

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A
INFORMATIKY

SPEKTROSKOPICKÉ DÁTA

2017/2018

Michal Chamula, Tomáš Bordáč,
Martina Bodišová, Patrik Fašang

Obsah

Obsah	2
Úvod	3
1.1 Podstata dokumentu.....	3
1.2 Určenie dokumentu	3
1.3 Rozsah Systému.....	3
1.4 Slovník cudzích pojmov	3
1.5 Referencie.....	4
2. Všeobecný popis	5
2.1 Perspektíva projektu	5
2.2 Funkcie produktu.....	5
2.3 Charakteristika používateľov.....	5
3. Špecifikácia požiadaviek.....	6
3.1 Načítanie vstupného súboru	6
4 Komunikácia s DLL	7
5 Návrh.....	8
1. Špecifikácia vonkajších interfejsov.....	8
1. Vstupný interface.....	8
1. Výstupný interface.....	9
2. Implementácia	10
6 Diagramy.....	11
1. Sekvenčný diagram	11
2. Triedne diagramy	11
7 Testovací scenár	13
8 Používateľská príručka	14
1. Ovládanie aplikácie	14
2. Načítanie DLL knižnice	14
3. Zadanie vstupných parametrov	14
4. Analýza výstupu	14
Záver	16

Úvod

1.1 Podstata dokumentu

Tento dokument popisuje požiadavky zadávateľa na softvér vyvíjaný v projekte Spracovanie spektroskopických dát.

1.2 Určenie dokumentu

Tento dokument je určený stakeholderom, ktorí na základe popísaných požiadaviek budú softvér vyvíjať. Finálna verzia tohoto dokumentu je odsúhlasená zadávateľom.

1.3 Rozsah Systému

Projekt je dynamicky linkovaná knižnica (DLL), ktorá slúži na rátanie komplexných, neanalytických funkcií. Neobsahuje grafické používateľské rozhranie.

1.4 Slovník cudzích pojmov

- **DLL** -(angl. Dynamic Link Library) je množina malých programov, ktorá môže byť použitá viac ako jedným programom v tom istom čase. Táto množina je zväčša uložená v súboroch so suffixom “.dll“
- **Stakeholder**- osoba alebo skupina osôb, ktorá sa podieľa na rovnakom projekte, napr. podnikaní, programovaní, vede a pod.
- **LabVIEW** – (angl. Laboratory Virtual Instrument Engineering Workbench) je vývojové prostredie určené na vizuálne programovanie s podporou čítať DLL knižnice napísané v jazyku c++.
- **Konvolúcia** – matematický operátor spracovávajúci dve funkcie. Je definovaný vzťahom:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha$$

1.5 Referencie

- [1] *"Decay time integrals in neutral meson mixing and their efficient evaluation"* - Till Moritz Karbach, Gerhard Raven, Manuel Schiller (CERN - Switzerland, NIKHEF - The Netherlands)
- [2] *"An isolated line-shape model to go beyond the Voigt profile in spectroscopic databases and radiative transfer codes"* - N.H. Ngo, D. Lisak, H. Tran, J.-M. Hartmann
- [3] *"Efficient computation of some speed-dependent isolated line profiles"* - H. Tran, N.H. Ngo, J.-M. Hartmann

2. Všeobecný popis

2.1 Perspektíva projektu

Projekt bude súčasťou väčšieho celku, ktorý má za úlohu analyzovať spektroskopické dáta. Optická Spektroskopia je oblasť fyziky, zaoberajúca sa štúdiom elektromagnetického žiarenia emitovaného alebo pohlteneho vzorkou. Získané informácie sa dajú použiť buď kvalitatívne (charakteristika vnútornej štruktúry vzorky, poprípade prostredia kde sa nachádza), alebo kvantitatívne (určenie koncentrácie známej vzorky).

2.2 Funkcie produktu

Softvér by mal byť schopný v optimálnom čase aplikovať rôzne transformácie na vstupné hodnoty – spektrá a modelovať ich tvar použitím funkcií opisujúcich žiarenie (absorpciu) vzoriek. Medzi tieto funkcie patria: Lorentzova, Gaussova, Voigtova alebo Hartmann–Tran.

2.3 Charakteristika používateľov

Finálny produkt bude využívať oddelenie experimentálnej Fyziky FMFI UK. Z používateľského hľadiska bude produkt použiteľný iba pod vývojovým prostredím LabVIEW.

3. Špecifikácia požiadaviek

3.1 Načítanie vstupného súboru

- 3.1.1 Systém prečíta vstupný súbor a dáta poukladá do objektov v DLL a pripraví tak namerané dáta na ďalšie spracovanie. Načítavanie dát je už implementované v knižnici DLL, ktorú sme zdedili. LabView načítané dáta zobrazí vo formulároch a nijak ich nemení. Umožňuje tieto dáta len prezerat', prípadne zvoliť transformačnú funkciu, ktorou sa tieto hodnoty prepočítajú a zobrazí počiatočný stav hodnôt, ktoré sme načítali so súboru v jednom formulári a zmenené dáta zvolenou funkciou v druhom formulári.

3.2 Zobrazenie načítaných vstupov

- 3.2.1 Prostredie LabView ponúka dva formuláre súvisiace s načítanými dátami.
- 3.2.2 Názov prvého formulára je **Data IN** (takto je nazvaný v LabView), ktorý ponúka používateľovi zadať:
- 3.2.2.1 **Name** - názov datasetu (merania) napr. pondelok, utorok , ...
- 3.2.2.2 Hodnotu **X-in**, kde si zvolíme z ktorého indexu chceme zobrazit' X-ovú súradnicu
- 3.2.2.3 Hodnotu **Y-in**, kde si zvolíme z ktorého indexu chceme zobrazit' Y-ovú súradnicu
- 3.2.2.4 Hodnotu **W-in**, kde si zvolíme z ktorého indexu chceme zobrazit' W (chybu merania)
- 3.2.3 Názov druhého formulára je **Data PAR In** (názov v LabView), ktorý ponúka používateľovi zadať:
- 3.2.3.1 **Name** - názov datasetu (merania) napr. pondelok, utorok , ... (mal by byť rovnaký ako v bode 2.1.1.1)
- 3.2.3.2 **Func.names** - umožní používateľovi zvoliť funkciu, ktorou sa majú vstupné hodnoty upraviť. Tieto hodnoty sú z popísané v bodoch 2.1.1.2, 2.1.1.3 a 2.1.1.4

3.3 Výstupy do LabView

- 3.3.1 Používateľ má k dispozícii v LabView ďalší formulár, v ktorom sa zobrazia transformované dáta. Keďže dáta sú uložené v poliach, tak sa zobrazujú iba z jedného indexu, ktorý môže používateľ meniť a tak si prezrieť postupne všetky hodnoty. Po načítaní súboru budú rovnaké ako vo formulári popísanom v časti 2.1.1. Až keď používateľ zvolí funkciu a stlačí tlačidlo na transformovanie dát, tieto dáta vo formulári prepočíta podľa zvolenej funkcie a vypíše ich znovu do tohto formulára. Nasledujúci popis počíta s tým, že už bola zvolená funkcia a stlačené tlačidlo na transformáciu dát.
- 3.3.2 Názov formulára v LabView je **Data_OUT_Fast**, ktorý obsahuje kolónky:
- 3.3.2.1 **Name** - názov datasetu (rovnaký ako v bode 2.1.1.1 a nebude sa meniť)
- 3.3.2.2 Hodnotu **X_Out**, kde si zvolíme z ktorého indexu chceme zobrazit' X-ovú súradnicu. Táto súradnica je už zmenená zvolenou funkciou.
- 3.3.2.3 Hodnotu **Y_Out**, kde si zvolíme z ktorého indexu chceme zobrazit' Y-ovú súradnicu. Táto súradnica je už zmenená zvolenou funkciou.
- 3.3.2.4 Hodnotu **W_Out**, kde si zvolíme z ktorého indexu chceme zobrazit' W (chybu merania). Táto chyba merania je stále rovnaká (ako v bode 2.1.1.4). Teda ju DLL nebude nijako meniť.

4 Komunikácia s DLL

DLL bude priamo komunikovať s LabView len cez funkciu `fast()`, ktorá načíta vstupný súbor. Dáta poukladá do štruktúr a nezmenené ich zobrazí v LabView. Používateľ bude môcť zvoliť funkciu, ktorou bude chcieť dáta transformovať/zmeniť. Ďalej sa budú dáta spracovávať podľa zvolenej funkcie.

5 Návrh

1. Špecifikácia vonkajších interfejsov

DLL knižnica vyvíjaná v tomto projekte bude komunikovať s prostredím LabView. Používateľ načíta dáta z grafického rozhrania v programe LabView a nastaví vstupné parametre v tabuľke, ktoré bude viesť DLL prečítať, spracovať a výsledok ponúknuť používateľovi.

1. Vstupný interface

1. Popis vstupného formuláru, Spektrum (obr.1):

1. Polia:

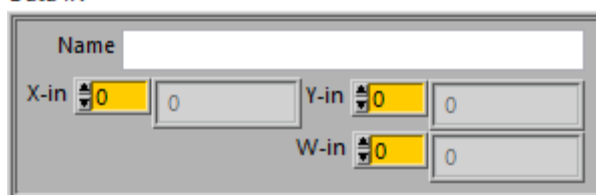
Name: meno dát

X-in: x-ová súradnica

Y-in: y-ová súradnica

W-in: neistota

Data IN



Obrázok 1: Vstupný formulár

2. Popis vstupného formuláru, Parametre (obr. 2)

1. Polia:

Param.strings: charakterizujúce mená parametrov a ich vlastnosti (meno, model, nezávislé parametre)

Param.values: charakterizujúce hodnoty parametrov (hodnota, neistota, škála pre GUI)

Func.names: (L je počet transformácií a funkcií modelu – max 6: XT, YT, BL, PK, RF, MC, ktoré treba vypočítať), prvý stĺpec obsahuje meno funkcie (XT, YT, BL, PK, RF, MC), druhý mená „skupín=groups“ v rámci danej funkcie zoradených do jedného stringu: Menno1@Meno2@...@MenoN

Func.par.adresses: 3xLxQ (Q je max počet skupín nachádzajúci sa niektorej z funkcií), prvý inde(page) definuje funkciu podľa poradia v poli Func.names následne každá skupina má jeden riadok, kde prvá hodnota hovorí koľko hodnôt je v danom riadku (adries – poradie v Param.strings a Param.values), poradie adries parametrov je pevne definované pre každú funkciu-skupinu

Data.names: Mená vektorov pred simulovaných dát pre funkciu REF

Data.length: Zodpovedajúca dĺžka(počet bodov) pre vektory pred simulovaných dát (uložené v binárnom súbore na disku)

The image shows a software dialog box titled "Data PAR in". It contains several input fields and spinners. At the top is a "Name" field. Below it is a "Param.strings" section with a spinner set to 0 and an empty text box. Next is a "Param.numbers" section with a spinner set to 0 and a text box containing the number "0". To the right of "Param.numbers" is a "Func.names" section with a spinner set to 0 and an empty text box. Below "Param.numbers" is a "Data.names" section with a spinner set to 0 and an empty text box. Below "Data.names" is a "Data_length" section with a spinner set to 0 and a text box containing the number "0". To the right of "Data.names" and "Data_length" is a "Func.Par.adresses" section with a spinner set to 0 and a text box containing the number "0".

Obrázok 2: Štruktúra vstupných dát

1. Výstupný interface

1. Popis výstupného formuláru, Spektrum (obr. 3):

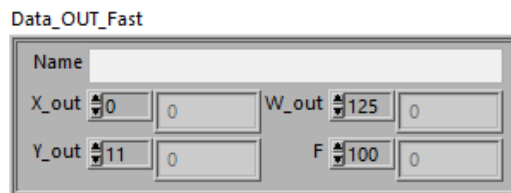
1. Polia:

Name: meno dát

X-in: x-ová súradnica

Y-in: y-ová súradnica

W-in: neistota



Obrázok 3: Štruktúra výstupných dát

2. Implementácia

Z LabView je z DLL volaná funkcia `fdata_fast()`, ktorú začneme upravovať ako prvú. Do jej tela zimplementujeme volanie funkcií na výpočet dát, ktoré sú popísané v katalógu požiadaviek.

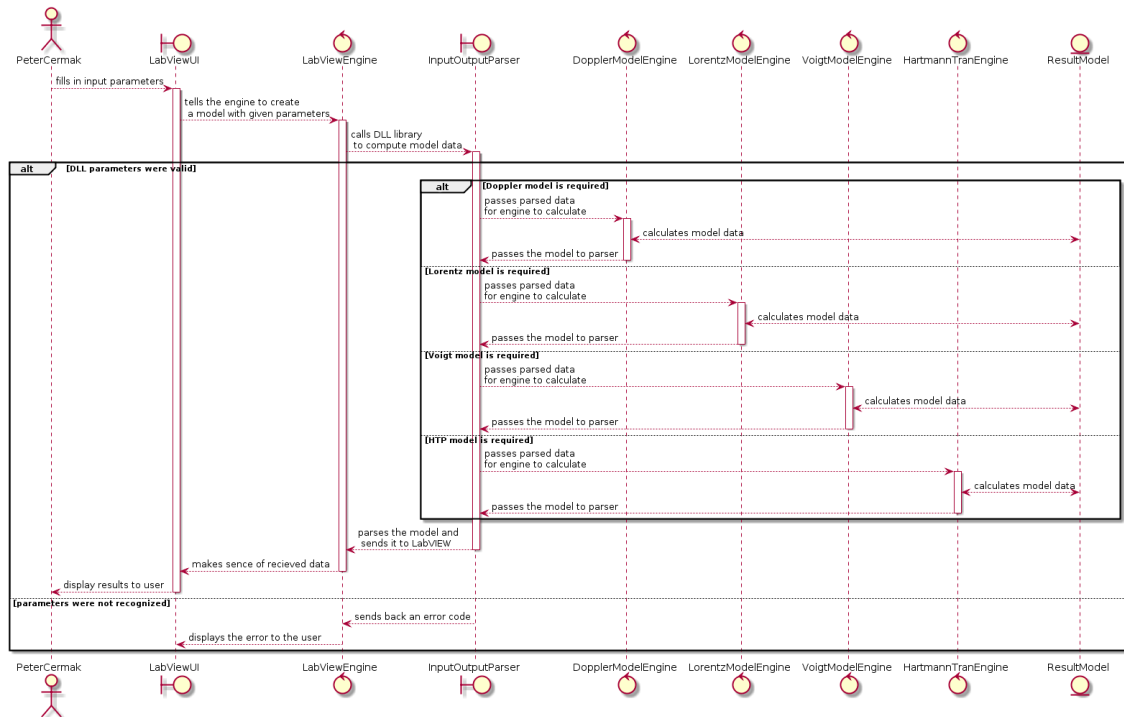
Volaná metóda má vstupné parametre pointre na 4 štruktúry (`TD1 *DataPARin`, `TD7 *DataIN`, `TDFast *DataOUT_F`, `TD10 *Error`) a výstupom je typ `<int32_t>` s ktorým ale nič nerobíme ani neupravujeme, takže ho ani neriešime. Všetky hodnoty s ktorými pracujeme sú iba v štruktúrach. Štruktúry sú pevne definované a nemožno ich meniť. Zavolaním tejto funkcie sa dáta vypočítajú, upraví a pošlú do štruktúry `TDFast *DataOUT_F` v ktorej zostanú uložené. Po zavolaní metódy sa vypočítané dáta vykreslia v tabuľke `Data_OUT_Fast`.

Funkcie a metódy pre výpočet dát budú dopísané do triedy `Transform`. Funkcia `fdata_fast()` už od staršej verzii programu vytvára inštanciu tohto objektu kde sú statické funkcie a metódy takže do jej tela treba implementovať volanie našich metód pre výpočet dát.

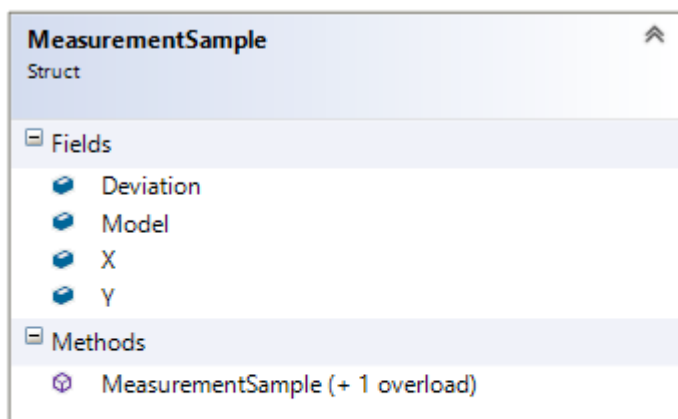
6 Diagramy

1. Sekvenčný diagram

Tento sekvenčný diagram detailnejšie popisuje fungovanie vyššie uvedeného použitia. Zameriava sa na interakciu medzi LabVIEW programom a DLL knižnicou, ktorá obsahuje moduly na spracovanie vstupu, transformáciu funkciami a vyprodukovanie modelov.



2. Triedne diagramy



uint : unsigned int
Typedef

7 Testovací scénár

- Vstup:
 - 1. Vyplnenie vstupných parametrov v grafickom prostredí LabView
 - 2. Zvolí sa funkcia, ktorou majú byť vstupné dáta spracované
 - 3. Spustíme proces vypočítania dát z prostredia LabView stlačením tlačidla
- Výstup:
 - 1. Kontrola výstupných parametrov naplnených v tabuľke data_out_fast
 - 2. Kontrola požadovaného grafického modelu
 - 3. Kontrola grafického modelu po zvolení údajov do transformačných funkcií

8 Používateľská príručka

1. Ovládanie aplikácie

Naša aplikácia je DLL knižnica, ktorá poskytuje výpočtovú funkcionálnosť. Je úzko spätá so softvérom LabVIEW, ktorý poskytuje aj užívateľské rozhranie na používanie tejto knižnice. Samozrejme, ako každú inú knižnicu, aj túto knižnicu je možné použiť mimo LabVIEW. Takéto využitie nie je očakávané a vyžadovalo by vytvorenie špecifických vstupných parametrov, ktoré LabVIEW generuje z užívateľského vstupu. Preto je z takéhoto hľadiska pre používateľa nepraktické.

2. Načítanie DLL knižnice

Závisí od konkrétnej konfigurácie LabVIEW aplikácie. V našom konkrétnom prípade sa musí DLL knižnica volať **Win32Project_AIstart** a musí byť umiestnená v adresári **data**.

3. Zadanie vstupných parametrov

Formát vstupu taktiež závisí od konkrétnej aplikácie. Na ukážke nášho prípadu sú k dispozícii dva spôsoby a to vyplnenie formulára, alebo import zo súboru.

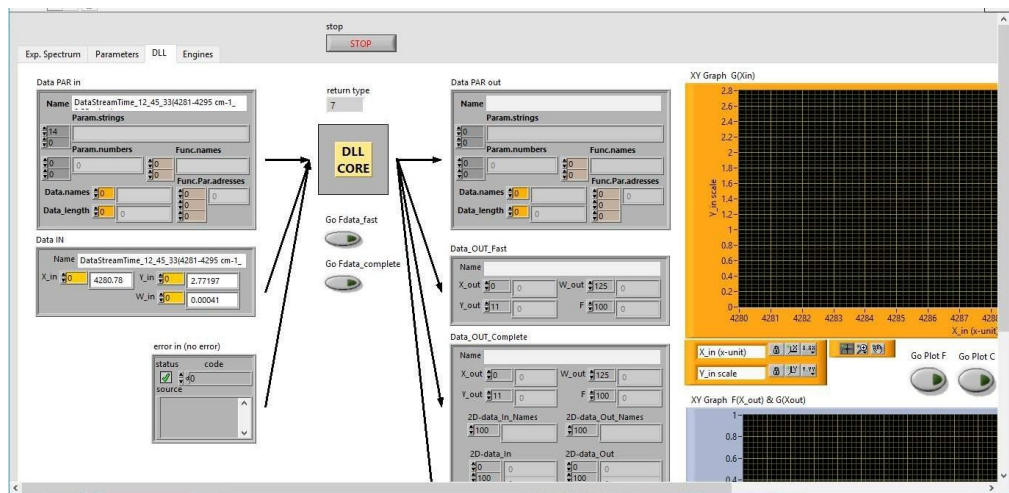
The screenshot displays the application's parameter input interface. At the top, there is a 'Run Track' button and a 'Key File path (P1)' field containing 'C:\Program Files (x86)\TI project\data\ParametersToLoad.txt'. To the right is a 'Load From File' button. Below these are two tables:

Param.strings	
XT@XOff@p0	C
YT@YTyp@function	C_VT
YT@YSpl@p0-0	C
YT@YSpl@p0-1	C
YT@YSpl@p1-0	C
YT@YSpl@p1-1	C
YT@YSpl@p2-0	C
YT@YSpl@p2-1	C
PK@C1@sample	C
PK@C1@A/S	C
PK@C1@shape	C_PT
PK@C1@center	C
PK@C1@surface	C
PK@C1@amplitude	C
PK@C1@w0	C
PK@C1@wD	C
PK@C1@d0	C

Param.numbers	
0.1	0
2	0
4280	0
2.8	0
4288	0
2.6	0
4295	0
2.8	0
11	0
0	0
4290	0
0.5	0
0.1	0
0.01	0
0.001	0
0.001	0
0.1	0
0	0

4. Analýza výstupu

Tlačidlami "Go_fdata_fast" a "Go_fdata_complete" sa púšťa výpočet so zadanými vstupnými parametrami. V poli "return" type sa zobrazuje výstupný kód a dáta, ktoré poskytne DLL knižnica sa použijú na vykreslenie grafu v pravej časti.



Záver