

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-104376-86447

RIADENIE MOBILNÉHO ROBOTA POMOCOU EEG
SIGNÁLOV MOZGU
DIPLOMOVÁ PRÁCA

Študijný program:	Robotika a kybernetika
Študijný odbor:	9.2.7 Kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vedúci záverečnej práce/školiťel:	prof. Ing. Peter Hubinský, PhD.

Bratislava 2020

Bc. Dávid Polák

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-104376-86447

**RIADENIE MOBILNÉHO ROBOTA POMOCOU EEG
SIGNÁLOV MOZGU
DIPLOMOVÁ PRÁCA**

2020

Bc. Dávid Polák



ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Dávid Polák**
ID študenta: 86447
Študijný program: robotika a kybernetika
Študijný odbor: kybernetika
Vedúci práce: prof. Ing. Peter Hubinský, PhD.
Miesto vypracovania: Ústav robotiky a kybernetiky

Názov práce: **Riadenie mobilného robota pomocou EEG signálov mozgu**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Témou práce je návrh riadenia pohybu mobilného robota pomocou EEG signálov mozgu. Cieľom je dosiahnuť jeho intuitívnejšie a presnejšie riadenie v porovnaní so štandardnými spôsobmi ovládania pomocou rôznych typov elektromechanických ovládačov.

Úlohy:

1. Naštudujte problematiku spracovania EEG signálov.
2. Navrhните a implementujte algoritmus, ktorá dokáže kategorizovať (rozpoznať) špecifické EEG signály v reálnom čase.
3. Navrhните a implementujte komunikáciu medzi riadiacim systémom a mobilným robotom
4. Verifikujte navrhnuté riešenie a zhodnot'te dosiahnuté výsledky.

Riešenie zadania práce od: 23. 09. 2019

Dátum odovzdania práce: 15. 05. 2020

Bc. Dávid Polák
študent

prof. Ing. Jarmila Pavlovičová, PhD.
vedúci pracoviska

prof. Ing. Jarmila Pavlovičová, PhD.
garantka študijného programu

Pod'akovanie

Touto cestou by som chcel pod'akovať tím, ktorí mi pomohli vo vypracovaní mojej záverečnej práce. Moje pod'akovanie patrí hlavne môjmu vedúcemu práce prof. Ing. Peter Hubinský, PhD. a vďačný som hlavne za jeho podporu a cenné pripomienky k diplomovému projektu.

ANOTÁCIA DIPLOMOVÉHO PRÁCE

Slovenská technická univerzita v Bratislave
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný odbor: 9.2.7 Kybernetika

Študijný program: Robotika a kybernetika

Autor: Bc. Dávid Polák

Diplomový projekt 1: Riadenie mobilného robota pomocou EEG signálov mozgu

Vedúci diplomového projektu 1: prof. Ing. Peter Hubinský, PhD.

Mesiac, rok odovzdania: Máj, 2020

Kľúčové slová: EEG, Mozog, Riadenie mobilného robota, BCI, kNN, DTW, Neurónové siete

Cieľom diplomovej práce bolo vytvoriť GUI aplikáciu, ktorá dokáže zaznamenať, zatriediť EEG signály a následne s pomocou rôznych klasifikačných metód roztriediť ich v reálnom čase. Následne, bude schopný aplikovať tieto príkazy na riadenie mobilného robota. Prvá kapitola sa venuje histórii EEG signálov, ako sa tvoria v mozgu a s topológiou hlavy. Po osvojení týchto znalostí nás oboznámi s viacerými typmi BCI prístrojov. Druhá kapitola sa venuje analýze signálov a ich filtrácii. V tejto časti je širšie rozpísané, že na aké šumy si treba dávať pozor a že kvôli čomu sa môžu pohoršiť naše výsledky pri zlej filtrácii. Po analýze a aplikácie týchto vedomostí pokračuje s ďalšou kapitolou, ktorá sa zaoberá s klasifikačnými metódami. Tejto časti sú predstavené dve metódy, ktoré sú klasifikačné neurónové siete a k-nearest neighbors s kombináciou dynamic time warping. Po teórii, práca implementuje túto časť a porovnáva spomínané dve metódy, aby sme sa dozvedeli, ktorá produkuje presnejšie výsledky. Posledná kapitola sa zaoberá len so samotným programom, ktorý bol vytvorený v procese. V danej časti je podrobne popísané, ako funguje a s akým spôsobom boli riešené jednotlivé problematiky. Práca podáva komplexný pohľad na vytvorenie systému, s ktorým sa dá jednoducho riadiť akýkoľvek prístroj s pomocou mozgových vln.

MASTER THESIS ABSTRACT

Slovak University of Technology in Bratislava
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION
TECHNOLOGY

Branch of Study: 9.2.7 Cybernetics

Study Programme: Robotics and Cybernetics

Author: Bc. Dávid Polák

Master Thesis: Mobile robot control using brain EEG signals

Supervisor: prof. Ing. Peter Hubinský, PhD.

Year, Month: 2020, May

Keywords: EEG, brain, Mobile robot control, BCI, kNN, DTW, Neural network

The aim of the diploma thesis was to create a GUI application that can record, classify EEG signals and then use the various classification methods to categorize them in real time. Subsequently, it will be able to apply these commands to control the mobile robot. The first chapter deals with the history of EEG signals, how they are formed in the brain and with the topology of the head. After mastering this knowledge, they will acquaint us with several types of BCI devices. The second chapter deals with the analysis of signals and their filtering. In this section, we get a better understanding of the noises we have to watch out for even eliminate and which noises can cause poor results when the filters are not correctly applied. After analyzing and applying this knowledge, it proceeds with the next chapter which deals with classification methods. In this section, two methods are presented, which are classification neural networks and k-nearest neighbors with a combination of dynamic time warping. Following the theory, the work implements knowledge and compares the two methods mentioned to find out which produces more accurate results. The last chapter deals only with the program itself, which was created in the process. This section describes in detail how it works and how the individual issues were addressed. The work provides a comprehensive view on how to create a system with which any device can be easily controlled with the help of brainwaves.

Obsah

Zoznam obrázok	9
Zoznam použitých skratiek	10
Úvod	11
1 Všeobecný prehľad EEG signálov a BCI	12
1.1 História EEG signálov	12
1.2 Vznik elektrické aktivity v mozgu	13
1.3 Topológia hlavy a umiestnenie senzorov	16
1.4 BCI systémy	17
2 Analýza EEG signálov	20
2.1 Butterworthov filter	20
2.2 Spracovanie signálov	21
3 Klasifikačné metódy	24
3.1 Dynamic time warping	25
3.2 Klasifikátor K-najbližších susedov	27
3.3 Neurónové siete	28
3.3.1 Model umelého neurónu	29
3.3.2 Aktivačná funkcia	30
3.3.3 Trénovanie neurónových sietí	31
3.3.4 Implementácia neurónového siete	32
3.4 Výsledky porovnania kategorizačných metód	34
4 GUI pre riadenie a trénovanie	37
4.1 Základné funkcie vytvorenej aplikácie	37
4.2 Vytvorenie trénovacích dát pomocou GUI	38
4.3 Trénovanie neurónovej siete a vytvorenie vzorcov pre kNN+DTW	40
4.4 Riadenie mobilného robota pomocou EEG signálov	43
4.5 Výsledky riadenia mobilného robota	45

Záver	47
Literatúra	48
Prílohy	50
Príloha A: Funkcia Dynamic Time Warping	50
Príloha B: Výsledky z klasifikácie neurónových sietí a kNN+DTW	51

Zoznam obrázok

Obr. 1. Hans Berger (1873-1941)	12
Obr. 2. Rozdelenie mozgovej kôry	13
Obr. 3. Základné typy EEG signálov	14
Obr. 4. Topologická mapa hlavy	16
Obr. 5. Vrstvy hlavy a typy BCI systémov	18
Obr. 6. Mikro-elektrody invazívnej metódy	18
Obr. 7. ECoG elektrody čiastočne invazívnej metódy	19
Obr. 8. Graf časovo priestorového rozlíšenia	19
Obr. 9. Porovnanie troch rôznych filtrov	21
Obr. 10. Surové dáta získané pomocou OpenBCI	22
Obr. 11. Butterworthov pásmový filter medzi 5-50 Hz	23
Obr. 12. Dáta získané po aplikácii filtrov	23
Obr. 13. Reprezentácia ED na grafe a), reprezentácia DTW na grafe b)	26
Obr. 14. Graf zobrazujúci tri rôzne časové rady (t_1 , t_2 , t_3)	27
Obr. 15. Algoritmus K-najbližších susedov + DTW výpočet vzdialenosti	28
Obr. 16. Model umelého neurónu	29
Obr. 17. Aktivačné funkcie použité pre klasifikačné siete	30
Obr. 18. Zobrazenie signálov prichádzajúce z prístroja OpenBCI	38
Obr. 19. Vytváranie záznamov na senzorické ovládanie	39
Obr. 20. Vytváranie záznamov na motorické ovládanie	40
Obr. 21. Sekcia pre tréning neurónovej siete	41
Obr. 22. Sekcia pre vytvorenie vzorcov pre kNN+DTW	42
Obr. 23. Sekcia pre riadenie virtuálneho robota	44

Zoznam použitých skratiek

EEG	Elektroencefalogram
EPSP	Excitačné postsynaptické potenciály
IPSP	Inhibičné postsynaptické potenciály
BCI	Brain Computer Interface
ECoG	Elektrokortikografia
MEG	Magnetoencephalography
PET	Positron Emission Tomography
fMRI	Funkčné magnetické rezonančné zobrazovanie
fNIRS	Near-infrared spectroscopy
DTW	Dynamic Time Warping
ED	Euclidean Distance
kNN	k-Nearest Neighbors

Úvod

V dnešnej dobe narastá komunikácia medzi človekom a počítačom a práve kvôli zefektívňovaniu tejto činnosti sú vytvárané nové typy periférnych zariadení. Jedným takýmto prístrojom je BCI, ktorý sa dostáva do širšieho povedomia. Princípom tejto komunikácie je, aby dané zariadenie nebolo ovládané motorickým pohybom rúk, nôh alebo hlasom človeka, ale pomocou impulzov generovaný mozgom resp. s pomocou nervového systému človeka. Táto interakcia medzi počítačom a človekom je už vyššia forma komunikácie, ktorá môže byť rýchlejšia aj pohodlnejšia ako súčasne existujúce periférne zariadenia napr. klávesnica, počítačová myš alebo pákové a hlasové ovládače.

Danú problematiku sme zvolili kvôli tomu, lebo v dnešnej dobe sa BCI používajú viac-menej len na zbieranie informácií o mozgových procesoch človeka a neaplikujú sa ako ovládacie zariadenie. Preto je prvotným cieľom tejto práce urobiť databázu signálov z celého mozgu, aby sa takto získané údaje mohli ľahšie analyzovať. S pomocou databázy bude možné nájsť tie dominantné signály, ktoré sú jedinečné a budú jednoducho rozoznateľné od ostatných signálov. Tieto signály prispievajú k nášmu základu, ako kategórie jednotlivých príkazov - jednoducho povedané tieto signály budú našimi „tlačidlami“. Aby tieto tlačidlá fungovali podľa našich predstáv, musíme zvoliť vhodnú formu klasifikácie, ktorá nám roztriedi naše signály. Jednotlivé mozgové aktivity meriame na povrchu hlavy. Komplikácia pri triedení môže nastať kvôli nezhodujúcim sa signálom pre samostatné mozgové aktivity. Nastáva to z toho dôvodu, že mozog pracuje paralelne na viacerých problémoch. Tvorí to tzv. šum, ktorý nám bráni pri absolútnej klasifikácii vstupných signálov. V ideálnom prípade by mal mať stroj, ovládaný pomocou BCI, správne klasifikovaných 80 - 90% signálov.

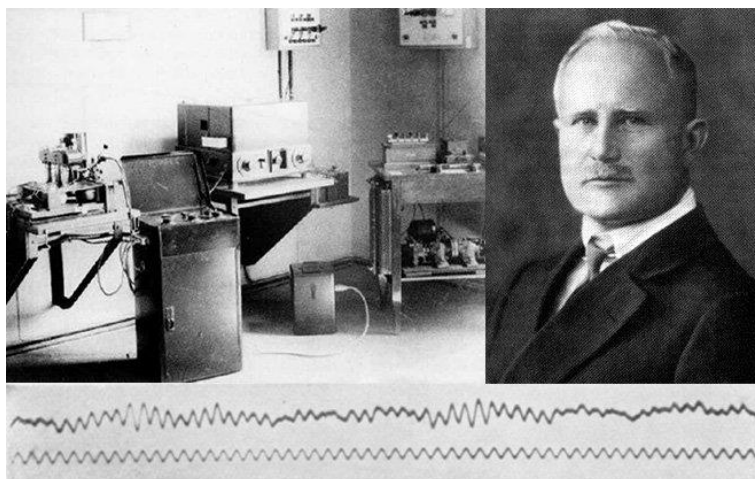
Ďalšia časť práce sa zaoberá naprogramovaním komunikácie medzi počítačom a mobilným robotom. V tejto časti už program bude posielat' klasifikované príkazy na vykonanie činnosti stroja v reálnom čase. Vo finálnej fáze bude uskutočnená validácia funkcionality práce resp. vykoná sa daný počet testov na dobrovoľníkoch. Tieto nám poskytnú spätnú väzbu o kvalite programu a jeho funkčnosti.

1 Všeobecný prehľad EEG signálov a BCI

Pred analýzou mozgových signálov musíme pochopiť hardvérovú časť, ktorú v našom prípade tvorí: *mozog, nervová sústava, senzory a prístroj na zbieranie mozgových vln*. Práve preto začneme históriou EEG, v rámci čoho sa dozvieme kto a ako ich zaznamenal prvýkrát na svete. Z hľadiska mozgu sa dozvieme, kde a ako sa tvoria EEG signály. Ďalšou časťou je preskúmanie topológie hlavy, z čoho dozvieme, kde je potrebné umiestniť jednotlivé senzory pre zvýšenie úspešnosti nálezu hľadaných vln. Posledná sekcia tejto kapitoly zhrnie jednotlivé typy BCI systémov, ktorá slúžia na zbieranie a zosilnenie požadovaných resp. predpokladaných signálov.

1.1 História EEG signálov

Hans Berger (Obr. 1) bol prvým človekom, ktorý zaznamenal elektrickú aktivitu ľudského mozgu v roku 1924. Tieto mozgové vlny zachytil pomocou Mannovho strunového galvanometra, ktorý mal citlivosť 1 mV/cm . Po dvoch rokoch, v roku 1926 si zaobstaral presnejší prístroj od firmy Siemens s vyššou citlivosťou – $130 \mu\text{V/cm}$. Po týchto rokoch uskutočnil desiatky experimentov, s pomocou čoho zdokonaľoval používané senzory a techniky merania. Prvú správu o mozgových vlnách publikoval v roku 1929, kde písal o alfa a beta vlnách. Tu definoval prvýkrát mozgové vlny ako „electroencephalogram“ signály (EEG).^[1]



Obr. 1. Hans Berger (1873-1941)

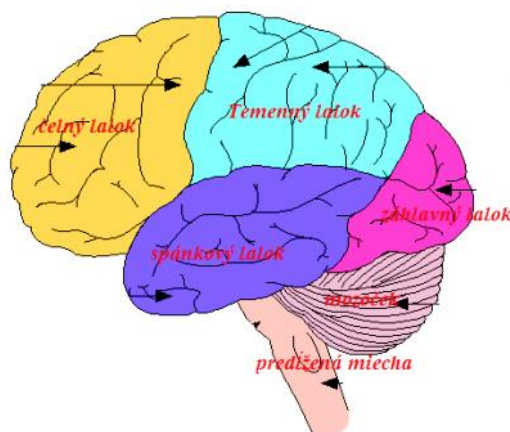
Po Bergerovej publikácii vo výskume mozgovej aktivity spravili významný pokrok Fisher a Lowenback. Prvýkrát sa realizovala demonštrácia epileptických hrotov (spikes) v roku 1934. V ďalších rokoch vznikli výskumné inštitúcie aj v Amerike a v Európe, ktoré

experimentovali a zbierali údaje o mozgových aktivitách. Eberhard Fetz - v roku 1969 – použil EEG signály prvýkrát používal EEG spätnoväzobné riadenie robotického ramena. ^[1]

1.2 Vznik elektrické aktivity v mozgu

Elektrický prúd vzniká v centrálnom nervovom systéme pri polarizácii neurónov v danom časovom intervale, ktoré sú zaznamenané ako EEG signály. Tieto aktivity sa objavujú na povrchu kože lebky, kde sú merateľné ako signály s úrovňou desiatok μV . Hlavnými zdrojmi elektrickej aktivity mozgu sú akčné potenciály a to *excitačné* a *inhibičné postsynaptické potenciály* (EPSP a IPSP). ^[2]

Na kôrových neurónoch sa nachádzajú akčné potenciály a hromadné excitačné postsynaptické potenciály, ktoré sú jedným z najdôležitejších podkladov synchronizovanej aktivity EEG. Podstatou synchronizácie výbojov talamických jadier je zmena ich membránového potenciálu. Na základe prichádzajúcich informácií z EPSP to aktivuje napäťové Ca^{2+} kanály. Následkom bude zmena membránového potenciálu, ktorá aktivuje ďalšie Ca^{2+} kanály a následkom tohto cyklu je akčný potenciál talamických neurónov. V tejto fáze nastáva hyperpolarizácia membrány, ktorá je obnovená tokom draslíka. S pomocou cholinergných vstupov kationov z mozgového kmeňa a predného mozgu je udržiavaný membránový potenciál talamických jadier, kvôli čomu je možný prevod senzorických informácií do mozgovej kôry v aktívnom stave. ^[2]



Obr. 2. Rozdelenie mozgovej kôry

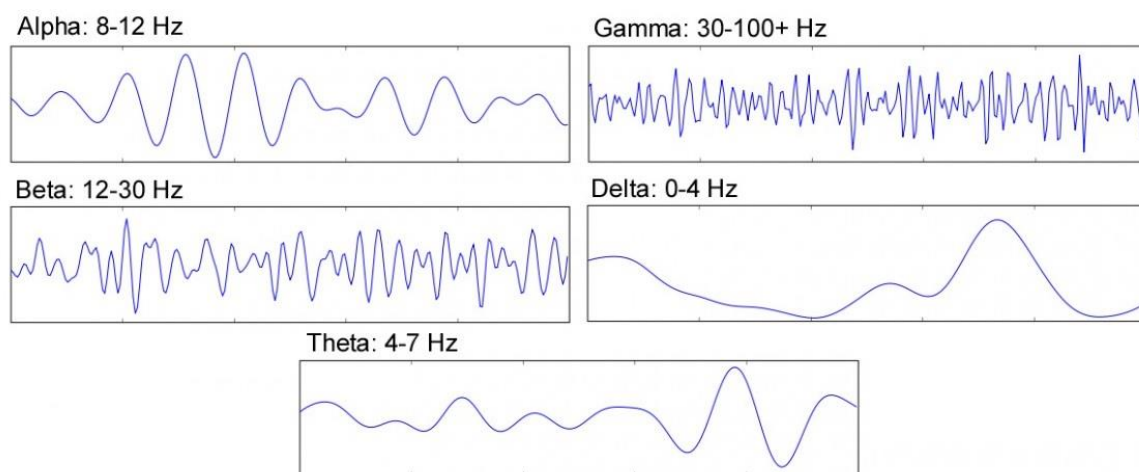
Priemerný ľudský mozog obsahuje 86 miliárd neurónových buniek, ktoré tvoria viacvrstvovú pavučinu a vedľa nich je ešte omnoho väčšie množstvo gliových buniek. Tieto majú za úlohu stabilizovať chemické prostredie, regulovať a chrániť neuróny. V najvrchnejšej vrstvy mozgovej kôry, ktorá je niekoľko milimetrov hrubá, môžeme nájsť cca. 70% všetkých neurónov mozgu. Práve preto môžeme povedať, že táto časť mozgu je

najvyvinutejšia. Táto časť je rozdelená na laloky, ktoré majú špecifické úlohy pri obsluhu človeka. Podľa použitého anatomického systému mozgovú kôru rozdelíme na 4 až 6 lalokov. Z hľadiska nášho zamerania je dôležitejšie rozčlenenie na 4 hlavné časti, ktoré vidíme na Obr. 2: *čelový lalok*, *temenný lalok*, *záhlavný lalok* a *spánkový lalok*.

Čelový lalok ručí za zložité duševné funkcie. Takáto funkcia je napr. rozhodovanie alebo uvažovanie, kde samozrejme patria aj motorické oblasti, ktoré plánujú a riadia pohyby. Ďalším lalokom je **temenný**, ktorý obsahuje somatosenzitívne oblasti, kde sú spracované hmatové informácie prichádzajúce z tela. Úlohou je integrácia rozličných typov informácií ako je napr. priestorové vnímanie. **Záhlavný lalok** zahŕňa desiatky menších oblastí, ktoré sú špecializované na spracovanie a interpretovanie informácií pochádzajúce zo zraku. Do **spánkového laloka** prichádzajú informácie z uší. Jeho kôra má špecifické oblasti, ktoré zaisťujú pochopenie reči. Vnútri sa nachádza hipokampus, ktorý má svoju úlohu pri tvorbe spomienok a spolu s okolitými oblasťami pomáha pri priestorovej orientácii.^[3]

Neurón dokáže vysielat' tisícky impulzov za sekundu, kým sa vytvorí okruh elektrických signálov, ktoré je možné v technickom zmysle zaznamenať. Tieto vlny porovnávajú potenciály dvoch bodov na povrchu kože lebky ako bipolárny záznam. Druhá metóda je unipolárny záznam, ktorý meria rozdiel elektrických potenciálov medzi mozgovým tkanivom a bodom s nulovým potenciálom.^[2]

Po získaní EEG signálov je tieto potrebné zosilniť a odfiltrovať z nich šum. Takto dostaneme výsledky, ktoré zaznamenáme do grafu. Aktuálny stav mozgu môžeme kategorizovať podľa frekvencií a amplitúdy jednotlivých vln. Medzi základné typy patria alfa, beta, gama, delta a theta vlny (Obr. 3).



Obr. 3. Základné typy EEG signálov

Alfa vlny sú tie signály, ktoré je možné zaznamenať pri všetkých vekových skupinách. Dospelý človek sa vie dostať do tohto stavu, keď si zavrie oči v bdelom stave. Tento signál je možné zachytiť na oboch stranách hlavy a je mierne silnejší na nedominantnej časti. Zdroj týchto vln je v temennom a záhlavovom laloku, ktoré reprezentujú bielu hmotu mozgu. Tieto časti tvoria tzv. most medzi vedomou a podvedomou myslou. Alfa vlna indukuje produkciu serotonínu, endorfinu a acetylcholínu. Z týchto hormónov je najdôležitejší serotonín, ktorý zvyšuje relaxáciu a zmieňuje pocit bolesti. ^[4]

Beta vlny sa produkujú pri konaní a správaní. Tieto signály súvisia so zberom informácií z okolia pomocou senzorov a to: **zrak, sluch, hmat, čuch a chuť**. Tieto signály je možné zvyčajne zaznamenať na oboch stranách čelného a temenného laloku. Beta vlny aktivujú výrobu dopamínu, adrenalínu, noradrenalínu a kortizolu. Kortizol má veľkú rolu pri učení a pri tvorbe spomienok. Signály tejto frekvencie sa vyskytujú v stave vedomia, ako je rozprávanie, riešenie problémov, pri posudzovaní a pri rozhodovaní. ^[4]

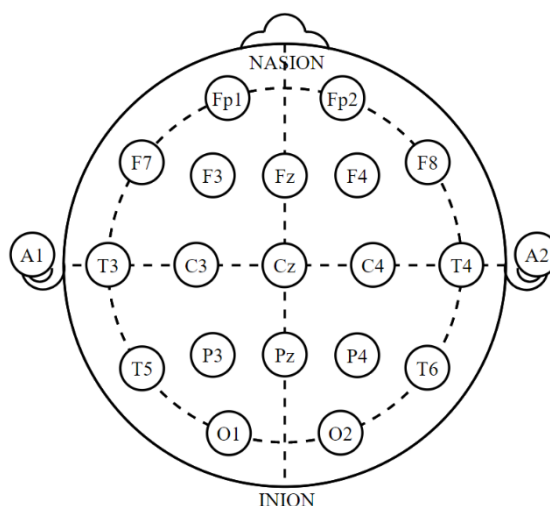
Gama vlny sú spárované s vnímaním a s vedomím, ktoré obvykle majú frekvenciu medzi 30 a 70 Hz. Daný typ signálov sa dá zaznamenať len s digitálnymi prístrojmi, lebo analógové sú zvyčajne obmedzené na 25 Hz. Gama signály sa vyskytujú pri hyperbdelosti a pri integrácii senzorických informácií. Ďalšou úlohou je kombinácia zmyslu a pamäti, aby sa takýmto spôsobom zabezpečila dokonalá skúsenosť. Dôležitú rolu hrajú pri tvorbe endorfinu a serotonínu. Pôvod týchto signálov vychádza zo somatosenzorickej kôry. ^[4]

V porovnaní s ostatnými typmi má najvyššiu amplitúdu a najpomalšie signály *delta* vlna. Tieto signály sa tvoria v šedej hmote mozgu. Najpomalšia aktivita mozgových vln je pozorovateľná vo všetkých štádiách spánku, najmä v jeho tretej a štvrtej etape, ktoré nazývame hlbokým spánkom. Aktuálny typ rytmu je dominantne viditeľný pri dojčatí. Delta vlny indikujú melatonín, rastový hormón a je abnormálnym prípadom, keď sú tieto signály zaznamenané pri dospelom človeku v bdelom stave. ^[4]

Theta vlny sú identifikované v rozsahu 4 – 7 Hz, ktoré súvisia s podvedomou aktivitou. Ich zdroj môžeme lokalizovať v oblasti talamu. Daný typ mozgovej aktivity je spozorovateľný pri hlbokej relaxácii a meditácii. Tieto signály majú vplyv na duchovný vzťah, ktorým sa venujú výskumy týkajúce sa získavania mimozmyslových schopností. Psychiatri a psychológovia používajú túto úroveň na zhromaždenie informácií o stave pacienta. Theta vlny indikujú hladinu serotonínu, acetylcholínu, anti-kortizolu a endorfinu. ^[4]

1.3 Topológia hlavy a umiestnenie senzorov

Pri získaní podstatných dát je dôležitým aspektom vyznať sa v topológii hlavy, aby jednotlivé senzory boli umiestnené v blízkosti zdroja. Skrátka, keď chceme zaobstarať dáta pochádzajúce zo zraku, nemali by sme senzory umiestniť na čelný lalok, pretože v tom prípade dostaneme nežiadúce informácie. Z tohto dôvodu vytvorila šandardy Medzinárodná Federácia pre elektroencefalografiu a klinickú neurofyziológiu, podľa ktorých je možné jednoducho umiestniť jednotlivé elektródy. Najpoužívanější štandard v rámci výskumov, ktoré vykonávajú s pomocou BCI systémov, používa 10-20 rozmiestnených elektród na povrchu hlavy. Aby toto rozmiestnenie bolo čo najpresnejšie dodržané, boli vytvorené elektródové čiapky. Tieto môžu byť vyrobené z textilu alebo môžu mať aj tuhú formu podobajúcu sa na prilbu. Pomocou týchto prístrojov môžeme urýchliť a spresniť jednotlivé záznamy mozgových aktivít.



Obr. 4. Topologická mapa hlavy

Rozmiestnenie senzorov na zbieranie EEG signálov je reprezentované na Obr. 4. Jednotlivé miesta elektród sú vyznačené veľkými písmami ktoré sa zhodujú prvými písmenami anglických názvov lalokov. Tieto sú:

- F – (Frontal) čelný lalok
- Fp – (Pre-Frontal) predný čelný lalok
- T – (Temporal) spánkový lalok
- C – (Central) zaznamená senzorické a motorické funkcie
- P – (Parietal) temenný lalok
- O – (Occipital) záhlavový lalok
- A – (Mastoid) kontralaterálne referenčné body pre ostatné signály

Na Obr. 4 je vidno, že po veľkých písmenách sú 'z'-čka (ktoré označujú stredný región) alebo čísla. Nepárnyimi číslami označujeme ľavú stranu hlavy a párne čísla sú na pravej strane.^[3]

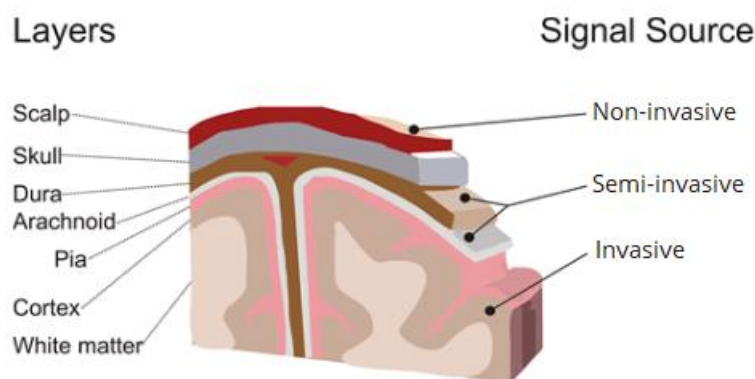
Ako sme už v predošlej podkapitole spomínali, senzory je možné zapojiť dvoma metódami: unipolárne a bipolárne. Unipolárna metóda pracuje na princípe, kedy každá elektróda berie potenciálny rozdiel od referenčného senzora, ktorý je umiestnený zvyčajne na ušnom laloku, označeného písmom (A). Alternatívou umiestnenia referenčnej elektródy je aj bod (Cz). Tieto body nie je odporúčané dať ďaleko od ostatných elektród (napr. na špičke nosa alebo na končatinách), lebo môžeme riskovať nedokonalé fungovanie resp. stratu informácií. Druhá metóda je bipolárna a na rozdiel od unipolárnej metódy pracuje na princípe, kde elektródy zaznamenávajú potenciálne rozdiely medzi dvoma aktívne párovanými elektródami. Takýmto spôsobom sú jednotlivé senzory spojené reťazci v pozdĺžnej alebo priečnej forme.^[3]

1.4 BCI systémy

BCI systémy sú prístroje, ktoré tvoria most medzi mozgom a počítačom. Jonathan R. Wolpaw z Wadsworth Center for Laboratories and Research v roku 1991 bol človek, ktorý prvýkrát implementoval funkčný BCI systém a zaznamenal EEG signály v digitálnej forme. Výsledkom projektu bolo riadenie kurzora počítača v jednej dimenzii, čo dosiahol pomocou merania amplitúdy μ (mV) signálu. Ďalším výsledkom tejto skupiny odborníkov bolo dosiahnutie riadenia polohy kurzora aj dvojrozmernom priestore v roku 2004.^[5]

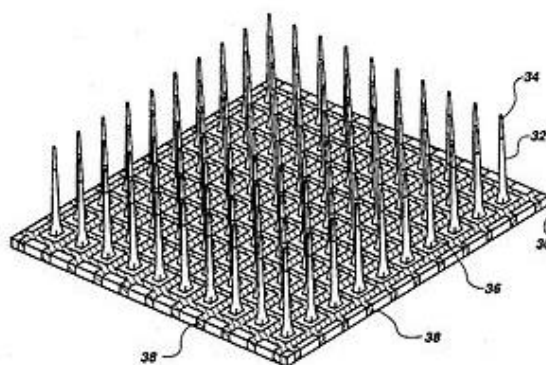
Avšak pred týmto prebehla digitalizácia EEG signálov, ktorá na prvý pohľad vyzerá jednoducho, ale v praxi je to oveľa zložitejší proces. V prvom kroku musíme s pomocou BCI systému zosilniť jednotlivé signály, ktoré prebehnú cez jednotlivé elektródy. Je to dôležité, kvôli faktu, že na povrchu lebky sú tieto signály rádovo v desiatkach mikrovoltov a elektronika nie je schopná spracovať takéto nízke napätie. Preto zosilnime mozgové vlny pomocou operačného zosilňovača, ktorý dobre filtruje šum a jeho zisk je v intervale 100 – 100 000. Po získaní dostatočne silného signálu je možné tento vzorkovať a pomocou AČ prevodníka digitalizovať do binárnej podoby. K digitálnemu signálu potom BCI pridá časovú stopu a získané údaje pošle do počítača.^[6]

BCI systémy môžeme rozdeliť na základe viacerých kritérií. My ich rozdelíme do troch kategórií. Aspekt, podľa čoho boli urobené tieto skupiny silne závisí od spôsobu ich implementácie. Jednotlivé triedy sú reprezentované na pravej strane obrázku (Obr.5) a na ľavej strane nájdeme popis, aké vrstvy sa nachádzajú v ľudskej hlave.



Obr. 5. Vrstvy hlavy a typy BCI systémov

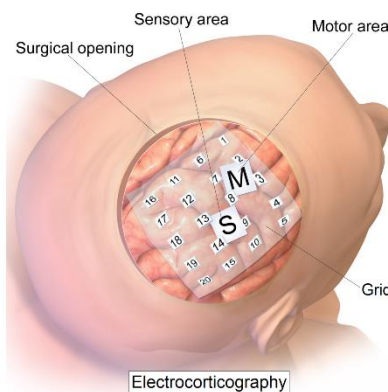
Prvou kategóriou sú invazívne BCI systémy ktoré sú implementované do mozgovej kôry pomocou neurochirurgickej operácie. Tieto triedu ešte je možné roztriediť na dve podkategórie. Prvá je *single unit BCI*, ktorý zaznamenáva signály v oblasti jednej mozgovej bunky a druhá je *multi unit BCI*, ktorá je schopný zbierať dáta z viacerých oblastí súčasne. Invazívne systémy majú elektródy zvyčajne 1,5 – 10 mm dlhé. Kvalita signálu je tu najvyššia a aj najšpecifickejšia, lebo je aplikovateľná na malé oblasti a zbiera dáta priamo z buniek. Ako všetky invazívne prístroje, aj tieto majú problém so skutočnosťou, že okolo elektródy sa môžu objaviť jazvy, ktoré zhoršujú kvalitu signálov. Objavujúce sa jazvy sú reakciou tela, keď nájde v sebe cudzí predmet. Ďalšou nevýhodou je, že samotná operácia je veľmi drahý a riskantný proces, kvôli čomu invazívnu metódu aplikujú väčšinou iba na nevidomých a ochrnutých pacientov.^[7]



Obr. 6. Mikro-elektrody invazívnej metódy

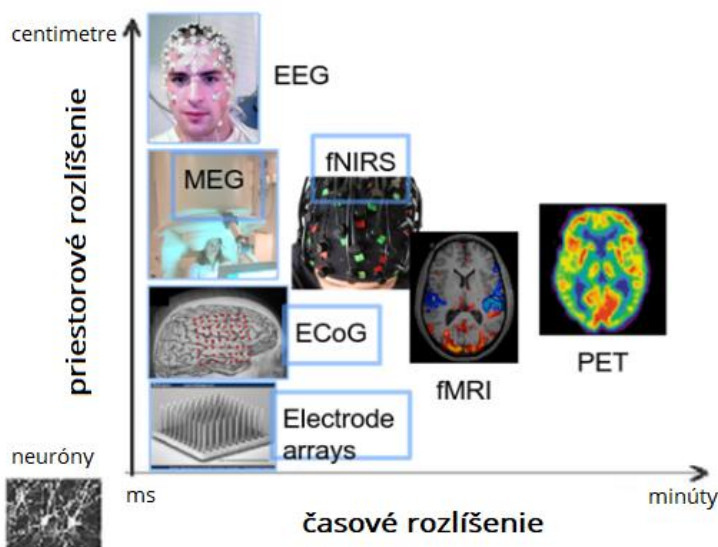
Druhú kategóriu tvorí čiastočne invazívny BCI systém. Túto metódu aplikovali prvýkrát v Montrealskom neurologickom ústave v 50. rokoch. Tento druh má elektródy implementované pod lebkou na povrchu mozgu a ostatné časti sú mimo lebky. Je to kvôli tomu čiastočne invazívne, pretože implementáciu robia podobným štýlom ako pri invazívnom, a to pomocou operácie. Kvôli tomu aj túto metódu aplikujú len v tom prípade,

keď je chirurgický zákrok nevyhnutný. Pri operácii elektródy umiestnia nad alebo pod epidurálnu vrstvu. Obvykle ECoG používajú v pásovom alebo mriežkovom formáte, ktorý môžu mať 4 až 256 elektród, čím pokryjú väčšiu časť, ako v prípade invazívneho typu.^[7]



Obr. 7. ECoG elektródy čiastočne invazívnej metódy

Poslednú kategóriu tvoria neinvazívne BCI systémy ktoré majú najväčšie zastúpenie oproti ostatným, lebo nie je potrebný chirurgický zásah. Do tejto skupiny patria: MEG, PET, fMRI, fNIRS a EEG. Nevýhoda tohto typu systému spočíva v nepresnosti týchto prístrojov resp. v tom, že majú dlhú časovú odozvu oproti invazívnym a čiastočne invazívnym metódam. Jeden signál môže zhrňať a väčšinou aj zahŕňa reakciu tisícok neurónov a kvôli tomu je oveľa zložitejšia analýza jednotlivých mozgových aktivít. Presnosť a oneskorenie jednotlivých prístrojov je vidno na nasledujúcom obrázku (Obr. 8).



Obr. 8. Graf časovo priestorového rozlíšenia

2 Analýza EEG signálov

Pri tvorbe aplikácií, ktorá nám umožňuje riadiť mobilného robota pomocou mozgových vĺn, je kľúčovým aspektom analýza EEG signálov. V tejto časti analýzy sa musíme rozhodnúť, z akej lokality budeme získavať mozgové vlny. Po viacerých testoch sme dospeli k tomu, že v tejto práci sa sústredíme na signály, ktoré sú tvorené naším vizuálnym systémom. Očné funkcie je možné roztriediť do dvoch skupín, a to motorické a senzorické.

Pri skúmaní motorickej funkcie oka, musíme elektródy umiestniť na pre-frontálny lalok Fp1 a Fp2, kde budú zaznamenávať očné pohyby a žmurknutie. Elektróda, ktorá je umiestnená na Fp1 bude zbierať dáta z ľavého oka a Fp2 z pravého. Výhodou signálov pochádzajúcich z pohybu oka je ich jednoznačnosť a fakt, že sa dajú rýchlo identifikovať na základe grafu. Aby sme mohli riadiť nejakého robota, musíme pripraviť aspoň 3 riadiace príkazy, ktoré ukazujú do jednotlivých smerov pohybu a tiež príkaz zastavenia. To znamená, že musíme nájsť aspoň 4 odlišných resp. jedinečných signálov a tak vytvoriť konkrétne kategórie jednotlivých príkazov.

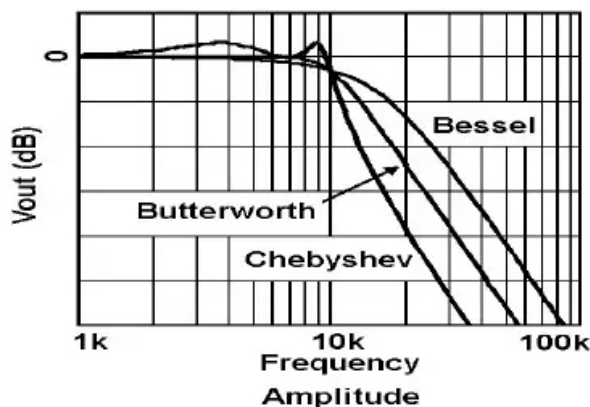
Druhá skúmaná vlastnosť oka, s ktorou sa budeme zaoberať, je jeho senzorická funkcia. Budú sa zaznamenávať informácie získané pomocou zraku. Vzniknuté signály budeme zbierať zo záhlavového laloku, kde budú jednotlivé elektródy umiestnené na miesta O1 a O2 (Obr. 4). V tomto prípade je nutnou podmienkou vytvoriť program, ktorý nám zobrazí štyri štvorce (na každú stranu monitoru jeden) blikajúcich na odlišných frekvenciách. Naším cieľom bude identifikácia jednotlivých frekvencií objavujúcich sa na záhlavovom laloku a ich kategorizácia do jednotlivých skupín.

Na vytvorenie ideálnych okolností pre tieto signály, musíme získať mozgové vlny pomocou OpenBCI. Po zaznamenaní takto získaných dát, je nutnou podmienkou filtrácia signálov. Je to dôležité kvôli získaniu jedinečných a normalizovaných dát, ktoré sa dajú zadeľiť do vopred stanovených kategórií a porovnať medzi sebou. Jednotlivé typy filtrov sú stručne opísané v nasledujúcich podkapitolách.

2.1 Butterworthov filter

Butterworthov filter slúži na spracovanie signálov. Je navrhnutý tak, aby mal v priestupnom pásme čo najrovnejšiu frekvenčnú odozvu. Daný typ filtra bol prvýkrát publikovaný v príspevku pod názvom „On the theory of filter amplifiers“ v roku 1930. Filter bol pomenovaný podľa britského inžiniera Stephena Butterwortha. V niektorých jeho publikáciách môžeme nájsť filter pod menom maximálne plochá aproximácia, keďže n -tá

mocnina pri prvom deriváte pri frekvencii nula je rovná nule. To v praxi znamená, že v priepustnom pásme nie je žiadne zvlnenie a DC zisk je maximálne plochý. Obr. 9 porovnáva logaritmickú amplitúdovú frekvenčnú charakteristiku troch rôznych filtrov, kde Butterworthov filter je kompromisom medzi filtermi Bessela a Chebysheva. Na obrázku vidíme, že náš filter v priepustnom pásme ešte nie je zvlnený ako v prípade filtra Chebysheva a pri vyšších frekvenciách je strmší ako Besselov filter.^[8]



Obr. 9. Porovnanie troch rôznych filtrov

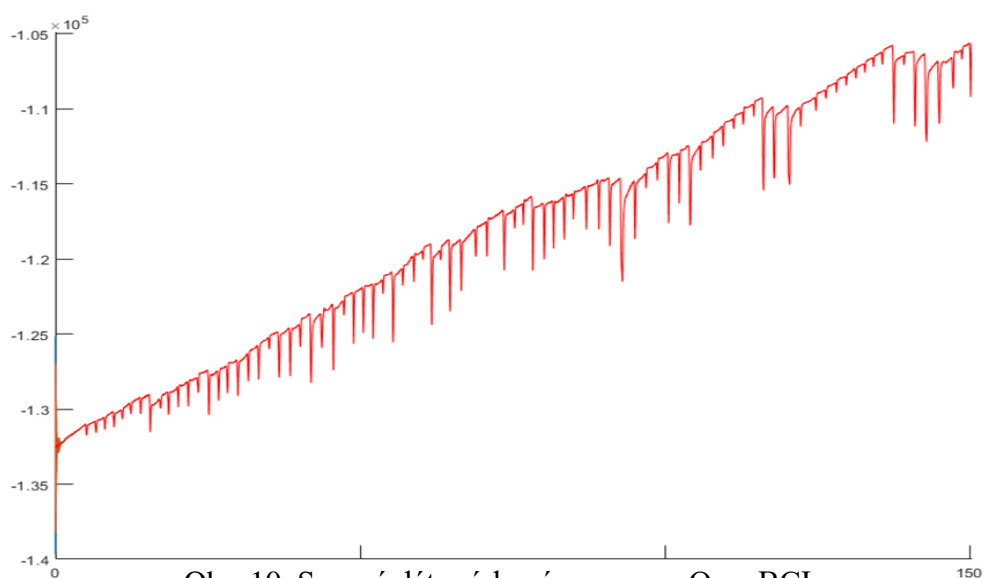
2.2 Spracovanie signálov

Keď chceme zabezpečiť, aby získané dáta boli čo najdôveryhodnejšie, podstatnou podmienkou je zosnímať signály s minimálnym šumom. Skreslenie zozbieraných dát sa nám môže preukázať ako na hardvérovej, tak aj na softvérovej strane. Pri hardvérovej časti tvorí najväčší problém to, že v európskych krajinách fungujú elektrické sústavy resp. zariadenia na 50 Hz a v Amerike na 60 Hz, čo môže spôsobovať saturáciu senzora. Práve preto, pri spracovaní zozbieraných signálov musíme brať do úvahy skutočnosť, že ľudské telo funguje ako anténa resp. zosilňovač signálov a zachytáva nežiaduce šumy z nášho okolia. Distorzia hardvérového signálu môže byť spôsobená aj dýchacími pohybmi, vibráciou svalov, srdca a ostatnými pohybmi skúmaného človeka. Do softvérových šumov môžeme zaradiť stratu informácií pri komunikácii medzi dvoma hardvérm, napr. medzi OpenBCI a PC. To v praxi môže znamenať, že v niektorých prípadoch bol náš softvér pomalší a nestihol zaznamenať všetky prichádzajúce signály z OpenBCI prístroja. Ďalšou chybou môžu byť aj zle zvolené parametre pri filtrácii dát.

Aby sme zabezpečili vhodné vzorky pre neurónové siete a DTW, musíme odstrániť šum a zbytočné vysoké resp. nízke harmonické signály. Po vyskúšaní viacerých filtrov sme prispeli k tomu, že najlepším kompromisom bude, keď použijeme Butterworthov filter.

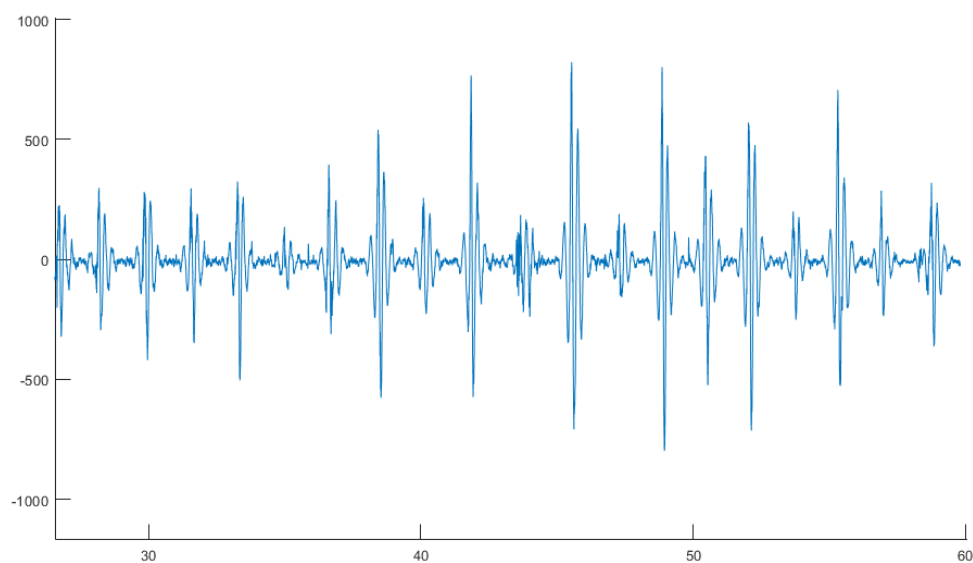
Tento typ filtra je pre nás vyhovujúci aj z dôvodu, že ho môžeme aplikovať viacerými spôsobmi. Po vyskúšaní rôznych metód sme skonštatovali, že použijeme vrubovú (notch) a pásmovú (bandpass) metódu.

Na nižšie uvedenom obrázku (Obr. 10) vidíme surové dáta získané z OpenBCI hardvéru, ktoré boli zaznamenané senzorom Fp1. V danom experimente bolo zaznamenané žmurkanie očami, pričom jednotlivé žmurknutia sú reprezentované skokmi dole. Pri manuálnom spracovaní dát je jednoduché určiť jednotlivé žmurknutie, ale pre neurónovú sieť by tento proces bol komplikovaný.



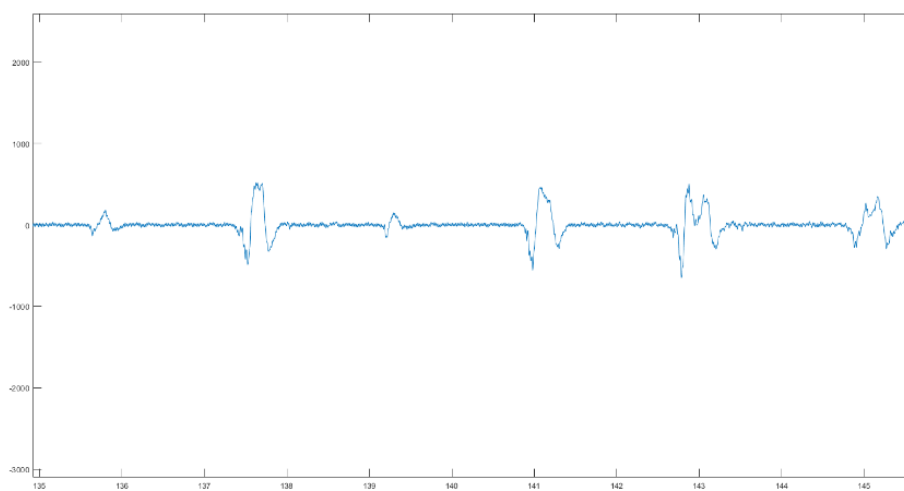
Obr. 10. Surové dáta získané pomocou OpenBCI

Na základe týchto znalostí, naším prvým krokom bude normalizácia získaných údajov a aplikácia pásmovej filtrácie. Pomocou pásmového filtra normalizujeme zozbierané dáta a následne odfiltrujeme nepodstatné harmonické signály, ktoré by tvorili nežiadúci šum a tým by znemožnili ich použitie pri vytváraní vzorov. Pri žmurknutí sme aplikovali filter s intervalom medzi 5-50 Hz a pri senzorickom zaznamenaní sme použili užší interval medzi 2-20 Hz. Po vyfiltrovaní zbytočných signálov dostaneme signály, ktoré sú reprezentované na nižšom obrázku (Obr. 11).



Obr. 11. Butterworthov pásmový filter medzi 5-50 Hz

Druhým krokom je aplikácia vrubového filtra, ktorý bolo nutné použiť aby sme odstránili 50 Hz šum. Tento šum je spôsobený elektrickými zariadeniami, ktoré používame v každodennom živote. Po použití tohto filtra dostaneme výsledný signál (Obr. 12). Po nasadení filtrov je možné dáta rozdeliť do jednotlivých kategórií a použiť v ďalšej časti našej práce.



Obr. 12. Dáta získané po aplikácii filtrov

3 Klasifikačné metódy

Po analýze jednotlivých dát, na ktoré boli aplikované filtre spomínané v predošlej kapitole, je potrebné aplikovať manuálnu klasifikáciu a vytvoriť databázu. Daný postup je nutný kvôli tomu, aby sme našli jedinečné atribúty danej triedy a aby sa dal automatizovať celý kategorizačný proces. Táto automatizácia nám umožní spracovanie stovák záznamov a tisícok vzoriek. Ďalšou výhodou je, že program bude schopný roztriediť signály a premeniť ich na príkazy v reálnom čase aj v budúcnosti. Toto nám v zásade umožní ovládať robota.

Klasifikácia je jedným z najčastejšie sa vyskytujúcich problémov pri rozhodovacích úlohách ľudskej činnosti. Najväčšia výzva pri klasifikácii je, keď objekt musí byť priradený do preddefinovanej skupiny alebo triedy na základe pozorovaných atribútov daného objektu. Medzi takéto klasifikačné problémy patrí: predikcia burzy, predikcia bankrotu, lekárska diagnostika, rozpoznanie znakov ale aj kontrola kvality. Do tradičnej klasifikácie patria štatistické metódy ako je napr. diskriminačná analýza, ktorá je založená na Bayesovskej teórii rozhodovania sa. Tieto postupy sú založené na pravdepodobnosti, v akej miere sa zhoduje skúmaný objekt s jednotlivými predlohami v danej triede. Jeden z hlavných vymedzovacích štatistických metód funguje adekvátne iba v prípade, keď sú splnené základné predpoklady. Postup sa dá najlepšie uplatniť, keď má používateľ veľmi dobré znalosti ohľadom tzv. úzkych miest používanej databázy. Ďalšou skupinou klasifikácie tvoria umelé neurónové siete. Klasifikačné neurónové siete sú sľubnou alternatívou klasických klasifikačných metód. Ich výhoda spočíva v triedení samo aditívnou riadenou metódou bez akejkoľvek špecifikácie konkrétnych funkcií alebo pravidiel. Štruktúra umelých neurónových sietí je nelineárna, vďaka čomu sú flexibilné pri modelovaní skutočného sveta. Ich ďalšia výhoda spočíva v tom, že používateľ nemusí poznať celý proces, len vstupné a výstupné údaje. Práve kvôli tomuto môžeme povedať, že neurónové siete fungujú ako čierna skrinka.^[10]

Na porovnanie úspechu našej klasifikácie sme vybrali dve metódy, ktoré uplatníme v našej práci. Pri oboch metódach je nutnou podmienkou vytvoriť si požadované množstvo tréningových dát, podľa čoho budú klasifikované prichádzajúce mozgové vlny v reálnom čase. Takto získané dáta triedia obe metódy súbežne v reálnom čase a identifikované výsledky bude program štatisticky vyhodnocovať podľa presnosti.

Prvou zvolenou metódou je kombinácia tradičnej štatistickej metódy *K-najbližších susedov* (kNN) a *Dynamic Time Warpingom* (DTW), kvôli tomu, pretože má jednoduchú štruktúru a je spoľahlivá. Táto metóda dokáže presne porovnať aj časovo nezhodujúce sa

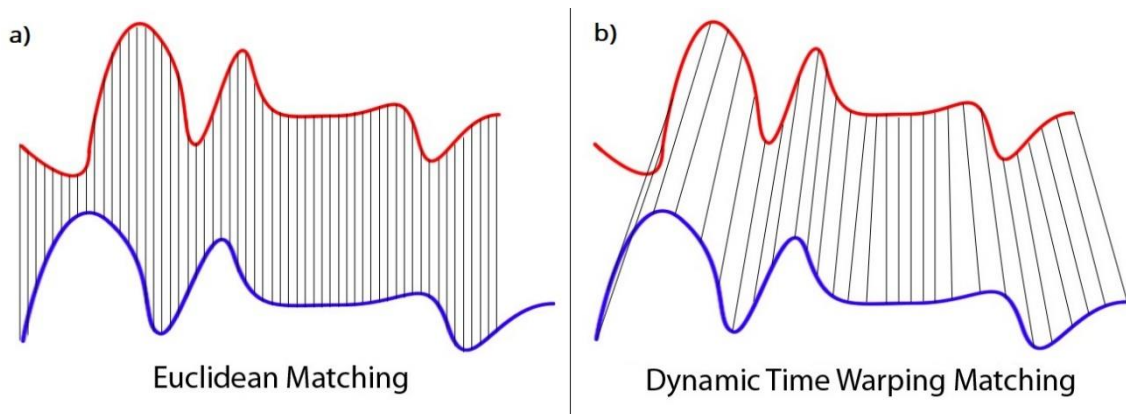
signály. Ďalšou výhodou danej metódy je aj to, že dokáže vytvoriť kategorizáciu aj s menším počtom vzoriek.

Druhá zvolená metóda je klasifikačná neurónová sieť, a to kvôli tomu, lebo sú stále najefektívnejšími nástrojmi pri ťažbe údajov a ich klasifikácií. Tento spôsob je dnes široko uznávaný ako výkonný nástroj na modelovanie. Vo väčšine aktuálnych softvérov je implementovaný na získavanie a spracovanie údajov. V danom princípe vidia vedci budúcnosť, ako vytvoriť spoľahlivý prístroj ktorý dokáže dekodovať signály prichádzajúce z mozgu človeka v reálnom čase. Ďalšou výhodou neurónovej siete sú vlastnosti, ako adaptivita a jednoduchosť, s ktorou sa dajú pridávať nové funkcie a s ktorou sa dá pretrénovať.

3.1 Dynamic time warping

V posledných rokoch klasifikácia a zoskupovanie jednotlivých údajov časových radov sa stali predmetom veľkého záujmu komunitách zaoberajúci sa databázami a pri ťažbe údajov. To znamená, že museli aplikovať nejaké metódy, pomocou ktorých sa dajú porovnať dve rozličné časové rady. Tieto spôsoby môžeme rozdeliť následne do dvoch kategórií, ktoré sú založené na vzdialenosti alebo na základe jedinečných vlastností.

Podľa metód založených na vzdialenosti je najskôr definovaná funkcia vzdialenosti na výpočet podobnosti medzi dvoma časovými radmi a potom je klasifikovaný do triedy podľa vzorky, od ktorej má najmenšiu funkciu vzdialenosti. Prvou takou metódou danej kategórie bola Euklidovská vzdialenosť (ED). Metódu prvýkrát aplikoval vo svojej práci Rakesh Agrawal v roku 1993, s podmienkou, že dve sekvencie sa považujú za podobné ak je Euklidovská vzdialenosť medzi nimi menšia ako preddefinovaný prah ϵ .^[11] V nasledujúcich rokoch sa daný postup veľmi rýchlo rozšíril aj napriek jeho známej slabej stránke, ktorou je citlivosť na skreslenie časovej osi (viď. Obr. 13a). V roku 1994 Donald J. Berndt a James Clifford predstavili novú metódu na meranie vzdialenosti pod menom Dynamic Time Warping (DTW), ako riešenie pre túto konkrétnu slabosť tejto metodológie. Flexibilita tohoto postupu spočíva v tom, že umožňuje zarovnať dve nelineárne časové rady, ktoré sú podobné, ale lokálne sa nezhodujú (viď. Obr. 13b). Napriek tomu, že má časovú zložitosť $O(n^2)$, DTW je najviac známym riešením v oblasti s časovými radmi. Daná metóda je aplikovaná v mnohých oblastiach vrátane lekárskeho, finančného, strojárskeho, bioinformatického ale aj priemyselných sférach.^[12]



Obr. 13. Reprezentácia ED na grafe a), reprezentácia DTW na grafe b)

Pri ED sú dané dve časové rady $Q = q_1, q_2, \dots, q_i, \dots, q_n$ a $C = c_1, c_2, \dots, c_1, \dots, c_m$, kde vzdialenosť medzi nimi je možné zadefinovať nasledovne:

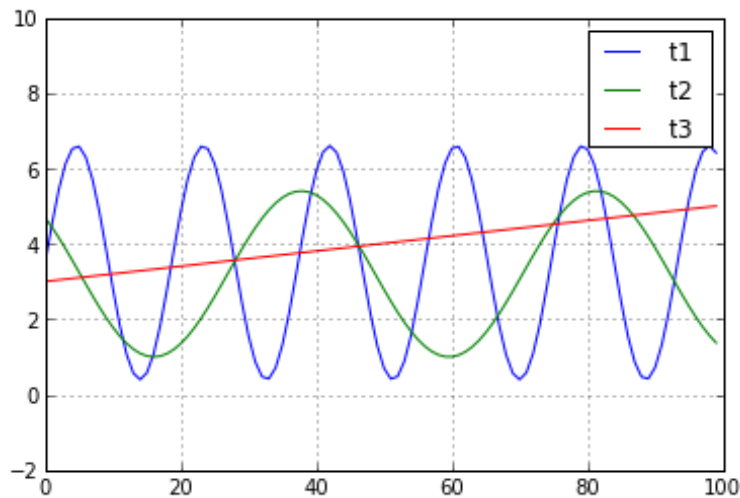
$$ED(Q, C) = \sqrt{\sum_{i=1}^n (q_i - c_i)^2} \quad (3.1)$$

Dajú sa vypočítať len v tom prípade, keď platí podmienka $n = m$. Silná strana danej metódy spočíva v tom, že je nesmierne jednoduchá a účinná a kvôli tomu je ešte stále veľmi populárna oproti iným spôsobom. Avšak okrem toho, že tieto dve časové rady majú byť rovnako dlhé, ED má ďalšiu nevýhodu vtom, že je veľmi citlivá aj na malé nezhody tvorené šumom alebo inými interferenciami, čo môže spôsobiť nerovnosť medzi dvoma časovými radmi.^[12]

Pri DTW sú dané dve časové rady $Q = q_1, q_2, \dots, q_i, \dots, q_n$ a $C = c_1, c_2, \dots, c_1, \dots, c_m$, kde vzdialenosť je počítaná dovedy, kým nenájde najlepšie zarovnanie. Na zarovnanie týchto časových radov sa používa matica n krát m , v ktorej (i, j) prvok sa rovná ku $(q_i - c_j)^2$. Rovnica reprezentuje náklady na zarovnanie bodu medzi elementom q_i z časového radu Q a c_j z časového radu C . Zrovnanie medzi bodmi je znázornené s deformačnou krivkou $W = w_1, w_2, \dots, w_i, \dots, w_n$, ktorá musí byť súvislá, monotónna a začínajúca sa od ľavého dolného rohu smerujúca do pravého horného rohu matice. Najlepšie zarovnanie dostaneme podľa deformačnej krivky, ktorá je tvorená najmenšími nákladmi v matici a súčet krivky tvorí vzdialenosť DTW. Rovnica použitá na výpočet matice je nasledujúca: ^[12]

$$DTW(Q, C) = \arg \min_{W=w_1, \dots, w_K} \sqrt{\sum_{k=1, w_k=(i,j)}^K (q_i - c_j)^2} \quad (3.2)$$

V praxi sme porovnali obe metódy, ktoré by boli presnejšie a lepšie využiteľné v našej práci. Prvým krokom bolo vytvorenie funkcie v programovom jazyku Python. Dané funkcie sú pripojené v časti Príloha A. Na porovnanie sme vybrali tri rôzne časové rady s označením $t1$, $t2$ a $t3$, ktoré sú reprezentované na Obr. 14. Pri prvom experimente sme použili ED, kde sme porovnali $ED(t1, t2)$, kde nám vyšla hodnota 26.9 a potom sme vypočítali $ED(t1, t3)$ a kde sme dostali hodnotu 23.2. Ako to vidíme aj na obrázku $t1$ sa viac podobá ku $t2$ ako $t1$ k $t3$ ale napriek tomu čísla hovoria pravý opak. V ďalšom pokuse sme použili tie isté časové rady, ktoré boli aplikované v predošlom experimente, ale teraz sme aplikovali DTW. Pri porovnaní $t1$ s $t2$ sme dostali hodnotu 17.9 a v prípade $t1$ s $t3$ hodnota bola 21.5. Ako je možné vidieť z hodnôt, druhá metóda je oveľa presnejšia ako predošlá. Po vyskúšaní viacerých vstupných dát sme skonštatovali, že pre naše ciele bude viac vyhovovať DTW ako ED.



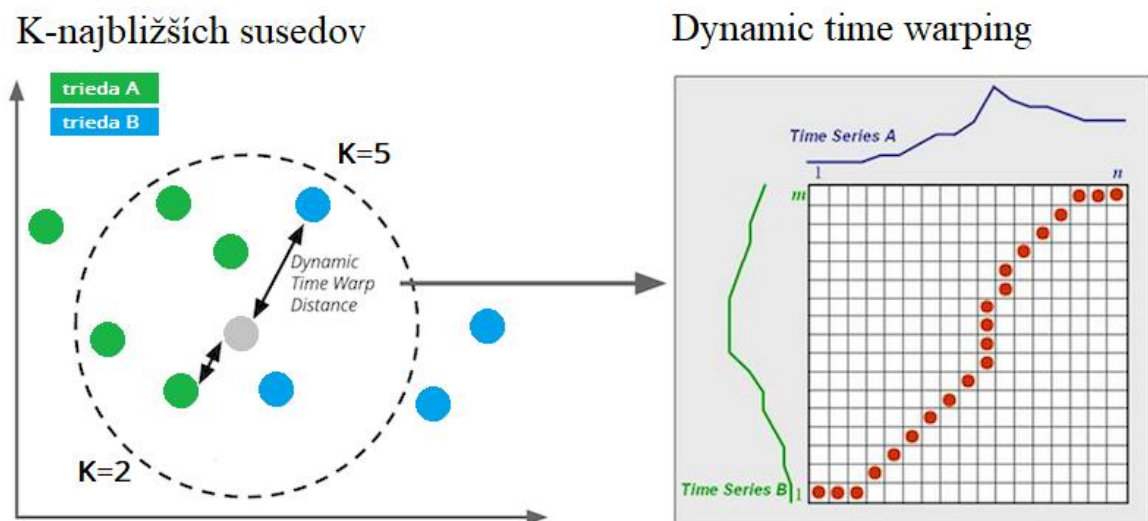
Obr. 14. Graf zobrazujúci tri rôzne časové rady ($t1$, $t2$, $t3$)

3.2 Klasifikátor K-najbližších susedov

V prípade, keď hľadáme vzory s pomocou analýzy v obrovskej dátovej množine, je veľmi ťažké nájsť jedinečné vlastnosti danej triedy, na základe ktorých ich vieme porovnať resp. klasifikovať. Práve preto je vhodné použiť metódu slúžiacu na rozpoznávanie vzorov. Na porovnanie jednotlivých predlôh, vytvorené v reálnom čase sme použili metódu DTW. Keďže samotné porovnávanie našu úlohu dostatočne nevyrieši, zvolený spôsob musíme skombinovať s ďalším klasifikátorom, ktorý bude v našom prípade štatistická metóda K-

najbližších susedov kNN. Táto metóda patrí medzi najjednoduchšie metódy strojového učenia. Algoritmus danej metódy klasifikuje objekty do dvoch alebo viac odlišných tried.

V prvej fáze porovnáva už známe vzory s ešte neznámymi a neidentifikovanými objektmi pomocou DTW. Toto porovnávanie sa vykonáva na základe jednotlivých atribútov, ktoré boli vytvorené podľa jednotlivých príkazov. Po výpočte vzdialenosti zoradíme hodnoty od najbližšiemu k nevzdialenejšiemu. Následne sa vykoná hlasovanie k -prvých prvkov. Každý prvok hlasuje za svoju triedu a do triedy, na ktorú prišlo najväčšie množstvo hlasov sa priradí neznámy objekt. Dobrým príkladom pre pochopenie danej problematiky nám bude slúžiť Obr. 15, kde sivý kruh na strede obrázku reprezentuje neznámy objekt, ktorý má vzdialenosť od modrého $k = 5$ a od zeleného má ich vzdialenosť hodnotu $k = 2$. V danom prípade by bol sivý prvok klasifikovaný do zelenej triedy.^[9]



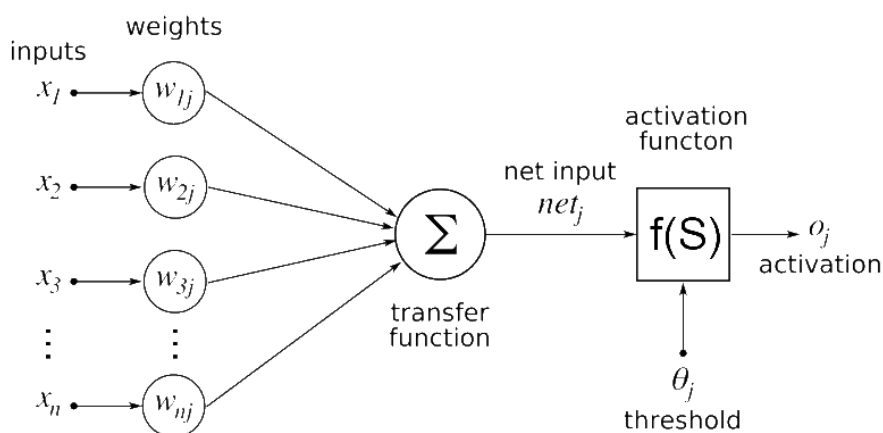
Obr. 15. Algoritmus K-najbližších susedov + DTW výpočet vzdialenosti

3.3 Neurónové siete

Väčšina klasifikátorov štatistických metód je založený na rozhodnutí podľa Bayesovej vety, kde jednotlivé elementy triedia podľa pravdepodobnosti. Po prvom pohľade na výstup klasifikačnej neurónovej siete môžeme vidieť, že dostaneme percentuálnu zhodu s jednotlivou triedou a vstupným elementom. Aj napriek spoločným menovateľom priame porovnanie medzi týmito metódami nie je možné. Nemôžeme ich porovnať, lebo neurónové siete sú nelineárne metódy bez kritériálnej funkcie, nakoľko štatistické metódy sú obvykle lineárne a založené na nejakých dopredu definovaných kritériách.^[10]

3.3.1 Model umelého neurónu

Základný model neurónu dostaneme po zostrojení z lineárneho člena, kde vstupné normalizované signály sú vynásobené synaptickými váhami, ktoré následne sumarizujeme. Na získanej hodnote zo sumátora aplikujeme aktivačnú funkciu, ktorá v dôsledku tvorí výstup jedného neurónu. Výstupná hodnota aktivačnej funkcie je zvyčajne v intervaloch $\langle 0, 1 \rangle$ alebo $\langle -1, 1 \rangle$. Ďalej model neurónu môže obsahovať prah („threshold“) značením θ_j (viď. Obr. 16), ktorý má za úlohu, aby vstup bol dostatočne znížený pred aktivačnou funkciou. V opačnom prípade, keď chceme zvýšiť vstup do aktivačnej funkcie použijeme predpätie („bias“).



Obr. 16. Model umelého neurónu

Formálna definícia jedného umelého neurónu je nasledujúca:

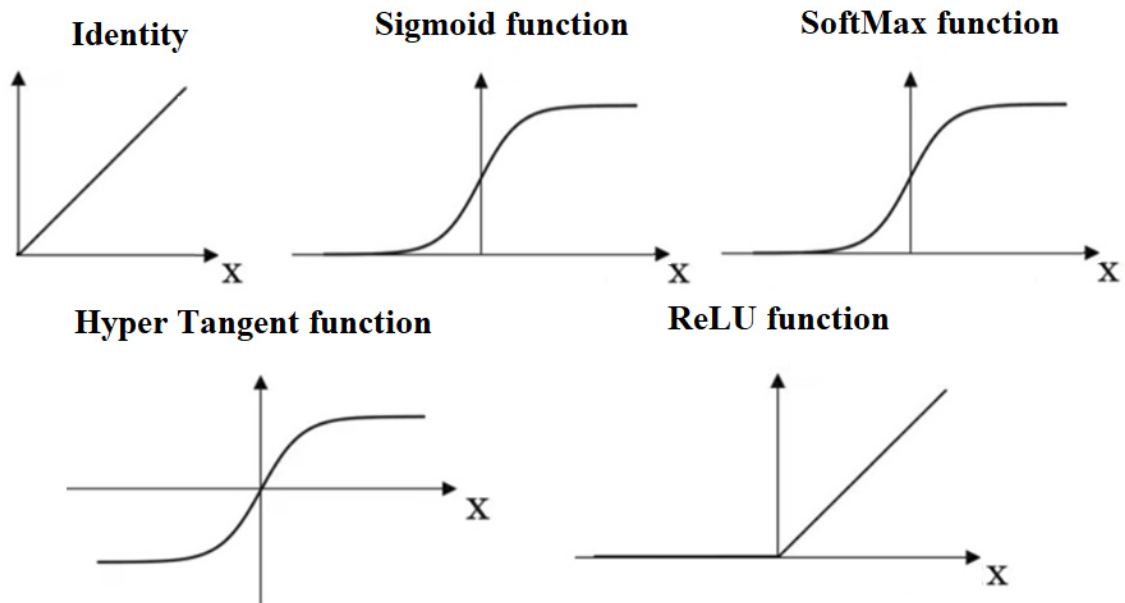
$$o_j = f(net_j - \theta_j) \quad (3.3)$$

$$net_j = \sum_{n=1}^j w_{nj} x_n \quad (3.4)$$

kde x_1, x_2, \dots, x_n sú vstupné signály pre neurónovú sieť, $w_{1j}, w_{2j}, \dots, w_{nj}$ sú jednotlivé synaptické váhy neurónu, net_j reprezentuje výstup lineárneho kombinátora, θ_j je hodnota prahu a $f(S)$ reprezentuje aktivačnú funkciu, kde je výstupom o_j .^[13]

3.3.2 Aktivačná funkcia

Z aktivačných funkcií existuje veľa variantov, ale nie každý typ je vhodnou voľbou pre klasifikačné siete. Aby sme zvýšili úspešnosť kategorizácie vybrali sme päť typov funkcií: *identity*, *sigmoid*, *softmax*, *tanh* a *relu*. Jednotlivé funkcie sú reprezentované na Obr. 17.



Obr. 17. Aktivačné funkcie použité pre klasifikačné siete

Prvou aktivačnou funkciou je *identity*, ktorú definujeme:

$$f(S) = a \cdot net \quad (3.5)$$

kde a je konštanta ovplyvňujúca strmlosť. Daná funkcia je málokedy využitá v praxi, lebo viac vrstiev lineárnych prvkov je možné nahradiť jednou nelineárnou metódou. Výstupné hodnoty sú pre tento typ v intervale $\langle -inf, inf \rangle$.^[13]

Druhá funkcia je *sigmoidálna* funkcia, ktorá má tvar:

$$f(S) = \frac{1}{1 + e^{-net}} \quad (3.6)$$

Výhoda sigmoidu spočíva v tom, že je nelineárny, diferencovateľný a spojitý. Kvôli týmto vlastnostiam pri aplikácii vo viacvrstvovej sieti je možné spočítať a vyhodnotiť ľubovoľnú reálnu funkciu. Pri klasifikácii sa Sigmoidová funkcia používa pre dvojtriednu logistickú regresiu. Výstupné hodnoty sú pre tento typ v intervale $\langle 0, 1 \rangle$.

Pre *softmax* aktivačnú funkciu platí:

$$f(S) = \frac{e^{net}}{\sum e^{net}} \quad (3.7)$$

ktorá je všeobecnejšou logistickou funkciou ako je sigmoid a práve preto ju môžeme považovať za jej vylepšenú formu. Tvar krivky a výstupnú hodnotu má podobnú, ako je to v prípade sigmoidu, ale jej veľká výhoda sa prejaví pri klasifikácii do viacerých tried.

Hyperbolický tangens je definovaný nasledovne:

$$f(S) = \frac{(e^{net} - e^{-net})}{(e^{net} + e^{-net})} \quad (3.8)$$

ktorý má tvar krivky podobný, ako predošlé dve funkcie. V niektorých prípadoch musíme mať výstupnú hodnotu v intervale $\langle -1, 1 \rangle$ a v tom prípade je tanh najlepšou voľbou. Ďalšia výhoda oproti sigmoidu spočíva v tom, že záporné vstupy budú mapované v blízkosti -1, kým pozitívne hodnoty budú vo veľmi úzkom priestore pod jednotkou.

Ako posledný reprezentant aktivačných funkcií, ktorý sme použili v našej práci je *reLU*. Je najjednoduchšou formou nelinearity, ktorú určíme nasledovným spôsobom:

$$f(S) = \begin{cases} 0 & \text{pre } x < 0 \\ x & \text{pre } x \geq 0 \end{cases} \quad (3.9)$$

Silnou stranou daného typu aktivačnej funkcie spočíva v tom, že viacvrstvová sieť zložená z takýchto neurónov je schopná spočítať teoreticky akúkoľvek pravdivostnú funkciu. Je veľmi jednoduchá a účinná pri Booleovských operáciách. Jej veľkou nevýhodou je však potlačenie záporných hodnôt na vstupe, kvôli čomu môžu nastať ťažkosti pri učení sietí.

3.3.3 Trénovanie neurónových sietí

Najčastejšia metóda učenia neurónových sietí je trénovanie iteračným spôsobom. To vo všeobecnosti znamená, že v každej iterácii sú jednotlivé váhy korigované podľa nasledujúceho vzťahu:

$$w_{nj}(i+1) = w_{nj}(i) + \Delta w_{nj}(i) \quad (3.10)$$

kde i je aktuálna iterácia, $w_{nj}(i)$ je váha neurónovej siete a $\Delta w_{nj}(i)$ reprezentuje korekciu, ktorú aplikujeme na váhu. Táto metóda je navrhnutá na báze učenia pomocou učiteľa. Učiteľa tu reprezentuje databáza, kde sú jednotlivé vzory roztriedené a slúžia ako správna odpoveď pre sieť. Pri tejto metóde je nevyhnutnou podmienkou dostatočne veľké množstvo údajov, aby bol výsledok čo najpresnejší. Počet iterácií treba zvoliť podľa množstva dát, pretože pri malom množstve môže nastať prípad, keď sieť nebude dostatočne natrénovaná a bude reagovať len na špecifické údaje.

3.3.4 Implementácia neurónového siete

Na klasifikáciu našich EEG signálov sme zvolili multiperceptronovú sieť, ktorá bola nakódovaná v programovacom jazyku Python. V jazyku C by síce prebehol trénovací cyklus oveľa rýchlejšie, ale nami zvolený jazyk bol vybraný kvôli ľahkej integrácii do GUI.

Program bol navrhnutý ako modul, čo znamená že aj bez GUI aplikácie je plne funkčný a integrovateľný s ostatnými programami. Samotný modul sa skladá z ôsmich skriptov, ktoré sú: *main.py*, *read_csv.py*, *read_mlp.py*, *write_mlp.py*, *mlp_train.py*, *back_propagation.py*, *forward_propagation.py* a *mlp_classifier.py*.

Skript *main.py* je mostom medzi GUI aplikáciou a ostatnými skriptami ktoré zabezpečujú prácu samotnej neurónovej siete. Pri spustení programového kódu je nutné zadať vstupné parametre, aby proces prebehol správnym spôsobom. Podľa funkcionality je možné zadať vstupné parametre dvomi spôsobmi. Prvú metódu je možné použiť v prípade, keď chceme natrénovať nanovo vytvorenú sieť alebo pretrénovať už existujúcu neurónovú sieť. Pri tejto metóde je nutné zadať nasledujúce vstupné argumenty:

- počet skrytých vrstiev (3),
- veľkosť jednotlivých skrytých vrstiev (4, 5, 5),
- typ aktivačnej funkcie pre jednotlivé skryté vrstvy (sigmoid, tanh, softmax),
- konštanta α zodpovedá miere učenia siete (0.1),
- maximálny počet iterácií (1000),
- počet výstupov z neurónového siete (2),
- typ aktivačnej funkcie pre výstupnú vrstvu (relu),
- meno súboru s trénovacími dátami (train.csv),
- počet riadkov trénovacích dát v súbore (200),
- počet stĺpcov trénovacích dát vrátane výstupného signálu v súbore (5),
- meno súboru s testovacími dátami (test.csv),
- počet riadkov v báze testovacích dát (100),
- počet stĺpcov trénovacích dát vrátane výstupného signálu v súbore (5),
- meno výstupného súboru pre natrénovanú sieť (mlp1.csv),
- voliteľný argument, meno vstupnej neurónovej siete (mlp.csv).

Druhá metóda je zadanie vstupných dát na klasifikáciu v reálnom čase. Tu je nutné zadať nasledujúce vstupné dáta:

- meno súboru obsahujúce váhy neurónovej siete (mlp.csv),
- počet prichádzajúcich dát na vstup do siete (5).

Po získaní vstupných parametrov skript najprv vyhodnotí, či sú argumenty relevantné alebo nie a následne zvolí metódu podľa počtu zadáných argumentov. Pri akejkol'vek nezhode skript vráti chybovú správu o chybnom parametre. Keď kontrola prebehne úspešne, skript postupne začne volať ostatné kódy k príslušnej metóde.

Skript *read_csv.py* je používaný na otvorenie súborov a načítanie dát, s ktorými bude sieť natrénovaná a otestovaná. Proces načítania prebehne podľa dopredu definovaných vstupných parametrov.

Programový kód s názvom *read_mlp.py* používa podobné nástroje, ako predošlý skript. Pomocou tohto programu vieme načítať jednotlivé váhy, ktoré reprezentujú neurónovú sieť. Prvým krokom je načítanie prvého stĺpca, kde sú uložené špecifikácie neurónovej siete s podobným štýlom ako bol deklarovaný pri vstupných parametrov v prvej metóde.

Skript *write_mlp.py* je presným opakom predošlého programu, pretože tento skript vytvorí .csv súbor pre váhy neurónovej siete. Do prvého stĺpca uloží vstupné podmienky, ktoré boli deklarované pri spustení programu a do ostatných stĺpcov uloží jednotlivé vrstvy siete.

V programovom kóde *mlp_trainer.py* prvým krokom je inicializácia jednotlivých váh v prípade, keď chceme vytvoriť novú neurónovú sieť. Váhy sú generované náhodným spôsobom v intervale $(-1, 1)$. Po vytvorení váh začne trénovací cyklus, kde v každom cykle pomieša náhodným spôsobom trénovacie dáta. Ako ďalšia fáza iterácií je volanie funkcie dopredaného šírenia a spätného šírenia. V cykle je aj periodicky volaná funkcia *mlp_classifier* (je súčasťou súboru *mlp_classifier.py*), ktorá nám slúži ako spätná väzba vyjadrujúca mieru úspešnosti siete pri klasifikácií.

Úlohou programového kódu *forward_propagation.py* je dopredné šírenie siete, kde najprv vynásobí vstupné dáta s váhami reprezentujúcimi prvú skrytú vrstvu a následne ich sumarizuje. Po sumarizácii hodnôt aplikujeme aktivačnú funkciu, ktorá bola zadaná pri spustení programu. Výstupná hodnota aktivačnej funkcie následne tvorí vstupnú hodnotu pre ďalšiu vrstvu. Daný postup je opakovaný dovtedy, kým šírenie siete nedosiahne poslednú „výstupnú“ vrstvu. Hodnoty, ktoré boli vypočítané vo výstupnej vrstve, sú uložené a následne ďalej použité v spätnom šírení.

Po výpočte dopredného šírenia je zavolaná funkcia *back_propagation*, ktorá je implementovaná v súbore *back_propagation.py*. Prvým krokom je inicializácia očakávaných výstupných hodnôt a následne definovanie zoznamu pre jednotlivé váhové korekcie. V druhom kroku program vypočíta lokálne gradienty podľa očakávaných výstupov

pre jednotlivé vrstvy neurónovej siete. Po dokončení výpočtov lokálnych gradientov ich vynásobíme s konštantou α , ktorá zodpovedá miere učenia neurónovej siete a výstupnými hodnotami, ktoré sme dostali v časti dopredné šírenie. Posledným krokom danej programovej funkcie je, že program sčíta korekciu Δw_{nj} s váhami w_{nj} .

Posledný skript modulu neurónovej siete je *mlp_classifier.py*, v ktorom sa nachádzajú dve hlavné funkcie. Prvá funkcia sa využíva pri tréovaní siete a druhá je pre klasifikáciu v reálnom čase. Najskôr sa otestuje s testovacími dátami, v akej miere je presná naša natrénovaná neurónová sieť. V prvom kroku program vezme všetky testovacie dáta a spustí ich cez sieť, ako to bolo aplikované v prípade dopredného šírenia. Následne len zozbiera výstupné dáta zo siete a uloží ich do zoznamu. Po dokončení tohto procesu program porovná jednotlivé výstupy so vzorovými a zostrojí maticu zmätku (confusion matrix). Podľa matice zmätku sa potom vypočíta presnosť siete, v akej miere vedela roztriediť jednotlivé vzory. Výstupom funkcie je presnosť neurónovej siete. Druhá funkcia v danom skripte berie dáta z OpenBCI, ktoré sú už po filtrácii. V tomto prípade sú vstupné reálne dáta pustené cez sieť a výstupné parametre sú aj výstupom funkcie, ktorá už ukazuje, do ktorej skupiny patrí daný signál.

3.4 Výsledky porovnania kategorizačných metód

Aby sme mohli porovnať zvolené metódy, museli sme vytvoriť databázu na tréovanie a testovanie. Vzory sme museli vytvoriť na motorické aj senzorické očné funkcie. Aby experiment bol čo najpresnejší, pripravili sme približne 225 (75%) tréovacích a 75 (25%) testovacích dát, ktoré boli rozdelené do 4 kategórií pre obidve metódy. K jednotlivým kategóriám príslušné príkazy sú: dopredu, doľava, doprava a stop.

Pri motorickej očnej funkcii sú prítomné 4 rozličné signály:

- Žmurknutie oboma očami je príkaz chod' *dopredu*,
- Žmurknutie ľavým okom je chod' *doľava*,
- Žmurknutie pravým okom je chod' *doprava*,
- Zatvorenie oka je príkaz zastavenia - *stop*.

V prípade senzorických očných funkcií sme nasníмали rôzne frekvencie, kde detegovaná frekvencia je príslušná k príkazu:

- 3 Hz, chod' dopredu,
- 5 Hz, chod' doľava,
- 7 Hz, chod' doprava,
- 11 Hz je stop.

Tieto frekvencie sme vybrali z dôvodu, že ide o prvočíselné hodnoty, ktoré sa dajú ľahko rozpoznať aj v prípade existencie ich násobkov (vyšších harmonických zložiek neharmonických periodických signálov). Hodnoty boli zvolené aj s ohľadom na obnovovaciu frekvenciu monitora, ktorý pracuje na 60 Hz. Čím vyššia bola frekvencia signálov, tým viac medzných zložiek sa vynechával a napr. sínusový signál bol skoro totožný s obdĺžnikovým. Pri testoch sme použili obdĺžnikové a sínusové signály.

V tabuľkách sú reprezentované najlepšie dosiahnuté výsledky a ostatné testy sa nachádzajú v Prílohe B. Tabuľky obsahujú matice zmätku, priemer a vážený priemer klasifikácie inštrukcií.

Tab. 1: Výsledky klasifikácie pomocou neurónovej siete na motorické očné funkcie

	Stop [počet]	Dopredu [počet]	Dol'ava [počet]	Doprava [počet]	Priemer [%]
Stop [počet]	58	0	1	0	98,31
Dopredu [počet]	1	64	2	0	95,52
Dol'ava [počet]	1	1	97	1	97,00
Doprava [počet]	0	0	4	74	94,87
Vážený priemer [%]					96,43

Tab. 2: Výsledky klasifikácie pomocou neurónovej siete na senzorické očné funkcie

	3 Hz [počet]	5 Hz [počet]	7 Hz [počet]	11 Hz [počet]	Priemer [%]
3 Hz [počet]	58	9	7	13	66,67
5 Hz [počet]	9	46	6	11	63,89
7 Hz [počet]	12	5	41	8	62,12
11 Hz [počet]	6	20	9	44	55,70
Vážený priemer [%]					62,10

Tab. 3: Výsledky klasifikácie pomocou kNN+DTW na motorické očné funkcie

	Stop [počet]	Dopredu [počet]	Dol'ava [počet]	Doprava [počet]	Priemer [%]
Stop [počet]	25	0	0	0	100,00
Dopredu [počet]	0	25	2	0	100,00
Dol'ava [počet]	0	0	24	1	96,00
Doprava [počet]	0	0	2	23	92,00
Vážený priemer [%]					97,00

Tab. 4: Výsledky klasifikácie pomocou kNN+DTW na senzorické očné funkcie

	3 Hz [počet]	5 Hz [počet]	7 Hz [počet]	11 Hz [počet]	Priemer [%]
3 Hz [počet]	10	3	5	7	40,00
5 Hz [počet]	4	14	2	5	56,00
7 Hz [počet]	1	4	17	3	68,00
11 Hz [počet]	8	5	5	7	28,00
Vážený priemer [%]					48,00

Pred testom, ktorého výsledky obsahuje Tab. 1 sme nastavili neurónovú sieť. Sieť má dve skryté vrstvy - obe majú aktivačnú funkciu tanh, kde prvá vrstva sa skladá z 160 prvkov a druhá z 80 prvkov. Výstupná vrstva má 4 výstupy a aktivačnú funkciu sigmoid. Hodnoty, ktoré sú reprezentované v Tab. 1 sme dosiahli za 50 cyklov učenia - vyšší počet asi by len zhoršil dosiahnuté výsledky. Pri druhom pokuse, kde sme trénovali sieť na senzorické očné funkcie, sme museli navýšiť počet prvkov v skrytých vrstvách na 250 a 125, lebo sieť nedokázala klasifikovať. Tieto neúspešné pokusy sú obsiahnuté v Prílohe B.

Po klasifikácii neurónovej siete sme urobili triedenie pomocou kNN+DTW, kde sme museli redukovať počet vzorov na tretinu, pretože to bolo časovo veľmi náročné. Aby bol náš experiment validný, počet vzorov sme vybrali náhodným spôsobom z pôvodnej množiny, a to z každej po 25 prvkov. Takto sme dostali výsledky Tab. 3 a Tab. 4.

Na experiment sme použili 3 rôzne databázy na senzorickú aj motorickú očnú funkciu. Ako je zrejmé z tabuliek, motorické očné funkcie sa dajú klasifikovať omnoho lepšie ako senzorické.

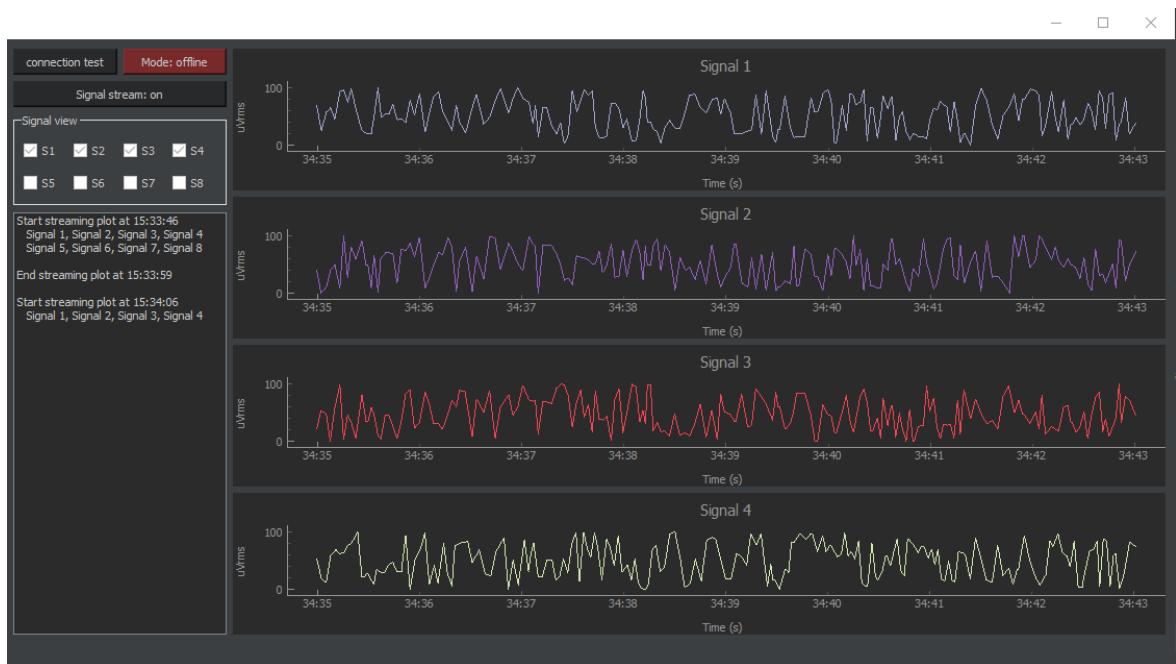
4 GUI pre riadenie a tréovanie

Vizuálnu časť diplomovej práce sme vytvorili pomocou Qt Designer od spoločnosti The Qt Company, kde sme implementáciu programu uskutočňovali pod Open Source Qt, ktoré patrí do licenčnej zmluvy GNU Lesser General Public Licence. Ďalšie časti boli vytvorené pomocou jazyka Python.

Na začiatku sme sa rozhodli pre vytvorenie jednej aplikácie, ktorá bude ovládať celý proces. Najjednoduchšiu realizáciu sme dosiahli výberom kartovej štruktúry (tab structure), kde jednotlivé karty plnia rozličné úlohy. Prvé dve karty obsahujú priestor na vytvorenie vzorových dát. Tretia karta je vytvorená na tréovanie neurónovej siete resp. na vytvorenie referenčných príkladov pre kNN + DTW. Na poslednej karte je predstavený náš virtuálny robot, ktorý je riadený pomocou EEG signálov.

4.1 Základné funkcie vytvorenej aplikácie

Po spustení nášho programu, na ľavej strane nájdeme jednotlivé karty, s ktorými sa budeme detailnejšie zaoberať v ďalších podkapitolách. V pravej časti aplikácie (Obr.18) v pravom hornom rohu vidíme, že sa tam nachádzajú tri tlačidlá. Prvé tlačidlo je *connection test*, ktorý slúži na otestovanie, či sa dá vytvoriť spojenie s prístrojom OpenBCI. Druhým tlačidlom je *Mode: offline/online*, ktorý funguje ako prepínač. V režime off-line môžeme rovnako generovať náhodné signály na grafe a spúšťať jednotlivé záznamové karty, pretože v danom prípade nie je nutnou podmienkou, aby bol prístroj pripojený k počítaču. Posledné tlačidlo slúži pre zobrazenie signálov, ktoré sú reprezentované na pravej strane (Obr. 18). Výber jednotlivých signálov, ktoré chceme v danom čase zobrazit', môžeme vybrať v sekcii *Signal view*. Pridať alebo odstrániť nežiadúce grafy signálov sa dá jedine vo vypnutom resp. off-line stave. V ľavom dolnom štvorci po stlačení tlačidla sa zobrazujú jednotlivé správy, ale takisto sa tu môžu zobrazit' aj chybové hlásenia.



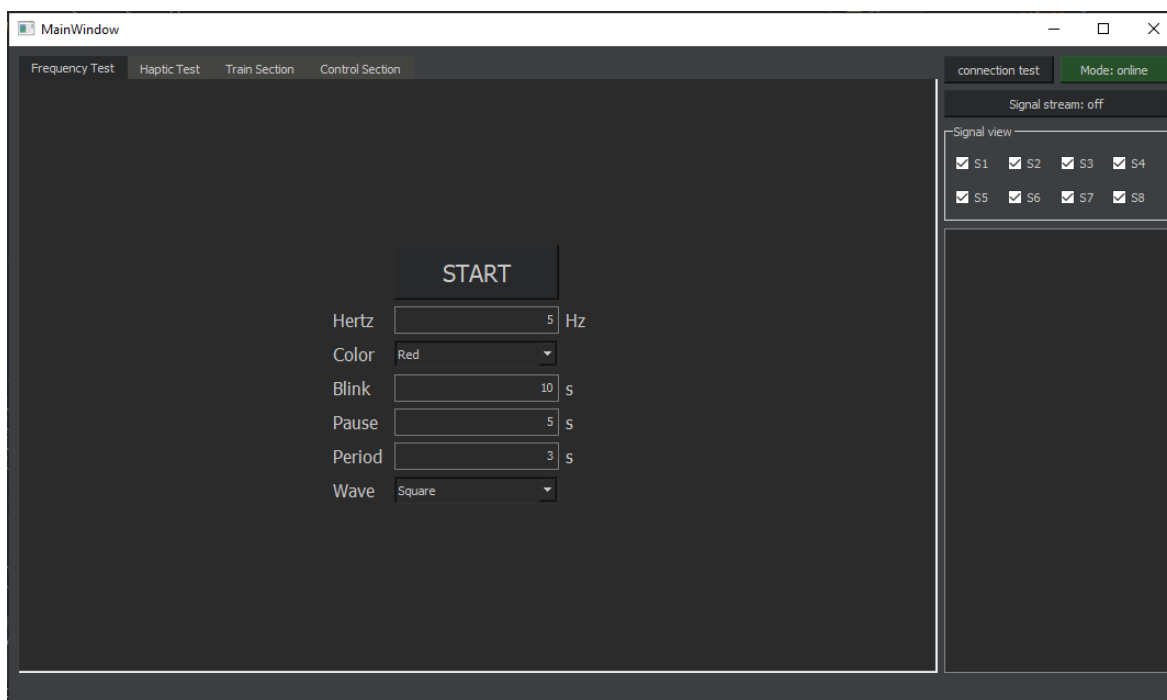
Obr. 18. Zobrazenie signálov prichádzajúce z prístroja OpenBCI

4.2 Vytvorenie trérovacích dát pomocou GUI

Po spustení GUI aplikácie sa nám objaví základná obrazovka s rozmerom 800x600, ktorá je zobrazená na Obr.19. Ako vidíme, na ľavej strane sa nachádzajú jednotlivé karty, ktoré obsahujú vstupné pole pre jednotlivé procesy. Na prvej karte, ktorú aktuálne vidíme, je priestor pre zaznamenanie a triedenie dát prichádzajúcich z OpenBCI. Aby sme dostali želané vzorky, ktoré sú vhodné pre spracovanie cez klasifikačné metódy, musíme zadať vhodné vstupné parametre. Každé prázdne pole očakáva pozitívne celé číslo, v opačnom prípade objaví sa chybové hlásenie. Vstupné parametre sú nasledujúce:

- Frekvencia (Herz) - sem zadáme hodnotu, na akej frekvencii bude blikať naša obrazovka,
- Farba (Color) - tu vyberieme jednu zo zopár dopredu definovaných farieb, ktoré slúžia na zmenu obrazovky pri blikaní,
- Blikanie (Blink) - tu sa definuje perióda, v akých časových úsekoch bude obrazovka v jednom cykle najdlhšie blikať,
- Pauza (Pause) - daným parametrom definujeme, na koľko sekúnd prestane obrazovka blikať na preddefinovanej frekvencii,
- Perióda (Period) - tu definujeme dĺžku periódy, kým bude naša aplikácia zaznamenávať dáta z prístroja OpenBCI ,

- Vlna (Wave) - tu môžeme vybrať z 4 typov priebehu vln ktoré môžu byť: obdĺžnikový, trojuholníkový, sínusový a pílovitý.



Obr. 19. Vytváranie záznamov na senzorické ovládanie

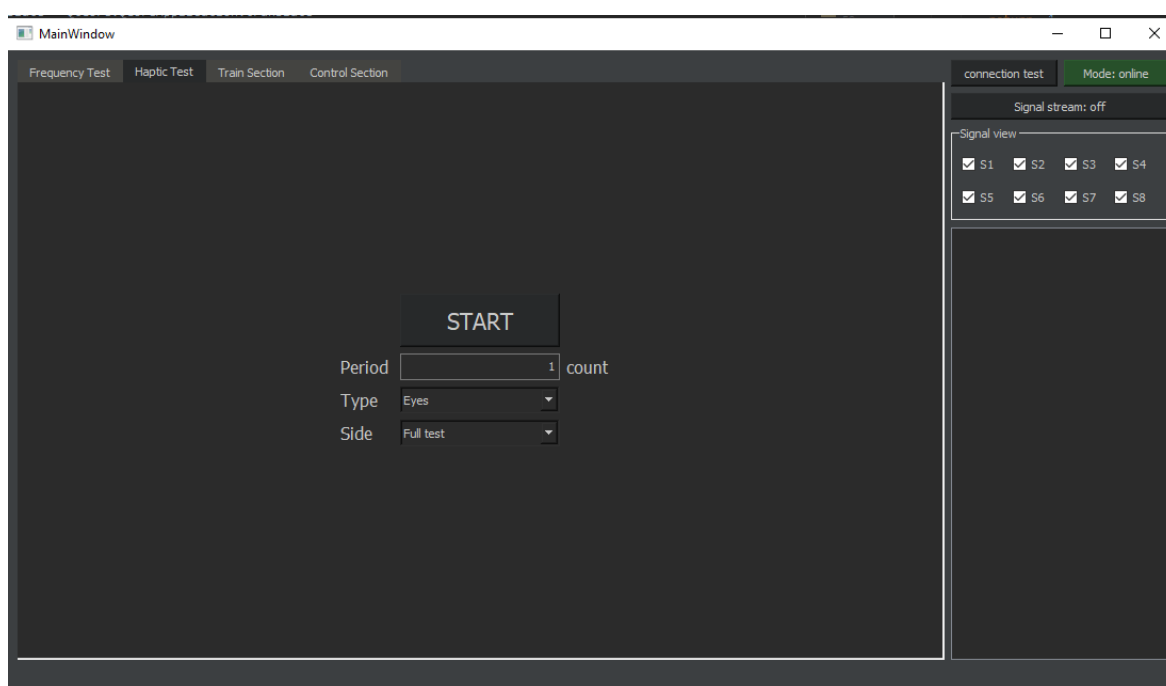
Po vyplnení prázdnych polí stlačíme gombík *štart*. Program sa potom rozdelí na dve vlákna. Prvé vlákno vytvorí kanál pre komunikáciu s prístrojom OpenBCI, kde zbiera dáta zo zariadenia a ukladá ich do zoznamu s časovou známkou. Druhé vlákno odstráni všetkých šesť polí slúžiacich pre vstupné dáta a vytvorí prázdne plátno. V strede plátna sa začne odpočítavanie do štartu aplikácie a tvorby záznamu. Toto vlákno zaznamená do zoznamu, v akej fáze je program a či práve bliká alebo je v stave pozastavenia. Tieto záznamy sú označené jednotkou alebo nulou a pri každom jednom je priradená aj tzv. časová známka. Po skončení periódy sa vlákna zlúčia a naše dve zoznamy sa spoja resp. synchronizujú do jedného spoločného dátového zoznamu na základe časovej známky, ktoré sú následne uložené do .csv súboru.

Na Obr. 20 vidíme druhú kartu, ktorá je pripravená na zaznamenanie motorických funkcií. Vstupné parametre sú nasledujúce:

- Perióda (Period) - sem zapíšeme celé číslo vyjadrujúce, koľkokrát chceme zopakovať zvolenú periódu,
- Typ (Type) - v tejto časti sú predefinované činnosti, na ktoré sa dajú urobiť motorické testy. Konkrétne typy sú: oči, ruky a nohy,

- Strana (Side) - tu určíme stranu na ktorú chceme urobiť jednotlivé vzory. Môžeme zvoliť jeden zo štyroch volieb: ľavá strana, pravá strana, obidve strany naraz alebo komplet (všetky predošlé voľby budú tvoriť jednu periódu)

Po stlačení tlačidla program takisto vytvorí dve vlákna, ako to bolo pri predošlej karte. Zmena sa prejaví pri druhom vlákne, keďže tu nebude blikat monitor. Aby bola vhodným spôsobom zachytená motorická interakcia, program vypíše na obrazovku odkaz, akú aktivitu musíme spraviť, aby sme skončili odpočítavanie. Čas na realizáciu jednej aktivity je individuálne uvedený podľa náročnosti aktivity. Po skončení periódy program zlúči vlákna a synchronizuje oba zoznamy, ktoré boli vytvorené pri zázname. Následne zlúči zoznam a uloží ho podobným spôsobom, ako to bolo v predošlom prípade.

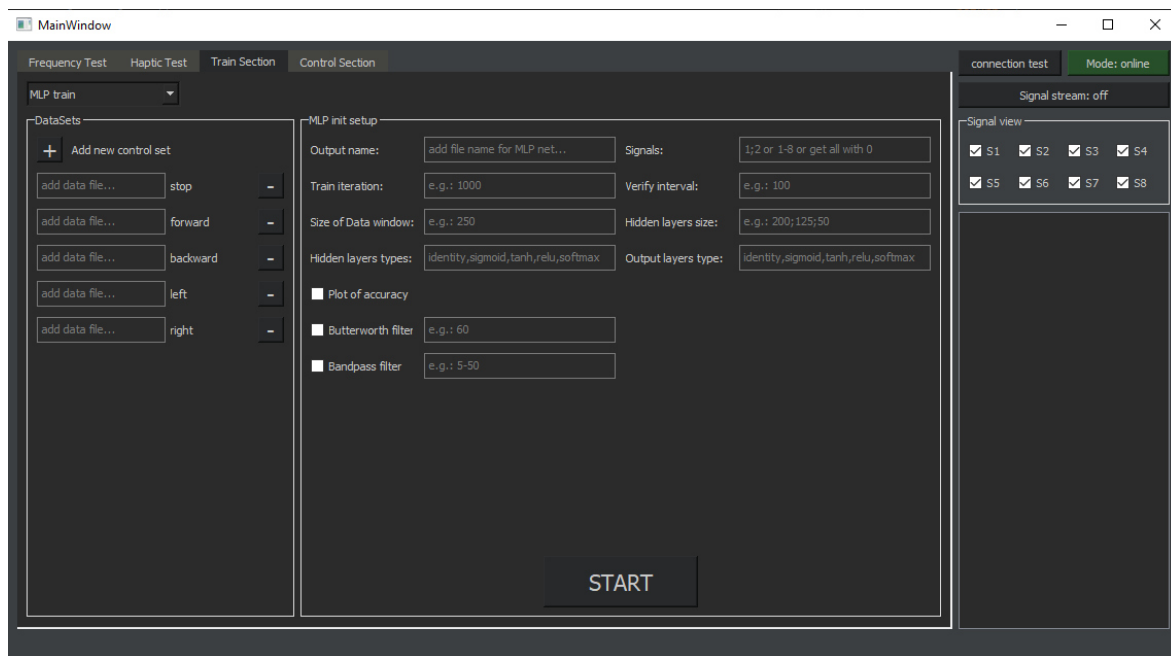


Obr. 20. Vytváranie záznamov na motorické ovládanie

4.3 Trénovanie neurónovej siete a vytvorenie vzorcov pre kNN+DTW

Tretia karta obsahuje tréningovú sekciu, kde sa množstvo vstupných dátových polí mení podľa typu klasifikačnej metódy. Vybrať môžeme medzi možnosťami, v závislosti od toho, či chceme trénovať neurónovú sieť alebo pripraviť vzory pre klasifikáciu kNN+DTW. Štruktúra rozloženia je pri oboch prípadoch realizovaná tým istým spôsobom. Ľavá strana tréningovej sekcie obsahuje tie isté elementy pri oboch spôsoboch vid'. Obr. 21. V tejto časti je už pár dopredu definovaných riadiacich príkazov, ktoré môžeme ľubovoľne odstrániť, ale zároveň je možné k nim pridať nové príkazy. Do jedného poľa je možné pridať aj viac súborov, ktoré chceme aby program načítal a použil ich na tréningovanie klasifikátora.

Jednotlivé súbory je možné pridať pomocou troch spôsobov. Prvý spôsob je, keď zadáme meno súboru – v tom prípade berie z predvoleného priečinka s názvom *data/*. Druhý spôsob je, keď napíšeme meno súboru a k tomu aj celú cestu (alebo môžeme na pole aj dvakrát kliknúť). Pri voľbe posledného spôsobu sa objaví prieskumník a v ňom môžeme ľubovoľne zvoliť súbory, ktoré chceme použiť na tréningovanie.



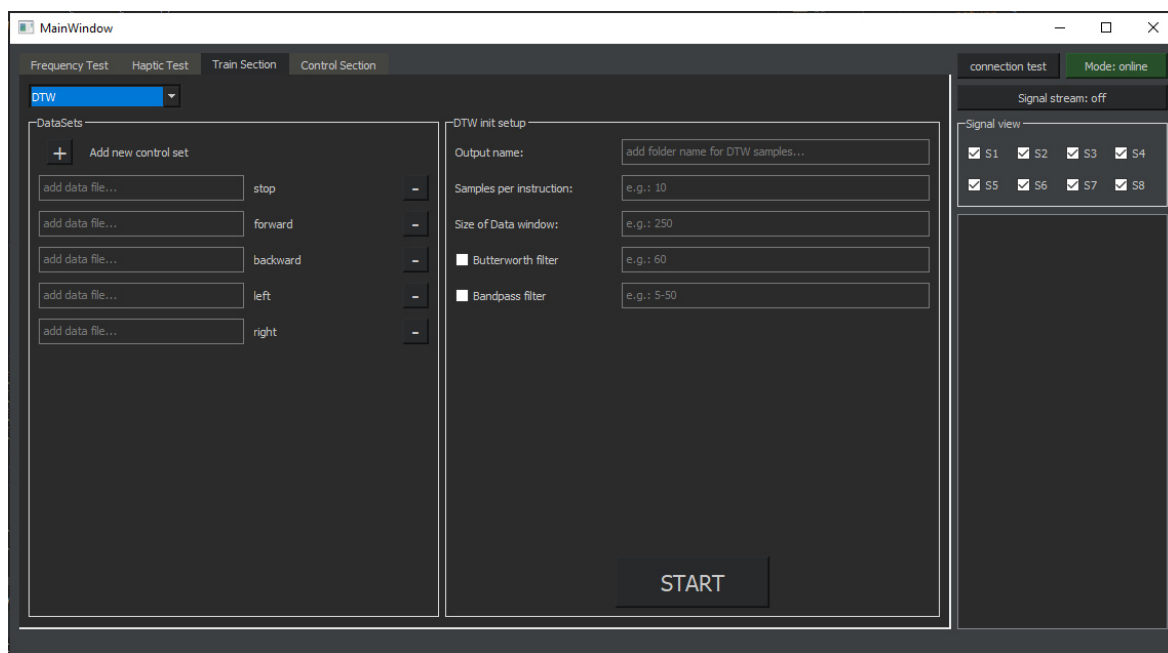
Obr. 21. Sekcia pre tréningovanie neurónového siete

Pravá strana karty je špecificky vytvorená pre danú klasifikačnú metódu ktorá očakáva nasledujúce vstupné parametre:

- Výstupné meno (Output name) - sem je potrebné napísať cestu plus meno súboru alebo len meno súboru, v ktorom si prajeme uložiť natréňovanú neurónovú sieť,
- Signály (Signals) – sem sa zadávajú signály, ktoré budú použité ako vstup neurónovej siete,
- Iterácia (Train iteration) - definujeme tu počet cyklov, počas ktorých sa bude sieť učiť,
- Overovací interval (Verify interval) - jeho uvedenie je nevyhnutné v prípade, keď chceme vidieť, s akou presnosťou dokáže klasifikovať testovacie dáta po uplynutí daných iterácií,
- Veľkosť dátového okna (Size of data window) - týmto vstupným parametrom je daný počet dát, ktoré budú slúžiť ako vstup do siete,
- Veľkosť skrytej vrstvy (Hidden layers size),
- Typ skrytých vrstiev (Hidden layers types),

- Typ výstupnej vrstvy (Output layers type),
- Graf presnosti (Plot of accuracy) - keď je odľaknutý, po štarte sa zobrazí aktuálna presnosť siete, ktorá sa obnoví po uplynutí každého overovacieho intervalu,
- Butterworthov filter - máme na výber si zvoliť aplikáciu filtra na naše vstupné signály, ale nie je to povinné,
- Bandpass filter - takisto si môžeme vybrať aplikáciu filtra.

Po stlačení tlačidla štart program načíta všetky zadané dátové súbory a začne sa predspracovanie dát. Skontroluje sa zadanie jednotlivých filtrov a začne sa ich vynechanie resp. ich aplikácia na jednotlivé signály, ktoré budú použité pri tréňovaní siete. V nasledujúcom kroku program rozdelí dáta na základe veľkosti dátového okna podľa zadanej hodnoty. K takto získaným hodnotám sa priradí hodnota, ktorá určuje ich triedu. Týmto sa končí fáza predspracovania a program zavolá modul na tréňovanie neurónovej siete. Na záver, uložíme do súboru aj sieť aj inicializačné (vstupné) dáta, pre budúce použitie (okrem iného aj pri kontrole robota).



Obr. 22. Sekcia pre vytvorenie vzorcov pre kNN+DTW

Keď vyberieme druhú možnosť, teda DTW (viď. Obr. 22) zadané vstupné parametre budú nasledujúce:

- Výstupné meno (Output name) - sem je potrebné napísať cestu plus názov priečinka alebo len samotný názov priečinka, v ktorom budú uložené jednotlivé vzorky, teda dáta tvoriace referenciu danej triedy,
- Signály (Signals) - tu sa zadávajú signály použité ako vzory pre DTW,

- Počet vzorov na inštrukciu (Samples per instruction) - na každý príkaz sa vytvorí podpriechinok, v ktorom budú jednotlivé vzory,
- Veľkosť dátového okna (Size of data window) - týmto vstupným parametrom je daný počet dát, ktoré budú vstupom DTW,
- Butterworthov filter - môžeme vybrať, či chceme alebo nechceme aplikovať filtráciu na naše signály,
- Bandpass filter - ten istý postup, ako v prípade predošlej filtrácie.

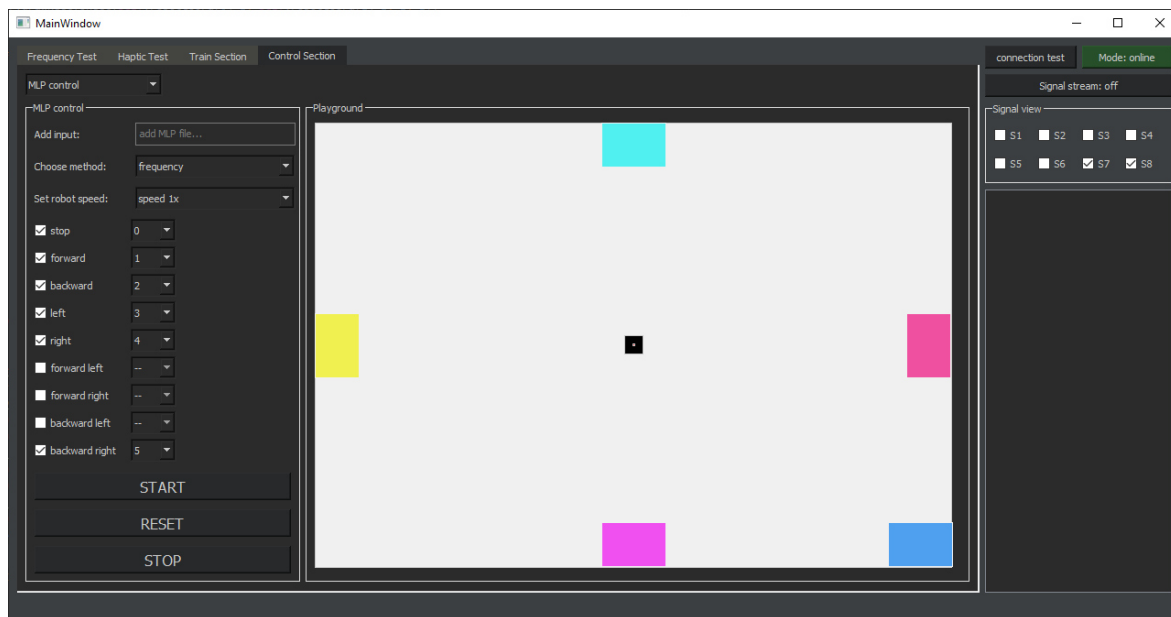
Prvým krokom je zadanie vstupných parametrov a následne stačí kliknúť na tlačidlo štart. Program najprv načíta všetky dáta a potom aplikuje filtre (v prípade, že táto možnosť bola zvolená). Po filtrácii program vytvorí špecifické dĺžky signálov podľa dátového okna a uloží ich ako .csv súbor do jednotlivých podpriechinok. Keď program dokončí túto fázu, skončí svoj proces a čaká na ďalšiu úlohu.

4.4 Riadenie mobilného robota pomocou EEG signálov

Posledná karta je vytvorená na validáciu našich metód. Keď dokážeme navrhnutými spôsobmi riadiť virtuálneho robota, potom je tento spôsob riadenia je aplikovateľný aj na reálny mobilný robot. Rovnako ako v prípade predošlej karty, aj tu si môžeme vybrať, akú metódu chceme aplikovať. Po výbere vhodnej metódy sa nám rozhranie programu rozdelí na dve časti a zobrazia dve tabuľky (combo boxes) zobrazené na Obr. 23.

Prvý box, ktorý sa nachádza na ľavej strane obrazovky, slúži na inicializáciu našej metódy. Táto časť je totožná v oboch prípadoch, líšia sa jedine v tom, že pokiaľ v prípade neurónovej siete zadáme meno súboru, tak pri DTW je do prvého riadku zadané meno priechinka. V druhom riadku môžeme zvoliť, podľa ktorej metódy chceme robota riadiť - pomocou senzorickej alebo motorickej očnej funkcie. V treťom riadku nastavujeme rýchlosť robota, kde už nájdeme niekoľko dopredu definovaných možností. Pod týmito parametrami sú definované jednotlivé riadiace príkazy. Na riadenie nášho virtuálneho robota je možné nastaviť maximálne 9 príkazov. Zvolíme pole odkazujúce na príkazy, ktoré chceme použiť pri reálnom časovom riadení. Po výbere príkazov, ku každému jednému je nutné priradiť referenčné číslo, keďže podľa tohto sa klasifikuje neurónová sieť. Keď príkazu *stop* priradíme číslo 0, funguje ako default funkcia a keď sieť nerozozná ani jednu kategóriu tak sa aplikuje tento príkaz. V inicializačnom bloku sa nachádzajú ešte tri tlačidlá - *start*, *reset* a *stop*. Po stlačení tlačidla štart, program načíta neurónovú sieť a pripojí sa na OpenBCI. Následne sa rozbehne plátno mapy virtuálneho robota. Keď program dokončí inicializačný proces, začne zaznamenávanie dát, na ktorý proces aplikuje filtre (ktoré boli zvolené počas

trénovania siete) a v poslednom kroku ich klasifikuje. Po klasifikácii dostaneme hodnotu, do ktorej triedy patrí aktuálny signál a podľa hodnoty sa vykoná pohyb virtuálneho robota. Tento cyklus sa opakuje, pokiaľ sa nestlačí tlačidlo *stop*. Tlačidlo *stop* slúži nielen na prerušenie zaznamenávacieho cyklu, ale takisto sa odpojí od OpenBCI a zastaví neurónovú sieť. Posledným tlačidlom je *reset*, ktoré slúži na nastavenie robota do pôvodného stavu čo znamená, že robot bude opäť v strede mapy. V prípade metódy DTW je celý postup totožný a líši sa len zobrazenie mapy.



Obr. 23. Sekcia pre riadenie virtuálneho robota

Na pravej strane sa nachádza mapa pre virtuálny robot nazvaná ako „Playground“. Po zmene veľkosti aplikácie sa dynamicky zmení aj veľkosť jednotlivých elementov. Čiernym štvorcikom je reprezentovaný robot, ktorý budeme ovládať. Po vykonaní príkazu sa robot premiestni na novú pozíciu a na mape zanechá stopu, kde sa predtým historicky nachádzal. Pri voľbe metóde „haptic“ je mapa celkom prázdna a je tam viditeľný jedine virtuálny robot. Metóda „frequency“ zobrazí na okrajoch mapy toľko farebných štvorcov, koľko zadaných príkazov máme v inicializačnej sekcii. Tieto štvorce začnú blikáť po stlačení tlačidla *start*. Frekvenciu blikania berú zo súboru, kde sú uložené jednotlivé váhy neurónovej siete.

Keď je príkaz *stop* nastavený na preddefinovanú hodnotu nuly (pri neurónových sieťach), vykoná pohyb jedine v prípade zaznamenania príkazu naznačujúceho pohyb. V prípade DTW funguje na báze prepínača - keď zaznamenaný jeden príkaz, tento dodržiava a opakuje ho, kým nepribudne ďalší.

4.5 Výsledky riadenia mobilného robota

Posledná fáza práce je overenie funkcionality aplikácie, ktorú sme vytvorili. Pomocou prvej z dvoch kariet sme nasníмали niekoľko záznamov na motorické a senzorické očné funkcie. Následne sme natrénovali neurónové siete a vytvorili sme databázu pre metódu kNN+DTW. Keďže naším účelom bola lepšia a dôkladnejšia validácia, zopakovali sme predošlé procesy trikrát s rôznymi nastaveniami. Po skončení procesu sme vybrali najlepšie výsledky produkujúce neurónové siete a databázu na kNN+DTW. Aby robot vedel reagovať čo najrýchlejšie a sieť dostávala relevantné dáta, klasifikácia prebieha zakaždým keď nazbiera dvadsať nových dát, potom ich prefiltruje a pošle signál na triedenie. To prakticky znamená, že keď OpenBCI zaznamenáva dáta na 250 hertzov tak program má 80 ms času na filtráciu a klasifikáciu dát.

Pri prvom pokuse sme použili natrénované neurónové siete, ktorého výsledky boli reprezentované v Tab. 1. Následne sme začali riadiť virtuálneho robota. Po skončení testu sme konštatovali, že robot reagoval na naše príkazy správne až s 90-95% presnosťou. Druhému experimentu slúžili základom tie isté dáta, avšak natrénované boli pomocou metódy kNN+DTW. Pri tejto metóde sme dostali podobné výsledky ako pri predošlej. Pri totožnom počte dát, ako bolo použité na trénovanie neurónovej siete pri kNN+DTW, sme zbadali, že triedenie je dosť pomalé a príkazy na ovládanie robota sa oneskorovali. Túto problematiku sme riešili redukovaním databázy na 10 vzorov na inštrukciu. V tomto prípade náš program už zvládol dostatočne rýchlo klasifikovať poskytnuté údaje, ale percentuálne sa zhoršilo triedenie, V našom prípade to znamená, že správne uhádnuť kategóriu pre dané signály už dokázal iba na 80-90%. Keď takým to istým spôsobom znížime počet testovacích a trénovacích dát pre neurónovú sieť, dokáže zareagovať správne iba na 60%.

Druhá skupina experimentov bolo zameraná na senzorické očné funkcie, kde sme takisto pracovali s „najlepšími“ dátami. Postupy boli podobné ako v predošlom experimente. V danom teste reagoval robot len približne na 30-40% a v ostatných prípadoch robot dostával zlé inštrukcie. Chyba mohla nastať kvôli OpenBCI, a to nesprávnym nasadením elektród, kedy prístroj nevedel zaznamenať relatívne slabé signály. Ďalší zdroj problémov mohol pochádzať zo strany subjektu - napr. jeho oči nemuseli správne zaregistrovať prichádzajúce impulzy.

Na záver môžeme povedať, že neurónové siete sú lepšou voľbou, keď máme k dispozícii väčšiu databázu, lebo po natrénovaní táto metóda klasifikuje najrýchlejšie. Kombinácia kNN a DTW je zase lepšia v situácií, keď nemáme dostatočne veľkú databázu a narýchlo potrebujeme vytvoriť jednu dátovú zložku pre riadenie.

Záver

Cieľom diplomovej práce bolo naštudovanie funkcie mozgu a detegovanie EEG signálov. Následne sme urobili prehľad viacerých typov BCI prístrojov a opísali sme ich fungovanie. Po získaní teoretických znalostí ohľadom funkcie mozgu a možnosti záznamu EEG signálov, bolo potrebné naštudovať filtračné metódy, ktoré nám pomáhali odstrániť nežiaduce signály. Pomocou týchto vedomostí sme následne zaznamenali EEG signály, na ktoré sme aplikovali navrhnuté filtre.

Po úspešnej analýze a filtrácii sme mohli začať implementáciu prvej časti našej aplikácie. Táto časť programu slúži na vytvorenie a zaradenie vzorov do jednotlivých kategórií, aby nebolo nutné vytvoriť databázu manuálnym spôsobom. Keď bola táto časť dokončená, mohli sme pristúpiť na druhú fázu, v ktorej bolo potrebné naštudovať jednotlivé klasifikačné metódy, ktoré sa dajú použiť na triedenie dát. Na tento účel sme zvolili dve metódy, a to klasifikačné neurónové siete a dynamic time warping (DTW) v kombinácii s metódou k-najbližších susedov. Po osvojení vymenovaných vedomostí práca prechádza na praktickú časť, kde sme na základe potrebných informácií vytvorili ďalšiu sekciu našej aplikácie. Paralelne s týmto procesom bol vytvorený program, ktorý dokáže načítať zaznamenané dáta, vyfiltruje ich aj podľa potreby a následne ich navzorkuje do dopredu definovanej formy. Po dokončení procesov predspracovania, sme implementovali samostatný modul neurónovej siete a kNN+DTW. Na validáciu tejto časti sme vytvorili databázu naplnenú tréningovými a testovacími dátami, ktoré boli vyskúšané na oboch metódach a následne sme ich porovnali, aby sme posúdili, ktorá z nich funguje lepšie.

V poslednej fáze sme vytvorili programovú časť, pomocou ktorej sme validovali dosiahnuté výsledky. Keďže aktuálna karanténna situácia nám nedovolila overiť výsledky na reálnom mobilnom robote na Ústave robotiky a kybernetiky, bol implementovaný virtuálny robot, ktorý sme riadili pomocou EEG signálov v reálnom čase. Klasifikačná sieť dokázala rozpoznať 4 príkazy ktoré sme natrénovali. Po viacerých testoch sme konštatovali, že vo väčšine prípadov správne rozpoznal jednotlivé triedy a vedeli sme riadiť našu simuláciu. Asi najväčším problémom bolo, že program bol vytvorený v jazyku Python a niektoré výpočty trvali dlhšie, kvôli čomu náš robot reagoval pomalšie resp. oneskorene. Túto problematiku je možné vyriešiť v budúcnosti jednoduchým spôsobom - grafickú časť je možné ponechať naprogramovanú v jazyku Python a výpočtovo náročné časti preložiť do pre daný cieľ vyhovujúcejšieho jazyka, napr. do jazyka C.

Literatúra

- [1] STONE, James L – HUGHES, John R., (2013). Early History of Electroencephalography and Establishment of the American Clinical Neurophysiology Society. In: *Journal of Clinical Neurophysiology*. DOI: 10.1097/WNP.0b013e31827edb2d. 2003, February, vol. 30, no. 1 - p 28-44
- [2] MYSLIVEČEK, Jaromír. Základy neurověd. 2. vydání. Praha : Triton, 2009. ISBN 978-80-7387-088-1.
- [3] ABHANG, Priyanka – GAWALI, Dr.Bharti, (2015). Correlation of EEG Images and Speech Signals for Emotion Analysis. In: *British Journal of Applied Science & Technology*. DOI: 10.9734/BJAST/2015/19000. 2015, vol. 10. p. 1-13.
- [4] KUMAR, J. Satheesh – BHUVANESWARI, P., (2012). Analysis of Electroencephalography (EEG) Signals and Its Categorization–A Study. In: *Procedia Engineering*. DOI: 10.1016/j.proeng.2012.06.298. vol. 38, p. 2525-2536.
- [5] FU, Michael – DALY, Janis – CAVUSOGLU, M.C.. (2006). *Assessment of EEG event-related desynchronization in stroke survivors performing shoulder-elbow movements*: Proceedings - IEEE International Conference on Robotics and Automation. p. 3158 - 3164. DOI: 10.1109/ROBOT.2006.1642182. ISBN 0-7803-9505-0
- [6] TEPLAN Michal, (2002). Fundamentals of EEG measurement. In: *Measurement Science Review* [online]. 2002, section 2, vol. 2, p. 1-11.
- [7] Webová stránka: <http://learn.neurotechedu.com/introtobci/>, názov článku: *Intro to Brain Computer Interface*
- [8] CALVANO José. V. – ALVES V. C. – LUBASZEWSKI M., (2000). *Fault detection methodology and BIST method for 2nd order Butterworth, Chebyshev and Bessel filter approximations*. Proceedings 18th IEEE VLSI Test Symposium, Montreal, Quebec, Canada. DOI: 10.1109/VTEST.2000.843861. 2000, p. 319-324.
- [9] FERREIRA, João – COSTA, Clauber – MIRANDA, Ricardo – FLORENCIO, Antonio, (2015). The use of the k nearest neighbor method to classify the representative elements. In: *Educación Química*. DOI: 10.1016/j.eq.2015.05.004. 2015, vol. 20. ISSN: 0187-893X

- [10] GUOQIANG, Peter Zhang, (2000). Neural Networks for Classification: A Survey. In: *Ieee Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*. DOI: 10.1109/5326.897072. 2000, November, vol. 30, no. 4. p. 451-462
- [11] LIABOTIS, Ioannis – THEODOULIDIS, Babis – SARAEE, Mohamad, (2006). Improving Similarity Search in Time Series Using Wavelets. In: *IJDWM*. DOI:10.4018/jdwm.2006040103. 2006, vol. 2. p. 55-81.
- [12] ROHIT, Kate,. (2015). Using dynamic time warping distances as features for improved time series classification. In: *Data Mining and Knowledge Discovery*. p. 30. 10.1007/s10618-015-0418-x.
- [13] Ing ČECHOVIČ, Lukáš, (2013). *Klasifikácia signálov pomocou umelých neurónových sietí*. Dizertačná práca. Fakulta riadenia a informatiky, 2013. Žilinská univerzita v Žiline

Prílohy

Príloha A: Funkcia Dynamic Time Warping

```
def dtw_distance(self, ts_a, ts_b, d=lambda x, y: abs(x - y)):
    # Create cost matrix via broadcasting with max int
    ts_a, ts_b = np.array(ts_a), np.array(ts_b)
    M, N = len(ts_a), len(ts_b)
    cost = sys.maxsize * np.ones((M, N))

    # Initialize the first row and column
    cost[0, 0] = d(ts_a[0], ts_b[0])
    for i in range(1, M):
        cost[i, 0] = cost[i - 1, 0] + d(ts_a[i], ts_b[0])

    for j in range(1, N):
        cost[0, j] = cost[0, j - 1] + d(ts_a[0], ts_b[j])

    # Populate rest of cost matrix within window
    for i in range(1, M):
        for j in range(max(1, i - self.max_warping_window),
                        min(N, i + self.max_warping_window)):
            choices = cost[i - 1, j - 1], cost[i, j - 1], cost[i - 1, j]
            cost[i, j] = min(choices) + d(ts_a[i], ts_b[j])
    return cost[-1, -1]

def dist_matrix(self, x, y):
    # Compute the distance matrix
    dm_count = 0

    # Compute condensed distance matrix of pairwise dtw distances
    if (np.array_equal(x, y)):
        x_s = np.shape(x)
        dm = np.zeros((x_s[0] * (x_s[0] - 1)) // 2, dtype=np.double)

        for i in range(0, x_s[0] - 1):
            for j in range(i + 1, x_s[0]):
                dm[dm_count] = self._dtw_distance(x[i, ::self.subsample_step],
                                                    y[j, ::self.subsample_step])
                dm_count += 1

    # Convert to squareform
    dm = squareform(dm)

    # Compute full distance matrix of dtw distneces between x and y
    else:
        x_s = np.shape(x)
        y_s = np.shape(y)
        dm = np.zeros((x_s[0], y_s[0]))
        dm_size = x_s[0] * y_s[0]

        for i in range(0, x_s[0]):
            for j in range(0, y_s[0]):
                dm[i, j] = self._dtw_distance(x[i, ::self.subsample_step],
                                                y[j, ::self.subsample_step])
            dm_count += 1
    return dm
```

Príloha B: Výsledky z klasifikácie neurónových sietí a kNN+DTW

Experiment 1 - Neurónová sieť:

- Train_data1.csv [911 vzorov], Test_data1.csv [304 vzorov]
- 4 príkazy na motorické očné funkcie
- 30 iterácií
- 2 skryté vrstvy, veľkosť 160,80, aktivačná funkcia: softmax, tanh
- 4 výstupy, aktivačná funkcia: sigmoid

	Stop [počet]	Dopredu [počet]	Doľava [počet]	Doprava [počet]	Priemer [%]
Stop [počet]	44	0	14	1	74,58
Dopredu [počet]	1	16	50	0	23,88
Doľava [počet]	1	0	98	1	98,00
Doprava [počet]	0	0	78	0	0,00
Vážený priemer [%]					49,12

Experiment 2 - Neurónová sieť:

- Train_data1.csv [911 vzorov], Test_data1.csv [304 vzorov]
- 4 príkazy na motorické očné funkcie
- 30 iterácií
- 2 skryté vrstvy, veľkosť 160,80, aktivačná funkcia: softmax, softmax
- 4 výstupy, aktivačná funkcia: softmax

	Stop [počet]	Dopredu [počet]	Doľava [počet]	Doprava [počet]	Priemer [%]
Stop [počet]	0	0	59	0	0,00
Dopredu [počet]	0	0	67	0	0,00
Doľava [počet]	0	0	100	0	100,00
Doprava [počet]	0	0	78	0	0,00
Vážený priemer [%]					25,00

Experiment 3 - Neurónová sieť:

- Train_data1.csv [911 vzorov], Test_data1.csv [304 vzorov]
- 4 príkazy na motorické očné funkcie
- 30 iterácií
- 2 skryté vrstvy, veľkosť 200,100, aktivačná funkcia: tanh, tanh
- 4 výstupy, aktivačná funkcia: sigmoid

	Stop [počet]	Dopredu [počet]	Doľava [počet]	Doprava [počet]	Priemer [%]
Stop [počet]	56	2	1	0	94,92
Dopredu [počet]	1	64	1	1	95,52
Doľava [počet]	0	2	92	6	92,00
Doprava [počet]	0	3	5	70	89,74
Vážený priemer [%]					93,05

Experiment 4 - Neurónová sieť:

- Train_data2.csv [1207 vzorov], Test_data2.csv [404 vzorov]
- 6 príkazov na motorické očné funkcie
- 100 iterácií
- 2 skryté vrstvy, veľkosť 250,125, aktivačná funkcia: tanh, tanh
- 6 výstupy, aktivačná funkcia: sigmoid

	Stop [počet]	Dopredu [počet]	Doľava [počet]	Doprava [počet]	45° Doľava [počet]	45° Doprava [počet]	Priemer [%]
Stop [počet]	56	0	0	0	1	2	94,92
Dopredu [počet]	1	65	1	0	0	0	97,01
Doľava [počet]	0	0	94	4	2	0	94,00
Doprava [počet]	0	2	6	44	0	4	78,57
45° Doľava [počet]	0	1	0	4	39	2	84,78
45° Doprava [počet]	5	0	1	2	0	68	89,47
Vážený priemer [%]							89,79

Experiment 5 - Neurónová sieť:

- Train_data3.csv [120 vzorov], Test_data3.csv [60 vzorov]
- 6 príkazov na motorické očné funkcie
- 200 iterácií
- 2 skryté vrstvy, veľkosť 250,125, aktivačná funkcia: tanh, tanh
- 6 výstupy, aktivačná funkcia: sigmoid

	Stop [počet]	Dopredu [počet]	Dolaava [počet]	Doprava [počet]	45° Dolaava [počet]	45° Dolaava [počet]	Priemer [%]
Stop [počet]	7	0	0	3	0	0	70,00
Dopredu [počet]	0	8	0	2	0	0	80,00
Dolaava [počet]	0	0	6	4	0	0	60,00
Doprava [počet]	0	0	2	8	0	0	80,00
45° Dolaava [počet]	0	0	0	2	8	0	80,00
45° Doprava [počet]	0	0	0	1	1	8	80,00
Vážený priemer [%]							75,00

Experiment 6 - Neurónová sieť:

- Train_data_hz.csv [452 vzorov], Test_data_hz.csv [151 vzorov]
- 4 príkazov na senzorické očné funkcie
- 50 iterácií
- 2 skryté vrstvy, veľkosť 160,80, aktivačná funkcia: tanh, tanh
- 4 výstupy, aktivačná funkcia: sigmoid

	3 Hz [počet]	5 Hz [počet]	7 Hz [počet]	11 Hz [počet]	Priemer [%]
3 Hz [počet]	11	6	6	14	29,73
5 Hz [počet]	9	10	12	8	25,64
7 Hz [počet]	7	7	18	3	51,43
11 Hz [počet]	5	13	10	12	30,00
Vážený priemer [%]					34,20

Experiment 7 - Neurónová sieť:

- Train_data_hz1.csv [383 vzorov], Test_data_hz1.csv [127 vzorov]
- 5 príkazov na senzorické očné funkcie
- 50 iterácií
- 2 skryté vrstvy, veľkosť 160,80, aktivačná funkcia: tanh, tanh
- 5 výstupy, aktivačná funkcia: sigmoid

	3 Hz [počet]	4 Hz [počet]	5 Hz [počet]	7 Hz [počet]	11 Hz [počet]	Priemer [%]
3 Hz [počet]	7	3	3	3	10	26,92
4 Hz [počet]	4	6	6	9	4	20,69
5 Hz [počet]	9	1	3	6	5	12,50
7 Hz [počet]	6	3	3	9	1	40,91
11 Hz [počet]	3	8	2	7	6	23,08
Vážený priemer [%]						24,82

Experiment 8 - Neurónová sieť:

- Train_data_hz2.csv [915 vzorov], Test_data_hz2.csv [304 vzorov]
- 4 príkazov na senzorické očné funkcie
- 100 iterácií
- 2 skryté vrstvy, veľkosť 200,100, aktivačná funkcia: tanh, tanh
- 4 výstupy, aktivačná funkcia: sigmoid

	3 Hz [počet]	5 Hz [počet]	7 Hz [počet]	11 Hz [počet]	Priemer [%]
3 Hz [počet]	31	25	11	20	35,63
5 Hz [počet]	16	20	9	27	27,78
7 Hz [počet]	22	16	16	12	24,24
11 Hz [počet]	10	14	23	32	40,50
Vážený priemer [%]					32,04

Experiment 9 – kNN+DTW:

- Train_data1.csv [911/3 vzorov], Test_data1.csv [304/3 vzorov]
- 4 príkazy na motorické očné funkcie
- Náhodne vybraných 75 vzorov na každý príkaz a 25 testovacích signálov

	Stop [počet]	Dopredu [počet]	Doľava [počet]	Doprava [počet]	Priemer [%]
Stop [počet]	24	0	1	0	96,00
Dopredu [počet]	0	25	0	0	100,00
Doľava [počet]	1	0	22	2	88,00
Doprava [počet]	0	1	1	23	92,00
Vážený priemer [%]					94,00

Experiment 10 - kNN+DTW:

- Train_data2.csv [1207/3 vzorov], Test_data2.csv [404/3 vzorov]
- 6 príkazov na motorické očné funkcie
- Náhodne vybraných 75 vzorov na každý príkaz a 25 testovacích signálov

	Stop [počet]	Dopredu [počet]	Doľava [počet]	Doprava [počet]	45° Doľava [počet]	45° Doprava [počet]	Priemer [%]
Stop [počet]	23	0	0	0	1	1	92,00
Dopredu [počet]	0	25	0	0	0	0	100,00
Doľava [počet]	0	0	24	0	1	0	96,00
Doprava [počet]	1	0	0	22	0	2	88,00
45° Doľava [počet]	0	0	2	1	20	2	80,00
45° Doprava [počet]	0	0	1	1	2	21	84,00
Vážený priemer [%]							90,00

Experiment 11 - kNN+DTW:

- Train_data_hz1.csv [383 vzorov], Test_data_hz1.csv [127 vzorov]
- 5 príkazov na senzorické očné funkcie
- Náhodne vybraných 75 vzorov na každý príkaz a 25 testovacích signálov

	3 Hz [počet]	4 Hz [počet]	5 Hz [počet]	7 Hz [počet]	11 Hz [počet]	Priemer [%]
3 Hz [počet]	6	2	5	4	8	24,00
4 Hz [počet]	6	7	6	3	3	28,00
5 Hz [počet]	7	3	9	5	1	36,00
7 Hz [počet]	2	9	3	9	2	36,00
11 Hz [počet]	5	4	4	3	9	36,00
Vážený priemer [%]						32,00