

Diplomová práce



České
vysoké
učení technické
v Praze

F2

Fakulta strojní
Ústav přístrojové a řídicí techniky

Softwarový nástroj pro předzpracování EEG signálů z MUSE

Bc. Petr Maršík

Září 2020

Vedoucí práce: Ing. Cyril Oswald, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Maršík** Jméno: **Petr** Osobní číslo: **395187**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Automatizační a přístrojová technika**
Specializace: **Automatizace a průmyslová informatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Softwarový nástroj pro předzpracování EEG signálů z MUSE

Název diplomové práce anglicky:

Software tool for preprocessing of EEG signals measured by MUSE

Pokyny pro vypracování:

1. Proveďte rešerši algoritmů vhodných pro základní předzpracování EEG dat.
2. Navrhněte a vytvořte softwarový nástroj pro měření a předzpracování EEG signálů měřených čelenkou MUSE s následujícími parametry:
 - a) programovací jazyk Python s využitím grafického uživatelského rozhraní
 - b) on-line záznam EEG signálů z čelenky MUSE
 - c) off-line vizualizace naměřených dat a off-line základní předzpracování dat pomocí zvolených algoritmů
 - d) export dat do Matlabu, CSV, Excel

Seznam doporučené literatury:

1. INTERAXON INC. Muse™ The Brain Sensing Headband - Technology Enhanced Meditation [online]. [vid. 2019-1226]. Dostupné z: <https://choosemuse.com/muse>
2. GKRAJČA, Vladimír a Jitka MOHYLOVÁ. Číslíkové zpracování neurofyzilogických signálů. V Praze: České vysoké učení technické, 2011. ISBN 978-80-01-04721-7.
3. COHEN, Mike X. Analyzing Neural Time Series Data: Theory and Practice [online]. Cambridge, Massachusetts: The MIT Press, 2014. Issues in Clinical and Cognitive Neuropsychology. ISBN 978-0-262-01987-3. Dostupné z: <http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=689432&lang=cs&site=ehost-live>

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Cyril Oswald, Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **30.04.2020** Termín odevzdání diplomové práce: **27.08.2020**

Platnost zadání diplomové práce: _____

Ing. Cyril Oswald, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Rád bych poděkoval panu Ing. Cyrilu Oswaldovi, Ph.D. za trpělivé vedení mé práce. Dále bych rád poděkoval svým rodičům za jejich duševní i materiální podporu během studia.

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

.....

Abstrakt / Abstract

Cílem této práce je vytvořit nástroj s uživatelským prostředím v programovacím jazyce Python, který umožňuje online sledování průběhů EEG signálů z nositelného EEG zařízení Interaxon Muse. Zároveň je zde rozebíráno offline předzpracování pro tento typ zařízení a aplikováno pomocí diskutovaných metod. Nakonec je analyzován výsledek předzpracování.

Klíčová slova: EEG, preprocessing, signal analysis, Interaxon Muse, Python, PyQt5, Muse-lsl, MNE-Python

The aim of this thesis is to develop a tool with a user interface in the programming language Python which allows users to view EEG signals from the wearable EEG device Interaxon Muse. Also, a research into offline preprocessing methods is conducted for this class of device and these methods are applied on the data, results of which are analyzed in the end.

Keywords: EEG, předzpracování, analýza signálu, Interaxon Muse, Python, PyQt5, Muse-lsl, MNE-Python

/ Obsah

1 Úvod	1
2 Předzpracování dat	3
2.1 Předzpracování a nastavení přístroje	5
2.2 Časová, frekvenční analýza a grafy	6
2.3 Filtrování	8
2.4 Segmentace	10
2.5 Referencování	11
2.6 Interpolace elektrod	11
2.7 Odstraňování trialů	11
2.8 Analýza nezávislých kompo- nent (ICA)	13
2.8.1 FastICA	18
2.9 Analýza hlavních kompo- nent (PCA)	19
3 Nástroj pro předzpracování EEG signálů	22
3.1 Muse	23
3.2 Python a knihovny	26
3.2.1 PyQt	26
3.2.2 PyQtGraph	27
3.2.3 MNE-Python	27
3.2.4 EEG-Notebooks	27
3.3 Komunikace přes LSL a OSC .	28
3.3.1 LSL	28
3.3.2 OSC	29
3.4 Popis programu	30
3.4.1 Záložka Online	31
3.4.2 Záložka Offline	32
3.5 Implementace GUI a ostat- ních funkcí	33
3.5.1 Komunikace	33
3.5.2 Formáty datových sou- borů	35
3.5.3 GUI	36
3.6 Implementace předzpracování .	38
4 Závěr	46
Literatura	48
A Zkratky	52
B Obsah CD	53

Tabulky / Obrázky

2.1. Technická specifikace EEG signálů	3
2.2. Frekvenční pásma EEG vln	4
3.1. Alternativní zařízení k Muse ..	22
3.2. Specifikace OSC pro Muse (Monitor)	29
3.3. Srovnání BT modulů.....	34
2.1. Muse: P300 na TP9	4
2.2. Diagram umístění elektrod v systému 10-20	5
2.3. Graf EEG časové řady ze 63 kanálů ve fázi odpočinku	6
2.4. Výkonová spektrální hustota EEG signálu	8
2.5. Výkonová spektrální hustota EEG signálu	9
2.6. Hranový artefakt a segmentace EEG (ERP)	10
2.7. Jednotlivý trial EOG průběhu obsahující mrknutí.....	12
2.8. Sdružená hustota pravděpodobnosti dvou gaussovských proměnných	15
2.9. Ilustrace metody PCA na případu dvou proměnných (2D).....	19
2.10. Graf kumulativního součtu singulárních hodnot	21
3.1. Popis čelenky Interaxon Muse MU-02 (2016)	23
3.2. Umístění senzorů na hlavě u Muse v systému 10-20	24
3.3. Přídavná elektroda (AUX) připojená k Muse.....	25
3.4. Srovnání GUI MNE LAB a EEGLAB	27
3.5. Architektura LSL systému	28
3.6. Záložka Online nástroje.....	31
3.7. Záložka Offline nástroje.....	32
3.8. CSV formát Muse LSL.....	35
3.9. PyQtGraph třídy GUI	36
3.10. Graf surových dat	38
3.11. Graf PSD surových dat	39
3.12. Porovnání mezi surovými a filtrovanými daty	41
3.13. Graf časové řady segmentovaných dat	42
3.14. Souhrnný graf EOG aktivity na EEG elektrodách	43
3.15. Časový průběh nezávislých komponent.....	43
3.16. Graf rozdělení komponent na skalpu.....	44

3.17.	Graf překrytí před a po ICA ..	44
3.18.	Graf porovnání před a po ICA .	45
3.19.	Graf skóre souvisejících s de- tekovanými komponenty	45

Kapitola 1

Úvod

V posledních letech se na trhu začala objevovat elektronická zařízení pracující na známých principech přístrojů používaných ve zdravotnictví, která jsou cenově dostupná pro širokou veřejnost a snadno nositelná, tzv. „Wearables“. Jedním z těchto zařízení, které aplikuje technologii elektroencefalografie (EEG), je čelenka Muse od společnosti Interaxon.

Elektroencefalografie (EEG) je metoda, která se používá k záznamu elektrických signálů aktivity mozku. EEG signál vzniká činností mozkových neuronů [1]. V případě použití čelenky Muse se jedná o neinvazivní metodu skalpové EEG, která měří signály pomocí senzorů umístěných na povrchu hlavy. Toto měření je prováděno obvykle pomocí mnoha elektrod umístěných na místech pomocí vybraných standardů. Skalповé měření EEG je méně přesné než invazivní měření EEG na povrchu mozku, protože nedetekuje samostatné vzruchy a aktivitu mozku, ale jejich sekundární projevy. Dochází tak k vzájemnému ovlivnění signálů. Existují však metody zpětného zjištění zdroje EEG signálu [2]. EEG je v klinické praxi rozšířenou metodou, která se používá při léčbě chorob s fyzickými i duševními projevy. Týká se to např. epilepsie nebo poruchy spánku, ale také výzkumu např. při ověřování, zda mají migrény vliv na zvlnění EEG signálu [3].

Častý způsob měření EEG u lékaře pomocí profesionálního přístroje je měření v klidu bez vnějších podnětů a výsledkem je záznam spontánní elektrické aktivity mozku [1]. Další způsob se zaměřuje na tzv. evokované potenciály (EP) resp. kognitivní evokované potenciály (ERP), které jsou důsledkem vnějších smyslových podnětů resp. identifikovatelných podnětů či událostí. Podle druhu podnětu jsou EP zrakové, sluchové nebo elektricky stimulované [4–5]. K EEG záznamu v případě měření EP (ERP) je vložen další, který slouží k označení pozic evokovaných stimulů. Výhodou EP (ERP) je vyrušení šumu v EEG signálu [6].

Jedním z hlavních problémů v přístrojové elektroencefalografii je zajištění kvality signálu. Předzpracování dat se využívá z důvodu dalšího zpracování tak, aby signál na výstupu z procesu předzpracování bylo možné snadněji podrobit analýze a interpretovat výsledek. Na zhoršení kvality signálu se podílí mnoho faktorů, ať už z vnějšího prostředí nebo ty, které vznikají přímo při měření EEG na subjektu měření. Tyto faktory se souhrnně nazývají artefakty a jejich odstranění nebo potlačení je hlavním úkolem předzpracování.

Další problém je technický, a tím je přenos dat ze zařízení měřící EEG signál - čelenka Muse. V případě této práce se jedná o komerční produkt cílený pro širokou veřejnost, a jeho hlavní předností je cenová dostupnost. Jeho hlavní zaměření je však jako fitness pomůcka a slouží jako prostředek ke zlepšení kvality meditace [7]. Výrobce tento konkrétní model už nevyrobí a podpora pro vývojáře byla ze strany výrobce ukončena před započítím práce na této závěrečné práci. Existuje však několik funkčních aplikací z doby před ukončením podpory. Další možností je volně dostupná knihovna muse-lsl od komunity vývojářů, která je napsaná v programovacím jazyku Python, a

která tento problém obchází [8]. V Pythonu je také napsaná knihovna MNE-Python [9]. Tato knihovna obsahuje nástroje potřebné k celkové analýze EEG signálu a je tak alternativou k Matlab toolboxům jako např. Fieldtrip toolbox nebo EEGLab.

Cílem práce je vytvořit softwarový nástroj, v návaznosti na předem zmíněné knihovny napsaném v programovacím jazyce Python, který bude umožňovat online záznam EEG signálu z čelenky Muse a také jeho uložení do souboru. Z uloženého záznamu pak bude možné data offline vizualizovat a předzpracovat pomocí vybraných algoritmů. Tento nástroj bude mít svoje grafické uživatelské prostředí (GUI), ze kterého bude ovládán. Tento nástroj bude dále usnadňovat použití zařízení Muse pro další projekty a výuku. Zároveň bude sloužit jako příklad aplikace, která používá protokol Lab Streaming Layer (LSL). Protokol LSL, který je používán řadou zařízení aplikujících EEG a které jsou podobné Muse, funguje jako systém pro jednotný sběr měření časových řad při výzkumných experimentech a umožňuje časovou synchronizaci a přístup k datům téměř v reálném čase [10].

Jako jeho další využití si dokáží představit v experimentech s EEG, ať už při experimentu sledování únavy řidiče v laboratorních podmínkách na simulátoru, nebo při experimentech s řadou EP (ERP) testů, jako např. N100 nebo P300.

Kapitola 2

Předzpracování dat

Termín předzpracování znamená transformaci nebo reorganizaci dat mezi procesem sběru surových dat a jejich následnou analýzou, přičemž cílem předzpracování je v obecném slova smyslu vybrat užitečné složky signálu. Signál EEG je signál s velmi nízkou amplitudou a je tak často významně ovlivněn šumem. Měření EEG pomocí povrchových elektrod navíc není příliš odolné vůči rušení a lebka, která do značné míry funguje jako filtr typu dolní propust, zapříčiňuje prostorový rozptyl signálů. [2, 6, 11–12].

Některé kroky předzpracování data nemění a pouze je organizují do skupin z celkového průběhu do epoch (segmentů), což umožňuje pozdější analýzu nad původními daty. Další kroky jsou schopny odstranit tzv. artefakty. Mezi biologické artefakty patří signály jako jsou tlukot srdce, tedy signál EKG, signál od svalových událostí jako pohyb očí či mrkání (EOG) nebo sevření čelistí či pohyb hlavy (EMG). Mezi technické artefakty pak patří síťový brum, impulsní rušení nebo šum od elektronických obvodů. Tuto kroky jsou schopny odstranit artefakty bez toho, aniž by se měnily jinak čistá data, např. pomocí odstraňování chybných signálů od elektrod nebo po segmentaci do epoch odstraňováním epoch s artefakty. Hlavním problémem předzpracování dat je, že neexistuje jednotný postup na všechna měření a závisí na použitém měřicím zařízení i návrhu experimentů. Ovšem obecně platí, že je velmi obtížné dostat z nekvalitních dat i sofistikovanými metodami data kvalitní, přičemž u signálu EEG existuje řada vodítek při vizuální kontrole signálu, jak odhadnout dostatečně kvalitní data [2, 6].

Signál EEG je nestacionární, nelineární a zašuměný [13] a lze ho považovat za realizaci stochastického procesu. Dále se dá říct, že se jedná o ergodický proces a v případě krátkých časových intervalů (epoch) je možné o něm mluvit vzhledem k celému průběhu realizace jako o kvazistacionárním signálu [2].

Vlastnost	Hodnota
Počet kanálů	Obvykle 8 – 256 (4 + 1 pro Muse)
Amplitudový rozsah	2 – 100 μV (až 300 μV při epileptických záchvatech)
Frekvenční rozsah	DC - 100 Hz , nejvíc výkonu mezi 0.5 – 30 Hz

Tabulka 2.1. Technická specifikace EEG signálů, převzato z [2].

V tabulce 2.1 jsou uvedena technická specifika EEG signálu, včetně rozsahu amplitud signálu, pohybující se v řádu nízkých stovek μV . Signál EEG se podle jeho frekvenčních pásem dělí na mnoho typů aktivit, přičemž 5 základních se označuje řeckými písmeny alfa až gama. V klinické praxi se často diagnostikují poruchy mozku pomocí vizuální kontroly EEG záznamu a spojují se právě s projevy v průbězích EEG vln. V tabulce 2.2 jsou uvedeny frekvenční pásma EEG vln se stavy spojovanými s těmito vlnami [11].

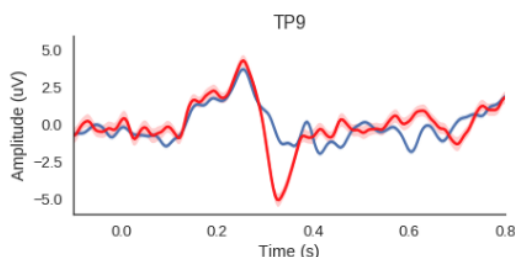
V úvodu jsem zmínil evokované a kognitivní evokované potenciály (EP a ERP). Soustředím se zde na tři příklady, které lze s pomocí Muse naměřit. Evokovaný potenciál je reakcí nervové soustavy na nějaký podnět. Protože se EP (ERP) měří pomocí

Označení EEG vlny	Frekvence	Stav
Gama (γ)	$> 35 \text{ Hz}$	Řešení problémů, koncentrace
Beta (β)	$12 - 35 \text{ Hz}$	Zaneprázdněnost, aktivita mysli
Alfa (α)	$8 - 12 \text{ Hz}$	Při zavřených očích, relaxace
Theta (θ)	$4 - 8 \text{ Hz}$	Únava, ospalost
Delta (δ)	$0.5 - 4 \text{ Hz}$	Spánek, snění

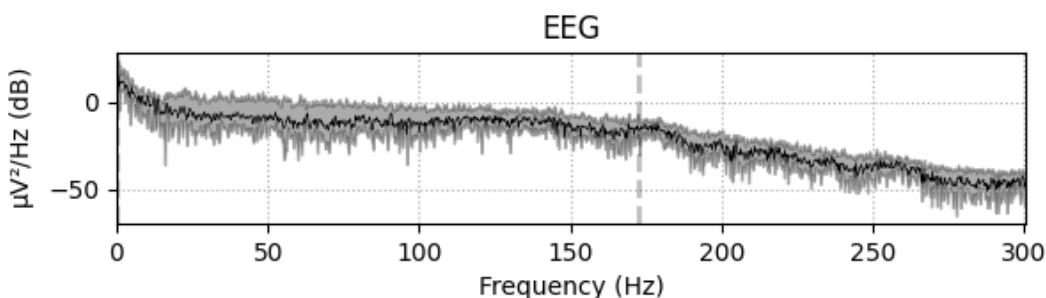
Tabulka 2.2. Frekvenční pásma EEG vln, převzato z [2, 14].

EEG, jde o měření změny elektrického napětí v čase. Kognitivní evokované potenciály (ERP) jsou pak charakteristické tím, že vznikají od identifikovatelných podnětů a jsou na tento podnět vázány. Původ jejich vzniku je v současnosti vysvětlován jako odraz aktivity komplexních neuronových sítí. Jsou to ty, které mají detekovat nové podněty a umožňovat rozlišovací schopnost člověka (např. selekce důležitých podnětů). Dále jsou klasifikovány podle latence na krátkolatentní (EP) a dlouholatentní (ERP), přičemž pomalé mozkové potenciály jsou vázány na vědomou i nevědomou kognitivní aktivitu a nejméněji reprezentují aktivitu CNS v průběhu běžného života. U EP (ERP) se sleduje řada parametrů, jako např. amplituda měřená od základní linie či amplituda mezi vrcholy, záporná či kladná polarita komponent, délka trvání jednotlivých komponent, latence, a tedy čas uběhnutý od náběhu stimulu a také přítomnost jednotlivých komponent v průběhu signálu. Protože se měření mnohokrát opakují během jedné sekce měření s jedním subjektem. Dále se při zpracování signál segmentuje a tyto segmenty se průměrují, tudíž se snižuje vliv šumu na signálu, a to je jedna z výhod EP (ERP). Dále jsou to testy poměrně citlivé, byť málo specifické a používají se především ve výzkumu [5].

Všechny tři EP (ERP), které zde zmíním, mají vizuální stimul. ERP jsou označeny jako N170 a P300 a jedno EP je označováno jako ustálený vizuální evokovaný potenciál (SSVEP). Komponenta N170 je velká ERP vlna, která se v signálu objevuje při detekování obličejů a je lokalizovatelná podle latence 170 ms po přijetí vjemu. Největší aktivitu způsobuje stimul obrázku lidské tváře ze předu nebo z profilu. Snad nejvíce známou komponentou ERP je vlna P300. Ta se objevuje přibližně 300 ms po přijetí vjemu nového, neočekávaného stimulu. Test je obvykle navržen jako tzv. „oddball“ paradigma, kdy prezentace sekvence opakujících se stimulů je nepravidelně přerušována deviantním stimulem. K této vlně je však možné dojít i pomocí zvukového nebo somatického stimulu. Posledním je SSVEP, což je potenciál evokovaný blikajícím stimulem s frekvencí v rozmezí $6 - 75 \text{ Hz}$. Aktivita se stejnou frekvencí může být detekována na týlním laloku mozku. Tohoto je často využíváno v aplikacích BCI také v důsledku vysokého frekvenčního rozlišení SSVEP [15].



Obrázek 2.1. ERP komponenta P300 při měření s Muse s největší amplitudou na elektrodě TP9, převzato z [16].



Obrázek 2.4. Výkonová spektrální hustota (PSD) EEG signálu, převzato a upraveno z [20].

Graf výkonové spektrální hustoty je v předzpracování důležitým ukazatelem funkčnosti metod předzpracování, např. při vyobrazení technických artefaktů jako rušení nebo pro ověření frekvenční skladby signálu pro měřené jevy jako je např. spánek.

2.3 Filtrování

Jedním z nejpoužívanějších nástrojů ve zpracování biosignálů je filtrování. Mezi hlavní cíle tohoto přístupu je odstranění šumu a jiných rušivých jevů, tzv. artefaktů. Artefakty rozlišujeme na dvě hlavní skupiny, tedy technické nebo také fyzikální a biologické artefakty, které mají řadu podskupin [2, 6]:

- Technické (fyzikální artefakty):
 - Síťový brum
 - Elektrostatické potenciály
 - Šum elektronických obvodů
- Biologické artefakty:
 - Mrkání
 - EOG - pohyb očí
 - EKG - srdeční rytmus
 - EMG - pohyb čelistí, hlavy, křeče

Stejně jako předzpracování, tak i filtrace závisí na balancování poměru signálu k šumu. Výzkumníci například v různých aplikacích používají filtry typu dolní propust s mezní frekvencí 30 Hz, protože u vyšších frekvencí je vyšší pravděpodobnost, že budou obsahovat šum a artefakty svalové aktivity (EMG). Jiné výzkumy se zase soustřeďují na evokované potenciály (ERP) a využívají jiné techniky vzhledem k faktu, že frekvence ERP jsou velmi blízké frekvencím alfa vln. Je tedy důležité aplikovat filtrování podle měřeného fenoménu [6, 21].

Účelem filtrů je zvýraznit specifické frekvence v signálu při potlačení jiných frekvencí a k tomu jsou určeny čtyři druhy filtrů: dolní a horní propust, pásmová propust a pásmová zádrž. Dolní, resp. horní propust propouští frekvence nižší resp. vyšší než mezní frekvence, pásmová zádrž nepřenáší jen definované pásmo frekvencí a pásmová propust přenáší jen definované pásmo frekvencí. Přenosové charakteristiky lze modelovat různými aproximacemi, mezi které patří např. Butterworthova, Čebyšelova nebo Besselova aproximace. Dále se filtry rozlišují na dva druhy, a to FIR, tedy filtry s konečnou dobou odezvy a IIR, což jsou filtry s nekonečnou dobou odezvy.

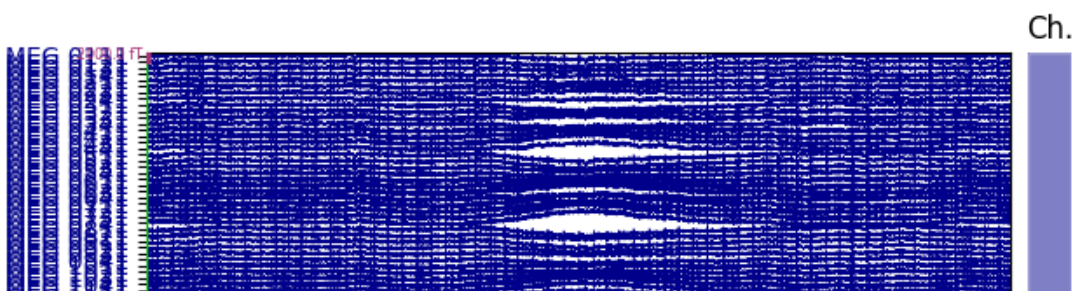
$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Mz^{-M}} \quad (9)$$

Ve frekvenční doméně je přenos soustavy filtru uveden v rovnici (9), kde koeficienty v čitateli b_k a koeficienty ve jmenovateli přenosu a_k mohou sloužit k obdržení výsledku výstupních dat $y(n)$ při zadání vstupních dat $x(n)$:

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k) \quad (10)$$

V případě filtrování signálu EEG se většinou více používá FIR filtrů, a to i v časově-frekvenční analýze. Další charakteristické vlastnosti FIR filtru jsou lineární fázová charakteristika, stabilita a proti IIR filtrům je FIR ve výpočtech časově náročnější. Dalším možným označením FIR filtrů je moving average filtr (MA), protože výstupem filtru je vážený průměr $N + 1$ posledních vzorků na vstupu. Filtry IIR jsou rekurzivní filtry obvykle nižšího řádu s nelineární fázovou charakteristikou, ovlivňující fázi a při nesprávném nastavení mohou být nestabilní. Při návrhu filtrů je nutné balancovat mezi kvalitou filtru a přesností i rychlostí výpočtu [2, 6, 22].

Jedním z jevů, které se při měření a následné analýze surových dat v EEG signálu objevují, jsou změny směrnice izoelektrické linie EEG signálu, jež jsou způsobené tzv. pomalými napětovými drifty. EEG signál je superponován na této DC složce signálu. Průběh takového artefaktu je znázorněn na obrázku 2.5. Ty jsou způsobeny na rozhraní elektrody a pokožky, mezi nimiž se nachází vrstva elektrolytu. Pro jeho potlačení na úrovni předzpracování se používá filtr typu horní propust s mezní frekvencí $0.5 - 1 Hz$. Používání hornípropustných filtrů je problematické z hlediska tvorby tzv. hranových artefaktů, které mohou trvat až $6 s$, což může znamenat problém při segmentaci signálů do epoch. Mnoho segmentačních postupů vytváří epochy kratší, než je možná doba trvání hranových artefaktů, tudíž je vhodné nasazovat tento typ filtru na kontinuálních datech [6, 21, 23].



Obrázek 2.5. Průběh surových EEG signálů v časové oblasti s viditelným driftem, převzato a upraveno z [24].

Mezi další artefakty, které je vhodné odfiltrovat, patří síťový brum na frekvenci $50 Hz$, který se v signálu často objevuje. Mnoho výzkumů se soustřeďuje na frekvenční pásmo do $30 Hz$ a tak používají filtr typu pásmová propust s mezními frekvencemi k potlačení pomalých driftů a vyšších frekvencí produkovaných EMG a jiným šumem. Tato strategie ilustruje souvislost mezi sledovanými jevy a filtrováním, neboť pokud např. použijeme metody časově-frekvenční analýzy pro extrakci výkonu v pásmu $2 - 20 Hz$, není nutné filtrovat dolnoproductním filtrem s mezní frekvencí $40 Hz$. Pro sledování rychlých jevů se tato horní frekvence posouvá až ke $100 Hz$ [6].

nekvadratickou funkci G , je aproximace vyjádřena pro prakticky libovolnou kvadratickou funkci G :

$$J(y) \propto [E\{G(y)\} - E\{G(\nu)\}]^2 \quad (28)$$

Následující funkce G se projevily v praxi jako velmi vhodné, protože nejsou příliš rychle rostoucí a poskytují odhadu potřebnou robustnost [26–27]:

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u, \quad G_2(u) = -\exp\left(-\frac{u^2}{2}\right), \quad 1 \leq a_1 \leq 2 \quad (29)$$

kde a_1 je vhodná konstanta, většinou se volí rovna jedné. Tímto je aproximace negativní entropie dobrým kompromisem mezi vlastnostmi dvou klasických metod popsaných výše. Mezi další metody odhadu ICA patří minimalizace společné informace a metoda maximální věrohodnosti.

Důležitým krokem před aplikací ICA na datech je jejich předzpracování, které usnadňuje odhad a zlepšuje podmínky ICA. Oba tyto kroky byly v předchozím textu už zmíněny, jedná se o centrování a bělení. Centrováním, jakožto nejzákladnějším postupem předzpracování, se myslí odečtení vektoru středních hodnot $\mathbf{m} = \mathbf{E}\{\mathbf{x}\}$ od vektoru \mathbf{x} . Vektor \mathbf{x} je potom vektor s nulovou střední hodnotou. V aplikaci ICA algoritmu to znamená, že vektor \mathbf{s} má také nulovou střední hodnotu. Je však možné po odhadu transformační matice \mathbf{A} přidat vektor středních hodnot \mathbf{s} zpět k centrovaným odhadům \mathbf{s} . Tento vektor středních hodnot je dán výrazem $\mathbf{A}^{-1}\mathbf{m}$, kde vektor \mathbf{m} je vektor středních hodnot odebraný v předzpracování [26].

Bělení je další metodou předzpracování a je zpravidla aplikována po centrování. Touto metodou lineárně transformujeme pozorovaný vektor \mathbf{x} tak, abychom dostali bílý vektor $\tilde{\mathbf{x}}$, což znamená, že jsou jeho prvky nekorelované a jejich rozptyl je jednotkový, nebo také že kovarianční matice $\tilde{\mathbf{x}}$ je rovna jednotkové matici:

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I} \quad (30)$$

V konečném důsledku bělení snižuje počet parametrů nutných k odhadu. Jednou z metod bělení je tzv. analýza hlavních komponent (PCA), která je probrána v další sekci. Prostředkem dosažení PCA jsou kromě adaptivních metod rozklad pomocí vlastních čísel (EVD) a rozklad pomocí singulárních čísel (SVD) [28].

Transformace pomocí bělení je vždy možná. V následujícím textu je popsána metoda rozkladu vlastních čísel kovarianční matice $E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T$, kde \mathbf{E} je ortogonální matice vlastních vektorů $E\{\mathbf{x}\mathbf{x}^T\}$ a \mathbf{D} je diagonální matice jejich vlastních čísel, tedy $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$. Bělení je tedy možno provést pomocí následující rovnice [26]:

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x} \quad (31)$$

kde matice $\mathbf{D}^{-1/2}$ je určena přes prvky jako $\mathbf{D}^{-1/2} = \text{diag}(d_1^{-1/2}, \dots, d_n^{-1/2})$. Bělení mění transformační matici na novou matici $\tilde{\mathbf{A}}$. Z rovnic (14) a (31) vychází:

$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{A}\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s} \quad (32)$$

Podstata užitečnosti funkce bělení je fakticky ve tvorbě nové ortogonální matice $\tilde{\mathbf{A}}$. Ortogonální matice má jen $n(n-1)/2$ stupňů volnosti, zatímco při použití matice \mathbf{A} bychom museli odhadnout n^2 parametrů. Z toho plyne, že proces bělení redukuje počet parametrů k odhadu. Zároveň je vhodné redukovat dimenzi dat podle velikosti vlastních čísel d_j kovarianční matice $E\{\mathbf{x}\mathbf{x}^T\}$, tedy aplikací PCA [26].

Kapitola 3

Nástroj pro předzpracování EEG signálů

V této části práce popisuji vývoj nástroje na streamování, nahrávání a předzpracování dat z čelenky Interaxon Muse MU-02 2016 (dále je Muse). V první sekci popisuji hardwarové vybavení Muse s jeho funkcemi a atributy a dále např. zařazení do skupiny zařízení Brain-Computer Interface (BCI). V následující, druhé sekci rozebírám programovací jazyk Python a použité knihovny pro streamování a nahrávání EEG z Muse a také knihovny použité k předzpracování dat. Ve třetí sekci této kapitoly se pak zmiňuji o komunikaci mezi Muse a počítačem. Obsahem předposlední sekce jsou praktické zkušenosti s implementací předzpracování naměřených dat. Poslední sekce zase popisuje programování uživatelského prostředí a ostatních funkcí finálního nástroje.

Muse patří do skupiny nositelných EEG headsetů, ze kterých je možné vytvořit BCI. Uvedu zde několik alternativních headsetů k Muse z různých cenových skupin, které se liší celou řadou parametrů a primárním zaměřením. Konkrétní model Muse, zmíněný v předchozím odstavci, který mi byl zapůjčen pro účely této diplomové práce se už nevyrábí. Jeho přímým nástupcem je Muse 2, vybavený pulzním oxymetrem a měří tepovou frekvenci pomocí metody fotopletysmografie s cenou cca 270 USD [30].

Mezi komerčně dostupné zařízení podobné Muse patří MindWave Mobile 2 od společnosti NeuroSky, Emotiv Insight a EPOC+ od společnosti Emotiv a DIY Neurotechnologist's starter kit od OpenBCI. Parametry jednotlivých zařízení jsou shrnuty v tabulce 3.1. Pokud je v řádku popsán jako „Podpora LSL“ heslo „Možná“, znamená to podporu pomocí komunitního softwaru, jinak je tento druh komunikace podporován přímo výrobcem.

Zařízení	Muse	Mobile 2	Insight	Epoch+	OpenBCI kit
Kanály	4+1 ref.	1+1 ref.	5+2 ref.	14+2 ref.	až 16
ADC rozlišení	12	12	15	16	24
Vzork. frekvence	256 Hz	512 Hz	128 Hz	256 Hz	250 Hz
Pohybový senzor	3 osy	N/A	9 os	9 os	3 osy
Podpora LSL	Možná	Možná	Možná	Možná	Ano
Výdrž baterie	cca 10 h	cca 8 h	cca 4 h	cca 6 h	cca 26 h
Cena	*230 USD	100 USD	300 USD	700 USD	1500 USD

Tabulka 3.1. Alternativní zařízení k Muse (*Muse už se neprodává), převzato a upraveno z [31].

Z tomto srovnání značně vybočuje řešení od OpenBCI, které je specifické svojí modularitou, kdy hlavní součástky kitu jsou headset a základní deska. Jde o řešení, které je otevřené pro komunitu vývojářů asi nejvíce ze všech srovnávaných zařízení.

■ 3.2.2 PyQtGraph

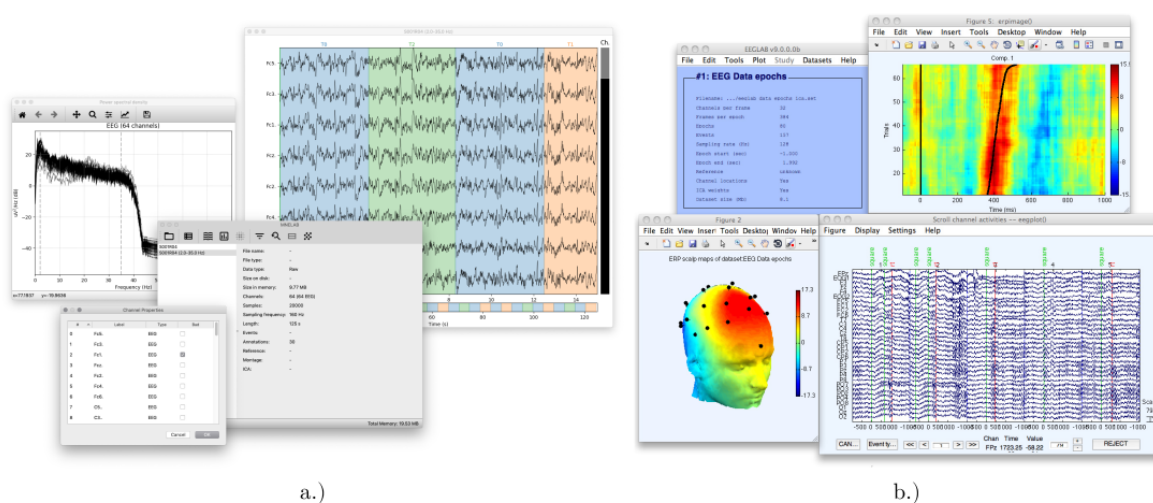
PyQtGraph je knihovna napsaná přímo v Pythonu s využitím NumPy pro numerické operace a Qt frameworku Graphics View, a tak je v ní zajištěno rychlé vykreslování. Je také postavená na PyQt4 resp. Pyside, ale dá se využívat i s novějšími verzemi těchto toolkitů, tedy PyQt5 a PySide2 (druhé v beta verzi). Výhodou této knihovny je možnost vložení grafu vytvořeného v této knihovně do PyQt GUI. Další předností je vytvoření aktualizujících se grafů a díky rychlosti knihovny lze pozorovat průběhy vykreslovaných veličin téměř v reálném čase. Mezi další přednosti knihovny je řada implementovaných nástrojů pro vědecké aplikace [44].

■ 3.2.3 MNE-Python

Knihovna MNE-Python je open-source knihovna napsaná v Pythonu, a slouží k analýze, vizualizaci EEG, MEG a dalších tříd dat. Tato knihovna obsahuje nástroje pro vstup/výstup dat v různých formátech, předzpracování, vizualizaci (i 3D), odhad zdroje signálu, časově-frekvenční analýzu, strojové učení a statistiku. Je také dostupná na Python Package Indexu. MNE-Python má také GUI MNELAB, které lze stáhnout separátně ke knihovně. K této knihovně existuje několik alternativních nástrojů v Pythonu jako PyEEG nebo NeuroPype, ale zmíním tu také knihovnu dostupnou knihovnu v Matlabu EEGLAB, která je ve své kompilované formě dostupná po vyplnění dotazníku. EEGLAB je velmi rozšířená knihovna a je možné se s ní setkat v analýze signálů z různých zařízení, dokonce i z Muse [9, 45].

■ 3.2.4 EEG-Notebooks

Jedná se o knihovnu programů ke spuštění prezentací pro měření různých EP (ERP) komponent. Při spuštění prezentace se začnou streamovat data z marker kanálu do LSL sítě. Jednotlivé ERP komponenty lze měřit a analyzovat z prostředí Jupyter Notebooks nebo i ve standardním Python skriptu. Tato knihovna lze používat se zařízeními Muse a OpenBCI Cython. V novější verzi této knihovny je streamování integrované do skriptu prezentací, což znamená, že není nutné spouštět stream z nástroje, ale stačí se k němu připojit pro online vizualizaci [15].



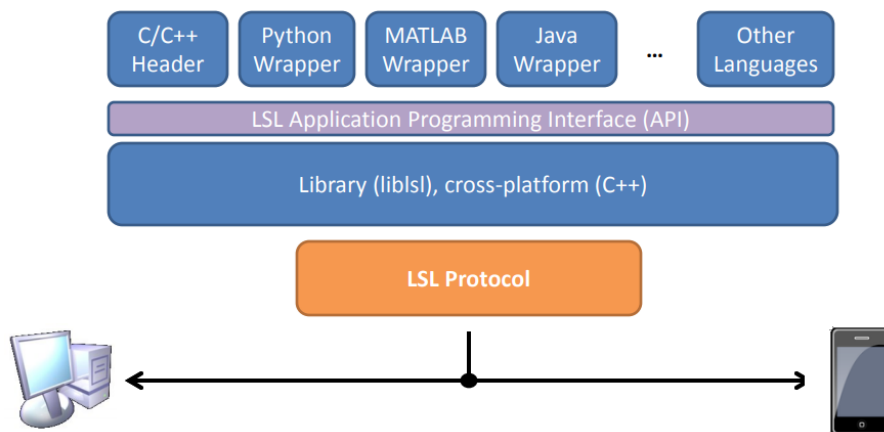
Obrázek 3.4. Srovnání grafických prostředí nástrojů MNELAB (vlevo) a EEGLAB (vpravo), převzato z [46–47].

3.3 Komunikace přes LSL a OSC

V této sekci budu rozebírat komunikační knihovny a protokoly, které používá Muse ke streamování dat do počítače. Mezi tyto prostředky patří protokol Open Sound Control (OSC) a Lab Streaming Layer (LSL). První jmenovaný byl navržen především pro multimédia, ke skládání a tvorbě hudby, druhý jmenovaný byl navržen prvotně jako nástroj pro přenos dat v laboratorním prostředí, zejména ve výzkumu EEG. Řešení LSL je v této diplomové práci preferováno a také použito při tvorbě zadaného nástroje, ale OSC je výrobcem podporované a nástroje výrobce, stejně jako mobilní aplikace k Muse využívají výhradně tento protokol.

3.3.1 LSL

Tento způsob komunikace byl vyvinut na univerzitě v San Diegu Christianem Kothem při působení v Swartz Center for Computational Neuroscience (SCCN). Není to jen komunikační protokol, celá distribuce LSL obsahuje protokol, knihovnu *liblsl* a soubor nástrojů které pracují na základech této knihovny. LSL je systém pro unifikovaný sběr dat z měření časových řad ve výzkumných experimentech a řeší vytváření sítí, časovou synchronizaci a přístup k zařízení v téměř reálném čase. Na obrázku 3.5 je vyobrazena základní rozložení komponent, LSL protokol ve spodní části obrázku je zodpovědná za jednotnost komunikace po síti, knihovna *liblsl*, která je multiplatformní, se stará o komunikaci mezi jednotlivými zařízeními a v horní části je vrstva aplikačního rozhraní pro jednotlivé programovací jazyky, jako je Matlab či Python. LSL je open-source licencované MIT licencí a podporuje řadu různých přístrojů, jako je EEG hardware, BCI, přístroje pro sledování pohybu očí (eye tracking) či přístroje pro sledování pohybu člověka (motion capture) [48].



Obrázek 3.5. Základní komponenty architektury LSL, převzato z [49].

Pro zajištění spolehlivosti přenosu přebírá atributy TCP protokolu, dotazování je implementované pomocí zpráv a je zajištěna typová bezpečnost. Jsou zde také definovány abstrakce pro používání klientskými programy, jako *Stream Outlets*, které poskytují streamované časové řady zařízením v síti, *Stream Inlets* pro přijímání streamovaných dat z připojeného *Outletu*, dále pak *Resolve Functions*, které např. objevují streamy dat na síti pomocí dotazování na jména či obsah meta-dat. Vzorčky dat jsou opatřeny časovými značkami pro jejich případnou vzájemnou synchronizaci.

Knihovna v Pythonu zapojená do aplikačního rozhraní LSL se jmenuje *pylsl*. Tuto knihovnu také používá knihovna Muse LSL, ve které autor Alexandre Barachant de-kódoval Bluetooth Low Energy protokol, který Muse používá. Knihovna Muse LSL je

klíčová pro streamování dat mezi počítačem a Muse přes Bluetooth s pomocí systému LSL [8].

■ 3.3.2 OSC

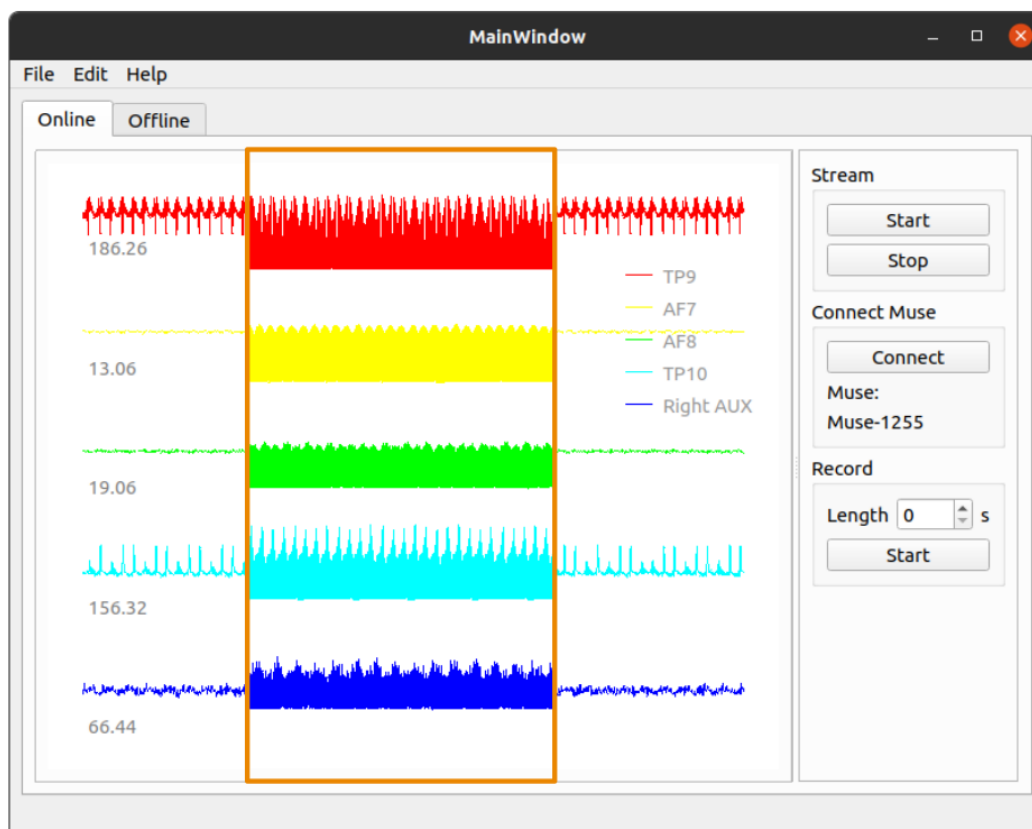
Vývoj protokolu OSC je vázán na Kalifornskou univerzitu v Berkley. Motivací k vývoji tohoto protokolu byly hlavně požadavky při práci se zvukovými signály spolu se synchronizací mezi zařízeními (např. v elektronických syntetizátorech). Specifikace 1.0 byla publikována v roce 2002, v několika publikacích se o něm hovoří jako o standardu. Topologie ani transportní vrstva nejsou striktně definovány, ale často se používá v modelu klient - server. Autoři [50] se zmiňují o nedostatečných implementacích OSC protokolu, přičemž jedním z nich je chybějící obousměrná komunikace. V mnoha implementacích funguje jen jednosměrná komunikace mezi klientem a serverem, kde klient odesílá pakety serveru. OSC zprávy jsou tvořeny specificky podle toho, zdali se jedná o jednu zprávu nebo o tzv. bundle, tedy několik zpráv. Společná struktura zprávy je oddělování jednotlivých částí cesty pomocí znaku „/“. Tento syntax reprezentuje stromovou strukturu, kde znak dopředného lomítka reprezentuje vnitřní uzel stromu, části cesty jsou pak názvy hran a volané metody jsou listy. V řetězci jednoduché zprávy se pak nachází ještě řetězec nesoucí typy argumentů a dále samostatné argumenty. Celá zpráva v rozhraní komunikačních aplikací abstrahována, a proto ji zde neuvádím. Je dostupná v specifikaci OSC nebo v následujícím citovaném zdroji. Na příkladu OSC komunikace Muse v tabulce 3.2 jsou ve sloupcích *OSC cesta* a *Hodnoty* patrné první dvě části jednoduché zprávy OSC. Přijaté zprávy jsou identifikovány adresou a na straně OSC serveru je nalezena metoda odpovídající argumentu a je následně volána [51].

Položka	OSC Cesta	Hodnoty	Vzorkování
Surové EEG data	<i>/muse/eeg</i>	<i>f f f f {+f}</i>	256 Hz
Abs. pásmo Delta	<i>/muse/ * /delta_absolute</i>	<i>f f f f f</i>	10 Hz
Abs. pásmo Theta	<i>/muse/ * /theta_absolute</i>	<i>f f f f f</i>	10 Hz
Abs. pásmo Alfa	<i>/muse/ * /alpha_absolute</i>	<i>f f f f f</i>	10 Hz
Abs. pásmo Beta	<i>/muse/ * /beta_absolute</i>	<i>f f f f f</i>	10 Hz
Abs. pásmo Gamma	<i>/muse/ * /gamma_absolute</i>	<i>f f f f f</i>	10 Hz
Ind. kontaktu elek.	<i>/muse/ * /horseshoe</i>	<i>f f f f</i>	10 Hz
Dotek čela	<i>/muse/ * /touching_forehead</i>	<i>i</i>	10 Hz
Info o baterii	<i>/muse/batt</i>	<i>i i i i</i>	0.1 Hz
Gyroskop	<i>/muse/gyro</i>	<i>f f f</i>	52 Hz
Akcelerometr	<i>/muse/acc</i>	<i>f f f</i>	52 Hz
Mrknutí	<i>/muse/ * /blink</i>	<i>i</i>	10 Hz
Sevření čelisti	<i>/muse/ * /jaw_clench</i>	<i>i</i>	10 Hz
Markery	<i>/Marker/{1, 2, 3, 4, 5}</i>	<i>i</i>	Okamžité

Tabulka 3.2. Specifikace OSC cest z aplikace Mind Monitor, popř. Muse-IO a Muse Direct (*Elements), převzato a upraveno z [52].

3.4.1 Záložka Online

Prostředí nástroje je zobrazeno na obrázku 3.6. Tato obrazovka je úvodní a zobrazuje okno vykreslovaného grafu, ovládací prvky po jeho pravé straně a lištu s menu. Prostředí programu je opatřeno popisky jen v angličtině.



Obrázek 3.6. Okno programu se zobrazenou záložkou režimu online s vloženým výstřížkem z nefiltrovaného průběhu v oranžovém ohraničení

Jednotlivé grafické prvky, tlačítka a boxy pro zadávání hodnot jsou propojeny s metodami v objektově orientovaném kódu programu a jejich vlastnosti jsou tyto:

- Levá část okna:
 - Graf - plocha pro vykreslování živých průběhů surových dat. Po kliknutí na jakýkoliv průběh signálu z implicitního filtrovaného stavu přepne průběh na stav nefiltrovaný, dalším kliknutím se vrátí zpět. Jedná se o FIR filtr typu pásmová propust s mezními frekvencemi 3 a 40 Hz. Pod každou křivkou se aktualizuje hodnota směrodatné odchylky signálu, doporučená hodnota pro začátek měření na všech elektrodách je menší než 10 (mimo nezapojené přídavné elektrody *Right AUX*). Legenda signálů na pravé straně okna lze volně přemístit, jednotlivé křivky lze přiblížit i oddálit.
- Pravá část okna:
 - Stream - Rámeček obsahuje dvě tlačítka Start a Stop, jimiž se zapíná a vypíná LSL stream dat. Po vypnutí streamu se Muse odpojí a při opětovném zapnutí se znovu spáruje s počítačem.

- Raw Data - Zde je možné sledovat charakteristiky surových dat pomocí dvou grafů, tedy časového průběhu po stisknutí *Plot Raw* a frekvenční charakteristiky pomocí *Plot PSD*. Pokud uživatel použije funkci anotace v časovém grafu, je možné odstranit nežádoucí části signálů i celé EEG kanály pomocí funkce po stisknutí tlačítka *Reject bads*.
- Rereferencing - V tomto boxu je rozvinovací seznam, kde je možné vybrat způsob rereferencování signálů.
- Filter - Box sloužící k filtrování signálů pomocí různých druhů filtrů, přičemž jejich použití záleží na sledovaném frekvenčním pásmu a tedy i sledování různých fenoménů v EEG.
- Epoch - Zde je možné segmentovat signál do epoch. A to jak signál naměřený bez markerů (např. v relaxovaném stavu), dále je možné použít experimentální funkci k vyhledání epoch pomocí EOG událostí, nebo použít markery k segmentaci EP (ERP) do trialů. Následně je možné zobrazit epochy v časovém grafu.
- ICA - Tento box je zaměřen na aplikaci analýzy nezávislých komponent (ICA). Po dokončení běhu algoritmu je možné zobrazit řadu grafů zobrazující komponenty a signál po aplikaci korekce ICA.
- Spodní část okna:
 - Data overview - Ve spodní části obrazovky se nachází přehled o datovém souboru surových dat. Je tak možné např. ověřit, že soubor byl naměřen s markery a jak dlouho trvalo měření.

3.5 Implementace GUI a ostatních funkcí

V této sekci se budu zabývat vývojem nástroje s vlastnostmi, které byly určeny v zadání práce. Především popíšu vlastnosti komunikace a jak jsem dospěl k rozhodnutí použít jednu variantu, také ale vývoj uživatelského prostředí a dalších funkcí, jako je export nahraných dat do různých typů souborů.

3.5.1 Komunikace

Nyní se lze k Muse připojit pomocí dvou způsobů. Jedním z nich je prostřednictvím protokolu OSC a druhým je pomocí systému LSL. Vzhledem k tomu, že Muse-IO ani Muse Direct od společnosti Interaxon už nejsou volně dostupné, přičemž dříve zmíněnou aplikaci se mi nepodařilo uvést do chodu a Muse Direct jsem při počátku práce na výsledném nástroji nedokázal najít, rozhodl jsem se, že dám přednost komunikaci pomocí LSL. Komunikace s Muse pomocí protokolu OSC je ovšem stále možná prostřednictvím aplikace Mind Monitor. Je však nutné připojit se k Muse pomocí mobilního telefonu. Mobilní aplikací požadovaný operační systém Apple iOS a Android má dnes naprostá většina zařízení na trhu. Avšak tato kombinace používá 2 sítě. Přes Bluetooth se mobilní telefon připojí k Muse a mobilní telefon musí být připojen na stejné Wi-Fi síti jako počítač, po které mezi sebou komunikují. Interaxon kromě v tomto odstavci zmíněných nástrojů nabízel ještě aplikaci MuseLab, kterou bylo možné stáhnout a nainstalovat alespoň na počítač s operačním systémem Windows. Tato aplikace funguje jako OSC server, a je možné ji nahradit vlastním řešením naprogramovaným např. v jazyce Python. Připojení Muse k počítači přímo přes Bluetooth je však z mého pohledu snazší řešení.

Vývoj nástroje a jeho testování probíhalo na operačním systému Ubuntu 20.04 LTS (Focal Fossa) a pro tento OS (případně další distribuce Linuxu) je primárně určen. V

prvotním testování komunikace byly zastoupeny operační systémy Ubuntu 19.10 (Eoan Ermine), Windows 7 SP1 a Windows 10 (build vyšší než 1703 - Creators Update) a u všech těchto OS jsem zjistil jednu nebo více konfigurací připojení, které fungovaly uspokojivě.

Muse používá Bluetooth modul verze 4.0 Low Energy. Zpětná kompatibilita tohoto modulu vůči modulu s Bluetooth verzí 2.0 na notebooku (modul Broadcom BCM92070MD) byla na operačním systému Linux i Windows vyhodnocena jako negativní. K dispozici jsem měl k otestování řadu Bluetooth USB donglů s různými čipy a také integrovaný modul v počítači, všechny ve verzi Bluetooth 4.0 LE.

Výrobce	Model	Ubuntu 20.04	Windows 7	Windows 10
Kuongshun	CRS 4.0	Ano	Ne	Ano
Asus	USB-BT400	Ano	Ne	Ano
Connect IT	CI-479	Ne	Ne	Ano
Silicon Labs	BLED112	Ne*	Ano	Ano
Broadcom	BCM20702	Ano	-	Ano

Tabulka 3.3. Srovnání Bluetooth modulů v testu připojení k Muse na různých OS.

V případě USB donglu BLED112 je rozpoznání počítačem s OS Linux funkční, ale nejspíše v důsledku chyby v jedné z použitých knihoven (pygatt) se mi na tomto operačním systému s Muse nepodařilo komunikovat. Tento dongle je ovšem preferovaný jednak v knihovně Muse LSL a jednak v knihovně EEG-Notebooks. Používá totiž jiný čip než ostatní a chová se odlišně při připojení k počítači, protože vytváří virtuální sériový port, přes který potom komunikace probíhá. Pro správné fungování tohoto donglu v operačním systému Windows je nutné si ze stránek výrobce stáhnout ovladače. Je také možné stáhnout si SDK pro vývojovou práci s Bluetooth LE.

Nejlepší zkušenosti jsem měl s donglem CRS 4.0, který poskytoval velmi stabilní připojení a zaznamenal jsem s ním nejméně chybějících vzorků v průběhu vývoje a testování na všech platformách.

Inicializace komunikace mezi Muse a počítačem pomocí LSL lze jednoduše spustit z terminálu. Knihovna Muse LSL má totiž naprogramované rozhraní příkazového řádku (CLI). Nejdůležitějším příkazem pro navázání komunikace a spuštění streamu je zařízeno pomocí příkazu:

```
muselsl stream (--argumenty)
```

Tato funkce je volána i v mém programu, jak popíši později. Má však mnoho argumentů, které specifikují režimy komunikace. Mezi argumenty patří *name* a *address*, které usnadňují hledání zařízení v LSL síti, např. "Muse-1255" resp. MAC adresa Bluetooth adaptéru. Dále jsou tu argumenty pro tzv. *backend*. Pro Windows 10 to buď "bluemuse" nebo "bgapi", pro Linux "gatt" nebo "bgapi" a pro Windows 7 lze použít pouze "bgapi". Argument "bluemuse" odpovídá používání aplikace BlueMuse na Windows 10, která zajišťuje připojení Muse a streamování dat. Argument "bgapi" odpovídá používání BLED112 donglu. Je zde také implicitní argument, který automaticky vyhledá nejlepší mód připojení. Posledním argumentem týkajícím se komunikace je *interface*. Specifikuje rozhraní a při použití Bluetooth stacku GATT je argumentem "hci0" a v případě použití BGAPI je jím "název sériového portu". Implicitně se streamují pouze

EEG data, je to zde ale možné pomocí argumentů změnit. Pomocí argumentů "acc" a "gyro" je možné streamovat také data z akcelerometru, resp. z gyroskopu.

Dále je zde funkce pro nahrávání dat, která se ale neobejde bez výše zapnuté funkce streamování, a je nutné ji zapnout prostřednictvím dalšího terminálového okna.

```
muselsl record (--argument)
```

Tento příkaz umožňuje nahrávání dat do csv souboru nebo několika souborů, pokud je specifikován argument "type". Může tak dojít k nahrávání dat ze všech senzorů Muse. Kromě toho lze použít argument "duration" pro specifikaci délky nahrávky v sekundách a také lze specifikovat jméno souboru pomocí argumentu "filename".

Jak už jsem se zmínil, na Windows 10 je možné stáhnout si aplikaci BlueMuse, která zajišťuje vyhledávání zařízení v LSL síti a připojení počítače k Muse. Vyžaduje aktualizaci Windows 10 na Creators Update (build 1703).

Na Linuxu se v průběhu vývoje ukazovaly problémy s připojením, konkrétně s knihovnou poskytující aplikační rozhraní k GATT v Pythonu, která se jmenuje *pygatt*. Pro skenování sítě je nutné mít v Linuxu oprávnění *root*. Je nutné mít nainstalovanou knihovnu *libcap* a pokud chce uživatel používat CLI knihovny Muse LSL, je nutné zadávat heslo k nejvyššímu oprávnění na Linuxu. Je možné to obejít nastavením oprávnění pro konfiguratoru Bluetooth připojení *hcitool* pomocí příkazu:

```
sudo setcap 'cap_net_raw,cap_net_admin+eip' 'which hcitool'
```

Je ale možné, že tento doporučený způsob nepomůže, jako se to stalo mně. V tom případě je ještě možné povolit běžnému uživateli operačního systému nepoužívat heslo při využívání práv *root* v souboru *sudoers*.

■ 3.5.2 Formáty datových souborů

Muse LSL používá pro ukládání dat jednoduchý souborový formát CSV s definovanou hlavičkou, kde na první pozici je časová známka, resp. unixový čas, následují vzorky EEG signálu všech elektrod a na poslední pozici je volitelně marker stimulu, tedy pokud Muse LSL detekuje stream markerů.

timestamps	TP9	AF7	AF8	TP10	Right AUX	Marker0
1575389775	-30.762	-21.484	16.602	-54.688	21.973	0
1575389775	37.598	-18.555	16.602	-10.742	-35.645	0
1575389775	194.336	-11.23	9.766	189.453	-74.707	0

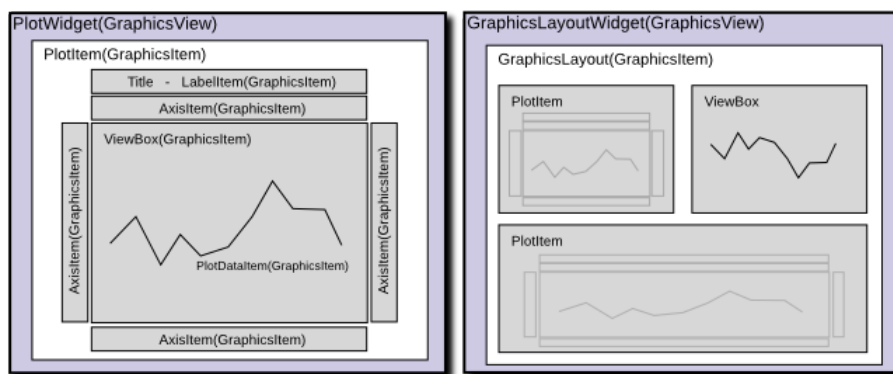
Obrázek 3.8. Ukázka CSV formátu nahrávky EEG dat pomocí Muse LSL

Formátů používaných pro zaznamenávání EEG dat je několik, obvykle se ale liší podle programu nebo prostředí, které ho používá. Obvyklá možnost výstupu téměř všech programů, toolboxů a programových prostředí pro zpracování EEG a podobných signálů je ve formátu RAW, který obsahuje hlavičku a změřená či minimálně zpracovaná data. V této práci využívám knihovnu MNE-Python, která je schopna nahrávat data z nejrůznějších formátů, jako např. European Data Format (EDF resp. EDF +), což je široce uznávaný standart pro sdílení EEG dat [53], nebo datový formát toolboxu EEGLAB (.set). Samotná knihovna MNE-Python má svůj nativní formát s příponou .fif, pomocí externích knihoven lze surová data uložit ve formátu .npy (Numpy) nebo .mat (Matlab).

■ 3.5.3 GUI

Grafické prostředí nástroje využívá knihovny PyQt5. Hlavní program je v podstatě celý sestaven kolem tříd této knihovny, přičemž mnou vytvořená hlavní třída GUI dědí ze základní třídy pro všechna rozhraní *QtWidgets* a nástroj je koncipován jako hlavní okno, tedy *QtWidgets.QMainWindow*. Dalším konceptem vytváření GUI v PyQt5 je vytváření šablony v samostatném grafickém prostředí Qt5 Designer, takže je poměrně snadné vytvořit libovolné rozložení programu a vkládat do něj různé ovládací prvky. Jednotlivé prvky GUI je nutné opatřit tzv. *layouty*, pokud má být okno velikostně škálovatelné. Tento program vytváří soubory ve formátu *.ui*, které je potom možné převést do Pythonu pomocí nástroje *pyuic5*, takže se z nich stane soubor s příponou *.py*. Toto řešení není nutné, je možné do souboru naimportovat šablonu bez nutnosti konverze. Třída hlavního okna potom dědí i z třídy *template.UiMainWindow*, což je jediná třída šablony.

Jedním z hlavních důvodů, proč tuto knihovnu používám je možnost vložení okna grafu knihovny PyQtGraph. Jak jsem již popsal v předchozích sekcích, jedná se o velmi rychlou knihovnu na vykreslování trendů téměř v reálném čase. Tato knihovna nabízí mnoho různých funkcí, např. také pro práci s 3D grafikou. Třídy, které slouží jako kontejner pro grafy a lze je vložit do PyQt5 GUI jsou dvě, a to *PlotWidget* a *GraphicsLayoutWidget*, viz obrázek 3.9, přičemž mnou vybraná třída je ta první jmenovaná. Vkládání do GUI se v programu Qt5 Designer provádí nejprve pomocí vložení generického Widgetu a následně použitím funkce *promote*.



Obrázek 3.9. Třídy knihovny PyQtGraph určené pro vkládání do PyQt5 GUI, převzato z [54].

Objekt vytvořený z třídy *PlotWidget* si lze všemožně upravovat, např. změnit barvu pozadí grafu z implicitní černé na bílou. Stejně tak lze do této třídy přidávat křivky, které jsou instancí třídy *PlotCurveItem* a také jim přidávat různé atributy, např. že je možné na ně kliknout a na základě této akce s nimi nějak naložit. V mém případě jsem ještě potřeboval aktualizovat tyto křivky tak, aby se přicházející data zobrazovala na grafu živě. Pro tento úkol lze vytvořit instanci třídy *QTimer*, která je zahrnutá v knihovně PyQt5, a následně ji použít k opakovanému volání metody, která data aktualizuje.

PyQt5 funguje ve smyčce událostí (event loop). Na konci programu ve funkci *main()* je volána metoda *QApplication.exec*, která vytváří nekonečnou smyčku a čeká na vnější události, jako např. kliknutí na tlačítko a v takovém případě volá příslušné metody. Dále je pro ovládání prvků v GUI využito mechanismu signálů a slotů. Signál je vyslán např.

po stisknutí tlačítka a k signálu je připojeno až několik slotů, tedy funkcí zavolaných po nastalé události [55].

Knihovna MNE-Python používá pro vykreslování grafů upravenou knihovnu Matplotlib, ve které jsou v různých grafech naprogramovány doplňující funkce, např. pro anotaci signálů pro případné odstranění částí signálu. Přístup k těmto grafům je v mém nástroji takový, že se obvykle po stisknutí tlačítka objeví vyskakovací samostatné okno s grafem.

Funkce zapnutí streamování a nahrávání, tak jak je to vidět na obrázku 3.6 je implementováno pomocí Python modulu *subprocess*. To je modul, který umožňuje vytvoření nového procesu. Těmito procesy jsou právě již zmiňované procesy Muse LSL, které lze vytvářet pomocí CLI, jak jsem uvedl výše. Konkrétně je to implementováno pomocí instance třídy *Popen*, ve kterém se specifikují argumenty, v mém případě jsou to příkazy CLI.

Celý proces inicializace streamování je jednoduchý, stačí stisknout tlačítka *Start* v boxu *Stream*. Následně je nutné připojit Muse pomocí tlačítka *Connect*. Nahrávání je možné stejně jako v případě CLI až při zapnutém streamu, a provádí se pomocí tlačítka *Stop* v boxu *Record*. Oba procesy stream a record lze také zastavit, v případě procesu stream je to vhodné udělat před zavřením nástroje. V programu je zahrnut i testovací režim, který vytváří stream náhodných čísel a pouští ho na síť LSL. Je tak možné si otestovat funkčnost nástroje i bez připojení k Muse.

Celý program byl vyvíjen a testován v integrovaném vývojovém prostředí (IDE) Pycharm. Jednou z jeho funkcí při vytváření projektů je tvorba virtuálního prostředí *venv*. To lze v Pythonu vytvořit i samostatně pomocí příkazové řádky pro Linux (v případě Windows vyměnit *python3* za *py -3*:

```
$ python3 -m venv nazev
```

kde argument *nazev* je libovolný název virtuálního prostředí. Aktivace virtuálního prostředí se provádí tímto způsobem, nejdříve pro Linux:

```
$ source nazev/bin/activate
```

A pro Windows:

```
> nazev\Scripts\activate
```

Pokud má počítač nainstalovaný Python ve verzi 3.8, je možné vytvořený nástroj pro předzpracování dat používat pomocí virtuálního prostředí. Stačí zkopírovat obsah CD do kořenového adresáře, vedle adresáře virtuálního prostředí. Instalace požadovaných knihoven se provádí pomocí Python instalátoru balíčku *pip*. Následujícím výrazem v již otevřeném aktivovaném virtuálním prostředí se nainstalují všechny potřebné moduly z textového souboru *requirements.txt*, který je také v příloze na CD.

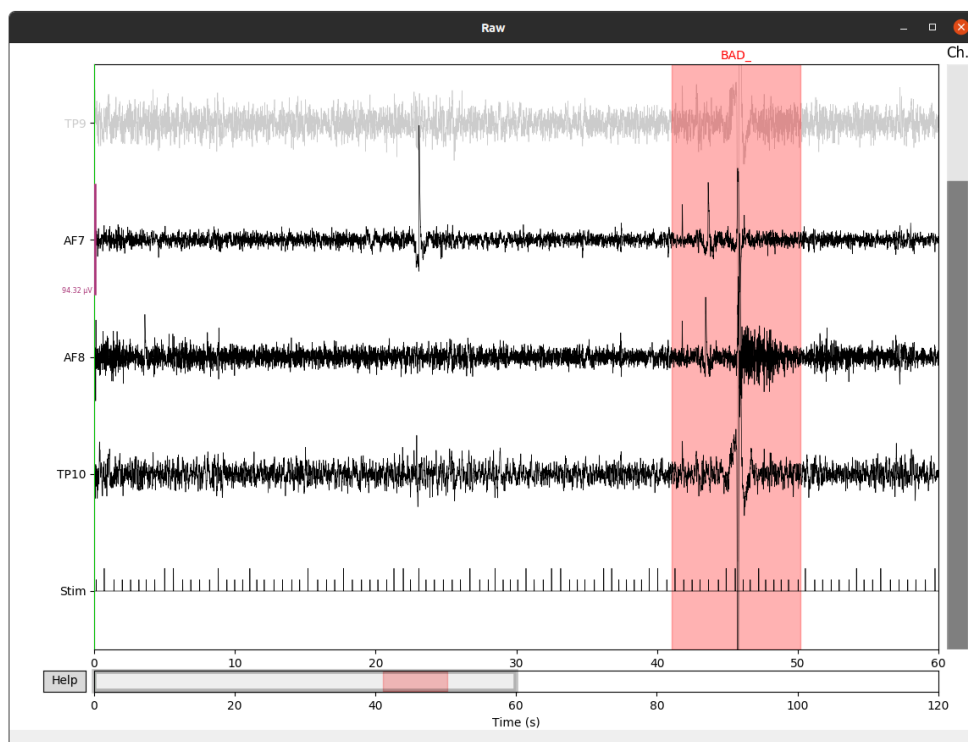
```
(venv) user@host:~/korenovy-adresar/$ pip install -r requirements.txt
```

3.6 Implementace předzpracování

Všechny funkce a algoritmy jsou napsány s pomocí knihovny MNE-Python, kde v hlavním skriptu je většina funkcí předzpracování a v samostatném skriptu importovaném do hlavního (s třídami GUI) jsou podpůrné funkce pro import dat. Knihovna MNE-Python je členěna do několika hlavních tříd, které nakládají s importovanými daty a aplikují na ně svoje metody.

Spojitá data jsou uložena v instancích třídy `Raw`, která mohou nebo nemusí být předzpracována, obvykle se touto instancí plní surová data přímo ze systémů získávání dat. Ostatní třídy se jmenují `Epochs` a `Evoked`, a staví na třídě `Raw`. Surová data mohou být segmentována do epoch, které obecně korespondují se segmenty dat po každém opakování stimulu a výsledkem může být objekt `Epochs`. Tyto segmenty pak mohou být dále zprůměrovány, aby vytvořily instanci objektu `Evoked` [9].

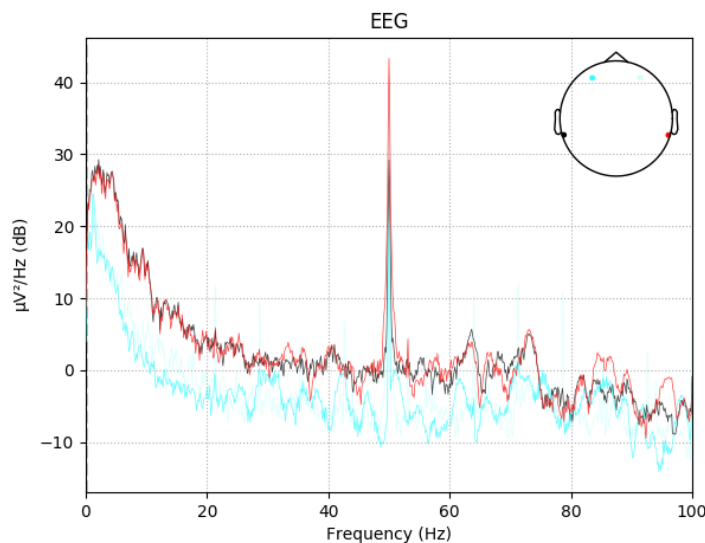
Vytvoření `Raw` objektu pro data z Muse obnáší několik kroků. Nejprve je nutné přečíst data z `.csv` souboru a získat data z hlavičky. Pro data měřená bez stimulačního streamu v datech chybí poslední sloupec `Marker0` a pokud nebyla při měření k dispozici přídavná elektroda `Right AUX`, je nutné ji odstranit z datasetu. Pro manipulaci s `.csv` soubory používám knihovnu `pandas` a její nástroje, včetně primární datové struktury `DataFrame`. Po načtení dat je nutné z nich zjistit několik povinných údajů, které jsou vstupními parametry pro vytvoření `Raw` objektu. Patří mezi ně názvy elektrod, typ elektrod (v případě Muse jen EEG), vzorkovací frekvence signálu, dále tzv. montáž, která koresponduje u Muse se systémem 10 - 20. Tyto údaje jsou uloženy v atributu `raw.info`. Do vytvoření instance třídy `Raw` je tato datová struktura zařazena hned za datovou strukturu dvou rozměrných dat časové řady EEG. Po vyplnění těchto dat je možné vyšetřit data v časovém grafu, což je metoda zavolaná na instanci třídy `Raw`.



Obrázek 3.10. Časový graf surových dat importovaných do programu, s možností anotace dat.

Inspekce časové řady hrubých dat je velmi důležitou součástí předzpracování EEG dat. Na obrázku 3.10 je zobrazeno okno časového grafu surových dat z Muse. Je možné si povšimnout červeného pásma manuálně přidané anotace, což knihovna MNE-Python umožňuje. Následně je přínosnou možností data vyčistit ručně. Do možnosti anotace by se také dalo zahrnout odstranění špatné elektrody (zde na obrázku jen ilustrace), kdy po kliknutí označení elektrody signál zeseďíví. Implicitní nastavení zobrazované délky signálu je 60 sekund.

Velmi užitečným nástrojem je pak zobrazení grafu odhadu spektrální výkonové hustoty surových dat, který je vykreslený na obrázku 3.11. Odhad je vytvořen pomocí Welchovy metody. V grafu surových dat je možné si povšimnout vyššího výkonu v oblasti okolo frekvence 50 Hz , tedy síťové frekvence v evropských zemích. Datové soubory, na kterých jsem tento program testoval, obsahují také ty, které jsou součástí knihovny EEG-Notebooks. Tyto soubory byly měřeny pravděpodobně v Severní Americe a mají tedy tento výkonový skok v grafu odhadu spektrální výkonové hustoty okolo frekvence 60 Hz . V pravém horním rohu obrázku je vidět rozmístění elektrod Muse v systému 10-20. V GUI je dále možné tomuto grafu omezit frekvenční rozsah zadáním mezních frekvencí k zobrazení. Poslední načtený graf může být uložen v menu *File - Export figure* pod jménem datového souboru s připojeným identifikátorem grafu ve formátu PNG. Pro export grafu typu jako je na obrázku 3.10 je toto nutností, protože nemá vlastní ovládací panel a možnost ukládání, jako tomu je u grafu PSD.



Obrázek 3.11. Frekvenční doména a graf odhadu spektrální výkonové hustoty surových dat

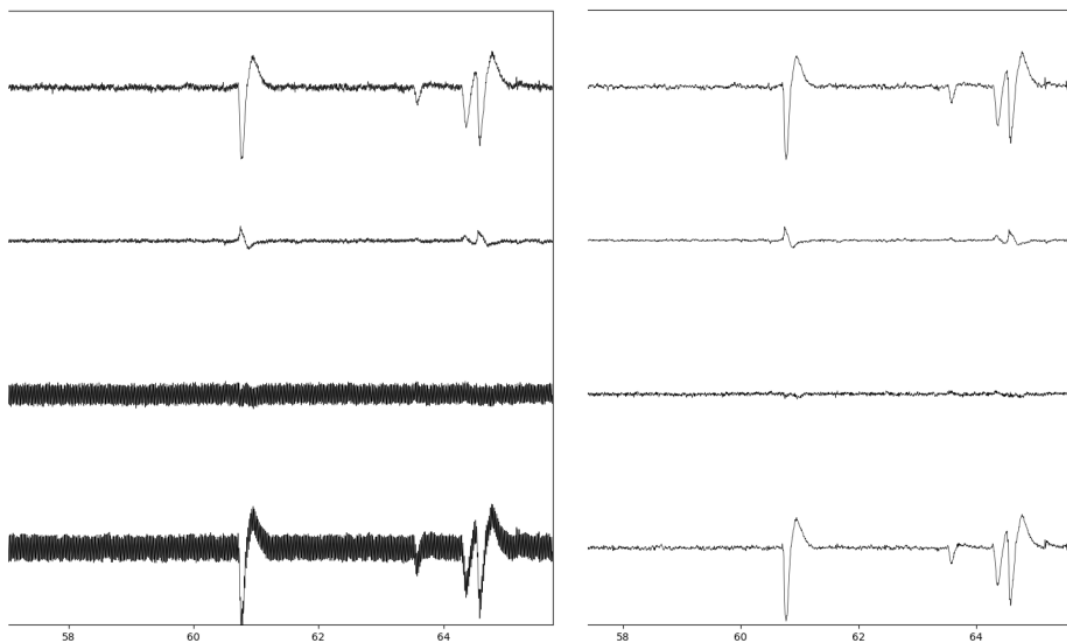
Dalším možným krokem předzpracování dat, který se provádí na spojitých signálech EEG, je převzorkování. Autoři knihovny MNE-Python [9] v dokumentaci nedoporučují aplikaci převzorkování na surových datech s markery, protože dochází k rozechvení časových trigger dat a pozdější segmentace do epoch pak trpí na sníženou přesnost časových událostí. Tento postup lze optimalizovat tak, že se nejdříve segmentují surová data a později se aplikuje převzorkování. Nicméně převzorkování je výhodné zejména pro zrychlení výpočtů, protože EEG data jsou vzorkována tak, že pro většinu rozsahu pozorování obvykle snadno splní Nyquistovu podmínku tak, aby nedocházelo k aliasingu.

Vzhledem k absenci přídavné elektrody jsem do grafického prostředí přidal funkci pro nakládání s touto elektrodou až na konec bloku prvotní práce se spojitými surovými daty, jak je vidět na obrázku 3.7. Pokud byla přídavná elektroda při měření přítomna, je nutné před importem dat zaškrtnout políčko *Aux channel present* a přejmenovat elektrodu dle pozice v systému 10-20, kde byla umístěna. V grafu surových dat, jako na obrázku 3.10, se potom zobrazí signál od této elektrody pod standardními signály Muse. Je však nutné podotknout, že všechny elektrody, a tedy i přídavná, jsou v programu napevno označeny jako EEG elektrody.

V první, teoretické kapitole jsem popisoval změnu reference EEG signálů jako lineární transformaci, ve které se signál nové referenční elektrody odečítá od každého signálu, včetně signálu nové reference. Je tak možné zahrnout do analýzy elektrodu, která předtím sloužila jako referenční elektroda. V případě mnoha přístrojů se ale data referenční elektrody nenahrávají a stejně tak je tomu v případě Muse, jejíž referenční elektroda je v systému 10-20 označena jako FPz. V nástroji pro předzpracování lze referenci měnit na již naměřených datech z implicitního způsobu použitým v Muse na průměr všech elektrod či průměr elektrod za ušima (TP9 a TP10), což je umístění blízké mastoidům. Platí však poučka, že reference by měla být zvolena tak, aby mozková oblast, která je podrobena zájmu měření byla co nejméně vzdálena nové referenci [21]. V případě nástroje a Muse se jedná o experimentální funkci a z článku [56] vyplývá, že autoři používají standardní referenci Muse k měření ERP komponent a porovnávají výsledky s mnohem flexibilnějším systémem s 64 elektrodami, který je rereferencován podle Muse.

Obvyklý postup předzpracování spojitých surových dat poté zahrnuje filtrování. Jak už bylo popsáno v předchozí kapitole, filtrování dat je způsob, který lze použít ke zvýšení poměru signálu k šumu, ale také je to metoda, která vnáší do dat určitá zkreslení a prioritní by mělo být měření kvalitních dat se snížením šumu a větším počtem trialů. Vzhledem k volbám autorů knihovny MNE-Python [9], ve které jsou filtry implicitně nastaveny jako FIR filtry, protože je jednodušší je nastavit, jsou vždy stabilní a mají dobře definované přechodové pásmo, jsou i v tomto nástroji použity FIR filtry jako hlavní způsob filtrování. V nástroji jsou použity tři druhy filtrů, které lze aplikovat na surová data typu *Raw*. Jde o filtry typu pásmová zádrž, který lze aplikovat na potlačení artefaktů síťového brumu, dále pak hornopropustní filtr, např. k potlačení artefaktů pomalých driftů v signálu a posledním je filtr typu pásmová propust, který bývá obvykle aplikován pro účely omezení frekvenčního pásma sledovaných trialů. Kombinací filtrů, i s vynecháním určitých kombinací je možné filtrovat signály pro různé sledovaná frekvenční pásma a pokusy.

Na obrázku 3.12 je vidět efekt použití filtrování na datech naměřených v klidu bez stimulů. Aplikován byl postupně filtr typu horní propust s mezní frekvencí 0.1 Hz a poté filtru typu pásmová zádrž s nastavenou frekvencí kolem 50 Hz a 100 Hz s nepropustným pásmem se šířkou 0.25 Hz a 0.5 Hz centrovaným kolem základní frekvence. Větší podíl na výsledku má jednoznačně filtr typu pásmová zádrž, jenž je navržen na utlumení síťového brumu v signálu. V grafu surových dat lze jednoduše pomocí kláves měnit rozsah časové osy grafu, tudíž je možné jednoduše analyzovat přítomnost artefaktů pomalých driftů signálu. Během měření mých dat jsem na tento artefakt narazil pouze výjimečně. Implicitní hodnota hornopropustního filtru 0.1 Hz vychází z doporučení a implicitního nastavení tohoto druhu filtru v [9]. Zejména v EEG záznamu EP (ERP) komponent však není nutné filtrovat data s tak nízkou frekvencí, pokud analyzujeme standardní, ne deviantní trialy. Je možné potom nastavit tento filtr např. na mezní frekvenci 1 Hz [57].

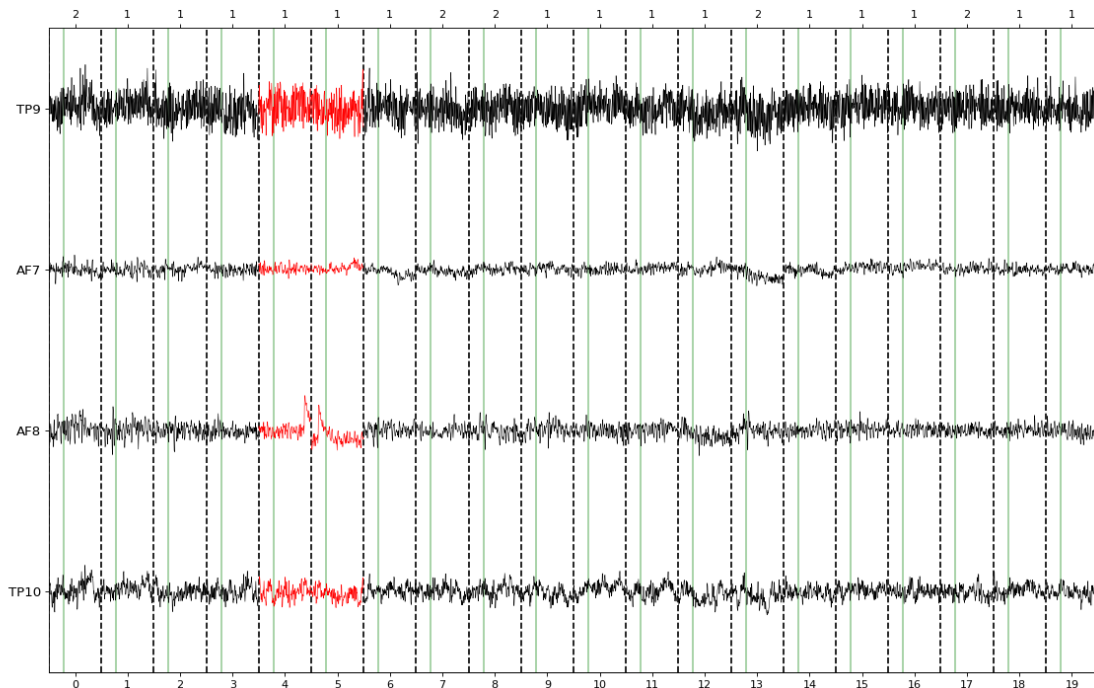


Obrázek 3.12. Srovnání nefiltrovaných surových dat (vlevo) s filtrovanými (vpravo), filtrem typu horní propust (0.1 Hz) a následně typem pásmová zádrž (50 Hz). Elektrody jsou pojmenovány shodně jako v předchozích případech, tedy postupně TP9, AF7, AF8 a TP10.

V následujícím logickém kroku je možné signály segmentovat do epoch. Mění se tím i objekt, který MNE-Python používá, tedy datová struktura *Epochs*, přičemž tento objekt vychází z objektu spojených surových dat *Raw*, který byl v programu používán doposud jako hlavní. Tato datová třída reprezentuje epochy popsané v předchozí kapitole a tím tedy reprezentuje data, která jsou časově spojena s opakovanými experimenty. Mezi tyto časová spojení patří např. počáteční čas stimulu nebo stisknutí tlačítka subjektem měření. Pomocí nástroje je možné segmentovat oba druhy dat doposud jmenované, tedy data s markery i bez. V knihovně MNE-Python jsou časové události extrahovány ze datové struktury surových dat *Raw*. Trigger kanál, mnohdy označovaný pomocí zkratk slova „Stimulation“, jako u Muse LSL „Stim“, umožňuje označit v datech okamžiky vzniku stimulu nebo reakce. V knihovně MNE-Python jsou tyto události označeny celými čísly a lze jim přiřadit názvy dle podmínek experimentu. V nástroji se tyto druhy epoch tvoří pomocí tlačítka *Marker Epochs* v boxu *Epoch* v hlavním okně nástroje. Po stisknutí tohoto tlačítka se zobrazí vyskakovací okno, ve kterém lze specifikovat další parametry segmentace, tedy např. které markery mají být použity k vytvoření událostí, dále je zde specifikace časového intervalu před a po náběhu stimulu. Lze zde také specifikovat použití a parametry korekce izolinie (baseline correction). V případě dat bez markerů lze využít segmentace do stejně dlouhých epoch např. při experimentech ve stavu odpočinku. Vytvořené epochy lze následně vykreslit v sekvenčním grafu, který je na obrázku 3.13. Protože jsou doposud dvě diskutované datové struktury odlišné, je v programu implementována funkce filtrování segmentovaných dat a také funkce pro převzorkování dat.

Kromě filtrování, které se používá po manuální detekci artefaktů v signálu, se segmentace do epoch v knihovně MNE-Python používá v souvislosti s automatickou detekcí artefaktů aktivity EOG a EKG. U Muse je velmi nepravděpodobné zachytit aktivitu EKG standardní elektrodou a je tak nutné spoléhat pro automatické odstranění těchto

artefaktů na nastavení prahu pro biologické rušení EEG elektrod. Aktivitu EOG je ovšem možné pozorovat pomocí standardních EEG elektrod, přičemž nejvíc patrná aktivita EOG je na elektrodách Muse za ušima (TP9 a TP10). Průběh aktivity EOG artefaktů, konkrétně mrkání, je viditelný na zmíněných elektrodách na obrázku 3.12, jako negativní skoky napětí v krátkém časovém intervalu.

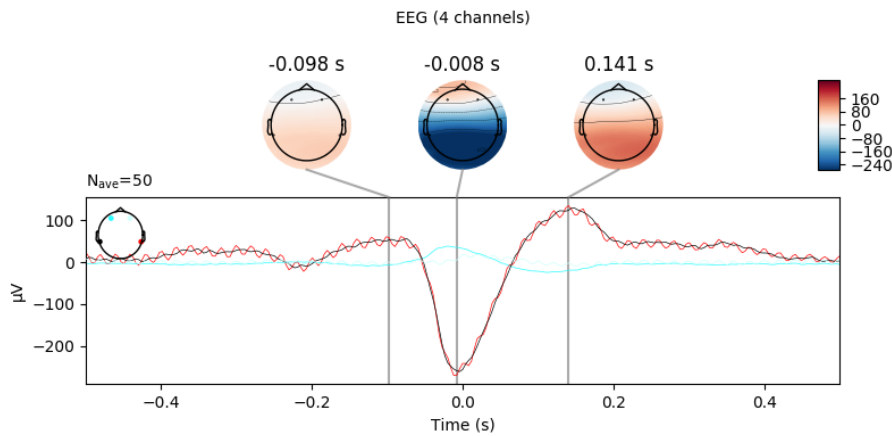


Obrázek 3.13. Graf časové řady segmentované do epoch, kde na horní ose jsou zobrazeny identifikátory markerů a na spodní ose jsou tyto události sekvenčně číslované. Možnost manuální anotace předvedena na epochách červeně vybarvených. Segmentace proběhla ze surových dat.

Pokud použijeme jednu z těchto elektrod k automatické identifikaci EOG artefaktů, můžeme vytvořit epochy s právě těmito artefakty. Existují dva způsoby, jak toho dosáhnout, přičemž jedním z nich je přiřadit jedné z elektrod, která nejlépe zachycuje EOG aktivitu, argument typu EOG elektrody. Tím však o tuto elektrodu v celkovém výčtu přijdeme. Preferovanou metodou použitou v nástroji je použití funkce vyšší úrovně, kterou knihovna poskytuje a která vytváří epochy a detekuje EOG události v jednom procesu. Jedná se však o experimentální funkci s nejistou úspěšností detekce.

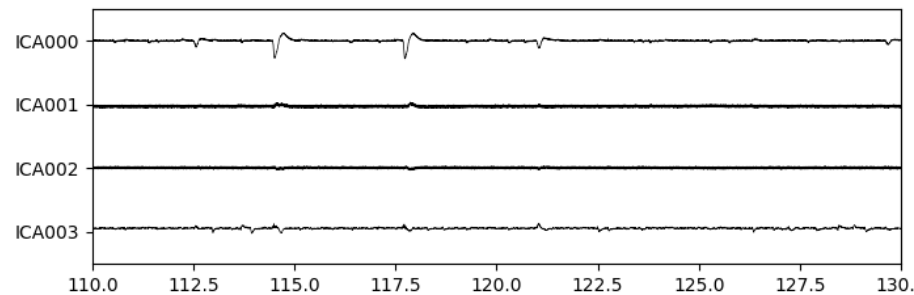
Posledním krokem předzpracování v mnou vybraném sledu metod je aplikace analýzy nezávislých komponent. Tuto metodu lze aplikovat na doposud oba typy datových struktur, tedy jak na spojitá surová data, a tedy i v počátečních krocích navrženého postupu předzpracování, tak i na segmentovaná data. ICA je v tomto programu implementovaná jako manuální metoda, tedy požaduje od uživatele volby parametrů a komponent. ICA je metoda korekce artefaktů, přičemž v implementaci jsem se soustředil na korekci EOG artefaktů. Protože je vhodné zobrazit aktivitu EOG artefaktů v signálu před začátkem analýzy, existuje v nástroji funkce vytváření epoch podle EOG událostí tlačítkem *EOG Epochs*, jehož funkce byla popsána v předchozím odstavci. Po vytvoření těchto epoch je vhodné takto automaticky detekované epochy zobrazit průměrováním epoch. Přípravou proměnných pro operaci vykreslování grafu vzniká datová struktura *Evoked*. Na obrázku 3.14 je vykreslen průběh EOG aktivity v průměrovaných

epochách. Tento graf také ukazuje tři lokální maxima v tzv. GFP, což je zjednodušeně řečeno směrodatná odchylka všech elektrod v čase [58].



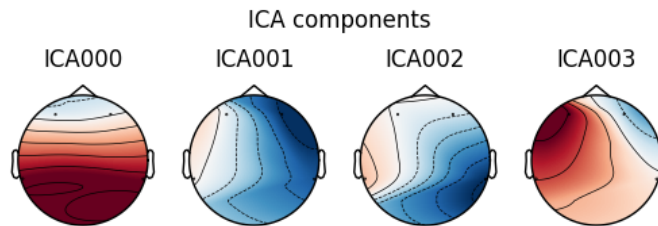
Obrázek 3.14. Souhrnný graf EOG aktivity na EEG elektrodách s topografickou mapou a třemi lokálními maximy v globálním poli výkonu (GFP)

Prvním doporučeným krokem je dle autorů [9] aplikace filtru typu horní propust s mezní frekvencí 1 Hz , což vede k potlačení pomalých driftů v signálu, které snižují nezávislost komponent, které jsou považovány za nezávislé zdrojové signály. Pomocí tlačítka *Run ICA* spustíme proces ICA dekompozice dat. V tomto kroku probíhá běh algoritmu PCA a bylo by možné vybrat počet prvních hlavních komponent pro ICA dekompozici. V případě Muse a za předpokladu, že počet elektrod odpovídá počtu hlavních komponent (i těch nezávislých), není vhodné redukovat dimenzi signálu, protože počet elektrod je nízký (4 až 5). Je zde také nastavena korekce nedeterminismu ICA dekompozice, což znamená, že v případě nepoužití korekce by mohlo dojít ke změně znamének u komponent při opakovaných spuštěních procesu. ICA algoritmy v knihovně MNE-Python ignorují časovou doménu, a tak lze aplikovat ICA na všechny tři druhy datových struktur. V nástroji je použit ICA algoritmus FastICA. Po dokončení běhu tohoto procesu je možné zobrazit několik užitečných grafů. Tlačítko *Plot sources* vyvolá graf časového průběhu nezávislých komponent, viz. obrázek 3.15. V případě tohoto grafu je možné vykreslovat nezávislé komponenty přímo z nefiltrovaných původních surových dat. Pokud uživatel klikne na průběh jednotlivých komponent, tak se tyto komponenty zařadí na seznam vyjmutých komponent. U Muse není vzhledem k nízkému počtu elektrod vhodné takto vyřazovat více jak jednu komponentu.



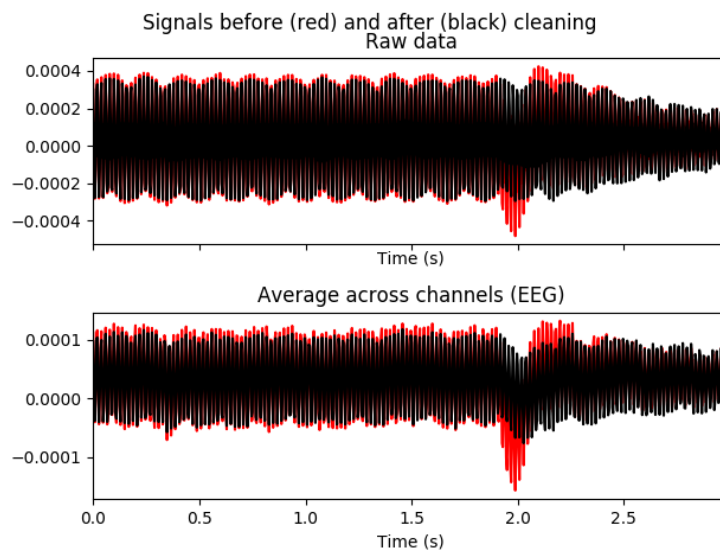
Obrázek 3.15. Graf časového průběhu nezávislých komponent

Všechny grafy v této sekci byly pořízeny ze stejného souboru dat, která jsem měřil za účelem zobrazení převážně artefaktů EOG a několika EMG aktivit, jako je pevné sevření čelisti a pohyb hlavou. Tyto data byly měřeny v běžných podmínkách a bez streamování stimulačního kanálu. Na obrázku 3.15 jsou zobrazeny jednotlivé nezávislé komponenty a je viditelné, že aktivita EOG se projevuje v tomto zobrazeném časovém okně především na první komponentě *ICA000*. Tento graf je důležitým pro počáteční vyhodnocení stavu po ICA dekompozici. Dalším grafem vykreslujeme distribuci komponent na skalpu, použitím tlačítka *Plot components*. Jsou to interpolované grafy na základě hodnot z inverzní transformační matice (tzv. demixovací matice).



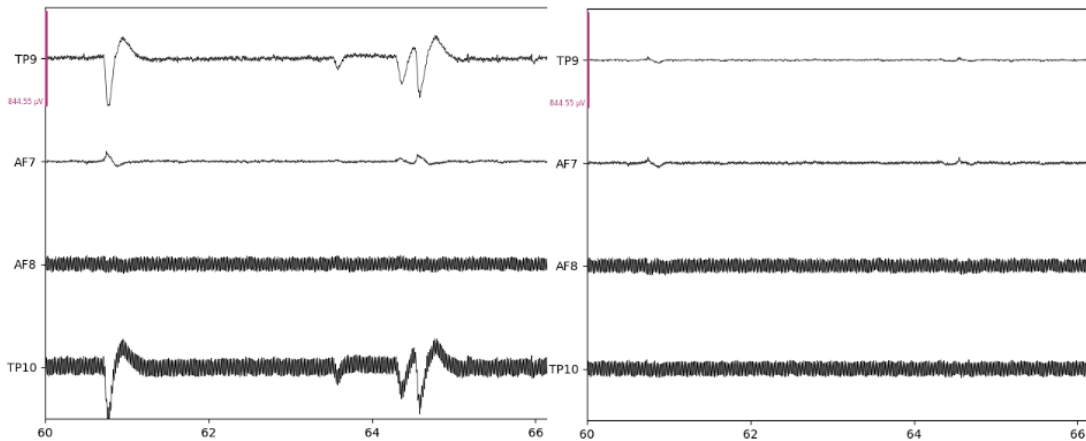
Obrázek 3.16. Graf rozdělení komponent na skalpu

Z obrázku je patrné, že z rozdělení komponent na není jednoduché určit, která komponenta má největší podíl na aktivitě EOG. Aktivita je kódována pomocí barev, modrá a červená ukazují aktivitu s opačnou amplitudou a méně výrazné barvy zobrazují aktivitu menší. První komponenta zmíněná v předchozím odstavci zobrazuje nejvíc aktivity v zadní partii hlavy, tedy tam, kde Muse měří aktivitu mrkání s nejvyšším potenciálem. Po kliknutí na konkrétní obrázek topografie (např. ICA000) vyskočí okno s podrobnostmi o signálu jako je rozptýl signálu, odhad PSD a barevně kódovaný obrázek epoch. Tyto nástroje slouží především jako vodítko k prvnímu seznámení se s artefakty v signálu. Teorii o tom, která z komponent nejvěrněji zachycuje aktivitu artefaktů lze zobrazit přesněji pomocí grafu srovnání původních surových signálů s korigovanými překrytím v jednom grafu. Je nutné zadat pořadové číslo komponenty a vykreslit graf pomocí tlačítka *Plot overlay*, viz. obrázek 3.17.



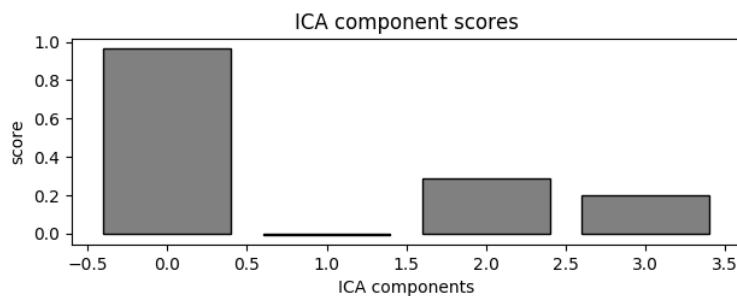
Obrázek 3.17. Graf porovnání mezi surovými a ICA korigovanými daty, v horním grafu příspěvky od všech elektrod, v dolním grafu jsou signály zprůměrované

V případě, že si je uživatel jistý výběrem komponenty k vyřazení, je možné stisknout tlačítko *Apply ICA* a vykreslit graf datové struktury, na kterou bylo aplikováno ICA. Protože jsem v tomto případě aplikoval ICA na spojitých surových datech, tedy na datové struktuře *Raw*, výsledným grafem s korigovanými signály je spojitý graf na obrázku 3.18. Na zpracování signálu před použitím ICA se podílel pouze filtr typu horní propust s mezní frekvencí 1 Hz. Na grafu je vidět ve dvou spodních elektrodách (AF8 a TP10) značný šum. Přesto byl ale algoritmus FastICA poměrně úspěšný a redukoval aktivitu EOG artefaktů v signálu.



Obrázek 3.18. Konečný graf porovnání mezi surovými a ICA korigovanými daty v čase

V knihovně MNE-Python existuje funkce k detekci komponent podobných EOG, která je založena na Pearsonově korelaci mezi filtrovanými daty a filtrovanými EOG kanály. Implicitně zadaný práh označující maximum, při jehož překročení jsou vlastnosti souboru klasifikovány jako anomálie, je při použití EEG elektrody namísto skutečné EOG elektrody nespolehlivý. Bráno jako experimentální funkce podobně jako v případě segmentace do epoch založeném na projevech EOG artefaktů v EEG signálu, jak jsem popsal výše, je možné zobrazit korelaci pro každou komponentu s EOG. Tato funkce se skrývá pod tlačítkem *Plot scores*, viz obrázek 3.19.



Obrázek 3.19. Graf skóreů souvisejících s detekovanými komponenty

Řešení ICA lze uložit a zpětně nahrát do programu s příponou *.fif* v nabídce *File*.

Kapitola 4

Závěr

Cílem diplomové práce bylo navrhnout a vytvořit nástroj pro předzpracování EEG signálů ze zařízení Muse. Tento nástroj měl mít grafické uživatelské prostředí a být schopen zobrazit živé průběhy dat z měření, vizualizovat a předzpracovat naměřená data a umožnit export dat do souborů různých formátů. Specifikace využití nástroje nebyla v zadání přesně stanovena, ale vzhledem k v práci popsaným komplikacím při snaze získat surová data z Muse do počítače k jejich analýze, se mi zdá pravděpodobné, že si nástroj uživatele najde a pomůže jim pracovat s čelenkou Muse na složitějších projektech.

V teoretické části práce jsem shrnul základy metod měření EEG signálů a využití EEG při měření různých fenoménů, které jsou v současnosti zkoumané nebo využíváné ve výzkumu, jako jsou EP (ERP) komponenty. Přes to je možné nezabývat se pouze těmito druhy dat a vyzkoušet studium relaxace, únavy či snad spánku, a vyzkoušet aplikaci těchto poznatků např. na detekci únavy. Dále byly předloženy základní metody předzpracování EEG signálu a diskutovány jejich přínosy a problémy při aplikaci na datech. Zároveň jsou zde uvedeny metody časové i frekvenční analýzy, stejně tak jako některé pokročilejší metody předzpracování, jako je např. ICA. Ačkoliv není téma předzpracování signálů tak obsáhlé jako soubor předzpracování a analýzy a např. ještě lokalizace signálu a jak je to častější v zadáních kvalifikačních prací, bylo možné ukázat na značnou důležitost tohoto uceleného kroku v celkovém procesu zpracování EEG dat.

V aplikační části práce jsem rozebíral hardwarové vybavení, se kterým musí program fungovat a vytěžit z něj maximum informací. Přesto, že se v případě Muse jedná o relativně levný měřicí přístroj, je možné s ním dosáhnout kvalitních výsledků při výzkumu alespoň dvou EP (ERP) paradigm, viz. [56]. Nicméně je pravdou, že počet elektrod Muse je omezujícím faktorem v dalších zajímavých výzkumech, přičemž přínosným by byla výroba a použití přídavné elektrody, která by byla schopna měřit i další signály jako EOG či EKG. V této části jsem také popisoval, jak se mi podařilo naprogramovat nástroj, který využívá v reálném světě používanou metodu přenášení dat v síti (LSL) pro laboratorní účely a specificky pro přístroje měřící EEG.

Nástroj má přidanou hodnotu v podobě offline zpracování naměřených signálů v jednom prostředí jako nahrávání a sledování živých průběhů signálů. Je tak možné si spolu se zapojením knihovny EEG-Notebooks vyzkoušet pokusy s EP (ERP) díky prezentačním stimulům, nahrávat a zároveň sledovat kvalitu nahrávaných dat. Dále je také možné pomocí nástroje analyzovat naměřený signál v časové i frekvenční oblasti, a to v různých stadiích procesu předzpracování. Jsou zde aplikovány klasické metody předzpracování jako je filtrování či změna reference signálu. Pro předpoklady správného statistického zpracování trialů je možné signál segmentovat do epoch (trialů) a dále s nimi nakládat. Povedlo se mi také ověřit funkčnost korekce artefaktů v signálu pomocí jednodušší metody závislé na očekávaných amplitudách signálu, ale i pokročilejší jako je analýza nezávislých komponent (ICA).

Literatura

- [1] PENHAKER, Marek. *Lékařské diagnostické přístroje*. První vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2004. ISBN 80-248-0751-3.
- [2] KRAJČA, Vladimír a Jitka MOHYLOVÁ. *Číslíkové zpracování neurofyziologických signálů*. V Praze: České vysoké učení technické, 2011. ISBN 978-800-1047-217.
- [3] BIGLARI, Habibeh, Alireza REZAYI, Hamid BIGLARI, Mehdi ALIZADEH a Farzad MD. Relationship Between Migraine and Abnormal EEG Findings in Children. *Iranian journal of child neurology*. 03, 2012, ročník 6, s. 21-24.
- [4] SOBOTKA, Pavel. *Patologická fyziologie*. 4., upr. vyd vyd. Praha: Karolinum, 2012. ISBN 978-80-246-2128-9.
- [5] BAREŠ, Martin. Kognitivní evokované potenciály. *Česká a slovenská neurologie a neurochirurgie*. 2011, ročník 5, s. 508-517 [vid. 2020-05-25]. Dostupné též na <https://bit.ly/34sNHF5>.
- [6] COHEN, M.X. *Analyzing Neural Time Series Data: Theory and Practice*. MIT Press, 2014. Issues in Clinical and Cognitive Neuropsychology. ISBN 978-0-262-01987-3.
- [7] INTERAXON INC. *The World's Leading EEG Device - Meditation Made Easy* [online]. [vid. 2020-05-25]. Dostupné též na <https://web.archive.org/web/20200102093919/https://choosemuse.com/muse/>.
- [8] BARACHANT, Alexandre, Dano MORRISON, Hubert BANVILLE, Jason KOWALESKI, Uri SHAKED, Sylvain CHEVALLIER a Juan Jesús Torre TRESOLS. *muse-lsl*. [vid. 2020-05-25]. Dostupné na DOI 10.5281/zenodo.3228861.
- [9] GRAMFORT, Alexandre, Martin LUESSI, Eric LARSON, Denis ENGEMANN, Daniel STROHMEIER, Christian BRODBECK, Roman GOJ, Mainak JAS, Teon BROOKS, Lauri PARKKONEN a Matti HÄMÄLÄINEN. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*. 2013, ročník 7, s. 267 [vid. 2020-05-25]. ISSN 1662-453X. Dostupné na DOI 10.3389/fnins.2013.00267. Dostupné též na <https://www.frontiersin.org/article/10.3389/fnins.2013.00267>.
- [10] KOTHE, Christian, David MEDINE, Chadwick BOULAY, Matthew GRIVICH a Tristan STENNER. *Lab Streaming Layer Documentation - Quick Start*. [vid. 2020-05-25]. Dostupné též na https://labstreaminglayer.readthedocs.io/info/getting_started.html.
- [11] SANEI, S. a J.A. CHAMBERS. *EEG Signal Processing*. Wiley, 2008. ISBN 978-0-470-02581-9.
- [12] GLAJCAR, Radek a Radek JANČA. Možnosti detekce SOZ v intrakraniálním EEG signálu. In: *Analýza a zpracování řečových a biologických signálů*. Praha: ČVUT, 2010. s. 29-33. ISBN 978-80-01-04680-7.
- [13] KLONOWSKI, Włodzimierz. Everything you wanted to ask about EEG but were afraid to get the right answer. *Nonlinear biomedical physics*. BioMed Central, 2009, ročník 3, č. 1, s. 1-5. ISSN 1753-4631.

- [14] *Chapter 2 - Technological Basics of EEG Recording and Operation of Apparatus*. [vid. 2020-05-25]. Dostupné na DOI 10.1016/B978-0-12-804490-2.00002-6.
- [15] GRIFFITHS, John, Jadin TREDUP a NEUROTECHX. *EEG-Notebooks - Democratizing the cognitive neuroscience experiment*. [vid. 2020-05-25]. Dostupné též na <https://neurotechx.github.io/eeg-notebooks/>.
- [16] BARACHANT, Alexandre. *P300 with Muse EEG headband*. [vid. 2020-05-25]. Dostupné též na <https://alexandre.barachant.org/blog/2017/02/05/P300-with-muse.html>.
- [17] ŠEBEK, Jakub. *Funkční konektivita zdrojů EEG pro časově vázané události*. Praha: ČVUT, 2018 [vid. 2020-05-25]. Diplomová práce. Dostupné též na <http://hdl.handle.net/10467/80722>.
- [18] FREEMAN, Walter J. a Rodrigo Quian QUIROGA. *Imaging Brain Function With EEG*. Springer New York, 2013 [vid. 2020-05-25]. ISBN 978-1-4614-4984-3. Dostupné na DOI 10.1007/978-1-4614-4984-3.
- [19] DORFER, Thomas A. *Artefact Correction with ICA*. [vid. 2020-05-25]. Dostupné též na <https://towardsdatascience.com/artefact-correction-with-ica-53afb63ad300>.
- [20] GRAMFORT, Alexandre, Martin LUESSI a Eric LARSON. *Compute the power spectral density of raw data*. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/2Pto7r4>.
- [21] LUCK, S.J. *An Introduction to the Event-Related Potential Technique*. MIT Press, 2014 [vid. 2020-05-25]. A Bradford Book. ISBN 978-0-262-52585-5.
- [22] HOLČÍK, Jiří a Martin KOMENDA. *Matematická biologie: e-learningová učebnice*. Masarykova univerzita, 2015 [vid. 2020-05-25]. ISBN 978-80-210-8095-9. Dostupné též na <https://portal.matematickabiologie.cz/>.
- [23] CHEVEIGNÉ, Alain de a Dorothée ARZOUNIAN. Robust detrending, rereferencing, outlier detection, and inpainting for multichannel data. *Neuroimage*. Elsevier, 2018, ročník 172, s. 903–912 [vid. 2020-05-25]. Dostupné na DOI 10.1016/j.neuroimage.2018.01.035.
- [24] MNE DEVELOPERS. *Filtering and resampling data*. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/2DIEbt2>.
- [25] HYVARINEN, Aapo, Juha KARHUNEN a Erkki OJA. *Independent Component Analysis*. New York: J. Wiley, 2001. ISBN 0-471-22131-7.
- [26] HYVÄRINEN, Aapo a Erkki OJA. Independent component analysis: algorithms and applications. *Neural networks*. Elsevier, 2000, ročník 13, č. 4-5, s. 411–430 [vid. 2020-05-25]. Dostupné na DOI 10.1016/S0893-6080(00)00026-5.
- [27] HOLČÍK, Jiří. *Analýza a klasifikace dat*. Brno: CERM, 2012. ISBN 0978-80-7204-793-2.
- [28] ZEMAN, Tomas. *BSS - Preprocessing Steps for Separation Improvement*. [vid. 2020-05-25]. Dostupné též na <http://noel.feld.cvut.cz/vyu/prj-czs-asi/zeman/pca.pdf>.
- [29] SHLENS, Jonathon. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*. 2014. [vid. 2020-05-25]. Dostupné též na <https://arxiv.org/pdf/1404.1100.pdf>.

- [30] INTERAXON INC. *Meditation Reimagined: Introducing Muse 2* [online]. [vid. 2020-05-25]. Dostupné též na <https://choosemuse.com/muse-2/>.
- [31] NEUROTECHX. *Consumer EEG Headsets* [online]. [vid. 2020-05-25]. Dostupné též na <http://learn.neurotechedu.com/headsets/>.
- [32] INTERAXON INC. *Technical specification, validation and research use* [online]. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/3kFhYGC>.
- [33] SAUER, Andrew. *Muse EEG Headset: Making Extra Electrode* [online]. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/3ar2ocU>.
- [34] EU-LIGHTS. *Muse: The Brain Sensing Headband*. [vid. 2020-05-25]. Dostupné též na <https://eu-lights.com/wp-content/uploads/2019/01/MU-02-BK-EN.jpg>.
- [35] INTERAXON INC. *Muse Translates Your Brainwaves Into the Guiding Sounds of Weather* [online]. [vid. 2020-05-25]. Dostupné též na <https://choosemuse.com/how-it-works/>.
- [36] INTERAXON INC. *Developer Kit* [online]. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/2XWRoiz>.
- [37] INTERAXON INC. *Letter for developers* [online]. [vid. 2020-05-25]. Dostupné též na <https://choosemuse.com/development/>.
- [38] DEMOS, J.N. *Getting Started with EEG Neurofeedback (Second Edition)*. New York: W. W. Norton, 2019 [vid. 2020-05-25]. ISBN 978-0-39371-254-4.
- [39] GRIFFITHS, John a Dano MORRISON. *Using an extra electrode with Muse* [online]. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/3aqVpAD>.
- [40] ZANDER, Thorsten O a Christian KOTHE. Towards passive brain-computer interfaces: applying brain-computer interface technology to human-machine systems in general. *Journal of Neural Engineering*. IOP Publishing, mar, 2011, ročník 8, č. 2, s. 025005 [vid. 2020-05-25]. Dostupné na DOI 10.1088/1741-2560/8/2/025005.
- [41] PYTHON SOFTWARE FOUNDATION. *General Python FAQ* [online]. [vid. 2020-05-25]. Dostupné též na <https://docs.python.org/3/faq/general.html>.
- [42] DEPARTMENT OF ATMOSPHERIC AND OCEANIC SCIENCES UW. *Matlab Introduction* [online]. [vid. 2020-05-25]. Dostupné též na <https://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>.
- [43] RIVERBANK COMPUTING LTD. *What is PyQt?* [online]. [vid. 2020-05-25]. Dostupné též na <https://riverbankcomputing.com/software/pyqt/intro>.
- [44] PYQTGRAPH. *Scientific Graphics and GUI Library for Python* [online]. [vid. 2020-05-25]. Dostupné též na <http://www.pyqtgraph.org/>.
- [45] DELORME, Arnaud a Scott MAKEIG. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*. Elsevier, 2004, ročník 134, č. 1, s. 9-21 [vid. 2020-05-25]. Dostupné na DOI 10.1016/j.jneumeth.2003.10.009.

- [46] MNELAB. *MNELAB Graphical User Interface* [online]. [vid. 2020-05-25]. Dostupné též na <https://raw.githubusercontent.com/cbrnr/mnelab/master/mnelab.png>.
- [47] DELORME, Arnaud. *EEGLAB Graphical User Interface* [online]. [vid. 2020-05-25]. Dostupné též na https://sccn.ucsd.edu/mediawiki/images/5/57/Eeglab_small.jpg.
- [48] KOTHE, Christian A., Tristan STENNER, Chadwick BOULAY a et. AL. *Lab Streaming Layer* [online]. [vid. 2020-05-25]. Dostupné též na <https://github.com/sccn/labstreaminglayer>.
- [49] KOTHE, Christian A. *Demo 1: The Lab Streaming Layer* [online]. [vid. 2020-05-25]. Dostupné též na ftp://sccn.ucsd.edu/pub/bcilab/lectures/Demo_1_The_Lab_Streaming_Layer.pdf.
- [50] SCHMEDER, Andrew, Adrian FREED a David WESSEL. Best Practices for Open Sound Control. In: *Linux Audio Conference*. Utrecht, NL, 2010.
- [51] RUČKA, Lukáš. *Implementace protokolu TUIO 2.0*. Masarykova univerzita, Fakulta informatiky, Brno, 2012 [vid. 2020-05-25]. Bakalářská práce. Dostupné též na https://is.muni.cz/th/tylmo/pb_rucka.pdf.
- [52] CLUSTERBUCK, James. *OSC Specification* [online]. [vid. 2020-05-25]. Dostupné též na <https://mind-monitor.com/FAQ.php>.
- [53] KEMP, Bob a Jesus OLIVAN. European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*. 10, 2003, ročník 114, s. 1755-61 [vid. 2020-05-25]. Dostupné na DOI 10.1016/S1388-2457(03)00123-8.
- [54] PYQTGRAPH. *Scientific Graphics and GUI Library for Python* [online]. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/3b1nzmf>.
- [55] VIKTORIN, Petr a Miro HRONČOK. *GUI v Pythonu: PyQt5* [online]. [vid. 2020-05-25]. Dostupné též na <https://naucse.python.cz/course/mi-pyt/intro/pyqt/>.
- [56] KRIGOLSON, Olave E, Chad C WILLIAMS, Angela NORTON, Cameron D HASSALL a Francisco L COLINO. Choosing MUSE: Validation of a low-cost, portable EEG system for ERP research. *Frontiers in neuroscience*. Frontiers, 2017, ročník 11, s. 109.
- [57] GRAMFORT, Alexandre, Martin LUESSI a Eric LARSON. *Background information on filtering*. [vid. 2020-05-25]. Dostupné též na <https://bit.ly/2ZbL7jt>.
- [58] MURRAY, Micah M, Denis BRUNET a Christoph M MICHEL. Topographic ERP analyses: a step-by-step tutorial review. *Brain topography*. Springer, 2008, ročník 20, č. 4, s. 249–264 [vid. 2020-05-25]. Dostupné na DOI 10.1007/s10548-008-0054-5.

Příloha A

Zkratky

EEG	Elektroencefalografie
EOG	Elektrookulografie
EMG	Elektromyografie
EKG	Elektrokariografie
EP (ERP)	Evokované potenciály (Kognitivní evokované potenciály), Evoked Potential (Event-related potential)
GUI	Grafické uživatelské prostředí, Graphical User Interface
LSL	Lab Streaming Layer
SSVEP	Ustálený vizuální evokovaný potenciál, Steady State Visually Evoked Potential
IFCN	Mezinárodní federace klinické neurofyzologie, International Federation of Clinical Neurophysiology
DFT	Diskrétní Fourierova transformace, Discrete Fourier Transform
FFT	Rychlá Fourierova transformace, Fast Fourier Transform
PSD	Spektrální výkonová hustota, Power Spectral Density
FIR	Konečná impulzní odezva, Finite Impulse Response
IIR	Nekonečná impulzní odezva, Infinite Impulse Response
MA	Klouzavý průměr, Moving Average
ICA	Analýza nezávislých komponent, Independent Component Analysis
PCA	Analýza hlavních komponent, Principle Component Analysis
EVD	Rozklad pomocí vlastních čísel, Eigenvalue Decomposition
SVD	Rozklad pomocí singulárních čísel, Singular Value Decomposition
BCI	Rozhraní Mozek-Počítač, Brain-Computer Interface
OSC	Open Sound Control
SDK	Software Development Kit
BLE	Bluetooth Low Energy
API	Rozhraní pro programování aplikací, Application Programming Interface
SCCN	Swartz Center for Computational Neuroscience
GATT	Generic Attribute Profile
BGAPI	BlueGiga API
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
CLI	Příkazový řádek, Command Line Interface
EDF	European Data Format

Příloha B

Obsah CD

Práce je napsána v PlainTeX šabloně CTUStyle. Generování .pdf souboru se provádí pomocí příkazu pdfcsplain, který je součástí distribuce TeXLive.

```
./
|--cd_dir_tree.txt
|--DP.pdf
|--zadani.pdf
|--program
|-- |--data
|-- |--dp_main.py
|-- |--dp_prep.py
|-- |--download_data_from_gdrive.py
|-- |--pylsl_dummy_stream.py
|-- |--requirements.txt
|--tex
```