



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název: Řízení robotů myšlenkou na bázi EEG signálu
Student: Michal Veljačik
Vedoucí: Ing. Miroslav Skrbek, Ph.D.
Studijní program: Informatika
Studijní obor: Znalostní inženýrství
Katedra: Katedra aplikované matematiky
Platnost zadání: Do konce letního semestru 2019/20

Pokyny pro vypracování

Seznamte se s EEG senzorem a knihovnami pro přenos dat ze senzoru. Systematicky nasnímejte dostatečné množství EEG signálů od uživatelů, kteří budou myslet na různé věci nebo budou upírat svoji pozornost různým způsobem. EEG data vhodně předzpracujte a podrobte je analýze například metodami shlukové analýzy nebo neuronovými sítěmi. Na základě analýzy sestrojte klasifikátor, který bude možné v reálném čase naučit rozlišovat různé mozkové aktivity konkrétního uživatele. Přesnost klasifikace vyhodnoťte. Klasifikátor navažte na minidron nebo jiného robota a realizujte jeho řízení myšlenkou. Rozsah práce upřesněte po dohodě s vedoucím práce.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Karel Klouda, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 12. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Řízení robotů myšlenkou na bázi EEG signálu

Michal Veljačik

Katedra Aplikované Matematiky
Vedoucí práce: Ing. Miroslav Škrbek, PhD

16. května 2019

Poděkování

Chtěl bych poděkovat vedoucímu práce za to, že vypsál takto inspirativní téma a za celkovou podporu při realizaci práce. Dále bych chtěl poděkovat všem uživatelům, kteří si ode mě nechali změřit mozkové vlny, a umožnili tak vznik této práce.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Michal Veljačik. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Veljačik, Michal. *Řízení robotů myšlenkou na bázi EEG signálu*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

V rámci této práce bylo vytvořeno experimentální BCI na bázi jednonárodního EEG signálu. Hlavním přínosem práce je implementace aplikace umožňující nahrávat a experimentovat se stopami mentálních úkolů a testovat tak jejich užitečnost pro nasazení v BCI. Zároveň bylo implementováno experimentální ovládání drona myšlenkou. Bylo zjištěno, že přesnost klasifikace EEG vln s časem výrazně degraduje.

Klíčová slova BCI, EEG, mozkové vlny, Python, strojové učení, klasifikace, SVM, extrakce příznaků, selekce příznaků

Abstract

In this thesis, an experimental single-channel EEG based BCI is implemented. Main contribution of this work is a Python application capable of recording and experimenting on EEG mental tasks. An experimental BCI for drone control by EEG signal was implemented and connected with the app. It was also discovered that with time the classification accuracy decreases drastically.

Keywords BCI, EEG, brainwaves, Python, machine learning, classification, SVM, feature extraction, feature selection

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza a návrh	5
2.1 Analýza	5
2.1.1 Lehký úvod do neurovědy	5
2.1.2 Současné řešení problému	7
2.1.3 Mindwave Mobile 2	8
2.1.4 Parrot Mambo Mission	10
2.2 Návrh	11
2.2.1 Akvizice dat	12
2.2.2 Předzpracování dat	14
2.2.3 Analýza dat	16
2.2.4 Klasifikace	17
2.2.5 Návrh aplikace	18
3 Realizace	21
3.1 Implementace	21
3.1.1 Použití aplikace	22
3.1.2 Formáty dat	23
3.1.3 Implementace modulu core	24
3.1.4 Praktické nasazení BCI	25
3.2 Experimentální část	26
3.2.1 Akvizice dat	26
3.2.2 Realizace experimentů	26
3.2.3 Ex1: rozlišitelnost VEP	28
3.2.4 Ex2: rozlišitelnost MI	29
3.2.5 Ex3: kombinace úkolů	29
3.2.6 Výsledky experimentů	30

Závěr	31
Literatura	33
A Seznam použitých zkratk	35
B Obsah přiložené SD karty	37

Seznam obrázků

2.1	10–20 Systém	8
2.2	Schema headsetu Mindwave	9
2.3	Parrot Mambo Mission	11
2.4	Tvar 2s EEG signálu různých mentálních úkolů.	13
2.5	Konstrukce nadroviny SVM	19

Seznam tabulek

2.1	Základní EEG rytmy	6
3.1	Přehled datasetů	27
3.2	Výsledky Ex1	27

Úvod

Nápad napojit fungování nějakého zařízení přímo na myšlenky uživatele není nikterak nový, technologie umožňující převést jej do praxe jsou ovšem lidstvu známy teprve necelé jedno století. První základy pro BMI (tedy Brain Machine Interface) byly položeny s objevením Elektroencefalografie, tedy metody pro snímání elektrické aktivity v mozku pomocí elektrod. Technika EEG byla zpočátku velmi primitivní: spočívala v umístění stříbrného plíšku pod kůži na hlavě uživatele a záznamu elektrické aktivity na pásku. Pomocí této základní techniky objevil v 20. letech 20. století Hans Berger takzvané *alpha* vlny, tedy specifický tvar vlny s frekvencí cca 12 Hz, který se objevil, pokud uživatel zavřel oči, a zase zmizel, pokud oči otevřel. Tímto bylo dokázáno jednak to, že se elektrická aktivita v mozku mění na základě vnějších okolností, a zároveň to, že je možné zvnějšku tuto aktivitu poměrně spolehlivě změřit.

V následujících letech EEG se stala hlavním prostředkem pro studium mozku v rámci neurovědy. EEG je používána pro diagnostiku epilepsie nebo studium anomálií v neurologické aktivitě mozku bez nutnosti otevírat lebku pacienta. Ačkoli z dostupných měřících technik není tou nejpřesnější (například oproti fMRI nebo ECoG), jistě je ve své dnešní podobě tou nejpohodlnější vzhledem ke své neinvazivnosti a mobilitě a je tak přirozenou volbou pro aplikace BCI.

Prvním obecně známým pokusem o přímé napojení EEG vln na nějaké externí zařízení bylo vytvoření ovládání hudebního nástroje pomocí alpha vln uživatele *music for solo performer*. Aby zařízení začalo hrát, musely být u uživatele rozpoznány *alpha* vlny, čehož šlo nejlépe docílit pomocí zavření očí a různých pohybů rukou. První odborná práce na téma BCI — brain computer interface — byla zveřejněna v sedmdesátých letech s ambiciózním podtitulem „vzhůru k ovládání počítačů, robotů a vesmírných lodí pomocí EEG“. Problém BCI se dostal do středu zájmu velkého množství badatelů. Nejčastějšími aplikacemi byly pomůcky pro pacienty postižené ochrnutím, pro které může BCI představovat jediný kanál pro komunikaci s okolím. Běžně se také můžeme setkat se studii zabývajícími se zařízeními replikujícími ztracené končetiny

nebo různá brain-to-text rozhraní.

Jakkoli je tato oblast široce popsána, podmínky vhodné k širokému nasazení BCI na bázi EEG do praxe nastaly až poměrně nedávno s pokroky v oblasti strojového učení a rozmachem snadno dostupných EEG zařízení, z nichž lze zmínit například Mindwave Mobile, Ultracortex nebo Emotiv headset. Tato zařízení svou komplexitou sice zdaleka nedosahují přesnosti laboratorních podmínek, jsou ale pro uživatele pohodlněji použitelná.

Nejjednodušší z těchto zařízení obsahují pouze jednu elektrodu a poskytují tak pro aplikace BCI pouze jeden kanál dat, tedy výrazně méně, než je obvyklé u laboratorních EEG aparatur. Při širokém rozšíření BCI do praxe však bude mít výsledné zařízení pravděpodobně malý počet elektrod. Proto má smysl zkoumat použitelnost jednoduchých EEG zařízení pro použití v BCI. Jedno takové zařízení, Mindwave Mobile 2, je jako zdroj EEG signálu použito v této práci. Že lze sestrojít BCI na bázi jednoho kanálu již bylo prokázáno například v práci [1], kde byl sestrojen jednoduchý klasifikátor schopný rozpoznat jednu třídu mentální aktivity oproti klidovému stavu.

Ambicí této práce je zjistit, jak dobře je pomocí takového zařízení možné interpretovat mentální aktivitu uživatele a na bázi těchto zjištění implementovat ovládání minidronu pomocí myšlenky.

Cíl práce

Nastudovat existující knihovny pro připojení EEG headsetu a minidronu do počítače. Navrhnout příkazy pro ovládání robotů pomocí jednobanového EEG. Experimentálně ověřit jejich klasifikovatelnost skrze naměření dat od dostatečného množství uživatelů. Vytvořit aplikaci realizující BCI na bázi jednobanového EEG. Aplikaci propojit s headsetem a s minidronem tak, aby skrze ni bylo možné realizovat ovládání robota myšlenkou.

Analýza a návrh

2.1 Analýza

Teoretická část se skládá ze tří sekcí: v první sekci jsou uvedeny neurovědecké poznatky, na kterých jsou BCI na bázi EEG postaveny. Ve druhé sekci je představena analýza obecných postupů při implementacích BCI na bázi EEG. Ve třetí sekci je popsán hardware, pomocí kterého je realizována praktická a experimentální část této práce.

2.1.1 Lehký úvod do neurovědy

V rámci lehkého úvodu do neurovědy je nejprve stručně představena metoda EEG. Následně je představena souvislost tvaru mozkových vln, prováděných mentálních aktivit a umístění měřících elektrod na hlavě uživatele. Nakonec jsou zkoumány různé druhy mentálních úkolů, které se běžně používají v rámci implementací BCI na bázi EEG.

Měřitelným projevem vědomí jsou elektrické signály v mozku, pocházející od miliard nervových buněk, které si mezi sebou tímto způsobem předávají informace. Jednotlivé neurony vzhledem k jejich mikroskopickému rozměru z vnějšku studovat nedokážeme, co ovšem rozpoznávat dokážeme, je koordinovaná aktivita jejich větších shluků.

Nejrozšířenější metodou pro snímání elektrické aktivity mozku je elektroencefalografie (EEG). Při EEG se na povrch hlavy rozmístí větší množství měřících a menší množství referenčních elektrod a následně jsou zaznamenávány rozdíly elektrického potenciálu naměřené na měřících a referenčních elektrodách.

Zachycený signál má přibližný tvar vlny. Podoba této vlny se mění s umístěním elektrody, mentálním stavem uživatele a druhem mentální aktivity, kterou uživatel v danou chvíli provádí, chování vlny ale vykazuje jisté univerzální vlastnosti, díky kterým rozlišujeme několik druhů „mozkových vln“. Nejznámější a první popsanou „mozkovou vlnou“ je *alpha* vlna, která se objevuje v zadní části mozku v době, kdy má uživatel zavřené oči.

Tabulka 2.1: EEG rytmy

Název	Frekvence	Mozková centra	Mentální aktivity
<i>delta</i>	0,5-2 Hz	celý mozek	Ospalost, stav vědomí
<i>theta</i>	4-8 Hz	frontální a temporální	bdění, stres
<i>kappa</i>	8-12 Hz	frontální a temporální	řešení problému
<i>alpha</i>	8-12 Hz	okcipitální lalok	vizuální podněty
<i>beta</i>	12,5-30 Hz	motorický kortex	pohyb končetin
<i>gamma</i>	25-50 Hz	frontální oblast mozku	vyšší funkce vědomí

Vlny se od sebe rozlišují podle převládající frekvence a lokality, ve které jsou detekovatelné. Souvislost těchto „mozkových vln“ s konkrétními mozkovými centry a mozkovými aktivitami je velice dobře popsána a při designu BCI na bázi EEG s výhodou používána.

Signál zachycený pomocí EEG specifickým způsobem osciluje a přibližně se tak podobá vlně. Podle přibližné frekvence této vlny rozlišujeme několik druhů EEG rytmů (populárně zvaných „mozkové vlny“). Přehled vlastností některých EEG vln je uveden v tabulce 2.1. Uvedené vlastnosti EEG rytmů jsou popsány v práci [2].

- **Alpha** vlna byla prvním objeveným specifickým tvarem EEG vlny. S frekvencí 8–12 Hz se nejzřetelněji projevuje nad occipitálním lalokem (mozkové centrum zodpovědné za zpracování vizuálních informací) v zadní části mozku v čase, kdy má uživatel zavřené oči. Některé zdroje uvádějí, že *alpha* vlnu je možné při zavřených očích rozpoznat i v přední části mozku.
- **Beta** vlna se s frekvencí 12,5–30 Hz nachází ve frekvenčním pásmu těsně nad *alpha* vlnou. Detekovat ji lze především v centrální části mozku nad motorickým kortexem (mozkové centrum zodpovědné za plánování a realizaci volných pohybů).
- **Gamma** vlnu nalezneme na frekvenčním spektru mezi 25–100 Hz, její největší aktivitu lze pozorovat v okolí frekvence 40 Hz. *Gamma* vlna je detekovatelná zejména v předních částech mozku, tedy oblastech zodpovědných za vyšší funkce vědomí.
- **Delta** vlna má ze zde uváděných vln největší amplitudu a zároveň nejnižší frekvenci (přibližně 0,5–4 Hz). Zřetelně se projevuje ve fázích hlubokého spánku.

Současná technologie EEG neumožňuje rozpoznávat myšlenky uživatele, ale „jen“ způsoby upírání pozornosti. Konkrétní mentální aktivity aktivují související mozková centra, čímž se signál zachycený elektrodami EEG jistým

očekávatelným způsobem mění. Při realizaci BCI se používají dvě hlavní techniky designu těchto způsobů upírání pozornosti, nadále v textu nazývaných jako *mentální úkoly*.

Technika ***Evoked Potentials*** spočívá v interpretaci reakce na nějaký externí podnět. Tímto podnětem může být obrázek promítaný na obrazovce počítače, elektrošok, lehký dotek kůže nebo reakce na zvuk. Tento podnět má v mozkových vlnách uživatele vyvolat detekovatelnou změnu, obvykle v průběhu následujících 0,5–2 s. Sofistikovanější implementací tohoto systému je navázání BCI na zařízení sestávající z LED panelu, na kterém každá dioda bliká jinou frekvencí. Uživatel předává příkazy BCI pomocí soustředěného pohledu na diody. Díky rozdílné frekvenci blikání je pak možné dobře rozpoznat, kterou diodu uživatel právě pozoruje. Příklad realizace BCI na bázi VEP pomocí různě barevných LED diod nalezneme ve studii [3].

Technika ***Mental Imagery*** je naopak založena na tom, že si uživatel daný podnět představuje. Do této kategorie spadají různé varianty slovních asociací (představování si slova, artikulace slova, vizualizace předmětu) nebo pohybu končetin (což je nejspíš nejčastější používaným druhem mentálních úkolů při realizaci BCI). Zároveň sem můžeme zařadit různé vyšší funkce vědomí, například provádění matematických výpočtů. Úspěšné použití této techniky nalezneme například ve studiích [4], [5] a [6].

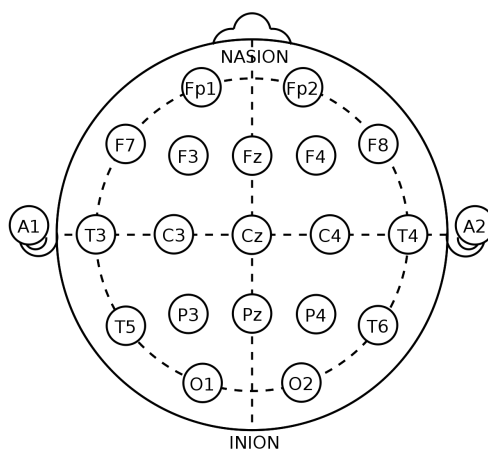
2.1.2 Současné řešení problému

V rámci této sekce jsou nejprve stručně popsány různé možné aplikace BCI, následně je představen obecný postup pro realizaci BCI na bázi EEG odvozený z literatury a nakonec jsou představeny na tento druh problému nejčastěji nasazované klasifikátory.

Nejčastějšími aplikacemi jsou komunikační pomůcky pro ochrnuté pacienty, protetické pomůcky nahrazující chybějící končetiny a různé video a VR hry.

Vzhledem k tomu, že se jedná o poměrně rozsáhle popsáný problém, lišící se pouze v parametrech, je z dostupné literatury možné vyvodit jakýsi obecný návod k realizaci BCI na bázi EEG:

- **Výber mentálních úkolů:** prvním krokem, obvykle zahrnutým do definice experimentu, je výběr studovaných mentálních úkolů. Výběr mentálních úkolů vychází z očekávaného použití BCI.
- **Výběr EEG kanálů:** obvyklá EEG aparatura se skládá z desítek až stovek elektrod, z nichž každá je napojena na vlastní výstupní kanál. Při výběru kanálů je nutné zohlednit jejich počet a umístění. Umístění vyplývá z povahy zvolených mentálních úkolů, počet kanálů zase výrazně ovlivňuje výpočetní nároky BCI.



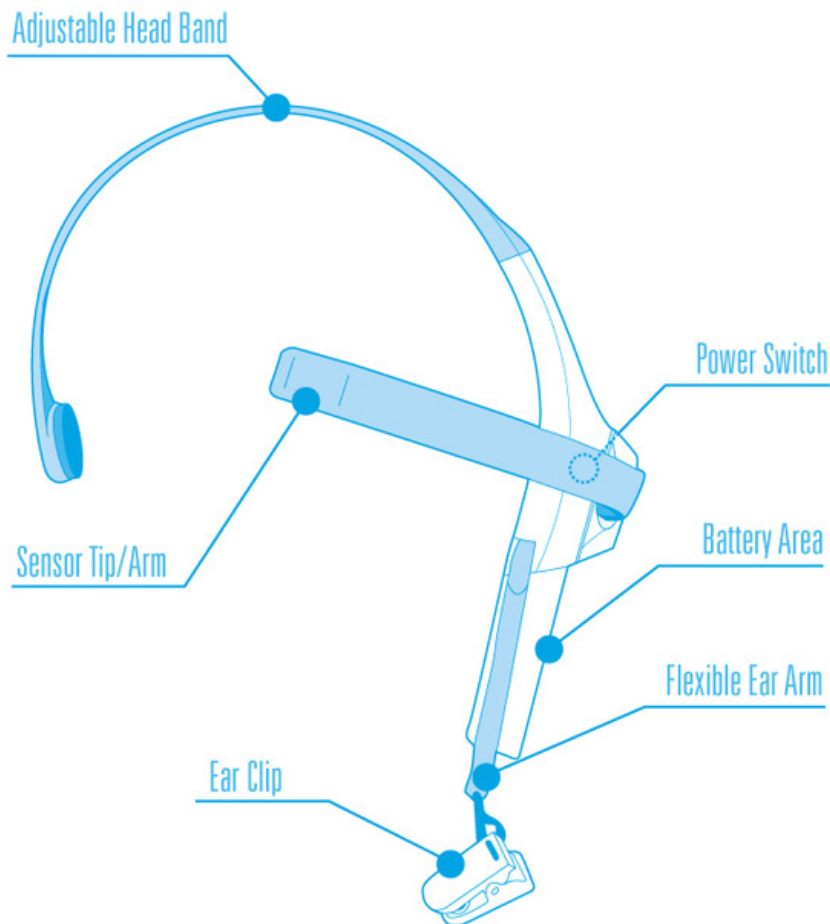
Obrázek 2.1: Mezinárodní 10–20 systém umístění EEG elektrod. Zdroj: [7]

- **Akvizice dat:** obvyklým způsobem pro sběr dat je nechat uživatele několikrát provádět různé mentální úkoly v rámci střídajících se kratších stop. Pomocí střídání je možné částečně odfiltrovat změny pomalých mozkových vln. Stopy jsou následně rozděleny na kratší časová okna, obvykle s mírným překryvem, která reprezentují instance dat.
- **Předzpracování dat:** v rámci předzpracování dat je obvykle provedena nějaká forma procedury *feature extraction* a *feature selection* (popsáno dále), případně jiné metody redukce dimenzionality. Následně mohou být dle požadavku zvoleného klasifikátoru data normalizována a jiným způsobem očištěna.
- **Klasifikace:** pro klasifikaci je třeba využít modely schopné poskytovat rozumné výsledky při malém množství dat a zároveň vysoké dimenzionalitě problému. Nejčastěji používanými klasifikátory jsou SVM a Naive Bayes.

2.1.3 Mindwave Mobile 2

Zařízení, pomocí kterého je v této práci implementováno BCI, se nazývá Mindwave Mobile 2. Jedná se o jeden z nejsnáze komerčně dostupných senzorů (cena v řádu nízkých stovek USD) a tomu odpovídá i jeho design: zjevnou nevýhodou je použití pouze jednoho kanálu na fixní Fp1 pozici (viz 2.1), naopak výhodou je jeho pohodlné použití díky bezdrátovému řešení a suchému provedení elektrody.

Mindwave obsahuje pouze jednu měřicí elektrodu. Jedná se o aktivní systém, kromě referenční elektrody tedy ještě navíc obsahuje i uzemnění. Měřicí elektroda se nachází na Fp1 pozici, tedy na levé části čela uživatele.



Obrázek 2.2: Headset Mindwave Mobile 2. Zdroj: [8]

Referenční a uzemňující elektrodu nalezneme na skřipci, který patří na levý ušní boltec uživatele (tedy A1 pozici dle mezinárodního 10–20 systému umístění elektrod).

Uvnitř zařízení se nachází ThinkGear čip, na kterém probíhá zpracování signálu a obsluha komunikace přes Bluetooth rozhraní. Signál je snímán s frekvencí 512 Hz. V rámci předzpracování je signál automaticky očištěn od šumu. V rámci tohoto čištění jsou odstraněny některé známé artefakty signálu a odfiltrovány frekvence nižší než 0,5 Hz a vyšší než 50 Hz. Kromě čištění signálu na čipu také probíhají výpočty různých metrik signálu, jež se odesílají souhrnně za poslední vteřinu.

Headset odesílá několik různých druhů dat, mezi nimi zejména:

- **RAW wave value:** hodnota naměřené EEG vlny po očištění signálu. Tyto hodnoty jsou v práci použity jako zdroj dat.
- **Poor quality:** indikátor nízké kvality signálu. Vyšší hodnoty značí horší signál. Hodnota 200 indikuje ztrátu kontaktu headsetu a uživatele.
- **eSense metriky** (*Meditation, Attention, Eye-blink strength*): proprietární metriky firmy NeuroSky. Nabývají hodnot 0–100. V rámci práce nejsou použity.
- **EEG Power:** Hodnoty osmi základních mozkových vln dle metodologie firmy NeuroSky (delta (0.5–2.75Hz), theta (3.5–6.75Hz), low-alpha (7.5–9.25Hz), high-alpha (10–11.75Hz), low-beta (13–16.75Hz), high-beta (18–29.75Hz), low-gamma (31–39.75Hz), a mid-gamma (41–49.75Hz). Tyto hodnoty nemají žádnou jednotku a dávají tak smysl jen při vzájemném porovnávání. V rámci práce nejsou použity.

Výhodou zařízení s nízkým počtem elektrod a umístěním elektrody na čele je jeho uživatelská přívětivost. Zatímco při použití elektrod v jiných oblastech hlavy je nutné kvůli zlepšení vodivosti používat tzv. „mokrý“ elektrody (kde je nutné kvůli vlasům používat vodivý gel), zařízení používající elektrody na čele se obejdou se „suchou“ variantou. Díky tomu je možné zařízení začít používat takřka okamžitě, což je z hlediska jeho použitelnosti pro BCI naprosto zásadní vlastností.

2.1.4 Parrot Mambo Mission

Parrot Mambo Mission je programovatelný minidron. Ovládá se pomocí ovladače podobného ovladačům používaným u herních konzol, na který je možné připevnit smartphone pro podpůrné funkce. Pomocí ovladače je možné minidron velice snadno a s vysokou přesností ovládat.

Minidron je možné vybavit příslušenstvím: puškou na kuličky nebo jeřábem. S pomocí jeřábu je minidron schopen z místa na místo přenášet drobné předměty až do hmotnosti 7 g. Zároveň je vybaven vertikální kamerou, pomocí které je možné snímat obraz pod dronem.

Napájení minidronu je realizováno vyměnitelnými bateriemi. Výdrž jedné baterie je 8 minut letu, nabíjení trvá přibližně 30 minut. K ostatním zařízením se minidron připojuje přes wifi nebo BLE standard 4.0. Rozhraní pro komunikaci minidrona s aplikacemi v jazyce Python použité v této práci ([9]) implementuje BLE spojení pomocí knihovny `bluepy`. Díky této závislosti je připojení na minidron možné pouze z linuxových systémů a při použití .



Obrázek 2.3: Minidron Parrot Mambo Mission. Zdroj vlastní.

2.2 Návrh

V práci je realizováno ovládání minidrona skrze BCI na bázi EEG signálu s použitím Mindwave Mobile 2 pro nahrávání EEG dat a minidrona Parrot Mambo Mission jako ovládaného robota.

Hlavní část práce je realizována ve formě aplikace v jazyce Python. Explorativní analýza dat byla provedena pomocí jupyter notebooků. S pomocí výsledné aplikace je možné realizovat online klasifikaci a provádět offline experimenty nad dříve naměřenými daty.

Při použití BCI jsou nejprve naměřena data pro kalibraci klasifikátoru, následně jsou automaticky zpracována a je nad nimi sestaven klasifikátor použitelný pro online klasifikaci daných mentálních úkolů. Trvanlivost klasifikátoru je díky unikátnosti a časové nestálosti EEG vln [2] omezená v řádu desítek minut, kalibraci je tedy nutné provádět při každém použití BCI znovu. Zároveň není možné pro analýzu použít souhrnný dataset a vytvořit tak univerzální klasifikátor.

2.2.1 Akvizice dat

Při akvizici dat jsou nahrávány stopy EEG signálu zachycující provádění jednoho mentálního úkolu uživatelem. Délka jedné takové stopy je 15 s. Vyšší počet nahraných stop pozitivně ovlivňuje odolnost klasifikátoru proti „dlouhým“ vlnám a prodlužuje tak trvanlivost klasifikátoru. Ze stejného důvodu jsou při nahrávání stop jednotlivé mentální úkoly střídány. Pro správnou funkčnost aplikace je nutné naměřit alespoň tři stopy každého mentálního úkolu, nejlépe ale alespoň pět.

Při nízkém počtu trénovacích stop je sice možné dosáhnout vysoké úspěšnosti klasifikátoru v rámci testovacích dat, dojde ale k jeho přeučení vlivem dlouhých vln, díky čemuž je pak klasifikátor pro online klasifikaci v zásadě nepoužitelný. Zároveň je nutné věnovat zvýšenou pozornost správnému označení naměřených stop a z výsledného datasetu odstranit stopy nepovedené (bylo experimentálně ověřeno, že k nefunkčnosti klasifikace spolehlivě stačí jediná špatně označená stopa).

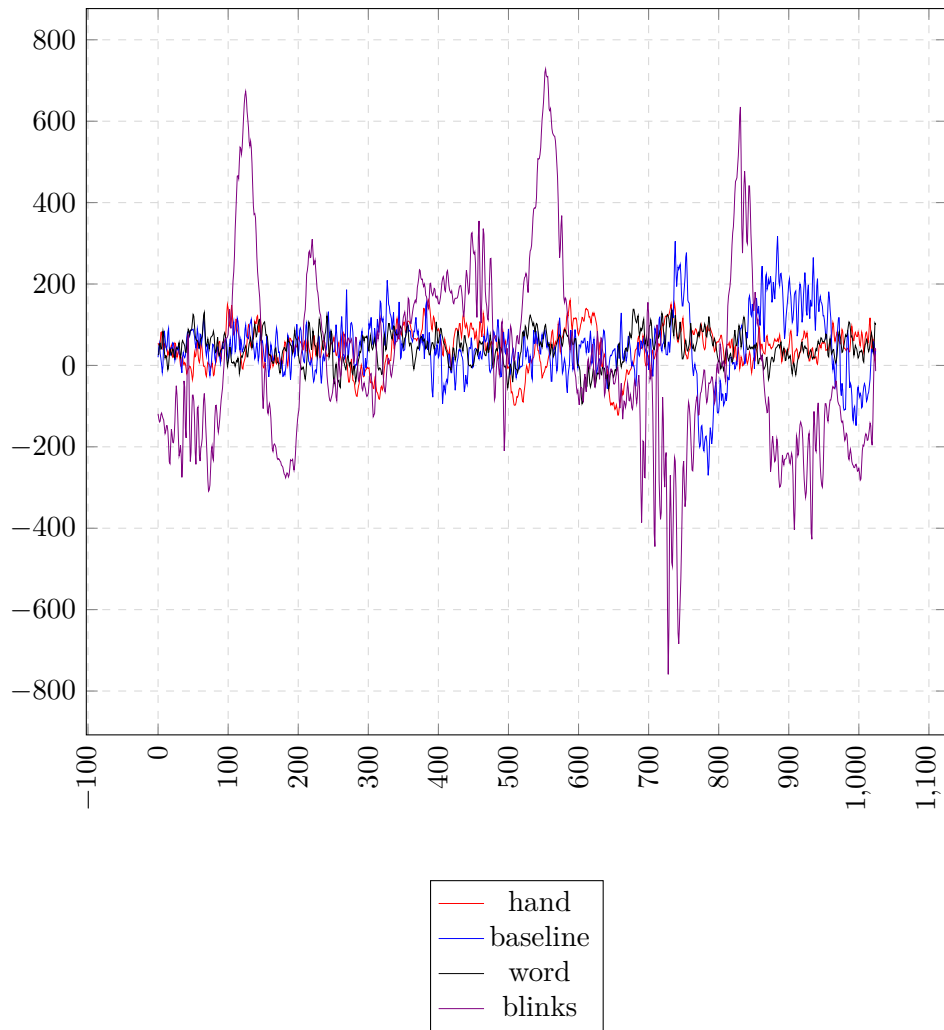
Vzhledem k logistické náročnosti získávání EEG dat je rozumné tato ještě v syrové formě ukládat. Data jsou ukládána odděleně po stopách tak, aby byla snadno použitelná pro různé opakované experimenty nad jejich podmnožinami. Samotné stopy je nutno rozdělit na kratší úseky. Délka takového úseku je jedním z hlavních parametrů experimentu. Kratší délka úseku má na úspěšnost klasifikace pozitivní vliv díky většímu množství instancí dat obsažených v naměřených stopách a zároveň může zkrátit délku výpočtu. Naopak delší časové úseky mají šanci lépe postihnout vliv pomalejších složek EEG signálu. V literatuře je délka úseku volena v rozsahu mezi 1 a 3 sekundami. V rámci této práce byla zvolena délka okna 2 s. Motivací pro tuto volbu je fakt, že čištění signálu prováděné headsetem filtruje signál s nižší frekvencí než 0,5 Hz, tudíž se při volbě 2s úseku může při analýze projevit i nejpomalejší vlna zachycená v datech. Pro ověření vhodnosti této délky bylo učiněno několik kontrolních měření úspěšnosti klasifikace při použití 1s a 3s úseků, obě tyto varianty vykázaly horší výsledky. V rámci práce jsou prezentovány pouze výsledky nad 2s okny, ve výsledné aplikaci lze ale délku okna v kódu poměrně snadno upravit (popsáno v dokumentaci kódu).

Dalším použitým parametrem je délka překryvu oken. Pokud se sousední okna mírně překrývají, je možné lépe zachytit charakter dat v časové doméně a zároveň získat větší množství instancí trénovacích dat. Délka překryvu byla experimentálně zvolena na 256 hodnot, tedy 0,5 s. Podobně jako délku okna je i tuto hodnotu poměrně snadné v rámci kódu aplikace změnit.

Headset zaznamenává data se vzorkovací frekvencí 512 Hz, naměřená data nabývají celočíselných hodnot v rozsahu $\langle -2048, 2047 \rangle$. Instance dat má při zvolené délce úseku 2 s podobu vektoru 1024 hodnot v daném rozsahu.

Instanci zkoumaných dat d_i lze tedy definovat takto:

$$d_i = (x_1, \dots, x_{1024}) \quad \forall x \in d_i : x \in \mathbb{Z} \wedge x \in \langle -2048, 2047 \rangle \quad (2.1)$$



Obrázek 2.4: Tvar 2s EEG signálu různých mentálních úkolů.

2.2.2 Předzpracování dat

Předzpracováním dat se myslí všechny operace provedené nad daty od jejich rozdělení do kratších oken po předání výsledného datasetu ke klasifikaci. Nad daty je nejprve provedena extrakce příznaků a následně jejich selekce dle očekávané relevantnosti. Nakonec jsou data normalizována.

Používat pro vstup klasifikátoru naměřené hodnoty přímo je velice nevýhodné. Pro ilustraci může sloužit následující experiment: mějme problém, ve kterém má klasifikátor rozpoznat, jestli v daném úseku došlo nebo nedošlo k mrknutí. Mrknutí, jak bylo zmíněno v analytické části, lze na záznamu EEG vlny snadno rozpoznat pouhým okem díky charakteristickému artefaktu. Na obrázku vidíme tři instance dat s mrknutím: ve dvou z nich k mrknutí došlo v rámci první vteřiny, v poslední v rámci druhé. Předpokladem k úspěšné klasifikaci v případě, že klasifikátor jako trénovací data dostane první dvě instance, je jeho schopnost vydedukovat, že důležitou vlastností dat není „vizuální“ podobnost tvaru vlny, ale přítomnost artefaktu mrknutí, na jehož přesném umístění nezáleží.

Sestojme druhý klasifikátor, který místo všech hodnot časové řady jako instance dat použije maxima a minima naměřených hodnot. Zatímco první klasifikátor dojde k závěru, že třídy lze rozlišit podle přítomnosti vyšších hodnot v první polovině dat (a druhou polovinu hodnot vyhodnotí jako pro klasifikaci neúčinnou, čímž zrychlí dobu výpočtu), druhý velice snadno zjistí, že maximum hodnot u dat popisujících mrknutí je výrazně vyšší než u dat bez mrknutí. Výpočetní nároky druhého klasifikátoru budou zjevně nižší, protože na rozdíl od mnohadimenzionálního prostoru hodnot operuje pouze ve dvou dimenzích.

Operaci, která instance dat z mnohadimenzionálního, informačně chudého prostoru hodnot zobrazuje do informačně bohatého prostoru příznaků, nazýváme extrakce příznaků (**feature extraction**).

Příznaky extrahované z časové řady se dělí do několika základních skupin:

- **Statistická doména:** běžné statistické příznaky typu *průměr*, *medián*, *maximum*, *minimum*, *kvantily*, *kurtosis*, *rozptyl* atd.
- **Frekvenční doména:** příznaky z frekvenční domény popisují vlastnosti frekvenčního spektra (tedy vln o rozdílné frekvenci, ze kterých se signál skládá). Frekvenční příznaky lze získat například pomocí FFT nebo vlnkové transformace (která navíc popisuje vlastnosti signálu i v časové doméně).
- **Časová doména:** příznaky časové domény popisují chování signálu v čase. Typickými příznaky časové domény jsou lineární trendy nebo autokorelace (popisující míru podobnosti signálu v čase).

Zajímavé netriviální příznaky, které byly vyhodnoceny jako vyjíměčně užitečné, jsou diskutovány v experimentální části práce.

Extrakci příznaků je možné realizovat pomocí různých softwarových nástrojů. Takový nástroj je obvykle uzpůsoben nejen pro analýzu EEG signálu, ale i různých jiných časových řad (data z termostatu, vývoj ceny akcií atd.). Z této univerzality plyne, že zdaleka ne všechny extrahované příznaky jsou pro zkoumání rozdílnosti konkrétních mentálních úkolů užitečné a reprezentují tak zbytečný šum, který bylo navíc výpočetně drahé získat.

Provedení extrakce příznaků je výpočetně nejnáročnější operací, kterou je v rámci implementace nutné realizovat, vyplatí se tedy věnovat zvýšenou pozornost selekci vhodných příznaků (**feature selection**).

Pro extrakci a selekci příznaků je v rámci práce použit balíček TSFRESH [10], který z časové řady extrahuje 794 vlastností pomocí 64 různých metod a pro selekci příznaků implementuje algoritmus FRESH [11]. Při použití metody `extract_features` jsou tedy data transformována do vektoru příznaků o 794 hodnotách. Příznaky nabývají pouze číselných hodnot, je tedy možné je bez dalšího zpracování dále používat.

Algoritmus FRESH testuje relevanci jednotlivých příznaků pro klasifikaci pomocí statistických testů. Příznak je označen za irelevantní pro klasifikaci, pokud je rozdělení jeho hodnot statisticky nezávislé na rozdělení příslušnosti prvků do tříd.

Algoritmus nejprve podle tvaru rozdělení hodnot zvolí vhodný statistický test a s jeho pomocí spočte p -hodnotu pro zamítnutí nulové hypotézy $H_0 = \{\text{Příznak je irelevantní}\}$ oproti alternativní hypotéze $H_1 = \{\text{Příznak je relevantní}\}$. Jedním možným způsobem výběru relevantních příznaků je stanovit pevnou maximální hladinu p -hodnoty a vybrat všechny příznaky, jejichž p -hodnota se bude nacházet pod touto hladinou. Tento přístup funguje v případě, že počet relevantních a irelevantních příznaků je přibližně vyrovnaný, ovšem v případě, že je mnohem více příznaků nerelevantních, samotná p -hodnota pro výběr vhodné podmnožiny nestačí: nutně budou vybrány i některé nerelevantní příznaky. Maximální množství takto vybraných příznaků (FDR - false discovery rate) je výhodné nějakým způsobem regulovat. Toho je v algoritmu FRESH je dosaženo pomocí parametru q Benjamini–Yekutielli procedury.

Benjamini–Yekutielli procedura zamítne ty nulové hypotézy H_0^ϕ , jejichž p -hodnoty jsou nižší než hodnota jejich úrovně zamítnutí r_ϕ , která je dána následující lineární posloupností:

$$r_\phi = \frac{\phi q}{n_\phi \sum_{\mu=1}^{\phi} \frac{1}{\mu}} \quad (2.2)$$

Kde ϕ značí pozici p -hodnoty hypotézy H_0^ϕ ve vzestupném pořadí p -hodnot, n_ϕ počet všech nulových hypotéz a q řídicí parametr procedury. [11]

V rámci realizace řešení bylo zjištěno, že ačkoli se přesné složení množiny vybraných příznaků napříč mezi datasey liší, některé příznaky bývají zvoleny častěji než ostatní. Na základě tohoto pozorování byla formulována hypotéza,

že některé příznaky jsou užitečné univerzálně. V případě, že by se podařilo univerzální příznaky nalézt, bylo by možné výrazně omezit náročnost výpočtu extrakce příznaků a zároveň zjistit, jaké informace obsažené v příznacích jsou pro rozlišování mentálních úkolů důležité. Výsledky analýzy příznaků jsou součástí přílohy této práce.

Poslední operací provedenou nad daty je jejich **normalizace**. Jelikož jednotlivé příznaky nabývají hodnot v různém rozsahu, při metodách klasifikace založených na vzdálenosti mezi hodnotami by v původní podobě příznaky nabývající vysokých hodnot zastínily vliv příznaků s nižšími hodnotami. Normalizace je transformací dat, která hodnoty jednotlivých příznaků převede do společného intervalu.

V rámci práce je normalizace provedena pomocí metody **MinMaxScaler**:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2.3)$$

kde $x = (x_1, \dots, x_n)$ jsou hodnoty normalizovaného příznaku v originálním datasetu a z_i je hodnota příznaku x ve vektoru, nad kterým je prováděna normalizace.

Výsledkem této metody jsou hodnoty v rozsahu $\langle 0, 1 \rangle$. Nutnou podmínkou pro funkčnost klasifikátoru je provést stejnou normalizaci nad daty i v rámci online klasifikace (s použitím hodnot z trénovacího datasetu).

2.2.3 Analýza dat

Analýza dat byla provedena ve dvou fázích. V explorativní fázi bylo zkoumáno, jestli a jakým způsobem jsou data klasifikovatelná a jakou mají podobu. V experimentální části je zkoumána frekvence výběru konkrétních příznaků napříč různými experimenty a na základě toho jsou formulována tvrzení o způsobu zachycení informace v EEG signálu.

V rámci experimentů jsou používána následující označení pro dílčí skupiny dat:

- Trénovací data: skupina stop použitá pro trénování klasifikátoru.
- Testovací data: podmnožina trénovacích dat, která jsou při trénování klasifikátoru skryta a po dokončení kalibrace je s jejich pomocí vyhodnocena kvalita modelu.
- Reálná data: skupina stop naměřená s časovým odstupem od trénovacích stop, pomocí které je simulováno chování BCI v reálných podmínkách. Pomocí této množiny je vyhodnocována trvanlivost klasifikátoru (neboli míra přeučení modelu na daný časový úsek EEG vln).

Pro explorativní analýzu byl použit program Jupyter notebook. Byly sestaveny tři notebooky s následujícím použitím:

- **top-k notebook**: experimentace se složením vybraných příznaků s možností volby parametrů (počet, metoda výběru).
- **PCA notebook**: experimentace s použitím PCA jako alternativní metody redukce dimenzionality vůči FRESH algoritmu.
- **classification notebook**: experimentace s kvalitou různých klasifikátorů nad výstupními daty z procedury extrakce nebo selekce příznaků. Jako vstup tohoto notebooku je možné použít výstupy předchozích dvou notebooků.

Vstupem **top-k notebooku** jsou data získaná pomocí extrakce příznaků nebo selekce příznaků. Smyslem tohoto notebooku je ve snadno interpretovatelné podobě získat představu o klasifikovatelnosti libovolného datasetu (podmnožiny stop).

V rámci notebooku se zvolí vstupní dataset a počet vybíraných příznaků. Výstupem notebooku je dataset omezený na vybrané příznaky a výpis těchto vybraných příznaků. Tento dataset je následně možné podrobit vizuální analýze například skrze graf paralelních souřadnic, na jejímž základě lze snadno formulovat tvrzení o předpokládané klasifikovatelnosti vybraných mentálních úkolů. Na základě této prvotní analýzy bylo konstatováno, že data klasifikovatelná jsou.

Druhou metodou pro zjištění klasifikovatelnosti dat je použití PCA. Při zvolení nízkého počtu principiálních komponent je pomocí PCA notebooku možné snadno interpretovat data stejným způsobem jako při použití **top-k notebooku**.

classification notebook porovnává kvalitu klasifikátorů **random forest**, **SVM** a referenčního **knn**. Pro hodnocení kvality se používá metrika *f1-score*, která je dobrým kompromisem mezi metrikami *precision* a *recall*. Bylo zjištěno, že na testovacích datech je téměř vždy dosaženo velmi dobré úspěšnosti klasifikace. Dále bylo také zjištěno, že ačkoli klasifikátor **random forest** dosahuje zdaleka nejlepších výsledků na testovacích datech, jeho výkon při použití na stopách mimo trénovací dataset je tristní a dochází tak u něj k přeučení na konkrétní časový úsek EEG vln. Na základě tohoto zjištění byla formulována metoda ověření kvality klasifikátoru na reálných, později naměřených datech, která odpovídá reálnému použití BCI. Přijatelných výsledků na reálných datech bylo dosaženo s použitím modelu **SVM**, který byl zvolen pro samotnou implementaci.

2.2.4 Klasifikace

Model **SVM** konstruuje nadrovinu, jejíž vzdálenost od všech instancí testovacích dat je co možná největší (pro příklad ve dvoudimenzionálním prostoru viz Obr. 2.5). Data jsou následně klasifikována podle toho, na které straně nadroviny se nacházejí. Úloha nalezení optimální (dostatečně široké, ale zároveň do-

statečně přesné) nadroviny je problémem kvadratického programování. Cena nesprávné kvalifikace je zde vyjádřena parametrem $C > 0$. Při hledání nadroviny se využívá zobrazení dat do vysokodimenzionálních prostorů, díky čemuž je možné klasifikovat i data, která by nadrovinou v původním prostoru nebylo možné rozdělit. Jelikož je toto zobrazení výpočetně náročnou operací, je nadrovina reprezentována implicitně skrze tzv. *support vectors* (na obrázku označené kroužkem, jedná se o vektory nejméně vzdálené od nadroviny), jejichž projekci do vysokodimenzionálního prostoru lze efektivně realizovat pomocí kernelové funkce $\phi(x)$.

Pomocí rozhodovací funkce 2.4 predikuje SVM příslušnost zkoumaného vektoru x do třídy $y \in \{-1, 1\}$.

$$\text{sgn} \left(\sum_{i=1}^l y_i \alpha_i K(x_i, x) + b \right), \quad (2.4)$$

kde x_i je pomocný vektor, $y_i \in \{-1, 1\}$ označuje příslušnost pomocného vektoru k třídě, α_i je váhou pomocného vektoru a $K(x_i, x) \equiv \phi(x_i)^T \phi(x)$ je kernelová funkce. [12]

V implementaci je použit RFB kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0, \quad (2.5)$$

kde $\|x_i - x_j\|$ značí eukleidovskou vzdálenost mezi vektory a γ parametr ovlivňující sílu vlivu jednotlivých instancí dat. Implementace SVM použitá v této práci je založena na LIBSVM [12] a implementována pro jazyk Python v balíčku `scikit-learn` [13]. Podpora vícetržní klasifikace je realizována formou `one-vs-one`.

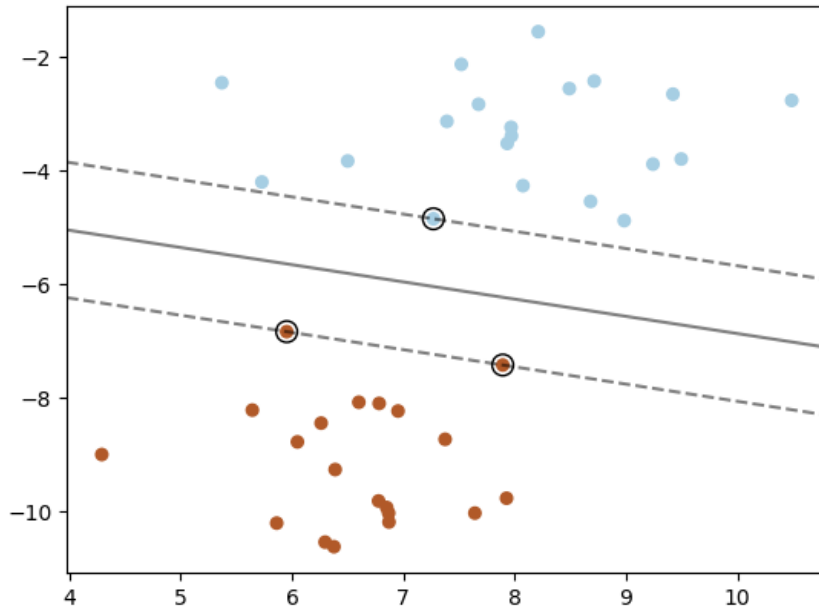
2.2.5 Návrh aplikace

Cílem práce je vyvinout BCI na bázi EEG a zároveň nad EEG daty provést odpovídající analýzu. Oba tyto základní úkoly jsou realizovány pomocí aplikace v jazyce Python. Očekávají se tři základní způsoby využití aplikace:

- Nahrávání EEG dat
- Provádění experimentů nad EEG daty
- Ovládání (nejen) robotů pomocí mentálních úkolů

Při prvním a druhém způsobu použití není nutné připojovat k aplikaci samotné BCI (minidron). Pro analýzu dříve naměřených EEG dat není nutné připojovat ani headset. Na základě toho byla aplikace rozdělena na tři logické moduly tak, aby bylo možné aplikaci používat i bez samotných zařízení.

Modul `server` obstarává komunikaci mezi headsetem a aplikací. Modul `controller` realizuje překlad identifikovaných mentálních úkolů na příkazy pro robota. Modul `core` realizuje zpracování dat.



Obrázek 2.5: Konstrukce optimální nadroviny SVM. Zdroj: [13]

`server` je navržen tak, aby v případě navázaného spojení dodával naměřené hodnoty EEG vlny klientovi. Je tedy snadno použitelný i v jiných aplikacích a zároveň snadno vyměnitelný. V implementaci serveru jsou použity skripty `python-mindwave-mobile` [14].

`controller` je v rámci práce navržen jako abstraktní použití BCI, které při dodržení jednoduchého komunikačního schématu může sloužit k různým aplikacím (rozpoznávání uživatelů, brain-to-text, ovládání robota). `controller`, který je v rámci práce realizován, je čistě demonstrativního charakteru. (Dron na zemi zavře jeřáb a přitom uchytí předmět, vzlétne, letí dopředu, přistane, jeřáb otevře a předmět pustí, to vše s využitím třech mentálních úkolů, které si uživatel sám definuje.) Pro realizaci ovládání minidronu je použit balíček `pyparrot` [9].

`core` realizuje analýzu a klasifikaci dat. Modul umožňuje nahrávání EEG dat, offline a online klasifikaci. Pro nahrávání dat je nutné, aby byl současně s jádrem spuštěn modul `server`. Pro online klasifikaci je navíc nutné spustit modul `controller`. Pro offline klasifikaci (`experimenty`) stačí spustit samotné `core`.

Aplikace se může nacházet ve třech základních stavech: nahrávání, kalibrace a nasazení.

- **Nahrávání:** při nahrávání je vytvářen nový dataset, který je následně možné použít pro kalibraci klasifikátoru. Naměřená data jsou uložena do souborů odděleně po jednotlivých stopách.
- **Kalibrace:** při kalibraci je na základě nějakého datasetu sestaven klasifikátor. Klasifikátor je následně možné použít pro nasazení, nebo pro opakované vyhodnocení kvality klasifikace nad cílovým datasetem. Pro použití jiného datasetu stačí kalibraci znovu spustit nad daným datasetem.
- **Nasazení:** při nasazení je realizováno BCI s použitím vybraného modulu `controller`, kterému jsou odesílány identifikátory předpověděných tříd.

V průběhu nahrávání není možné provést kalibraci ani nasazení online klasifikace. V průběhu nasazení naopak není možné kalibrovat ani nahrávat nová data. Z kalibrovaného stavu je možné nahrávat nové datasety nebo nasadit klasifikátor se současnou kalibrací do provozu.

Realizace

Sekce implementace popisuje realizaci aplikace a způsoby jejího použití. Sekce experimenty popisuje analýzu výsledků odděleně podle dílčích experimentů. V sekci výsledky jsou představeny univerzálně užitečné vlastnosti časových řad pro rozlišování mentálních úkolů. Na základě univerzálních vlastností jsou formulovány hypotézy o tom, jakým způsobem se od sebe mentální úkoly odlišují.

3.1 Implementace

Realizovaná aplikace je experimentálním prototypem BCI na bázi EEG. Pro implementaci byl vybrán jazyk Python. Aplikace je rozdělena na tři samostatně spustitelné logické moduly `server`, `core` a `controller`.

Modul `server` je jednoduchou nástavbou nad knihovnou `python-mindwave-mobile` [14], která implementuje přijímání dat z headsetu skrze rozhraní bluetooth a jejich dekódování dle protokolu ThinkGear [15]. Server se připojí na headset a následně čeká na spojení s jádrem. V okamžiku, kdy je spojení navázáno, zahodí neaktuální data ze socketu a začne každou vteřinu odesílat naměřené hodnoty. V případě přerušení spojení čeká na jeho opětovné navázání.

Modul `controller` implementuje funkcionalitu BCI. Při dodržení komunikačního protokolu (popsáno v dokumentaci souboru `dummy_controller.py`) je možné tento modul snadno vyměnit. `controller` realizovaný v této práci umožňuje otevírat a zavírat jeřáb minidrona, s minidronem vzlétnout, odletět vpřed nebo vzad a přenést tak pomocí myšlenky předmět z bodu A do bodu B.

Modul `core` implementuje uživatelské rozhraní aplikace a veškeré operace spojené s akvizicí, zpracováním a klasifikací dat. Funkcionalita a implementace modulu `core` je popsána v samostatné části.

Součástí implementace jsou pomocné skripty pro realizaci experimentů (`fast_crunch.py`), testování BCI a funkčnosti ovládání robota (`dummy_bci.py`)

a `dummy_controller.py`). Podrobnější informace o implementaci těchto skriptů jsou k nalezení v jejich dokumentaci.

3.1.1 Použití aplikace

Dle návrhu je aplikaci možné použít ke třem různým účelům:

- nahrávání EEG dat,
- experimentování s dříve nahranými daty,
- použití nějaké aplikace BCI (ovládání minidronu, `brain-to-text`,...).

Aplikace se ovládá pomocí rozhraní CLI implementovaného v modulu `core`. Skrze příkazy zadávané pomocí znaků lze spustit různé funkce aplikace.

Smyslem nahrávání je uložit hodnoty EEG vlny v perzistentní podobě odděleně po stopách jednotlivých mentálních úkolů. **Nahrávání** stop lze shrnout do následujících kroků:

1. Uživatel zapne headset a spustí modul `server`.
2. Uživatel zvolí možnost nahrání nové stopy (příkaz `N` v CLI), následně zadá jméno mentálního úkolu. Po krátkém odpočtu je zahájeno nahrávání mentálního úkolu.
3. Uživatel nahraje požadovaný počet stop opakovanou volbou příkazu `R` s následnou specifikací mentálního úkolu.
4. Uživatel ukončí nahrávání pomocí příkazu `E`.
5. Uživatel může zahájit nové nahrávání nebo spustit kalibraci.
6. Po ukončení nahrávání nalezne uživatel nový dataset v samostatném adresáři ve složce `data`.

Po dokončení nahrávání všech stop je automaticky vygenerován vysvětlující soubor `command.json`, který popisuje automatické přiřazení tříd (čísel) mentálním úkolům. Přečíslováním mentálních úkolů lze snadno změnit originální složení tříd (například pro `one-vs-rest` klasifikaci).

Pomocí aplikace je možné nad daty provádět různé experimenty a vyhodnocovat tak užitečnost konkrétních mentálních úkolů pro klasifikaci. Cílový dataset se musí nacházet ve složce `data`. Pro spuštění kalibrace nad cílovým datasetem uživateli stačí v rozhraní CLI zvolit možnost `C` a následně specifikovat adresář s cílovým datasetem.

Po dokončení kalibrace je do konzole vypsána zpráva o kvalitě sestrojeného klasifikátoru. Zároveň se do adresáře cílového datasetu uloží soubory obsahující data získaná při zpracování dat (viz Formáty dat).

Pro zprovoznění BCI uživatel provede následující kroky:

1. Uživatel nahraje trénovací data dle návodu popsaného výše.
2. Uživatel provede kalibraci klasifikátoru.
3. Uživatel spustí nějaký modul `controller`.
4. Uživatel zvolí v CLI možnost D, čímž spustí online klasifikaci.

Modul `controller` implementovaný v rámci této práce umožňuje ovládat robota myšlenkou. Pomocí tří mentálních příkazů je možné ovládat jeřáb minidrona (jeho zavření nebo otevření), jeho vzlet či přistání a krátký let dopředu s následným otočením o 180°. Teoreticky by tedy měl dron bezpečně operovat jen na pevně omezeném prostoru a jeho ovládání by tak mělo být bezpečné.

3.1.2 Formáty dat

Pro perzistentní uložení dat jsou v implementaci používány následující soubory:

- Soubory typu `.rawdata`: tyto soubory obsahují hodnoty naměřené při snímání stopy mentálního úkolu.
- `selected_features.csv`: uložené hodnoty extrahovaných příznaků.
- `selected_features.csv`: uložené hodnoty vybraných příznaků.
- `selected_features.txt`: uložené názvy vybraných příznaků.

Naměřené EEG hodnoty je výhodné ukládat ve formátu který je snadné nahrát do tabulkových procesorů pro vizuální analýzu stopy. Zároveň není vhodné již při tomto ukládání stopy dělit na menší časová okna. Jednotlivé EEG stopy jsou ukládány ve formátu `.rawdata`, který na první řádce obsahuje informace o stopě a na dalších řádkách pak samotné hodnoty.

Hodnoty extrahovaných příznaků se ukládají do `.csv` souboru. Nad danou časovou řadou s danou délkou oken se jedná o deterministickou operaci, v zájmu ušetření výpočetních prostředků a času uživatele je tedy při běhu programu proveden test na přítomnost souboru `extracted_features.csv` a v případě jeho nalezení jsou hodnoty extrahovaných příznaků načteny z něj místo jejího opětovného provedení. V případě změny nějakého parametru (délka okna, definice tříd, složení stop, ...) je nutné starý soubor `extracted_features.csv` odstranit a provést extrakci příznaků znovu.

Hodnoty vybraných příznaků se ukládají do souboru `selected_features.csv`. Tento soubor je přepsán při každém běhu programu nad daným souborem stop. Seznam jmen vybraných příznaků se ukládá do souboru `extracted_features.txt`. Pomocí těchto souborů je možné provádět automatickou analýzu vybraných příznaků v rámci obsáhlejších experimentů pomocí skriptu `analyse_features.py`.

3.1.3 Implementace modulu core

Logický modul `core` se skládá z následujících submodulů:

- `core.py`: Jádru celé aplikace. Implementuje zpracování EEG dat a realizuje klasifikaci.
- `recorder.py`: Realizuje akvizici dat.
- `cruncher.py`: Realizuje předzpracování dat.
- `deployer.py`: Realizuje online klasifikaci a offline simulaci online klasifikace.
- `messages.py`: Pomocný modul pro centralizaci hlášek vypisovaných programem.

Aplikaci je možné spustit pomocí skriptu `runcore.py`. Tento skript vytvoří instanci jádra aplikace a následně spustí smyčku CLI. Pomocí CLI je možné jádru předávat příkazy, například příkaz spuštění nahrávání stop (N), příkaz k nahrání stopy (R), příkaz ke kalibraci klasifikátoru (C) nebo příkaz k otestování offline klasifikace (T). Jádro automaticky kontroluje, jestli jsou k provedení příznaku vhodné podmínky, a v negativním případě příkaz neprovede. Parametrem konstruktora jádra je název datového adresáře (v základním spouštěcím skriptu se jedná o adresář `data`). Do tohoto adresáře jsou ukládány nahrávané stopy. Jádro zároveň předpokládá, že se v adresáři nacházejí všechny dataseťy, které má uživatel v úmyslu používat. Dostupnost takového adresáře z pracovního adresáře programu je nutnou podmínkou pro funkčnost aplikace.

Soubor `recorder.py` implementuje třídu `Recorder`, která je zodpovědná za nahrávání stop a jejich ukládání do perzistentní podoby tak, jak byla popsána v sekci Formáty dat. Při zahájení tvorby nového datasetu vytvoří adresář pro uložení stop, následně se pak při jednotlivých nahrávaních připojuje na `server` a získaná data pak zapisuje do souborů `.rawdata`. V momentě, kdy bylo dosaženo dostatečného množství dat specifikovaného délkou stopy, spojení přeruší. Komunikace se serverem je realizována pomocí knihovny `multiprocessing`.

Veškeré zpracování dat od rozdělení do oken po predikci třídy realizuje třída `Cruncher` implementovaná v souboru `cruncher.py`. Instance třídy je vytvořena při prvním spuštění kalibrace, při opětovných kalibracích je pak jen vyměněn trénovaný klasifikátor. Při kalibraci je nejprve provedeno načtení hodnot EEG vlny ze souborů `.rawdata`, následně jsou tyto rozděleny do oken a takto předány pro extrakci a selekci příznaků. Nad výsledným datasetem je následně natrénován klasifikátor SVM. Kvalita sestrojeného klasifikátoru je vyhodnocena na testovacích datech a výsledek je vypsán do konzole. Výstupem kalibrace je uložení instance klasifikátoru, který je následně možné použít buď pro nasazení, nebo experimenty s offline klasifikací. Pro manipulaci s daty jsou

využity kontejnery z knihoven `pandas` a `numpy`. Implementace klasifikátoru pochází z knihovny `scikit-learn` [13]. Extrakce a selekce příznaků je implementována pomocí knihovny `TSFRESH` [10].

V souboru `deployer.py` je implementována třída `Deployer`, která je zodpovědná za nasazení klasifikátoru do provozu. V rámci třídy jsou implementovány čtyři metody nasazení: offline klasifikace, která simuluje chování nad reálnými daty včetně délky běhu metody `extract_features` nad jednotlivými okny; souhrnná (bulk) offline klasifikace, kdy jsou jako vstup metody `extract_features` použity celé stopy; testovací nasazení, ve kterém jsou analyzovaná okna naplněna náhodnými hodnotami (toto nasazení lze použít pro test spojení s modulem `controller`); a konečně nasazení online klasifikace, které se připojí na moduly `server` a `controller`, a realizuje tak samotné BCI.

3.1.4 Praktické nasazení BCI

V rámci testování online klasifikace bylo zjištěno, že ačkoli se délka běhu metody `extract_features` při extrakci pouze vybraných příznaků výrazně zkrátila, délka analýzy jednotlivých oken zůstala nepříjemně dlouhá. Experimentální snižování počtu extrahovaných příznaků jen na absolutní minimum nikdy nezrychlilo výpočet pod 2 s. Z tohoto důvodu je bohužel nutné konstatovat, že knihovna `TSFRESH` není pro implementaci online klasifikace ideální. Ze stejného důvodu bylo zvoleno on-demand schema online klasifikace, kdy modul `core` provede analýzu prováděného příkazu pouze po obdržení požadavku od modulu `controller` (neprovádí tak analýzu průběžně, jak by pro BCI bylo ideální). Délka trvání extrakce vybraných příznaků nad jednotlivými okny je obvykle 5–7 s, což v kombinaci s odpočtem před zahájením nahrávání příkazu a samotným 2s nahráváním způsobuje pro praktické použití BCI nepříjemně dlouhou latenci.

Přijatelných výsledků online klasifikace bylo dosaženo pouze s použitím mentálního úkolu mrkání proti základnímu stavu (viz Experimenty). Výkon při online klasifikaci je ale bohužel ještě výrazně horší než výsledky klasifikace na reálných datech. Možným vysvětlením je nutnost uživatele soustředit se při používání BCI na minidrona, z čehož vyplývá jeho nižší soustředěnost na provádění mentálního úkolu.

Při testování nasazení BCI do provozu bylo zjištěno, že ovládání minidrona skrze aplikaci poměrně nebezpečné pro minidron a jeho okolí. Testovací dron, patrně vlivem závady na rotoru, v klidovém stavu driftoval a v omezených prostorách laboratoře tak v čase mezi příkazy obvykle narazil do zdi. Občas byla navíc pozorována značná prodleva mezi odesláním příkazu a jeho provedením. Vzhledem k nízké frekvenci zadávání příkazů (optimisticky jeden příkaz za 10 sekund) a jejich nízké přesnosti kvalifikace nelze nasazení BCI do ostrého provozu příliš doporučit.

Funkčnost BCI je vhodné před nasazením ověřit pomocí testovacích skriptů `dummy_drone.py` a `dummy_bci.py`, které simulují chování minidrona a jádra

aplikace. Pomocí prvního skriptu lze provést test správnosti identifikace příkazů bez nutnosti startovat minidron. Pomocí druhého je možné předávat dronu konkrétní příkazy. Teprve v případě uspokojivých výsledků obou testů je rozumné BCI nasazovat do provozu.

3.2 Experimentální část

V rámci experimentální části byly provedeny tři základní experimenty. Zkoumána byla zejména úspěšnost klasifikace na reálných datech s použitím různých vybraných mentálních úkolů. Bylo zjištěno, že trvanlivé klasifikátory je možné sestrojít pouze u dvoutřídních problémů. Přehled vybraných příznaků napříč různými datasety a experimenty lze nalézt v příloze ve složce `data`.

3.2.1 Akvizice dat

Bylo naměřeno celkem devět datasetů od šesti uživatelů. Přehled datasetů je uveden v tabulce 3.2.1. Mentální úkoly s prefixem `c_` byly naměřeny při zavřených očích. Počet stop je uveden ve formátu `trénovací / reálné`, kde trénovací stopy byly pořízeny s časovým odstupem minimálně 10 minut.

Při volbě mentálních úkolů nasbíraných od jednotlivých uživatelů bylo nutné vyvážit jistou konzistenci naměřených dat (aby bylo možné něco o mentálních úkolech tvrdit, měly by být naměřené na více než jednom uživateli), zároveň ale bylo cílem také zjistit, které mentální úkoly jsou pro tento problém relevantní. Výsledný mix mentálních úkolů je kompromisem mezi těmito nutnostmi: obvykle byly naměřeny nějaké úkoly spojené s evokovanými potenciály a zároveň i úkoly realizované pomocí techniky MI.

Použité mentální úkoly patří do základních skupin popsanych v analytické části práce. Úkoly označené názvem nějaké končetiny spočívají v upírání pozornosti způsobem simulace pohybu dané končetiny. Úkoly `word` spočívají v artikulaci slova nebo vizualizaci předmětu. Při úkolech `baseline` uživatelé svou pozornost žádným způsobem neupírali.

3.2.2 Realizace experimentů

Uváděné experimenty byly provedeny pomocí skriptu `fast_crunch.py`. Jako cílové datasety byly použity relevantní podmnožiny stop původních datasetů, vhodně přeznačené skrze soubor `command.json`. Pro každý cílový dataset byl nejprve natrénován klasifikátor, následně byla vyhodnocena jeho kvalita nad testovacími daty a v případě existence reálných dat i nad reálnými daty. Výsledky experimentů jsou popsány v tabulce 3.2.2 metrikou *f1-score*. Podrobné výsledky experimentu (matice zmatení a ostatní metriky) jsou popsány v příloze, společně s původními datasety, pomocí kterých lze experiment snadno opakovat. Návod pro spuštění experimentů je popsán v dokumentaci pomocného skriptu `fast_crunch.py`.

Název	Mentální úkoly	Počet stop
user_k	baseline, hand, word, closed	20 / 8
user_l	baseline, blinks, hand, leg, word	19 / 0
user_m1	baseline, c_word, c_motor, closed, blinks	14 / 0
user_m2	baseline, c_hand, blinks	9 / 0
user_m3	open, closed	9 / 5
user_m4	baseline, gaze	10 / 5
user_t	baseline, math, sound, finger, word	25 / 10
user_u	blinks, c_hand, c_word, baseline	13 / 7
user_v	baseline, closed, blinks, leg, hand	11 / 0

Tabulka 3.1: Přehled naměřených datasetů. Naměřené dataset sestává z naměřených (trénovacích) stop nějakého výběru mentálních úkolů. Některé datasety obsahují (reálné) stopy, které byly naměřeny s časovým odstupem. Počet stop je uváděn ve tvaru trénovací / reálné, 0 značí, že daný dataset reálné stopy neobsahuje. Prefix c_ značí, že byl daný mentální úkol prováděn při zavřených očích.

Dataset	Ex1A	Ex1B	Ex2	Ex3
user_k	1.0 / 0.75	-	0.85 / 0.53	0.87 / 0.42
user_l	-	-	-	0.78
user_m1	1.0	1.0	0.92	0.72
user_m2	1.0	0.86	-	0.96
user_m3	0.91 / 0.94	-	-	0.96 / 0.93
user_m4	-	-	-	0.78 / 0.88
user_t	-	-	0.78 / 0.47	0.68 / 0.28
user_u	1.0 / 0.79	0.74 / 0.83	0.91 / 0.66	0.72 / 0.54
user_v	1.0	0.86	-	0.92

Tabulka 3.2: Výsledky klasifikace v Ex1–Ex4. Sloupce popisují *f1-measure* klasifikátorů natrénovaných nad relevantními podmnožinami datasetů v jednotlivých experimentech nad testovacími/reálnými daty. – indikuje, že dataset pro daný experiment žádné relevantní stopy neobsahuje a v experimentu tak nebyl použit.

3.2.3 Ex1: rozlišitelnost VEP

VEPs, neboli Visually Evoked Potentials, jsou technikou založenou na detekci reakce mozku na nějaký externí vizuální stimul. V rámci tohoto experimentu bylo zkoumáno, jak dobře jsou klasifikovatelné úkoly na základě změny vizuální informace zpracovávané mozkiem při otevřených, zavřených a mrkajících očích.

- Hypotéza: půjde o snadno rozlišitelné úkoly vlivem artefaktů mrknutí a přítomností *alpha* vlny při zavřených očích.
- Zkoumané mentální úkoly: `baseline`, `closed`, `blinks`
- Použité datasety: `user_k`, `user_m1`, `user_m2`, `user_m3`, `user_u`, `user_v`
- Poznámka: Úspěšnost klasifikace je vyhodnocována při použití úkolu `blinks` a bez něj zvažt. Datasety `user_m3` a `user_k` neobsahují úkol `blinks`, použity jsou pouze v části experimentu bez něj.

Experiment byl proveden ve variantě včetně úkolu `blinks` a ve variantě bez něj. U většiny experimentů nejsou stopy `closed` díky kontaminaci skrze provádění úkolů pomocí techniky MI „čistě“, zároveň je složení datasetu značně nevyrovnané ve prospěch úkolu `closed`. Přesnost klasifikace tedy přesně neodráží výkon klasifikátoru, v tabulce výsledků Ex1 3.2 je proto místo přesnosti uvedena metrika váženého `f1-score` ve tvaru `test / real`.

V příložených archivech `ex1_blinks` a `ex1_noblinks` jsou výsledky experimentu podrobně zaznamenány včetně matic zmatení a ostatních standardních metrik.

- Úspěšnost klasifikace: vzhledem k porovnávání dvoutřídní a třítřídní klasifikace je nižší výkon druhého experimentu očekávatelný. Ukazuje se, že při přítomnosti úkolu `Blinks` je pro klasifikátor problematické rozlišit klidový stav od zavřených očí (klasifikátor se přeučí na úkol `closed`). Dále bylo zjištěno, že pro spolehlivé rozlišení úkolu `blinks` dostačují dvě stopy v rámci trénovacích dat.
- Analýza příznaků bez `blinks`: dva příznaky (`ar-coefficient` a `augmented-dickey-fuller`) byly vybrány u všech šesti datasetů. Alespoň pěti datasety bylo vybráno dalších 10 příznaků, mezi nimi zejména příznaky popisující autokorelaci. Alespoň jednou bylo vybráno 360 příznaků.
- Analýza příznaků s `blinks`: 200 příznaků bylo vybráno při analýze všech čtyř datasetů. Alespoň jednou bylo vybráno 390 příznaků.

Z prvního experimentu lze vyvodit dvě základní zjištění: úkol `blinks` je sice velmi dobře použitelný pro dvoutřídní klasifikaci, při použití více tříd je ale lepší jej nepoužívat. Zároveň bylo zjištěno, že ze všech 794 příznaků je jich více než polovina natolik irelevantních, že nebyly vybrány ani jednou.

3.2.4 Ex2: rozlišitelnost MI

Motor imagery je technikou mentálních úkolů, při kterých uživatel simuluje nějakou mentální činnost. Typicky se jedná o simulaci pohybu nějaké končetiny nebo vizualizaci či artikulaci nějakého slova nebo souvisejícího předmětu. Pokud by bylo možné tyto dvě třídy rozlišovat s použitím jednokanálového BCI ideálně při zavřených očích, bylo by možné sestrojít trojtřídní BCI a tím i použitelné ovládání robota. Rozlišitelnost úkolů těchto dvou typů je zkoumána v rámci tohoto experimentu.

- Hypotéza: Bude možné od sebe odlišit úkoly typu Mental Imagery a úkoly typu Word Imagery, jelikož tyto úkoly aktivují různá mozková centra a produkují různé tvary EEG vlny. Rozdíl by měl být rozeznatelný i při značné vzdálenosti Fp1 pozice od obou center.
- Zkoumané mentální úkoly: `motor_imagery`, `word_imagery`, `baseline`
- Použité datasety: `user_k`, `user_l`, `user_m1`, `user_t`, `user_u`, `user_v`
- Poznámka: Mentální úkoly označené prefixem `c_` byly prováděny při zavřených očích.

Úspěšnost klasifikátorů na trénovacích datech je stále ucházející, trvanlivost klasifikátoru je nicméně slabá. Jak je patrné z matic zmatení, lepšího a vyrovnanějšího skóre bylo dosaženo u datasetu `user_u`, ve kterém byly úkoly prováděny se zavřenýma očima. To může souviset s odstřížením vizuálních podnětů a tím i větší soustředěnosti uživatele na provádění mentálních úkolů, nicméně vzhledem k nedostatku dat toto nelze konstatovat obecně.

Pro dataset `user_t` bylo vybráno pouze 17 příznaků, což se pravděpodobně negativně projevilo na trvanlivosti klasifikátoru. Žádný příznak nebyl vybrán vždy, 3 příznaky byly vybrány čtyřikrát a 32 příznaků třikrát. Zajímavé příznaky jsou `change-quantiles` (vybrány pro čtyři datasety) a `welch-density` (vybrané `user_t` a zároveň ostatními datasety).

Spolehlivá rozlišitelnost dvou základních technik MI tedy potvrzena nebyla. Z univerzálních příznaků lze odvodit, že důležité vlastnosti signálu popisují jiné příznaky než u předchozího experimentu.

3.2.5 Ex3: kombinace úkolů

V rámci tohoto experimentu byla provedena klasifikace celých původních datasetů, které obvykle obsahují čtyři až pět různých mentálních úkolů. Smyslem tohoto experimentu bylo identifikovat případně skutečně univerzální příznaky napříč všemi datasety.

- Hypotéza: Vzhledem k velkému počtu tříd bude dosaženo horších výsledků.
- Zkoumané mentální úkoly a datasety: všechny

Z výsledků vyplývá, že ve většině případů dojde u klasifikátoru na reálných datech k přeučení na dvě třídy, do kterých jsou následně zařazeny téměř všechny výsledky. Úspěšnost klasifikace samotných tříd přitom vůbec nemusí být vysoká, viz např. matici zmatení datasetů `user_k` nebo `user_u`. Analýza společných vybraných příznaků přinesla zjištění, že téměř padesát příznaků bylo vybráno u všech devíti datasetů.

Jako závěr experimentu lze formulovat tvrzení, že více než dvoutřídní klasifikace nebyla u žádného datasetu úspěšná na reálných datech.

3.2.6 Výsledky experimentů

Základním zjištěním učiněným v experimentální části je nízká trvanlivost klasifikátorů, tedy jejich přeučení na konkrétní časový úsek a nemožnost použít je mimo něj. Nízká trvanlivost byla sice od začátku očekávána (již v návrhu bylo počítáno s tím, že uživatel bude muset při každém použití BCI znovu nahrávat trénovací data), nicméně nepoužitelnost BCI rozlišujícího více než dvě třídy po deseti minutách je jistým zklamáním.

Nabízí se otázka, jestli tyto výsledky nejsou způsobeny nezkušeností uživatelů: data byla sbírána od uživatelů, kteří před prvním měřením neměli žádnou zkušenost s prováděním mentálních úkolů—možným vysvětlením slabých výsledků klasifikace je i jejich nezkušenost a nepřesnost či nevědomá nedůslednost při provádění mentálních úkolů. Zároveň nebylo nasbíráno dostatečné množství dat pro to, aby bylo možné nízkou trvanlivost klasifikátorů jednoznačně potvrdit. Je možné, že dobře klasifikovatelné mentální úkoly pro BCI na Fp1 pozici existují, v rámci této práce se je však nalézt nepodařilo.

Závěr

V rámci bakalářské práce jsem se zabýval implementací jednokanálového BCI na bázi EEG signálu. Hlavním cílem práce bylo realizovat ovládání robota myšlenkou skrze klasifikaci EEG signálu. Zjistil jsem, že ačkoli se jedná o již dlouho známý problém v oblasti zájmu mnoha badatelů, masivní rozšíření jeho aplikací do praxe zatím nenastalo.

Svou prací jsem učinil pokus o implementaci takového praktického nasazení. Pomocí existujících knihoven jsem propojil zařízení Neurosky Mindwave Mobile 2 a Parrot Mambo Mission s vlastní aplikací v jazyce Python. Na základě rešerše jsem navrhl způsob zpracování EEG signálu, pomocí explorativní analýzy vybral vhodný klasifikátor a klasifikaci pak prakticky implementoval. Také jsem naměřil EEG vlny šesti různých uživatelů, kteří prováděli různé mentální úkoly, s cílem vybrat z nich užitečné způsoby upírání pozornosti pro klasifikaci EEG signálu. V rámci implementace jsem vyvinul aplikaci schopnou nahrávat a klasifikovat EEG data. Aplikaci jsem propojil s experimentálním ovládáním minidronu, pomocí kterého je při použití vhodných mentálních úkolů možné myšlenkou přenést předmět z bodu A do bodu B.

Hlavním přínosem mojí práce je aplikace pro nahrávání a experimentaci s klasifikovatelností EEG vln. Dalším přínosem je zjištění, že EEG vlny sice je možné dobře klasifikovat i do většího množství tříd, trvanlivost takové klasifikace je však velmi krátká. Dostatečně trvanlivých výsledků klasifikace jsem dosáhl při klasifikaci dvou mentálních úkolů.

Navázat na tuto práci je možné například vylepšením extrakce příznaků, která v momentální podobě představuje úzké hrdlo aplikace vzhledem ke své výpočetní náročnosti. Zároveň lze pokračovat cíleným hledáním kvalitních mentálních úkolů nebo souvislostí mezi vybíranými příznaky EEG vln.

Literatura

- [1] Ullah, K.; Ali, M.; Rizwan, M.; aj.: Low-cost single-channel EEG based communication system for people with lock-in syndrome. In *2011 IEEE 14th International Multitopic Conference*, Karachi, Pakistan: IEEE, 2011, ISBN 978-1-4577-0657-8, ISSN 10.1109/INMIC.2011.6151455, s. 120–125, doi:10.1109/INMIC.2011.6151455. Dostupné z: <http://ieeexplore.ieee.org/document/6151455/>
- [2] Teplan, M.: Fundamental of EEG Measurement. *MEASUREMENT SCIENCE REVIEW*, ročník 2, 01 2002.
- [3] Wang, Y.; Gao, X.; Hong, B.; aj.: Brain-Computer Interfaces Based on Visual Evoked Potentials. *IEEE Engineering in Medicine and Biology Magazine*, ročník 27, č. 5, 2008: s. 64–71, ISSN 0739-5175, doi:10.1109/MEMB.2008.923958. Dostupné z: <http://ieeexplore.ieee.org/document/4625401/>
- [4] Millan, J. R.; Renkens, F.; Mourino, J.; aj.: Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, ročník 51, č. 6, June 2004: s. 1026–1033, ISSN 0018-9294, doi:10.1109/TBME.2004.827086.
- [5] Bhattacharyya, S.; Shimoda, S.; Hayashibe, M.: A Synergetic Brain-Machine Interfacing Paradigm for Multi-DOF Robot Control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, ročník 46, č. 7, July 2016: s. 957–968, ISSN 2168-2216, doi:10.1109/TSMC.2016.2560532.
- [6] R, S. S.; Rabha, J.; Nagarjuna, K. Y.; aj.: Motor Imagery EEG Signal Processing and Classification Using Machine Learning Approach. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Oct 2017, s. 61–66, doi:10.1109/ICTCS.2017.15.
- [7] contributors, W. C.: Electrode locations of International 10-20 system for EEG (electroencephalography) recording. 2015. Dostupné z: <https://www.10-20.org/>

- [//commons.wikimedia.org/w/index.php?title=File:21_electrodes_of_International_10-20_system_for_EEG.svg&oldid=169572190](https://commons.wikimedia.org/w/index.php?title=File:21_electrodes_of_International_10-20_system_for_EEG.svg&oldid=169572190)
- [8] MindWave Diagram. 2011. Dostupné z: <http://support.neurosky.com/kb/mindwave/mindwave-diagram>
- [9] McGovern, A.: Pyparrot: Python interface for Parrot drones. 2017. Dostupné z: <https://github.com/amymcgovern/pyparrot>
- [10] Christ, M.; Braun, N.; Neuffer, J.; aj.: Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh A Python package). *Neurocomputing*, 05 2018, doi:10.1016/j.neucom.2018.03.067.
- [11] Maximilian, C.; W., K.-L.; Feindt, M.: Distributed and parallel time series feature extraction for industrial big data applications. 2016.
- [12] Chang, C.-C.; Lin, C.-J.: LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, ročník 2, 2011: s. 27:1–27:27, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [13] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.
- [14] Schirrmeister, R. T.: Python-mindwave-mobile. 2019. Dostupné z: <https://github.com/robintibor/python-mindwave-mobile>
- [15] ThinkGear Serial Stream Guide. 2017. Dostupné z: http://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_protocol

Seznam použitých zkratk

BCI Brain Computer Interface

BLE Bluetooth Low Energy

CLI Command Line Interface

EEG Elektroencefalografie

VEP Visually Evoked Potentials

Obsah přiložené SD karty

readme.txt	stručný popis obsahu SD karty
app	adresář se spustitelnou formou implementace
src	
├─ data	data naměřená v rámci práce
├─ jupyter	adresář s jupyter notebooky
├─ thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
text	text práce
├─ thesis.pdf	text práce ve formátu PDF