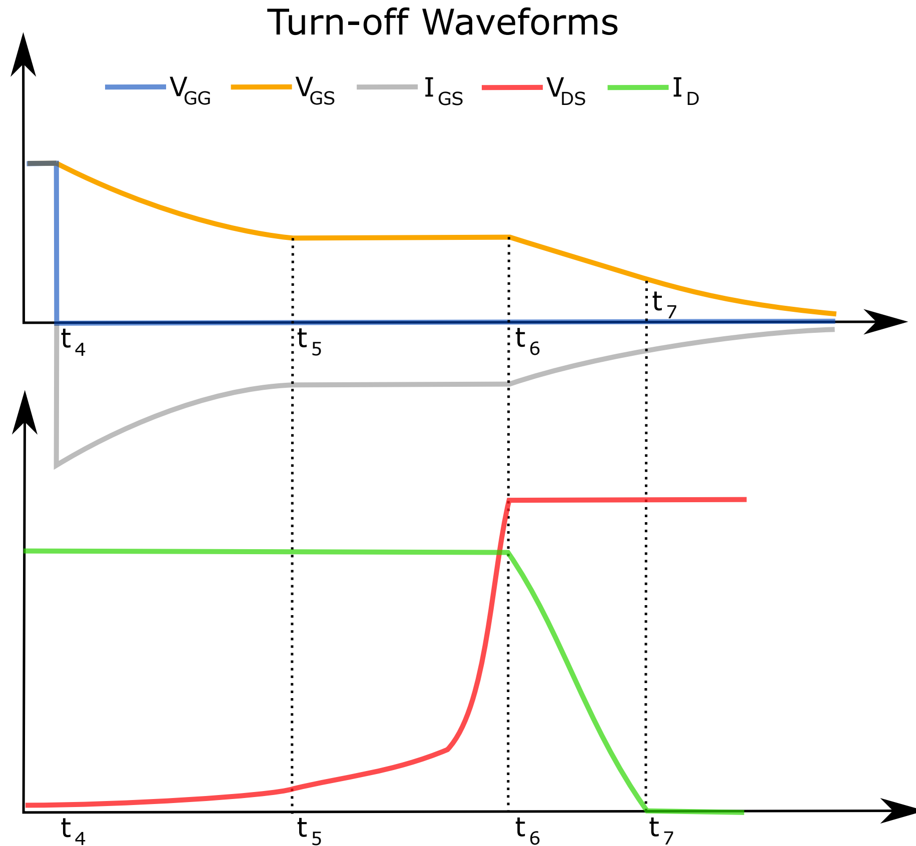


Figure 2.18: Turn-off event

During turn-off, the gate capacitances are discharged by letting V_{GG} become zero or a negative value. This will create a negative current flowing from the gate capacitors through the gate resistor. The first time interval, $\boldsymbol{t_4}$ to $\boldsymbol{t_5}$ is termed the turn-off delay time. In this interval, the charge used to lower the on-state resistance is released. Thus, moving the MOSFET from the ohmic region back into the active region, increasing the on-state resistance during this interval. Once $\boldsymbol{t_5}$ is reached, the

voltage commutation interval begins. However, the gate-source voltage cannot decrease until the drain-source voltage has increased. Hence, the MOSFET is moved further into the active region. During this interval, the gate-drain capacitance will change and the effect of this is seen on the voltage rise of V_{DS}. Once V_{DS} has reached the supply voltage, the drain-source current will start to decrease. $\boldsymbol{t_6}$ to $\boldsymbol{t_7}$ is the current commutation interval. The rate of change will be determined by the transconductance curve and equation 2.2. At $\boldsymbol{t_7}$, I_D has decreased to zero and the MOSFET has been moved from the active region into the cutoff region. Consequently, the MOSFET is completely turned off.

2.4 Si and SiC Comparison

Wide band gap (WBG) materials have been emerging as the silicon technology is reaching its limits. These materials have characteristics that allow for higher efficiency, higher switching frequency and higher temperature operation. Silicon carbide (SiC) is one of the promising WBG materials [5]. Figure 2.19 shows a comparison of Si, SiC and gallium nitride (GaN).

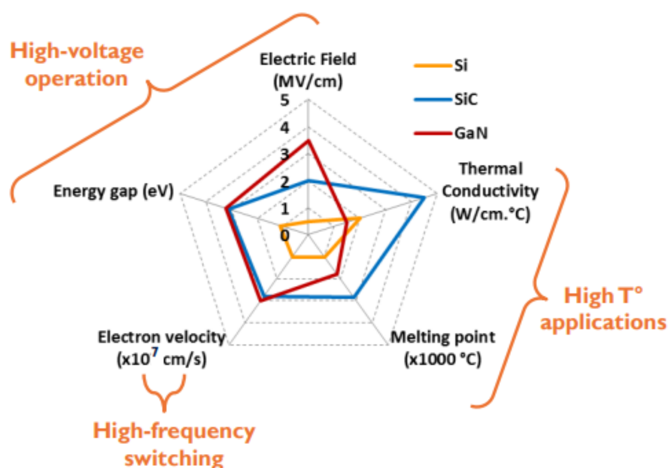


Figure 2.19: GaN, SiC and Si comparison for different material properties [6]

As SiC and GaN have higher energy gap and higher breakdown electric field compared to Si, they can deliver the same voltage ratings with smaller drift region or higher doping levels. The drift region determines the breakdown voltage and is the dominating resistance for the conduction losses of a MOSFET. Hence, decreasing this region will reduce the drift resistance and subsequently lower the on-state resistance. According to [7], SiC can, in theory, reduce the resistance per unit area of the drift region to 1/300 compared to Si at the same breakdown voltage. Lower resistance per unit area for SiC allows for lower losses at the same ratings compared to Si. Besides, it allows for the same resistance at higher blocking voltages. For high blocking voltages, Si MOSFETs have high conduction losses. To combat this, IGBTs have an injection layer that can inject carriers in to the drift layer, allowing for higher blocking voltages while having low resistance. Hence, IGBTs have lower conduction losses and are more competitive for high power applications. However, SiC MOSFETs allow for higher blocking voltages at lower resistance compared to Si MOSFETs. This might make SiC MOSFETs able to compete with Si IGBTs. The advantage of a MOSFET structure

is that the switching event is faster compared to an IGBT. IGBTs are bipolar devices and have longer switching times due to conductivity modulation. Carriers are injected into the drift layer, allowing lower on-state voltage. However, these carriers must be removed when turning off. As a consequence, there will be a so called tail-current at the end of an IGBT switching, lasting for some time after the collector-emitter voltage has increased. Tail-current makes the switching take longer and increases the switching losses. MOSFETs are unipolar devices and do not have a tail-current. Consequently, the switching losses in a MOSFET are lower compared to an IGBT. Subsequently, SiC MOSFETs can switch at higher frequency. Higher switching frequency reduces the size of the passive elements, which may allow the system to be denser. Additionaly, SiC MOSFETs have an intrinsic body diode. This body diode may replace the requirement for separate diodes parallel to the switching device. Thus, it may reduce the size of modules and converters. Moreover, SiC has higher electron velocity compared to Si, enabling faster carrier movement within the material. This makes SiC based semiconductor devices able to switch faster, further strengthening the competetiveness of SiC MOSFETs.

Another characteristic of SiC is that it has much higher thermal conductivity and melting point compared to Si. Higher thermal conductivity and melting point allow for higher temperature operation. I.e. SiC can dissipate more heat and operate at higher temperatures. This is attractive for high power applications. Using SiC in place of Si might allow for denser and cheaper systems. E.g. fewer SiC devices may be needed compared to Si devices to do the same task, due to SiC's thermal properties. Furthermore, this might open new high frequency applications, since SiC can both switch faster and tolerate higher operating temperatures.

Due to all these characteristics of WBG materials, SiC MOSFETs might be able to replace Si IGBTs in many applications [8][9].

3 Modeling of SiC MOSFETs

Models of components are important when simulating systems. Simulations can be used to predict the response of the system, subsequently allowing the designer to find potential flaws or shortcomings in the design before implementing a prototype. Therefore, it is of great importance that the models used in simulations accurately represent the real components that will be used in the system. However, the model chosen for the simulations will often have certain limitations. There will be a compromise between fast simulation speed and accuracy of the model. A detailed and accurate model can be vastly time consuming to simulate, while a simple model described by a few equations will give fast results, but at a price of reduced accuracy. This section will give an introduction to modeling of SiC MOSFETs and present several models proposed in the literature and by different manufacturers. To begin with, section 3.1 will discuss some of the important aspects of MOSFET modeling.

3.1 What Makes a Good MOSFET Model?

The behavior of a MOSFET depends on the structure, design and the operating condition. Providing an accurate IV-characteristic is essential for the accuracy and reliability of the simulation. In connection to this, the transconductance should be accurate and smooth continuous in the operating range to avoid convergence issues. Furthermore, the model should have a well defined and efficient parameter extraction methodology. This will allow users to easily utilize measurements or values from a device to obtain a model. Moreover, temperature is an important factor in the modeling design of a device. In most cases, temperature will change the behavior of the device by a noticeable degree. E.g. the thermal ionization in the semiconductor material depends heavily on temperature. As a consequence of this, the inversion layer might be more easily created at higher temperatures, consequently lowering the threshold voltage of the device. In many applications, there will be some sort of temperature difference and thermal considerations are in several levels of the design process. Thus, the model should include temperature dependency and be well behaved for the temperature range of interest. Finally, computational efficiency is an important aspect of simulation models. In theory, a model could be described by all its physical and mathematical equations and relationships. However, solving such a detailed description would require a lot of computational power and time. Usually, when simulating a circuit, it is simulated over a wide range of possible operating points in order to evaluate the performance. Hence, the model should be fast and accurate. A compromise will often have to be made; either increasing accuracy at the cost of more simulation time or vice versa. Therefore, the model to be used in simulations should be chosen based on the level of the design process. E.g. to assess a new circuit design, thermal considerations may be neglected. Thus, allowing for a simpler model, which will give results faster. The thermal considerations come later in the design process and a more advanced model may be utilized then.

3.2 SiC MOSFET Models

Models used in simulations may be chosen based on the level of the design process. Some models will be simple and fast, while others will be computationally expensive and highly accurate. SiC MOSFET modeling can be divided into five levels [10].

1. Behavioral Models:

Behavioral models are simple models that simulate the devices without considering the physical mechanisms of operation. Such models are usually implemented using mathematical fitting methods. As a result, behavioral models are fast to simulate at the cost of low accuracy.

2. Semi-physics-based Models:

   Semi-physics-based models are partially based on the device physics. For example, some parts of the model are based in physics equations, while other parts are defined by mathematical fitting methods.

3. Physics-based Models:

   Physics-based models use semiconductor physics. The electrical and thermal behavior will be obtained by solving equations based on semiconductor physics.

4. Semi-numerical Models:

   When a physics-based model contains equations or relations that is solved by numerical methods, such as Fourier or Laplace, the model is said to be semi-numerical.

5. Numerical Models:

   Numerical models are used when a complete and complex mathematical model of the system are to be used. The mathematical model is solved numerically due to its complexity, which makes it hard to find an analytical solution. These types of models require great detail of material properties and device geometry.

The list presents five different types of models. These are used to characterize the SiC MOSFET models presented in the coming sub-sections. Generally, the computational complexity of the model types is in the order of the list; behavioral models are the fastest, while numerical models are the slowest. Numerical and semi-numerical models include many equations that require detailed information of device structure and material properties. That information might not be easily obtainable and is not shown in datasheets. As a result, many proposed models are usually behavioral, semi-physics-based or physics-based. Some manufacturers even provide their own models. Although the first three model types will be less accurate compared to numerical and semi-numerical, they can still provide the necessary results needed to evaluate a circuit or design. Physics-based models utilize equations based on semiconductor physics. However, certain constants used in these equations may have its own relations, and is based on material properties and device structure. This can make it hard to use the same model on other devices that may have slightly different structure and material properties. Behavioral models are based on fitting equations to measurement data of the devices. Models of this type may have excellent match to the measurement data or datasheet curves. However, the performance of the model can vary. There are many elements in a MOSFET model, such as IV-characteristics, CV-characteristics, breakdown, body diode and reverse recovery of body diode to mention a few. In modeling, there will be a trade-off between having accurate results and complexity of the model.

The following sub-sections will explore different MOSFET models. Firstly, established models for low voltage (LV) complementary metal-oxide-semiconductors (CMOS) will be briefly mentioned, followed by models proposed in the literature. Various models with and without temperature considerations will be presented. Additionally, some manufacturer models will be discussed. The models presented will be of the first three types; behavioral, semi-physics-based and physics-based.

3.2.1 Established Models

For integrated circuits and LV CMOS technology, Berkeley short-channel IGFET model (BSIM) is the industry standard. It is a detailed and comprehensive model of a variety of effects related to MOSFET physics, including the sub-100nm regime. All the effects included in the model are described in its user manual [11], which is more than 170 pages long. BSIM requires a great number of parameters and detailed information about MOSFET structure and physical properties. This can introduce many challenges when extracting parameters from datasheets or measurements to model a new or pre-existing device where a simulation model is not provided by the manufacturer. BSIM can be classified as a numerical model. Other models are also well established, such as the Enz-Krummenacher-Vittoz (EKV) model. The EKV MOSFET model characterizes all modes of operation for a MOSFET and includes modeling of several physical effects [12, 13]. It is in principle formulated as a single expression, subsequently preserving continuity of derivatives in the model. Compared to BSIM, the number of parameters are reduced, making it easier to extract parameters and establish a model for the device. Moreover, other models based on curve fitting also exist. The Curtice-Ettenberg WBG FET model uses a certain expression for the drain current and uses data to fit the expression and obtain accurate static characteristics [14, 15].

3.2.2 Charge-Based Model

The charge-based model proposed in [16] is a compact model able to predict the behavior of the device on the basis of a single equation. It can be classified as a physics-based model. The proposed model contains the intrinsic capacitances C_{GD}, C_{GS} and C_{DS} and an R_{on} on drain. Parasitic elements and breakdown modeling are not included. The drain current is obtained through an analytical expression that represents the inversion charge density at the source and drain electrodes. Equation 3.1 shows the expression of the drain current as the difference between the forward current I_F and the reverse current I_R.

$$I_{DS} = I_F - I_R = \beta \int_{V_S}^{\infty} \frac{-Q_i}{C_{ox}} \mathrm{d}V - \beta \int_{V_D}^{\infty} \frac{-Q_i}{C_{ox}} \mathrm{d}V \tag{3.1}$$

where β is the transmission coefficient, C_{ox} the oxide capacitance per unit area and Q_i is the inversion charge. Furthermore, the model represents the parasitic capacitances as defined in [17]. Equation 3.2 shows the definition of the capacitances.

$$\begin{cases} C_{GS} = C_m + C_{oxs} + \frac{1}{\frac{1}{C_{oxc}} + \frac{1}{C_c}} \\ C_{GD} = \frac{1}{\frac{1}{C_{oxd}} + \frac{1}{C_{gdj}}} \\ C_{DS} = C_{dsj} \end{cases} \tag{3.2}$$

C_m, C_{oxs}, C_{oxc} and C_{oxd} are related to the gate oxide layer and do not change with the applied voltage. However, C_c, C_{gdj} and C_{dsj} originates from the depletion layer and depends on the gate-source voltage, gate-drain voltage and drain-source voltage respectively. By defining the drain-current and the parasitic capacitances, the model may be able to accurately represent both static and dynamic behavior of the device. Although the model is very compact with few parameters, the equation describing the drain current can be difficult to compute directly and might require simplifications of the transition between the forward current and reverse current. Additionally, the model suffers from lack of thermal considerations. Hence, the model cannot be used in thermal analysis of the circuit design.

3.2.3 Non-segmented Model

One way to solve convergence issues is to use non-segmented, smooth continuous equations to describe the static and dynamic characteristics of SiC MOSFETs [18]. In this cases it leads to a behavioral model. The equations used has no physical meaning. Figure 3.1 shows the model of a SiC MOSFET proposed in [18].

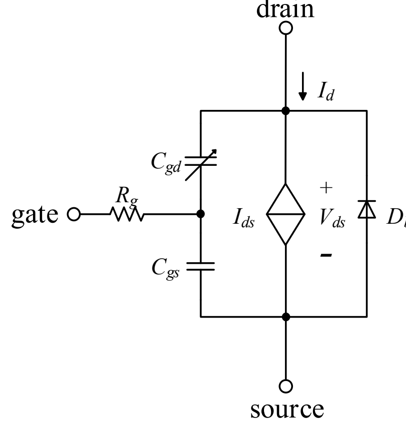


Figure 3.1: Non-segmented model proposed in [18]

The model includes a voltage-dependent current source I_{ds}, a gate-drain voltage-dependent capacitance C_{gd}, a gate-source constant capacitance C_{gs}, an internal gate resistance R_g and a body diode D_b. Parasitic elements and breakdown modelling are not included. The voltage-dependent current source describes static IV-characteristic of the SiC MOSFET. It is obtained as in equation 3.3,

$$I_{ds} = k \cdot \{1 + \tanh\left[a \cdot (V_{gs} + m) + b \cdot (V_{gs} + n)^2\right]\} \cdot \frac{p(V_{gs}) \cdot V_{ds}}{1 + q(V_{gs}) \cdot V_{ds}} \tag{3.3}$$

where k, a, b, m and n are parameters related to the transfer characteristic, while p and q are parameters related to the output characteristic. Furthermore, the gate-drain capacitance C_{gd} has strong non-linearity, and the equivalent current source shown in figure 3.2 is used to describe the voltage-dependent capacitance C_{gd}. Moreover, the non-linear characteristic of C_{gd} is realized by the constant capacitance C_0. Equation 3.4 and 3.5 represent the model for the nonlinear capacitance C_{gd}.

$$E_{gd} = f(V_{gd}) \tag{3.4}$$

$$i_0 = C_0 \frac{\mathrm{d}E_{gd}}{\mathrm{d}t} = C_0 \frac{\mathrm{d}E_{gd}}{\mathrm{d}V_{gd}} \cdot \frac{\mathrm{d}V_{gd}}{\mathrm{d}t} \tag{3.5}$$

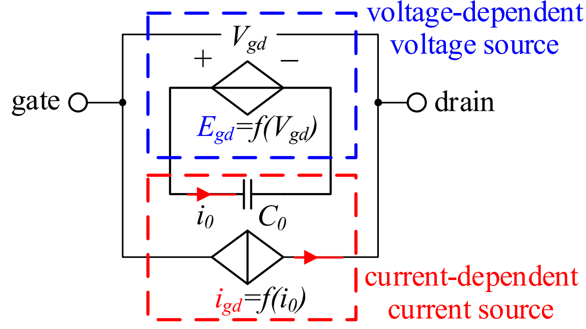


Figure 3.2: Voltage-dependent capacitance model used in [18]

The current-dependent current source in the gate-drain capacitance model is given as in equation 3.6.

$$i_{gd} = f(i_0) = \frac{1}{C_0} i_0 = \frac{\mathrm{d}E_{gd}}{\mathrm{d}V_{gd}} \cdot \frac{\mathrm{d}V_{gd}}{\mathrm{d}t} \tag{3.6}$$

In a physical model, i_{gd} will be as in equation 3.7.

$$i_{gd} = C_{gd} \frac{\mathrm{d}V_{gd}}{\mathrm{d}t} \tag{3.7}$$

Combining equation 3.6 and 3.7 results in equation 3.8.

$$C_{gd} = \frac{\mathrm{d}E_{gd}}{\mathrm{d}V_{gd}} \tag{3.8}$$

An expression for C_{gd} is found through data extraction from the CV-characteristics in the datasheet of the chosen SiC MOSFET, followed by curve fitting. The results are shown in equations 3.9 and 3.10.

$$C_{gd} = s \cdot \frac{1}{1 + \exp\left\{\frac{h - V_{gd}}{f}\right\}} + r \tag{3.9}$$

$$E_{gd} = s \cdot f \cdot \ln\left[1 + \exp\left\{\frac{V_{gd} - h}{f}\right\}\right] + r \cdot V_{gd} \tag{3.10}$$

where s, h, f and r are parameters related to the C-V characteristics.

The internal resistance R_g and the gate-source capacitance C_{gs} are constant and are found in the datasheet. However, the body diode D_b uses an inherent diode model in PSpice, and the modeling of the drain-source capacitance C_{ds} is considered in the body diode.

The results show good agreement between the datasheet and the simulated curves. However, there is a noticable difference when it comes to the body diode curve.

This simple model will give fast simulation results with less convergence problems due to the continuous smooth equations describing the SiC MOSFET.

A similar approach has been used in [19], where a genetic algorithm iterates to fit the model parameters to the original data. The model is divided into a static and dynamic model. The static model represents the forward and reverse conduction characteristics of the SiC power MOSFET, whereas the dynamic part aims at reproducing the switching waveforms of the device. Furthermore, the parasitic inductances and resistances of the three pins of the device are included in the dynamic model. Additionally, the dynamic model includes a model for the internal body diode of the SiC MOSFET.

Nonetheless, the temperature will change the behavior of the SiC MOSFET and these effects are completely ignored in these models. Thus, they can only provide accurate results for simulations at the temperature used when extracting the data from the datasheet or from measurements. As temperature is an important aspect in the design of power electronic converters, more advanced models should be looked at.

3.2.4 Electro-Thermal Model

As temperature is an important aspect of the design of power electronic converters, a model taking that into account is necessary. By introducing temperature dependency, the simulations can more accurately represent the usage of a real device in a circuit design.

A fast and accurate SiC MOSFET model is proposed in [20]. The model is based on 25 model equations. Many of these equations are temperature dependent, meaning that as the temperature changes, the characteristics of the SiC MOSFET will also change. In this model, the intrinsic capacitances C_{GD}, C_{GS} and C_{DS}, an R_{DD} on drain, breakdown characteristics and an empirical description of the third quadrant operation are included. Thus, the model can be classified as a semi-physics-based model, since it has a mix of physics-based and behavioral equations. Parasitic elements are not included. The channel current is generated by a low current region and high current region. Equations 3.11 - 3.16 show the equations used for the MOSFET channel current.

$$I_{D\boldsymbol{x}} = \begin{cases} 10K_{P\boldsymbol{x}}[(V_{GS} - V_{T\boldsymbol{x}})V_{DSM} - \mathbf{P_{VFx}}^{y\boldsymbol{x}-1} V_{DSM}^{y\boldsymbol{x}} (V_{GS} - V_{T\boldsymbol{x}})^{2-y\boldsymbol{x}}] & , V_{DSM} \leq V_{DSS\boldsymbol{x}} \\ \frac{1}{2} K_{P\boldsymbol{x}} (V_{GS} - V_{T\boldsymbol{x}})^2 & , V_{DSM} \geq V_{DSS\boldsymbol{x}} \end{cases} \tag{3.11}$$

$$V_{T\boldsymbol{x}} = \mathbf{V_{Tx0}} - \mathbf{V_{Tx1}}(T_M - 27) \tag{3.12}$$

$$V_{DSS\boldsymbol{x}} = \frac{V_{GS} - V_{T\boldsymbol{x}}}{\mathbf{P_{VFx}}} \tag{3.13}$$

$$K_{P\boldsymbol{x}} = \mathbf{K_{Px0}} - \mathbf{K_{Px1}}(T_M - 27) \tag{3.14}$$

$$y\boldsymbol{x} = \frac{10}{10 - \mathbf{P_{VFx}}} \tag{3.15}$$

$$R_{DD} = \mathbf{R_{D0}} - \mathbf{R_{D1}}(T_M - 27) \tag{3.16}$$

where **x** should be replaced by L and H, representing low current region and high current region respectively. Furthermore, the bold variables are product-dependent parameters. Additionally, the intrinsic capacitances are defined as in equation 3.17.

$$C_{GD} = \begin{cases} \mathbf{C_{GD0}} & for\ V_{DS} \leq 0 \\ \mathbf{C_{GD0}} + \Delta C_{GD} & for\ 0 < V_{DS} \leq 380 \\ \mathbf{C_{GDF}} & for\ V_{DS} > 380 \end{cases} \tag{3.17}$$

where ΔC_{GD} is defined as in equation 3.17.

$$\Delta C_{GD} = 25.9 \cdot 10^{-3}(\mathbf{C_{GD0}} - \mathbf{C_{GDF}}) \left(0.064 V_{DS} - \frac{V_{DS}^{0.15}}{0.15}\right) \tag{3.18}$$

Data extraction from the manufacturers datasheet is used to fit the parameters of the model. The model is written in Verilog-A, a language for describing analog circuits. Moreover, an inverter design is tested using the proposed SiC MOSFET model in the simulation program Questa-ADMS, and then compared to a simulation of the inverter design in OrCAD PSpice using the model of the manufacturer of the SiC MOSFET. There is no visible difference between the switching characteristics of the inverter using different engines and SiC MOSFET models. However, a significant reduction in simulation time were observed for the Questa-ADMS simulation using the proposed model. According to the authors, this is, in part, because of better modeling and a more efficient simulation engine.

A SiC MOSFET model for short-circuit analysis is proposed in [21]. It is based on non-segmented equation modeling, making it a behavioral model. Equation 3.3 describes the drain-current of the device, similar to that of the non-segmented model presented in 3.2.3 with a few adjustments. Temperature dependency is introduced by an additional function multiplied with the drain current. The short-circuit current is defined as in 3.19;

$$I_{SC} = I_{DS} \cdot f(T_j) \tag{3.19}$$

where the function $f(T_j)$ is defined by equation 3.20.

$$f(T_j) = 1 + (T_j - 25) \cdot \frac{\left(p_1 \cdot T_j^{p_2}\right)}{ln(p_3 \cdot T_j) + p_4 \cdot T_j^{p_5}} \tag{3.20}$$

where p_1-p_5 are related parameters from data extraction of short-circuit characteristics. The definition of the gate-drain capacitance is also similar to that of the model proposed in [18]. However, another term is added to the equations, shown in equation 3.21 and 3.22.

$$C_{GD} = s_{11} \cdot \frac{1}{1 + \mathrm{e}^{\frac{s_{21} - V_{GD}}{s_{31}}}} + s_{41} \cdot \frac{1}{1 + \mathrm{e}^{\frac{s_{51} - V_{GD}}{s_{61}}}} + s_{71} \tag{3.21}$$

$$E_{GD} = s_{11} \cdot s_{31} \cdot \cdot \ln\left[1 + \mathrm{e}^{\frac{V_{GD} - s_{21}}{s_{31}}}\right] + s_{41} \cdot s_{61} \cdot \cdot \ln\left[1 + \mathrm{e}^{\frac{V_{GD} - s_{51}}{s_{61}}}\right] + s_{71} \cdot V_{GD} \tag{3.22}$$

where s_{11}-s_{71} are parameters related to the CV-characteristics. This model is able to both represent the static characteristics and CV-characteristics, and the short-circuit characteristics. However, the temperature-dependency is only used during short-circuit condition. This means that temperature changes occurring due to normal operation, ambient temperature changes and load changes are not accounted for. Hence, the model may be accurate and perform well when operating at 25^oC, while performing worse for other temperatures.

3.2.5 Curtice-Ettenberg Based Model

In [22], an approach to make an universal behavioral model is proposed. It is based on the Curtice-Ettenberg WBG FET model [14, 15] with some adjustments. The drain current is defined by equation 3.23.

$$I_{DS} = f_g(V_{GS}) \cdot f_d(V_{GS}, V_{DS}) \tag{3.23}$$

The first component, f_g, is related to the transfer characteristics, while the second component, f_d, is related to the shape of output characteristics.

$$f_g(V_GS) = \frac{H_t V_{GS}^{2B_t}}{\left[V_{GS}^{2C} + \mathrm{e}^{2C}\right]^{\frac{B_t}{C}}} \tag{3.24}$$

where C is constant, while B_t and H_t are linear temperature-dependent functions described by equation 3.25 and 3.26,

$$B_t = B_0 + \alpha_B \cdot (T_j - T_{nom}) \tag{3.25}$$

$$H_t = H_0 + \alpha_H \cdot (T_j - T_{nom}) \tag{3.26}$$

where B_0 and H_0 represent parameter values at nominal temperature T_{nom}, α_B and α_H are temperature coefficients, and T_j is the junction temperature. f_d is defined in equation 3.27.

$$f_d = \tanh\left[A_d V_{DS}^{k_d}\right] \cdot (1 + \lambda \cdot (V_{DS} - V_{DS,tr})) \tag{3.27}$$

where λ is the channel length modulation factor, A_d a scaling function, k_d constant coefficient and $V_{DS,tr}$ a drain-source voltage parameter chosen before the parameter extraction. The scaling function is defined as in equation 3.28.

$$A_d = A_g(T_j) \cdot V_{GS}^{k_g(T_j)} \tag{3.28}$$

where A_g and k_g are quadratic temperature dependent polynomials as given in equation 3.29 and 3.30 respectively.

$$A_g(T_j) = A_{g0} + \alpha_{A1} \cdot (T_j - T_{nom}) + \alpha_{A2} \cdot (T_j - T_{nom})^2 \tag{3.29}$$

$$k_g(T_j) = k_{g0} + \alpha_{k1} \cdot (T_j - T_{nom}) + \alpha_{k2} \cdot (T_j - T_{nom})^2 \tag{3.30}$$

where A_0 and k_0 are parameters at T_{nom}, $\alpha_{A1}, \alpha_{A2}, \alpha_{k1}$ and α_{k2} are temperature coefficients. Measurement data is used for parameter extraction of the proposed model. The fitting of the curves are done to find all the parameters as follows. Firstly, fitting of f_g is performed at nominal temperature to find initial estimations of C, B_0 and H_0. This is repeated for each temperature, yielding several values of B_t and H_t from which α_B and α_H can be estimated. A similar procedure is carried out for the fitting of f_d. The model has been fitted to four different devices with roughly the same rating and the results show good match for all four devices over the temperature range. Some discrepancies are noted for high temperature. However, the results are promising and this model is a candidate for an universal SiC MOSFET model. Although the model is able to recreate the static characteristics over a wide temperature range for different MOSFETs, the model is lacking in areas such as reverse recovery, dynamic switching behavior and breakdown. These are left to be considered in an improved model.

3.3 Manufacturer Models

Some manufacturers provide models of their products to use in simulation programs. These models can vary in terms of implementation and accuracy. In this section, a selection of different manufacturer provided models will be briefly looked at. Both IV- and CV-characteristics will be simulated and compared to the datasheet for a Cree model, an STmicroelectronics model and a ROHM model. The simulation circuits can be found in appendix A and B.

3.3.1 Cree Wolfspeed: C3M0075120K SPICE Model

The model from Cree can be classified as a semi-physics-based model. It includes drain, gate and source parasitic elements, intrinsic capacitances C_{GD} and C_{GS}, and an intrinsic body diode that includes C_{DS}. Breakdown is not covered in the model. The static characteristics are realised by a voltage-controlled current source (VCCS) from drain to source. Cree has based their equations for the static characteristics on the EKV model [12, 13], with a few simplifications. Equation 3.31 represents the drain-source current.

$$I_D = 2g_m\Phi_t^2 K_s \left[\ln\left(1 + e^{\frac{V_{GS}-V_{th}}{2K_s\Phi_t}}\right)^k - \ln\left(1 + e^{\frac{V_{GS}-V_{th}-nV_{DS}^a}{2K_s\Phi_t}}\right)^k \right](1+\lambda V_{DS}) \tag{3.31}$$

where g_m is the transconductance parameter, Φ_t the thermal voltage, K_s the sub-threshold slope parameter, k is the law exponent, λ the channel length modulation parameter, and n and a are triode region parameters [23]. Figure 3.3 shows the simulated output curves compared to the datasheet.

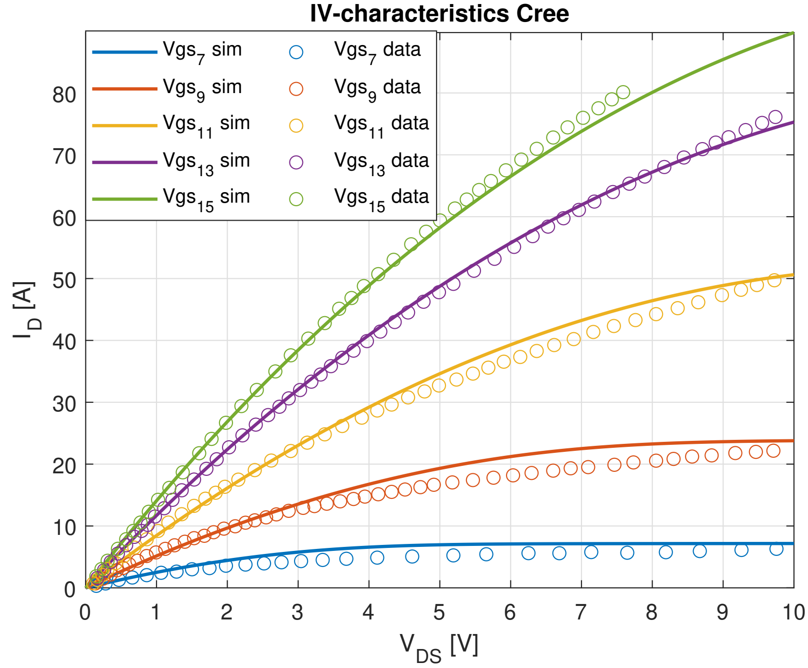


Figure 3.3: Simulation of C3M0075120K spice model, IV characteristics compared with datasheet

The circles represent points from the datasheet, obtained using a free online tool called "Web-PlotDigitizer" [24]. The simulated lines show acceptable match with the datasheet points. Thus, the model is able to accurately reproduce the static characteristics.

Another important aspect of the modeling is the dynamic characteristics. The gate-drain capacitance is represented by a VCCS in series with a small resistance. Equation 3.32 represents the characteristic of the capacitance value as the gate-drain voltage changes.

$$k_1 \left[1 + (\text{limit}(-V_{gd}, 0, 600)) \cdot \left(1 + k_a \cdot \frac{(1 + \tanh\left[k_b \cdot (-V_{gd}) - k_c\right)]}{2} \right) \right]^{-k_2} \cdot \frac{\mathrm{d}V_{gd}}{\mathrm{d}t} \tag{3.32}$$

where k_1, k_2, k_a, k_b and k_c seems to be fitting parameters. The gate-source capacitance is constant. The implementation of the capacitors gives CV-characteristics as shown in figure 3.4.

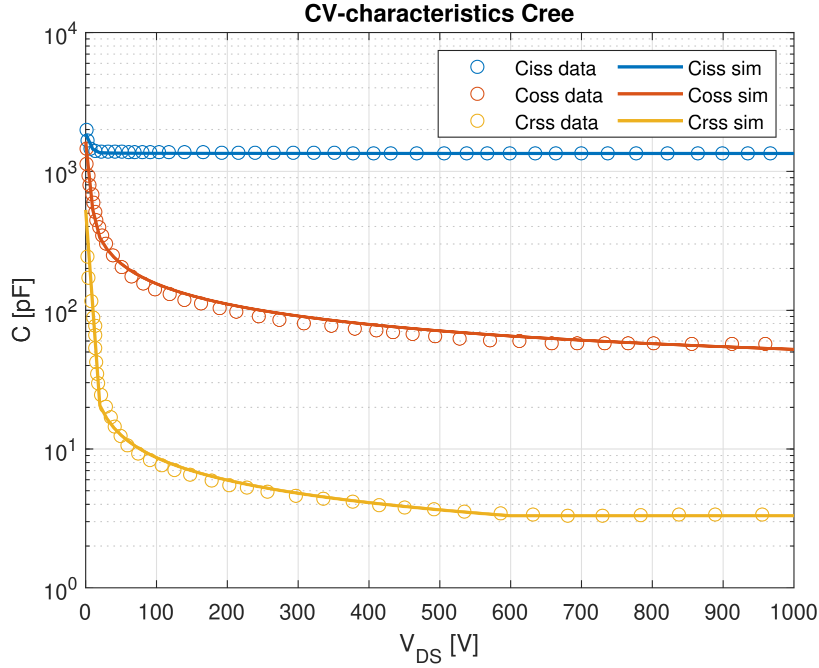
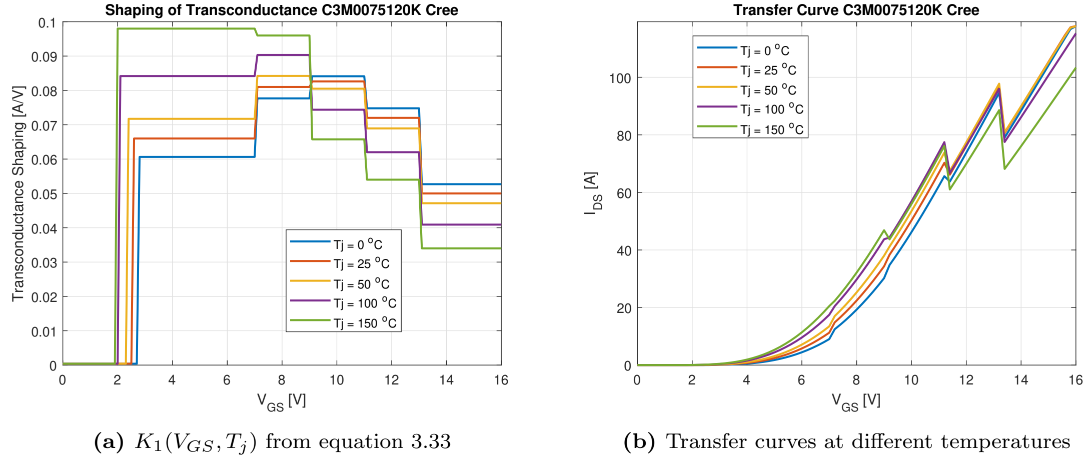


Figure 3.4: Simulation of C3M0075120K spice model, CV characteristics compared with datasheet

The model is able to accurately reproduce the CV-characteristics. However, the model has convergence issues. In the model file from Cree, the first term of the drain current is implemented like shown in equation 3.33.

$$2g_m \Phi_t^2 K_s = K_1(V_{GS}, T_j) \cdot (V_{GS} - V_{th}) + K_2(T_j) \tag{3.33}$$

where K_1 is a conditional expression that uses different functions based on the gate-source voltage and junction temperature and K_2 is a function of the junction temperature. Since the transconductance, g_m, depends on the gate-source voltage, it is likely that K_1 is used to shape the transconductance. The transfer curves and shaping functionality for C3M0075120K can be seen in figure 3.5.



(a) $K_1(V_{GS}, T_j)$ from equation 3.33

(b) Transfer curves at different temperatures

Figure 3.5: Discontinuities in the model

The transfer curves in figure 3.5b are discontinuous. This poses a problem during simulation of the device, as it will have convergence related issues. $K_1(V_{GS}, T_j)$ is constant for certain gate-source voltage intervals. For example, when gate-source voltage is between 7 and 9V, the function has a constant value based on the junction temperature. When the gate-source voltage increases to a value larger than 9V, a step occurs. These steps happen at predefined values for V_{GS} and makes the transfer curves discontinuous. Nevertheless, this was explored in [23], and a solution was proposed to make the transconductance shaping continuous, subsequently reducing convergence related issues. $K_1(V_{GS}, T_j)$ was changed from having steps to being smooth continuous. Equation 3.34 shows the implementation.

$$K_1(V_{GS}, T_j) = (p_{100} \cdot T_j^2 + p_{101} \cdot T_j + p_{102}) \cdot \left(\frac{V_{GS}}{(p_{200} \cdot T_j + p_{201})^2} \right) \cdot \mathrm{e}^{-\left(\frac{V_{GS}^2}{2 \cdot (p_{200} \cdot T_j + p_{201})^2} \right)} \tag{3.34}$$

where p_{100}, p_{101}, p_{102}, p_{200} and p_{201} are fitting parameters. Although the static characteristics from the simulations were smooth continuous, the model still has discontinuities without the improvements proposed by [23].

In summary, the Cree C3M0075120K SPICE model is classified as a semi-physics-based model. Parts of the model use equations that are not based in physics, e.g. the implementation of the transconductance shaping or the gate drain capacitance. However, most of the drain-source current are physics based, on a modified EKV model. Despite accurate results when comparing to the datasheet, the simulations using this model have high chances of convergence related issues. The main drawback is the implementation of the transconductance shaping. To reduce this issue, the improved version proposed by [23] should be used. This is also the version used in later sections of this thesis.

3.3.2 STMicroelectronics: SCT30N120_V3 SPICE Model

STMicroelectronics provides several versions of their components [25], ranging from basic to a more or less complete model where self-heating and package thermal modeling are included. SCT10N120_V3

includes temperature dependence, capacitance profiles, parasitic elements, breakdown and body diode modeling. The drain current is realized by a single VCCS. It is based on a LEVEL 1 MOS model with additional temperature-dependent coefficients un, V_{lin} and $kpsat$. Some parameters are based on look-up tables in LTspice and certain equations are not physics-based. Thus, the model is classified as semi-physics-based. Equation 3.35 and 3.36 show the expression for the drain current.

$$I_D = \begin{cases} 0 & \text{, if } V_{GS} < V_{th} \\ un \cdot A \cdot V_{DS} \cdot x_\lambda \cdot \left(2(V_{GS} - V_{th}) - \frac{V_{lin}}{kpsat} V_{DS}\right) \frac{V_{lin}}{2} & , V_{DS} < (V_{GS} - V_{th}) \frac{kpsat}{V_{lin}} \\ un \cdot A \cdot x_\lambda \cdot (V_{GS} - V_{th})^2 \frac{kpsat}{2} & , V_{DS} > (V_{GS} - V_{th}) \frac{kpsat}{V_{lin}} \end{cases} \tag{3.35}$$

$$x_\lambda = (1 + \lambda V_{DS}) \tag{3.36}$$

where V_{GS} and V_{DS} are gate-source and drain-source voltages, V_{th} the threshold voltage, A the chip area, λ the channel length modulation parameter, un, $kpsat$ and V_{lin} are dependent on junction temperature (T_j). Equations 3.37 shows the modeled temperature dependency of un and $kpsat$.

$$un(T_j) = b \cdot \left(\frac{T_j + 273}{300}\right)^{unt} \tag{3.37}$$

$$kpsat(T_j) = kpsat0 \cdot \left(\frac{T_j + 273}{300}\right)^{ksat} \tag{3.38}$$

where b, unt, $kpsat0$ and $ksat$ are fixed value parameters hardcoded in the library file. V_{lin} is defined by a look-up table in the code. Based on the junction temperature, V_{lin} will be linearly interpolated between two points in the table. It looks like in figure 3.6.

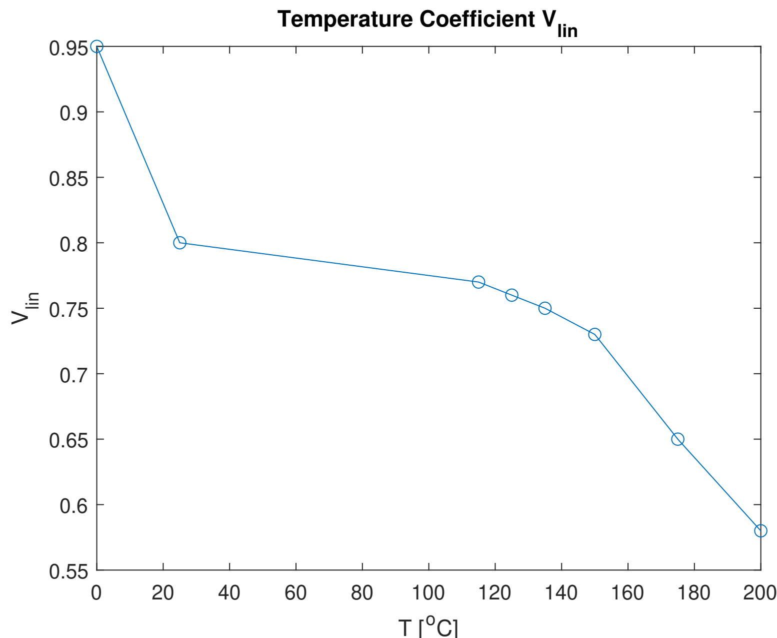


Figure 3.6: V_{lin} look-up table from LTspice illustrated

The model includes self-heating which will increase the junction temperature during conduction, i.e. when the drain current is different from zero. The V_{lin} parameter will change based on the junction temperature. This may introduce convergence related issues as V_{lin} is not smooth continuous. The changes it introduces on the drain current can be discontinuous when the temperature increases above certain values and moves V_{lin} to a new linearly interpolated line between two table values. Hence, it can be difficult for the software to find a solution if the operating point has a junction temperature close to the table values. Figure 3.7 shows the simulated IV-characteristics of the STMicroelectronics model.

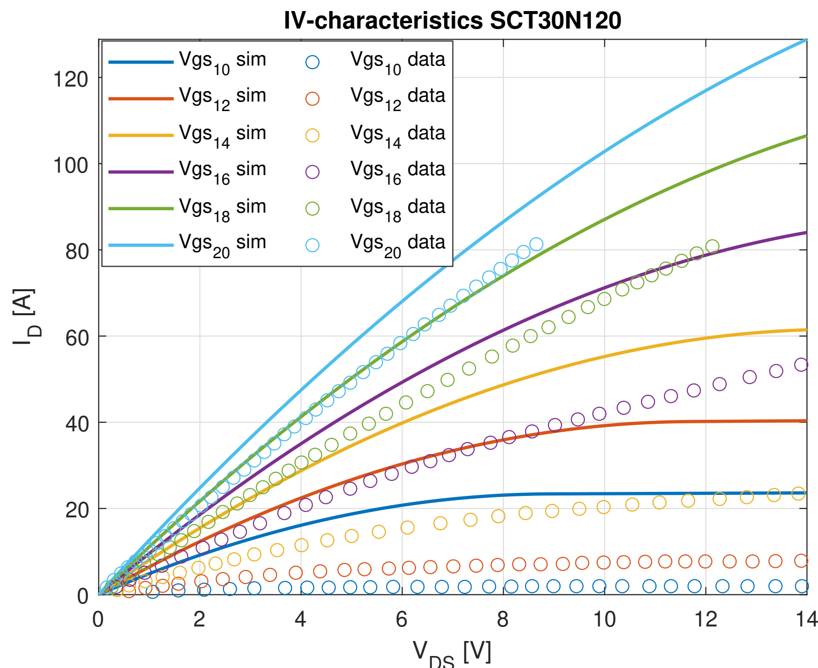


Figure 3.7: Simulation of SCT30N120_V3 spice model, IV characteristics compared with datasheet

The simulations do not match the extracted datasheet values well. The trend is similar to that of the datasheet, however, the accuracy is very low.

For the CV-characteristics, the STMicroelectronics model uses a single VCCS to represent the gate-drain capacitance. Equation 3.39 shows the drain-gate VCCS.

$$I_{C_{GD}} = 1 \cdot V_{alfa} \cdot I_{Vmiller} \tag{3.39}$$

$I_{Vmiller}$ is the current flowing through an RC-circuit with fixed values. The applied voltage to this small RC-circuit is the drain-gate voltage. Furthermore, V_{alfa} is found by equation 3.40

$$V_{alfa} = a_1 \cdot \left(10^{\left(a_2 - a_3 \cdot \tanh \left[a_4 \cdot \log \left(|V_{DS}| - a_5 \right) \right] \right)} + a_6 \right) \tag{3.40}$$

where a_1-a_6 are fitting parameters. This structure is also used to represent the drain-source capacitance.

Figure 3.8 shows the CV-characteristics of the ST model.

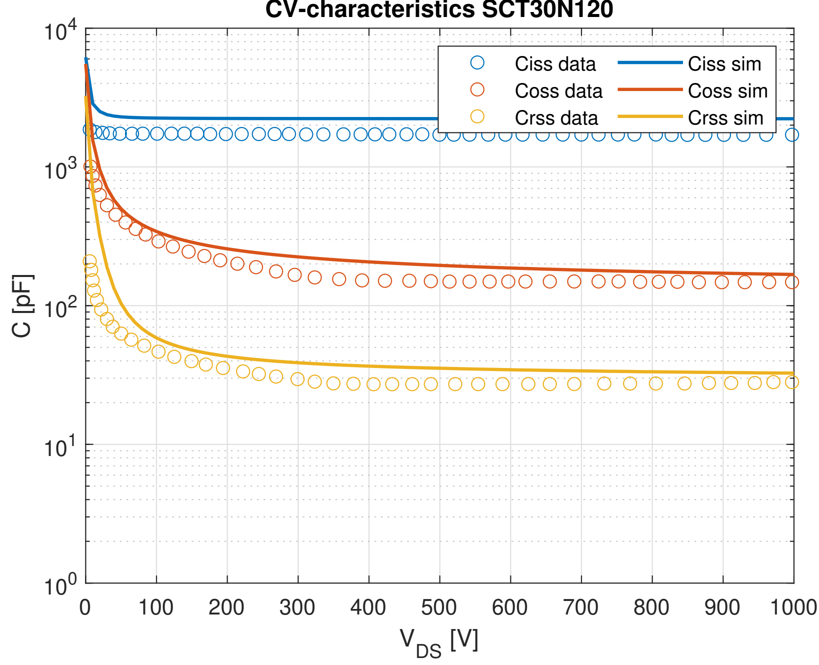


Figure 3.8: Simulation of SCT30N120_V3 spice model, CV characteristics compared with datasheet

The simulations definitely have the same shape as the datasheet values. However, the accuracy is not great, as the simulated values have overall higher values compared to the datasheet.

In summary, the STMicroelectronics model can be classified as semi-physics-based. It introduces temperature dependency and self-heating, but the results show low accuracy when compared to the datasheet. The modified LEVEL 1 MOS model used might be too simple to be able to represent the characteristics accurately. Nevertheless, the implementation of the capacitances shows better fit to the datasheet.

3.3.3 ROHM: SCT2080KE SPICE Model

The model from ROHM includes a thermal model. Parasitic elements are only included at source. In the model files, mathematical equations not based in physics are used to describe the characteristics, making it a behavioral model. Four equations make up the static characteristics, shown in equations 3.41 to 3.46.

$$R1(I,T) = r_0 \cdot I \cdot \mathrm{e}^{\frac{T-T_0}{a_1} \cdot \mathrm{e}^{\frac{T-T_0}{a_2}}} + a_3 \cdot I \cdot |I|^{a_4} \cdot \mathrm{e}^{\frac{T-T_0}{a_5}} \tag{3.41}$$

$$V1(I,T) = b_0 \cdot \arcsin \frac{I}{b_1} \cdot \mathrm{e}^{\frac{T-T_0}{b_2}} + b_3 \cdot I \cdot \mathrm{e}^{\frac{T-T_0}{b_4}} \tag{3.42}$$

$$V2(V,T) = c_0 \cdot V^{c_1 \cdot \mathrm{e}^{\frac{T-T_0}{c_2}}} \cdot \mathrm{e}^{\frac{T-T_0}{c_3} \cdot \mathrm{e}^{\frac{T-T_0}{c_4}}} \tag{3.43}$$

$$I1(V,W,T) = f_1(V,T) \cdot f_2(V,W,T) \tag{3.44}$$

$$f_1(V,T) = V \cdot \left(1 + \left(d_1 + d_2 \cdot \tanh\left[\frac{V}{d_3}\right]\right) \cdot \mathrm{e}^{\frac{T-T_0}{d_4}}\right) \tag{3.45}$$

$$f_2(V,W,T) = \frac{W}{\left(|W| + \left(10 \cdot d_1 + 10 \cdot d_2 \cdot \tanh\left[\frac{V}{d_3}\right]\right) \cdot \mathrm{e}^{\frac{T-T_0}{d_4}}\right)} \tag{3.46}$$

where $r_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4, c_0, c_1, c_2, c_3, c_4, d_1, d_2, d_3, d_4$ are fitting parameters and T_0 is the reference temperature 25°C. $R1(I,T)$ is used to determine the on-state resistance through a voltage controlled voltage source (VCVS). $V1(I,T)$ and $V2(V,W,T)$ are used in a separate sub-circuit to produce a current that is fed along with drain-source voltage and junction temperature to $I1(V,W,T)$ in a VCCS, producing the drain current.

Figure 3.9 shows the IV-characteristics of the ROHM model compared to the datasheet.

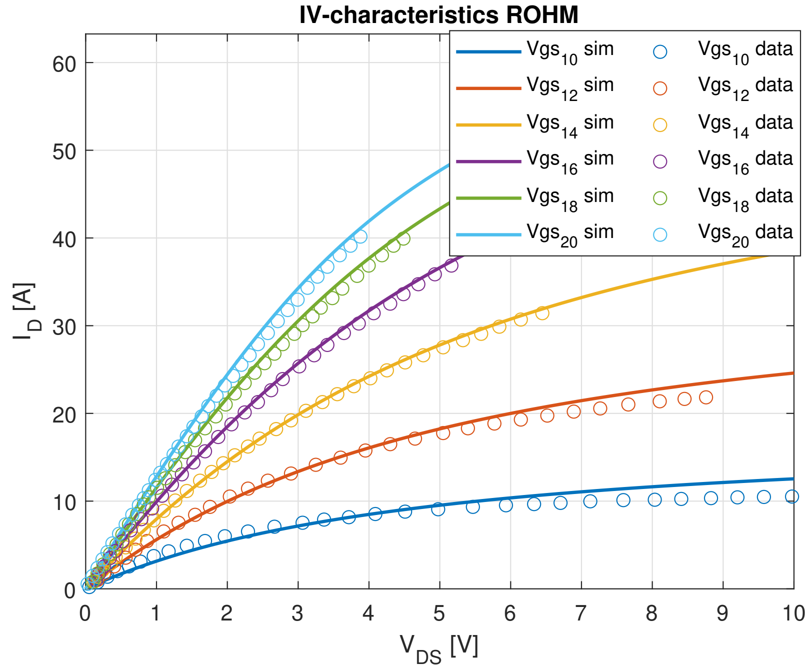


Figure 3.9: Simulation of SCT2080KE spice model, IV characteristics compared with datasheet

The ROHM model simulation results clearly follow the datasheet with good accuracy.

Furthermore, the definition of the intrinsic capacitances are also based on mathematical fitting. The gate-drain capacitance is modeled using a VCCS parallel with a capacitor, C_0, with a value fixed to 1pF. Equations 3.47 to 3.50 show the implementation of the capacitance.

$$I_{CGD} = C_0 \cdot \frac{\mathrm{d}V_{GD}}{\mathrm{d}t} \cdot \left((A + b_1 B^{k_1}) \cdot C\right) \tag{3.47}$$

$$A = a_1 \cdot \Big(\mathrm{MIN}\big(\mathrm{MAX}(V_{GD}, a_2), a_3\big) - a_2\Big) \tag{3.48}$$

$$B = \left(1 - \frac{b_2 \cdot \tanh\left[\frac{\mathrm{MIN}(V_{GD}, a_2)}{b_3}\right]}{b_4}\right) \tag{3.49}$$

$$C = c_1 \cdot \tanh\Big[(V_{GD} + c_2) \cdot c_3\Big] + c_4 \tag{3.50}$$

where $k_1, a_1, a_2, a_3, b_1, b_2, b_3, b_4, c_1, c_2, c_3$ and c_4 are fitting parameters. The CV-characteristics are shown in figure 3.10.

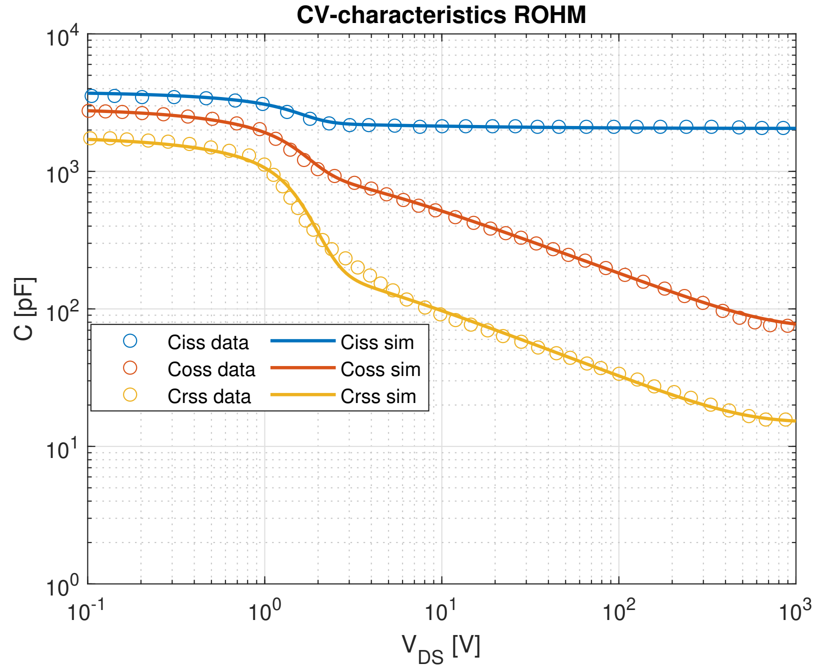


Figure 3.10: Simulation of SCT2080KE spice model, CV-characteristics compared with datasheet

The simulations show good match compared to the datasheet. Compared to Cree and STMicroelectronics, the CV-characteristics look different. The reason for this is that the ROHM datasheet used provided CV-characteristics in logarithmic scale on the x-axis while Cree and STMicroelectronics provided linear scale.

In summary, the model seems to be based solely on curve fitting to match the characteristics. There is no clear indication of material nor structure properties in any of the parameter values. However, the model seems promising as it is able to reproduce both IV- and CV-characteristics accurately.

3.4 Universal Model

Various models have been presented in section 3. Some models are based on mathematical functions that have to be fitted to measurement data in order to recreate the behavior of the device. Others are based in physics equations and relations which require detailed information about material properties and device structure. The models are summarized in table 3.1.

Table 3.1: Classifications of presented models

Model	Classification	Temperature Considerations
BSIM [11]	Numerical	Yes
EKV-model [12, 13]	Physics-based	Yes
Charge-based model [16, 17]	Physics-based	No
Non-segmented model [18]	Behavioral	No
Electro-thermal model [20]	Semi-physics-based	Yes
Curtice-Ettenberg based model [22]	Behavioral	Yes
Cree model	Semi-physics-based	Yes
STMicroelectronics model	Semi-physics-based	Yes
ROHM model	Behavioral	Yes

Focus will be on models that can be used to represent more than the devices of only one manufacturer. Information given in the datasheet is not enough to establish a detailed physical model of most devices. Hence, the most promising modeling scheme seems to be behavioral or semi-physics-based modeling. Therefore, mathematical fitting equations will be the largest contributor to represent the behavior of the device. Thus, by using expressions with enough parameters and a well defined algorithm for optimization of the fitting parameters, the behavior of different manufacturer devices can be modeled based on a single universal model. The improved Cree model [23] and the ROHM model are the chosen models. Both the ROHM model and Cree model includes thermal modeling of the device. In addition, they showed great promise when comparing the simulations to their respective datasheets, both for static (IV-characteristics) and dynamic (intrinsic capacitances) behavior. The Curtice-Ettenberg based model in [22] is also promising based on its fitting capabilities for a range of temperatures. However, it is fitted to the IV-characteristics using a described method without considering the effects of junction temperature, and covers only the IV-characteristics of the MOSFET. Those are the reasons why it is not chosen. Moreover, the improved Cree model and the ROHM model are already implemented in LTspice. This will save time compared to using the other models, as they must be implemented in the LTspice environment before optimizing using an algorithm. Both the improved Cree model and the ROHM model will be optimized using a genetic algorithm to fit $I_D V_{DS}$-characteristics from measurements.

4 Method of Data Extraction and Optimization

In this section, the measurements, data extraction and optimization methods will be explored. IV-characteristics of MOSFETs will be measured by a commercial curve tracer. Details regarding measurement method and how measurements are performed will be discussed. Furthermore, the developed algorithm for optimization will be explained.

4.1 Curve Tracer

Curve Tracer (CT) is a specialized test equipment that measures and analyzes semiconductor devices such as diodes, thyristors, MOSFETs, IGBTs and BJTs. A CT from KEYSIGHT Technologies is used in this thesis [26], shown in figure 4.1.

Figure 4.1: Curve Tracer from KEYSIGHT Technologies [26]

It is used to obtain measurements of device characteristics. The CT can measure characteristics in different ways. For example, it can be used to obtain static characteristics and device capacitances. CT measurements must be defined by the user, and can be done in multiple ways. For IV- and CV-characteristics, the CT will use different schematics based on what characteristic is being measured and step through different values of e.g. a voltage pulse or current pulse. For each step, the CT will measure either current or voltage of the device. In this thesis, the CT will be used to measure $I_D V_{DS}$-characteristics. These measurements will be used in the optimization algorithm to evaluate model performance. Two different approaches for measuring $I_D V_{DS}$-characteristics will be presented and compared. The first approach, termed V-Ipulse, is to apply several drain current pulses at constant gate voltage levels and measure the drain-source voltage. The second approach, termed Vpulse-Vpulse, is to apply several voltage pulses over drain-source at constant gate voltage levels and measure the drain current. Both methods will produce $I_D V_{DS}$-characteristics of the device. The choice between these is discussed in the following sub-sections.

The two different methods have been performed on the same Cree MOSFET, C3M0075120D [27], with other settings set equal. Table 4.1 shows the most important settings from the CT menu for each test.

Table 4.1: Curve Tracer Settings for IV-measurements

	V-Ipulse		Vpulse-Vpulse	
Setting	I pulse	V_{GS}	V_{DS} pulse	V_{GS} pulse
Start	0A	7V	0V	7V
Stop	40A	15V	15V	15V
Step	4A	1V	1.5V	1V
No. of steps	11	9	11	9
Compliance	20V	100mA	60A	100mA
Pulse Delay	0s	-	200μs	0s
Pulse Width	200μs	-	200μs	1.2ms

"Start", "Stop" and "Step" options define all the points the CT will test on the device, and subsequently measure the device. "Compliance" is an option to limit either the voltage for the current source or limit the current for the voltage source. "Pulse Delay" indicates how long it takes from the start of a new pulse period to when the pulse is applied. For V-Ipulse it is not necessary to set pulse delay, as the gate voltage is automatically set up in advance before the current pulse is applied. However, this setting is important for Vpulse-Vpulse, as the gate voltage must be set up before applying the voltage pulse in the power loop. In addition to these options, there are also measurement settings. Table 4.2 shows the measurement settings used for both methods.

Table 4.2: Measurement Settings Curve Tracer

Setting	V-Ipulse	Vpulse-Vpulse
Measurement Delay	AUTO	AUTO
Measurement Time	10μs	10μs
Pulse Period	AUTO	AUTO

Figure 4.2 sums up and illustrates how the different options affect the measurement. "Measurement Delay" indicates the waiting time before starting the measurement. "Measurement Time" sets the point in time at which the measurement is performed. "Pulse Period" is the time for completing one pulse. Both "Measurement Delay" and "Pulse Period" are set to "AUTO" for both methods. "AUTO" is a built-in option which automatically selects a value for the measurement. For "Pulse Period", it will automatically choose a value based on a set of conditions. One of these conditions is; if the sum of pulse delay and width are less than 3ms, 5ms will be selected for the pulse period. For "Measurement Delay", it sets the optimum value automatically so that the measurement is completed when the transition from peak to peak is started by the pulse output channel. I.e. based on the "Measurement Time", it will select a delay such that measurement is completed before the end of the pulse width.

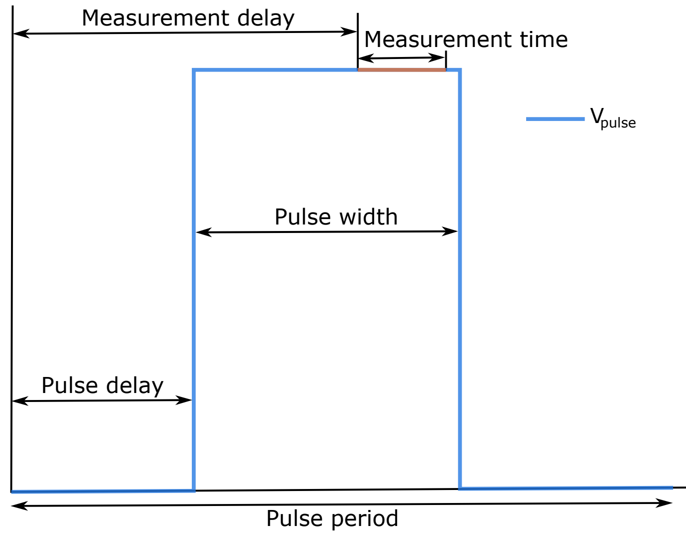
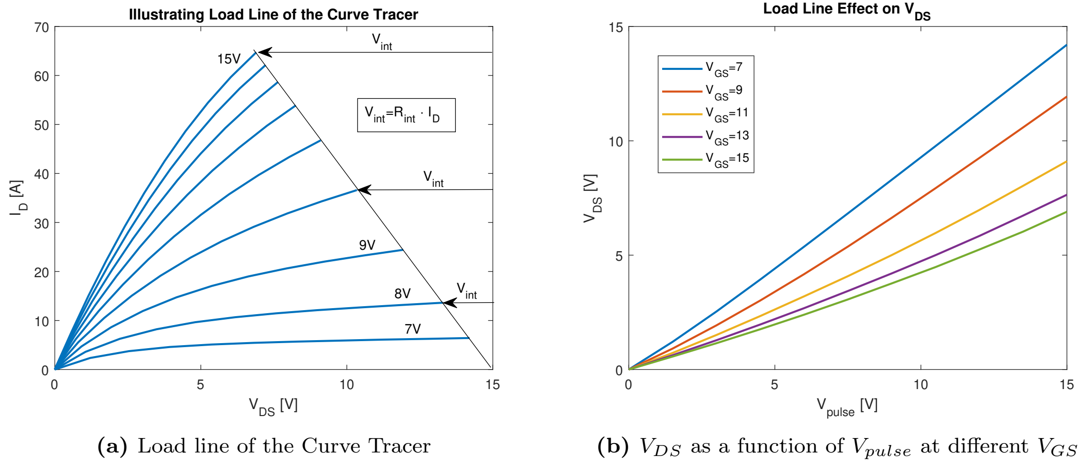


Figure 4.2: Illustration of how the different settings affect the measurement

4.1.1 Load Line

When taking measurements, the drain-source voltage will not be equal to the supply. Figure 4.3 illustrates this issue.



(a) Load line of the Curve Tracer

(b) V_{DS} as a function of V_{pulse} at different V_{GS}

Figure 4.3: Load line

In the CT there are different connections and wires that will have small resistance values. The sum of these small resistances is termed the internal resistance, R_{int}, and causes the effect seen in figure 4.3. Since the $R_{ds,ON}$ for a MOSFET is small, close to R_{int} for high gate-source voltages, there will be a noticeable voltage across R_{int}. Subsequently, the measurement taken by the curve tracer will not occur at the same drain-source voltage for different gate voltages. For some gate voltages,

a large portion of the line will be missing. E.g. at $V_{GS} = 15V$, the line stops at approximately 7V V_{DS}. However, this does not change the characteristic of the device, but must be taken into account in the optimization algorithm or by increasing the supply voltage at higher gate voltages to ensure the full range is shown.

4.1.2 Comparison of Measurement Methods

Measurements using both methods have been carried out on the C3M0075120D Cree SiC MOSFET [27]. Figure 4.4 shows the obtained data from both methods with settings as listed in table 4.1 and 4.2.

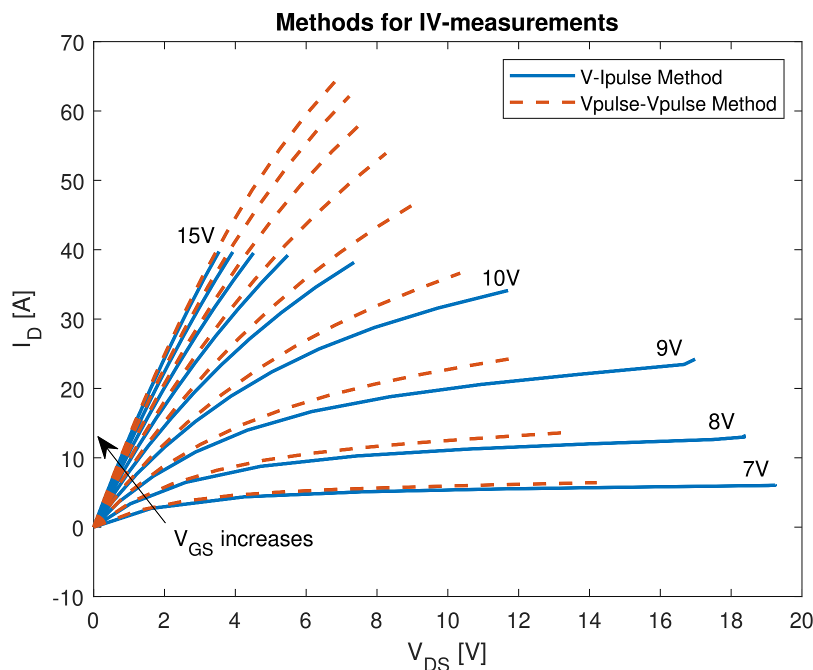


Figure 4.4: Different methods for obtaining IV-characteristic

For the first values of V_{DS}, both methods yield approximately the same current values. However, as the drian-source voltage increases above 3 to 4 volts, the difference becomes noticable. Before using the data in the optimization algorithm, it is important to investigate the differences seen in the figure, and subsequently conclude which of the methods is better suited to be used for measurements.

"Oscilloscope View" is a tool in the CT that can be used to see the time-domain waveforms of a single selected step out of all the measurement steps. Certain steps from each method have been selected for approximately the same drain current and drain-source voltage to explore potential differences. Figure 4.5 and 4.6 shows the comparison between time-domain waveforms.

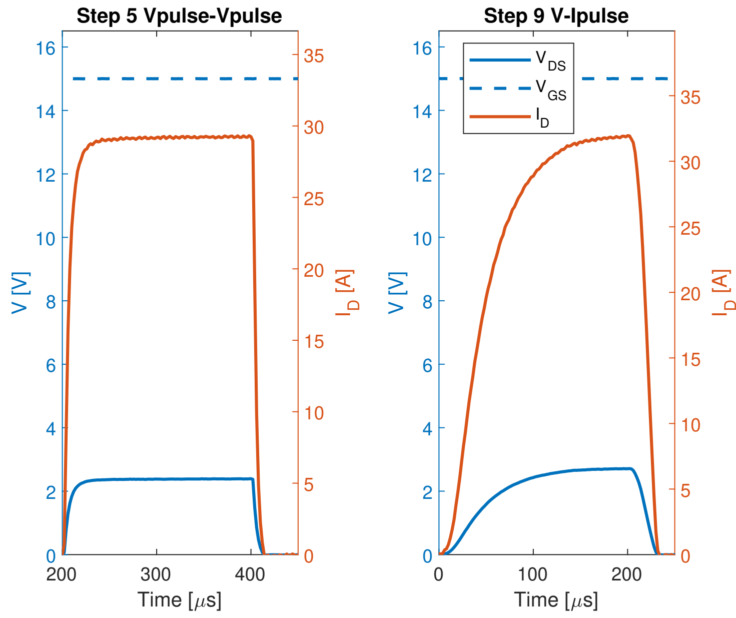


Figure 4.5: Waveforms of one pulse from the two methods at V_{GS}=15V

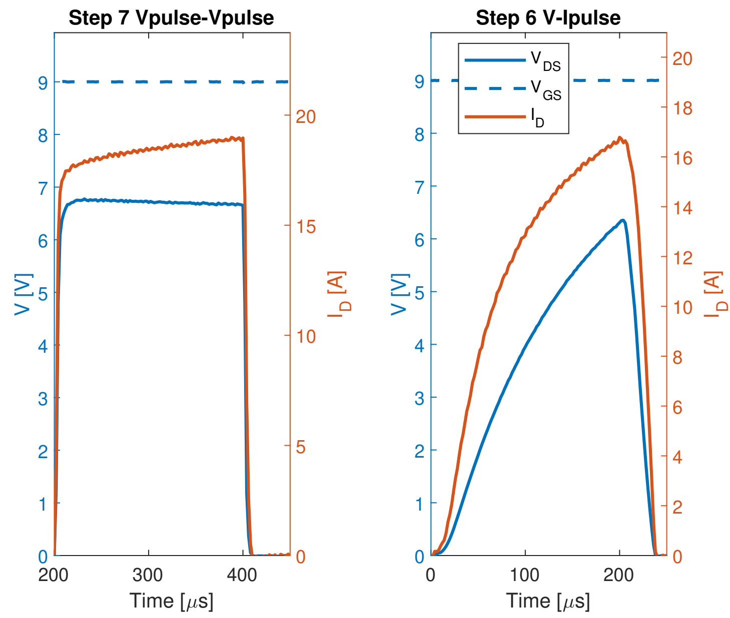


Figure 4.6: Waveforms of one pulse from the two methods at V_{GS}=9V

In figure 4.5, step 5 of Vpulse-Vpulse method is shown next to step 9 of V-Ipulse method for $V_{GS} = 15$V. In step 5 Vpulse-Vpulse, the CT will try to apply a voltage pulse $V_{pulse} = 1.5 \cdot 4 = 6$V. Due to the internal resistance of the equipment, the measured drain-source voltage, V_{DS}, is lower. For step 9 V-Ipulse, the CT will try to apply a current pulse $I_{pulse} = 4 \cdot 8 = 32$A.

In figure 4.6, step 7 Vpulse-Vpulse is shown next to step 6 V-Ipulse at $V_{GS} = 9$V. Step 7 Vpulse-Vpulse will have a voltage pulse $V_{pulse} = 6 * 1.5 = 9$V. For step 6 V-Ipulse, the CT will try to apply a current pulse of $I_{pulse} = 5 * 4 = 20$A.

From the waveforms, the voltage pulse in the Vpulse-Vpulse method has a low rise time compared to the current pulse in the V-Ipulse method. This is especially seen in figure 4.6, where the current pulse does not reach a plateau. It is important that the pulse reaches a plateau and the measurement is taken during a more or less constant value of current or voltage pulse depending on the method. Another test was done with increased pulse width to show the effects on the waveforms and the IV-curves produced by the measurements. Figure 4.7 shows step 6 V-Ipulse waveforms at V_{GS}=9V for different pulse width.

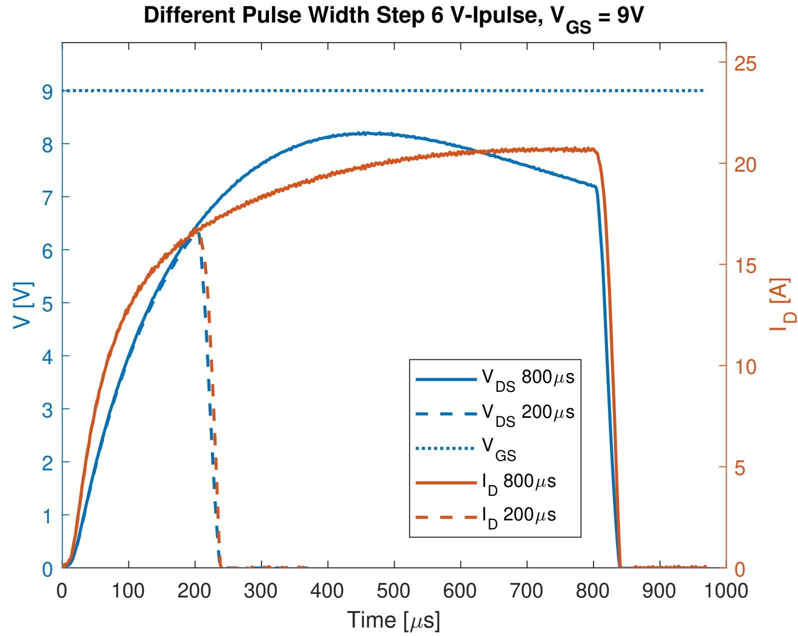
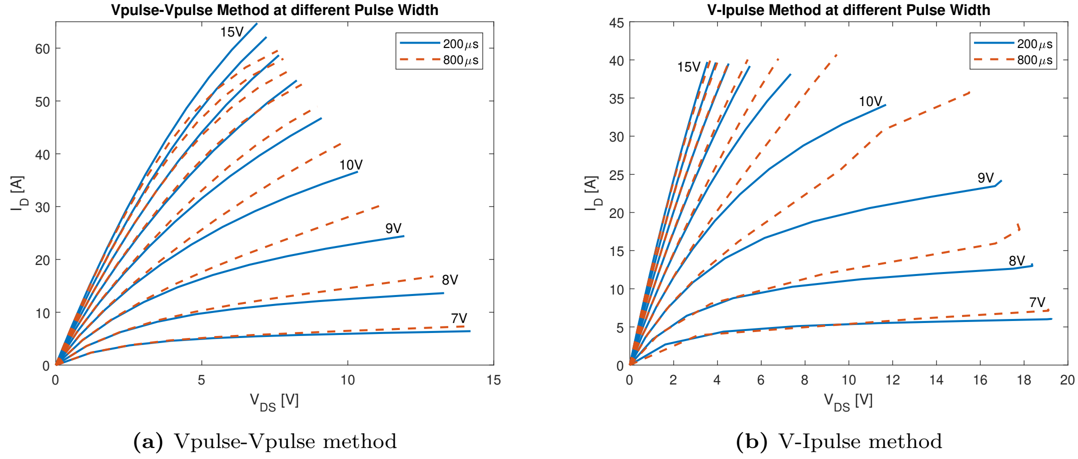


Figure 4.7: Different pulse width for the same step in the V-Ipulse method at V_{GS}=9V

From the CT settings for V-Ipulse method, the current pulse at step 6 is 20A. It tries to apply it, but it takes a long time for the current to reach the value. The current pulse flats out towards the end for the 800μs pulse width. However, it is still very slow and the drain-source voltage varies even though the current has flattened out. Consequently, "Measurement Delay" and "Measurement Time" will have an impact on the measurement. With "AUTO" delay, the measurement is taken at the end of the pulse, just before the drain current and drain-source fall back to zero. Figure 4.8 shows the IV-curves for each method at different pulse width.



(a) Vpulse-Vpulse method

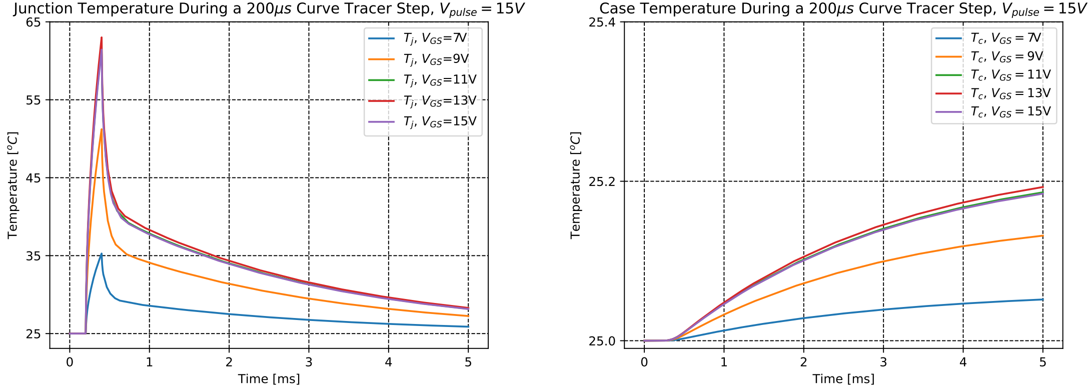
(b) V-Ipulse method

Figure 4.8: Effects of pulse width on IV-characteristics produced by CT

Both methods show differences when increasing the pulse width with "AUTO" measurement delay. For the Vpulse-Vpulse method, the characteristics from different pulse widths are almost equal until V_{DS} reaches 4V. At that point 800μs shows higher values for V_{GS}=7-11V and lower values for high V_{GS}=12-15V.

For the V-Ipulse method, the characteristics from different pulse widths are matching closely up to V_{DS}=4V. However, for higher V_{DS}, the 800μs shows higher values for most of the gate voltages. Furthermore, the difference is particularly large at V_{GS}=8-10V.

The causes of these differences must be investigated. Firstly, the measurement delay will impact the measurements by looking at the time-domain waveforms. In the case of figure 4.7, the drain current and drain-source voltage have large variations during the pulse. These variations might be a result of temperature changes occuring on the junction. The impact of junction temperature changes on the measurements will be investigated. Once the MOSFET starts conducting, the temperature of the die will increase immediately. However, only the case temperature can be measured by the CT. Thus, to investigate the temperature changes of the die, simulations have been performed on the thermal model provided by Cree in LTspice. In the simulation, ambient temperature is set to room temperature (25°C) by connecting a voltage source in series with a resistor to the case temperature pin in the model. The junction temperature pin is left open. This is to allow the junction temperature to vary. A constant gate voltage is applied and in the power loop, a voltage pulse will be applied with equal settings as used in the CT. This simulation is repeated for different gate voltages at the same V_{pulse} in the power loop. Case and junction temperature are shown in figure 4.9 in time domain. The simulation circuit can be found in appendix C.

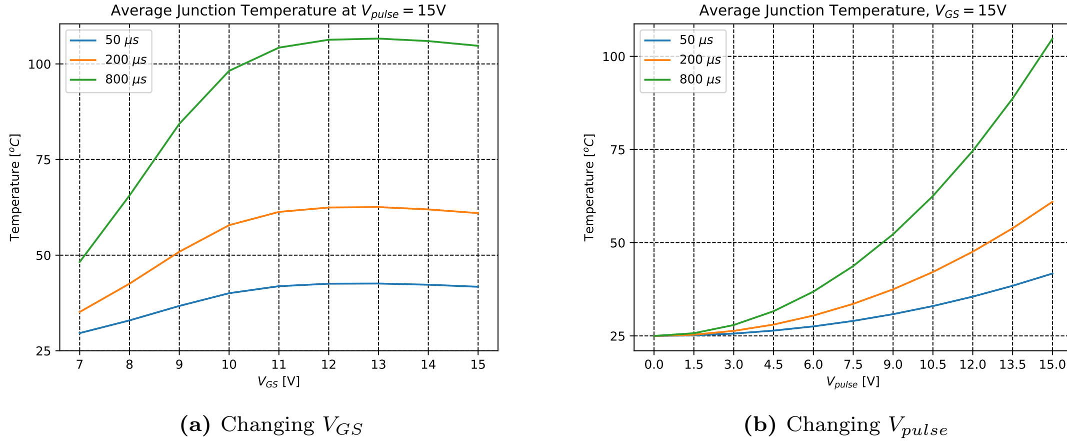


(a) Junction temperature during a measurement step at different gate voltage

(b) Case temperature during a measurement step at different gate voltages

Figure 4.9: Simulation of junction and case temperature during a measurement step at different gate voltages

Junction temperature increases significantly during the applied pulse in the simulations. The peak value increases with the gate voltage when keeping the value of the applied voltage pulse constant. However, the case temperature shows only a small increase. After a complete CT test, the case temperature of the device increases by approximately half a degree. Since only the case temperature is measured by the CT during the characterization, the changes on junction will remain hidden. Thus, the characteristics are said to be for room temperature (25°C), while in reality, each measurement point of the characteristic is for a different temperature compared to room temperature. Figure 4.9 indicates that the pulse period is important for the measurements. Each line in the figure represents one step performed by the CT. Short pulse period wont allow the junction cool down to its original value before starting the next step. This will add to the temperature difference already occurring. Therefore, the pulse period should be long enough to let the junction cool down. 5 ms pulse period is used in the CT settings. Additionally, the pulse width of V_{pulse} will have an impact on the junction temperature increase. The peaks occur at the end of V_{pulse}. I.e. the heating happens during V_{pulse}. Short pulse width will give the junction temperature less time to increase. Thus, allowing for smaller pulse period and less heating of the component overall. Yet, the measurements are taken close to the end of the pulse, meaning the junction temperature will still vary for the same drain-source voltage at different gate voltages. Since higher gate voltages allow higher drain current at the same drain-source voltage, the power dissipated in the die will increase with both gate and drain-source voltage. To investigate this, the average junction temperature during the measurement has been simulated. The average junction temperature is measured as the average of the junction temperature over the last $10\mu s$ of V_{pulse}. This is to mimic the CT settings, which use a measurement time of $10\mu s$. Results from simulations using different pulse width are shown in figure 4.10.

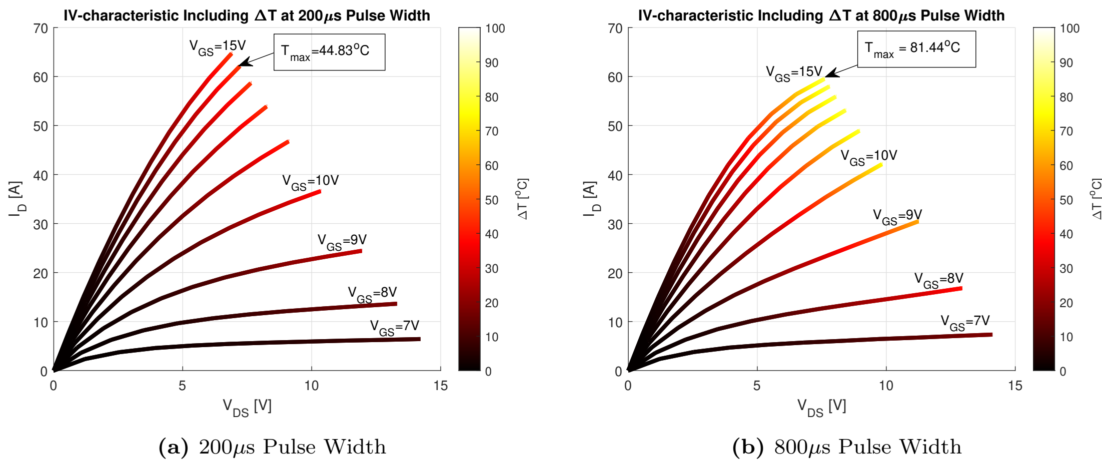


(a) Changing V_{GS}

(b) Changing V_{pulse}

Figure 4.10: Temperature changes on junction for different pulse widths

The temperature difference is significantly visible when both V_{GS} and V_{pulse} are high. However, figure 4.10b shows that there is a small temperature difference for low V_{pulse}. The first few steps of V_{pulse} have approximately the same average junction temperature. Regardless, when V_{pulse} increases past 3V, the differences between pulse widths are clear. Longer pulse width means more energy fed to the device, generating heat. This results in a larger average junction temperature during the measurements of drain current and drain voltage. Hence, introducing inaccuracies in the measurements since it is meant to be taken at 25°C, while the junction temperature is much larger when performing the measurement in reality. To illustrate the issue, the temperature difference between junction and case, ΔT, at each measurement point have been mapped on top of the IV-curves from the Curve Tracer measurements. The resulting figure, 4.11, shows how the temperature difference between junction and case varies during the measurements.



(a) 200μs Pulse Width

(b) 800μs Pulse Width

Figure 4.11: IV-characteristics including ΔT mapped as third variable to a colormap

It is observed that ΔT increases as the drain current and gate voltage increases. Higher drain current translates to higher power consumption for the MOSFET, and more energy is converted into heat. The conduction losses during the pulse is given by equation 4.1.

$$P = I_D \cdot V_{DS} \tag{4.1}$$

Thus, a source of error is introduced in the CT measurements. ΔT is not the same for all the measurements. However, the measurements are taken at the same temperature if the case temperature is used as reference. It is not possible to measure the junction temperature using the CT. Yet, the temperature difference that occurs in a measurement between junction and case can be found through simulation using the manufacturer thermal model or using thermal calculations. ΔT found through simulations have been verified by thermal calculations using equation 4.2.

$$\Delta T = Z_{thJC} \cdot P_{MOSFET} \tag{4.2}$$

where Z_{thJC} is the junction to case thermal impedance and P_{MOSFET} is the power consumed by the MOSFET. The integral of IV-characteristic serves as the power reference for each measurement point and the thermal impedance found in the component's datasheet. More detailed explanations are given in appendix D. Furthermore, the temperature difference, ΔT, between case (room temperature) and junction temperature is higher for long pulse widths. Consequently, the measurements should be taken as early as possible and the pulse width should be reduced to mitigate large ΔT.

Two methods have been explored for $I_D V_{DS}$-characterization using the curve tracer. Equal settings have been used for both methods. From the settings, the measurement delay is set such that the measurement is taken during the last $10\mu s$ of the applied pulse. Thus, it is important to have stable voltage and current signals, such that they are constant during the measurement. Junction temperature has also been investigated. Changes on the junction temperature occurs when the MOSFET starts to conduct. When the drain current and drain-source voltage increases, the temperature difference between junction and case increases. Based on the pulse rise time, junction temperature and the results presented in this subsection, the Vpulse-Vpulse method is the better choice for IV-characterization when using the CT. The voltage pulse has a very small rise time compared to the V-Ipulse method, which allows for lower measurement delay. It is still important that the signals are stable during the measurement. By looking at the oscilloscope views, the measurement delay could be set to $40\mu s$ and the pulse width set to $50\mu s$ for the Vpulse-Vpulse method. Furthermore, the pulse period should be high to ensure the junction temperature has fallen to its original value before applying another voltage pulse. Vpulse-Vpulse method has been selected for $I_D V_{DS}$-characterization in this thesis. The main reason behind this choice is based on the time-domain waveforms. Vpulse-Vpulse method shows a fast rise time and stable signals during the pulse. This can allow for lower measurement delay. Additionally, the Vpulse-Vpulse method shows more stable curves for higher pulse width.

4.2 Parameter Extraction and Optimization

The improved Cree model [23] and the ROHM model will be used for optimization and fitting to the CT measurements. To limit the scope of the thesis, the main focus will be on static behavior by optimizing model parameters to fit the CT measurements when considering the changes on the junction temperature occurring in the measurements. This subsection will give a brief introduction to evolutionary algorithms and explain the chosen algorithm for finding and optimizing the parameters of the different models. Elements of the algorithm, such as parameter limits, procedure and termination criteria, will be discussed.

4.2.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) are algorithms based on the theory of evolution of species through natural selection [28], which means they are based on the evolutionary theory of Charles Darwin [29]. Some of Charles Darwin's discoveries and observations heavily influence EAs. Firstly, more offsprings are produced than could possibly survive. Secondly, characteristics varies between individuals of the population, which can impact the rate of survival and reproduction. Lastly, some of these characteristics are heritable. Additionally, Gregor Mendel's work on genetics [30], proving that evolution in organisms happen due to changes in heritable traits, serve as a foundation of EAs. The emergence of EAs started in the 1950s and -60s, when several independent studies were performed on using evolutionary systems as an optimization tool for engineering problems [31, 32]. The idea was to evolve a population of individuals, where an individual represents a possible solution to a given problem, using operators inspired by natural selection and variation. It is a heuristic approach to find a solution to a problem. Fit individuals survive and the populations evolve heavily influenced by the fittest individuals. The genetic algorithm is one of the early EAs to emerge [33]. Today, there are also different variations of EAs inspired by other elements and phenomenons found in nature. Some examples are; particle swarm optimization algorithm (PSO) which is motivated by intelligent collective behavior of animals such as flock of birds or schools of fish [34], and ant colony optimization algorithm (ACO) which is inspired from the foraging behavior of ants [35].

EAs are suitable for solving problems that have a vast number of possible solutions. Evolution can be abstracted as a method for searching large number of possibilities in an ever changing environment. Highly fit organisms are more likely to survive and reproduce, passing on their genes to the next generation. In the biological world, there is no single perfect solution, but many creative and viable solutions. Evolution could be described as a method for designing innovative solutions to complex problems [32]. In a perfect scenario, all possibilities would be calculated and compared to find the best solution to a problem. However, that is not feasible for many problems, as the number of possibilities are enormous and would take an infinitely long time to calculate. Evolution has proved itself to work in an ever changing and complex environment. Yet, solving problems using EAs are not guaranteed to yield the best or most optimal solution in the search space. It will find solutions that in many cases are good enough. In this thesis, the genetic algorithm (GA) is the chosen algorithm to optimize model parameters in SiC MOSFET models.

4.2.2 Genetic Algorithm

Genetic algorithms (GAs) were invented by Johan H. Holland in the 1960s and were developed by Holland and his colleagues at the University of Michigan [32]. Holland's GA is a method for moving one population to a new population using operators of crossover, mutation and inversion, inspired by natural selection [33]. Several "chromosomes" - termed individuals in this thesis - consisting of genes, make up a population. Each gene is represented as a bit, e.g. 0 or 1. There is no exact answer to when a GA should be used, though there are some aspects that makes it a viable option. GAs are good candidates for solving problems where the search space is not well known. Additionally, some problems may not have the need for exact or optimal solutions. Investigating and comparing different SiC MOSFET models to potentially find one that can be used to model SiC MOSFETs on a general scale, can be a difficult task, especially when the models vary in implementation. The improved Cree model is semi-physics-based. It includes a factor that has a few parameters that can be modified to fit measurement data. On the other hand, the ROHM model has twenty non-physical parameters that can be altered to obtain good fit to measurement data. Finding the best solution for each model may result in different optimization strategies and implementations. As long the model has changeable parameters, the GA can optimize different models with the same method. Moreover, finding the optimal solution by simulating the model for all possible parameter values is not feasible. Additionally, the search space may not be well known and certain parameter values can result in simulation errors or convergence issues. Regardless, a GA can perform well under these conditions and can be used to find possible solutions for both models within the same algorithm. The same algorithm can be used to optimize both the improved Cree model and the ROHM model. A global solution is not mandatory and may not be found by the GA as it can get stuck in a local minimum. However, the solution found by the GA might be sufficiently good enough for representing the measurement data. Nevertheless, the chosen genetic operators are crucial for the performance of the GA. In this sub-section, different parts of the algorithm will be presented. That entails how it is structured and what methods have been decided for the different steps in the GA.

4.2.3 Implementation of the GA

This section will explore the different elements of the GA.

GA Process - Flowchart

A general overview of the structure and flow of the GA will be given in this section. A flowchart of the GA is shown in figure 4.12.

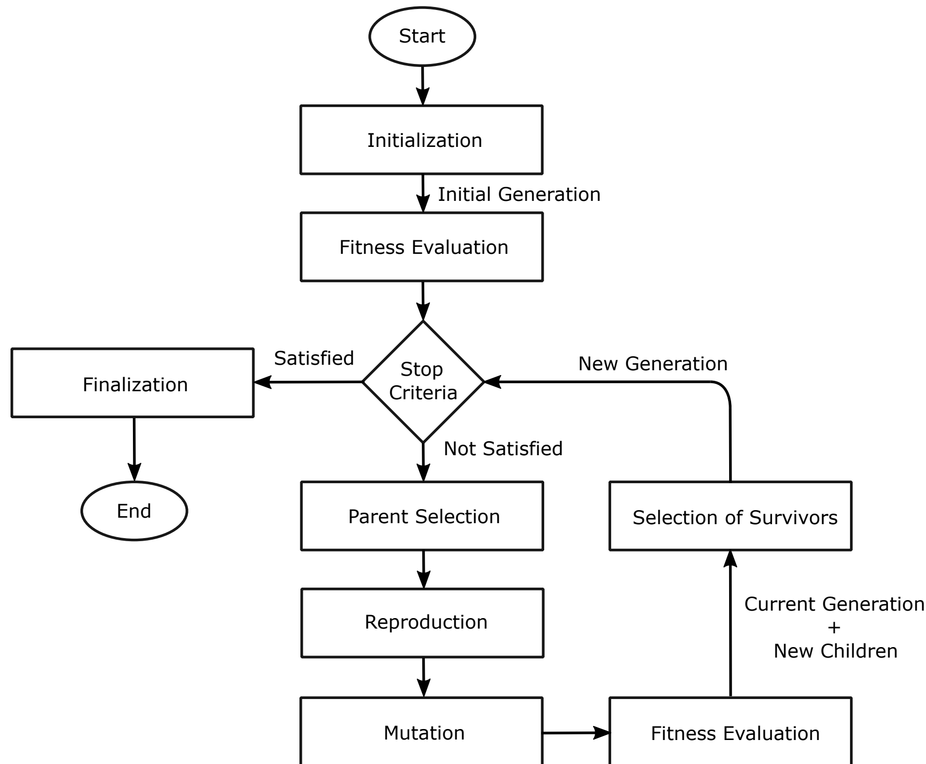


Figure 4.12: Simplified flowchart of GA

The GA takes in several arguments, configuring the algorithm as well as loading in the measurement data to be compared. Initialization fills up the initial population by creating individuals with random gene values. Then the initial population is simulated in LTSpice and sent to fitness evaluation. Each individual in the population is compared with the measurement data and given a fitness value. Lower fitness value translates to a fitter individual in this implementation. The fittest individual is then compared to the stop criteria to see if the algorithm should end or evolve further. If satisfied, the algorithm enters a finalization step where the fittest individual is returned to the outer scope of the algorithm. If not satisfied, the algorithm enters the evolution steps. Here the genetically inspired operators are performed on the population. Parent selection is where individuals are selected for reproduction. Reproduction defines the method for transferring genes from both parents to the off-spring (child). Thereafter, the child may mutate in the mutation step. Each gene

of the child has a chance of mutating based on a given mutation rate that is chosen by the user of the algorithm. Once all the children have been produced, a sorting algorithm will select survivors from all the individuals - now containing individuals from old population and new children - based on fitness. The fittest individuals survive and move on to the new population, the next generation. This procedure is then repeated until the stop criteria is fulfilled.

Class Structure and Encoding

Object-oriented programming (OOP) has been chosen as the primary programming method for the algorithm. Different objects such as population, individuals and genes can be defined more intuitively using OOP. They have been defined as in figure 4.13.

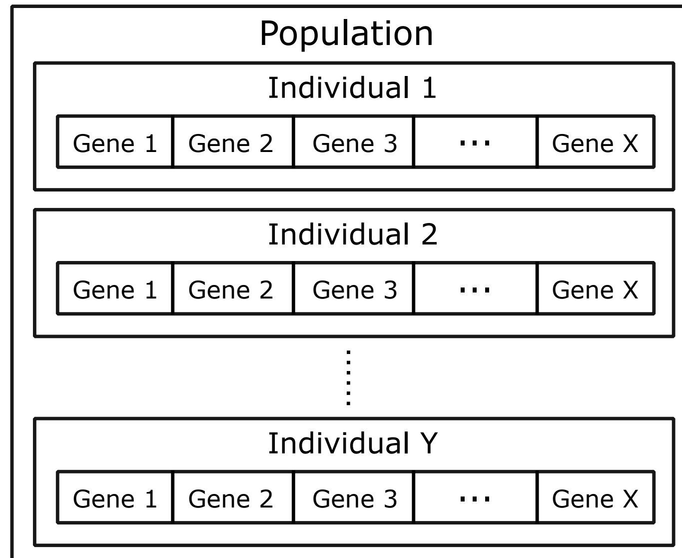


Figure 4.13: Class Structure

A population contains a number of individuals given by the population size decided by the user of the algorithm. Each individual in the population has a set of genes, each representing a model parameter in an LTspice model. The genes are defined as class objects. A single gene object contains a parameter name, a parameter value and a set of bounds, a lower and upper bound. An individual will then be a vector of several gene class objects. The parameter value is represented as a real number.

Initialization

The initialization step will load a wide range of arguments passed by the user. It will create the genes representing model parameters and fill the population with a number of individuals equal to the population size. These individuals will have random values on their genes, limited by the gene boundaries. Once the initial population has been established, each individual in the population is simulated. I.e. the individual's genes are written to the provided model before instantiating

an LTspice simulation. After the simulations, the individuals are evaluated based on the fitness function, which will be explained in another paragraph. To be able to evaluate different parts of the algorithm, it is necessary to use seeds. A seed is passed to a built-in python function and is used to determine the sequence of numbers used by the random functions of python. If the same seed is passed to the algorithm while all other functionality remains unchanged, the GA will give exactly the same results as previous runs with equal seed.

Simulation in LTSpice

Each new individual will be simulated through LTSpice. It has to be configured in such a way that it will produce IV-characteristics. There are several ways this can be achieved, two of which will be discussed. The first simulaton method is to use transient analysis of the circuit in LTSpice. Transient analysis gives the circuit response over time. This will include dynamic behavior. To produce IV-curves using this method, two measurement commands must be used. One to measure the drain current and another to measure the drain-source voltage. These values are stored in a separately generated log-file by LTSpice, which can be read by the algorithm to get the results. Furthermore, the circuit has been configured to mimic the CT measurements. Internal resistance of the CT has been calculated using the IV-measurements and added in series with the voltage source applying the voltage pulse. Additionally, the case temperature is set to room temperature, 25C^o. The voltage source and measurement commands have been configured the same way as would be for a CT step. The voltage pulse lasts 200μs, where the last 10μs of the pulse is when the average drain-source voltage and drain current are measured. That means the simulation needs a total of $9 \cdot 11 = 99$ steps to obtain the necessary data used in the fitness evaluation.

The second method is DC sweep analysis. DC sweep is an analysis of static characteristics, such as the IV-characteristics. It will find steady state operating points for different V_{pulse} values, using V_{pulse} as the x-axis unlike transient analysis, which uses time along the x-axis. The DC sweep simulation will have equally many steps as the transient analysis. However, each step will be faster to calculate since the time response is ignored. Moreover, the self-heating component of the ROHM and Cree models will not be active during a DC sweep. I.e. the junction temperature will be the same as the temperature set on the case, 25C^o. To adjust for temperature changes that occur on junction during the CT measurements, the junction temperature must be set. The temperature is set for each measurement point by values from the thermal calculations. A More in-depth explanation of how to add a different temperature to each measurement step can be found in appendix E.

The two simulation methods produce equal results. Thus, other requirements may be used to select a preferred method. Transient analysis may seem to be a good choice as it replicates the actual CT measurements. However, the nature of a transient analysis is to simulate dynamic behavior and the simulation tool will spend unnecessary time to calculate other elements such as the switching behavior. DC sweep finds the steady state operating point. Hence, skipping all the calculations of dynamic behavior, subsequently making it complete a simulation faster compared to transient analysis. Table 4.3 shows the time investment per simulation using the different methods.

Table 4.3: Comparison of simulation time using the two methods

CPU	AMD A10-7300 Radeon R6, 1.90GHz	
Method	Transient	DC-sweep
Min	20s	2s
Max	40s	6s
Timeout	40s	40s

DC-sweep is much faster compared to transient analysis. Some simulations using the transient analysis were aborted due to timeout. Timeout is used to prevent LTspice from being stuck for an excessive amount of time, e.g. convergence issues. If the simulation is not completed within a certain amount of time equal to the timeout value, then it will automatically close and the algorithm continues. Time is an important factor since the GA will usually use a decently large population size to create large diversity in the beginning. Furthermore, several iterations are required to be confident the algorithm has converged and each individual must be simulated once. As DC sweep is several times faster than transient analysis, it allows the algorithm to work faster and find a possible solution earlier compared to using transient analysis. Based on this, DC sweep is the selected simulation method used to simulate individuals in the GA.

Parameter Bounds and the Search Space

In maximization and minimization problems there will always be one or more objective functions limited by a set of bounds. In this case, the objective is to minimize the difference between measurement data and simulation results. The simulation results changes based on the model parameter values. It is indirectly a function of many parameters. Each model parameter will have a certain effect on the simulation results. This means that limits on the model parameters will also limit the the scope of the simulation results. However, these bounds are set by the user of the algorithm. Choosing bounds can drastically change the time requirements for the algorithm to find a good solution, as it does not have to simulate individuals guaranteed to be weak. Some examples will be given on the approach behind setting certain limits.

If parameters are used in special mathematical operators, such as logarithms or inverse trigonometric functions, certain parameter values can produce undefined results. Since complex analysis is not used, negative values can not be given to a logarithm. That will result in an error. Likewise for inverse sine, values more negative than -1 and larger than 1 will result in an error. Consequently, certain special functions are automatically limited, which will impose a limitation on the parameter value.

Other examples are exponents. If a parameter is used in an exponent it might be safe to assume that the parameter should have the same sign. Most MOSFETs have similar shape on their IV-characteristics and in many cases the models of these characteristics contain exponents. By changing the sign of a parameter in an exponent, the result can be drastically different.

Lastly, parameters can exist as factors, addends and quotients. For quotients it is important to avoid division by zero. Furthermore, if the result is supposed to be positive, an outer factor multiplied with a positive number must be positive as well.

In conclusion, the parameter limits will in special cases have to be limited carefully. However, most variables will not be limited by the function it is used in, it will instead be limited by reason.

The search space will be given by the limits of the parameters. Moreover, certain parameter values can cause errors or convergence issues in the simulation. Handling of such cases must be added. If a simulation takes longer than a timeout variable, it will be aborted and properly handled in the other steps of the algorithm. Otherwise, if the simulation produces an error, it will be handled as if it was aborted due to timeout.

Fitness Evaluation

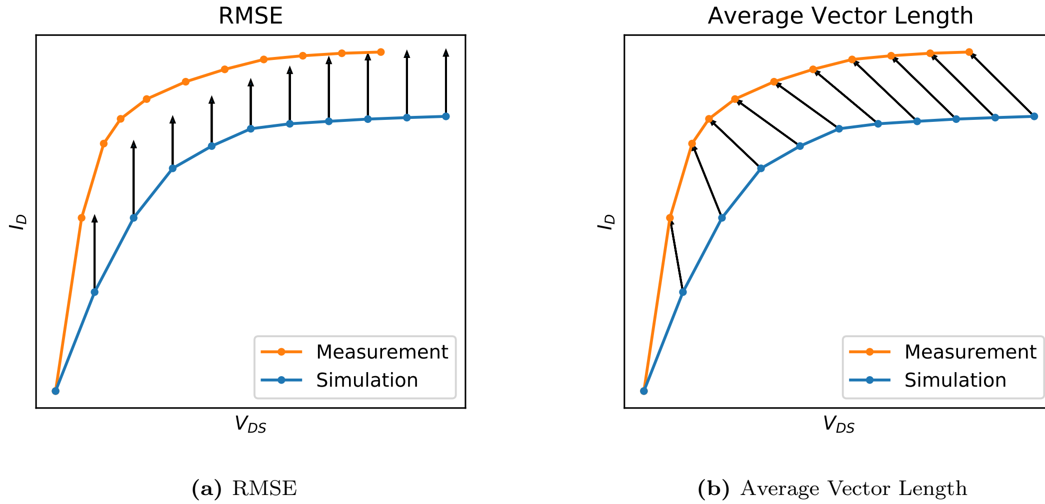
Fitness evaluation implements the objective function of the algorithm. It defines what the algorithm will focus on and the results can vary greatly depending on the chosen fitness function. Thus, it is important to carefully choose a good fitness function for the application. The objective in this thesis is to fit an LTspice model to CT measurements of IV-characteristics performed on SiC MOSFETs. That means the algorithm must be fed measurement data of IV-characteristics. The objective is to fit a simulation model to IV-measurements of SiC MOSFETs. This can also be described as a minimization problem, where the objective is to minimize the difference between the measurements and the curves produced by the simulation model. A commonly used method used to determine the fitness is the "Root Mean Square Error" (RMSE). RMSE obtains the error between two data-sets and is given by equation 4.3.

$$\mathbf{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{4.3}$$

where y represents the measurement data, $\hat{y}$ represents the simulation results and n represents the number of measurement points from the CT measurement. However, there is one drawback when using RMSE in this case. For many fitting operations, one of the axes is constant. E.g. time along x-axis. That means the error between two data-sets are only on one axis, in one dimension. However, the IV-curves produced by the simulations can have variations in two dimensions. The model can produce a large current for small voltages or opposite. This will move the curve along both axes. Consequently, the RMSE must be expanded to include two dimensional error or changed to accommodate for these variations. The chosen strategy is to use the average length of the vector that occurs between measurement and simulation points. It will be defined by equation 4.4.

$$\overline{|\vec{\mathbf{A}}|} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2} \tag{4.4}$$

where (x_i, y_i) is the measurement data point, $(\hat{x}_i, \hat{y}_i)$ is from the simulation results and n is the total number of points. The root in equation 4.4 calculates the vector length between a measurement point and the equivalent simulation point. After summing up all the vector lengths, it is divided by n which results in the average vector length. This fitness function takes into account the error that can occur on both drain current and drain-source voltage for each measurement point. An example of how the presented fitness functions will behave is shown in figure 4.14.



(a) RMSE

(b) Average Vector Length

Figure 4.14: Different fitness functions

The vectors in the figures represent the error that will be minimized by the objective function to obtain better fit. RMSE takes only into account the error in one dimension, in this case the y-axis. However, both drain current and drain-source voltage from simulations can be different compared to the measurement data. Thus, using RMSE makes it possible to achieve a low fitness value for a bad fit. It will make the algorithm optimize only for the drain current or drain-source voltage. Average vector length method does not suffer from the same problem. It will try to minimize the error vectors that spans in two dimensions. By using the average vector length to calculate fitness, it follows that smaller fitness is equivalent to a better match. I.e. if the calculated fitness is zero, there is a perfect fit between the measurement points and simulation data.

Stop Criteria

Stop criteria, also called termination criteria, can be defined in multiple ways. One approach is to use a number of generations or simulations as maximum iterations. The GA will run until it has calculated an amount of generations or individuals and then go to the finalization step. Additionally, a fitness condition can be added, such that when the fitness value if the fittest individual in a population is below the given value, it will break off and proceed to the finalization step. However, a max iteration limit should be used in parallel with the fitness condition. The GA may not be able to find a solution satisfying the fitness condition, or the model may not have enough flexible parameters preventing it from accurately reproducing the measurement data. Alternatively, how many generations the current fittest individual has survived can also be used as a stop criteria. E.g. the GA runs for some time and in the last 50 generations, the fittest individual has remained the same. The GA might have converged and found a good fit for the model.

Parent Selection

A major part of any GA is how to perform selection. I.e. how to select individuals that will produce off-springs. Several selection methods exist and are used in various implementations [32]. Some

methods will be presented here before explaining the selected method.

Elitism is a selection method where only the fittest individuals of a population are selected as parents. A disadvantage with such an implementation may be that the GA could prematurely converge to a sub-par solution. However, elitism is often used as an addition to other selection methods. In certain selection methods, the fittest individuals may be lost in the next generation. Thus, adding an element of elitism to the selection process can retain some number of the best individuals at each generation.

Fitness-Proportionate Selection is a method where fit individuals are more likely to be selected. A common way to implement this method is by "roulette wheel" sampling. That means each individual will occupy a slice proportional to their fitness. The wheel is then spun several times to determine the parents. Subsequently, this process is repeated until a given number of children have been produced. Fitness-proportionate selection can suffer from premature convergence as with elitism. Additionally, it may suffer from time complexity. The roulette wheel slices must be calculated and depending on the implementation, the population may have to be sorted. For very large populations, this can be a time consuming procedure.

Steady State Selection is a selection method that focuses on retaining a large portion of the current population. Only a few individuals from the old population are replaced by new children. In some applications, the initial individuals are completely random and may, as a result, have bad fitness values. Keeping these in the next generation can make the GA slow.

Rank Selection is an alternative method to prevent premature convergence. The individuals in the population are ranked based on certain criteria. It could be based on fitness, distance from other individuals or both combined. Ranking avoids giving certain individuals the largest share of off-springs. A possible disadvantage with rank selection is that by reducing the selection pressure of the fittest individuals, the GA may be slower in finding new highly fit individuals. Like fitness-proportionate selection, rank selection can also suffer from time complexity. The rank must be calculated for each individual and then sorting the entire population by rank. For large populations, it may be a time consuming process.

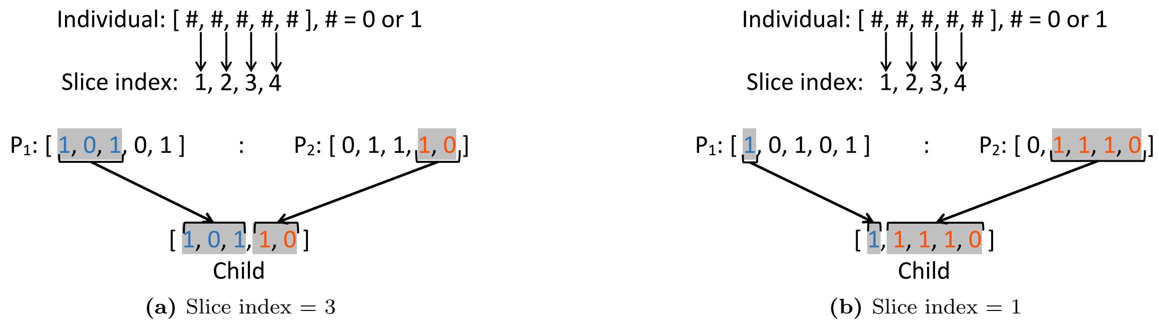
Tournament Selection is a selection method that uses several "tournaments" to select individuals. A tournament is a comparison between a few randomly selected individuals of the population. The size of the tournament, i.e. the number of selected individuals must be chosen. The fittest individual among the ones in the tournament is selected. Two tournaments or the two fittest individuals in a single tournament can be selected for reproduction. This process is repeated until enough off-springs have been produced. Choosing small tournaments keeps diversity, though it may come at a cost of convergence speed. By choosing large tournaments, weak individuals are less likely to be selected, causing a loss of diversity.

The chosen selection method in the GA is the fitness-proportionate selection method with "roulette wheel" sampling. This selection method makes it so all individuals have a chance on reproducing, though an element of elitism is added since the fittest individuals are more likely to be selected. As fit individuals have low fitness value, a mapping is required to allocate the shares of the wheel.

After calculating the portions of the wheel for each individual, the wheel is spun twice to select the parents for reproduction. If the same parent is selected twice, the wheel is spun until the second parent is not equal the first. As previously mentioned, fitness-proportionate selection may suffer from premature converge. However, countermeasures have been added in other parts of the algorithm to negate this. Another selection method is used when determining the new population. Fitness-proportionate selection is only used to select two individuals that will produce an off-spring. Moreover, the time complexity issue as a consequence of having large population sizes will not pose a problem for the algorithm. This is because the simulation time for each individual will automatically constraint the population size. As each simulation can take several seconds, having a large population size will make the algorithm use a long time before completing an iteration and creating the next generation. Time complexity related to sorting algorithms will not be a limiting factor in the algorithm, the limitation is on the simulation time.

Reproduction

An important aspect of a GA is the reproduction step, often called crossover. This is where new children are made. Based on evolutionary theory, children inherits genes from their parents. A commonly used method is to determine one or more indices that will be used to "cut" different parts of the genes from each parent and merge them to create the new child. This is described by Holland as crossover [33]. An example would be as shown in figure 4.15.



(a) Slice index = 3

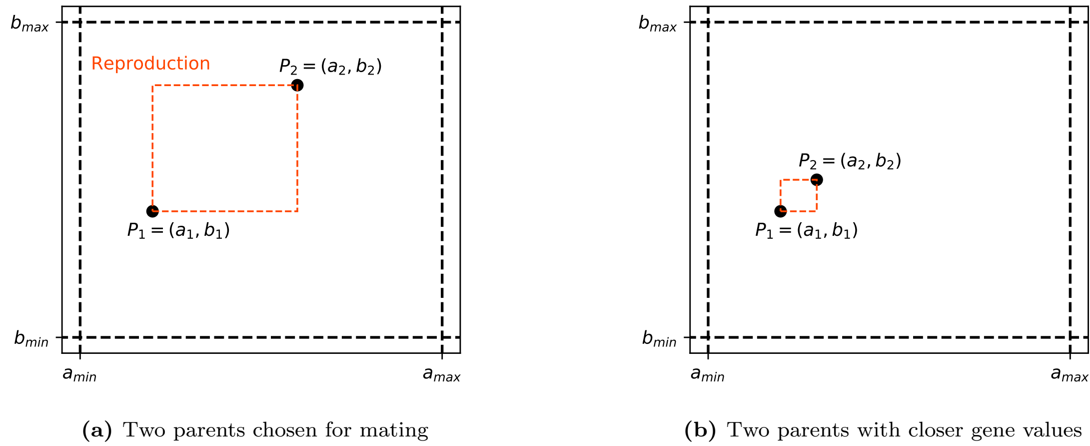
(b) Slice index = 1

Figure 4.15: Crossover

In the example above, each individual has 5 genes which can have values 0 or 1. A child will get direct copies of a given number of genes from each parent. As the slice index is 3 in figure 4.15a, the child gets the first three genes from parent P_1 and the last two genes from parent P_2. Thus, the new child is similar to its parents, but not identical. Such a strategy can work when the genes are represented as bits. It poses a problem when the genes are represented as real numbers. The main reason for this is that a bit has only two values, 0 or 1. A real number has infinitely many values. Thus, if a gene can have real values limited by two arbitrary boundaries, it can not explore values other than those given by the parents. That means the individuals will not have the opportunity to explore the space between the parents. A bit can only have one of two values. For each copy there is a chance that a gene will change value, covering that gene's whole space with a single change. This will severely limit the real valued genes. In order to explore a larger range of values for genes, another reproductive strategy has to be defined. When choosing other strategies, it is important to keep in mind that the reproduction phase seeks to create children that can be related to their

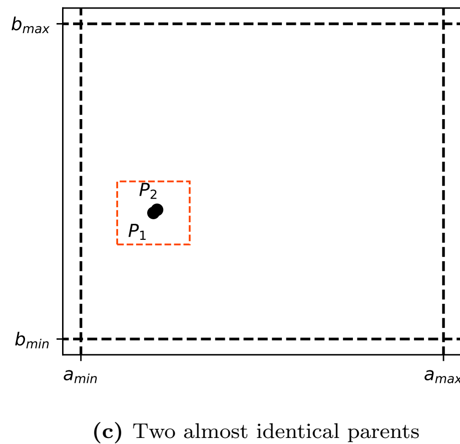
parents. Having the reproduction strategy as a new random value for each gene of the child is not a good strategy. That is the same as having a mutation rate of 100%. It must instead use and combine the parents' genes. A big part of evolutionary algorithms is that they test many different individuals and focus on where the results are good. Therefore, it has been decided to use a strategy that restricts the value of the genes to that in between those of the parents. Subsequently limiting the search space, while at the same time exploring other gene values. Imagine a problem with an objective function applied to individuals comprised of 2 genes, (a, b). Both a and b are defined as in equation 4.5. The possible space is given by the boundary conditions, which in this case happens to be the minimum and maximum values of a and b. Consequently, the solution space becomes a two-dimensional area. An illustration of the process can be seen in figure 4.16.

$$x \in [x_{min}, x_{max}], \qquad x_{min}, x_{max} \in \mathbb{R} \wedge x_{min} < x_{max} \tag{4.5}$$



(a) Two parents chosen for mating

(b) Two parents with closer gene values



(c) Two almost identical parents

Figure 4.16: Illustration of implemented reproduction

Figure 4.16a shows two parents with big differences on the genes. If these are chosen for reproduction, the space in which the child can appear is limited by the dashed red rectangle. This ensures the child to receive genes that are a result from mixing the parents' genes. As time goes on and the population evolves and the individuals become more fit, the individuals selected as parents may inevitably become similar. In figure 4.16b, two parents with similar genes are chosen for reproduction. The space in which the child can explore becomes narrower as time goes on. Thus, at some point, the strategy becomes inefficient. In figure 4.16c, two almost identical parents have been chosen for reproduction. If the normal procedure was to be used, the rectangle would be small and the child will inherit close to exact copies of the parents' genes. I.e. it would get stuck at a possible solution to the problem. However, this solution might be a local minimum or a point close to a local minimum. To avoid getting stuck at such a point, another layer will be added to the reproduction phase. This layer entails a comparison of the two parents before selecting a method. If the parents are different, the previously described method will be used and the the red rectangle shown in figure 4.16a and 4.16b illustrates the child space. If the parents are almost identical, another method is used. The new method selects a parent and uses its gene values as the intersection point between two diagonals which creates a new rectangle as shown in figure 4.16c. In order to create the new child space, a percentage reduction and increase is applied to the genes of the parent in order to establish new temporary boundaries for the genes of the child. By doing this, the local search space is not converging to an infinitely small rectangle, i.e. a single point. Yet, reproduction can be looked at as a local search. As generations pass and individuals converge, it will not be able to explore the whole space defined by the gene boundaries. It will converge the population towards individuals matching closely to the fittest individual. The fittest individual after several generations may be a local minimum or a point close to a local minimum. It is not guaranteed to be the best solution in the whole space. Therefore, an additional concept, called mutation, is added that ensures the possibility of searching outside the limits of reproduction.

Mutation of individuals

So far, parent selection and reproduction have been defined. These are integral parts of a GAs. Yet, another important aspect is the mutation operator. The reproduction phase will narrow the diversity of the population and performs only a local search close to the parents. Mutation is the operator that allows individuals to explore the whole solution space. Mutation may occur after a child is produced. Each gene of the child has a chance of mutating given by a mutation rate chosen by the user. When a gene mutates, the gene value is changed to a random value limited by the gene's bounds.

Selection of Survivors

After selecting parents and producing children, where some may have mutated, the population is larger than the initial population. To determine which individuals are selected as survivors, a selection method presented in [36] will be used. It proposes a non-dominated sorting genetic algorithm (NSGA). To get the new population, all individuals are compared in a non-dominated sorting algorithm, and placed in fronts based on how they dominate other individuals. The best individuals will be placed in the first front. As individuals in this implementation will have fitness values represented as floats, comparing individuals based on fitness value may yield only a single individual per front. To adjust for this, only the first few decimals of the fitness value is compared to evaluate domination. Hence, there is a possibility of having fronts with more than a single

individual. After allocating all the individuals in different fronts, another procedure takes place. This will insert the fronts into the new population. However, once the population reaches its population size, the last front that would push the number of individuals above the population size will be applied to a crowding distance operator. The overall procedure is shown in figure 4.17

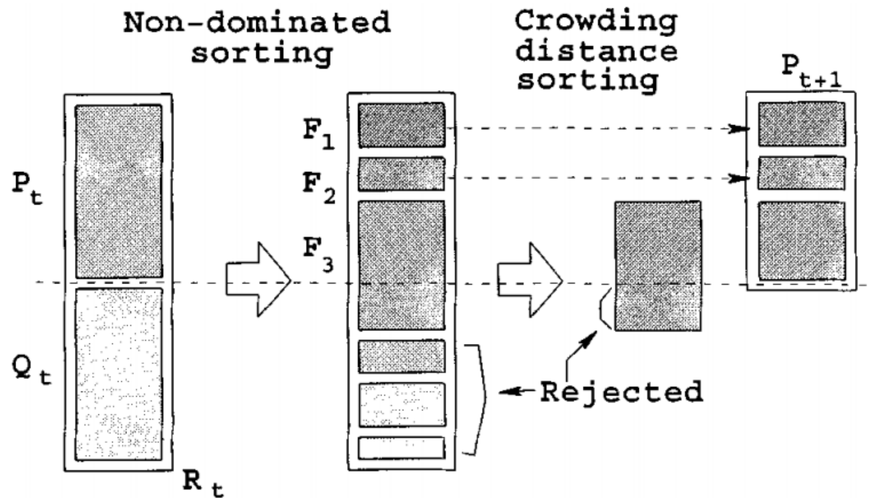


Figure 4.17: NSGA-II procedure, showing non-dominated sort and crowding distance operator [36]

P_t represents the previous population, Q_t is the new children, $F_1, F_2, F_3...$ represent the fronts created by the non-dominated sorting and P_{t+1} represents the new population, the next generation. Each front is placed directly into the new population until doing so would make the new population larger than the population size. The last front will then be applied to the crowding distance operator to determine which individuals survive to the next generation. From figure 4.17 this happens already at the third front. The crowding distance operator is applied and some individuals from the third front are rejected along the fronts with worse domination rank such as F_4, F_5 etc.

Crowding distance operator is a process targeting diversity preservation. Individuals in the front will have the same domination rank, however, they may be located at different points in the solution space. The objective is to select individuals that are located in a lesser crowded regions. To determine whether an individual is in a more crowded region, the genes of the individual are compared with all other individuals. The distance between two individuals is calculated as a normalized difference for each gene and summarized. Individuals with the largest distance to other individuals are selected to be a part of the new population.

Adding the NSGA selection method to the GA will introduce both elitism and diversity preservation. If the fittest individual still exists among the old population, it will be brought over to the new population. Thus, the best solution is preserved. At the same time, crowding distance is added for diversity preservation.

5 Results

In this section, results from running the algorithm to optimize the improved Cree model and the ROHM model will be presented. The models will be compared and discussed. Furthermore, several runs of the algorithm have been performed to evaluate the performance. These results will be discussed in a separate sub-section.

5.1 Fitting of IV-curves

Both models have been optimized with the GA using the same settings. The settings used is given in table 5.1.

Table 5.1: GA settings

Setting	Cree	ROHM
Population Size	100	100
Nr. Generations	100	100
Mutation Rate	10%	10%
Simulation Method	DC-sweep	DC-sweep
Nr. Parameters	6	20

Individuals in a population have different number of genes depending on which model is used in the algorithm. The improved Cree model has 6 parameters, while ROHM has 20. Results from optimizing both the improved Cree model and the ROHM model are shown in figure 5.1 and 5.2 respectively. The same CT measurements of a Cree C3M0075120D SiC MOSFET are used for both models.

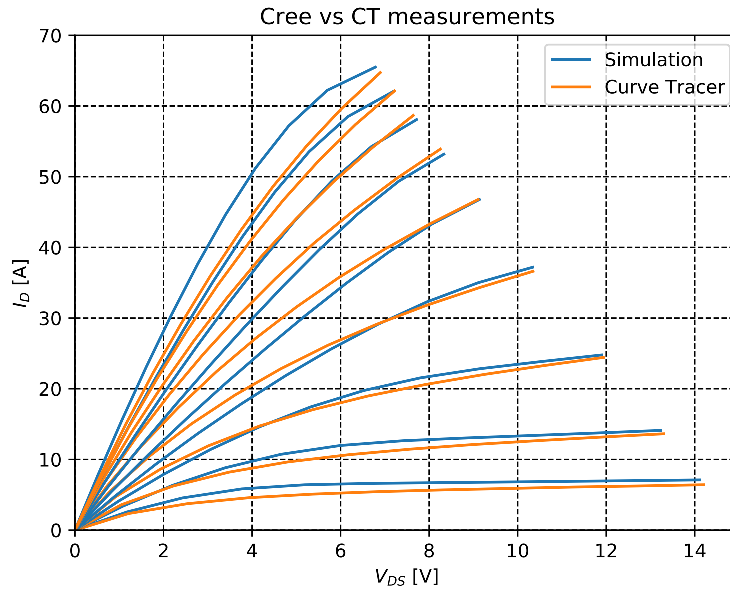


Figure 5.1: Fittest individual of last generation, improved Cree model

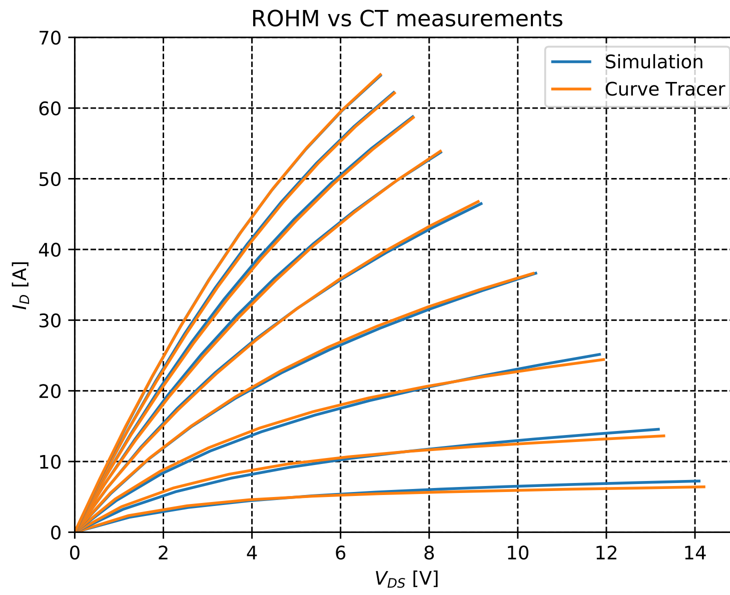


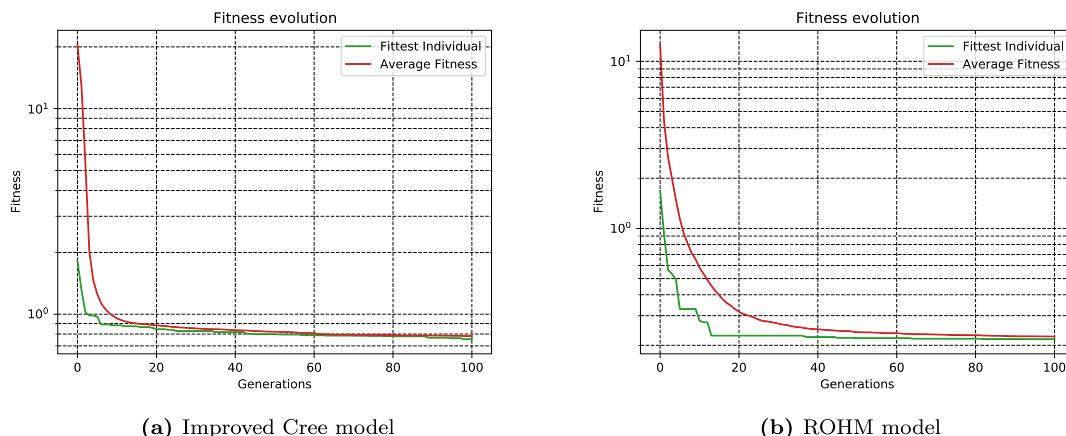
Figure 5.2: Fittest individual of last generation, ROHM model

The orange lines are the CT measurements, and the blue lines are results from simulation of the fittest individual in the last generation. The improved Cree model achieved a fitness value of approximately 0.78, while the ROHM model achieved a fitness value of approximately 0.22. As the fitness value is calculated as the average vector length between measurement and simulation points, 0 fitness value corresponds to perfect match of all points. Thus, the GA is able to adjust the parameters of the models to obtain a match. Both models are clearly following the same trend as the CT measurements, however, there are large errors at certain points for the improved Cree model. It seems that the model enters saturation abruptly compared to the measurements, especially for the high gate voltages. Adjusting the saturation modeling of the improved Cree model may yield better results. The ROHM model obtains a very good match for the high gate voltages (V_{GS}=10-15V), while a slight error occurs on the last part of the low gate voltages (V_{GS}=7-9V). Compared to the improved Cree model, ROHM has 20 parameters that are changed. This may allow for better adjustments of the fitting, as changing a single parameter in the ROHM model might have less impact on the curves.

Based on these results, both models are able to reproduce the $I_D - V_{DS}$-characteristics. However, with different accuracy. Overall, the ROHM model obtains best fit of the two models when optimizing with the GA. This can indicate that the ROHM model is more versatile than the improved Cree model. However, additional tests and analyzes must be performed to make that conclusion. The next section will evaluate the algorithm and explore the evolution of the models during the optimization process.

5.2 Evaluation of the Algorithm

The initial generation of the GA will contain individuals with completely random genes, only limited by the parameter bounds. This will create many individuals with a bad fitness value. During the parent selection step, these individuals have less chance of being selected as parents compared to the best individuals. The fitness evolution of the population for both the improved Cree model and the ROHM model is shown in figure 5.3.



(a) Improved Cree model

(b) ROHM model

Figure 5.3: Fitness evolution

The green line shows the evolution of the fittest individual and the red line shows the average fitness of the population. Both models have the same trend. The initial population has a very high average fitness value, but over generations, the individuals produce children with better genes that replace the majority of the bad individuals in the previous generation. Both models have very fast improvements during the first generations. Then, slows down after 10-20 generations for both models. The average fitness quickly drops to a fitness value approximately equal to the fittest individual. It is slower for the ROHM model. This might be a result of the number of genes each individual have, i.e. the number of changeable parameters in the model. When the model has many parameters, it might take more time to find good genes. After 100 generations it seems that both models have converged since the average fitness value is approximately equal to the fittest individual. The improvements from running the algorithm further will be tiny. To investigate convergence of the models, parameters from the individuals of the last generation can be plotted. The GA is run several times using different seeds. All other settings are as in table 5.1. The final generations for the improved Cree model and the ROHM model are shown in figure 5.4 and 5.5 respectively.

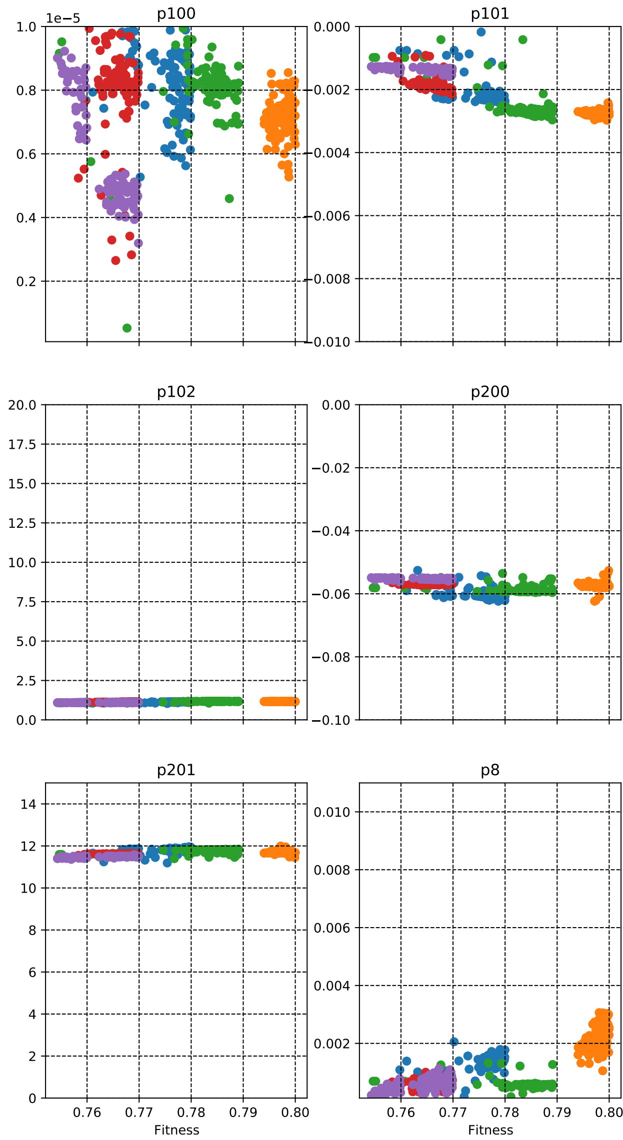


Figure 5.4: Parameter values of all individuals in last generation, improved Cree model

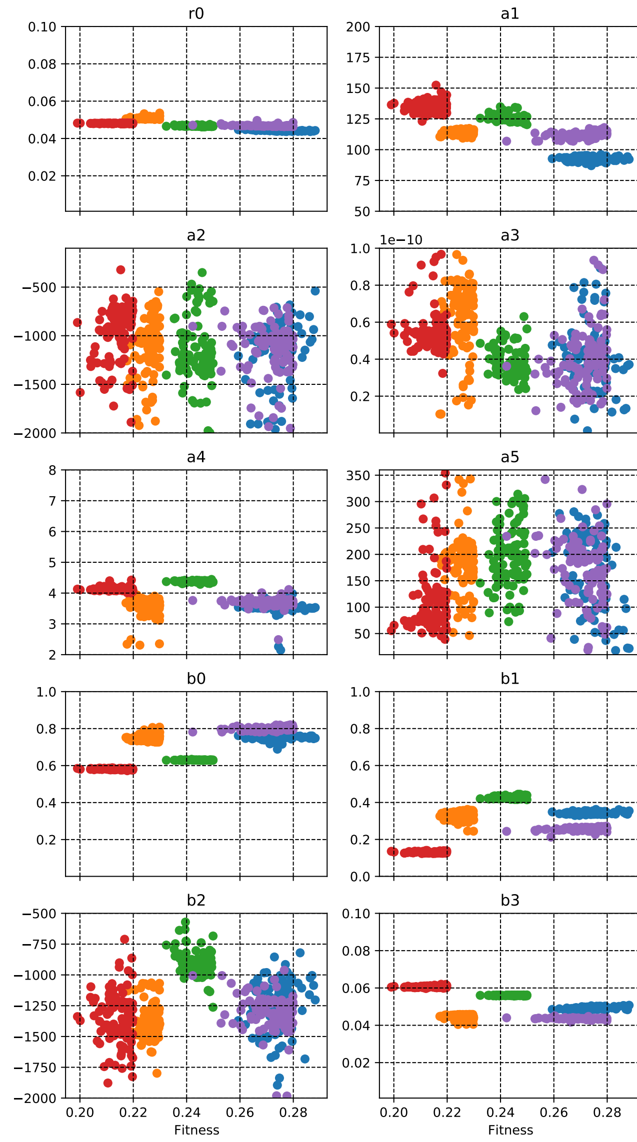


Figure 5.5: Parameter values of all individuals in last generation, ROHM model

The improved Cree model has 6 parameters that are changed in the optimization. Figure 5.4 shows the last generation from five runs using a new seed each run. Each color represents one of the runs. Some parameters, p_{102}, p_{200} and p_{201} converge to a small interval. The values found for these parameters are almost the same after each run. They might have a strong impact on the shape of the curves and quickly gets targeted during the reproduction phase. The rest of the parameters have more of a spread. For example, p_{100} is almost covering the whole interval set by the bounds. This can indicate algorithm didn't run long enough for that parameter to converge, it has low impact on the shape of the curves which will translate to low impact on the fitness value, or the bounds set are too narrow. However, the fitness values of all the individuals range from 0.75 to 0.80. That is a small difference and barely noticeable on the curves.

The ROHM model has 20 parameters that are changed in the optimization. Figure 5.5 shows 10 of the parameters of the last generation from five runs using a new seed each run. Each color represents one of the runs. In this case, the parameters are even more spread compared to the improved Cree model. Of the parameters shown, only r_0 converges to approximately the same value throughout the runs. a_1, a_4, b_0, b_1 and b_3 seems to converge for each run, but at different values between the runs. This can indicate that more than one optimal value for that parameter exists or there are strong correlation between certain parameters. These parameters may have a large impact on the shape of the curves, which impacts the fitness values. Therefore, the spread is small. a_2, a_3, a_5 and b_2 have a much larger spread and cover most of the interval. This can indicate algorithm didn't run long enough for that parameter to converge, it has low impact on the shape of the curves, or the bounds set are too narrow. Moreover, the fitness value ranges from 0.20 to 0.30, which is a larger spread compared to the improved Cree model.

In summary, the ROHM model has more spread compared to the improved Cree model, even though the ROHM model achieved better fitness value in all runs. Each run used equal settings with a new seed each time. Therefore, it is not expected to obtain the same convergence each run. For that to happen, in the same amount of generations, the sequence of random numbers would have to be almost equal. A few conclusions can be drawn by these results; 100 generations are not enough to make all parameters converge and despite this, the algorithm is able to achieve relatively close fitness values between the runs. The algorithm was also tested with four different mutation rates for five different seeds. These results are shown in figure 5.6 and 5.7.

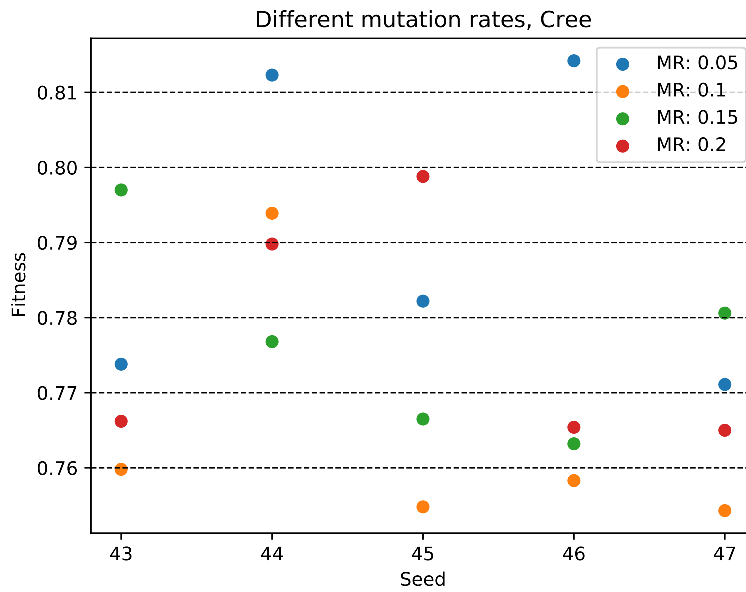


Figure 5.6: 20 runs of Cree model, different mutation rates for various seeds

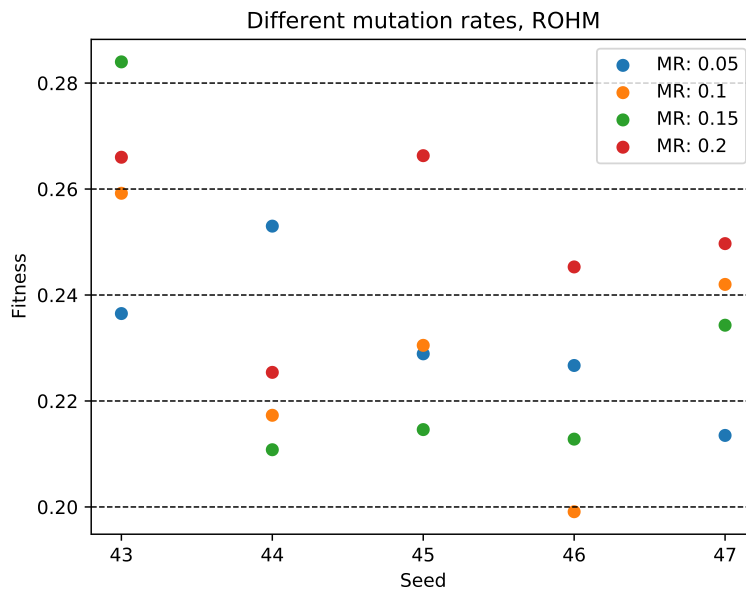


Figure 5.7: 20 runs of ROHM model, different mutation rates for various seeds

Figure 5.6 shows the results of running the GA on the improved Cree model with different mutation rates. The results varies between a fitness value of approximately 0.81 and 0.75. For the ROHM model in figure 5.7, the results varies between a fitness value of approximately 0.28 and 0.20. For the Cree model, 10% mutation rate seems to perform the best based on the results from the chosen seeds. However, for the ROHM model there is no indication of a best mutation rate. Higher mutation rates means that a gene has higher chances of mutating, i.e. assigned a random value between its bound rather than a value determined by the reproduction step. A high mutation rate will increase the chance of more than one gene mutating, especially for models with many parameters such as the ROHM model. The results are inconclusive when considering mutation rates and why the algorithm converges to solutions with different fitness values. Despite this, the overall spread of the fitness values are not large when compared to the average fitness value of the first generations of a run, as seen in figure 5.3. For both the improved Cree model and the ROHM model, the first few generations have an average fitness value much greater than the fittest individual. Additionally, the stop criteria was 100 generations, meaning the algorithm runs until it has evolved through 100 generations. Like for the parameter values in figure 5.4 and 5.5, using a number of generations as stop criteria will most likely not produce equal results, unless the number of generations are set to infinite. Nevertheless, the overall results indicates that the ROHM model is able to fit the measurement data better compared to the improved Cree model.

6 Conclusions

The main goal of this thesis is to develop an algorithm that can optimize MOSFET models to measurement data. Several papers have shown different ways of modeling and how to fit to data, some including thermal considerations. However, only case temperature is considered when fitting measurements to models including thermal modeling. From the CT measurements in section 4.1 it is shown that for different pulse width for either current or voltage pulse, depending on method, the measurements are not the same. Large discrepancies are seen when the power consumed by the MOSFET is high. Through simulations it was shown that the junction temperature changes drastically depending on which measurement point is investigated. This was also confirmed by thermal calculations using the thermal impedance given by the datasheet. For many measurement points, junction temperature increases significantly and has an impact on the $I_D V_{DS}$-characteristics. By optimizing models taking the junction temperature changes into account, the optimized model is more accurate of a real model when representing the $I_D V_{DS}$-characteristics.

A genetic algorithm has been implemented and used to optimize the improved Cree model and the ROHM model based on measurements taken with a curve tracer. Changes on junction temperature occurring during the measurements have been investigated and accounted for in the optimization. Results show that the ROHM model obtains a better fit compared to the improved Cree model. It has 20 parameters that can be changed and may as a consequence allow for better fit, since changing only one parameter can have less impact compared to changing one parameter in the improved Cree model, which has only 6 parameters. Running the algorithm 20 times using different mutation rates and seeds showed that ROHM, in all cases, obtained a better fit. However, each run did not converge to exactly the same solution. This can be explained by the stop criteria used, as each run ended due to a limit of 100 generations. When using various seeds, different sequences of random numbers are used in many functions defining the selection, reproduction and mutation steps. Thus, it is not given that different runs running for 100 generations are bound to achieve the same solution.

The use of a genetic algorithm works to obtain a match between measurement data and the model. This opens the opportunity to test many different implementations of models without having prior knowledge about how the model work. Models can be optimized and compared to evaluate what models work well. Consequently, these models can be investigated further to understand why they perform well or what they might be missing in order to perform well. The same procedure and principles can be used on different parts of models. I.e. the GA with its implementation is not necessarily limited to only fitting $I_D V_{DS}$-characteristics. It may also be adjusted, along with the simulation configuration, to optimize different characteristics, such as CV-characteristics, dynamic switching behavior, reverse recovery of intrinsic diode and more. Ultimately, by optimizing a model for many different aspects based on measurement data, an accurate and universal model may be found. This can help obtaining more accurate SiC MOSFET models to better explore the advantages of WBG materials in different applications.

6.1 Further Work

This thesis scratches only the surface of the possibilities of optimizing SiC MOSFET models using a GA. Only one manufacturer device was measured using the curve tracer and subsequently used as measurement data in the GA. To further strengthen the possibility of models being candidates for

universal models, more devices from different manufacturers must be measured and used as measurement data in the GA. Additionally, optimization could be expanded to include measurements at higher case temperatures. There are also other aspects of SiC MOSFET modeling that can be explored such as CV-characteristics, dynamic switching event and reverse recovery. Further work based on this thesis may go in several directions, two of which will be presented.

The first path is to explore the many other aspect of SiC MOSFET modeling. By using the designed GA or another implementation of an optimization algorithm to optimize different parts of the model based on measurement data. The following list contains suggestions to what aspects can be optimized.

- Perform IV-measurements at higher temperatures, e.g. 25^oC, 75^oC and 125^oC, and optimize the model based on the measurements.
- CV-characteristics can be measured by the curve tracer and used in the same way as the IV-characteristics for optimizing the model.
- Measurements of the dynamic switching event can be performed through double pulse tests and used to adjust the internal resistances and inductances of the component.
- Measure the reverse recovery waveforms of the intrinsic body diode and use the measurements to optimize the reverse recovery part of the model.
- Measure other manufacturer devices and use these as input to the GA. Then, compare how the models are able to predict the behavior of different devices.

The GA and the simulation must be changed in order to accommodate for all these optimizations. However, the underlying principles will remain the same. If the GA is expanded to take into consideration more aspects of SiC MOSFET modeling, the goal of finding an universal model may be closer.

The second path is to investigate and evaluate the performance of the GA. It may be that evolutionary algorithms such as ACO or PSO algorithms are better suited to find a good solution faster than the resulting GA. Additionally, the GA uses only a single objective function to obtain a good solution. There might be a possibility of exploring multi-objective optimization. The following list presents possible tasks for pursuing this path.

- Redefine the current objective function to a multi-objective problem.
- Compare GA using single objective and multi-objective function.
- Investigate and compare gene values of individuals in population.
  - Single objective vs multi-objective.
  - How does the parameter values change during the optimization?
  - Parameter correlation analysis to determine why the spread on certain parameters occur. Low impact on output vs many possible sets of solutions due to strong correlation.
- Optimize the GA based on the previous analyzes
- Compare optimization using GA vs other types of EAs such as ACO or PSO.

Appendices

A IV-characteristics Simulation Circuit

Simulation circuit used to obtain the IV-characteristics. After running the simulation, Ctrl+L is used to open the log file containing the results of the measurement commands. Right-click the variable name of the measured variable and select plot data. Right-click on the plot, hover over "File" and select "Export data as TEXT". This will create a .log file containing all the measurements in .CSV format. The .log files were imported to Matlab to obtain better looking graphs. The MOSFET model must be changed when simulating models from other manufacturers.

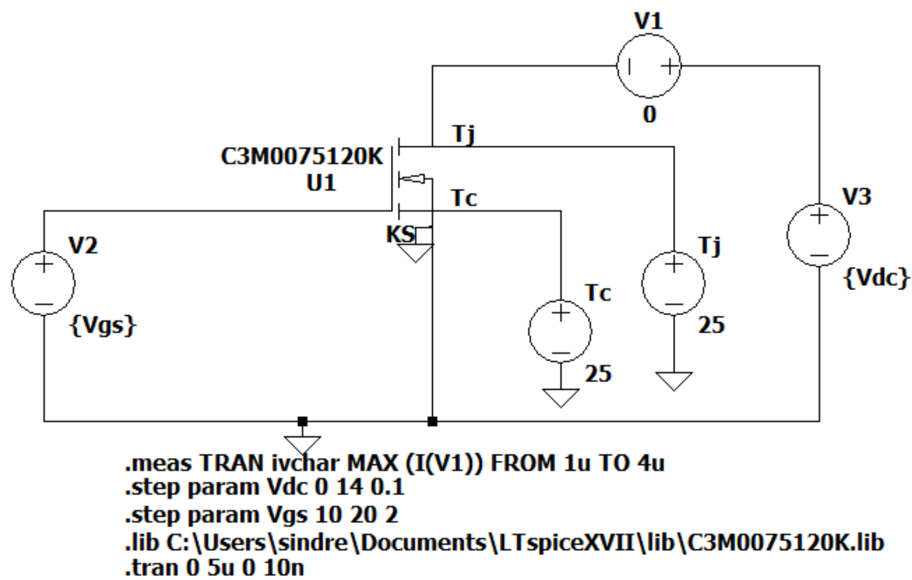


Figure A.1: IV-characteristic simulation circuit

B CV-characteristics Simulation Circuits

Simulation circuits used to obtain the CV-characteristics of the manufacturer models. Same procedure is used as in appendix A to obtain the .log file. In this case, the current is measured and must be manipulated after simulation to obtain the correct value for the capacitors. Hence, after plotting in LTspice, Right-click the variable name and add the following expression to obtain the correct values for the capacitances:

$$\frac{"VarNameCurrentMeas"}{2 \cdot \pi \cdot f_{AC} \cdot \left(\frac{A_{V_{AC}}}{\sqrt{2}}\right)}$$

where "VarNameCurrentMeas" is the current measurement variable name, f_{AC} the frequency of the applied AC voltage and $A_{V_{AC}}$ the amplitude of the AC voltage signal.

B.1 Ciss Simulation Circuit

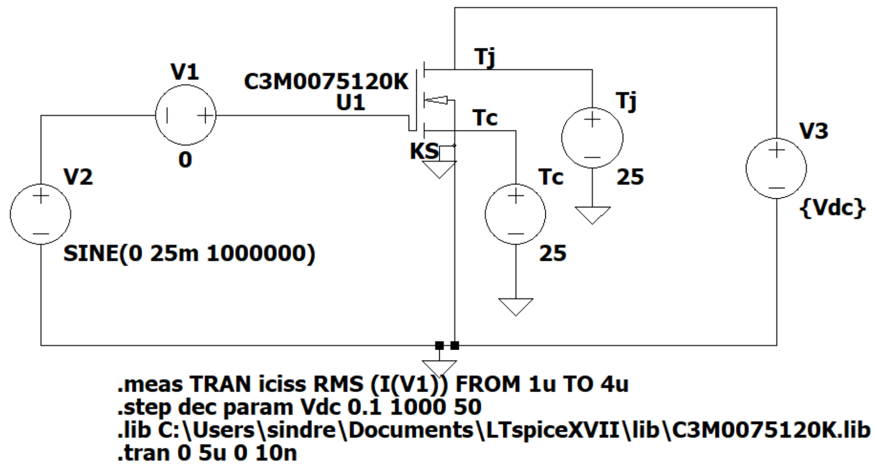


Figure B.1: Ciss simulation circuit

B.2 Coss Simulation Circuit

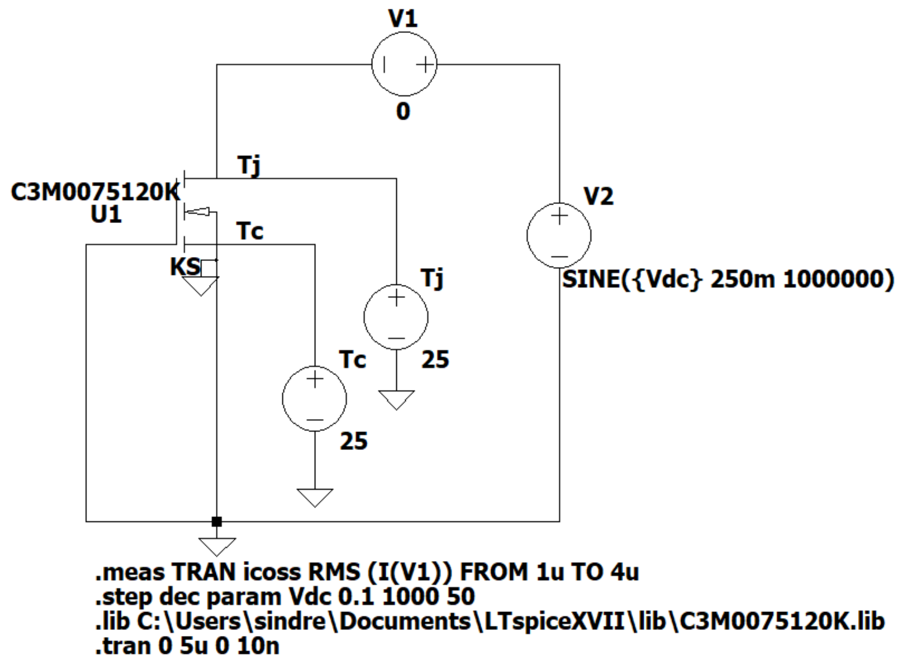


Figure B.2: Coss simulation circuit

B.3 Crss Simulation Circuit

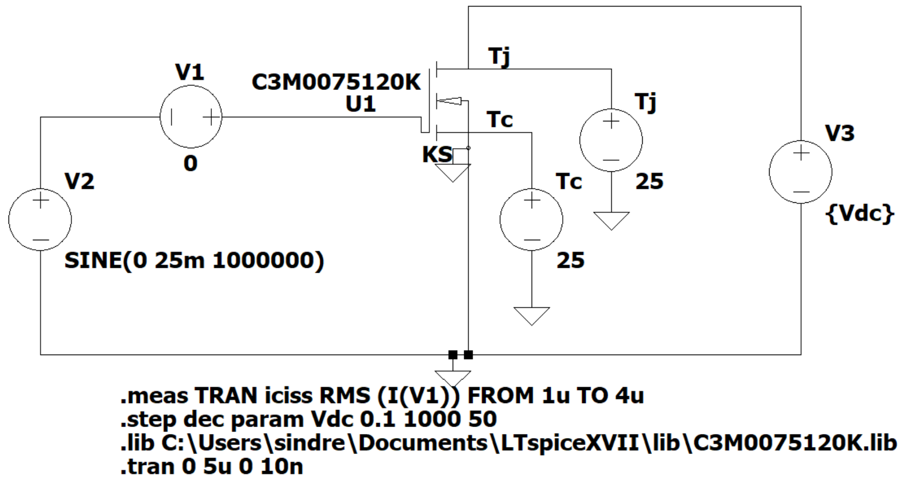


Figure B.3: Crss simulation circuit

C Junction Temperature Simulation Circuit

Simulation circuit used to obtain ΔT between junction and case during the measurements done by the curve tracer. Same settings have been used as in the curve tracer. The measurement delay has been chosen such that the last $10\mu s$ of the pulse width is when the measurement is taken. This is to mimic the "AUTO" setting on the Curve Tracer. Python has been used to implement scripts that run and analyze results from LTSpice simulations through a package called PySpicer [37].

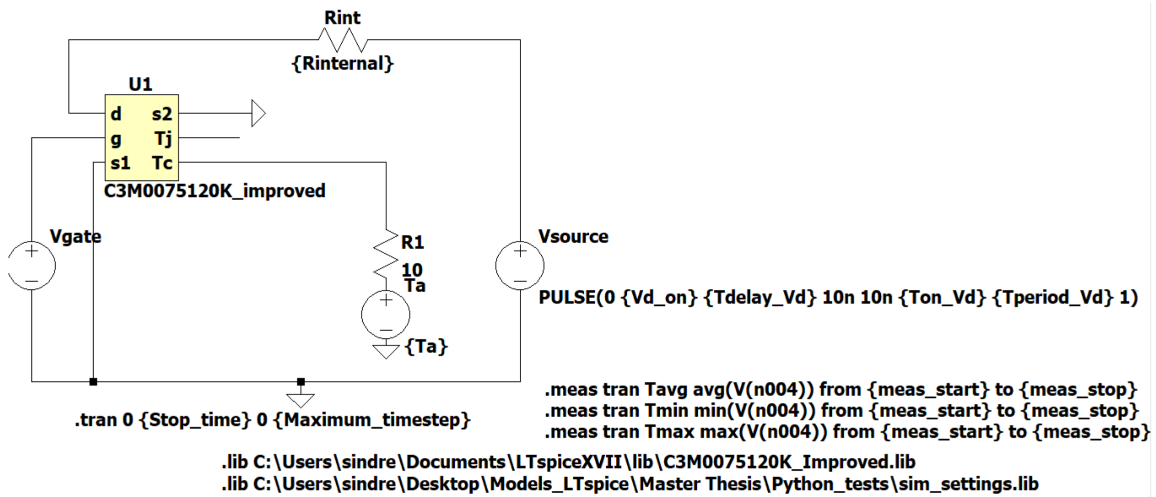


Figure C.1: Simulation circuit used to simulate junction and case temperature

D Verification of Temperature Simulations

The manufacturer-provided model was used to produce the junction temperature at all measurement steps. In order to remove potential error sources, the junction temperatures from simulations are compared with manual calculations using the thermal impedance and power consumption of the MOSFET. Integrating the IV-characteristic curves will yield the power curve of the MOSFET as shown in figure D.1.

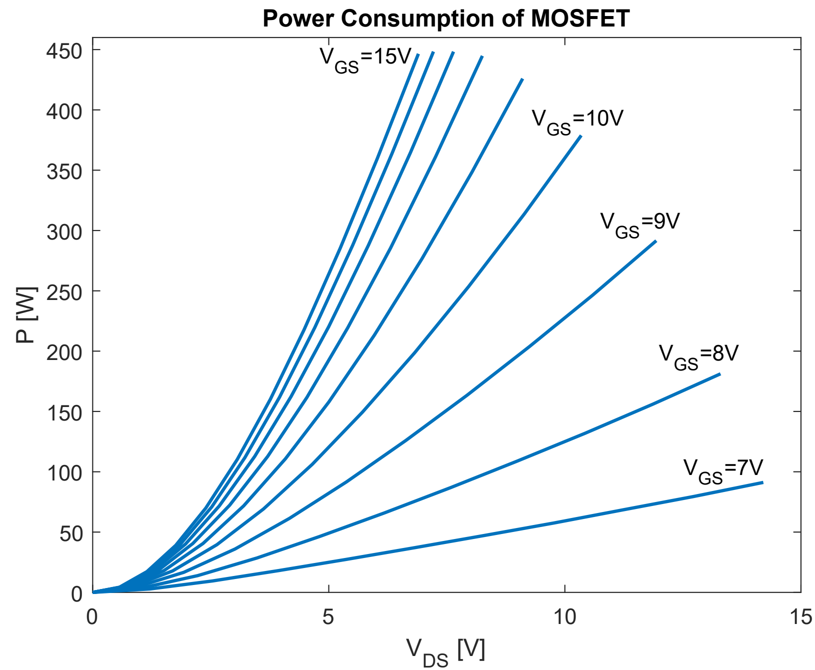


Figure D.1: Power curve, IV-curves integrated by V_{DS}

Multiplying by the correct thermal impedance gives the temperature difference that occurs between junction and case. As the thermal impedance varies with pulse width and duty cycle, the correct thermal impedance must be found from the component's datasheet, using the transient thermal impedance (junction - case) figure. The results are shown in figure D.2.

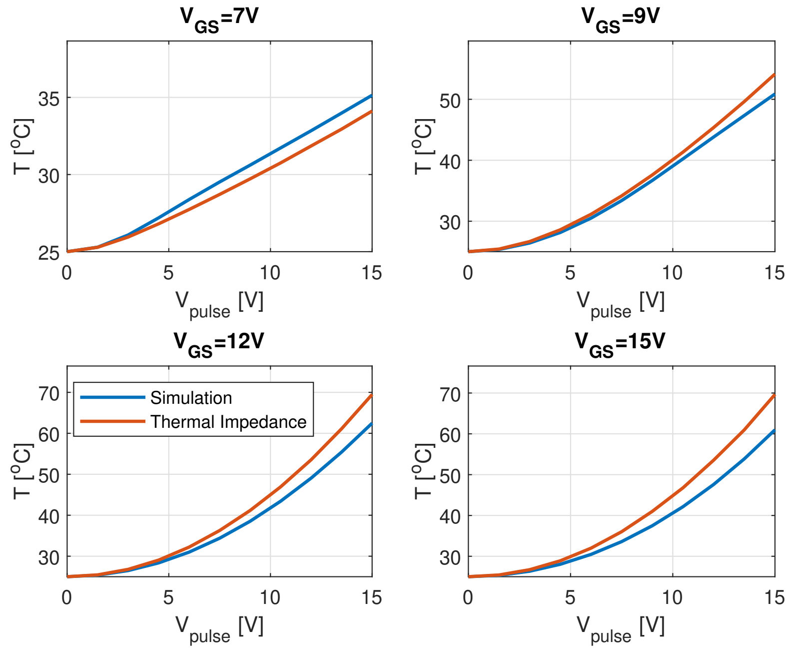


Figure D.2: Verification of temperature simulation

The simulations matches the calculations using thermal impedance. For high gate voltages, the thermal impedance calculations show even higher temperatures compared to the simulations. The reason for the difference could be due to higher power than in reality. By integrating the IV-curves, only the power at the measurement is captured. To obtain greater accuracy, V_{DS} and I_D curves during the measurement should've been multiplied together to obtain the power curve during the pulse. Then the total energy consumed could've been obtained by integrating the power curve. However, for the sake of confirming changes of junction temperatures during measurement steps, the method should suffice. This means that the simulations can be used to show the temperature differences that appear during the measurements. It also confirms a source of error of the CT measurements, namely higher junction temperature for several measurement points although the case temperature stays the same. In conclusion, the measurements are said to be taken at 25^oC while in actuality, the junction temperature may be far greater than 25^oC during the measurement.

E Setting Junction Temperature LTSpice

Achieving the temperature changes at each point can be done using the LTSpice function command along a Behavioral Voltage Source (BVS). The BVS will produce a voltage signal given by a function defined in a simulation settings file, which is created before simulation. A python function takes in the temperature data, which is calculated using the method described in appendix D and creates an LTSpice function command. The LTSpice function uses two variables to index an LTSpice table with the temperature data. To obtain the correct temperature for a given step, the BVS uses the voltage pulse and the gate voltage values as variables in the function. The indexing of the table will be as in equation E.1.

$$index(V_G, V_{pulse}) = (V_G - V_{Gmin}) \cdot X_{pulse} + \frac{V_{pulse}}{V_{step}} \tag{E.1}$$

where V_{Gmin} is the lowest gate voltage used in the measurements, X_{pulse} is the number of applied voltage steps, and V_{step} is the step value of the applied voltage. These values are set in the CT settings. Furthermore, an LTSpice table has syntax as in equation E.2.

$$\text{Table}(\{index(V_G, V_{pulse})\},\ \mathbf{0}, n_0, \mathbf{1}, n_1, ..., \mathbf{i}, n_i) \tag{E.2}$$

where $index(V_G, V_{pulse})$ is the expression that determines the index value, the bold numbers $\mathbf{0}, \mathbf{1}, ..., \mathbf{i}$ represents the indices of the table, and $n_0, n_1, ..., n_i$ represents the values located at the indices in the table. If the expression is a value between two indices in the table, LTSpice will use linear interpolation between the closest indices to find a value. In summary the implementation of the LTSpice function will be as in equation E.3.

$$\text{.func tempcorrection}(V_G, V_{pulse}) \quad \text{Table } (\{index(V_G, V_{pulse})\}, \mathbf{0}, T_0, \mathbf{1}, T_1, ..., \mathbf{i}, T_i) \tag{E.3}$$

A python function is made to automatically produce this function and adding it to the simulation settings file. It requires temperature data and test settings for gate voltage and pulse voltages as two separate vectors. E.g. the settings used for measurements in the thesis are $V_{gate} = [7, 8, 9, 10, 11, 12, 13, 14, 15]$ and $V_{pulse} = [0, 1.5, 3, 4.5, 6, 7.5, 9, 10.5, 12, 13.5, 15]$. That means the table will have a total of $9 \cdot 11 = 99$ indices and elements. The function can also be used in a BVS for transient analysis to ensure the same temperature occurring during the measurement.

References

[1] S. B. Kongerød, “Advanced gate drivers for next generation high power converters,” *Norwegian University of Science and Technology (NTNU)*, 2019.

[2] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronic Converters, Applications and Design*, 3rd ed. JOHN WILEY & SONS, INC., 2003.

[3] D. Peftitsis, *Lecture Notes ELK-20*. NTNU Trondheim, 2019.

[4] Wolfspeed, “Silicon carbide power mosfet c3m^{tm} mosfet technology,” Aug 2018, accessed 11.13.2019. [Online]. Available: https://www.wolfspeed.com/downloads/dl/file/id/176/product/14/c3m0065090d.pdf

[5] J. Rabkowski, D. Peftitsis, and H. Nee, “Silicon carbide power transistors: A new era in power electronics is initiated,” *IEEE Industrial Electronics Magazine*, vol. 6, no. 2, pp. 17–26, June 2012.

[6] M. Rosina, “Gan and sic power device: market overview,” Nov 2018, accessed 12.12.2019. [Online]. Available: http://www1.semi.org/eu/sites/semi.org/files/events/presentations/02_Milan%20Rosina_Yole.pdf

[7] ROHM, “Sic power devices and modules application note,” Aug 2014, accessed 12.12.2019. [Online]. Available: http://rohmfs.rohm.com/en/products/databook/applinote/discrete/sic/common/sic_appli-e.pdf

[8] M. Östling, R. Ghandi, and C. Zetterling, “Sic power devices — present status, applications and future perspective,” in *2011 IEEE 23rd International Symposium on Power Semiconductor Devices and ICs*, May 2011, pp. 10–15.

[9] X. She, A. Q. Huang, O. Lucía, and B. Ozpineci, “Review of silicon carbide power devices and their applications,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8193–8205, Oct 2017.

[10] H. A. Mantooth, K. Peng, E. Santi, and J. L. Hudgins, “Modeling of wide bandgap power semiconductor devices—part i,” *IEEE Transactions on Electron Devices*, vol. 62, no. 2, pp. 423–433, Feb 2015.

[11] N. Paydavosi, T. H. Morshed, D. D. LU, W. Yang, M. V. Dunga, X. Xi, J. He, W. Liu, M. Cao, X. Jin, J. J. Ou, M. Chan, A. M. Niknejad, and C. Hu, “Bsim4v4.8.0 mosfet model,” 2013, accessed 24.02.2020. [Online]. Available: http://ngspice.sourceforge.net/external-documents/models/BSIM480_Manual.pdf

[12] C. C. Enz, F. Krummenacher, and E. A. Vittoz, “An analytical mos transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications,” *Analog Integrated Circuits and Signal Processing*, vol. 8, no. 1, pp. 83–114, Jul 1995. [Online]. Available: https://doi.org/10.1007/BF01239381

[13] M. Bucher, C. Lallement, C. Enz, F. Théodoloz, and F. Krummenacher, “The epfl-ekv mosfet model equations for simulation,” *Version 2.6*, 1998.

[14] W. R. Curtice and M. Ettenberg, "A nonlinear gaas fet model for use in the design of output circuits for power amplifiers," in *1985 IEEE MTT-S International Microwave Symposium Digest*, June 1985, pp. 405–408.

[15] S. Maas, "Fixing the curtice fet model," in *Microwave Journal*, March 2002.

[16] R. Zhou, M. Shintani, M. Hiromoto, and T. Sato, "A charge-based sic power mosfet model considering on-state resistance," in *Proceedings of International Symposium on Nonlinear Theory and Its Applications*, 2016.

[17] N. Phankong, T. Funaki, and T. Hikihara, "Characterization of the gate-voltage dependency of input capacitance in a sic mosfet," *IEICE Electronics Express*, vol. 7, no. 7, pp. 480–486, 2010.

[18] H. Li, X. Zhao, R. Hao, and K. Sun, "A non-segmented pspice model of sic mosfets," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2017, pp. 4823–4828.

[19] A. AlHoussein, H. Alawieh, Z. Riah, and Y. Azzouz, "A new modeling approach for predicting the static and dynamic behavior of sic power mosfets," in *2018 International Symposium on Electromagnetic Compatibility (EMC EUROPE)*, Aug 2018, pp. 648–653.

[20] D. Lena, I. Buraioli, A. Bocca, D. Demarchi, and A. Macii, "An accurate electro-thermal model of sic power mosfets for fast simulations," in *2018 IEEE International Conference on Industrial Technology (ICIT)*, Feb 2018, pp. 623–628.

[21] X. Zhao, H. Li, Y. Wang, Z. Zhou, K. Sun, and Z. Zhao, "A temperature-dependent pspice short-circuit model of sic mosfet," in *2019 IEEE Workshop on Wide Bandgap Power Devices and Applications in Asia (WiPDA Asia)*, May 2019, pp. 1–5.

[22] A. Stefanskyi, L. Starzak, and A. Napieralski, "Universal behavioural model for sic power mosfets under forward bias," in *2018 25th International Conference "Mixed Design of Integrated Circuits and System" (MIXDES)*, June 2018, pp. 343–348.

[23] H. Hove, O. Spro, G. Guidi, and D. Peftitsis, "Improved sic mosfet spice model to avoid convergence errors," Oct. 2019.

[24] A. Rohatgi, "Webplotdigitizer," 2019, accessed 04.02.2020. [Online]. Available: https://apps.automeris.io/wpd/

[25] ST Microelectronics, "Um1575 user manual, spice model tutorial for power mosfets," Nov 2013, accessed 31.01.2020. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/user_manual/04/4d/16/0d/d9/49/46/29/DM00064632.pdf/files/DM00064632.pdf/jcr:content/translations/en.DM00064632.pdf

[26] KEYSIGHT Technologies, "B1505ap pre-configured power device analyzer / curve tracer," 2020, accessed 25.02.2020. [Online]. Available: https://www.keysight.com/en/pd-2111407-pn-B1505AP/pre-configured-power-device-analyzer-curve-tracer-b1505a-with-modules-and-fixture?nid=-32851.1013306&cc=NO&lc=eng

[27] Cree Wolfspeed, "C3m0075120d - silicon carbide power mosfet," accessed 06.05.2020. [Online]. Available: https://www.wolfspeed.com/media/downloads/1450/C3M0075120D.pdf

[28] D. Câmara, "1 - evolution and evolutionary algorithms," in *Bio-inspired Networking*, D. Câmara, Ed. Elsevier, 2015, pp. 1 – 30. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781785480218500016

[29] C. Darwin, *On the Origin of Species, or the Preservation of Favoured Races in the Struggles for Life.* John Murray, London, 1859.

[30] G. Mendel, *Versuche über Pflanzen-Hybriden.* Naturforschender Verein in Brünn, 1866.

[31] K. De Jong, D. Fogel, and H.-P. Schwefel, *A history of evolutionary computation.* Handbook of Evolutionary Computation, IOP Publishing Ltd, 01 1997, pp. A2.3:1–12.

[32] M. Mitchell, *An introduction to genetic algorithms.* MIT press, 1998.

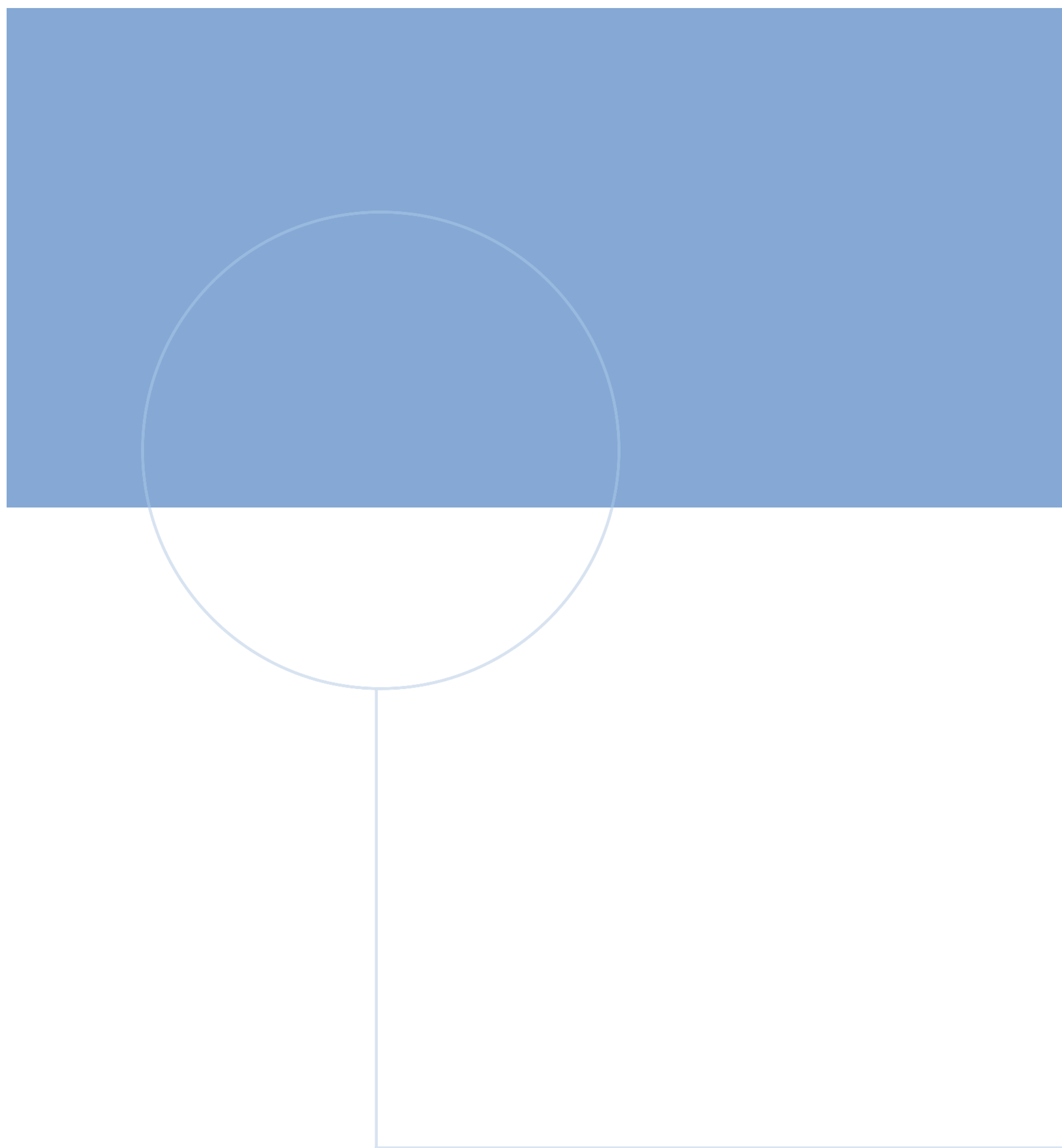
[33] J. H. Holland, *Adaptation in Natural and Artificial Systems*, 1975, vol. University of Michigan Press. (Second edition: MIT Press, 1992).

[34] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, no. 22, pp. 387–408, Jan 2018.

[35] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[37] N. Brum, "Ltspice python tools," accessed 20.03.2020. [Online]. Available: https://www.nunobrum.com/pyspicer.html



NTNU

Kunnskap for en bedre verden