BLOKOVĚ ORIENTOVANÁ ANALÝZA DLOUHODOBÝCH ZÁZNAMŮ BIOLOGICKÝCH SIGNÁLŮ

P. Vlk[1,2], J. Otáhal[2], P. Jiruška[2]

[1] Fakulta elektrotechnická, České vysoké učení technické v Praze
[2] Fyziologický ústav, Akademie věd České republiky

Abstrakt

Při návrhu analýzy dlouhodobých záznamů většinou postupujeme ve dvou krocích. V prvním se soustředíme na charakteristický segment záznamu. Na jeho základě sestavujeme jednotlivé kroky analýzy, sledujeme dílčí výsledky, optimalizujeme parametry a ošetřujeme chybové situace. Druhým krokem je aplikace návrhu. K tomu je ve většině případů zapotřebí obohatit původní návrh o rozsáhlou správu souborů a tvorbu mezivýsledků a jejich výsledné sloučení. V rámci výzkumu epilepsie jsme proto pro potřeby analýzy dlouhodobých EEG záznamů vyvinuli v prostředí Matlab aplikaci, která spojuje oba kroky do jediného. Umožňuje dynamickou změnu procesu analýzy za použití rychle dostupných funkcí. Obdobně jako Simulink k tomu využívá princip vzájemně propojených bloků.

1 Úvod

V současné době existuje řada aplikací, které umožňují analýzu biologických signálů. Základním kritériem, podle kterého mohou být tyto aplikace rozděleny, je účel, ke kterému jsou využívány. Zejména na poli výzkumu často narážíme na omezení a nedostatky, které tyto aplikace mají při použití ke konkrétním účelům daného projektu. Specifita daných výzkumných projektů před nás staví nutnost vytvářet vlastní algoritmy, které splňují potřeby daného projektu.

V rámci experimentálního výzkumu epilepsie se setkáváme se značným objemem různorodých dat, které je nutno analyzovat jako celek. Jedná se zejména o vícekanálové EEG záznamy a případně přidružené obrazové a zvukové záznamy, které jsou získány dlouhodobou monitorací spontánní, či evokované EEG aktivity v modelech epilepsie. Počet a charakter snímaných EEG kanálů se liší v závislosti na uspořádání experimentu. Variabilním faktorem je také délka záznamů. Ta je ovlivněna mnoha faktory a pohybuje se minimálně v rámci týdnů. Z technických důvodů nejsou výsledné záznamy ryze kontinuální a obsahují výpadky v záznamu o různé délce.

V prostředí Matlab existuje řada dostupných aplikací a toolboxů, které lze využít zpravidla pro analýzu krátkých úseků výše zmíněných EEG dat. Vybrané aplikace umožňují přidávat vlastní rozšíření, nicméně jsme limitováni původním návrhem vnitřní struktury. Jejich využití za účelem zpracování dlouhodobých záznamů ve většině případů není možné a takový účel vyžaduje vytvoření zcela nové aplikace nebo algoritmu. Nová aplikace musí brát v potaz výpadky v kontinuitě záznamu a kombinovat manuální zpracování multimediálních záznamů (nejčastěji značení záchvatů a nežádoucích artefaktů). Dále je třeba zajistit rozsáhlou správu souborů a tvorbu mezivýsledků a jejich výsledné sloučení. Bez takové správy dochází zpravidla k redundantním výpočtům z důvodu optimalizace analýzy nebo v důsledku nečekaného přerušení průběhu aplikace.

Při vývoji aplikace splňující potřeby konkrétního experimentu zpravidla narážíme na otázku všestrannosti výsledné aplikace. Z dlouhodobého hlediska je efektivní vytvářet banky funkcí a algoritmů, které lze využít v budoucnosti bez nutnosti jejich dalších modifikací. Na základě této myšlenky vzniká většina toolboxů, jež jsou volně k dispozici od jiných autorů. Vývoj dostatečně generalizovaných kódů však trvá podstatně déle, než striktně specializovaných. Zejména na poli analýzy dlouhodobých záznamů biologických signálů jako celku, se upřednostňuje specializovaný přístup. Generalizovaná varianta není díky specifitě daných signálu často možná. Vývoj projektu a získané výsledky často vyžadují modifikaci metodických postupů, které jsou spojeny i s nezbytností změnit kód aplikace. Specifickými důvody mohou být např. odlišnost datových typů, ošetření specifických artefaktů a rušení v záznamech či chyb zanesených lidským faktorem.

Faktem zůstává, že za použití všestranných kódů (toolboxů) lze velmi rychle provádět předběžnou analýzu. Lze tak ušetřit značné množství času ve fázi návrhu cílové analýzy. Cílem této

práce je návrh a vytvoření robustní aplikace, která by umožnila efektivně a flexibilně modifikovat a okamžitě aplikovat specifický typ analýz na záznamy dlouhodobých biologických signálů a jejíž vlastnosti by umožnily kompenzovat potřebu modifikace původních algoritmů

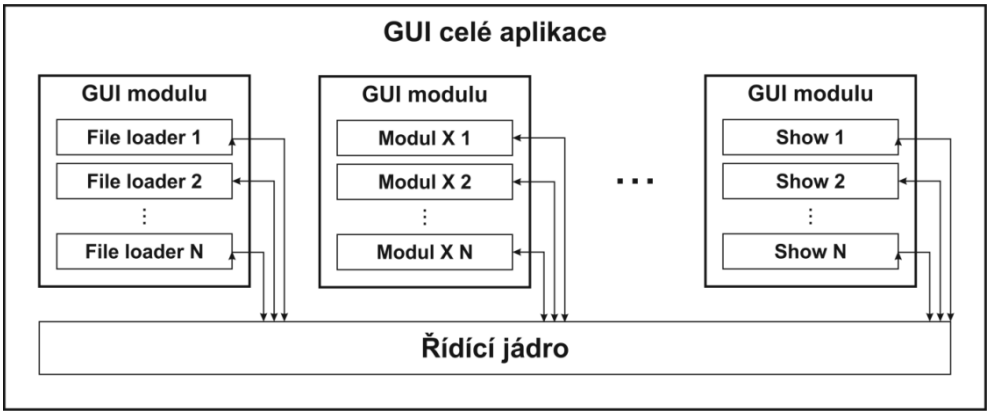
2 Teoretický koncept

Základním předpokladem pro všestrannost a rozšiřitelnost aplikace, je vhodný návrh její vnitřní architektury. V této práci jsme využili princip vzájemně propojených bloků, které mezi sebou komunikují na základě sady společných příkazů. Tyto příkazy zastřešují pouze meziblokovou komunikaci a jsou nezávislé na charakteru analyzovaných dat. K tomuto odlišení slouží princip základních datových typů, které jednotlivé bloky dědí. Tímto způsobem je vytvořena hierarchická struktura, jejímž vrcholným bodem je zásuvný modul aplikace. Jeho jednotlivé instance (bloky) slouží ke specifickému zpracování daného datového typu/typů, jež je povolen jako vstupní blok/bloky.

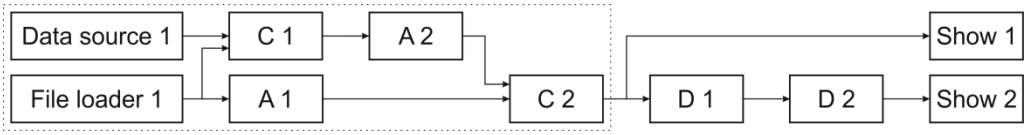
Předchozí krok umožňuje volnost k vytváření vlastních modulů, které mohou mít jak univerzální, tak striktně specifický účel. Lze také vytvářet vlastní datový typ, aniž by došlo k narušení konzistence výsledné sítě bloků. Pro správu jednotlivých modulů slouží podpůrný rámec. Ten je tvořen dvěma prvky. Prvním je řídící jádro, které slouží pro sběr informací o jednotlivých modulech a v případě potřeby upravuje jejich nastavení. Kontroluje tak konflikt jmen, načítá uložená nastavení modulů, poskytuje seznamy existujících bloků na základě jejich datového typu, tvoří diagramy jednotlivých sítí a zajišťuje ukládání a načítání schémat vytvořených analýz.

Druhým prvkem je grafické prostředí řídícího jádra, které obsahuje jeho volání a funkce společné pro všechny moduly. Pro každý modul je vytvořena samostatná záložka s identickým prostorem pro grafické prostředí modulu. Jednotlivé instance (bloky) modulu mají tedy společné grafické prostředí.

Výsledná architektura (Obr. 1) dovoluje analýzu jak krátkodobých, tak dlouhodobých záznamů biologických signálů. To je zajištěno existencí specializovaného modulu. Ten dává pokyny pro načítání jednotlivých souborů tvořících dlouhodobý záznam a výsledky na základě nastavených skupin slučuje. Sám se následně chová jako zdroj dat pro další kroky analýzy (Obr. 2).



Obr. 1: Schéma vnitřního uspořádání aplikace



Obr. 2: Demonstrace principu analýzy dlouhodobých signálů. Ohraničená skupina bloků zpracovává jednotlivé soubory na základě pokynů z bloku *D 1*. Blok *D 2* dává pokyny bloku *D 1* pro zpracování jednotlivých skupin souborů. Blok *Show 1* slouží k zhodnocení zpracování prvního souboru a je následně vyčištěn a ignorován. K zobrazení výsledné analýzy slouží blok *Show 2*. Aplikace podporuje nejen vstup dat ve formě souborů (*File loader 1*), ale také dle potřeby syntetizovaná data (*Data source 1*).

2.1 Společné jádro modulů

Jelikož je celá architektura navržené aplikace postavena na zásuvných modulech, je zapotřebí, aby jádro všech modulů bylo identické. Toho je dosaženo pomocí dědičnosti tříd. Výsledná třída daného modulu musí navíc obsahovat implementaci sady abstraktních metod takto zděděných. Stejným způsobem je určován také datový typ daného modulu. Ulehčuje se tak implementace nových modulů a zajišťuje vzájemná kompatibilita a komunikace.

Výsledná síť bloků je poté schopna automaticky reagovat na řadu situací. Jde například o změnu či odstranění předchozího bloku, kdy dojde k resetování nebo přepočítání obsahu všech návazných bloků. V případě přílišné náročnosti na RAM či fyzickou paměť je počítáno se sadou příkazů pro jejich automatické uspoření. Jedná se o ukládání mezikroků a jejich následné čtení z disku pro uspoření RAM či zpracování dat při každém jejich vyžádání pro uspoření obojího. Samo jádro také umožňuje adekvátní chování vůči modulům s poloautomatickým či manuálním zpracováním dat.

Společné pro všechny bloky jsou také metody pro smazání, přejmenování, uložení, načtení a převedení do původního nastavení. V jádru je také obsažen mechanizmus, který umožňuje zpracování dlouhodobých záznamů. Ten zaručuje připravenost sítě na načtení dalšího souboru a vyčkávání jednotlivých bloků na přepočítání všech jejich vstupů. Je tak zaručena správná komunikace bloků v místech větvení sítě.

2.2 Konzistence sítě bloků

Pro zajištění robustnosti výsledné sítě jsme stanovili dvě základní bezpečnostní opatření, která nemohou být součástí jádra jednotlivých modulů. Nicméně musí být respektována. Každý jednotlivý blok si sám sestavuje seznam ostatních bloků, které mohou sloužit jako jeho vstup. Vstupním blokem nesmí být daný blok samotný, ani žádný z bloků, jež na něj v síti navazují. Druhé opatření se týká zdrojů dat během analýzy více souborů. Modul dávající pokyny pro načítání nových souborů a jejich souhrnného zpracování musí kontrolovat, zda jsou všechny předřazené datové zdroje kompatibilní. Na demonstrovaném principu analýzy (Obr. 2) se jedná o blok D 1 a D 2. Pokud by existovalo v síti spojení, které by obcházelo blok D 1, nebyl by daný návrh funkční.

Aby byla zajištěna co největší všestrannost vytvářených modulů, předpřipravili jsme čtyři všeobecné datové typy, které jsou nejvíce použitelné v rámci zpracování biologických signálů. První typ slouží k uchovávání časových řad z jednoho a více kanálů o stejném počtu vzorků. Druhý typ obsahuje tabulku určující úseky a události v prvním typu. V třetím typu jsou uloženy atributy, které popisují úseky či události v druhém typu. A čtvrtý typ obsahuje informaci o rozdělení obsahu druhého typu na základě třetího do skupin.

2.3 Správa mezikroků

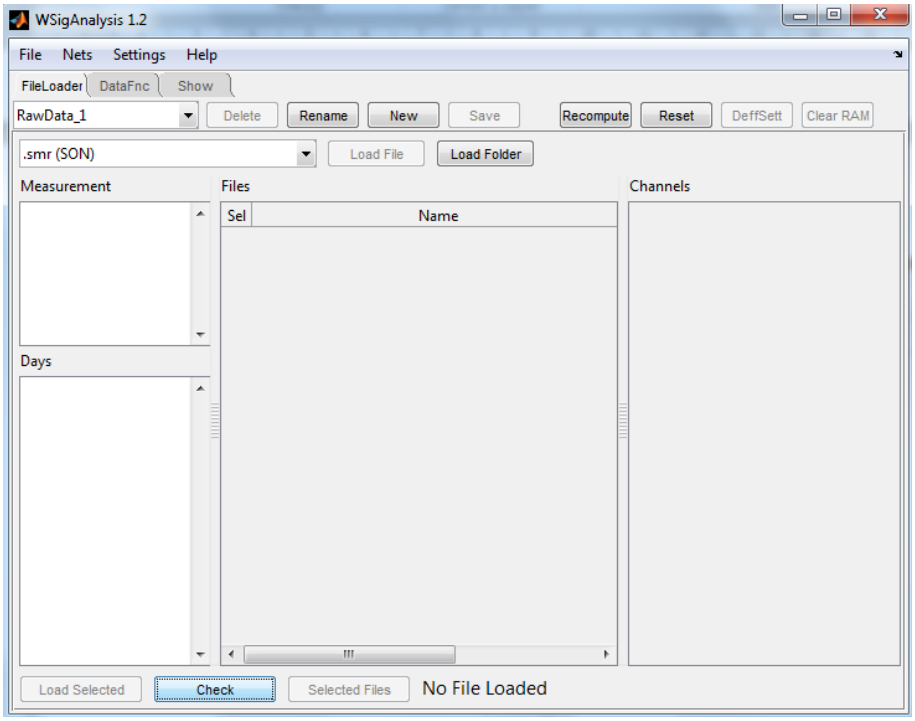
Jednotlivé bloky jsou díky společnému jádru schopny ukládat pro každý zpracovávaný soubor své výsledky. Při opětovném načtení schématu analýzy jsou tyto výsledky načteny bez zbytečných výpočtů. Nicméně musí dojít ke kontrole validity načítaných výsledků. Načítaný blok a ostatní jemu předcházející nesmí mít odlišná nastavení ani propojení. Při zpracování dlouhodobých záznamů je navíc potřeba ukládat dílčí výsledky z bezpečnostních důvodů. Moduly dávající pokyny pro načítání nových souborů musí být schopny vytvářet skupiny na základě seznamu pro analýzu vybraných souborů. Tyto moduly musí být dále schopny fungovat jako zdroj dat pro další kroky a vytvářet obdobný seznam. Díky tomu je možné rozdělit analýzu na dílčí celky a průběžně hodnotit výsledky již po zpracování první skupiny.

3 Vývoj aplikace

Jelikož je aplikace (Obr. 3) teprve v rané fázi vývoje, je zprovozněno a otestováno pouze jádro její architektury bez funkcí pro vyšší správu sítí. Komunikace mezi bloky funguje dle očekávání a neprojevily se nedostatky v kontrolních mechanizmech. Nicméně v současné době nemáme plně funkční modul dávající pokyny pro načítání nových souborů. Princip postupného načítání funguje, nejsme však momentálně schopni otestovat jeho chování jako zdroj dat pro další kroky.

Současně jsou rozpracovány základní čtyři moduly. Prvním je modul pro načítání souborů, který se nachází již ve fázi optimalizace. Druhý slouží pro vizualizaci kombinace základních datových typů. Jeho dílčí část je již schopna zobrazovat dlouhodobé záznamy s označenými událostmi a úseky.

Využili jsme zde principy rychlého překreslování grafů. Třetí a čtvrtý modul jsou v současnosti spojeny do jednoho. V budoucnu budou sloužit pro řízení načítání nových souborů a aplikaci externí uživatelem vybrané funkce na vstupní data.



Obr. 3: Náhled vlastní aplikace – modul File loader

4 Diskuze

Teoretický koncept aplikace slibuje budoucí urychlení výzkumné práce na našem pracovišti. Nelze však zatím s jistotou říci, zda aplikace samotná dokáže dostát očekáváním. Absence plně funkčního modulu pro řízení analýzy více souborů znemožňuje testování plné funkčnosti. V případě, že aplikace nebude schopna zpracovávat dlouhodobé záznamy dle našich předpokladů, ztrácí tím jednu ze svých hlavních výhod. Druhá hlavní výhoda a to blokový přístup je nicméně plně funkční. Jeho současnému využití brání opět nedostatek potřebných modulů.

Budoucí praktické využití ukáže, zda navržený přístup nepodporuje přílišnou tvorbu jednotlivých modulů. Ty by se poté staly nepřehlednými a musely by se členit do skupin, což by si vyžádalo potenciálně značný zásah do uživatelského rozhraní. Dalším krokem vývoje navržené aplikace bude praktické testování při analýze experimentálních dat, které prokáže skutečné klady a zápory aplikace.

5 Poděkování

Tato práce byla podpořena granty Ministerstva zdravotnictví České republiky (IGA NT/14489-3), Nadačního fondu Neuron (2012/10) a Grantové agentury České republiky (P303/14-02634S).

Ing. Pavel Vlk

Oddělení Vývojové Epileptologie, Fyziologický ústav AV ČR, v. v. i.
Vídeňská 1083, 142 20, Praha 4, Česká Republika
pavelvlk1988@gmail.com, Tel.: +420 241 062 495