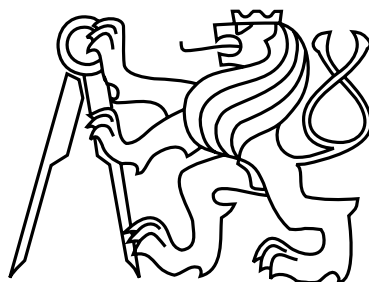


České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra řídicí techniky



Bakalářská práce

Anotátor dlouhodobých EEG záznamů

Václav Příhoda

Vedoucí práce: Ing. Václav Gerla, Ph.D.

květen 2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Václav Příhoda**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Anotátor dlouhodobých EEG záznamů**

Pokyny pro vypracování:

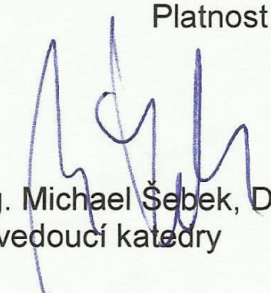
1. Navrhněte a implementujte programový modul, který bude umožňovat zobrazení a anotaci vícekanálového EEG záznamu. Klíčové vlastnosti tohoto modulu:
 - a) práce s dlouhodobými biomedicínskými signály (především celonoční EEG záznamy),
 - b) přehledné a intuitivní GUI v prostředí programu MATLAB,
 - c) možnost změny velikosti okna, ve kterém se budou zobrazovat jednotlivé signály,
 - d) možnost změny měřítka časové a amplitudové osy (pro výběr aktivního úseku záznamu bude použit scrollbar),
 - e) pokud je součástí analyzovaného záznamu hodnocení experta, zobrazí se tato informace společně s EEG signálem,
 - f) možnost manuální anotace (např. značení artefaktů, nebo EEG grafoelementů),
 - g) jasně definovaná datová struktura pro uložení nastavení všech použitých metod.
2. Ověřte realizovaný modul nad reálnými klinickými záznamy.

Seznam odborné literatury:

- [1] V. Gerla, V. Djordjevic, L. Lhotska, V. Krajca. PSGLab Matlab Toolbox for Polysomnographic Data Processing: Development and Practical Application, Proceedings of 10th International Conference on Information Technology and Applications in Biomedicine, IEEE, 2010. Online: <http://bio.felk.cvut.cz/psglab/>
- [2] A. Delorme and S. Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of Neuroscience Methods, 134(1):9-21, 2004.
- [3] E. Niedermeyer, F. Lopes da Silva. Electroencephalography - Basic principles, clinical applications and related field. Philadelphia: Lippincott William & Wilkins, S. 1165-97, ISBN 0-7817-5126-8, 2005.

Vedoucí: Ing. Václav Gerla, Ph.D.

Platnost zadání: do konce zimního semestru 2013/2014


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 2. 1. 2013

Poděkování

Rád bych zde poděkoval Ing. Václavu Gerlovi, Ph.D. vedoucímu této bakalářské práce za veškeré připomínky, konzultace a rady. Dále chci poděkovat své rodině, za neutuchající podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 23. 5. 2013

.....

Abstrakt

Cílem práce je návrh a implementace programového modulu, který umožňuje prohlížení a anotaci dlouhodobých biomedicínských záznamů. Tento modul je součástí vyvíjeného grafického prostředí pro PSGLab. Práce je rozdělena na dvě části. První část je teoretická, má sloužit pro získání přehledu o elektroencefalografii a tvorbě GUI v programu Matlab. Druhá část práce je praktická a obsahuje popis ovládání a popis funkcí vytvořeného modulu. V praktické části je také řešeno ověření funkčnosti na reálných datech.

Klíčová slova: EEG, GUI, artefakty, grafoelementy, anotátor

Abstract

The aim of this thesis is the proposal and implementation of software module, which enables viewing and annotation of long-term biomedical records. This module is a part of the developed graphical user interface for PSGLab. The thesis is divided into two parts. The first part is theoretical and its goal is to give an overview of electroencephalography and creation of GUI in Matlab. The second, practical part includes a description of control and functions in the created module. Finally, the practical part includes verification of the functionality on real data.

Keywords: EEG, GUI, artifacts, graphoelements, annotator

Obsah

1	Úvod	1
2	Elektroencefalografie	2
2.1	Vznik a vývoj EEG	2
2.2	Záznam EEG	3
2.2.1	Rozmístnění elektrod	3
2.3	Frekvenční pásma EEG signálu	4
2.4	Grafoelementy	5
2.5	Artefakty	6
3	Matlab a GUI	7
3.1	Handle Graphics	7
3.2	Figure	8
3.2.1	Uicontrol	8
3.2.2	Ostatní grafické objekty Figure	9
3.2.3	Axes	10
3.3	Funkce Callback	10
4	PSGLab	11
4.1	Hlavní metody PSGLabu	11
4.2	GUI pro PSGLab	12
4.2.1	Načtení a předávání dat	12
5	Návrh a tvorba anotátoru	13
5.1	Ovládací prvky anotátoru	13
5.1.1	Horizontální posuvník	14
5.1.2	Channel options menu	14
5.1.3	Tags menu	16
5.1.4	Ovládání pomocí kláves	16
5.2	Vykreslení	17
5.2.1	Vykreslení záznamu	18
5.2.2	Vykreslení značek	19
5.3	Nastavení - Preferences	20
5.3.1	Window size menu	20
5.3.2	Visible channel menu	21
5.3.3	Uložení nastavení	22

5.4	Anotace záznamu	22
5.4.1	Struktura dat	22
5.4.2	Přidání značky	23
5.4.3	Odebrání značky	24
6	Ověření na reálných datech	26
6.1	Záznamy skupiny Biodat	26
6.2	Dlouhodobé záznamy	27
6.3	Ostatní záznamy	28
7	Závěr	29
	Literatura	30
A	Uživatelská příručka	31
B	Obsah přiloženého CD	34

Seznam obrázků

2.1	Rozmístnění elektrod v systému 10-20 [4]	3
2.2	Průběhy EEG signálu [4]	5
2.3	Epileptický grafoelement [7]	5
2.4	Artefakty [7]	6
3.1	Hierarchie grafických objektů	7
3.2	Ukázka všech dostupných stylů v rámci Uicontrol	8
4.1	Výsledek segmentace signálu [11]	12
5.1	GUI EEG anotátoru	13
5.2	Channel option menu	15
5.3	Preferences	21
5.4	Struktura ukládání parametrů značky	23
6.1	Testování anotace	26
6.2	Testování anotace dlouhodobého záznamu	27
6.3	Ukázka okna nastavení pro záznam s 65 kanály	28
6.4	Ukázka vykreslení záznamu s 65 kanály	28
A.1	Hlavní okno anotátoru	31
A.2	Nastavení vlastností záznamu	32

Seznam tabulek

2.1	Rozdělení frekvenčních pásem	4
-----	--	---

Seznam použitých zkratek

EEG Elektroencefalografie

EOG Elektrookulografie

EKG Elektrokardiografie

EMG Elektromyografie

GUI Graphical user interface

GUIDE Graphical user interface development environment

Kapitola 1

Úvod

Vznik elektroencefalografie nastal před více jak osmdesáti lety, když se v roce 1924 Hansu Bergerovi podařilo zaznamenat elektrické potenciály na povrchu hlavy. Přesto je dnes stále rozhodující subjektivní hodnocení lékaře, během kterého lékař musí zkontrolovat i několik hodin trvající záznam. Tento způsob je značně neefektivní. Hodnocení záznamu probíhá na základě zkušeností. Stejně tak je problematický samotný záznam, neboť záznamy různých pacientů se mohou značně lišit. Dlouhodobým cílem je vyvinout nástroj, který by lékařům usnadnil jejich práci.

Cílem práce je návrh anotátoru dlouhodobých EEG záznamů. Jak již napovídá slovo anotátor, modul by měl být schopen vkládat a odstraňovat značky do zobrazeného záznamu. Na rozdíl od EEGLabu [1] umožní přidávat značky k vybranému kanálu. Jedním z požadavků je vytvoření grafického prostředí, zkráceně GUI. GUI by mělo usnadnit práci při anotaci záznamu specialisty. Tato práce je součástí vytvářeného grafického rozhraní pro PSGLab, na kterém se se svou bakalářskou prací podílí Petr Štěpánek.

Kapitola 2 seznamuje s historií elektroencefalografie, popisem frekvenčních pásem a grafoelementů. Kapitola 3 je úvodem do problematiky tvorby GUI v programu Matlab. Tato kapitola obsahuje popis základních grafických objektů. Kapitola 4 se věnuje PSGLab toolboxu, obsahuje výčet hlavních funkcí. Zmiňuje také vývoj grafického rozhraní PSGLabu. V kapitole 5 se práce věnuje samotnému návrhu anotátoru. Obsahuje popis hlavních ovládacích prvků, způsob vykreslení záznamu a značek. Věnuje se způsobu vkládání a odstraňování značek. Kapitola 6 se zabývá ověřením funkce anotátoru na reálných datech. Příloha A obsahuje příručku pro uživatele, příloha B obsah přiloženého CD.

Kapitola 2

Elektroencefalografie

Elektroencefalografie zkráceně nazývaná EEG, je neinvazivní metoda záznamu elektrických potenciálů z povrchu hlavy. Pro lékaře je EEG hlavní pomůcka pro diagnózu epilepsie, zranění hlavy, poruchy spánku, psychických poruch, neurologických poruch a kontrola centrální nervové soustavy u novorozenců [2].

2.1 Vznik a vývoj EEG

Základem nervové soustavy je neuron. Mozek obsahuje přibližně 100 miliard neuronových buněk. Neuron přenáší a zpracovává informace. Objevitelem neuronu je J.E. Purkyně, který ho v roce 1835 popsal. Podstatné části neuronové buňky jsou synaptická spojení, tělo buňky a axon. Synaptická spojení jsou místa, kde dochází k příjmu informací a ukládání znalostí od jiných neuronů. Tělo buňky (soma) obsahuje jádro, zde dochází ke zpracování přijatých informací. Axon pak přenáší tyto zpracované informace do dalších neuronů. Přenos signálu mezi neurony je elektrochemický proces, ve kterém dochází k pohybu nabitých iontů, tím v jejich okolí vzniká elektrický potenciál.

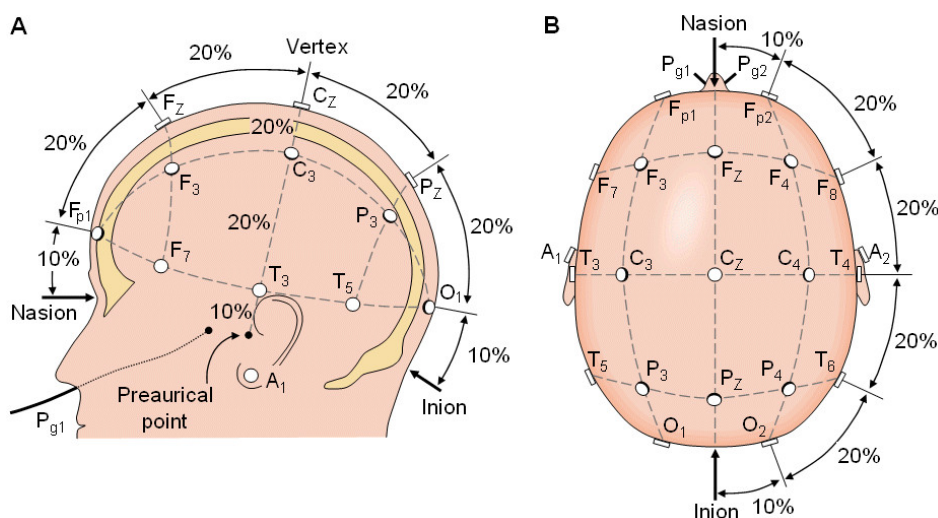
V roce 1875 Richard Caton zaznamenal elektrické potenciály na mozku králíků, koček a opic. Objevitelem lidské elektroencefalografie je Hans Berger. Se studiem lidské EEG začal v roce 1920. V roce 1924 začal u pacientů zaznamenávat EEG. Nejprve prováděl měření na lidech se zlomeninami lebečních kostí. Brzy přišel na to, že zlomená lebeční kost není výhodou pro záznam a zjistil, že záznam může být stejně dobrý, nebo i lepší i přes nepoškozenou lebeční kost. Hans Berger také zavedl pojem *elektrenkephalogram*. V roce 1953 Eugen Aserinsky a Nathaniel Kleitman popsali REM spánek [3].

2.2 Záznam EEG

Ke snímání EEG slouží elektroencefalograf. Jeho součástí jsou elektrody, které snímají elektrické potenciály. Tyto elektrické potenciály jsou součtem všech elektrických potenciálů velké skupiny neuronů v okolí elektrod. Signál se snímá z povrchu hlavy, ale je velmi slabý. To je způsobeno průchodem přes mozkovou kůru, lebku a skalp. Amplituda signálu je v řádů desítek μV . Pro lepší zpracování je potřeba měřený signál zesílit. Při měření je zapotřebí dát pozor na odstínění a na správný kontakt elektrody s pokožkou, k tomu se na elektrody nanáší vodivý gel.

2.2.1 Rozmístnění elektrod

Při prvním měření Hansem Bergerem elektrody byly umístěny vpředu a vzadu hlavy. Při měřeních bylo dokázáno, že se EEG aktivita liší v různých částech hlavy. To vedlo ke snahám rozmístit elektrody tak, aby se zlepšil záznam určité aktivity. Sledování různých typů aktivit vedlo k využití více elektrod. Zvýšení počtu elektrod následovala snaha umísťovat elektrody pomocí standardu, který by umožňoval porovnávat záznam napříč časem a různými pacienty. V roce 1958 byl mezinárodně přijat systém rozmístnění elektrod navržený Herbertem Jasperem. Tento systém se značí 10-20. Název 10-20 je odvozen od způsobu rozmístnění elektrod. Vzdálenosti mezi elektrodami jsou 10% nebo 20% z celkové vzdálenosti. Tento standard je navržen pro 21 elektrod.



Obrázek 2.1: Rozmístnění elektrod v systému 10-20 [4]

Elektrody, které se nacházejí nad levou hemisférou jsou označeny lichými čísly, zatímco elektrody nad pravou hemisférou jsou označeny sudými čísly. Elektrody s označením *z* se nacházejí uprostřed. Písmenem *F* se značí elektrody ve frontální (čelní) oblasti, *Fp* frontopolární, *C*

centrální, *P* pariteální (temenní), *O* okcipitopolární (zátylní), *T* temporální (spánkový). Nasion (kořen nosu) a Inion jsou anatomicky definované výstupky na lebce, jsou to základní body pro rozmístnění elektrod [5]. Rozmístnění a popis elektrod je vidět na obr. 2.1.

2.3 Frekvenční pásma EEG signálu

Podle frekvence dělíme signál do pěti hlavních frekvenčních pásem. Se vzrůstající frekvencí jsou tato pásma pojmenována delta, theta, alfa, beta a gamma. Alfa a beta vlny popsal Hans Berger. H. Jasper a Andrews použili označení gamma pro vlny s frekvencí vyšší jak 30 Hz. Delta pásmo bylo představeno W. Grey Walterem k popisu signálů s nižší frekvencí než je rozsah alfa pásma. Také stanovil, že theta pásmo se nachází v rozmezí 4 - 7.5 Hz [6].

Delta δ	0.5 - 4 Hz
Theta θ	4 - 8 Hz
Alfa α	8 - 13 Hz
Beta β	13 - 30 Hz
Gamma γ	Více jak 30 Hz

Tabulka 2.1: Rozdělení frekvenčních pásem

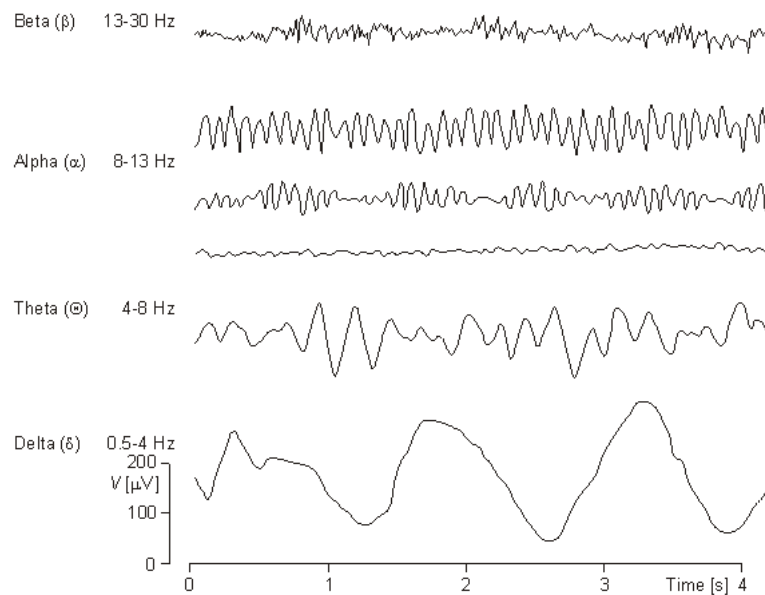
Delta - Tyto vlny jsou spojeny hlavně s hlubokým spánkem. Projevují se u dětí do prvního roku. Přítomnost delta vln během bdělosti je u dospělého člověka vždy patologická [5]. Mohou se snadno zaměnit s artefakty, způsobenými svaly na krku.

Theta - Označení theta je nejspíše odvozen od původu těchto vln v thalamu. Theta vlny mají významnou roli u kojenců. Větší aktivita theta vln v bdělém stavu u dospělého člověka může být patologických projevem. U theta vln se zkoumá jejich vliv na emoce a dospívání.

Alfa - Alfa vlny se projevují v zadní polovině hlavy nad týlní oblastí mozku. Jsou nejvýznamnějším projevem bdělého člověka. Alfa vlny jsou spojeny s uvolněnou myslí, bez jakéhokoliv soustředění. Tlumí se při duševní činnosti, otevřených očích nebo zaslechnutím neznámého zvuku. Jsou charakteristické pro fázi těsně před usnutím.

Beta - Beta vlny jsou spojené s aktivním myšlením při řešení problémů a vnímání světa. Jsou pozorovatelné u dospělých lidí. Vlny s vysokou amplitudou lze získat, pokud je člověk v panickém stavu. Aktivita beta vln je zaznamenána hlavně přes čelní a centrální oblast.

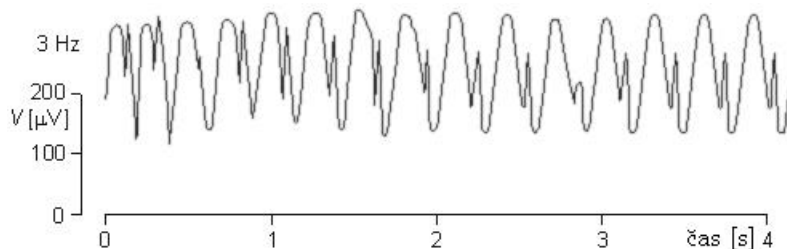
Gamma - Amplitudy těchto vln jsou velmi nízké a jejich výskyt je vzácný. Detekcí těchto vln může vést k potvrzení určitého onemocnění mozku.



Obrázek 2.2: Průběhy EEG signálu [4]

2.4 Grafoelementy

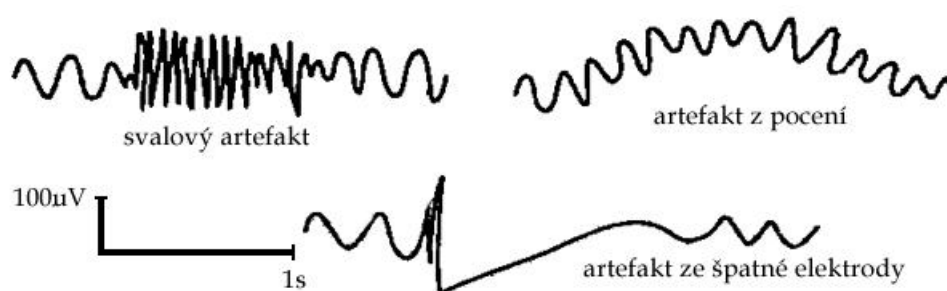
Detekce grafoelementů je metoda analýzy EEG. Detekci, která je časově náročná, provádí lékaři. Při dlouhodobých záznamech je to velmi náročná práce. Mezi grafoelementy řadíme různé osamocené vlny, hroty, popř. jejich složitější komplexy. Jako příklad může sloužit komplex hrotu a vlny. Hrot je úzká vlna trvá okolo 200-500 *ms*. Tento grafoelement má pak souvislost s epileptickým záchvatem. Dalším příkladem grafoelementu je tzv. lambda vlna, která je spjata s upoutáním zraku [7].



Obrázek 2.3: Epileptický grafoelement [7]

2.5 Artefakty

Artefakty jsou rušivě působící grafoelementy. Artefakty nejsou způsobeny činností mozku. Artefakty je možné rozdělit podle jejich původu na biologické a technické. Mezi biologické můžeme zařadit artefakt svalový, oční, pulsový a artefakt z pocení. U technických artefaktů to pak bývá síťové rušení s kmitočtem 50 Hz nebo špatný kontakt elektrody s lebkou na obr.2.4. Řadu těchto projevů má za úkol odstranit analogový filtr při záznamu, další možností je číslicová filtrace při diskrétním zpracování [7]. Srdce vytváří dva typy artefaktů, elektrické a mechanické. Oba jsou snadno rozpoznatelné synchronizováním s naměřeným EKG záznamem. Artefakty způsobené svaly jsou nejvýraznějším rušením EEG záznamu. Tato svalová aktivita téměř vždy překryje EEG záznam, protože má větší amplitudu a frekvenci [8].



Obrázek 2.4: Artefakty [7]

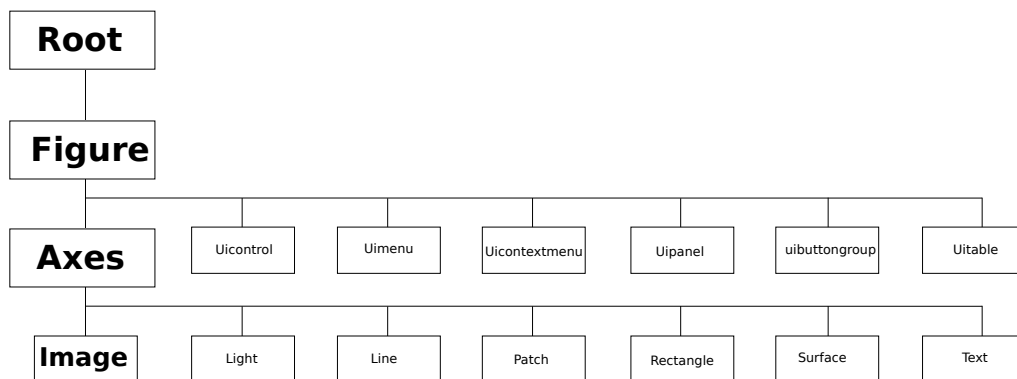
Kapitola 3

Matlab a GUI

Matlab je vysokoúrovňový programovací jazyk a interaktivní prostředí pro numerické výpočty a vizualizaci. Matlab se využívá pro analýzu dat, vývoj algoritmů a vytváření modelů a aplikací, zpracování signálů, obrazu a videa. Matlab je také schopný vytvářet Graphical user interface, zkráceně GUI. Matlab poskytuje nástroj pro automatické generátorování GUI. Nazývá se Graphical User Interface Development Environment, zkráceně **GUIDE**. Po jeho spuštění je možnost vytvářet grafické prostředí interaktivně.

3.1 Handle Graphics

Grafické objekty jsou základní elementy používané k zobrazení jakéhokoli grafického výstupu [9]. Handle Graphics je grafický systém implementovaný do systému Matlab, s jehož pomocí lze efektivně pracovat s grafickými objekty, ať už se jedná o 2D a 3D grafy nebo o interaktivní tlačítka, textová pole, menu atd. Handle je proměnná, která je při vytvoření objektu naplněna číslem, které identifikuje daný grafický objekt. Každý objekt má různou hodnotu handle. Na obr. 3.1 je znázorněné hierarchické uspořádání grafických objektů. Základním objektem



Obrázek 3.1: Hierarchie grafických objektů

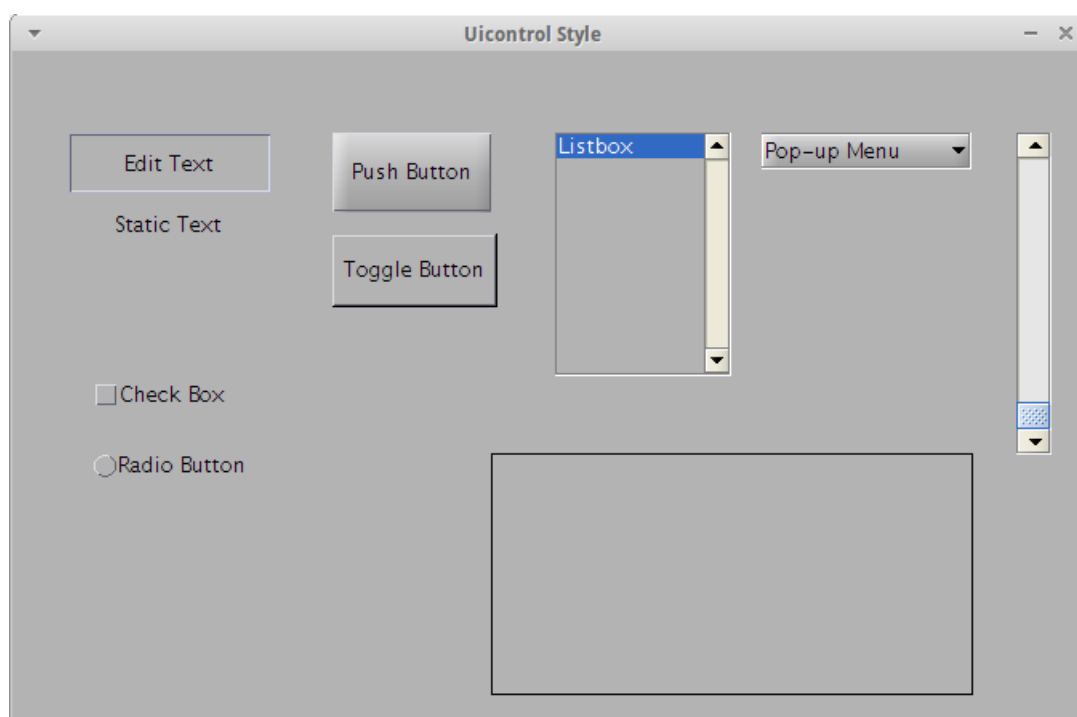
je **Root**, tomu jsou podřízeny všechny zbylé objekty. Objekt **Root** odpovídá obrazovce počítače. Hierarchický systém grafických objektů umožňuje dědičnost vlastností.

3.2 Figure

Figure je okno, ve kterém se zobrazuje grafika a uživatelské rozhraní. Pomocí příkazu **h=figure** se vytvoří okno s implicitní velikostí a je vytvořeno na dané pozici obrazovky. V proměnné *h* je uložena hodnota handle. Příkazem **get(h)**, lze zjistit všechny vlastnosti vytvořeného objektu. Pokud se objekt figure vytvoří bez uložení jeho hodnoty handle, je možné použít příkaz **gcf** (get current figure), který určí hodnotu handle vybraného objektu typu figure. Příkazem **set** lze tyto vlastnosti upravovat.

3.2.1 Uicontrol

Grafické objekty Uicontrol jsou podřízeny objektům figure. Jelikož jsou tyto prvky vytvářeny uvnitř objektů figure, jsou jejich souřadnice počítány uvnitř figure a nikoliv v rámci celé obrazovky (Root). Při vytváření objektu uicontrol je nutné vybrat si jeden z deseti různých stylů viz. obr. 3.2.



Obrázek 3.2: Ukázka všech dostupných stylů v rámci Uicontrol

Checkbox je zaškrťovací políčko. Mění svůj vzhled v případě kliknutí myši na tento objekt.

Hodnota `Value` je vlastnost objektu a nabývá dvou stavů. Pokud není označen, hodnota `Value` je `0` a pokud je označen hodnota `Value` je `1`.

Edit je grafický objekt, do kterého lze vkládat text. Objekt má vlastnost `Value`, v té je uložen vložený text ve tvaru string.

Listbox zobrazí seznam prvků, kde je možné označit buď jeden prvek, nebo více prvků současně. Hodnota `Value` obsahuje index vybraného prvku. V případě označení více prvků obsahuje vektor těchto indexů.

Popupmenu je podobný objektu `listbox`. Je možné vybrat pouze jeden prvek v seznamu. Výhoda je především v úspoře místa. `Value` obsahuje index (pořadové číslo) vybraného prvku.

Pushbutton je tlačítko. Důležitá vlastnost tohoto objektu je jeho `Callback` funkce. K zavolání `Callback` funkce dojde po stisku tlačítka.

Radiobutton se chová obdobně jako objekty typu `checkbox`. V případě skupiny těchto tlačítek je možné vybrat pouze jednu volbu. Ke sdružování těchto prvků tomu slouží objekt `uibuttongroup`.

Slider nebo-li posuvník. Pro správnou funkci je potřeba definovat rozsah `MIN` a `MAX`. V tomto rozsahu se slider pohybuje. V hodnotě `Value`, je uložena aktuální číselná poloha posuvníku.

Text je textové pole, do kterého není možné zapisovat. Používá se pro označení ovládacích prvků.

Togglebutton je tlačítko podobné jako `pushbutton`. Graficky vyjadřující svůj stav. Po stisknutí tlačítka se nevrátí do původní polohy jako tlačítko `pushbutton`.

Frame je grafické ohraničení. Je doporučeno používat spíše `uipanel` [10].

3.2.2 Ostatní grafické objekty Figure

Podle hierarchického znázornění na obr. 3.1 jsou patrné další grafické objekty Figure, které jsou na stejné úrovni.

Uimenu slouží k tvorbě menu.

Uicontextmenu se vytváří v případě, že je potřeba, aby se kontextové menu objevilo při kliknutí pravým tlačítkem myši na graf.

Uipanel vytvoří grafický kontejner pro grafické objekty Figure.

Uibuttongroup je také grafický kontejner. Řídí přepínání jednotlivých objektů typu radiobutton, které obsahuje. Může obsahovat i další grafické objekty.

Uitable vytvoří grafický objekt - tabulku.

3.2.3 Axes

Grafický objekt axes je nepostradatelný při tvorbě grafů. Objekt axes se vytvoří příkazem **axes**. Stejně jako u objektu figure je přístup k vlastnostem zajištěn příkazy **set** a **get**. Pokud není známa hodnota handle objektu axes, stačí použít příkaz **gca** (get current axes).

3.3 Funkce Callback

Funkce Callback je funkce, která se vyvolá po vykonání akce. Akce může být například stisk tlačítka, klávesy nebo pohyb posuvníku. Tato funkce umožňuje řídit, jak bude GUI reagovat na vyvolané události. Každý komponent má specifický druh této funkce. Příklad vytvoření Callback funkce pro prvek pushbutton.

```
pb=uicontrol('Style','pushbutton','String','Action','Callback',@Action);
```

V ukázce je využito volání anonymní funkce **Action**. Stisknutím tlačítka dojde k zavolání a provedení funkce **Action**.

Kapitola 4

PSGLab

PSGLab je MATLAB toolbox pro zpracování biomedicínských signálů, především polygrafických signálů. Tento toolbox je vyvíjen výzkumnou skupinou Biodat. Toolbox umožňuje nalézt důležité informace skryté v signálu a jejich správnou interpretaci. PSGLab byl vyvíjen především pro hodnocení klinických dat. Hlavním cílem je usnadnění práce specialistům.

Polygrafie je metoda snímání více fyziologických parametrů současně. Při polygrafii se zaznamenává mozková aktivita (EEG), oční pohyby (EOG) a svalová aktivita (EMG). Snímat lze také záznam srdeční aktivity (EKG), množství ventilovaného vzduchu v plicích, tělesnou teplotu, pohyby a pozice těla.

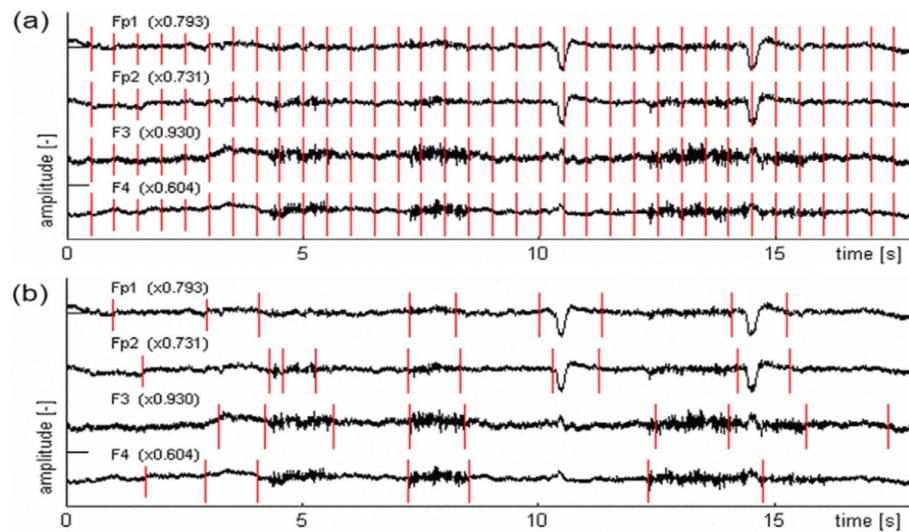
4.1 Hlavní metody PSGLabu

PSGLab zahrnuje metody na předzpracování, klasifikaci a vizualizaci dat. Používá předdefinovaný formát pro ukládání dat ve všech fázích procesu (vstupní signály, klasifikace, hranice segmentů, grafy a průběžné výsledky). PSGLab umožňuje práci s daty různých formátů, podporované formáty jsou např. ASCII, EaSys, BrainQuick a binární formáty, které se používají při záznamu polygrafických signálů.

Metody předzpracování signálu zahrnují filtraci síťového rušení, algoritmus na převzorkování, FIR filtry pro separaci jednotlivých frekvenčních pásem EEG signálu, průměrování signálu, spojení dvou nahrávek při záznamu odlišnými technikami. Dále obsahuje metody na odhalení artefaktů, způsobených svalovou aktivitou.

PSGLab umožňuje jak konstatní, tak adaptivní segmentaci a možnost klasifikace. Pokud ke klasifikaci chybí trénovací množina, umožňuje využít metody shlukové analýzy. K tomu

se používá algoritmus K-means nebo hierarchické shlukování. Pokud je dostupná testovací množina je použit K-NN klasifikátor [11].



Obrázek 4.1: Výsledek segmentace signálu [11]

4.2 GUI pro PSGLab

V předcházející kapitole byl stručně vypsán seznam funkcí PSGLabu pro zpracování polygrafických signálů. Komunikace s aplikací přes textové rozhraní, je pro neznalého uživatele velmi složitá nebo naprosto neúčinná. GUI umožňuje mnohem přívětivější interakci s uživatelem. Pro většinu z nich je důležité aby GUI bylo intuitivní. To je důvod vzniku GUI pro PSGLab. GUI vytváří Petr Štěpánek, který v rámci své bakalářské práce navrhl hlavní strukturu a s ní spojené potřeby. Anotátor dlouhodobých EEG záznamů je jedním z modulů vyvíjeného grafického rozhraní PSGLabu.

4.2.1 Načtení a předávání dat

Aplikace umožňuje načíst data ve formátu **.d*. To je zajištěno pomocí funkce **EASY2**. Po načtení se uživateli zpřístupní možnosti prohlížení a segmentace. Aby se v těchto modulech dalo pracovat s daty, je třeba je nejprve předat. K předávání dat se používá dvojice příkazů. Příkaz **getappdata** slouží pro získání dat, které jsou uloženy v objektu. Pro uložení dat do objektu slouží příkaz **setappdata**.

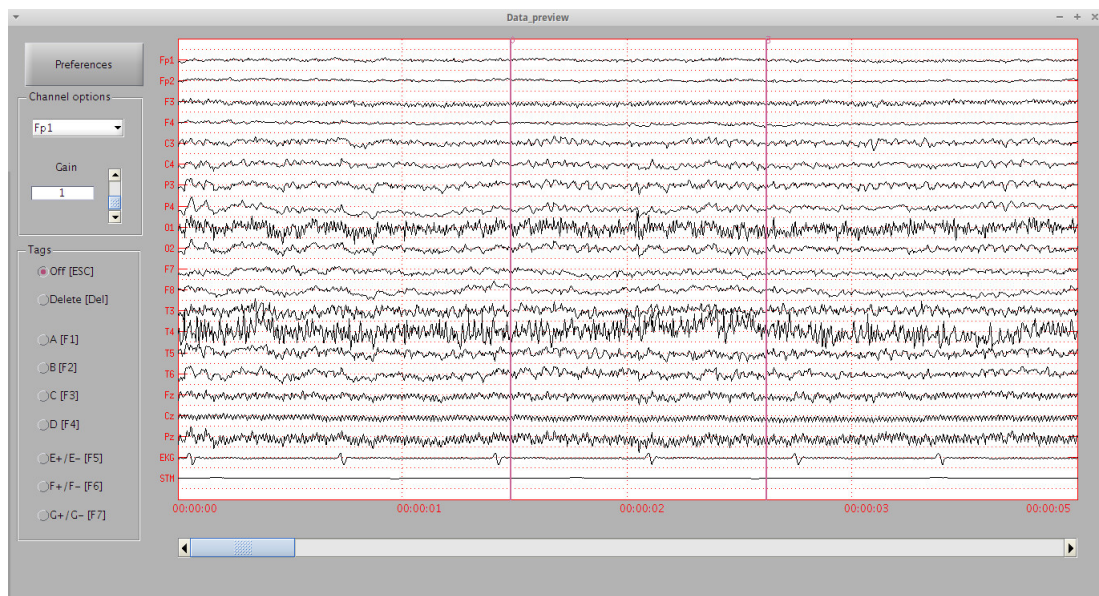
Kapitola 5

Návrh a tvorba anotátoru

Anotátor záznamů je součástí vytvářeného grafického rozhraní pro PSGLab. Umožňuje prohlížení EEG záznamů, vkládání a odebírání značek (tagů), do tohoto záznamu. Prohlížení umožňuje vypnutí a zesílení vybraných kanálů nebo změnit velikosti zobrazeného časového úseku. Níže popisovaný zdrojový kód se nachází v souborech *Data_preview.m* a *Preferences.m*.

5.1 Ovládací prvky anotátoru

Na obr. 5.1 je patrné uspořádání grafických objektů v anotátoru. Největší část zabírá objekt typu axes, ve kterém se záznam vykresluje. Posuvník, který je horizontální, je důležitou



Obrázek 5.1: GUI EEG anotátoru

součástí grafického rozhraní. Pomocí změny jeho polohy je umožněno prohlížení záznamu. Ohraničené menu s názvem *Channel options* se skládá ze třech grafických objektů. Popupmenu, textové pole a vertikální posuvník. Popupmenu obsahuje seznam všech načtených kanálů. Změnou polohy posuvníku je možné zesílit nebo zeslabit takto vybraný kanál. Aktuální hodnotu je možné sledovat v textovém poli, a jelikož je toto pole editovatelné, tak lze hodnotu zesílení vložit přímo do tohoto pole. Ohraničené menu s názvem *tags* je objekt typu `uibuttongroup`. Sdružuje všechny objekty typu `radiobutton` v sobě obsažené. Toto menu slouží k přidávání a odebírání značek. Tlačítko s názvem *Preferences* otevírá nové okno s nastavením. Manuál k ovládání anotátoru je přiložen v příloze A.

5.1.1 Horizontální posuvník

Pro správnou funkci posuvníku je nutné nastavit jeho parametry. Mezi parametry patří minimální, maximální hodnota posuvníku a velikost kroku.

```
step=1/(((handles.nsamp/handles.fsamp)/handles.ViewSet.WinSize));
Sstep=[step, step];
set(handles.slider1, 'Min', 0);
set(handles.slider1, 'Max', ...
((handles.nsamp/handles.fsamp))/handles.ViewSet.WinSize)-1);
set(handles.slider1, 'SliderStep', Sstep);
```

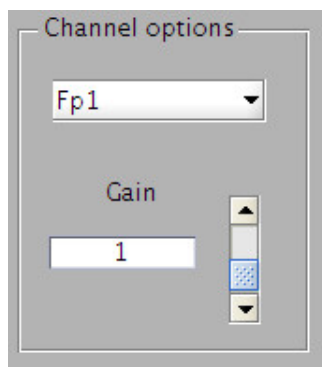
Handler posuvníku odpovídá hodnotě *handles.slider1*. Funkce **slider1_Callback** zpracovává akci, pokud k ní dojde.

```
function slider1_Callback(hObject, eventdata, handles)
handles.sval=get(handles.slider1,'Value');
fig(hObject,eventdata,handles);
```

Ve funkci dojde k načtení aktuální hodnoty posuvníku. Tato hodnota se uloží do proměnné *handles.Sval*. Poté je zavolána funkce vykreslení **fig**, aby došlo k aktualizaci.

5.1.2 Channel options menu

Pro toto menu je důležitá proměnná *handles.gain*. Ta obsahuje pro každý kanál hodnotu zesílení. Ta je zpočátku nastavena pro všechny kanály na *1*. Tímto menu je možné hodnotu *handles.gain* u každého kanálu změnit. Popupmenu se naplní seznamem kanálů následujícím příkazem.



Obrázek 5.2: Channel option menu

```
set(handles.popupmenu1, 'String', char(handles.channel_name((1:1:end),:)));
```

Funkce **popupmenu1_Callback** zpracovává akci, vykonanou objektem popupmenu. Jedná se o výběr prvku ze seznamu. Po výběru prvku (názvu kanálu) dojde k načtení jeho hodnoty *handles.gain*, která se zobrazí v editovatelném textovém poli. Příkaz **guidata** uloží a aktualizuje hodnoty handles.

```
function popupmenu1_Callback(hObject, eventdata, handles)
set(handles.edit4,'String',...
handles.gain(1,end+1-get(handles.popupmenu1, 'Value')));
guidata(hObject, handles);
```

Zesílení signálu je možné v intervalu $\langle 0; 10 \rangle$. Funkce **slider2_Callback** reaguje na akci vyvolanou tímto vertikálním posuvníkem. Hodnotu posuvníku je možné sledovat v editovatelném textovém poli, tato hodnota se pak ukládá do proměnné *handles.gain*.

```
set(handles.edit4,'String',num2str(get(handles.slider2,'Value')));
handles.gain(1,end+1-get(handles.popupmenu1, 'Value')) ...
= get(handles.slider2,'Value');
```

Hodnotu zesílení lze měnit také přes editovatelné textové pole. Slouží k tomu funkce **edit4_Callback**. Do proměnné *sliderValue2* je načtena hodnota z textového pole. U hodnoty se ověří, zda leží v intervalu minimální a maximální hodnoty posuvníku. Také je ošetřeno, aby textové pole nereagovalo na symboly. Pokud je v textovém poli skutečně číselná hodnota, která je větší jak 0 a menší jak 10, je na tuto hodnotu nastaven posuvník.

```
sliderValue2 = str2num(get(handles.edit4,'String'));

if (isempty(sliderValue2) || sliderValue2 < get(handles.slider2, 'Min'))
```

```

|| sliderValue2 > get(handles.slider2, 'Max'))
else

    set(handles.slider2, 'Value', str2num(get(handles.edit4, 'String')));
end

...
guidata(hObject, handles);

```

5.1.3 Tags menu

Tags menu je vytvořeno pro potřeby vkládání značek do záznamu. Sdružuje pět objektů typu `radiobutton`. Výběrem vhodné značky (tagu) se spustí režim anotace. Značky jsou dvojího typu. Značky *A*, *B*, *C*, *D* jsou značky nepárové. Značky *E*, *F*, *G* jsou značky párové. Menu umožňuje odstranění značek a vypnutí značkovacího režimu. Přepínání je možné provádět i pomocí tlačítek na klávesnici. Klávesy *F1* - *F7* jsou vyhrazeny pro přepínání typu značek. Klávesa *Delete* spustí režim odstraňování značek a klávesa *Escape* ukončí značkovací režim. Reakci na změnu tlačítka vyhodnocuje funkce `uipanel_Tags_SelectionChangeFcn`.

5.1.4 Ovládání pomocí kláves

Z důvodu zefektivnění režimu anotace je umožněno pro ovládání použít klávesnici. Stiskem kláves *F1* - *F7* dojde k vybírání typu značky. Klávesa *Escape* ukončuje režim anotace a klávesa *Delete* spouští režim odstraňování značek. Pro rychlejší prohlížení záznamu se využívá směrových, kurzorových šipek. Funkce, která kontroluje zda došlo ke stisknutí kláves, se nazývá `keyDownListener`.

```
set(hObject, 'KeyPressFcn', @keyDownListener)
```

Funkce `get(gcf, 'CurrentKey')` určí název stisklé klávesy. Na ukázce je předvedeno porovnání s názvem klávesy *rightarrow* tj. šipka doprava. V případě, že je tato klávesa stisklá, dojde k posunutí horizontálního posuvníku *handles.Sval* o dvojnásobek kroku.

```

function keyDownListener(hObject, eventdata, handles)
...
kkey =get(gcf, 'CurrentKey');
...
if(strcmp(kkey, 'rightarrow')==1)

```

```

handles.Sval=get(handles.slider1,'Value');
max=get(handles.slider1,'Max');
step=get(handles.slider1,'SliderStep');
if((handles.Sval+(step(1)*2))<=max)

    set(handles.slider1,'Value',handles.Sval+(step(1)*2));
end

if((handles.Sval+(step(1)*2))>max)
    set(handles.slider1,'Value',max);
end

..

```

5.2 Vykreslení

Pro vykreslení je zapotřebí načíst data způsobem, jak bylo vysvětleno v kapitole 4.2.1.

```

handles.hPSGLab_gui = getappdata(0, 'hPSGLab_gui');
handles.channels = getappdata(handles.hPSGLab_gui, 'channels');
handles.channel_name = getappdata(handles.hPSGLab_gui, 'channel_name');
handles.chena = getappdata(handles.hPSGLab_gui, 'chena');
handles.fsamp = getappdata(handles.hPSGLab_gui, 'fsamp');
handles.tag_pos = getappdata(handles.hPSGLab_gui, 'tag_pos');
handles.tag_type = getappdata(handles.hPSGLab_gui, 'tag_type');
handles.nsamp = getappdata(handles.hPSGLab_gui, 'nsamp') ;
handles.nch = getappdata(handles.hPSGLab_gui, 'nch');

...

if ( getappdata(handles.hPSGLab_gui, 'dataPreviewFlag') == 1 )
handles.ViewSet = getappdata(handles.hPSGLab_gui, 'dataPreviewSetting');

...

else
    ViewSet.Visible=ones(1,handles.nch);
    ViewSet.WinSize=5;
    handles.ViewSet=ViewSet;
end;

```

Data se načítají ve funkci **Data_preview_OpeningFcn**, která se vykonává při spuštění modulu. Nejprve je do proměnné *handles.hPSGLab_gui* načtená celá struktura, poté se do jednotlivých handles načtou data, která obsahují informaci o kanálech (channels), počtu vzorků (nsamp), vzorkovací frekvenci (fsamp) a pokud jsou v záznamu již přidány značky, tak se načtou i informace o značkách (tags). Pokud existuje nastavení pro tento záznam jsou načtené hodnoty handles.ViewSet, které obsahují informaci o velikosti okna nebo kanálech, které se nemají vykreslovat. Pokud nastavení neexistuje, použijí se implicitní hodnoty. Vykreslování pak zajišťuje funkce **Fig**.

5.2.1 Vykreslení záznamu

Délka vykresleného záznamu může být různá, a záznam se dá posunovat. Proměnná *LoB* (Lower boundary) obsahuje index prvního vzorku vykreslovaného záznamu. V proměnné *UpB* (Upper boundary) je uložen index posledního vykresleného vzorku.

```
if (handles.Sval) == 0
    LoB=1;
    UpB=handles.fsamp*handles.ViewSet.winsize;
else
    LoB=floor((handles.Sval*handles.fsamp*handles.ViewSet.WinSize));
    UpB=floor(LoB+handles.fsamp*handles.ViewSet.WinSize);
end
```

Proměnná *handles.Sval* obsahuje aktuální pozici posuvníku. Pokud je rovna 0, pak se vykresluje od prvního vzorku. Matlab indexuje od čísla 1. Hodnota posledního indexu je určena součinem vzorkovací frekvence (*handles.fsamp*) a velikostí okna (*handles.ViewSet.WinSize*). Velikost okna je určena v sekundách. Pokud se změní hodnota posuvníku, bude posunut o tuto hodnotu i vykreslený záznam.

Aby se signál vykreslil správně s ohledem na čas, je nutné vytvořit si časovou osu, která přiřadí jednotlivým vzorkům v rozmezí *LoB* - *UpB* přesný časový okamžik.

```
Taxis=[(LoB/handles.fsamp):(1/handles.fsamp):(UpB/handles.fsamp)];
Ts=length(Taxis);
```

Proměnná *Taxis* je vektor, hodnoty tohoto vektoru poslouží jako data pro osu *x*. Proměnná *Ts* obsahuje délku vektoru *Taxis*.

```

for i=1:handles.nch
    ...
if handles.ViewSet.Visible(1,end-i+1)==1

    plot(Taxis(1:Ts),(handles.channels((handles.nch-i+1),...
        (LoB:UpB))*handles.gain(1,i))+offset*(i-1),...
        'Color',[0 0      0],'HitTest','on','ButtonDownFcn',@DownButtonFunction,...
        'DisplayName',{num2str(i)});
    ...
end
end

```

Na ukázce kódu je znázorněno, jakým způsobem se vykreslují jednotlivé kanály. Na začátku je cyklus `for`. Cyklus se opakuje tolikrát, kolik kanálů (*handles.nch*) záznam obsahuje. Před samotným vykreslením je provedena kontrola, zda je daný signál nastavený, aby byl viditelný. Pro vykreslování je využita funkce **plot**.

Osa *x* je určena hodnotami z vektoru *Taxis(1:Ts)*, hodnoty na ose *y* pak vektorem hodnot *handles.channels(číslo_signálu,LoB:UpB)*. Jednotlivé kanály jsou vykreslovány od posledního. Hodnota, která odpovídá danému indexu je poté popřípadě přenásobena hodnotou zesílení (*handles.gain*) a nakonec se přičte násobek hodnoty *offset*. To má za následek nepřekrývání signálu a jeho posunutí na ose *y*. Mezi další vlastnosti funkce **plot** patří nastavení barvy, názvu a *callback* funkce.

Nezbytné je nastavit vlastnosti os. Osa *x* je vypnuta a časový údaj je prezentován pomocí statického textového pole. Časový údaj je převeden do formátu HH:MM:SS.

```

axis([LoB/handles.fsamp,UpB/handles.fsamp,...
-offset, offset*(handles.nch)]);
set(gca,'YTickLabel',{char(handles.channel_name((end:-1:1),:))});

```

Příkazem **axis** se nastavuje ohraničení os. Vlastnost *YTickLabel* přidává popisky, v tomto případě názvy kanálů v ose *y*.

5.2.2 Vykreslení značek

Pro vykreslování značek, které záznam již obsahuje, jsou důležité dvě struktury *handles.tag_pos* a *handles.tag_type*. *Handles.tag_type* je vektor a obsahuje číselné typy značek, *handles.tag_pos*

má pak na odpovídajícím indexu informaci o pozici značky. Pro vykreslování vložených značek je použita vlastní struktura a má i odlišný způsob vykreslení.

```
for i=1:length(handles.newTag.type(1,:))
    ...
    if(handles.newTag.type(1,i)>4)
        ...
    else
        plot([handles.newTag.pos(1,i)/handles.fsamp...
            handles.newTag.pos(1,i)/handles.fsamp],...
            [(offset)*(handles.newTag.type(2,i)-1.5),...
            offset*(handles.newTag.type(2,i)-0.5)],...
            'HitTest','on','ButtonDownFcn',...
            @TagsButtonFunction,'DisplayName',num2str(i));

        handles.Ttext(1,i)=text(handles.newTag.pos(1,i)/handles.fsamp,...
            offset*(handles.newTag.type(2,i)-1.5)+100,...
            char(handles.newTag.type(3,i)),'Color',linecolor);
        ...
    end
end
```

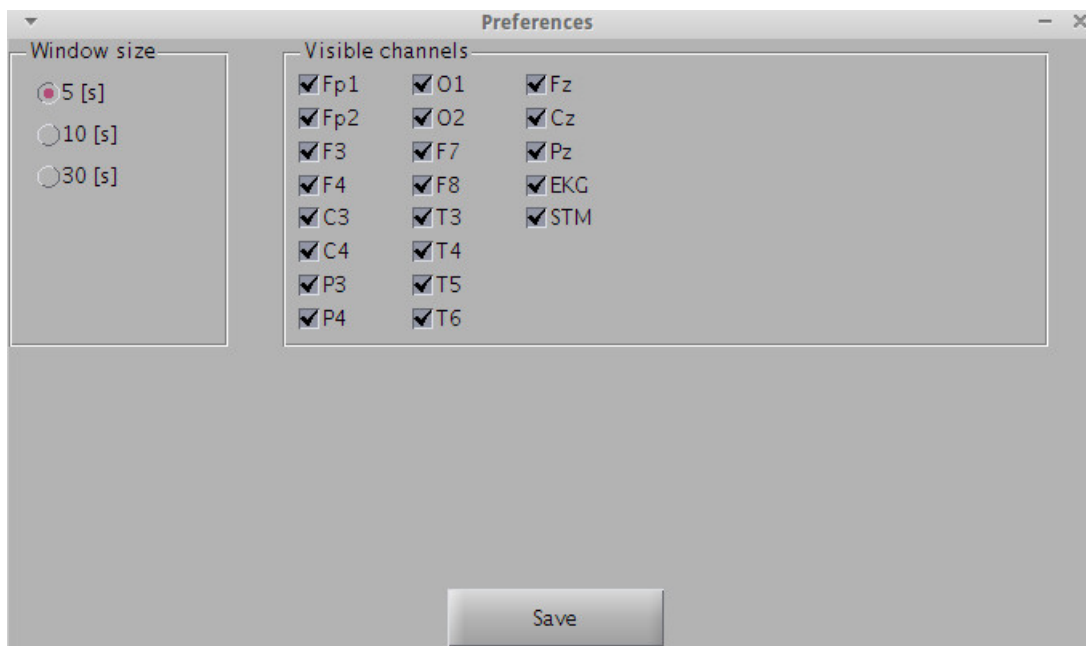
Cyklus `for` se provede tolikrát, kolik je záznamů v *handles.newTag_type*. Značka (tag) se vykreslí vertikálně v jednom místě na ose *x*. Ke značce je také přidán textový popis, který informuje o typu značky. Pro každý typ značky je přiřazena specifická barva. Značky mají vlastní *callback* funkci podle toho, zda se jedná o značku párovou (**TagsButtonFunctionPair**) nebo nepárovou (**TagsButtonFunction**).

5.3 Nastavení - Preferences

Z důvodu úspory místa a lepší přehlednosti GUI, je vytvořené oddělené okno s dalším nastavením. Okno se vytvoří stisknutím tlačítka Preferences. Smysl tohoto okna je možnost měnit nastavení. Zejména délku vykresleného záznamu nebo vypnutí zobrazování vybraných kanálů. Toto nastavení je následně uloženo.

5.3.1 Window size menu

Podobně jako v kapitole 5.1.3, i zde jsou objekty `radiobutton`, které sdružuje objekt typu `uibuttongroup`. Možnosti délky záznamu jsou předdefinované. V nabídce je na výběr 5, 10



Obrázek 5.3: Preferences

a 30 sekund nebo 1, 5 a 10 minut. V případě, že je načten záznam kratší než jsou uvedené možnosti, zobrazí se pouze volby, jejichž délka je kratší než záznam samotný viz. obr. 5.3.

5.3.2 Visible channel menu

Menu obsahuje dynamicky vytvořené objekty typu checkbox. V závislosti na počtu kanálů se vytvoří příslušný počet objektů. Rozmístění je navrženo tak, aby se do každého sloupce vešlo 8 objektů. Změna označení objektu způsobí změnu zobrazení kanálu v záznamu.

```
for i=1:ceil(handles.nch/8)
    for j=1:8
        ...
        handles.cb(count)= uicontrol('units','normalized','Style','checkbox',...
            'String',char(handles.channel_name(count,:)),...
            'pos',[0.01+0.15*(i-1) 1-0.12*j .2 .1],'HandleVisibility','on',...
            'Parent',h2,'Callback',@CheckboxFunction);
        ...
    end
end
```

5.3.3 Uložení nastavení

Uložení nastavení a vykreslení záznamu nastane po stisknutí tlačítka s popisem *Save. Callback* funkce tlačítka se nazývá **closeCallback**. Pro uložení dat se využívá funkce **setappdata** viz. kapitola 4.2.1. Proměnná *dataPreviewFlag* se nastaví na *1*, to indikuje zápis dat. Po uložení dat se volá funkce **Data_preview**, pro nové vykreslení záznamu a následně se příkazem **close** zavírá okno *Preferences*.

```
hf=findobj('Name','Preferences');

setappdata(handles.hPSGLab_gui, 'dataPreviewFlag',1);
setappdata(handles.hPSGLab_gui, 'dataNewTag',1);
setappdata(handles.hPSGLab_gui, 'ViewSet', handles.ViewSet);
setappdata(handles.hPSGLab_gui,'dataPreviewSetting',handles.ViewSet);

Data_preview;

close(hf);
```

5.4 Anotace záznamu

Anotací záznamu je myšleno vkládat, popřípadě odstraňovat značky. Značky jsou dvojího typu. Jednak jsou značky nepárové. Nepárové se nazývají, protože stačí pouze jedna souřadnice na ose *x*, k jejich označení. Tyto značky oznamují výskyt události, to může být podání prášků pacientovi, nebo usnutí pacienta. Párová značka označuje začátek a konec události. Například charakteristická věténka, která jsou typickým projevem epilepsie.

5.4.1 Struktura dat

K ukládání vložených značek bylo potřeba navrhnout vhodnou strukturu pro ukládání dat tak, aby bylo možné přidanou značku jednoznačně spojit s kanálem, ve kterém došlo k výběru. Došlo k vytvoření struktury *newTag*, ta obsahuje informaci o typu značky *newTag.type* a informaci o pozici značky *newTag.pos* viz. obr. 5.4. U struktury *newTag.type* první řádek obsahuje informaci o typu značky, ta odpovídá hodnotě *handles.switchO*. Druhý řádek obsahuje číslo kanálu, to je potřebná informace pro vykreslení značky. Ve třetím řádku je uložen název značky ve formátu string. Struktura *handles.pos* obsahuje informaci o začátku značky na prvním řádku. Na druhém řádku se nachází informace o konci značky. Druhý řádek se používá pouze v případě párové značky.

newTag.type

	1	2	3		n
Typ značky					
Číslo kanálu					
Název značky					

newTag.pos

	1	2	3		n
Pozice začátek					
Pozice konec					

Obrázek 5.4: Struktura ukládání parametrů značky

5.4.2 Přidání značky

Možnost anotace záznamu, tzn. přidávat značky do záznamu se spustí ve chvíli, kdy je vybrán typ značky. To se provede buď označením příslušného grafického objektu nebo pomocí kláves *F1 - F7*. Výběrem jedné z těchto značek se do proměnné *handles.switch0* uloží kladné číslo v rozmezí 1 až 7 podle toho, jaký typ značky byl vybrán. Do proměnné *handles.tagname* se uloží název značky.

V kapitole 5.2.1 bylo zmíněno, že každý vykreslený kanál má *callback* funkci. To znamená, že při stisknutí levého tlačítka myši na daný průběh se zavolá *callback* funkce, v tomto případě s názvem **DownButtonFunction**.

```
if(handles.switch0>0)

    if(handles.switch0<5)
        ...
    else

        channum = str2num(get(hObject,'DisplayName'));
        p=get(gca,'CurrentPoint');

        set(hObject,'color','blue');
        pause(0.1);
        set(hObject,'color',[0 0 0]);

        x(1)=p(1,1);

        while (waitforbuttonpress==1)
```

```

        waitforbuttonpress
    end
    ...
    p=get(gca,'CurrentPoint');
    x(2)=p(1,1);
    ...
    handles.newTag.type(1,length(handles.newTag.type(1,:))+1)=handles.switch0;
    handles.newTag.type(2,length(handles.newTag.type(2,:))=channum;
    handles.newTag.type(3,length(handles.newTag.type(3,:))=handles.tagname;
    handles.newTag.pos(1,length(handles.newTag.pos(1,:))+1)=x(1)*handles.fsamp;
    handles.newTag.pos(2,length(handles.newTag.pos(2,:))=x(2)*handles.fsamp;

end
end

```

Na začátku funkce dojde ke kontrole hodnoty *handles.switch0*. Pokud není větší jak 0, neprovádí se žádná akce, a tak stisk tlačítka myši v záznamu nemá žádný vliv. Druhá podmínka rozlišuje mezi značkou nepárovou a značkou párovou. Do proměnné *channum* se načte číslo kanálu. Příkazem **get(gca,'CurrentPoint')** je získána poloha, kde došlo k označení. Následuje grafické potvrzení, že byl daný kanál označen. Toto potvrzení se projeví v podobě probliknutí daného kanálu v modré barvě. Následuje cyklus while, který čeká označení druhé polohy značky, tzn. druhé stisknutí tlačítka myši. To má na starosti funkce **waitforbuttonpress**. Označení je opět potvrzeno graficky. Poté už pouze dojde k uložení dat do vytvořené struktury a zavolání funkce **fig** pro vykreslení.

5.4.3 Odebrání značky

Možnost odebírat značky se aktivuje označením tlačítka Delete, nebo stiskem příslušné klávesy *Delete*. Hodnota *handles.switch0* se nastaví na -1. K odebírání značek slouží funkce **TagsButtonFunctionPair** a **TagsButtonFunction**.

```

function TagsButtonFunction(hObject,eventdata)
    ...
    index=str2num(get(hObject,'DisplayName'));

    if(handles.switch0==-1)

```

```

delete(hObject);

delete(handles.Ttext(1,index));

handles.newTag.type(:,index)=[];
handles.newTag.pos(:,index)=[];
end
...

```

Ukázka obsahuje princip odstranění značky. Kromě odstranění značky musí dojít i k odstranění textu, který slouží jako popis značky. Následně je odstraněn celý sloupec ve struktuře, ve které jsou značky uchovány. Princip je obdobný i pro odstranění značek párových.

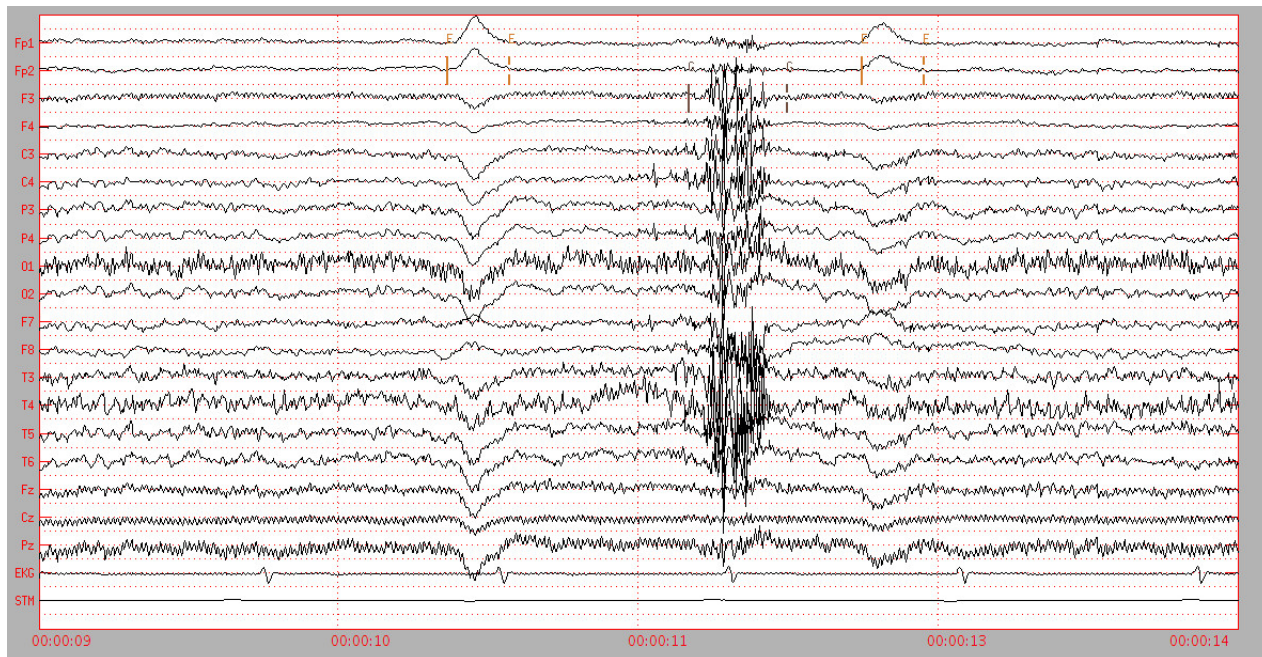
Kapitola 6

Ověření na reálných datech

V průběhu tvorby anotátoru se pracovalo s reálnými daty. Jednalo se o krátké záznamy naměřené skupinou Biodat. Pomocí těchto záznamů došlo k odstranění chyb a vylepšení funkčnosti. Poté se mohly otestovat dlouhodobé záznamy.

6.1 Záznamy skupiny Biodat

Záznamy naměřené skupinou Biodat byly v průběhu tvorby anotátoru velice užitečné. Datovou strukturou se nijak neliší od ostatních záznamů. Tyto záznamy obsahovaly 21 kanálů. Jejich výhodou je především jejich délka. Ta nepřesahuje jednu minutu, proto nebyl toolbox

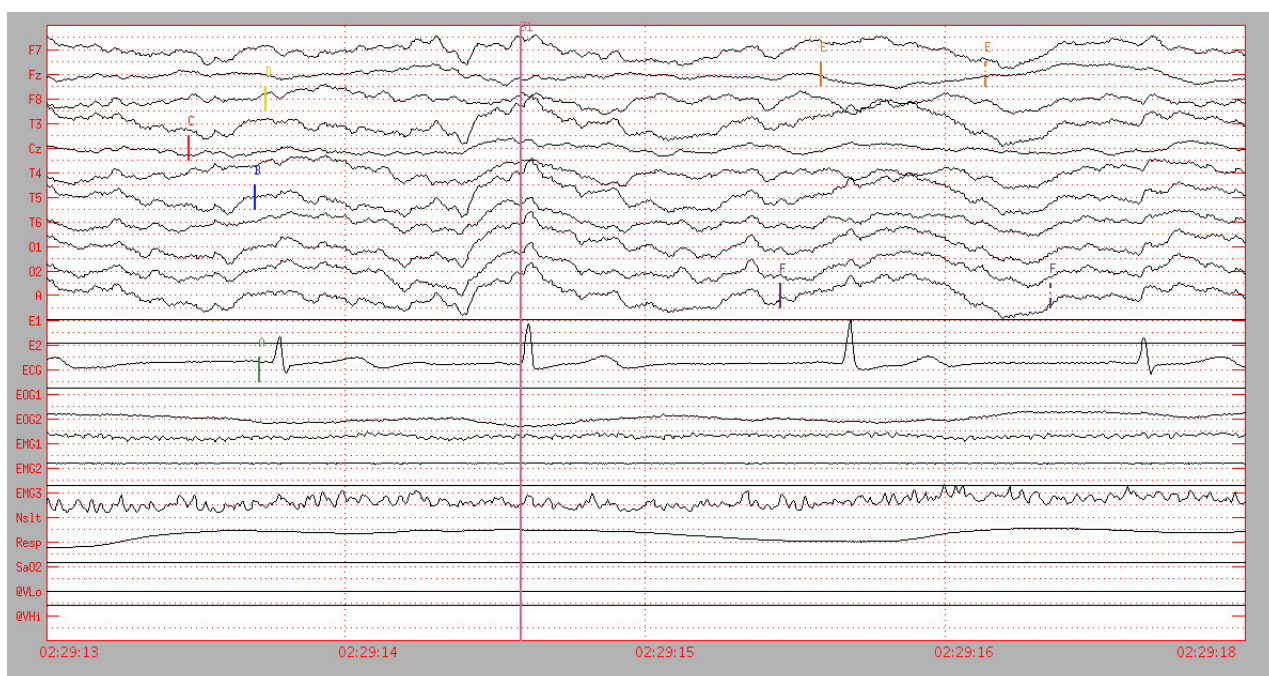


Obrázek 6.1: Testování anotace

zatížen načítáním velkým objemem dat. Na obrázku 6.1 jsou vidět přidané párové značky *E* a *G*. Značka *E* ohraničuje artefakt mrknutí oka, značka *G* pak artefakt způsobený svalovou aktivitou.

6.2 Dlouhodobé záznamy

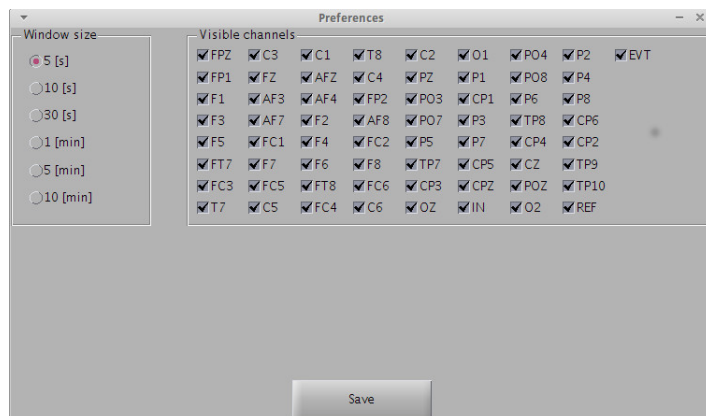
Dlouhodobé záznamy, které se použily k ověření funkčnosti, jsou ze spánkové laboratoře psychiatrického centra v Bohnicích a z nemocnice Na Bulovce. Jsou téměř osmihodinové, což představuje přibližně 7 miliónů vzorků pro každý kanál. Tyto záznamy obsahují 24 kanálů. Na obrázku 6.2 je vidět vykreslení značky, kterou záznam již obsahoval. Je to značka, která je vykreslena napříč všemi signály a její značka je 36. Do záznamu byly pokusně vloženy jak párové tak nepárové značky.



Obrázek 6.2: Testování anotace dlouhodobého záznamu

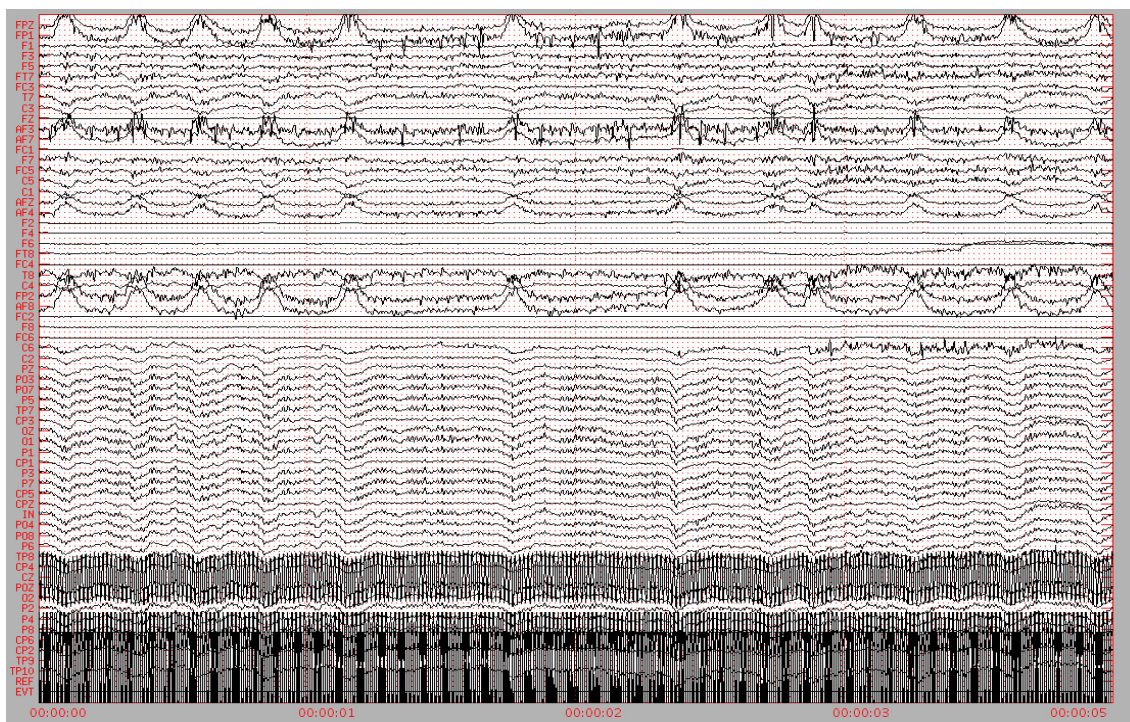
6.3 Ostatní záznamy

Do této kategorie můžeme zařadit záznamy různých délek z různých pracovišť. Jedním z těchto záznamů je experimentální záznam s 65 kanály. Na obrázku 6.3 je patrné, že u tohoto záznamu lze vybírat z větší možnosti rozsahů velikosti okna. Především bylo otestováno, že dynamické vytváření zaškrtačích políček (checkbox) umožňuje vykreslení velkého množství těchto prvků.



Obrázek 6.3: Ukázka okna nastavení pro záznam s 65 kanály

Na obrázku 6.4 je zřejmé, že pro kvalitní zobrazení takto vysokého počtu kanálů je třeba monitor s vysokým rozlišením.



Kapitola 7

Závěr

Cílem této práce bylo navrhnout a implementovat anotátor dlouhodobých záznamů. Anotátor je součástí grafického prostředí PSGLab. Modul umožňuje zobrazení a anotaci dlouhodobých, vícekanálových EEG záznamů. Umožňuje vybrat si vhodnou velikost vykresleného okna. U jednotlivých kanálů je možné měnit jejich amplitudu nebo jejich vykreslování úplně vypnout. V případě, že záznam již obsahuje značky vytvořené expertem, je tento modul schopný tyto značky vykreslit. Pro uložení nových značek vytvořených tímto anotátorem je navržena vhodná datová struktura. Pro uložení nastavení se používá struktura, kterou navrhl a implementoval Petr Štěpánek.

Velký důraz se kladl na to, aby grafické prostředí bylo intuitivní a obsahovalo co nejméně ovládacích prvků. Ověření funkce anotátoru probíhalo po celou dobu jeho vývoje. Testování na dlouhodobých záznamech mělo přínos v optimalizaci vykreslování záznamu. Přesto po určitém sledu událostí může dojít k výskytu situací, které jsou neočekávané. V průběhu testování byla snaha objevit co nejvíce těchto situací, aby nedošlo k výskytu chyb popř. pádu celého modulu. K objevení těchto nečekaných stavů je zapotřebí ještě mnoho dalších experimentů.

Vývoj touto prací nekončí. Do budoucna se počítá s rozšiřováním funkcí PSGLabu včetně anotátoru. Nabízí se možnost vytvoření databáze grafoelementů, implementace inkrementálního způsobu učení, vykreslování výsledků segmentace signálu nebo vykreslení výsledků shlukové analýzy.

Literatura

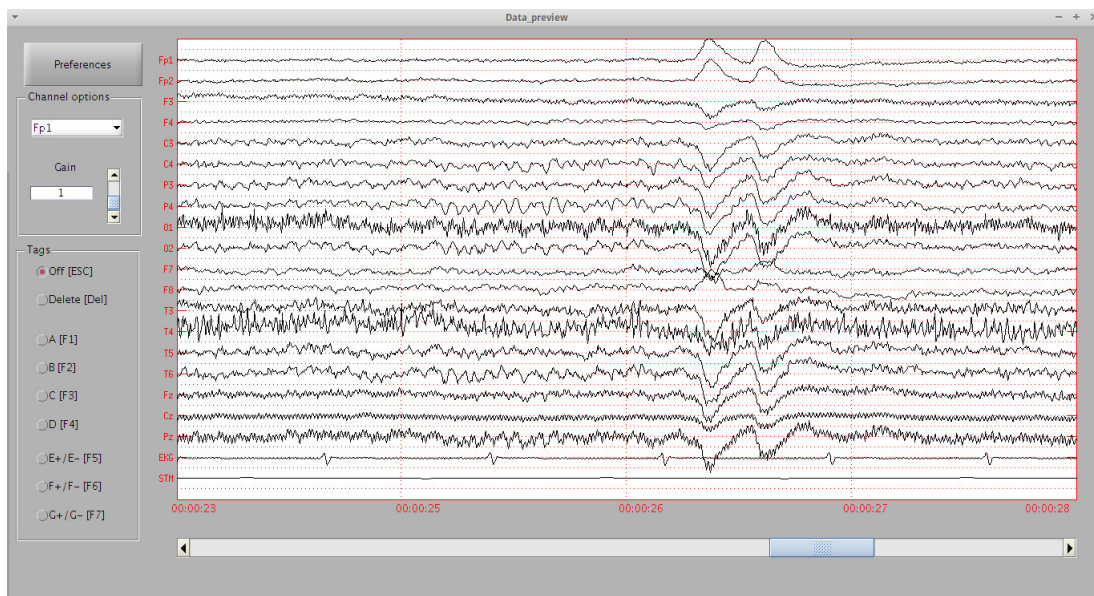
- [1] DELORME Arnaud a Makeig SCOTT. “EEGLAB”. In: *Journal of Neuroscience Methods* vol. 134.issue 1 (2004), pp. 9–21. URL: <<http://linkinghub.elsevier.com/retrieve/pii/S0165027003003479>>.
- [2] KRAJČA Vladimír a Jitka MOHYLOVÁ. *Číslíkové zpracování neurofyzilogických signálů*. 1. vyd. České vysoké učení technické, 2011.
- [3] LOPES DA SILVA Fernando a Ernst NIEDERMEYER. *Electroencephalography. Basic principles, clinical applications, and related fields*. 5th ed. Lippincott Williams, 2005.
- [4] PLONSEY Robert a Jaakko MALMIVUO. *Bioelectromagnetism. Principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, 1994.
- [5] RIEGER Josef. *Zpracování dlouhodobých EEG záznamů*. Diplomová práce. 2004.
- [6] SANEI Saeid a Jonathon CHAMBERS. *EEG signal processing*. John Wiley, 2007.
- [7] RIEGER J. a L. LHOTSKÁ a V. KRAJČA. “Long-Term EEG Records Analysis”. In: *Advances in Electrical and Electronic Engineering* (2005).
- [8] STERN John M. a Jerome ENGEL. *Atlas of EEG patterns*. Lippincott Williams, 2005.
- [9] ZAPLATÍLEK Karel a Bohuslav DOŇAR. *MATLAB*. 1. vyd. BEN, 2004.
- [10] *MATLAB Documentation. Uicontrol*. URL: <<http://www.mathworks.com/help/matlab/ref/uicontrol.html>> (visited on 05/15/2013).
- [11] GERLA Václav et al. “PSGLab Matlab toolbox for polysomnographic data processing”. In: *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine* (2010), pp. 1–4. URL: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5687737>>.

Příloha A

Uživatelská příručka

Ke spuštění grafického rozhraní PSGLabu dojde spuštěním **PSGLab_gui.m**. K načtení dat se mohou využít data ve složce *Test_data*. Po načtení se zpřístupní tlačítko *View Data*. Stisknutím dojde ke vzniku nového okna a vykreslení záznamu viz. obr. A.1. K prohlížení záznamu slouží horizontální posuvník. Ten se ovládá buď tažením myši a nebo kurzorovými šipkami.

V tomto okně má uživatel možnost upravovat velikost amplitud vykreslených kanálů. K tomu slouží *Channel options* menu. Výběrem v rolovacím menu si uživatel vybere kanál. V textovém poli se zobrazí aktuální hodnota zesílení. Uživatel tuto hodnotu upraví buď změnou hodnoty v textovém poli nebo pohybem vertikálního posuvníku.



Obrázek A.1: Hlavní okno anotátoru

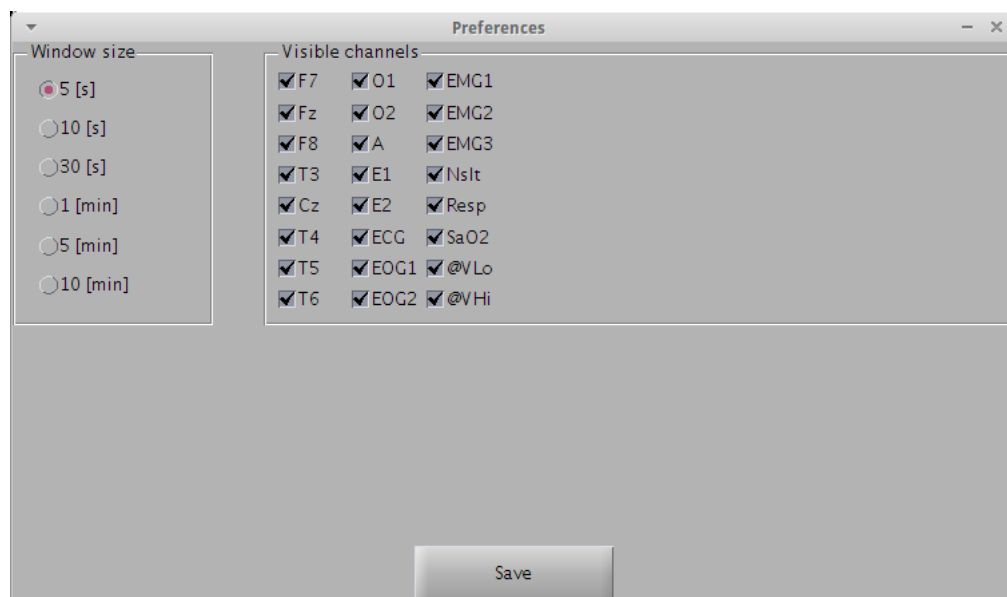
V případě, že uživatel chce do záznamu vložit značku, stačí si vybrat jednu z nabízených v

menu *Tags*. Popis v závorce uvádí název kláves, kterými se může provést výběr. Není nutné pro tuto činnost používat myš. Značky *A*, *B*, *C*, *D* jsou nepárové, značky *E*, *F*, *G* jsou párové. Zvolením typu značky se spustí režim anotace. Jednou značkou lze označit pouze jeden kanál. Pro případ nepárové značky stačí kliknout v daném místě na vykreslený kanál. Pro potvrzení výběru změni vykreslený kanál na krátký časový úsek svou barvu, krátce na to dojde k vykreslení značky. Pokud chce uživatel vložit párovou značku, je postup podobný. Prvním kliknutím do vybraného kanálu dojde pouze k probliknutí jako potvrzení výběru. Zatím se nic nevykresluje, program čeká na druhé stisknutí tlačítka myši. Po druhém stisknutí opět dojde k probliknutí zvoleného kanálu a následuje vykreslení značky. V případě, že vzdálenost mezi začátkem a koncem značky je větší než zobrazený úsek je nutné použít k posunu záznamu kurzorové šipky. To z důvodu čekání na druhý stisk tlačítka myši.

Pokud dojde k omylu nebo špatnému vložení značky do záznamu, lze značky snadno odstranit. Výběrem možnosti *Delete* nebo klávesou *Delete* se spustí režim odstraňování značek. K odstranění dojde kliknutím na danou značku.

K ukončení režimu anotace nebo režimu odstraňování značek dojde vybráním možnosti *Off* nebo stiskem klávesy *Escape*.

Pokud uživateli nevyhovuje délka vykresleného záznamu, má možnost toto nastavení změnit. Stiskem tlačítka *Preferences* dojde k vytvoření nového okna viz. obr. A.2. V tomto okně si



Obrázek A.2: Nastavení vlastností záznamu

uživatel může vybrat jednu z nabízených možností délky okna. Další možností v tomto okně

je vypnutí vykreslování vybraných kanálů. Uživatel pouze kliknutím na příslušné políčko rozhodne o vykreslení libovolného kanálu. Aby se změny projevily, musí uživatel tyto změny potvrdit stisknutím tlačítka *Save*. Tyto hodnoty nastavení se ukládají.

Příloha B

Obsah přiloženého CD

Přiložené CD obsahuje PSGLab Anotator zabalený v archivu zip. Archiv obsahuje testovací data. Dlouhodobé záznamy z lékařských pracovišť není možné zveřejňovat, protože obsahují osobní informace pacientů. Ve složce Text se nachází samotný text bakalářské práce.

```
.  
|--- PSGlab Anotator 1.0.zip  
|--- Text  
    |--- bp_Prihoda.pdf
```