

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## Diplomová práce

**Ustálené vizuální evokované  
potenciály (SSVEP) a jejich  
aplikace v systému BCI**

# Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 9. května 2012

Martin Fornbaum

# Abstract

This dissertation deals with SSVEP application in the field of BCI. There are analyzed methods of measure brain waves, the principles of evoked potentials and how we can process EEG data in the first part. The next section is designed the BCI on the principle of SSVEP. It is about the control of robotics car which turned there where the user concentrates their attention. It was necessary to take into consideration principles of the brain and the individual differences of each person during the measuring and design modules of the system by sufficiently robust manner. Implementation of modules needed for the BCI is described in subsequent chapters. This includes modules for retrieval of the data, processing the EEG and classification. The last part of the dissertation includes an validation of the individual functions of each modules and the evaluation of the results.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Elektroencefalografie</b>	<b>2</b>
2.1	Elektroencefalograf . . . . .	2
2.1.1	Elektrody . . . . .	3
2.2	Snímání signálu EEG . . . . .	4
2.2.1	Frekvence v EEG záznamu . . . . .	4
2.2.2	Artefakty . . . . .	8
<b>3</b>	<b>Evokované potenciály</b>	<b>9</b>
3.1	Princip vzniku evokovaného potenciálu . . . . .	9
3.2	Vizuální evokované potenciály . . . . .	10
3.2.1	Stimulace zrakové dráhy . . . . .	10
3.2.2	Ustálené vizuální evokované potenciály . . . . .	13
3.3	Sluchové evokované potenciály . . . . .	14
3.3.1	Faktory ovlivňující záznam . . . . .	15
3.4	Somatosenzorické evokované potenciály . . . . .	16
3.5	Motorické evokované potenciály . . . . .	16
3.6	Endogenní evokované potenciály . . . . .	17
3.6.1	Stimulace . . . . .	17
<b>4</b>	<b>Zpracování EEG dat</b>	<b>18</b>
4.1	Základní zpracování signálu . . . . .	18
4.1.1	Vzorkování signálu . . . . .	18
4.1.2	Zpracování kanálů . . . . .	20
4.2	Fourierova transformace . . . . .	21
4.2.1	Diskrétní Fourierova transformace . . . . .	21
4.2.2	Rychlá Fourierova transformace . . . . .	23
4.2.3	Krátkodobá Fourierova transformace . . . . .	24

<b>5</b>	<b>Klasifikace dat</b>	<b>25</b>
5.1	Příznakové rozpoznávání . . . . .	25
5.1.1	Diskriminační funkce . . . . .	26
5.1.2	Příznakový vektor . . . . .	26
5.1.3	Klasifikátory . . . . .	28
<b>6</b>	<b>Návrh a implementace BCI</b>	<b>31</b>
6.1	Součásti BCI . . . . .	31
6.2	Modul pro stimulaci . . . . .	32
6.2.1	Popis programu Presentation . . . . .	34
6.2.2	Požadavky na scénář . . . . .	36
6.2.3	Implementace scénáře . . . . .	37
6.3	Modul pro načítání dat EEG . . . . .	40
6.3.1	Načítání ze souboru . . . . .	40
6.3.2	Načítání v reálném čase . . . . .	42
6.4	Modul pro zpracování dat . . . . .	45
6.4.1	Uložení dat . . . . .	45
6.4.2	Transformace kanálů . . . . .	45
6.4.3	DFT a FFT . . . . .	46
6.5	Modul pro klasifikaci . . . . .	47
6.5.1	Implementace klasifikátoru . . . . .	47
6.6	Modul pro výběr vhodné stimulační frekvence . . . . .	50
6.6.1	Algoritmus výběru frekvencí . . . . .	51
6.7	Modul ovládání modelu auta . . . . .	51
6.8	Demonstrační programy . . . . .	51
6.8.1	Program pro načítání dat ze souboru . . . . .	53
6.8.2	Program pro real-time komunikaci . . . . .	56
<b>7</b>	<b>Ověření funkčnosti modulů</b>	<b>62</b>
7.1	Načítání EEG dat . . . . .	62
7.2	Zpracování EEG dat . . . . .	62
7.2.1	FFT a DFT . . . . .	63
7.3	Klasifikace . . . . .	64
7.4	Celkový test . . . . .	66
7.4.1	Popis experimentu . . . . .	66
7.4.2	Zpracování dat . . . . .	66
7.4.3	Závěr experimentu . . . . .	70
<b>8</b>	<b>Závěr</b>	<b>71</b>

# 1 Úvod

Již od vynalezení prvních strojů bylo zapotřebí je nějak ovládat. Po mnoho století se jednalo o ovládání čistě manuální. Příchod počítačů však tento způsob změnil a začaly se vymýšlet nové způsoby pro ovládání všech možných zařízení. První myšlenka, která většinu lidí napadne, je ovládat stroje přirozenou řečí. Komu však tohle nestačí, může v myšlenkách plout do vod méně probádaných, čímž je například ovládání strojů přímo pomocí mozku.

Ovládání zařízení pomocí mozku může znít trochu jako ze sci-fi knihy, ale pravdou je, že některé z těchto systémů jsou již spolehlivé a připravené k reálnému nasazení. Samozřejmě, že se nejedná o řízení věcí myšlenkou, ale jsou pro to využívány známé principy fungování mozku. Všechny tyto systémy jsou souhrnně označovány zkratkou BCI (brain computer interface).

Cílem této práce je navrhnout systém BCI pro ovládání robotického vozítka pomocí ustálených vizuálních evokovaných potenciálů. Pro vytvoření celého systému je zapotřebí prostudovat principy a způsob zpracovávání ustálených vizuálních evokovaných potenciálů a následně naimplementovat potřebné moduly.

Podobné systémy na těchto principech na své reálné využití zatím čekají, ale lze předpokládat, že postupem času budou BCI systémy běžným pomocníkem pro tělesně hendikapované. Problémem zde stále zůstává nutnost přítomnosti elektrod na skalpu, což výrazně ovlivňuje pohodlnost použití.

## 2 Elektroencefalografie

Elektroencefalografie patří k základním metodám pro vyšetření mozku. První zachycení elektrické aktivity mozku bylo provedeno už na konci 19. století, přičemž založení skutečné elektroencefalografie se odehrálo ve 20. letech 20. století. V průběhu dalších let došlo k velkému rozvoji elektroencefalografie a ve 30. letech už byly známy základní patologie. Postupem času se vyvíjela zařízení pro měření EEG, zejména s rozvojem výpočetní techniky přišly velké změny, nicméně základní křivka zůstává vizualizována stejným způsobem jako na počátku.

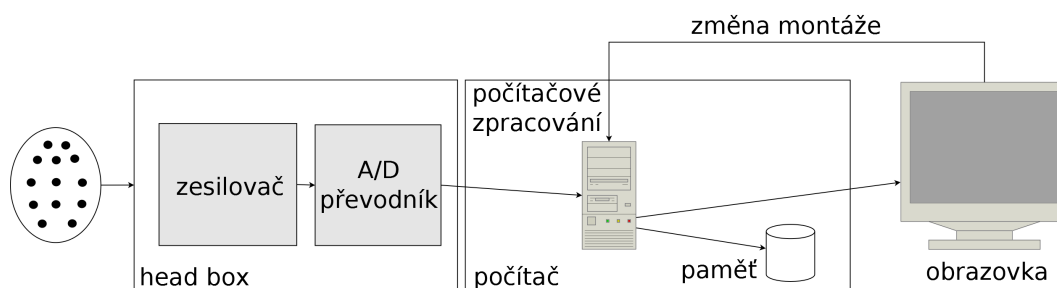
Toto vyšetření patří k tzv. neinvazivním vyšetřením a používá se jako základní vyšetření všech poruch mozku v neurologii a ve většině případů i v psychiatrii. Záznam zachycený při elektroencefalografii se nazývá elektroencefalogram (EEG) a odráží se v něm mozková aktivita - různé fáze spánku a stavy vědomí, projevy metabolických poruch, vlivy drog či toxických látek. [8]

### 2.1 Elektroencefalograf

Elektroencefalograf je přístroj sloužící k získání EEG. Jelikož jeho vývoj začal již na konci 19. století, nemohla být využita číslicová technika a jednalo se o přístroj plně analogový. Až v 80. letech přišel nový model, který se používá dodnes.

Analogové přístroje využívaly k zobrazování osciloskopickou obrazovku s jedním a později i více kanály. Záznam probíhal na papír přímo písíci galvanometry s inkoustem.

Číslicový elektroencefalograf je rozdělen na dvě hlavní části (obrázek 2.1). V tzv. head boxu je celá vlastní elektronika a bývá umístěn co nejbližší k hlavě pacienta. V této části dochází k zesílení signálu, filtraci a k digitalizaci signálu. Další část je samotný počítač, ve kterém dochází k zobrazování dat pomocí softwaru od výrobce.



Obrázek 2.1: Schéma elektroencefalografu

### 2.1.1 Elektrody

Pro různé typy vyšetření jsou používány různé druhy elektrod. Elektrody lze například rozdělit takto:

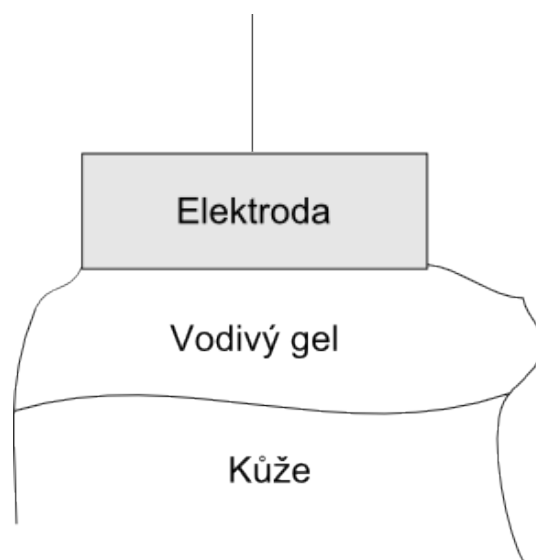
- Povrchové - kovové deskové elektrody nebo suché elektrody (izolované/neizolované);
- Podpovrchové - jehlové elektrody vpichové nebo implantabilní elektrody;
- Mikroelektrody - kapilární elektrody pro snímání z buněk nebo implantabilní elektrody (snímání z mozku);

Jak již bylo řečeno, elektroencefalografie je neinvazivní vyšetření, tudíž se používají povrchové elektrody. Elektrody musí být nepolarizovatelné, čemuž odpovídají vzácné kovy (zlacené elektrody) nebo nejčastěji používané elektrody typu AgCl, které se skládají z vrstvy čistého stříbra pokrytého tenkou vrstvou chloridu stříbrného. Tyto elektrody se musí kombinovat s vodivými gely. [4]

#### Vodivé pasty a gely

Pro dobré vodivé spojení mezi elektrodou a tělem se používají vodivé gely, které vyplní prostor mezi elektrodou a tělem pacienta (obrázek 2.2).





Obrázek 2.2: Schéma elektroencefalografu

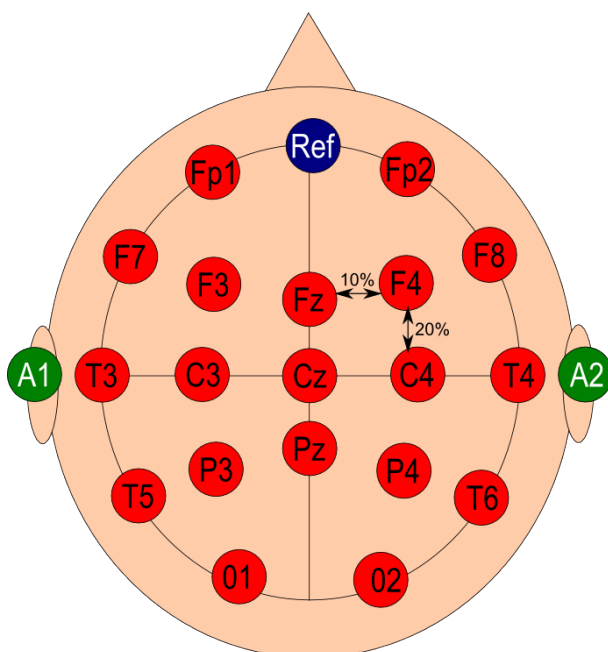
## 2.2 Snímání signálu EEG

Jelikož jsou elektrody přiloženy na lebku pacienta, jsou amplitudy mozkových vln tlumeny průchodem přes vrchní část lebky na úroveň řádově desítek mikrovolt. Dalším problémem je, že EEG signál vzniká jako vážená sumace velkého množství neuronů, takže nelze odlišit akční potenciál jednotlivých buněk. Důsledkem bývá na první pohled nepravidelný průběh, ve kterém jsme schopni pozorovat pouze signály s určitou periodou. [2]

Rozložení elektrod na povrchu lebky vyplývá z definovaných míst na lebce a následném rozdělení všech vzdáleností po 10 % a 20 %. Tento způsob rozložení elektrod se nazývá systém 10-20 (obrázek 2.3) a definuje umístění 19-ti základních měřících elektrod. Snímání větším počtem elektrod definuje systém 10-10 a pro experimentální účely lze použít až 128 elektrod. [8]

### 2.2.1 Frekvence v EEG záznamu

EEG signál je tvořen vlnami sinusového tvaru a dělí se podle frekvencí. Každá z těchto frekvencí se objevuje za určité situace a v závislosti na stavu subjektu. Nejčastěji se objevují mozkové vlny Delta, Theta, Alfa, Beta a Gama.



Obrázek 2.3: Systém rozložení elektrod 10-20

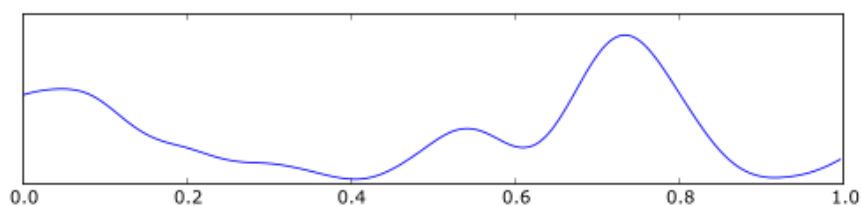
Každý typ mozkových vln má uvedený rozsah frekvencí, které se v různých zdrojích literatury nemusí shodovat, neboť konkrétní frekvence, na které se vlna vyskytuje se liší v závislosti na pacientovi a podmínkách. [1]

### Delta

Vlny delta jsou mozkové vlny o frekvencích menších než 4 Hz. Tyto vlny se běžně objevují u dospělých v nejhlubší fázi non-REM spánku nebo u dětí. Na EEG záznamu je lze poznat podle vysokých amplitud a nízké frekvence (viz obrázek 2.4). Ačkoliv jsou tyto frekvence známkou tvrdého spánku, ukázalo se, že někteří lidé jsou schopni se během meditace dostat do této fáze, ba co víc, dokonce v ní i komunikovat.

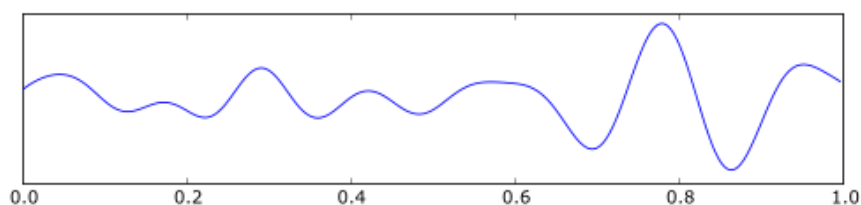
### Théta

Objevuje se v rozmezí mezi 4 - 7 Hz. Běžně může být pozorována u dětí do 13-ti let nebo u dospělých ve stavu ospalosti nebo po probuzení. Théta vlny



Obrázek 2.4: Ukázka delta vln

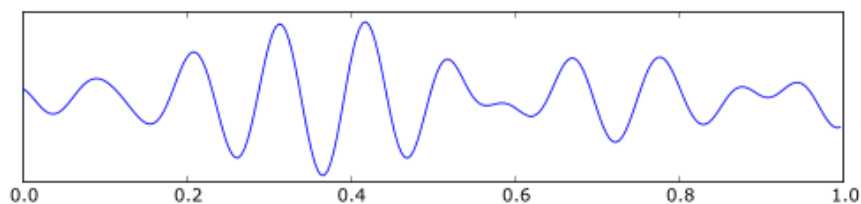
lze najít i během meditace. Dále jsou tyto frekvence spojovány s kreativitou a intuicí, ale také s denním sněním, úzkostí nebo depresemi (viz obrázek 2.5).



Obrázek 2.5: Ukázka théta vln

## Alfa

Vyskytuje se na frekvencích mezi 8 a 12Hz. Tuto aktivitu lze nalézt v zadních částech hlavy na obou stranách, nicméně amplitudy bývají větší v dominantní hemisféře mozku<sup>1</sup>. Alfa vlny se vyskytují v bdělém stavu při zavřených očích a během relaxace (viz obrázek 2.6).



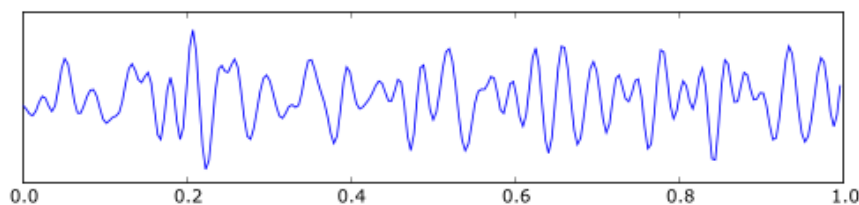
Obrázek 2.6: Ukázka alfa vln

<sup>1</sup>V mozkové kůře se nacházejí sídla vyšší nervové činnosti (řeč, logika...). Dominantní hemisférou mozku nazýváme hemisféru, ve které je těchto sídel buď více, nebo jsou více rozvinuté. U praváků zpravidla levá hemisféra a u leváků pravá.

## Beta

Beta vlny se vyskytují na frekvencích 12 - 30 Hz (viz obrázek 2.7). Převládají v normálním bdělém stavu a jsou známkou logicko-analytického myšlení. Jelikož beta vlny se nacházejí ve velkém rozsahu, jsou dále děleny na:

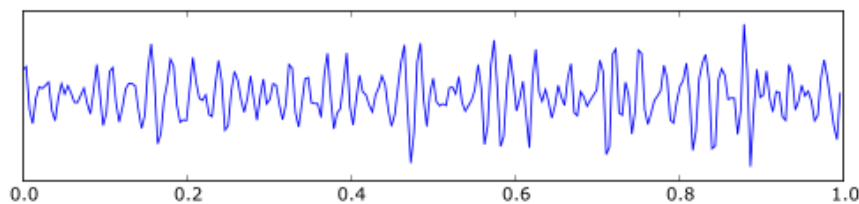
- Nízká beta (12 - 15 Hz) - Stav relaxovaného soustředění
- Střední beta (15 - 18 Hz) - Stav pohotovosti, nikoliv ale podrážděnosti
- Vysoká beta (nad 18 Hz) - Stav podrážděnosti a ostražitosti



Obrázek 2.7: Ukázka beta vln

## Gama

Gama se vyskytuje na frekvencích vyšších než 30 Hz (viz obrázek 2.8). Vyskytují se ve stavech extrémního vypětí, soustředěnosti nebo ve velkém stresu. Podílí se také na řízení mozkových procesů, fungování paměti a podporuje učení.



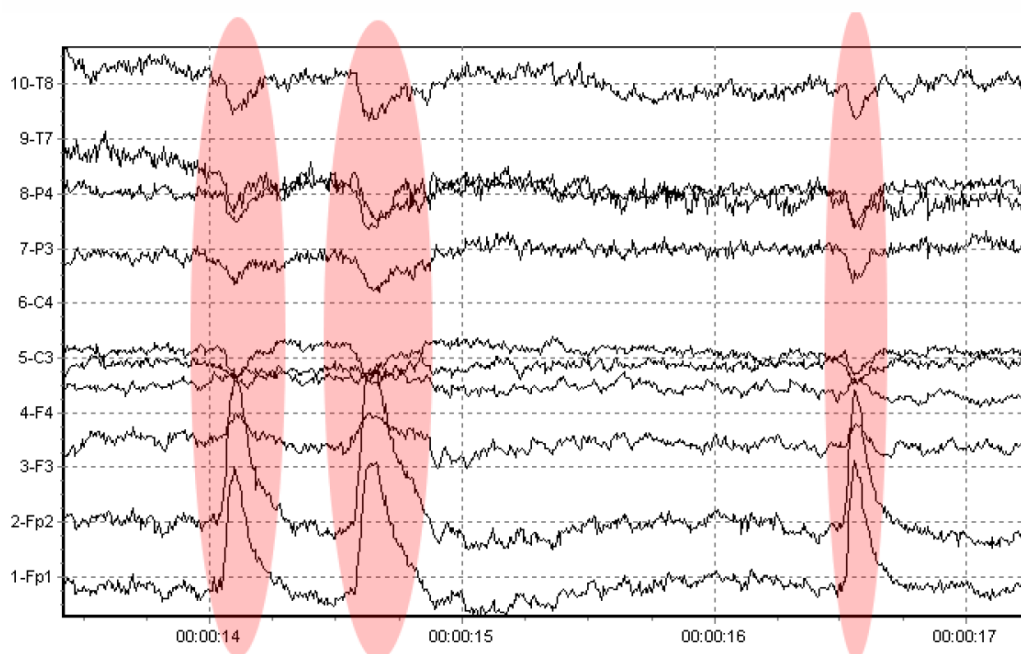
Obrázek 2.8: Ukázka gama vln

## 2.2.2 Artefakty

Kromě výše popsaných vln se mohou v záznamu vyskytovat ještě další vlnové tvary, které nazýváme artefakty. Základní dělení artefaktů je na biologické a technické.

### Biologické artefakty

Biologické artefakty jsou zapříčiněny pacientem (i neúmyslně). Jedná se například o mrkání, pohyby očí, polykání nebo třeba kašláni (obrázek 2.9).



Obrázek 2.9: Biologické artefakty způsobené mrkáním pacienta.

### Technické artefakty

Původem technických artefaktů jsou elektrody, přístroj nebo například vnější rušení. U elektrod se může jednat o vadnou elektrodu nebo špatný kontakt s lebkou pacienta. Přístroj může být poškozený nebo pokud je napájen ze sítě, bude se zcela jistě objevovat v datech vlnění o frekvenci 50 Hz.

## 3 Evokované potenciály

Vyšetření evokovaných potenciálů je nejmladší neurofyzilogická metoda, která se začala využívat v klinické praxi až v posledních dvaceti letech. Po druhé světové válce byly zaznamenány různé nepravidelnosti v EEG záznamu při různých činnostech pacientů. Tyto nepravidelnosti vyvolané na základě podnětu byly nazvány jako evokované potenciály. Jelikož je zde nutné provádět průměrování velkého počtu epoch, došlo k výraznému rozšíření této oblasti až s rozvojem výpočetní techniky, která tyto úkony dokázala provádět sama. Kapitola čerpá z [3].

Postupem času se začaly vyšetření standardizovat a dnes se v praxi používá několik druhů vyšetření:

- VEP (visual evoked potentials) - vizuální evokované potenciály
- BAEP (brainstem auditory evoked potentials) - kmenové sluchové evokované potenciály
- SEP (somatosensory evoked potentials) - somatosenzorické evokované potenciály
- MEP (motor evoked potentials) - motorické evokované potenciály
- ERP (endogenous event-related cognitive potentials) - endogenní kognitivní evokované potenciály

Jelikož se práce zabývá výhradně aplikací vizuálních evokovaných potenciálů, bude tato problematika popsána detailněji než ostatní druhy.

### 3.1 Princip vzniku evokovaného potenciálu

Evokovaný potenciál je reakce nervové soustavy na stimulaci receptorů. Tato odpověď může být zachycena kdekoliv mezi receptorem a korovým analyzátozem<sup>1</sup>. To znamená, že při stimulaci periferního nervu můžeme zachytit potenciál na končetině, na různých místech míchy a nakonec na skalpu.

---

<sup>1</sup>neurony kůry, seskupené do okřsků, v nichž je prováděna analýza a syntéza informací přicházejících z receptorů

Evokované potenciály jsou trifázického tvaru, který má tzv. P-N-P charakter. Znamená to, že isoelektrická linie se nejprve vychýlí do pozitivní deflexe, za kterou následuje dominantní negativní výchylka a následně další pozitivní. Tento tvar má evokovaná odpověď tehdy, pokud registrační elektroda je blíže ke generátoru než elektroda referenční.

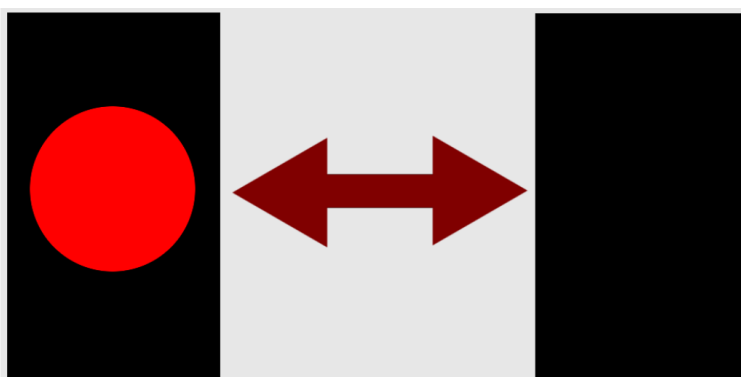
## 3.2 Vizuální evokované potenciály

Zrakový evokovaný potenciál vzniká současnou depolarizací neuronů v optických oblastech okcipitálního<sup>2</sup> laloku.

### 3.2.1 Stimulace zrakové dráhy

Pro stimulaci zrakové dráhy se používají dva základní způsoby. První z nich je stimulace zábleskem, která se využívá u nespolupracujících pacientů. Tato metoda spočívá ve změně jasu v zorném poli.

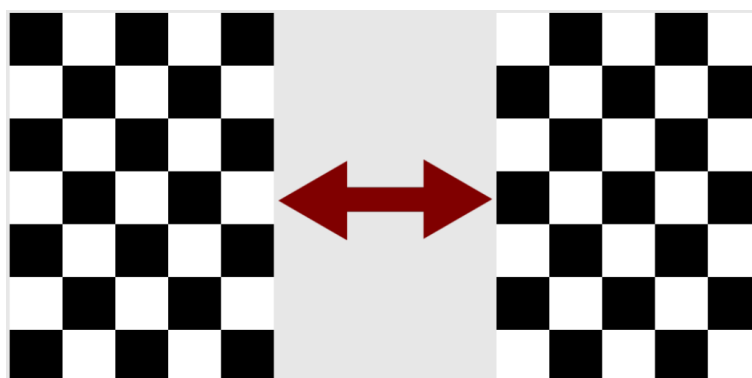
Druhý způsob je stimulace pomocí strukturovaného podnětu, který zpravidla obsahuje tmavé a světlé plochy. Jednou možností je stimulace obrazem, který se objevuje a mizí (obrázek 3.1), ale častěji používanou technikou je záměna inverzních vzorů (obrázek 3.2). V tomto případě zůstává celkový jas stejný a podnět vyvolává změny jasu v různých částech zorného pole.



Obrázek 3.1: Stimulace obrazem, který se objevuje a mizí

---

<sup>2</sup>týlního



Obrázek 3.2: Stimulace inverzním vzorem

### Technické parametry ovlivňující záznam

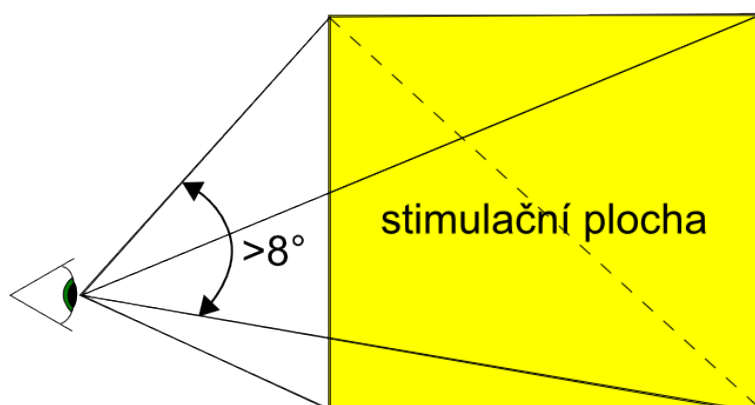
- Velikost stimulační plochy a prvků obrazce
- Luminance stimulační plochy
- Kontrast mezi prvky
- Frekvence stimulace

**Velikost stimulační plochy a prvků obrazce** je rozhodující pro výsledky měření. Lze stimulovat celé zorné pole pacienta (tzv. full-field stimulation) nebo polovinu zorného pole (half-field stimulation). Při stimulaci celým polem by měla být stimulační plocha větší než  $8^\circ$  (viz obrázek 3.3). Se zvětšováním stimulační plochy se mírně zvyšuje amplituda vlny P100.

Nejčastěji je pro stimulaci používán vzor horizontálně nebo vertikálně se střídajících pruhů nebo čtverce tvořící šachovnici. Velikost čtverců v šachovnici je rozhodující faktor. Při zmenšování čtverců šachovnice se zmenšují amplitudy a prodlužují latence vln. Pokud se naopak výrazně zvětší čtverce, stává se dominantní složkou stimulace jas a záznam se přibližuje tvaru sledovaného při stimulaci zábleskem.

**Luminance stimulační plochy** ovlivňuje záznam tak, že při jejím snížení se zmenšují amplitudy a prodlužují latence vln. Snížení průměrné luminance o jednu logaritmickou jednotku zmenší amplitudu o 18% a prodlouží latenci o 18 ms. [3]





Obrázek 3.3: Velikost stimulační plochy pro full-field stimulaci

**Kontrast** mezi světlými a tmavými prvky výrazně ovlivňuje záznam. Vyjadřuje se v procentech a lze ho vypočítat jako

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \cdot 100 \quad (3.1)$$

přičemž  $L_{max}$  je luminance světlých prvků a  $L_{min}$  luminance tmavých prvků. Zvyšování kontrastu až do pásma saturace (mezi 20-40%) vede ke zvyšování amplitud a snižování latence vln. Další zvyšování kontrastu zvyšuje amplitudy jen nepatrně.

Byla provedena vyšetření i s jinou barvou než černá a bílá (např. červená-černá, zelená-červená), ale záznam byl prakticky stejný. Při použití různých barev na stimulaci ustálených vizuálních evokovaných potenciálů se výsledky liší (viz dále).

**Frekvence stimulační** se určuje jako počet změn stimulačního obrazce za sekundu a udává se v hercích (Hz). Standardně používaná frekvence je 2 Hz, kdy lze vyhodnocovat jednotlivé odpovědi na stimuly. Při zvyšování frekvence nad 5 Hz začínají vlny interferovat a vytváří tak sinusový průběh, který frekvenčně odpovídá stimulaci. Tento typ záznamu se nazývá steady-state VEP (SSVEP, ustálené vizuální evokované potenciály).

### Biologické faktory ovlivňující záznam

- Věk

- Zraková ostrost
- Tělesná teplota a pohlaví

**Věk** pacienta ovlivňuje záznam výrazněji, pokud jsou používány nižší luminance a menší velikosti čtverců. Asi do 20 let dochází ke zkracování latence P100 a naopak po 50. roku se latence zvětšuje.

**Zraková ostrost** opět ovlivňuje latenci a amplitudu vln. Čím je menší zraková ostrost, tím jsou amplitudy P100 nižší a latence delší.

**Tělesná teplota** nemá prokazatelně vliv na VEP, nicméně je to jeden z faktorů, který pravděpodobně může za rozdíly mezi záznamem u mužů a žen.

**Pohlaví** ovlivňuje jak amplitudu tak latenci vln. Ženy mají průměrně nižší latenci P100. Jako důvod se uvádí vyšší tělesná teplota a menší hlava, tudíž i kratší zraková dráha. Amplituda P100 je u žen vyšší než u mužů. Jako důvod se udávají hormonální rozdíly.

### **3.2.2 Ustálené vizuální evokované potenciály**

Jak již bylo řečeno, ustálené vizuální evokované potenciály (SSVEP) jsou speciálním případem VEP, kdy je sítnice buzena v periodických cyklech rychle za sebou, tudíž nejsou evokovány jednotlivé vlny, ale vytváří se periodické sinusové vlnění o frekvenci stejné, jakou je buzena sítnice.

#### **Stimulace**

Způsoby stimulace zrakové dráhy byly popsány výše. Jelikož u SSVEP jsou třeba vyšší frekvence pro stimulaci, plyne z toho omezení zejména pro stimulaci pomocí monitoru.

U stimulace pomocí monitoru narážíme na problém, že většina monitorů má obnovovací frekvenci 60 Hz, takže je tu velká limitace, která se týká

maximální frekvence. Pokud se tedy bude střídát pozadí s nějakým obrazcem, dosáhneme frekvence maximálně 30 Hz.

U stimulace pomocí diody může být naopak problém s požadavkem na velikost stimulační plochy.

### **Frekvence**

Frekvence, kterými je třeba pro SSVEP stimulovat se pohybují mezi 5 Hz a 75 Hz, přičemž jsou rozděleny do tří pásem. Nízké se uvádějí v rozmezí do 12 Hz, střední od 12 Hz do 30 Hz a vysoké nad 30 Hz.

Podle [9] nejvyšší amplitudy jsou kolem 10 Hz a dále mezi 16 - 18 Hz. Ve vysokém pásmu jsou amplitudy nižší. Při volbě frekvence je třeba vyhnout se frekvencím, na kterých má pacient pásmo alfa vln, neboť ty se mohou v záznamu objevovat běžně a mohly by být zaměněny s odpovědí na stimulaci.

### **Barvy**

U stimulace klasických VEP bylo řečeno, že barva nemá na záznam žádný vliv. To ale neplatí pro SSVEP, neboť v [9] se uvádí, že barvy mají vliv na stimulaci pro různé frekvence. Červená barva má silné odezvy kolem 11 Hz, ale v okolí se rychle zmenšují. Oproti tomu modrá barva má na 13 Hz slabší odezvu než barva červená, nicméně není tolik závislá na použité frekvenci. Žlutá barva má opět nižší amplitudu a menší závislost na zvolené frekvenci.

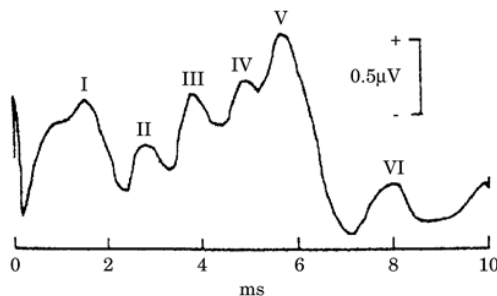
## **3.3 Sluchové evokované potenciály**

Sluchové evokované potenciály jsou elektrické odpovědi na stimulaci zvukovým podnětem. Podle latence je lze rozdělit na kmenové sluchové evokované potenciály (BAEP) s latencí do 10 ms, sluchové evokované potenciály se střední latencí 10 - 50 ms od podnětu, pomalé korové odpovědi (50 - 300 ms) a pozdní korové odpovědi s ještě větší latencí.

V praxi se při vyšetření využívají jen BAEP, ostatní jsou využívány pro experimentální účely.

Pro stimulaci sluchové dráhy se převážně používá cvaknutí. Méně používaný je tón, kde kromě intenzity je třeba ještě určit frekvenci a délku trvání.

Při měření BAEP se v prvních 10 ms nachází pět až sedm vln, které jsou označovány římskými číslicemi I - VII (obrázek 3.4). Generátor vlny I je periferní část sluchového nervu. U ostatních vln nejsou generátory známy.



Obrázek 3.4: Kmenové sluchové evokované potenciály

### 3.3.1 Faktory ovlivňující záznam

Stejně jako u VEP jsou zde biologické a technické faktory ovlivňující záznam. Mezi biologické se zde řadí věk, teplota a pohlaví. Z technických faktorů záznam ovlivňuje frekvence stimulace a intenzita.

Je prokázáno, že děti mají větší amplitudy vln při vyšetření BAEP a odlišné poměry amplitud I a V v porovnání s dospělými. Velmi malé děti mají vyšší latence vln a k hodnotám dospělých se dostávají kolem 1. až 2. roku.

Pohlaví má vliv na latenci i amplitudy vln. Ženy mají nižší latenci vln dokonce i po odečtení vlivu velikosti hlavy.

Zde má i tělesná teplota vliv na záznam, jak bylo dokázáno během operací srdce při umělém podchlazení. Při nižší teplotě se prodlužují latence BAEP, a to exponenciálně. Amplitudy vln se zvyšují do 28°C a dále pak lineárně klesají.

Při zvyšování frekvence stimulace narůstá latence jednotlivých vln a amplituda se zvětšuje. Při vyšších frekvencích mohou být některé vlny špatně

rozlišitelné.

Intenzita stimulace má zásadní vliv na BAEP. Při vyšší intenzitě se snižují latence a zvyšují amplitudy. Vyšší vliv má zvyšování intenzity na vlnu I než na vlnu V.

### **3.4 Somatosenzorické evokované potenciály**

SEP vznikají jako reakce na stimulaci příslušných nervů. Standardně se používá stimulace pomocí elektrického stimulu. Intenzita stimulu je u každého jedince individuální, ale je třeba, aby došlo k podráždění nervové dráhy. Zde záleží na tom, jaký nerv je stimulován. V případě n. medianus (středový nerv ruky) se za běžných okolností intenzita stimulu pohybuje okolo 2 - 8 mA. Pokud stimulujeme n. tibialis (holenní nerv), je to obvykle 4 - 8 mA. Obecně intenzita neovlivňuje tvar evokovaných odpovědí.

Frekvence stimulu na rozdíl od intenzity má vliv na záznam. Čím větší frekvence stimulace, tím menší amplitudy vln jsou v záznamu. Běžně používané frekvence jsou 3 - 5 Hz pro n. medianus a 1 - 5 Hz pro n. tibialis. Délka vyšetření pro obě končetiny se stimulací běžnou frekvencí se pohybuje okolo 45 - 60 minut, proto se nižší frekvence používají jen vyjíměčně pro experimentální účely.

### **3.5 Motorické evokované potenciály**

Tento druh evokovaných potenciálů se výrazně liší od všech předcházejících. U předcházejících probíhalo měření tak, že byly stimulovány periferie a záznam byl odečítán na skalpu. U MEP je to přesně naopak. Stimuluje se mozek nebo mícha a evokované potenciály se registrují na periférii.

Stimulace může probíhat buď invazivně nebo neinvazivně. Mezi neinvazivní metody patří magnetická a elektrická stimulace. Invazivní stimulace míchy a mozku je využívána během operací, zejména neurochirurgických nebo ortopedických.

Měření motorických evokovaných potenciálů může přispět k diagnostice různých neurologických onemocnění.

## **3.6 Endogenní evokované potenciály**

ERP (event-related potentials) jsou evokované potenciály vázány na událost. Pokud subjekt očekává nějaký podnět, který přijde, je amplituda vlny vyšší než u podnětu neočekávaného nebo nezájímavého. Nejrozšířeněji používaný potenciál je vlna P3 (někdy označovaná jako P300).

Typický průběh má charakter N-P-N-P-N, kde lze identifikovat vrcholy N1, P2, N2, dále nejvýraznější P3 a občas také N4. Při stimulaci vizuálním podnětem se ještě navíc vyskytuje vlna P1, která je podobná vlně P100 zaznamenávané u VEP. Dále při vizuální stimulaci je vyšší latence vlny P3.

### **3.6.1 Stimulace**

Tento druh evokovaných potenciálů lze stimulovat různými způsoby. Podněty mohou být vizuální, sluchové, somatosenzorické nebo i kombinované.

U stimulace sluchovým podnětem se zpravidla používá detekce dvou druhů tónu. Jeden má zpravidla frekvenci 1000 Hz, druhý 2000 Hz a jsou prezentovány v náhodném pořadí v poměru 4:1. Nižší tón se nazývá maskovací a vyšší tón je nazýván terčový. Subjekt je instruován, aby terčové podněty rozpoznal, případně je i počítal.

U vizuální stimulace je použit stejný postup jako u sluchové. Opět je zde maskovací a terčový podnět. Jejich trvání je obvykle 200 ms a objevují se v poměru 5:1.

## 4 Zpracování EEG dat

V této kapitole budou popsány postupy zpracovávání EEG signálu, které jsou využívány při hledání ustálených vizuálních evokovaných potenciálů. [6] Existuje nespočet metod zpracování signálu jako takového a většina z nich má široké uplatnění i za hranicemi oblasti EEG. Postupy, které jsou zde popsány, lze využít na libovolný signál nebo dokonce pro operace s obrazovými daty.

První podkapitola se zabývá základním zpracováním signálu. Z EEG ve své podstatě získáváme spojitý signál z každé elektrody. Je jasné, že takové signály nelze zpracovávat, neboť je třeba signál nejprve navzorkovat. Když je signál navzorkovaný, stále máme oddělené kanály a tudíž pro lepší práci s daty je vhodné zvolit nějaký způsob pro zprůměrování signálů, aby vznikl pouze jediný. Například pro zpracování vizuálních evokovaných potenciálů je vhodné průměrovat pouze hodnoty z elektrod okcipitálního laloku a ostatní kanály nezapočítávat vůbec (nebo s malou váhou).

Další části této kapitoly se zabývají samotným zpracováním předpřipraveného signálu do tvaru, který již člověku může napovědět, zda měření bylo úspěšné a zda se v datech objevují hledané frekvence. Jedná se zejména o různé druhy Fourierovy transformace.

Existuje i mnoho dalších metod zpracování EEG záznamu, které ovšem pro dané téma nemají význam, proto zde nebudou uvedeny.

### 4.1 Základní zpracování signálu

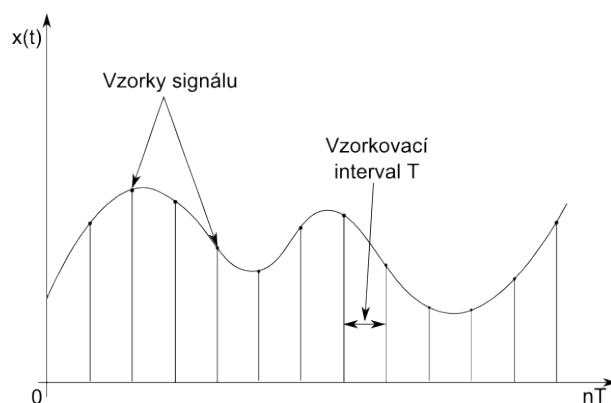
Jak bylo popsáno výše, prvním krokem k získání použitelného signálu je vzorkování a následné průměrování jednotlivých kanálů.

#### 4.1.1 Vzorkování signálu

Při zpracování počítačem se vždy setkáváme s diskrétními daty, neboť paměť počítače není neomezená a ani zpracování neprobíhá nekonečně rychle. Z tohoto důvodu je třeba získávaný signál, který je spojitý, vzorkovat. Vzorkování je tedy proces, ve kterém je spojitý signál transformován na posloup-

nost diskretních vzorků (viz obrázek 4.1). Perioda vzorkování se označuje  $T_s$  a označuje čas mezi jednotlivými vzorky. Převrácenou hodnotou periody je vzorkovací frekvence.

$$f_s = \frac{1}{T_s} \quad (4.1)$$



Obrázek 4.1: Vzorkování analogového signálu

Signál je třeba vzorkovat s dostatečnou vzorkovací frekvencí  $f_s$ , aby data ztratila co nejméně informace. Při vzorkování se ztrátě informace nelze vyhnout. V případě špatně zvolené vzorkovací frekvence může docházet k aliasingu.

### Volba vzorkovací frekvence

Při volbě vzorkovací frekvence je třeba se řídit vzorkovací (Shanonovou) větou, která říká, že pokud je signál  $x(t)$  spojitý v čase a obsahuje složky s frekvencemi menšími než  $f_{max}$ , pak může být rekonstruován z posloupnosti diskretních vzorků  $x(nT)$ , pokud pro vzorkovací frekvence  $f_s = \frac{1}{T_s}$  platí:

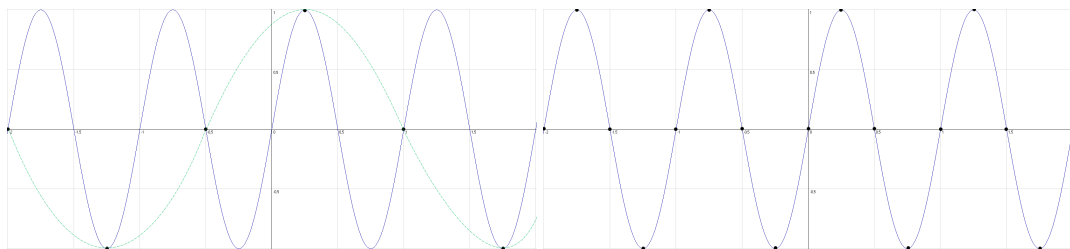
$$f_s \geq 2 \cdot f_{max} \quad (4.2)$$

Frekvence  $\frac{f_{max}}{2}$  se nazývá Nyquistova frekvence (kritická vzorkovací frekvence).



## Aliasing

Pokud vzorkovací frekvence je menší než Nyquistova frekvence dochází k aliasingu. Znamená to, že v rekonstruovaném signálu se místo původní frekvence objeví frekvence jiná, která poruší výsledný signál (viz 4.2 vlevo, kde modrá křivka znázorňuje původní signál, černé tečky jsou jednotlivé vzorky a zelená čárkovaná čára ukazuje signál po rekonstrukci). Pokud se dodrží Shannonova věta, k aliasingu nedochází, jak je vidět na obrázku 4.2 vpravo.



Obrázek 4.2: Vlevo příklad špatného vzorkování. Dochází ke zkreslení původního signálu. Vpravo vhodné vzorkování

Používaný přístroj umožňuje záznam vzorkovací frekvencí  $f_s = 1000\text{Hz}$ . Jelikož budeme zkoumat mozkové vlny maximálně o  $f_{max} = 50\text{Hz}$ , je tato vzorkovací frekvence více než dostatečná a k aliasingu by nemělo docházet.

### 4.1.2 Zpracování kanálů

Jelikož zdrojová data jsou měřena nezávisle na několika elektrodách, je třeba z dat vytvořit jeden signál. Pro účely SSVEP se využívají zejména elektrody O1 a O2. Dále lze použít ještě elektrody Pz, P3 a P4, u kterých už nejsou hledané odezvy tolik patrné.

Nejprve se každému kanálu přiřadí váha, s kterou má být do výsledného signálu započítán. Výsledný signál vznikne vypočítáním váženého průměru z těchto dat. Označíme-li tedy signál z každé elektrody  $el_k$  a jednotlivé váhy  $a_k$  vznikne jednoduchý vztah:

$$\overline{el} = \frac{\sum_{i=1}^n a_i \cdot el_i}{\sum_{i=1}^n a_i} \quad (4.3)$$

S takto upraveným signálem lze pracovat daleko pohodlněji. Samozřejmě jde opět o další ztrátu informace, ale se správně zvolenými váhami to v dané oblasti nehraje výraznou roli.

## 4.2 Fourierova transformace

Fourierova transformace slouží k převodu signálu do frekvenční oblasti. Vychází z Fourierových řad, které umožňují zapsat libovolnou periodickou funkci pomocí součtu goniometrických funkcí. Dle vstupních dat lze rozdělit Fourierovu transformaci na spojitou a diskrétní. V praxi se při zpracování setkáváme výhradně s diskrétním signálem, jak bylo popsáno výše.

### 4.2.1 Diskrétní Fourierova transformace

Diskrétní Fourierova transformace (DFT) je diskrétní transformace využívající Fourierovu analýzu. Jak již název napovídá, očekává na vstupu diskrétní signál, který je často tvořen navzorkovaným spojitým signálem. Vstupní signál musí mít konečný počet vzorků. Pokud tato podmínka není splněna, lze pro výpočet použít pouze část signálu, který už samozřejmě obsahuje  $N$  vzorků. V praxi se zpravidla zpracovává signál v okně o určité délce, které se postupně posouvá.

Pro výpočet  $N$ -bodové DFT signálu s  $N$  vzorky lze použít následující vzorce.

$$X_{DFT}[k] = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}; k = 0, 1, \dots, N-1 \quad (4.4)$$

$$X_{DFT}[k] = \sum_{n=0}^{N-1} x(n)[\cos(2\pi nk/N) - j\sin(2\pi nk/N)] \quad (4.5)$$

Zpětná transformace je dána vztahem:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_{DFT}[k]e^{j2\pi nk/N} \quad (4.6)$$

Jelikož  $e^{j2\pi nk/N}$  je periodická funkce, je periodická i DFT (případně IDFT). Z tohoto důvodu stačí počítat vzorky pouze přes jednu periodu.

## Symetrie DFT

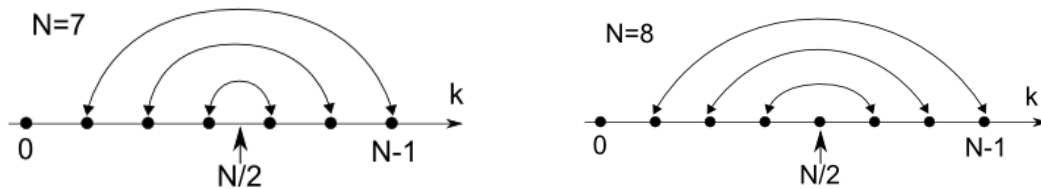
DFT reálného signálu vykazuje komplexně sdruženou symetrii okolo počátku (viz obrázek 4.3) tj.:

$$X_{DFT}[-k] = X_{DFT}^*[k] \quad (4.7)$$

Jelikož je DFT periodické, lze napsat:

$$X_{DFT}[-k] = X_{DFT}[N - k] \quad (4.8)$$

Pokud tedy spočítáme spektrum ze signálu o N vzorcích, dostaneme spek-



Obrázek 4.3: Symetrie DFT pro lichý a sudý počet vzorků

trum souměrné podle středu ( $N/2$ ). Není tedy potřeba počítat DFT pro všech N vzorků, ale stačí pouze pro  $N/2$  vzorků.

## Frekvence v DFT

Po výpočtu DFT je získána posloupnost N prvků, které jsou souměrné podle středu. Každé z těchto čísel udává četnost frekvence v signálu. Je však třeba zjistit, jaká frekvence odpovídá jakému prvku posloupnosti.

Frekvenci konkrétního členu posloupnosti lze jednoduše zjistit, pokud je známá vzorkovací frekvence  $f_s$  signálu. Máme-li N vzorků signálu a vzorkovací frekvenci  $f_s$  bude každý prvek posloupnosti odpovídat frekvenci:

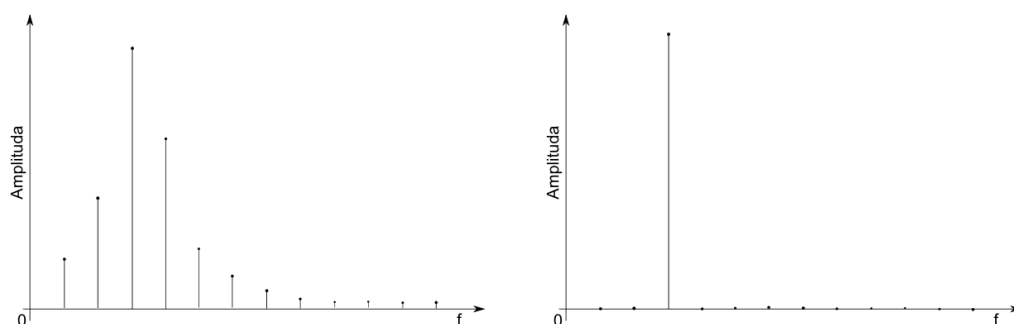
$$f[k] = k \cdot \frac{f_s}{N} \quad (4.9)$$

Maximální frekvence, kterou posloupnost DFT může obsahovat, je polovina vzorkovací frekvence a nachází se na pozici  $X_{DFT}[N/2]$  pro sudý počet N. Pro lichý je to o prvek blíže (viz obrázek 4.3).

Jelikož frekvence jsou diskrétní, může se stát, že signál obsahuje frekvence, které spektrum neobsahuje (například frekvence ve spektru jdou po 1 Hz a signál obsahuje frekvenci, která není celočíselná). V takovém případě dochází k tzv. prosakování ve spektru.

### Prosakování ve spektru

Vyskytuje se tehdy, pokud u vzorkovaného analogového signálu počítáme DFT z  $N$  vzorků a v těchto vzorcích není obsažen celočíselný počet period sinusoid obsažených ve výstupním signálu. Znamená to, že se v DFT vyskytnou spektrální čáry i jinde, než ve frekvenci vstupního signálu (obrázek 4.4).



Obrázek 4.4: Na levém obrázku dochází k prosakování ve spektru. V pravém nikoliv

### 4.2.2 Rychlá Fourierova transformace

Hlavním problémem DFT byla složitost výpočtu, který je  $O(N^2)$  operací. Rychlá Fourierova transformace (FFT) optimalizuje výpočet na složitost  $O(N \log N)$ . S prvním algoritmem FFT přišel v roce 1965 J. W. Cooley a John Tukey a jedná se o nejpoužívanější algoritmus FFT.

Základní princip, který je využíván pro výpočet FFT je rozklad signálu na liché a sudé vzorky, z kterých je dělána transformace zvlášť.  $N$ -bodová DFT může být totiž zapsána jako součet dvou  $N/2$  bodových transformací vytvořených ze sudých a lichých vzorků.

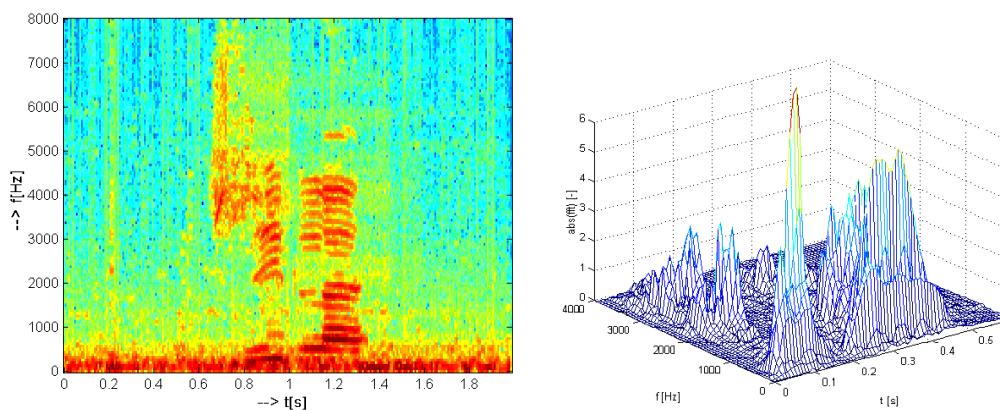
Malou cenou za efektivitu algoritmu je nutnost počítat FFT vždy z počtu vzorků, který je roven mocnině dvou, na rozdíl od DFT, kterou lze vypočítat z libovolného počtu vzorků. V případě, že nemáme správný počet vzorků, lze doplnit signál nulami na nejbližší mocninu dvou.

### 4.2.3 Krátkodobá Fourierova transformace

Krátkodobá Fourierova transformace je zachycení frekvenčního spektra v čase. Signál je rozdělen na bloky. Pro každý blok je vypočteno spektrum a vykresleno do tzv. spektrogramu.

Výsledný spektrogram nejvíce ovlivňuje délka jednotlivých bloků. Pokud je blok větší (více vzorků pro DFT), je přesnější určení jednotlivých frekvencí, ale naopak se vytrácí informace o časovém umístění daných frekvencí. Pokud jsou naopak bloky malé, lze přesně identifikovat čas, ale frekvenční spektrum bude mít menší rozlišení.

Spektrogram může být vykreslován buď v 2D zobrazení, kde intenzitu určuje barva (obrázek 4.5 vlevo), nebo ve 3D zobrazení, kde je intenzita vynesena na třetí osu (obrázek 4.5 vpravo).



Obrázek 4.5: Vlevo ukázka 2D spektrogramu, vpravo 3D spektrogramu

## 5 Klasifikace dat

Klasifikace dat a rozpoznávání je široce používaná oblast v mnoha oborech. Jedná se o zařazování objektů reálného světa do tříd. Tyto třídy jsou buď předem definovány (klasifikace) nebo se třídy vytvářejí za běhu (rozpoznávání). Aby bylo možné jednotlivé objekty od sebe odlišit, je třeba zvolit vhodné veličiny, které je charakterizují. Výběr vhodných veličin použitých pro klasifikaci nebo rozpoznávání je u každého problému jiný, vždy záleží na povaze úlohy.

Základní dělení metod pro klasifikaci a rozpoznávání je na příznakové a strukturální. Příznakové využívají tzv. příznakový vektor, který je složený z veličin charakterizujících objekt. Příznakové metody tento vektor porovnávají a na základě toho se rozhodují o zařazení do třídy. Strukturální metody nejprve nadefinují množinu základních popisných elementů - primitiv a objekty pomocí nich popisují. Jelikož pro realizaci této práce nelze využít strukturální metody, budou popsány jen příznakové. Více informací o klasifikaci v [5].

### 5.1 Příznakové rozpoznávání

Prvním krokem pro realizaci příznakového rozpoznávání je vybrat příznaky, charakterizující klasifikované objekty. Tyto příznaky se uspořádají do příznakového vektoru. Na základě tohoto příznakového vektoru rozhoduje klasifikátor o příslušnosti objektu k třídě. Klasifikátor lze nastavit dvěma způsoby:

- Analýzou problému a definováním rozhodovacího pravidla před klasifikací
- Sestavením rozhodovacího pravidla s použitím objektů, jejichž správná klasifikace je předem známa. Tomuto způsobu se říká učení (nebo trénování) klasifikátoru.

### 5.1.1 Diskriminační funkce

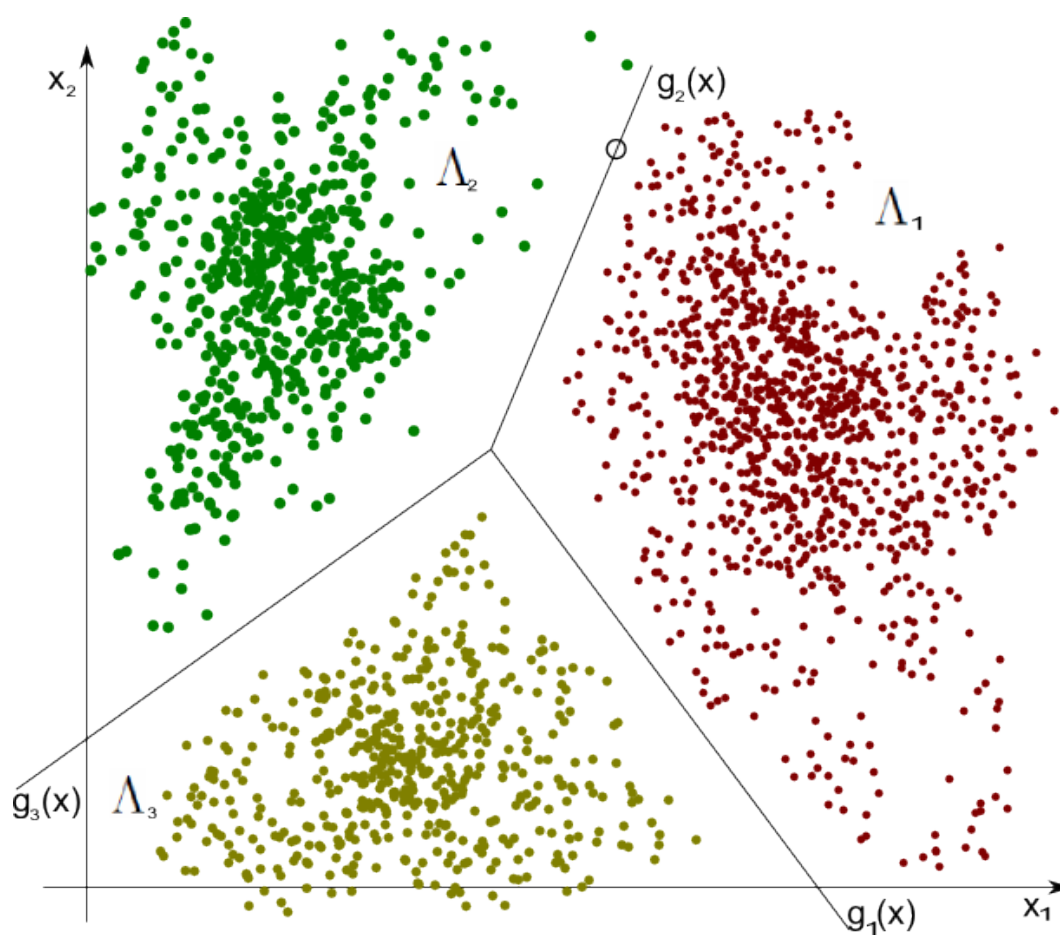
Máme příznakový vektor  $x^T = [x_1, x_2, \dots, x_n]$  o  $n$  prvcích. Příznakový prostor je  $n$ -rozměrný a označíme jej  $\Lambda$ . Klasifikace probíhá do  $R$  tříd a každá třída je označena identifikátorem  $w_1, w_2, \dots, w_R$ . Funkce  $w = d(x)$  je rozhodovacím pravidlem, které popisuje toto přiřazení. Rozhodovací pravidlo vymezuje  $R$  disjunktních podprostorů  $\Lambda$  označených  $\Lambda_1, \Lambda_2, \dots, \Lambda_R$ . Nadplochy, které jsou společné dvěma množinám  $\Lambda_i, \Lambda_j$  nazýváme rozdělovací nadplochy. Tyto nadplochy nazýváme diskriminační funkce a označujeme je  $g_i(x), i = 1, \dots, R$ . Každá diskriminační funkce je přiřazena jedné z tříd. Vektor  $x$  zařadíme do třídy  $\Lambda_i$  tehdy, pokud má diskriminační funkce přiřazená této třídě pro tento vektor  $x$  vyšší hodnotu než diskriminační funkce všech ostatních tříd. Pokud by vektor ležel na rozdělovací nadploše, nelze o příslušnosti k třídě rozhodnout. Pro  $n = 2$  a  $R = 3$  je vidět příklad na obrázku 5.1.

### 5.1.2 Příznakový vektor

Příznakový vektor je vektor, který je složený z jednotlivých příznaků popisujících objekt. Tyto příznaky jsou získány různým způsobem. Nejjednodušší možností je změřit fyzikální vlastnosti objektů (např. velikost, hmotnost, tvar...). Tomuto způsobu se říká selekce příznaků. Druhou možností je tzv. extrakce příznaků, která získává příznaky z původních příznaků nějakou transformací. V mnoha oblastech se používají například Fourierovy koeficienty, které vypovídají o určitých vlastnostech objektu.

Některé úvahy mohou jít směrem, že pokud příznakový vektor obsahuje velké množství příznaků, je objekt popsán přesněji, tudíž klasifikace bude probíhat s vyšší přesností. Tyto úvahy naráží na dva hlavní problémy. První z problémů je ten, že vyšší počet příznaků v žádném případě nezaručuje vyšší spolehlivost klasifikace, dokonce v mnohých případech spolehlivost snižuje. Druhý problém nastává v rychlosti, kdy větší příznakový vektor znamená delší čas potřebný pro zpracování a navíc samotná extrakce příznaků trvá nepochybně déle.

U klasifikace je vždy největším problémem vybrat správné příznaky, neboť pro každou úlohu musí příznaky vybírat člověk na základě zkušeností, důkladné analýzy problému a mnohdy také intuice.



Obrázek 5.1: Dvouzměrný příznakový prostor se třemi třídami



### 5.1.3 Klasifikátory

Klasifikátor je stroj, který zobrazuje příznakový vektor na identifikátor třídy. Existuje mnoho typů klasifikátorů založených na různých principech. Pro naši potřebu budou uvedeny jen nejpoužívanější.

#### Minimální vzdálenost od těžiště třídy

Tato metoda klasifikace je jednou z nejjednodušších. Zpravidla je předložena množina trénovacích dat, které mají určenou příslušnost ke své třídě. Z těchto dat je vypočteno těžiště třídy  $t^T = [t_1, t_2, \dots, t_n]$ , které je reprezentováno bodem v klasifikačním prostoru. Druhou možností je zadat těžiště přímo, pokud známe přesné hodnoty ideálního příznakového vektoru pro konkrétní třídu.

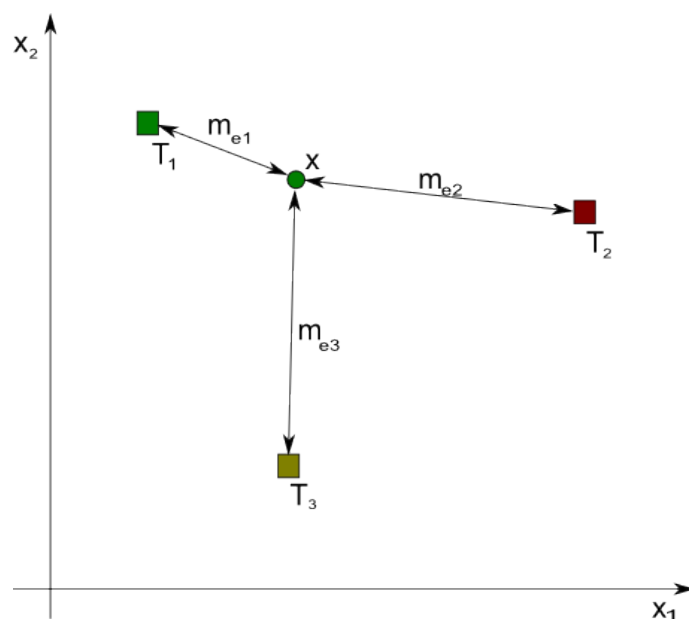
Při klasifikaci je získán příznakový vektor objektu a počítají se vzdálenosti k těžištím všech tříd. Zpravidla je používána Euklidovská vzdálenost, ale lze použít i jiné metriky.

$$m_e = \sqrt{\sum_{i=1}^n (t_i - x_i)^2} \quad (5.1)$$

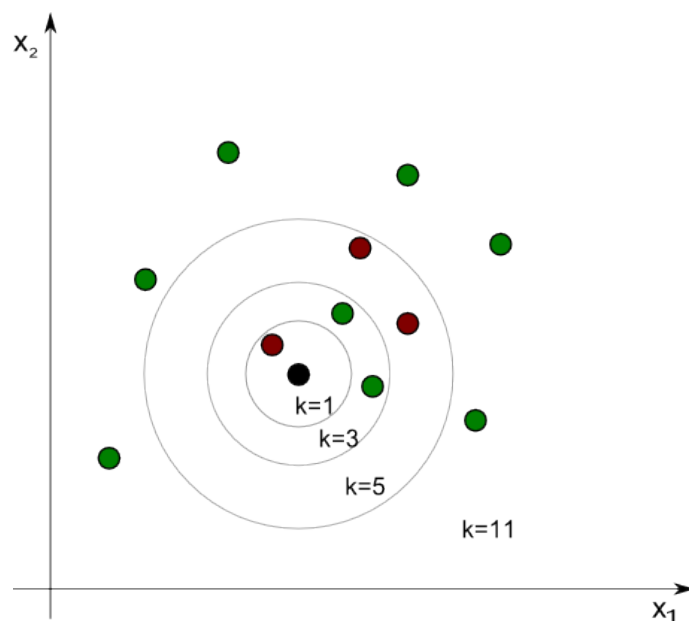
Objekt bude zařazen do třídy, jejíž těžiště je nejbližší příznakovému vektoru klasifikovaného objektu (obrázek 5.2).

#### K-nejbližších sousedů

Metoda k-nejbližších sousedů je zobecnění výše uvedeného způsobu klasifikace. Spočívá v zařazení objektu do třídy dle  $k$  nejbližších objektů z trénovací množiny. Často používaná hodnota je  $k = 1$ , kde je objekt zařazen do stejné třídy jako nejbližší výskyt prvku z trénovací množiny. V praxi se používají další prvočísla, ale více sousedů než 13 se běžně nepoužívá. Pro  $k > 1$  se uvažuje  $k$  nejbližších sousedů a objekt je zařazen do třídy, která má v daném okolí nejvyšší zastoupení. Jak je vidět na obrázku 5.3 na volbě  $k$  závisí výsledek klasifikace. V tomto případě pro  $k = 1$  a  $k = 5$  bude objekt zařazen do třídy červených, pro  $k = 3$ ,  $k = 7$  a  $k = 11$  se zařadí k třídě zelených.



Obrázek 5.2: Dvourozměrný příznakový prostor se třemi třídami. Čtverce znázorňují těžiště tříd. Kolečko znázorňující příznakový vektor objektu je klasifikováno do zelené třídy dle kritéria nejmenší vzdálenosti k těžišti



Obrázek 5.3: Klasifikace pomocí algoritmu k-nejbližších sousedů. Výsledky klasifikace jsou rozdílné pro různé hodnoty  $k$

**Shluková analýza**

Shluková analýza je metoda, která slouží k nalezení shluků v množině dat obsahující neklasifikované příznakové vektory. Cílem je nalézt shluky těchto vektorů, tj. skupiny, jejichž prvky jsou si vzájemně blízké.

## 6 Návrh a implementace BCI

BCI (brain computer interface) je jeden ze způsobů, jak komunikovat s počítačem. Základní způsob komunikace s počítačem je pomocí klávesnice, později se začalo pracovat na komunikaci v přirozeném jazyce a nejnovější způsob je komunikace pomocí mozku, která zatím nemá velké uplatnění, ale lze předpokládat velký rozvoj do budoucnosti.

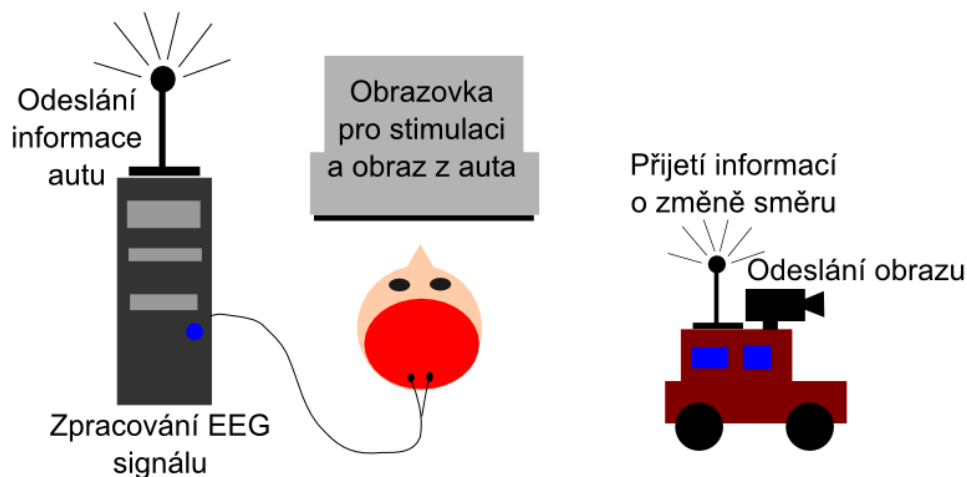
Způsobů, jakými lze ovládat zařízení pomocí mozku, je mnoho a záleží jen na autorovi BCI, jaký zvolí. Samozřejmě k volbě způsobu ovládání je třeba znát základní fyziologii mozku a diagnostické metody - tedy v našem případě EEG. Tato práce se zabývá využitím ustálených vizuálních evokovaných potenciálů v BCI.

Pro využití SSVEP v BCI se nabízí hned několik možných využití. Jelikož se bude jednat o rozpoznávání několika frekvencí, které budou prezentovány, první věc, která člověka napadne, je řízení směru. Může se jednat o řízení kurzoru myši na obrazovce, ovládání jednoduché hry a nebo řízení modelu auta. Pro práci jsem vybral možnost řízení modelu auta pomocí SSVEP.

Jednotlivé moduly BCI byly většinou naimplementovány v javě. Jedinou výjimkou je stimulační scénář, ke kterému byl použit software Presentation. Tyto moduly jsou používány v rámci dvou programů, na kterých bylo prováděno testování. Tyto programy obsahují další užitečné funkce pomáhající určit správnou funkcionalitu modulů. Moduly lze jednoduše použít v jiném programu, neboť jsou to samostatné třídy, popřípadě celé balíky.

### 6.1 Součásti BCI

K úspěšnému sestavení BCI je třeba několik součástí. V první řadě se jedná o zařízení pro měření EEG, které umožňuje real-time zpracování ve vytvořeném programu. Další součástí je počítač, na kterém budou EEG data zpracovávána, a z kterého bude prezentován stimulační scénář na monitor. Dále je třeba mít bezdrátovou webkameru, která bude umístěna na autě a bude posílat obraz cesty před autem. Směr auta bude řízen pomocí bluetooth zařízení na autě a to bude ovládat program v počítači pro zpracování EEG dat. Pro lepší představu je zde obrázek 6.1.



Obrázek 6.1: Navržené BCI

V tomto návrhu se počítá s jedním počítačem a jedním programem, který bude obstarávat stimulaci i zpracování dat. Druhou možností je udělat dva nezávislé programy, z nichž jeden bude zajišťovat stimulaci a zobrazovat obraz z webkamery a druhý zpracovávat data a odesílat změny směru jízdy.

## 6.2 Modul pro stimulaci

Modul pro stimulaci by měl obsahovat dvě základní věci. Jedna z nich je samotná stimulace, druhá zpětná vazba, která uživateli zobrazuje aktuální akci. Modul pro stimulaci nebyl plně implementován. Byla implementována pouze část, potřebná pro testování ustálených vizuálních evokovaných potenciálů. Pro naše potřeby stačilo naprogramovat scénář v softwaru Presentation.

**Stimulace** je základní předpoklad pro správnou funkci BCI. Pro stimulaci mohou být využity různé obrazce, které budou dostatečně stimulovat zrakovou dráhu uživatele. Umístěny by měly být na kraji monitoru, aby se vzájemně co nejméně ovlivňovaly. Toto umístění je vhodné ještě z důvodu, že uživateli snadno napoví směr. Pokud bude stimul na levé části obrazovky,



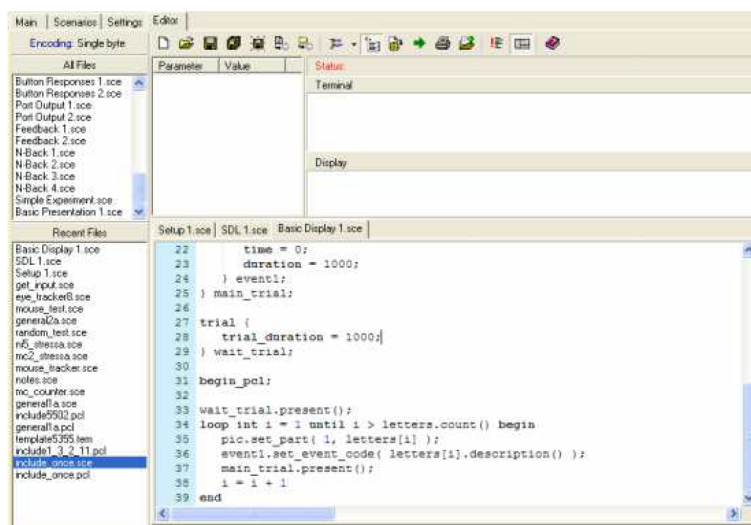
Obrázek 6.2: Návrh obrazovky pro stimulaci

bude zajišťovat pohyb doleva.

Na monitor lze umístit větší množství stimulů, ale je třeba si uvědomit, že čím více stimulů bude, tím více se mohou ovlivňovat a výsledná klasifikace nemusí být tolik spolehlivá. Pro začátek je dobrou volbou umístit na každou stranu monitoru jeden stimul a ovládání auta nadefinovat tak, že pojede konstantní rychlostí a uživatel bude moci pouze zatáčet doleva nebo doprava.

Pro samotnou stimulaci lze využít software Presentation. Tento software umožňuje vytvořit celkem široké spektrum scénářů pro stimulaci, ale jeho použití je limitováno naším požadavkem na umístění obrazu z webkamery na obrazovku. Z tohoto důvodu může být používán úspěšně pro testovací účely, ale pro konečnou verzi BCI není úplně vhodný.

**Zpětná vazba** je důležitá pro uživatele, aby věděl, zda se vykonává akce, jakou požaduje. Stejně jako u většiny programů v počítači se ikona po klepnutí označí, i zde je třeba dát uživateli najevo, že danou akci provedl nebo provádí. V našem návrhu je z tohoto důvodu umístěna na auto webkamera a jako zpětná vazba slouží obraz z kamery. Uživatel tudíž vidí, zda zatáčí nebo ne. Je však ještě navíc vhodné využít nějakou další signalizaci o zatáčení, například umístění šipek na obrazovku a zvýraznění jedné z nich v případě zatáčení. Stimulační obrazovka by měla vypadat jako na obrázku 6.2.



Obrázek 6.3: Editor softwaru Presentation

## 6.2.1 Popis programu Presentation

Presentation je komerční software firmy Neurobehavioral Systems. Tento program umožňuje pomocí dvou vestavěných jazyků PCL a SDL programovat různé scénáře pro stimulaci, jak vizuální, tak i sluchovou. Podporovány jsou i nejrůznější rozhraní pro vstup a výstup. Pro naše účely byl používán paralelní port pro posílání značek (tzv. markerů) do EEG záznamu. Více informací na [7].

Software presentation obsahuje přímo textový editor pro psaní kódu jazyků PCL a SDL (obrázek 6.3). Pokud jsou používány nějaké vstupní a výstupní porty, je třeba provést nastavení v záložce Settings.

### Součásti scénáře

Každý soubor by se měl skládat ze tří částí. První z nich je hlavička, druhá část je kód jazyka SDL a třetí kód jazyka PCL.

**Hlavička** je umístěna na začátku souboru a definuje základní vlastnosti jako je velikost písma, fontu nebo i nastavení portu. Pokud se nenastaví žádné hodnoty, jsou použity defaultní.

**Kód jazyka SDL** je další část programu. Nachází se za příkazem *begin*; a definují se v ní jednotlivé prvky stimulace, což mohou být například obrázky nebo zvuky. Zvláštním objektem je tzv. *trial*, který v sobě sdružuje několik jednoduchých objektů. SDL není klasický programovací jazyk, slouží pouze k definici objektů.

Následující příklad definuje dva objekty. První objekt je typu *picture* a je v něm nastavena pouze barva pozadí na modrou. Druhý objekt je typu *bitmap*, ve kterém je nastavena cesta k obrázku a barva, která má být průhledná.

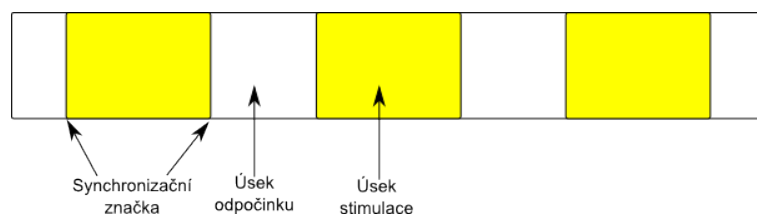
```
picture {
    background_color = 0,0,255;
} default;
bitmap {
    filename = "pic1.jpg";
    transparent_color = 0,0,0;
} pic1;
```

**Kód jazyka PCL** je uvozen příkazem *begin\_pcl*; Nachází se za úsekem definice objektů jazyka SDL. PCL lze oproti SDL považovat za jednoduchý programovací jazyk. PCL obsahuje základní jazykové konstrukce jako jsou smyčky nebo podmínky, má několik datových typů a dokonce lze používat i pole. Jazyk SDL nadefinuje objekty, které pak jsou v PCL dostupné a lze je využívat a měnit jim některé vlastnosti.

Následující příklad ukazuje definici v SDL a následné použití v PCL. V SDL je nadefinován obrázek s názvem *pic1* a *trial* s názvem *trial1*. Presentation automaticky vytvoří v PCL proměnné *pic1* a *trial1*, pomocí nichž PCL manipuluje s objekty. V tomto případě je nastavena x souřadnice části obrázku a následně je *trial* prezentován.[1]

```
begin;
trial {
    picture {
        bitmap map1;
        x = 0; y = 0;
    } pic1;
} trial1;
```





Obrázek 6.4: Rovržení úseků blikání ohraničených markery

```
begin_pcl;
pic1.set_part_x( 1, 100 );
trial1.present();
```

## 6.2.2 Požadavky na scénář

Jelikož má být scénář využíván zejména pro testování, je třeba, aby bylo možné snadno měnit určité parametry.

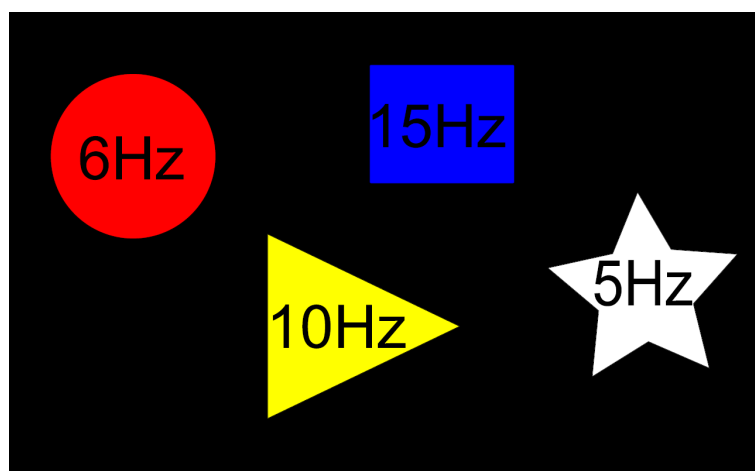
První z nich je frekvence stimulů. Jelikož obrazovka je schopna se překreslovat 60x za sekundu, nejvyšší možná frekvence, kterou lze zobrazit, je 30Hz (1 překreslení obrázku a 1 překreslení zhasnuto). Pokud chceme zachovat stejnou délku obrázku a pauzy, budou použitelné frekvence velice omezené. I při těchto možnostech lze pohodlně vybrat vhodné frekvence.

Tvar a barvy stimulů jsou důležité pro různé testy. Nejjednodušší způsob, jak je ve scénáři měnit, je předpřipravit bitmapové obrázky, které jsou následně do scénáře načítány.

Jeden ze základních požadavků je mít více stimulů v jednom scénáři. Stimuly musí mít různé frekvence a různé pozice na obrazovce.

Scénář by měl umožnit definovat úseky blikání a pauzy pro odpočinek. Každý úsek by měl být označen v záznamu markery, které lze posílat přes paralelní port. Markery v záznamu mohou vypadat jako na obrázku 6.4.

Možný scénář, který lze připravit, může vypadat jako na obrázku 6.5.



Obrázek 6.5: Možný scénář splňující požadavky

### 6.2.3 Implementace scénáře

Stimulační scénář je převzatý z [1] a byl jen modifikován vždy pro konkrétní experiment. Inspirací pro scénář byl kód z webových stránek výrobce.

#### Jazyk SDL

Jak již bylo zmíněno, je výhodnější vytvořit soubor bitmapových obrázků než definovat různé tvary v kódu SDL. Každý obrázek použitý pro stimulaci je třeba na počátku načíst. V příkazu se definuje cesta k souboru s obrázkem. Načtení se provede následovně:

```
bitmap { filename = "obr1\\blue_box.png"; } blueBox;
```

Tímto způsobem jsou načteny všechny požadované obrázky.

V části SDL jsou ještě nedefinovány tři objekty typu `trial`. První z nich obsahuje obrázek a je aktualizován při probíhající stimulaci z jazyka PCL.

Další z `trialů` zajišťuje pauzu mezi jednotlivými bloky stimulace. Poslední je v kódu umístěn pouze z důvodu logování. Tyto `triale` obsahují objekt `stimulus_event`, který zajišťuje zápis přesných časů spuštění do logovacího souboru.

V následujícím kódu se nacházejí dva trialy. První je spuštěn bezprostředně po skončení blikání. Do logovacího souboru zapíše čas spuštění. Od tohoto okamžiku trvá 4999 milisekund. Následně je spuštěn druhý z trialů, který zapíše do logovacího souboru svůj čas spuštění, což je milisekunda před začátkem blikání (za tímto trialem následuje sekvence blikání).

```
trial {
  stimulus_event {
    picture {
      background_color = 0, 0, 0;
    } pic2;
    code = "stim2";
    duration = 4999;
  } stim2;
} trial2;
trial {
  stimulus_event {
    picture {
      background_color = 0, 0, 0;
    } pic3;
    code = "stim3";
    duration = 1;
  } stim3;
} trial3;
```

## Jazyk PCL

Kód jazyka PCL se skládá ze dvou funkcí a hlavního programu. Na počátku je definován paralelní port, za kterým následují definice používaných polí.

**Funkce `add_stim`** slouží k přidání nového stimulu na obrazovku. Parametry funkce jsou dva obrázky, pozice  $x$  a  $y$ , délka trvání jednoho z obrázků a poslední je délka trvání obou obrázků dohromady. Funguje to tak, že se střídají definované obrázky (jeden z obrázků má zpravidla barvu pozadí). Jeden se zobrazuje vždy několik cyklů po sobě dle zadané hodnoty parametru. Parametr délky trvání neoznačuje čas, ale počet, kolikrát za sebou má být jeden obrázek prezentován.

**Funkce `iterate`** je volána v každém průchodu smyčkou. Projde všechny stimuly přidané pomocí funkce `add_stim` a nastaví jim obrázek, který má být prezentován.

**Hlavní program** nejprve přidá do scénáře stimuly voláním funkce `add_stim` a dále se stará o spouštění připravených trialů. Volání funkce `iterate` ve smyčce zajišťuje blikání stimulů dle předchozí definice.

```
add_stim( blueBox, background, -500,0, 10, 5 ); # 6Hz
add_stim( blueBox, background, 500,0, 4, 2 ); #15Hz
loop int j = 1 until j > 10 begin #hlavní smyčka, 10 opakování
    stim2.set_event_code("konec blikani");
    trial2.present();
    stim3.set_event_code("zacatek blikani");
    trial3.present();
    oport.send_code( 1 ); #synchronizační značka
    loop int k~= 1 until k~> 420 begin
        iterate();
        k~ = k~ + 1;
    end;
    oport.send_code( 2 ); #synchronizační značka
    j = j + 1;
end;
```

V příkladu výše se nejprve přidají dva stimuly na pozice (-500; 0) a (500; 0). Stimuly mají různou frekvenci. U prvního stimulu je předposlední parametr 10 a poslední 5. Tyto čísla udávají počty iterací pro jednotlivé obrázky. Znamená to, že perioda stimulu bude 10 překreslení a z toho 5 překreslení pro jeden z obrázků. Pokud tedy proběhne 60 překreslení za sekundu, blikání bude probíhat s frekvencí 6 Hz.

Po přidání stimulů se spustí smyčka, která proběhne 10x. Uvnitř smyčky se spustí dva trial, ve kterých zůstane černá obrazovka, jen se zalogují časy. Poté se vyšle synchronizační značka a spustí se smyčka s blikáním. Smyčka probíhá 420x, z čehož při 60 překreslení za sekundu snadno dopočítáme čas blikání, což dělá 7 sekund. Po skončení této smyčky se vyšle na paralelní port opět synchronizační značka.

## 6.3 Modul pro načítání dat EEG

Pro samotné řízení modelu auta je třeba zajistit načítání dat přímo z přístroje a to v reálném čase. Používaný software od výrobce EEG zařízení umožňuje posílat v reálném čase po síti data, která mohou být zpracována dalším programem.

Standardní výstup softwaru BrainVision je soubor v definovaném formátu, který lze zpracovávat zpětně až po ukončení měření. Tento typ souboru lze využít pro testování stimulátorů a nastavení klasifikátorů. Načítání dat bylo implementováno oběma způsoby.

### 6.3.1 Načítání ze souboru

Jako první bylo vytvořeno načítání dat ze souboru, neboť se jedná o jednodušší způsob. Základ načítání ze souboru vytvořil Martin Prantl, jehož kódy byly lehce modifikovány.

Načítání umožňuje kromě EEG dat načítat i uložené markery, k čemu slouží jedna z tříd. Třídy se nacházejí v balících *load* a *data*. Složení jednotlivých balíků je následující:

- data
  - Constants
  - DataFields
- load
  - FileLoading
  - Load
  - LoadEEG
  - LoadVHDR
  - LoadVMRK
  - MyException

## Balík data

Balík *data* obsahuje třídu pro ukládání dat a třídu s konstantami. Obě třídy mají jen několik statických proměnných. Je to z důvodu viditelnosti těchto dat v celém projektu. Předpokládá se, že tyto data budou používána v převážné většině tříd a předávání by bylo nepohodlné.

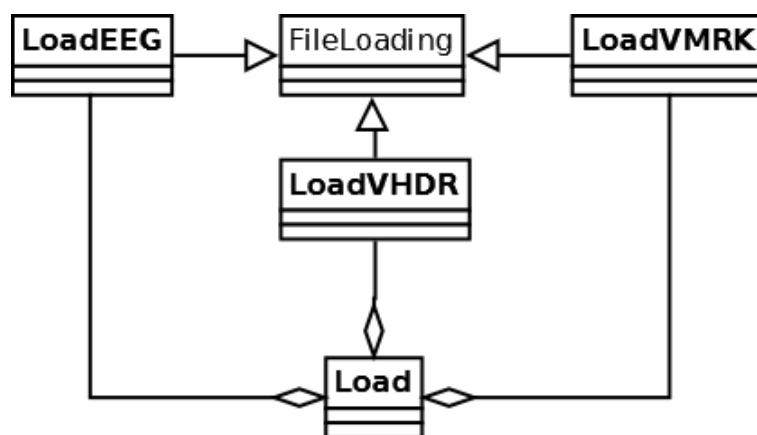
**Třída Constants** obsahuje konstanty dané formátem načítaných souborů. Jedná se o různé konstanty týkající se markerů, jednotlivých kanálů a reprezentaci čísel. Tyto konstanty jsou poté pro větší přehlednost používány při načítání a práci s daty.

**Třída DataFields** definuje několik statických polí. Do těchto polí budou ukládány kromě EEG dat i informace o markerech a další informace o datech.

- *eegData* – dvourozměrné pole typu *short* pro data EEG záznamu. Obsahuje EEG záznam z každé elektrody. Z těchto dat jsou dále extrahovány pouze požadované kanály.
- *channelInfos* – dvourozměrné pole typu *Object* nesoucí informace o jednotlivých kanálech (např. název kanálu)
- *markerInfos* – dvourozměrné pole typu *Object* uchovává informace o markerech a jejich pozicích
- *transformedData* – pole typu *double* obsahuje EEG data extrahovaná z požadovaných kanálů a připravená na další zpracování

## Balík load

Třídy v balíku *load* se starají o načítání dat. Jedná se o načítání různých typů souborů, které dohromady dávají kompletní informace o měření. Třídy *LoadEEG*, *LoadVMRK* a *LoadVHDR* dědí od abstraktní třídy *FileLoading*. Všechny tři jsou obsaženy ve třídě *Load*, která se stará o kompletní načtení dat. Diagram tříd lze vidět na obrázku 6.6.



Obrázek 6.6: UML diagram tříd pro načítání dat ze souboru

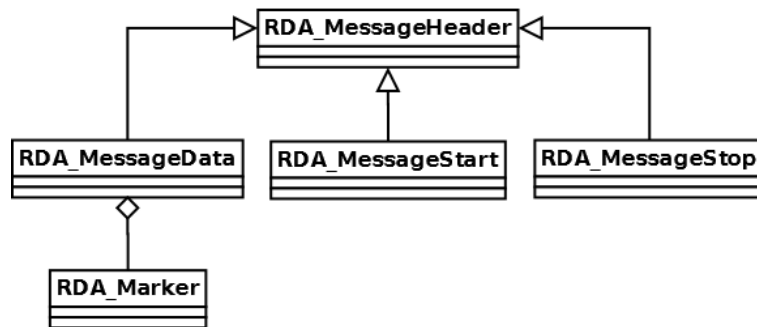
**Třída FileLoading** je abstraktní třída a obsahuje metody pro vytvoření datového proudu ze souboru. O načítání konkrétního typu souboru se starají oddělené třídy, které dodefinovávají abstraktní metody *load* a *allocateMemForData*.

**Třídy LoadEEG, LoadVHDR a LoadVMRK** zajišťují načítání jednotlivých typů souborů. Výstupem EEG zařízení je totiž trojice souborů. Soubory typu *.vhdr* a *.vmrk* jsou textové a *.eeg* je soubor binární. Tyto soubory jsou načítány a ukládány do připravených polí třídy *DataFields*.

**Třída Load** zastřešuje předchozí třídy. Umožňuje načítat jen konkrétní soubor, ale v drtivé většině případů je třeba načíst veškerá data. K tomuto účelu obsahuje metodu *loadAll*.

### 6.3.2 Načítání v reálném čase

Načítání dat v reálném čase umožňuje BrainVision Recorder pomocí síťové komunikace. Tento způsob je již náročnější než načítání dat ze souboru. Jelikož je třeba načítat data a zároveň je nějakým způsobem zpracovávat, nevyhneme se práci s vlákny. Základní síťovou komunikaci se softwarem BrainVision Recorder vytvořil Michal Patočka. Kód na nižší úrovni zůstal zachován,



Obrázek 6.7: UML diagram tříd datových objektů

přepracovány byly pouze části pro ukládání dat. Tyto změny byly nevyhnutelné kvůli rozdílnému zaměření cílového programu.

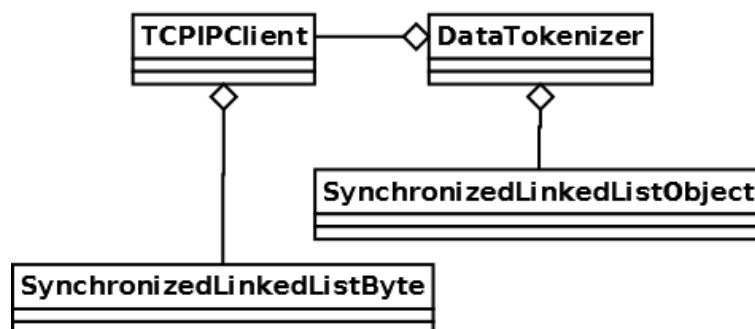
Sít'ová komunikace probíhá způsobem klient-server. Je nadefinováno několik typů zpráv. Všechny tyto zprávy obalují třídy z balíku *data.objects*. K naplnění těchto datových objektů slouží třída *DataTokenizer* z balíku *data*. Pro komunikaci na nižší úrovni využívá *DataTokenizer* třídu *TCPClient*, která poskytuje přijatá data od serveru.

### Balík *data.objects*

Balík obsahuje několik tříd, reprezentující jednotlivé typy zpráv, které server posílá. Diagram tříd na obrázku 6.7.

- *RDA\_Marker* – reprezentuje marker. Obsahuje několik důležitých informací jako je název markeru, k jaké elektrodě patří a jaká je jeho pozice.
- *RDA\_MessageData* – třída, která kromě samotných dat může obsahovat jeden i více markerů.
- *RDA\_MessageHeader* – hlavička každého datového objektu je označena unikátní posloupností bytů. Tuto hlavičku obsahují všechny ostatní datové objekty kromě *RDA\_Marker*. Nese informaci o velikosti následujícího datového bloku.
- *RDA\_MessageStart* – objekt posílaný zpravidla na počátku sít'ové komunikace. Jsou v něm informace o vzorkovací frekvenci a seznam jed-





Obrázek 6.8: UML diagram tříd datových objektů

notlivých kanálů s jejich jmény.

- RDA\_MessageStop – objekt zasílaný serverem na konci komunikace.

Jelikož se počítá s pevně zavedeným nastavením, které se nemění, je bezpochyby nejdůležitější z datových objektů *RDA\_MessageData*. Dále lze situaci zjednodušit tím, že při použití v navrženém BCI nemají markery význam, tudíž zpracovávat má smysl pouze pole s EEG daty.

## Balík data

Balík data obsahuje čtyři třídy. Jejich diagram je na obrázku 6.8.

**Třída TCPIPClient** slouží pro základní komunikaci se serverem. Naváže spojení se serverem a přijímá data po bytech. Tyto data ukládá do třídy *SynchronizedLinkedListByte*. Aby byla zajištěna správná funkcionality, běží klient ve vlákne. Z tohoto důvodu je třeba mít zajištěnou synchronizaci při přístupu k datovému úložišti.

**Třída DataTokenizer** vybírá data uložená třídou *TCPIPClient* a vytváří z nich datové objekty. Tyto objekty jsou následně ukládány do synchronizovaného spojového seznamu typu Object, reprezentovaného třídou *SynchronizedLinkedListObject*. Opět je třeba mít zajištěnou synchronizaci, neboť *DataTokenizer* běží ve vláknu.

## 6.4 Modul pro zpracování dat

Po načtení je třeba data uložit do vhodné struktury, provést transformaci jednotlivých kanálů a následně z nich extrahovat užitečné příznaky. V našem případě se jedná o frekvence mozkových vln v závislosti na frekvenci stimulace. Pro zjištění frekvenčního spektra je třeba využít Fourierovu transformaci a vybrat vhodné Fourierovy koeficienty odpovídající použitým frekvencím (případně jejich nejbližšího okolí).

### 6.4.1 Uložení dat

*DataTokenizer* zajistil uložení všech příchozích bloků do spojového seznamu. Pro použití v BCI je třeba vybrat ze všech datových objektů pouze typy *RDA\_MessageData* a uložit je do sktruktury, která bude pro naši potřebu ideální.

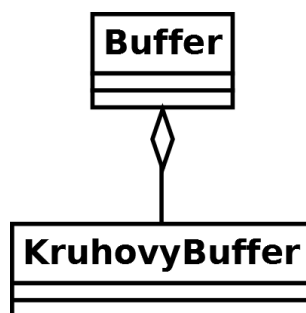
**Kruhový buffer** byl zvolen jako optimální způsob ukládání dat. Kruhový buffer je pole o pevně dané velikosti, které po svém naplnění začne přepisovat od začátku svá data. Tato vlastnost sice znamená ztrátu dat, ale jsou to zpravidla data, která nejsou aktuální. Pokud se tedy počítač při výpočtech zdrží, další průběh se bude počítat z dat, která jsou nejnovější. Kdyby byl použit standardní buffer, zpracovávala by se neaktuální data a zpoždění by dále narůstalo. Kruhový buffer je implementován ve třídě *KruhovyBuffer*.

### 6.4.2 Transformace kanálů

Příchozí datové objekty obsahují hodnoty ze všech elektrod. Jelikož nás zajímají jen určité elektrody, je třeba z dat vybrat pouze data z konkrétních elektrod a smíchat je dle důležitosti do jednoho signálu.

V třídě *Buffer* se nachází metoda *vyberElektrod*, která ze zadaného pole z objektu *RDA\_MessageData* vytvoří nové pole, které bude obsahovat pouze požadovaná data. Tyto data jsou ukládána do kruhového bufferu, který tato třída obsahuje (obrázek 6.9).

```
private double[] vyberElektrod(float[] pole, int o1, int o2,
```



Obrázek 6.9: UML diagram bufferu

```

    int pz, int p3, int p4) {
double[] temp = new double[POCET_HODNOT];
for (int i = 0; i < temp.length; i++) {
    temp[i] = (o1 * pole[01 + 19 * i] +
              o2 * pole[02 + 19 * i] +
              pz * pole[PZ + 19 * i] +
              p3 * pole[P3 + 19 * i] +
              p4 * pole[P4 + 19 * i])
              / (o1 + o2 + p3 + p4 + pz);
}
return temp;
}

```

Metoda požaduje několik argumentů. Prvním argumentem je pole s původními daty. Další argumenty jsou čísla označující váhu, s kterou mají být započítána do výsledného pole. Pomocí těchto vah je počítán vážený průměr. Jsou zde brány v úvahu pouze elektrody, které mají pro SSVEP význam. S ostatními elektrodami se nepočítá. Zpravidla se nastavuje vyšší váha pro elektrody O1 a O2.

### 6.4.3 DFT a FFT

Rychlá a diskretní Fourierova transformace je používána pro získání příznaků pro klasifikaci. Rychlá Fourierova transformace byla převzata od Nizara Kerkeniho z open-source projektu jEDF a nachází se ve třídě *FastFourierTransform*. Jelikož vzorkovací frekvence přístroje je 1000 Hz a pro FFT je třeba

počet vzorků, který je roven mocnině dvou, nikdy nebudou výsledné členy posloupnosti odpovídat celočíselným frekvencím a docházelo by k prosakování ve spektru.

Z tohoto důvodu byla naimplementována i DFT, kterou lze spočítat z libovolného počtu vzorků a tudíž lze ovlivnit, jakým frekvencím budou spektrální čáry odpovídat. Nevýhodou je samozřejmě rychlost DFT, ale v našem případě je dostačující. Diskrétní Fourierova transformace se nachází ve třídě *DFT*.

## 6.5 Modul pro klasifikaci

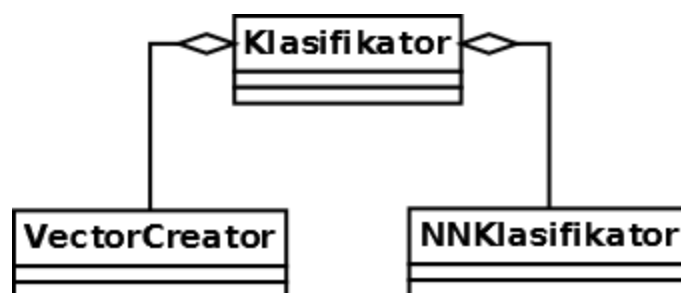
Z charakteru dat je jasně patrné, že klasifikátor bude vytvořen na základě příznakové metody. Příznakový vektor se bude skládat z hodnot určených Fourierových koeficientů, případně jejich druhých mocnin. Pokud tedy budou určeny dvě frekvence, bude probíhat klasifikace do tří tříd a klasifikační prostor bude dvourozměrný.

Možností pro klasifikaci je několik. Nejjednodušší metoda je nastavit ručně prahovou hodnotu, po jejíž překročení bude vektor klasifikován jako jedna či druhá frekvence. Pokud prahovou hodnotu nepřekročí, nebude určena žádná frekvence. Tento způsob má nevýhodu, že u každého člověka se liší amplitudy vln a tudíž nelze určit prahovou hodnotu obecně.

Tento problém může řešit metoda nejkratší vzdálenosti od těžiště. Subjekt bude instruován, aby se zaměřil na jednu frekvenci. Následně se pořídí množina trénovacích dat. Stejně tak se to udělá pro druhou frekvenci a pak i pro třídu reprezentující stav, kdy se subjekt nesoustředí na žádnou z frekvencí. Složitější způsoby klasifikace v tomto případě nemají význam, neboť je jasné, že klasifikační prostor bude rozdělen velice jednoduše, což vychází ze zcela průhledného složení příznaků.

### 6.5.1 Implementace klasifikátoru

Veškeré třídy pro klasifikaci se nachází v balíku *classification*. Jedná se o tři třídy, které předpokládají real-time načítání dat do bufferu popsáno výše. Implementovaná metoda klasifikace se řídí kritériem nejmenší vzdálenosti



Obrázek 6.10: Diagram tříd klasifikace

těžišť tříd. Třída *VectorCreator* se stará o vytvoření těžiště třídy z trénovacích dat. Ve třídě *Klasifikator* jsou uloženy jednotlivé třídy s jejich vektory reprezentujícími těžiště. Tato třída se dále stará o spouštění trénování a spuštění samotné klasifikace. Poslední třída *NNKlasifikator* má za úkol samotnou klasifikaci konkrétního vektoru do nadefinovaných tříd. Diagram tříd je na obrázku 6.10.

### Příznakový vektor

Příznakový vektor se skládá z Fourierových koeficientů. Jelikož klasifikátor je udělán obecně pro libovolný počet tříd, je zpracováván celý výstup Fourierovy transformace a příznakový vektor je vybírán až při samotné klasifikaci ve třídě *NNKlasifikator*. Je možné ponechat jako příznakový vektor celý výstup Fourierovy transformace. Na tento způsob se váže velké množství chyb, které jsou zapříčiněny náhodnou mozkovou aktivitou na frekvencích nesouvisejících se stimulací.

Nejspolehlivější způsob je použít jako příznaky frekvence, kterými se stimuluje. Jelikož může docházet k nepřesnostem, je vhodné ještě započítávat i koeficienty okolo s nižší vahou.

### Třída *VectorCreator*

Tato třída má za úkol z příchozích dat vytvořit těžiště. Zpravidla se jedná o data získávaná za účelem natrénování konkrétní třídy pro klasifikaci. Výpočet těžiště může probíhat libovolnou dobu v závislosti na tom, kolik dat je zapotřebí. Algoritmus stále dokola dělá několik úkonů, které po skončení

zprůměruje.

1. Načtení nejnovějších  $n$  vzorků
2. Výpočet DFT
3. Normalizace jednotlivých koeficientů DFT vzhledem k celkovému součtu koeficientů DFT
4. Součet normalizovaných DFT

Po ukončení je následně součet všech normalizovaných DFT vydělen počtem iterací. V kódu to může vypadat následovně:

```
double[] teziste;  
while (running) {  
    double[] ft = vypocetDFT(nejnovejsiData); //výpočet DFT  
    double suma = 0;  
    for (int i = 0; i < teziste.length; i++) {  
        suma += ft[i]; //celkový součet koef.  
    }  
    for (int i = 0; i < teziste.length; i++) {  
        teziste[i] += ft[i] / suma; //normalizace a součet  
    }  
    counter++;  
}  
for (int i = 0; i < vektor.length; i++) {  
    teziste[i] = teziste[i] / counter; //zprůměrování  
}
```

Z důvodu velkého počtu vektorů se ukládá jen jejich průměr, reprezentující těžiště. Pro klasifikaci dle kritéria nejbližší vzdálenosti k těžišti třídy je to dostačující, avšak znemožňuje to využít jiné metody klasifikace např. klasifikaci dle  $k$  nejbližších sousedů.

### Třída NNKlasifikator

Tato třída zajišťuje klasifikaci aktuálních dat do tříd, které jsou již natrénovány. V konstruktoru je předána HashMapa, obsahující stimulační frekvenci

jako klíč a bod v klasifikačním prostoru reprezentující těžiště dané třídy, který je polem reálných čísel.

Třída je opět navržena pro real-time zpracování, proto je oddělena od třídy *Thread*. V metodě *run* cyklicky probíhá klasifikace, která obsahuje několik kroků:

1. Načtení aktuálních dat z bufferu
2. Fourierova transformace a extrakce příznaků
3. Výpočet euklidovských vzdáleností vektoru ke všem těžištím tříd
4. Výběr nejmenší vzdálenosti a zařazení do příslušné třídy

### **Třída Klasifikátor**

*Klasifikátor* řídí proces trénování klasifikátoru a samotné klasifikace. Obsahuje *HashMap*, která ukládá těžiště tříd a informace o tom, jaké frekvenci náleží. Obsahuje metody pro začátek a konec trénování a také metody pro začátek a konec klasifikace.

## **6.6 Modul pro výběr vhodné stimulační frekvence**

V části zabývající se ustálenými vizuálními evokovanými potenciály bylo řečeno, že amplitudy vln závisí na frekvenci stimulace. Ačkoliv pokusy ukazují konkrétní frekvence, na které člověk reaguje lépe nebo jaké jsou vhodnější pro měření SSVEP, je zcela nepochybné, že na každého člověka působí optimálně jiné frekvence. Z tohoto důvodu může být součástí BCI také modul zajišťující výběr optimálních frekvencí.

Na rozdíl od předchozích částí byl tento modul navržen pro zpracování z naměřeného záznamu, nikoliv real-time. Princip získání nejvhodnějších frekvencí je následující. Uživateli jsou postupně prezentovány podněty s různou frekvencí (např. od 1 Hz do 20 Hz po 1 Hz). Pro každou frekvenci musí být záznam dostatečně dlouhý pro zpracování a každá frekvence musí být

ohraničena markery s identifikátorem frekvence. Ze záznamu se poté pomocí algoritmu získají frekvence, u kterých měly SSVEPy nejvyšší amplity.

### 6.6.1 Algoritmus výběru frekvencí

Předpokládejme, že jsou data uložená ve správném formátu, obsahují markery, které ohraničují vždy dostatečně velký úsek dat.

Nejprve je záznam rozsegmentován podle markerů a je vypočítán práh pro každou frekvenci. Následně se vezme úsek dat, kde osoba nebyla stimulovaná a určí se opět práh pro každou frekvenci. Dále se spočítá pro každou frekvenci podíl prahu se stimulací a prahu bez, což vypovídá o velikosti odezvy na stimulaci vůči klidovému stavu.

Pokud máme pro každou frekvenci určený tento poměr, lze vybrat nejvyšší hodnotu, která odpovídá nejlépe rozpoznatelné frekvenci. Pokud je třeba zvolit více frekvencí, je vhodné, aby nebyly harmonické. Vybere se tedy frekvence s nejvyšší odezvou a vyškrtne se poloviční a dvojnásobná frekvence. Ze zbylých frekvencí se opět vybírá frekvence s nejvyšším poměrem.

Celý algoritmus výběru frekvencí lze vidět na obrázku 6.11.

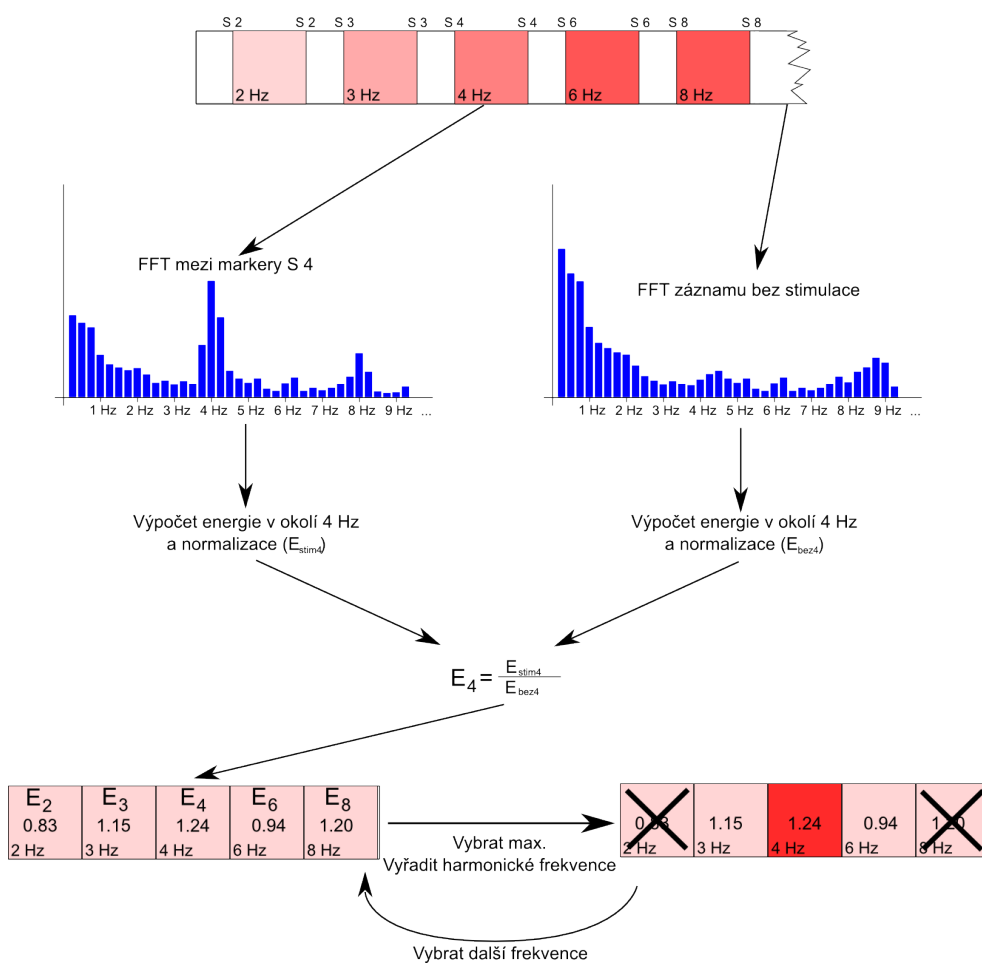
## 6.7 Modul ovládání modelu auta

Po zpracování dat klasifikátor určí, do jaké třídy jsou aktuální data zařazována. Pokud jsou tedy tři třídy (jedna určuje směr vlevo, druhá směr vpravo a třetí směr rovně), lze vyslat autíčku pomocí bezdrátového zařízení pokyn pro určité chování. Předpokládá se, že model auta jede rovně konstantní rychlostí a odesílány jsou pouze pokyny o zatáčení vlevo a vpravo, na které reaguje odpovídajícím způsobem. Implementace tohoto modulu není součástí práce.

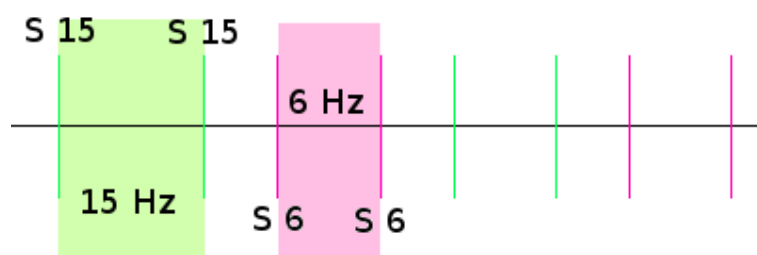
## 6.8 Demonstrační programy

Pro ověření funkčnosti jednotlivých modulů byly vytvořeny dva programy. Nejedná se přímo o součást BCI, ale pro bezchybnou implementaci modulů





Obrázek 6.11: Schéma algoritmu pro výběr vhodné stimulační frekvence



Obrázek 6.12: Umístění markerů v záznamu

bylo zapotřebí vytvořit tyto programy z důvodu testování. Jeden z programů umožňuje načítání dat ze souboru a druhý je navržen pro real-time komunikaci.

### 6.8.1 Program pro načítání dat ze souboru

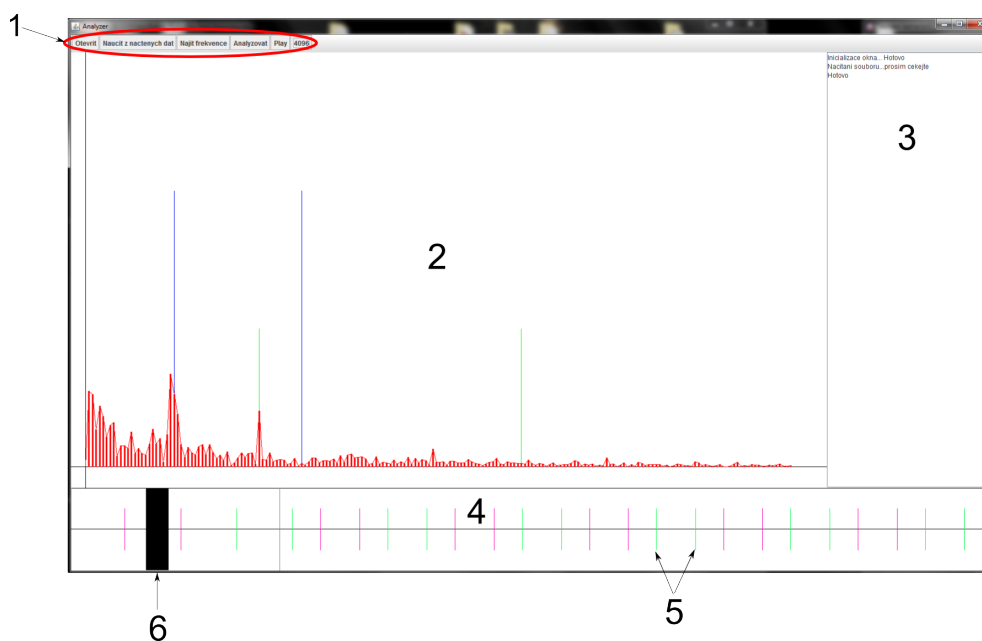
V tomto programu bylo naimplementováno několik funkcí. Jedná se o načtení dat ze souboru, dále v načtených datech umožňuje najít nejvýraznější frekvence. Z dat lze také natrénovat klasifikátor a následně klasifikovat do tří tříd. Vizuálně program zobrazuje časovou osu záznamu, frekvenční spektrum a po trénování zobrazí klasifikační prostor s jednotlivými třídami, z čehož lze předpokládat následnou úspěšnost klasifikace. Pro objektivní testování byla přidána ještě třída pro výpočty úspěšnosti klasifikace.

#### Formát a načtení vstupních dat

Program na vstupu očekává data s přesně definovanými markery. Každý úsek se stimulací musí být označen markery s identifikátorem charakterizujícím frekvenci (obrázek 6.12). Markery jsou totiž jediný prostředek, který program má k rozlišení typu dat.

Pro načtení dat je třeba zmáčknout tlačítko *Otevírat*. Následně se otevře dialogové okno pro výběr souboru. Po otevření se načte základní rozhraní složené z několika částí (obrázek 6.13).

1. Ovládací prvky programu

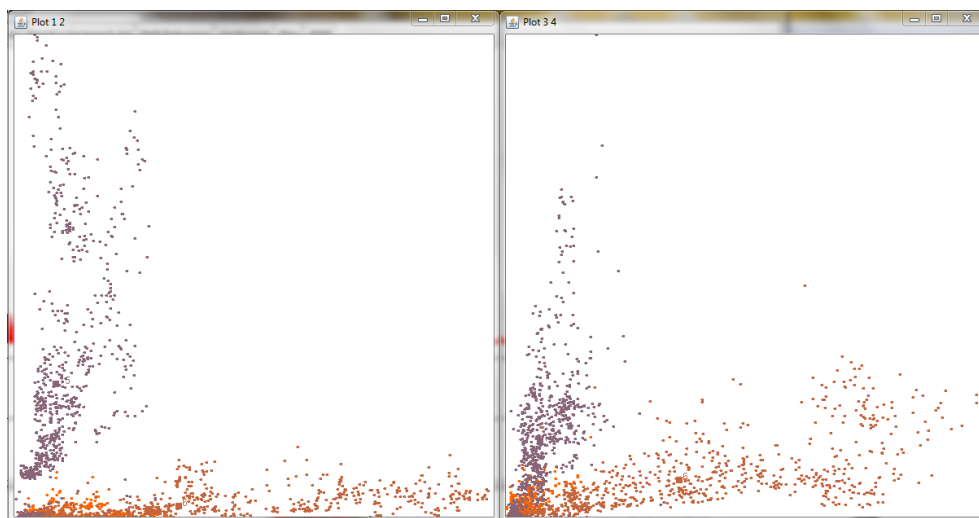


Obrázek 6.13: Základní rozhraní programu

2. Zobrazení frekvenčního spektra ze zvolené oblasti
3. Panel zobrazující informace o stavu programu a výsledcích klasifikace
4. Časová osa
5. Markery na časové ose označující úsek stimulace
6. Aktuální rozsah dat, ze kterých se vypočítává frekvenční spektrum

### Hledání optimálních frekvencí

Pokud je načtený záznam, ve kterém se nachází větší množství úseků dat s různou stimulační frekvencí, lze vybrat frekvenci, která měla nejvyšší odezvu. Po zmáčknutí tlačítka *Najít frekvence* se zobrazí dialogové okno, které se ptá, kolik frekvencí je třeba vybrat. Po potvrzení může výpočet chvilku trvat. Následně jsou vybrané frekvence vypsány v pravém panelu.



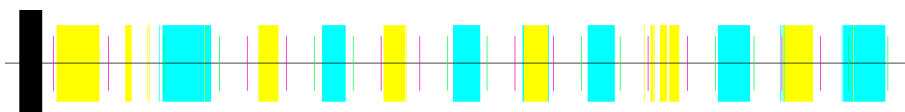
Obrázek 6.14: Klasifikační prostor s třemi třídami

### Trénování dat

Jsou-li načtena trénovací data, lze z nich natrénovat klasifikátor pomocí tlačítka *Naucit z nactených dat*. Po zmáčknutí tlačítka je uživatel dotázán, které frekvence se budou hledat. Tento klasifikátor byl navržen pouze pro dvě frekvence. Přednastaveny jsou hodnoty 6 Hz a 15 Hz, což jsou hodnoty, s kterými byla prováděna většina experimentů. Když proběhne úspěšné natrénování klasifikátoru, jsou zobrazeny dvě okna zobrazující klasifikační prostor, jednotlivé výskyty a těžiště. Jedno z oken vynáší na osy jako příznaky hledané frekvence a druhé jejich první harmonické frekvence. Jedna z tříd je zpravidla podél jedné osy, druhá podél druhé osy a třetí třída reprezentující data bez stimulace je poblíž počátku (obrázek 6.14).

### Klasifikace

Klasifikaci je možno provádět na libovolných datech, pokud je klasifikátor natrénovaný. Pokud je však zapotřebí získat statistiky o úspěšnosti klasifikace, musí být data známá a označená markery. Klasifikace se spustí tlačítkem *Analyzovat*. Po výpočtu je barvami označena časová osa a do pravého panelu jsou vypsány informace o úspěšnosti. Ideální stav klasifikace je takový, že osa je obarvena pouze mezi dvěma markery. Relativně úspěšná klasifikace na ose



Obrázek 6.15: Časová osa po klasifikaci

vypadá jako na obrázku 6.15. Žlutá barva označuje úsek dat klasifikovaných jako 6 Hz, azurová barva data klasifikované jako 15 Hz. Červené čárky ohraňují místa se stimulační frekvencí 6 Hz a zelené čárky místa, kde probíhala stimulace 15 Hz.

### Statistiky úspěšnosti klasifikace

Statistiky počítá jedna z tříd. Je třeba říci, že se nejedná o testování klasifikátoru, jako spíše testování experimentu celkově. Data, která jsou klasifikována, nemusí být vždy korektní. Mohou být ovlivněna různými artefakty a nesoustředěností osoby. Dalším důvodem je místo v záznamu, kde stimulace začíná nebo končí. V tomto místě je velká část zpracovávaných dat z úseku bez stimulace, tudíž dochází k chybné klasifikaci na počátku a na konci každé epochy.

Tato chyba však není nijak kritická pro využití v BCI, neboť se jen zpozdí odpověď a model auta zatočí o něco později. Daleko větší chyba je, pokud je v místě bez stimulace klasifikována jedna z frekvencí. Z důvodu různé závažnosti chyb byla do hodnocení úspěšnosti zanesena váha jednotlivých chyb.

Statistiky vypisují úspěšnost rozpoznávání, která hodnotí všechny chyby stejně a tudíž má nižší hodnoty. Druhou měřenou statistikou je tzv. vážená chyba, která rozděluje výše popsané „malé“ a „velké“ chyby. Váhy těchto chyb lze měnit, ale v programu byla nadefinována velká chyba jako 1, malá chyba jako 0,5 (poloviční chyba) a správná klasifikace jako 0.

### 6.8.2 Program pro real-time komunikaci

Tento program byl vytvořen pro otestování základní funkčnosti real-time komunikace a pro ověření použitelnosti tohoto přístupu pro BCI. Může být

použit jako základní kostra pro program řídicí BCI. Další možností je doplnit program o modul pro řízení modelu auta a použít přímo tento program.

Pokud by byl tento program použit pro finální BCI, bude obsahovat několik nástrojů, které jsou zbytečné. Nicméně pro ověření funkčnosti byly tyto nástroje nutné.

## Načítání dat

Načítání dat probíhá postupně v závislosti na serveru, který data poskytuje. Postup získávání a ukládání dat byl popsán v implementaci modulu pro real-time načítání dat. Jelikož server vytváří software BrainVision Recorder bylo by nutné veškeré testy provádět v laboratoři. Dokonce i testy modulů nesouvisejících s načítáním dat. Z tohoto důvodu byla vytvořena třída *Test*.

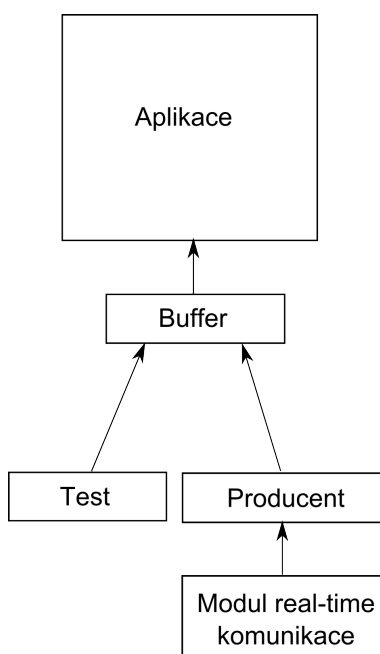
## Testovací generátor dat

Pro testování programu bylo nutné vytvořit třídu, která by generovala data. Tyto data jsou generována dle našich požadavků v závislosti na testovaném modulu. Generovaná data mají většinou povahu sinusového signálu, který může být doplněn o určené množství šumu. Lze střídat několik různých frekvencí v určených časových intervalech.

Tato třída *Test* nenahrazuje server, ale generuje data přímo do bufferu, čímž odřízne celou vrstvu starající se o komunikaci se serverem (obrázek 6.16). Výhodou tohoto přístupu je to, že lze odladit vyšší vrstvy programu na datech, která zcela jistě nejsou poškozena a pokud při použití v laboratoři dochází k problému, bude to zcela jistě vinou síťové komunikace.

## Vizualizace dat

Aktuální data, která jsou v bufferu, lze vizualizovat několika způsoby. Jedná se o normální zobrazení dat z bufferu, zobrazení frekvenčního spektra vytvořeného pomocí DFT a FFT a také spektrogramu.

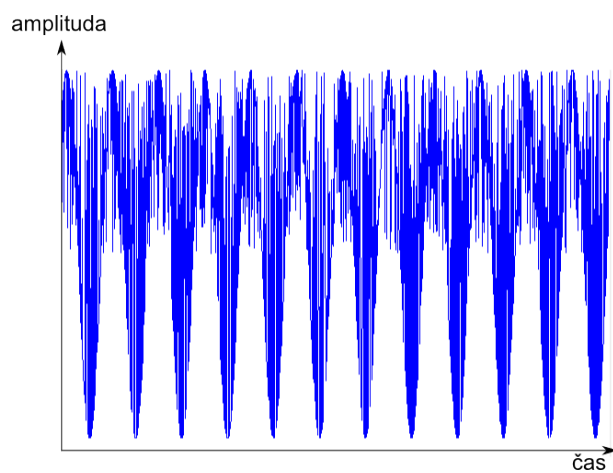


Obrázek 6.16: Schéma napojení generátoru dat na aplikaci

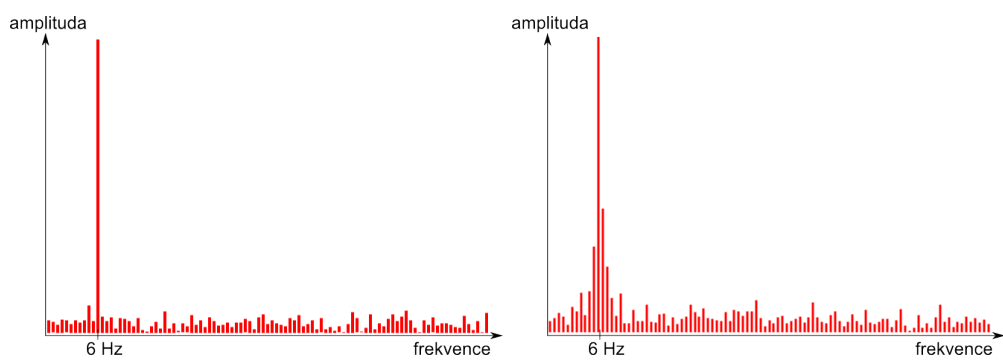
**Hrubá data** zobrazují v grafu hodnoty z bufferu. Nejsou nijak upravována. Na obrázku 6.17 je ukázka dat pro sinus s frekvencí 6 Hz a s 50% šumem.

**Frekvenční spektrum** je možno zobrazit dvěma způsoby. Jeden způsob je pomocí FFT a druhý pomocí DFT. Rozdíl v těchto možnostech je ten, že každé z těchto spekter má spektrální čáry na jiných frekvencích. Při používané vzorkovací frekvenci 1000 a velikosti bufferu 2000 jsou spektrální čáry DFT po 0,5 Hz, zatímco FFT je má umístěné mimo „hezké“ frekvence. Jak lze vidět na obrázku 6.18, zatímco spektrum z DFT obsahuje spektrální čáru přímo na 6 Hz, u FFT se 6 Hz nachází mezi dvěma čarami, čímž dochází k rozlítí mezi více čar, což je nežádoucí.

**Spektrogram** Poslední implementovanou možností zobrazení dat je spektrogram. Spektrogram uchovává kromě aktuálních dat i data starší a ukazuje vývoj frekvencí v čase. Implementován byl 2D spektrogram, u kterého tmavší barva znamená vyšší četnost výskytů určité frekvence. Na obrázku 6.19 lze vidět spektrogram obsahující nejprve jen šum, v další části se přidává 6 Hz frekvence, která je dále vystřídána 10 Hz frekvencí.

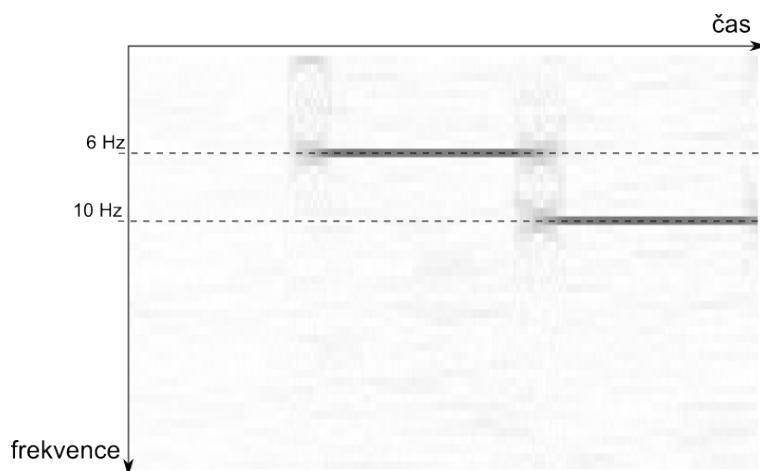


Obrázek 6.17: Vizualizace dat z bufferu



Obrázek 6.18: Vlevo spektrum vypočtené metodou DFT, vpravo metodou FFT





Obrázek 6.19: Spektrogram

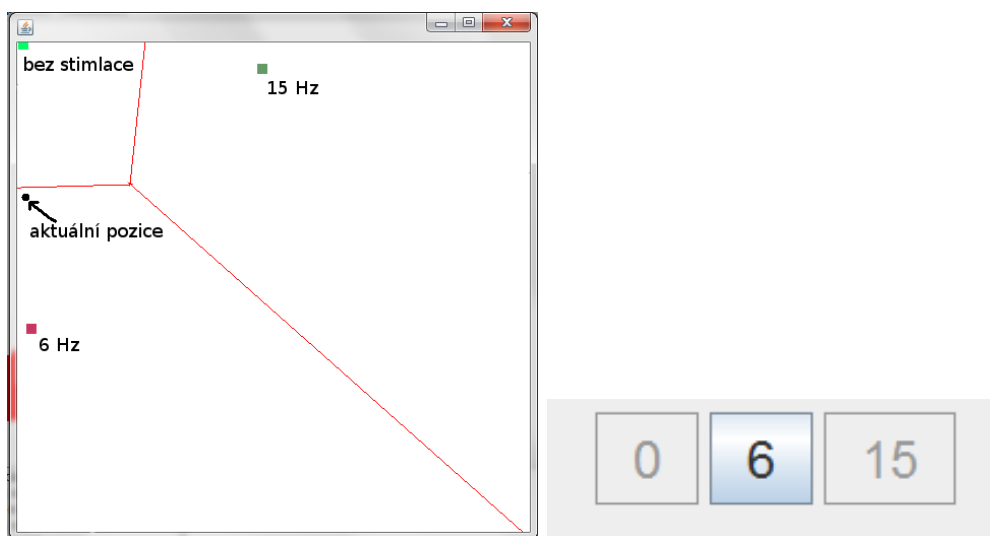


Obrázek 6.20: Tlačítka znázorňující klasifikační třídy

### Trénování klasifikátoru

Pro natrénování klasifikátoru slouží položky v menu *Klasifikace*. Pokud je osoba stimulovaná nějakou frekvencí a buffer je dostatečně naplněný, lze začít trénování tlačítkem *Začít ucení*. Pro ukončení trénování dané frekvence slouží tlačítko *Ukončit ucení*. Po ukončení učení se program ještě zeptá, k jaké frekvenci tyto data náleží. Pokud trénujeme data bez stimulace, vyplníme políčko číslem 0 nebo -1. Tímto způsobem lze nadefinovat libovolný počet tříd. U tohoto programu již neplatí omezení na tři třídy jako v programu pro načítání dat ze souboru.

Po natrénování určité frekvence se v pravé části programu objeví tlačítka s identifikátorem frekvence (obrázek 6.20). Tyto tlačítka jsou neaktivní. Po začátku klasifikace se stává aktivní tlačítko, které odpovídá aktuálně klasifikovaným datům.



Obrázek 6.21: Klasifikace aktuálních dat

## Klasifikace

Pokud proběhlo úspěšně trénování klasifikátoru, lze spustit klasifikaci. Klasifikace se spouští tlačítkem *Zacít klasifikaci*. Po spuštění klasifikace se otevře okno vykreslující podprostor klasifikačního prostoru pro dva příznaky a zobrazuje tři třídy a pozici aktuálních dat. Tento prvek má význam zejména pokud probíhá klasifikace do tří tříd. Konkrétní příklad může vypadat jako na obrázku 6.21, kde příznakový vektor aktuálních dat spadl do třídy 6 Hz.

## 7 Ověření funkčnosti modulů

Ověřování funkčnosti probíhalo různým způsobem v závislosti na možnostech. Jelikož všechny moduly na sobě více či méně závisí, byl občas problém testovat vše jednotlivě.

### 7.1 Načítání EEG dat

Při načítání dat nelze přesně kontrolovat vstup. Ověřit správnost lze jen porovnáním se softwarem BrainVision Analyzer, který lze brát jako referenční. Druhou možností je ověřit ostatní moduly na testovacím generátoru dat a při výměně generátoru za modul načítání dat musí vše fungovat stejně jako předtím. V opačném případě se zcela jistě jedná o chybu v načítání dat. V tomto případě byla použita druhá metoda, neboť přístup k softwaru BrainVision Analyzer je omezený a nachází se na jediném počítači v laboratoři.

Načítání dat ze souboru se ukázalo funkční bez jakéhokoliv problému a klasifikace probíhala relativně spolehlivě. Malé chyby v klasifikaci nejsou v žádném případě způsobené načítáním dat.

Načítání dat v reálném čase lze porovnávat během měření. Pokud sledujeme záznam v softwaru BrainVision Recorder a vizualizaci dat v programu, je vidět stejná křivka. Tento fakt prokazuje správné načítání hodnot, což následně potvrdila i klasifikace. Zde však občas nastává problém se stabilitou, kdy z neznámého důvodu přestane pracovat posílání dat a je třeba spojení znovu obnovit.

### 7.2 Zpracování EEG dat

Zpracování EEG dat v tomto případě spočívá v průměrování signálů z různých elektrod a následném zpracování pomocí Fourierovy transformace. Průměrování jednotlivých kanálů je zcela průhledná operace popsána v implementaci.

### 7.2.1 FFT a DFT

Zajímavější je to ovšem s Fourierovou transformací. Správnost výsledků lze ověřit prostudováním frekvenčního spektra. Tento krok je ovšem spojený s vizualizací a vygenerováním vhodného signálu pro zpracování. Jak již bylo řečeno, program používá diskrétní a rychlou Fourierovu transformaci. Pokud bude vygenerován sinusový signál o frekvenci například 6 Hz při vzorkovací frekvenci 1000 Hz a Fourierova transformace bude počítána z 2000 vzorků, teoreticky by mělo vyjít spektrum následovně.

**Pro DFT** zůstávají všechny parametry tak, jak byly zadány. Nejprve je třeba zjistit po jaké frekvenci se budou jednotlivé spektrální čáry vyskytovat. Zadání příkladu je tedy  $f[k] = 6Hz$ ,  $f_s = 1000Hz$ ,  $N = 2000$  a  $k$  je neznámé. Ze vzorce 4.9 je tedy třeba vyjádřit  $k$ .

$$k = \frac{f[k] \cdot N}{f_s}$$

Po dosazení vyjde

$$k = \frac{6 \cdot 2000}{1000} = 12$$

Nyní víme, že jediná čára celého spektra bude na dvanácté pozici, ostatní hodnoty budou nulové.

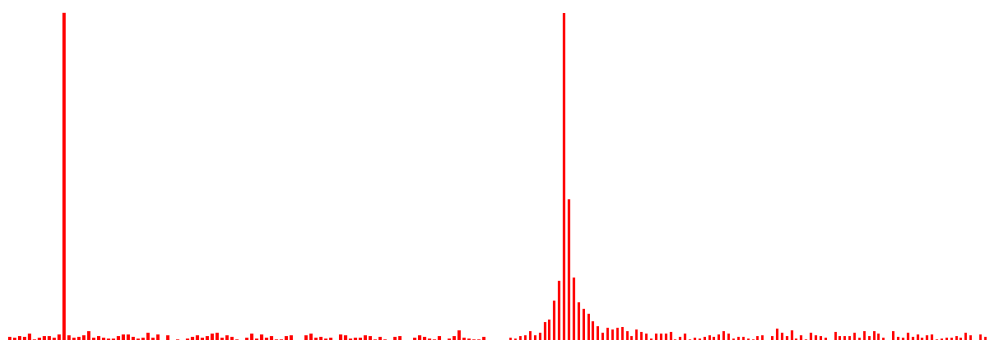
**Pro FFT** nastává oproti DFT jedna změna. Jelikož pro FFT je nutné, aby bylo  $N$  mocninou 2, bude doplněna posloupnost hodnot nulami na délku 2048. Všechny hodnoty zůstávají stejné jako v předchozím případě, jen  $N = 2048$ . Po dosazení tedy dostaneme:

$$k = \frac{6 \cdot 2048}{1000} = 12,288$$

Jelikož se požadovaná frekvence nachází mimo spektrální čáru, bude rozložena do více čar, nicméně nejvyšší amplitudy by měla nabývat stále dvanáctá hodnota.

### Výsledky testu

Pro oba případy bylo vykresleno spektrum (obrázek 7.1). Kromě sinusového signálu bylo do generovaných dat přidáno malé množství šumu, aby bylo



Obrázek 7.1: Spektrum DFT vlevo a FFT vpravo

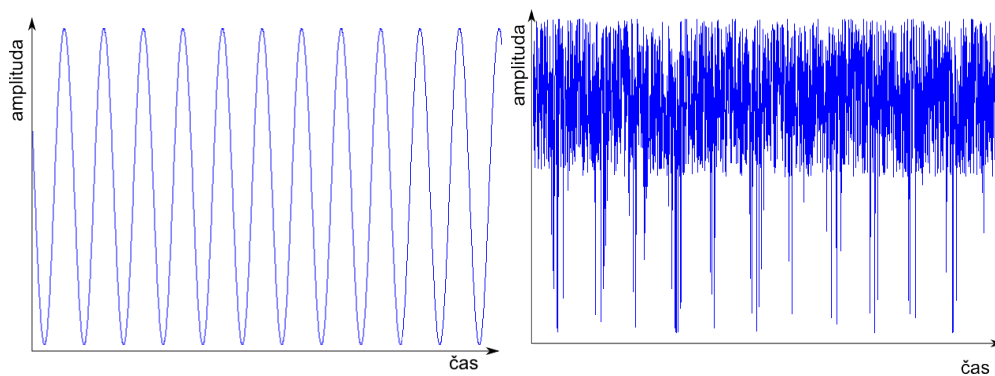
možné odpočítat spektrální čáry. Test dopadl dle předpokladů. Pro DFT se nachází jediná spektrální čára na dvanácté pozici. Spektrum FFT se rozlilo okolo dvanácté spektrální čáry z důvodu neceločíselné hodnoty. Test vyšel přesně dle předpokladů, lze tedy považovat implementaci DFT a FFT za správnou.

### 7.3 Klasifikace

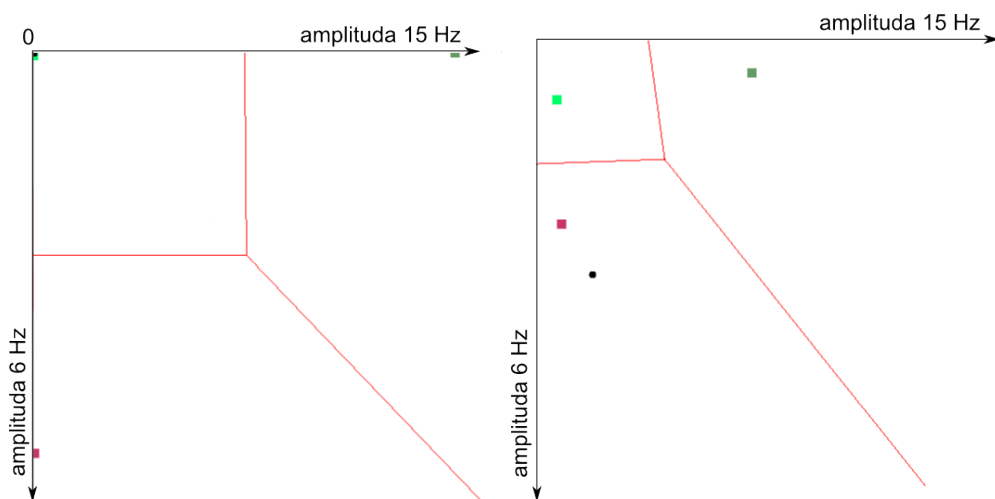
Klasifikátor byl testován zejména pro periodické signály s různou úrovní šumu. Na reálných datech byl testován také, ale tam to již nevypovídá tolik o kvalitě klasifikátoru jako o celkovém úspěchu či neúspěchu experimentu.

Jelikož v BCI je třeba klasifikace do tří tříd, základní test byl navržen jako střídání šumu, 6 Hz sinusovky a 15 Hz sinusovky. Samozřejmě, že taková klasifikace probíhá naprosto spolehlivě. Příznakový vektor tvoří amplitudy okolí 6 Hz a 15 Hz. Pokud tento případ bude vyneseno do příznakového prostoru, vidíme výsledek na obrázku 7.3 vlevo. Nenachází se žádný šum, jednotlivé třídy leží přímo na ose a dokonce většina výskytů dané třídy se nachází přímo v nejbližším okolí těžiště třídy.

Stejným způsobem funguje klasifikace i pro větší procenta šumu v signálu s tím rozdílem, že se zmenšují vzdálenosti tříd. Klasifikátor funguje zcela spolehlivě do velmi vysokého procenta šumu. Až kolem 95% šumu nastávají občasné chyby klasifikace, ale většina dat je i přesto klasifikována správně. Z dat s takovou hodnotou šumu již jen těžko pouhým okem rozpoznat o jaký signál se původně jednalo. Na obrázku 7.2 je zobrazen sinus s frekvencí 6 Hz.



Obrázek 7.2: Funkce sinus o frekvenci 6 Hz vlevo bez šumu, vpravo s 95% šumem



Obrázek 7.3: Příznakový prostor pro 3 třídy bez šumu vlevo a s 95% šumem vpravo

Vlevo bez šumu, vpravo s 95% šumem. Pokud se podíváme na příznakový prostor v tomto případě (obrázek 7.3 vpravo), třídy jsou již daleko blíže u sebe než v signálu bez šumu. Kromě toho nejsou přímo na osách, neboť kvůli šumu je výrazný i příznak, který by se u určité frekvence vůbec neměl vyskytovat. Jednotlivé výskyty mají od těžiště tříd větší vzdálenost. Z tohoto důvodu může občas docházet k chybě klasifikátoru.

Klasifikátor je schopen správně klasifikovat data i s velkým množstvím šumu. Lze tedy předpokládat, že při klasifikaci EEG dat bude úspěšnost relativně vysoká.

## 7.4 Celkový test

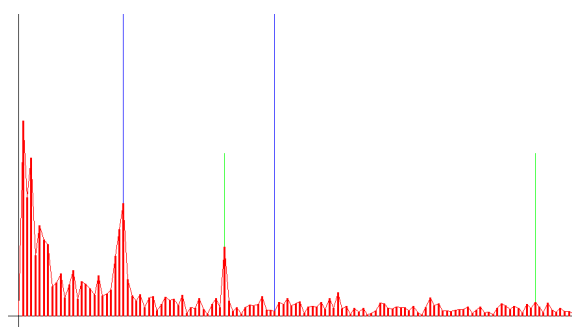
V tomto testu se jednalo o zjištění úspěšnosti klasifikace na reálných datech. Testování probíhalo na programu pro načítání dat ze souboru. Jelikož tento program obsahuje stejný způsob zpracování a klasifikace, lze výsledky považovat za objektivní i v rámci zpracování v reálném čase v druhém programu.

### 7.4.1 Popis experimentu

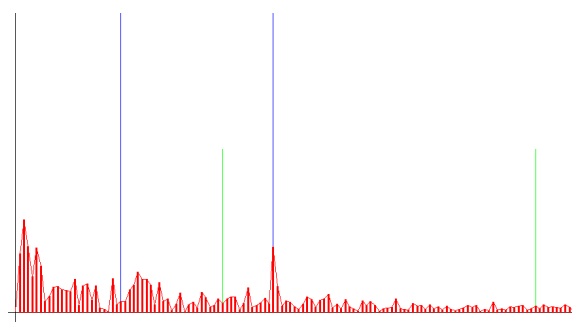
Pro tento experiment byla použita data naměřená na jaře roku 2010. Subjekty byli muži ve věku 18 - 25 let. Celkový počet subjektů byl 5. Každý z nich sledoval postupně čtyři scénáře. Jednalo se o stimulaci čtverečkem a šipkou v červené a modré barvě zobrazovanými na monitoru počítače. Frekvence blikání byly 6 Hz a 15 Hz. Pro každý scénář bylo naměřeno 12 epoch o délce 5 - 10 sekund.

### 7.4.2 Zpracování dat

U naměřených dat byly kromě statistik sledovány i grafy frekvenčních spekter. Ve všech případech by grafy měly vypadat přibližně jako na obrázcích 7.4, 7.5 a 7.6. V grafu vždy první modrá čára označuje 6 Hz a druhá modrá 15 Hz, což jsou stimulační frekvence. Zelené čáry jsou jejich první harmonické frekvence. Na obrázku 7.4 se jedná o stimulaci 6 Hz stimulem. Je vidět vysoká amplituda v okolí spektrální čáry 6 Hz a dokonce i na 12 Hz. U obrázku 7.5 se jedná o 15 Hz stimulaci. Amplituda již není tak vysoká. Poslední

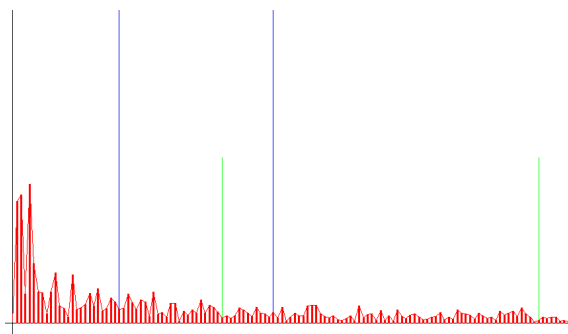


Obrázek 7.4: Frekvenční spektrum při 6 Hz stimulaci



Obrázek 7.5: Frekvenční spektrum při 15 Hz stimulaci





Obrázek 7.6: Frekvenční spektrum bez stimulace

z obrázků 7.6 ukazuje spektrum normálního stavu bez stimulace.

Tyto obrázky demonstrují relativně dobrá data. U mnoha subjektů frekvenční spektra při stimulaci nevykazují žádné pravidelnosti a pouhým okem (většinou ani pomocí klasifikátoru) nelze dosáhnout vysokých úspěšností při klasifikaci.

### Subjekt 1

Sada dat	Úspěšnost	Chyba	Vážená úspěšnost	Vážená chyba
Trénovací data	0.8265	0.1735	0.8874	0.1126
Data 1	0.7776	0.2224	0.8765	0.1235
Data 2	0.6817	0.3183	0.8246	0.1754
Data 3	0.6971	0.3029	0.8291	0.1709
Průměr	0.7457	0.2542	0.8544	0.1456

Z výsledků je patrné, že osoba reagovala na modrý stimul méně než na červený. Dále je také zřejmé, že chyby při klasifikaci jsou většinou malé (tzn. neoznačení žádné frekvence během blikání), neboť vážená úspěšnost rozpoznávání je řádově o 10% vyšší. Na tomto faktu se velkou měrou podílí postupný náběh a seběh na začátcích a koncích epoch.

### Subjekt 2

Sada dat	Úspěšnost	Chyba	Vážená úspěšnost	Vážená chyba
Trénovací data	0.7234	0.2766	0.8218	0.1782
Data 1	0.7878	0.2122	0.8817	0.1183
Data 2	0.6248	0.3752	0.7126	0.2874
Data 3	0.6835	0.3165	0.7913	0.2087
Průměr	0.7049	0.2951	0.8019	0.1982

U druhého subjektu jsou výsledky velmi podobné prvnímu.

### Subjekt 3

Sada dat	Úspěšnost	Chyba	Vážená úspěšnost	Vážená chyba
Trénovací data	0.7858	0.2142	0.8819	0.1181
Data 1	0.7653	0.2347	0.8784	0.1216
Data 2	0.7748	0.2252	0.8747	0.1253
Data 3	0.5563	0.4437	0.7604	0.2396
Průměr	0.7206	0.2795	0.8489	0.1512

Třetí osoba měla dobré reakce i na modrou barvu, jak je vidět z řádku Data 2, avšak poslední sada dat zůstává v úspěšnosti rozpoznávání daleko pod ostatními. Šlo zřejmě o nesoustředěnost subjektu.

### Subjekt 4

Sada dat	Úspěšnost	Chyba	Vážená úspěšnost	Vážená chyba
Trénovací data	0.6846	0.3154	0.7673	0.2327
Data 1	0.6758	0.3242	0.7502	0.2498
Data 2	0.6365	0.3635	0.6933	0.3067
Data 3	0.5953	0.4047	0.6700	0.3300
Průměr	0.6481	0.3520	0.7202	0.2798

U tohoto subjektu jsou celkově úspěšnosti rozpoznávání velice nízké. Subjekt se pravděpodobně soustředil nesprávným způsobem a systém BCI by v tomto případě selhával.

### Subjekt 5

Sada dat	Úspěšnost	Chyba	Vážená úspěšnost	Vážená chyba
Trénovací data	0.7065	0.2935	0.8051	0.1949
Data 1	0.7461	0.2539	0.8268	0.1732
Data 2	0.6615	0.3382	0.7617	0.2383
Data 3	0.5711	0.4289	0.7154	0.2846
Průměr	0.6713	0.3286	0.7773	0.2228

Výsledky opět ukazují horší odezvu subjektu na modrou barvu.

### 7.4.3 Závěr experimentu

Ze získaných hodnot je jasně patrné, že stimulace modrou barvou na většinu subjektů nepůsobí tak výrazně jako barva červená. Dále lze pozorovat, že většina klasifikačních chyb je na začátcích a koncích epoch, kde je dlouhý náběh a seběh. Pokud bychom tyto chyby nebrali jako chyby (tj. nastavit parametry vážené chyby na 1; 0; 0), dostali bychom se v některých případech s úspěšností klasifikace až k 95%. Při nastavení, jaké bylo použito v tabulkách (neoznačení dat, které mají být označeny je poloviční chyba), se úspěšnost klasifikace pohybuje u červené barvy od 75% do 90% v závislosti na subjektu. U modré barvy se pak úspěšnost pohybuje řádově o 10% níže.

Pokud se jedná o použitelnost tohoto principu pro oblasti BCI, může být systém relativně spolehlivý. Zejména pokud subjekt bude mít odezvu, zda jeho soustředění je správné nebo ne. Je jasné, že každý člověk nedosáhne vysoké spolehlivosti hned. Největší neznámou je zde vždy testovaná osoba.

## 8 Závěr

Diplomová práce se zabývá aplikací ustálených vizuálních evokovaných potenciálů v oblasti BCI. Jelikož se jedná o téma zacházející ve velké míře do medicíny, bylo zapotřebí nastudovat množství materiálů týkajících se vyšetření mozku a evokovaných potenciálů. Na základě těchto znalostí bylo možné připravit stimulační scénář a následně testovat moduly pro zpracování a klasifikaci dat. Tyto moduly byly vyrobeny jako možná součást připravovaného BCI, kde je ovládáno robotické vozítko právě pomocí ustálených vizuálních evokovaných potenciálů.

Práce je členěna na osm kapitol. Po krátkém úvodu se nachází kapitola s názvem Elektroencefalografie. V této části jsou stručně probrány základní principy měření EEG, typy mozkových vln a také problémy, na které lze při měření narazit. Následuje kapitola Evokované potenciály obsahující souhrn informací o jednotlivých typech evokovaných potenciálů. Důraz je především kladen na vizuální evokované potenciály, které jsou zde popsány detailněji. Další část se zabývá metodami pro zpracování EEG. Jedná se zejména o různé druhy Fourierových transformací. Poslední kapitola teoretické části se zaměřuje na klasifikaci. Kvůli povaze zpracovávaných dat jsou zde probrány pouze příznakové metody. Návrh a implementace BCI se zabývá už vlastním návrhem systému ovládání robotického vozítka pomocí SSVEP. Jsou zde popsány veškeré postupy při výrobě stimulačního scénáře, zpracování dat, klasifikace a další. Jednotlivé moduly popsané v této části jsou v další kapitole testovány. Testování modulů je zakončeno komplexním testem pro ověření funkce celkového principu a zjištění využitelnosti SSVEP v oblasti BCI.

Během implementace a testování nenastaly výraznější komplikace. Většina modulů pracujících s real-time daty byla testována převážně na generovaných datech, neboť propojení s přístrojem pro měření EEG bylo možné pouze v laboratoři. I přes tento fakt při následném testu v laboratoři vše fungovalo jak mělo a klasifikace probíhala u měřené osoby spolehlivě.

Program pro načítání dat ze souboru byl používán při vyhodnocení spolehlivosti klasifikátoru. Z výsledků je vidět, že klasifikace dosahuje úspěšnosti až 90%. Jelikož tyto testy byly prováděny na náhodných subjektech bez zpětné vazby, lze předpokládat, že při ovládání robotického vozítka bude úspěšnost klasifikace ještě vyšší.

Diplomová práce ukázala, že lze použít princip SSVEP pro ovládání robo-

tického vozítka. K realizaci tohoto systému práce přispěla několika moduly, které byly ověřeny a lze je nasadit. Práci lze považovat za úspěšnou, neboť přispěla k vytvoření zajímavého BCI systému a splnila veškeré požadavky na ni kladené.

# Literatura

- [1] FORNBAUM, M. *Bakalářská práce: Ustálené vizuální evokované potenciály a jejich aplikace v oblasti BCI*. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky, 2010.
- [2] HEŘMAN, P. *Biosignály z pohledu biofyziky*. Petr Heřman – DÚLOS, 2006.
- [3] KAŇOVSKÝ, P. *Evokované potenciály v klinické praxi*. Brno : Institut pro další vzdělávání pracovníků ve zdravotnictví, 2000.
- [4] KOLÁŘ, R. *Lékařská diagnostická technika*. VUT Brno, 2006.
- [5] MAŘÍK, V. *Umělá inteligence 1*. Academia Praha, 1993.
- [6] MAUTNER, P. Přednášky KIV/AZS. Přednášky z předmětu KIV/AZS.
- [7] *Presentation documentation*. Neurobehavioral Systems, 2012. Dostupné z: <http://www.neurobs.com/presentation/docs>.
- [8] ROZMAN, J. *Elektronické přístroje v lékařství*. Academia, 2006.
- [9] ZHU, D. A Survey of Stimulation Methods Used in SSVEP - Based BCIs. 2010. Dostupné z: <http://www.hindawi.com/journals/cin/2010/702357.html>.