

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Spracovanie a vizualizovanie dát
z kapacitných meraní štruktúr na báze GaN**

Diplomová práca

2020

Bc. Ivana Patlevičová

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Spracovanie a vizualizovanie dát
z kapacitných meraní štruktúr na báze GaN**

Diplomová práca

Študijný program: Informatika
Študijný odbor: 9.2.1. Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: Ing. Miroslav Michalko, PhD.
Konzultant: prof. Ing. Ľubica Stuchlíková, PhD.

Košice 2020

Bc. Ivana Patlevičová

Názov práce: Spracovanie a vizualizovanie dát z kapacitných meraní štruktúr na báze GaN

Pracovisko: Katedra počítačov a informatiky, Technická univerzita v Košiciach

Autor: Bc. Ivana Patlevičová

Školiteľ: Ing. Miroslav Michalko, PhD.

Konzultant: prof. Ing. Ľubica Stuchlíková, PhD.

Dátum: 4. 5. 2020

Kľúčové slová: Python, DLTFs, GaN, PyQt5, PyQtGraph

Abstrakt: Táto práca sa zaoberá tvorbou aplikácie na zvýšenie efektívnosti evaluácie dát nameraných na progresívnych polovodičových štruktúrach na báze GaN, metódou Fourierovej spektroskopie hlbokých hladín (DLTFs). V analytickej časti je stručne charakterizované matematicko-fyzikálne pozadie metódy DLTFs, polovodičový materiál GaN a existujúce softvérové riešenia. Na základe výstupov analýzy s využitím štandardov softvérového inžinierstva bol pripravený návrh aplikácie. Softvérové riešenie bolo implementované a otestované. Výstupom práce je funkčná aplikácia, ktorá poskytuje užívateľsky priaznivú podobu vizualizácie nameraných výsledkov a základnú evaluáciu.

Thesis title: Processing and visualization of data from capacity measurements of structures based on GaN

Department: Department of Computers and Informatics, Technical University of Košice

Author: Bc. Ivana Patlevičová

Supervisor: Ing. Miroslav Michalko, PhD.

Tutor: prof. Ing. Lubica Stuchlíková, PhD.

Date: 4. 5. 2020

Keywords: Python, DLTFs, GaN, PyQt5, PyQtGraph

Abstract: This work is focused on the development of application with aim to increase the efficiency of data evaluation measured on progressive semiconductor structures based on GaN, the method Deep-Level Fourier Spectroscopy (DLTFs) is used. The analytical part briefly characterizes the mathematical and physical background of the DLTFs method, semiconductor material GaN and existing software solutions. Based on the outputs of the analysis an application design was prepared, using software engineering standards. The software solution was implemented and tested. The output of the work is a functional application that provides a user-friendly form of visualization of measured data and basic evaluation.

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

Spracovanie a vizualizovanie dát z kapacitných meraní štruktúr na báze GaN

Processing and visualization of data from capacity measurements of structures based on GaN

Študent: **Bc. Ivana Patlevičová**

Školiteľ: **Ing. Miroslav Michalko, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce: **prof. Ing. Ľubica Stuchlíková, PhD.**

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Oboznámte sa so základmi matematickej analýzy spracovania nameraných dát metódou Fourierova spektroskopia hlbokých hladín.
2. Navrhňte aplikáciu pre automatizovanú analýzu veľkého množstva dát získaných z experimentálnych meraní z hľadiska: vybraných parametrov, implementácie knižnice identifikovaných pascí a spracovania kapacitných transient s cieľom dosiahnutia potrebnej úrovne exponenciality a použitia vybraných dekonvolučných algoritmov.
3. Vytvorte riešenie pre automatizovanú analýzu veľkého množstva dát získaných z experimentálnych meraní metódou Fourierova spektroskopia hlbokých hladín.
4. Urobte optimalizáciu aplikácie, otestujte ju a vypracujte dokumentáciu v zmysle katedrových štandardov.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 04.05.2020

Dátum zadania diplomovej práce: 31.10.2019



prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som záverečnú prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice, 4.5.2020

.....

Vlastnoručný podpis

Podakovanie

Na tomto mieste by som chcela úprimne poďakovať môjmu vedúcemu práce Ing. Miroslavovi Michalkovi, PhD, ako aj svojej konzultantke prof. Ing. Ľubici Stuchlíkovej, PhD. a jej študentovi Ing. Jakubovi Drobnému za ich čas, cenné rady a odborné vedenie pri tvorbe tejto práce.

Obsah

Motivácia	1
1 Formulácia úlohy	3
2 Štruktúry na báze GaN	5
3 Hlboké energetické hladiny	7
3.1 Elektricky aktívne poruchy	8
3.2 Energetické hladiny polovodičov	9
3.3 Procesy v polovodičoch súvisiace s emisiou a záchyтом	10
3.4 Identifikovanie poruchy	12
3.5 Teplotná korekcia Arrheniovej závislosti	14
4 Fourierova spektroskopia hlbokých hladín	15
4.1 Porovnanie DLTS a DLTFs	15
4.2 Fourierova transformácia	17
4.2.1 Definícia diskkrétnej Fourierovej transformácie	19
4.2.2 Predpis lichobežníkového pravidla	19
4.3 Priama a nepriama evaluácia	20
5 Systém pracoviska DL8000	21
5.1 Funkcie systému	21
5.2 Meracie pracovisko	22
5.3 Softvérové vybavenie	23
5.4 Vyhodnotenie analýzy systému	27
6 Metodológia a návrh	28
6.1 Opis domény a konceptuálneho modelu	28

6.2	Prípady použitia a proces identifikácie objektov a akcií	29
6.2.1	Povolená postupnosť stavov programu	30
6.2.2	Konceptuálny model programu	32
6.3	Návrh používateľského rozhrania	32
6.3.1	Sled obrazoviek	34
6.3.2	Iteratívne prototypovanie	34
6.4	Výber platformy a technológií	35
7	Implementácia programu pre spracovanie a vizualizovanie dát	39
7.1	Štruktúra vstupných dát	39
7.2	Priblíženie softvérového riešenia	41
7.2.1	Štruktúra projektu	41
7.2.2	Používateľské rozhranie a jeho ohraničenie	42
7.2.3	Výber vstupných dát, kontrola konzistencie a štruktúry . . .	44
7.2.4	Vizualizácia vstupných súborov	45
7.2.5	Evaluácia vstupu	46
7.2.6	Export analyzovaných a vizualizovaných dát	47
8	Výsledky práce	49
9	Ďalšia práca	53
10	Záver	55
	Literatúra	57
	Zoznam skratiek	60
	Zoznam príloh	63
A	Používateľská príručka	64
A.1	Opis aplikácie pre vizualizáciu a evaluáciu dát	64
A.2	Obsah CD média	64
A.3	Používanie aplikácie	65
A.3.1	Definovanie vstupných meraní	66
A.3.2	Vizualizácia Tempscan meraní	68
A.3.3	Evaluácia Transient meraní	70

A.3.4	Exportovanie vizualizovaných a evaluovaných dát	72
A.4	Spustenie aplikácie	73
B	Systémová príručka	74
B.1	Funkcie aplikácie a analyzovanie riešenia	74
B.2	Priblíženie aplikácie	74
B.2.1	Popis softvérového riešenia	74
B.2.2	Štruktúra projektu a popis tried	75
B.2.3	Popis vstupných dát	80
B.2.4	Preklad a spustenie aplikácie	81
B.3	Vyhodnotenie riešenia	82

Zoznam obrázkov

2.1	Principiálna schéma AlGaIn/GaN HEMT (Gate, Source a Drain elektródy tranzistora) (Samco, b. r.)	6
3.1	Hlboké energetické hladiny spôsobené prítomnosťou intrinzických defektov, dislokácií a ich komplexmi v zakázanom pásme energetického pásmového diagramu polovodiča GaN. Nulová energia zodpovedá hornému okraju valenčného pásma, hrubá čierna čiara zodpovedá dolnému okraju vodivostného pásma (Meneghini et al., 2017)	7
3.2	Princíp DLTS (Pohorelec, 2016)	8
3.3	Energetický diagram so znázornenými energetickými hladinami (E_i je intrinzická hladina)	9
3.4	Proces emisie a zachytenia hlbokou hladinou	11
4.1	Vstupné meracie parametre metódy DLTFs: záverné napätie U_R , plniace napätie U_P , doba plniaceho impulzu t_P , šírka periódy snímanej kapacitnej odozvy T_W a kapacitná transienta (Drobný, 2018)	16
4.2	Namerané spektrum DLTFs s kapacitnými transientami vo vybraných bodoch (Pohorelec, 2016)	16
4.3	Príklad využitia priamej evaluácie DLTFs. Body Arrheniovej závislosti (vpravo) sú vypočítané podľa nameraných kapacitných transient pri rôznych teplotách (vľavo) a vopred definovaných evaluačných parametroch: hodnotiaci trieda 60, rozptyl časovej konštanty 10% a maximálny teplotný rozdiel 10K (Kósa, 2016)	20
4.4	Príklad DLTFs nepriamej evaluácie. Každý bod Arrheniovej krivky (vpravo) je vypočítaný pomocou emisnej časovej konštanty τ zodpovedajúcej maximu alebo minimu korelačnej krivky DLTFs (vľavo) (Kósa, 2016)	20

5.1	Výskumné a meracie pracovisko	22
5.2	Pohľad na kryostat s nakontaktovanou vzorkou	23
5.3	Softvér Dlts 4.1, 2016-02-14, 2016-11-20 by S. Weiss (<i>PhysTech</i> , b. r.)	24
5.4	Porovnanie dvoch meraní softvérom a zobrazenie príslušných pa- rametrov merania (<i>PhysTech</i> , b. r.)	25
5.5	Dôkladnejšia vizualizácia meraní iným softvérom (Kósa et al., 2016)	25
5.6	Arrheniova závislosť (<i>PhysTech</i> , b. r.)	26
5.7	Nameraná transienta (<i>PhysTech</i> , b. r.)	26
6.1	Diagram reprezentujúci koncept domény	29
6.2	Stavový diagram programu	31
6.3	Koncept programu vo forme diagramu interakcie hlavných tried .	33
6.4	Diagram sledu obrazoviek	35
7.1	Softvér na preklad dát	40
7.2	Štruktúra projektu a obsah jednotlivých balíčkov	42
7.3	Pridané vstupné súbory na úvodnej obrazovke	43
7.4	Potvrdenie ukončenia aplikácie	43
7.5	Výstraha o nepridaných vstupných súboroch	44
7.6	Výstraha o nekorektnej štruktúre vstupného súboru	45
7.7	Výstraha o pokuse o pridanie duplicitného súboru	45
7.8	Výstraha o pokuse o odstránenie súboru bez jeho zvolenia	45
7.9	Záložka s vizualizáciou Tempscan súborov a zobrazenie ich para- metrov v tabuľke	46
7.10	Záložka s evaluáciou	47
7.11	Záložka s možnosťami exportovania grafu do súboru	48
8.1	Vizualizácia dát pomocou programu Origin	50
8.2	Finálna vizualizácia tempscan súborov a ich parametrov	51
8.3	Evaluácia dát	52
A.1	Záložka s informáciami	66
A.2	Potvrdenie ukončenia aplikácie	66
A.3	Záložka bez zadaných vstupných súborov	67
A.4	Tempscam dáta na vstupe	68
A.5	Transient dáta na vstupe	68

A.6	Hlásenia	69
A.7	Definovanie vstupných Tempscan dát	70
A.8	Vizualizácia Tempscan dát	70
A.9	Definovanie vstupných Transient dát	71
A.10	Hlásenie o nezadaných alebo zle formátovaných vstupných parametroch	71
A.11	Výsledná evaluácia dát	72
A.12	Exportovanie priamo z grafického komponentu	72
A.13	Exportovanie z príslušnej záložky	73
B.1	Softvér na preklad dát	80
B.2	Štruktúra projektu a obsah jednotlivých balíčkov	81

Zoznam tabuliek

2.1	Porovnanie fyzikálnych parametrov vybraných polovodičov a GaN pri $T = 300\text{K}$ (Ozpineci et al., b. r.)	5
6.1	Porovnanie vybraných rámcov	37

Motivácia

Elektronika súčasnej doby si vyžaduje polovodičové súčiastky, ktoré majú špičkovú kvalitu, a keďže nároky na pracovné frekvencie a výkony neustále stúpajú, trvalý vývoj v oblasti polovodičových štruktúr je nevyhnutný. Zlepšenie účinnosti výkonových elektronických zariadení je rozhodujúce napríklad pre zníženie strát energie pri spínaní. Predpokladá sa, že celková čistá spotreba elektrickej energie na svete sa medzi rokmi 2003 a 2030 zdvojnásobí zo 14 781 miliárd kilowatthodín na 30 116 miliárd kilowatthodín, čo je priemerný ročný nárast 2,7% (Salbu et al., 2008). Ak nahradíme štandardný polovodič kremík vo výkonových zariadeniach širokopásmovým polovodičom nitridom gália („Gallium Nitride“) GaN, je možné dosiahnuť zníženie energetických strát o 10% až 25% (Chen, 2019). Zariadenia na báze GaN sú účinnejšie, sú schopné pracovať vo vysokých pracovných teplotách, sú odolnejšie voči žiareniu ako zariadenia na báze kremíka.

Avšak elektricky aktívne poruchy v súčiastkách na báze GaN späté s procesom výroby a spracovaním, alebo čistotou jednotlivých vrstiev štruktúr vykazujú negatívny vplyv na elektrické vlastnosti a tým aj na kvalitu zariadení. Preto je veľmi dôležité určiť parametre týchto porúch a pochopiť okolnosti ich vzniku a vplyv na elektrofyzikálne vlastnosti a parametre súčiastok. Fourierova spektroskopia hlbokých hladín (Deep-Level Transient Fourier Spectroscopy - DLTFs) (Weiss; Kassing, 1988) patrí medzi popredné metódy v oblasti identifikácie elektricky aktívnych porúch, pričom vykazuje vysokú citlivosť a spoľahlivosť. Pravdaže pri vyhodnocovaní zložitých štruktúr je pre dosiahnutie požadovanej úrovne spoľahlivosti nevyhnutné zrealizovať veľké množstvo časovo náročných meraní. Následné vyhodnocovanie taktiež predstavuje časovo, ale aj softvérovo náročný proces. Súčasne využívané licencované softvérové riešenie je optimalizované pre štandardné polovodiče ako je kremík, preto je potrebné navrhnuť a implementovať efektívnejšiu alternatívu.

Cieľom tejto diplomovej práce je návrh a implementácia efektívnejšej alternatívy procesu evaluácie dát spojených so základným výskumom emisných a zachytných procesov v progresívnych polovodičových štruktúrach na báze GaN. Výsledky evaluácie majú potenciál prispieť k pochopeniu procesov v štruktúrach a následnej optimalizácii technologického procesu ich prípravy.

Základným predpokladom je dôkladná analýza procesu evaluácie veľkého množstva meraní a s ním súvisiacimi teoretickými podkladmi. K motivácii prispieva aj nadviazanie na bakalársku prácu, v ktorej bolo implementované len parciálne riešenie vizualizácie nameraných výsledkov pre zjednodušenie procesu, ktorý sa využíva pri evaluácii nameraných výsledkov.

Ďalšou motiváciou je možnosť rozvoja v programovaní, pri používaní nových jazykov a technológií. Subjektívnou motiváciou je aj zdokonalenie sa a získanie nových skúseností v súvislosti s dátovou analýzou, ale aj oboznámenie sa s matematicko-fyzikálnym pozadím procesu. Súčasne je veľkou výzvou aj rozvoj kreativity pri procese návrhu a dizajnu používateľského rozhrania, pre zlepšenie použiteľnosti a užívateľského zážitku (User Experience - UX), čo licencovaný softvér neposkytuje.

1 Formulácia úlohy

V rámci tejto diplomovej práce by sme chceli vytvoriť aplikáciu, ktorá umožní dosiahnuť zvýšenie efektívnosti v procese vyhodnocovania výsledkov, nameraných pri experimentoch na polovodičových štruktúrach s využitím Fourierovej spektroskopie hlbokých hladín (DLTFS). Tieto experimentálne merania sú realizované za účelom skúmania kvality progresívnych polovodičových štruktúr na báze GaN v Ústave elektroniky a fotoniky (FEI STU v Bratislave), v spolupráci s ktorým je táto práca realizovaná.

Medzi parciálne ciele, ktoré by sme chceli naplniť realizáciou tejto práce patrí priblíženie pojmového aparátu týkajúceho sa experimentálnej činnosti a používanej metódy. Dôležité je tiež ozrejmiť význam získaných výsledkov a dopad pri praktickom využití skúmaných materiálov. Podstatné je tiež oboznámenie sa s matematicko-fyzikálnym pozadím procesu analýzy, čo je nevyhnutné pre dôkladné pochopenie problému a špecifikovanie požiadaviek a podmienok pre implementáciu programu.

Snažíme sa, aby jeho prínosom bolo značné zjednodušenie procesu a zníženie časovej náročnosti voči použitiu pôvodného licencovaného softvéru. Tento licencovaný softvér je taktiež predmetom analýzy, keďže chceme vymedziť priestor pre zlepšenie užívateľského zážitku (UX), hlavne prehľadným a ľahko použiteľným rozhraním, a tiež chceme podchytiť to, v čom je tento softvér neefektívny, alebo ťažkopádny pri používaní.

Na základe dôkladnej analýzy navrhujeme softvérové riešenie pre vyhodnotenie experimentálnych meraní s využitím DLTFS. Popíšeme riešenie z hľadiska architektúry, výberu platformy a jazyka, návrhu užívateľského rozhrania, resp. výberu knižníc a rámcov pre zjednodušenie realizácie riešenia.

Podľa návrhu vytvoríme a popíšeme proces realizácie softvérového riešenia. Aplikácia bude implementovaná tak, aby zahŕňala špecifikované vlastnosti a po-

stupy, ktoré zvolíme pri dizajne architektúry a užívateľského rozhrania. Implementujeme prostriedky pre zjednodušenie procesu a dosiahnutie vyššie uvedeníh čiastkových cieľov.

Pre zvýšenie presnosti a správnosti fungovania aplikácie uskutočníme testovanie užívateľmi, na základe čoho softvér optimalizujeme, aby vyhovoval požiadavkám používateľov priamo z domény. Pre naše softvérové riešenie následne vytvoríme potrebnú dokumentáciu.

2 Štruktúry na báze GaN

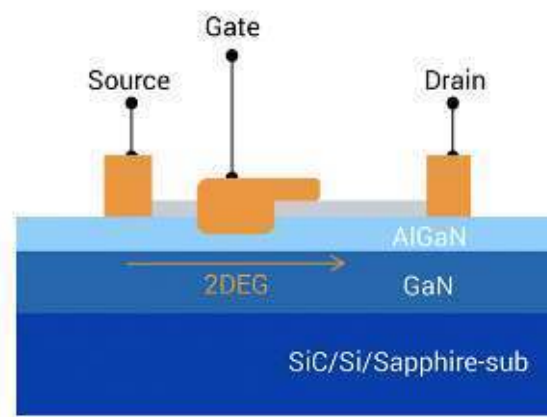
Kremík, najčastejšie používaný polovodič v elektronike, je ľahko dostupný a lacný, a zároveň kremíkové technológie sú už vynikajúco zvládnuté a dokonale odladené, takže ho dokážeme pripraviť veľmi čistý a so zanedbateľnou koncentráciou porúch. V oblasti vysokých frekvencií, výkonov a účinností, však kremík začína dosahovať svoje fyzikálne limity.

Širokopásmový polovodič GaN, teda prichádza ako perspektívny materiál a nástupca kremíka pre svoje vlastnosti (Tab. 2.1), ktoré spĺňajú rastúce nároky na súčiastky v aplikáciách určených pre vysoké frekvencie a výkony, ale aj nároky na objem súčiastky, s čím súvisia aj požiadavky na chladenie, či globálne náklady na výrobu a údržbu systému.

Tabuľka 2.1: Porovnanie fyzikálnych parametrov vybraných polovodičov a GaN pri $T = 300\text{K}$ (Ozpineci et al., b. r.)

Parametre		Si	4H-SiC	GaN
šírka zakázaného pásma	E_g (eV)	1,12	3,26	3,45
dielektrická konštanta	ϵ_r	11,9	10,1	9,0
intenzita el. poľa, pri ktorej dochádza k prerazu	E_{BR} (kV.cm ⁻¹)	300	2400	5000
pohyblivosť elektrónov	μ_e (cm ² .V ⁻¹ .s ⁻¹)	1500	1000	3D 900 2DEG 2000
pohyblivosť dier	μ_p (cm ² .V ⁻¹ .s ⁻¹)	600	115	850
tepelná vodivosť	λ (W.cm ⁻¹ .K ⁻¹)	1,5	4,9	1,3
saturačná rýchlosť	v_{sat} ($\times 10^7$ cm.s ⁻¹)	1,0	2,0	2,2
intrinická koncentrácia	n_i (cm ⁻³)	$1,0 \times 10^{10}$	$5,0 \times 10^{-9}$	$1,9 \times 10^{-10}$

Tranzistory s vysokou pohyblivosťou elektrónov (High Electron Mobility Transistor - HEMT) na báze GaN vykazujú vysoké spínacie frekvencie (desiatky až stovky GHz) a vyšší výkon. Ich využitie má perspektívu najmä v prichádzajúcich nových technológiách sietí (5G) (Polyakov et al., 2015). Princíp činnosti HEMT tranzistora je založený na využití prítomnosti dvojrozmerného elektrónového plynu (2 Dimensional Electron Gas - 2DEG). Tento 2DEG plyn vzniká na rozhraní dvoch rôznych polovodičových materiálov, ktoré majú rozdielnu šírku zakázaného pásma, najčastejšie na rozhraní AlGaN/GaN (Obr. 2.1). V tejto vrstve dochádza k vytvoreniu vysokej koncentrácie elektrónov, ktoré majú vysokú pohyblivosť (Tab. 2.1).

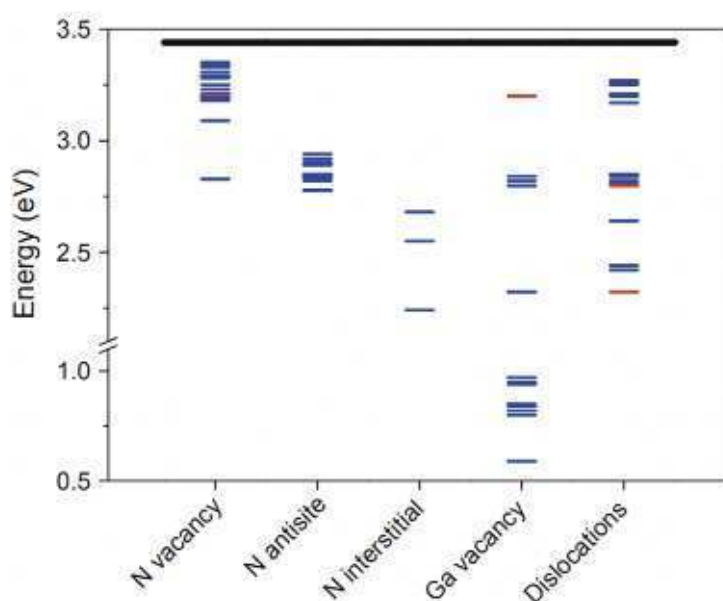


Obr. 2.1: Principiálna schéma AlGaN/GaN HEMT (Gate, Source a Drain elektródy tranzistora) (Samco, b. r.)

Vlastnosti týchto tranzistorov však významne ovplyvňuje prítomnosť elektricky aktívnych porúch. Práve pomocou metódy DLTFs v procese základného výskumu smerujeme k optimalizácii takýchto súčiastok s cieľom podstatne zlepšiť ich účinnosť. Emisno-záchytné procesy prebiehajúce v týchto štruktúrach výrazne podmieňujú výkonnosť HEMT na báze GaN, teda nevyhnutnou súčasťou výskumu je analýza takýchto procesov. Podľa záverov tejto analýzy je možné potom realizovať modifikáciu štruktúry už počas výrobného procesu, čo môže výrazne zlepšiť kvalitu súčiastok.

3 Hlboké energetické hladiny

Poruchy môžu ovplyvniť vlastnosti polovodičových materiálov a štruktúr pozitívne, no aj negatívne. Niektoré vo forme prímiesí vykazujú dobrý vplyv na vlastnosti súčiastok, a teda sú pridávané zámerne pre podporenie fyzikálne užitočnej vlastnosti. Naopak negatívne - nežiadúce poruchy alebo defekty kryštalickej štruktúry (Obr. 3.1) môžu vzniknúť pri nedodržaní požadovaných podmienok pri procese výroby, ktorá predstavuje často extrémne náročné procesy vyžadujúce si daný tlak, teplotu alebo čistotu (Šebok, 2010).



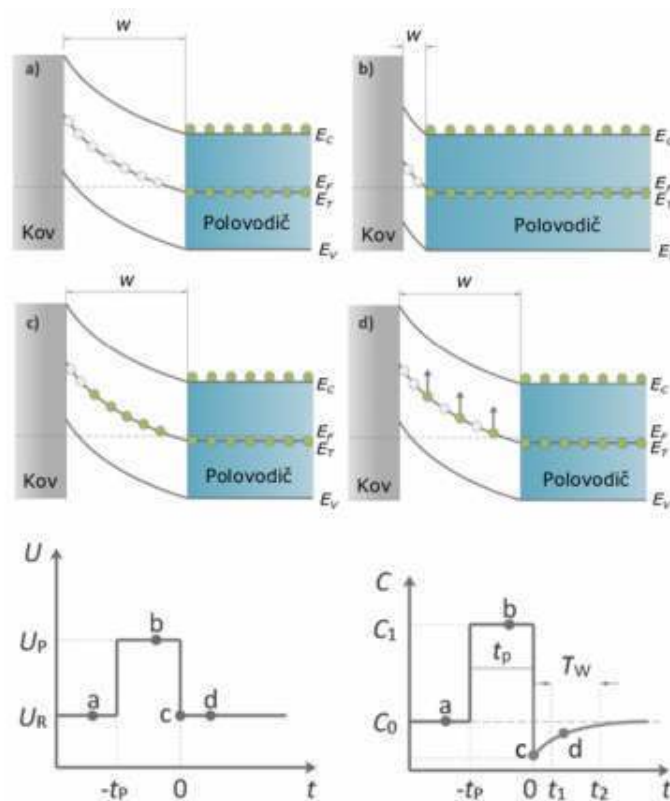
Obr. 3.1: Hlboké energetické hladiny spôsobené prítomnosťou intrinzičných defektov, dislokácií a ich komplexmi v zakázanom pásme energetického pásmového diagramu polovodiča GaN. Nulová energia zodpovedá hornému okraju valenčného pásma, hrubá čierna čiara zodpovedá dolnému okraju vodivostného pásma (Meneghini et al., 2017)

Pojem porucha, je často nahrádzaný pojmi defekt, plytká alebo hlboká energetická hladina, nečistota alebo pasca (Schroder, 2006). Prítomnosť poruchy v polovodiči sa prejaví v pásmovom energetickom diagrame polovodiča vznikom energetickej hladiny E_T v zakázanom pásme polovodiča (Obr. 3.1).

Čo sa týka porúch v GaN, najčastejšie identifikovanou poruchou sú dislokácie. Tie sú radené medzi atómové poruchy kryštalickej mriežky, konkrétne lineárne, alebo tiež nazývané ako čiarové poruchy (Pietriková et al., 2007).

3.1 Elektricky aktívne poruchy

Spektroskopia hlbokých hladín (Deep-Level Transient Spectroscopy - DLTS) (Obr. 3.2) je metóda využívaná na identifikáciu elektricky aktívnych porúch v laboratóriu ÚEF FEI STU v Bratislave. Tejto metóde, resp. jej modifikácii venujeme priestor v samostatnej kapitole, no v tomto bode musíme upriamiť pozornosť na fakt, že touto metódou je možné skúmať len elektricky aktívne poruchy (Deen et al., 2006).



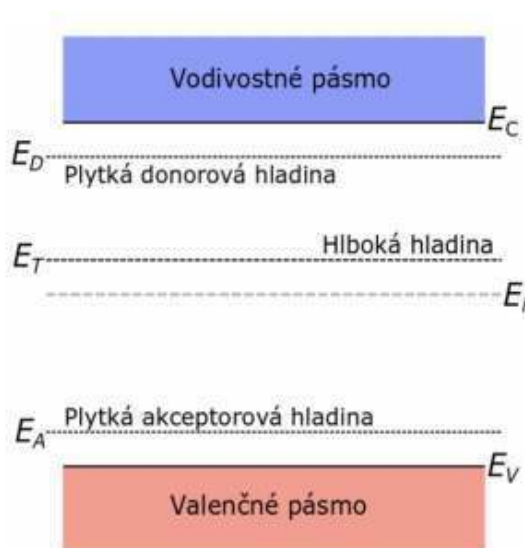
Obr. 3.2: Princíp DLTS (Pohorelec, 2016)

Preto je nevyhnutné definovať a vysvetliť rozdiel pri poruchách, ktoré sú elektricky aktívne, alebo neaktívne. Tie, ktoré sú schopné pracovať s elektrickým nábojom, teda presnejšie povedané môžu náboj zachytiť, či emitovať predstavujú elektricky aktívne poruchy (Deen et al., 2006). Interakcia s elektrickým nábojom je formou reakcie na vonkajšie prostredie a jeho zmeny.

Na druhej strane elektricky neaktívne poruchy neinteragujú a nevykazujú žiadnu odozvu na zmeny vonkajšieho prostredia, a teda ich ani nie je možné zachytiť metódou DLTS.

3.2 Energetické hladiny polovodičov

Dosiaľ sme sa venovali tomu, ako ovplyvňujú poruchy elektrické vlastnosti materiálov a snažili sa globálne popísať problematiku. V tejto časti sa sústreďíme na tzv. energetický pásmový diagram (Obr. 3.3). Ten totiž ilustruje vlastnosti materiálu, pričom je postavený na kvantovej fyzike a pásmovej teórii tuhých látok (Redhammer et al., 2005).



Obr. 3.3: Energetický diagram so znázornenými energetickými hladinami (E_i je intrinzická hladina)

Na pásmovom diagrame (Obr. 3.3) je možné identifikovať jednotlivé pásma. Vodivostné pásmo a jeho dolný okraj s energiou E_C , a tiež valenčné pásmo s energiou horného okraja E_V . Na intervale v rozmedzí hodnôt energie E_V a E_C sa na-

chádza oblasť nazývaná zakázané pásmo E_g , keďže sa tu v prípade ideálneho polovodiča nenachádzajú prípustné energetické hladiny (pre diery a elektróny).

Za predpokladu výskytu porúch, alebo prímiesí môže nastať prípad, kedy v zakázanom pásme vznikne energetická hladina, na ktorej je lokalizovaný elektrón, alebo diera. Donorová hladina E_D vzniká v blízkosti vodivostného pásma a zachytáva elektrón. Na akceptorovej hladine E_A môže byť lokalizovaná diera. Pri platnosti nasledujúcej nerovnosti, nazývame takéto hladiny plytkými (Kósa, 2011):

$$\begin{aligned} E_F - E < 4kT \\ E - E_F < 4kT \end{aligned} \implies |E_F - E| < 4kT \quad (3.1)$$

Kde E_F je Fermiho hladina konkrétneho druhu polovodiča, T je absolútna teplota a Boltzmanova konštanta je označená ako k . V polovodiči typu N je plytká hladina ohraničená zdola hodnotou $(E_F - 4kT)$ a zhora spodným okrajom vodivostného pásma E_C . Pri type P je ohraničenie stanovené hodnotou $(E_F + 4kT)$ zhora a horným okrajom valenčného pásma E_V zdola.

Výskyt poruchy kryštalickej mriežky spôsobuje prítomnosť energetických hladín v zakázanom pásme. Ak nerovnosť 3.1 neplatí, takúto energetickú hladinu označujeme ako hlbokú. Atómy nečistôt alebo prímiesí, všeobecne porúch vytvárajú vyššie popísané energetické hladiny.

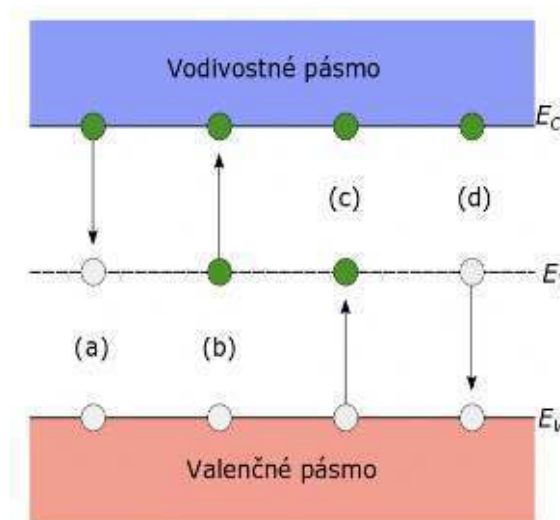
3.3 Procesy v polovodičoch súvisiace s emisiou a zachytením

Energetické hladiny zakázaného pásma, ktorých vznik je podmienený prítomnosťou poruchy, majú priamy vplyv na elektrické vlastnosti polovodiča, keďže sa podieľajú na procesoch zachytenia a emisie nosičov náboja. V súvislosti s týmito procesmi sa odohrávajú ďalšie sprievodné javy ovplyvňujúce charakter porúch, ktoré sa v dôsledku toho tiež označujú pojmami ako rekombinačné alebo zachytné centrá. Podľa čoho je pochopiteľné aj ďalšie pomenovanie - pasce, keďže voľný nosič elektrického náboja môže byť hladinou emitovaný, alebo zachytený.

Interakcia hlbokých hladín zakázaného pásma a voľných nosičov náboja nachádzajúcich sa vo valenčnom a vodivostnom pásme bola popísaná a pomenovaná modelom *SRH* (Shockley-Read-Hall), pričom táto problematika bola ďalej študovaná a rozoberaná v ďalších prácach (Goudon et al., 2007). Procesy je možné

diferencovať podľa ich vlastností a spôsobu interakcie pásem E_C a E_V s voľným nosičom náboja, nasledovným delením prechodov:

- zachytenie voľného elektrónu z vodivostného pásma hlbokou hladinou (Obr. 3.4 a),
- emisia elektrónu smerom z hlbkej hladiny do vodivostného pásma (Obr. 3.4 b),
- zachytenie voľnej diery z valenčného pásma - emisia elektrónu hlbokou hladinou do valenčného pásma (Obr. 3.4 c),
- emisia diery do valenčného pásma - zachytenie elektrónu hlbokou hladinou smerom z valenčného pásma (Obr. 3.4 d).



Obr. 3.4: Proces emisie a zachytenia hlbokou hladinou

Hlboké hladiny môžeme podľa interakcie deliť nasledovne:

- hlboké hladiny donorového - elektrónového typu interagujúce s vodivostným pásmom,
- hlboké hladiny akceptorového - dierového typu s valenčným,

pričom vzdialenosť od pásem E_C a E_V určuje typ pasce. Majoritnou pascou pre polovodiče typu n je pasca elektrónového typu, a teda dierová je minoritnou (Schroder, 2006).

Aproximáciu pre Shottkyho diódu popíšeme nasledujúcimi vzťahmi (Weiss, 1991). Koncentrácia obsadených pascí (obsahujúcich elektróny) počas emisného procesu je definovaná nasledovne:

$$n_{Te}(t) = N_T \exp\left(-\frac{t}{\tau_e}\right) \quad (3.2)$$

Z toho je zrejmá závislosť od exponenciálnej zmeny emisnej časovej konštanty τ_e a koncentrácie pascí N_T , pričom pre emisnú časovú konštantu platí:

$$\tau_e = \left(\sigma_n \nu_{th,n} X_n N_C \exp\left(-\frac{E_C - E_T}{kT}\right) \right)^{-1} \quad (3.3)$$

kde σ_n reprezentuje záchytný prierez, $\nu_{th,n}$ predstavuje veličinu tepelná rýchlosť, X_n symbolizuje faktor entropie a N_C je hustota stavov.

Vyššie spomenutá tepelná rýchlosť $\nu_{th,n}$ je definovaná pomocou hustoty stavov N_C a efektívnej hmotnosti elektrónov m_n^* nasledovne:

$$\nu_{th,n} = \sqrt{\frac{3kT}{m_n^*}} N_C = 2M \left(\frac{2\pi m_n^* kT}{\hbar^2} \right)^{\frac{3}{2}} \quad (3.4)$$

Emisná časová konštantá τ_e je inverznou funkciou ku funkcii reprezentujúcej emisnú rýchlosť elektrónov e_n (Weiss, 1991):

$$e_n = \tau_e^{-1} \quad (3.5)$$

Úpravou vzťahu 3.3 (logaritmovaním a prepísaním) dostávame Arrheniovu závislosť v nasledujúcej podobe:

$$\ln(\tau_e \nu_{th,n} N_C) = \frac{E_C - E_T}{kT} \frac{1}{T} - \ln(X_n \sigma_n) \quad (3.6)$$

Arrheniovu závislosť je možné získať ako výsledok priamej alebo nepriamej evaluácie, čo sú postupy využívané pracoviskom pri realizovaní experimentálnych meraní pomocou metódy DLTS.

3.4 Identifikovanie poruchy

Pomocou metódy DLTS je možné určiť typ hlbokkej hladiny podľa polaroty DLTS signálu, ale aj identifikovať poruchu na základe dôležitých parametrov pasce (Lang, 1974). Medzi základné parametre hlbokkej hladiny patrí aktivačná energia ΔE

($\Delta E_T = E_C - E_T$) a záchytný prierez nosičov náboja $X_n\sigma_n$, ktoré je možné dostať z Arrheniovej závislosti (vzťah 3.6).

Výsledkom lineárnej regresie z grafu $\ln(\tau_e\nu_{th,n}N_C) = f(1/T)$ je lineárna funkcia:

$$y = A_R x + B_R \quad (3.7)$$

kde A_R predstavuje smernicu a B_R symbolizuje priesečník regresnej priamky. Potom vzhľadom na analógiu Arrheniovej závislosti (vzťah 3.6) s rovnicou regresnej priamky (vzťah 3.7) platí:

$$\ln(\tau_e\nu_{th,n}N_C) = \frac{\Delta E}{k} \frac{1}{T} - \ln(X_n\sigma_n) \quad (3.8)$$

$$A_R = \frac{\Delta E}{k}, \quad (3.9)$$

$$x = \frac{1}{T}, \quad (3.10)$$

$$B_R = -\ln(X_n\sigma_n) \quad (3.11)$$

z čoho je možné odvodiť vzťahy pre základné parametre hlbokých hladín:

$$\Delta E = \frac{A_R k}{q}, \quad (3.12)$$

$$X_n\sigma_n = \exp(-B_R) \quad (3.13)$$

K ďalším dôležitým parametrom pri identifikovaní hlbokých hladín patrí koncentrácia porúch, ktorú možno odhadnúť v najjednoduchšom prípade (platí pre jednoduchú hlbokú energetickú hladinu s koncentráciou obsadených hlbokých hladín N_T , ktorá je homogénna) zo vzťahu:

$$\frac{N_T}{N_D} \approx 2 \cdot \frac{\Delta C}{C_0} \quad (3.14)$$

kde N_T je koncentrácia obsadených hlbokých hladín, N_D je koncentrácia donorových prímiesí, ΔC je rozdiel kapacít meraných v rôznych časoch (amplitúda) a C_0 predstavuje celkovú kapacitu štruktúry nameranú pri pripojenom reverznom napätí U_R .

3.5 Teplotná korekcia Arrheniovej závislosti

Vo vzťahu 3.8 nebola uvažovaná závislosť emisno-záchytných procesov od teploty. Keďže nezávislosť od teploty nie je všeobecne platná, je nutné vykonať tzv. teplotnú korekciu T^2 všetkých bodov Arrheniovej závislosti, ktoré boli vyrátané (Sieghard, 1991), pre správne určenie parametrov hlbokaj pasce. Po odstránení tejto závislosti argumentu logaritmu (vzťah 3.8) dosiahneme správne hodnoty predpisu funkcie regresnej priamky pre Arrheniovu závislosť.

Len pre štandardné emisné procesy je možné aplikovať teplotnú korekciu, pričom platí, že hustota stavov $N_{C,V}$ je úmerná s teplotou $T^{-\frac{3}{2}}$ a tepelná rýchlosť $\nu_{th,n}$ je úmerná hodnote teploty $T^{-\frac{1}{2}}$. Za toho predpokladu môžeme pre Arrheniovu závislosť definovať nasledujúci vzťah, tiež všeobecne označovaný názvom teplotná korekcia (Kósa, 2011):

$$\ln(\tau \times C \times (T/300)^2) \quad (3.15)$$

kde C je kapacita:

$$C = \nu_{th,n}(300K) \times N_{c,v}(300K) \quad (3.16)$$

Implementácia teplotnej korekcie je veľmi dôležitá, keďže správne určenie parametrov pasce je možné len pri tomto štandardnom modeli. Dosiahneme tým teda spresnenie výpočtu a porovnateľné výsledky výpočtov so systémom DLTS, čo uľahčí proces testovania funkcionality softvéru.

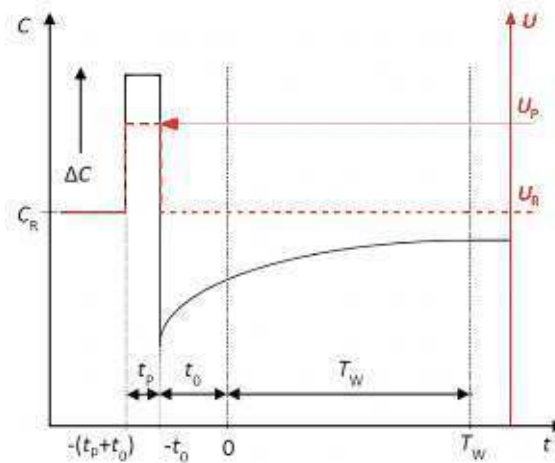
4 Fourierova spektroskopia hlbokých hladín

Predmetom výskumu realizovaného na pracovisku FEI STU v Bratislave sú aj DLTS kapacitné merania transient (kapacitných odoziev v danom čase) vysoko-výkonových a vysokofrekvenčných tranzistorov zhotovených na báze GaN.

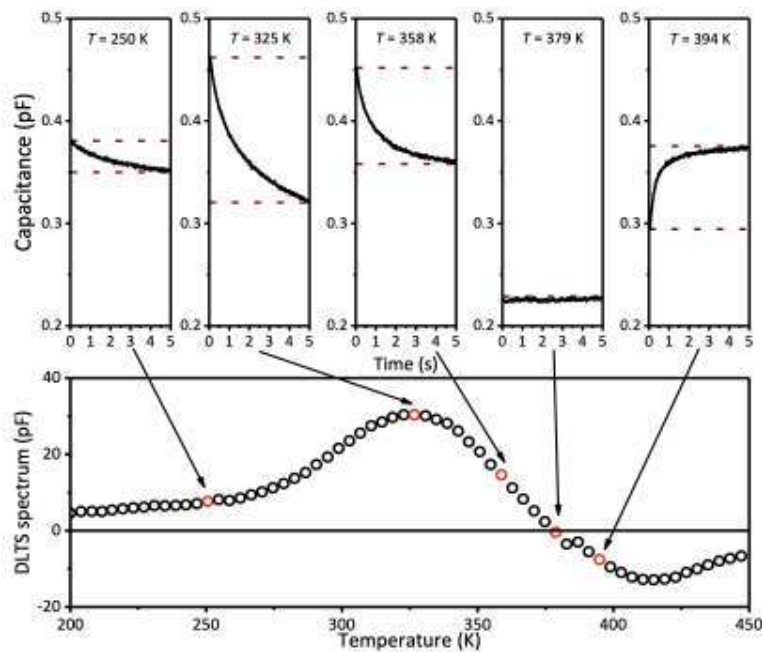
D. V. Lang v sedemdesiatych rokoch minulého storočia postavil základy klasickej metódy spektroskopie hlbokých hladín (Deep-Level Transient Spectroscopy - DLTS) slúžiacej na štúdium emisných a záchytných procesov v polovodičových materiáloch a štruktúrach (Lang, 1974). Avšak táto štandardná metóda je časovo mimoriadne náročná. Jej modifikácia Fourierova spektroskopia hlbokých hladín (Deep-Level Transient Fourier Spectroscopy - DLTFs) (Weiss; Kassing, 1988) nielen významne skracuje čas merania, ale okrem iného poskytuje niekoľkonásobne viac nameraných dát v jednom teplotnom cykle a možnosť ich presnejšej matematickej evaluácie.

4.1 Porovnanie DLTS a DLTFs

V analógovej DLTS je kapacita meraná len v časoch t_1 a t_2 , zatiaľ čo v DLTFs je nameraná celá kapacitná transienta v N bodoch (Obr. 4.1). Ak takto namerané transiency pri rôznych teplotách vynásobíme korelačnými funkciami, získame DLTFs signál (Obr. 4.2). Z nameranej transiency sú aplikáciou numerickej Fourierovej transformácie odvodené diskkrétne Fourierove koeficienty (Weiss; Kassing, 1988). Pomocou nich je možné vyrátať časovú konštantu τ_e a dôležité parametre (Weiss, 1991) pre identifikáciu pasce (aktivačná energia, efektívny záchytný prierez a koncentrácia porúch).



Obr. 4.1: Vstupné meracie parametre metódy DLTS: záverné napätie U_R , plniace napätie U_P , doba plniaceho impulzu t_P , šírka periódy snímanej kapacitnej odozvy T_W a kapacitná transients (Drobný, 2018)



Obr. 4.2: Namerané spektrum DLTS s kapacitnými transientami vo vybraných bodoch (Pohorelec, 2016)

Medzi hlavné nevýhody (Schroder, 2006) kapacitnej DLTS patrí to, že neme-
ria celú kapacitnú transientsu, čo pri komplexných emisných procesoch neposta-
čuje pre dosiahnutie významných výsledkov. Pre zostavenie Arrheniovej závis-
losti je preto potrebné veľké množstvo meraní. Ďalším negatívom (Šarocký, 2018)

je mylný predpoklad týkajúci sa teplotnej závislosti časovej konštanty, no nie amplitúdy (dôvodom nesplniteľnosti je teplotná závislosť Fermiho hladiny).

4.2 Fourierova transformácia

Fourierova transformácia je využívaná pri metóde DLTFs pre určitú "konverziu", či ináč povedané mapovanie časovo závislého signálu na signál závislý od frekvencie. Všeobecná definícia tvaru Fourierovej transformácie $F(j\omega)$ (4.1) a aj inverznej Fourierovej transformácie (4.2) je nasledovná:

$$F(j\omega) = \int_{-\infty}^{\infty} f(t) \exp(-j\omega t) dt \quad (4.1)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) \exp(-j\omega t) dt \quad (4.2)$$

kde

$$\omega = 2\pi f$$

Pri využití vyjadrenia pomocou harmonických signálov (goniometrických funkcií sínus a kosínus pre popis periodických javov) a za predpokladu, že má transients ($f(t)$ časový signál, ktorý bol nameraný) exponenciálny charakter môžeme signál vyjadriť nasledovným spôsobom:

$$f(t) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 t + b_n \sin n\omega_0 t) \quad (4.3)$$

pričom

$$\omega_0 = \frac{2\pi}{T_W} = 2\pi f_0$$

kde Fourierove koeficienty predstavujú členy a_0, a_n, b_n a počet členov Fourierovho radu vyjadruje hodnota n . Ďalej pri exponenciálnom Fourierovom rade pre analytické (spojité) Fourierove koeficienty (Sieghard, 1991, *PhysTech, Theory Manual*, b. r.) platí:

$$c_n = \frac{1}{T_W} \int_0^{T_W} f(t) \exp(-in\omega_0 t) dt \quad (4.4)$$

Ak platí, že koeficienty a_n , b_n a $f(t)$ sú reálne, potom môžeme napísať nasledovné, z čoho vyplýva, že koeficienty tvoria reálnu a imaginárnu zložku koeficientu c_n :

$$c_n = \frac{1}{2}(a_n - ib_n) \quad (4.5)$$

Kapacitnú transientu, ktorá má exponenciálny charakter môžeme vyjadriť pomocou reálnej funkcie - platného exponenciálneho časového zákona (Weiss, 1991):

$$f(t) = A \exp\left(-\frac{t + t_0}{\tau}\right) + B \quad (4.6)$$

pričom A predstavuje amplitúdu, B je posun, časové oneskorenie je vyjadrené pomocou t_0 a τ predstavuje časovú konštantu. Pre $f(t)$ sú spojité Fourierove koeficienty vyjadrené vzťahmi:

$$a_0 = \frac{2A}{T_W} \exp(-t_0/\tau)(1 - \exp(-T_W/\tau))\tau + 2b \quad (4.7)$$

$$a_n = \frac{2A}{T_W} \exp(-t_0/\tau)(1 - \exp(-T_W/\tau)) \frac{\frac{1}{\tau}}{\frac{1}{\tau^2} + n^2\omega_0^2} \quad (4.8)$$

$$b_n = \frac{2A}{T_W} \exp(-t_0/\tau)(1 - \exp(-T_W/\tau)) \frac{n\omega_0}{\frac{1}{\tau^2} + n^2\omega_0^2} \quad (4.9)$$

Exponencialita transienty je daná charakterom koeficientov, je možné ju overiť platnosťou nasledujúcich podmienok:

$$a_k < a_n < \frac{k^2}{n^2} a_k \quad , \text{ pre } n < k \quad (4.10)$$

$$\frac{n}{k} b_k < b_n < \frac{k}{n} b_k \quad , \text{ pre } n < k \quad (4.11)$$

$$\frac{b_n}{a_n} \frac{a_k}{b_k} = \frac{n}{k} \quad (4.12)$$

Mieru exponenciality nameranej kapacitnej transienty je možné vyjadriť využitím koeficientov prvého a druhého rádu:

$$\frac{b_2}{2b_1} \frac{a_1}{a_2} = 1 \quad (4.13)$$

Hodnotu amplitúdy A je možné definovať pomocou ktoréhokoľvek koeficientu a_n úpravou rovnice 4.8:

$$A = a_n \frac{T_W}{2} \frac{\exp(t_0/\tau)}{(1 - \exp(-T_W/\tau))} \frac{\frac{1}{\tau^2} + n^2 \omega_0^2}{\frac{1}{\tau}} \quad (4.14)$$

Časovú konštantu τ získame ako hodnotu pomeru dvoch koeficientov, a práve pri priamej evaluácii je využívaný tento postup určenia časovej konštanty:

$$\tau(a_n, a_k) = \frac{1}{\omega_0} \sqrt{\frac{a_n - a_k}{k^2 a_k - n^2 a_n}} \quad (4.15)$$

$$\tau(b_n, b_k) = \frac{1}{\omega_0} \sqrt{\frac{k b_n - n b_k}{k^2 n b_k - n^2 k b_n}} \quad (4.16)$$

$$\tau(a_n, b_n) = \frac{1}{n \omega_0} \frac{b_n}{a_n} \quad (4.17)$$

4.2.1 Definícia diskkrétnej Fourierovej transformácie

Diskrétnu Fourierovu transformáciu $F(N)$ (Sieghard, 1991, *PhysTech, Theory Manual*, b. r.) je možné definovať nasledovne:

$$F_N = \sum_{k=0}^{N-1} f_k \exp(-i2\pi nk/N) \quad (4.18)$$

$$F_N = NC_n^D \quad (4.19)$$

$$F_n = \frac{f_0}{2} + \sum_{k=1}^{N-1} f_k \exp(-i2\pi nk/N) + \frac{f_N}{2}, \text{ pre } n = 0, 1, \dots, N-1 \quad (4.20)$$

kde

$$\exp(-i2\pi nk/N) = \cos\left(\frac{2\pi kn}{N}\right) - i \sin\left(\frac{2\pi kn}{N}\right) \quad (4.21)$$

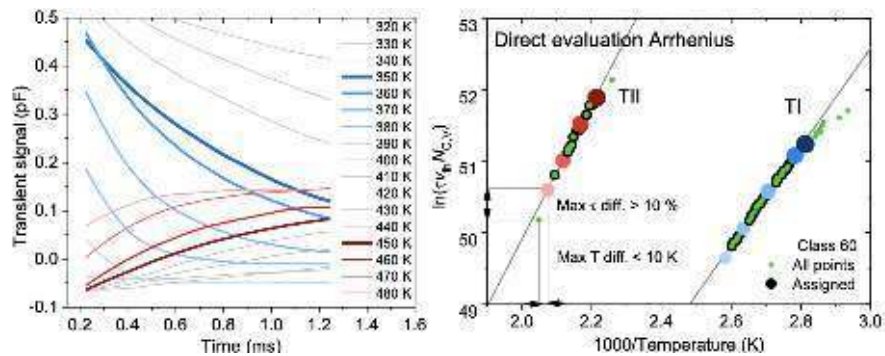
4.2.2 Predpis lichobežníkového pravidla

Lichobežníkové pravidla pre N rovnakých intervalov má nasledujúcu podobu:

$$\int_a^b f(x) dx \approx \sum_{i=1}^n \frac{f(x_{i-1}) + f(x_i)}{2} (x_i - x_{i-1}) = \frac{b-a}{n} \left(\frac{f(x_0)}{2} + \sum_{i=1}^{n-1} f(x_i) + \frac{f(x_n)}{2} \right) \quad (4.22)$$

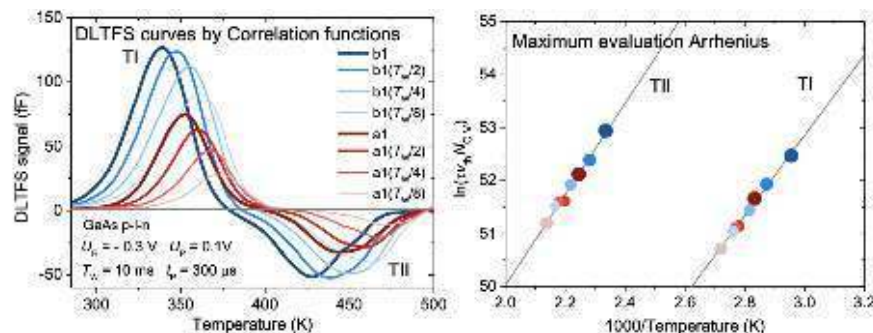
4.3 Priama a nepriama evaluácia

V rámci evaluácie metódou DLTFs je pri splnení podmienok exponenciality možné využiť priamu metódu (Obr. 4.3) evaluácie na každú nameranú kapacitnú transientu. Priama evaluácia je realizovaná tak, že je zostavená Arrheniova závislosť a vypočítané jednotlivé hodnoty ako aj časová konštanta priamo využitím napr. signálov $a_n(T)$, či $b_n(T)$ aplikovaním uvedeného matematického aparátu.



Obr. 4.3: Príklad využitia priamej evaluácie DLTFs. Body Arrheniovej závislosti (vpravo) sú vypočítané podľa nameraných kapacitných transient pri rôznych teplotách (vľavo) a vopred definovaných evaluačných parametroch: hodnotiaca trieda 60, rozptyl časovej konštanty 10% a maximálny teplotný rozdiel 10K (Kósa, 2016)

Pri nepriamej evaluácii (Obr. 4.4) je Arrheniova závislosť na rozdiel od priamej tvorená z maxim funkcie DLTS signálu v závislosti od teploty.



Obr. 4.4: Príklad DLTFs nepriamej evaluácie. Každý bod Arrheniovej krivky (vpravo) je vypočítaný pomocou emisnej časovej konštanty τ zodpovedajúcej maximu alebo minimu korelačnej krivky DLTFs (vľavo) (Kósa, 2016)

5 Systém pracoviška DL8000

Už od počiatku osemdesiatych rokov sa metóda DLTS stala veľmi významnou pri skúmaní vlastností polovodičov, a to práve pre vysokú citlivosť v procese určenia parametrov pascí a ich koncentrácie v materiáli (Lang, 1974).

Digitálny DLTS systém, ktorý ďalej popíšeme, je flexibilný vzhľadom na problémy súvisiace s analógovými systémami DLTS (*PhysTech*, b. r.). Bol vytvorený v rámci kooperácie Univerzity v Kassel so spoločnosťou PhysTech (Weiss, 1991), pričom pri systémoch BIO-RAD pracovišok DL8000 sa dá považovať za bežnú súčasť (*PhysTech*, b. r.).

5.1 Funkcie systému

Systém DL8000 operuje s využitím rozličných módov a voliteľnými možnosťami nastavenia. Medzi funkcie a dôležité charakteristiky systému (*PhysTech, Software Manual*, b. r.) patrí:

- modulárny a flexibilný hardvér,
- podporovanie rozličných regulátorov teploty a kryostatov,
- Fourierova transformácia - frekvenčné okná, a tiež definovanie korelačných funkcií užívateľom,
- DLTFs evaluácia - vyhodnotenie,
- automatická kontrola kontaktu a kompenzácia kapacity,
- rutinné merania a vylepšený softvér,
- tri FET (Field Effect Transistor) prúdové prechodové merania,
- veľký rozsah merania kapacity a koncentrácie.

5.2 Meracie pracovisko

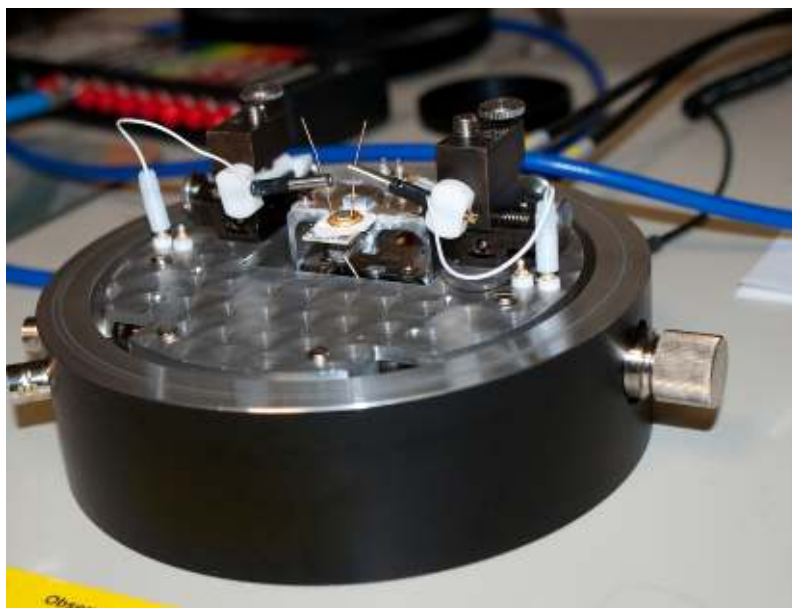
Na Ústave elektroniky a fotoniky, FEI STU v Bratislave je situované meracie pracovisko, ktoré je zobrazené na Obr. 5.1, pričom práve s týmto laboratóriom je realizovaná spolupráca. Obr. 5.2 zobrazuje autentický záber na kryostat s nakontakovanou vzorkou.



Obr. 5.1: Výskumné a meracie pracovisko

Diagnostika polovodičových materiálov je vykonávaná za pomoci pracoviska DL8000 (Digital Deep Level Transient Spectrometer) systému BIO-RAD. Vďaka tomuto systému, ktorý využíva dostatočne citlivé metódy sú uskutočňované a vyhodnocované experimentálne merania, a teda ich výsledky prispievajú zo širšieho uhla pohľadu ku komplexnej diagnostike vlastností testovaných materiálov a štruktúr.

Systém pracoviska využíva metódu DLTFIS (modifikáciu klasickej DLTS), teda v procese spracovania dát, ktoré boli namerané počas experimentu, je aplikovaná Fourierova transformácia (popísaná v časti 4.2). Namerané dáta predstavujú tzv. transienty, ktoré sú kapacitnou odozvou v danom čase.



Obr. 5.2: Pohľad na kryostat s nakontaktovanou vzorkou

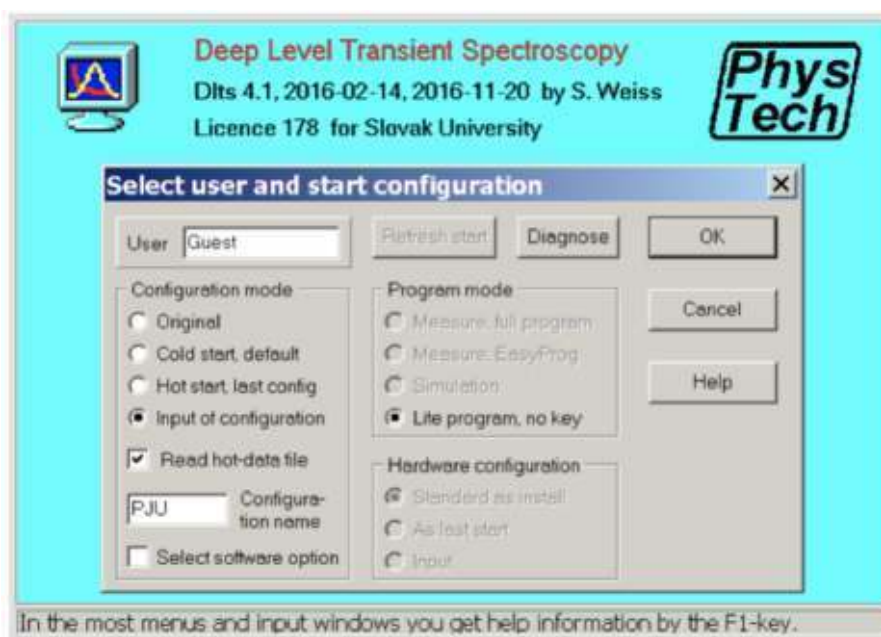
5.3 Softvérové vybavenie

DLTS 4.1 (2016-02-14, 2016-11-20 by S. Weiss) je poslednou verziou licencovaného softvéru spoločnosti PhysTech určeného pre operačné systémy rodiny Windows pre DL8000, ktorého úvodnú obrazovku demonštruje Obr. 5.3. To znamená, že nebude ďalej upravovaný a distribuovaný medzi pracoviská, ktorých je štandardnou súčasťou.

Tento fakt predstavuje prvú z negatívnych vlastností, keďže diagnostika nových štruktúr na báze progresívnych materiálov si vyžaduje pravidelnú modifikáciu existujúceho softvéru, úpravu pre novšie platformy, prípadne korekciu nájdených nedostatkov, či vylepšovanie funkcionality. To je extrémne dôležité najmä pri takto rozsiahlych a komplexných systémoch. Systém však poskytuje riadenie procesu merania, teda samotný experiment a tiež uchovanie nameraných dát.

Aj samotný charakter experimentov si môže časom vyžadovať špecifické úpravy v softvéri, ktoré by odzrkadľovali potenciálne ešte nedefinované požiadavky. Ako už bolo spomenuté vyššie, softvérové riešenie je komplexné, teda v určitom ohľade natoľko zložité, že predstavuje značné obmedzenia pre pracovisko.

Potrebné úpravy sa nemusia týkať len funkcionality, či úprav spätých s platformou, ohľad treba brať aj na efektivitu užívateľského rozhrania. Pri komplexných programoch je veľmi dôležité prehľadné a intuitívne grafické rozhranie, ktoré do-



Obr. 5.3: Softvér DLts 4.1, 2016-02-14, 2016-11-20 by S. Weiss (*PhysTech*, b. r.)

káže prispieť k efektívnosti pri používaní. V tomto prípade neposkytuje v súčasnej dobe očakávaný užívateľský zážitok (*UX*) a potrebnú úroveň použiteľnosti (*User Experience Basics*, b. r.).

Závažné problémy vznikajú pri skúmaní zložitých štruktúr s potenciálne veľkým množstvom porúch. Príčinou je nutnosť realizácie veľkého množstva meraní, a následného spracovania, resp. vizualizovania. Spracovanie a vyhodnocovanie nameraných dát je časovo náročné a pracné, keďže v softvéri nie je implementovaná možnosť filtrácie dát podľa parametrov, resp. podmienok merania.

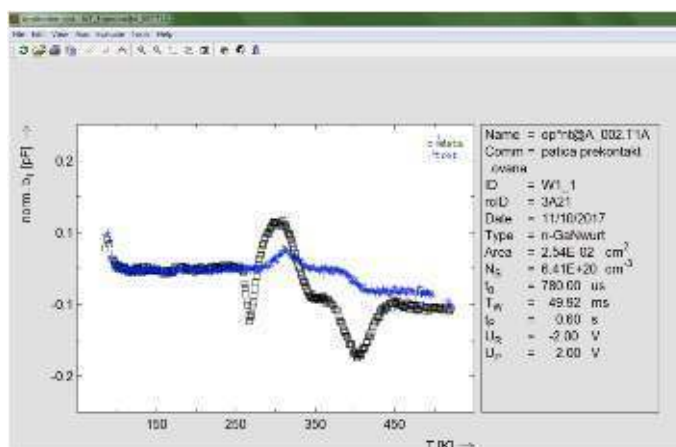
Dosiaľ boli používané dodatočné programy na filtráciu (*MS Excel*) a vizualizáciu resp. analýzu (*MathLab*¹, *Origin*² alebo *OriginPro*³). Softvér je obmedzený na vizualizáciu maximálne dvoch meraní (s potrebným rozlíšením), avšak je potrebné často porovnať viaceré merania zhodujúce sa v určitých parametroch (Obr. 5.4 vs. 5.5). Bez filtrácie to predstavuje obmedzenie aj pri výbere meraní, ktoré majú byť vizualizované.

Po vykonaní experimentu riadeného systémom je na výstupe sada nameraných dát, ktorá obsahuje viacero setov. Pričom zväčša každý set pozostáva z troch

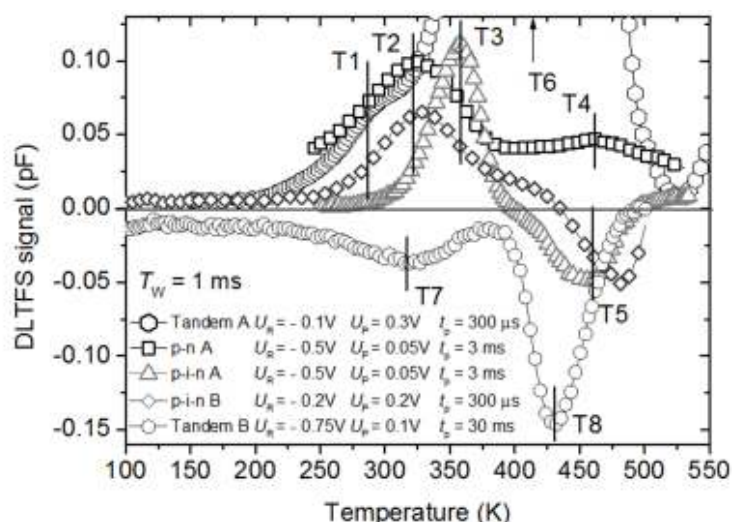
¹http://www.mathlab.mtu.edu/mediawiki/index.php/Main_Page

²<https://www.originlab.com/Origin>

³<https://www.originlab.com/index.aspx?go=PRODUCTS&PID=1834>



Obr. 5.4: Porovnanie dvoch meraní softvérom a zobrazenie príslušných parametrov merania (*PhysTech*, b. r.)

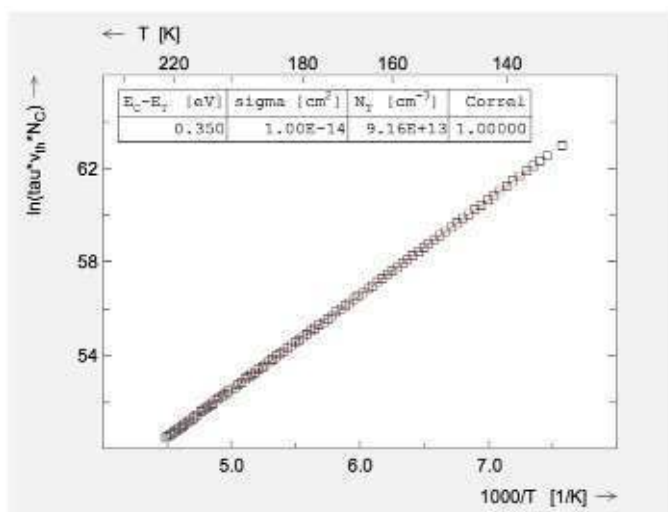


Obr. 5.5: Dôkladnejšia vizualizácia meraní iným softvérom (Kósa et al., 2016)

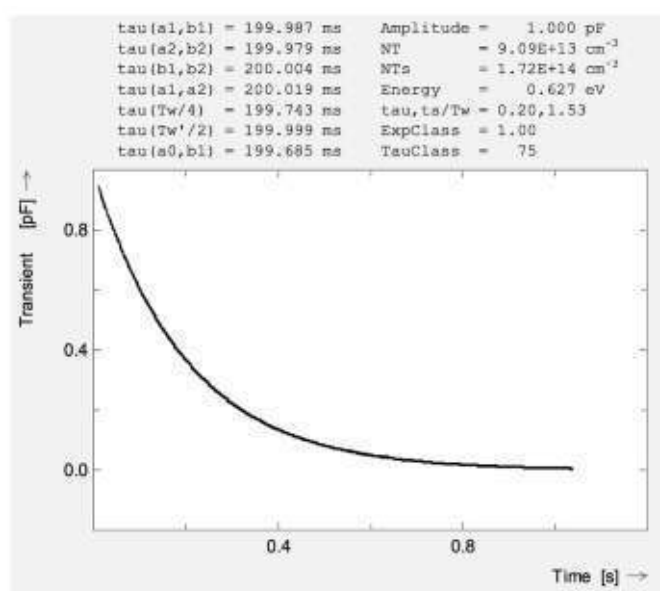
meraní pri rôznych podmienkach, teda disponujú medzi sebou rôznymi hodnotami parametrov.

Okrem riadenia merania a vizualizácie softvér poskytuje tiež vyhodnotenie - evaluáciu nameraných dát metódami (Weiss, 1991), ktoré sme podrobnejšie popísali v časti 4.3. Pri priamej evaluácii je Arrheniova závislosť (Obr. 5.6) vytváraná z nameraných transient (Obr. 5.7) využitím Fourierových koeficientov, za predpokladu, že sú splnené podmienky exponenciality.

Optimalizácia spôsobu výberu vstupných transient a kontroly podmienok exponenciality by podporila efektivitu procesu. Implementáciou separácie exponen-



Obr. 5.6: Arrheniova závislosť (PhysTech, b. r.)



Obr. 5.7: Nameraná transients (PhysTech, b. r.)

ciálnych častí transient, resp. užívateľom voleného faktoru citlivosti by sa dali zjednodušiť určité časovo náročnejšie kroky.

Pri ďalšej metóde nazývanej nepriama evaluácia je Arrheniova závislosť tvorená z maxím DLTS spektier, pričom môžu byť vybrané manuálne alebo automaticky. Metóda je použiteľná, aj keď transients vykazujú rozličnú mieru exponenciality. Korelačné funkcie sú pri tejto metóde definované softvérom, nie hardvérom, teda je na výber škála dvadsiatich ôsmich korelačných funkcií využívaných

pri transformácii nameraných kapacitných transient na DLTS signál, ktorý je závislý od teploty (Šarocký, 2018).

5.4 Vyhodnotenie analýzy systému

Aj napriek rôznym funkciám a širokej škále nastavení, systém nepokrýva špecifické potreby pracoviska. Integrácia viacerých rozšírení, ale aj zjednodušenie rozhrania a implementovanie najdôležitejších funkcií do jedného softvérového riešenia by sa mohli pozitívne podpísať na efektívite výskumného procesu. Na tomto mieste by sme chceli v prehľadnej forme zhrnúť hlavné nedostatky, ktoré sme identifikovali pri analýze používaného softvérového riešenia:

- užívateľské a systémové obmedzenia,
- spôsob výpisu a porovnanie parametrov merania - parciálne implementované pri bakalárskej práci (Patlevičová, 2018),
- porovnanie najviac dvoch priebehov merania - parciálne implementované pri bakalárskej práci (Patlevičová, 2018),
- vyhodnotenie v setoch - nutnosť zhody v parametroch merania - parciálne implementované pri bakalárskej práci (Patlevičová, 2018),
- časovo náročná vizualizácia,
- zefektívnenie aplikácie podmienok exponenciality transient - definovanie faktoru citlivosti,
- optimalizácia procesu analýzy a vyhodnotenia pri väčšom množstve dát.

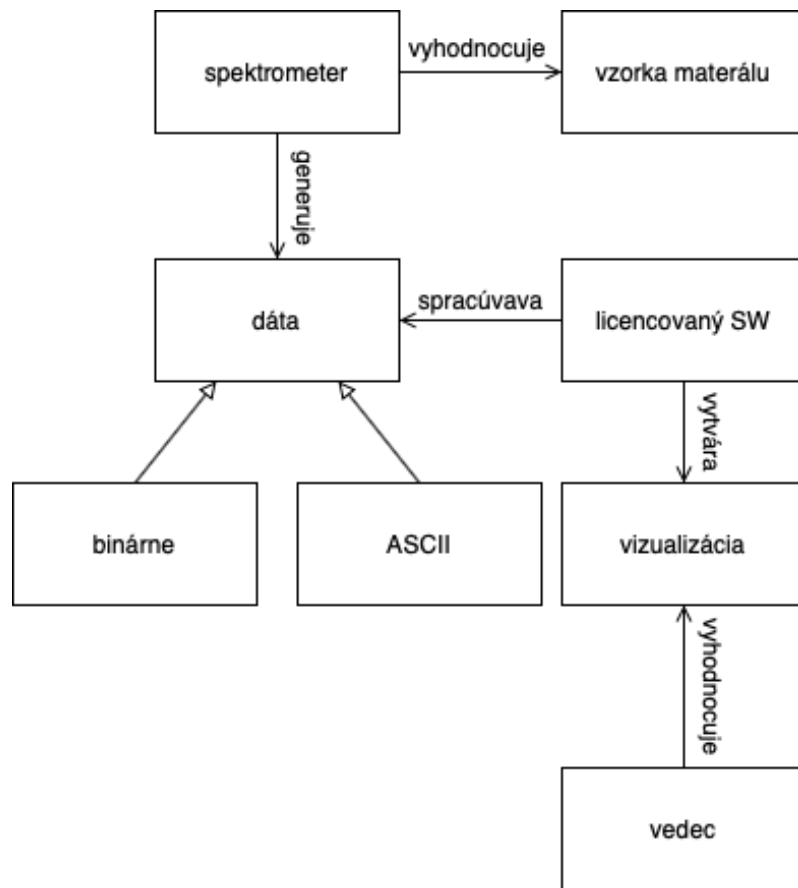
6 Metodológia a návrh

Po oboznámení sa s teoretickými základmi problematiky a po analýze pracoviska využívajúceho licencovaný softvér sa pokúsime v tejto časti namodelovať a navrhnuť program pokrývajúci nájdené nedostatky. Pri dizajnovaní a návrhu použijeme štandardizované metódy a postupy, ktoré nám následne uľahčia a urýchlia proces implementácie programu.

6.1 Opis domény a konceptuálneho modelu

Pre dôkladný návrh softvéru je podstatné poznať pozadie procesu a nevnímať implementovaný softvér len ako časť procesu vytrhnutú z kontextu. Pre lepšie pochopenie fungovania pracoviska na FEI STU sme ho osobne niekoľkokrát navštívili, čo nám poskytlo lepšiu predstavu a hlbšie poznanie danej domény. Jej fungovanie sme sa pokúsili znázorniť pomocou diagramu zahŕňajúceho podstatné prvky a akcie medzi nimi.

Na Obr. 6.1 je jednoduché identifikovať podstatné časti domény, pričom ich interakcie znázorňujú šípky resp. stručný popis v ich blízkosti. Modelovaný proces teda môžeme popísať nasledovne. Skúmaná polovodičová štruktúra je vyhodnocovaná spektrometrom, pričom namerané hodnoty, ako aj podmienky merania a ďalšie informácie o ňom sú zaznamenané v podobe dát v binárnej forme. Tie je možné exportovať do textovej (ASCII) podoby, keďže sú kvôli licencií zabezpečené šifrovaním. Namerané dáta vo forme súborov predstavujú vstup licencovaného softvéru, ktorý ich analyzuje a vykonáva nad nimi operácie, vďaka ktorým generuje grafické znázornenie samotných dát, alebo ich spracovanej podoby. Tieto vizualizácie ďalej pracovníci laboratória skúmajú a identifikujú z nich základné parametre elektricky aktívnych porúch a kvalitatívne vyhodnotenie vplyvu vybraných parametrov na distribúciu týchto porúch.



Obr. 6.1: Diagram reprezentujúci koncept domény

6.2 Prípady použitia a proces identifikácie objektov a akcií

Po oboznámení sa s doménou je počas návrhu možné prirodzene rozpoznať prípady použitia softvéru. Tie reprezentujú scenáre (postupnosť krokov), ktoré je možné za pomoci programu realizovať a dosiahnuť stav, do ktorého sa používateľ chce dostať.

Sú konceptom, ktorému rozumie dobre nielen používateľ (keďže chápe opis použitia programu za danej situácie v prirodzenej reči a jednotlivé kroky), ale aj programátor, ktorý na základe týchto informácií pochopí lepšie požiadavky na program a následne dokáže vlastnosti softvéru zovšeobecniť a implementovať. Počas modelovania prípadov použitia (*Use Cases*, b. r.) dochádza k ozrejmieniu použitia a fungovania systému, a to vysvetlením jednotlivých čiastkových

cieľov s jednotlivými krokmi. Tento proces môže tiež poukázať na nezrovnalosti, alebo odhaliť pri diskusii s užívateľom jednoduchšie riešenia. V našom prípade sú hlavné elementy určené nasledovne:

- aktér - používateľ systému, čiže zamestnanec laboratória,
- cieľ - analýza nameraných dát (vizualizácia Arrheniovej závislosti a výpočet parametrov poruchy),
- systém - postupnosť krokov pre dosiahnutie cieľa, zahŕňajúca podmienky a funkčné požiadavky:
 - vedec daného laboratória spustí program,
 - zvolí vstupné transieny (alebo tempscan dáta pre vizualizáciu) zo súborového systému,
 - opakovane môže pridať viacero vstupných súborov, ktoré sa zobrazia v zozname vstupných súborov, dáta s nevyhovujúcou štruktúrou budú odfiltrované,
 - vstupné dáta môže vizualizovať,
 - definuje podmienky evaluácie,
 - systém ho upozorní pri výskyte dát nespĺňajúcich podmienky, tie budú musieť byť z procesu evaluácie vylúčené,
 - po zvolení začatia evaluácie (za predpokladu splnenia podmienok a dostatočného počtu vstupných súborov) je zobrazená vizualizácia Arrheniovej závislosti spolu s vypočítanými parametrami,
 - vizualizáciu a evaluáciu môže exportovať do viacerých grafických, alebo aj textových formátov (*png, jpg, tiff, svg* a *csv*) a ukončiť program.

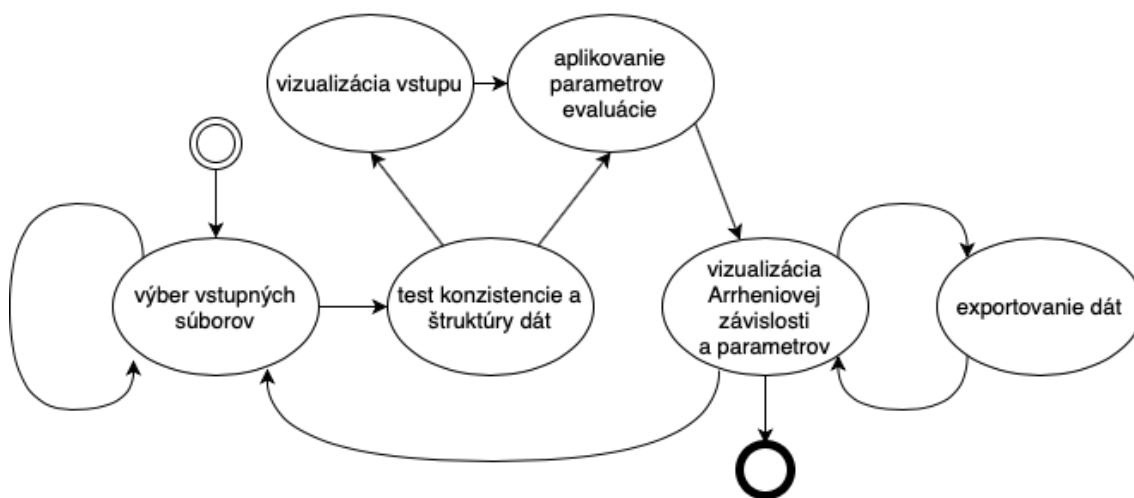
6.2.1 Povolená postupnosť stavov programu

Pomocou stavového diagramu (*State Machine Diagrams*, b. r., Fowler, 2003), podľa štandardu UML¹ sa snažíme poukázať na sled krokov v závislosti na logike programu. Postupnosť stavov je podmienená vlastnosťami, ktorými má navrhovaný

¹<https://www.uml.org/what-is-uml.htm>

system disponovať. Stavový diagram na Obr. 6.2 demonštruje povolenú postupnosť, a tiež vymedzuje prípady, kedy je prechod medzi jednotlivými stavmi v určitom smere zakázaný.

Z východzieho stavu je povolený prechod len do stavu, kedy užívateľ môže opakovane voliť vstupné súbory (transienty resp. tempscan súbory). Pred vizualizáciou a evaluáciou (vizualizácia Arrheniovej závislosti a parametrov pasce) je potrebné vykonať test konzistencie a štruktúry dát, nevyhovujúce nie sú pridané do zoznamu reprezentujúci vstup. Pre evaluáciu je potrebné zadať a aplikovať príslušné parametre, dáta nevyhovujúce zadaným parametrom evaluácie sú zo zoznamu zvolených súborov vylúčené. Preto je povolený prechod zase späť do stavu výberu vstupu. Po splnení podmienok (parametre a štruktúra) je možné prejsť do stavu s vizualizáciou Arrheniovej závislosti. Je možné zas zmeniť vstupné dáta a proces opakovať podľa povolených prechodov až to tohoto stavu, z ktorého je možné prejsť do finálneho stavu, teda program ukončiť. Ako rozšírenie doplníme ešte stav, v ktorom je možné dáta exportovať do ponúkaných grafických, či textových formátov a zvoliť cestu, kam bude exportovaný súbor uložený.



Obr. 6.2: Stavový diagram programu

6.2.2 Konceptuálny model programu

Po vytvorení stavového diagramu, čomu predchádzalo transformovanie požiadaviek do podoby prípadov použitia, sme ďalej pokračovali v návrhu štruktúry softvéru. Fungovanie domény je možné modelovať pomocou konceptuálneho modelu, v tomto prípade chceme navrhnuť už koncept softvérového riešenia. Preto sme zvolili formu diagramu tried (Obr. 6.3) pre naplánovanie štruktúry nášho softvérového riešenia.

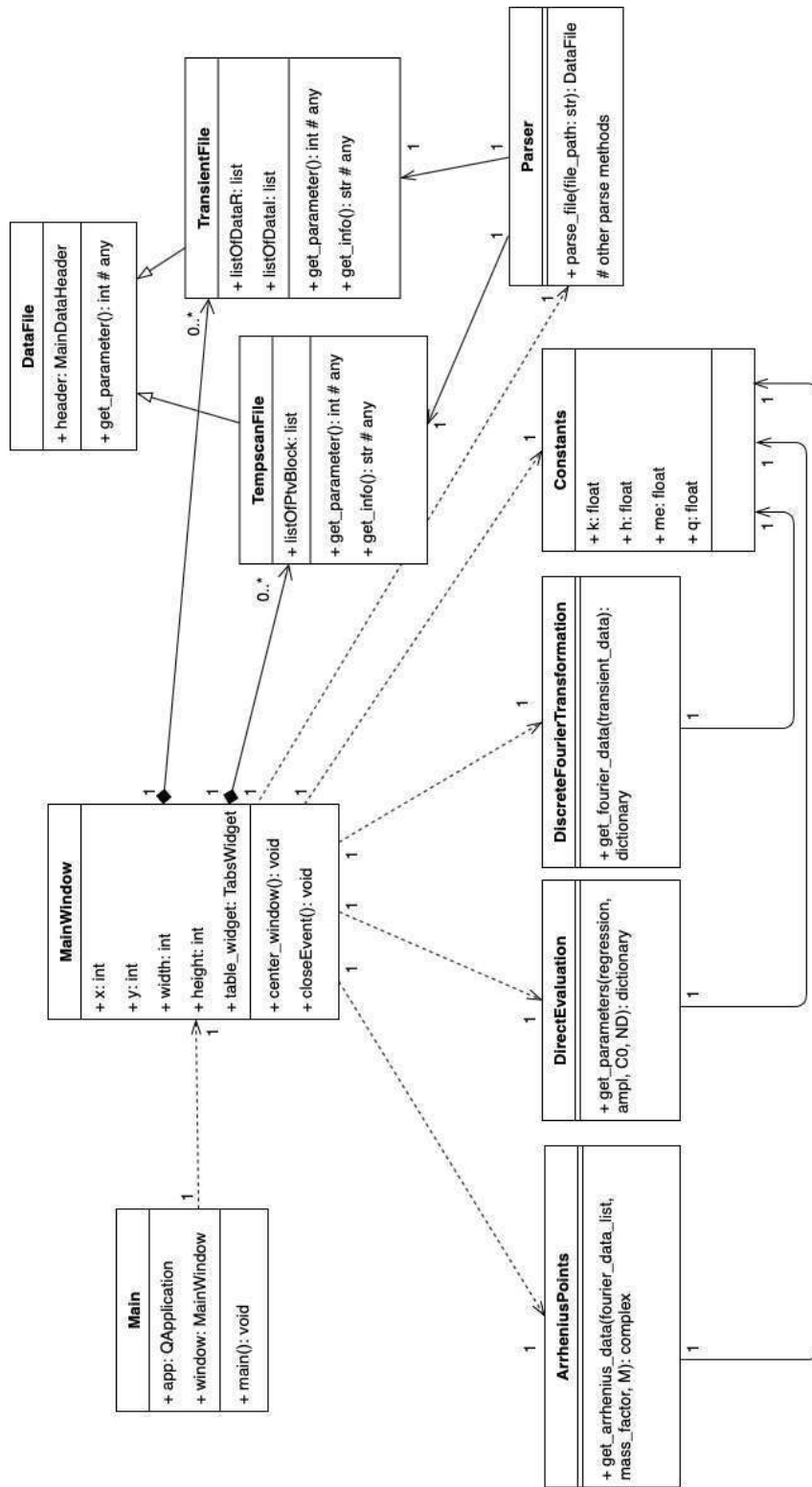
Diagram tried (*UML Class and Object Diagrams Overview*, b. r., Fowler, 2003) zo štandardu UML, je používaný pri náčrte fungovania softvéru. Znázorňuje jednotlivé triedy, čiže entity, alebo objekty danej domény. Poukazuje na ich štruktúru a vlastnosti, tiež zobrazuje vzťahy medzi nimi. Podľa tohoto návrhu je pri vývoji jednoduchšie implementovať program, keďže z diagramu tried je jasná hierarchia objektov, interakcie a volané metódy, ich atribúty a iné podrobnosti. Predstavuje istú formálnu abstrakciu softvéru, z ktorej zreteľne vyplýva, čo má daná trieda obsahovať, a čo má poskytovať, teda ako komunikuje s inými časťami softvéru.

V procese návrhu sme ďalej postupovali náčrtom konceptu programu, a to vo forme diagramu tried, ktorý je znázornený na Obr. 6.3. Podľa tejto abstrakcie fungovania programu v grafickej podobe budeme postupovať pri implementácii.

Považujeme to za veľmi dobrý spôsob naplánovania, ktorý nám jednoznačne uľahčí samotné programovanie a zamedzí vzniku nezrovnalostí, duplicity kódu, alebo zlému rozloženiu funkcionality v rámci tried. Takýto postup by mal podporiť rýchlosť a prehľadnosť implementácie, pričom tiež by mal obmedziť improvizáciu (často aj chaotickú) a nedodržanie mnohých ďalších konvencií pri programovaní.

6.3 Návrh používateľského rozhrania

Pomocou grafického rozhrania používateľ komunikuje s programom, preto je mimoriadne dôležité navrhnuť ho a implementovať tak, aby pôsobilo jednoducho a prirodzene objaviteľne, a tiež použiteľne.



Obr. 6.3: Koncept programu vo forme diagramu interakcie hlavných tried

Komplexné a rozsiahle používateľské rozhrania neposkytujú užívateľský zážitok (User Experience - UX²), na ktorý sú v súčasnosti kladená značné nároky a požiadavky. Ak je rozhranie navrhnuté tak aby bolo dobre použiteľné, pri interagovaní dochádza k istej miere pôžitku, z toho ako sa jednoducho a prirodzene používa, aktér je spokojný a pracuje rýchlejšie, keďže nemusí jednotlivé akcie dlho hľadať alebo rozmýšľať ako postupovať aby dosiahol svoj cieľ.

Všeobecne povedané ak je komunikácia čistá a jasná, nedochádza k nedorozumeniam a nezrovnalostiam. Táto paralela k používateľským rozhraniám je podľa nášho názoru veľmi výstižná, preto je naším cieľom navrhnuť rozhranie tak, aby nepôsobilo mäťuco, ale naopak jednoznačne a intuitívne. Pri návrhu sa sústreďíme na UX a použiteľnosť, tiež porovnáme viaceré rámce pre tvorbu grafického rozhrania, aby sme zvolili knižnicu poskytujúcu potrebnú funkcionálnosť.

6.3.1 Sled obrazoviek

Používateľské rozhranie sme sa rozhodli rozdeliť do viacerých obrazoviek tak, aby sme dosiahli väčšiu prehľadnosť. Ak by sme jednotlivé čiastkové úlohy rozmiestnili v rámci jednej obrazovky, užívateľa by sme tým mohli zmiast a znemožnili by sme mu jednoznačne identifikovať úlohy, resp. postupnosť krokov programu.

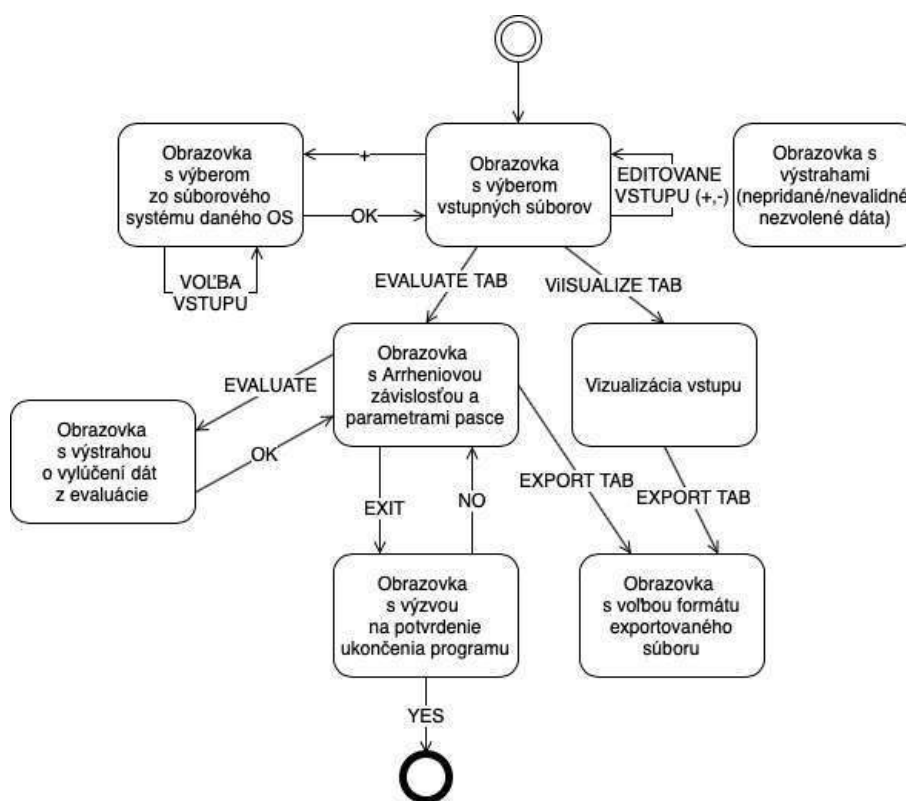
Kvôli vyššie uvedenému sme navrhli pomocou diagramu sledu obrazoviek, na Obr. 6.4, (Screen Sequence Diagram - SSQ³) postupnosť grafického zobrazenia jednotlivých krokov programu. Tento diagram sme vytvorili prihliadajúc na stavový diagram (Obr. 6.2), pričom prechody medzi stavmi sú vyjadrené šípkami s názvom tlačidla, alebo akciou, ktoré prechod zabezpečia. Vytvorenie tohoto diagramu nám ďalej uľahčuje proces prototypovania a aj samotnej implementácie.

6.3.2 Iteratívne prototypovanie

Na základe diagramov, ktoré boli pripravené a popísané v predošlých častiach sme vytvorili prvý papierový prototyp. Tento prototyp sme spolu s opisom scenáru predstavili užívateľom z domény, ale aj náhode vybraným. Pri vykonávaní scenáru sme merali čas, a tiež zaznamenávali kroky, ktoré predstavovali tú najnáročnejšiu úlohu pre testujúcich. Po jednotlivých testovaniach sme s aktérom ešte

²<https://www.usability.gov/what-and-why/user-experience.html>

³<https://kurzy.kpi.fei.tuke.sk/guis-en/student/07.html>



Obr. 6.4: Diagram sledu obrazoviek

prekonzultovali poznamenané časti a poznačili si ďalšie pripomienky a návrhy na úpravu.

Druhý prototyp sme vytvorili tak, že sme vychádzali z prvého, no zahrnuli sme návrhy od používateľov. Proces testovania a merania času testu sme opakovali, s tým, že sme oslovili ďalších ľudí, ktorí sa prvého testu nezúčastnili, aby sme zistili aká bude odozva pri prvom použití. Prirodzene priemerný čas použitia sa znížil, a aj noví používatelia dosiahli lepší priemerný čas, než užívatelia v prvej iterácii. Usudzujeme, že sme implementáciou zmien plynúcich z prvej iterácie podporili objaviteľnosť funkcionalít.

6.4 Výber platformy a technológií

Voľba použitých softvérových rámcov, technológií a v prvom rade platformy je nesmierne dôležitým krokom v procese návrhu. Výber platformy môže do určitej miery ovplyvniť budúcnosť a životný cyklus projektu, ktorého cieľom je vyvíjať a udržiavať softvér. Preto pri výbere treba klásť dôraz na aktuálne požiadavky, no

tiež uvažovať do budúcnosti a brať ohľad na možné rozšírenia a ďalšie aplikácie, či funkcie, ktoré bude potrebné upraviť, či navyše dodať.

Pozornosť chceme upriamiť hlavne na výber programovacieho jazyka, ktorý poskytuje multiplatformovosť, teda nebude limitovať využitie softvéru len na konkrétnej rodine operačného systému, či dokonca verzii (tak ako to bolo pri dosiaľ používanom licencovanom programe). Zvolená platforma teda musí povoľovať určitú mieru škálovateľnosti a možnosť optimalizácie, keďže ide pri tomto softvérovom riešení o zväčšené množstvo vstupných dát, ktoré majú byť spracované, analyzované a vizualizované. Optimalizácia sa môže dotknúť aj náročnosti pri matematických operáciách vykonávaných nad dátami.

Zvažovali sme programovacie jazyky, ktoré by mohli podať najlepší výkon, čo sa týka rýchlosti spracovania, no uvedomujeme si negatívne stránky vznikajúce pri takýchto riešeniach. Prvým adeptom, bol programovací jazyk C++⁴ predstavujúci nadstavbu štandardného jazyka C⁵, ktorý je pomerne blízko hardvéru, keďže nie je postavený na žiadnej virtualizačnej vrstve. Aj pôvodný licencovaný softvér bol vyvíjaný v tomto jazyku. Uvedomujeme si, že nie sú prístupné architektúry, ktoré by vedeli maximálne využiť rýchlosť takýchto implementácií. Takže teoreticky by sme s použitím takéhoto jazyka vedeli dosiahnuť rýchle riešenie, no to by bolo citeľné až pri veľmi veľkom množstve dát a hardvérovej architektúre s väčším počtom jadier procesoru, alebo inými vysoko-rýchlostnými rozšíreniami. Avšak keďže výpočtová technika, ktorá je pri tomto výskume používaná disponuje štandardnými parametrami, nevidíme možnosť využitia potenciálu takejto rýchlosti.

Ďalším, menej rýchlym adeptom na programovací jazyk, ktorý sme zvažovali je Python⁶. Aj napriek tomu, že vznikol pomerne dávno predstavuje perspektívnu cestu, keďže sa v súčasnosti často využíva pri dátovej analýze v kombinácii s rôznymi rámcami poskytujúcimi vizualizáciu. Nevýhodu týkajúcu sa rýchlosti prevyšujú možnosti softvérovej optimalizácie a paralelizácie, ale aj významne veľká komunita, vďaka ktorej je dostupná široká škála rámcov a knižníc. Dá sa povedať, že je momentálne veľmi populárnym jazykom, ktorý má pomerne jednoduchú syntax. Medzi jeho špecifiká patrí to, že je interpretovaný a interaktívny, pričom jeho vývoj je vedený ako open source projekt so súčasnou verzou 3.7.

⁴<https://isocpp.org>

⁵<http://www.bell-labs.com/usr/dmr/www/chist.html>

⁶<https://www.python.org>

Na základe vyššie uvedeného sme sa teda rozhodli pre implementáciu v jazyku Python, kvôli jeho vlastnostiam a širokému využitiu (*Defining the 20 Most Popular Programming Languages*, b. r.), rozvoju a rozšíreniam vytvoreným komunitou. Konkrétne použité matematické knižnice ozrejmime v nasledujúcej časti práce, pri opise implementácie.

Ďalej by sme ešte chceli popísať výber rámca pre grafické používateľské rozhranie. Navštívili sme viacero dostupných zdrojov, ktoré porovnávajú aktuálne možnosti tvorby (*5 open source Python GUI frameworks*, b. r., *The 6 Best Python GUI Frameworks for Developers*, b. r.) používateľského rozhrania ku natívnej aplikácii napísanej v jazyku Python. Vlastnosti vybraných multiplatformových technológií sme porovnali v Tab. 6.1.

Tabuľka 6.1: Porovnanie vybraných rámcov

Názov	Dostupnosť (licencie)	Nástroj na tvorbu GUI	Poznámka
<i>Kivy</i>	MIT License	Kivy Designer	multimediálne aplikácie
<i>PyQt</i>	komerčná a GPL	Qt Designer	oblúbený komunitou, prirodzený vzhľad na rôznych platformách
<i>Tkinter</i>	Python license	Pygubu	štandardný toolkit
<i>WxPython</i>	wxWindows Library License	wxDesigner	narastajúca komunita, prirodzený vzhľad na rôznych platformách

Z vybraných rámcov sme zvolili PyQt⁷, ktorý ponúka využitie populárnej knižnice Qt⁸. Tento rámec je multiplatformový a dostupný (*What is PyQt?*, b. r.) pod dvoma licenciami Riverbank Commercial License, ale aj GNU GPL (General Public License) v3. Navyše je podporovaný nástroj Qt Designer⁹, pomocou ktorého je možné namodelovať používateľské rozhranie, pričom tento nástroj vygeneruje potrebný kód, čo značne uľahčuje samotnú tvorbu rozhrania. Podľa našudovaných zdrojov je momentálne pre nový vývoj odporúčaná verzia PyQt5 a Qt v5.

⁷<https://riverbankcomputing.com/software/pyqt/intro>

⁸<https://www.qt.io>

⁹<https://www.qt.io/blog/2019/06/03/qt-design-studio-1-2-released>

V neposlednom rade by sme radi vyzdvihli množstvo dokumentácie dostupnej k vybranej technológii, čo poukazuje na to, že je pomerne jednoduché sa s ňou oboznámiť. Oblíbenosť tiež potvrdzuje široká komunita, veľa diskusií, návodov a tutoriálov.

7 Implementácia programu pre spracovanie a vizualizovanie dát

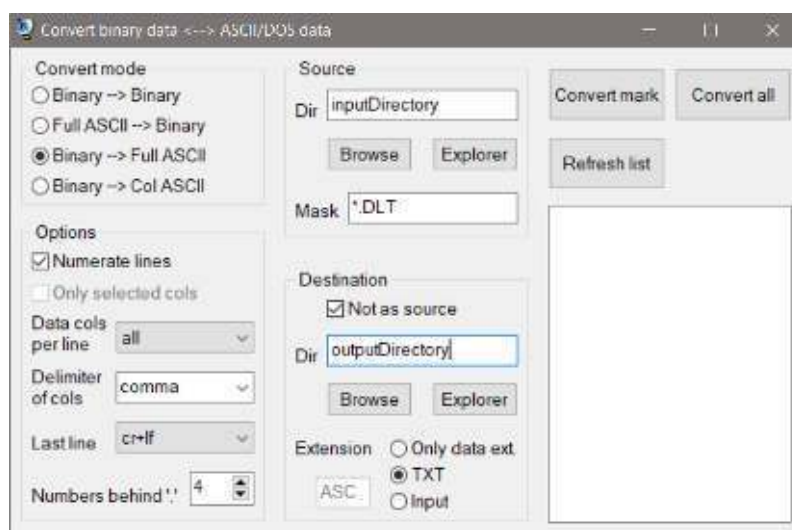
Podľa záverov analytickej časti a aparátu v nej zhrnutom, a taktiež berúc ohľad na prieskum v oblasti možností platforiem pre implementáciu sme navrhli vyššie popísané softvérové riešenie aplikácie. Zohľadnili sme naštudované informácie a vhodnosť dostupných softvérových možností, pričom sme sa snažili držať štandardného prístupu pri návrhu softvéru. Pokúsili sme sa obsiahnuť časť funkcionality licencovaného softvéru aby sme dosiahli menej komplexné riešenie, ktoré bude jednoduché pri používaní.

V tejto časti podrobne popíšeme jednotlivé časti softvéru a spôsob ich implementácie, ako aj štruktúru nameraných dát, ktoré nám boli poskytnuté pracovníkom. Okrem toho priblížime aj spôsob reprezentácie týchto dát v programe a ako dátový tok ovplyvňuje dostupnosť jednotlivých častí programu, pričom sa chceme poukázať na návaznosť na postupy zvolené pri návrhu.

7.1 Štruktúra vstupných dát

Pôvodné dáta generované licencovaným softvérom majú binárny formát s príponou *.DLT*, špecifickou pre toto softvérové riešenie. Avšak keďže ide o program chránený licenciou, dáta ním generované sú zabezpečené šifrovaním. To znamená, že nie sú čitateľné inými softvérmi, bez toho aby bolo presne dané akým kľúčom, alebo postupom boli zabezpečené.

Pretom sme zvolili možnosť preložiť binárne šifrované dáta do čitateľnej podoby pomocou programu, ktorý poskytovaný spolu s licencovaným softvérom. Definovanie pravidiel konverzie dát je možné vidieť na Obr. 7.1. Podľa zvolených pravidiel sme dáta preložili a ďalej očakávali, že ich forma bude dodržaná.



Obr. 7.1: Softvér na preklad dát

Prekladom bol zmenený typ súboru z binárneho na textový formát, no očakávame zachovanie názvu súboru, ktorý sám o sebe nesie informácie o priebehu meraní. A práve názov využívame prvotne pri validácii vstupných súborov. Má totiž obsiahnuté informácie nielen o type merania respektíve type dát, ale aj o sade, sete a počte meraní celého experimentu. Keďže uvažujeme dva typy vstupných súborov, osobitne popíšeme štruktúru ich názvu, ale aj vnútornú hierarchiu informácií.

Prvým podporovaným typom sú Tempscan súbory, ako demonštratívny príklad uvedieme nasledujúci názov *TS_925@A_012.T1A.TXT*. Znak *@* a *.* oddeľujú jednotlivé informácie, ktoré názov poskytuje. Presnejšie prvá časť (*TS_925*) reprezentuje sadu, nasledujúca (*A_012*) označuje set a posledná (*T1A*) určuje typ merania, konkrétne to, že ide o Tempscan súbor. Vnútorná štruktúra pozostáva z hlavičky a viacerých tzv. *PTV* blokov. Každé pole hlavičky sa nachádza na novom riadku, tak isto aj každý blok dát sa nachádza na jednom riadku, pričom jednotlivé položky sú oddelené čiarkou. Pri parsovaní vstupu sme využili tieto vlastnosti, a tak identifikovali jednotlivé časti a uložili ich do podoby, v ktorej budú ľahšie dosiahnuteľné programom.

Túto formu popíšeme po ozrejmnení ďalšieho typu a tým je Transient súbor, ktorého názov môže byť nasledovný *T001.Y1A*. Prvá časť (*T001*) predstavuje označenie poradia transienty a v druhej časti (*Y1A*) je dôležité *Y*, čo značí, že ide o Transient súbor. Čísllice v druhej časti sa pohybujú v intervale 1 až 4, čo sa odkazuje

na nastavenie priebehu merania. Vnútrná štruktúra obsahuje tiež hlavičku v rovnakej forme ako pri Tempscan súboroch, no dátová časť pozostáva z dvojíc hodnôt oddelených opäť čiarkou a novým riadkom.

Oba typy dát disponujú hlavičkou s rovnakou formou, preto sme definovali triedu *DataFile*, od ktorej budú triedy *TransientFile* a *TempscanFile* dediť, aby sme zamedzili duplicitu kódu a využili princípy objektovo orientovaného programovania. Rodičovská trieda teda obsahuje ako členskú premennú list reprezentujúci dáta z hlavičky. Pomocou metód, ktoré pristupujú k prvkom listu skrz index dosiahneme to, že vieme jednoznačne pristúpiť k informácii obsiahnutej v hlavičke. Takéto metódy sme nazvali príznačne, aby jednoznačne značili aké dáta z hlavičky poskytujú.

Podobne pomocou indexu a príznačného názvu je možné pristupovať k dátam z jednotlivých *PTV* blokov, ktoré sú reprezentované v triede *TempscanFile* ako členská premenná typu list, pričom každá položka prezentuje jeden blok vo forme osobitného listu. V triede *TransientFile* sú dáta reprezentované dvoma listmi, a to jedným pre reálnu a ďalším pre imaginárnu zložku komplexného čísla.

Ďalšie dáta v programe sme nereprezentovali vo forme samostatných tried, ale ako dátové štruktúry typu slovník (*dict*). Volili sme túto možnosť pre jednoduchosť používania a volania pomocou definovaného kľúča, konkrétne ako návratovú hodnotu príslušných funkcií.

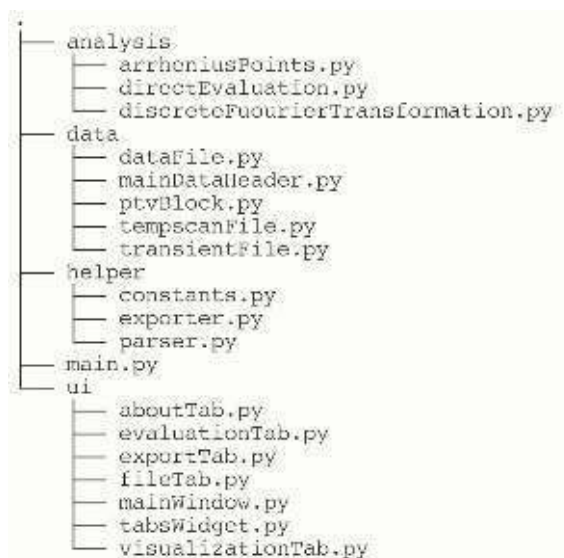
7.2 Priblíženie softvérového riešenia

Podrobnejšie sa budeme venovať štruktúre projektu a dôvodom, prečo sme zvolili dané členenie. Popíšeme aj obmedzenia používateľského rozhrania, berúc ohľad na stanovenú logiku a jednotlivé funkcionality programu, ktoré sú rozčlenené do samostatných obrazoviek tzv. *Tabov*.

7.2.1 Štruktúra projektu

Projekt s hierarchiou balíčkov zobrazenou na Obr. B.2 je dostupný na vzdialenom repozitári *GitHub*, pričom už počas vývoja sme využívali repozitár na správu verzií a zálohu. Jednotlivé funkcie sme rozdelili do osobitných tried reprezentovaných súbormi, pričom príslušnosť v danom balíčku zaručuje prehľadnosť zabezpečenú logikou usporiadania. Aj samotné užívateľské rozhranie sme rozdelili

do viacerých tried aby sme sa vyhli príliš komplexným riešeniam v jednom súbore.



Obr. 7.2: Štruktúra projektu a obsah jednotlivých balíčkov

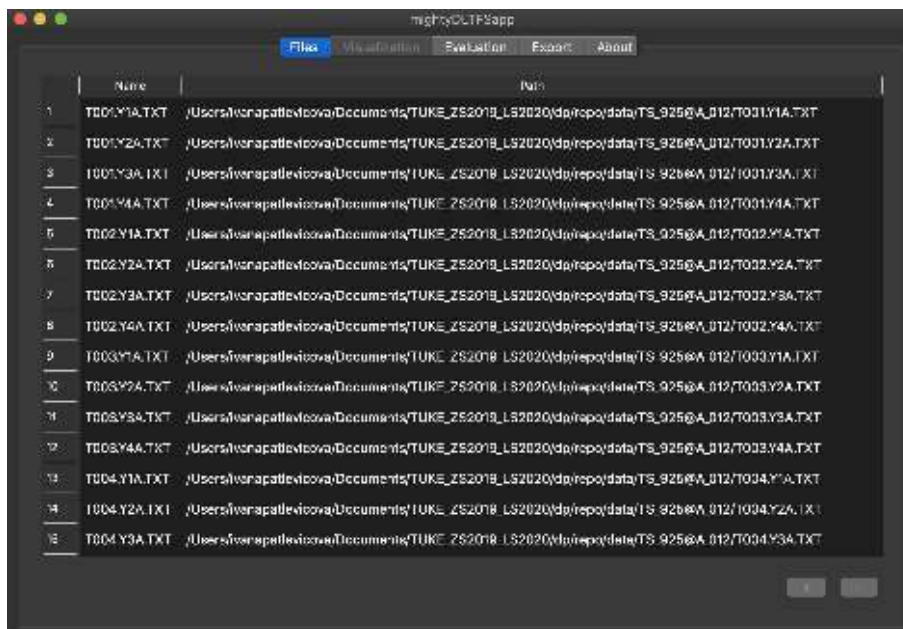
7.2.2 Používateľské rozhranie a jeho ohraničenie

Po spustení programu je používateľovi zobrazené používateľské rozhranie, ktoré je rozdelené podľa funkcií aplikácie do jednotlivých záložiek, alebo tzv. tabov aby bola dosiahnutá vyššia prehľadnosť.

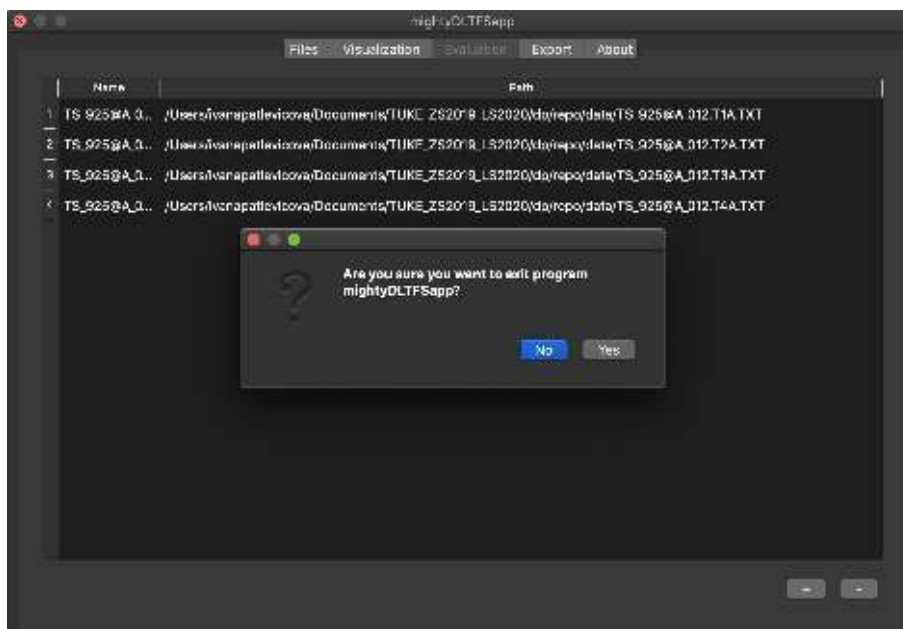
Možnosť prechodu do jednotlivých tabov je podmienená logikou programu, ktorú sme navrhli vo forme diagramov (Obr. 6.2 a 6.4). Až po pridaní vstupných súborov je možné prejsť z tabu *Files* do nasledujúcich podľa toho aké súbory boli zvolené za vstup. Tab *Visualization* je primárne určený pre *Tempscan* súbory na ich vizualizáciu a zhrnutie ich dôležitých parametrov v prehľadnej tabuľke. *Transient* súbory sú vstupom pre *Evaluation* tab. Možnosť exportovania grafov zo záložiek *Evaluation* a *Visualization* je dostupná v tabe *Export*. Tento tab je povolený, ak je vytvorený graf daného typu.

Za predpokladu, že v spustenom programe boli pridané vstupné dáta môže užívateľské rozhranie vyzeráť ako na Obr. 7.3, kedy je povolená evaluácia, kvôli pridaným transientam. Po úspešnom dokončení scenáru, ale aj kedykoľvek používateľ uzná za vhodné, môže program ukončiť. Bude upozornený oznámením

pre potvrdenie ukončenia (Obr. 7.4). Na obrázku s týmto oznámením môžeme vidieť, že pri pridaných transient dátach je povolená vizualizácia.



Obr. 7.3: Pridané vstupné súbory na úvodnej obrazovke



Obr. 7.4: Potvrdenie ukončenia aplikácie

Využitím *PyQt5* sme zabezpečili, že aplikácia bude pôsobiť prirodzene na každej platforme, keďže sú využívané grafické komponenty s dizajnom danej plat-

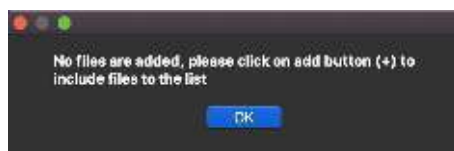
formy, na ktorej je program spustený. Pre jazyk používateľského rozhrania sme použili angličtinu, keďže mnohé využívané termíny v rozhraní poskytujú presnejšiu informáciu práve v tomto jazyku. Knižnicu *PyQtGraph*¹ sme zvolili pre grafické elementy vizualizácie a evaluácie, keďže poskytuje potrebné matematické a vedecké typy grafov v graficky priaznivej podobe. Navyše je možné si konkrétnu časť grafu priblížiť alebo oddialiť podľa potreby používateľa pre podrobnejšie preskúmanie dopsiahnutých výsledkov.

7.2.3 Výber vstupných dát, kontrola konzistencie a štruktúry

Dôležitou podmienkou fungovania implementovaného softvéru je voľba korektných dát. Štruktúru a spôsob interpretácie sme už popísali vyššie a ďalej by sme chceli naviazať na nami vytvorenú triedu *Parser*, ktorá ma za úlohu identifikovať validné a konzistentné dáta a preložiť ich do programom čitateľnej podoby. V tejto triede sú definované viaceré metódy pre validáciu, a tiež pre postupné parsovanie jednotlivých častí vstupných súborov (hlavička, *PTV* bloku a pod.).

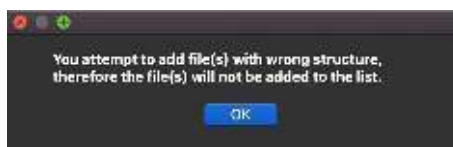
Z vyššie uvedeného teda vyplývajú ďalšie obmedzenia používateľského rozhrania, ktoré sú užívateľovi oznamované vo forme výstrah s informáciou:

- o nepridaní súborov (Obr. 7.5),
- o nedodržanej štruktúre dát (Obr. 7.6),
- o duplicitě vstupných súborov (Obr. 7.7),
- o pokuse o zmazanie nedefinovaného súboru (Obr. 7.8).

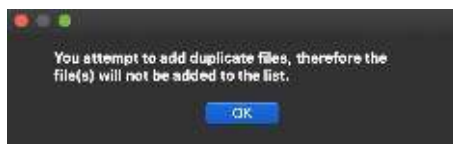


Obr. 7.5: Výstraha o nepridaných vstupných súboroch

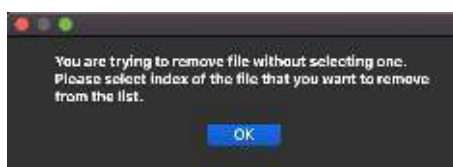
¹<http://www.pyqtgraph.org>



Obr. 7.6: Výstraha o nekorektnej štruktúre vstupného súboru



Obr. 7.7: Výstraha o pokuse o pridanie duplicitného súboru



Obr. 7.8: Výstraha o pokuse o odstránenie súboru bez jeho zvolenia

Ak prebehla validácia a parsovanie dát správne, výsledný zoznam objektov je uložený v triede *TabsWidget*, kde reprezentuje stále aktualizovaný obsah. Tento zoznam je dostupný v jednotlivých taboch ako *@property*. Z triedy *FileTab* je jeho hodnota aktualizovaná podľa zadaných validných vstupných súborov a triedy *EvaluationTab* a *VisualizationTab* využívajú dostupné dáta pre realizáciu ich konkrétnych funkcionalít. Pri každej zmene vstupu sú dáta v jednotlivých záložkách aktualizované.

7.2.4 Vizualizácia vstupných súborov

V záložke vizualizácia je možné zobrazíť vstupné dáta zadané v záložke *Files*. Avšak je možné tak vykonať len s jedným typom dát, teda môžeme vizualizovať *Tempscan* súbory a zobrazíť ich parametre v prehľadnej tabuľke (Obr. 7.9), alebo je možné zobrazíť obsah maximálne siedmich *Transient* súborov (pri väčšom množstve by došlo ku neprehľadnosti).

Trieda *VisualizationTab* má referenciu na stále aktuálny zoznam dát zadaných ako vstup, preto sa dynamicky dostupnosť tejto záložky mení, podľa aktuálneho stavu. V triede je okrem iného definovaná inicializácia grafického rozhrania a metóda pre jeho aktualizáciu podľa vstupu. Logika dostupnosti záložky je defino-

vaná v triede *TabsWidget*, ktorá za určitých podmienok (napr. validita, počet a typ vstupu) mení stav tabu.



Obr. 7.9: Záložka s vizualizáciou Tempscan súborov a zobrazenie ich parametrov v tabuľke

7.2.5 Evaluácia vstupu

Vychádzajúc z matematicko-fyzikálnych podkladov, ktoré sme analyzovali v kapitolách 4 a 3 a využitím knižníc ako *PyQtGraph*, *NumPy* a *scikit-learn*² (poskytuje prediktívnu dátovú analýzu), sme implementovali záložku *Evaluation* (Obr. A.11). Po jej inicializácii a za predpokladu, že sú zvolené vstupné súbory typu *Transient* sú zobrazené informácie o dátach v časti *Data parameters* a zvyšné časti obrazovky sú pripravené na naplnenie dátami, po definovaní parametrov evaluácie a stlačení príslušného tlačidla. Uživatelské rozhranie je ošetrené, tak aby zamedzilo nedefinovaniu parametrov a pokusu o evaluáciu.

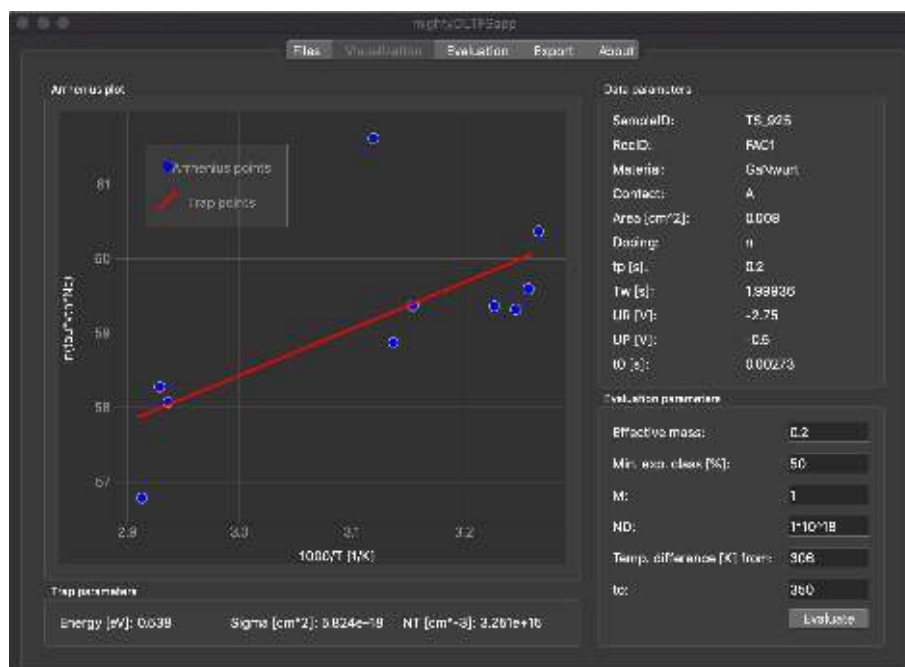
Pre odfiltrovanie bodov Arrheniovej závislosti a dosiahnutie potrebnej úrovne presnosti a spoľahlivosti evaluácie užívateľ môže dedefinovať príslušné parametre

²[https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)

[LinearRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)

v časti *Evaluation parameters*, pričom sú povolené ako vstup celé, ale aj desatinné čísla príslušného rozsahu. Tieto parametre určujú faktor citlivosti pri evaluácii. Po kliknutí na tlačidlo *Evaluate* budú zadané parametre zohľadnené a dáta nevyhovujúce definovaným podmienkam sú vylúčené. Na výstupnom grafe je teda zobrazená výsledná Arrheniova závislosť pozostávajúca z vyhovujúcich bodov a lineárna regresia (reprezentujúca pascu) je znázornená priamkou.

Úspešným scenárom je teda dosiahnutie priamej evaluácie s aplikovanými parametrami definovanými užívateľom, v podobe grafu obsahujúceho body Arrheniovej závislosti a vyznačené body reprezentujúce pascu. Pričom v časti *Trap parameters* sú pro evaluácii zobrazené vypočítané parametre danej pasce.



Obr. 7.10: Záložka s evaluáciou

7.2.6 Export analyzovaných a vizualizovaných dát

Okrem spomínaných funkcionalít realizujúcich zobrazenie a evaluáciu, sme chceli zahrnúť aj možnosť exportovania vizualizovaných dát, preto sme sa rozhodli vyčleniť túto možnosť do samostatnej časti užívateľského rozhrania.

No použitá knižnica *PyQtGraph* poskytuje exportovanie a modifikovanie grafu priamo kliknutím na grafický element. V tomto prípade sú pre export a konverziu

dostupné formáty ako *png*, *jpg*, *tiff*, *svg* alebo *csv*. Po zvolení formátu a lokácie je exportovaný súbor uložený.

Avšak my sme sa rozhodli využiť aj *API* (*Application Programming Interface*), ktoré knižnica poskytuje a doplnili sme funkcionality pre export z osobitnej záložky (Obr. 7.11), aby bola táto funkcia ľahšie objaviteľná. Funkcionality potrebná pre exportovanie je dodefinovaná v triede *Exporter*. Po zadaní názvu a zvolení požadovaného formátu (*png*, *jpg* alebo *tiff*) je možné po stlačení príslušného tlačidla exportovať graf zo záložiek *Evaluation* alebo *Visualization* (podľa typu zadaných vstupných súborov).



Obr. 7.11: Záložka s možnosťami exportovania grafu do súboru

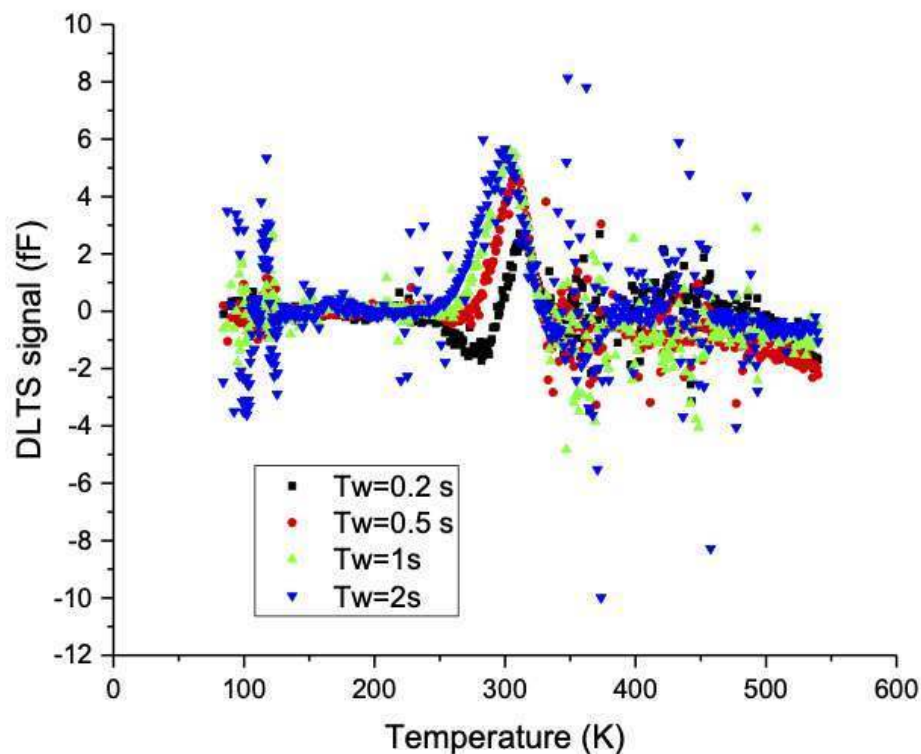
8 Výsledky práce

Výsledky analýzy poukázali na zložitosť využívaného programu, ktorý nie je prispôbený pre špecifické požiadavky pracoviska na Ústave elektroniky a fotoniky, FEI STU v Bratislave. Preto sme navrhli a implementovali softvérové riešenie poskytujúce vybranú funkcionalitu potrebnú pri vyhodnocovaní nameraných dát štruktúr na báze GaN v užívateľsky priaznivejšej podobe.

V procese návrhu sme realizovali testovanie prototypu v dvoch iteráciách, pričom sme študovali ťažšie objaviteľné časti a pripomienky užívateľov. Indikované nedostatky sme pretvorili a integrovali do samotného programu. Týmto procesom sme podporili úroveň jednoduchosti a objaviteľnosti funkcií programu, čo v porovnaní s pôvodným - komplexným licencovaným softvérom prináša vyššiu úroveň užívateľského zážitku (UX). To prispieva spolu s jednoduchosťou používania k zrýchleniu a zefektívneniu procesu vyhodnocovania nameraných dát.

Naše softvérové riešenie poskytuje možnosť vizualizácie nameraných dát v prehľadnej forme, spolu s tabuľkou príslušných parametrov (Obr. 8.2), pričom pôvodný softvér nedokázal vizualizovať viac než dve merania, teda nebola podporená možnosť porovnania viacerých dát naraz. Takúto vizualizáciu bolo možné pôvodne realizovať len pomocou ďalšieho programu, pričom bolo nutné dáta exportovať a spracovať do podoby vhodnej pre zvolený grafický softvér (Obr. 8.1).

Ďalej sme implementovali možnosť evaluácie dát s tým, že užívateľ dodefinuje parametre potrebné pre určenie presnosti analýzy. Táto funkcionalita poskytuje tiež zobrazenie parametrov vstupných dát a hodnoty parametrov nájdenej pasce v prehľadnej podobe (Obr. A.11). Implementovali sme aj exportovanie vizualizovaných dát vo viacerých formátoch, a to okrem východzej možnosti knižnice, aj využitím aplikačného programového rozhrania knižnice *PyQtGraph*. Validitu dát sme prediskutovali s pracovníkmi domény a možné vylepšenia sme sformulovali v samostatnej kapitole našej práce.

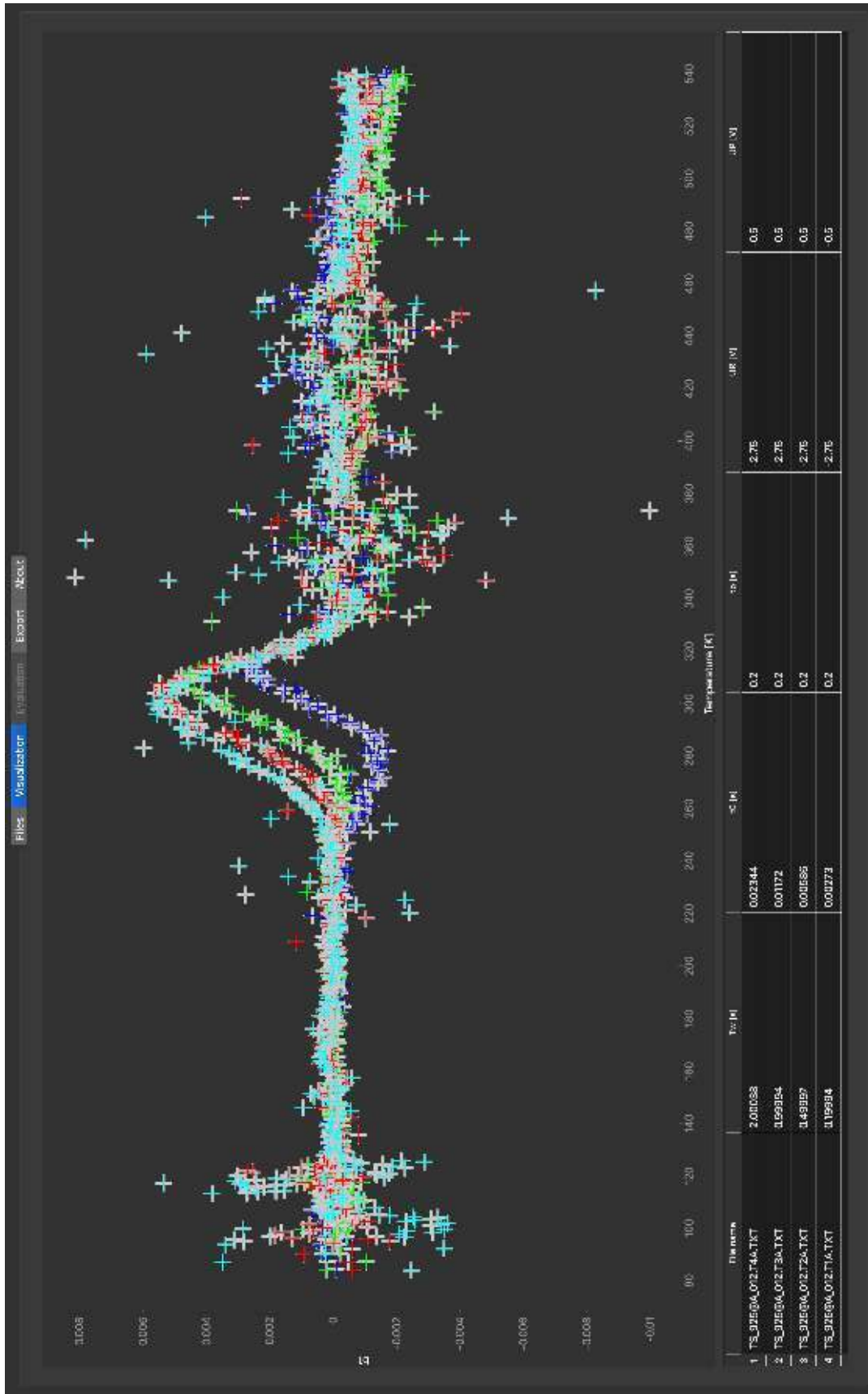


Obr. 8.1: Vizualizácia dát pomocou programu Origin

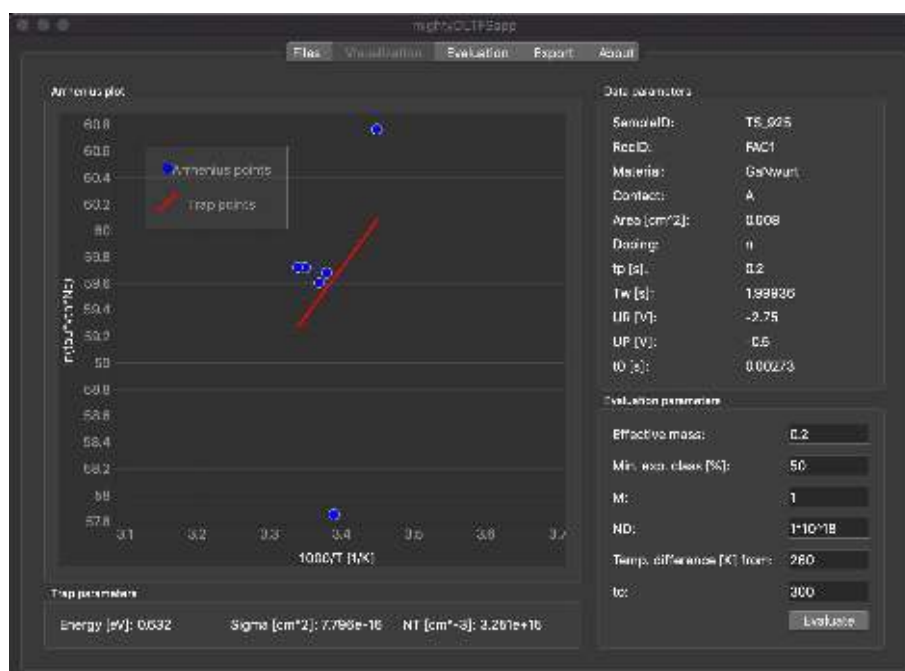
Okrem vyhodnotenia z hľadiska použiteľnosti a jeho možností, sme implementovaný program ešte počas návrhu vyhodnocovali aj z hľadiska bezpečnosti. Realizovanie bezpečnostnej analýzy aplikácie našej diplomovej práce nám poskytlo hlbší pohľad do danej problematiky. Poukázala na jednotlivé hrozby týkajúce sa zabezpečenia softvéru. Vďaka tejto analýze, tiež vďaka spracovaniu jednotlivých hrozieb sme dospeli k uvedomeniu si podstaty a dôležitosti zabezpečenia. Keďže bola aplikácia v procese tvorby, bolo jednoduché možné vylepšenia bezpečnosti zahrnúť do implementácie (najmä pri spracovaní vstupu a exportovaní dát), pričom sme sa vyhli dodatočnej modifikácii dokončenej aplikácie, čo predstavuje zdĺhavejší a náročnejší proces.

Využili sme modelovanie hrozieb, ktoré slúži k analyzovaniu bezpečnosti softvéru. Týmto prístupom sme posúdili uvažované riziká a možnosti zabezpečenia. Využívaný softvér *Microsoft Threat Modelling Tool*¹ nám pomohol identifikovať a kvantifikovať hrozby, či bezpečnostné problémy. Vytvorili sme model aplikácie pre vizualizáciu a analýzu dát, potrebný pre vykonanie bezpečnostnej analýzy

¹<https://www.microsoft.com/en-us/download/details.aspx?id=42518>



Obr. 8.2: Finálna vizualizácia tempscan súborov a ich parametrov



Obr. 8.3: Evaluácia dát

programom. Podľa výstupu analýzy vo forme správy sme kategorizované hrozby, popísali a zhodnotili možnosť ich výskytu a navrhli potrebné opatrenia.

Dúfame, že naše softvérové riešenie a jeho ďalšie prípadné modifikácie poskytnú potrebnú funkčnosť aj v praxi, a že uľahčí časovo náročný proces evaluácie a analyzovania dát nameraných na špecifických polovodičových štruktúrach.

9 Ďalšia práca

Softvérové riešenie, ktoré sme navrhli, implementovali v jazyku *Python* a popísali v predošlých kapitolách pokrýva základnú požadovanú funkcionálnu pre evaluáciu dát získaných pri experimentálnych meraniach polovodičových štruktúr na báze GaN. Považujeme ho za akýsi štartovací bod, na ktorom je možné stavať a ďalej dopĺňať ďalšie špecifické funkcionality, či metódy a postupy vhodné pre vyhodnocovanie materiálov. Vidíme aj perspektívu zefektívnenia procesu, a to vo forme paralelizácie výpočtov a operácií vykonávanými nad dátami.

Vytvorenú implementáciu by bolo možné tiež doplniť o viaceré parametre evaluácie a bližšie študovať vplyv ich zmeny na dosiahnuté výsledky. Týmto postupom by sa dalo dosiahnuť ešte citlivejšie vyhodnotenie, ktoré by poskytlo spoľahlivejšie výstupy. Vidíme aj možnosť integrovania už implementovaného riešenia pre filtráciu a výber parametrov nameraných dát, ktoré bolo vytvorené v jazyku *Java* a preto by bolo potrebné logiku programu preložiť do teraz využitej platformy.

Implementovaný softvér podporuje vizualizáciu a spracovanie dvoch typov dát, a to *Tempscan* a *Transient* dáta. Navrhujeme integrovanie ďalších typov, ktoré sú generované pri experimentálnych meraniach. Vyžadovalo by si to však bližšie preštudovanie ich štruktúry a pochopenie reprezentácie týchto dát. Pravdaže by bolo potrebné pochopiť aj podstatu úkonov, ktoré môžu byť nad takýmito dátami realizované.

Ďalej navrhujeme podporu exportovania viacerých formátov, nie len grafických, ale aj textových, pre lepšiu integráciu s viacerými dostupnými štandardnými softvérmi. Za dobrý príklad považujeme napríklad vytvorenie *JSON* súboru, alebo *excel* dokumentu. Vidíme možnosť aj v exportovaní jednotlivých častí výstupu (všetky body Arrheniovej závislosti alebo len identifikované pasce).

V neposlednom rade je tu obrovský potenciál v sústredení sa na vývoj použí-

vatelského rozhrania. Pochopiteľne s rastúcim množstvom funkcionalít sa môže zhoršovať prehľadnosť a objaviteľnosť používateľského rozhrania. V našom riešení sme sa snažili vytvoriť ľahko objaviteľné používateľské rozhranie, ktoré bude aj jednoducho použiteľné a priaznivé z hľadiska UX. Avšak po doplnení ďalších funkcionalít bude potrebná istá prestavba používateľského rozhrania, aby bola udržiavaná prehľadnosť a jednoduchosť. Preto by sme nechali osobitný priestor po vytvorení komplexnejšieho *backend*-u (logiky programu) na predizajnovanie a vytvorenie nového *frontend*-u, teda používateľského rozhrania.

10 Záver

V spolupráci s Ústavom elektroniky a fotoniky, FEI STU v Bratislave sme realizovali túto diplomovú prácu zameranú na zefektívnenie procesu evaluácie pri experimentálnej činnosti týkajúcej sa progresívnych polovodičových štruktúr na báze nitridu gália - GaN. Výskum v oblasti polovodičových materiálov a ich poruchovosti je doménou uvedeného pracoviska.

Venovali sme sa bližšie ozrejmieniu podstaty takejto výskumnej činnosti zameranej na GaN štruktúry a celkovému prínosu nameraných a interpretovaných výsledkov. Analyzovali sme metódu Fourierova spektroskopu hlbokých hladín (Deep-Level Transient Fourier Spectroscopy - DLTF), ktorú využíva licencovaný softvér používaný pracoviskom, aby sme nadobudli potrebný matematicko-fyzikálny aparát a pochopili princíp jej fungovania. Následne sme sa sústredili na samotný softvér a jeho funkcie. Keďže predstavuje veľmi komplexné riešenie, postupovali sme tak, že sme odčlenili funkcionalitu špecifickú pri vyhodnocovaní materiálov na báze GaN a zamerali sme sa na jej podrobnejšie preskúmanie. Identifikovali sme nedostatky, ktoré spôsobujú časovú náročnosť procesu vyhodnocovania, a preto robia samotnú analýzu nameraných dát časovo náročnou a znižujú jej efektívnosť.

Podľa výstupov analýzy sme spolu so špecifikovaním požiadaviek od samotných zamestnancov pracoviska pokračovali návrhom softvérového riešenia pre vizualizáciu a spracovanie nameraných dát. Návrh sme spracovali podľa štandardov využívaných v softvérovom inžinierstve. Zjednodušením a odčlenením potrebných funkcií, spolu s prehľadným používateľským rozhraním sme sa snažili podporiť použiteľnosť a efektívnosť pri využívaní navrhovaného programu. Separátne sme sa venovali aj používateľskému rozhraniu, aby sme zachovali jeho jednoduchosť a intuitívnosť, s cieľom vyhnúť sa komplexnému a neprehľadnému riešeniu prinášajúcemu ťažko objaviteľné funkcionality. Pre dosiahnutie tohoto cieľa, sme

realizovali testovanie navrhnutých prototypov v dvoch iteráciách, kedy sme získali spätnú väzbu a pripomienky na vylepšenie rozhrania, ktoré sme následne aj do implementovaného softvérového riešenia zapracovali.

Aplikáciu sme implementovali podľa popísaného softvérového návrhu. Pre implementáciu sme zvolili aktuálne často využívaný jazyk práve pri dátovej analýze - *Python*, pričom dôvody jeho výberu, ako aj voľbu ostatných technológií sme ozrejmili pri návrhu. Pre grafické užívateľské rozhranie sme volili knižnicu *PyQt5* predstavujúcu riešenie s prirodzeným dizajnom grafických komponentov na ktorejkoľvek podporovanej platforme. Docielili sme multiplatformové softvérové riešenie podporujúce efektívnosť procesu dátovej analýzy v oblasti vyhodnocovania poruchovosti špecifických štruktúr, navyše podporujúce možnosť exportovania výsledkov evaluácie a vizualizácie dát. Implementované riešenie sme otestovali na nameraných dátach a výsledky sme prekonzultovali so samotnými členmi domény.

Keďže paradigmou spadá jazyk našej implementácie aj medzi objektovo-orientované programovanie, snažili sme sa využiť jeho princípy a vytvoriť prehľadný a modulárny zdrojový kód. To by malo podporiť jednoduché doplnenie uvedených, ale aj ďalších možných rozšírení. Na záver sme vypracovali potrebnú dokumentáciu k nášmu riešeniu. V závere oceňujeme príležitosť vyskúšať si používanie nových technológií, platforiem a knižníc, čo obohatilo naše zručnosti a vedomosti.

Táto práca vznikla s podporou Vedeckej grantovej agentúry MŠVVaŠ SR a SAV, projekt VEGA 1/0668/17 a zo spoločného podniku ECSEL JU na základe dohody o grante č. 783274, projekt 5G_GaN2. Tento JU dostáva podporu z programu výskumu a inovácií Európskej únie v rámci programu Horizont 2020 a Francúzska, Nemecka, Slovenska, Holandska, Švédska, Talianska, Luxemburska, Írska.

Literatúra

5 open source Python GUI frameworks [<https://opensource.com/resources/python/gui-frameworks>]. [Online; accessed 14-March-2020].

Defining the 20 Most Popular Programming Languages [<https://towardsdatascience.com/one-word-to-define-each-of-the-20-most-popular-programming-languages-b1ef06ca8716>]. [Online; accessed 14-March-2020].

OZPINECI, B.; TOLBERT, L. M. *Information Bridge: DOE Scientific and Technical Information. Comparison of Wide-Bandgap Semiconductors* [<http://www.ornl.gov/~webworks/cppr/y2001/rpt/118817.pdf>]. [Online; accessed 23-August-2019].

PhysTech [<http://www.phystech.de/products/dlts/dlts.htm>]. [Online; accessed 27-October-2019].

PhysTech, Software Manual [<http://www.phystech.de/products/dlts/dlsoftw.htm>]. [Online; accessed 27-October-2019].

PhysTech, Theory Manual [<http://www.phystech.de/products/dlts/dltstheo.htm>]. [Online; accessed 27-October-2019].

SAMCO. *AlGaIn/GaN Etch for GaN Power Device Fabrication* [<https://www.samcointl.com/featured-solutions/algan-and-gan-etch/>]. [Online; accessed 10-January-2020].

State Machine Diagrams [<https://www.uml-diagrams.org/state-machine-diagrams.html>]. [Online; accessed 8-January-2020].

The 6 Best Python GUI Frameworks for Developers [<https://blog.resellerclub.com/the-6-best-python-gui-frameworks-for-developers/>]. [Online; accessed 14-March-2020].

- UML Class and Object Diagrams Overview* [<https://www.uml-diagrams.org/class-diagrams-overview.html>]. [Online; accessed 8-January-2020].
- Use Cases* [<https://www.usability.gov/how-to-and-tools/methods/use-cases.html>]. [Online; accessed 27-February-2020].
- User Experience Basics* [<https://www.usability.gov/what-and-why/user-experience.html>]. [Online; accessed 25-November-2019].
- What is PyQt?* [<https://riverbankcomputing.com/software/pyqt/intro>]. [Online; accessed 14-March-2020].
- CHEN, A., 2019. *Gallium nitride is the silicon of the future* [In: Gallium Nitride Valley <https://www.theverge.com/2018/11/1/18051974/gallium-nitride-anker-material-silicon-semiconductor-energy>]. [Online; accessed 14-September-2019].
- DROBNÝ, J., 2018. *Elektrická charakterizácia MOS-HEMT štruktúr na báze GaN*. Diplomová práca. FEI STU v Bratislave.
- PATLEVIČOVÁ, I., 2018. *Výber a vizualizácia dát získaných z experimentálnych meraní v polovodičových infraštruktúrach*. Bakalárska práca. FEI TUKE.
- ŠAROCKÝ, T., 2018. *Vyhodnocovanie dát z experimentálnych meraní polovodičových materiálov a štruktúr*. Diplomová práca. FEI TUKE.
- MENEGHINI, M.; MENEGHESSO, G.; ZANONI, E., 2017. *Power GaN Devices* [In: Springer International Publishing]. [Online; accessed 24-April-2020].
- KÓSA, A., 2016. *Deep Level Transient Spectroscopy study of emission and capture processes in multilayer semiconductor structures*. Dizertačná práca. FEI STU v Bratislave.
- KÓSA, A.; STUHLÍKOVÁ, L.; HARMATHA, L.; MIKOLÁŠEK, M.; KOVÁČ, J.; SCIANA, B.; DAWIDOWSKI, W.; RADZIEWICZ, D.; TLACZALA, M., 2016. Defect distribution in InGaAsN/GaAs multilayer solar cells. *In Solar Energy*. Roč. 132, s. 587–590. ISSN 0038-092X.
- POHORELEC, O., 2016. *Kvalita polovodičových štruktúr a prvkov na báze InGaAs*. Diplomová práca. FEI STU v Bratislave.
- POLYAKOV, A. Y.; LEE, I. H., 2015. *Deep traps in GaN-based structures as affecting the performance of GaN devices* [In: Materials Science and Engineering <https://doi.org/10.1016/j.mser.2015.05.001>]. [Online; accessed 28-April-2019].

- KÓSA, A., 2011. *Skúmanie vlastností polovodičových štruktúr metódou DLTS*. Bakalárska práca. FEI STU v Bratislave.
- ŠEBOK, J., 2010. *Diagnostika polovodičových štruktúr a materiálov spektroskopiou hlbokých hladín*. Písomná práca k dizertačnej skúške, FEI STU v Bratislave.
- SALBU, B.; SKIPPERUD, L., 2008. *NATO Advanced Workshop on Radiological Risks in Central Asia* [Springer]. Nuclear risk in Central Asia.
- GOUDON, T.; MILJANOVIC, V.; SCHMEISER, Ch., 2007. On the Shockley-Read-Hall Model: Generation-Recombination in Semiconductors. *SIAM Journal of Applied Mathematics*. Roč. 67, s. 1183–1201. Dostupné z DOI: 10.1137/060650751.
- PIETRIKOVÁ, A.; BANSKÝ, J., 2007. *Základy inžinierstva materiálov*. Letná 9, 042 00 Košice: elfa, s.r.o. ISBN 80-8086-001-7.
- DEEN, M. J.; PASCAL, F., 2006. Electrical characterization of semiconductor materials and devices. In: *Springer Handbook of Electronic and Photonic Materials*. Springer, s. 409–438.
- SCHRODER, D. K., 2006. *Semiconductor material and device characterization*. John Wiley & Sons.
- REDHAMMER, R.; STUHLÍKOVÁ, L.; HULÉNYI, L., 2005. *Elektronické prvky a systémy*. STU v Bratislave FEI.
- FOWLER, M., 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)*. Addison-Wesley Professional. ISBN 0321193687.
- SIEGHARD, W., 1991. *Semiconductor Investigations with the DLTFs*. Dizertačná práca. Department of Physics Gesamthochschule Kassel University of the Country of Hessen.
- WEISS, S., 1991. *Semiconductor Investigations with the DLTFs (Deep-Level Transient Fourier Spectroscopy) Method*. Dizertačná práca. Department of Physics Gesamthochschule Kassel University of the Country of Hessen.
- WEISS, S.; KASSING, R., 1988. Deep Level Transient Fourier Spectroscopy (DLTFs) - A technique for the analysis of deep level properties. *Solid-State Electronics*. Roč. 31, č. 12, s. 1733–1742.
- LANG, D. V., 1974. Deep-Level Transient Spectroscopy: A New Method to Characterize Traps in Semiconductors. Roč. 45, s. 3023–3032.

Zoznam skratiek

A amplitúda exponenciálnej kapacitnej transienty (funkcie).

A_R smernica regresnej priamky.

B posunutie exponenciálnej kapacitnej transienty (funkcie).

B_R priesečník regresnej priamky.

C kapacita.

C_0 celková kapacita nameraná pri reverznom napätí.

E_A akceptorová hladina.

E_C energia dolného okraja vodivostného pásma.

E_D donorová hladina.

E_F Fermiho hladina.

E_T energetická hladina.

E_V energia horného okraja valenčného pásma.

E_g zakázané pásmo.

$F(N)$ diskretná Fourierova transformácia.

$F(j\omega)$ všeobecná Fourierova transformácia.

N počet diskretných ekvidištančných časových intervalov.

N_C hustota stavov vo vodivostnom pásme.

N_D koncentrácia donorových prímiesí.

N_T koncentrácia pascí.

SRH Shockley-Read-Hall štatistika.

T absolútna teplota.

T^2 teplotná korekcia.

T_W šírka periódy snímanej kapacitnej transiency.

U_P plniaci impulz.

U_R záverné napätie.

X_n faktor entropie.

ΔE_T aktivačná energia pasce.

$\nu_{th,n}$ tepelná rýchlosť.

σ_n záchytný prierez.

τ_e emisná časová konštanta návratu populácie elektrónov do termodynamickej rovnováhy.

a_0, a_n, b_n Fourierove koeficienty.

c_n spojitý Fourierov koeficient.

$f(t)$ Fourierova transformácia.

k Boltzmannova konštanta.

m_n^* efektívna hmotnosť elektrónov.

t_0 časové oneskorenie kapacitnej transiency (po plniacom impulze).

t_P doba plniaceho impulzu.

e_n emisná rýchlosť elektrónov.

2DEG 2 Dimensional Electron Gas.

DLTFS Deep-Level Transient Fourier Spectroscopy.

DLTS Deep-Level Transient Spectroscopy.

GaN nitrid gália („Gallium Nitride“).

HEMT High Electron Mobility Transistor.

UX User Experience.

Zoznam príloh

Príloha A Používateľská príručka

Príloha B Systémová príručka

Príloha C CD médium – záverečná práca v elektronickej podobe

A Používateľská príručka

A.1 Opis aplikácie pre vizualizáciu a evaluáciu dát

Implementovaný softvér bol vytvorený pre špecifické potreby pracoviska na Ústave elektroniky a fotoniky, FEI STU v Bratislave, keďže ich softvérové vybavenie neumožňuje dosiahnuť potrebnú úroveň efektívnosti. Hlavným dôvodom je jeho komplexnosť, preto sme vyčlenili základnú funkcionálnu pre evaluáciu a zapracovali sme ju do tohoto riešenia. Poskytuje možnosť vizualizácie a evaluácie dát špecifikovaním vstupných parametrov, ktoré definujú úroveň presnosti. Podporuje aj doplnkovú možnosť exportovania výstupov vizualizácie a evaluácie vo viacerých formátoch (png, jpg, tiff, svg alebo csv).

A.2 Obsah CD média

Spustiteľný formát aplikácie ako aj ostatné prílohy sú dostupné na priloženom CD médiu. Obsahom priloženého CD média je konkrétne:

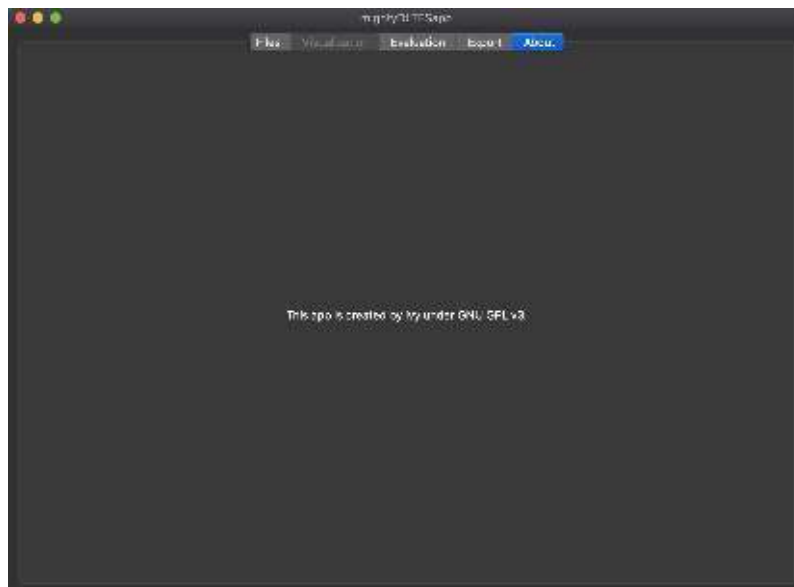
- adresár /doc/ obsahujúci diplomovú prácu vo formáte PDF, spolu so Systémovou a Používateľskou príručkou,
- adresár /tex/, v ktorom sa nachádzajú zdrojové texty práce a oboch príručiek,
- adresár /bin/ obsahuje spustiteľný formát implementovanej aplikácie,
- adresár /src/ obsahujúci komentované zdrojové kódy aplikácie v konkrétnych súboroch v hierarchickej štruktúre.

A.3 Používanie aplikácie

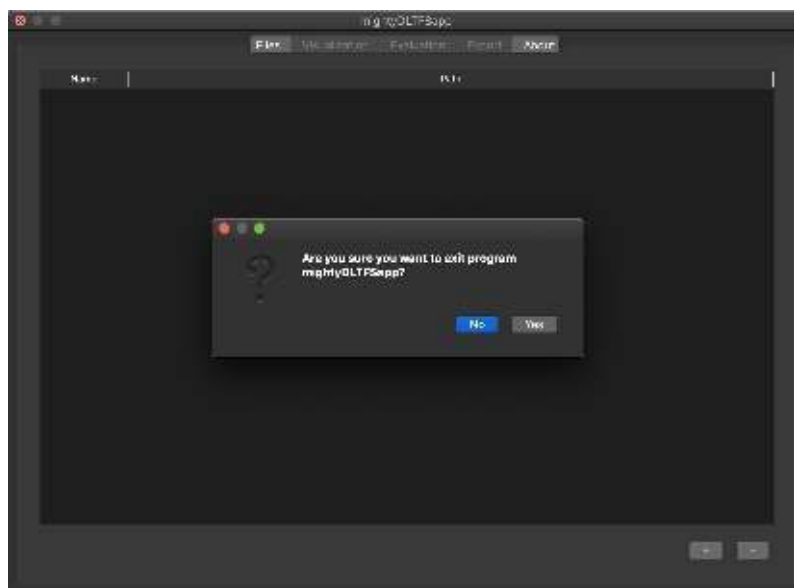
Cieľom bolo vytvoriť ľahko použiteľnú aplikáciu s intuitívnym a jednoducho objaviteľným používateľským rozhraním. Práve preto je grafické používateľské rozhranie aplikácie rozdelené do viacerých záložiek, aby boli jednotlivé funkcie separované a prehľadné. Jednotlivé záložky sú blokované do bodu kým nie je možný prechod ku danej funkcionalite. Štruktúra rozhrania je nasledovná:

- Files;
 - poskytuje možnosť pridávať a odoberať vstupné súbory,
 - zabezpečuje validáciu vstupných súborov,
- Visualization;
 - za predpokladu, že sú na vstupe dáta typu Tempscan poskytuje možnosť vizualizácie týchto dát,
 - tiež zobrazuje prehľadné zhrnutie ich parametrov v tabuľke,
- Evaluation;
 - pri zadaní vstupných dát typu Transient, poskytuje evaluáciu,
 - umožňuje dodefinovanie evaluačných parametrov,
- Export;
 - umožňuje exportovanie aktuálne realizovanej vizualizácie alebo evaluácie do viacerých formátov,
- About;
 - zahŕňa informácie o autorovi a licencií (Obr. A.1).

Pri pokuse o ukončenie aplikácie, je zobrazené oznámenie s potvrdením ukončenia (Obr. A.2).



Obr. A.1: Záložka s informáciami

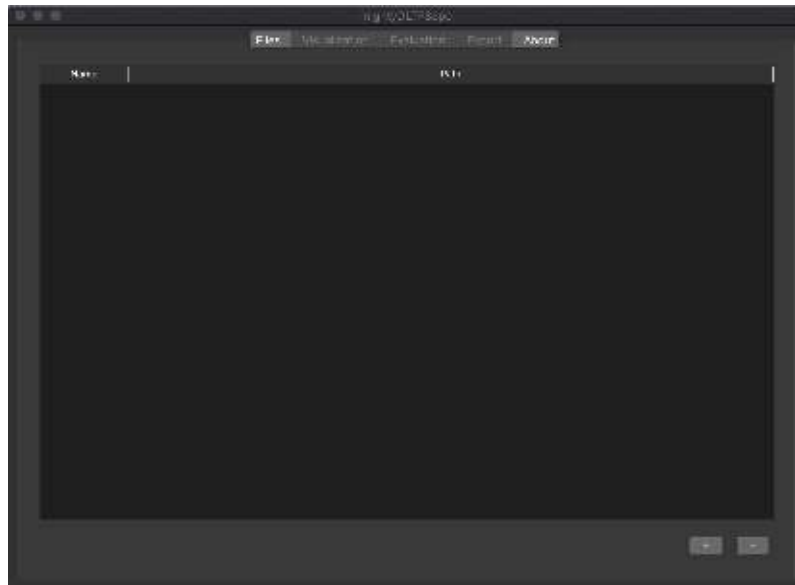


Obr. A.2: Potvrdenie ukončenia aplikácie

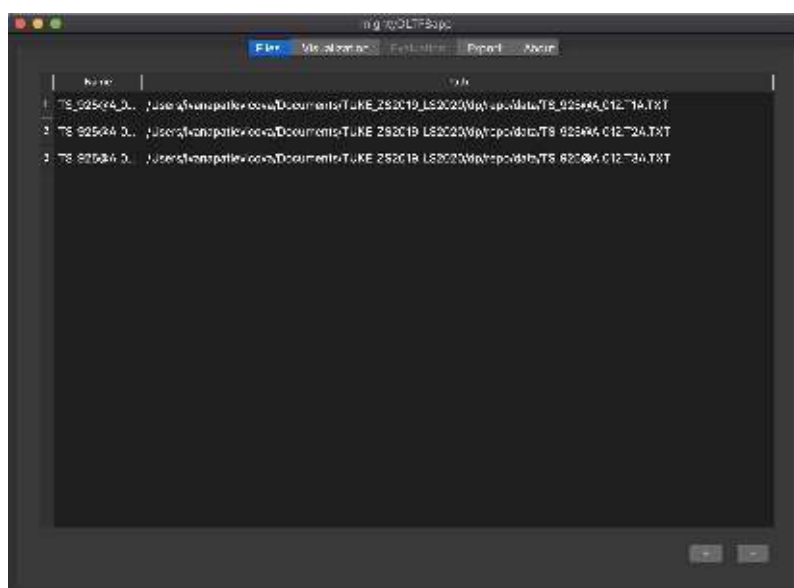
A.3.1 Definovanie vstupných meraní

Po spustení aplikácie, je zobrazená záložka Files, v tomto kroku je potrebné vybrať vstupné merania (Obr. A.3). Po kliknutí na príslušné tlačidlá (Obr. A.4 a A.5) je možné súbory do zoznamu pridávať a naopak, aj odoberať. Pri pokuse o nevalidnú akciu, je zobrazený príslušný oznam s vysvetlením situácie, ktorá nastala a nápodvedou na jej riešenie. Môžu nastať nasledovné situácie:

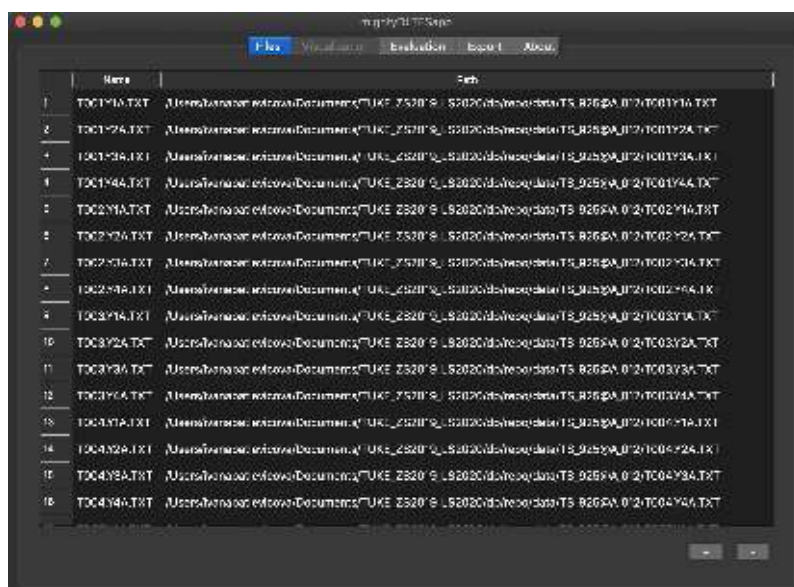
- zadanie rôznych typov súborov (Obr. A.6 a),
- zvolenie duplicitných súborov (Obr. A.6 b),
- Tempscan dáta nemožno evaluovať (Obr. A.6 c),
- nepridané žiadne vstupné dáta (Obr. A.6 d),
- pokus o odstránenie súboru bez jeho zvolenia (Obr. A.6 e),
- nevalidny formát alebo štruktúra súboru (Obr. A.6 f).



Obr. A.3: Záložka bez zadaných vstupných súborov



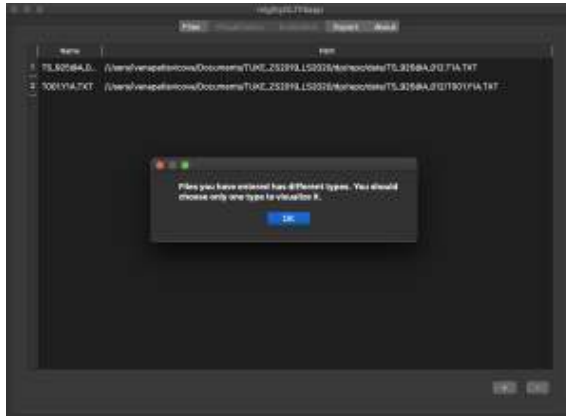
Obr. A.4: Tempscam dáta na vstupe



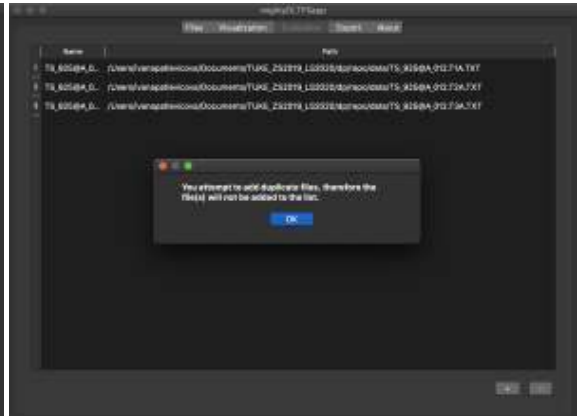
Obr. A.5: Transient dáta na vstupe

A.3.2 Vizualizácia Tempscam meraní

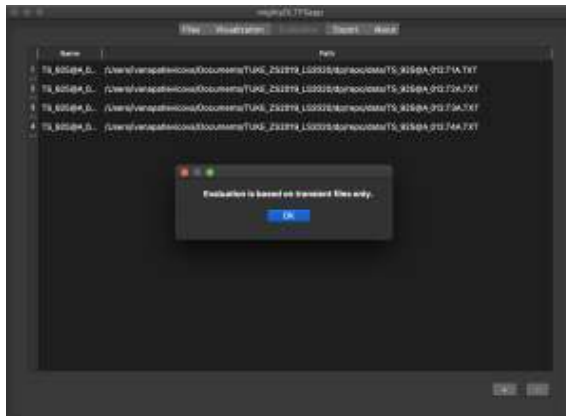
Po zadaní validných dát typu Tempscam (Obr. A.7) je možné prejsť do záložky Visualization (Obr. A.8) a zobrazíť načítané dáta v grafe. Parametre vstupných dát sú zobrazené v prehľadnej tabuľke pod grafom. Táto záložka je automaticky aktualizovaná podľa zadaného vstupu.



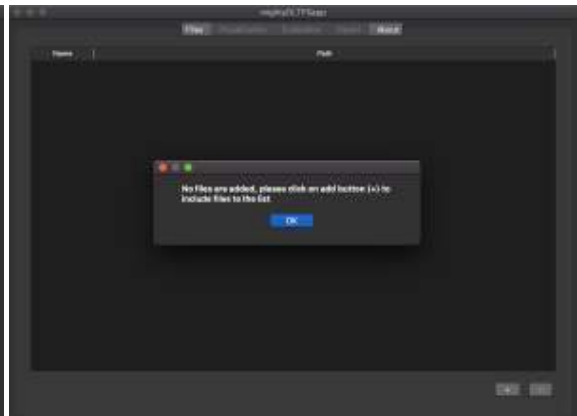
(a) Rôzne typy vstupných súborov



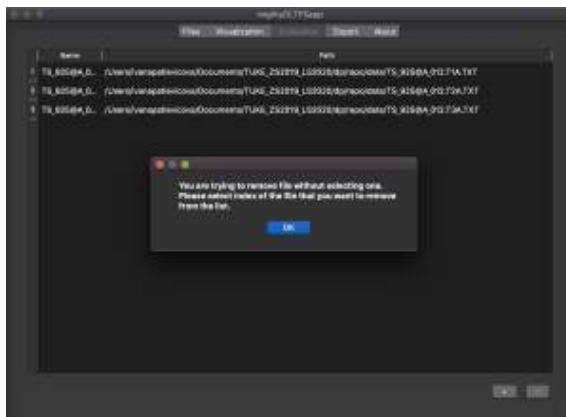
(b) Duplicita vstupných súborov



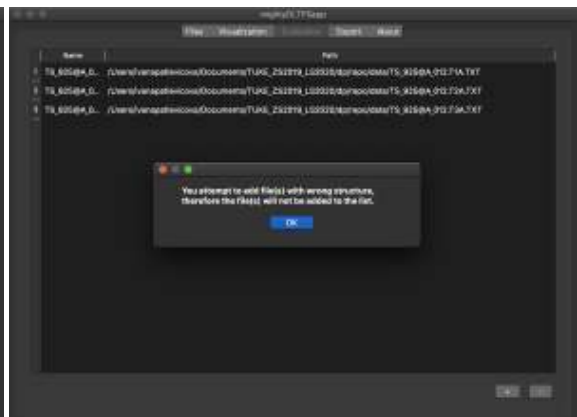
(c) Evaluácie nie je možná na Tempscan dátach



(d) Prázdny vstup

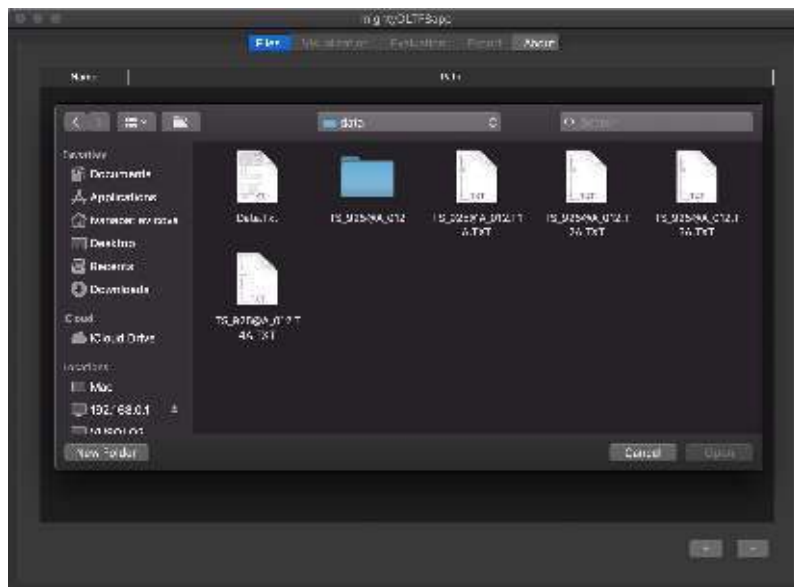


(e) Nezvolený súbor pre odstránenie

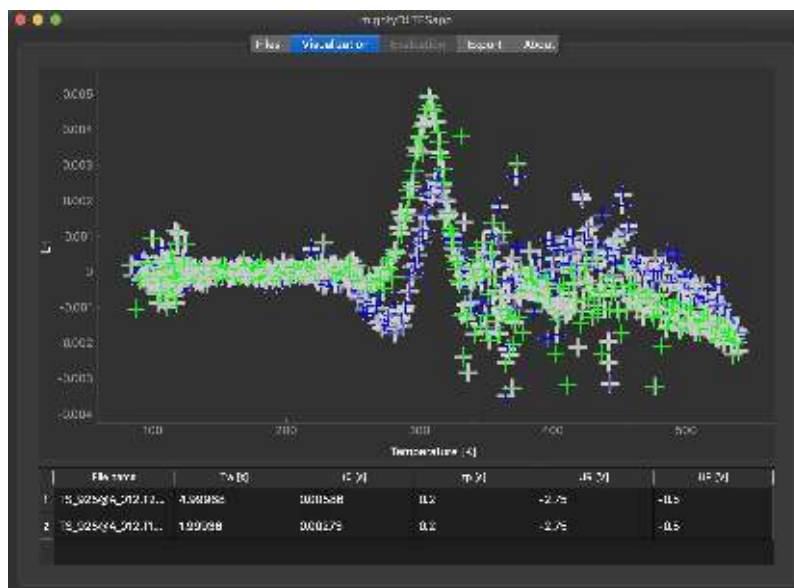


(f) Nevalidná štruktúra dát

Obr. A.6: Hlásenia



Obr. A.7: Definovanie vstupných Tempscan dát

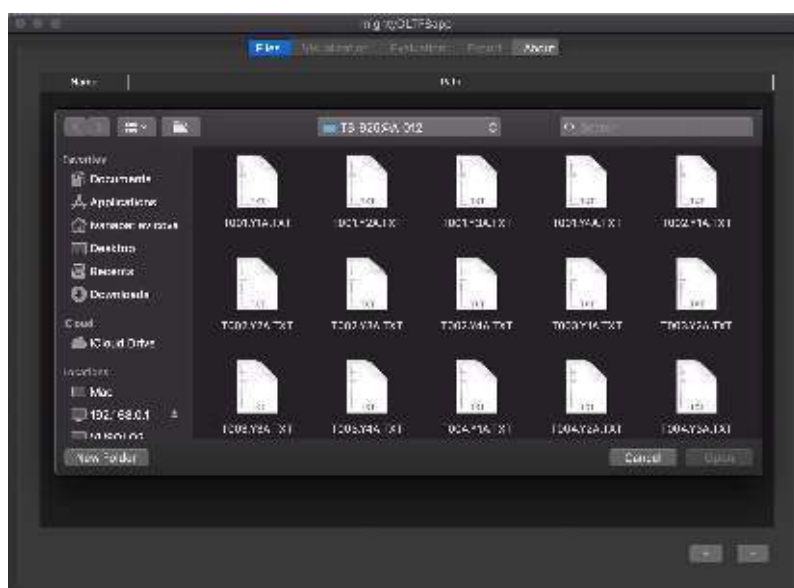


Obr. A.8: Vizualizácia Tempscan dát

A.3.3 Evaluácia Transient meraní

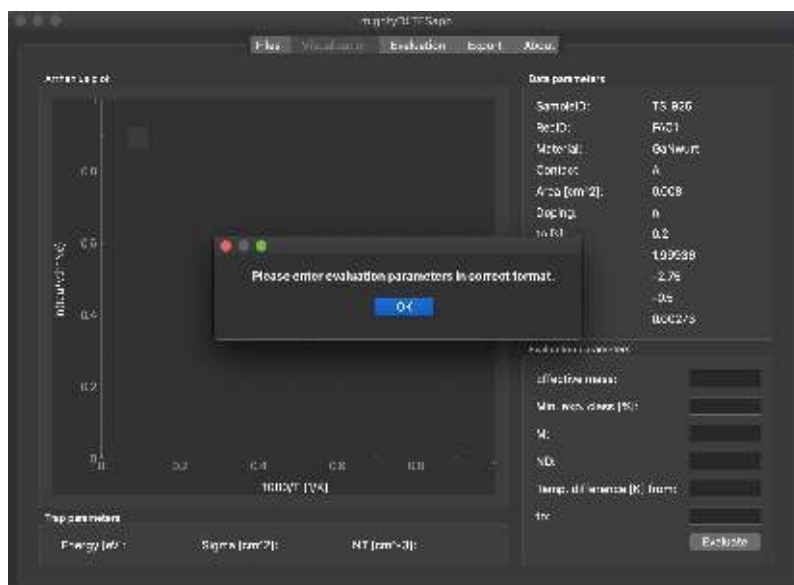
Za predpokladu, že boli zadané validné Transient dáta (Obr. A.9), je možné prejsť do záložky Evaluation, kde sú automaticky načítané informácie o meraní v časti Data parameters.

Pokiaľ nebol zadaný správny typ dát, prechod do tejto časti, nie je možný a informácia o tejto situácii je zobrazená v záložke Files.



Obr. A.9: Definovanie vstupných Transient dát

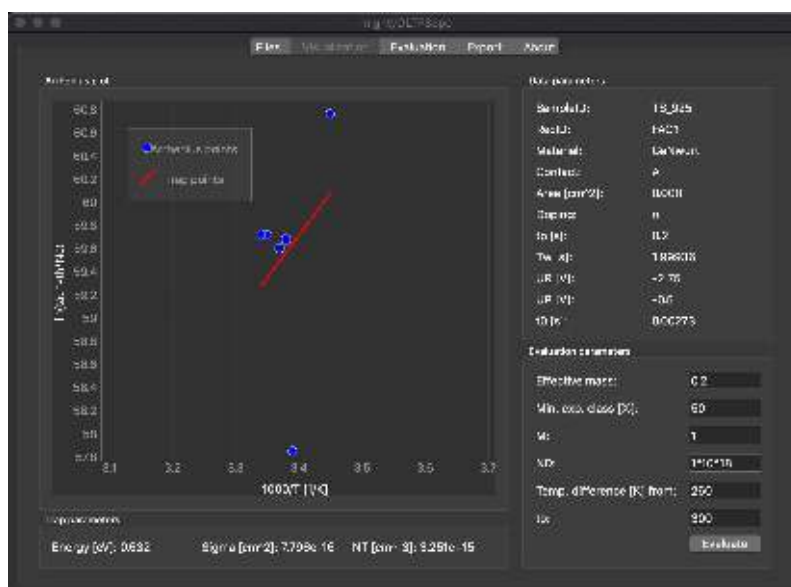
Definovaním vstupných parametrov je možné spustiť evaluáciu stlačením príslušného tlačidla Evaluate. Ak neboli definované parametre, alebo mali zlý formát je zobrazená hláška s upozornením na tento stav (Obr. A.10).



Obr. A.10: Hlásenie o nezadaných alebo zle formátovaných vstupných parametroch

Pokiaľ boli definované správne je aktualizovaný graf zobrazujúci Arrheniovu závislosť a nájdenú pascu, spolu jej s parametrami v časti Trap parameters (Obr.

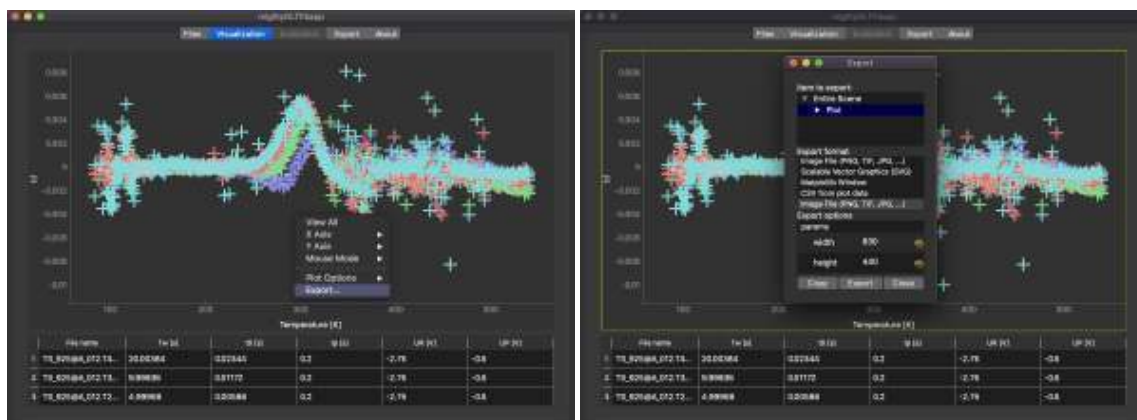
A.11). Pri zmene vstupných súborov je možné proces opakovať.



Obr. A.11: Výsledná evaluácia dát

A.3.4 Exportovanie vizualizovaných a evaluovaných dát

Vizualizované ale aj evaluované dáta je možné exportovať dvoma spôsobmi. Prvým je kliknutie na graf (Obr. A.12 a) a dodefinovanie potrebných informácií, ako formát súboru a lokácia pre uloženie (Obr. A.12 b).



(a) Exportovanie vizualizácie priamo z grafu

(b) Špecifikovanie exportovaného súboru

Obr. A.12: Exportovanie priamo z grafického komponentu

Druhou možnosťou je prechod do záložky Export (Obr. B.1) a dodefinovanie

názvu a formátu exportovaného súboru, pričom po kliknutí na príslušné tlačidlo bude súbor uložený.



Obr. A.13: Exportovanie z príslušnej záložky

A.4 Spustenie aplikácie

Aplikácia bola vytvorená využitím jazyka Python (verzia 3.7), ktorý je platformovo nezávislý a ďalších knižníc ako napr. PyQt5 a PyQtGraph. Jej výsledná podoba bola exportovaná do samostatne spustiteľného formátu pre danú platformu. Pomocou programu py2app sme vytvorili spustiteľný formát aplikácie (.app). Pre ďalšie platformy dodatočne vygenerujeme potrebné formáty (napr. .exe) a dodáme prostredníctvom repozitára, kde je aplikácia uchovaná. Bez potreby inštalácie je aplikáciu možné použiť na rôznych platformách, spustením súboru, keďže využité technológie podporujú multiplatformovosť. Aplikáciu sme testovali na zariadení s nasledujúcimi parametrami:

- OS: macOS Catalina Version 10.15.4,
- RAM: 8 GB 2133 MHz LPDDR3,
- DISK: 500 GB SSD,
- PROCESOR: 3,1 GHz Dual-Core Intel Core i5.

B Systémová príručka

B.1 Funkcie aplikácie a analyzovanie riešenia

Aplikácia pre vizualizáciu a evaluáciu experimentálnych meraní na polovodičových štruktúrach (na báze GaN, ale aj iných materiálov) je určená pre pracoviská disponujúce systémom DL8000, konkrétne bola vyvíjaná pre využitie pracoviskom na Ústave elektroniky a fotoniky, FEI STU v Bratislave.

Poskytuje možnosť spracovania dát generovaných systémom a preložených do textovej podoby doplnkovým softvérom poskytovaným spolu s programovým vybavením pracoviska. Spracované dáta je možné vizualizovať (Tempscan merania) alebo realizovať priamu evaluáciu (Transient dáta) definovaním vstupných parametrov určujúcich citlivosť výstupu. Ďalšou funkcionalitou je možnosť exportovania grafov vizualizácie a evaluácie do formátov png, jpg, tiff, svg alebo csv.

B.2 Priblíženie aplikácie

V tejto časti priblížime softvérové riešenie, jeho štruktúru a samotné metódy jednotlivých tried.

B.2.1 Popis softvérového riešenia

Aplikáciu sme implementovali v jazyku Python s využitím knižníc PyQt5, PyQtGraph, NumPy a scikit-learn. Softvérové riešenie je rozdelené do balíčkov pre zvýšenie prehľadnosti a logické odčlenenie funkcionalít.

Po zadaní vstupných súborov je možné programom realizovať vizualizáciu alebo evaluáciu spracovaných dát. Pri korektných Tempscan dátach na vstupe je povolená vizualizácia a následne export grafu. Po zadaní valídnych Transient dá-

tach a definovaní parametrov evaluácie, je možné spustiť priame vyhodnotenie a zobraziť parametre pasce. Takúto evaluáciu je možné tiež exportovať v spomenu- tých formátoch.

B.2.2 Štruktúra projektu a popis tried

Projekt zahŕňa štyri balíky a súbor `main.py`, ktorý obsahuje funkciu spúšťajúcu aplikáciu. Jednotlivé balíky a ich obsah popíšeme v nasledujúcich častiach. Sú- stredíme sa na vybrané hlavné funkcie daných tried, pomocné funkcie sú kome- tované v zdrojových textoch priložených na CD médiu.

Balík `analysis`

Balík `analysis` obsahuje súbory s triedami reprezentujúcimi potrebnú funkciona- litu pre realizovanie priamej evaluácie nameraných dát. Ďalej popíšeme jednotlivé súbory obsahujúce triedy a metódy poskytujúce danú funkcionálnosť.

`arrheniusPoints.py` – Class `ArrheniusPoints`

Poskytuje výpočet bodov Arrheniovej závislosti.

- `def get_arrhenius_data(fourier_data_list, mass_factor, M)`
 - Pomocou vstupných parametrov vypočíta a vráti bod Arrheniovej zá- vislosti ako komplexné číslo.

`directEvaluation.py` – Class `DirectEvaluation`

Definuje výpočty jednotlivých parametrov pasce.

- `def get_parameters(regression, ampl, CO, ND)`
 - Vráti vypočítané parametre pasce.
 - Pre výpočet sú použité vstupné parametre.

`discreteFourierTransform.py` – Class `DiscreteFourierTransform`

Obsahuje logiku potrebnú pre získanie veličín dôležitých pri analýze dát.

- `def get_fourier_data(transient_data)`
 - Vyráta a vráti slovník s vypočítanými parametrami.

Balík data

Balík data obsahuje implementáciu reprezentácie vstupných súborov v programe. Jednotlivé triedy ďalej popíšeme.

dataFile.py – Class DataFile

Predstavuje všeobecný typ vstupných dát načítaných do programu. Definuje viaceré metódy pre získanie požadovaného parametra, a tiež pomocné metódy pre získanie hodnoty na indexe alebo vypísanie informácií.

- `def get_*(self)`
 - Metódy vrátia požadovaný parameter *.

mainDataHeader.py – Class MainDataHeader

Reprezentuje hlavnú dátovú hlavičku vstupného súboru.

- `def __init__(self, list_of_header_data)`
 - Inicializuje triedu hodnotou parametra.

ptvBlock.py – Class PtvBlock

Obsahuje tzv. PTV blok dát merania.

- `def __init__(self, list_of_ptv_data)`
 - Inicializuje triedu nastavením parametra.

tempscanFile.py – Class TempscanFile

Reprezentuje spracovanú formu merania typu Tempscan.

- `def __init__(self, header_data, list_of_ptv_blocks, file_name)`
 - Inicializuje triedu vstupnými parametrami.
- `def print_list(self)`
 - Vypíše list blokov.
- `def get_x_points(self)`

- Vrátí list všetkých X-ových bodov blokov.

- `def get_y_points(self)`

- Vrátí list všetkých Y-ových bodov blokov.

transientFile.py – Class `TransientFile`

Je spracovanou formou merania typu `Transient`.

- `def __init__(self, header_data, list_of_data_r, list_of_data_i, file_name)`

- Inicializuje triedu vstupnými parametrami.

- `def get_*(self)`

- Metódy vrátia požadovaný parameter *.

Balík helper

Balík helper zahŕňa všetky pomocné triedy pre fungovanie aplikácie. Jednotlivé funkcie tried popíšeme.

constants.py – Class `Constants`

Obsahuje hodnoty používaných konštánt.

- `k = 1.38066E-23`

- Boltzmann constant.

- `h = 6.62618E-34`

- Planck constant.

- `me = 9.10953E-31`

- Mass of electron.

- `q = 1.60200E-19`

- Elementary charge.

exporter.py – Class `Exporter`

Trieda disponujúca funkcionalitou potrebnou pre export dát.

- `def export(self, fileName=None, toBytes=False, copy=False)`
 - Exportuje dáta vo vybranom formáte.

parser.py – Class Parser

Poskytuje funkcie pre spracovanie a validáciu vstupu. Obsahuje viacero pomocných funkcií, no sústredíme sa na hlavnú z nich.

- `def parse_file(self, file_path)`
 - Spracuje súbor na ceste definovanej parametrom a vráti ho ako objekt naplnený príslušnými hodnotami.

Balík ui

Balík ui poskytuje jednotlivé časti používateľského rozhrania a funkcie potrebné pre jeho kontrolu a používanie. Bližšie priblížime komponenty a ich funkcie.

aboutTab.py – Class AboutTab

Obsahuje komponenty používateľského rozhrania záložky About.

evaluationTab.py – Class EvaluationTab

Obsahuje komponenty používateľského rozhrania záložky Evaluation a funkcie pre jej ovládanie resp. fungovanie. Definuje viaceré pomocné inicializačné a aktualizované funkcie pre jednotlivé grafické komponenty, taktiež pomocné funkcie pre spracovanie vstupu evaluácie. Sú tu definované aj metódy pre aktualizovanie globálnych dát vlastnených objektom, ktorý vytvoril Class EvaluationTab a drží na neho referenciu.

- `def try_evaluation(self, input_params)`
 - Obsahuje logiku priamej evaluácie a realizuje ju využitím vstupných parametrov.
- `def init_*(self)`
 - Inicializuje jednotlivé časti záložky.
- `def evaluate(self)`

- Riadi beh evaluácie, vytvára upozornenia pre užívateľa.
- `def update(self)`
 - Zabezpečuje aktualizovanie dát po opakovanej evaluácii.
- `def global_data(self)`
 - Predstavuje property a `global_data.setter` globálneho zoznamu súborov, ktorý je automaticky aktualizovaný.

exportTab.py – Class `ExportTab`

Obsahuje komponenty používateľského rozhrania záložky `Export` a funkcie pre jej ovládanie resp. fungovanie. Zabezpečuje svojimi funkciami aktualizovanie stavu tlačidiel. Nastavuje tlačidlám kontrolné funkcie.

- `def export(self, plt)`
 - Pokúša sa vytvoriť a exportovať súbor zvoleného typu za pomoci importovaného pomocného export-éra.

fileTab.py – Class `FileTab`

Obsahuje komponenty používateľského rozhrania záložky `Files` a funkcie pre jej ovládanie resp. fungovanie. Zabezpečuje aktualizáciu zoznamu vstupných súborov, ako aj jej globálnej reprezentácie. Funkcie poskytujú hlásenia užívateľovi o nevalídnych akciách. Nastavuje funkcionality tlačidlám.

- `def add_button_handler(self)`
 - Pridanie funkcionality tlačidlu 'pridať'.
- `def remove_button_handler(self)`
 - Pridanie funkcionality tlačidlu 'odobrať'.

mainWindow.py – Class `MainWindow`

Obsahuje hlavné okno aplikácie a nastavenia jeho vlastností.

- `def closeEvent(self, event)`
 - Kontroluje udalosť - pokus o zatvorenie aplikácie.

tabsWidget.py – Class TabsWidget

Kontroluje dostupnosť jednotlivých záložiek a posúvanie dát medzi nimi. Aktualizuje dostupnosť záložiek podľa logiky aplikácie.

- def process_update_of_*(self, empty)
 - Aktualizuje záložku * podľa hodnoty parametra.

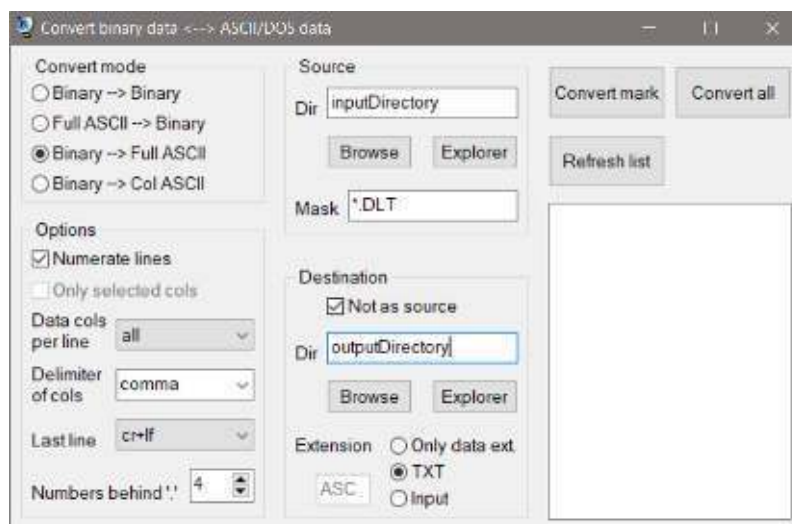
visualizationTab.py – Class VisualizationTab

Obsahuje komponenty používateľského rozhrania záložky Visualization a funkcie pre jej ovládanie resp. fungovanie. Zabezpečuje aktualizovanie vizualizácie pri zmene vstupu, ako aj aktualizáciu zobrazených parametrov.

- def update(self)
 - Aktualizuje jednotlivé časti rozhrania.

B.2.3 Popis vstupných dát

Generované binárne (.DLT) súbory je potrebné preložiť do textovej podoby (.TXT) pomocou softvéru, ktorý je súčasťou programového vybavenia systému DL8000, podľa nastavení zobrazených na Obr. B.1.



Obr. B.1: Softvér na preklad dát

B.2.4 Preklad a spustenie aplikácie

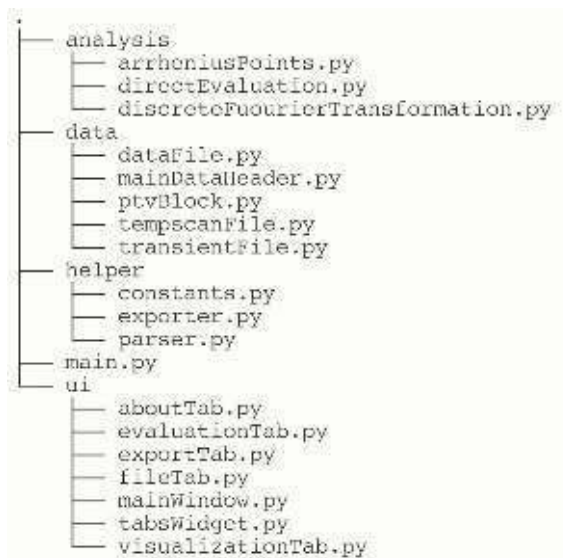
Preklad a spúšťanie programu môžeme realizovať napr. v prostredí Pycharm, v ktorom bol projekt aj vyvíjaný. Prostredie disponuje potrebnými prostriedkami pre interpretovanie a spustenie programu (za predpokladu nainštalovaného interpretéra a potrebných modulov).

Softvérové požiadavky

Aplikácia bola vyvíjaná v jazyku Python verzia 3.7. Odporúčame využívať túto verziu a novšie. Ďalej je potrebné disponovať knižnicami PyQt5, PyQtGraph, NumPy a scikit-learn, ktoré je možné dosiahnuť ako doplnkové balíčky v prostredí Pycharm.

Zoznam zdrojových textov v hierarchickej štruktúre

Všetky komentované zdrojové kódy je možné nájsť v prílohe na CD médiu (pričinnok /src/), tiež sú zálohované na vzdialenom repozitári GitHub. Projekt má štruktúru zobrazenú na Obr. B.2.



Obr. B.2: Štruktúra projektu a obsah jednotlivých balíčkov

B.3 Vyhodnotenie riešenia

Aplikácia je určená pre zjednodušenie procesu evaluácie meraní realizovaných na polovodičových štruktúrach. Poskytuje potrebnú funkcionálnu pre základnú evaluáciu. Dodefinovaním ďalších vstupných parametrov by sa dalo dosiahnuť vyšší level presnosti a citlivosti. Testovanie používateľmi poukázalo na dobrú objaviteľnosť grafického rozhrania a jeho funkcií.