

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Martin Kamenšek

**Simulacija premikanja grafičnega  
kazalca z zamišljanjem motoričnih  
aktivnosti**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJ

MENTOR: prof. dr. Franc Jager

Ljubljana 2014



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\LaTeX$ .*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Razvijte grafični sistem za simulacijo pomikanja grafičnega kazalca po računalniškem zaslonu z uporabo posnetkov elektroencefalograma (EEG), medtem ko si testna oseba zamišlja motorične aktivnosti pomikanja grafičnega kazalca. Za izvirne posnetke uporabite posnetke mednarodne podatkovne baze EEGMMI DS (EEG Motor Movement/Imagery DataSet). Arhitektura razvitega sistema naj temelji na nevronskih mrežah s povratnim razširjanjem, katere vhodi so frekvenčne komponente amplitudnih spektrov možganskih valov  $\mu$  in  $\beta$  signalov EEG. Razvijte tri protokole, ki vključujejo: 1) odpiranje in zapiranje leve in desne pesti; 2) odpiranje in zapiranje leve in desne pesti z uporabo optimizirane učne množice; in 3) simultano odpiranje in zapiranje obeh pesti. Vrednotite zmogljivost razvitega sistema.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Martin Kamenšek, z vpisno številko **63080223**, sem avtor diplomskega dela z naslovom:

*Simulacija premikanja grafičnega kazalca z zamišljanjem motoričnih aktivnosti*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Franca Jagra,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 7. oktober 2014

Podpis avtorja:





# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Vmesnik med možgani in računalnikom . . . . .	1
1.2	Namen naloge . . . . .	1
<b>2</b>	<b>Metode</b>	<b>3</b>
2.1	Možganska aktivnost . . . . .	3
2.1.1	Električna aktivnost . . . . .	3
2.1.2	Elektroencefalografija . . . . .	6
2.1.3	Motorični korteks . . . . .	8
2.2	Podatkovna zbirka	
	EEG Motor Movement/Imagery Dataset . . . . .	8
2.2.1	Sistem BCI2000 . . . . .	9
2.2.2	Protokol EEG snemanj . . . . .	10
2.2.3	Uporaba podatkov . . . . .	12
2.3	Filter s končnim impulznim odzivom . . . . .	12
2.4	Nevronska mreža . . . . .	15
2.4.1	O nevronske mreži . . . . .	15
2.4.2	Nevronska mreža s povratnim razširjanjem . . . . .	17
2.5	Protokoli premikanja kazalca . . . . .	18
2.5.1	Prvi protokol . . . . .	19

## KAZALO

2.5.2	Drugi protokol . . . . .	19
2.5.3	Tretji protokol . . . . .	21
2.5.4	Učenje mreže . . . . .	21
2.6	Metrike zmogljivosti . . . . .	22
2.7	Implementacija . . . . .	23
2.7.1	Branje podatkov . . . . .	23
2.7.2	Filtriranje in izračun značiln . . . . .	25
2.7.3	Ustvarjanje učne množice in učenje nevronske mreže . . . . .	26
2.7.4	Grafični vmesnik . . . . .	28
<b>3</b>	<b>Rezultati</b>	<b>31</b>
3.1	Prvi protokol . . . . .	31
3.2	Drugi protokol . . . . .	33
3.3	Tretji protokol . . . . .	35
<b>4</b>	<b>Diskusija in nadaljnje delo</b>	<b>43</b>
4.1	Splošno o rezultatih . . . . .	43
4.2	Nadaljnje delo . . . . .	44

# Povzetek

Cilj diplomske naloge je bil narediti sistem, ki na podlagi posnetkov elektroencefalograma (EEG) premika grafični kazalec po zaslonu v štirih smereh. Uporabljeni so EEG posnetki, dobljeni med zamišljanjem motoričnih aktivnosti, kot so stiskanje leve pesti, stiskanje desne pesti, stiskanje obeh pesti in mirovanja. Posnetki so del podatkovne zbirke EEG posnetkov EEG Motor Movement/Imagery Dataset.

Predstavljeni so trije protokoli obdelave EEG posnetkov in učenja nevronske mreže. Konfiguracija nevronske mreže je podobna za vse tri protokole, ima en skriti nivo z 64 nevroni in izhodni nivo s štirimi nevroni. Namen prvega protokola je iskanje enostavnih relacij med amplitudnima spektroma  $\mu$  in  $\beta$  valov ter namišljenimi motoričnimi akcijami stiskanja leve in stiskanja desne pesti. Z uporabo celotnega amplitudnega spektra  $\mu$  in  $\beta$  kot vhoda nevronske mreže je bila dosežena klasifikacijska natančnost 52%. V drugem protokolu je za učenje nevronske mreže uporabljeno le povprečje  $\mu$  in  $\beta$  amplitudnih spektrov, uporabljenih v prvem protokolu, poleg tega pa je učna množica optimizirana. S tem protokolom je natančnost klasifikacij dosegla 80.8%. S tretjim protokolom se je poizkusilo razširiti nabor uporabljenih tipov posnetkov tudi na posnetke stiskov obeh pesti hkrati. Končna rešitev je uporabila dve nevronske mreži. S prvo nevronske mreže je dosegla enake rezultate kot drugi protokol, mreža s klasifikacijo stiskov obeh pesti in mirovanja pa je dosegla klasifikacijsko natančnost 69.7%.

**Ključne besede:** Elektroencefalogram, Vmesnik možgani-računalnik, Zamišljanje motoričnih aktivnosti, Nevronska mreža



# Abstract

The goal of this thesis was to create a system which can perform cursor movements on a computer monitor in four directions, based on electroencephalogram (EEG) recordings. The EEG recordings were recorded while imagining of motor activities such as opening and closing of individual fists, opening and closing of both fists simultaneously. The recordings are part of a dataset of EEG recordings EEG Motor Movement/Imagery Dataset.

This thesis documents three different protocols for processing EEG records and training a neural network. Neural network configuration is similar for all three protocols and consists of one hidden layer with 64 neurons and an output layer with four neurons. The purpose of the first protocol is to search for simple relations between amplitude spectra of  $\mu$  and  $\beta$  brainwaves and imaginary motor activities of opening and closing individual fists. By using the entire amplitude spectra of  $\mu$  and  $\beta$  as an input to a neural network, a classification accuracy of 52% was achieved. In the second protocol only the average of amplitude spectra of  $\mu$  and  $\beta$  was used. Additionally the training set was optimized, which resulted in a classification accuracy of 80.8%. The third protocol is an attempt to expand the learning set and include the recordings of opening and closing of both fists simultaneously as well. The final solution used two neural networks. The first neural network is identical to the one used in second protocol, the second neural network was used to classify imaginary motor activities of opening and closing both fists simultaneously and achieved classification accuracy of 69.7%.

**Keywords:** Electroencephalogram, Brain-computer interface, Imagining motor activity, Neural Network



# 1. Uvod

## 1.1 Vmesnik med možgani in računalnikom

Vmesnik med možgani in računalnikom (angl. Brain-Computer Interface - BCI) predstavlja sistem, znotraj katerega poskušamo s pomočjo elektrod ali kakšnih drugih merilnih naprav, prepoznavati posameznikovo možgansko aktivnost in s to interpretacijo upravljati računalnik. Na področju rehabilitacije bi lahko s pomočjo BCI sistemov tako namesto klasične pasivne protetike uporabljali popolnoma interaktivno robotsko protetiko, upravljali elektrificiran invalidski voziček, ali pa nenazadnje vzpostavili primitiven komunikacijski kanal s pacienti, ki sicer niso zmožni interakcije z okoljem, recimo ljudje s sindromom zaklenjenosti (locked-in syndrom). Seveda pa je možna aplikacija takšne vrste tehnologije tudi na drugih področjih, kot je na primer zabavna industrija.

## 1.2 Namen naloge

Namen diplomske naloge je ustvariti simulacijo, ki bo na podlagi EEG posnetkov omogočila vzpostaviti osnovno komunikacijo med človekom in računalnikom. Ta komunikacija bo temeljila na detektiranju in prepoznavanju namišljenih aktivnosti rok iz EEG signala, kar se bo manifestiralo kot premiki grafičnega kazalca po zaslonu. Simulacija mora dovoliti premike kazalca v dveh dimenzijah, torej štirih smereh (levo, desno, gor, dol). Motorična aktivnost se odraža kot aktivacija specifičnih nevronske povezav v motoričnem

korteksu. Zelo zanimiva lastnost motoričnega korteksa pa je v tem, da se že ob samem zamišljanju teh motoričnih aktivnosti aktivirajo v velikem delu iste povezave kot ob dejanski motorični akciji. Za uporabnike takega sistema je zamišljanje aktivnosti veliko lažje, kot dejansko izvajanje takih aktivnosti, za nekatere pa celo nemogoče, zato bomo v tej nalogi uporabili le posnetke namišljenih motoričnih aktivnosti. V nalogi smo uporabili zbirko EEG posnetkov možganske motorične aktivnosti, EEG Motor Movement/Imagery Dataset [1, 2, 3]. Na podlagi te zbirke posnetkov smo v prvem protokolu poskusili naučiti nevronske mrežo razlikovati med amplitudnimi spektri zamišljene motorične aktivnosti stiska leve in stiska desne pesti, kar pa se ni izkazalo za najbolj uspešno. V drugem protokolu smo sistem nekoliko nadgradili in iz baze posnetkov ustvarili podmnožico učnih primerov s katerimi smo nevronske mrežo uspešno naučili razlikovati med amplitudnimi spektri zamišljanja stiska leve in stiska desne pesti. Nazadnje smo v tretjem protokolu učno množico in sistem razširili tako, da, poleg zamišljanja stiskov posamezne pesti, sistem prepozna tudi namišljene stiske obeh pesti hkrati, vendar pa, zaradi prekrivanja amplitudnih spektrov namišljenih motoričnih aktivnosti, učenje mreže ni uspelo tako kot bi si želeli.



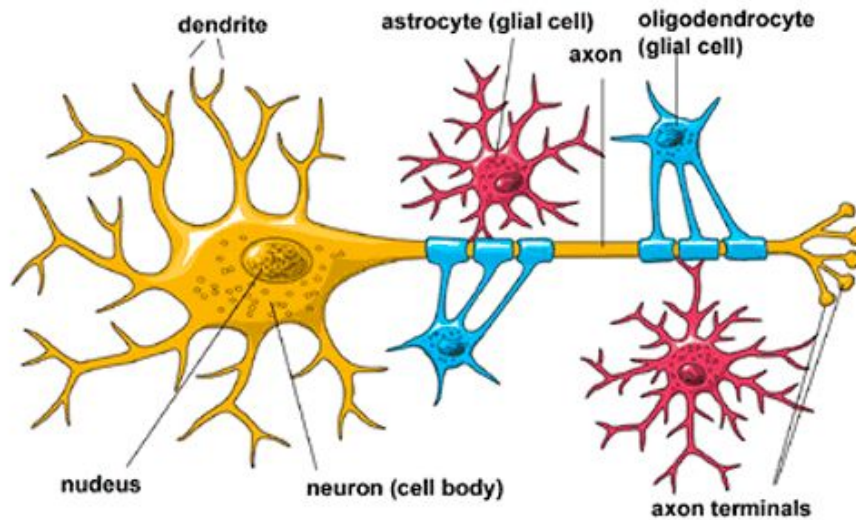
## 2. Metode

### 2.1 Možganska aktivnost

#### 2.1.1 Električna aktivnost

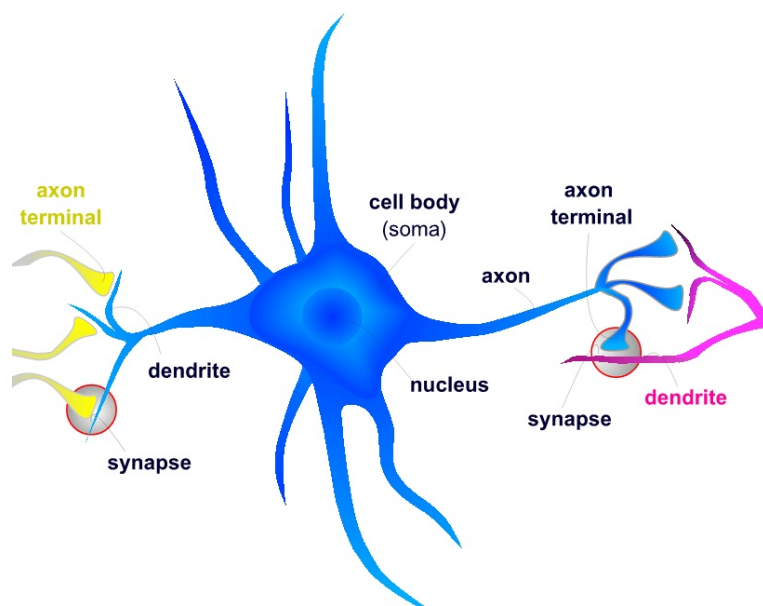
Možgani so organ, sestavljen iz živčnih celic imenovanih nevroni in živčno opornega tkiva imenovanega nevroglia (slika 2.1). Nevroglijo sestavlja več različnih vrst glija celic, nekatere delujejo kot fizična opora in preprečujejo poškodbe nevronov, druge skrbijo za vzdrževanje primerne okolja in preskrbo nevronov s hranili in kisikom. Vendar pa nevroglia ni omejena le na vzdrževanje že obstoječega sistema, nekatere vrste glija celic lahko tvorijo oziroma pomagajo pri tvorbi novih povezav med nevroni, nekatere pa celo sodelujejo pri opravljanju pomembnih funkcij, kot je dihanje. Kljub njeni nepogrešljivosti pri delovanju celotnega živčnega sistema pa za cilj te naloge podrobnosti njenega delovanja niso bistvene.

Drugi tip celic so nevroni, ki so formalno osnovna enota sposobna procesiranja informacije. Nevron sam po sebi ni sposoben zahtevnih procesiranja informacij, zato je ključno, da se nevroni povezujejo v večje mreže, kjer je posamezen nevron lahko povezan z več deset tisoč drugimi nevroni. Nevroni (slika 2.2) se med seboj povezujejo preko dveh tipov izrastkov, vhodnih dendritov in izhodnih aksonov, oziroma bolj točno, preko posebnih povezav imenovanih sinapse (slika 2.3), ki se praviloma tvorijo med končiči aksona in dendriti. Aktiviranje sinapse v povezanem dendritu povzroči neravnovesje ionov, kar pomeni pozitivno ali pa negativno spremembo električnega poten-

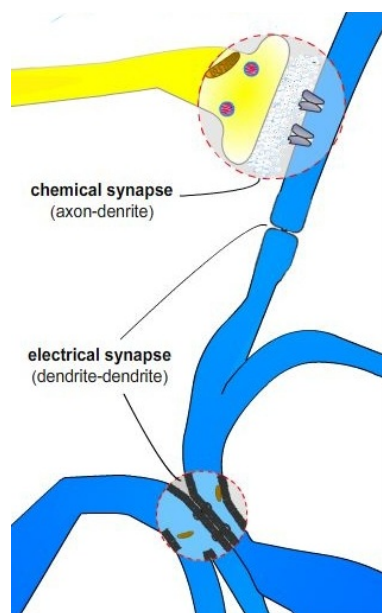


Slika 2.1: Primer nevronske celice in glia celic [4].

ciala. V primeru, da skupni potencial dendritov nekega nevrona doseže dovolj visok nivo se na korenu povezave med somo in aksonom ustvari neravnovesje ionov imenovano tudi akcijski potencial. Akcijski potencial se nato kot električni impulz širi proti koncu aksona do sinaps, ki ga vežejo na naslednje nevrone. Ustvarjanje akcijskega potenciala zaniha tudi do 100 mV od mirovnega stanja, kar je sicer težko zaznati, vendar pa velika stopnja povezanosti in interakcije med nevroni pomeni, da se v kratkem času proži večje število nevronov, njihov kumulativni električni potencial pa je lažje zaznati. Visoka stopnja povezanosti pomeni tudi, da se v nekaterih primerih na nivoju posameznih nevronov, pa tudi na nivoju širših povezav med nevroni, vzpostavi ciklično proženje akcijskih potencialov imenovano tudi ritem. V tej nalogi smo se oprli na  $\mu$  in  $\beta$  ritma, ki se pojavljata v frekvenčnih področjih 8 do 13 Hz in 18 do 24 Hz.



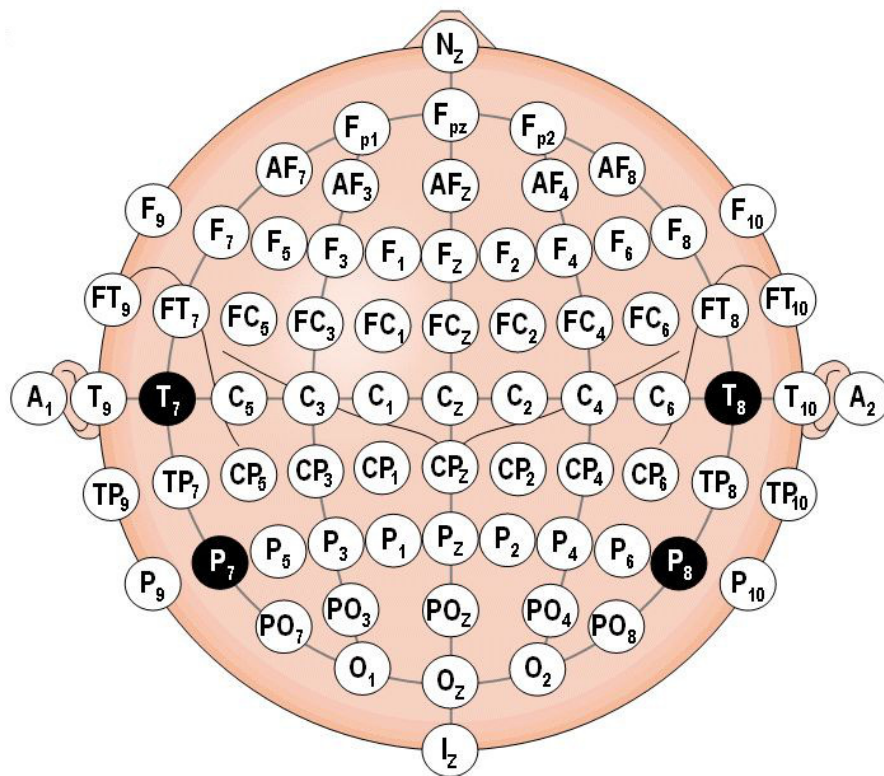
Slika 2.2: Shema posplošene zgradbe nevrona [5].



Slika 2.3: Shema prikazuje dve vrsti sinaps. Kemične sinapse povezujejo aksone in dendrite, električna sinapsa pa med seboj povezuje dendrite [5].

### 2.1.2 Elektroencefalografija

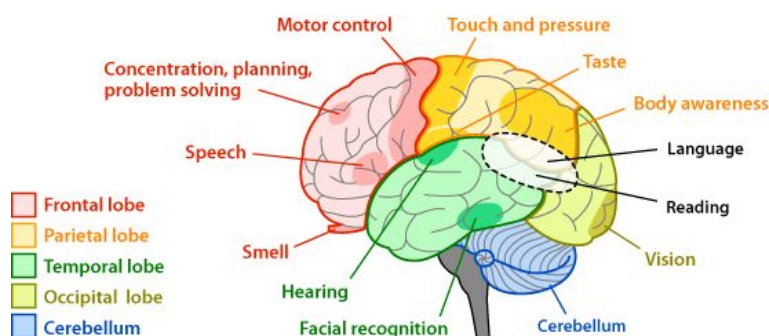
Postopek merjenja in beleženja elektromagnetnih nihanj v možganih imenujemo elektroencefalografija. Pristopov k merjenju je veliko in jih lahko v splošnem razdelimo v tri skupine: invazivne, polinvazivne in neinvazivne. Invazivni pristop zahteva vsajanje elektrod v samo možgansko tkivo, kar na kratki rok prinese najbolj točne meritve, vendar velik problem povzroča brazgotinjenje poškodovanega tkiva, ki ošumi signal. Druga skupina pristopov prav tako zahteva invazivni poseg v lobanjo, vendar pa tu elektrod ne vsajajo v možgansko tkivo, ampak le nastavijo na površino možgan ali pa lobanje. Ta pristop tudi omogoča natančne meritve, vendar pa zaradi svoje invazivnosti ni primeren za splošno uporabo. Zadnja skupina so neinvazivni pristopi polaganja elektrod na skalp, kar je relativno enostavno za uporabo, vendar pa so meritve precej slabše zaradi dušenja in motenj, ki jih povzročajo lobanjske kosti in plast mišic. V tej nalogi smo uporabili podatkovno zbirko meritev, ki so bile pridobljene na neinvazivni način s porazdelitvijo elektrod kot pri mednarodnem 10-10 sistemu. Ta sistem predvideva razporeditev 74 elektrod v mrežo, kot je prikazano na sliki 2.4, kjer sta tako navpični kot vodoravni razmak med elektrodami natanko deset odstotkov širine oziroma višine mreže.



Slika 2.4: Razporeditev merilnih elektrod po mednarodnem 10-10 sistemu [6].

### 2.1.3 Motorični korteks

V literaturi [7] so identificirana možganska področja, kjer se procesirajo določene funkcije (slika 2.5), kot so na primer vid, sluh, branje, pisanje, motorična aktivnost itd. V tej nalogi smo se osredotočili na upravljanje z motoričnimi aktivnostmi, zato nas zanima le motorični korteks. Delovanje motoričnega korteksa je seveda zelo kompleksno in še ni popolnoma raziskano, kljub temu pa je odkritih že nekaj zelo zanimivih lastnosti. Juliana Dushanova in John Donoghue sta v članku [8] ugotovila, da se ob opazovanju izvajanja motoričnih aktivnosti aktivira tudi del nevronskega vzorca, katerih aktivacija je sicer značilna za izvajanje motorične aktivnosti. Podobno sta Marc Jeannerod in Victor Frank v članku [9] pokazala podobnost v nevronske aktivnosti med izvajanjem motorične aktivnosti in zamišljanjem motorične aktivnosti.



Slika 2.5: Možganski režnji glede na opravljane funkcije [10].

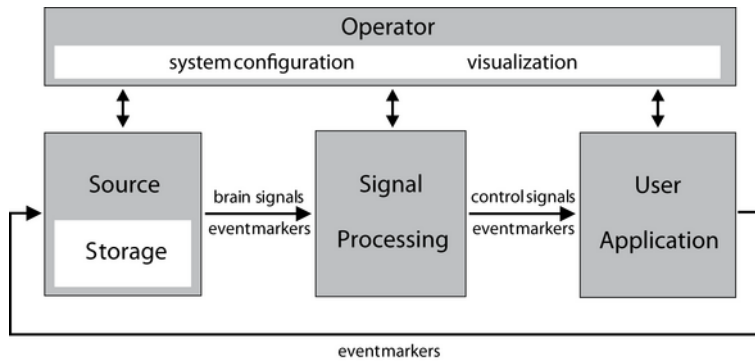
## 2.2 Podatkovna zbirka

### EEG Motor Movement/Imagery Dataset

Zbirka posnetkov (podatkovna zbirka) možganske aktivnosti EEG Motor Movement/Imagery Database [11, 3], ki smo jo uporabili v diplomski nalogi, je bila pridobljena s pomočjo sistema BCI2000, opisanega v raziskavi [1].

### 2.2.1 Sistem BCI2000

BCI sistem je odvisen od dejavnikov, kot so postopki merjenja možganske električne aktivnosti oziroma signalov, metode procesiranja in identificiranja značilke v izmerjenih aktivnosti, algoritmov, ki značilke prevedejo v smiselne ukaze, izhodnih sistemom, ki zvajajo ukaze ter prikaza povratne informacije uporabniku. Zaradi te kompleksnosti je za kvalitetno študijo potrebno dobro definirati in predvideti postopke evalvacije večih kombinacij signalov, alternativnih metod identificiranja značilke itd. Zelo veliko BCI sistemov je narejenih z namenom demonstracije učinkovitosti določene kombinacije BCI metod, vendar pa tak pristop zelo otežuje nadaljnje študije, ki bi gradile na takih izhodiščih. McFarland, Hinterberger, Birbaumer in Wolpaw so razvili in v članku [1] opisali koncept in implementacijo splošnega BCI sistema, ki omogoča enostavno zamenjavo in spreminjanje uporabljenih BCI metod in algoritmov. Sistem je sestavljen iz štirih modulov, njegova shema pa je prikazana na sliki 2.6. Naloga podatkovnega modula (angl. Source Module) je zajemanje oziroma digitaliziranje možganske električne aktivnosti in shranjevanje zajetih EEG posnetkov. Posnetke lahko zajema iz zunanjih naprav ali datotek, shranjuje pa jih v posebej strukturirane datoteke. Zajeti EEG posnetki se neobdelani posredujejo v modul za obdelavo EEG posnetkov, kjer se izvedejo metode in algoritmi za odkrivanje značilke in nato ugotovljene značilke prevedejo v tako imenovane ukaze, ki jih izvede naslednji modul. Kako se sistem odziva na ukaze in kako se ta odziv prikaže uporabniku je definirano v modulu za povratno informacijo (angl. User Application Module). Odzivi so lahko v obliki grafičnih vmesnikov, slušnih signalov, ali pa kot nadzorni signali za krmiljenje priključene fizične naprave. Vodja eksperimenta ima s pomočjo nadzornega modula (angl. Operator Module) nadzor nad celotnim sistemom. Modul mu omogoča spreminjanje nastavitve vseh ostalih modulov, omogoča mu zagon ali zaustavitev sistema in nenazadnje modul prikazuje tudi grafični vmesnik za prikaz trenutnega stanja sistema.



Slika 2.6: Shema sistema BCI2000 [1].

### 2.2.2 Protokol EEG snemanj

Podatkovna zbirka je sestavljena iz 1512 EEG snemanj, posnetih na 109 udeležencih. Vsak udeleženec je sodeloval pri 14 snemanjih: pri dveh enominutnih referenčnih snemanjih (baseline), in sicer eno z odprtimi in eno z zaprtimi očmi, ter pri treh dvominutnih snemanjih za vsako od štirih tipov nalog. V vsaki izmed nalog se je na zaslonu pred udeležencem prikazala tarča na enem izmed robov zaslona (levo, desno, zgoraj, spodaj), udeleženec pa si je moral glede na vrsto naloge zamišljati ali dejansko odpirati in zapirati ustrezno pest oziroma prste na nogi, dokler ni tarča izginila. Naloge so bile sledeče:

1. tarča se pojavi ali na levi ali na desni strani zaslona. Udeleženec odpira in zapira istoležno pest dokler tarča ne izgine, nato se sprosti
2. tarča se pojavi ali na levi ali na desni strani zaslona. Udeleženec si zamišlja odpiranje in zapiranje istoležnih pesti dokler tarča ne izgine, nato se sprosti
3. tarča se pojavi ali na zgornji ali spodnji strani zaslona. Udeleženec odpira in zapira obe pesti (če je tarča zgoraj) ali pa stiska prste na obeh nogah (če je tarča spodaj) dokler tarča ne izgine, nato se sprosti
4. tarča se pojavi ali na zgornji ali spodnji strani zaslona. Udeleženec



si zamišlja odpiranje in zapiranje obeh pesti (če je tarča zgoraj) ali stiskanja prstov na obeh nogah (če je tarča spodaj) dokler tarča ne izgine, nato se sprosti

Za vsakega udeleženca je bila ustvarjen mapa z udeleženčevo zaporedno številko v katerega so bili shranjeni posnetki opravljeni med snemanjem. Vsak posnetek je shranjen v svoji datoteki, iz imena datoteke pa je s pomočjo tabele 2.1 možno ugotoviti v okviru katere naloge je bil posnetek zajet. Datotekam s posnetki so bile dodane tudi spremljajoče istoimenske anotatorske datoteke s končnico ".event", kjer je za vsak segment v posnetku definiran njegov začetek kot zaporedna številka vzorca ter tip tega segmenta.

Naloga	Regularni izraz, ki sprejme ime datoteke posnetka
Referenčni posnetek zaprte oči	S*R00.edf\$
Referenčni posnetek odprte oči	S*R01.edf\$
Naloga 1	S*R(0[37] 11).edf\$
Naloga 2	S*R(0[48] 12).edf\$
Naloga 3	S*R(0[59] 13).edf\$
Naloga 4	S*R(0[6] 10 14).edf\$

Tabela 2.1: Imena datotek glede na nalogo v kateri je bil signal posnet.

Posnetki so bili posneti z 64 elektrodami in vzorčeni s frekvenco 160 Hz, tako da je v vsaki časovni enoti vzorec sestavljen iz 64 vrednosti. Povprečen posnetek je dolg 20000 vzorcev oziroma 125 sekund, s segmenti dolžine približno 800 vzorcev oziroma trajanjem 5 sekund. Vsak segment predstavlja periodično motorično aktivnost, definirano v spremljajočih anotacijskih datotekah, kjer je posamezen segment označen z eno izmed treh različnih označb podanih v tabeli 2.2. Za lažje razumevanje smo v nadaljevanju množico označb razširili in jih poimenovali glede na smer v katero naj bi se grafični kazalec simulacije premaknil ob dani aktivnosti (tabela 2.3).

Labela	Pomen v nalogi 1 in 2	Pomen v nalogi 3 in 4
T0	Mirovanje	Mirovanje
T1	Stiskanje leve pesti	Stiskanje obeh pesti hkrati
T2	Stiskanje desne pesti	Stiskanje prstov na obeh nogah

Tabela 2.2: Privzete označbe segmentov.

Labela	Pomen
H0	Mirovanje (Nalogi 1 in 2)
H1	Stiskanje leve pesti (Nalogi 1 in 2)
H2	Stiskanje desne pesti (Nalogi 1 in 2)
V0	Mirovanje (Nalogi 3 in 4)
V1	Stiskanje obeh pesti hkrati (Nalogi 3 in 4)
V2	Stiskanje prstov na obeh nogah (Nalogi 3 in 4)

Tabela 2.3: Prilagojene označbe segmentov.

### 2.2.3 Uporaba podatkov

Kot že naslov naloge pove, smo uporabili le posnetke zamišljanja motoričnih aktivnosti, torej posnetke, ki spadajo v kontekst Naloge 2 in Naloge 4.

V poglavju 2.1.3 smo omenili, da nas zanima le aktivnost motoričnega korteksa, še pred tem pa smo v poglavju 2.1.2 opisali podložnost neinvazivnih pristopov merjenja EEG signalov šumenju. Zaradi tega smo se odločili uporabiti le elektrodi C3 in C4, ki sta neposredno nad motoričnim korteksom in posledično najmanj izpostavljeni šumu. Podobno sta naredila tudi McFarland in Wolpaw v članku [12].

## 2.3 Filter s končnim impulznim odzivom

Filtriranje je transformacija vhodnega signala s funkcijo filtra v nek izhodni signal. V nalogi smo obravnavali le frekvenčni območje 8 do 13Hz ter 18 do 24 Hz, zato smo z uporabo nizko prepustnega filtra iz osnovnega signala poskusili izločiti nepotrebno frekvenčno območje, višje od 24 Hz.

Digitalne filtre lahko v splošnem opišemo z diferencialno enačbo 2.1, kjer

$x$  predstavlja vhodni signal,  $y[n]$  odziv filtra v  $n$ -tem koraku,  $a_k$  in  $b_m$  pa predstavljata poljubna števila. Filtre v taki obliki imenujemo tudi filtri z neskončnim enotnim odzivom (angl. Infinite Impulse Response filter - IIR filter). IIR filtri so rekurzivni, saj je odziv filtra v nekem koraku odvisen od poljubnega števila odzivov predhodnih korakov. Izhod IIR filtrov se tako ne bo nujno ustalil v končnem številu korakov, saj je izhod filtra odvisen od vsaj enega predhodnega izhoda in je možno, da bo vsakem nadaljnjem koraku vrednost izhoda višja od predhodnika. V podrobnosti IIR filtrov se tu ne bomo spuščali, bralec pa lahko o njih izve več v literaturi [13].

$$y[n] = \sum_{m=0}^K a_m y[n-m] + \sum_{m=0}^M b_m x[n-m] \quad (2.1)$$

Poseben primer so filtri, kjer so vsi koeficienti  $a_k$  enaki 0. Take filtre imenujemo filtri s končnim enotnim odzivom (angl. Finite Impulse Response filter - FIR filter). FIR filtri so po definiciji (enačba 2.2) neodvisni od odziva v predhodnih korakih oziroma z drugimi besedami FIR filtri so nerekurzivni, saj so odvisni zgolj od preteklih vrednosti vhodnega signala. To pomeni, da se pri filtriranju s FIR filtrom izhod zagotovo v končno mnogo korakov ustali pri vrednosti 0.

$$y[n] = \sum_{m=0}^M b_m x[n-m] \quad (2.2)$$

Postopek filtriranja lahko opišemo tudi kot konvolucijo jedra filtra z matriko, ki predstavlja vhodni signal. V primeru opisanega FIR filtra lahko jedro filtra  $h$  opišemo z matriko 2.3, izhodni signal pa z enačbo 2.4.

$$h = [b_M \dots b_0] \quad (2.3)$$

$$y = x * h \quad (2.4)$$

Omenili smo, da želimo filter uporabiti tako, da prepušča le določen fre-

kvenčni spekter, vendar pa bi bilo ustvarjanje ustreznega jedra filtra na do sedaj omenjen način zelo težko. Na tem mestu se lahko uporabi lastnost  $Z$  transform, posplošenega diskretnega Fourierjevega transform, imenovano konvolucijski teorem [14]. Teorem pravi, da lahko Fourierjev transform konvolucije dveh spremenljivk zapišemo tudi kot produkt Fourierjevega transform posameznih spremenljivk, kot je zapisano v enačbi 2.5, kjer  $\mathcal{F}$  predstavlja Fourierjev transform.

$$\mathcal{F}(x) \cdot \mathcal{F}(h) = \mathcal{F}(x * h) \quad (2.5)$$

Z drugimi besedami, konvolucijo v realnem prostoru lahko zapišemo kot množenje v frekvenčnem prostoru, kjer  $X(Z)$  predstavlja  $Z$  transform vhodnega signala,  $H(Z)$  pa  $Z$  transform jedra filtra:

$$x(n) * h(n) \iff H(Z)X(Z) \quad (2.6)$$

Sedaj lahko definiramo  $H(Z)$  tako, da bo končni filter prepuščal le frekvence nižje od  $\frac{1}{6}$  frekvence vzorčenja 160 Hz:

$$H(Z) = (1 - Z^{-6}) / (1 - Z^{-1}) \quad (2.7)$$

Sledi še izračun inverzne  $Z$  transformacije,  $Z^{-1}(H(Z))$ , in zapis filtra z rekurzivno diferenčno enačbo:

$$h(n) = x(n) - x(n - 6) + y(n - 1) \quad (2.8)$$

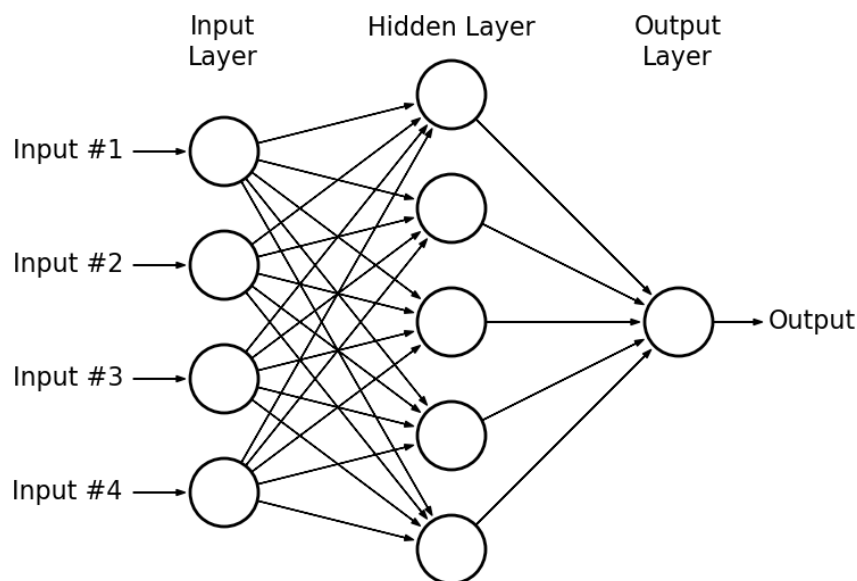
Dobljena enačba 2.8 predstavlja FIR filter implementiran kot IIR filter.  $Z$  nekaj izpeljevanja lahko zapišemo ekvivalentno jedro FIR filtra:

$$h(n) = x(n) + x(n - 1) + x(n - 2) + x(n - 3) + x(n - 4) + x(n - 5) \quad (2.9)$$

## 2.4 Nevronska mreža

### 2.4.1 O nevronski mreži

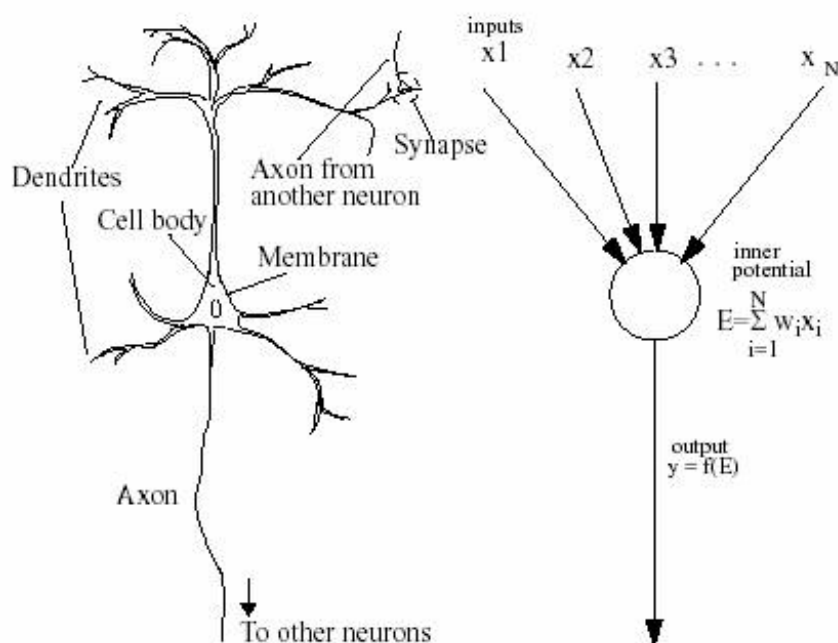
Nevronska mreža je model procesiranja informacij, zgrajen iz velikega števila procesnih elementov [15]. Splošna shema nevronske mreže je prikazana na sliki 2.7, kjer krogi predstavljajo procesne elemente oziroma umetne nevrone. Umetni nevron na posameznem nivoju ima vhodno povezavo do vseh umetnih nevronov predhodnega nivoja. Izjeme so umetni nevroni v vhodnem in izhodnem nivoju, ki imajo edini stik z okoljem izven nevronske mreže. Elementi vhodnega nivoja prejemaajo vsak po en vhod iz zunanjega sveta, ki ga nato procesirajo in posredujejo naprej po nevronske mreži, dokler signal ne doseže izhodnega nivoja. Izhodni nivo ima prav tako lahko več elementov, s tem pa se kompleksnost odzivov nevronske mreže poljubno večja.



Slika 2.7: Splošna shema nevronske mreže [16].

Umetni nevron deluje podobno kot posplošen biološki nevron (slika 2.8).

Množica sinaps med dendriti in aksoni nevrona se v umetnem nevronu kaže kot množica vhodov iz predhodnega nivoja nevronske mreže ali pa iz zunanega sveta. Podobno se tudi prepustnost sinaps pojavi kot spremenljiva utež na vhodni povezavi umetnega nevrona. Bolj kot je sinaps obremenjena, močnejša in bolj prepustna postane, in podobno pri umetnem nevronu, več koristne informacije neka povezava poda, večja bo njena utež.



Slika 2.8: Primerjava nevrona in umetnega nevrona nevronske mreže [17].

Pri biološkem nevronu se akcijski potencial sproži, ko skupna intenziteta vhodnih dražljajev preseže nek prag, pri umetnem nevronu pa je izhod definiran kot funkcija nad vsoto produktov vrednosti vhodnih povezav in njihovih uteži. To funkcijo imenujemo tudi pragovna funkcija in je lahko poljubna. Izhod umetnega nevrona lahko zapišemo z enačbo 2.10, kjer  $\phi$  predstavlja pragovno funkcijo,  $w_{kj}$  utež na povezavi med  $k$ -tim umetnim nevronom predhodnega nivoja in  $j$ -tim umetnim nevronom trenutnega nivoja,  $x_k$  pa izhod  $k$ -tega nevrona predhodnega nivoja.

$$o_j = \phi\left(\sum_{k=1}^n w_{kj}x_k\right) \quad (2.10)$$

Nevronsko mrežo, ki smo jo uporabili v tej nalogi, lahko opišemo kot učni model z nadzorovanim učenjem, kar pomeni, da je v fazi učenja nevronske mreže za vsak vhod natanko definiran tudi pričakovani odziv nevronske mreže. Po primerjavi dejanskega in pričakovanega odziva nevronske mreže in izračunu napake se uteži povezav med umetnimi nevroni prilagodijo tako, da se ta napaka čim bolj zmanjša. Obstaja več pristopov popravljanja uteži, eden izmed najbolj poznanih je povratno razširjanje, ki smo ga uporabili v tej nalogi in ga bomo na kratko pisali v nadaljevanju.

### 2.4.2 Nevronska mreža s povratnim razširjanjem

Pri učenju nevronske mreže s povratnim razširjanjem se izračun napake prične v nevronih izhodnega nivoja in nato nadaljuje v predhodnih nivojih. Pri vsakem umetnem nevronu se izvedeta dva koraka.

Z izračunom napake pričnemo v izhodnem nivoju. Funkcija izračuna napake je lahko poljubna, primer izračuna kvadrata napake pa je podan v enačbi 2.11, kjer  $t$  predstavlja pričakovan,  $o$  pa dejanski odziv nevronske mreže.

$$E = (t - o)^2 \quad (2.11)$$

Napaka izhodnega nivoja je posledica kumulativne napake predhodnih nivojev. Upoštevati moramo, da kumulativno napako transformira tudi pragovna funkcija trenutnega umetnega nevrona. Izračun kumulativne napake predhodnega nivoja iz že izračunane napake trenutnega umetnega nevrona lahko tako zapišemo z enačbo 2.12, kjer indeks  $j$  predstavlja trenutni umetni nevron, indeks  $k$  teče po vseh povezanih nevronih naslednjega nivoja, indeks  $i$  pa teče po vseh nevronih predhodnega nivoja.  $\phi'$  predstavlja prvi odvod

pragovne funkcije  $\phi$  trenutnega nevrona [15].

$$\delta_j = \begin{cases} (o_j - t_j)\phi'(\sum_i w_{ij}x_i) \dots \text{izhodni nivo} \\ (\sum_k \delta_k w_{jk})\phi'(\sum_i w_{ij}x_i) \dots \text{ostali nivoji} \end{cases} \quad (2.12)$$

Napaka se nato porazdeli po vseh povezanih nevronih predhodnega nivoja glede na uteženost povezav po enačbi 2.13, kjer  $x_i$  predstavlja izhod  $i$ -tega nevrona predhodnjega nivoja.

$$\frac{\partial E}{\partial w_{ij}} = \delta_j x_i \quad (2.13)$$

Končno spremembo uteži povezave lahko izračunamo z enačbo 2.14, kjer  $\alpha$  predstavlja nastavljeni parameter hitrosti učenja. Večji kot je  $\alpha$  parameter, večje bodo spremembe uteži in hitreje, vendar manj natančno, bo mreža skonvergirala k rešitvi [15].

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}} \quad (2.14)$$

## 2.5 Protokoli premikanja kazalca

Tekom izdelave diplomske naloge smo poizkusili priti do rešitve z različnimi pristopi, nekateri so bili uspešni, spet drugi ne. V nadaljevanju bomo opisali tri pristope oziroma protokole, ki temeljijo na uporabi nevronske mreže in se zaporedno nadgrajujejo. Prvi protokol predstavlja začetno fazo, kjer smo poskušali poiskati dokaj trivialno rešitev, ki je bila podobna linearni kombinaciji posameznih frekvenc amplitudnega spektra. Velika nihanja v amplitudah posameznih frekvenc znotraj istega tipa segmentov so nas prisilila, da smo pristop zamenjali in poskusili s povprečno vrednostjo, ki bi vsaj delno zadušila vplive velikih nihanj amplitud. Ta pristop se je izkazal kot precejšnja izboljšava, zato smo ga v okviru drugega protokola še nadgradili z dodatno nevronske mrežo in tako vzpostavili tretji protokol. Tretji protokol



predstavlja končno rešitev, kjer smo uporabili dve nevronske mreži, eno za prepoznavanje stiskov leve, stiskov desne pesti in mirovanja, ter drugo za prepoznavanje stiskov obeh pesti hkrati in mirovanja.

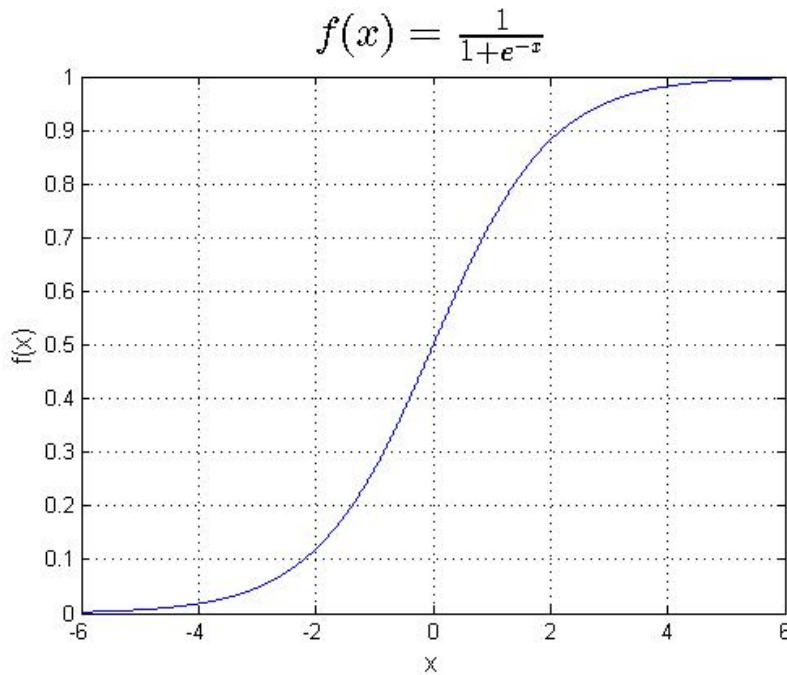
### 2.5.1 Prvi protokol

V prvem protokolu smo se osredotočili zgolj na premikanje kazalca v vodoravni smeri, torej levo, desno oziroma brez premika, uporabili pa smo vse segmente zamišljanja stiskanja leve in desne pesti. V vsakem segmentu smo signala iz kanalov C3 in C4 s hitrim Fourierjevim transformom (angl- Fast Fourier Transform - FFT) transformirali v amplitudni spekter in iz njega izluščili frekvenčni območji  $\mu$  in  $\beta$  ter tako dobili dva amplitudna vektorja. V nevronske mreži uporabljamo sigmoidno pragovno funkcijo 2.15, ki preslikava vhod kot je prikazano na sliki 2.9. Amplitudni vektor zajema vrednosti od 10 do 150, zato bi ga pragovna funkcija zelo slabo razpršila, mreža pa bi bila nezmožna razlikovati med posameznimi učnimi primeri. To težavo smo rešili s normalizacijo oziroma preslikavo amplitudnih vektorjev na interval  $[0, 1]$ . Za vsak amplitudni vektor smo poiskali njegovo najvišjo in najnižjo vrednost, nato pa iz celotne množice amplitudnih vektorjev poiskali najvišjo in najnižjo vrednost, tako smo dobili skrajni vrednosti množice amplitudnih vektorjev. Za vsak amplitudni vektor smo od celotnega vektorja odšteli prej dobljeno najnižjo vrednost, ter celoten vektor nato delili z velikostjo razlike med najvišjo in najnižjo vrednostjo. Tako dobljeni vektor je sedaj normaliziran na intervalu  $[0, 1]$  in ga lahko vključimo v učno množico.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.15)$$

### 2.5.2 Drugi protokol

Težava prvega protokola je veliko število vhodov, ki niso dovolj konsistentni, da bi nevronska mreža lahko vzpostavila ustrezen model, dodatno povzroča težave še prekrivanje amplitudnega spektra med različnimi tipi segmentov.



Slika 2.9: Graf pragovne funkcije.

Tip segmenta	Pogoj za sprejem
H0	$ avgC3 - avgC4  <  avgC4  * 0.2$
H1	$avgC3 > avgC4 * 1.1$
H2	$avgC3 < avgC4 * 0.97$

Tabela 2.4: Kriterij sprejetja segmenta v učno množico

Zato smo v tem protokolu kot vhodni vektor nevronske mreže uporabili vektor dveh aritmetičnih sredin (en vektor povprečne amplitude  $\mu$  in  $\beta$  spektra za kanal C3 in en za kanal C4). Ta pristop se je izkazal kot precejšnja izboljšava, vendar pa je še vedno prihajalo do težav s prekrivanjem amplitud, zato smo se odločili za dodaten, morda malo sporen, korak in iz učne množice odstranili primere po kriteriju v tabeli 2.4. Iz te reducirane množice segmentov smo nato kot učno množico ohranili tiste primere, ki padejo tako v srednjih 90% glede na povprečno vrednost C3 kot tudi v srednjih 90% glede na povprečno vrednost C4.

Tip segmenta	Pogoj za sprejem
V0	$ avgC3 - avgC4  <  avgC4  * 0.15$
V1	$ avgC3 - avgC4  <  avgC4  * 0.15$

Tabela 2.5: Kriterij sprejetja segmenta v učno množico.

### 2.5.3 Tretji protokol

V tretjem protokolu smo poleg premikanja kazalca vodoravno dodali še premikanje navpično. Ena izmed možnih implementacij je uporaba podatkov iz naloge 3, v kateri udeleženec stiska pesti obeh rok hkrati, če je tarča prikazana zgoraj in prste na obeh nogah, če je tarča prikazana spodaj. Ker bi bilo takšno aktivno upravljanje z nogami za udeleženca lahko naporno, smo se odločili za izvedbo samo z uporabo podatkov premikanja rok. Premike kazalca navzgor smo tako povezali s stiskanjem obeh pesti hkrati, aktivno premikanje kazalca navzdol pa smo nadomestili s samodejnim počasnim drsenjem kazalca navzdol po zaslonu. Podatke smo obdelali in oklestili podobno kot v drugem protokolu, le da smo sedaj po kriteriju v tabeli 2.5 dodali tudi segmente stiskov obeh pesti in segmente mirovanj. V tem protokolu so se pokazale težave pri razlikovanju med segmenti stiskov desne roke in segmenti stiskov obeh rok, zato smo uporabili dve nevronske mreže. Prvo nevronske mrežo uporabljamo tako kot v drugem protokolu za klasifikacijo stiskov leve pesti, desne pesti in mirovanja, drugo mrežo pa za klasifikacijo stiskov obeh pesti ter mirovanja. S kombinacijo teh dveh nevronske mreže smo pokrili tako premike kazalca v vodoravno kot tudi v navpični smeri.

### 2.5.4 Učenje mreže

**Prvi protokol** V prvem protokolu smo tako za učenje kot za merjenje uspešnosti uporabili celotno učno množico, dodatno pa smo uspešnost preverili še na posameznih manjših podmnožicah učne množice. Za dobro konvergenco nevronske mreže in konsistentne rezultate čez več poskusov se je izkazalo, da je potrebno vsaj 200 iteracij učenja nad celotno učno množico.

**Drugi in tretji protokol** V drugem in tretjem protokolu smo učno množico naključno razdelili na tri dele. Prvi del je zajemal 70% začetne množice in je predstavljal novo učno množico, drugi in tretji del sta zajemala vsak po 15% začetne množice in sta predstavljalata validacijsko in testno množico. Ob pričetku učenja se natančnost klasifikacij veča, s časoma pa lahko pride do prevelikega prilagajanja učni množici, kar bi pomenilo slabše prileganje validacijski in testni množici. V postopku učenja smo v rednih intervalih testirali odzive nevronske mreže nad validacijsko množico, če se šestkrat zaporedoma uspešnost klasifikacij nad validacijsko množico ni povečala predpostavimo, da je mreža dosegla najboljšo možno naučenost in postopek učenja prekinemo. Dodatni končni kriterij je bilo 1000 ponovitev učenja nad celotno učno množico, vendar je učni postopek že pri 150 do 200 ponovitvah učenja zadostil pogoju šestih zaporednih uspešnih klasifikacij nad validacijsko množico. Končno uspešnost učenja nevronske mreže preverimo s pomočjo ločene testne množice, tako preverimo, da se je nevronska mreža uspešno naučila splošnih klasifikacijskih pravil in ne le prepoznavanja učnih primerov.

## 2.6 Metrike zmogljivosti

Uspešnost učenja nevronske mreže smo merili s tremi metrikami zmogljivosti, in sicer pozitivno napovedljivost, občutljivost in natančnost. Vse tri metrike zmogljivosti lahko preberemo iz tako imenovane matrike zmogljivosti, predstavljeni v tabeli 2.6. Metrika pozitivne napovedljivosti lahko preberemo za vsako vrsto odziva mreže po vrsticah in predstavlja delež učnih primerov klasificiranih kot nek tip, ki so pravilno klasificirani 2.16. Naslednjo metriko, metriko občutljivosti, lahko preberemo za vsak stolpec in predstavlja delež učnih primerov nekega tipa, ki so bili pravilno klasificirani 2.17. Zadnja uporabljena metrika, natančnost, pa povzame rezultate celotne matrike razvrstitev in predstavlja delež učnih primerov, ki je bil klasificiran pravilno 2.18.

$$+P = \frac{TP}{TP + FP} \quad (2.16)$$

$$Se = \frac{TP}{TP + FN} \quad (2.17)$$

$$Acc = \frac{TP + TN}{P + N} \quad (2.18)$$

		Tip segmenta			
		H0/V0	H1	H2	V1
Odziv mreže	H0/V0	PK	NK	NK	NK
	H1	NK	PK	NK	NK
	H2	NK	NK	PK	NK
	V1	NK	NK	NK	PK

Tabela 2.6: Matrika razvrstitev (PK .. Pravilno Klasificirano; NK ... Nepravilno Klasificirano).

## 2.7 Implementacija

Protokole opisane v poglavju 2.5 smo večinoma implementirali s pomočjo standardnih orodji, ki so na voljo v programskem okolju Matlab. Dodatno je bilo potrebno namestiti le za Matlab prilagojeno različico orodja WFDB [18]. Ta vsebuje potrebne metode za branje podatkov iz datotek omenjenih v poglavju 2.2.2.

### 2.7.1 Branje podatkov

V poglavju 2.2.2 smo omenili, da je za vsakega udeleženca ustvarjen imenik v katerem se nato shranjujejo njegovi posnetki. Pri branju podatkov smo tako iterirali čez vse imenike udeležencev in včitavali datoteke, katerih imena so ustrezala regularnim izrazom Referenčnega posnetka z odprtimi očmi, Naloge 2 in Naloge 4 iz tabele 2.1.

Pred včitanjem posamezne datoteke je bilo potrebno ugotoviti dolžino posnetka v datoteki ter identificirati začetke in tipe segmentov v datoteki.

Dolžino oziroma število vzorcev v posnetku smo izračunali po enačbi 2.19, kjer  $t$  predstavlja trajanje posnetka v sekundah,  $F_s$  pa frekvenco vzorčenja. Oba parametra smo pridobili z ukazom `wfdbdesc` (psevdo koda 2.1).

$$N = t * F_s; \quad (2.19)$$

```

1 [sigInfo, Fs] = wfdbdesc(file_name);
2 t = sigInfo.LengthSamples;
3
4 N = t * Fs;
```

Psevdo koda 2.1: Izračun dolžine posnetka.

Nadalje smo z ukazom `rdann` iz spremljajoče anotatorske datoteke prebrali pričetke in tipe segmentov. Struktura anotatorske datoteke sicer predvideva polje namenjeno označevanju tipa segmenta, vendar pa je pri tej podatkovni zbirki tip segmenta opisan v polju namenjenemu komentarjem. Primer psevdokode včitavanja podatkov in rezanja posnetka na segmente je prikazan v psevdo kodi 2.2.

```

1 channels = [9, 13]; % C3 ... idx9; C4 ... idx13
2 [ann, ~, ~, ~, ~, comments] = rdann(file_name, 'event');
3 [~, signal, ~] = rdsamp(file_name, channels, N, 1);
4
5 for i = 1:length(ann)
6     if i < length(ann)
7         tmp = signal(ann(k):ann(k+1), :);
8     else
9         tmp = signal(ann(k):N, :);
10
11     len = length(tmp);
12     segment = tmp(1:min(resolution, end), :);
13     if length(segment) < resolution
14         segment = [segment;
15                     zeros(
```

```

16         resolution=length(segment),
17         length(channels)
18     )];
19
20     type = comments{i};

```

Psevdo koda 2.2: Rezanje posnetka na segmente.

Zaradi poenostavitve shranjevanja v lokalne strukture smo vse segmente skrčili oziroma razširili z nizom ničel na dolžino 1024 vzorcev. Poleg samega segmenta smo shranili tudi enako dolg izsek referenčnega posnetka, označbo segmenta, definirano po tabeli 2.3 ter tarčno vrednost, potrebno za učenje nevronske mreže (tabela 2.7).

Tip segmenta	Tarčna vrednost
H0	[1; 0; 0; 0]
H1	[0; 1; 0; 0]
H2	[0; 0; 1; 0]
V0	[1; 0; 0; 0]
V1	[0; 0; 0; 1]

Tabela 2.7: Tarčne vrednosti tipov segmentov pri učenju nevronske mreže.

## 2.7.2 Filtriranje in izračun značilik

V poglavju 2.3 smo podali primer nizkoprepustnega filtra, ki prepušča  $\mu$  in  $\beta$  frekvenčni spekter in duši frekvence višje od 26.67 Hz. V končni implementaciji smo se omejili le na uporabo  $\mu$  spekter, ki zajema 8 do 13 Hz, zato bomo nizkoprepustni filter prilagodili temu in frekvenco dušenja pomaknili na 16 Hz. Jedro takega FIR filtra lahko tokrat brez izpeljevanja zapišemo z enačbo 2.20.

$$y(n) = x(n) + x(n-1) + \dots + x(n-8) + x(n-9) \quad (2.20)$$

Primer uporabe takega filtra je podan v psevdokode 2.3. Uporaba pa-

parametra 'same' pomeni, da bo rezultat konvolucije enako dolg kot je vhodni segment.

```
1 filter = ones(10,1)/10;  
2 segment_filtered = conv(segment, filter, 'same');
```

Psevdo koda 2.3: Konvolucija segmenta z jedrom filtra.

V poglavju o protokolih (poglavje 2.5) smo opisali značilke kot par povprečnih vrednosti amplitudnega spektra izračunanega za signal elektrode C3 in signal elektrode C4. Nad vsakim filtriranim segmentom smo tako z uporabo vgrajene Matlabove funkcije izračunali absolutno vrednost hitrega Fourierjevega transformata z ločljivostjo 1024 vzorcev. Na koncu še izračunamo značilke z uporabo Matlabove funkcije *mean* ter jih dodamo v seznam vseh značilk. Celoten postopek je zelo enostavno prikazan v psevdokodi 2.4.

```
1 amplitudni_spekter = abs(fft(segment_filtered, 1024));  
2 znacilke(:,end+1) = mean(amplitudni_spekter);
```

Psevdo koda 2.4: Izračun amplitudnega spektra in dodajanje značilk v seznam.

### 2.7.3 Ustvarjanje učne množice in učenje nevronske mreže

Z drugim protokolom smo uvedli redukcijo seznama značilk glede na pogoje v tabeli 2.8. Pri reduciranju smo odstranili istoležne elemente tudi iz seznama referenčnih posnetkov, seznama označb segmentov ter seznama tarčnih vrednosti. Nadalje iz reducirane množice poskusimo izločiti še skrajno izstopajoče elemente tako, da izločimo zgornjih in spodnjih pet odstotkov elementov seznama značilk. Tako oklesteno množico značilk sprejmemo kot učno množico pri učenju nevronskih mrež.

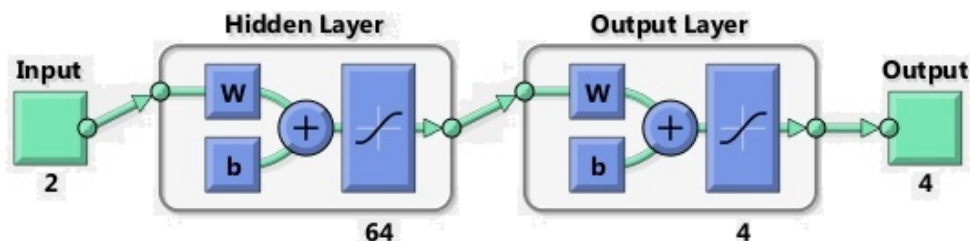
Za upravljanje grafičnega kazalca smo uporabili dve nevronske mreži. Prva nevronska mreža je namenjena klasifikaciji segmentov zamišljanja stiskov leve, stiskov desne roke in mirovanja, druga nevronska mreža pa klasifikaciji segmentov zamišljanja stiskov obeh rok in mirovanja. To razdelitev



Tip segmenta	Tarčna vrednost
H0	$ avgC3 - avgC4  <  avgC4  * 0.2$
H1	$avgC3 > avgC4 * 1.1$
H2	$avgC3 < avgC4 * 0.97$
V0	$ avgC3 - avgC4  <  avgC4  * 0.15$
V1	$ avgC3 - avgC4  <  avgC4  * 0.15$

Tabela 2.8: Kriterij sprejetja segmenta v učno množico.

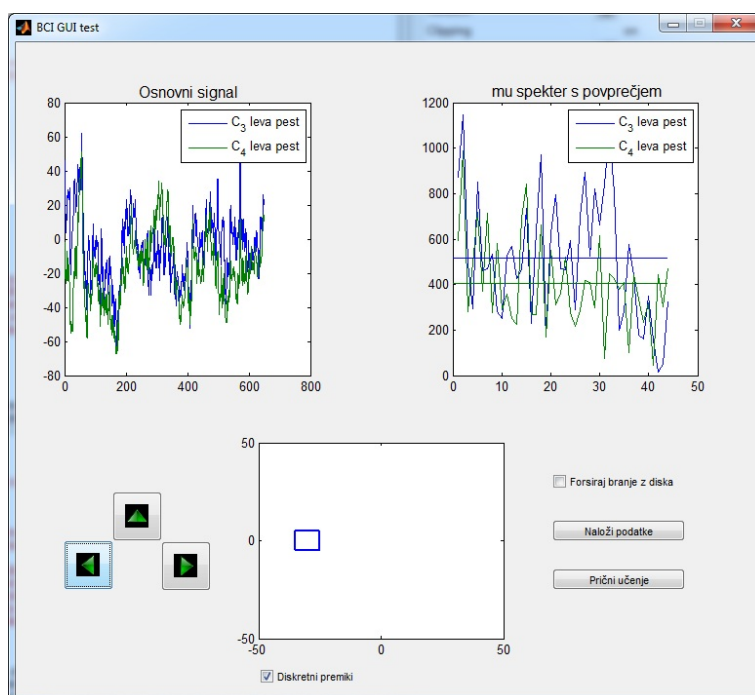
klasifikacij moramo seveda upoštevati tudi pri posredovanju učnih primerov v fazi učenja nevronske mreže. Obe mreži imata enako konfiguracijo z dvema vhodi, enim skritim nivojem s 64 vozlišči in štirimi izhodi (slika 2.10). Za učenje nevronske mreže smo uporabili algoritem povratnega razširjanja, ki je del okolja Matlab.



Slika 2.10: Shema uporabljene nevronske mreže.

### 2.7.4 Grafični vmesnik

Grafični vmesnik je prikazan na sliki 2.11. Razdeljen je na tri dele: gumbe za nalaganje podatkov in učenje nevronske mreže, grafe za prikaz obdelovanega signala, ter treh smerne gumbe in risalno površino za simulacijo premikanja grafičnega kazalca.



Slika 2.11: Slika grafičnega vmesnika.

**Nalaganje podatkov in učenje** Pred uporabo sistema je potrebno naložiti podatke in zagnati učni proces nevronske mreže.

Nalaganje podatkov uporabnik doseže s pritiskom na gumb Naloži podatke. Nalaganje podatkov iz izvornih datotek je zelo počasen proces, zato sistem po vsakem branju izvornika podatke obdela in shrani v strukture, kot je opisano v poglavju 2.7.2. Te strukture nato shrani v datoteko, ki jo je kasneje mogoče zelo hitro včitati. Z aktiviranjem stikala Obvezno branje izvornika lahko uporabnik sistem prisili, da ob včitavanju podatkov ne uporabi

datoteke z že strukturiranimi podatki, temveč, da vse podatke ponovno včita iz izvornih datotek.

Naslednji korak je proces učenja nevronske mreže, ki se prične s pritiskom na gumb Prični učenje. Sistem pred predajo podatkov nevronske mreže preveri, ali so podatki včitani in ustrezno strukturirani, ter v primeru neskladnosti uporabniku sporoči napako. Potek učenja nevronske mreže lahko uporabnik spremlja preko posebnega vmesnika, ki je del Matlabovega skupka orodij za delo z nevronske mreže.

**Smerni gumbi in premikanje kazalca po podlagi** Smerni gumbi omogočajo simulacijo premikov grafičnega kazalca v treh smereh, v četrto smer, navzdol, pa kazalec drsi sam. Samodejno drsenje navzdol je realizirano z uporabo periodičnega časovnika s periodo ene sekunde. Ko uporabnik pritisne na enega izmed smernih gumbov se aktivira drugi enkratni časovnik, ki za dve sekundi prepreči premikanje kazalca navzdol. Poleg tega se ob pritisku na smerni gumb iz seznama značilk izbere naključna značilka, katere označba ustreza smeri, ki jo predstavlja gumb. Odvisno od pritisnjene gumba se značilka klasificira z nevronske mreže za premike navzgor ali pa mreže za vodoravne premike. S stikalom Diskretni premiki lahko uporabnik izbira med dvema načinoma izračuna odmika kazalca glede na odziv nevronske mreže. Sam izračun premikov je prikazan v psevdokodi 2.5.

```
1 if znacilka_labela == 'H1' || znacilka_labela == 'H2'
2     odziv = mreza_vodoravno(znacilka);
3 else
4     odziv = mreza_navpicno(znacilka);
5
6 if diskretniPremiki
7     smer = vec2ind(odziv);
8
9     switch smer
10         case 2
11             dX = dX - 1;
```

```
12         case 3
13             dX = dX + 1;
14         case 4
15             dY = dY + 1;
16 else
17     dX = odziv[3] - odziv[2];
18     dY = odziv[4];
```

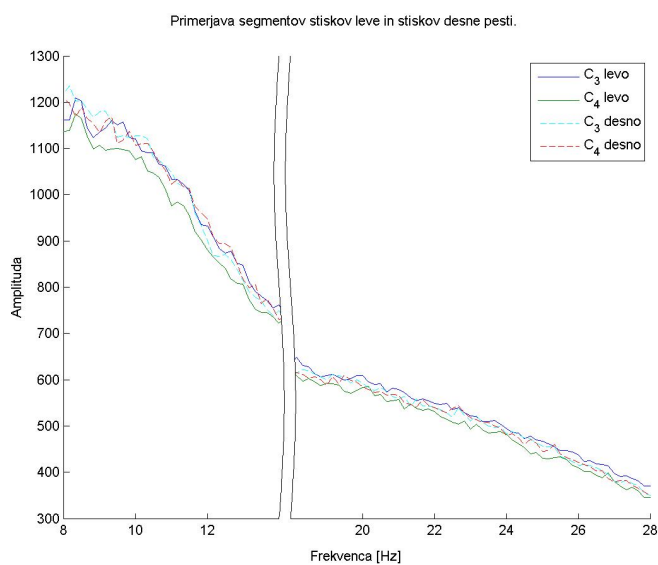
Psevdo koda 2.5: Premikanje kazalca glede na odziv nevronske mreže.

**Grafi** Zaradi lažje predstave o obnašanju izvornih podatkov se ob pritisku na smerni gumb izrišeta dve stopnji v obdelavi signala, na podlagi katerega je bila izračunana izbrana značilka. Levi graf prikazuje segment izvornega signala elektrod C3 in C4 po filtriranju z nizko prepustnim FIR filtrom. Izračunan amplitudni spekter tega segmenta in povprečje tega spektra pa sta izrisana na desnem grafu.

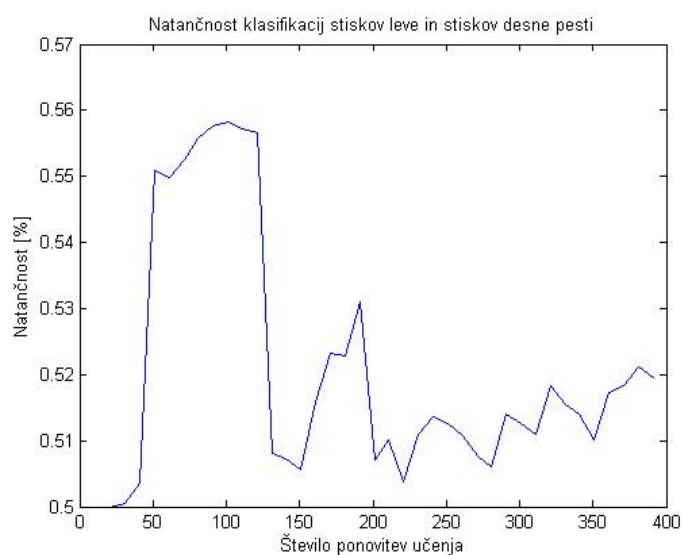
## 3. Rezultati

### 3.1 Prvi protokol

V okviru prvega protokola (Poglavje 2.5.1) smo hoteli ugotoviti, ali obstaja enostavna relacija med posameznimi harmoniki  $\mu$  in  $\beta$  spektra ter motoričnimi akcijami. Iz primerjave segmentov stiskov leve pesti in stiskov desne pesti na sliki 3.1 lahko vidimo, da je zaradi velikega prekrivanja amplitudnih spektrov že na pogled težko razlikovati med tema dvema tipoma. Nadalje lahko iz slike 3.2 vidimo, da ima tudi nevronska mreža zelo velike težave s prepoznavanjem posameznih tipov segmentov z le nekaj več kot 50% natančnostjo klasifikacij tudi po večkratnem učenju nad isto učno množico.



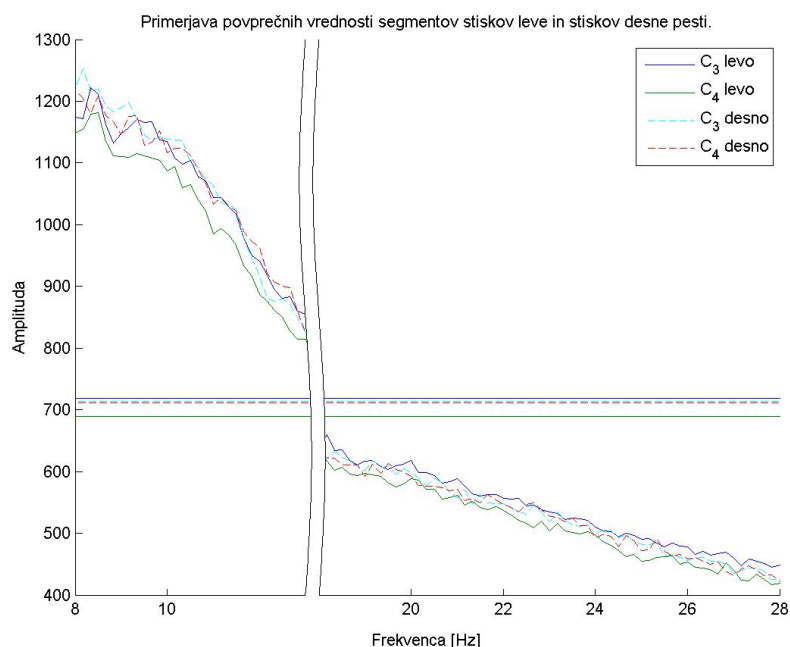
Slika 3.1: Primerjava stiskov leve pesti (neprekinjena črta) in stiskov desne pesti (prekinjena črta) v  $\mu$  in  $\beta$  amplitudnih spektrih.



Slika 3.2: Natančnost klasifikacij v odvisnosti od števila ponovitev učenja.

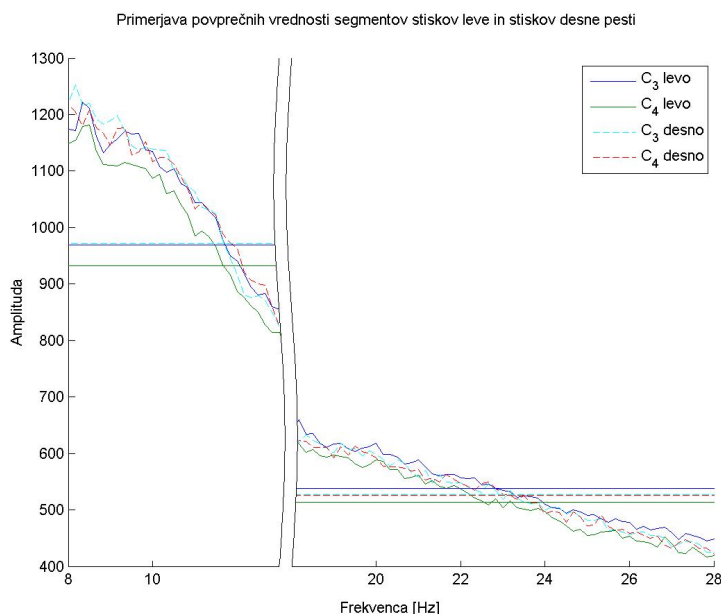
## 3.2 Drugi protokol

Prvi protokol se ni izkazal kot najbolj uspešen, zato smo ga poskusili izboljšati. Kot je opisano v poglavju 2.5.2, smo poizkusili z odstranjevanjem učnih primerov, za katere smo menili, da so neustrezni. Za vsak potencialni učni primer smo izračunali povprečje  $\mu$  in  $\beta$  amplitudnih spektrov, sprejeli pa smo le srednjih 90% segmentov, ki so ustrezali pogojem podanim v tabeli 2.4.



Slika 3.3: Povprečna vrednost  $\mu$  in  $\beta$  amplitudnega spektra.

Poleg reduciranja učne množice smo poizkusili še z učenjem nevronske mreže kar na povprečnih vrednostih amplitudnih spektrov. Prvo smo poizkusili s povprečno vrednostjo celotnega  $\mu$  in  $\beta$  amplitudnega spektra, primer katerega je podan na sliki 3.3, nato pa še s posameznima  $\mu$  in  $\beta$  spektroma, prikazanima na sliki 3.4.



Slika 3.4: Povprečna vrednost posameznih amplitudnih spektrov.

Učenje nevronske mreže s povprečjem celotnega  $\mu$  in  $\beta$  amplitudnega spektra nad reducirano učno množico se je izkazalo kot bistveno izboljšanje v primerjavi s prejšnjimi metodami, vendar pa je iz slike 3.5 razvidno, da z 67% občutljivostjo, segmenti mirovanja delno posegajo v amplitudne spektre drugih tipov in s tem predstavljajo precejšnjo težavo pri učenju nevronske mreže. Zelo slabo se je izkazal  $\beta$  amplitudni spekter, kjer je metrika občutljivosti za segmente stiskov desne pesti dosegla le okoli 11%, iz slike 3.6 pa lahko tudi vidimo, da je nevronska mreža skoraj 89% segmentov stiska desne pesti prepoznala kot segmente mirovanja. Ta rezultat pomeni, da v  $\beta$  spektru prihaja do zelo velikega deleža prekrivanja med vsaj dvema tipoma segmentov in posledično metoda s povprečjem amplitudnega spektra za to območje ne bo delovala sprejemljivo dobro. Na drugi strani pa imamo  $\mu$  amplitudni spekter, ki ima v skoraj vsakem pogledu lepšo matriko razvrstitev (slika 3.7) kot prej omenjena spektra, še vedno pa lahko vidimo, da prihaja do težav s segmenti mirovanja.

Iz predstavljenih matrik razvrstitev je razvidno, da veliko večino težav



**Matrika razvrstitev  $\mu$  in  $\beta$  spektra**

Odziv mreže	mirovanje	619 18.6%	158 4.7%	184 5.5%	64.4% 35.6%
	leva pest	124 3.7%	920 27.6%	1 0.0%	88.0% 12.0%
	desna pest	172 5.2%	0 0.0%	1150 34.6%	87.0% 13.0%
		67.7% 32.3%	85.3% 14.7%	86.1% 13.9%	80.8% 19.2%
	mirovanje	leva pest	desna pest	<b>Tip segmenta</b>	

Slika 3.5: Matrika razvrstitev po učenju nevronske mreže s povprečno vrednostjo  $\mu$  in  $\beta$  amplitudnega spektra.

povzročajo segmenti mirovanja, medtem ko je razločevanje segmentov stiskov leve in segmentov stiskov desne pesti skoraj trivialno, kar je morda še bolj očitno, če nevronske mreži podamo le te segmente. Matrika razvrstitev, prikazana na sliki 3.8, je tako skoraj idealna, vse metrike zmogljivosti pa dosežejo vrednosti večje od 99%.

### 3.3 Tretji protokol

S tretjim protokolom smo poskusili razširiti delovanje drugega protokola tako, da bi le-ta bil sposoben usmerjati grafični kazalec tudi navzgor z uporabo segmentov stiskov obeh pesti hkrati. Pri tem so se pokazale precejšnje težave pri razločevanju med segmenti stiskov obeh pesti, segmenti mirovanja ter se-

**Matrika razvrstitev beta spektra**

<b>Odziv mreže</b>	mirovanje	2481 48.3%	163 3.2%	1141 22.2%	65.5% 34.5%
	leva pest	128 2.5%	970 18.9%	0 0.0%	88.3% 11.7%
	desna pest	108 2.1%	4 0.1%	146 2.8%	58.6% 43.4%
		91.3% 8.7%	85.3% 14.7%	11.3% 88.7%	70.0% 30.0%
	mirovanje	leva pest	desna pest	<b>Tip segmenta</b>	

Slika 3.6: Matrika razvrstitev po učenju nevronske mreže s povprečno vrednostjo  $\beta$  amplitudnega spektra.

**Matrika razvrstitev  $\mu$  spektra**

<b>Odziv mreže</b>	mirovanje	723 21.9%	186 5.6%	187 5.7%	66.0% 34.0%
	leva pest	105 3.2%	878 26.6%	4 0.1%	89.0% 11.0%
	desna pest	111 3.4%	0 0.0%	1112 33.6%	90.9% 9.1%
		77.0% 23.0%	82.5% 17.5%	85.3% 14.7%	82.1% 17.9%
	mirovanje	leva pest	desna pest	<b>Tip segmenta</b>	

Slika 3.7: Matrika razvrstitev po učenju nevronske mreže s povprečno vrednostjo  $\mu$  amplitudnega spektra.

**Matrika razvrstitev  $\mu$  spektra**

<b>Odziv mreže</b>	leva pest	<b>1056</b> 44.6%	<b>4</b> 0.2%	<b>99.6%</b> 0.4%
	desna pest	<b>8</b> 0.3%	<b>1299</b> 54.9%	<b>99.4%</b> 0.6%
		<b>99.2%</b> 0.8%	<b>99.7%</b> 0.3%	<b>99.5%</b> 0.5%
	leva pest	desna pest		<b>Tip segmenta</b>

Slika 3.8: Matrika razvrstitev po učenju nevronske mreže le s segmenti stiskov leve pesti in segmenti stiskov desne pesti.

gmenti stiskanja desne pesti. Na sliki 3.9 je prikazana matrika razvrstitev pri klasifikaciji vseh uporabljenih tipov segmentov z eno nevronske mreže. Vidimo lahko, da je le 19.9% segmentov stiskanja obeh pesti hkrati klasificiranih pravilno, 44% je klasificiranih kot stiski desne pesti, 34% pa je klasificiranih kot mirovanje. Ob podrobnejši primerjavi je iz slike 3.10 razvidno, da se nevronska mreža ob klasifikaciji segmentov stiska obeh rok in segmenti mirovanja odreže sprejemljivo dobro, z 46.8% točnostjo klasifikacij segmentov mirovanja. Nasprotno je iz matrike razvrstitev na sliki 3.11 očitno, da z 0% točnostjo klasifikacij segmentov desne pesti, nevronska mreža ni zmožna razlikovati med segmenti stiskov obeh pesti in segmenti stiskov desne pesti.

Kot kompromis smo v končno rešitev vključili dve nevronske mreže. Prvo nevronske mreže smo uporabili le za klasifikacijo segmentov stiskov leve pesti, stiskov desne pesti in mirovanja, drugo pa le za klasifikacijo segmentov stiskov obeh pesti hkrati in mirovanja.

**Matrika razvrstitev  $\mu$  spektra**

<b>Odziv mreže</b>	mirovanje	<b>1763</b> 30.8%	<b>237</b> 4.1%	<b>327</b> 5.7%	<b>508</b> 8.9%	<b>62.2%</b> 37.8%
	leva pest	<b>75</b> 1.3%	<b>836</b> 14.6%	<b>5</b> 0.1%	<b>23</b> 0.4%	<b>89.0%</b> 11.0%
	desna pest	<b>8</b> 0.1%	<b>0</b> 0.0%	<b>713</b> 12.5%	<b>648</b> 11.3%	<b>52.1%</b> 47.9%
	obe pesti	<b>16</b> 0.3%	<b>2</b> 0.0%	<b>273</b> 4.8%	<b>292</b> 5.1%	<b>50.1%</b> 49.9%
		<b>94.7%</b> 5.3%	<b>77.8%</b> 22.2%	<b>54.1%</b> 45.9%	<b>19.9%</b> 80.1%	<b>62.9%</b> 37.1%
	mirovanje	leva pest	desna pest	obe pesti	<b>Tip segmenta</b>	

Slika 3.9: Matrika razvrstitev po učenju nevronske mreže tudi s segmenti stiskov obeh pesti.

**Matrika razvrstitev  $\mu$  spektra**

<b>Odziv mreže</b>	mirovanje	<b>470</b> 18.2%	<b>248</b> 9.6%	65.5% 34.5%
	obe pesti	<b>534</b> 20.7%	<b>1331</b> 51.5%	71.4% 28.6%
		46.8% 53.2%	84.3% 15.7%	<b>69.7%</b> <b>30.3%</b>
	mirovanje	obe pesti		<b>Tip segmenta</b>

Slika 3.10: Matrika razvrstitev po učenju nevronske mreže s segmenti stiskov obeh pesti ter mirovanja.

**Matrika razvrstitev  $\mu$  spektra**

<b>Odziv mreže</b>	desna pest	<b>0</b> 0.0%	<b>0</b> 0.0%	NaN% NaN%
	obe pesti	<b>1004</b> 41.9%	<b>1391</b> 58.1%	58.1% 41.9%
		0.0% 100%	100% 0.0%	<b>58.1%</b> <b>41.9%</b>
	mirovanje	obe pesti		
		<b>Tip segmenta</b>		

Slika 3.11: Matrika razvrstitev po učenju nevronske mreže s segmenti stiskov obeh pesti ter stiski desne pesti.



## 4. Diskusija in nadaljnje delo

### 4.1 Splošno o rezultatih

Rezultati prvega protokola so nas nekoliko presenetili, saj smo, morda malce naivno, pričakovali, da bo nevronska mreža sposobna poiskati vsaj tako dobro kombinacijo harmonikov, kot so to storili v članku [12] z uporabo autoregresivnega modela. Predpostavljamo, da je tako slab odziv nevronske mreže predvsem posledica dokaj velikega nihanja harmonikov znotraj množice segmentov istega tipa. Ta nihanja so bila dovolj velika, da so posegla tudi v amplitudni spekter drugih tipov segmentov, s čimer pa smo nevronske mreže dovajali slabe učne primere ali pa celo protiprimere.

Drugi protokol z 82% natančnostjo smatramo za uspešnega. Kljub temu menimo, da pristop z reduciranjem učne množice morda ni najbolj primeren oziroma bi ga morali izboljšati, da bi preprečili potencialno prirejanje informacije zajete v izvorni podatkovni zbirki. Indikacija škodljivosti reduciranja učne množice je morda prav slika 3.8, kjer nevronska mreža doseže več kot 99% natančnost, občutljivost in točnost. Nekoliko nas je presenetilo, da je nevronska mreža, navkljub omenjenemu potencialnemu prirejanju učne množice, zelo veliko segmentov napačno kalsificirala kot segmente mirovanja.

Vpeljava segmentov stiskanja obeh pesti hkrati v učno množico se je izkazala za izredno problematično. Nevronska mreža je imela težave že z ločevanjem segmentov stiskanja obeh pesti in segmentov mirovanja, pri ločevanju segmentov stiskanja obeh pesti in segmentov stiskanja desne pesti pa je skoraj popolnoma odpovedala. Pričakovali smo, da bo nevronske mreže več težav

povzročalo ločevanje med segmenti stiskanja obeh pesti in segmenti mirovanja, saj so si ti že zaradi razmerja med povprečnima amplitudama spektrov elektrod C3 in C4 blizu.

## 4.2 Nadaljnje delo

Nadaljnje raziskave bi radi opravljali z realnočasovnimi BCI sistemi in dejanskimi subjekti, ne le s posnetki. Za to bi bilo potrebno nekoliko prilagoditi pristop, ki smo ga ubrali pri tej nalogi. Glavna sprememba bi bil pristop k opazovanju in procesiranju signala. V nalogi smo signal razrezali na segmente dolge pet sekund in le-te obravnavali kot celoto, oziroma drugače povedano, za obdelavo signala smo uporabili nedrseče okno širine petih sekund. V primeru realnočasovnega BCI sistema tak pristop ne bi bil učinkovit, saj bi se sistem odzval le vsakih pet sekund. Rešitev vidimo v vzorčenju signala s prekrivajočimi okni širine nekaj desetink sekunde. V obstoječem sistemu, bi lahko ekvivalent te rešitve opisali kot zvezno drseče okno širine nekaj desetink sekunde.

Postopek učenja in klasifikacije bi spremenili tako, da bi namesto značilke enega samega okna upoštevali značilke niza oken. S tem bi pridobili vpogled v časovno spremenljivost signala, ki nudi veliko več informacij kot samostojen izsek signala. Učni model nevronske mreže bi tako morali zamenjati z modelom, ki je sposoben izkoristiti to dodatno informacijo. Primera takega modela sta skriti model Markova [19] in nevronska mreža z odločitvenimi drevesi [20].

Področje procesiranja signalov in strojnega učenja je zelo široko, če upoštevamo še nepopolno raziskanost možganske aktivnosti lahko vidimo, da je opisana različica nadaljnjega dela le kaplja v morje.

# Literatura

- [1] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, “Bci2000: a general-purpose brain-computer interface (bci) system,” *Biomedical Engineering, IEEE Transactions on*, zv. 51, št. 6, str. 1034–1043, 2004.
- [2] Schalk Lab, “Bci2000,” september 2014, Dosegljivo: [www.bci2000.org](http://www.bci2000.org)
- [3] G. AL, A. LAN, G. L, H. JM, I. PCh, M. RG, M. JE, M. GB, P. C-K, and S. HE, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,” *Circulation*, zv. 101, str. e215–e220, junij 2000.
- [4] Beyond the Dish, “How Neural Stem Cells Become Neurons and Glia,” september 2014, Dosegljivo: <http://beyondthedish.wordpress.com/2013/11/22/how-neural-stem-cells-become-neurons-and-glia/>
- [5] Robert Stufflebeam, “Introduction to Neurons, Synapses, Action Potentials, and Neurotransmission,” september 2014, Dosegljivo: [http://www.mind.ilstu.edu/curriculum/neurons\\_intro/neurons\\_intro.php](http://www.mind.ilstu.edu/curriculum/neurons_intro/neurons_intro.php)
- [6] BrainMaster Technologies, Inc., “10-10 system chart,” september 2014, Dosegljivo: <http://www.brainmaster.com/kb/entry/438/>
- [7] D. Cordes, V. M. Haughton, K. Arfanakis, G. J. Wendt, P. A. Turski, C. H. Moritz, M. A. Quigley, and M. E. Meyerand, “Mapping functio-

- nally related regions of brain with functional connectivity mr imaging,” *American Journal of Neuroradiology*, zv. 21, št. 9, str. 1636–1644, 2000.
- [8] J. Dushanova and J. Donoghue, “Neurons in primary motor cortex engaged during action observation,” *European Journal of Neuroscience*, zv. 31, št. 2, str. 386–398, 2010.
- [9] M. Jeannerod and V. Frak, “Mental imaging of motor activity in humans,” *Current opinion in neurobiology*, zv. 9, št. 6, str. 735–739, 1999.
- [10] ASU - Ask A Biologist, “Brain Regions and Functions,” september 2014, Dosegljivo: <https://askbiologist.asu.edu/what-your-brain-doing>
- [11] Schalk Lab, “EEG Motor Movement/Imagery Dataset,” september 2014, Dosegljivo: <http://www.physionet.org/physiobank/database/eegmmidb/>
- [12] J. R. Wolpaw and D. J. McFarland, “Control of a two-dimensional movement signal by a noninvasive brain–computer interface in humans,” zv. 101.
- [13] E. Grivel and M. Najim, “Infinite impulse response filters,” *Digital Filters Design for Signal and Image Processing*, str. 173–189.
- [14] J. Proakis and D. Manolakis, *Digital Signal Processing*. Prentice Hall international editions, Pearson Prentice Hall, 2007.
- [15] R. Hecht-Nielsen, *Neurocomputing*. Addison-Wesley Publishing Company.
- [16] astroML, “Neural Network Diagram,” september 2014, Dosegljivo: [http://www.astroml.org/book\\_figures/appendix/fig\\_neural\\_network.html](http://www.astroml.org/book_figures/appendix/fig_neural_network.html)
- [17] Cecilia Hemming, “Using Neural Networks in Linguistic Resources,” september 2014, Dosegljivo: <http://www.hemming.se/gslt/LingRes/NeuralNetworks.htm>

- 
- [18] Physionet, “The WFDB Software Package,” april 2014, Dosegljivo: <http://www.physionet.org/physiotools/wfdb.shtml>
- [19] Z. Ghahramani, “An introduction to hidden markov models and bayesian networks,” *International Journal of Pattern Recognition and Artificial Intelligence*, zv. 15, št. 01, str. 9–42, 2001.
- [20] I. Ivanova and M. Kubat, “Decision-tree based neural network,” in *Machine Learning: ECML-95*, str. 295–298, Springer, 1995.