

Electrofiziologie – digitalizarea și prelucrarea semnalelor EMG asociate

Lucrare de disertație

**prezentată ca cerință parțială pentru obținerea titlului de
Master în domeniul *Inginerie electronică, telecomunicații și tehnologii
informaționale*
programul de studii de masterat *Ingineria Informației și a Sistemelor de
Calcul***

Coordonatori științifici

Prof. Dr. Ing. Corneliu Burileanu

Ing. Ana-Antonia Neacșu

Absolvent

Ing. Ioana Bădițoiu

6/16/2020

etti.pub.ro/absolvire/vizualizeaza2.php

Universitatea "Politehnica" din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației
Programul de masterat **Ingineria informației și a sistemelor de calcul**

Anexa 2

TEMA LUCRĂRII DE DISERTAȚIE
a masterandului BĂDIȚOIU I.C. Ioana , 421-IIIS.

1. Titlul temei: Electrofiziologie – digitalizarea și prelucrarea semnalelor EMG asociate

2. Descrierea contribuției originale a masterandului (în afara părții de documentare) și specificații de proiectare:

Scopul acestui proiect este de a dezvolta o platformă inteligentă (hardware și software) capabilă să capteze în timp real activitatea electrică ca răspuns la stimularea unui nerv a mușchiului și să ofere instrumentele software necesare pentru aplicații bazate pe tehnici de Inteligență Artificială în scopul dezvoltării aplicațiilor în diverse domenii: asistență medicală, securitate, divertisment, etc.

Pentru început, se va crea un set de date EMG de suprafață preluate la nivelul antebrațului, ce va conține o serie de gesturi executate de diferiți subiecți. Ulterior se va realiza o analiză extinsă a metodelor și algoritmilor incluse în instrumentele software de Inteligență Artificială.

Pentru validarea soluțiilor, se va dezvolta un cadru de recunoaștere automată a gesturilor (AGR) folosind platforma. AGR va fi un sistem complex care permite etichetarea în timp real a gesturilor din setul de date. Mai precis, proiectul va include achiziția semnalului care urmează să fie etichetat, pre-procesarea semnalului și, în final, clasificarea (rezultatul sistemului este gestul pe care utilizatorul l-a efectuat). Platforma va fi structurată în două componente - modulul de achiziție EMG și instrumente software de Inteligență Artificială.

Modulul de achiziție EMG va fi un dispozitiv de achiziție neintruziv care poate colecta și monitoriza activitatea EMG la nivelul antebrațului. Arhitectura acestui dispozitiv va fi compusă dintr-o rețea de mai mulți senzori EMG și o unitate centrală (Microcontroller/Microcomputer) responsabilă de sincronizarea semnalelor captate și comunicarea cu instrumentele software de Inteligență Artificială.

3. Resurse folosite la dezvoltarea proiectului:

OpenBCI Cyton Board, Senzor de Activitate Musculară MyoWare, Olimex EKG – EMG Shield, Python, Arduino IDE

4. Proiectul se bazează pe cunoștințe dobândite în principal la următoarele 3-4 discipline:

MEDE, TAPSM, NA

5. Proprietatea intelectuală asupra proiectului aparține: U.P.B.

6. Data înregistrării temei: 2019-11-28 10:16:21

Conducător(i) lucrare,
Prof. dr. ing. Corneliu BURILEANU

Student,

Ana-Antonia NEACȘU

Responsabil program master,
Prof. dr. ing. Radu DOGARU

Decan,
Prof. dr. ing. Mihnea UDREA

Cod Validare: **ae4b2cccfb**

Declarație de onestitate academică

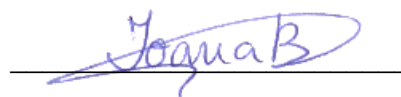
Prin prezenta declar că lucrarea cu titlul “ *Electrofiziologie – digitalizarea și prelucrarea semnalelor EMG asociate*”, prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității “Politehnica” din București ca cerință parțială pentru obținerea titlului de Master în domeniul *Inginerie electronică, telecomunicații și tehnologii informaționale*, programul de studii *Ingineria Informației și a Sistemelor de Calcul* este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurărilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, 16.06.2020

Absolvent Ioana BĂDIȚOIU



(semnătura în original)

Copyright © 2020 , *Ioana BĂDIȚOIU*

Toate drepturile rezervate

Autorul acordă laboratorului “Speed” din cadrul UPB dreptul de a reproduce și de a distribui public copii pe hârtie sau electronice ale acestei lucrări, în formă integrală sau parțială.

CUPRINS

Cuprins	9
Listă figuri.....	11
Listă tabele	13
Listă acronime.....	15
Introducere	17
Motivația lucrării.....	17
Obiectivele lucrării	18
Structura lucrării.....	19
CAPITOLUL 1 EMG – Aspecte principale	21
1.1 Introducere.....	21
1.2 EMG – Fundalul anatomic și fiziologic	22
1.3 Zgomotul electric și factorii care afectează semnalul EMG	25
1.4 Detecția și descompunerea semnalelor EMG.....	27
CAPITOLUL 2 Componente Hardware și Metode de Implementare	29
2.1 Introducere.....	29
2.2 Schema Propusă	30
2.3 Setul de date EMG	30
2.4 Componente Hardware.....	31
2.4.1 OpenBCI	31

2.4.2	ArduinoUNO.....	32
2.4.3	Senzorii MyoWare	33
2.4.4	Olimex EKG/EMG	34
CAPITOLUL 3 Experimentul.....		37
3.1	Achiziția de date.....	37
3.2	Potențiale soluții în analiză.....	41
3.2.1	Olimex.....	41
3.2.2	MyoWare	42
3.2.3	OpenBCI	44
3.3	Procesarea semnalului EMG	48
3.4	Alcătuirea bazei de semnale	51
CAPITOLUL 4 Antrenarea modelului și rezultate.....		57
4.1	Platforma Kaggle.....	57
4.2	TensorFlow.....	58
4.2.1	Keras	59
4.2.2	Funcții de activare	63
4.2.3	Matricea de confuzie	65
4.2.4	Acuratețea	65
Concluzii		67
Concluzii generale.....		67
Contribuții personale		68
Dezvoltări ulterioare.....		68
Bibliografie		71
Anexa 1		73
Anexa 2		77
Anexa 3		79

LISTĂ FIGURI

Figura 1.1 – Descompunerea semnalului EMG de suprafață [1].....	24
Figura 1.2 – Procesarea semnalului [3].....	27
Figura 2.1 – Schema bloc.....	30
Figura 2.2 – Setul de gesturi: (1) poziție neutră, (2) extensia dorsală a încheieturii, (3) flexia palmară a încheieturii.....	30
Figura 2.3 – Setul de gesturi: (1) flexia degetelor, (2) extensia degetelor.....	31
Figura 2.4 – Setul de gesturi: Supinație.....	31
Figura 2.5 – Setul de gesturi: Pronație.....	31
Figura 2.6 – Placa de dezvoltare OpenBCI [8].....	32
Figura 2.7 – Placa de dezvoltare ArduinoUNO [12].....	33
Figura 2.8 – Senzor MyoWare [9].....	33
Figura 2.9 – Plasarea senzorului MyoWare [9].....	34
Figura 2.10 – Scutul Olimex [17].....	35
Figura 3.1 – Anatomia mușchilor mâinii și ai antebrațului [10].....	38
Figura 3.2 – Plasarea electrozilor pe antebraț.....	40
Figura 3.3 – Principiile colectării semnalului EMG.....	41
Figura 3.4 – Olimex: evaluarea unui singur mușchi.....	42
Figura 3.5 – Montaj pregătire mișcare clește prindere cu servomotor.....	43
Figura 3.6 – Conexiuni Senzor MyoWare – placa de dezvoltare ArduinoUNO.....	43
Figura 3.7 – MyoWare: evaluarea unui singur mușchi.....	44
Figura 3.8 – Schema utilizând nouă electrozi (limitare fizică conexiuni existente).....	45
Figura 3.9 – Electrozii utilizați.....	46
Figura 3.10 – Conectarea electrozilor.....	47

Figura 3.11 – Înregistrarea semnalelor EMG în softul OpenBCI	48
Figura 3.12 – Filtru Butterworth [6]	49
Figura 3.13 – Ilustrarea algoritmului SVM [6]	51
Figura 3.14 – Grafic gesturi subiect 1	53
Figura 3.15 – Grafic gesturi subiect 2	54
Figura 4.1 – Arhitectura TensorFlow [13]	59
Figura 4.2 – Sumarul modelului	61
Figura 4.3 – Modelul utilizat.....	62
Figura 4.4 – Funcția de activare ReLU [20]	63
Figura 4.5 – Exemplu rulare	64
Figura 4.6 – Reprezentare de bază a matricei de confuzie [15]	65
Figura 4.7 – Matricea de confuzie rezultată în urma rulării.....	65
Figura 4.8 – Matricea de confuzie și acuratețea generată în urma primelor teste efectuate	66

LISTĂ TABELE

Tabelul 3.1 – Mușchii și funcțiile acestora	39
Tabelul 3.2 – Mușchii ce pot fi măsurați și gesturile rezultate	40

LISTĂ ACRONIME

A

ANN = Artificial Neural Network (Rețea Neuronală Artificială)

API = Application Programming Interface

AR = auto-regresiv

B

BCI = Interfață creier-calculator (Brain-Computer Interface)

C

CPU = Central Processing Unit

D

DRNN = Dynamic Recurrent Neural Network (Rețele Neuronale Recurente Dinamice)

E

EMG = Electromiografie

G

GPU = Graphics Processing Unit

I

IA = Inteligență artificială

K

KNN = k-nearest neighbors

M

MUAP = Potențialul de acțiune al unității motorii (Motor Unit Action potential)

MAV = Valoarea medie absolute (Mean Absolute Value)

R

RMS = Rădăcina medie patrată (Root Mean Square)

S

SVM = Support Vector Machine

T

TPU = Tensor Processing Unit

W

WVD = Distribuția Wigner-Ville

INTRODUCERE

MOTIVAȚIA LUCRĂRII

Recunoașterea gesturilor are numeroase aplicații în diverse domenii. Dezvoltarea protezelor pentru mâini robotizate are ca scop redarea persoanelor cu dizabilități, capacitatea de a recupera funcționalitatea necesară manipulării obiectelor mediului lor zilnic și întreprinderii activităților cotidiene.

Însă nu numai persoanele cu dizabilități ar putea beneficia de sisteme Brain-Computer Interface (BCI). Un posibil scenariu ar fi controlul unui braț robotic pe o distanță lungă și interacțiunea acestuia cu mediul. De exemplu, interacțiunea în zonele contaminate radioactiv ar putea fi posibilă, ceea ce ar putea contribui la minimizarea riscurilor în cazul producerii catastrofelor.

Semnalele electrice transmise de creier prin sistemul nervos sunt asociate cu tipul de mișcare pe care membrele trebuie să-l execute. Senzorii mioelectrici sunt dispozitive neintruzive care permit captarea de semnale electrice din sistemul nervos periferic. Relația dintre semnalele originare din creierul care tinde să genereze o acțiune și cele mioelectrice ca urmare a acestora, este slab corelată. Din acest motiv, este necesar să se studieze interacțiunea lor pentru a dezvolta algoritmi care permit recunoașterea gesturilor și transformarea lor în comenzi care activează mișcările corespunzătoare ale protezei.

OBIECTIVELE LUCRĂRII

În prezent, domeniul cercetării Brain-Computer Interface (BCI) se confruntă cu următoarele probleme:

- ✓ Nu există un proces bine documentat pentru configurarea și implementarea ușoară a sistemelor BCI;
- ✓ Sistemele BCI care funcționează bine sunt în prezent costisitoare sau exclusive pentru un grup mic de specialiști;
- ✓ Dispozitivele mai accesibile, cu resurse reduse se confruntă cu probleme de precizie;
- ✓ Sunt necesare cunoștințe multidisciplinare precum biologie, procesarea semnalului și tehnologia informației pentru înțelegerea funcționării unui sistem BCI. Astfel de informații de fond sunt rareori disponibile într-o singură resursă sau arie.

Electromiografia (EMG) este o tehnologie promițătoare pentru a controla protezele sau brațele robotizate. Electromiografia de suprafață (sEMG) este o metodă non-invazivă și, prin urmare, poate fi aplicată ușor, fără nicio intervenție medicală. În cazul unui persoane cu un membru amputat, EMG permite utilizarea semnalului de activare a aceluiași mușchi care a fost cândva folosit pentru a mișca membrul pierdut. Acest lucru permite, un control mai natural și mai intuitiv al unui dispozitiv robotic protetic.

Una dintre barierele tehnologice care împiedică utilizarea pe scară largă a semnalelor EMG pentru a controla dispozitivele protetice este aceea că semnalele fiabile și curate necesită un hardware scump.

Însă, recent, au fost introduse dispozitive precum OpenBCI. Acesta este o interfață BCI ieftină, care poate fi folosită și pentru măsurarea semnalelor EMG. În special, introducerea ADS1299 de către Texas Instruments în 2012, un circuit integrat specific pentru măsurători biopotențiale care integrează amplificatoare operaționale și un convertor analog-digital de precizie, a permis proiectarea dispozitivelor BCI cu costuri reduse și portabile.

Asociate cu noua generație de proteze printate 3D, metodele mai ieftine pentru măsurarea semnalelor EMG pot avea un impact profund în calitatea vieții celor care nu pot accesa protezele scumpe. Utilizarea unor astfel de dispozitive cu costuri reduse vine, însă, cu prețul semnalelor de calitate scăzută, adesea cu un raport slab semnal-zgomot și variații mari.

Cu toate acestea, domeniul de cercetare al învățării automate oferă o serie de metode care pot rezolva problema clasificării.

Acest proiect abordează problemele de mai sus, aducând următoarele contribuții:

1. Dezvoltarea unei platforme capabile să capteze în timp real activitatea electrică ca răspuns la stimularea unui nerv al muchiului;
2. Crearea unui set de date EMG de suprafață preluate la nivelul antebrațului, ce va conține o serie de gesturi executate de diferiți subiecți. Ulterior se va realiza o analiză extinsă a metodelor și algoritmilor incluse în instrumentele software de Inteligență Artificială;
3. Pentru validarea soluțiilor, se va dezvolta un cadru de recunoaștere automată a gesturilor (AGR) folosind platforma;

4. Platforma va fi structurată în două componente – modulul de achiziție EMG și instrumente software de inteligență artificială.

STRUCTURA LUCRĂRII

În *Capitolul 1* se va face o introducere în domeniul semnalelor EMG, prezentând detalii legate de fundalul anatomic și fiziologic, zgomotul și factorii care afectează acest tip de semnal, modul de detecție și descompunere al semnalului. *Capitolul 2* prezintă concepte teoretice ce țin de arhitectură, etape de dezvoltare, resurse necesare și expune unele noțiuni de lingvistică. *Capitolul 3* prezintă detalii legate de modul în care a fost realizat studiul, precum achiziția de date și procesarea semnalului EMG de suprafață, iar *Capitolul 4*, prezintă modalitatea de antrenare a modelului pe setul de date construit anterior, precum și rezultatele studiului. Capitolul final sumarizează concluziile tezei, subliniază contribuțiile autorului și prezintă câteva posibilități de îmbunătățire sau dezvoltări ulterioare.

CAPITOLUL 1

EMG – ASPECTE PRINCIPALE

1.1 INTRODUCERE

Semnal biomedical este un semnal electric achiziționat de la orice organ care reprezintă o variabilă fizică de interes. Acest semnal este în mod normal o funcție de timp și poate fi descris în termeni de amplitudine, frecvență și fază.

Semnalul electromiografic (EMG) este un semnal biomedical care măsoară curenții electrici generați în mușchi în timpul contracției acestora reprezentând activități neuromusculare.

Sistemul nervos controlează activitatea musculară (contracție sau relaxare). Prin urmare, semnalul EMG este un semnal complex, controlat de sistemul nervos și care depinde de proprietățile anatomice și fiziologice ale mușchilor.

Semnalul EMG capătă zgomot în timpul călătoriei prin diferite țesuturi. Mai mult decât atât, dacă detectorul EMG se află la suprafața pielii, acesta poate colecta semnale de la diferite unități motorii la un moment dat și poate genera interacțiunea diferitelor semnale. Detectarea semnalelor EMG cu metode puternice și avansate devine o cerință foarte importantă în ingineria biomedicală. Motivul principal al interesului pentru analiza semnalului EMG este diagnosticul clinic și aplicațiile biomedicale. Domeniul de gestionare și reabilitare a dizabilității motorii este identificat ca unul dintre domeniile importante de aplicare. Formele și ritmurile de ardere ale potențialelor de acțiune ale unității motorii (MUAP) în semnalele EMG oferă o sursă importantă de informații pentru diagnosticul

tulburărilor neuromusculare. Odată ce algoritmi și metode adecvate pentru analiza semnalului EMG sunt disponibile cu ușurință, natura și caracteristicile semnalului pot fi înțelese în mod corespunzător și se pot realiza implementări hardware pentru diferite aplicații legate de semnalul EMG. Până în prezent, s-au făcut cercetări și eforturi ample în această zonă, dezvoltarea unor algoritmi mai buni, modernizarea metodologiilor existente, îmbunătățirea tehnicilor de detectare pentru reducerea zgomotului și obținerea de semnale EMG precise [1].

Abia la mijlocul anilor 1980, tehnicile de integrare în electrozi au fost suficient de avansate pentru a permite producerea în lot a amplificatoarelor necesare. În prezent, sunt disponibile în comerț o serie de amplificatoare adecvate. La începutul anilor 1980, au devenit disponibile cabluri care produc artefacte în intervalul dorit de microvolți. În ultimii 15 ani, cercetările au dus la o mai bună înțelegere a proprietăților înregistrării semnalelor EMG de suprafață. Recent, electromiografia de suprafață este din ce în ce mai folosită pentru a înregistra mușchii superficiali în protocoalele clinice, unde electrozii intramusculari fiind folosiți doar pentru mușchiul profund.

Puține implementări hardware au fost realizate pentru controlul protetic al mâinilor și interacțiunea om-mașină. Este destul de important să se efectueze o investigație pentru a clasifica problemele reale ale analizei semnalelor EMG și pentru a justifica măsurile acceptate. Tehnologia înregistrării EMG este relativ nouă. Există încă limitări în detectarea și caracterizarea neliniarităților existente în electromiografia de suprafață (sEMG, o tehnică specială pentru studierea semnalelor musculare), estimarea fazei, dobândirea de informații exacte datorate derivării de la normalitate. Algoritmi tradiționali de reconstrucție a sistemului au diverse limitări și o complexitate computațională considerabilă și multe arată o varianță ridicată.

Progresele recente în tehnologiile de prelucrare a semnalelor și modelele matematice au făcut posibilă dezvoltarea tehnicilor avansate de detectare și analiză a semnalelor EMG. Diverse tehnici matematice și de inteligență artificială (AI) au capatat o atracție extinsă. Modelele matematice includ transformarea undelor, abordările în timp-frecvență, transformarea Fourier, distribuția Wigner-Ville (WVD), măsuri statistice și statistici de ordin superior. Abordările AI pentru recunoașterea semnalului includ rețele neuronale artificiale (ANN), rețele neuronale recurente dinamice (DRNN) și sisteme fuzzy logice. Algoritmul genetic a fost, de asemenea, aplicat în cipuri hardware evolutive pentru cartografierea intrărilor EMG la acțiunile dorite ale mâinii [1].

Transformarea Wavelet este, de asemenea, potrivită pentru semnalele non-staționare precum EMG. Abordarea în timp-frecvență folosind WVD în hardware ar putea permite un instrument în timp real care poate fi utilizat pentru antrenamentul specific al unității motorii în situații de răspuns biologic. Metodele statistice de ordin superior pot fi utilizate pentru analiza semnalului EMG datorită proprietăților unice ale acestora aplicate seriilor de timp aleatorii. Bispectrul sau spectrul de ordinul trei are avantajul de a suprima zgomotul gaussian.

1.2 EMG – FUNDALUL ANATOMIC ȘI FIZIOLOGIC

Electromiografia este uneori denumită și activitate mioelectrică. Țesutul muscular conduce potențialul electric similar modului în care o fac nervii, iar numele dat acestor semnale electrice este potențialul de acțiune musculară. EMG de suprafață este o metodă de înregistrare a informațiilor prezente în aceste potențiale de acțiune musculară. Când se detectează și înregistrează semnalul

EMG, există două probleme principale care influențează fidelitatea acestuia. Prima este raportul semnal-zgomot. Adică raportul dintre energia din semnalele EMG și energia din semnalul de zgomot. În general, zgomotul este definit ca un semnal electric care nu face parte din semnalul EMG dorit. Cealaltă problemă este denaturarea semnalului. Contribuția relativă a oricărei componente de frecvență din semnalul EMG nu trebuie modificată.

Se pot utiliza două tipuri de electrozi pentru a obține semnalul muscular:

- ✓ electrodul invaziv
- ✓ electrodul non-invaziv

Când EMG este achiziționat de la electrozii montați direct pe piele, semnalul este compus din toate potențialele de acțiune ale fibrelor musculare care apar în mușchii care stau la baza pielii. Aceste potențiale de acțiune apar la intervale aleatorii. Deci, în orice moment, semnalul EMG poate avea atât tensiune pozitivă, cât și negativă [1].

Potențialele individuale de acțiune ale fibrelor musculare sunt uneori dobândite folosind sârmă sau electrozi cu ac așezați direct în mușchi. Combinația potențialelor de acțiune din toate fibrele musculare ale unei singure unități motorii este potențialul de acțiune al unității motorii (MUAP) care poate fi detectat de un electrod de suprafață al pielii (neinvaziv) situat în apropierea acestui câmp sau de un ac electrod (invaziv) introdus în mușchi. Ecuația de mai jos arată un model simplu al semnalului EMG [5]:

$$x(n) = \sum_{r=0}^{N-1} h(r)e(n-r) + w(n) \quad (1)$$

Unde $x(n)$ este semnalul EMG modelat, $e(n)$ este punctul prelucrat, reprezentând impulsul de ardere, $h(r)$ reprezintă MUAP, $w(n)$ este zgomotul Gaussian alb de medie zero și N este numărul unităților motorii de ardere.

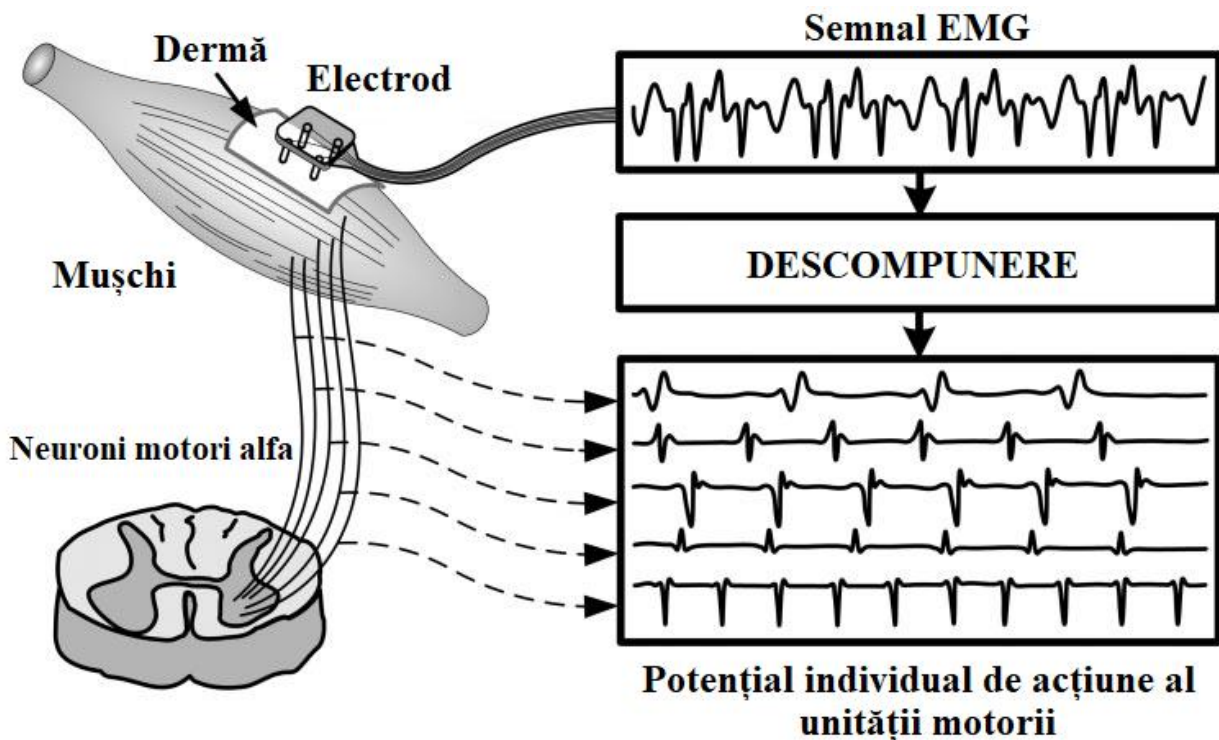


Figura 1.1 – Descompunerea semnalului EMG de suprafață [1]

Semnalul este preluat de la electrod și amplificat. De obicei, un amplificator diferențial este utilizat ca amplificator pentru prima etapă. Pot urma etapele suplimentare de amplificare. Înainte de a fi afișat sau stocat, semnalul poate fi procesat pentru a fi eliminat zgomotul de joasă frecvență sau de înaltă frecvență sau alte artefacte posibile. Frecvent, utilizatorul este interesat de amplitudinea semnalului. În consecință, semnalul este rectificat și mediat într-un anumit format pentru a indica amplitudinea EMG.

Sistemul nervos este atât sistemul de control, cât și cel de comunicație al organismului. Acest sistem este format dintr-un număr mare de celule conectate excitabile numite neuroni care comunică cu diferite părți ale corpului cu ajutorul unor semnale electrice, care sunt rapide și specifice. Sistemul nervos este format din trei părți principale: creierul, măduva spinării și nervii periferici. Neuronii sunt unitatea structurală de bază a sistemului nervos și variază considerabil ca mărime și formă. Aceștia reprezintă celule extrem de specializate care conduc mesaje sub formă de impulsuri nervoase de la o parte a corpului la alta.

Un mușchi este alcătuit din mănunchiuri de celule specializate capabile de contracție și relaxare. Funcția principală a acestor celule specializate este aceea de a genera forțe, mișcări și capacitatea de a comunica, cum ar fi vorbirea, scrierea sau alte moduri de exprimare. Țesutul muscular are extensibilitate și elasticitate. Are capacitatea de a primi și de a răspunde la stimuli și poate fi relaxat sau contractat. Țesutul muscular are patru funcții cheie:

- ✓ producerea mișcării
- ✓ mișcarea substanței în interiorul corpului
- ✓ asigurarea stabilizării

- ✓ generarea căldurii.

Trei tipuri de țesut muscular pot fi identificate pe baza structurii, proprietăților contractile și a mecanismelor de control [1]:

- (i) Mușchi scheletici
- (ii) Mușchi netezi
- (iii) Mușchi cardiac

EMG se aplică la studiul mușchiului scheletic. Țesutul muscular al scheletului este atașat de os și contracția acestuia este responsabilă de susținerea și deplasarea scheletului. Contracția mușchiului scheletului este inițiată de impulsuri în neuroni către mușchi și este, de obicei, sub control voluntar. Fibrele musculare scheletice prezintă neuroni pentru a putea realiza contracția. Acest tip particular de neuron se numește „neuron motor” și se apropie de țesutul muscular, dar nu este în realitate conectat la acesta. Un neuron motor furnizează de obicei stimularea multor fibre musculare.

În general, corpul uman este neutru din punct de vedere electric; are același număr de sarcini pozitive și negative. Dar în stare de repaus, membrana celulară nervoasă este polarizată datorită diferențelor în concentrații și compoziție ionică în întreaga membrană plasmatică. Există o diferență de potențial între fluidele intracelulare și extracelulare. Ca răspuns la un stimul din neuron, o fibră musculară se depolarizează pe măsură ce semnalul se propagă de-a lungul suprafeței sale și fibrele se răsfrâng. Această depolarizare, însoțită de o mișcare de ioni, generează un câmp electric în apropierea fiecărei fibre musculare. Un semnal EMG este trenul potențialului de acțiune al unității motorii (MUAP) care arată răspunsul muscular la stimularea neurală. Semnalul EMG apare în mod aleatoriu și este, în general, modelat ca un proces de impuls filtrat în care MUAP este filtrul și procesul de impulsuri reprezintă impulsurile neuronale [2].

1.3 ZGOMOTUL ELECTRIC ȘI FACTORII CARE AFECTEAZĂ SEMNALUL EMG

Intervalul de amplitudine al semnalului EMG este de 0-10 mV (între +5 și -5) înainte de amplificare. Semnalele EMG dobândesc zgomot în timpul călătoriei prin diferite țesuturi. Este important să fie înțelese caracteristicile zgomotului electric. Zgomotul electric, care va afecta semnalele EMG, poate fi clasificat în următoarele tipuri [1]:

1. *Zgomot inerent în echipamentele electronice*

Toate echipamentele electronice generează zgomot. Acest zgomot nu poate fi eliminat; utilizarea componentelor electronice de înaltă calitate nu poate decât să o reducă.

2. *Zgomot ambiental*

Radiația electromagnetică este sursa acestui tip de zgomot. Suprafețele corpurilor noastre sunt inundate constant cu radiații electro-magnetice și este practic imposibil de evitat expunerea la acestea pe suprafața Pământului. Zgomotul ambiental poate avea o amplitudine care este cu una până la trei ordine de mărime mai mare decât semnalul EMG.

3. *Artefact de mișcare*

Când este introdus în sistem artefactul de mișcare, informațiile sunt alterate. Artefactul de mișcare provoacă nereguli în date. Există două surse principale pentru acesta: interfața electrodului și cablul electrodului. Artefactul de mișcare poate fi redus prin proiectarea corespunzătoare a circuitelor și setării electronice.

4. *Instabilitatea inerentă a semnalului.*

Amplitudinea EMG este de natură aleatoare. Semnalul EMG este afectat de viteza de ardere a unităților motoare, care, în cele mai multe condiții, se aprind în regiunea de frecvență de la 0 la 20 Hz. Acest tip de zgomot este considerat nedorit, iar eliminarea lui este importantă.

Factorii care afectează în principal semnalul EMG pot fi, de asemenea, clasificați. Acest tip de clasificare este setată astfel încât algoritmi de analiză a semnalului EMG să poată fi optimizați și echipamentele să poată fi proiectate în mod consecvent. Factorii care afectează semnalul EMG se încadrează în trei categorii de bază:

1. *Factorii cauzali*

Aceasta este afectarea directă a semnalelor. Factorii cauzali pot fi împărțiți în două clase [2]:

- (i) **Extrinseci:** Aceasta se datorează structurii și plasării electrodului. Factorii precum aria suprafeței de detecție, forma electrodului, distanța dintre suprafața de detectare a electrodului, locația electrodului în raport cu punctele motorii din mușchi, locația electrodului muscular pe suprafața musculară în raport cu marginea laterală a mușchiului, orientarea suprafețelor de detectare în raport cu fibrele musculare au în principal efect asupra semnalului EMG.
- (ii) **Intrinseci:** Factorii fiziologici, anatomici, biochimici au loc datorită numărului de unități motorii active, compoziției tipului de fibre, fluxului de sânge, diametrului fibrei, adâncimii și locației fibrelor active și cantității de țesut între suprafața mușchiului și electrodului.

2. *Factorii intermediari*

Factorii intermediari sunt fenomene fizice și fiziologice influențate de unul sau mai mulți factori cauzali. Motivele care stau la baza acestui fapt pot fi aspectele de filtrare cu filtru bandă a electrodului doar cu volumul său de detecție, suprapunerea potențialelor de acțiune în semnalul EMG detectat sau viteza de conducere a potențialului de acțiune care se propagă de-a lungul membranei fibrei musculare. Chiar și apariția musculaturii din apropiere poate cauza factori intermediari.

3. *Factorii determinanți*

Aceștia sunt influențați de factorii intermediari. Numărul de unități motorii active, viteza de ardere și interacțiunea mecanică între fibrele musculare au o influență directă asupra informațiilor din semnalul EMG și a forței înregistrate. Amplitudinea, durata și forma potențialului de acțiune al unității motorii pot fi, de asemenea, responsabile.

Maximizarea calității semnalului EMG se poate face prin următoarele moduri [2]:

1. Raportul semnal-zgomot trebuie să conțină cea mai mare cantitate de informații provenite de la semnalul EMG posibil și cantitatea minimă de contaminare cu zgomot;
2. Distorsiunea semnalului EMG trebuie să fie cât mai mică posibil, fără o filtrare inutilă și nu se recomandă denaturarea vârfurilor de semnal și a filtrelor notch.

În timpul procesării semnalului EMG sunt analizate doar valori pozitive. Când se efectuează rectificarea cu jumătate de undă, toate datele negative sunt aruncate și se păstrează datele pozitive. Valoarea absolută a fiecărui punct de date este utilizată în timpul rectificării cu undă completă. De obicei pentru rectificare, este preferată rectificarea cu undă completă.

1.4 DETECȚIA ȘI DESCOMPUNEREA SEMNALELOR EMG

Detectarea precisă a evenimentelor discrete din EMG (cum ar fi schimbarea de fază a modelului de activare asociat cu inițierea răspunsului motor rapid) este o problemă importantă în analiza sistemului motor. Au fost propuse mai multe metode pentru detectarea sincronizării timpului de pornire și oprire a mușchiului.

Cea mai frecventă metodă de rezolvare a evenimentelor legate de motor din semnalele EMG constă în inspecția vizuală de către observatori instruiți. „Metoda cu un singur prag”, care compară semnalul EMG cu un prag fix, este cea mai intuitivă și obișnuită metodă computerizată pentru a localiza timpul debutului activității de contracție musculară. Această tehnică se bazează pe compararea semnalelor prime rectificate și un prag de amplitudine a cărui valoare depinde de puterea medie a zgomotului de fundal. Metoda poate fi utilă pentru a depăși unele dintre problemele legate de inspecția vizuală. Cu toate acestea, acest tip de abordare nu este, în general, satisfăcătoare, deoarece rezultatele măsurate depind puternic de alegerea pragului [7].

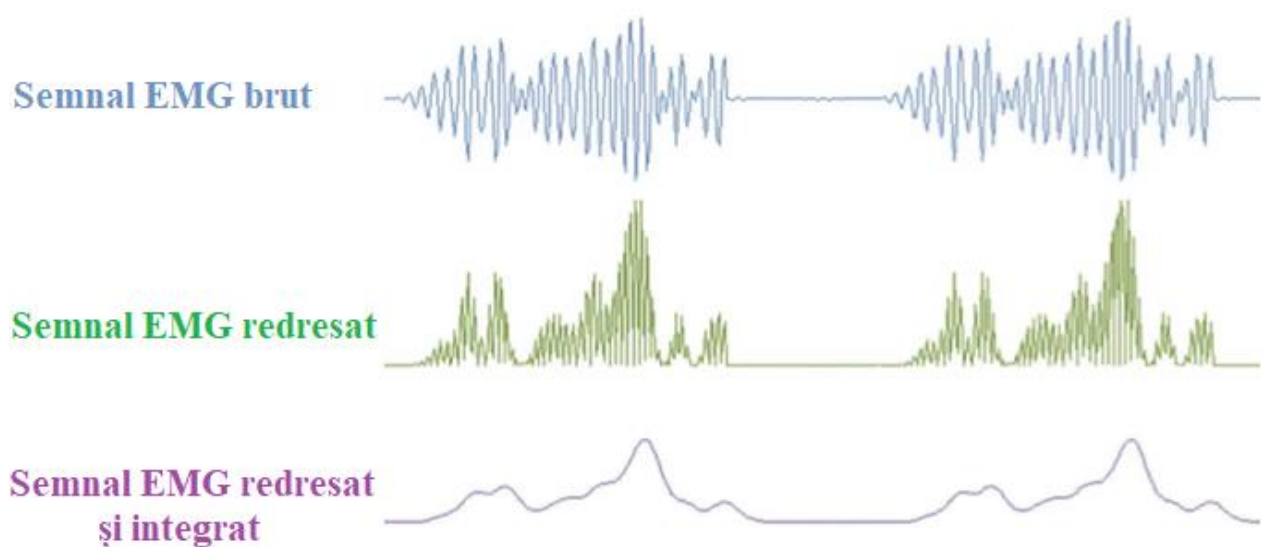


Figura 1.2 – Procesarea semnalului [3]

Semnalele EMG sunt suprapunerea activităților mai multor unități motorii. Este necesar să descompunem semnalul EMG pentru a dezvălui mecanismele referitoare la controlul muscular și nervos. În ceea ce privește descompunerea EMG, au fost concepute diverse tehnici. Descompunerea semnalului EMG s-a făcut prin potrivirea spectrului de undă și prin analiza principală a componentelor coeficienților de undă.

Tehnica dezvoltată pentru descompunerea semnalului EMG cu mai multe unități constă din patru proceduri separate: procedura de decompoziție a semnalului, procedura de detectare a vârfurilor, procedura de clasificare și procedura de separare. Conform unor cercetători, numai coeficienții de undă ai benzilor de frecvență joasă sunt mai importanți în diferențierea caracterizării potențialului de acțiune (AP) față de benzile superioare. Acest concept este unul subiectiv conceput empiric [4].

CAPITOLUL 2

COMPONENTE HARDWARE ȘI METODE DE IMPLEMENTARE

2.1 INTRODUCERE

În această lucrare, se propune crearea unui set de date EMG de suprafață preluate la nivelul antebrațului, ce va conține o serie de gesturi executate de diferiți subiecți. În cele ce urmează, se oferă mai întâi o imagine de ansamblu asupra schemei propuse. Sunt introduși subiecții recrutați pentru acest studiu, fapt urmat de descrierea protocolului experimentului. Experimentul servește drept dovadă de concept pentru schema propusă. De asemenea, se vor prezenta instrumentele hardware utilizate în cadrul studiului desfășurat.

2.2 SCHEMA PROPUȘĂ

În figura prezentată mai jos, este descrisă serializarea acțiunilor executate pentru identificarea mișcărilor.

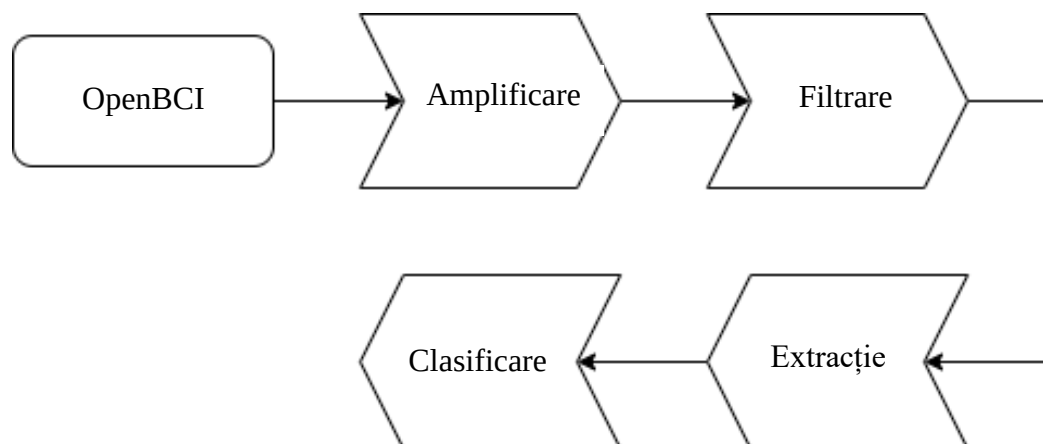


Figura 2.1 – Schema bloc

2.3 SETUL DE DATE EMG

Am stabilit să recunoaștem șase gesturi astfel încât să se genereze un set de date biometrice care să conțină gesturile de bază. Scopul este de a putea clasifica suficiente gesturi astfel încât o persoană cu dizabilități să poată comunica eficient cu ceilalți, să apuce obiecte și să indice diferite direcții.

Gesturile propuse sunt:

- ✓ Extensia dorsală a încheieturii
- ✓ Flexia palmară a încheieturii
- ✓ Extensia degetelor
- ✓ Flexia degetelor
- ✓ Supinație
- ✓ Pronație

Gestul inițial, cu care se începe experimentul este poziția neutră, în care se și revine după efectuarea anumitor gesturi.



Figura 2.2 – Setul de gesturi: (1) poziție neutră, (2) extensia dorsală a încheieturii, (3) flexia palmară a încheieturii



Figura 2.3 – Setul de gesturi: (1) flexia degetelor, (2) extensia degetelor

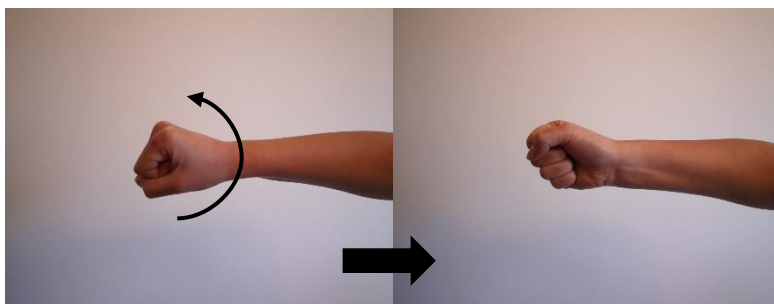


Figura 2.4 – Setul de gesturi: Supinație

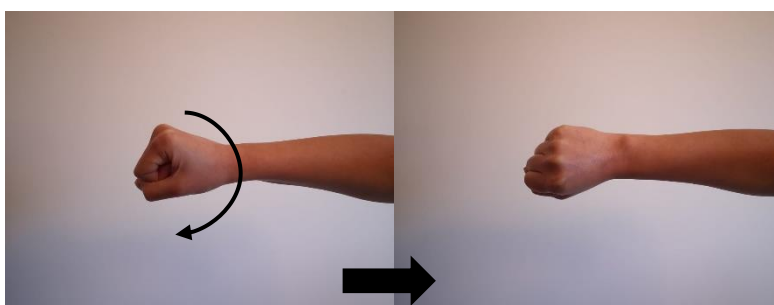


Figura 2.5 – Setul de gesturi: Pronație

Datele sunt colectate de la 10 subiecți cu vârste cuprinse între 20 și 55 de ani, atât femei cât și bărbați. Fiecare participant este instruit să execute toate gesturile de 4 ori cu pauze de două minute după fiecare ciclu. Fiecare gest este menținut pentru patru secunde, durată în care se înregistrează activitatea EMG. Astfel, există siguranța că datele au fost corect etichetate.

2.4 COMPONENTE HARDWARE

2.4.1 OPENBCI

Hardware-ul utilizat în această teză este placa OpenBCI de 32 biți (Figura 2.6). Aceasta este o interfață neurală cu 8 canale cu un procesor pe 32 de biți, compatibilă cu Arduino. La baza dispozitivului se află microcontrolerul PIC32MX250F128B, oferindu-i mai multă memorie locală și viteze de procesare mai rapide. Datele sunt prelevate la 250 Hz pe fiecare din cele opt canale [8].

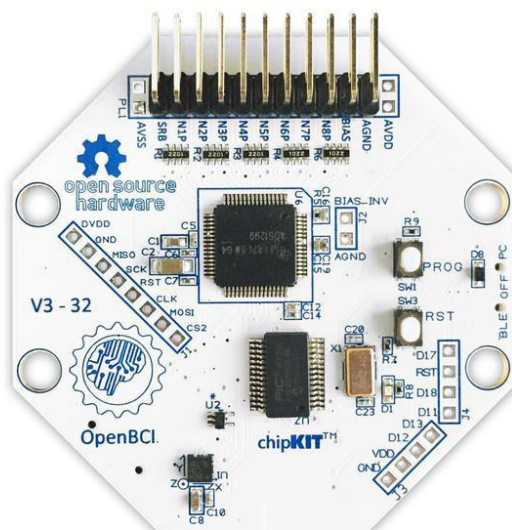


Figura 2.6 – Placa de dezvoltare OpenBCI [8]

Acest circuit este conceput special pentru măsurători biopotențiale. Firmware-ul OpenBCI este expus pe placă, putând fi utilizat pentru a proba activitatea creierului (EEG), activitatea musculară (EMG) și activitatea inimii (ECG). Placa comunică wireless cu un computer prin intermediul dongle-ului USB programabil OpenBCI, care se bazează pe modulul radio RFDuino [8].

În lucrarea de față, plăca OpenBCI se va utiliza pentru vizualizarea activității musculare.

2.4.2 ARDUINO UNO

În primul rând, Arduino este una dintre cele mai populare plăci de dezvoltare utilizate în dezvoltarea de proiecte electronice, deoarece este simplu de configurat, are un număr destul de mare de pini și este compatibil cu un număr mare de auxiliare care permit adăugarea diverselor funcționalități.

Specificații tehnice [12]:

- ✓ Microcontroller: ATmega328P
- ✓ Memorie flash: 32 KB
- ✓ SRAM: 2 KB
- ✓ EEPROM: 1 KB
- ✓ Viteza ceasului: 16 MHz
- ✓ 14 pini digitali: pot fi folosiți atât ca intrare, cât și ca ieșire. În plus, unii dintre ei au funcții specializate:
 - Pini 0 (RX), 1 (TX): acești pini sunt folosiți pentru a primi (RX) și a transmite (TX) date seriale TTL
 - Pini 2, 3: întreruperile externe
 - Pini 3, 5, 6, 9, 10, 11: PWM pe 8 biți
 - SPI: Pini 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)
 - Pinul 13: LED; există un LED încorporat conectat la pinul digital 13
- ✓ 6 pini analogici: sunt folosiți ca intrări analogice



Figura 2.7 – Placa de dezvoltare ArduinoUNO [12]

2.4.3 SENZORII MYOWARE

Senzorii musculari MyoWare reprezintă cea mai recentă tehnologie de senzori de electromiografie de la Advancer Technologies.

Majoritatea sistemelor standard de măsurare EMG sunt scumpe și nu sunt fezabile pentru ceea ce se propune în lucrarea de față, și anume un dispozitiv cu performanțe mari la costuri nu foarte ridicate; produsele de la companii de echipamente de cercetare precum BIOPAC, DelSys sau Biometrics Ltd costă mult peste 6.000 USD sau chiar 10.000 USD [3].

În Figura 2.4 este prezentată placa principală a senzorului, care colectează și amplifică semnalele EMG.

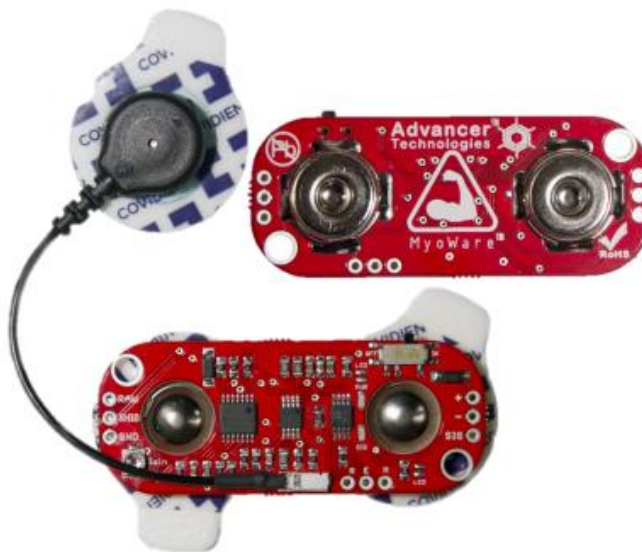


Figura 2.8 – Senzor MyoWare [9]

Deși MyoWare desemnează ce electrod trebuie să fie așezat la capătul mușchiului și care la mijlocul acestuia, în realitate, atâta timp cât doi electrozi sunt plasați pe direcția fibrelor musculare, nu contează așezarea. Electroful de referință („R”) trebuie plasat, în mod ideal, pe o proeminență osoasă, iar electrozii „M” și „E” pot avea orice orientare.

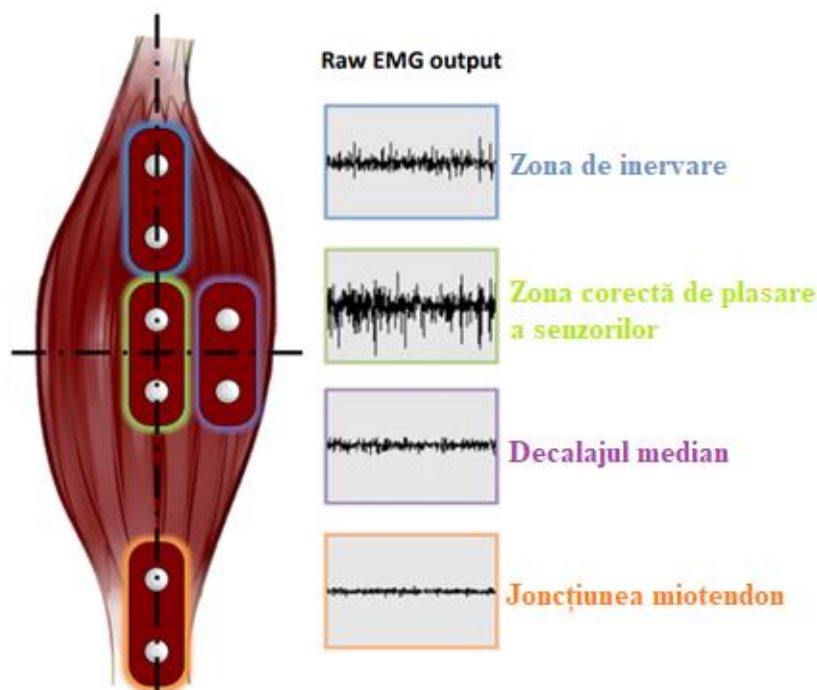


Figura 2.9 – Plasarea senzorului MyoWare [9]

MyoWare oferă, de asemenea, 3 conexiuni („R”/„E”/„M”) pentru fiecare dintre acești electrozi, în cazul în care se dorește utilizarea altor electrozi în afara celor încorporați [9].

Senzorul emite semnale redresate și integrate în mod implicit, însă oferă și posibilitatea de a obține semnale brute, dacă se dorește acest lucru.

2.4.4 OLIMEX EKG/EMG

Acesta este un scut EKG/EMG care permite plăcilor Arduino să capteze semnale de electrocardiografie/electromiografie. Acesta poate monitoriza bătăile inimii și înregistra pulsul, dar poate fi utilizat și pentru recunoașterea gesturilor prin analizarea activității musculare așa cum se dorește în acest proiect.

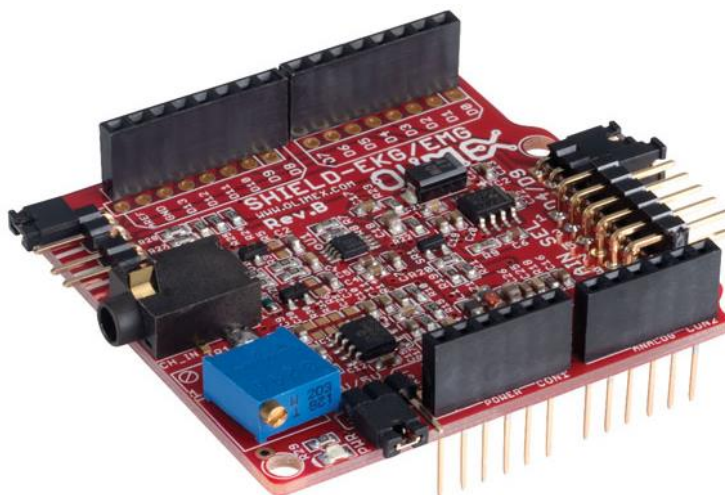


Figura 2.10 – Scutul Olimex [17]

Dispozitivul EKG-EMG convertește semnalul diferențial analog (bio-potențialele ECG/EMG generate de mușchi), atașat la intrările sale CH1_IN+/CH1_IN, într-un singur flux de date ca semnal de ieșire. Semnalul de ieșire este analog și trebuie discretizat în continuare pentru a da opțiunea procesării digitale. De obicei, acest lucru se face printr-un convertor Analog-Digital dedicat, încorporat în MCU-ul plăcii de bază (cum ar fi: ArduinoUno, OLIMEXINO-328, OLIMEXINO-32U4, OLIMEXINO-STM32, etc) [17].

Specificații tehnice [17]:

- ✓ Conectori așezați în stivă – până la 6 canale pot fi stivuite și conectate la intrările analogice A0-A6
- ✓ Generarea semnalului de calibrare prin ieșirile digitale D4/D9
- ✓ Potentiometru precis pentru calibrare
- ✓ Conector de intrare pentru electrozi standard sau activi
- ✓ Funcționează atât cu plăci Arduino de 3.3V cât și cu cele de 5V

CAPITOLUL 3

EXPERIMENTUL

3.1 ACHIZIȚIA DE DATE

Cei afectați de accidente în urma cărora au rezultat amputaje de mână, prezintă încă mușchii antebrăului, care sunt parțial responsabili de mișcările degetelor și ale mâinii. Astfel, acești mușchi sunt potriviți pentru a activa acțiunile de control pe un membru protetic. În mod ideal, pacientul trebuie să fie capabil să folosească aceiași mușchi pe care îi utiliza pentru diverse gesturi înainte de pierderea mâinii, ceea ce duce la o interacțiune naturală cu dispozitivul protetic. Electrozii pasivi de suprafață sunt cei mai rentabili și cea mai rapidă opțiune de a obține semnale EMG.

Cu toate acestea, utilizarea semnalelor EMG de suprafață aduce limitări, în sensul că este posibilă numai măsurarea mușchilor superficiali, semnalele EMG putând fi corupte de alte activități de natură electrică reflectate rezidual la nivelul epidermei. Tabelul 3.1 prezintă mușchii superficiali care sunt responsabili de mișcările mâinii. Mușchiul flexor superficial al degetelor (Flexor Digitorul Superficialis) este, însă, o excepție, deoarece este parțial acoperit de Brachio Radialis și mușchiul pronator (Pronator Teres) și de tendoanele palmar lung (Palmaris Longus) și flexor radial al carpului (Flexor Carpi Radialis). Deoarece tendoanele palmar lung și flexor radial al carpului nu afectează semnalele EMG, electrozii plasați pe aceste zone sunt utili pentru a măsura activitatea mușchiului flexor superficial al degetelor [10]. Prima provocare a fost găsirea mușchilor dați.

Mușchii și acțiunea acestora [11]:

- Mușchiul flexor radial al carpului (palmarul mare): realizează flexia mainii pe antebraț acționând asupra articulației radiocarpene și mediocarpene. Realizează abducția mainii. Participă slab la flexia antebrațului pe braț și pronație.
- Mușchiul palmar lung (mușchiul palmar mic): flexia mâinii (slab flexor al antebrațului); nu are nici o acțiune asupra înclinărilor laterale pentru că trece prin axul sagital al carpului.
- Mușchiul flexor ulnar al carpului (cubital anterior): realizează flexia și adducția mâinii. Participă parțial la flexia cotului.
- Mușchiul lung extensor radial al carpului: este extensor al mâinii, abductor al mâinii. Participă la flexia antebrațului pe braț.
- Mușchiul scurt extensor radial al carpului: extensor al mâinii, participă parțial la flexia antebrațului.
- Mușchiul extensor ulnar al carpului (cubital posterior): realizează extensia și adducția mâinii. Participă slab la extensia antebrațului.
- Mușchiul flexor superficial al degetelor: realizează flexia falangei mijlocii pe falanga proximală și a acesteia pe metacarpene. Participă la flexia mâinii și parțial la flexia antebrațului.
- Mușchiul extensor comun al degetelor: la nivelul degetelor I-IV realizează extensia falangei distale pe cea medie, a acesteia pe cea proximală, a falangei proximale pe metacarpian și a mâinii pe antebraț. Extensia falangei proximale este puternică în timp ce extensia celorlalte falange e insuficientă, ea fiind completată de interosoși.

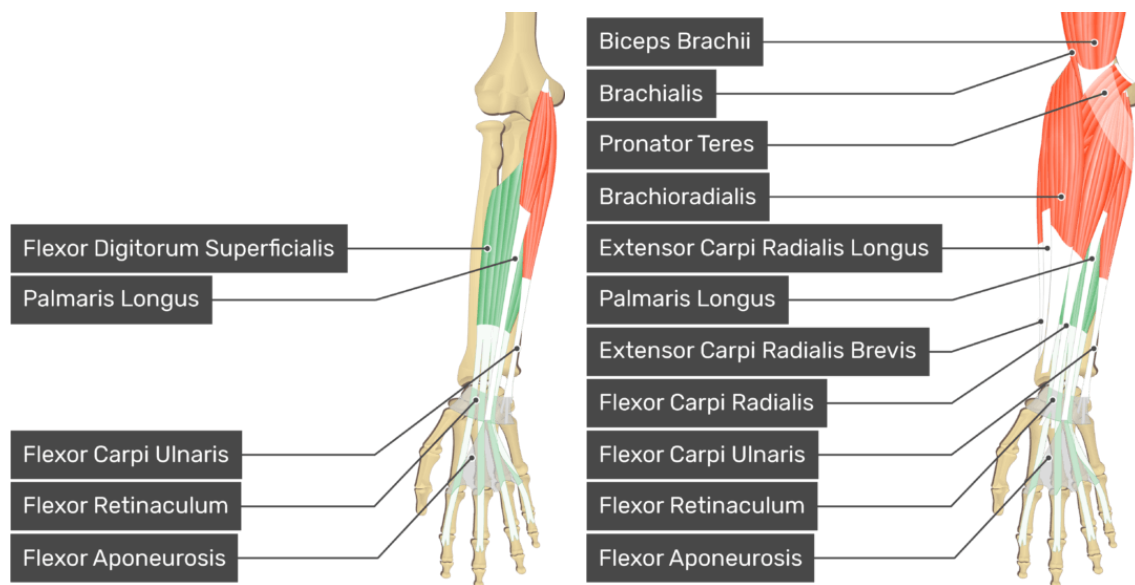


Figura 3.1 – Anatomia mușchilor mâinii și ai antebrațului [10]

Pentru a găsi locația mușchilor, a fost necesar ca brațul participantului să fie comparat cu o figură anatomică a mușchilor antebrațului, ca în reprezentarea de mai sus (Figura 3.1). În mod empiric, am constatat că cea mai eficientă metodă pentru găsirea unui mușchi este de a găsi tendonul de lângă încheietura mâinii și apoi de a flexa mușchiul executând mișcări pronunțate. Astfel, se poate simți forma mușchiului prin deplasarea de-a lungul tendonului până la începerea mușchiului. După ce s-a găsit mușchiul presupus, doi electrozi au fost atașați conform setării bipolare de pe centrul mușchiului

de-a lungul fibrelor. Ulterior mușchiul a fost măsurat pentru diferite mișcări. Un indiciu că mușchiul a fost identificat în mod corect, este dat atunci când mișcările preconizate se potrivesc cu cele măsurate.

Dacă corelațiile măsurate între mușchi și mișcări se conformează corelațiilor din Tabelul 3.1, este probabil să se găsească mușchiul corect. Această abordare părea a fi cea mai convenabilă pentru a găsi mușchii utili. Tabelul 3.2 prezintă corelațiile determinate empiric între mușchi și mișcări. Tabelul 3.1 comparativ cu tabelul 3.2 arată că nu orice mușchi ar putea fi măsurat cu instrumentele date. Spre exemplu, mușchiul flexor ulnar al carpului nu a putut fi măsurat. Pe de altă parte, semnalele ar trebui să poată fi clasificate, deoarece nici o mișcare nu are aceeași combinație musculară.

Mușchi/ Funcție	Încheietură				Degete		Articulație radioulnară	
	Flexie palmară	Extensie dorsală	Adducție ulnară	Adducție radială	Flexie palmară	Extensie dorsală	Supinație	Pronație
Flexor radial al carpului	x			x				x
Flexor ulnar al carpului	x		x					
Palmar lung	x						x	x
Lung extensor radial al carpului	x	x						
Scurt extensor radial al carpului		x						
Extensor ulnar al carpului		x	x					
Flexor superficial al degetelor	x				x			
Extensor comun al degetelor		x				x		
Pronator								x

Tabelul 3.1 – Mușchii și funcțiile acestora

Mușchi/ Funcție	Încheietură				Degete		Articulație radioulnară	
	Flexie palmară	Extensie dorsală	Adducție ulnară	Adducție radială	Flexie palmară	Extensie dorsală	Supinație	Pronație
Palmar lung	x						x	x
Flexor superficial al degetelor	x				x			
Extensor comun al degetelor		x				x		
Pronator								x

Tabelul 3.2 – Mușchii ce pot fi măsurați și gesturile rezultate

Poziția și orientarea electrozilor senzoriali musculari au un efect vast asupra puterii semnalului. Electrozii trebuie să fie amplasați în mijlocul corpului muscular și trebuie să fie aliniați cu orientarea fibrelor musculare. Plasarea senzorului în alte locații va reduce rezistența și calitatea semnalului senzorului datorită reducerii numărului de unități motorii măsurate și interferenței atribuite intersecției.

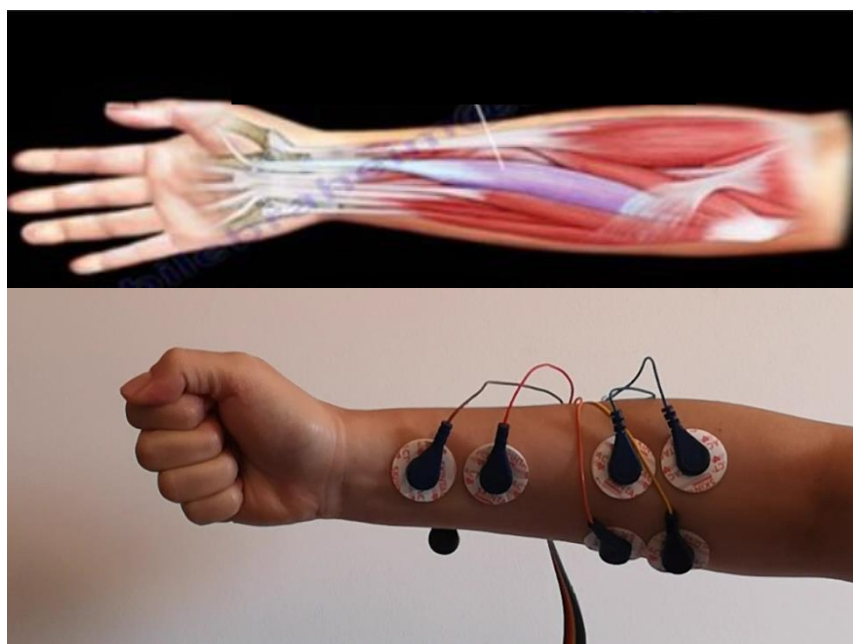


Figura 3.2 – Plasarea electrozilor pe antebraț

Doi electrozi sunt plasați la 1-2 cm unul de celălalt paralel cu lungimea fibrelor musculare pe linia mediană a mușchiului. Apoi, un alt electrod, electrodul de referință, este plasat pentru a scădea zgomotul de semnal și tensiunile comune între cei doi electrozi musculari. Acesta trebuie poziționat cât mai departe posibil de electrodul principal, în mod ideal pe o proeminență osoasă (în cazul de față a fost plasat pe cot).

Pentru colectarea datelor de la dispozitivul OpenBCI, a fost ales un singur canal și trei electrozi au fost plasați pe brațul drept al subiectului, în următorul mod: doi dintre aceștia au fost atașați la pinii 1+/1- și celălalt a fost atașat la pinul de jos, funcționând ca masă comună. În timpul procesului, s-a constatat că rezultatele variau în funcție de plasarea electrozilor. Semnalul EMG este produs de schimbarea potențialului electric, care este generat de celulele musculare. Ca urmare, gesturile produse de diferite părți ale mușchiului vor genera potențiale distincte. Poziția optimă a electrozilor este determinată prin efectuarea mai multor încercări ale experimentului de achiziție (prezentat în Figura 3.2). În plus, s-a demonstrat că mediul de testare este un alt parametru care trebuie controlat pentru a obține date brute de înaltă calitate.

În cazul utilizării senzorilor MyoWare, deși aceștia desemnează ce electrod trebuie să fie așezat la capătul mușchiului și care la mijlocul acestuia, în realitate, atâta timp cât doi electrozi sunt plasați pe direcția fibrelor musculare, nu contează așezarea. Electrocul de referință („R”) trebuie plasat, în mod ideal, pe o proeminență osoasă, iar electrozii „M” și „E” pot avea orice orientare.

3.2 POTENȚIALE SOLUȚII ÎN ANALIZĂ

Principiile colectării semnalului EMG de suprafață sunt prezentate în Figura 3.3.

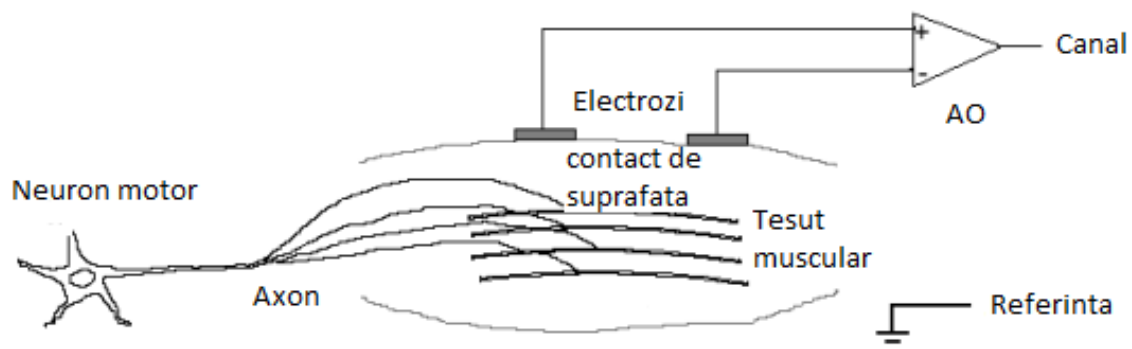


Figura 3.3 – Principiile colectării semnalului EMG

Inițial, am luat în analiză mai multe opțiuni:

1. Olimex utilizat în ecosistemul ArduinoUNO
2. Senzorii MyoWare conectați la placa ArduinoUNO
3. OpenBCI Cynton cu ecosistemul propriu

Cele trei soluții candidate vor fi descrise în cele ce urmează. Din considerente de substanță, se va califica pentru concluzii finale soluția care va fi identificată ca fiind cea mai robustă în raport cu enunțul studiului.

3.2.1 OLIMEX

Shield-ul Olimex EKG-EMG este un modul de extensie pentru platforma Arduino menit să ofere capabilități de colectare și monitorizare a semnalului electromiografic funcționând cu un electrod de referință și alte două intrări diferențiale.

Dispozitivul EMG Olimex este alcătuit din următoarele elemente constructive [17]:

- ✓ Mecanism de protecție la supratensiuni pe intrarea dinspre electrozi, pentru evitarea deteriorării datorate în principal încărcării electrostatice potențiale la nivelul electrozilor și firelor de conexiune;
- ✓ Un filtru de rejecție a frecvențelor înalte, în afara intervalului de interes
- ✓ Un amplificator de instrumentație urmat de un filtru trece sus cu un pol situat la 0,16 Hz
- ✓ Un al doilea etaj de amplificare cu câștig reglabil urmat tot de un filtru trece sus cu un pol situat la 0,16 Hz
- ✓ Un filtru Besselworth de ordin 3 cu $F_c=40\text{Hz}$.

Semnalul rezultat este transferat către o intrare analogică în Arduino Uno, acesta putând fi achiziționat în calculator prin intermediul interfeței seriale, cu ajutorul programului listat în *Anexa 3*.

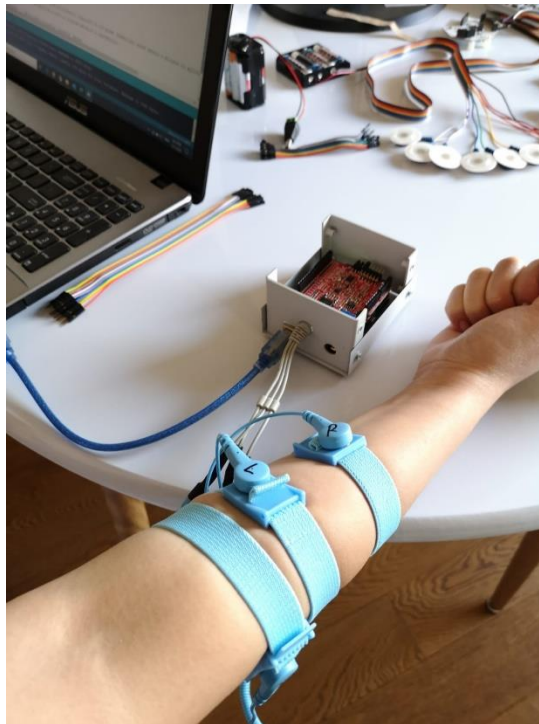


Figura 3.4 – Olimex: evaluarea unui singur mușchi

3.2.2 MYOWARE

Majoritatea sistemelor standard de măsurare EMG sunt scumpe și nu sunt fezabile pentru uz personal. Produsele de la companii de echipamente de cercetare precum BIOPAC, DelSys sau Biometrics Ltd costă mult peste 6.000 de dolari, putând ajunge chiar și până la 10.000 de dolari. Prin urmare, senzorul muscular MyoWare oferă un punct de plecare foarte avantajos, cu un raport calitate-preț bun [9].

Această soluție reprezintă o a doua etapă în studiul sistemelor BCI și propune simularea mișcării unui braț. În acest punct se studiază plasarea optimă a senzorului Myo pe unul din mușchii antebrăului, astfel încât semnalul să fie maxim, iar mișcarea să se poată distinge pe un clește de prindere cu servomotor.

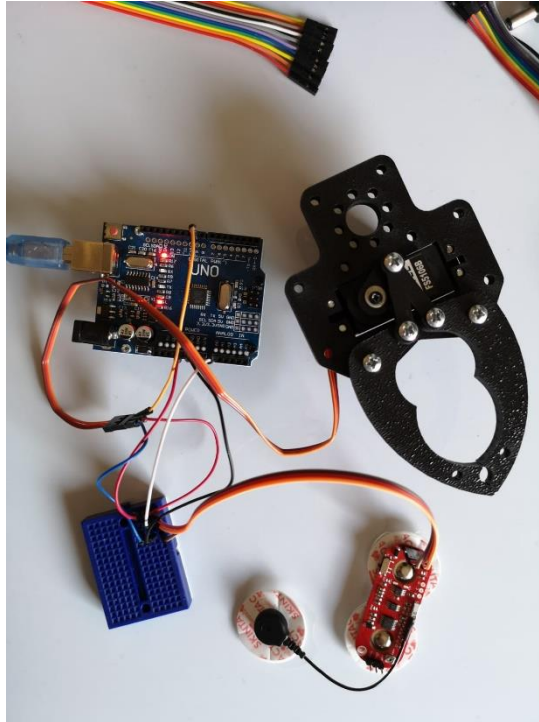


Figura 3.5 – Montaj pregătire mișcare clește prindere cu servomotor

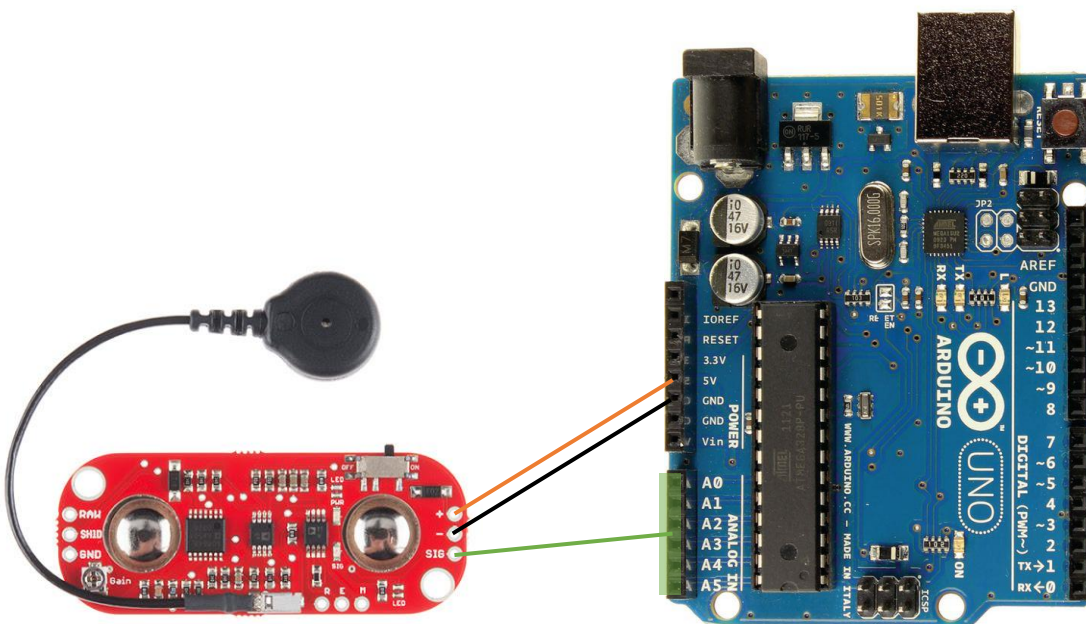


Figura 3.6 – Conexiuni Senzor MyoWare – placa de dezvoltare ArduinoUNO

Electrozii au fost plasați așa cum a fost descris anterior, de-a lungul unui singur mușchi. Aplicația care evidențiază mișcarea se regăsește în *Anexa 2*.

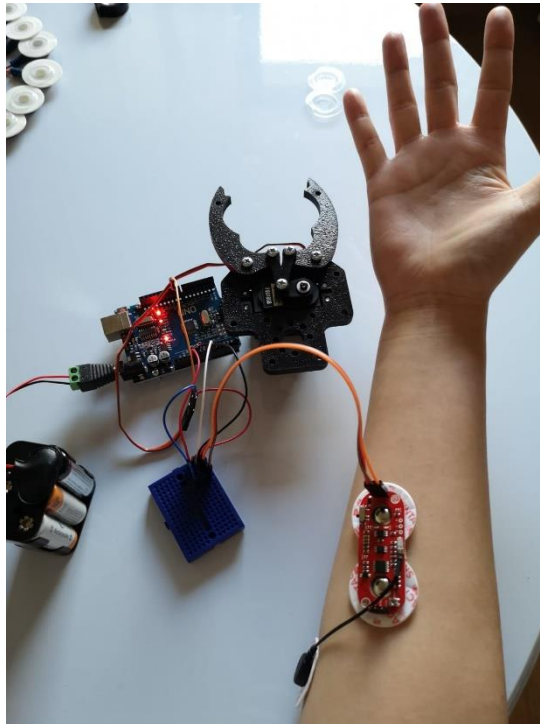


Figura 3.7 – MyoWare: evaluarea unui singur mușchi

3.2.3 OPENBCI

Pentru cea de-a treia soluție, care va fi utilizată și descrisă în detaliu în cadrul acestei teze, se utilizează placa de dezvoltare OpenBCI.

Inițial, pentru colectarea datelor de la dispozitivul OpenBCI, a fost ales un singur canal și trei electrozi au fost plasați pe brațul drept al subiectului, în următorul mod:

- ✓ doi dintre aceștia au fost atașați la pinii 1 P/N
- ✓ cel de-al treilea a fost atașat la pinul de jos, funcționând ca masă comună.

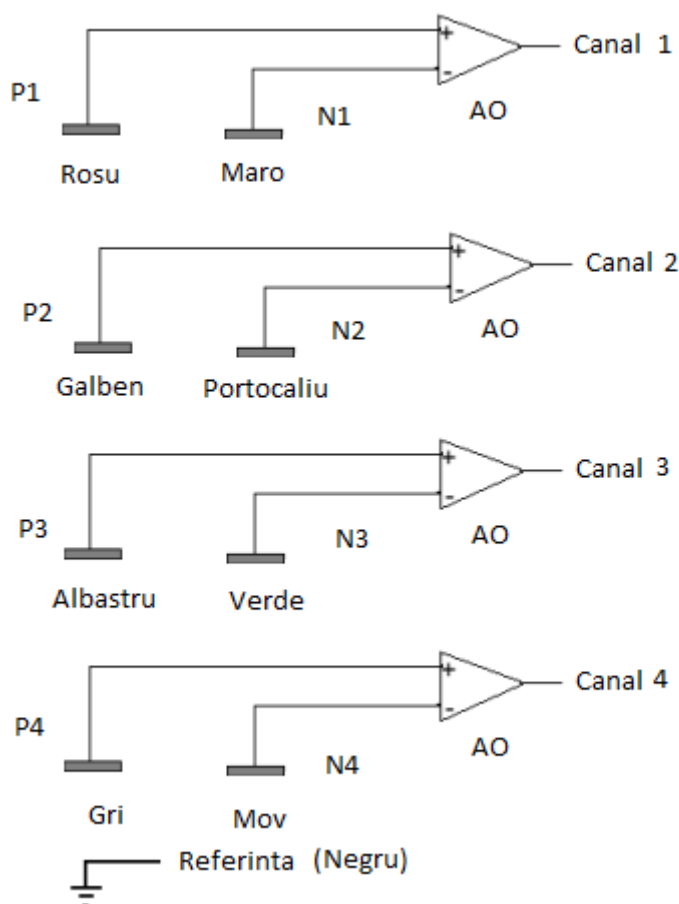


Figura 3.8 – Schema utilizând nouă electrozi (limitare fizică conexiuni existente)

În timpul procesului, s-a constatat că rezultatele variaau în funcție de plasarea electrozilor. Semnalul EMG este produs de schimbarea potențialului electric, care este generat de celulele musculare. Ca urmare, gesturile produse de diferite părți ale mușchiului vor genera potențiale distincte. Așezarea optimă a electrozilor este determinată prin efectuarea mai multor încercări ale experimentului de achiziție.

Așa cum se poate observa și în Figura 3.8, placa are pentru fiecare canal doi pini. Un pin N și un pin P, care pot fi conectați la un amplificator diferențial, ceea ce duce la măsurători bipolare. Conexiunea depinde de setarea multiplexorului circuitului.

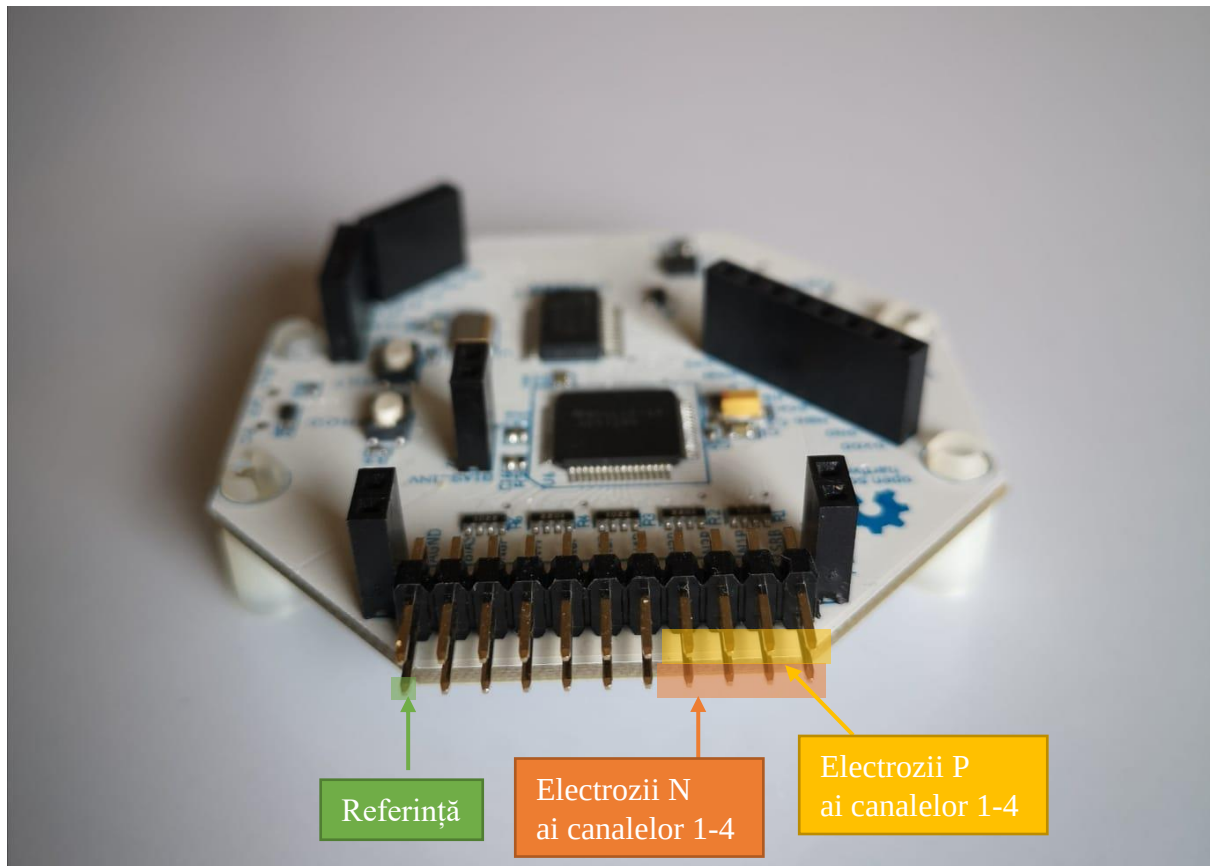


Figura 3.9 – Electrozii utilizați

Ordinea se citește de la dreapta la stânga. Rândul superior sunt pinii xP (semnal), iar pinii inferiori sunt pinii xN (de referință).

Această soluție este considerată optimă și a fost aleasă ca implementare principală în cadrul acestei lucrări datorită numărului de canale și posibilității plasării acestui tip de electrozi (se pot observa în Figura 3.10) cât mai eficient în funcție brațul subiectului.

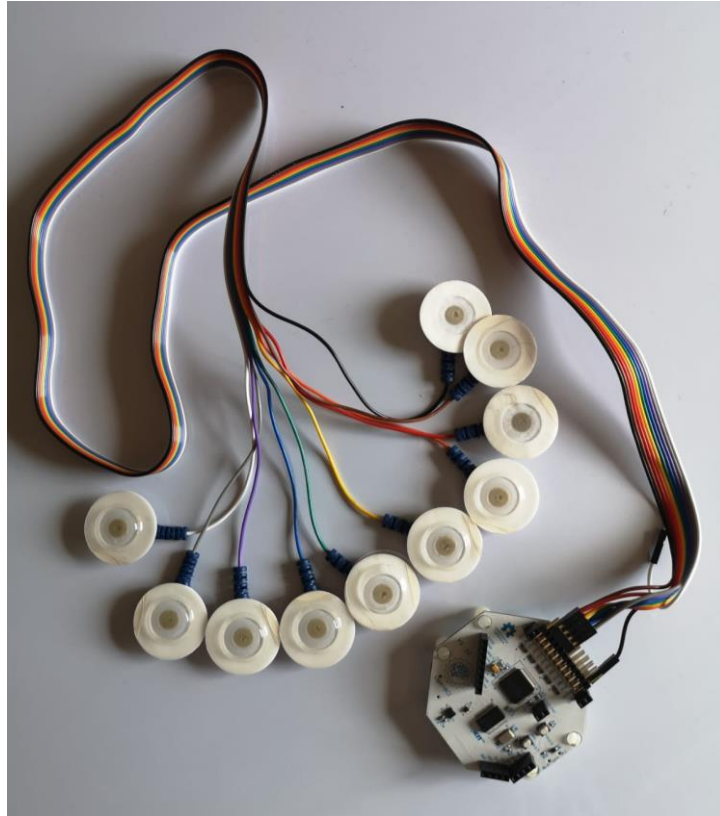


Figura 3.10 – Conectarea electrozilor

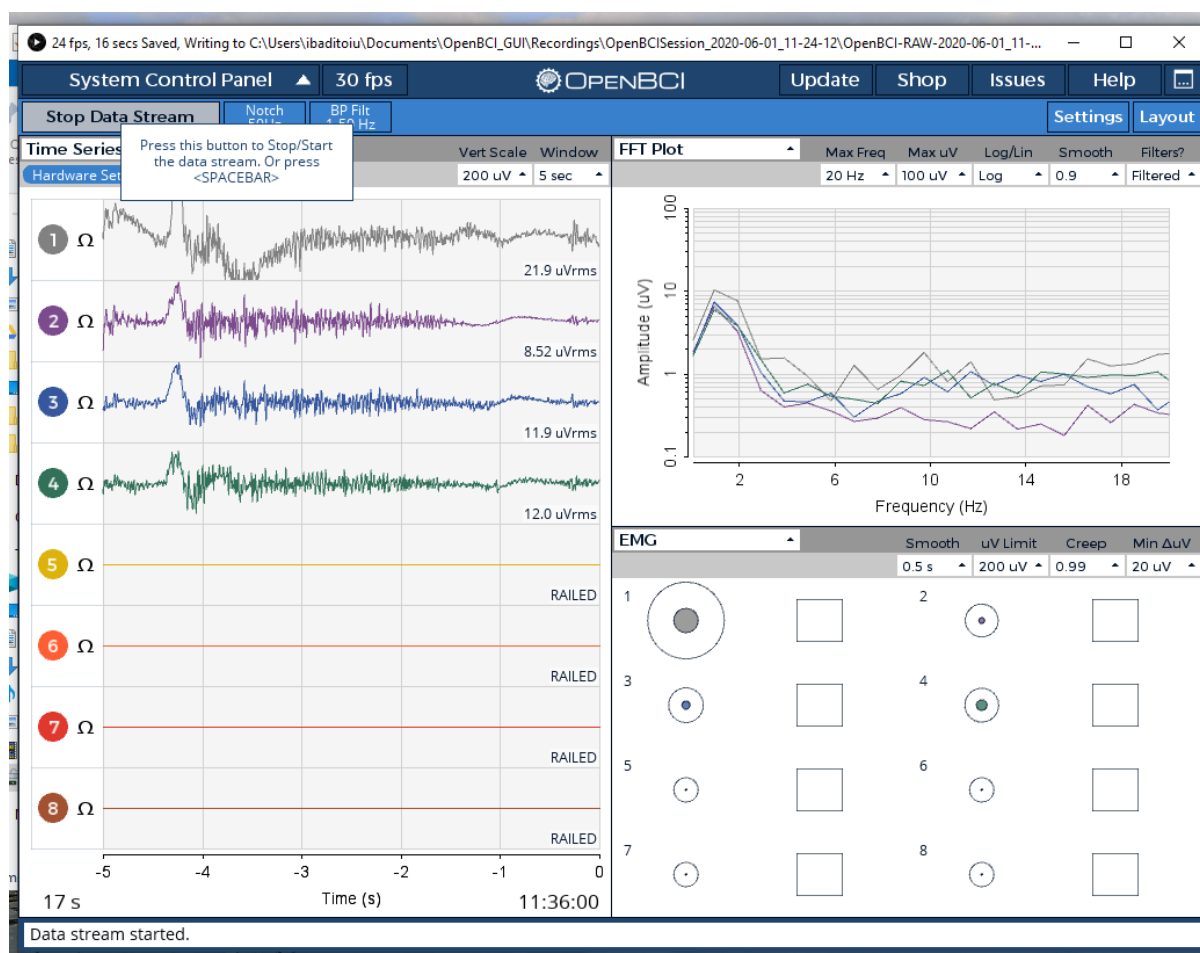


Figura 3.11 – Înregistrarea semnalelor EMG în softul OpenBCI

3.3 PROCESAREA SEMNALULUI EMG

Recunoașterea semnalelor EMG de suprafață ale gestului este compusă din trei secvențe: pre-procesarea datelor, extragerea caracteristicilor și clasificarea propriu-zisă. Pentru pre-procesarea datelor, s-au luat în calcul mai multe variante, una dintre ele utilizând un algoritm de segmentare pentru obținerea de date mai curate și pentru reducerea cantității de date. Algoritmul de extragere a caracteristicilor comprimă segmentele de semnal EMG de suprafață în vectori de trăsături. Aceste trăsături sunt proiectate astfel încât să sublinieze caracteristicile specifice clasei de gest ale semnalului EMG. Clasificatorul trebuie instruit cu vectorii de trăsături pentru a distinge gesturile diferite unele de altele cu o precizie ridicată.

În consecință, procedura adoptată a fost de sincronizare a măsurabilelor cu înregistrările video ale mișcărilor bazat pe marcare temporală. Totodată, s-au determinat maximele locale pentru o cât mai bună sincronizare între marcasele de timp ale înregistrării video și marcasele de timp ale măsurabilei. Astfel, a rezultat marcarea claselor pentru setul de antrenare. De asemenea, și pentru setul de test s-a procedat asemănător, realizând o clasificare apriori din considerente de validare post factum întregului proces.

1. Pre-procesare a datelor

OpenBCI are un filtru notch încorporat care ajută la eliminarea zgomotului de 50 Hz. În plus, se obțin rezultate bune dacă se reduce drastic influența curentului alternativ (AC) asupra semnalului [8].

Inițial, fără a utiliza filtrul Notch, testele reieșeau necorelabile chiar și cu o poziționare în diferite zone în raport cu perturbatorii electromagnetici din mediul de lucru. Filtrul notch a îmbunătățit rezultatele, motiv pentru care s-a optat pentru utilizarea acestuia. Au fost parcurse diversele tipuri de filtre disponibile în aplicația OpenBCI, rezultatele optime concretizându-se în mediul de lucru doar din perspectiva filtrului notch de 50 Hz.

Cu toate acestea, pentru rezultate optime, s-a studiat și folosirea filtrului Butterworth pentru a procesa semnalul original. Filtrul Butterworth este un filtru de procesare a semnalului și prezintă o curbă de răspuns plană în frecvență în banda de trecere (prezentat în Figura 3.12). În acest experiment, banda a fost aleasă de la 10 Hz la 50 Hz, cu o undulare de 3 dB și o atenuare de 40 dB.

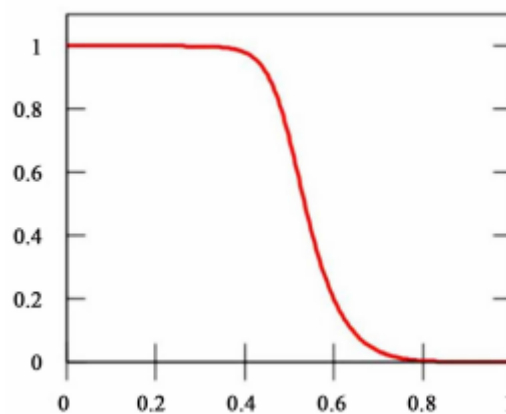


Figura 3.12 – Filtru Butterworth [6]

2. Extragerea caracteristicilor

Extragerea caracteristicilor este efectuată pe semnalele EMG de suprafață pentru a păstra tiparele semnalului cu scopul de a distinge diferite gesturi. În experimentul nostru, valorile medii absolute (MAV), rădăcina medie pătratică (RMS), variația și coeficienții modelului auto-regresiv (AR) sunt utilizate la modelarea diferitelor semnale EMG și apoi sunt introduse în rețea pentru clasificare. O scurtă descriere a calculului este prezentată mai jos.

Valoarea medie absolută este definită ca [6]:

$$MAV = \frac{1}{N} \sum_{n=1}^N |x_n|$$

Unde N este lungimea semnalului și x_n reprezintă semnalul EMG într-un segment.

Varianța – ecuația generală utilizată pentru a calcula varianța este [6]:

$$Var = \sqrt{\frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})^2}$$

Unde $\bar{x}$ este valoarea medie a semnalului EMG, N este lungimea semnalului și x_n reprezintă semnalul EMG într-un segment.

Radacina medie pătratică poate fi exprimată ca [6]:

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2}$$

Unde N este lungimea semnalului și x_n reprezintă semnalul EMG într-un segment.

Modelul auto-regresiv al unui semnal poate fi exprimat ca [6]:

$$x(n) = - \sum_{k=1}^p a_k x(n-k) + u(n)$$

unde a_k se referă la coeficientul de ordin k al semnalului, p se referă la ordinul modelului și $u(n)$ este zgomotul alb.

3. Clasificare

În recunoașterea modelului semnalului EMG de suprafață, diferiți algoritmi au fost folosiți pentru a atribui vectori de caracteristică. Grupul de clasificatori ai rețelei neuronale este utilizat pe scară largă datorită capacității sale de extindere și a capacității de a suporta atât cazuri complexe, cât și simple. Cu toate acestea, alegerea caracteristicilor și a restricțiilor de timp fac ca acest clasificator să fie excesiv de complex. Alții au preferat să utilizeze abordarea logică Fuzzy, care permite să insereze experiența utilizatorului în sistem și să se contrazică pentru schimbarea tiparelor.

În experimentul prezentat, am selectat, pentru început, algoritmul k-nearest neighbors (KNN) datorită simplității și practicabilității sale. Principiul KNN este clasificarea obiectelor printr-un vot plural al vecinilor săi, obiectul fiind atribuit clasei celei mai comune printre vecinii săi cei mai apropiați.

Cu toate acestea, clasificarea obiectelor necesită, de obicei, o cantitate mare de calcul, deoarece trebuie să calculăm distanța dintre fiecare punct de testare și restul pentru a obține cei mai apropiați k vecini. Din acest motiv a fost Support Vector Machine (SVM), care oferă o precizie ridicată în calibrare și clasificare. SVM este un clasificator discriminator definit în mod formal de un hiperplan de separare. Este utilizat pe scară largă în clasificarea binară, deoarece beneficiază de principiul de minimizare a riscurilor structurale și de evitare a supraajustării. Când sunt disponibile date de instruire limitate, SVM depășește de obicei metodele tradiționale de estimare a parametrilor.

Obiectivul algoritmului SVM este de a găsi o graniță de decizie într-un spațiu N-dimensional care să clasifice distinct datele (Figura 3.4). Maximizarea distanței de marjă între două hiperplane oferă o anumită întărire, astfel încât datele viitoare să poată fi clasificate cu mai multă încredere.

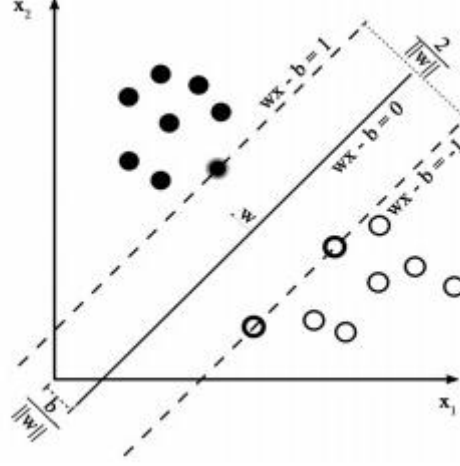


Figura 3.13 – Ilustrarea algoritmului SVM [6]

Se presupune ca există două gesturi care trebuie clasificate: G1 și G2. Se definește setul de antrenare cu eșantioanele:

$$\{(x_i, g_i)\}, i = 1, \dots, n$$

Unde $x_i \in R^d$, reprezentând un vector de caracteristici al unui gest.

$$g_i = \begin{cases} +1, & \text{dacă } x_i \text{ aparține G1 } (wx_i - b \geq 1) \\ -1, & \text{dacă } x_i \text{ aparține G2 } (wx_i - b \leq -1) \end{cases}$$

Un plan de separare poate fi scris ca:

$$wx - b = 0$$

Apoi, singura problemă este de a găsi marja maximă (w minim), care poate fi rezolvată prin funcția kernel.

Acuratețea este definită ca:

$$ACC = \frac{T_p}{T_p + F_p}$$

3.4 ALCĂTUIREA BAZEI DE SEMNALE

Așa cum a fost menționat și anterior, semnalele au fost preluate de la mai mulți subiecți, astfel încât să se poată crea o bază de semnal suficient de mare care să poată fi împărțită în setul de date de antrenare și setul de date testare.

Gesturile au fost executate în ordinea stabilită de fiecare subiect, cu o pauză de două minute între repetiții. Așa cum se poate observa și pe grafic, în Figura 3.14, fiecare gest a fost menținut aproximativ 4 secunde astfel încât să se poată înregistra activitatea EMG, iar prelucrarea și etichetarea datelor să se facă în mod corect.

Fiecare ciclu de gesturi a fost salvat într-un fișier separat, iar înregistrările au fost comparate astfel:

- ✓ Pentru aceeași persoană s-au comparat înregistrările de la aceeași mână
- ✓ Pentru aceeași persoană s-au comparat înregistrările de la mâna stângă și mâna dreaptă
- ✓ Pentru persoane diferite s-au comparat înregistrările de la mâinile stângi, respectiv, de la mâinile drepte.

Analizând datele, se poate concluziona cu ușurință că rezultatele pot fi afectate de mai mulți factori precum:

- ✓ poziționarea senzorilor; este dificil de identificat fasciculul muscular și poziționat senzorii astfel încât semnalul să fie maxim la persoane diferite
- ✓ sexul persoanei sau gradul de dezvoltare al musculaturii – în general bărbații au mușchii mai proeminenți, fiind mai ușor de identificat
- ✓ mâna de la care se preiau datele – mâna dominantă, cu care executăm în general majoritatea activităților, are musculatura mai bine definite
- ✓ vârsta și lipsa de activitate fizică; acestea determină atrofierea mușchilor – este bine să se seteze un interval de vârstă concludent pentru studiu

Mai jos, în Figura 3.14 (Subiectul 1) și 3.15 (Subiectul 2), este prezentată o comparație pe două seturi de gesturi, executate de persoane diferite. Se pot observa cu ușurință diferențele, provenite chiar din factorii prezentați anterior.

Subiectul 1 – sex: feminin, vârsta: 24 ani, nu suferă de afecțiuni musculare, mâna: dreapta

Subiectul 2 – sex: masculin, vârsta: 51 ani, nu suferă de afecțiuni musculare, mâna: dreapta

Gesturile au fost împărțite în următoarele clase:

- ✓ 0 – Extensia dorsală a încheieturii
- ✓ 1 – Flexia palmară a încheieturii
- ✓ 2 – Extensia degetelor
- ✓ 3 – Flexia degetelor
- ✓ 4 – Supinație
- ✓ 5 – Pronație

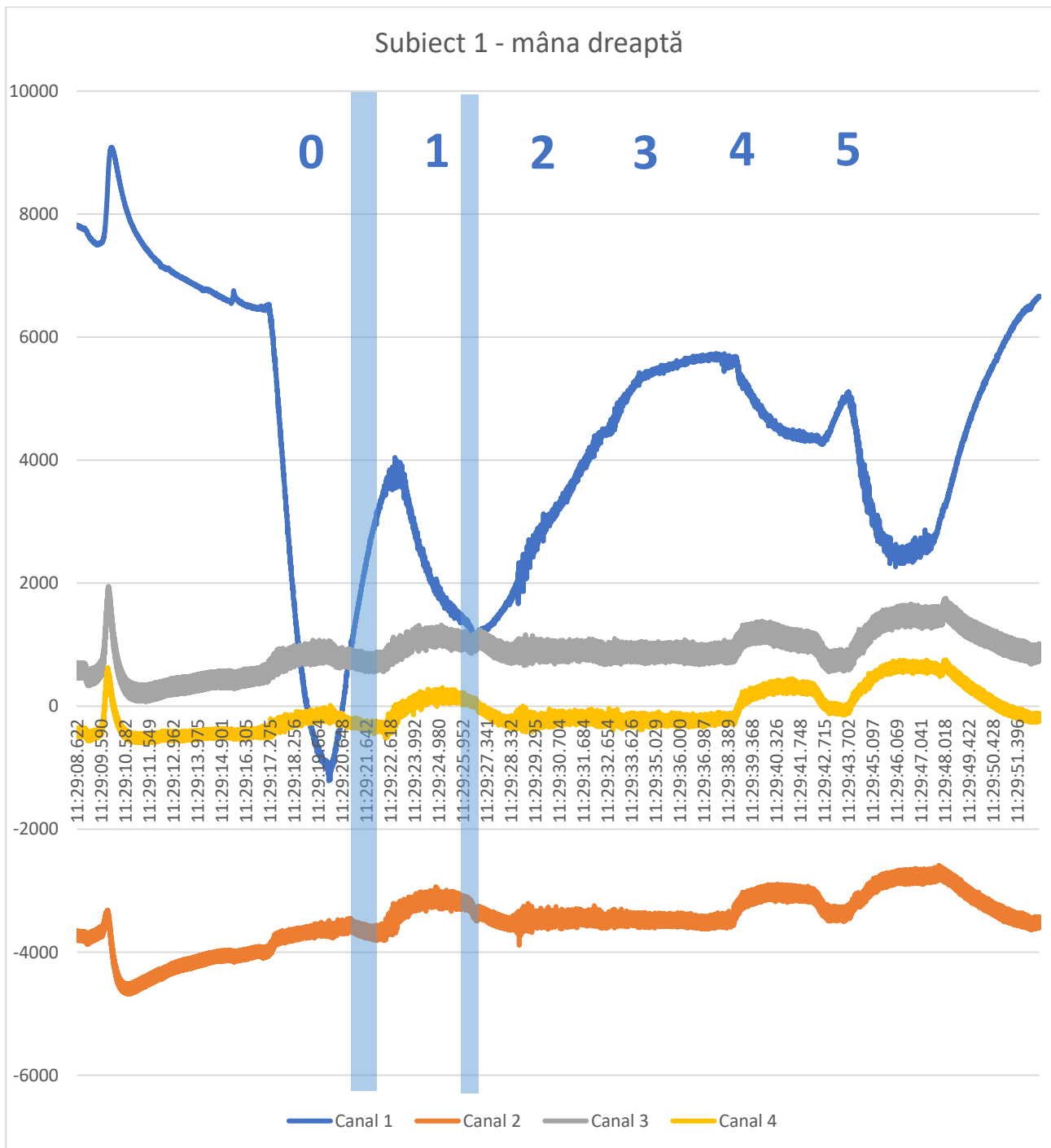


Figura 3.14 – Grafic gesturi subiect 1

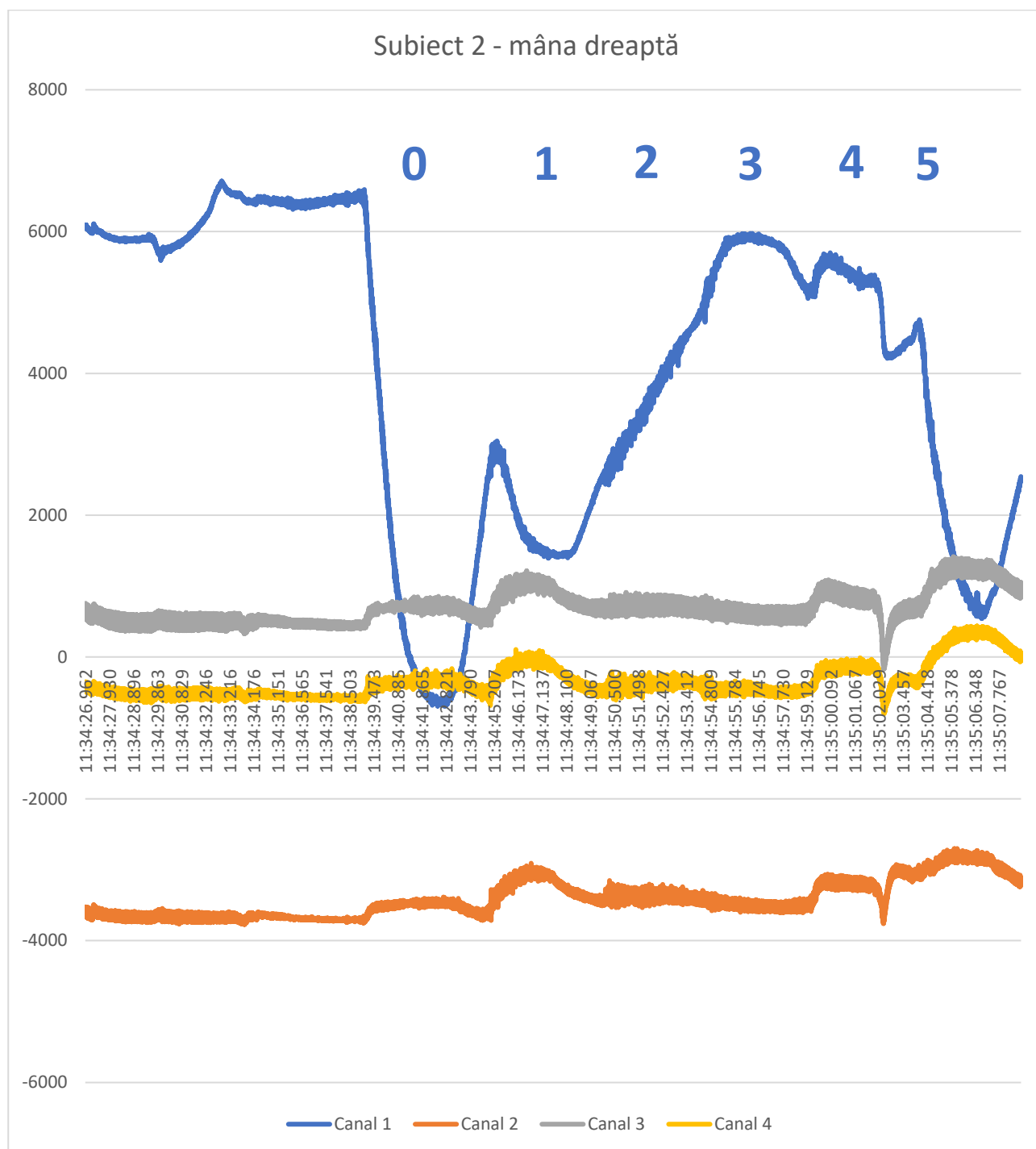


Figura 3.15 – Grafic gesturi subiect 2

S-a încercat extragerea caracteristicilor gesturilor din grafice prin împărțirea semnalului în funcție de durata unui gest (aproximativ patru secunde), dar și de punctele de inflexiune de pe fiecare canal. Se pot observa pe grafic și cele două reveniri în poziția neutră între:

- ✓ extensia dorsală a încheieturii (clasa 0) și flexia plamară a încheieturii (clasa 1)
- ✓ flexia plamară a încheieturii (clasa 1) și extensia degetelor (2).

Celelalte gesturi nu au necesitat revenirea în poziția neutră (extensia degetelor – flexia degetelor) sau, poziția de revenire nu a impactat rezultatul și nu se observă o inflexiune foarte vizibilă pe grafic (supinație – pronație).

Setul de date fost împărțit în 70% date de antrenare și 30% date testare. Datele de antrenare au fost etichetate cu clasele corespunzătoare, menționate mai sus.

CAPITOLUL 4

ANTRENAREA MODELULUI ȘI REZULTATE

4.1 PLATFORMA KAGGLE

Kaggle este o comunitate online de oameni de data science și practicieni de învățare automată. Aceasta permite utilizatorilor să găsească și să publice seturi de date, să exploreze și să construiască modele într-un mediu bazat pe data science, să lucreze cu alți oameni de știință și ingineri de învățare automată și să intre în diverse competiții.

Kernel-urile Kaggle sunt în esență notebook-urile Jupyter rulate din browser. Aceasta înseamnă că nu mai este nevoie de configurarea unui mediu local, codul scris putând fi accesat de oriunde din lume fără a fi necesară crearea unui repository pe GitHub/GitLab sau alte platforme de acest tip. Notebook-urile Jupyter constau dintr-o secvență de celule, în care fiecare celulă este formatată fie în Markdown (pentru scrierea textului), fie într-un limbaj de programare la alegere (pentru scrierea codului) [19].

Un alt avantaj este acela că puterea de procesare a notebook-ului provine de la serverele din cloud, nu de la mașina locală. Se poate adăuga gratuit un singur NVIDIA Tesla P100 la notebook. Mediile GPU (Graphics Processing Unit) au un procesor și o memorie principală mai reduse, dar sunt o modalitate excelentă de a realiza viteze semnificative pentru anumite tipuri de studii, cum ar fi antrenarea rețelelor neuronale pe bazele de date de imagine. Unul dintre avantajele majore ale utilizării notebook-urilor, spre deosebire de o mașină locală sau a propriei mașini virtuale, este că acest mediul este deja preconfigurat cu software-ul GPU și pachete care pot fi consumatoare de timp

în cazul configurării. Disponibilitatea gratuită a GPU este limitată, în orele de vârf, fiind posibilă plasarea într-o coadă.

De asemenea, se poate adăuga gratuit un TPU (Tensor Processing Unit) v3-8 la notebook. TPU-urile sunt acceleratoare hardware specializate în sarcini de învățare profundă. Acestea sunt suportate de Tensorflow 2.1 atât prin API-ul Keras de nivel înalt, cât și, la un nivel inferior, în modele care utilizează o buclă de instruire personalizată. Disponibilitatea TPU gratuită este, de asemenea, limitată.

Notebook-urile Kaggle rulează într-un mediu de calcul la distanță. În momentul scrierii, fiecare sesiune de editare a notebook-ului este prevăzută cu următoarele resurse [19]:

- ✓ 9 ore timp de execuție
- ✓ 5 Gigabytes spațiu de stocare pe disc cu salvare automată (în directorul /kaggle/working)
- ✓ Spațiu suplimentar care nu va fi salvat în afara sesiunii curente (exterior directorului /kaggle/working)

Specificații CPU (Central Processing Unit) [19]:

- ✓ 4 nuclee CPU
- ✓ 16 Gigabytes RAM

Specificații GPU [19]:

- ✓ 2 nuclee CPU
- ✓ 13 Gigabytes RAM

4.2 TENSORFLOW

TensorFlow este o platformă de învățare automată end-to-end disponibilă gratuit. Sau, altfel spus, un strat de infrastructură pentru o programare diferențiată. Are un ecosistem cuprinzător și flexibil de instrumente, biblioteci și resurse, care permite cercetătorilor să accedă la cele mai noi tehnologii de învățare automată și dezvoltatorilor să construiască și să implementeze cu ușurință aplicații bazate pe acestea [16].

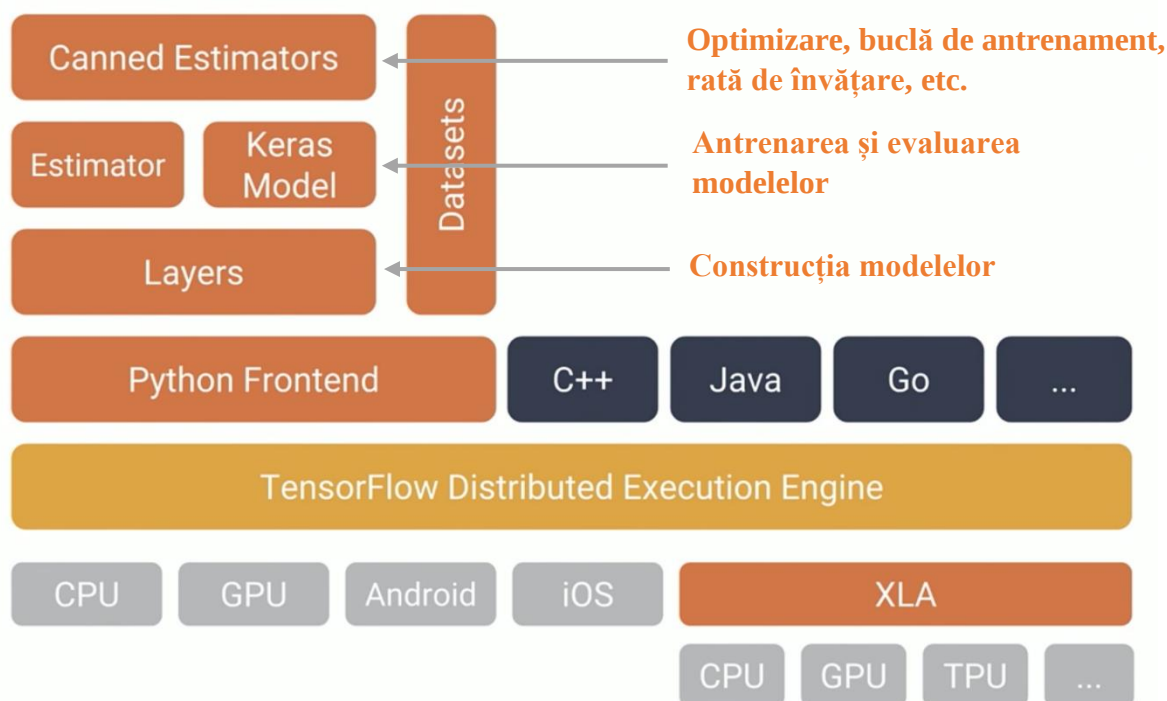


Figura 4.1 – Arhitectura TensorFlow [13]

Acesta combină patru abilități cheie [16]:

- ✓ Executarea eficientă a operațiilor de tensor la nivel de bază pe CPU, GPU sau TPU
- ✓ Calcularea gradientului expresiilor arbitrare diferențiale
- ✓ Scalarea calculului pe mai multe dispozitive (de exemplu, supercomputerul Summit de la Laboratorul Național Oak Ridge, care se întinde pe 27.000 de GPU-uri)
- ✓ Exportul de programe („grafice”) în sisteme de rulare externe, cum ar fi servere, browsere, dispozitive mobile și încorporate.

4.2.1 KERAS

Keras este API-ul de nivel înalt al TensorFlow 2.0: o interfață abordabilă, extrem de productivă pentru rezolvarea problemelor de învățare automată, cu accent pe învățarea profundă modernă. Acesta oferă abstracții esențiale și elemente de bază pentru dezvoltarea și livrarea soluțiilor de învățare a mașinilor cu viteză mare de iterație [14].

Keras se poate rula pe TPU sau pe clustere mari de GPU, iar modelele se pot exporta pentru a rula în browser sau pe un dispozitiv mobil.

Structurile de bază ale datelor Keras sunt straturile și modelele. Cel mai simplu tip de model este modelul Secvențial, reprezentând o stivă liniară de straturi [14]. În cadrul studiului de față, s-a utilizat API-ul funcțional care poate gestiona modele cu topologie neliniară, modele cu straturi partajate și modele cu intrări sau ieșiri multiple.

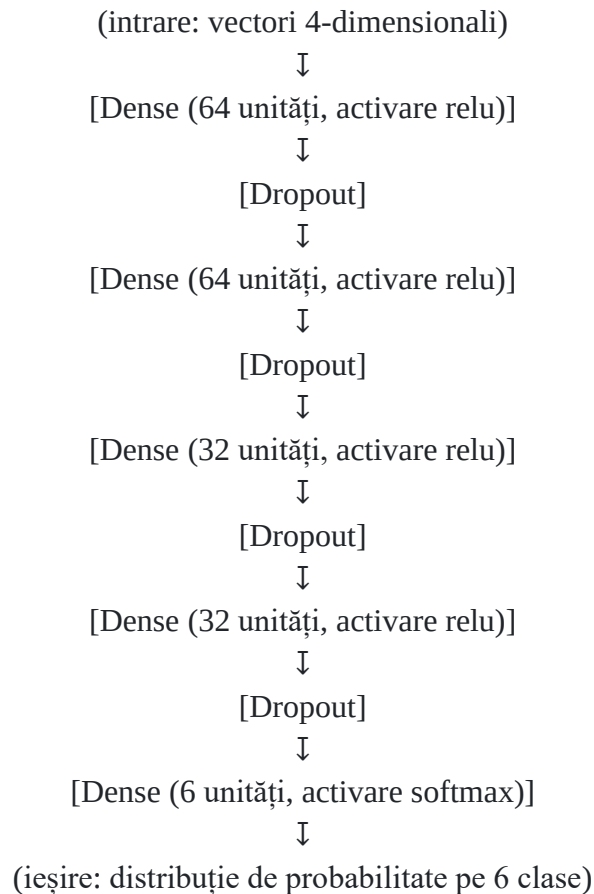
Ideea principală este că un model de învățare profundă este de obicei un grafic aciclic direcționat de straturi. Prin urmare, API-ul funcțional este o modalitate de a construi grafice de straturi.

Un flux de lucru end-to-end este, în general, format din:

- 2 Antrenare
- 3 Validare pentru un set generat din datele de antrenare originale
- 4 Evaluare pe datele de test

În cazul de față, se consideră setul de date creat și descris anterior (*Capitolul 3*) prin prelucrarea semnalelor EMG de suprafață.

Se consideră modelul utilizat (acesta este un grafic cu cinci straturi):



Pentru a vizualiza sumarul modelului se apelează: `model.summary()`

```

Model: "model_3"
-----
Layer (type)                Output Shape              Param #
-----
input (InputLayer)          (None, 4)                 0
-----
hidden_1 (Dense)            (None, 64)               320
-----
dropout_1 (Dropout)         (None, 64)               0
-----
hidden_2 (Dense)            (None, 64)               4160
-----
dropout_2 (Dropout)         (None, 64)               0
-----
hidden_3 (Dense)            (None, 32)               2080
-----
dropout_3 (Dropout)         (None, 32)               0
-----
hidden_4 (Dense)            (None, 32)               1056
-----
dropout_4 (Dropout)         (None, 32)               0
-----
output (Dense)              (None, 6)                198
-----
Total params: 7,814
Trainable params: 7,814
Non-trainable params: 0
-----

```

Figura 4.2 – Sumarul modelului

Un mod standard de a salva un model funcțional este de a apela *model.save()* pentru a salva întregul model ca un singur fișier. Ulterior, se poate recrea același model din acest fișier, chiar dacă codul care a construit modelul nu mai este disponibil.

Acest fișier salvat include:

- ✓ arhitectura modelului
- ✓ scorurile modelului (care au fost învățate în timpul antrenamentului)
- ✓ configurarea antrenării modelului, dacă există (în varianta în care a fost compilat)
- ✓ optimizatorul și starea acestuia, dacă există (pentru a putea reporni antrenamentul din starea în care a rămas).

Modelul poate fi exprimat și ca un grafic utilizând: *tf.keras.utils.plot_model(model, "model.png")* sau *tf.keras.utils.plot_model(model, "model.png", show_shapes=True)*. Codul aferent regăsește în Anexa 1.

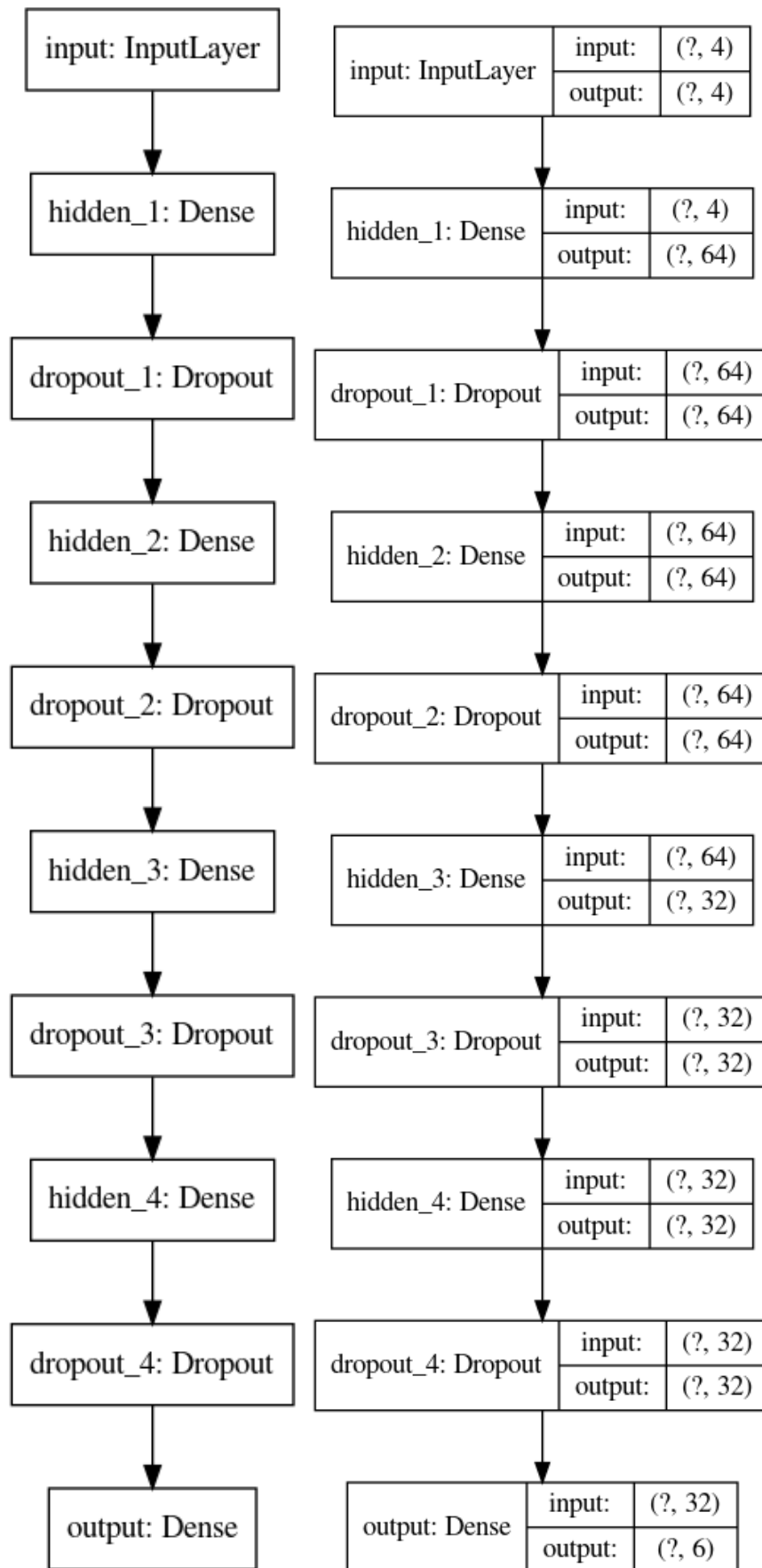


Figura 4.3 – Modelul utilizat

4.2.2 FUNCȚII DE ACTIVARE

Funcția de activare *ReLU* – aplică funcția de activare liniară rectificată. Cu valori implicite, aceasta returnează activarea standard ReLU: $\max(x, 0)$, maximul în valoare de 0 și tensorul de intrare. Modificarea parametrilor impliciți permite utilizarea pragurilor diferite de zero, modificarea valorii maxime a activării și utilizarea unui multiplu diferit de zero pentru intrare pentru valori sub prag [20].

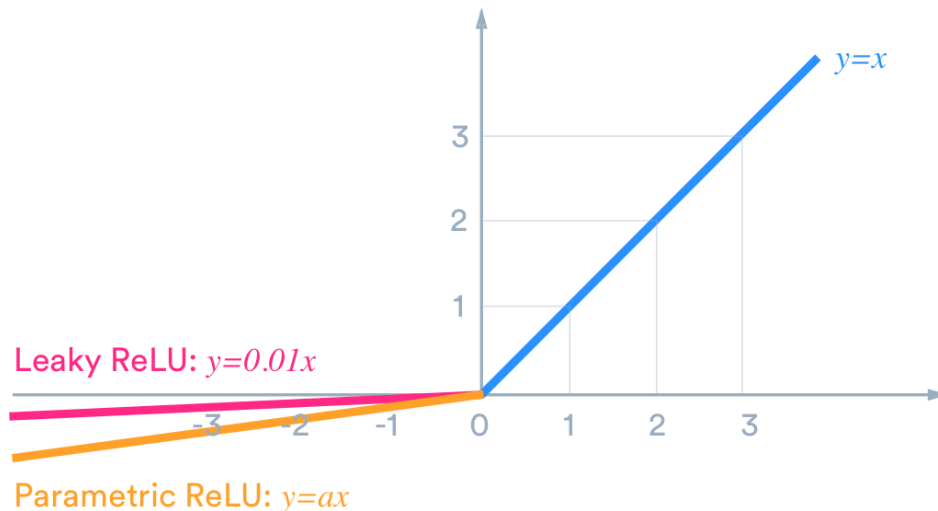


Figura 4.4 – Funcția de activare ReLU [20]

Funcția de activare *softmax* – convertește un vector real într-un vector cu probabilități categorice. Elementele vectorului de ieșire sunt în intervalul $(0, 1)$ și se însumează la 1. Fiecare vector este gestionat independent. Argumentul axei stabilește cărei axe de intrare este funcția aplicată [20].

Softmax este adesea folosit ca activare pentru ultimul strat al unei rețele de clasificare, deoarece rezultatul ar putea fi interpretat ca o distribuție de probabilitate.

Softmax-ul fiecărui vector x este calculat ca $\exp(x)/\text{tf.reduce_sum}(\exp(x))$. Valorile de intrare sunt probabilitatea de înregistrare a probabilității rezultate.

Dropout este o tehnică de îmbunătățire a preciziei prin reducerea supraîncadrării. Funcționează prin eliminarea aleatorie a unităților (și a conexiunilor lor) din rețea în timpul antrenamentului. Acest lucru împiedică fenomenul co-adaptării unităților, forțând fiecare neuron să învețe o caracteristică utilă pentru calcularea ieșirii corecte. În special, dropout-ul aleator al unităților stratului interesat în timpul antrenamentului este reglat de probabilitatea de cădere p , pentru care o valoare comună este $p = 0,5$. La fiecare trecere înainte, fiecare unitate este anulată independent și aleatoriu, pornind de la distribuția Bernoulli parametrizată de p . Mai mult, rezultatele sunt înmulțite cu $1/(1-p)$. Dropout-ul este activ numai în timpul antrenamentului: în timpul inferenței, acesta este dezactivat (sau, echivalent, probabilitatea de cădere p este setată la 0). Dropout-ul este aplicat în general imediat înainte de straturile CONV sau FC [15].

Optimizatorul Adam este o metodă de coborâre a gradientului stocastic, care se bazează pe estimarea adaptativă a momentelor de ordin întâi și ordin doi. Metoda este eficientă din punct de vedere al calculului, are o cerință de memorie redusă și este invariantă la redimensionarea în diagonală a gradientilor [15].

Deoarece setul de date nu este foarte mare, s-au utilizat tablouri NumPy în memorie pentru a antrena și evalua modelul.

`tf.keras.Model.fit` primește trei argumente importante:

- ✓ *epochs*: instruirea este structurată în epoci. O epocă este o iterație prin întregul set de date de intrare (aceasta se face în loturi mai mici, numite batch-uri).
- ✓ *batch_size*: modelul structurează datele în loturi mai mici și se iterează peste aceste loturi în timpul antrenamentului. Acest număr întreg specifică dimensiunea fiecărui lot. Ultimul lot poate fi mai mic dacă numărul total de eșantioane nu este divizibil după dimensiunea lotului.
- ✓ *validation_data*: atunci când se realizează prototipul unui model, se dorește monitorizarea cu ușurință a performanțelor sale pe unele date de validare. Se dă acest argument (un tuplu de intrări și etichete) care permite modelului să afișeze pierderile și valorile în modul inferență pentru datele introduse, la sfârșitul fiecărei epoci.

```
Train on 2755 samples, validate on 2755 samples
Epoch 1/1024
- 0s - loss: 330.5030 - accuracy: 0.3292 - val_loss: 13.1684 - val_accuracy: 0.6958

Epoch 00001: val_loss improved from inf to 13.16836, saving model to model.h5
Epoch 2/1024
- 0s - loss: 86.6518 - accuracy: 0.4436 - val_loss: 5.1821 - val_accuracy: 0.7946

Epoch 00002: val_loss improved from 13.16836 to 5.18209, saving model to model.h5
Epoch 3/1024
- 0s - loss: 48.2784 - accuracy: 0.4661 - val_loss: 3.2276 - val_accuracy: 0.5434

Epoch 00003: val_loss improved from 5.18209 to 3.22760, saving model to model.h5
Epoch 4/1024
- 0s - loss: 30.9859 - accuracy: 0.4842 - val_loss: 1.4531 - val_accuracy: 0.6548

Epoch 00004: val_loss improved from 3.22760 to 1.45311, saving model to model.h5
Epoch 5/1024
- 0s - loss: 20.8403 - accuracy: 0.5154 - val_loss: 1.0907 - val_accuracy: 0.6120

Epoch 00005: val_loss improved from 1.45311 to 1.09075, saving model to model.h5
Epoch 6/1024
- 0s - loss: 12.6828 - accuracy: 0.4915 - val_loss: 1.1756 - val_accuracy: 0.6007

Epoch 00006: val_loss did not improve from 1.09075
Epoch 7/1024
- 0s - loss: 8.7933 - accuracy: 0.4316 - val_loss: 1.5853 - val_accuracy: 0.2432
```

Figura 4.5 – Exemplu rulare

4.2.3 MATRICEA DE CONFUZIE

O matrice de confuzie descrie performanța modelului de clasificare. Cu alte cuvinte, matricea confuziei este o modalitate de a rezuma performanța clasificatorului. Figura următoare prezintă o reprezentare de bază a unei matrice de confuzie [18]:

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Figura 4.6 – Reprezentare de bază a matricei de confuzie [15]

Coloanele matricei reprezintă etichetele de predicție, iar rândurile reprezintă etichetele reale. Matricea de confuzie este întotdeauna o matrice 2-D de formă $[n, n]$, unde n este numărul de etichete valabile pentru o sarcină de clasificare dată. Atât predicția cât și etichetele trebuie să fie tablouri unidimensionale de aceeași formă pentru ca această funcție să ruleze.

```
[[1035  44  167  68  16  2]
 [ 70 541 107  0  0  1]
 [  0  27 606 17  2  0]
 [  0  0  70 1167 41  0]
 [  2  0  1  215 727  0]
 [  0 25  56 401  61 314]] confusion matrix
75.91215632024901 acc
```

Figura 4.7 – Matricea de confuzie rezultată în urma rulării

4.2.4 ACURATEȚEA

Inițial, acuratețea rezultată în urma rulării programului pe setul de antrenament creat a fost aproximativ 60%. Deoarece acesta nu reprezenta un rezultat satisfăcător din punct de vedere al studiului efectuat, s-a încercat îmbunătățirea prin adăugarea unui alt strat în modelul creat.

```
Epoch 02048: val_loss did not improve from 0.94945  
<keras.callbacks.callbacks.History object at 0x7f7e0c748cd0>  
[[942  37 168  44 141  0]  
 [140 290 288  0  0  1]  
 [  0 129 467  16 40  0]  
 [  0  0 166 791 321  0]  
 [  1  0 102 196 646  0]  
 [  0 132 153 361  57 154]] confusion matrix  
56.89088708282898 acc
```

Figura 4.8 – Matricea de confuzie și acuratețea generată în urma primelor teste efectuate

Acuratețea s-a îmbunătățit, crescând până la 79%. Însă aceasta este impactată și de setul de date. Populația de măsurători a fost limitată datorită contextului curent, putându-se obține rezultate mai bune prin lărgirea ariei subiecților.

S-a încercat și o structură de tip codor-decodor a rețelei cu patru straturi și 32, 64, 64, 32 neuroni npe fiecare strat, însă nu s-au putut observa alte îmbunătățiri.

CONCLUZII

CONCLUZII GENERALE

Semnalele EMG sunt utilizate în multe aplicații clinice și biomedicale, ca instrument de diagnostic pentru identificarea bolilor neuromusculare, evaluarea durerilor de spate scăzute, kinesiologiei și tulburărilor de control motor. Semnalele EMG sunt de asemenea utilizate ca semnal de control pentru dispozitive protetice, cum ar fi mâinile, brațele și membrele protetice. EMG poate fi utilizat și pentru a sesiza activitatea musculară izometrică în care nu se produce nicio mișcare. Aceasta permite definirea unei clase de gesturi subtile folosite pentru a controla interfețele fără a fi observate și fără a perturba mediul înconjurător.

Aceste semnale pot fi utilizate pentru a controla o proteză sau ca un semnal de control pentru un dispozitiv electronic, cum ar fi un telefon mobil.

Totodată, se poate recunoaște vorbirea prin observarea activității EMG a mușchilor asociați cu aceasta. Este destinat utilizării în medii zgomotoase și poate fi util pentru persoanele fără corzi vocale și persoanele cu afazie. EMG a fost de asemenea utilizat ca semnal de control pentru calculatoare și alte dispozitive. Un dispozitiv de interfață bazat pe EMG ar putea fi utilizat pentru a controla obiecte în mișcare, cum ar fi roboți mobili sau scaune cu roțile electrice.

Dispozitivele electromiografice cu costuri reduse precum placa OpenBCI au devenit recent populare. De asemenea, evoluția accelerată a domeniului imprimantelor 3D conduce la o mai ușoară realizare a dispozitivelor protetice personalizate. Datorită costurilor lor reduse, atât BCI-urile, cât și

dispozitivele protetice imprimate tridimensionale ar putea fi utilizate ca o alternativă ieftină pentru amplificarea forței de manipulare.

Însă nu numai persoanele cu dizabilități ar putea beneficia de sisteme BCI. Un alt posibil scenariu ar fi controlul unui braț robotic pe o distanță lungă și interacțiunea acestuia cu mediul. De exemplu, interacțiunea în zonele cu risc radiologic, chimic sau biologic ar putea fi posibilă, ceea ce ar putea contribui la minimizarea riscurilor intervenționale în cazul catastrofelor.

Semnalele EMG au fost vizate drept control pentru sistemele de zbor. Grupul Human Senses de la NASA Ames Research Center din Moffett Field, CA încearcă să avanseze interfețele om-mașină prin conectarea directă a unei persoane la un calculator. În acest proiect, un semnal EMG este utilizat pentru a înlocui tastele de comandă mecanice și tastaturile. EMG a fost, de asemenea, utilizat în cercetarea pentru o armură care angajează gesturi bazate pe EMG pentru a manipula întrerupătoarele și butoanele de control necesare pentru zbor în combinație cu un ecran pe bază de ochelari.

CONTRIBUȚII PERSONALE

În termeni de contribuții personale, m-am concentrat, în primul rând, pe investigarea diferitelor abordări pentru plasarea electrozilor. Locațiile mușchilor de suprafață ai antebrațului trebuiau identificate și poziționate în scopul maximizării semnalului colectat pentru mișcările propuse. Din constrângerile hardware au putut fi utilizate doar semnale provenite din patru canale, ceea ce donotă faptul că simultan au putut fi măsurate maxim patru grupe de mușchi. Gesturile propuse pentru a fi clasificate au fost următoarele:

- ✓ Extensia dorsală a încheieturii
- ✓ Flexia palmară a încheieturii
- ✓ Extensia degetelor
- ✓ Flexia degetelor
- ✓ Supinație
- ✓ Pronație

Acestea au provenit dintr-un studiu al corelațiilor între mușchi și mișcări.

În cadrul acestui studiu, am investigat dacă și ce metode ale învățării automate oferă rate de clasificare utile pentru a extrage informația din zgomotului electric multiplu existent la nivelul corpului.

Am prezentat diferitele etape de preprocesare consacrate în literatura EMG, și anume, valoarea medie absolută, varianța, rădăcina medie pătratică și modelul auto-regresiv.

Platforma de învățare automată utilizată pentru antrenarea modelului a fost TensorFlow, împreună cu bibliotecile specifice.

DEZVOLTĂRI ULTERIOARE

Având în vedere analiza mai multor dispozitive precum: OpenBCI Cyton, Myo și Olimex și a metodelor deja expuse, frontul de lucru are o deschidere de o amplitudine care ar permite

îmbunătățirea proceselor de achiziție și creșterea acurateții recunoașterii mișcărilor. De exemplu, ar putea fi cercetate metode de analiză și preprocesare mai complexe, rafinarea acestora ducând la o creștere a competitivității soluțiilor.

Metodele de recunoaștere prin modele statistice sunt deja mai ieftine și eficiente decât rafinarea de natura hardware în zona de colecție neintruzivă a semnalelor electrice.

Pentru a îmbunătăți calitatea semnalelor, un test de impedanță ar putea fi integrat în rutina unei măsurători. De asemenea, utilizarea electrozilor activi ar putea crește calitatea semnalelor EMG măsurate.

BIBLIOGRAFIE

- [1] Adnan, A.A., Albarahany, L.Q., “EMG Signals Detection Technique in Voluntary Muscle Movement”
- [2] Ismail, S., Nur, H.B., Bun, S.C., Mohd. Zuhir H., Bolong, N., “Electromyogram (EMG) Signal Processing Analysis for Clinical Rehabilitation Application”
- [3] <https://medium.com/@leex5202/an-unofficial-introductory-tutorial-to-myoware-muscle-sensor-development-kit-e2169948e63> - accesat la data de 2.05.2020
- [4] Daniel Alte, “Control of a robotic arm using a low-cost BCI”
- [5] Ke, X., Weichao, G., Lei, H., Xinjun, S., Xiangyang, Z., “A Prosthetic Arm Based on EMG Pattern Recognition”
- [6] Narjes, M., Mostafa, K., Khaled, C., Milad Ghantous, Mohamad, H.H., “Pattern Recognition of EMG Signals: Towards Adaptive Control of Robotic Arms”
- [7] Nurhazimah, N., Mohd A.A.R. “Electromyography (EMG) based Signal Analysis for Physiological Device Application in Lower Limb Rehabilitation”
- [8] <https://docs.openbci.com/docs/Welcome.html> - accesat la data de 3.05.2020
- [9] <https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf> - accesat la data de 30.05.2020

- [10] <https://www.getbodysmart.com/wrist-hand-digits/flexor-digitorum-superficialis> accesat la data de 31.05.2020
- [11] Elena Tatiana Avramescu, “Bazele anatomice ale mișcării”
- [12] <https://store.arduino.cc/arduino-uno-rev3> - accesat la data de 31.05.2020
- [13] <https://developers.googleblog.com/2017/09/introducing-tensorflow-datasets.html> accesat la data de 05.06.2020
- [14] <https://www.tensorflow.org/guide/keras/overview> accesat la data de 06.06.2020
- [15] https://keras.io/guides/functional_api/ accesat la data de 06.06.2020
- [16] <https://opensource.com/article/17/11/intro-tensorflow> accesat la data de 10.06.2020
- [17] <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/open-source-hardware> accesat la data de 10.06.2020
- [18] https://www.tensorflow.org/tutorials/customization/custom_training_walkthrough accesat la data de 12.06.2020
- [19] <https://elitedatascience.com/beginner-kaggle> accesat la data de 12.06.2020
- [20] <https://keras.io/api/layers/activations/> accesat la data de 12.06.2020

ANEXA 1

```
import tensorflow as tf
from keras.layers import *
from keras.models import *
from keras.utils import *
from keras.callbacks import *
import numpy as np
import matplotlib.pyplot as plt

def get_accuracy_from_confusion_matrix(confusion_matrix):
    acc = 0.0
    for k in range(confusion_matrix.shape[0]):
        acc += confusion_matrix[k][k]
    return 100.0 * acc / np.sum(confusion_matrix)

def create_confusion_matrix(label_test, label_predict):
    label_test = np.asarray(label_test, dtype=int)
```

```
label_predict = np.asarray(label_predict, dtype=int)
nr_l = int(label_test.max() + 1)
nr_t = label_test.shape[0]
confusion_matrix = np.zeros((nr_l, nr_l), dtype=np.int32)
for i in range(nr_t):
    confusion_matrix[label_test[i]][label_predict[i]] += 1
return confusion_matrix

if __name__ == '__main__':
    training = np.loadtxt('../input/setdate/training_data.txt',
delimiter=" ")
    x_train = training[:, 1:]
    y_train = to_categorical(training[:, 0], 6)
    label_train = np.argmax(y_train, axis=1)
    test = np.loadtxt('../input/setdate/test_data.txt', delimiter=" ")
    x_test = test[:, 1:]
    y_test = to_categorical(test[:, 0], 6)
    label_test = np.argmax(y_test, axis=1)
    input = Input((4,), name='input')
    hidden = Dense(64, activation='relu', name='hidden_1')(input)
    dropout = Dropout(0.2, name='dropout_1')(hidden)
    hidden = Dense(64, activation='relu', name='hidden_2')(dropout)
    dropout = Dropout(0.2, name='dropout_2')(hidden)
    hidden = Dense(32, activation='relu', name='hidden_3')(dropout)
    dropout = Dropout(0.2, name='dropout_3')(hidden)
    hidden = Dense(32, activation='relu', name='hidden_4')(dropout)
    dropout = Dropout(0.2, name='dropout_4')(hidden)
    output = Dense(6, activation='softmax', name='output')(dropout)
    model = Model(input=[input], output=[output])
    model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
    model = load_model('model.h5')
    accuracy = model.fit(x_train, y_train,
callbacks=[ModelCheckpoint('model.h5', save_best_only=True, verbose=1)],
batch_size=64, epochs=100, verbose=2, validation_data=(x_test, y_test))
```

```
print(accuracy)
model.save('model.h5')
y_predict = model.predict(x_test)
label_predict = np.argmax(y_predict, axis=1)
confusion_matrix = create_confusion_matrix(label_test, label_predict)
print(confusion_matrix, 'confusion matrix')
acc = get_accuracy_from_confusion_matrix(confusion_matrix)
print(acc, 'acc')

#tf.keras.utils.plot_model(model, "model.png")
#tf.keras.utils.plot_model(model, "model.png", show_shapes=True)
#model.summary()
#history_dict = history.history
#print(history_dict.keys())

#plt.plot(accuracy.history['accuracy'])
#plt.plot(accuracy.history['val_accuracy'])
#plt.title('Model accuracy')
#plt.ylabel('Accuracy')
#plt.xlabel('Epoch')
#plt.legend(['Train', 'Test'], loc='upper left')
#plt.show()

#plt.plot(accuracy.history['loss'])
#plt.plot(accuracy.history['val_loss'])
#plt.title('Model loss')
#plt.ylabel('Loss')
#plt.xlabel('Epoch')
#plt.legend(['Train', 'Test'], loc='upper left')
#plt.show()
```


ANEXA 2

```
// Senzor MyoWare

#include <Servo.h>    // Utilizare librerie Servo
Servo MyoWare;       // Denumirea servo 'MyoWare'
int EMGsig;          // Stocare valoare semnal EMG
int servoPos;        // Valoare (unghiulara) pentru servo
int prag = 300;      // Prag semnal EMG signal de la care incepe
miscarea. Intervalul orerational este 0-1023.

void setup() {
    Serial.begin(9600); // Startare comunicatie seriala
    MyoWare.attach(9);  // Info rutina servo cu privire ca pin-ul utilizat
este 9.
}

void loop() {
    EMGsig = analogRead(A0); // Citirea valorii analogice a semnalului
prelucrat de MyoWawe pe intrarea A0.
    if (EMGsig < prag){      // If semnal EMG signal sub prag
        servoPos = 88;      // Servo ramane deschis la 88 grade.
    }
}
```

```
} else {          // If semnal EMG este peste prag,
  servoPos=map(EMGsig,prag,1023,88,128);
  // Unghiul servo va fi mapat cu EMG signal,
  // apare miscare de la 300(pragul de pornire)-1023 in intervalul
  unghiular 88-128 grade (dimensiuni alese pentru o miscare cu amplitudine
  relevanta a falcilor clestelui),
  // 88 si 128 pot fi modificate dar cu grija sa nu duca la griparea
  mecanica a angrenajului.
}

MyoWare.write(servoPos); // Misca servo in pozitia 'servoPos' degree
Serial.print(servoPos);
Serial.print(" grade, cu EMG: ");
Serial.println(EMGsig); // Display servo si valoarea EMGsig.
delay(1000); // 1 secunda (1000ms) intarziere care sa evite miscari
haotice. Se poate ajusta experiential.
}
```

ANEXA 3

```
// EMG utilizand Olimex EKG/EMG Shield

const int analogInPin = A0;
unsigned long LoopTimer = 0;

int sensorValue = 0;

const int LoopTime = 10000;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    if (micros() > LoopTimer)
```

```
{  
  LoopTimer += LoopTime;  
  sensorValue = analogRead(analogInPin);  
  sensorValue = sensorValue >> 2;  
  Serial.write(sensorValue);  
  Serial.flush();  
}  
}
```