



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

ŘÍZENÍ VIRTUÁLNÍHO ROBOTA POMOCÍ EEG

VIRTUAL ROBOT CONTROL USING EEG

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL DRLA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN TINKA

BRNO 2021

Zadání bakalářské práce



Student: **Drla Michal**
Program: Informační technologie
Název: **Řízení virtuálního robota pomocí EEG**
Virtual Robot Control Using EEG
Kategorie: Bioinformatika

Zadání:

1. Nastudujte problematiku komunikace mezi mozkiem a počítačem (BCI) využívající elektroencefalografu (EEG). Seznamte se se zařízením OpenBCI. Prostudujte metody strojového učení pro rozpoznávání EEG signálů. Seznamte se s nástroji pro tvorbu 3D obsahu Blender a herním enginem Unity či Godot.
2. Navrhněte vlastní implementaci rozpoznávání EEG signálů a navrhněte jednoduchou aplikaci, ve které bude možné pomocí EEG ovládat virtuálního robota.
3. Navrženou aplikaci pro řízení virtuálního robota implementujte s vámi navrženým rozpoznáváním EEG signálů.
4. Ověřte správnost fungování vaší aplikace na lidech, pokud to umožní epidemiologická situace. Navrhněte možné pokračování práce. Klíčové vlastnosti práce prezentujte formou krátkého videa a plakátu.

Literatura:

- Wolpaw, J. a Wolpaw, E. W., eds. Brain-computer interfaces: principles and practice. Oxford University Press, 2012. ISBN 978-0-19-538885-5.
- Siuly, S., Li, Y. a Zhang, Y. EEG Signal Analysis and Classification. Springer International Publishing, 2016. Health Information Science. ISBN 978-3-319-47652-0. Dostupné z: <http://link.springer.com/10.1007/978-3-319-47653-7>.
- Cohen, M. X. Analyzing neural time series data: theory and practice. The MIT Press, 2014. Issues in clinical and cognitive neuropsychology. ISBN 978-0-262-01987-3.
- Luck, S. J. An introduction to the event-related potential technique. Second edition. The MIT Press, 2014. ISBN 978-0-262-52585-5.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a částečně 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Tinka Jan, Ing.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 11. listopadu 2020

Abstrakt

Tato práce si kladla za cíl implementovat aplikaci, ve které je možné ovládat virtuálního robota pomocí EEG. V práci je vysvětlena teorie nutná k pochopení, jak fungují BCI systémy, které zpracovávají signály EEG. Jsou zde uvedeny nejen základy analýzy EEG, ale také ukazuje biologii lidského mozku a signály, které lze získat. Další důležitou částí je teorie neuronových sítí, které byly v implementaci využity. V implementaci jsou popsány skripty, které byly využity pro sbírání dat, návrh neuronové sítě a tvorba demonstrační aplikace. Výsledky testování jsou uspokojivé. Neuronová síť vyhodnocovala správně signály EEG a uživatel byl schopen ovládat virtuálního robota.

Abstract

This bachelor thesis aimed to create an application where is user able to control the virtual robot with an EEG signal. The thesis contains a brief introduction that explains how BCI systems which are using EEG work. This introduction not only explains the basics of EEG analysis but also explains brain biology and shows different signals which are extractable from the brain. This thesis also explains the theory of neural networks which are used to implement the analysis. In implementation are shown scripts that were used to collect data and there is also shown the design of the neural network. Results of testing are good, the neural network was making correct decisions and the user was able to control the virtual robot.

Klíčová slova

BCI, EEG, neuronové sítě, perceptron, analýza EEG, klasifikace dat, CNN, RNN

Keywords

BCI, EEG, neural networks, perceptron, analysis of EEG, clasification of data, CNN, RNN

Citace

DRLA, Michal. *Řízení virtuálního robota pomocí EEG*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Tinka

Řízení virtuálního robota pomocí EEG

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Tinky. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Michal Drla
6. května 2021

Poděkování

Rád bych poděkoval vedoucímu mé práce Ing. Janu Tinkovi za odborné vedení a poskytnuté materiály.

Obsah

1	Úvod	2
2	Mozek a signály produkované mozkem	3
2.1	Historie zkoumání elektrických signálu mozku	3
2.2	Neuron	3
2.3	Anatomie mozku	4
2.4	Získávání dat pro BCI	6
2.5	Formy signálů EEG	9
3	Zpracování signálu	14
3.1	Předzpracování signálu	14
3.2	Filtrování signálu	15
3.3	Extrakce rysů	15
3.4	Analýza dat	17
3.5	Neuronové sítě	19
4	Implementace aplikace BCI	25
4.1	Návrh systému BCI	25
4.2	Použité knihovny a vývojářská prostředí	26
4.3	Návrh a implementace pomocných skriptů a sběru dat	26
4.4	Neuronová síť	36
4.5	Testovací aplikace	40
4.6	Testování systému	44
5	Závěr	46
	Literatura	48
A	Plakát	53
B	Obsah příloženého paměťového média	54

Kapitola 1

Úvod

Technologie Brain computer interface (BCI) je poměrně starou technologií. Již v roce 1924 Hans Berger popsal elektrické signály, které jsou produkovány v lidském mozku. Od té doby s technologií pracovalo velké množství vědeckých týmů, které se v počátcích snažily zpracovat artefakty z EEG. Dnes lze s pomocí systémů BCI ovládat složité aplikace, které mohou pomoci lidem, kteří by tyto aplikace nebyli schopni ovládat normálním způsobem.[22]

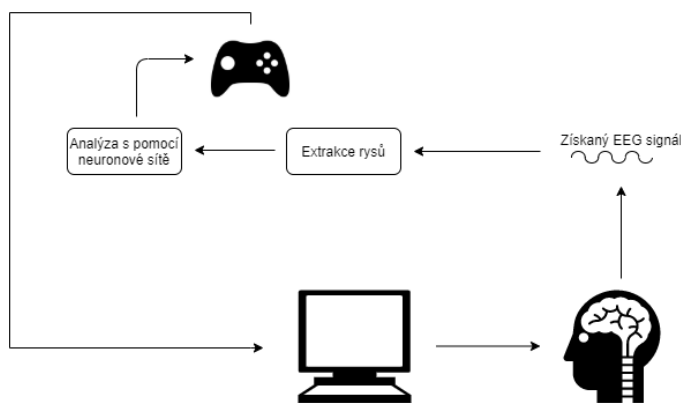
Systémy BCI využívají toho, že centrální nervová soustava reaguje na každou situaci produkováním elektrického signálu, který je v nějaké formě zachycen a analyzován. Na základě této analýzy lze určit, jakou činnost chtěl uživatel provést. Systém BCI lze rozdělit do částí, které jsou ukázány na obrázku 1.1.[43]

Tato práce si dává za cíl vytvořit jednoduchou aplikaci, s pomocí které bude možné ovládat virtuálního robota.

V této práci jsou nejdříve popsány teoretické základy, ze kterých systémy BCI vychází.

V první sekci jsou vysvětleny základní principy fungování mozku a shrnuty poznatky o základních signálech, které mozek produkuje. Následně je popsán proces extrakce rysů a jejich analýzu.

Následovat bude popis implementace aplikace a zhodnocení informací získaných z vývoje.



Kapitola 2

Mozek a signály produkované mozkiem

V této kapitole jsou uvedeny základní informace o mozku a signálech, které produkuje. V úvodní sekci je představena historie získávání signálu mozku, která ukazuje jak daleko se za poměrně krátkou dobu tato oblast rozvinula. Následuje popis anatomie mozku, kde je důležité nejdříve vzpomenout neuron, který je základní stavební buňkou mozku. Následně se práce věnuje způsobům, získávání signálů z mozku. V závěrečné sekci jsou shrnuty poznatky o EEG signálech, které lze získat s pomocí elektrod.

2.1 Historie zkoumání elektrických signálu mozku

Centrální nervová soustava (CNS) využívá elektrických signálů k řízení jednotlivých částí těla. Tento objev uskutečnila v roce 1870 dvojice Eduard Hitzig a Gustav Fritsch, kteří stimulovali oblast mozku psa, což způsobilo pohyb části jeho těla. Tento experiment prokázal, že je možné využít elektrické stimuly k ovládní různých částí těla.[43]

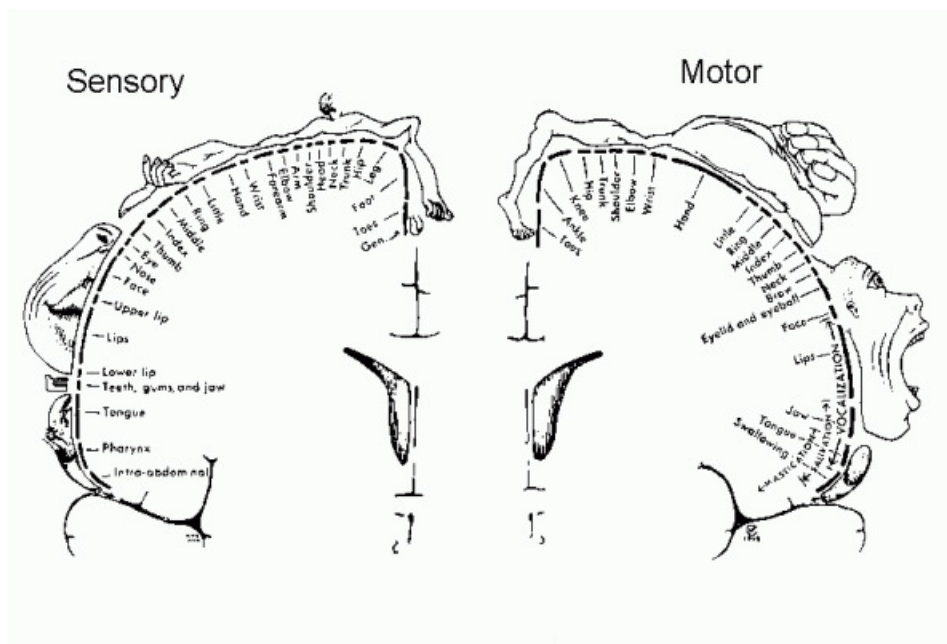
Dalším důležitým průlomem bylo přiřazení jednotlivých částí těla ke korespondujícím oblastem v mozku. Tento model vytvořil Wilder Penfield, který vytvořil schéma nazvané motor homoculus (obrázek 2.1), které je dodnes hojně využíváno i přes určitou nepřesnost. Ve stejné době byly učiněny i první měření aktivity neuronu, které vedly k rozvoji oblasti technologií BCI.[43]

2.2 Neuron

Neuron je základní stavební buňkou mozku. Hlavním úkolem neuronů je zachytávat informace, zpracovat je a následně je převést dále. V mozku lze nalézt různé druhy neuronů, které se liší stavbou (například pyramidové neurony, košíkové neurony [37]), avšak všechny sdílí základní strukturu, která je v následujících odstavcích popsána.

Každý neuron se skládá z buněčného jádra, na které navazují dendrity, což jsou výběžky, které mají za úkol, přijímat informace a předávat je dále do axonu. Axon je ocasovitý výběžek, který na rozdíl od dendritů předává informaci z neuronu dále.[13]

Příchozí elektrické signály z dendritů se sečtou a pokud překročí určitou hranici, tak axon přeneše nervový vzruch do dalšího neuronu. Toto se děje s pomocí generování akčních potenciálů, které s pomocí chemických reakcí se přemístí do dalšího neuronu.[13]



Obrázek 2.1: Penfieldova mapa mozku, autor Julia Nosek [31]

2.3 Anatomie mozku

Koncový mozek se skládá ze dvou symetrických částí, takzvaných hemisfér. Povrch mozku pokrývá mozková kůra. Mozková kůra je výrazně zvrásněná. Toto zvrásnění zvyšuje množství nervové tkáně, která se vejde do malé oblasti.[28]

V šedé kůře mozkové probíhá zpracování informací předaných mozku a jejich vyhodnocování. Mozková kůra se skládá z nervových buněk, které mají šedou barvu. Podobně zbarvené jsou i další části mozku, jako například mozeček nebo thalamus, který je součástí mezimozku.[28]

Pod šedou kůrou mozkovou se nachází bílá hmota mozková. Tato hmota se skládá z axonů, které jsou omotané bílým myelinem. Myelin je lipoprotein, který izoluje axon od okolního prostředí. Hlavní účel bíle hmoty mozkové je přenos informací mezi částmi mozku, případně z jiných částí centrální nervové soustavy do mozku a naopak.[43] Mozek se skládá i z dalších oblastí, které navazují na výše popsané části. V této práci je nejdůležitější šedá kůra mozková, protože tato oblast ovládá většinu kognitivních funkcí a je poměrně dobře přístupná.[43] Další část je zaměřená na základní popis jednotlivých oblastí mozkové kůry a shrnutí toho, jak je lze využít v systémech BCI.

Mozková kůra

Mozkovou kůru lze rozdělit na několik základních částí (laloků):

- čelní lalok
- temenní lalok
- týlní lalok

- spánkový lalok

Čelní lalok

Čelní lalok se nachází v přední části mozku. Zde se nachází řečové centrum (Brocovo centrum řeči). V jeho horní části je umístěné pohybové centrum, které má na starost koordinaci a veškerý pohyb. Rozdělení oblastí v mozku je přímo úměrné množství pohybů, které jednotlivé oblasti koordinují.[28]

Temenní lalok

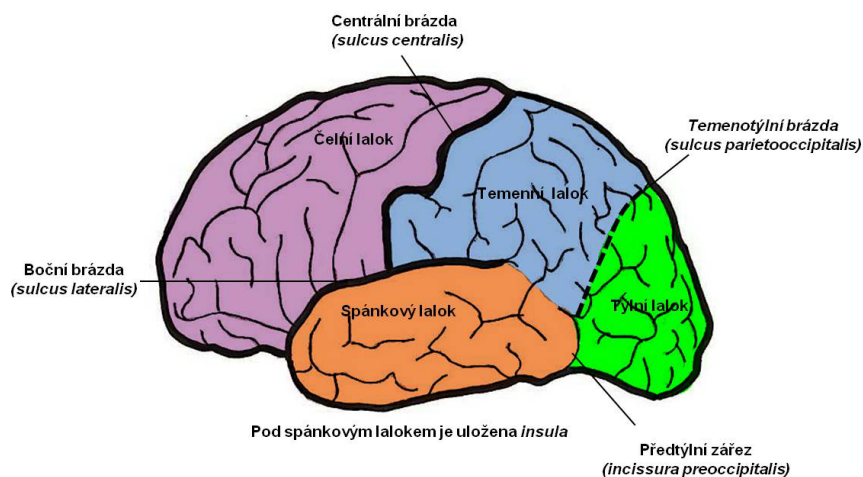
Temenní lalok, jak je ukázáno na obrázku 2.2 se nachází za čelním lalokem. V této části mozku se především nachází sensorická centra. Podobně jako v motorické oblasti platí, čím více sensorů se nachází na určité části těla, tím větší oblast je v mozku dedikována. Například konečky prstů mají dedikovanou poměrně velkou část.[28]

Týlní lalok

Týlní lalok se nachází v zadní části mozku viz obrázek 2.2. V této oblasti mozku se nalézá takzvaný hipokampus, který je zodpovědný za krátkodobé uchovávání vzpomínek a jejich přeměnu na dlouhodobé vzpomínky. Jedinci s poškozenou oblastí tohoto mozku mají problém si myšlenky krátkodobě zapamatovat a nejsou schopni vytvářet nové dlouhodobé vzpomínky.[28]

Spánkový lalok

Spánkový lalok se nachází ve spodní části mezi čelním a temenním lalokem. Dominantní je zde centrum zraku. V tomto centru jsou zpracovávány informace, které byly získány levým a pravým okem. Další oblastí je zraková asociativní oblast, která se snaží zpracované informace interpretovat a umožňuje člověku získat celkový přehled o situaci, kterou vidí.[28]



Obrázek 2.2: Základní schéma mozkové kůry, autor Daniel Kosec [21]

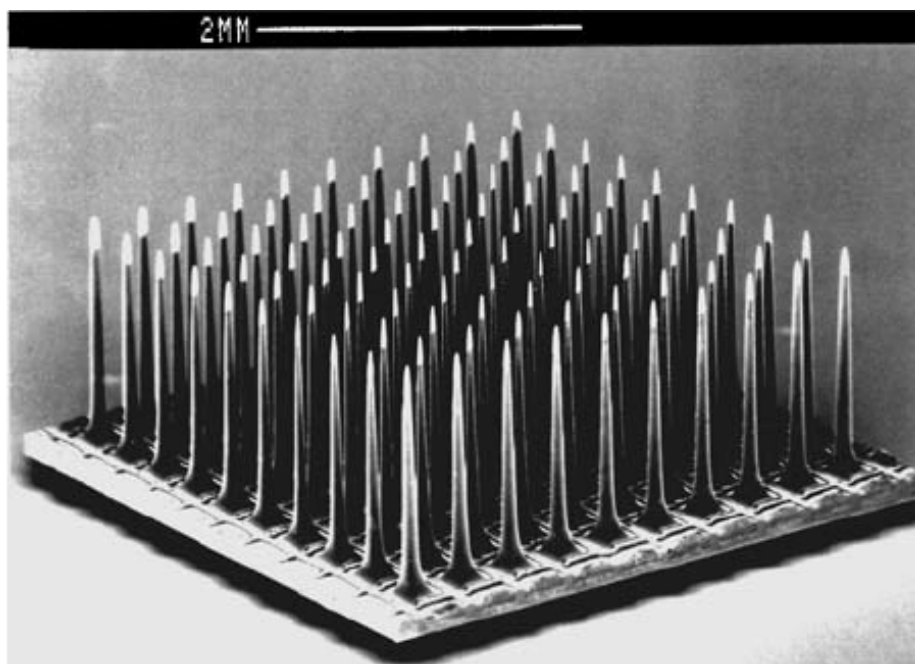
2.4 Získávání dat pro BCI

Technologie BCI zpracovává elektrický signál, který je produkovaný neurony v mozku. Tento signál se zachytává pomocí elektrod. Získaný signál lze rozdělit do několika kategorií: potenciál lokálního pole (LFP), elektrokortikografie (ECoG) a elektroencefalografie (EEG). Tyto signály se liší v kvalitě, kdy LFP má nejlepší rozlišení a naopak EEG nejhorší. Liší se také ve formě elektrod, které jsou pro získávání daného signálu použity a způsobem zavedení těchto elektrod.[43]

Potenciál lokálního pole

Potenciál lokálního pole je signál získaný přímo z šedé kůry mozkové. Elektrody musí být zavedeny do mozkové tkáně. Elektrody jsou implantovány přímo na povrch šedé kůry mozkové. K výrobě mohou být použity různé materiály. Často se používají tenké skleněné trubičky, případně se vyrábí z různých kovů.[7] Průměr jednotlivých elektrod je minimálně $10\ \mu\text{m}$ a maximální velikost se většinou pohybuje okolo $100\ \mu\text{m}$. [43]

Zavedením těchto elektrod přímo do mozkové kůry lze získat velice kvalitní signál, který se skládá z akčního potenciálu neuronů a dalších membránových potenciálů. Signál získaný s pomocí těchto elektrod, je z velice malé oblasti mozku, což může vést ke zkreslování dat. Proto se často využívá větší množství elektrodových matic (jak je ukázáno na obrázku 2.3), které pokrývají větší plochu a tedy má signál lepší prostorové pokrytí. Zavedení elektrod do mozkové kůry je náročné a může přinášet zvýšená zdravotní rizika.[7]



Obrázek 2.3: Ukázka matice mikroelektrod, autor Kim Seung-Jae [17]

Elektrokortikografie

Elektrokortikografie je metoda, která na rozdíl od LFP nepoužívá signál získaný přímo z jednotlivých neuronů. Pro tuto metodu se používají často ocelové elektrody, které jsou

zavedeny na povrch šedé kůry mozkové. Proces zavádění je oproti předchozí metodě jednodušší a představuje menší riziko.[7]

Signál získaný z těchto elektrod již nepracuje s membránovými potenciály jednotlivých neuronů, ale měří signál, který je generován větším shlukem neuronů. Díky tomu, že jsou elektrody umístěny pod lebku, tak nedochází ke zkreslení signálů. Dnes se ECoG využívá při studiu jevů v šedé kůře mozkové.[7]

Elektroencefalografie

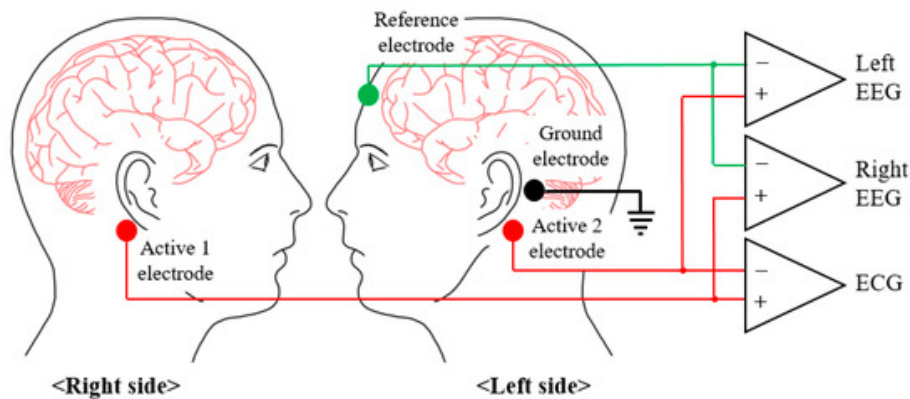
Elektroencefalografie na rozdíl od předchozích metod není invazivní. Tato metoda je nejstarší mezi uvedenými a použití je snadné.[7]

Pro metodu EEG se využívají dva základní typy elektrod. Pro získávání signálu lze využít takzvané vlhké elektrody, které používají nějaký vodivý materiál nejčastěji gel, který je aplikován na skalp. Na tuto vrstvu je umístěna elektroda.

Druhou variantou je použití suchých elektrod. Ty nevyužívají pomocnou vrstvu a lze je umístit přímo na skalp. Tyto elektrody jsou nejčastěji vyráběny ze vzácných kovů - zlato, platina a případně z oceli. Vývoj těchto elektrod je velice náročný, protože signál, který se s jejich pomocí zachytává, je náchylný na zanesení okolním elektrickým šumem. Tento šum může veškerá měření znehodnotit. Výhody tohoto typu elektrod jsou značné a jsou tedy centrem vývoje.[43]

Dále lze elektrody rozdělit na elektrody aktivní nebo pasivní. Aktivní elektrody zesilují EEG signál přímo v elektrodě. Naopak pasivní signál pouze předávají dále do systému, kde je pak zesílen.[43]

Pro správné získávání EEG signálu je nutné využít minimálně tři elektrody, kde jedna se připojí k zemi (země je zde dohodnuté místo, které má pro systém nulový potenciál, například spánkové kosti na obrázku 2.4) a zbylé dvě elektrody měří hodnoty na skalpu. Ukázka tohoto zapojení je na obrázku 2.4. Výsledný EEG signál je rozdílem referenční elektrody s elektrodou aktivní.[43]



Obrázek 2.4: Schéma minimálního počtu elektrod, autor Joong Woo Ahn [2]

Pro volbu umístění referenční elektrody existuje více možností. Lze využít metodu spojených uší, kdy se umístí elektrody na ušní lalůčky a referenční hodnota se měří pro každou hemisféru zvlášť. Další alternativou jsou spánkové kosti na skalpu, toto umístění je ukázáno na obrázku 2.4. Obě tyto možnosti jsou často využívány, avšak neexistuje žádný vědecký podklad jejich výhodnosti.[43] Naopak dnes často používaná je metoda běžného průměru, která využívá zprůměrování všech hodnot napětí na elektrodách. Výsledek je považován za

referenční. Výhodnost této metody stoupá s použitím většího množství elektrod a tedy i její přesností.[43]

Dalším důležitým parametrem, který je nutné u EEG sledovat je vzorkovací frekvence. Tato hodnota říká, kolik vzorků za sekundu AD převodník, který navazuje na elektrody, předává dále do systému. Tato informace je kriticky důležitá nejen, aby se dále v aplikaci vědělo s jakým množstvím dat se bude pracovat, ale také protože Nyquist-Shannonův teorém říká, že hodnoty frekvence nad polovinou vzorkovací frekvence jsou nepoužitelné a tedy je není možné použít. V tomto pásmu totiž dochází k takzvanému aliasingu. To je efekt při kterém signály přestávají být odlišitelné a může to vést i na vytváření artefaktů, které mohou znehodnotit celé měření. Za tímto účelem se tedy používají pásmové propusti, které propouští pouze signál, který bude splňovat výše zmíněný teorém.[42]

Důležitým parametrem, který se sleduje u EEG systémů je tzv. prostorové rozlišení. Prostorové rozlišení ovlivňuje schopnost určit, v které oblasti mozku proběhl významnější elektrický signál. Prostorové rozlišení je závislé na počtu elektrod umístěných na skalpu a tedy hodnoty, které se získávají z těchto elektrod jsou diskrétní a vztahuje se na ně Nyquistův teorém. Při nízkém počtu elektrod dochází k prostorovému aliasingu, který znemožňuje určit s dostatečnou přesností oblast vzruchu, a tedy celé měření znehodnotit. Prostorové rozlišení lze zvýšit využitím většího množství elektrod, které se rozmístí na skalp.[34]

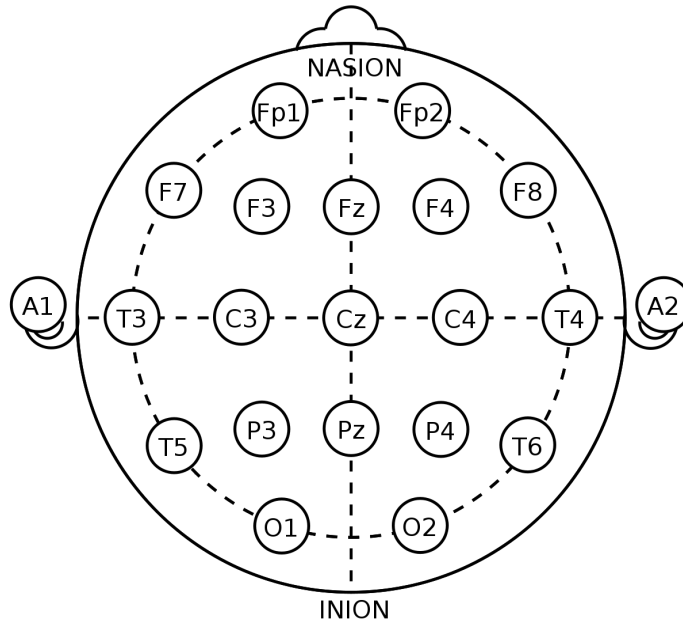
Ne vždy je, ale možné zvýšit počet elektrod. Z tohoto důvodu byly vypracovány metody, které se snaží zvýšit prostorové rozlišení. Mezi nejvýznamnější metody patří povrchový Laplacian. Tato metoda se snaží zlepšit prostorové rozlišení zvýšením citlivosti jednotlivých elektrod. Využitím této metody se sníží nutný počet elektrod a zároveň se zvýší kvalita výsledného prostorového signálu.[9] Další důležitou vlastností povrchového Laplacianu je nezávislost na referenčních elektrodách a schopnost vyfiltrovat artefakty způsobené signálem z jedné vadné elektrody. Dále je schopen vyfiltrovat artefakty způsobené okolním prostředím mimo elektrody.[32]

Umístění elektrod

Rozmístění elektrod na skalpu je často ovlivněno jejich účelem. Reflektují snahu získat co nejlepší měření signálu v určité oblasti mozku. I přesto byly vyvinuty standarty, které se snaží pokrýt co nejlépe důležité oblasti skalpu, tak aby signál získaný s pomocí těchto elektrod byl co nejlepší.[43]

Mezi nejznámější systémy používané pro rozmístění elektrod patří systém 10-20 a systémy odvozené, tj. systémy 10-10 a 10-5. Pro použití 10-20 systému se nejdříve skalp rozdělí do kvadrátu. Osy vedou od nosu po zátylek a mezi ušima. Následně se rozmístí elektrody, tak aby mezera byla 20 % z celkové délky daného kvadrátu.[43] Ukázkou použití tohoto systému pro 21 elektrod je uvedena na obrázku 2.5. Obrázek ukazuje, že tento systém definuje standart označování jednotlivých elektrod. Elektrody, které jsou označeny lichým číslem se umísťují na levou část skalpu, se sudým číslem na pravou. Elektrody vedoucí centrální částí skalpu jsou označeny malým písmenem z.[43]

Systém 10-20 je dnes velice oblíbený, protože tento systém lze přizpůsobit na velké množství elektrod a s pomocí tohoto systému lze dosáhnout kvalitních všeobecných měření, které jsou užitečné především v raných fázích výzkumu.[43]



Obrázek 2.5: 10/20 systém umístění elektrod, Wikipedia [20]:

2.5 Formy signálů EEG

Systém BCI působí jako prostředník mezi aplikací a uživatelem. Systém převádí signály EEG na příkazy, které dokáže pochopit aplikace. Aby bylo možné produkovat signály EEG, které je schopen systém BCI zpracovat, je nutné, aby uživatel stimuloval konkrétní části mozku stejnou činností opakovaně.[43] V následující kapitole jsou nejdříve představeny jednotlivá pásma signálu EEG, která se využívají v aplikacích BCI. Na závěr jsou popsány jednotlivé druhy potenciálu, které se využívají v systémech BCI.

Frekvenční pásma

Signál v mozku lze rozdělit do několika pásem podle frekvencí, na kterých se nachází. Některé metody analýzy EEG signálu sledují výkyvy v jednotlivých pásmech a podle toho se následně rozhoduje, jaká činnost byla prováděna.[43] Rozdělení jednotlivých pásem se často liší mezi autory vědeckých studií. Jednotlivé hranice se mohou drobně lišit. V této práci se uvažují pásma definovaná dle následující tabulky 2.1, která byla zveřejněna v knize *Introduction to EEG and Speech-Based Emotion Recognition*[29].

Frekvenční pásmo	Frekvence	Stav
Delta	0.5 - 4 Hz	Spánek
Beta	4 - 8 Hz	Maximální uvolnění, vnitřní soustředění
Alpha	8 - 12 Hz	Uvolnění, lehké soustředění
Beta	12 - 35 Hz	Nervozita, aktivita, zvýšená pozornost
Gamma	>35 Hz	Soustředění

Tabulka 2.1: Tabulka frekvenčních pásem a korespondujících stavů a aktivit

Způsoby generování EEG

Signál EEG lze pro systém BCI generovat různými způsoby. Mezi nejběžnější metody patří využití evokovaných potenciálů nebo představivosti. V následující části jsou představeny tyto vybrané metody:

- sluchové evokované potenciály
- somatosensorické evokované potenciály
- zrakové evokované potenciály
- pohybové evokované potenciály
- pohybová představivost

Při popisu výsledných signálů se v odborné literatuře často uvádí číslo, před kterým je písmeno P nebo N (například P100, N75). Toto číslo říká, jak dlouho po provedení konkrétní činnosti, případně zaznamenání stimulu, dojde k největší amplitudě mozkové aktivity v některé oblasti mozku. Označení P zde znamená pozitivní vrchol a N negativní vrchol. Číselná hodnota říká jaké zpoždění v milisekundách daný signál má.[24]

Sluchové evokované potenciály

Sluchové evokované potenciály (zkratka AEP z anglického auditory evoked potential) se většinou využívají při diagnostice ztráty sluchu. Využívají se k tomu krátké zvukové signály a následně se sleduje latence EEG signálu, které tento zvuk vyvolal. Signál má většinou větší množství menších amplitud, které může být poměrně složité analyzovat.[24]

Při využití této metody je nutné jedno ucho zastínit takzvaným „ružovým šumem“ (šumem, který má zastoupení určité frekvence), který omezuje průchodnost ostatních zvuků. Druhé ucho naslouchá velkému množství krátkých zvuků (nejčastěji kliknutí).[24]

Somatosensorické evokované potenciály

Somatosensorické evokované potenciály (zkratka SEP) vznikají stimulací různých částí těla s pomocí elektrického proudu. V lékařství se tato metoda využívá především pro zhodnocení průchodnosti nervových zakončení, případně na monitorování nervů při operaci páteře.[24]

SEP produkují poměrně výraznou vlnu N75 a následně vlnu P100. Pokud některá z těchto vln chybí, tak to může znamenat poškození stimulovaných nervových zakončení.[24]

Tato metoda je poměrně nevhodná pro systémy BCI, které mají za cíl zjednodušit ovládání aplikací, protože její použití vyžaduje stimulaci proudem, která může být pro uživatele nepříjemná.

Zrakové evokované potenciály

Zrakové evokované potenciály (zkratka VEP) se využívají v různých systémech BCI. Lze je využívat na ovládání kompenzačních pomůcek, komunikaci i zábavu. VEP využívá různých forem vizuálních stimulů, které jsou prezentovány uživateli.[16]

VEP se rozděluje do kategorií, podle toho zda jsou závislé nebo nezávislé na zbytku systému, dle typu stimulu, případně podle modulace. Následující část, která popisuje jednotlivé typy VEP podle posledního parametru (modulace).[16]

Časová modulace VEP (zkratka t-VEP) je specifická tím, že jednotlivé oblasti, které blikají jsou na sobě časově nezávislé a nepřekrývají se. Dochází tedy k tomu, že krátká nepravidelná zablikání generují amplitudy s krátkou odezvou. Při využití této metody, se musí věnovat zvýšená pozornost určení správné oblasti, na kterou se uživatel soustředil. Signál generovaný s pomocí t-VEP má často velice nízkou frekvenci (méně než 5 Hz), tato frekvence v mnohých systémech BCI bývá filtrována, proto se tato metoda často nepoužívá.[16]

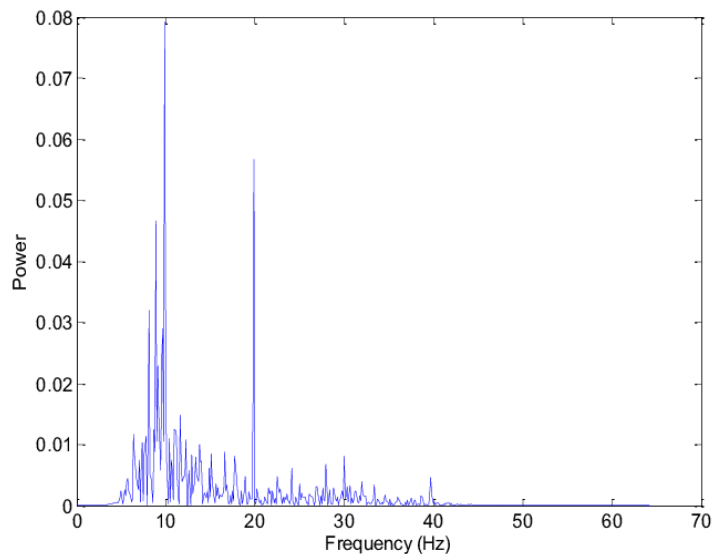
Další metodou je **frekvenční modulace VEP** (f-VEP). Při této metodě jednotlivé oblasti blikají na specifické frekvenci. Pokud se toto blikání udrží konstantní, tak je produkován EEG signál s podobnou frekvencí. Pro analýzu tohoto signálu je následně vhodné jej převést do spektra (například pomocí rychlé Fourierovy transformace, viz 3.3). Následně tento signál lze velice jednoduše analyzovat a přináší velice konstantní výsledky. Frekvence tohoto signálu se pohybuje nad signálem produkováným pomocí t-VEP (>5 Hz). Tento signál je tedy často v pásmu, které zůstane po aplikování pásmové propusti a lze jej dobře analyzovat. Často tato forma signálu také bývá nazývána ustálený vizuálně evokovaný potenciál (SSVEP). Signál SSVEP je generován pravidelným blikáním na konkrétní frekvenci. Signál v mozku tuto frekvenci napodobuje. Jak je vidět z obrázku 2.6 frekvenční doména tohoto signálu obsahuje i další amplitudy ve frekvencích, které jsou násobkem původní frekvence. Tyto amplitudy jsou postupně slabší a ve vysokých frekvencích zanikají úplně. Jednoduchost a stálost této metody z ní dělá jednu z nejoblíbenějších metod pro získávání příkazů od uživatelů pomocí BCI.[16]

Poslední metoda využívá pseudonáhodných vizuálních stimulů, které jsou prezentovány uživateli. Tato metoda se nazývá **pseudonáhodně kódovaná VEP** (c-VEP). C-VEP podobně jako t-VEP vyžaduje přesné časování, aby nedocházelo k překrývání a její vyhodnocování je velice složité. Systémy BCI využívající c-VEP se dočkaly pouze omezeného vývoje, mezi nejznámější patří systém Ericha Suttera [38]. Tyto systémy jsou velice složité na tvorbu a je velice těžké je udržet stabilní, proto jejich využití dnes moc nevidíme.[16]

Pohybové evokované potenciály a pohybová představitost

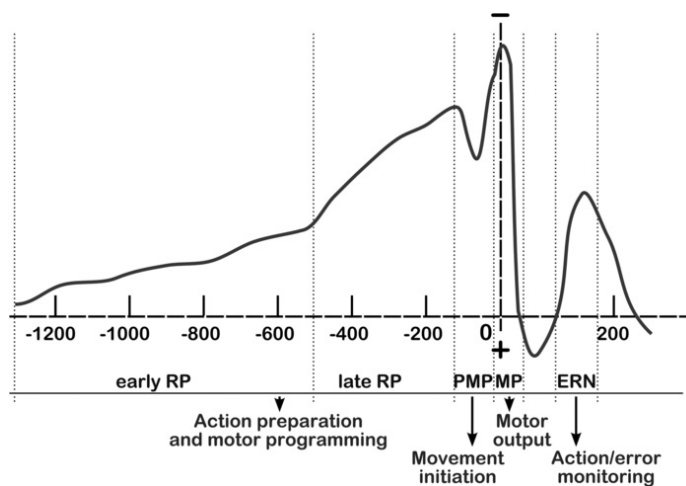
Jedním z cílů systémů BCI je umožnit pacientům s poškozením pohybového aparátu ovládat kompenzační pomůcky, které jim umožňují pohyb. To ovšem neznamená, že by oblast pohybově evokovaných potenciálů (MRP z anglického movement related potential) byla opomíjena. MRP lze využít při částečném ochrnutí, kdy pohyby částí těla jsou převáděny na jiné pohyby, případně je lze využít při léčbě nemocí, které omezují pohyb (jako je například Parkinsonova choroba).[12]

V systémech BCI lze využít, kteroukoliv část těla. Nejčastěji se však jedná o ruce. Uživatelé pohybují například prsty, případně celým zápěstím. Následně se určí oblast mozku,



Obrázek 2.6: Ukázka frekvenční domény signálu, produkovaného s pomocí SSVEP, autor Omar Trigui [39]

ze které pochází daný signál. MRP signál je charakteristický výraznou amplitudou, která nastane se zpožděním přibližně 500 ms od provedení dané činnosti.[12] Signál má frekvenci 13-15 Hz a jak je na obrázku 2.7 možné vidět, signál dosáhne maxima těsně po zahájení pohybu. Následuje velký pokles a znovu zvýšení amplitudy, která nedosáhne původního maxima.



Obrázek 2.7: Ukázka signálu produkovaného pomocí MRP, autor Roland A. Carlstedt [8]

Alternativou k MRP je **pohybová představivost** (MI z anglického motor imagery). Pohybová představivost nabízí možnost uživatelům s těžkým poškozením pohybového aparátu využívat systémy BCI postavené pro MRP. Základem této metody je představa uživatele, jak provádí konkrétní pohybovou činnost. Tato představa následně generuje signál velice podobný signálu, který je pozorován u MRP. Systémy BCI není třeba nějak zásadně

upravovat a jedinou nevýhodou oproti pohybově evokovaným potenciálům je složité pochopení tohoto konceptu uživatelem. Taktéž samotná schopnost představit si daný pohyb může být u uživatele omezená.[12]

Shrnutí

Pro ovládání aplikace s použitím systémů BCI, lze použít velké množství metod a zvolení vhodné metody je velice důležité. Zvolení nevhodné metody může mít za následek, že uživatel nebude schopen ovládat aplikaci, protože jeho postižení bude bránit využití dané metody. Při volbě je důležité zohlednit k jakému účelu bude systémem BCI použit, citlivost elektrod, které se využijí, případně další omezení, která mohou vyplývat z postižení uživatele.

Kapitola 3

Zpracování signálu

Signál získaný z elektrod může sloužit k ovládní různých aplikací. Aby jej bylo možné využít v praxi, tak je nejdříve nutné výstup z elektrod zpracovat.[43]

Zpracování je v této práci rozděleno do čtyř základních fází: Předzpracování, filtrování, extrakce rysů a analýza signálu.

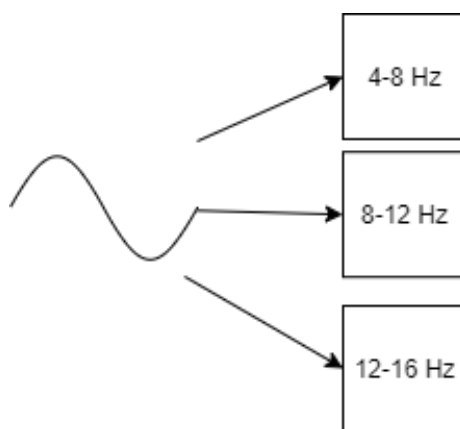
3.1 Předzpracování signálu

Získaný signál z elektrod obsahuje velké množství šumu. Ten je způsobem pohybem očí, svalovou aktivitou případně i dýcháním. Tyto artefakty mohou znatelně ztížit analýzu signálu, a proto se je nutné jej odstranit.[1] Základem předzpracování je kovarianční adaptace maticové evoluční strategie (CMA-ES). Algoritmus CMA-ES se využívá při složitých optimalizačních problémech. CMA-ES spočítá pravděpodobnostní distribuci vzorků, které mají nejlepší výsledky a tímto se snaží najít nejvhodnější oblasti s dobrými výsledky.[4]

3.2 Filtrování signálu

Filtrování signálu s pomocí digitálních filtrů hraje pro analýzu EEG signálů velice důležitou roli. Signál získaný z elektrod je od určité frekvence poškozen aliasingem jak bylo zmíněno v předchozí části (viz 2.4), a tedy je nepoužitelný. Proto je vhodné signál vyfiltrovat za použití pásmové propusti, ta zajistí, že pouze frekvence, která splňuje Nyquistův teorém bude zahrnuta v analýze. Další důležitý filtr je pásmová zádrž. Tento filtr hraje klíčovou roli ve filtraci šumu, který je generovaný elektrickou sítí.[43] Frekvence, kterou je nutné vyfiltrovat v evropských zemích i v České republice je 50 Hz.[19]

Další využití filtrů může být při analýze za použití takzvaných filtračních bank. Filtrační banky jsou pásmové propusti, které rozdělí signál do daných oblastí. Ukázka filtračních bank je na obrázku 3.1. Rozdělený signál lze dále jednodušeji analyzovat a zaměřit se pouze na podstatné pásmo.[26]



Obrázek 3.1: Schéma filtrační banky

3.3 Extrakce rysů

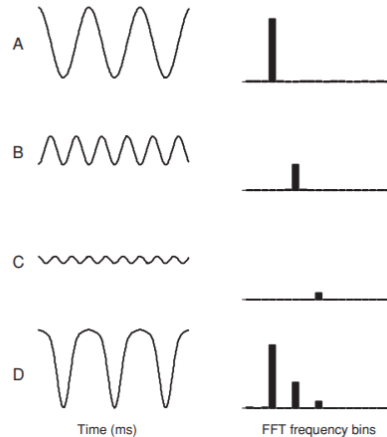
Signál získaný filtrováním, je dále předán algoritmům, které mají za úkol provést extrakci rysů z EEG signálu. Rys je hodnota, která ukazuje charakteristiku části signálu. Extrakce rysů se využívá pro získání dat, která jsou důležitá pro následnou analýzu. Pokud by extrakce rysů nebyla využita, tak by signál obsahoval velké množství artefaktů a nebylo by jej možné použít. [3]

Metod, které byly adaptovány pro extrakci rysů z EEG signálu je velké množství. V tomto textu jsou představeny tři metody: Rychlá Fourierova transformace (FFT), vlnková transformace (WT) a autoregresivní model (AR). Každá z těchto metod má své výhody a nevýhody, a proto si jsou nejdříve představeny a následně je zhodnoceno, která metoda je nevhodnější.[43]

Rychlá Fourierova transformace

Rychlá Fourierova transformace je algoritmus, který implementuje diskrétní Fourierovu transformaci. FFT vytváří reprezentaci frekvenčního spektra signálu. Rozlišení dané frekvence lze určit jako $f_s / \text{Pocet} - \text{hodnot} - \text{spektra}$, kde f_s je vzorkovací frekvence daného signálu. Samotné FFT vytváří z N vstupních vzorků N výstupních vzorků, které jsou roz-

děleny do frekvenčního spektra o velikosti $f_s/2$. Ukázkou takového rozložení lze vidět na obrázku 3.2. Každá hodnota na grafu říká jaká byla konkrétní amplituda na dané frekvenci. Tato reprezentace je výhodná především kvůli rychlosti výpočtu a následné jednoduchosti při analýze. FFT přináší i nevýhody především ve formě poměrně nízkého rozlišení, kdy veškerá data mají diskrétní charakter a může se tedy stát, že některé důležité prvky se v této transformaci mohou ztratit. Proto je FFT primárně vhodná pro extrakci ze signálu, který má hodně výrazné rysy a nemůže se stát, že by se důležité hodnoty ztratily.[43]



Obrázek 3.2: Ukázkou převodu signálu z časové do frekvenční domény pomocí FFT, autor Jonathan R. Wolpaw[43]:

Vlnková transformace

Vlnková transformace podobně jako FFT převádí hodnoty signálu do frekvenčního spektra. Na rozdíl od FFT tato metoda nepřevádí hodnoty na diskrétní, ale zachovává spojitou formu signálu. Rozlišení těchto hodnot je tedy značně lepší. Data která se analyzují v následném procesu dávají podrobnější informace o rysech, které by v FFT zanikly. Vlnková transformace zachovává časovou informaci signálu, která může být při analýze užitečná.[23]

Nevýhodou tohoto přístupu je jeho zvýšená náročnost na výpočet oproti FFT. I přesto tato metoda patří k nejoblíbenějším metodám pro extrakci rysů z EEG.[23]

Speciální formou WT je diskrétní vlnková transformace (DWT), která diskrétně vzorkuje jednotlivé vlnky na rozdíl od základní metody, která vzorkuje spojitě. Tato metoda si zachovává dobré rozlišení a schopnost zachytit časovou informaci. DWT zrychluje proces extrakce rysů avšak není schopná dosáhnout rychlosti FFT. DWT je vhodné použít při snaze přiblížit se rychlosti FFT a zachování časové domény signálu.[23]

Autoregresní model

Autoregresní model (AR) podobně jako předchozí metody konvertuje původní spektrum signálu na frekvenční spektrum. Tato metoda předpokládá, že vstupní hodnoty jsou bílý šum, který byl zpracován filtrem s nekonečnou impulzní odezvou. Signál získaný filtrací má vlastnost, že jeho posunutá část nijak nesouvisí s původním signálem. EEG signál má velice podobnou charakteristiku, protože je tvořen aktivacemi neuronů, které nejsou nikdy

úplně stejné. Navíc je signál často znečištěn různými formami artefaktů, které mohou být způsobeny pohybem svalů na hlavě, případně pohybem očí. AR model může dosáhnout lepšího rozlišení než FFT metoda u kratších bloků signálu. AR model je také schopen zvýraznit ostré hrany, které lze často sledovat v EEG signálu.[43] Samotné rozlišení závisí na vhodné volbě řádu, což je hodnota, která říká kolik výsledků z předchozích výpočtů se v aktuálním výpočtu využije.[40]

Určení vhodného řádu je u této metody velice náročný a důležitý proces. Hodnota řádu totiž určí, jak moc výsledný signál bude vyhlazený. Pokud se zvolí nízká hodnota, signál bude mít potlačené vrcholy a tedy následná analýza bude náročná a může se stát, že nebude možné získat relevantní data. Naopak moc vysoká hodnota znamená, že vrcholy budou výrazné, ale taktéž se mohou zvýraznit artefakty, které jsou způsobené pohybem očí nebo svalovou aktivitou na hlavě. Data získaná z této extrakce jsou nekvalitní a výsledná analýza povede na falešná pozitiva, která lze velice špatně vyfiltrovat. Z tohoto důvodu není vždy ideální používat AR a je vhodné využít některou z jednodušejší aplikovatelných metod extrakce rysů EEG signálu.[40]

Shrnutí

Metod na extrakci rysů se dnes využívá velké množství. V této kapitole jsou shrnuty tři nejdůležitější metody. Vybrat vhodnou metodu je velice náročný proces, který nesmí být v systému BCI opomíjen. Každá z těchto metod se využije pro jiné rysy. Nejvíce dnes používanou metodou je FFT, která je oblíbená především kvůli její jednoduchosti a rychlosti. Data získaná touto metodou jsou dobře analyzovatelná dalšími algoritmy a ztráta dat způsobená nižším rozlišením je pro většinu aplikací zanedbatelná. Další metody se používají především v aplikacích, které vyžadují větší rozlišení. Například při klasifikaci řeči a dalších jemných mozkových aktivitách.[43]

3.4 Analýza dat

Po úpravě dat a extrakci rysů přichází na řadu finální analýza dat. Při analýze dat se zjišťuje uživatelský záměr a transformuje se z EEG signálu na příkazy, které lze předat aplikaci, ta na základě příkazů provede danou činnost. Pro analýzu dat se využívají tzv. překladové algoritmy.[43] Nejdůležitějším členem překladového algoritmu je matematický model. Model se skládá z jedné nebo více matematických rovnic. Jako vstup přijímají extrahované rysy, které byly získány s pomocí algoritmů ukázaných v předešlé kapitole. Výstupem je příkaz, který bude předán aplikaci. Model může reprezentovat jednoduchá lineární rovnice (například 3.1, kde x jsou vstupní data) nebo soustava rovnic, případně neuronová síť.[43]

$$y = 2x + 1 \tag{3.1}$$

Při tvorbě matematických modelů, je často nutné zvolit parametry. Lze je zadat ručně, nebo s pomocí trénovacích dat lze model vytrénovat, tak aby našel co nejvhodnější parametry pro analýzu konkrétního datasetu. Dalším důležitou vlastností je přesnost modelu, která je označována jako „fitness modelu“.[43] Metody učení modelu a využití pro neuronové sítě jsou zmíněny v dalších sekcích.

Následující část zhodnocuje základní modely, které jsou používány pro klasifikaci EEG signálu. Následuje sekce věnující se strojovému učení. Jsou zde ukázány rozdíly mezi učením s učitelem a bez učitele. Dále budou představeny základní typy neuronových sítí používaných v systémech BCI a určí se, které jsou nejvhodnější pro použití.

Modely

Pro klasifikování signálu EEG se využívá velké množství matematických modelů. Následující část představuje tři základní modely, které patří k těm jednodušším a jsou velice často používané pro klasifikaci signálu EEG.[43]

Klasifikace s pomocí lineární diskriminační funkce

Lineární diskriminační funkce rozděluje daný prostor do různých tříd, které mají charakteristiky dle rovnice 3.2. Kde g jsou funkce rozdělující prostor, r a s jsou třídy v daném prostoru a \vec{x} je vektor příznaků. Jednotlivé třídy určují jednotlivé vlastnosti, které lze sledovat v signálu EEG.

$$g_r(\vec{x}) > g_s(\vec{x}), r, s \in C, r \neq s \quad (3.2)$$

Nejčastěji používaná lineární diskriminační funkce má následující tvar:

$$Y = b_0 + \sum_{i=1}^n b_i x_i \quad (3.3)$$

Y označuje výsledek, který bude dále předán do aplikace. b_0 je konstanta, b_i jsou váhy a x_i jsou samotné vstupní hodnoty, které jsou využity k predikci Y . S pomocí této metody lze schopni určit zda se jedná o hledanou vlastnost signálu EEG. Tato metoda se využívá v případech, že známých hodnot, které jsou spojené s jednotlivými rysy EEG. Použitím diskriminační rovnice je možné správně určit hodnoty, které zachycený EEG představuje.[44]

Bayesovská klasifikace

Bayesovská klasifikace je metoda, která se snaží určit správné příkazy z rysů určením pravděpodobnosti jejich výskytu v daném signálu EEG. K jednotlivým typům rysů se vždy přiřadí nějaká konkrétní hodnota, která reprezentuje výsledek. Následně se s pomocí Bayesova vzorce dopočítává šance, že se konkrétní hodnota vyskytne v EEG. Jako ukázka Bayesova vzorce je použita 3.4.[27]:

$$P(v_j|A) = \frac{P(A|v_j) \times P(v_j)}{P(A)} \quad (3.4)$$

Kde v_j je hypotéza z množiny hypotéz a A jsou jednotlivé atributy, které popisují data. Problémem naivního bayesovského klasifikátoru je, že vypočítat pravděpodobnost, všech hypotéz na všech prvcích je velice náročné, a tedy používat tuto metodu na velkém množství dat může být značně nepraktické. Proto se vypočítává častěji spojení jednotlivých atributů vůči hypotéze dle vzorce:

$$P(a_1, a_2, a_3 \dots, a_n) = \prod_i P(a_i|v_j) \quad (3.5)$$

Kde a_1, a_2, \dots, a_n jsou jednotlivé hodnoty množiny A . Tímto sjednocením hypotéz mnohonásobně se zrychluje proces Bayesova výpočtu, a tedy je možné tuto metodu aplikovat v reálném prostředí.[27]

Metoda podpůrných vektorů

Metoda podpůrných vektorů (také nazývaná SVM z anglického support-vector machine) se snaží optimalizovat jednotlivé parametry modelu, aby bylo možné vylepšit výsledky analýzy. Jak může být ze základního popisu SVM patrné, tato metoda se velice podobá neuronovým sítím, které jsou popsány v dalších kapitolách. Avšak tyto dva modely se od sebe liší přístupem k jednotlivým druhům chybovosti. Neuronové sítě se především snaží snížit chybovost učení, naopak SVM se snaží snížit chybovost intervalu.[41]

SVM využívá často lineární diskriminační funkci, která je vysvětlena dříve v této kapitole. SVM ji rozšiřuje o pomocné vektory, které vytváří další ohraničení okolo samotného vektoru funkce (vektor funkce je orientovaná úsečka, která je definovaná funkcí). S jejich pomocí se snaží více generalizovat výsledky, aby byl algoritmus schopen určit jednotlivé rysy i z jiných datasetů než originálních. Tyto hranice se posouvají podle toho, zda významný rys zapadá do tohoto ohraničení. Pokud je rys pevně v některé třídě, tak se na úpravě hodnot nepodílí.[43]

3.5 Neuronové sítě

Neuronové sítě se těší velké oblibě. Jejich schopnost učit se a přizpůsobovat se velkému množství dat z nich činí výborný nástroj pro analýzu.[6] Vzhledem k tomu, že mají neuronové sítě široké uplatnění, tak se využívá i velké množství variant neuronových sítí.[11] V této práci jsou popsány kromě základní varianty i rekurentní neuronové sítě (RNN) a konvoluční neuronové sítě (CNN). Tyto modifikace pracují se základním modelem neuronové sítě (základní model neuronové sítě je popsán v následující části), který rozšiřují o metody (například konvoluce), které mají zlepšit schopnost učení. Následující sekce popisuje základní strukturu neuronové sítě a ukazuje způsoby učení, které lze využít. Dále jsou představeny dvě zmíněné modifikace a jejich využití mimo oblast klasifikace signálu EEG.

Struktura neuronových sítí

Neuronové sítě fungují na podobném principu jako neurony v mozku, které jsou popsány v kapitole 2.2. Neuronové sítě se skládají ze základních jednotek tzv. perceptronů a vstupních a výstupních proměnných. Jednotlivé perceptrony, které jsou podrobněji popsány v sekci 3.5 se umísťují mezi vstupní a výstupní hodnoty do tzv. skrytých vrstev. V neuronových sítích lze upravit vstupní hodnoty a získat výstupní, ale není možné ovlivnit co se děje ve skrytých vrstvách.

Velikost neuronových sítí je různá. Existují neuronové sítě o jedné skryté vrstvě, nebo se využívají neuronové sítě, které obsahují desítky vrstev. Větší množství skrytých vrstev, znamená mnohem náročnější ladění a ne vždy platí, čím větší tím lepší. Proto je vhodné se při návrhu neuronové sítě zamyslet, jaký je nejnižší možný počet skrytých vrstev nutný pro danou aplikaci.[14] Neuronová síť, která vznikne spojením skrytých vrstev se nazývá vrstevnatá síť, protože vzniká větší množství vrstev, které jsou navzájem propojeny.[14]

Perceptron

Perceptron je základní jednotkou každé neuronové sítě. Inspirací návrhu perceptronu byl biologický neuron, který byl popsán v kapitole 2.2. Perceptron má podobnou funkci jako biologický neuron. Přijímá vstupní hodnoty, které následně transformuje za pomoci aktivační funkce a výsledek předá do dalších vrstev neuronové sítě.[5] Jednotlivé vstupní hodnoty mají

k sobě přidanou určitou váhu, která říká jak důležitý je konkrétní vstup. Úprava těchto vah, je zásadní součástí procesu učení neuronových sítí. Každý neuron má navíc vlastní hodnotu nazývanou bias, která upravuje výsledek. Tato hodnota podobně jako váha patří k parametřům, které se snaží neuronová síť při procesu učení co nejlépe vyvážit. Samotné hodnoty spojené s biasem se následně předají tzv. aktivační funkci, která je transformuje na výstupní hodnotu. Ta je dále předána do dalších vrstev. V praxi se využívá velké množství aktivačních funkcí a tato práce se zaměřuje pouze na několik základních:

- funkce binárního kroku
- lineární funkce
- sigmoidní funkce
- ReLU
- SoftMax

Funkce binárního kroku

Funkce binárního kroku je jednou z nejjednodušších aktivačních funkcí. Tato funkce zůstává deaktivována do té doby než dostane číslo, které je větší než prahová hodnota. Tato funkce neprovádí tedy žádnou složitou transformaci a je možné takovou funkci jednoduše naprogramovat. Funkci binárního kroku lze definovat pomocí rovnic 3.6. Jak lze u uvedených rovnic pozorovat, pokud bude zadáno číslo menší než 0, funkce nebude aktivována a funkční hodnota bude 0. Jestliže vstupní hodnota bude větší nebo rovna 0, tak dojde k aktivaci a do neuronové sítě bude předána hodnota 1. Tato funkce není schopná rozlišit mezi různými hodnotami (výsledkem vždy bude buď hodnota 0 nebo 1) a má značně omezené využití.[36]

$$\begin{aligned} F(x) &= 0, x < 0 \\ F(x) &= 1, x \geq 0 \end{aligned} \tag{3.6}$$

Lineární funkce

Lineární aktivační funkce přímou úměrou transformuje vstup z výstupu. Lineární funkce se snaží řešit problémy předchozí metody. Obecná rovnice 3.7 obsahuje parametr a , který je možné nastavit na libovolnou hodnotu, která může usnadnit následnou klasifikaci. Hlavní problém této metody je, že derivace funkce je hodnota parametru a . Hodnota není nijak závislá na vstupu a tedy nelze posoudit vliv vah vstupů na výstup. Bude sice docházet k postupné úpravě vah a biasu, ale samotný krok (velikost změny) zůstane stejný. Nelze provádět jemné úpravy, které jsou v neuronových sítích důležité. Lineární funkce není schopná analyzovat složitá data, která mají více výstupů a proto není moc používaná.[36]

$$F(x) = ax \tag{3.7}$$

Sigmoidní funkce

Sigmoidní funkce transformuje vstupní hodnoty na hodnoty sigmoidní funkce $F(x) \in \langle 0, 1 \rangle$. Derivace funkce není pouze hodnota jako u lineární funkce a její gradient není konstantní. Z obecné definice této funkce 3.8 lze vidět, že funkční hodnota se blíží

k 0 v $-\infty$ a k 1 v $+\infty$. Sigmoidní funkce nemůže dosáhnou vyšších hodnot než 1 a jsou všechny vyšší hodnoty zařezány. Což v některých případech může být nežádoucí. I přesto patří Sigmoidní funkce k nejoblíbenějším aktivačním funkcím. Především pokud se využijí na trénování data, která jsou v intervalu $< 0, 1 >$; v tomto případě není problém obor hodnot sigmoidní funkce.[36]

$$F(x) = \frac{1}{e^{-x}} \quad (3.8)$$

ReLU

ReLU funkce (zkratka pro rectified linear unit) je nelineární aktivační funkcí, která patří k nejefektivnějším. Při využití této funkce nejsou vždy aktivní všechny perceptrony, ale pouze jejich část. ReLU je proto schopna ušetřit poměrně velké množství výkonu. Jak ukazuje definice 3.9, ReLU lze označit jako kombinaci binárního kroku a lineární funkce bez parametru. Nevýhodou ReLU je, že funkce je v 0 v intervalu $< -\infty, 0 >$, proto hodnoty z této části jsou vždy nulové.[36]

$$F(x) = \max(0, x) \quad (3.9)$$

Tento problém se snaží vyřešit její varianty, kde nejpoužívanější je parametrizovaná ReLU. Z definice 3.10 lze vidět, že tato varianta upravuje zápornou část tak, že využívá lineární funkci, která je ovlivněna parametrem a . Díky tomuto parametru je tato metoda vhodnější pro učení a je mnohem flexibilnější v úpravách pro vyvážení chyb.[36]

$$F(x) = \max(ax, x) \quad (3.10)$$

Softmax

Softmax využívá sigmoidní funkce k tomu, aby určila pravděpodobnost náležitosti vstupních hodnot do tříd příslušných výsledných hodnot. Softmax, podobně jako ReLU, je možné využít pro komplexní klasifikaci s velkým počtem výsledků. Výstupní vrstva má v tomto případě stejný počet proměnných, jako je počet námi hledaných rysů. Softmax je oblíbená v komplexních neuronových sítích, které vypočítávají velké množství hodnot.[36] Vzorec 3.11 popisuje, že výstup Softmax funkce pro vstup \vec{z}_j je poměr jeho exponenciální hodnoty a součtu exponenciálních hodnot všech vstupů. To znamená, že pro každý vstup je vypočítána pravděpodobnost výskytu rysu. Softmax je vhodné použít jako poslední vrstvu, která určí pravděpodobnostní zastoupení hledaných rysů.

$$\sigma(\vec{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \quad \text{pro } j = 1 \dots K \quad (3.11)$$

Metody učení neuronových sítí

Při trénování neuronových sítí existují různé způsoby, jak se neuronové sítě poskytují data. Tuto oblast si pro je možné rozdělit do tří základních kategorií:

- učení s učitelem
- učení bez učitele
- částečné učení s učitelem

Učení s učitelem

Pokud se využívá metoda trénování s učitelem, tak se nejdříve označí data určená neuronové síti. Neuronová síť se následně učí tím, že se snaží předpovědět, jaké označení daný vzorek bude mít. Následně neuronová síť porovná tuto předpověď s opravdovým výsledkem. Pokud předpověď byla správná, tak neuronová síť pokračuje. Pokud ne, neuronová síť upraví váhy na jednotlivých perceptronech a pokračuje dále v předpovídání s upravenými váhami.[30]

Následně takto trénovaná neuronová síť zpracuje validační sérii dat. Ta ukáže, jak moc se trénování podařilo a pomůže odhalit případné problémy s neuronovou sítí, případně i samotnými daty. Tato metoda je velice vhodná v situacích, kdy se disponuje velkým množstvím klasifikovaných dat. Pokud ale takové daty nejsou dostupné, tak je vhodnější využít některou z dalších metod.[18]

Částečné učení s učitelem

Tato metoda pracuje s teorií, že v běžném prostředí je možné získat snadno velké množství dat, ale tato data často nejsou označená a nelze je předem klasifikovat, případně by tato klasifikace byla časově náročná.[35]

Metoda využívá malého označeného datasetu, který byl předem vytvořen a velkého datasetu neoznačených dat, na kterých následně trénuje. Nejdříve se použijí označená data jako u trénování s učitelem a následně nechá neuronová síť vytvořit označení u neoznačeného datasetu. Dále se propojí vstupy neoznačeného datasetu se vstupem z označeného a výstupy. Po propojení znovu neuronová síť znovu trénuje na vytvořeném datasetu, jako v první fázi.[35]

Trénování bez učitele

V případě, že problém, který se klasifikuje neumožňuje vytvořit ani malé množství označeného datasetu, tak lze využít trénování bez učitele.

Při trénování bez učitele se využívají algoritmy, které prohledávají zadaná data a snaží se nalézt podobné vlastnosti mezi daty. Následně se zadávají různé úkoly, co se má s těmito daty dělat. Mezi nejčastější patří shlukování. Algoritmus se snaží roztrždit data, podle společných vlastností do skupin. Dalším úkolem může být detekce anomálií v datech. Poslední činností bývá spojování různých dat k sobě a vytváření asociací.[30]

U trénování bez učitele je velice složité určit správnou odpověď. Proto je složité zhodnotit přesnost algoritmu. Z tohoto důvodu má metoda omezené využití. I přesto existují oblasti, kde by bylo velice komplikované získat klasifikovaná data. Metoda je dnes například využívána při odhalování nádorů v lidském těle.[30]

Modifikace neuronových sítí

V předchozí sekci jsou popsány základní formy neuronových sítí. Tato základní varianta má značná omezení. Tato omezení se snaží odstranit různé modifikace. Následující část popisuje dva druhy neuronových sítí, které jsou dnes v oblasti klasifikace EEG oblíbené: rekurentní neuronové sítě a konvoluční neuronové sítě.

Rekurentní neuronové sítě

Klasické neuronové sítě rozhodují o klasifikaci dané části datasetu pouze podle aktuálně dostupných informací - nemají paměť. Tuto nevýhodu se snaží odstranit rekurentní neuronové sítě.[25]

Rekurentní neuronové sítě si předávají výstupy mezi vrstvami podobně jako základní neuronové sítě, ale tento výstup je vrácen zpátky do vrstvy jako další vstup. Tímto si vytváří paměť. Následná klasifikace zpracovává nejen aktuální informaci, ale i informaci z předchozích dat. Rekurentní neuronové sítě jsou schopny analyzovat data v sekvenční návaznosti a extrahovat z těchto dat kontext, který by se při použití základních neuronových sítí vytratil.[25]

Schopnost neuronové sítě si pamatovat je velice užitečná, ale přináší také nové problémy. Mezi nejvýznamnější patří ztráta nebo exploze gradientu. Exploze gradientu znamená, že neuronová síť si nashromáždila takové množství starých dat, že veškeré nové vstupy jsou silně ovlivněny minulými vstupy a neuronová síť ztrácí schopnost se učit.[33]

Tento problém se snaží adresovat různé metody, kde mezi nejpoužívanější patří krátkodobá paměť (kterou navrhl ve své práci Hochreiter a Schmidhuber). Metoda se nazývá dlouho-krátkodobá paměť (LSTM z anglického long short-term memory).[15] LSTM nastavuje interval zapomínání. Neuronová síť po určité době zapomene předchozí výsledky a pracuje pouze s aktuálními.

Při trénování se neuronová síť vyhne situaci, kdy by nová data nebyla schopná ovlivnit neuronovou síť, kvůli přehlcené paměti.[25]

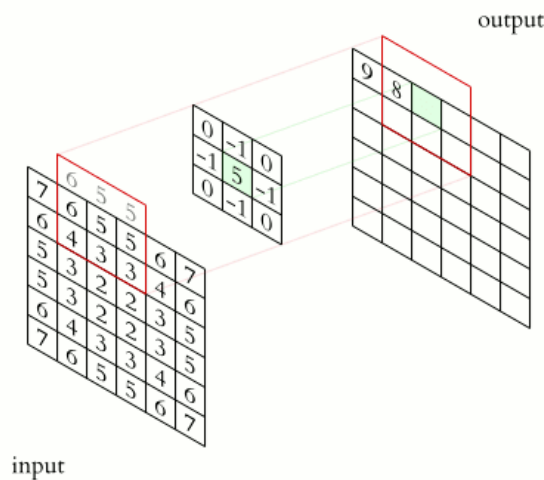
Rekurentní neuronové sítě se využívají na včasné odhalování nemocí (parkinsonova choroba).[45] Dále se používají na rozpoznávání řeči.

Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN) patří mezi další modifikace, které lze úspěšně použít při klasifikaci dat v BCI. Tato metoda je podobně jako RNN schopna zpracovávat data v posloupnosti. Získá se mnohem přesnější informace o aktuálních datech a lze je mnohem přesněji klasifikovat.[25]

Konvoluční neuronové sítě, fungují na jiném principu než RNN. CNN se skládá se tří typů vrstev. Konvoluční vrstva, pooling a plně propojená vrstva. Konvoluční a pooling vrstva slouží k extrahování jednotlivých rysů z dat. Následná vrstva ze standardní neuronové sítě klasifikuje tyto extrahované rysy.[46]

Konvoluční vrstva využívá takzvaných filtrů. Filtry jsou pole hodnot, které se aplikuje na data za využití operace konvoluce, která spočítá výslednou hodnotu znásobením hodnot filtru s hodnotami v datech. Ukázku použití filtru lze vidět na obrázku 3.3. Důležitou otázkou může být, co se udělá s oblastmi, které nejsou pokryty daty. Zde se využívá padding, kdy se prázdné oblasti doplní nulami, aby bylo možné aplikovat filtr. Při tvorbě neuronové sítě je důležité správně zvolit vhodnou velikost filtrů a jejich posun. Všeobecně lze říct, že menší filtry podávají lepší výsledky za cenu nižší rychlosti. Konvoluce je velice užitečná metoda, zvýrazňuje jednotlivé rysy v datech, proto bude jednodušší data klasifikovat.[46]



Obrázek 3.3: Aplikace filtru na data v konvoluční síti, Wikipedia [10]:

Další metoda využívaná v CNN je tzv. pooling. Tato metoda podobně jako konvoluce aplikuje filtr na data. Neprovádí žádnou složitou matematickou operaci, ale pouze vybere/určí hodnotu dle typu poolingu. Mezi nejčastější patří max pooling, kdy je vybrána nejvyšší hodnota ve filtru, případně average pooling, který zprůměruje hodnoty ve filtru.[46]

Konvoluční neuronové sítě se nejčastěji využívají při klasifikaci obrázků, kde jsou velice snadno aplikovatelné a dosahují velice dobrých výsledků. V oblasti EEG se začaly využívat neuronové sítě poměrně nedávno a ukazuje se, že mohou mít dobré výsledky při aplikaci na signálu, který je kontaminován velkým množstvím šumu.[46]

Zhodnocení

Každá modifikace neuronových sítí má jiné použití. Všechny výše představené neuronové sítě lze využít při klasifikaci EEG signálu. Použití každého typu neuronové sítě závisí na problému, který se snaží BCI systém odstranit. Pro levnější systémy BCI může být vhodnější spíše konvoluční síť. CNN je schopná zpracovat signál, který je ovlivněn velkým množstvím šumu. Naopak kvalitní signál může být vhodnější zpracovat pomocí RNN nebo standardní neuronové sítě.

Kapitola 4

Implementace aplikace BCI

Cílem této práce bylo vytvořit rozhraní, s pomocí kterého bude možné ovládat demonstrační aplikaci za využití signálů EEG. Analýzu provádí neuronová síť. Neuronová síť vyžaduje pro své fungování velké množství dat, které nejsou pro systém OpenBCI veřejně dostupné. V rámci implementace jsem vytvořil pomocné skripty, které sbírají data a vytváří zpětnou vazbu. S pomocí této vazby, lze lépe zvolit způsob generování EEG signálu (viz 2.5) a lépe optimalizovat neuronovou síť.

Nejdříve je představen návrh celého systému a jsou popsány knihovny a vývojářská prostředí, které byly využity pro implementaci. V další části jsou popsány pomocné skripty. Následně je představena architektura neuronové sítě a experimenty, které vedly k volbě této architektury.

Poslední část se zabývá tvorbou demonstrační aplikace, kterou jsem vytvořil v Unity a Blenderu.

4.1 Návrh systému BCI

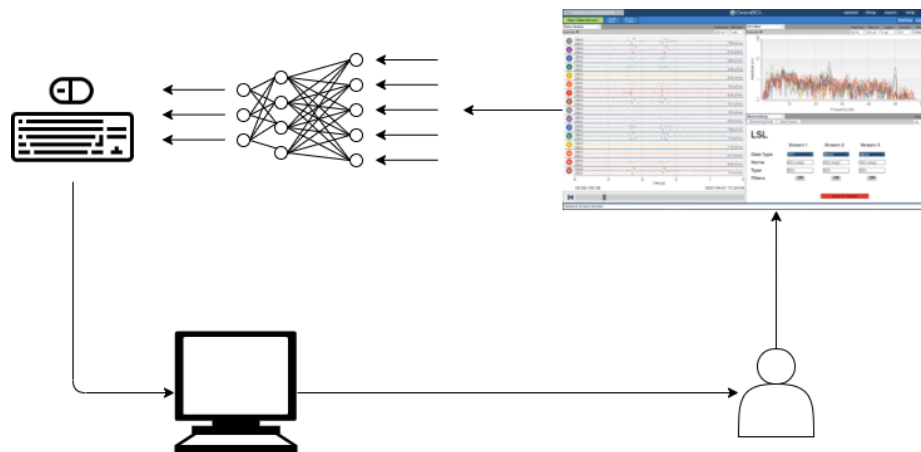
Návrh systému BCI se skládal z neuronové sítě, která vyhodnocuje data získaná z OpenBCI GUI a dále je součástí implementace demonstrační aplikace.

Schéma celého systému je ukázáno na obrázku 4.1. Uživatel provádí činnost, která generuje signál EEG. Způsob generování signálu EEG byl zvolen na základě experimentů, které jsou implementovány s pomocí skriptů (ty jsou popsány v sekci 4.3). Pro jednoduché systémy BCI jsou nejvhodnější motoricky evokované potenciály nebo zrakové evokované potenciály. Obě formy byly představeny v sekci 2.5. Získání EEG signálu je dle návrhu provedeno s pomocí OpenBCI GUI, které zaznamenává signál a posílá jej dále. Data jsou předána neuronové síti, jejíž architekturu představuje sekce 4.4. Neuronová síť data vyhodnotí a stiskne příslušnou virtuální klávesu, která předá robotovi příkaz k provedení akce.

Prostředí demonstrační aplikace mělo dle návrhu vypadat jako ulice, ve které se pohybuje robot. Plocha byla ohraničena bariérami, které zajišťují, aby se robot nepohyboval mimo vytvořenou oblast. V prostředí byl umístěn i předmět, který robot uchopoval. Dále zde byly umístěny různé objekty, které vytvářely dojem reálného prostředí.

Robot podle návrhu měl mít čtyři kola, s pomocí zajišťují pohyb s dostatečnou flexibilitou. Dále měl být robot vybaven ramenem, které slouží k uchopení předmětu.

Demonstrační prostředí dle návrhu mělo být uvedeno hlavní nabídkou, kde může uživatel hru zapnout nebo vypnout. Důležité je, aby se mohl uživatel do hlavní nabídky kdykoliv vrátit.



Obrázek 4.1: Návrh BCI systému, který využívá pro získání EEG OpenBCI GUI a následnou analýzu neuronovou síť. Neuronová síť provádí stisknutí virtuální klávesnice. Na základě stisků provádí virtuální robot akce na které může uživatel reagovat.

4.2 Použité knihovny a vývojářská prostředí

Systém BCI je velice komplexní. Při tvorbě této aplikace bylo nutné použít velké množství knihoven a vývojářských prostředí.

Jazyk Python byl využit pro tvorbu datasetů, testování různých forem generování EEG signálů a analýzu dat. Pro testování vizuálních vjemů byla využita knihovna PsychoPy. Architektura neuronové sítě a její testování bylo provedeno s pomocí frameworku Tensorflow. Pro získávání EEG signálu jsem využil systém OpenBCI¹. OpenBCI využívá k zaznamenávání EEG signálu 16 suchých elektrod, kterou jsou umístěny do plastové helmy. Součástí tohoto systému je OpenBCI GUI. Tato aplikace nabízí možnost nahrávat data ze systému a předávat je dále s pomocí LSL protokolu.

Tvorba testovací aplikace, je rozdělena do dvou částí. Tvorba modelu robota a tvorba demonstrační aplikace.

Model robota, jsem vytvořil v návrhovém prostředí Blender². Virtuální prostředí jsem navrhl ve videoherním Enginu Unity³.

4.3 Návrh a implementace pomocných skriptů a sběru dat

Systémy BCI vyžadují pro fungování správnou volbu vzniku EEG (činnosti, kterou bude uživatel provádět, aby byl generován pravidelný analyzovatelný signál). Systémy BCI využívají různé formy vzniku EEG signálu (viz 2.5).

V této práci je využit systém OpenBCI, který patří k levnějším a obsahuje pouze 16 elektrod. Pro tento systém jsou nejvhodnější buď vizuálně evokované potenciály 2.5, nebo motoricky evokované potenciály (případně MI) 2.5.

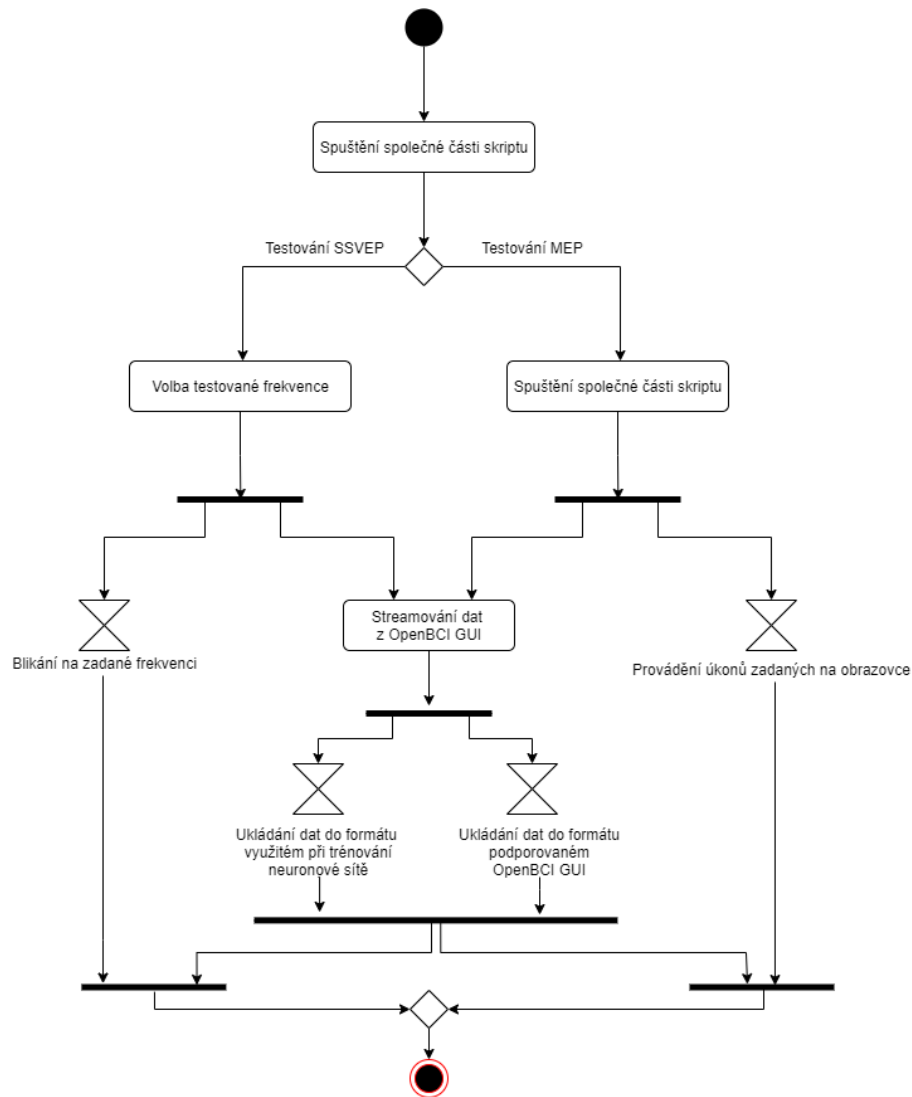
Určit správně, která z těchto metod je nejlepší je složité, proto jsou navrženy dva skripty, které mají výběr metody zjednodušit. Prvním skript testuje vizuálně evokované potenciály (konkrétně SSVEP) s pomocí knihovny PsychoPy. Druhý se zaměřuje na pohy-

¹<https://openbci.com/>

²<https://www.blender.org/>

³<https://unity.com/>

bově evokované potenciály. Komunikace s OpenBCI je složitá. Bylo by nevhodné, aby každý skript měl vlastní rozhraní pro komunikaci mezi headsetem a aplikací. Z tohoto důvodu bylo vytvořeno jednotné rozhraní, které využívají oba skripty. Skripty se spouští pomocí souboru `main.py`. Následně si může uživatel vybrat, co chce testovat. Použití těchto skriptů je ukázáno na obrázku 4.2. Tento diagram je důkladněji vysvětlen v dalších částech.



Obrázek 4.2: UML diagram zobrazující průběh skriptů. Všechny skripty jsou spouštěny zároveň a na základě uživateli dochází k zanořování a volání pomocných skriptů.

Skript pro testování SSVEP

První skript se zaměřuje na testování SSVEP (viz 2.5). Pro tvorbu tohoto skriptu byla zvolena knihovna PsychoPy⁴. Časování při blikání, je pro testování SSVEP velice důležité a PsychoPy nabízí přesné časovače, které jsou schopny startovat pravidelné události. PsychoPy dále nabízí nástroje pro výpočet snímků, které je třeba prezentovat, aby bylo do-

⁴<https://www.psychopy.org/>

saženo požadované frekvence. Knihovna je schopná rychle měnit barvy, tak aby bylo možné dosáhnout efektu blikání a je schopná vytvářet vzory, které se využívají při generování SSVEP (třída `GratingStim`).

Po spuštění hlavního souboru skriptu a výběru SSVEP je uživatel požádán o zadání frekvence, která bude generována. Zvolení frekvence bylo ponecháno na uživateli, protože každý displej má jinou obnovovací frekvenci a tento fakt je vhodné brát do úvahy při volbě frekvence. Po zvolení frekvence bude uživateli prezentován vizuální stimul, který bude blikat 120 sekund. Ve stejný moment jsou také spuštěny dva procesy, které zaznamenávají data pro OpenBCI GUI a trénování neuronové sítě. Data jsou generována pomocí modulu `StreamingLSL`, který je společný pro testování SSVEP a MEP (více k tomuto modulu v sekci 4.3). Oba procesy končí ve stejný moment jako prezentování stimulu.

Skript pro testování MEP

Pokud chce uživatel testovat MEP, jsou mu postupně zadány tři úkoly. Pohyb levou rukou, pohyb pravou rukou a žádný pohyb (meditace). Každý úkol trvá padesát sekund a po ukončení dané činnosti je pětisekundová pauza, během které se může uživatel připravit na další úkol. Pro každou činnost jsou vytvořeny samostatné datasety, určené pro neuronovou síť a pro OpenBCI GUI (podobně jako u předchozího skriptu 4.3).

Modul pro komunikaci s EEG headsetem

Pro získávání dat z elektrod je využívána aplikace OpenBCI GUI. OpenBCI GUI nabízí možnost přehrávání dříve nahraných dat. Data jsou předána aplikaci v textovém souboru, který je představen v části 4.3. Dále aplikace vykresluje měřená data, což je užitečné při prvotní analýze na začátku testování. S pomocí vykresleného spektra lze jednoduše určit například použitelnost SSVEP metody. Tyto vlastnosti byly užitečné, kvůli aktuální epidemiologické situaci. Počet sezení byl omezený a nebyl prostor analyzovat nasbíraná data. Přehrávání dat bylo vhodnou alternativou k živému analyzování.

OpenBCI GUI nabízí možnost streamovat data s pomocí různých protokolů. Pro tento typ aplikace je nejvhodnější protokol LSL (protokol lab streaming layer určený ke sběru dat při experimentech⁵). Rozhraní, které se využívá v Pythonu pro komunikaci s pomocí protokolu LSL je `pylsl`⁶. OpenBCI GUI odesílá data v různých formátech podle toho, který typ dat posíláme. Pokud například se zpracovávají hodnoty, které se získaly z elektrod, aplikace je pošle v jednom seznamu. Naopak pokud se zpracovává spektrum, tak jsou data poslána v šestnácti blocích, kde je v každém bloku 125 hodnot, které reprezentují frekvenci od 1 do 125 Hz. Je tedy nutné mít implementovány rozdílné funkce pro ukládání dat spustitelných v OpenBCI GUI a dat určených pro neuronovou síť.

Datasety

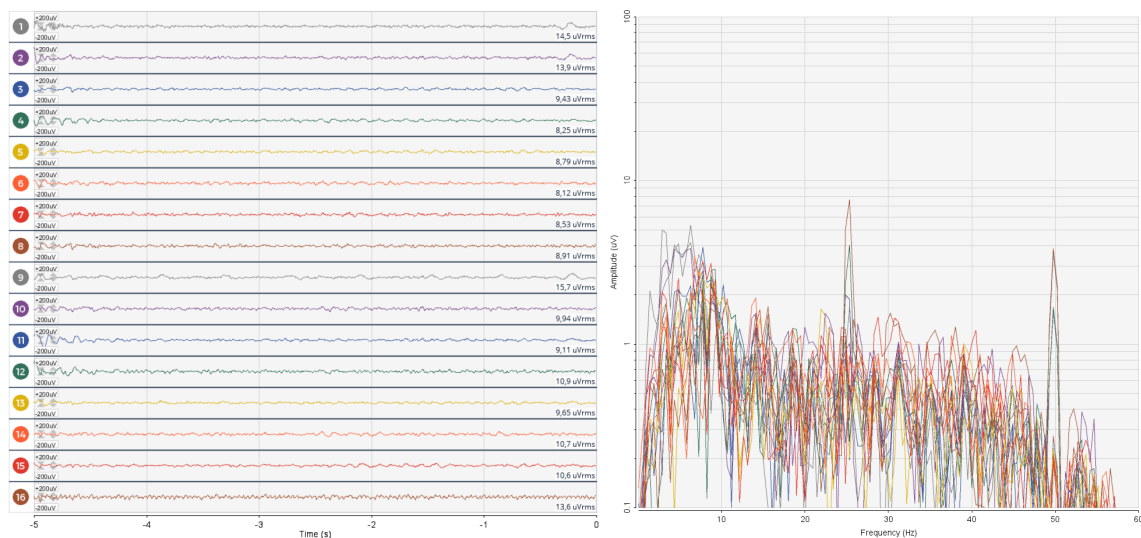
Skripty po zapnutí generovaly dva druhy datasetů. První byl určený pro trénování neuronové sítě a druhý bylo možné spustit pomocí aplikace OpenBCI GUI.

Aplikace OpenBCI GUI odesílala data, která reprezentují signál generovaný mozkiem buď ve formě rychlé fourierovy transformace (tyto hodnoty jsou určené pro trénování neuronovou sítí), nebo aktuální hodnoty v mikrovolttech. Přibližné vizuální odhady těchto hodnot

⁵<https://github.com/sccn/labstreaminglayer>

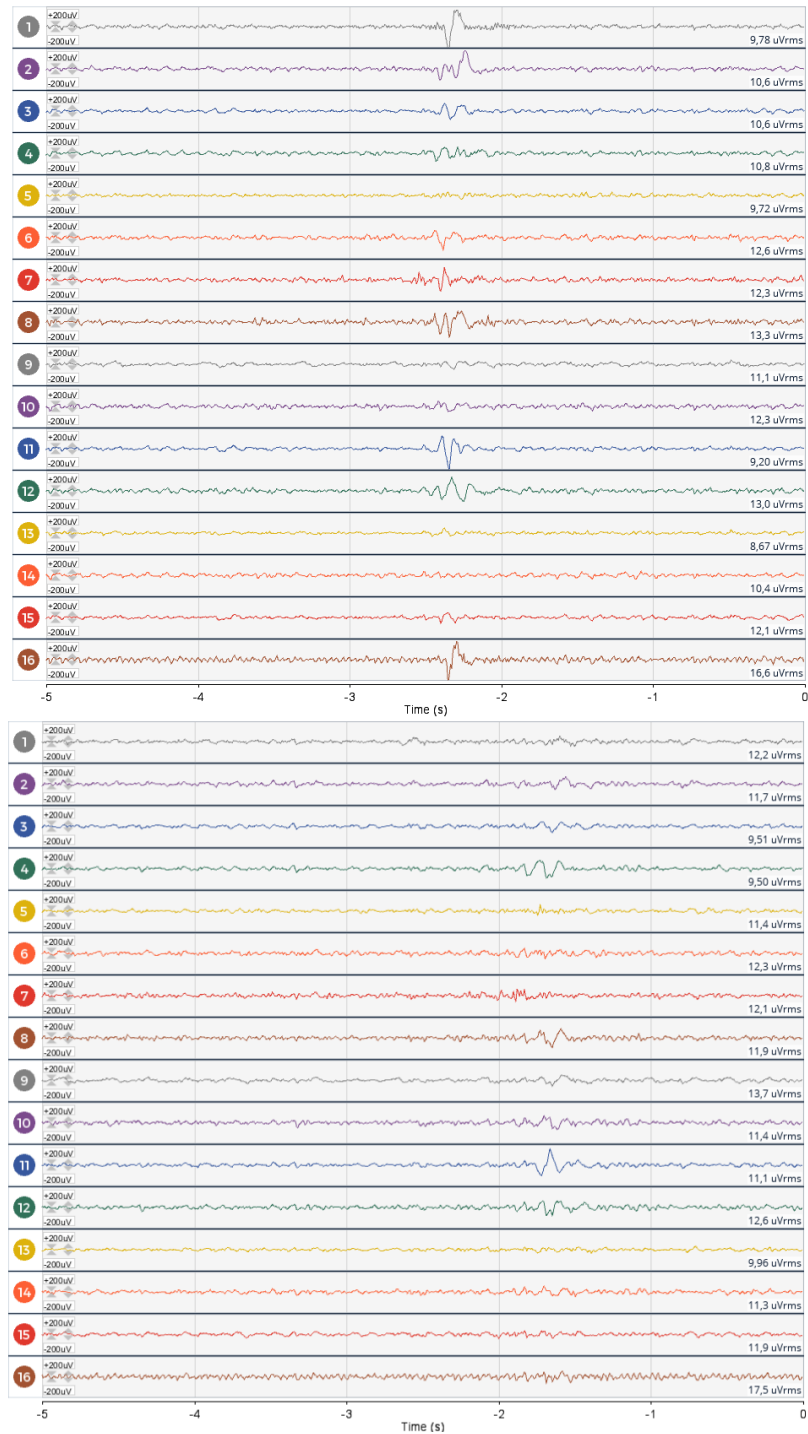
⁶<https://pypi.org/project/pylsl/>

je možné vidět na obrázku 4.3. Vlevo je ukázka signálu produkovaného v klidovém stavu a vpravo je převod těchto hodnot do spektra pomocí FFT.

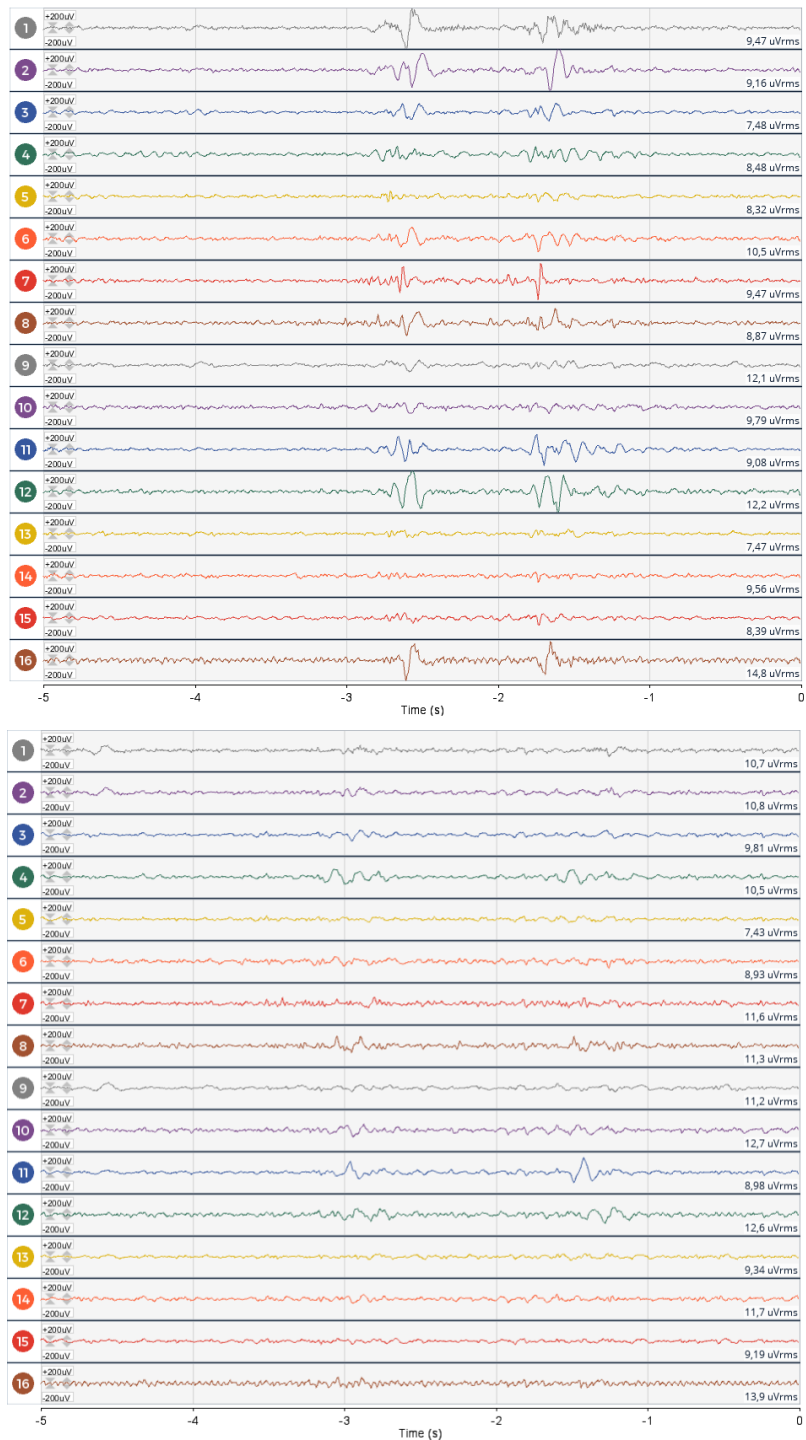


Obrázek 4.3: Porovnání běžného vykreslení mozkového signálu a jeho reprezentace ve spektru. (ukázka vytvořená v aplikaci OpenBCI GUI)

Na následujících dvou stranách jsou ukázány jednotlivé pohyby rukou (obrázky 4.4 a 4.5), které v této práci byly měřeny. Obrázky zobrazují EEG signál v základní nepřevedené formě, na který byla aplikována pásmová zadrž a pásmová propust (viz 2.4), které filtrují šum generovaný zásuvkami a hodnoty, které jsou ovlivněny aliasingem. Obrázek 4.4 zachytává měřená mávnutí - levou rukou jedenkrát, pravou rukou jedenkrát. Obrázek 4.5 zachytává mávnutí - levou rukou dvakrát a pravou rukou dvakrát. Obrázky ukazují, že rozdíly mezi levou a pravou rukou jsou poměrně výrazné. Levá ruka má mnohem výraznější vrcholy. Na elektrodě 11 (elektroda F3 v systému 10-20 2.5) jsou u pohybu levé ruky vrcholy záporné. Naopak u pravé ruky jsou kladné.



Obrázek 4.4: Porovnání jednotlivých signálů pohybu rukou. Jsou zde ukázány následující pohyby: mávnutí levé ruky (obrázek výše), mávnutí pravé ruky (obrázek níže). Levá ruka má výraznější vrcholy a na elektrodě 11 má výraznou negativní výchylku. Naopak pravá ruka má na elektrodě 11 výchylku pozitivní (ukázka vytvořená v aplikaci OpenBCI GUI)



Obrázek 4.5: Porovnání jednotlivých signálů pohybu rukou. Jsou zde ukázány následující pohyby: mávnutí levé ruky dvakrát (obrázek výše), mávnutí pravé ruky dvakrát (obrázek níže). Signál při dvou mávnutích reaguje podobně jako při jednom mávnutí (obr. 4.4), hlavní rozdíl je, že signál se po malém intervalu zopakuje. (ukázka vytvořená v aplikaci OpenBCI GUI)

Data určená pro trénování neuronové sítě byla uložena do proměnné `numpy array`, která má tvar $(-1, 16, 60)$, kde 16 je počet kanálů, které se zpracovávají a 60 je maximální frekvence, která není ovlivněna aliasingem (viz 2.4). Data byla následně uložena do `numpy` souborů. Každý soubor obsahuje 5 sekund záznamu přičemž záznamy byly pořízené rychlostí přibližně 25 vzorků za sekundu. Tedy každý soubor obsahuje mezi 120-130 vzorky. Takto vytvořený dataset lze následně jednoduše načíst s pomocí funkce `numpy.load` a lze jej předat neuronové síti.

OpenBCI GUI při každém sezení vytváří nový záznam. Tyto záznamy bylo složité využít k analýze, případně k dalšímu streamování, protože se jedná o záznam dlouhé relace. Skripty tedy implementují vlastní systém ukládání dat, tak aby je bylo možné následně lépe analyzovat. OpenBCI nenabízí popis, v jakém formátu mají být data uložena. V rámci implementace byl nejdříve zkoumán formát uložených souborů. Ze zkoumání vyplynulo, že soubor musí obsahovat hlavičku se základními informacemi, které jsou určené pro OpenBCI GUI.

```
%OpenBCI Raw EEG Data
%Number of channels = 16
%Sample Rate = 125 Hz
%Board = OpenBCI_GUI$BoardCytonSerialDaisy
```

Výpis 4.1: Hlavička souboru

Hlavička 4.1 obsahuje název, informaci o počtu kanálů, vzorkovací frekvenci a název desky, která byla použita. Tyto informace OpenBCI GUI použije k rozlišení mezi různými typy systémů OpenBCI, proto je důležité tyto informace mít zapsané správně. Dále následují data, která jsou indexována od 0 do 125. Tento index se periodicky opakuje. Dále se ukládají data ze samotných elektrod. Po EEG hodnotách by byly umístěny další hodnoty z měření (akcelerometr a podobně), ale tato data GUI nepotřebuje, tak jsou pouze vyplněná nulami. Nakonec je čas měření. Vytvořený soubor je možné následně spustit pomocí OpenBCI GUI a informace o měřeních EEG jsou znovu přehrávány podobně jako v reálném testování. Hlavní rozdíl mezi přehráváním ze souboru a praktickým měřením je stabilita. Data ze souboru jsou přehrávána v přesně daných intervalech, naopak headset může mít občas problémy dosáhnout správného počtu vzorků.

Experimenty provedené pomocí skriptů

Experimenty byly rozděleny do dvou částí. V první části se byly testovány metody vzniku EEG signálu a hledala se ta nejvhodnější. Druhá část byla zaměřena na tvorbu datasetů, použitých pro trénování neuronové sítě.

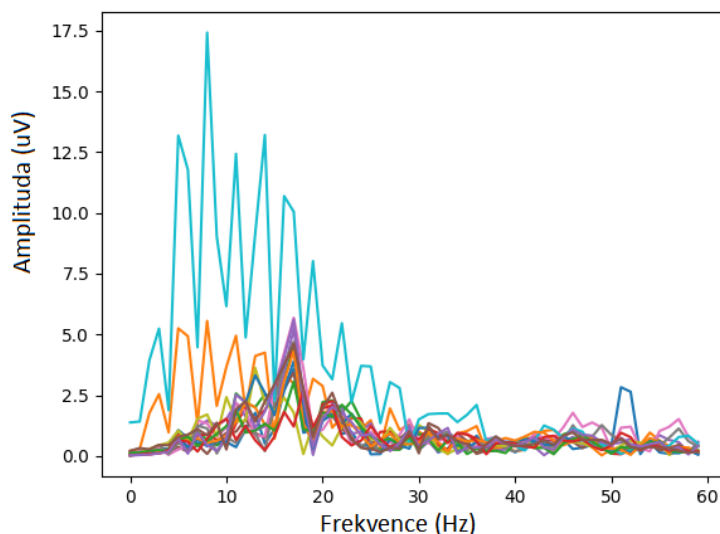
Výběr vhodné metody pro vznik EEG

Zvolit vhodnou metodu, kterou se bude vytvářet signál EEG, je při návrhu systému BCI důležité. V původním návrhu se mělo provést velké měření na více lidech, aby data obsahovala anomálie, které se mohou u některých lidí vyskytovat. Z důvodu špatné epidemiologické situace toto nebylo možné provést.

Testování SSVEP

Při omezeném testování, které umožnila epidemiologická situace bylo zjištěno, že signál produkovaný pohledem na blikající čtverec neodpovídá tomu, jak by signál SSVEP měl vypadat. Jak lze z grafu 4.6 vidět, byl produkován ustálený signál na špatné frekvenci. Čtverec blikal na frekvenci 10 Hz a ustálený signál se pohybuje na frekvenci přibližně 17 Hz. Při zadání jiné frekvence se výsledek téměř neměnil. Navíc je možné vidět, že elektroda 10, která odpovídá elektrodě F8 na schématu 10-20 (obrázek 2.5 kapitola 2.4), má pravidelný signál, který ale neodpovídá amplitudě signálu SSVEP. Navíc je tento signál generován na elektrodě, která neměří oblast zrakového centra (viz 2.3).

Chybějící amplituda SSVEP signálu může být způsobena různými faktory (každý člověk reaguje jinak na vizuální vjemy). Zjištění problému by vyžadovalo testování na více lidech, které nebylo možné. Pro ověření, zda není problém v časovačích PsychoPy (případně jiný problém v implementaci), jsem navrhl další dva skripty. Implementovány byly v PyQT⁷ a v PyGame⁸. Vytvořené skripty byly použity na další experimentování. Bohužel tyto experimenty nevedly na žádné průkazné výsledky. Měření dosáhla podobných hodnot jako v případě PsychoPy. V některých případech měření byla i horší a nedařilo se vygenerovat žádnou amplitudu.



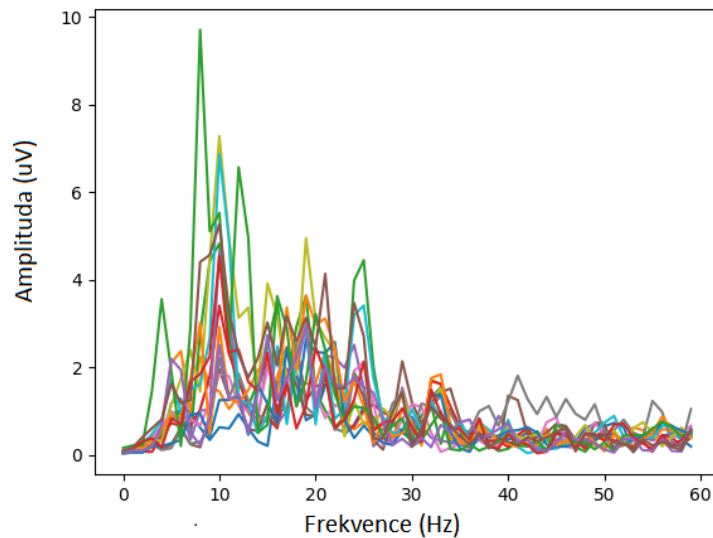
Obrázek 4.6: Graf SSVEP, který produkovalo blikání

⁷<https://riverbankcomputing.com/software/pyqt/intro>

⁸<https://www.pygame.org/wiki/about>

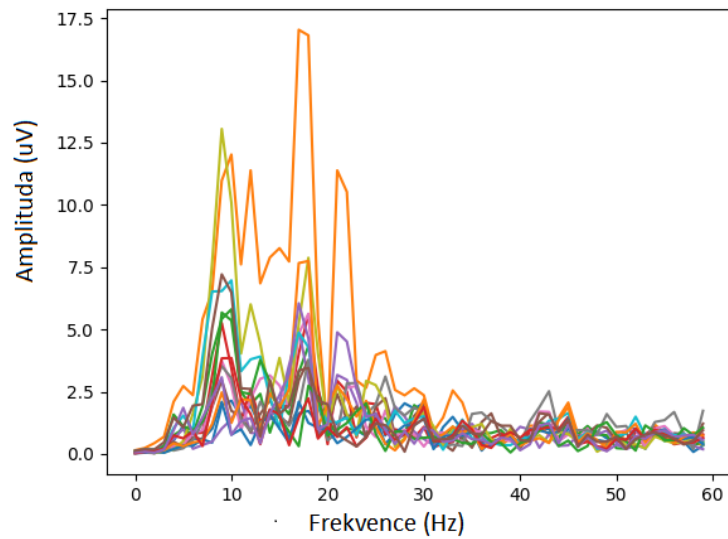
Z tohoto důvodu není možné využít metodu SSVEP pro ovládání aplikace a bylo třeba přejít na testování MEP.

Testování MEP bylo prováděno s pomocí druhého skriptu, který byl popsán 4.3. Při testování bylo zjištěno, že pohyb pravým zápěstím produkuje výrazné vzruchy v měřeném EEG signálu. Když se převedly tyto hodnoty do spektra a vynesly se na osu, vznikl graf 4.7. Na grafu lze vidět zelenou čarou výraznou amplitudu na elektrodě 13 (elektroda T3 obr. 2.5). Výrazná amplituda je také žlutou barvou na elektrodě 10 (elektroda C3 obr. 2.5). Takto výrazné amplitudy je možné využít při učení neuronové sítě.



Obrázek 4.7: Graf MEP, při pohybu pravým zápěstím

Signál získaný testováním levé ruky je ukázán na grafu 4.8. Hodnoty jsou podobné signálu, který byl generován pravou rukou. Signál je ovšem výraznější a je generován i na elektrodách 4 a 12 (elektrody C4 a F4). Signál se tedy liší mezi měřeními pravé a levé ruky proto je vhodný pro BCI aplikaci. Bylo tedy možné vytvořit dataset pro neuronovou síť. Data se získávala opakovaným prováděním činností, které zadával testovací skript (viz 4.3). Neuronová síť vyžaduje velké množství dat. Dataset byl vytvářen během několika sezení. Jak již bylo zmíněno epidemiologická situace neumožňovala získávání dat od více lidí, proto bylo této části věnováno více času a měřil jsem pouze svůj signál EEG.



Obrázek 4.8: Graf MEP, při pohybu levým zápěstím

Shrnutí

Skripty popsané v této části byly důležité pro výběr vhodného způsobu generování EEG signálu. S pomocí těchto skriptů byl tento proces zjednodušen a jednotlivé testy byly identické. Průběžné výsledky je tedy možné porovnat. Využití jednotných skriptů značně zrychlilo sběr dat. Experimenty prováděné s pomocí těchto testů ukázaly, že není možné využít SSVEP signál pro BCI aplikaci, protože signál byl nepřesný. Zjištění důvodů by vyžadovalo testování na více lidech, což v současné situaci není možné. Pro aplikaci jsou tedy využity signály generované s pomocí MEP, které jsou výrazné a dobře rozlišitelné.

V rámci těchto skriptů je navíc implementována predikce pomocí modelu. Trénování neuronové sítě je popsáno v následující části a popis ovládání je uveden v sekci 4.5.

4.4 Neuronová síť

Analýzu EEG signálu lze provádět pomocí různých matematických modelů. Detaily byly popsány v sekci 3.4. V této práci pro analýzu EEG dat, byly využity neuronové sítě. Neuronová síť je schopna dosáhnout podobných výsledků jako jiné metody, bez nutnosti provádění složitých výpočtů programátorem.

Implementace neuronové sítě v této práci je provedena v souboru `CNNTensorFlow.py`.

V souboru lze nalézt také funkce určené k přípravě dat pro neuronovou síť. V následující části je popsána příprava, která se provádí před každým experimentem, popis daného experimentu a výběr nejlepší architektury neuronové sítě. Na závěr je vybraná architektura zhodnocena.

Průběh experimentů

Před prováděním samotných experimentů je nejdříve nutné připravit data na trénování. Data jsou rozmístěna po složkách podle toho, která činnost byla prováděna. Jsou to následující: `Left_Hand`, `Right_Hand`, `Meditation`. Poté jsou složky rozmístěny podle toho k jaké činnosti v rámci neuronové sítě budou využity (trénování, validace, testování). Před každým experimentem byla data rozmístěna do těchto kategorií v poměru 80:10:10. Následně jsou data načtena a označena hodnotami 0-2, kde 0 je levá ruka, 1 je pravá a 2 je meditace. Označená data se předávají hlavní funkci, která vyjme označení dat a zbytek předá neuronové síti, která na nich trénuje. Na závěr je provedeno testování, ze kterého byla vypočítána chybová matice. Chybová matice říká, jaké procento z celkového počtu vzorků se umístilo v dané kombinaci. Vzhledem k tomu, že počet vzorků dat levé ruky, pravé ruky a meditace není stejný, tak nejdůležitější sledovaná hodnota je chybovost, což je hodnota reprezentující procento vzorků, které se umístily při testování do špatné kategorie. Nejlepší model bude mít ve výsledku nejnižší chybovost.

Experimenty

V rámci experimentů byly testovány různé architektury neuronových sítí. Využití neuronových sítí v systémech BCI je poměrně nový fenomén, a proto jsou různé názory na to, která architektura neuronové sítě je nejvhodnější. Mezi nejčastěji používané neuronové sítě (jak bylo zmíněno v sekci 3.5) patří rekurentní neuronové sítě a konvoluční neuronové sítě. Obě tyto modifikace mají odlišné výhody, proto byly v následující části testovány. Další architekturou, která se dnes začíná využívat je kombinace RNN a CNN. V této architektuře se provádí konvoluce jako u běžné CNN, ale využívá se také vrstev LSTM, které vytváří v neuronové síti paměť. Tato paměť je následně užitečná při vyhodnocování.[47]

Všechny neuronové sítě byly navrženy s pomocí frameworku Tensorflow⁹. Tensorflow nabízí nástroje pro tvorbu neuronových sítí. S pomocí Tensorflow je možné navrhovat architektury pro konvoluční i rekurentní neuronové sítě, ale je možné jednotlivé vrstvy neuronových sítí kombinovat dle potřeby. Tensorflow také nabízí nástroje pro vyhodnocování výsledků trénování.

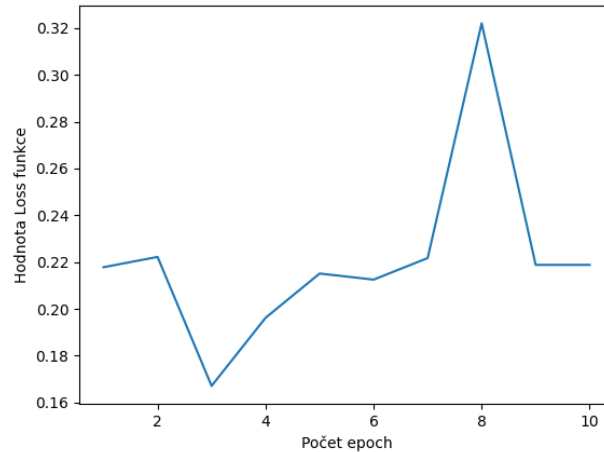
Rekurentní neuronová síť

Rekurentní neuronové sítě, jak bylo zmíněno v části 3.5, se nejčastěji využívají při analýze řeči. Podobně jako v oblasti rozpoznávání řeči při analýze EEG signálu je užitečná paměť.

⁹<https://www.tensorflow.org/>

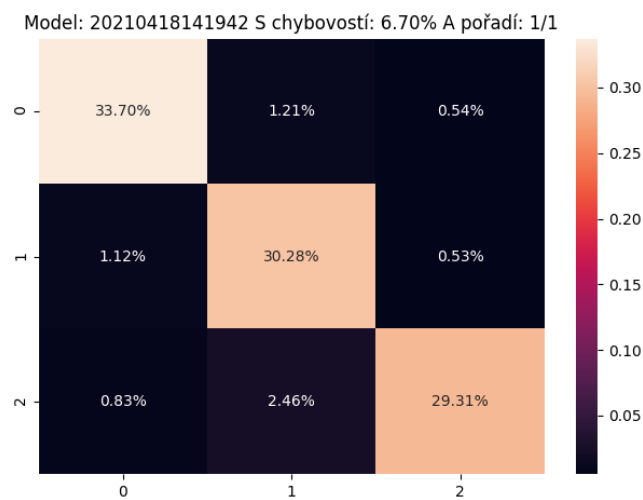
Při analýze se nesleduje pouze aktuální stav signálu, ale i předchozí hodnoty, které podávají lepší informaci o jeho průběhu.

Při trénování neuronové sítě je zvolen počet takzvaných epoch. Epocha je počet opakování trénování neuronové sítě na datasetu. Tato hodnota je pro správné trénování neuronové sítě velice důležitá. Pokud se zvolí velká dochází ke zhoršení výsledků neuronové sítě. Vybraná neuronová síť byla testována na deseti epochách a výsledek lze vidět na grafu 4.9. Z grafu je možné vidět, že nejlepší hodnoty funkce bylo dosaženo ve třetí epoše, proto v dalších testováních byla zvolena právě tato hodnota.



Obrázek 4.9: Průběh loss funkce pro deset epoch

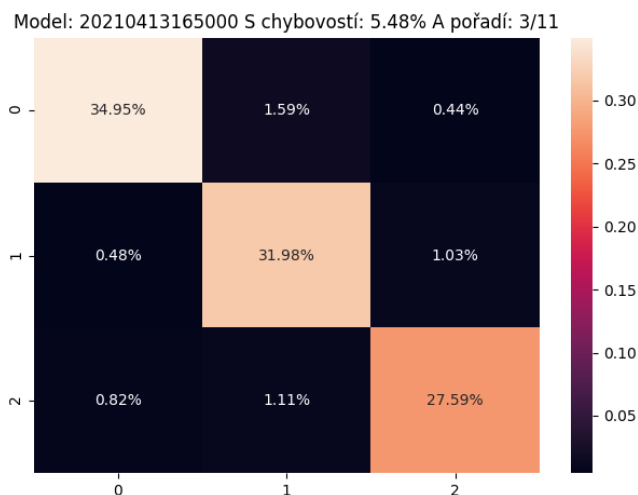
Při testování se ukázalo, že velké neuronové sítě měly nejen vyšší časovou náročnost na zařízení kde byly trénovány, ale také byly méně přesné na datasetu, který byl použit k trénování. Jak je ukázáno na obrázku 4.10, který zobrazuje chybovou matici pro neuronovou síť s téměř 5 miliony parametrů, tak neúspěšnost modelu je 6,7 % (procentuální hodnota, kolikrát se model mýlil ve svých predikcích).



Obrázek 4.10: Chybová matice velké rekurentní neuronové sítě

Naopak menší neuronová síť má neúspěšnost 5,48 %. Výsledky této neuronové sítě jsou ukázány na obrázku 4.11. Vyšší neúspěšnost velkých neuronových sítí je dána poměrně nízkým množstvím dat, které bylo možné během omezených sezení nasbírat. Podobná situace, byla i u dalších typů neuronových sítí, a proto dále v práci tento parametr není zmíněn.

Celková úspěšnost rekurentní neuronové sítě v porovnání s ostatními zde testovanými sítěmi byla horší. Z tohoto důvodu nebylo s touto modifikací prováděno další testování.



Obrázek 4.11: Chybová matice malé rekurentní neuronové sítě

Konvoluční neuronová síť

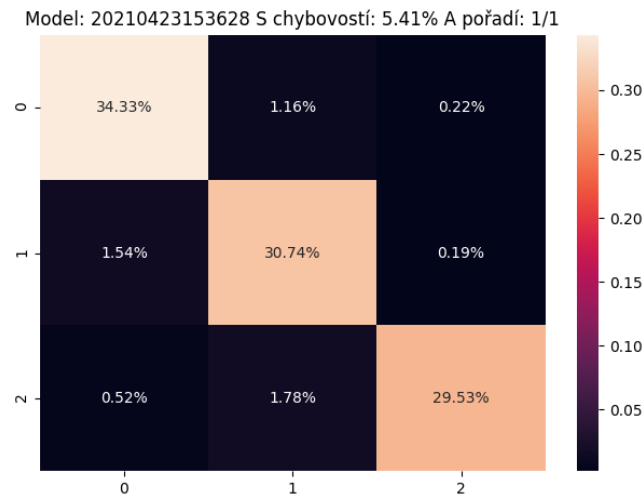
Konvoluční neuronové sítě se nejčastěji využívají pro analýzu obrázků, jak bylo řečeno v sekci 3.5. CNN lze použít i pro rozpoznávání EEG signálu. Proces konvoluce, který je v těchto neuronových sítích využíván zvýrazní hrany v signálu. Při následné analýze je tedy jednodušší najít místa, kde došlo k výrazné mozkové aktivitě. Architektury konvoluční neuronové sítě byly testovány podobně jako předchozí architektury. Velikost neuronové sítě hraje stejnou roli jako u RNN, proto není nutné jednotlivé výsledky prezentovat a je představen si pouze nejlepší výsledek.

Výsledek, který byl získán testováním různých parametrů konvoluční neuronové sítě je uveden na obrázku 4.12. Chybová matice má lepší rozložení než předchozí matice, která reprezentovala RNN, ale stále nedosahuje úrovně, která by byla dostatečně přesná pro ovládání virtuálního robota. Proto následující část popisuje jak byla testována kombinace CNN a RNN.

Propojení konvoluční neuronové sítě a rekurentní neuronové sítě

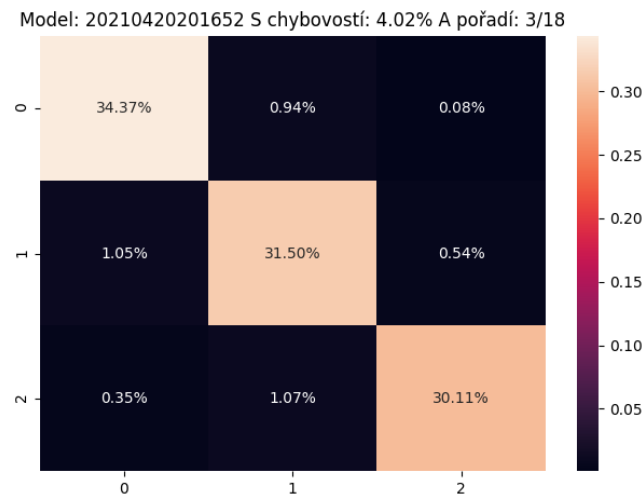
Když se rozdělí neuronové sítě do různých kategorií, tak se nabízí otázka, proč je nezkombinovat a nevyužít výhod obou modifikací neuronové sítě. Testování této otázky se věnuje následující část.

Podobně jako v předchozích částech byly testovány různé velikosti neuronové sítě. Výsledky byly stejné jako v předchozích částech, proto nebudou zmíněny. Z různých architektur neuronové sítě byla vybrána architektura s následující chybovou maticí 4.13. Matice má nej-



Obrázek 4.12: Chybová matice konvoluční neuronové sítě

nižší neúspěšnost ze všech zde uvedených architektur (4,02 %). Takto nízká neúspěšnost značně zvyšuje schopnost neuronové sítě správně analyzovat data.



Obrázek 4.13: Chybová matice kombinované rekurentní a konvoluční neuronové sítě

Zvolená architektura je ukázána na obrázku 4.1. Nejdříve bylo aplikováno několik konvolučních vrstev, které jsou následovány pooling vrstvou. Cílem těchto vrstev je snížit počet kanálů na minimum a zjednodušit následnou klasifikaci. Následuje LSTM, která slouží jako paměť pro neuronovou síť. Nakonec byly dvě plně propojené vrstvy, kde jedna slouží k analýze a druhá určí finální pravděpodobnosti.

Shrnutí

Využití neuronových sítí v oblasti analýzy EEG signálu je nový fenomén. Z tohoto důvodu nebylo provedeno velké množství studií a je tedy nutné provést testování pro výběr nejvhodnější architektury neuronové sítě. V této sekci byly představeny provedené experi-

Vrstvy sítě	Počet filtrů	Velikost filtrů	Posun	Výstup
Vstupní data	-	-		$207413 \times 16 \times 60$
Konvoluce (ReLU)	256	3	1	$207413 \times 14 \times 256$
Konvoluce (ReLU)	256	3	1	$207413 \times 12 \times 256$
Max pooling	-	2	2	$207413 \times 6 \times 256$
LSTM (Tanh)	-	-	-	207413×256
Dense	-	-	-	207413×256
Dense (Softmax)	-	-	-	207413×3

Tabulka 4.1: Architektura kombinované sítě CNN a RNN. Sít se skládá z několika konvolučních vrstev, které zvýrazní vzruchy a vrstvy LSTM, která pomáhá v následné vyhodnocování

menty. Z experimentů bylo patrné, že nejvhodnější architektura je kombinace CNN a RNN, která využívá výhod těchto modifikací neuronových sítí. Experimenty také ukázaly, že bylo lepší vytvářet menší neuronové sítě, které projdou menším počtem epoch. Menší neuronové sítě dosahovaly vyšší úspěšnosti a bylo možné strávit méně času trénováním neuronové sítě. Při trénování zvolené architektury se dosáhlo úspěšnosti 95,98 %. Takto vytrénovaný model je již možné využít při ovládání virtuálního robota.

Vytvoření prostředí pro testování bylo složité a proces tvorby je představen v následující sekci.

4.5 Testovací aplikace

Druhou částí implementace bylo vytvořit testovací prostředí, ve kterém bude možné vyzkoušet funkčnost vytvořeného modelu neuronové sítě z předchozí části. Pro prostředí má formu hry, kde uživatel s pomocí EEG signálů ovládá čtyřkolového robota s ramenem. V prostředí je také umístěna malá nádržka, kterou je možné uchopit a přenést ji na jiné místo. Tvorba hry byla rozdělena do dvou částí - tvorba modelu robota a tvorba herního prostředí.

K vymodelování robota byl využit program Blender¹⁰. Blender je opensource nástroj určený k tvorbě 3D modelů, animaci filmů nebo tvorby fotorealistických scénérií. Při tvorbě modelu je možné využít velké množství nástrojů, mezi které patří různé druhy osvětlení, vyhlazování objektu, zvětšování počtu ploch, zrcadlení a další.

Pro tvorbu herního prostředí a vytvoření skriptů, které ovládají robota byl zvolen herní engine Unity¹¹. Unity obsahuje nástroje pro snadné importování modelů. Unity také obsahuje velké množství knihoven, které lze využít pro skriptování. Hlavním programovacím jazykem využitým v Unity je C#. Programátor si může do Unity importovat vlastní modely, nebo má také možnost využít Unity asset store. Což je online obchod, kde mohou uživatelé prodávat své modely (assety), které navrhly. Cena assetů se může značně lišit. Především

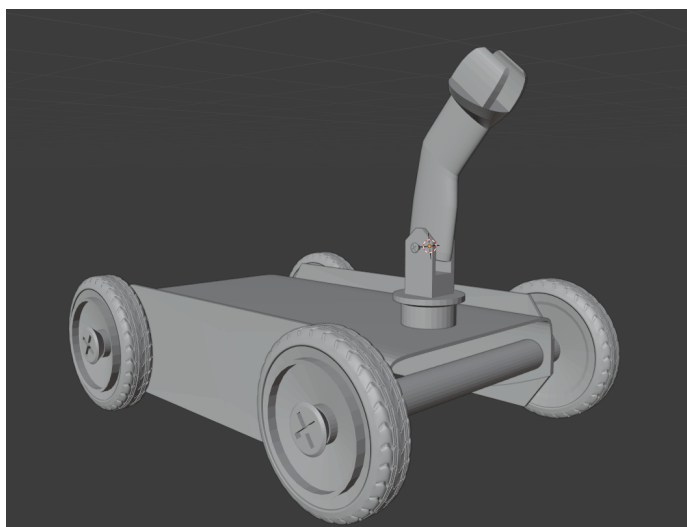
¹⁰<https://www.blender.org/>

¹¹<https://unity.com/>

díky Asset storu, je Unity vhodný pro začínající herní vývojáře, protože je možné získat nástroje, které by si musel vývojář vytvářet u jiných enginů sám.

Tvorba modelu

Při tvorbě modelu se vycházelo z robotů využívaných při neutralizování bomb. Model využitý pro demonstraci nepotřebuje mít velké množství detailů, proto některé části robota byly zjednodušeny a upraveny tak, aby byly více přizpůsobené demonstračním účelům. Výsledný model je ukázaný na obrázku 4.14. Robot není obarven, protože Unity nabízí dostatečné nástroje k vytvoření jednoduché barvy. Aplikování barev a materiálů v Blenderu by mohlo zkomplikovat následný import do Unity.



Obrázek 4.14: Výsledný model robota

Model má vytvarovanou základnu, do které jsou napojeny dvě osy. Do os byla umístěna kola, které se mohou pohybovat nezávisle na sobě pokud je třeba. Každé kolo má navíc v sobě zabudovaný šroub. Šroub zajišťuje rotaci do stran. Ve střední části základny je umístěné pohyblivé rameno. Po vytvoření modelu jej bylo třeba připravit na exportování do Unity. Soubor s modelem byl exportován do formátu fbx, který je podporován Unity a při přípravě tohoto exportu bylo třeba zvolit vhodné transformace, které budou na modelu provedeny při generování exportu. Orientace os je v Blenderu jiná než v Unity. Bylo třeba změnit například rotaci osy Y. Pokud by se rotace neupravily došlo by k převrácení modelu v Unity, což může přinášet další komplikace při programování.

Tvorba hry

Tvorbu testovacího prostředí v Unity lze rozdělit do tří základních částí:

- tvorba menu
- návrh prostředí
- programování skriptů pro pohyb a interakci s objektem

Tvorba menu

Při startu hry je uživateli prezentováno menu s možností hru zapnout nebo ukončit (viz obrázek 4.15). Po stisknutí tlačítka **START**, je hráč umístěn do testovacího prostředí. Do hlavního menu se může uživatel vrátit kdykoliv stisknutím tlačítka escape na klávesnici.



Obrázek 4.15: Herní menu prezentované uživateli

Herní menu pracuje s konceptem scén v Unity. Scény jsou jednotlivá prostředí do kterých lze rozmístit modely a vytvářet unikátní oblasti. Jednotlivé scény lze duplikovat. V práci byly vygenerovány dvě scény, jedna určená pro menu a druhá pro samotnou hru. Následně byly scény načteny pomocí **Scene manager**, který jim přiřadil index. V kódu je možné následně tyto scény měnit zavoláním metody **LoadScene**, kde je jako argument předán index dané scény.

Návrh prostředí

Při návrhu prostředí bylo bráno v potaz, aby uživatel měl dostatek prostoru pro ovládání robota, zároveň testovací plocha nemohla být moc velká, protože by to vedlo na komplikace při tvorbě. Robot byl umístěn na silnici, která je ohraničená domy a zátarasy. Proti robotovi je umístěn kyblík, který může uživatel pomocí virtuálního ramene zvednout. Hotové prostředí je ukázáno na obrázku 4.16. Model robota byl vytvořen v Blenderu a byl představen v předchozí sekci. Ostatní modely byly převzaty z Asset store. Jedná se o: Conifers [BOTD]¹² (modely stromů), Snaps Prototype | Asian Residential¹³ (modely budov, automobilů a další dekorace) a Road Architect¹⁴ (model silnice a zátarasů).

¹²<https://assetstore.unity.com/packages/3d/vegetation/trees/conifers-botd-142076>

¹³<https://assetstore.unity.com/packages/3d/environments/urban/snaps-prototype-asian-residential-143457>

¹⁴<https://github.com/MicroGSD/RoadArchitect>



Obrázek 4.16: Herní prostředí

Programování skriptů pro pohyb a interakci s objektem

Předtím než bylo možné programovat skripty, které budou řídit pohyb a chování robota, bylo nutné přiřadit modelu určité vlastnosti s pomocí kterých bude interagovat s okolním prostředím. Robotu je nejdříve přidán atribut `RigidBody`, který přidává vlastnosti jako je váha, odpor objektu a podobně. Dalším atributem jsou takzvané `Collider`. `Collider` zajišťují, že objekty budou mít oblast která je neprostupná. Pro správné fungování model potřebuje `Collider` pro tělo a pro každé kolo zvlášť. U kol jsou `Collider` velice důležité, protože budou využity ve skriptech pro rozpohybování robota.

Pohyb robota byl implementován ve skriptu `CarMovement`. Zde je aktualizována hodnota `MotorTorque` jednotlivých `Collider`ů. Následně je model kol rotován podle posunu, který kola provedly. Úchop je implementován ve skriptu `GrabMovement`. Do modelu byl přidán prázdný objekt, který je umístěn ve středu ramene. Když se robot přiblíží ke kyblíku a uživatel zmáčkne přiřazenou klávesu (viz níže) dojde k rotaci ramene a přemístění kbelíku na pozici prázdného objektu. Navíc je kbelíku odstraněna gravitace. Stisknutím stejné klávesy se rameno natočí nahoru a kbelík upustí.

Ovládání a propojení se systémem BCI

Pohyb robota je řízen klávesami W, A, S, D, úchop je pomocí klávesy E. V práci byla snaha oddělit analýzu od testovací aplikace, protože testovací aplikace je určena k potvrzení funkčnosti a nepředpokládá se, že by byla využita v navazujících pracích.

Samotné ovládání s pomocí OpenBCI je řízeno s pomocí skriptů, které byly představeny v sekci 4.5. Tyto skripty jsou rozšířené o funkci `controlByModel`, která získává data z OpenBCI GUI, ty předává neuronové síti, která vytváří predikce a tyto predikce jsou následně analyzovány.

Při analýze se rozhoduje, podle toho kolikrát určila neuronová síť konkrétní pohyb. Stisk kláves byl implementován za pomoci knihovny `pynput`¹⁵. `Pynput` nabízí rozhraní pro

¹⁵<https://pypi.org/project/pynput/>

stisk virtuální klávesnice, na kterou testovací aplikace reaguje stejně jako na stisk fyzických kláves. Pynput také nabízí možnost kontroly, zda jsou klávesy stisklé a možnost dlouhého stisku, který je uvolněn až zavoláním metody `Release`. S pomocí této knihovny je tedy možné ovládat testovací aplikaci bez nutnosti složitěho napojování.

Shrnutí

V rámci práce byla vypracována testovací aplikace, na které je možné ověřit funkčnost analýzy EEG signálu. Aplikace obsahuje dostatek prostoru pro základní testování pohybu a úchopu pomocí jednoduchého robota. V rámci tvorby prostředí byl vytvořen jak vlastní model robota, tak byly využity veřejně dostupné assety s pomocí kterých bylo vytvořeno testovací prostředí, kde byl robot umístěn. Robot byl vytvořen v Blenderu, při jeho tvorbě byly využity různé nástroje, které umožnily tvorbu složitějšího modelu. Aplikaci lze ovládat nezávisle na EEG analyzátoru. Jednotlivé příkazy z EEG jsou předávány pomocí knihovny pynput. Toto rozdělení bylo implementováno, protože se předpokládalo, že navazující práce nebudou potřebovat tuto testovací aplikaci, ale budou chtít primárně využít analýzu EEG.

4.6 Testování systému

Systém byl testován v průběhu několika sezení. Testování probíhalo ve třech fázích. Testování analýzy, aplikace a spojení obou částí systému. Praktické testování systému je složité. Systém by muselo otestovat více lidí a následně by museli vyhodnotit své dojmy v dotazníku. Testování tohoto rozsahu nebylo možné, proto je testování zaměřeno na ověření funkčnosti jednotlivých částí systému a základní testování funkčnosti celého systému.

Testování analýzy pomocí neuronových sítí

Při testování analýzy docházelo k predikcím na dosud nepoužitých datech. Výsledek těchto predikcí byl poté porovnán s reálnými hodnotami pomocí funkce `confusion_matrix`, která je součástí knihovny `scikit-learn`¹⁶. Následně byla vygenerována grafická podoba matice chyb, která znázorňuje procentuální zastoupení správných a špatných odpovědí. Matice chyb jednotlivých neuronových sítí byly představeny v sekci 4.4. Z různých variant byla vybrána ta nejvhodnější architektura, která byla testována i praktickými testy, kdy bylo sledováno jak moc přesná je neuronová síť při provádění analýzy pohybu. Z mého pozorování vyplynulo, že neuronovou síť lze využít pro ovládání virtuálního robota.

Testování herního prostředí

U herního (testovacího) prostředí byla ověřována správná funkčnost praktickými pokusy, které byly provedeny bez použití analýzy EEG. V těchto testech jsem zjišťoval, zda ovládání je dobře nastavené a pohyb, který je ovládán klávesami W, A, S, D bude použitelný i pro nepřímé ovládání. Poté se testovalo, zda je prostředí vhodné pro systém BCI a zda bude možné jednoduše robota ovládat. Důležitým parametrem byla velikost plochy, po které se robot pohybuje. Po prvních testech jsem naznal, že oblast je malá a mohl by být problém robota otočit, proto jsem se rozhodl na základě testů plochu zvětšit. V původní verzi byly domy blízko u sebe a prostředí působilo stísněným dojmem. Ve finální verzi je mezi domy větší vzdálenost, což zvětšuje plochu po které se může uživatel pohybovat. Na základě testů

¹⁶<https://scikit-learn.org/stable/>

také byla upravena funkce uchopení, kdy je možné kbelík uchopit pouze pokud je robot blízko.

Testování celého systému

Celý systém byl testován během posledního sezení. Výběr pokusů je možné vidět v souboru `video.mp4`. Během sezení jsem prováděl řadu pokusů. Nejdříve jsem testoval pouze schopnost ovládat kola, následně rozjezd a složitější pohyb po testovacím prostoru. Ovládání modelu bylo v rámci testování uspokojivé. Predikce byly pro kontrolování kol poměrně přesné. Otáčení kol doleva mělo mnohem méně chybných predikcí, než otáčení kol doprava. Podobný trend byl sledován i u dalších činností. Pokud byla činnost prováděna levou rukou, predikce byly přesné a nestávalo se, že by neuronová síť udělala chybu. Při testování funkcí vázaných na pravou ruku docházelo více k chybám, kdy buď pohyb nebyl vůbec zaznamenán, nebo neuronová síť měla problém rozhodnout, která ruka pohyb udělala. Tento problém se snížil zvětšením datasetu, který byl několikrát na základě těchto testů rozšířen.

Bohužel není možné kvůli aktuální epidemiologické situaci tento problém kompletně odstranit, protože by vyžadoval sbírání dat od více dobrovolníků, což aktuální opatření neumožňují. I přesto se povedlo dosáhnout uspokojivých výsledků a bylo možné ovládat robota pomocí signálů EEG.

Kapitola 5

Závěr

Cílem této práce bylo vytvořit systém BCI, který se skládá z analyzátoru signálu EEG a demonstrační aplikace.

Práce se nejdříve věnuje teoretickému úvodu do problematiky systémů BCI a neuronových sítí.

V práci byl využit systém OpenBCI pro získávání dat. Tento systém využívá 16 suchých elektrod. Data z elektrod lze získat buď využitím OpenBCI GUI, které je vyvíjeno firmou OpenBCI, nebo využitím knihovny Brainflow. Brainflow v této práci nebyl zmíněn, protože ve finální verzi použit nebyl ale funkce, která získává data z elektrod je v rámci práce implementována a je určena pro navazující práce. Místo Brainflow jsem v práci využil OpenBCI GUI, které nabízí možnost přehrávat data, což v době zhoršené epidemiologické situace a omezených možností sezení bylo výhodné. OpenBCI GUI je také schopno vizualizovat data v přímém přenosu, což je užitečné pro rychlou analýzu a sníží se tím riziko problémů dále při vývoji. Toto rozhodnutí vedlo k tomu, že nebylo možné využít knihovnu Brainflow ve výsledné aplikaci a tím zajistit, že program bude jednoduchý. Data získaná knihovnou Brainflow jsou podobná datům z GUI, ale nejsou úplně stejná. Bylo by nutné vytvořit nové datasey, s pomocí kterých by se trénovala neuronová síť. Omezený čas, kdy bylo možné využívat OpenBCI, ale neumožnilo tyto datasey vytvořit, proto bylo využito GUI.

Při analýze signálu jsem došel k závěru, že nejvhodnější je použít motoricky evokované potenciály a kombinaci konvoluční a rekurentní neuronové sítě.

Při implementaci se ukázalo, že některé signály, které by měly být poměrně výrazné se vůbec nepromítaly do signálu získaných z elektrod OpenBCI, toto představovalo zásadní problém při vývoji, který by bylo vhodné detailněji prostudovat. To však nebylo možné kvůli aktuálním opatřením, proto byly více testovány alternativní metody vzniku signálu EEG.

Dále v rámci práce bylo implementováno testovací prostředí. Samotné prostředí bylo vyvíjeno v herním enginu Unity a model virtuálního robota byl vytvořen v programu Blender.

Výsledná aplikace je funkční a demonstruje možnosti systémů BCI. Dále byl vytvořen v rámci práce plakát, který ukazuje základní fungování aplikace a video které demonstruje funkčnost aplikace.

Aplikace byla vytvořena tak, aby bylo možné využít kteroukoliv její část pro další vývoj. Dá se předpokládat, že v pozdějších fázích vývoje nebude potřeba demonstrační aplikace. Z tohoto důvodu byla aplikace rozdělena na analýzu EEG signálu a samotnou demonstrační aplikaci, která je ovládána generováním virtuálních stisků klávesnice. Tímto způsobem je

možné ovládat i jiné aplikace, takže napojení na další systémy by mělo být značně zjednodušeno a aplikace je dobrý základ pro budoucí vývoj.

Aplikaci lze dále rozšířit o analýzu dalších forem pohybu (například pohyb nohou). Hlavním cílem dalších prací by mělo být využití Brainflow pro získávání signálu EEG a využití pohybové představivosti, která je možnou alternativou k pohybově evokovaným potenciálům. Pokud by byla využita knihovna Brainflow, tak by bylo možné využít aplikaci pro ovládání kompenzační pomůcek jako je například invalidní vozík případně ovládání umělých končetin. Pro tyto účely je vhodné aplikaci rozšířit o analýzu signálu EMG, který je vhodný pro přesnější analýzu v kombinaci se signálem EEG. Aplikaci lze, ale také dále rozvíjet za využití již implementované analýzy, kde data jsou získána s pomocí OpenBCI GUI. Aplikaci je možné rozšířit tak, aby ovládala základní funkce počítače, což by bylo vhodné řešení pro osoby se zhoršenou schopností pohybu.

Literatura

- [1] AGGARWAL, S. a CHUGH, N. Signal processing techniques for motor imagery brain computer interface: A review. *Array*. 1. vyd. 2019, 1-2, č. 1, s. 100003. DOI: <https://doi.org/10.1016/j.array.2019.100003>. ISSN 2590-0056. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S2590005619300037>.
- [2] AHN, J. W., KU, Y. a KIM, H. C. *A Novel Wearable EEG and ECG Recording System for Stress Assessment*. Multidisciplinary Digital Publishing Institute, Apr 2019. Dostupné z: <https://www.mdpi.com/1424-8220/19/9/1991/html>.
- [3] AL FAHOUM, A. S. a AL FRAIHAT, A. A. Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains. *ISRN Neuroscience*. 1. vyd. Hindawi Publishing Corporation. Feb 2014, sv. 2014, č. 1, s. 730218. DOI: 10.1155/2014/730218. Dostupné z: <https://doi.org/10.1155/2014/730218>.
- [4] ALER, R., GALVÁN, I. M. a VALLS, J. M. Applying evolution strategies to preprocessing EEG signals for brain–computer interfaces. *Information Sciences*. 1. vyd. 2012, sv. 215, č. 2, s. 53–66. DOI: <https://doi.org/10.1016/j.ins.2012.05.012>. ISSN 0020-0255. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0020025512003568>.
- [5] ATHAIYA, A. Activation function in neural networks. *International Journal of Engineering Applied Sciences and Technology*. 2. vyd. April 2020, č. 4, s. 310–316. DOI: 10.33564/ijeast. ISSN 2455-2143. Dostupné z: <https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf>.
- [6] BISHOP, C. M. Neural networks and their applications. *Review of Scientific Instruments*. 1. vyd. 1994, sv. 65, č. 6, s. 1803–1832. DOI: 10.1063/1.1144830.
- [7] BUZSÁKI, G., ANASTASSIOU, C. a KOCH, C. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature reviews. Neuroscience*. 1. vyd. Květen 2012, sv. 13, č. 1, s. 407–20. DOI: 10.1038/nrn3241.
- [8] CARLSTEDT, R. A. a BALCONI, M. *Handbook of sport neuroscience and psychophysiology*. 1. vyd. Routledge, 2019. ISBN 9781138852181.
- [9] CARVALHAES, C. a DE BARROS, J. A. The surface Laplacian technique in EEG: Theory and methods. *International Journal of Psychophysiology*. 1. vyd. 2015, sv. 97, č. 3, s. 174–188. DOI: <https://doi.org/10.1016/j.ijpsycho.2015.04.023>. ISSN 0167-8760. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0167876015001749>.

- [10] COMMONS, W. *File:2D Convolution Animation.gif* — *Wikimedia Commons, the free media repository*. 2020. Dostupné z:
https://commons.wikimedia.org/w/index.php?title=File:2D_Convolution_Animation.gif&oldid=487222533.
- [11] CRAIK, A., HE, Y. a CONTRERAS VIDAL, J. L. Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering*. 1. vyd. 2019, sv. 16, č. 3, s. 1–14. DOI: 10.1088/1741-2552/ab0ab5.
- [12] CUNNINGTON, R. Movement-related potentials in Parkinsons disease. Motor imagery and movement preparation. *Brain*. 1. vyd. 1997, sv. 120, č. 8, s. 1339–1353. DOI: 10.1093/brain/120.8.1339.
- [13] GAUTAM, A. Nerve Cells. In: VONK, ed. *Encyclopedia of Animal Cognition and Behavior*. Leden 2017, s. 1–3. DOI: 10.1007/978-3-319-47829-6_1282-1. ISBN 978-3-319-55064-0.
- [14] GURNEY, K. *An Introduction to Neural Networks*. 1. vyd. CRC Press, 2014. ISBN 0-203-45151-1.
- [15] HOCHREITER, S. a SCHMIDHUBER, J. Long Short-Term Memory. *Neural Computation*. 1. vyd. 1997, sv. 9, č. 8, s. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- [16] KAYA, I., BOHÓRQUEZ, J. a ÖZDAMAR. A BCI Gaze Sensing Method Using Low Jitter Code Modulated VEP. *Sensors*. 1. vyd. 2019, sv. 19, č. 17, s. 3797. DOI: 10.3390/s19173797.
- [17] KIM, S.-J., MANYAM, S. C., WARREN, D. J., NORMANN, R. A., GALASHAN, F. O. et al. *FIGURE 1. The Utah electrode array (UEA). It shows 100 microelectrodes...* Oct 2020. Dostupné z:
https://www.researchgate.net/figure/The-Utah-electrode-array-UEA-It-shows-100-microelectrodes-in-a-10-by-10-configuration_fig1_7281464.
- [18] KOLEKTIV, M. *Supervised Learning*. Prosinec 2020. Dostupné z:
<https://www.mathworks.com/discovery/supervised-learning.html>.
- [19] KOLEKTIV swissgrid. *Frequency*. 2021. Dostupné z:
<https://www.swissgrid.ch/en/home/operation/regulation/frequency.html#:~:text=ThestandardfrequencyinEurope's,of60Hertzisused>.
- [20] KOLEKTIV wiki. *10–20 system (EEG)*. Wikimedia Foundation, 14. Jan 2021. Dostupné z:
[https://en.wikipedia.org/wiki/10\T1\textendash20_system_\(EEG\)#/media/File:21_electrodes_of_International_10-20_system_for_EEG.svg](https://en.wikipedia.org/wiki/10\T1\textendash20_system_(EEG)#/media/File:21_electrodes_of_International_10-20_system_for_EEG.svg).
- [21] KOSEC, D. *Jak funguje mozek?* Jan 2020. Dostupné z:
<https://codeoflife.cz/telo/jak-funguje-mozek/>.
- [22] KÜBLER, A. The history of BCI: From a vision for the future to real support for personhood in people with locked-in syndrome. *Neuroethics*. 1. vyd. Jul 2020, sv. 13, č. 2, s. 163–180. DOI: 10.1007/s12152-019-09409-4. ISSN 1874-5504. Dostupné z:
<https://doi.org/10.1007/s12152-019-09409-4>.

- [23] KUMAR, N., ALAM, K. a SIDDIQI, A. H. Wavelet Transform for Classification of EEG Signal using SVM and ANN. *Biomedical and Pharmacology Journal*. 1. vyd. 2017, sv. 10, č. 4, s. 2061–2069. DOI: 10.13005/bpj/1328.
- [24] LEIF, S. a LAGUNA, P. *Bioelectrical signal processing in cardiac and neurological applications*. 1. vyd. Elsevier Academic Press, 2006. ISBN 978-0-12-437552-9.
- [25] LI, G., LEE, C. H., JUNG, J. J., YOUN, Y. C. a CAMACHO, D. Deep learning for EEG data analytics: A survey. *Concurrency and Computation: Practice and Experience*. 3. vyd. 2019, sv. 32, č. 18. DOI: 10.1002/cpe.5199.
- [26] LOTTE, F. A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain-Computer Interfaces. In: MIRANDA, E. R. a CASTET, J., ed. *Guide to Brain-Computer Music Interfacing*. Springer, 2014. ISBN 1447165845. Dostupné z: <https://hal.inria.fr/hal-01055103>.
- [27] MACHADO, J., BALBINOT, A. a SCHUCK, A. A study of the Naive Bayes classifier for analyzing imaginary movement EEG signals using the Periodogram as spectral estimator. In: MACHADO, J., ed. *2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*. Institute of Electrical and Electronics Engineers (IEEE), 2013, s. 1–4. DOI: 10.1109/BRC.2013.6487514. ISBN 9781467330244.
- [28] MANASCO, M. H. *INTRODUCTION TO NEUROGENIC COMMUNICATION DISORDERS*. 3. vyd. World Headquarters, 2014. ISBN 9780323290920.
- [29] MEHROTRA, S. C., ABHANG, P. A. a GAWALI, B. W. *Introduction to EEG- and Speech-Based Emotion Recognition*. 1. vyd. Academic Press, 2016. ISBN 978-0-12-804490-2.
- [30] NASTESKI, V. An overview of the supervised machine learning methods. *Horizons.b*. 1. vyd. 2017, sv. 4, č. 1, s. 51–62. DOI: 10.20544/horizons.b.04.1.17.p05.
- [31] NOSEK, J. *Scientific Contributions & Legacy*. 2021. Dostupné z: <https://wilderpenfieldassignment.weebly.com/scientific-contributions--legacy.html>.
- [32] NUNEZ, P., SILBERSTEIN, R., CADUSCH, P., WIJESINGHE, R., WESTDORP, A. et al. A theoretical and experimental study of high resolution EEG based on surface Laplacians and cortical imaging. *Electroencephalography and Clinical Neurophysiology*. 1. vyd. 1994, sv. 90, č. 1, s. 40–57. DOI: [https://doi.org/10.1016/0013-4694\(94\)90112-0](https://doi.org/10.1016/0013-4694(94)90112-0). ISSN 0013-4694. Dostupné z: <https://www.sciencedirect.com/science/article/pii/0013469494901120>.
- [33] PHILIPP, G., SONG, D. a CARBONELL, J. The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions. *ArXiv: Learning*. 1. vyd. 2017, č. 1.
- [34] RYNNÄNEN, O., HYTTINEN, J. a MALMIVUO, J. Study on the spatial resolution of EEG - Effect of electrode density and measurement noise. In: MALMIVUO, J., ed. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*. Institute of Electrical and Electronics Engineers (IEEE

-), Září 2004, sv. 6, s. 4409–12. DOI: 10.1109/IEMBS.2004.1404226. ISBN 0-7803-0785-2.
- [35] SAMMUT, C. a WEBB, G. I. *Encyclopedia of machine learning and data mining*. 1. vyd. Springer, 2017. ISBN 978-1-4899-7686-4.
- [36] SHARMA, S. a SHARMA, S. International Journal of Engineering Applied Sciences and Technology. *Activation functions in neural networks*. 1. vyd. 2020, sv. 4, č. 12. DOI: 10.33564/ijeast.
- [37] STAHL, S. M. *Stahls essential psychopharmacology: neuroscientific basis and practical applications*. 4. vyd. Cambridge University, 2013. ISBN 1107686466.
- [38] SUTTER, E. E. The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications*. 1. vyd. 1992, sv. 15, č. 1, s. 31–45. DOI: 10.1016/0745-7138(92)90045-7.
- [39] TRIGUI, O., ZOUCHE, W. a MESSAOUD, M. B. A comparison study of SSVEP detection methods using the Emotiv Epoc headset. *2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. 1. vyd. 2015, č. 1. DOI: 10.1109/sta.2015.7505108.
- [40] VAZ, F., DE OLIVEIRA, P. G. a PRINCIPE, J. A study on the best order for autoregressive EEG modelling. *International Journal of Bio-Medical Computing*. 2. vyd. 1987, sv. 20, č. 1, s. 41–50. DOI: [https://doi.org/10.1016/0020-7101\(87\)90013-4](https://doi.org/10.1016/0020-7101(87)90013-4). ISSN 0020-7101. Dostupné z: <https://www.sciencedirect.com/science/article/pii/0020710187900134>.
- [41] WANG, L. *Support vector machines: theory and applications*. 1. vyd. Springer, 2005. ISBN 3540243887.
- [42] WEIERGRÄBER, M., PAPAZOGLU, A., BROICH, K. a MÜLLER, R. Sampling rate, signal bandwidth and related pitfalls in EEG analysis. *Journal of Neuroscience Methods*. 1. vyd. 2016, sv. 268, č. 1, s. 53–55. DOI: <https://doi.org/10.1016/j.jneumeth.2016.05.010>. ISSN 0165-0270. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0165027016300887>.
- [43] WOLPAW, J. R. *Brain-computer interfaces: principles and practice*. 1. vyd. Oxford University Press, 2012. ISBN 978-0-19-538885-5.
- [44] WOLPAW, J. R., MILLÁN, J. del a RAMSEY, N. F. Brain-computer interfaces: Definitions and principles. In: RAMSEY, N. F., ed. *Brain-Computer Interfaces*. Elsevier, 2020, s. 15–23. DOI: 10.1016/b978-0-444-63934-9.00002-0. ISBN 978-0-444-63934-9. Dostupné z: <https://doi.org/10.1016/b978-0-444-63934-9.00002-0>.
- [45] XU, S., WANG, Z., SUN, J., ZHANG, Z., WU, Z. et al. Using a deep recurrent neural network with EEG signal to detect Parkinson's disease. *Annals of Translational Medicine*. 1. vyd. 2020, sv. 8, č. 14, s. 874–874. DOI: 10.21037/atm-20-5100.
- [46] YAMASHITA, R., NISHIO, M., DO, R. K. G. a TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*. 1. vyd. 2018, sv. 9, č. 4, s. 611–629. DOI: 10.1007/s13244-018-0639-9.


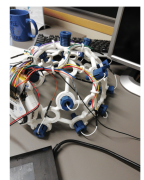
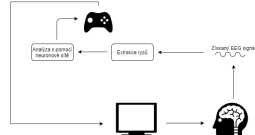

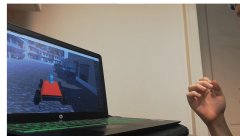
- [47] ZHANG, X., CHEN, F. a HUANG, R. A Combination of RNN and CNN for Attention-based Relation Classification. *Procedia Computer Science*. 1. vyd. 2018, sv. 131, č. 1, s. 911–917. DOI: <https://doi.org/10.1016/j.procs.2018.04.221>. ISSN 1877-0509. Recent Advancement in Information and Communication Technology:. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S187705091830601X>.

Příloha A

Plakát

Ovládání virtuálního robota pomocí EEG

Autor: Michal Drla
Vedoucí: Ing. Jan Tinka
xdrlam00@stud.fit.vutbr.cz

Motivace <ul style="list-style-type: none">• Lidé s omezením pohybu nemusí být schopni ovládat kompenzační pomůcky.• Technologie BCI nabízí možnosti nepřímého ovládní, pomocí signálů z mozku.• Je možné ovládat například invalidní vozíky.• Využití neuronových sítí v systémech BCI není ještě plně prozkoumáno. 	Použité technologie <ul style="list-style-type: none">• Tensorflow• OpenBCI• Unity• Blender 
Navrhnutý systém <ul style="list-style-type: none">• Systém je rozdělený na dvě části: analyzátor a testovací hra.• Rozdělení systému zajišťuje jednoduché využití analyzátoru v jiných systémech.• Pro získávání EEG signálu je využito OpenBCI GUI. 	Herní prostředí <ul style="list-style-type: none">• Robot se dokáže pohybovat dopředu, dozadu, zatáčet a zvedat kyblík.• Herní prostředí bylo vytvořeno v Unity.• Robot se ovládá klávesami W, A, S, D a E. 
Ovládání systému <ul style="list-style-type: none">• Systém se ovládá pohybem ruky.• Pohyb ruky lze nahradit polybovou představivostí.• Dvě mávnutí levou rukou rozpohybují robota, dvě mávnutí pravou rukou zvednou kyblík.• Mávnutí levou rukou provede zatočení doleva, mávnutí pravou rukou doprava.• Provedení má zpoždění 5 sekund, aby bylo možné provádět více příkazů. 	



Obrázek A.1: Plakát ukazující BCI systém

Příloha B

Obsah přiloženého paměťového média

- *example_data/* - složka s ukázkovým souborem s daty určenými pro OpenBCI GUI
- *game/* - složka s testovací hrou, která byla využita pro testování analýzy signálu EEG
- *poster/* - složka s plakátem představujícím základní informace o práci
- *video/* - video demonstrující funkčnost práce ve třech experimentech
- *src/* - složka se zdrojovými soubory
- *README.md* - soubor obsahující informace o struktuře adresáře
- *models/* - složka obsahující natrénovaný model
- *src_latex/* - složka se zdrojovými soubory práce v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u