

Universität Stuttgart



INSTITUT FÜR GROSSFLÄCHIGE MIKROELEKTRONIK

PROF. DR.-ING. N. FRÜHAUF

Automated Extraction of TFT Parameters for Simulation

Research thesis

21. July 2023

Autor : Arpan Biswas

Supervisor : Dr.-Ing. Patrick Schalberger

Co-supervisor : M. Sc. Florian Kleber

Kurzfassung

Diese Arbeit zielt auf die Entwicklung eines automatisierten Werkzeugs zur Extraktion von Simulationsparametern von Dünnschichttransistoren (TFT) für SPICE-Simulationen ab. In der großflächigen Mikroelektronik sind TFT die grundlegenden Komponenten in mikroelektronischen Geräten, zu denen auch Displays und Sensor-Arrays gehören. Je komplexer die Schaltkreise werden, desto genauere Simulationen sind vor der Herstellung der Geräte erforderlich. Ein automatisiertes Tool hilft bei der Ermittlung der Parameter für die SPICE-Simulation und ermöglicht es, das Verhalten des Bauelements anhand der Simulationsergebnisse vor der Fertigung zu bestimmen.

Simulation-Frameworks wie SPICE werden für Schaltungs- oder TFT-Simulationen verwendet, die eine manuelle Extraktion der TFT-Modellparameter erfordern. Das Hauptziel dieser Arbeit ist es, ein Werkzeug zu entwickeln, das die Extraktion von Parametern aus gemessenen TFT-Kennlinien automatisiert und damit den Simulationsprozess automatisiert.

Die Funktionalität des automatisierten Tools wird erweitert, um eine vergleichende Analyse zwischen gemessenen und simulierten Graphen aus den Eingangs- und Ausgangskennlinien des TFT zu ermöglichen, wobei ein iterativer Optimierungsalgorithmus verwendet wird.

Abstract

This thesis aims at the development of an automated tool for extraction of simulation parameters from Thin Film Transistors (TFT) for SPICE simulation. In large area microelectronics, TFTs are the foundational components in microelectronic devices which includes displays, sensor arrays. The more complex the circuitry becomes, more accurate simulations are required before fabrication of the devices. An automated tool helps to get parameters for the SPICE simulation and helps to know the device behaviour through simulation results before fabrication.

Simulation framework like SPICE are used for circuit or TFT simulations, which requires manual extraction of TFT model parameters. The central aim of this thesis is to design a tool that automates the extraction of parameters from measured TFT characteristic curves, thus automating the simulation process.

The functionality of the tool is expanded to provide comparative analysis between measured and simulated graphs from the TFT input and output characteristic curves, using an iterative optimization algorithm.

Contents

Kurzfassung	ii
Abstract	iii
Abkürzungen	vi
1. Introduction	1
1.1. Background	1
1.2. Objectives and Scope of the Thesis	1
1.3. Structure of the work	2
2. Theory	3
2.1. Transistor	3
2.1.1. MOSFET	3
2.1.2. N-channel enhancement type MOSFET and depletion type MOS-FET	4
2.1.3. Thin Film Transistors (TFTs)	5
2.1.4. Indium Gallium Zinc Oxide (IGZO) TFT	6
2.1.5. Architecture	8
2.1.6. Electrical Characteristics	10
2.1.7. Device Modelling and Importance of Parameter Extraction in TFT Models	13
3. Fabrication and process flow of IGZO TFT	15
3.1. Process flow and fabrication steps	15
4. Optimization Techniques for Parameter Extraction	20
4.1. Basics of Parameter Extraction	20
4.2. Different Optimization Techniques used in Parameter Extraction	21
4.2.1. Least Squares Method	21
4.2.2. Gradient Descent Method	23

4.2.3. Genetic Algorithms	24
4.2.4. Particle Swarm Optimization	26
4.3. Selection criteria for optimization techniques	27
5. Methodology and Results	28
5.1. Differential Evolution	28
5.2. Applied Parameter Extraction Procedure	31
5.2.1. Data Acquisition	31
5.2.2. Parameter Extraction	32
5.2.3. Iterative Simulation with Differential Evolution	35
5.3. Results	36
5.3.1. Extraction for Sample A1 10x10 (Enhancement type)	37
5.3.2. Extraction for Sample A1 10x5 (Enhancement type)	42
5.3.3. Extraction for Sample A1 10x20 (Enhancement type)	44
5.3.4. Extraction for Sample D1 10x10 (Depletion type)	45
6. Discussion, Conclusion, and Future Work	48
6.1. Summary and Conclusion	48
6.2. Future Work	49
A. Programmcode	50
Literaturverzeichnis	51

Abkürzungen

<i>TFT</i>	Thin Film Transistor
<i>SPICE</i>	Simulation Program with Integrated Circuit Emphasis
<i>MOSFET</i>	Metal-oxide-semiconductor field-effect transistor
<i>n-channel</i>	Negative channel
<i>p-channel</i>	Positive channel
<i>FET</i>	Field-effect transistor
<i>IGZO</i>	Indium gallium zinc oxide
<i>a-Si</i>	Amorphous silicon
<i>poly-Si</i>	Poly-silicon
<i>LCD</i>	Liquid-crystal displays
<i>I-V</i>	Current-Voltage
I_D	Drain current
V_{GS}	Gate-source voltage
V_{DS}	Drain-source voltage
V_{th}	Threshold voltage
μ	Mobility
C_{ox}	Oxide layer capacitance
W	Channel width
L	Channel length

<i>SS</i>	Sub-threshold slope
<i>CAD</i>	Computer-Aided Design
<i>MoTa</i>	Molybdenum Titanium Aluminium
<i>HF</i>	High Frequency
<i>TMA</i>	Tetramethylammonium Hydroxide
<i>SiN</i>	Silicon-Nitride
<i>SiOx</i>	Silicon-Oxide
<i>ITO</i>	Indium Tin Oxide
<i>PECVD</i>	Plasma-enhanced chemical vapour deposition
<i>VI</i>	VIA or Holes
<i>NoMoTa</i>	Nobelium Molybdenum Tantalum
<i>SEMI</i>	Semiconductor Equipment and Materials International
<i>PSO</i>	Particle swarm optimization
<i>DE</i>	Differential Evolution
<i>rand</i>	Random
<i>bin</i>	Binomial
<i>exp</i>	Exponential
<i>MSE</i>	Mean Squared Error
<i>MAE</i>	Mean Absolute Error
<i>RMSE</i>	Root Mean Squared Error
U_g	Gate voltage
U_d	Drain voltage
I_{dsat}	Saturation drain current
μ_{sat}	Saturation mobility

κ	kappa (Output resistance coefficient)
η	eta (Static feedback factor)
λ	lambda (Channel length modulation factor)
nfs	Bulk charge effect parameter
$nsub$	Bulk surface doping parameter
$theta$	Mobility degradation factor

1. Introduction

1.1. Background

In microelectronics, Thin Film Transistors (TFTs) form the base for the development of large area microelectronic devices, such as displays and sensor arrays. The most prominent feature of Amorphous Oxide TFTs is that they operate with good performances even if they are fabricated at low temperatures without a defect passivation treatment. The TFT mobilities exceed $10\text{ cm}^2/(\text{Vs})$, which are more than ten times larger than those of conventional amorphous semiconductor devices [1]. The increasing complexity in circuit design needs for precise simulations prior to the fabrication process.

Traditional simulation frameworks, such as SPICE, has the capability to conduct these necessary simulations. However, the extraction of the TFT model parameters for these frameworks are a manual process, leading to time inefficiency and the potential for human error.

This thesis aims to resolve this inefficiency by creating a tool designed specifically for TFT simulation. This tool's primary objective is to automate the extraction of parameters from the measured TFT characteristic curves. Moreover, the tool provides a comparative analysis between measured and simulated graphs from the TFT input and output characteristic curves, using an iterative optimization algorithm.

1.2. Objectives and Scope of the Thesis

The objective of this thesis is around the development and evaluation of an automated tool that extracts TFT parameters from measured input and output characteristic curves. By conducting iterative SPICE simulations aimed at Model level 3, until achieving a

predetermined difference tolerance or iteration number, this tool is expected to reduce the time and effort compared with manual parameter extraction, while increasing the accuracy and reliability of the results.

1.3. Structure of the work

This thesis comprises six chapters:

Chapter 1: Introduction - Presents the background, objectives and scope of the study.

Chapter 2: Theory - Explains the fundamentals of Thin Film Transistor, the materials used in TFTs with a focus on IGZO, the electrical characteristics of TFTs, and the importance of parameter extraction in TFT models.

Chapter 3: Fabrication and Process Flow of IGZO TFT - Provides an overview of the fabrication process and describes the process flow, along with discussing the characteristics of IGZO TFTs.

Chapter 4: Optimization Techniques for Parameter Extraction - Discusses the basics of parameter extraction, different optimization techniques used in parameter extraction, their advantages and disadvantages, and the selection criteria for optimization techniques.

Chapter 5: Methodology and Results - Describes the methodology used in the study, explains the applied parameter extraction procedure, discusses performance metrics used for evaluation, and the results of the comparative study.

Chapter 6: Discussion, Conclusion, and Future Work - Summarizes the work and findings, and suggests directions for future work.

2. Theory

This section describes the transistor's operating principle, structure, and materials. The electrical features, such as transfer curves, mobility, and threshold voltage, are then presented. Furthermore, device modelling and importance of parameter extractions are outlined.

2.1. Transistor

A transistor is a three-terminal device that is used primarily for amplifying or switching electronic signals and electrical power [2]. Essentially, it is a type of semiconductor device with three layers of semiconductor material, that are the emitter, base, and collector. The operation of a transistor involves ability to control electronic signals, achieved by manipulating the properties of a portion of the semiconductor [3].

2.1.1. MOSFET

A MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) is a four terminals: source, drain, gate, and body (or substrate). It works by altering the channel width along which charge carriers flow (electrons or holes). The charge carriers enter the channel at the source and exit via the drain. The width of the channel is controlled by the voltage on the gate which is located between the source and the drain [4]. MOSFETs play an important role in electronic devices for signal amplification, control, and switching applications.

2.1.2. N-channel enhancement type MOSFET and depletion type MOSFET

There are two types of MOSFETs: enhancement-type and depletion-type. Each of these types is further divided into negative-channel (n-channel) and positive-channel (p-channel) types. This thesis focuses on n-channel enhancement-type and n-channel depletion-type MOSFETs.

An n-channel enhancement MOSFET is a four-terminal device that typically remains 'off' when there is no voltage across the gate and source. It is built on a p-type substrate, referred to as the body, into which two heavily doped n-type regions are infused, creating the source and drain [5]. The gate is separated from the other terminals by a dielectric layer (metal oxide), which covers the area between the source and drain. Metal is applied on top of this oxide layer to form the gate electrode, with similar metal contacts made to the source, drain, and substrate.

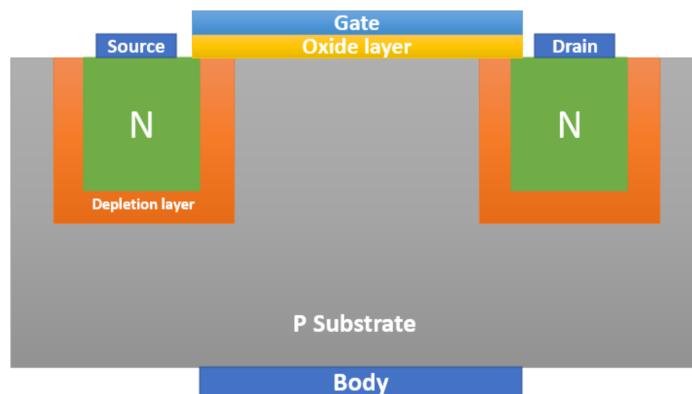


Figure 2.1.: n-channel enhancement type MOSFET

The operation of the device is further detailed. In the absence of a gate voltage, even a positive drain-source voltage yields no current in an enhancement-type MOSFET as there is no channel formation beneath the oxide layer. When a positive voltage is applied to the gate, the resulting electric field pushes the free holes deeper into the substrate, creating a carrier-depletion region and attracting electrons from the source and drain into the channel region. This accumulation of electrons effectively forms a conductive channel connecting the source and drain. The threshold voltage refers to the minimum gate-source voltage needed to create a conductive channel. As this voltage increases beyond the threshold, the carrier density in the induced channel also increases, leading to higher drain current. However, the voltages across the channel are no longer equal. The voltage at the drain

terminal becomes the smallest at the point where the channel is thinnest, leading to a pinch-off situation when the drain-source voltage increases. This causes the MOSFET to enter a saturation region where the drain current becomes independent of the drain-source voltage [6].

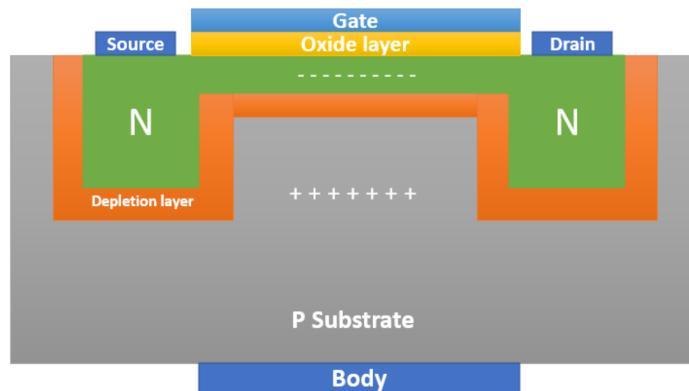


Figure 2.2.: n-channel depletion type MOSFET

In contrast, the structure of the n-channel depletion-type MOSFET is similar to the enhancement-type, with one difference: it has a conductive channel at zero gate-source voltage, while the enhancement-type only forms a channel when the gate-source voltage exceeds the threshold. This occurs because the dielectric layer in the depletion-type MOSFET is heavily doped with positive metal ions during fabrication. This allows the device to form a conductive channel between the drain and source at zero gate-source voltage due to the attraction of electrons by the electric field generated by these ions. Positive gate-source voltage attracts more electrons, leading to a wider channel and lower channel resistance. However, a negative gate-source voltage that reaches the threshold or pinch-off voltage can make the current approach zero, thereby cutting off the n-channel depletion-type MOSFET. Therefore, the threshold voltage of the n-channel depletion-type MOSFET is negative.

2.1.3. Thin Film Transistors (TFTs)

Thin Film Transistors (TFTs) are a type of Field-Effect Transistor (FET) more commonly used in display technologies. Most importantly TFTs are used in active matrix liquid crystal displays (AM-LCD). In active matrix displays TFTs are arranged in a matrix structure, where it works as simple ON/OFF switches, the speed of the switching depends on the refresh rate of the LCD [7].

The principle working of a Thin-Film Transistor (TFT) is the same as that of a standard Field-Effect Transistor (FET). A TFT works by modifying the electrical conductivity of a semiconductor layer, the channel, with an electric field generated by a voltage applied to the gate electrode.

A TFT is made up of three main parts: the gate, the source, and the drain. It works on the principle of controlling the flow of current between the source and the drain via the gate voltage. The transistor is built on a glass substrate onto which a thin film of a semiconductor material, most commonly amorphous silicon or poly-silicon, is deposited. This semiconductor layer serves as the current path between the source and the drain [8].

The channel is not conductive when no voltage is applied to the gate (in an enhancement mode TFT), and no current flows between the source and the drain. When a positive voltage is applied to the gate, an electric field is formed that draws carriers (electrons in an n-type TFT or holes in a p-type TFT) to the region of the semiconductor close to the gate dielectric, resulting in the formation of a conductive channel.

Once the channel has been constructed, a voltage put between the source and drain will cause current to flow through it. The higher the gate voltage, the more carriers are attracted, and the channel becomes more conductive. As a result, the gate voltage controls the current between the source and the drain, allowing the TFT to function as a voltage-controlled switch or amplifier.

2.1.4. Indium Gallium Zinc Oxide (IGZO) TFT

Indium Gallium Zinc Oxide (IGZO) is a thin-film transistor (TFT) technology that has gained popularity in recent years due to its high electron mobility, transparency, and flexibility.

The IGZO TFT employs a semiconductor composed of indium, gallium, and zinc oxides. It was created to address the limitations of amorphous silicon (a-Si) and poly silicon (poly-Si) TFTs. Traditional a-Si has low electron mobility, restricting TFT speed, but poly-Si can have abnormalities due to grain boundaries in the material. The electron mobility of IGZO-TFT is 20-50 times that of amorphous silicon, which is commonly utilized in liquid-crystal displays (LCDs) and e-papers. As a result, IGZO-TFT improves flat-panel display speed, resolution, and size [9].

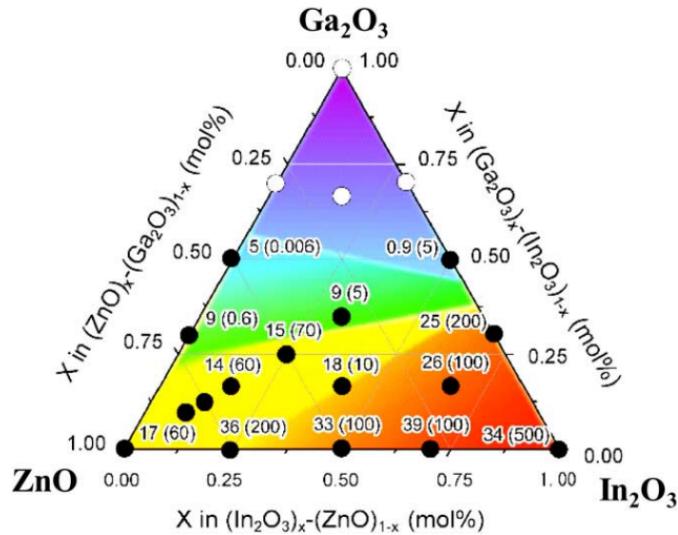


Figure 2.3.: Hall mobility and carrier density of a-IGZO [1].

In the structure of IGZO TFTs, the semiconductor layer is replaced with thin layer of IGZO material. With the added benefits of IGZO layer the TFT has high electron mobility, which allows faster switching and low power consumption. The thickness of IGZO also determines if the TFT is depletion type or enhancement type, with more thick layer more electrons can flow through the channel and hence results in depletion type TFT.

IGZO TFTs exhibit a range of unique characteristics that differentiate them from other types of transistors.

1. **High Mobility:** Compared to amorphous silicon TFTs, IGZO TFTs have significantly higher electron mobility, approximately 10 to 20 times higher. This results in faster switching speeds and lower power consumption, making IGZO TFTs suitable for high-resolution and high-refresh-rate displays [10].
2. **Transparency:** IGZO is transparent due to the large band gap of the material. This transparency feature, combined with the high mobility, allows IGZO TFTs to be used in a variety of applications, including touch screen displays, flexible displays, and high-resolution screens [10].
3. **Stability:** IGZO TFTs are stable under prolonged application of bias, and they exhibit superior bias-stress stability. IGZO has excellent chemical and thermal stability, making these TFTs highly reliable and durable [1].

4. **Low Off-state Current:** IGZO TFTs are known for their extremely low off-state current, which significantly reduces power consumption, especially when the display is in standby or idle mode [1].
5. **Excellent Subthreshold Swing:** IGZO TFTs have excellent sub threshold swing close to the theoretical limit (60 mV/dec at room temperature), which indicates the device's efficient control of the channel by the gate [11].
6. **Fabrication Process:** The fabrication process of IGZO TFTs is compatible with existing flat panel display production lines, making IGZO a cost-effective option for upgrading display technology.

These characteristics make IGZO TFTs ideal for a variety of applications, such as high-quality displays. The low power consumption of IGZO TFTs also leads to the extended battery life of portable devices.

2.1.5. Architecture

A TFT primarily consists of five components: a gate, dielectric layer, semiconductor layer, and source and drain electrodes. Contrary to the commonly used semiconductor substrate in MOSFETs, TFTs employ glass as the substrate. The choice of glass is attributed to its non-conductive nature, exceptional optical transparency, and resistance to reactions with chemicals utilized in semiconductor processing [7]. TFTs can be subdivided into five structures based on the arrangement of their functional layers, namely, Top-gate staggered TFT, Top-gate coplanar TFT, Bottom-gate staggered TFT, Bottom-gate coplanar TFT, and Dual-gate TFT. A schematic diagram of these TFT structures can be found in Figure 2.4.

The primary distinction between single and dual gate TFTs lies in their gate count and arrangement. A single gate TFT has one gate either at the top or bottom, whereas a dual gate TFT includes two gates situated both above and beneath the semiconductor channel. As illustrated in Figure 2.4e, a dual-gate TFT structure comprises a bottom gate with its associated bottom dielectric layer, source and drain electrodes, a semiconductor layer, a top gate, and its associated top dielectric layer. The dual gates in this TFT setup play an integral role in charge accumulation within the semiconductor layer. When a bias is applied to the dual gates, two discrete channels form within the semiconductor layer,

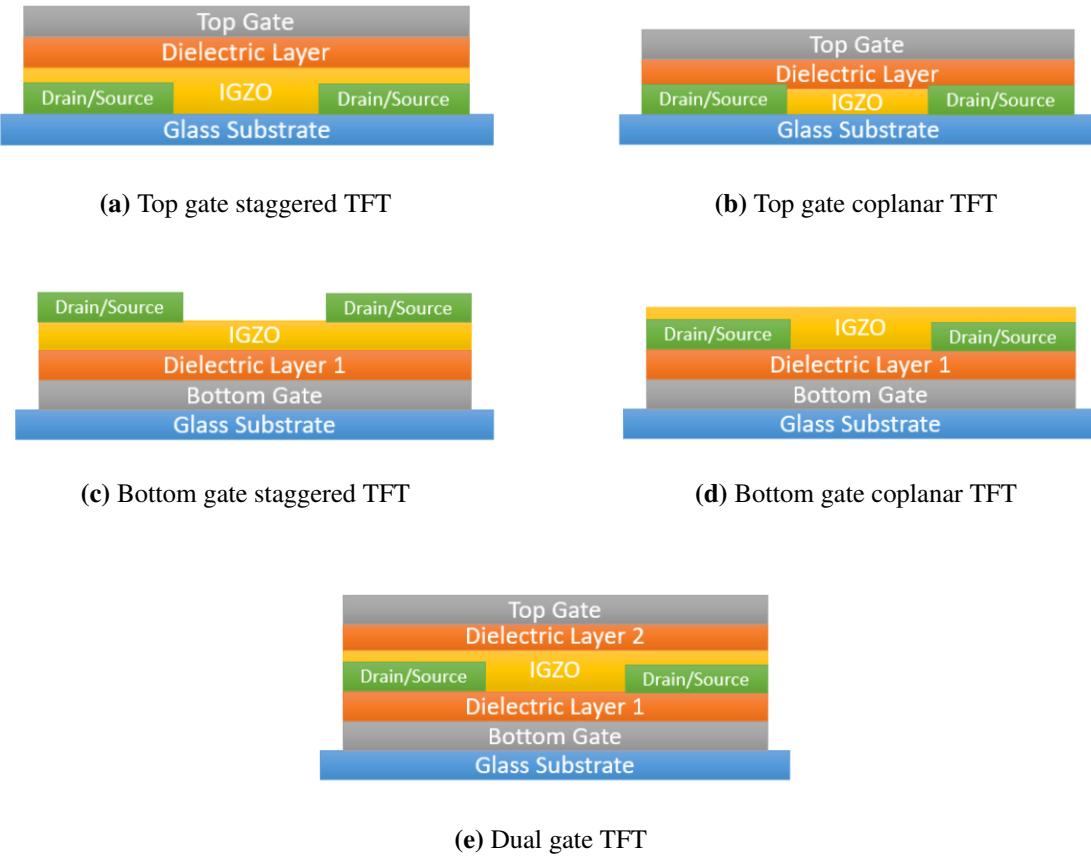


Figure 2.4.: Different types of TFT architectures.

thereby contributing to the superior performance of dual-gate TFTs, such as increased mobility and greater stability under gate bias stress [12].

1. **Top-gate Staggered TFT:** The gate in this configuration is located above the semiconductor channel, but the source and drain electrodes are staggered. The source and drain are not in direct contact with the gate insulator and are located on the other side of the channel [13].
2. **Top-gate Coplanar TFT:** The gate, source, and drain electrodes are all on the same side of the semiconductor layer in a top-gate coplanar arrangement [13].
3. **Bottom-gate Staggered TFT:** In this configuration, the gate electrode is located beneath the semiconductor layer, with the source and drain electrodes staggered.
4. **Bottom-gate Coplanar TFT:** The gate is located at the bottom of the structure, with the source and drain electrodes in direct contact with the gate insulator on the same side of the semiconductor layer.

5. **Dual-gate TFT:** The dual-gate TFT structure has two gates, one above and one below the semiconductor layer. Each gate in this configuration controls the channel formation in the semiconductor layer individually, providing greater control. When compared to single-gate arrangements, dual-gate TFTs often exhibit improved performance, such as higher mobility and stability [14].

This thesis will focus on fabrication and parameter extraction of only Dual-gate IGZO TFTs.

2.1.6. Electrical Characteristics

1. Transfer curve:

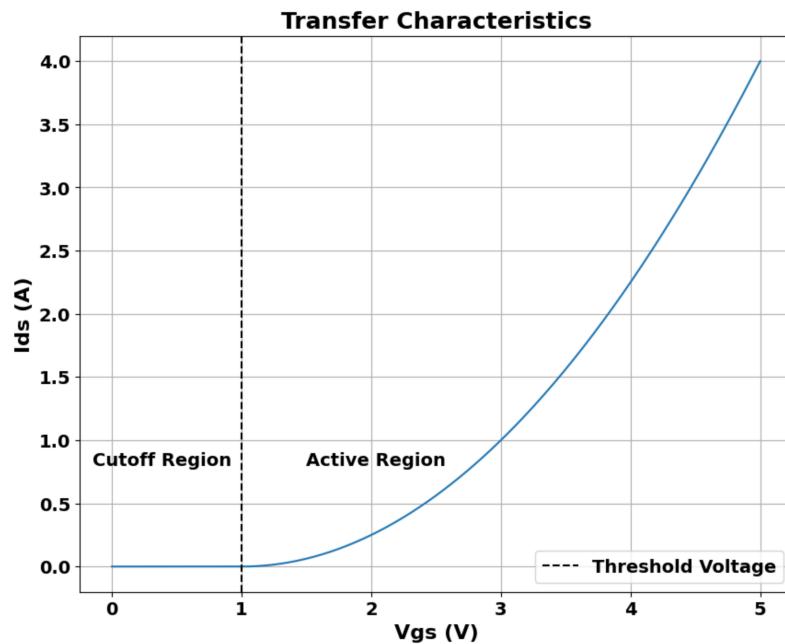
The transfer and output characteristics curves are critical components in TFT measurement. They provide crucial insights into the device's behaviour and performance under varying voltage conditions. The Current-Voltage (I-V) characteristic curve of a TFT includes both the input and output characteristic curves.

The input characteristic of the TFT, describes the relationship between the output current (I_D) and the input voltage (V_{GS}) for a given constant drain-source voltage (V_{DS}), portraying the effects of the gate voltage on the drain current.

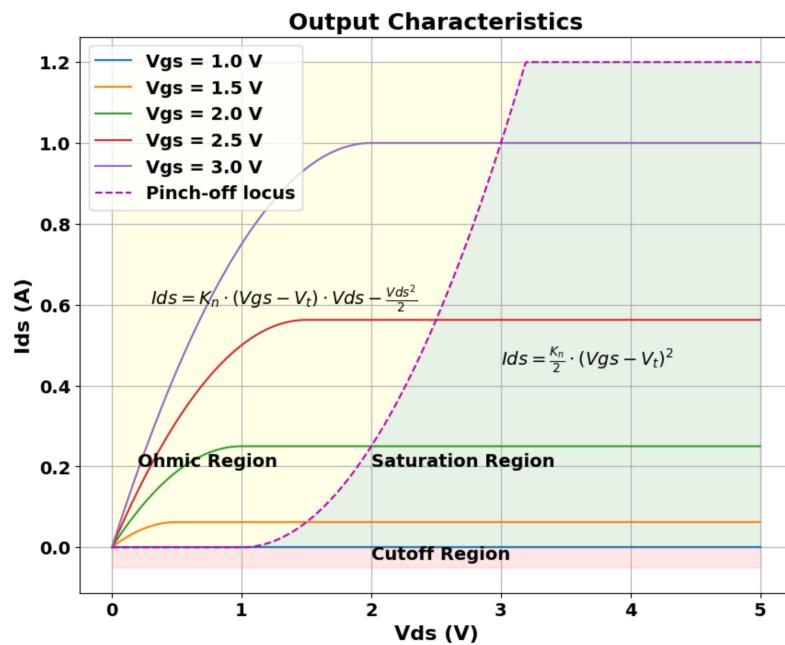
The output characteristic curve of a TFT shows the output current (I_D) varies with the output voltage (V_{DS}) for different levels of input voltage (V_{GS}). The output curves can have varying heights depending on the value of (V_{GS}).

The transfer curve essentially assists in determining the behaviour of the transistor at various operating points. The transfer curve is divided into three sections: cut-off, triode (or linear), and saturation.

- a) **Cut-off:** When $(V_{GS}) < (V_{th})$ (threshold voltage) is reached, the transistor is turned off, and the drain current (I_D) is essentially zero.
- b) **Triode:** When $(V_{GS}) > (V_{th})$ and $(V_{DS}) < (V_{GS} - V_{th})$, the transistor operates in the linear or triode region. In this region, the transistor behaves like a voltage-controlled resistor.



(a) Input characteristics of MOSFET



(b) Output characteristics of MOSFET

Figure 2.5.: Input characteristics curve (a) and output characteristics curve (b) of the MOSFET.

- c) **Saturation:** Saturation occurs when $(V_{GS}) > (V_{th})$ and $(V_{DS}) > (V_{GS} - V_{th})$. In this region, (I_D) is nearly constant, and the transistor behaves like a current source. The general form of the equation representing this relationship in the saturation region is:

$$I_D = \mu(C_{ox}/2)(W/L)((V_{GS} - V_{th})^2) \quad (2.1)$$

where, μ is the charge-carrier mobility, C_{ox} is the oxide capacitance per unit area, W/L is the channel width-to-length ratio, V_{GS} is the gate-to-source voltage, and V_{th} is the threshold voltage.

2. Threshold voltage:

This is the smallest voltage differential between the gate and the source required for current to flow from the drain to the source through the channel. The gate voltage is the voltage at which the transistor begins to conduct. The threshold voltage can be calculated using the transfer curve as the voltage at which the drain current begins to noticeably increase.

The TFT's transfer characteristics can be used to calculate the threshold voltage. This is typically accomplished by projecting the I_D against V_{GS} curve's linear area back to the x-axis (V_{GS}), where $I_D = 0$. This can be accomplished in two ways:

- a) **Linear Extrapolation Method:** A line is fitted to the linear part of the I_D vs V_{GS} curve and then extrapolated back to $I_D = 0$. The V_{GS} at this point is taken as V_{th} .
- b) **Square Root Extrapolation Method:** A line is fitted to the linear part of the $\sqrt{(I_D)}$ vs V_{GS} curve and then extrapolated back to $\sqrt{(I_D)} = 0$. The V_{GS} at this point is taken as V_{th}

These methods give an estimate of the threshold voltage, while the threshold voltage of a specific transistor is determined by several parameters, including the channel shape of the semiconductor layer, the thickness of the dielectric layer, the gate material, and the excess charge in the dielectric [12].

3. Mobility:

Mobility is a measure of how quickly carriers (electrons or holes) can move through a semiconductor material when an electric field is applied. High mobility is desirable for quick response. The mobility can be calculated from the transfer curve in

the saturation region using the equation 2.2. If we plot $\sqrt{I_D}$ versus V_{GS} and fit a straight line to the data in the saturation region, the slope of this line can be used to calculate mobility, given the values of C_{ox} , W , and L .

$$\sqrt{I_D} = \sqrt{(\mu(C_{ox}/2)(W/L)((V_{GS} - V_{th})^2))} \quad (2.2)$$

4. Sub-threshold slope: The sub-threshold slope describes the rate at which the transistor turns on as the gate-to-source voltage rises above the threshold voltage. It is commonly expressed in decibels per decade (dB/decade) or millivolts per decade (mV/dec). As the gate voltage increases, a steep (low value) sub-threshold slope indicates a rapid transition from the off state to the on state. The semi-logarithmic plot of I_D vs V_{GS} can be used to calculate SS . It is equal to the inverse of the line's slope in the sub-threshold region:

$$\frac{dV_{GS}}{d(\log_{10}(I_D))} = SS \quad (2.3)$$

2.1.7. Device Modelling and Importance of Parameter Extraction in TFT Models

Device modelling is an important part of electronics simulation, fabrication, and circuit realisation. For design and analysis of electronic circuits various Computer-Aided Design (CAD) tools, and SPICE models are used. Within these tools, mathematical models and equations are used to represent device behaviour. These models allow to predict and optimize the performance of a circuit before it is physically fabricated, saving resources and time [15]. The necessity for precise mathematical compact models grows more as devices become more complicated.

Parameter extraction involves determining the values of the electrical parameters in the model, such as threshold voltage, mobility, and sub-threshold swing, etc. These values are physical quantities that must be accurately measured or estimated from measured data. Having accurate parameter values ensures that the model correctly represents the real transistor behaviour, which in turn leads to more effective and efficient designs.

Compact models for semiconductor devices are broadly categorized into three types:

- **Physical Models:** These models describe the device's behaviour using parameters that have direct physical meaning. For example, parameters might involve the de-

vice's size, carrier mobility, threshold voltage, and so on [16]. Physical models are derived from fundamental physical laws governing device operation, such as quantum mechanics and thermodynamics.

- **Empirical Models:** These models describe the device's behaviour using sophisticated mathematical equations and parameters with no physical relevance [16]. Typically, these models are based on curve fitting to experimental data. While they cannot give information on the device's actual operation, they can accurately mimic the device's behaviour in the conditions for which they were designed.
- **Semi-empirical models:** the device's behaviour is expressed using both device physics parameters and empirical parameters for better fitting in all device regions [16].

In summary, semi-empirical models provide a balance between physical insight provided by physical models and empirical model fitting accuracy. It enables accurate models, efficient designs, quality control in manufacturing, and the ongoing evolution of TFT technology.

3. Fabrication and process flow of IGZO TFT

The detailed manufacturing technique of Indium Gallium Zinc Oxide (IGZO) Thin-Film Transistors (TFTs) is covered in this chapter. The systematic process flow involved in the production of IGZO TFTs, including the techniques and materials utilized at each step, is examined.

3.1. Process flow and fabrication steps

The fabrication was divided into 8 substrates with 4 batches of different semiconductor layer thickness shown in Table 3.1.

Table 3.1.: IGZO Thickness for Various Substrates

Group	A1	A2	B1	B2	C1	C2	D1	D2
IGZO Thickness (nm)	50	50	70	70	90	90	110	110

Here are the fabrication steps:

- 1. Sputter MoTa onto glass substrate and structure it:**



Figure 3.1.: Cross-sectional view after fabrication of the bottom gate

The sputtering of a Molybdenum Titanium Aluminium (MoTa) layer is the initial step in the process. A 6000 High Frequency (HF) sputtering system with three oscillations and a speed of 0.32 m/min is used to accomplish this. The pre-sputtering settings are set to 500 W for 2 minutes. Following sputtering, the substrate is coated with an photoresist coater and a hotplate set to 115°C. This procedure ensures that the substrate is uniformly coated.

The coated substrate is then exposed for 3.5 seconds in soft contact mode and a mask to protect particular areas from exposure. The following development technique employs Tetramethylammonium Hydroxide (TMA 7:1) developer for approximately 25 seconds, with the exact timing adjusted based on visual inspection.

The MoTa layer is then etched with an Aluminium Etchant. The etching is done in a water bath that is kept at a constant temperature of 39°C. The etching process continues until visual confirmation is obtained. After etching, the remaining coating is removed with a series of solvent washes.

2. Fabricate double layer dielectric (SiN and SiO_x):



Figure 3.2.: Cross-sectional view after fabrication of the first dielectric layer

The first gate oxide layer is formed in the second step. This layer is created in a KAI PECVD machine utilizing a plasma-enhanced chemical vapour deposition (PECVD) method. The substrate is next coated, exposed, developed in a manner similar to the previous stage, and the mask employed is designated as VI (VIA or holes).

Following that, the substrate is hard baked in a convection oven for 30 minutes at 120°. Then the plasma etcher is then used to construct vias in the gate oxide layer. Finally, the coating is removed by repeating the solvent cycle from the first stage.

3. Fabricate drain-source metal (MoTa and ITO):



Figure 3.3.: Cross-sectional view after fabrication of the drain and source layer

MoTa and Indium Tin Oxide (ITO) are sputtered together to generate the drain-source metal in the third stage. MoTa is initially sputtered onto the substrate and constructed with a Nobelium Molybdenum Tantalum (NoMoTa) mask. Then, an ITO layer is sputtered onto the MoTa layer, followed by ITO patterning. Following the ITO layer, the MoTa layer is patterned, and ultimately, the photo resist is removed. The substrate is then annealed for an hour in a 250°C oven to cover the MoTa with ITO. This prevents the MoTa from oxidizing during annealing and isolates it from the IGZO semiconductor layer. If MoTa comes into direct contact with IGZO, oxidation and oxygen vacancies arise at the MoTa-IGZO junction, causing the transistor to malfunction [17].

4. Fabricate IGZO layer and anneal substrate at 300°C:



Figure 3.4.: Cross-sectional view after fabrication of the semiconductor layer

The IGZO semiconductor layer is subsequently produced on the substrate. The IGZO layer is sputtered onto the substrate, patterned with a Semiconductor Equipment and Materials International (SEMI) mask. Then A1, B1, C1, and D1 substrates are etched with old etchant, whereas A2, B2, C2, and D2 substrates are etched with new etchant to check if etchant play a role in changing the behaviour

of the transistors. Then the photo resist is eventually removed. The substrate is annealed in an oven at 300°C for two hours.

Table 3.2.: Actual IGZO Thickness for Various Substrates

Group	A1 and A2	B1 and B2	C1 and C2	D1 and D2
IGZO Thickness (nm)	50	70	90	110
Actual IGZO Thickness (nm)	44.5	59.9	71.1	85
Etching time (s)	30	35	40	45

5. Fabricate second dielectric layer (SiO_x) and Anneal substrate at 250°C overnight:

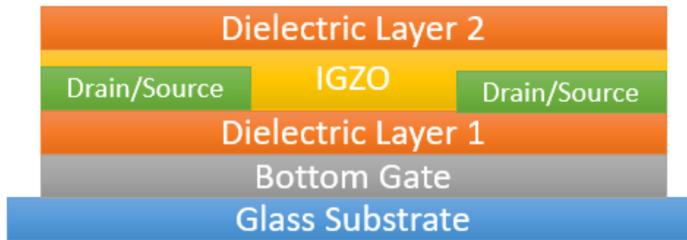


Figure 3.5.: Cross-sectional view after fabrication of the second dielectric layer

The next stage is to use KAI-PECVD to create another dielectric layer of 130 nm SiO_x. This is due to the fact that hydrogen created during silicon nitride sputtering can permeate the semiconductor layer and affect transistor performance. As a result, this dielectric layer is made entirely of silicon oxide. Following patterning, the substrate is baked for 30 minutes at 120°C (lithography process), etched in plasma, and the photo resist is removed. The substrate is then annealed overnight in a 250°C oven to remove any moisture within the substrate as the oxide layer absorbs moisture from the environment and can affect the transistor characteristics [18].

6. Sputter a double layer (MoTa and ITO) as a top gate and anneal substrate at 250°C overnight:

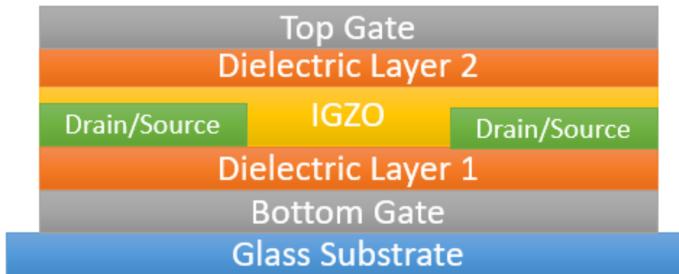


Figure 3.6.: Cross-sectional view after fabrication of the top gate

Finally, a top gate is sputtered with a double layer of 70 nm MoTa and 50 nm ITO. After that, the substrates are annealed overnight at 250°C. ITO is utilized to avoid MoTa oxidation during overnight annealing.

This completes the fabrication process of IGZO TFT. The manufactured TFT has width ranging from 5 μm - 1000 μm and length from 5 μm , 10 μm , and 20 μm .

This concludes the chapter on the fabrication and process flow of IGZO TFTs. The next chapter will explore the measurement techniques and python scripts developed for parameter extraction from the transistors.

4. Optimization Techniques for Parameter Extraction

4.1. Basics of Parameter Extraction

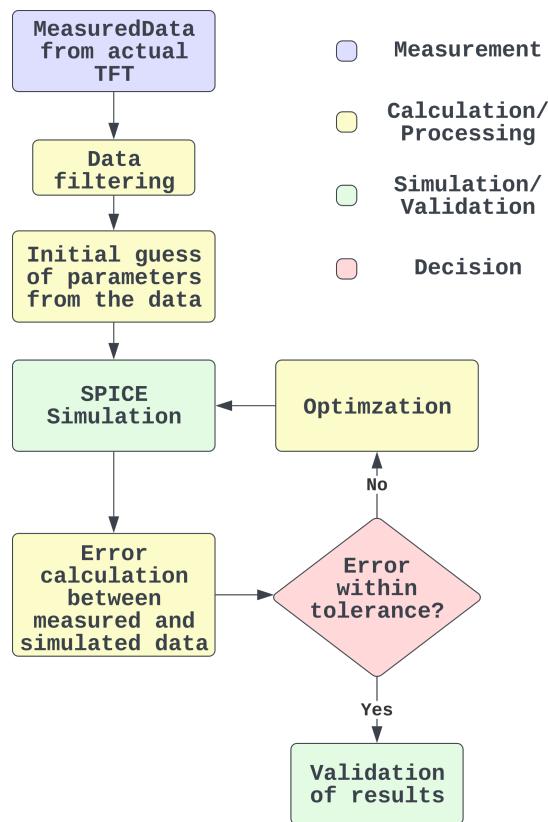


Figure 4.1.: Flow diagram showing extraction processes

Parameter extraction is a critical step in the modelling and simulation of semiconductor devices, particularly TFTs. The performance of which has an impact on digital devices such as screens, solar cells, and integrated circuits.

The primary goal of parameter extraction is to get model parameters that can properly reflect the device behaviour using simulation framework like SPICE. These parameters are usually physical characteristics that characterize the behaviour of the semiconductor device. The threshold voltage (the minimum gate voltage required to create a conducting path between the source and the drain), mobility (which measures the speed at which carriers move through the device under the influence of an electric field), sub-threshold swing (which describes the sharpness of the transition from the off state to the on state in a transistor), and many other parameters are some of the important parameters needed for accurate simulation.

In general, parameter extraction is a multi-step process that involves experimental observations and computational methods. The process begins with data collection, which is accomplished by electrical characterisation such as current-voltage (I-V) measurements.

The data is then fitted to a model using computational methods. This step involves the use of optimization techniques, which try to minimize the deviation between the experimental data and the model's simulation predictions. These algorithms will tweak the model's parameters until an optimal combination that matches the measured data is reached.

4.2. Different Optimization Techniques used in Parameter Extraction

There are many optimization methods that can be used in parameter extraction. Some of the most common ones include:

4.2.1. Least Squares Method

The least squares method is a great tool for data fitting problems. It works by minimizing the sum of the squares of the differences between measured and predicted values. When the errors in the measured data are uncorrelated and have equal variances, the method provides an unbiased estimate. The least squares method, on the other hand, can be sensitive to outliers in the data, which can distort the resulting parameter estimates.

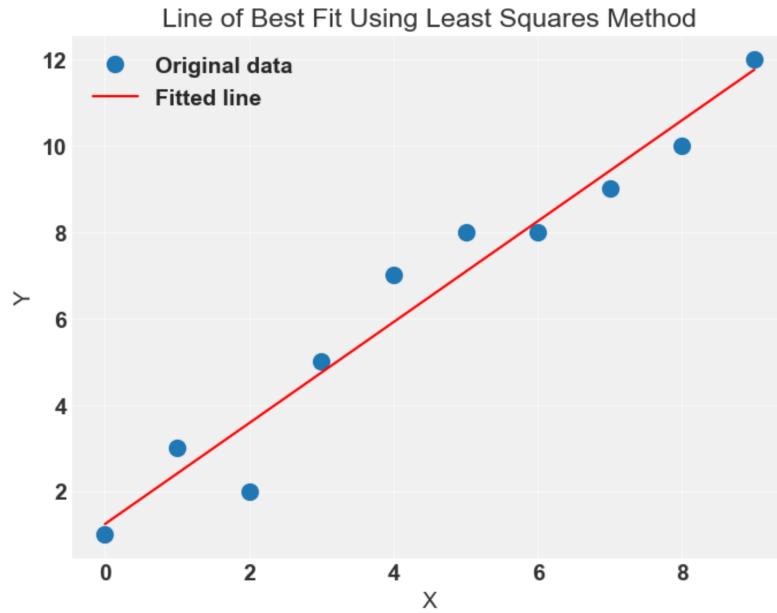


Figure 4.2.: The graph represents a set of data points on a two-dimensional grid along with a line of best fit, derived using the least squares method.

Figure 4.2, shows an example of line of best fitted around noisy data points calculated using the least squares method. This method seeks a line that minimizes the sum of the squares of the vertical distances between each data point and the line, providing the best estimate of a linear relationship between the X and Y variables. This line's slope and intercept represent, the predicted change in the Y variable for a unit change in the X variable and the predicted Y value when X equals zero.

The equation for the line of best fit is:

$$y = m \cdot x + c \quad (4.1)$$

Here, 'm' is the slope of the line and 'c' is the y-intercept.

The slope 'm' and y-intercept 'c' can be calculated from the data by minimizing the sum of squared vertical distances between the data points and the line:

$$m = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} c = \frac{\sum y - m \sum x}{n} \quad (4.2)$$

Where, Σ denotes the sum of the elements, x and y are the data points, n is the number of data points.

The least squares method is simple to implement and provides good results for linear problems. However, it can be sensitive to outliers, and it assumes that the error has a normal distribution, which is not the case in real problems.

4.2.2. Gradient Descent Method

The gradient descent algorithm is a first-order optimization algorithm that works by iteratively adjusting the model parameters in the direction of the objective function's steepest descent [19]. This method can be useful for navigating parameter space, but its success is heavily dependent on the careful selection of initial parameters and iteration step size. Gradient descent may become stuck in local minima, making it unsuitable for optimization problems with many local minima spread across the parameter search space [20].

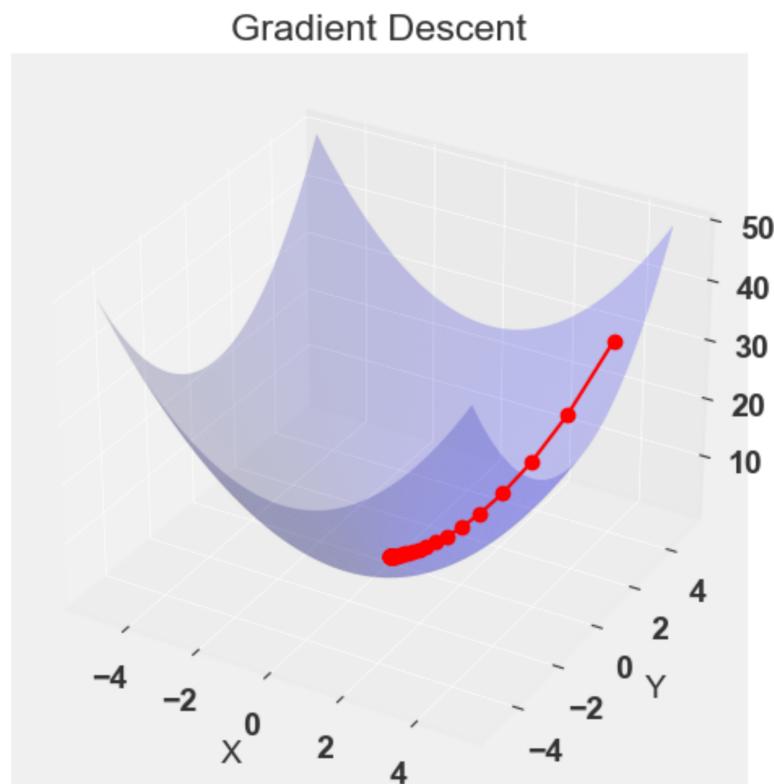


Figure 4.3.: 3D plot of gradient descent optimization for $z = x^2 + y^2$.

Figure 4.3 shows a 3D plot of the gradient descent process, an optimization technique widely used in machine learning to minimize cost functions. The surface plot displays the cost function, here a simple quadratic function $z = x^2 + y^2$, with each point (x, y, z) denoting a pair of parameters (x, y) and their corresponding cost z . The cost signifies the performance of the model, with lower costs implying better fits to the data. The highlighted path, depicted as a red line with 'o' markers, demonstrates the steps of the gradient descent algorithm. It starts from an arbitrary high point on the surface and progressively moves towards the global minimum of the cost function. At each step, it moves in the direction opposite to the gradient (the direction of steepest ascent), descending down the

cost function surface, until it converges at the minimum point, where the cost is minimized.

Gradient descent employs an equation based on the gradient for the cost function minimization. The gradient descent update rule can be expressed in the context of minimizing a cost function represented by $z = f(x, y)$ as follows:

$$x_{n+1} = x_n - \alpha \frac{\partial f}{\partial x}(x_n, y_n) \quad (4.3)$$

$$y_{n+1} = y_n - \alpha \frac{\partial f}{\partial y}(x_n, y_n) \quad (4.4)$$

where (x_n, y_n) are the current parameter values, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ are the partial derivatives of the cost function with respect to x and y , respectively, and α is the learning rate that determines the step size at each iteration. The learning rate determines the algorithm's convergence speed, and an acceptable value must be determined to balance convergence speed and stability. This approach lowers the cost function towards the global minimum by iteratively updating parameter values based on the gradient descent.

Gradient descent can provide good results when the error surface is convex and differentiable. However, the choice of the step size can greatly influence the convergence of the algorithm. A large step size may lead to overshooting, while a small step size may result in slow convergence.

4.2.3. Genetic Algorithms

The genetic algorithm is a powerful optimization tool that can solve both constrained and unconstrained problems. Its methodology is based on natural selection and biological evolution principles, hence the name. The procedure entails iteratively updating a population of solutions known as "individuals." During each iteration, parents from the current population are chosen to produce offspring for the next generation. The population converges towards an optimal solution as generations succeed one another [21].

Genetic algorithms excel at dealing with non-linear, multi-modal, and discontinuous objective functions and can effectively explore a large search space.

The fundamental steps of a genetic algorithm can be visualized through Figure 4.4, which begins with the creation of an initial population, followed by scoring and scaling of this

population. The elite individuals are then retained, parents are selected, and crossover and mutation events produce offspring. This cycle repeats from the scoring and scaling stage [21].

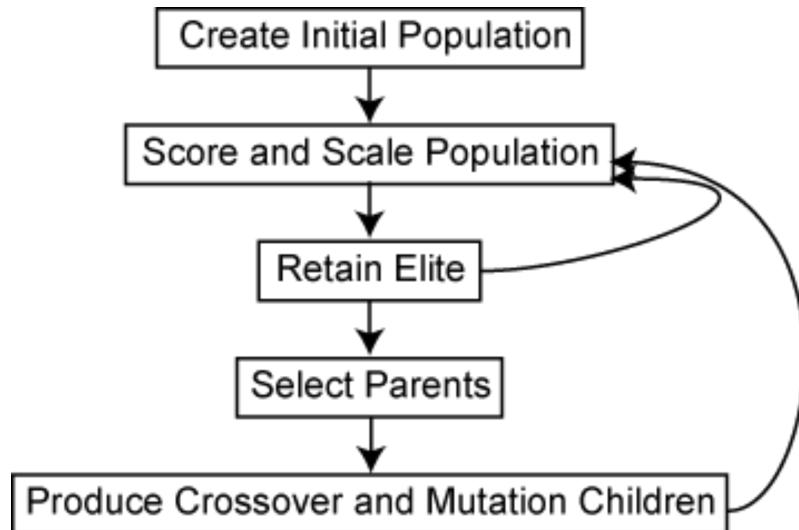


Figure 4.4.: Genetic Algorithm Flowchart [21].

Genetic algorithms predominantly employ three types of operations to generate a new population [21]:

- Selection, which stochastically picks the parents that will contribute to the next generation. The selection is usually dependent on the individuals' scores.
- Crossover, which merges two parents to create offspring for the succeeding generation.
- Mutation, which introduces random changes to a single parent to generate offspring.

An offspring can be an elite child, a crossover child, or a mutation child. An elite child is a clone of its parent, a crossover child inherits traits from both parents, and a mutation child bears a change from its parent.

Genetic algorithms are robust and can find global optima in complex search spaces. However, they can be computationally intensive and may require careful tuning of their parameters (like crossover rate, mutation rate) for effective performance.

4.2.4. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic optimization technique inspired by bird flocking and fish schooling [22]. In PSO, each solution, or "particle," moves around the search space based on its own experience as well as the experience of neighbouring particles, with the goal of finding the optimal position. This is a heuristic approach because it can never guarantee that the true global optimal solution exists. However, it is frequently discovered that the solution obtained by PSO is very close to the global optimal [22].

Figure 4.5, depicts the Particle Swarm Optimization (PSO) technique for optimizing a two-dimensional Rosenbrock function:

$$f(x,y) = (a-x)^2 + b(y-x^2)^2 \quad (4.5)$$

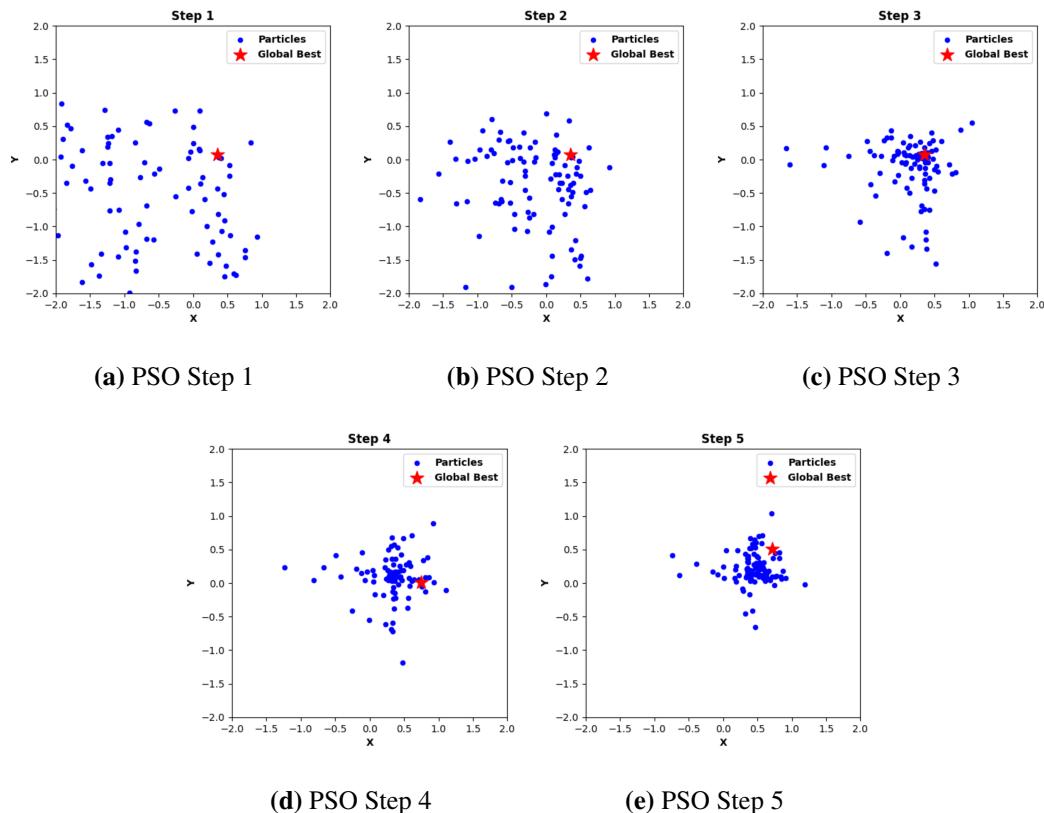


Figure 4.5.: Particle Swarm Optimization (PSO) for optimizing a two-dimensional Rosenbrock function.

Each graph depicts a step in the PSO algorithm, with the particle positions (blue dots) and the global optimum position (red star) shown. The particles advance towards the global

best position as the process progresses, guided by their personal best positions and the overall best position obtained by any particle. The graphs show the particles convergence towards the optimum during the duration of the iterations.

PSO is easy to implement and has fewer parameters to adjust compared to genetic algorithms [22]. However, it can sometimes be trapped in local optima, especially in complex search spaces.

4.3. Selection criteria for optimization techniques

Choosing the right optimization technique for parameter extraction depends on several factors. These factors may include the nature of the objective function (linear, non-linear, multi-modal, etc.), the presence of noise in the data, the dimensionality of the problem, and the available resources. Prior knowledge of the problem can also help in the selection of the appropriate method. For example, if the error surface is known to be convex, gradient descent may be a good choice. On the other hand, for complex and multi-modal error surfaces, population-based methods like genetic algorithms or PSO could be more effective.

Table 4.1.: Advantages and Disadvantages of Different Optimization Techniques

Technique	Advantages	Disadvantages
Least Squares Method	Simple implementation, good results for linear problems	Sensitivity to outliers, assumes normal distribution of errors
Gradient Descent Method	Good results for convex and differentiable error surfaces	Choice of step size affects convergence
Genetic Algorithms	Robust, finds global optima in complex search spaces	Computationally intensive, requires parameter tuning
Particle Swarm Optimization	Easy implementation, fewer parameters to adjust	Potential to get trapped in local optima

For this thesis genetic algorithm specifically differential evolution algorithm is implemented as it is a powerful tool for non-linear problems and converges well towards global minima and works well with defined bounds.

5. Methodology and Results

This study's methodology was designed to extract critical parameters from the Indium Gallium Zinc Oxide (IGZO) Thin Film Transistor (TFT) and use these parameters to generate accurate simulations. Data collecting, parameter extraction for initial guess to define bounds, iterative simulation utilizing Differential Evolution (DE) (genetic algorithm), and data visualization are the four primary components of the process.

5.1. Differential Evolution

Differential Evolution is a stochastic population based optimization algorithm that excels at solving optimization problems in continuous domains [23]. In this thesis, the main objective was to optimize the match between the simulated and measured TFT device properties. The method starts with a population of candidate solutions (various sets of parameters) and evolves it over generations. New candidate solutions are generated at each generation by merging existing solutions according to the algorithm's mutation and crossover principles. The algorithm then employs a selection process to determine which solutions will be passed down to the next generation based on their quality i.e., how well the simulated characteristics match the observed features with a particular set of parameters.

The three main operators of the DE algorithm are mutation, crossover, and selection. A new trial vector is created during the mutation phase by adding the weighted difference between two population vectors to a third vector [24]. The mutation operation in DE can be represented by the following equation:

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \quad [24] \quad (5.1)$$

Here, $V_{i,G}$ is the mutated vector in the generation G , $X_{r1,G}$, $X_{r2,G}$, and $X_{r3,G}$ are randomly selected vectors from the population at the current generation G , and F is a real and

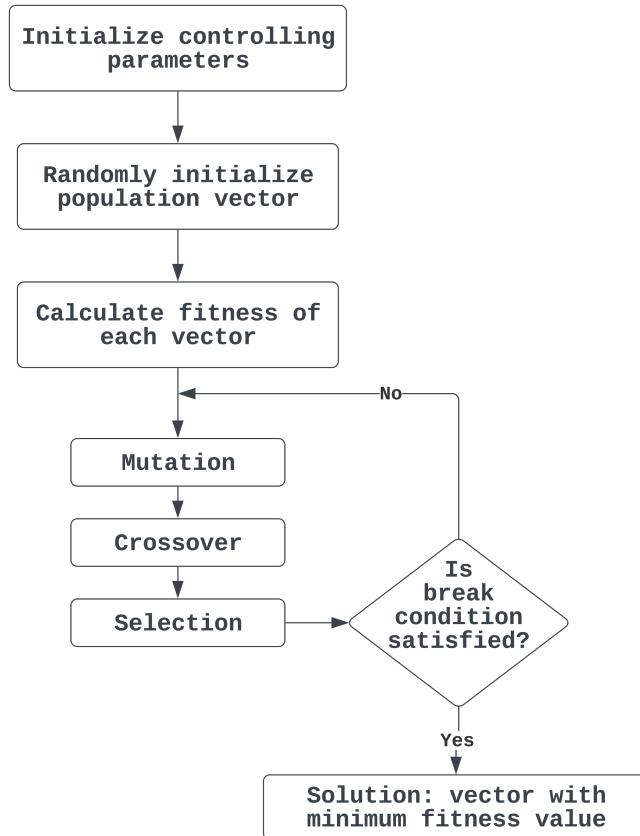


Figure 5.1.: Differential Evolution Flowchart.

constant factor which controls the amplification of the differential variation ($X_{r2,G} - X_{r3,G}$) [24].

There are various strategies for generating the donor vector:

- *rand/1*: A donor vector is created by adding the weighted difference between two randomly selected vectors to a third one: $v = x_a + F \cdot (x_b - x_c)$.
- *rand/2*: Similar to *rand/1* but uses two differential vectors: $v = x_a + F \cdot (x_b - x_c + x_d - x_e)$.
- *best/1*: The base vector is the current best solution in the population: $v = x_{best} + F \cdot (x_a - x_b)$.
- *best/2*: Similar to *best/1* but uses two differential vectors: $v = x_{best} + F \cdot (x_a - x_b + x_c - x_d)$.

- ***rand-to-best/1***: A hybrid of '*best/1*' and '*rand/1*': $v = x_a + F \cdot (x_{best} - x_a) + F \cdot (x_b - x_c)$.

Here, x_a, x_b, x_c, x_d, x_e are randomly chosen individuals from the population, x_{best} is the best individual in the population, F is a scaling factor [25].

During the crossover phase, components from the trial vector (V_i, G) and a target vector (X_i, G) are combined to create a recombinant vector (U_i, G). There are two main strategies:

- **Binomial (bin)**: For each parameter, a random number is generated. If the random number is below the crossover rate, the parameter value from the donor is selected, otherwise, the parameter value from the target is selected [25].
- **Exponential (exp)**: A random parameter is selected. From that parameter onwards, values are taken from the donor until a randomly generated number exceeds the crossover rate, the remaining parameters are taken from the target [25].

Crossover phase is done using the following equation:

$$U_{i,j,G} = \begin{cases} V_{i,j,G} & \text{if } (rand_j \in [0, 1] \leq Cr) \text{ or } (j = rnbr(i)) \\ X_{i,j,G} & \text{otherwise} \end{cases} \quad [24] \quad (5.2)$$

In this case, $U_{i,j,G}$ is the $j-th$ parameter of the recombinant vector of the $i-th$ vector at the generation G , $rand_j$ is a uniformly distributed random number within $[0, 1]$, Cr is the crossover constant, and $rnbr(i)$ is a randomly chosen index for the $i-th$ vector.

Finally, during the selection step, the algorithm determines whether the freshly created vector will be included in the new population, often based on the vector's fitness.

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad [24] \quad (5.3)$$

In this equation, $X_{i,G+1}$ is the $i-th$ vector at the next generation $G+1$, $f()$ is the objective function, and $f(U_{i,G})$ and $f(X_{i,G})$ represent the objective function value of the recombinant and target vector at generation G , respectively. The vector with the lower objective function value will be selected for the next generation.

The DE algorithm in this problem uses the analytic methodology's initial parameters as the search space's boundaries. As a result, the search space is multidimensional, with each

dimension representing a parameter. DE alters these parameters within their constraints with the goal of minimizing the target function, which in this case is the difference between simulated and measured device characteristics.

The objective function denotes the solution's quality, with a lower value suggesting a better answer. The objective function for this study is specified as the Mean Squared Error (MSE) between the measured and simulated features using a set of parameters. The purpose of the DE method is to discover the parameters that minimize this MSE.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (5.4)$$

Where: n is the total number of observations or instances, Y_i is the actual value for the $i-th$ instance, \hat{Y}_i is the predicted or estimated value for the $i-th$ instance.

The DE algorithm converges to a solution after multiple iterations (or generations), producing a set of optimized parameters that best suit the simulated features to the measured data.

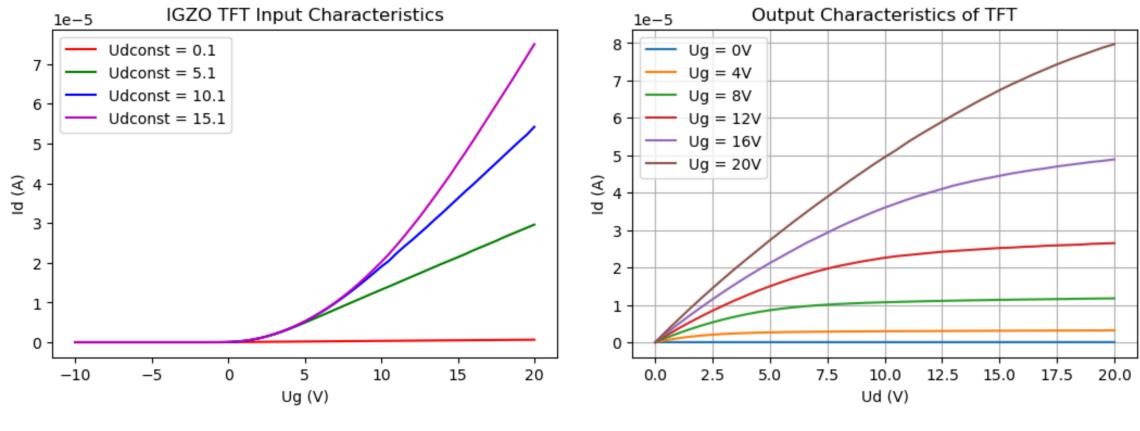
The Differential Evolution strategy *DE/best/1/bin* was chosen for this study due to its optimal balance of exploration and exploitation, by the binomial crossover. The strategy employs the best current solution, pushes convergence towards the minima of the solution space. It is well-suited to the task of parameter extraction in Thin Film Transistor models.

5.2. Applied Parameter Extraction Procedure

The parameter extraction process encompasses a series of steps aimed at extracting the parameters from the measured data and applying these parameters in simulations using Differential Evolution (DE). The process begins with data acquisition, progresses through parameter extraction, and culminates in iterative simulation.

5.2.1. Data Acquisition

The process begins by measuring the TFT's input characteristics by reading drain current (I_D) from -10 V to 20 V of gate voltage (U_g) at 0.1 V, 5.1 V, 10.1 V, and 15.1 V of constant drain voltage (U_d).



(a) Measured Input characteristics of IGZO TFT C1 sample 1 ($10 \mu\text{m} \times 10 \mu\text{m}$) (b) Measured Output characteristics of IGZO TFT C1 sample 1 ($10 \mu\text{m} \times 10 \mu\text{m}$)

Figure 5.2.: Input characteristics curve (a) and output characteristics curve (b) of IGZO TFT C1 sample 1 ($10 \mu\text{m} \times 10 \mu\text{m}$).

This was achieved by reading in two types of experimental data files: input and output characteristics of the IGZO (Indium Gallium Zinc Oxide) TFT (Thin Film Transistor) device, using the read data function in Python. The input file comprises the applied gate voltage (U_g) and the corresponding drain current (I_d) for different constant drain voltages ($U_{d\text{const}}$). Simultaneously, the output file houses the drain voltage (U_d) and corresponding drain currents for various constant gate voltages ($U_g\text{const}$). All data is stored in NumPy arrays, which allow for efficient data manipulation and mathematical operations.

5.2.2. Parameter Extraction

The next step involves extracting essential parameters from the measured data. These parameters include Threshold Voltage (V_{th}), mobility (μ), kappa (κ), eta (η), and lambda (λ).

- **The threshold voltage (V_{th})** is calculated using the *func_vth* function. This function estimate the threshold voltage by analysing the TFT's input characteristics, specifically, observing the steepest slope on the square root of the drain current versus gate voltage plot.

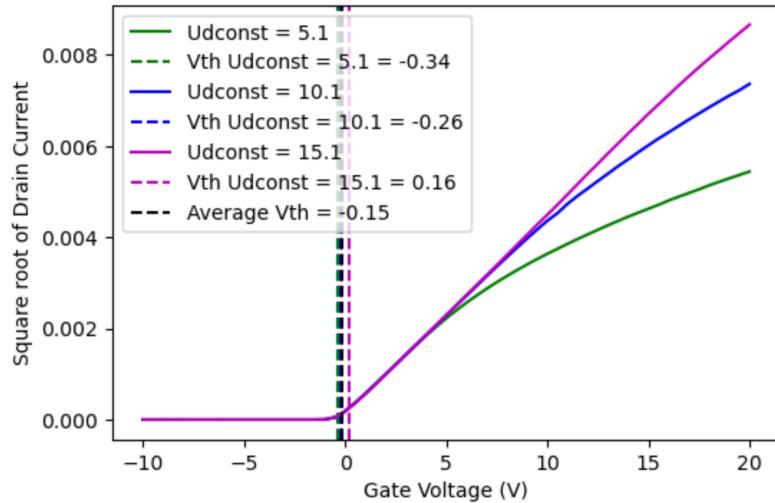


Figure 5.3.: Average V_{th} Calculated.

The function calculates the V_{th} using a list of measured drain currents (Id_data_list). For each set of drain current measurements, it computes the square root of the absolute values of the drain current ($\sqrt{Id_data} = \sqrt{|Id_data|}$), calculates the slopes between consecutive points on the square root of drain current against gate voltage plot ($slopes = \frac{\sqrt{Id_data_{i+1}} - \sqrt{Id_data_i}}{U_{g_{i+1}} - U_{g_i}}$), and identifies the point of maximum transconductance by finding the index of the steepest slope ($steepest_slope_index = argmax(slopes)$).

The threshold voltage (V_{th}) for each set of measurements is then estimated by extrapolating the tangent at the point of maximum transconductance to the x-axis.

$$V_{th} = point_x - \frac{point_y}{slopes_{steepest_slope_index}} \quad (5.5)$$

The function finally returns the average of all these V_{th} values ($V_{th_average} = \frac{1}{n} \sum_{i=1}^n V_{th_i}$), providing an estimate of the device's threshold voltage.

- **The mobility parameter**, indicative of the TFT's ability to conduct current, is calculated using the *func_mobility* function, which operates on the principle of the field effect mobility in the saturation region of the TFT.

The function *func_mobility* calculates the average saturation mobility of a TFT device. It does this by iterating over each set of drain current (Id) measurements in Id_input_{meas} , corresponding to the different constant drain voltages ($Udconst_values$). For each Id set, it identifies the saturation region where the gate

voltage (U_g) is greater than the sum of the constant drain voltage and the threshold voltage ($U_g > (U_{dconst} + V_{th})$), and computes the saturation mobility using the equation:

$$\mu_{sat} = \frac{2 \cdot Id_{sat}}{C_{ox} \cdot \frac{W}{L} \cdot (U_{gsat} - V_{th})^2} \quad (5.6)$$

where, W is the channel width, L is the channel length, and C_{ox} is the gate oxide capacitance. This process yields a list of saturation mobility values (milities), and the function returns their average.

- **Output resistance coefficient**, $kappa$, is computed using the *calculate_kappa* function. Kappa is determined by analysing the slope of the log plot of the drain current versus the gate voltage. The function computes the output resistance coefficient, using a set of measured drain currents, represented as *Id_input_meas*. For each set of *Id* measurements, it calculates the base-10 logarithm of the absolute drain current values (represented as $\log_{10}(|Id_input_meas|)$).

Then, it computes the derivative of $\log_{10}(Id)$ with respect to the gate voltage U_g (represented as *Ug_input_meas*), yielding $\frac{d\log_{10}(Id)}{dV_g}$.

The κ factor is then calculated from the minimum value of this derivative using the equation:

$$\kappa = \frac{1}{1 - \min\left(\frac{d\log_{10}(Id)}{dV_g}\right)} \quad (5.7)$$

This process is repeated for each set of *Id* measurements and the average of all *kappa* values is returned as *mean_kappa*.

- **The static feedback factor**, eta , is determined by analysing the slope of the natural log of the drain current versus the drain voltage plot in the calculate *eta* from output characteristic curve.

The function *calculate_eta_from_output* calculates the average The static feedback factor. The function begins by iterating over each set of measured drain currents, represented as *Id_output_meas*. For each *Id* dataset, it first checks if the corresponding gate voltage $U_{g_output} > V_{th}$. If not, the loop continues to the next set of data. If the condition is met, it takes the natural logarithm of the absolute drain current values $\ln(|Id_{data}|)$.

It then performs a linear regression on the plot of $\ln(Id)$ versus the drain-source voltage *Ud_output_meas*. The slope of this regression line gives the value of η .

This process is repeated for each Id dataset, and the function returns the average of all η values, denoted as $mean_{\eta}$.

- Finally, the function `calculate_lambda` is used to derive the **channel length modulation factor**, often denoted as λ . This process begins with a set of measured drain currents, Id_output_meas , and drain-source voltages, Ud_output_meas . The function first calculates the derivative of Id with respect to Ud , represented as:

$$\frac{dId}{dUd} = \frac{\nabla Id_output_meas}{\nabla Ud_output_meas} \quad (5.8)$$

Then, λ is calculated as the average of these derivatives in the saturation region, typically the last 10 points, represented as $\lambda = np.mean(\frac{dId}{dUd}[:, -10 :])$. This process yields the average channel length modulation factor, which is an indication of the change in the effective channel length due to the drain-source voltage. The function returns this λ value.

5.2.3. Iterative Simulation with Differential Evolution

After the extraction of these parameters, they are utilized as the initial inputs for the Differential Evolution (DE) algorithm. Here certain range from initial guess defines the bounds of the algorithm search space. Bounds for each parameters were fixed after achieving satisfactory results. Parameters like nfs , $nsub$, $theta$ was selected according to best simulation results as these parameters can not be calculated directly.

Then the DE algorithm iteratively refines these parameters to minimize the difference between the measured and simulated characteristics of the TFT. The DE algorithm is chosen due to its ability to handle a large number of parameters and its efficiency in escaping local minima, which is critical for an accurate and optimal solution in this case. Here the user can select between two differential evolution algorithms, i.e., "`scipy.optimize.differential_evolution`", which is a free open source Python library and takes bounds and iteration number as the arguments, while other "`differential_evolution_custom`" function is a custom function which lets user modify parameters like iteration numbers, population size, mutation and crossover probability factors.

The optimization is done in following steps:

1. First the *generate_netlist* function creates two netlist files (input.cir and output.cir) for NGSPICE, which define the circuit and the model parameters for the TFT.
2. The parameters are passed as arguments to the function. Then the *run_ngspice* function runs the NGSPICE simulator with the generated net-lists and extracts the simulation results.
3. Then the *deviation* function calculates the root mean square error (RMSE) and mean absolute error (MAE) between the simulated and measured data. This function serves as the objective function for the optimization process, which aims to minimize these errors.
4. Next step is optimization, where the DE function takes the objective function, the bounds for the parameters, and other DE parameters as inputs. It starts with an initial guess for the parameters and iteratively adjusts them to minimize the deviation calculated by the previous function. It then returns the best solution found and its fitness value (i.e., the value of the *deviation* function).
5. Finally the best parameters are found, and simulated graphs visualized to see if simulated curves are within 20% bounds of the measured curves.

5.3. Results



Figure 5.4.: TFT Parameter Extraction Graphical User Interface.

Finally, the code was exported to a Graphical User Interface (GUI). The GUI has five sections:

- **File Selection:** where the measured input and output characteristic files is selected in .dat format.
- **MOSFET Parameters:** where the MOSFET properties like thickness of oxide layer, dielectric constant, width, and length are extracted from the input .dat file automatically.
- **Optimization Algorithm:** here the optimization algorithm is selected, i.e., either Python SciPy library (Number of iteration as argument), or Differential Evolution Custom (Number of iterations, population size, mutation factor, crossover probability as arguments). It also has the run button which when pressed runs the algorithm.
- **Output Predicted Parameters:** where the final predicted parameters are displayed after the run button is pressed. It also has "Copy Model Parameter Set" button which copies final parameters in the following example format:

```
.model testfet nmos level=3
+is=0 u0=3.2210727210844734 vto=3.3479248993296475
+gamma=0 pb=0 fc=0 tpg=0 tox=65e-9 eta=3.012268141
+nfs=1583932371229.9458 nsub=1446504158896810.8
+kappa=0.59480011721806 theta=-0.015 lambda=1.8890e-07
```

This can be directly used in .cir file for SPICE simulations.

- **Output Graphs:** this displays input characteristic curve, Output characteristic curve, initial V_{th} calculation, measured and simulated input characteristic curve for comparison, and measured and simulated input characteristic curve for comparison.

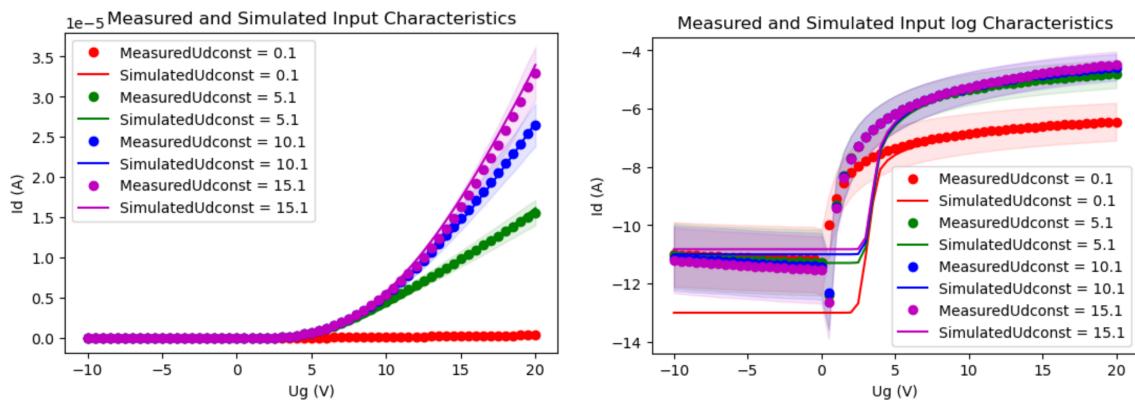
5.3.1. Extraction for Sample A1 10x10 (Enhancement type)

This section shows simulation result for A1 sample of $10\mu\text{m} \times 10\mu\text{m}$ dimension and IGZO thickness of 44.5 nm. Differential evolution custom function was used with population size as 50, mutation factor as 0.5, crossover probability of 0.7. The main difference can be seen when iteration numbers were set and population size was changed.

- **Number of iteration = 1 and population size = 50:** Figure 5.5, shows the simulated and measured curves comparison, where it can be observed that even with one iteration the algorithm already performs a lot better matching the measured curves. And table 5.2, shows the initial and final predicted parameters. Similarly, table 5.1, shows the value of deviation function for one iteration.

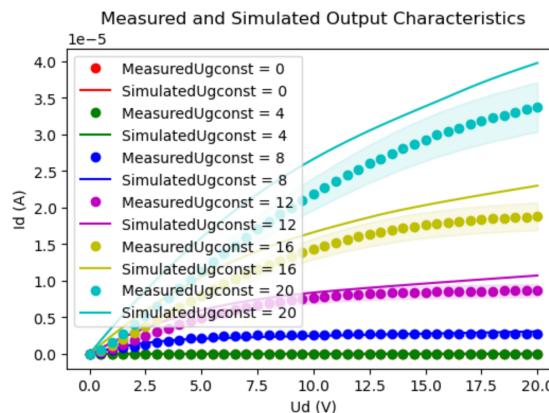
Table 5.1.: Best Fitness per Generation

Generation	Best Fitness
1	0.0003172



(a) Measured and Simulated Input characteristics

(b) Measured and Simulated Input log characteristics



(c) Measured and Simulated Output characteristics

Figure 5.5.: Input characteristics curve (a), Input log characteristics curve (b), and output characteristics curve (c) A1 (10x10) iteration = 1 and population size = 50.

Table 5.2.: Initial and Final Values for Parameters (1 iteration)

Parameter	Initial Guess	Final Values
Vth	2.96 V	3.4611 V
u0	3.497 cm ² /Vs	3.1937 cm ² /Vs
kappa	0.595	0.5949
eta	3	2.9498
LAMBDA	1.889e-07 V ⁻¹	1.8836e-07 V ⁻¹
nfs	2e12	1.1592e12
nsub	1e15	9.1849e14
theta	-0.02	-0.015

- **Number of iterations = 10:** Figure 5.6, shows the simulated and measured curves comparison, where it can be observed that with 10 iterations the algorithm refines the matching of the output characteristic curve. And table 5.4, shows the final predicted parameters. Similarly table 5.3, shows the value of deviation function for one iteration, where it can be seen that with further iterations the deviation value refines.

Table 5.3.: Best Fitness per Generation

Generation	Best Fitness
0	0.000317
4	0.000208
10	0.000208

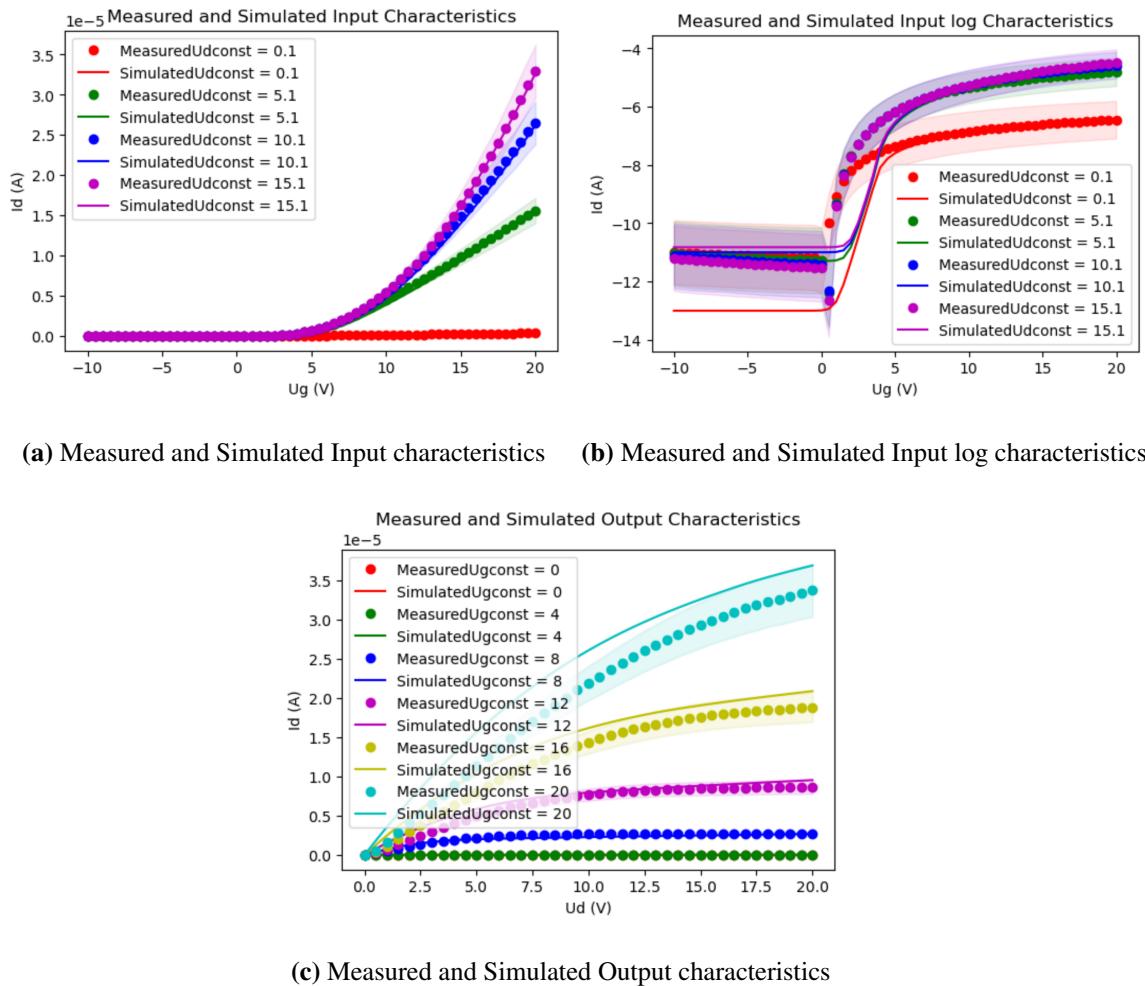


Figure 5.6.: Input characteristics curve (a), Input log characteristics curve (b), and output characteristics curve (c) A1 (10x10) iteration = 10 and population size = 50.

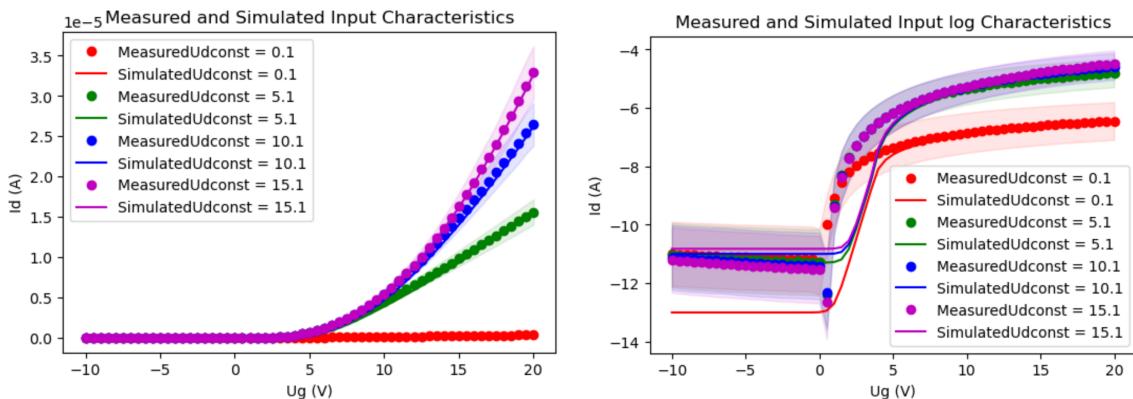
Table 5.4.: Initial and Final Values for Parameters (10 iterations)

Parameter	Initial Guess	Final Values
Vth	2.96 V	3.4611 V
u0	3.497 cm ² /Vs	3.1383 cm ² /Vs
kappa	0.595	0.5948
eta	3	2.9
LAMBDA	1.889e-07 V ⁻¹	2.834e-07 V ⁻¹
nfs	2e12	3e12
nsub	1e15	1.5e15
theta	-0.02	-0.015

- **Number of iterations = 100:** Figure 5.7, shows the simulated and measured curves comparison for 100 iterations. And table 5.6, shows the initial and final predicted parameters.

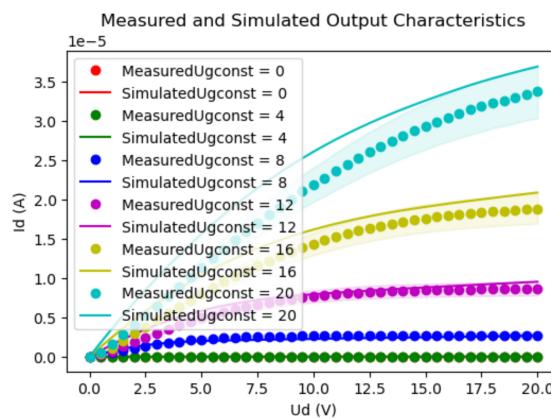
Table 5.5.: Evolution of Function Through 100 Steps

Generation (iteration)	Best Fitness (MSE)
0	0.000317
10	0.0002082
100	0.0002082



(a) Measured and Simulated Input characteristics

(b) Measured and Simulated Input log characteristics



(c) Measured and Simulated Output characteristics

Figure 5.7.: Input characteristics curve (a), Input log characteristics curve (b), and output characteristics curve (c) A1 (10x10) iteration = 100 and population size = 50.

Table 5.6.: Initial and Final Values for Parameters (100 iterations)

Parameter	Initial Guess	Final Values
Vth	2.96 V	3.4611 V
u0	3.497 cm ² /Vs	3.1383 cm ² /Vs
kappa	0.595	0.5948
eta	3	2.9
LAMBDA	1.889e-07 V ⁻¹	2.8286e-07 V ⁻¹
nfs	2e12	3e12
nsub	1e15	1.5e15
theta	-0.02	-0.015

And table 5.5, shows the value of deviation function for 100 iterations, where it can be seen that after 10 iterations there is not much of change in fitness values. Hence, with just 10 iterations the program produces comparable results and good simulation outputs.

5.3.2. Extraction for Sample A1 10x5 (Enhancement type)

This section shows simulation result for A1 sample of $10\mu\text{m} \times 5\mu\text{m}$ dimension and IGZO thickness of 44.5 nm. Differential evolution custom function was used with population size as 50, mutation factor as 0.5, crossover probability of 0.7, and number of iterations as 10.

Figure 5.8, shows the simulated and measured curves comparison, where it can be observed that the optimization performs worse for length values below $10\mu\text{m}$. And table 5.7, shows the initial and final predicted parameters.

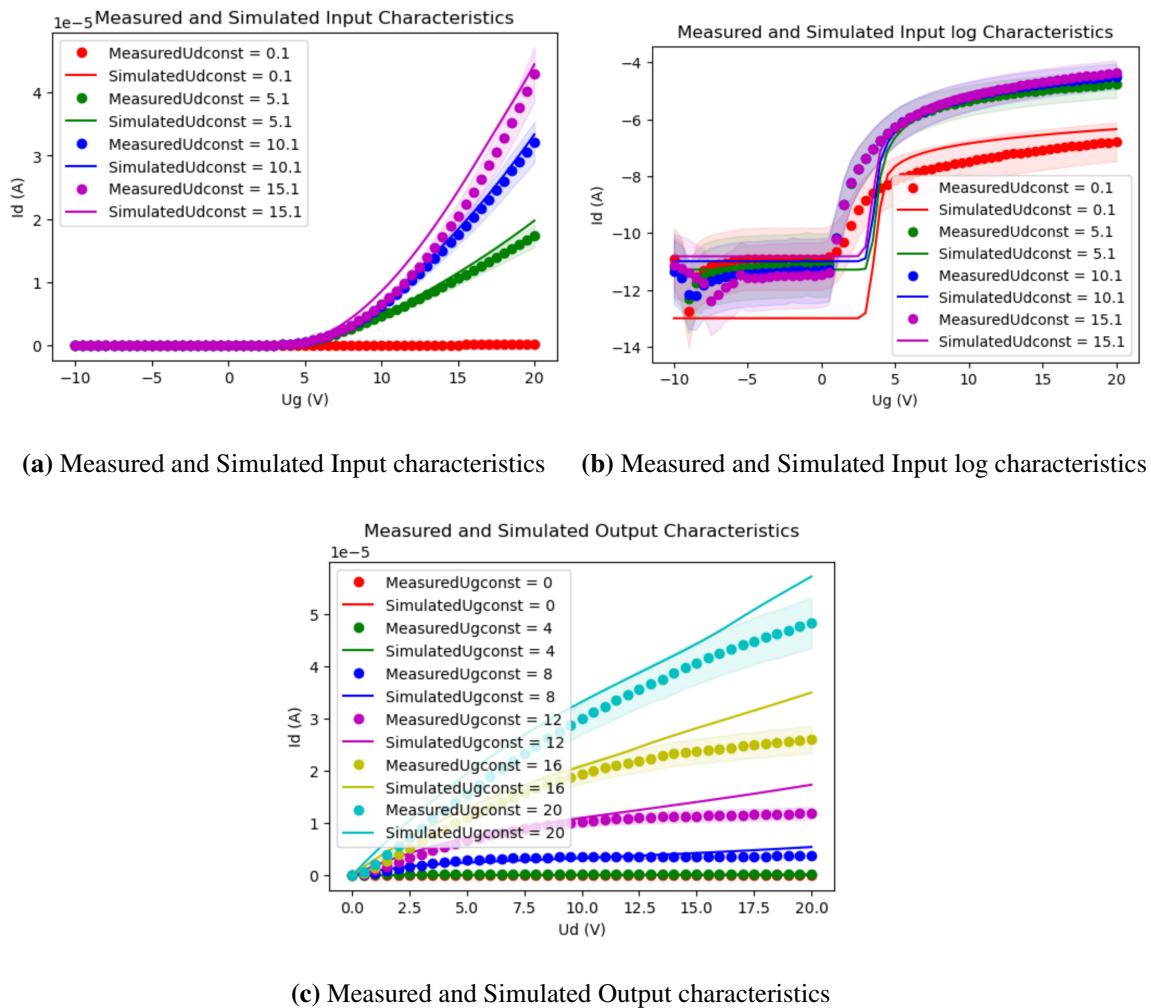


Figure 5.8.: Input characteristics curve (a), Input log characteristics curve (b), and output characteristics curve (c) A1 (10x5).

Table 5.7.: Initial and Final Values for Parameters

Parameter	Initial Guess	Final Values
Vth	3.44 V	3.9376 V
U0	2.29337 cm ² /Vs	2.0067 cm ² /Vs
kappa	0.484	0.4835
eta	3	2.9
LAMBDA	3.39733e-07 V ⁻¹	4.5765e-07 V ⁻¹
nfs	2e12	2656911005042.6074
nsub	1.5e15	1.5e15
theta	-0.02	-0.015

5.3.3. Extraction for Sample A1 10x20 (Enhancement type)

This section shows simulation result for A1 sample of $10\mu\text{m} \times 20\mu\text{m}$ dimension and IGZO thickness of 44.5 nm. Python's SciPy Differential evolution function was used with number of iterations as 10.

Figure 5.9, shows the simulated and measured curves comparison, where it can be observed that the simulated curves are within the bounds of measured curves. And table 5.8, shows the initial and final predicted parameters.

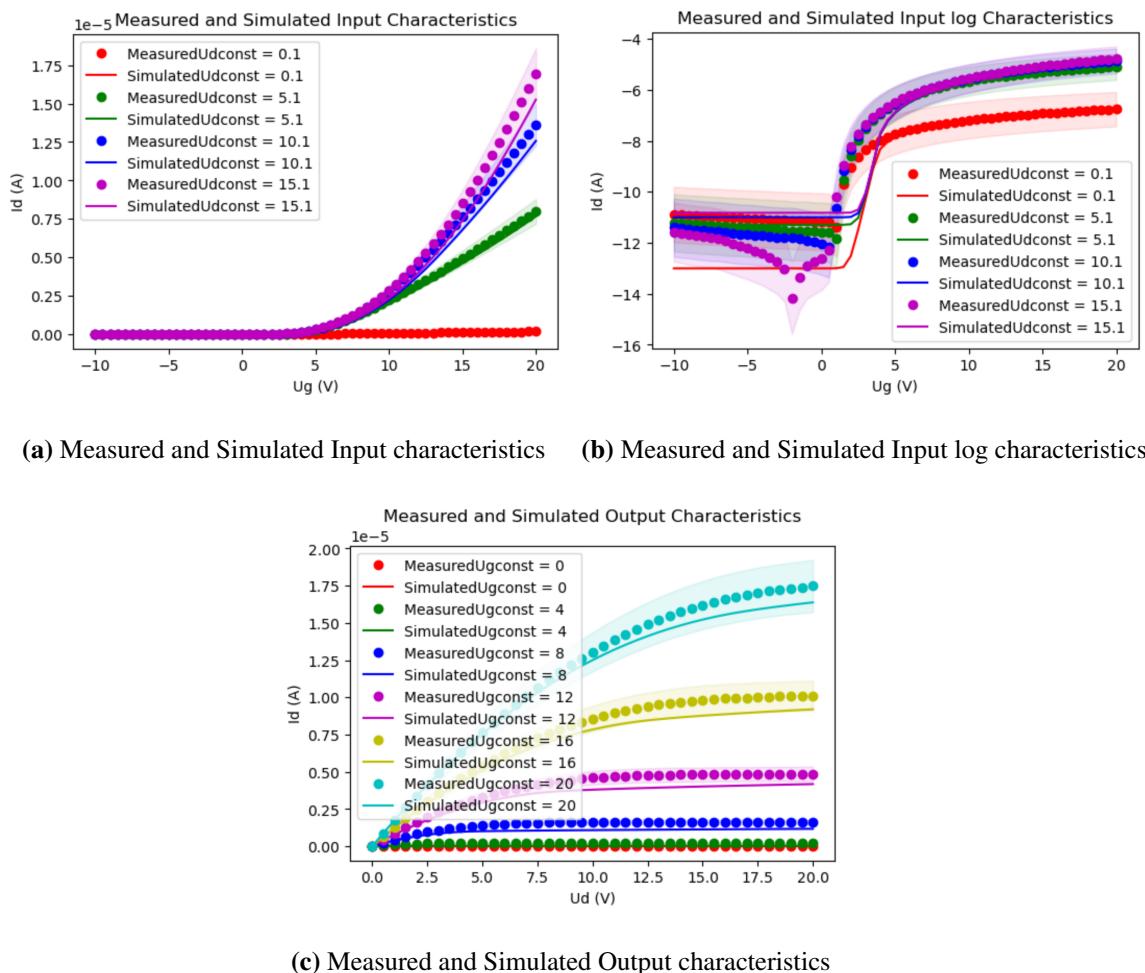


Figure 5.9.: Input characteristics curve (a), Input log characteristics curve (b), and output characteristics curve (c) A1 (10x20).

Table 5.8.: Initial and Final Values for Parameters

Parameter	Initial Guess	Final Values
Vth	2.93 V	3.3899 V
U0	3.6386 cm ² /Vs	3.2173 cm ² /Vs
kappa	0.722	0.8627
eta	3	3.1118
LAMBDA	5.1790e-08 V ⁻¹	6.1754e-08 V ⁻¹
nfs	2e12	2004573510235.264
nsub	1e15	1.5383e15
theta	-0.02	-0.012

5.3.4. Extraction for Sample D1 10x10 (Depletion type)

This section shows simulation result for D1 sample of $10\mu\text{m} \times 10\mu\text{m}$ dimension and IGZO thickness of 85 nm. Python's SciPy Differential evolution function was used with number of iterations as 10.

Figure 5.10, shows the simulated and measured curves comparison, where it can be observed that the simulated curves almost matches the measured curves. And table 5.9, shows the initial and final predicted parameters, where it can be observed that the Vth is negative and hence the TFT is a depletion type MOSFET.

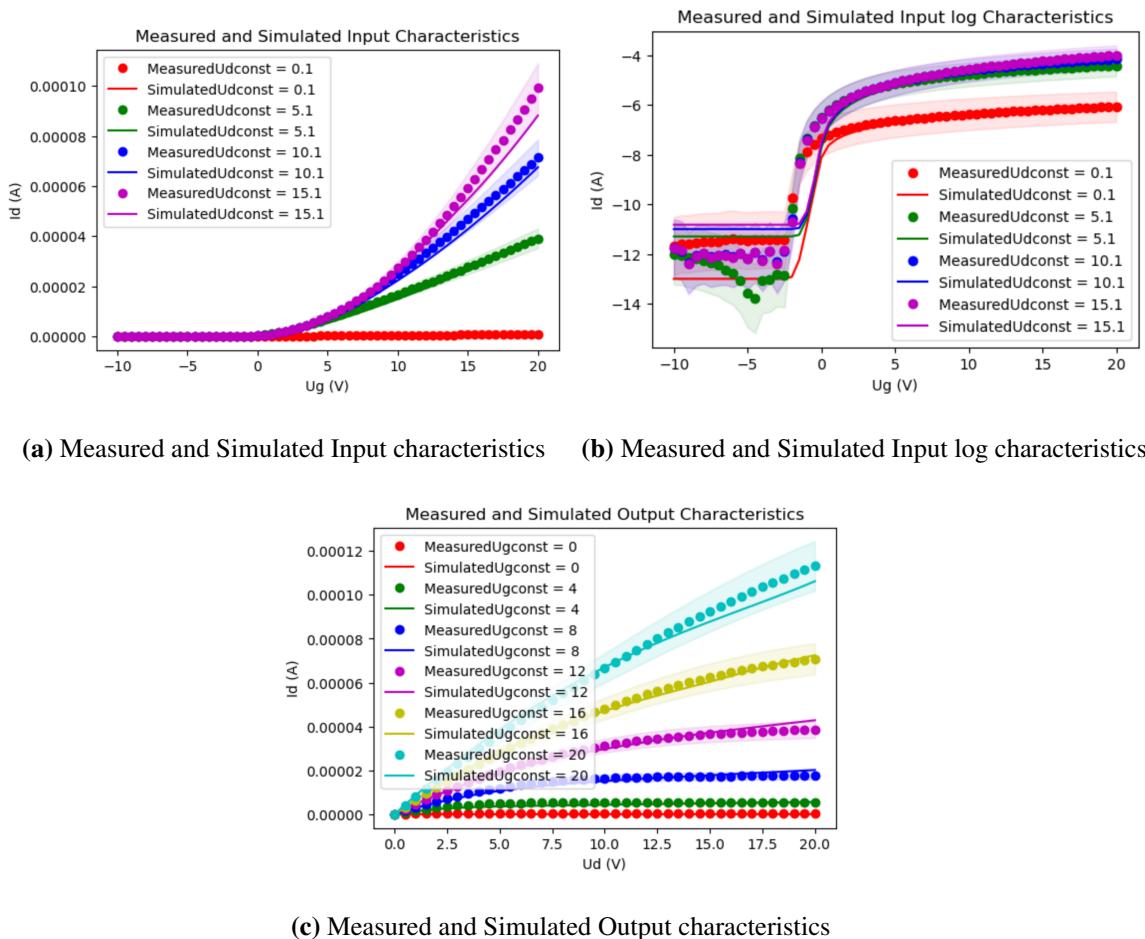


Figure 5.10.: Input characteristics curve (a), Input log characteristics curve (b), and output characteristics curve (c) D1 (10x20).

Table 5.9.: Initial and Final Values for Parameters

Parameter	Initial Guess	Final Values
Vth	-0.78 V	-0.27629 V
U0	7.0447 cm ² /Vs	6.1641 cm ² /Vs
kappa	0.688	0.9534
eta	3	2.96321
LAMBDA	1.0314e-06 V ⁻¹	1.4255e-06 V ⁻¹
nfs	2e12	1661961142492.1772
nsub	1e15	1.1473e15
theta	-0.02	-0.012

Finally, it can be observed that the optimization algorithm performs well for enhancement and depletion type TFTs, and also can perform well with other architecture or materials of transistors, as the code uses basic transistor equations. This concludes this chapter on Methodology and Results. The next chapter will discuss about the conclusion from the results, and outlines future work that can be implemented.

6. Discussion, Conclusion, and Future Work

This chapter provides a summary of the work conducted, the conclusions drawn from the study, and recommendations for future research in the area of Indium Gallium Zinc Oxide (IGZO) Thin Film Transistor (TFT) parameter extraction.

6.1. Summary and Conclusion

The purpose of this study was to compare different optimization techniques for parameter extraction of IGZO TFTs and determine the most effective approach. The importance of parameter extraction lies in its ability to predict the electrical behaviour of TFTs, allowing for the design and performance analysis of TFT-based circuits.

From the extensive study, it has been determined that Differential Evolution (DE) emerged as the best optimization technique for this application. The results showed that DE offered the most reliable and consistent results. It was also observed that with just 10 iterations DE performed reliably producing consistent output. This demonstrated an optimal balance between computational efficiency and the quality of the extracted parameters. Additionally, DE performed best with devices that had a channel length above $10 \mu\text{m}$. This may be attributed to the TFT characteristics, and SPICE simulation to be more suited to the larger device sizes. And the algorithm is designed to work with any MOS architecture and is not limited to IGZO TFT.

In conclusion, the Differential Evolution algorithm is the most suitable optimization technique for parameter extraction of TFTs, particularly for devices with a channel length above $10 \mu\text{m}$ and for an optimal iteration count of 10.

6.2. Future Work

Several areas can be explored further to build upon this research. The first being, DE algorithm itself; while it proved the most effective in this study, there is scope for enhancing its performance. DE's bound space can be improved, which could yield more precise and efficient parameter extraction. Also other mutation and crossover schemes can be utilized to check the performance.

Moreover, the application of alternative optimization techniques, such as Particle Swarm Optimization (PSO), could be investigated. PSO may provide insights and results different from DE, providing better understanding of the parameter extraction process. A combination of both can also provide better performance.

Another aspect for future research lies in the development of a batch mode for the parameter extraction process. Such a system could simultaneously process multiple TFTs, potentially greatly improving the efficiency of the extraction process and facilitating the high-throughput analysis of TFT devices.

A. Programmcode

There were two Python files created as NGSPICE configurations needed to be defined OS specifically. The program file can be found as *TFT_Parameter_Extraction_Windows.py* and *TFT_Parameter_Extraction_Linux.py*.

And "*TFT_Parameter_Extraction v1.2.exe*" executable application for windows can directly be used for parameter extraction.

Bibliography

- [1] KAMIYA, TOSHIO; NOMURA, KENJI; HOSONO, HIDEO: Origins of high mobility and low operation voltage of amorphous oxide TFTs: Electronic structure, electron transport, defects and doping. *Journal of Display Technology, IEEE*, 5.7: 273-288, 2009.
- [2] UTMEL: How Do Transistors Work. <https://www.utmel.com/blog/categories/transistors/how-do-transistors-work>, 2023.
- [3] MALVINO, ALBERT PAUL, DAVID J. BATES, AND PATRICK E. HOPPE: Electronic Principles. *Glencoe*, 1993.
- [4] ELECTRONICS FOR YOU: MOSFET Basics, Working, Applications. <https://www.electronicsforu.com/technology-trends/learn-electronics/mosfet-basics-working-applications>, 2023.
- [5] AUTODESK: What is Field Effect Transistor (FET). <https://www.autodesk.com/products/fusion-360/blog/what-is-field-effect-transistor-fet/>, 2023.
- [6] RAKESH AXE: Characteristics Of MOSFET. <https://rakeshaxe.blogspot.com/2023/03/Characteristics-Of-mosfet.html>, 2023.
- [7] TECHOPEDIA: What Does Thin-Film Transistor (TFT) Mean. <https://www.techopedia.com/definition/3630/thin-film-transistor-tft>, 2023.
- [8] BROTHERTON, S. D.: Introduction to Thin Film Transistors: Physics and Technology of TFTs. *Springer Science & Business Media*, 1996.
- [9] WIKIPEDIA: Indium gallium zinc oxide. https://en.wikipedia.org/wiki/Indium_gallium_zinc_oxide, June 2023.
- [10] ZHU, YING, ET AL.: Indium gallium zinc oxide thin-film transistors: Materials, devices, and applications. *Journal of Semiconductors*, 42.3: 031101, 2021.

- [11] KLAUK, H.: Organic thin-film transistors. *Chemical Society Reviews*, 39(7): 2643-2666, 2010.
- [12] GE, GE: Design and optimization of electric circuits based on depletion-type and enhancement-type indium gallium zinc oxide transistors. *University of Stuttgart, Institute for Large Area Microelectronics*, 2022.
- [13] OUYANG, YUNYI: Investigation of the influence of process steps on the properties of indium gallium zinc oxide semiconductor transistors. *University of Stuttgart, Institute for Large Area Microelectronics*, 2022. Supervised by Yannick Schellander and Florian Kleber.
- [14] DR.-ING. PATRICK SCHALBERGER: Personal Communication. *Dual-gate TFTs often exhibit improved performance, such as higher mobility and stability*, 2023.
- [15] BHATTACHARYYA, A. B.: Compact MOSFET Models for VLSI Design. *Wiley-IEEE Press*, 2010.
- [16] BARTOLO, N. G. S.: Parameter Extraction, Modelling and Circuit Design for Electrolyte-Gated Transistors on Paper. Doctoral Dissertation, 2018.
- [17] DR.-ING. PATRICK SCHALBERGER: Personal Communication. *If MoTa comes into direct contact with IGZO, oxidation and oxygen vacancies arise at the MoTa-IGZO junction, causing the transistor to malfunction*, 2023.
- [18] DR.-ING. PATRICK SCHALBERGER: Personal Communication. *The substrate is then annealed overnight in a 250°C oven to remove any moisture within the substrate as the oxide layer absorbs moisture from the environment and can affect the transistor characteristics*, 2023.
- [19] NG, A.: Machine Learning. *Coursera*, n.d., <https://www.coursera.org/learn/machine-learning>.
- [20] SKAHILL, BRIAN E., AND JOHN DOHERTY: Efficient accommodation of local minima in watershed model calibration. *Journal of Hydrology*, 329.1-2: 122-139, 2006.
- [21] MATHWORKS: What Is the Genetic Algorithm? MathWorks - Makers of MATLAB and Simulink, 2023. <https://de.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>.

- [22] TAM, A.: A Gentle Introduction to Particle Swarm Optimization. *Optimization*, September 16, 2021. <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/>.
- [23] MACHINE LEARNING MASTERY: Differential Evolution from Scratch in Python. <https://machinelearningmastery.com/differential-evolution-from-scratch-in-python/>, 2023.
- [24] DAS, SWAGATAM, AND PONNUTHURAI NAGARATNAM SUGANTHAN: Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15.1: 4-31, 2010.
- [25] PABLO RMIER: A Tutorial on Differential Evolution with Python. <https://pablormier.github.io/2017/09/05/a-tutorial-on-differential-evolution-with-python/>, 2023.