

Sveučilište u Rijeci

TEHNIČKI FAKULTET

Sveučilišni diplomski studij elektrotehnike

Sveučilišni diplomski studij strojarstva



Komponente mehatroničkih sustava

Projekt

Određivanje napetosti mišića ljudske ruke

Studenti:

Valentina Canjuga, 0069072036

Daniel Đuranović, 0069077256

Ana Markota, 0069073332

Rijeka, rujan 2021.

Sadržaj

1. Uvod.....	1
2. Elektromiografija (EMG)	3
2.1. Nastajanje EMG signala	4
2.2. Akcijski potencijal.....	5
2.3. Detekcija i mjerenje EMG signala	6
2.4. Čimbenici koji utječu na EMG signal	8
2.6. EMG Pojačala	9
2.7. A/D Pretvorba	11
2.7.1. Razlučivost.....	11
2.7.2. Frekvencija uzorkovanja.....	11
3. Shimmer3 senzorski sustav.....	13
3.1. Postupci pripreme kože.....	15
3.2. Elektrode.....	15
4. Predobrada EMG signala pomoću MatLab-a u svrhu daljnje analize.....	18
4.1. Kompenzacija pomaka od osnovne linije i spektar snage	18
4.2. Filtriranje.....	22
4.3. Punovalno ispravljanje.....	25
4.4. Izgladivanje punovalno ispravljenog signala.....	26
4.4.1. Algoritam usrednjavanja.....	27
4.4.2. Algoritam srednje kvadratne vrijednosti.....	27
4.4.3. Utjecaj odabira širine prozora na fazno kašnjenje.....	28
4.5. MVC normalizacija i koeficijent.....	29

4.5.1. Utjecaj odabira algoritma zaglađivanja i širine prozora na MVC koeficijent	31
5. Procjena napetosti mišića u realnom vremenu	33
5.1. MVC_coeff_evaluation.m	33
5.2. MVCMeasure.m	35
5.3. RealTimeEMG.m	35
5.3. Postupak testiranja s utezima	36
5.4. Rezultati mjerenja s utezima.....	37
6. Zaključak	39
7. Popis literature	41
8. Tablica individualnih doprinosa	43

Privitak 1 - Kompenzacija_PI_ispravljanje.m skripta

Privitak 2 - MVCMeasure.m skripta

Privitak 3 - MVC_coeff_evaluation. m skripta

Privitak 4 - RealTimeEMG.m skripta

Privitak 5 - Upute

1. Uvod

Cilj ovog projekta je određivanje napetosti mišića mjerenjem površinskih EMG signala pomoću *Shimmer3* senzorskog modula. Mjereni podaci koji su dobiveni pomoću *Shimmer3* senzorskog modula, nakon sakupljanja na senzoru, prebačeni su u MatLab kako bi se podaci mogli dalje obrađivati.

Mjerenja su izvedena na rukama, gdje se i postavljaju elektrode na točno određena mjesta kako bi se dobili čim bolji podaci o radu mišića. Mjerenja su izvedena na kolegama i kolegicama iz dviju grupa kako bi se moglo doći do nekih razlika o napetosti mišića zbog individualnih razlika pojedinaca. Na mjerene signale imaju utjecaj spol, težina, snaga itd. Svi ti utjecaji su promatrani na konačnim rezultatima.

Također, mjerenja su izvedena tako da ispitanik kojemu su elektrode spojene na ruci, tom se istom rukom najprije oslanja na poledinu stola. S tim mjerenjem su dobiveni podaci iz kojih se nakon obrade očitala maksimalna amplituda koja se kasnije koristila kao točka za usporedbu s ostalim mjerenjima. Kasnije, kako bi se napravilo više usporedbi dodane su razne težine utega, čiji su se podaci zatim uspoređivali sa početno dobivenom amplitudom.

U drugom poglavlju započinje cijela priča oko EMG signala. Objašnjeno je što je EMG signal, kako nastaje, gdje mu je izvor i što utječe na samo očitavanje signala. Zatim, spomenuta su EMG pojačala i njihov utjecaj na otklanjanje smetnji kako bi se došlo do pouzdanijeg i točnijeg signala.

U trećem poglavlju opisan je *Shimmer3* senzorski modul, uz opis elektroda i pripremu kože u svrhu boljeg očitavanja signala

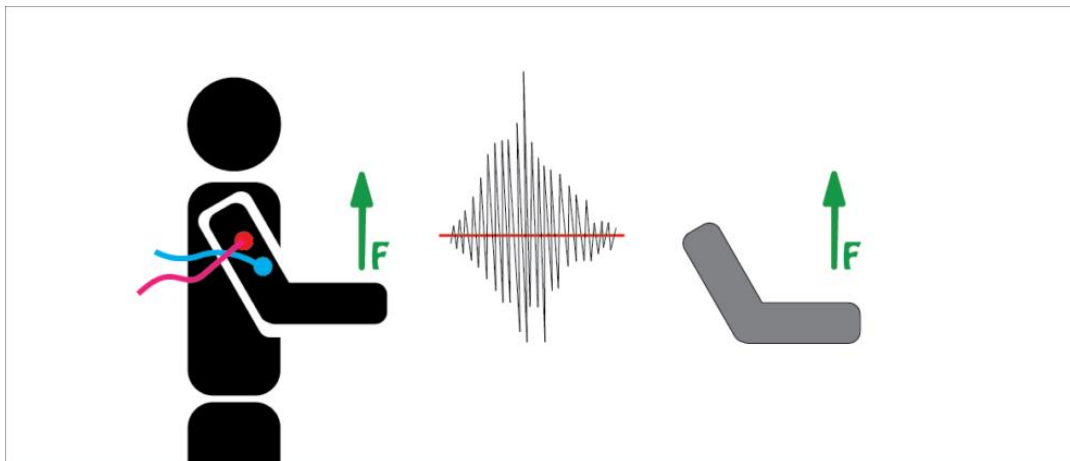
U četvrtom poglavlju započelo se sa opisivanjem očitanih i kasnije obrađivanih podataka, te i sam tijekom obrade očitanih podataka iz *Shimmer3* modula. Svaki signal koji se očita pomoću senzorskog modula sprema se kao neobrađeni („sirovi“) podatak i kao takav se ne može koristiti dalje za neku analizu. Na neobrađenim podacima se nakon sakupljanja izvrši predobrada. Svaki dio obrade signala se detaljno opisao od sakupljanja samih signala, njihovog ispravljanja do normalizacije obrađenog signala.

U petom poglavlju opisana su i prikazana prethodno navedena mjerenja na kolegama. Opisane su implementacije podataka u *MatLabu*, postupci isto koji su opisani u prethodnim poglavljima se ovdje na praktičan način i iskorištavaju. Zatim, prikazani su rezultati mjerenja sa dobivenim grafovima i tablicama za bolje shvaćanje cijelog mjerenja.

U zaključku su predstavljeni dobiveni rezultati cjelokupnog projekta kao i sva mjerenja, te je predloženo idejno rješenje kako ovaj sustav implementirati kao povratnu vezu.

2. Elektromiografija (EMG)

„EMG (elektromiografija) je dijagnostička metoda za ispitivanje funkcije perifernog živčanog sustava (perifernih živaca i odgovora mišića) u stanju mirovanja i pokreta. Njome se može brzo potvrditi ili postaviti dijagnoza bolesti ili povrede perifernog živčanog sustava, disfunkcija mišića ili problemi s prijenosom signala od živca do mišića.“ [1]. Signali od živaca do mišića nazivaju se mioelektrični signali te oni nastaju fiziološkim promjenama u stanju membrane mišićnih vlakana. Motorni neuroni prenose mioelektrične signale koji uzrokuju mišićnu kontrakciju . EMG se izvodi pomoću posebnog uređaja koji se zove elektromiograf, te različitih vrsta elektroda (površinskih i igličnih elektroda) koje registriraju i prenose neuromišićne potencijale u uređaj. EMG prevodi te signale u grafove (numeričke vrijednosti), prikazane potencijale promatra na ekranu, istodobno analizirajući i njihov karakter. Slika 2.1. prikazuje shemu povezivanja elektroda na ruci čovjeka. Kretanjem ruke elektrode evidentiraju potencijale koji se javljaju u mišićima ruke. Ti signali se spremaju na određene senzorske module kako bi ih se kasnije obrađivalo i koristilo za analize.

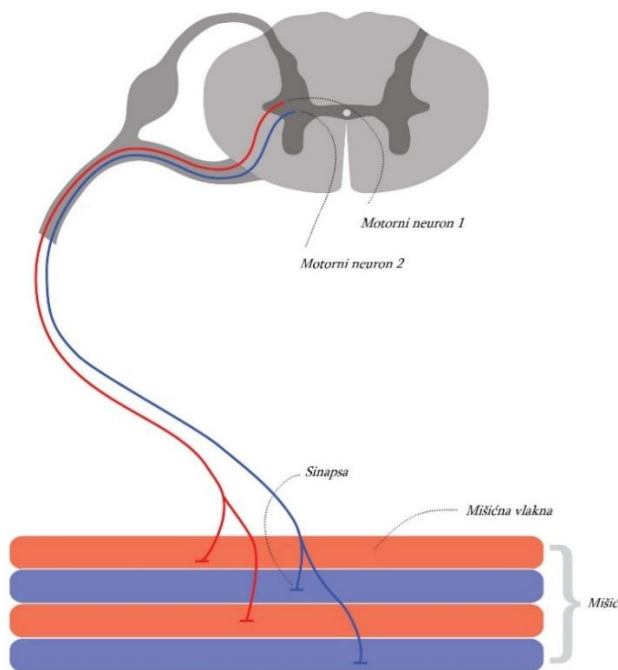


Slika 2.1.: Kontrakcija mišića i izlazni impuls [2]

No prije svega, učinkovita i smisljena upotreba takve opreme zahtjeva, kao i uvijek, poznavanje osnovne teorije i njeno razumijevanje. Stoga u sljedećem dijelu biti će opisani neki od osnovnih pojmova, nastanka mioelektričnih signala i princip njihova mjerenja.

2.1. Nastajanje EMG signala

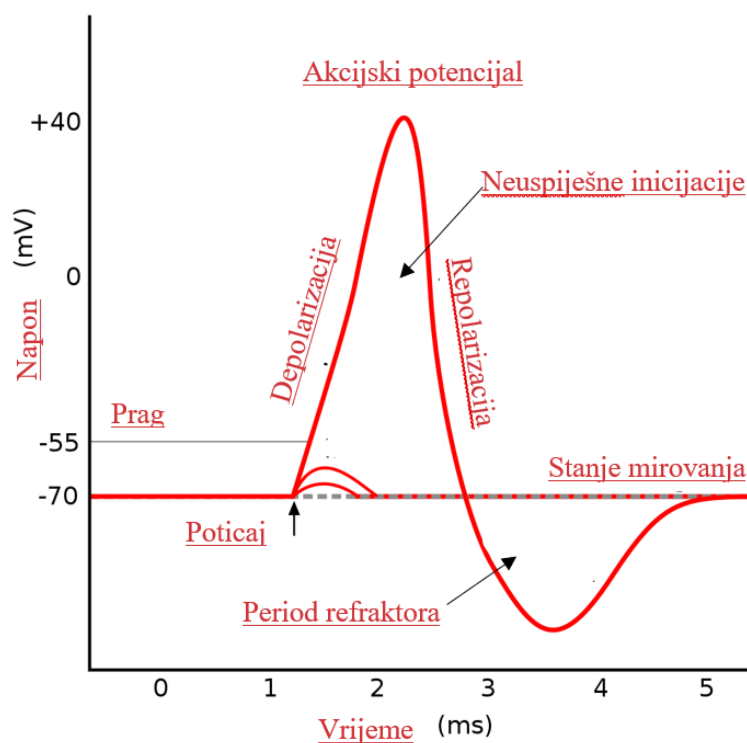
Da bi mišić proizveo silu mišićnim vlaknima je potreban impuls, koji dolazi od motornog neurona. Motorni neuroni su neuroni locirani u centralnom živčanom sustavu čiji aksoni izlaze iz središnjeg živčanog sustav i direktno ili indirektno kontroliraju mišiće. On je ujedno i najmanja funkcionalna jedinica koja opisuje kontrolu procesa mišićnog stezanja. Centralni živčani sustav aktivira motorni neuron i električni impuls putuje niz motorni neuron do svake sinapse. Sinapsa je neuromišićna veza između motornog neurona i mišićnih vlakana koja motornom neuronu omogućava prijenos signala do mišićnih vlakana. Nakon što signal dođe preko sinapse na mišićna vlakna dolazi do stezanja mišića. Cijeli proces prikazan je na Slici 2.2. Čak i u stanju mirovanja mišićno tkivo proizvodi električne signale. Unutrašnjost mišićnog vlakna ima membranski potencijal mirovanja od -70 do -90 mV [3]. Ovaj potencijal varira u zavisnosti o prisutnosti različitih koncentracija iona natrija , kalija i klora.



Slika 2.2.: Prikaz veze između centralnog živčanog sustava i mišića [4]

2.2. Akcijski potencijal

Podražaj neurona dovodi do promjena na membrani koja je propusna za određene ione. Ioni su čestice koje imaju električni naboj, a taj naboj nastaje zbog razlike u broju protona i elektrona. Prolaskom iona kroz mišićnu membranu nastaje akcijski potencijal. Akcijski potencijal brza je promjena membranskoga potencijala, nakon koje slijedi povratak na membranski potencijal mirovanja (Slika 2.3.). Ciklus depolarizacije - repolarizacije tvori električni dipol (dva naboja suprotnih predznaka) koji putuje uz površinu mišićnog vlakna. Akcijski se potencijal širi bez slabljenja cijelom duljinom mišićnoga vlakna u oba smjera. Dok putuje duž mišićnoga vlakna, veličina i oblik akcijskoga potencijala ostaju jednaki, tj. veličina se ne smanjuje razmjerno s udaljenošću. [3]

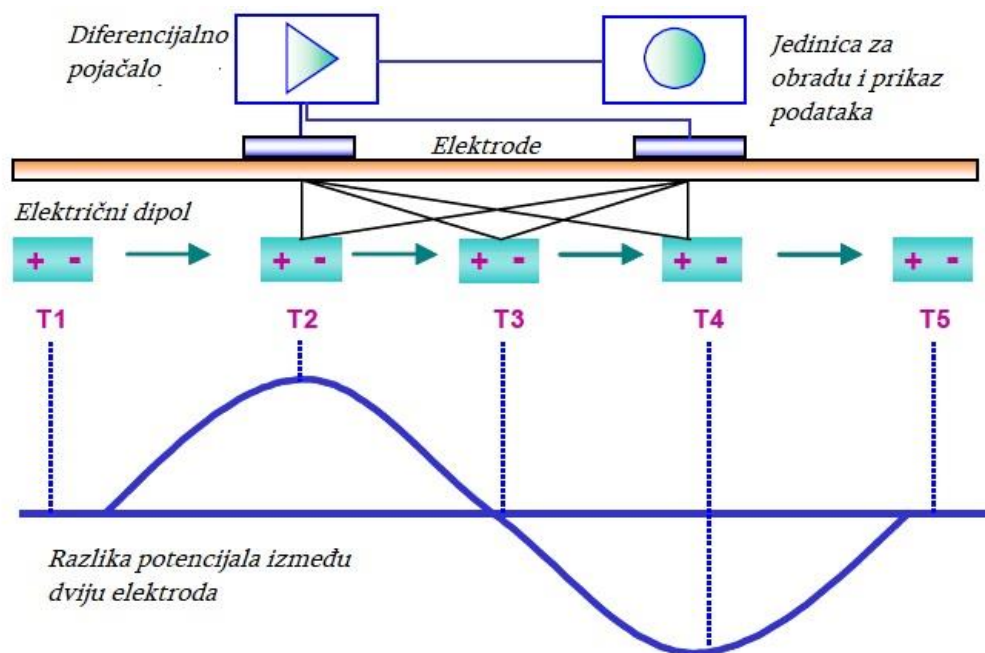


Slika 2.3.: Akcijski potencijal [3]

EMG signal temelji se na akcijskim potencijalima koji se javljaju na membrani mišićnih vlakana kao što je gore opisano. Nakon početnog uzbuđenja električni dipol putuje duž mišićnog vlakna brzinom od 2-6 m/s. [3]

2.3. Detekcija i mjerenje EMG signala

Za mjerenje EMG signala tipično se koriste konfiguracije dviju (bipolarnih) elektroda i diferencijalno pojačanje (najčešće instrumentacijsko pojačalo). Radi jednostavnosti prvo je objašnjeno detektiranje EMG signala na jednom mišićnom vlaknu čiji je princip prikazan na Slici 2.4. Ovisno o prostornoj udaljenosti između elektroda 1 i 2 dipol tvori razliku potencijala između elektroda. kaj tvori promjenu??

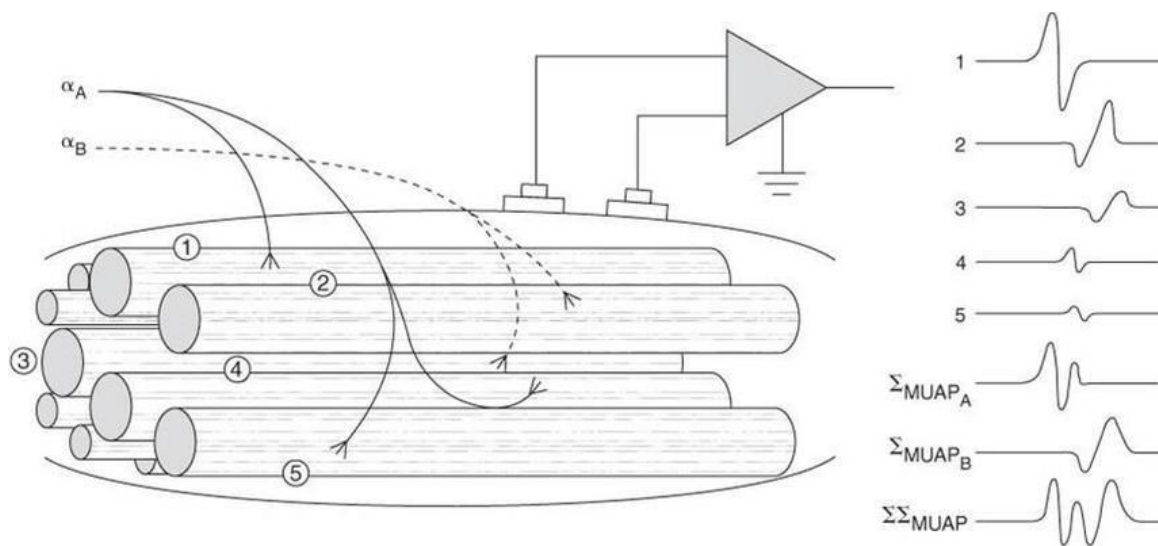


Slika 2.4.: Primjer signala kroz jedno mišićno vlakno [3]

U trenutku T1 generiran je akcijski potencijal (el. dipol) i putuje kroz mišićno vlakno prema paru elektroda. Kako se dipol približava prema paru elektroda mjeri se rastuća razlika potencijala između elektroda koja je najveća u trenutku T2, što znači najkraću udaljenost do elektrode 1. Kada

dipol dođe na jednaku udaljenost između elektroda razlika potencijala je nula (trenutak T3). Daljnjim gibanjem razlika potencijala prelazi nulu i najveća je opet u trenutku T4 (najkraća udaljenost do elektrode 2), ali je sada negativnog predznaka. Iz ovoga se može primijetiti zašto monopolarni akcijski potencijal stvara bipolarni signal unutar diferencijalnog pojačanja. Odnosno konačno promotri li se na apcisi sa Slike 2.4. razlika potencijala između dviju elektroda, uočava se da je to izmjenični odnosno bipolarni signal. [3]

No, u stvarnosti se unutar mišića nalazi mnogo mišićnih vlakana tj. svaki motorni neuron podražuje nekoliko stotina mišićnih vlakana (premda taj broj varira u zavisnosti od vrste mišića kroz koje taj signal prolazi), to znači da ovaj par elektroda detektira sve signale iz svih vlakana ovisno o njihovoj prostornoj udaljenosti i razlučivosti. Akcijski potencijal motorne jedinice (engl. “*Motor unit action potential*” - MUAP), predstavlja sumu električnih aktivnosti svih mišićnih vlakana unutar te motorne jedinice (Slika 2.5.). Dakle, motorne jedinice s više mišićnih vlakana imaju veći MUAP.

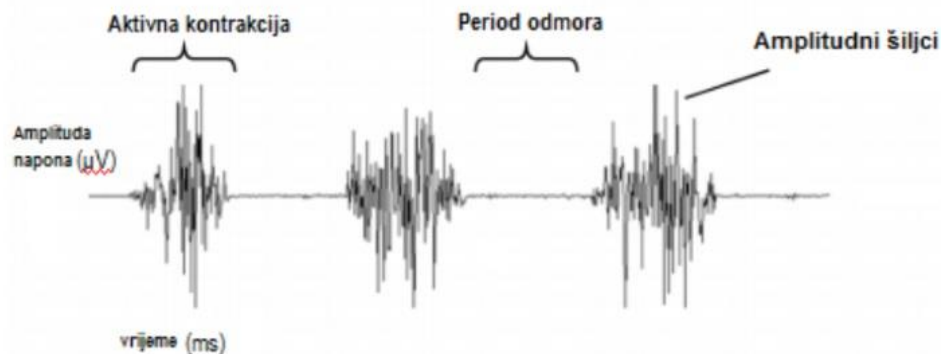


© 2010 Human Kinetics. Reprinted by permission from G. Kamen, 2004.

Slika 2.5.: MUAP signal [5]

Iz Slike 2.5. se može primijetiti da mišićno vlakno koje se nalazi dalje od elektroda (npr. vlakno 5) daje manji doprinos ukupnoj sumi nego recimo vlakno koje je bliže elektrodama (npr. vlakno 1 ili 2).

Zapravo, signali pojedinačnih mišićnih vlakana se ne razlikuju samo po amplitudi, nego se razlikuju već i po frekvenciji i vremenu okidanja te to utječe na finalni oblik superponiranog MUAP-a kojeg elektrode “vide” (Slika 2.5.). No kada takav signal mjerimo u realnosti na njega djeluju različiti čimbenici (npr. šum) te takav nefiltrirani i neobrađeni signal se naziva “sirovim” EMG signalom (engl. “*raw EMG*”) koji je prikazan na Slici 2.6.



Slika 2.6: Primjer realnog EMG signala [3]

2.4. Čimbenici koji utječu na EMG signal

Na putu od mišićne membrane do elektroda, na EMG signal može utjecati nekoliko vanjskih čimbenika koji mijenjaju njegov oblik i karakteristike. Ti čimbenici su:

- *Karakteristike tkiva* - Ljudsko tijelo je dobar električni vodič, ali nažalost električna vodljivost varira ovisno o vrsti tkiva, debljini, fiziološkim promjenama i temperaturi.
- *Fiziološki „Cross Talk”* - Susjedni mišići mogu proizvesti značajnu količinu EMG-a koju detektira lokalno mjesto elektrode. Mora se voditi računa o uskim rasporedima unutar mišićnih skupina. Primjer su EKG skokovi koji mogu ometati EMG snimanje, posebno kada se izvodi na gornjim mišićima trupa i ramena.
- *Promjene u geometriji između mišića i elektrode* - Svaka promjena udaljenosti između izvora signala i mjesta detekcije mijenja očitavanje EMG-a. To je svojstveni problem svake dinamike pokreta, a mogu biti uzrokovane i vanjskim pritiskom.

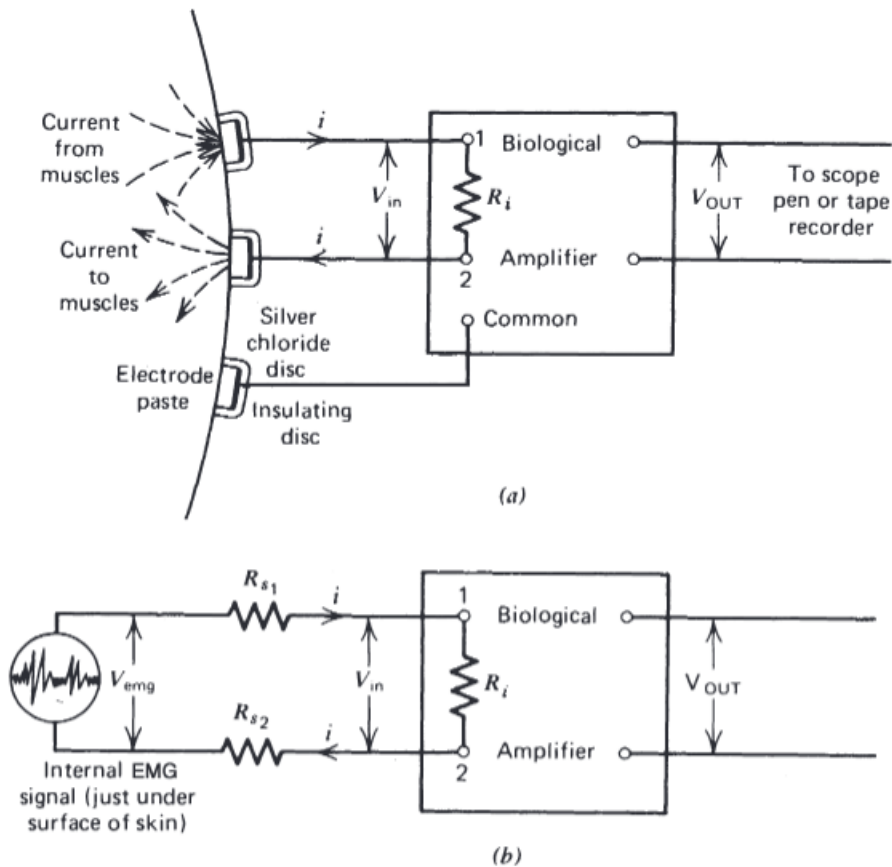
- *Vanjska buka (smetnje)* -Vrlo „bučna“ električna okruženja koja djeluju na izravne smetnje, a nastala su nepravilnim uzemljenjem drugih vanjskih uređaja.
- *Elektrode i pojačala* -Odabir kvalitetnih elektroda i pojačala pojačala mogu utjecati na sadržaj signala EMG.

2.6. EMG Pojačala

EMG pojačala djeluju kao diferencijalna pojačala (najčešće instrumentacijska) i njihova je glavna svrha sposobnost odbacivanja ili uklanjanja smetnji. Diferencijalno pojačanje otkriva potencijalne razlike između elektroda te poništava istofaznu smetnju koja sadrži najviše smetnji, te tako poboljšava pouzdanost i točnost. Jedan od kriterija za odabir kvalitetnog pojačala je faktor potiskivanja istofazne smetnje CMRR (engl. „*Common Mode Rejection Ratio*“). CMRR predstavlja odnos između protufaznog i istofaznog pojačanja te bi njegov iznos trebao biti što je moguće veći ($> 95\text{dB}$). Najsuvremeniji koncepti preferiraju upotrebu EMG predpojačala. Ova minijturna pojačala obično su ugrađena u kabele ili su smješteni na vrhu elektroda (aktivne elektrode). [3]

Ulazna impedancija pojačala trebala bi biti dovoljno velika kako ne bi atenuirala (prigušila) EMG signal koji je spojen na ulazne stezaljke pojačala. Ako se razmotri pojačalo prikazano na Slici 2.7.a), aktivni ulazni priključci su 1 i 2 s referentnim (*common*) priključkom c. Svaki spoj elektroda-koža ima konačnu impedanciju koja ovisi o mnogim čimbenicima kao što su: debljina sloja kože, čišćenje kože prilikom pripreme, površina elektrode i dr. . Na Slici 2.7.b), spoj elektroda-koža zamijenjen je aproksimiranim otporima R_{S1} i R_{S2} . Ovo je pojednostavljen prikaz stvarnoga stanja, mnogo točnija reprezentacija bila bi da se uključi i kapacitivni efekt između elektrode i kože. Čim je diferencijalno pojačalo spojeno na elektrode, EMG signal uzrokuje protok struje kroz otpore R_{S1} i R_{S2} , te ulaznu impedanciju pojačala R_i . Protok struje kroz otpore R_{S1} i R_{S2} uzrokuje pad napona, pa će tako napon V_{in} na ulaznim stezaljkama biti manji nego stvarni V_{EMG} . Na primjer ako su $R_{S1} = R_{S2} = 10\text{ k}\Omega$ i $R_i = 80\text{ k}\Omega$ uz $V_{EMG} = 2\text{ mV}$, napon na ulaznim stezaljkama pojačala V_{in} bio bi 1.6 mV . Ako se boljom pripremom kože smanje vrijednosti otpora R_{S1} i R_{S2} na recimo $1\text{ k}\Omega$ i ulazni otpor R_i pojačala je bliže realnoj vrijednosti kao npr $1\text{ M}\Omega$, vrijednost ulaznog napona V_{in} iznosila bi 1.98 mV , što je značajno manji pad. V_{in} bi bio skoro jednak ako bi vrijedilo

$R_i \gg R_s$, te autor preporučuje da je ulazna impedancija diferencijalnog pojačala barem 10x veća od impedancije elektroda, koja inače iznosi oko $50\text{ k}\Omega$. [6]



Slika 2.7.: a) Protjecanje struje kroz elektrode i pojačalo

b) Pojednostavljeni nadomjesni model [6]

2.7. A/D Pretvorba

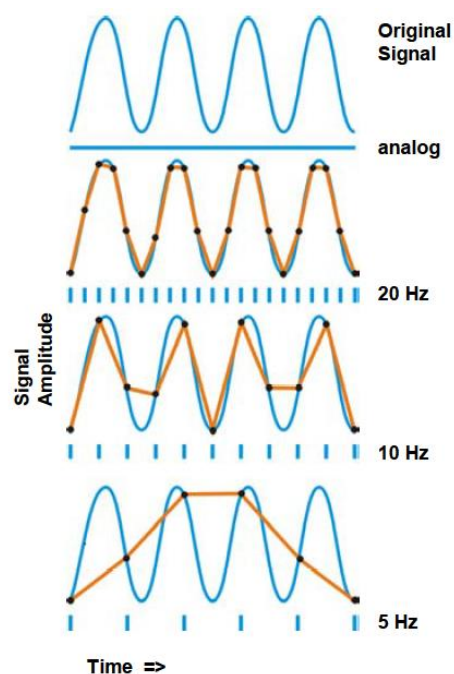
Kako bi se mjereni signal mogao prikazati, te zatim analizirati putem računala, potrebno ga je iz analognog oblika pretvoriti u digitalni. Postupak pretvaranja iz analognog u digitalni oblik naziva se Analogno-Digitalna pretvorba (A/D pretvorba skraćeno). Kod A/D pretvorbe postoje dva bitna parametra nekakvog A/D pretvornika koji se moraju uzeti u obzir, a to su njegova razlučivost i frekvencija uzorkovanja.

2.7.1. Razlučivost

“Razlučivost pretvarača pokazuje broj različitih, tj. Diskretnih vrijednosti koje može proizvesti u dopuštenom rasponu vrijednosti analognih ulaza” [7] Tako npr za 12-bitni A/D pretvornik (kao što u sebi sadrži *Shimmer3* modul [8]), raspon A/D pretvornika je 4096 razina (2^{12}).

2.7.2. Frekvencija uzorkovanja

Frekvencija uzorkovanja izuzetno je bitan parametar kako bi se vjerno reproducirao signal na računalu. Prema teoremu uzorkovanja ili *Nyquistovom* teoremu, frekvencija uzorkovanja A/D pretvornika mora biti najmanje dvostruka veća od maksimalne frekvencije koja se javlja u signalu. Ne zadovolji li se taj uvjet, dolazi do preklapanja spektra (takozvanog *Aliasina*) koji je prikazan na slici 2.8 te se signal ne reproducira dobro .



Slika 2.8.: Pojava Aliasinga kod nezadovoljenog Nyquistovog teorema [3]

U praksi se rijetko kad uzima granična vrijednost frekvencije uzorkovanja (dvostruko veća), uglavnom se uzima frekvencija uzorkovanja koja je 3-6 puta veća od maksimalne frekvencije u signalu. Spektar EMG signala obično se nalazi između 10 i 250 Hz, uz preporuke da je frekvencijski pojas pojačala od 10 do 500 Hz. [3]

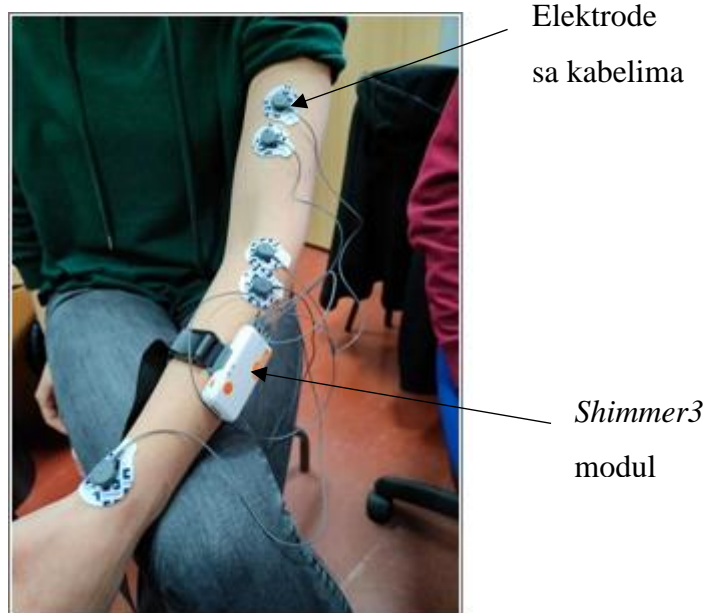
3. Shimmer3 senzorski sustav

Shimmer („*Sensing Health with Intelligence, Modularity, Mobility and Experimental Reusability*“) tvrtka se bavi proizvodnjom kompaktnih senzora koji se mogu jednostavno nositi. *Shimmerova* senzorska tehnologija omogućuje prikaz i prijenos biofizičkih i kinetičkih podataka u stvarnom vremenu koji mogu biti pohranjeni ili prenošeni preko Bluetootha na drugi uređaj (npr. računalo ili pametni telefon). *Shimmerov* senzorski sustav snima i prenosi širok raspon podataka, ovisno o tome što je potrebno za mjerenje i okolinu u kojoj se mjeri.

Shimmer3 modul je mali, lagan i kompaktan uređaj koji sadrži inercijski senzorski sustav uključujući integrirani akcelerometar, žiroskop, magnetometar i visinometar.

Spajanje *Shimmera* zahtijeva nekoliko koraka. Prvo se odredi mišić koji se želi mjeriti, na njemu se odredi područje uzdužnih mišićnih vlakana kako bi izmjerili naponske razlike između dvije mjerne točke. Alkoholom se očisti područje kože gdje se planira postaviti elektroda, radi stvaranja boljeg kontakta. Za jedan mišić se postavljaju tri elektrode, dvije su pozitivan i negativan ulaz a treća predstavlja referentnu točku.

Slika 3.1. prikazuje *Shimmer3* modul spojen na ruci. Preko spojenih elektroda na modul se prenose podaci o kontrakciji mišića koji se dalje mogu obrađivati preko računala. Modul obavlja analogno-digitalnu pretvorbu i obavlja spremanje podataka u stvarnom vremenu. Podaci se mogu „streamati“ preko Bluetootha na računalo ili spremati lokalno na ugrađenu *MicroSD* memorijsku karticu. Također, ima bateriju koja omogućava obavljanje velikog broja mjerenja izvan laboratorijskih uvjeta, daleko od izvora napajanja.



Slika 3.1.: Spojeni Shimmer3 modul

Slika 3.2. prikazuje *Shimmer3* modul u kutiji, sa raznim kabelima za napajanje i s pločom. Preko ploče se na računalu prepoznaje sam modul kako bi se kasnije s njim moglo raditi. Ploča također služi i kao punjač za module.



Slika 3.2.: Shimmer EMG senzor [9]

3.1. Postupci pripreme kože

Postupci pripreme kože neophodni su kako bi se mjesta na koja se lijepe elektrode pripremila za prihvat elektroda.

Neki od koraka pripreme nanošenja elektroda su uklanjanje dlaka i čišćenje kože. Za mjesta na koja se stavljaju elektrode preporuča se da se dlake uklone jer se sa čistom kožom uspostavlja bolji signal. Za čišćenje kože dostupne su razne abrazivne i vodljive paste za čišćenje koje uklanjaju mrtve stanice kože i očiste kožu od prljavštine i znoja. Abrazivna pasta sadrži smjesu supstanci za čišćenje kože, koristi se za efektivno pranje ruku pri radu u industriji, rudarstvu i radionicama.

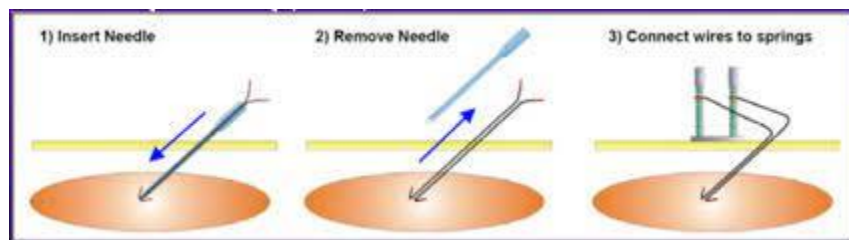
Alternativno se može koristiti brusni papir uz mekan i kontroliran pritisak u tri ili četiri zamaha uz kombinaciju s čišćenjem alkoholom. Čista upotreba alkohola može biti druga alternativa ako se koristi s tekstilnim ručnikom (koji omogućuje meko trljanje). Sve navedene metode uključuju blago trljanje vanjskog sloja kože kako bi se osigurao što bolji kontakt s elektrodama.

Za projekt su korištene dezinfekcijske maramice kako bi se područja na ruci očistila od nečistoća i kako bi se elektrode zatim mogle postaviti na za to predviđena mjesta.

3.2. Elektrode

Elektrode se postavljaju na mjesta na koži gdje su referentne točke kako bi se očitavali određeni i željeni signali. Postoje površinske elektrode i elektrode od tanke žice ili igle.

Zbog svog neinvazivnog karaktera, u većini slučajeva površinske elektrode koriste se u kineziološkim studijama. Osim jednostavnog rukovanja, njihov glavni nedostatak je što se mogu detektirati samo za površinske mišiće. Za dublje mišiće (prekrivene površinskim mišićima ili kostima) neizbježne su elektrode od tanke žice ili igle.



Slika 3.3.: Primjer elektroda od fine žice [3]

Upotreba elektroda od fine žice obično se koristi zbog pokreta mišića. Tanke i fleksibilne elektrode od fine žice česti su izbor za invazivnu primjenu elektroda u dubljim mišićnim slojevima. Signali se mjere i obrađuju poput pravilnih površinskih EMG signala. Uz primjenu visokopropusnog filtra, uklanjaju se osnovni pomaci koji se obično pojavljuju iz artefaktnih kretanja žice u mišićnom tkivu.

Za pravilno rukovanje elektrodama potrebno je izabrati dobro vodljive elektrode. Primjerice, elektrode s mokrim gelom (omogućava dobru vodljivost) imaju najbolje vrijednosti impedancije kože. Manja površina elektroda osigurava bolju selektivnost mjerenja i povećava vrijednost impedancije. Odabirom najbliže moguće udaljenosti između elektroda također se povećava selektivnost. Za najbolju selektivnost, elektrode se postavljaju paralelno sa smjerom mišićnih vlakana i pri tome se koristi dominantni srednji dio. Perporuča se izbjegavati područje motoričkih točaka ako je moguće. Motorne točke su područja s velikom gustoćom završnih motora mišića. Te točke se mogu detektirati generatorima snage podražaja niske frekvencije koji proizvode pravokutne impulse. Pripaziti da mjesto elektrode ostane na aktivnoj mišićnoj masi tijekom kontrakcije mišića.

Treba postaviti najmanje jednu neutralnu referentnu elektrodu, odnosno odabrati električki nepovezani dio tijela. Tipično se bira u blizini područja, poput zglobova, koštanog područja, frontalne glave itd. Postavljanjem referentne elektrode uklanja se tjelesni šum i moguće je lakše čitanje željenog signala.

Slika 3.4. prikazuje površinske elektrode, slične korištenima za projekt. Za njih se obavlja predhodno priprema kože i zatim se lijepe na određena mjesta na tijelu. Slika desno prikazuje kabele koji se s jedne strane spajaju na gornju stranu elektrode, a s druge strane se spajaju na *Shimmer3* modul i tako prenose signale sa tijela na modul.



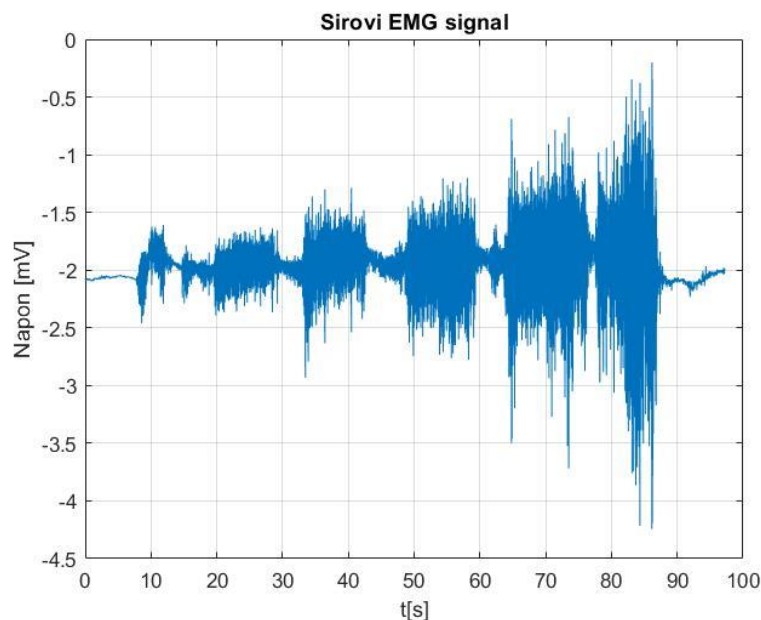
Slika 3.4. Površinske elektrode i kabeli [9]

4. Predobrada EMG signala pomoću MatLab-a u svrhu daljnje analize

Kako bi signal bio spreman za daljnju analizu i procesiranje kojem će biti podvrgnut, potrebno ga je obraditi. U procesu predobrade signal se prilagođava potrebama daljnjeg rada kako u kasnijoj fazi njegove analize ne bi došlo do poteškoća, te su u daljnjem tekstu opisani načini kako je to postignuto putem *MatLab-a*.

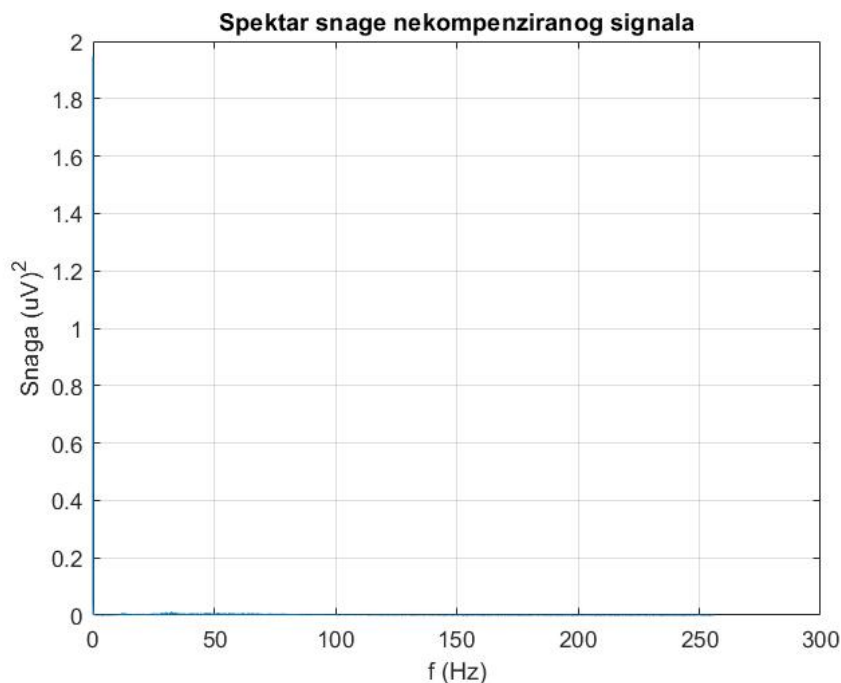
4.1. Kompenzacija pomaka od osnovne linije i spektar snage

Srednja vrijednost EMG signala jednaka je nuli, stoga i srednja vrijednost cjelokupnog spektra EMG signala mora biti jednaka nuli. Uzimajući u obzir kako za osnovnu liniju uzimamo $0 \mu\text{V}$, pomak izmjenjenog signala od osnovne linije prilikom mjerenja ne smije postojati. Ako takav pomak postoji, on se mora kompenzirati tako da se od izmjenjenog signala oduzme njegova srednja vrijednost. U slučaju da se to ne napravi svi zaključci dobiveni iz amplitude signala nisu točni i ne mogu se smatrati vjerodostojnima. Na slici 4.1. prikazan je jedan uzorak sirovog EMG signala u vremenu koji je izmjeren pomoću *Shimmer3* senzora.



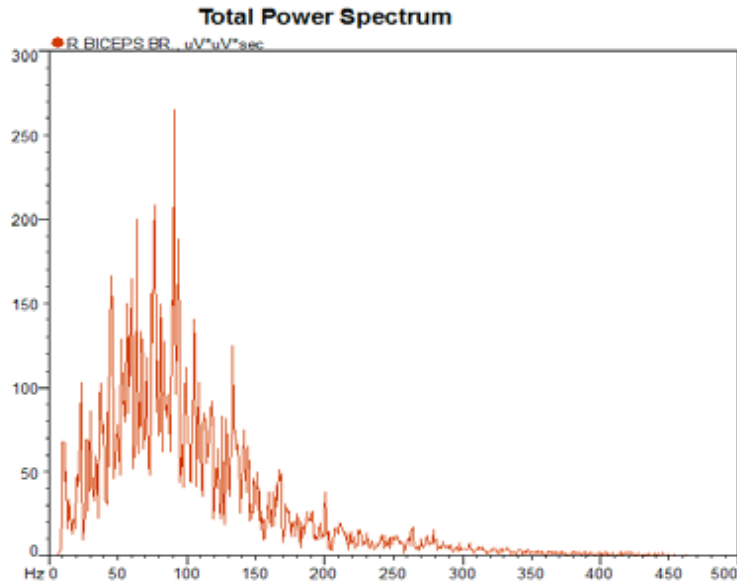
Slika 4.1.: Sirovi EMG signal izmjeren pomoću *Shimmer3* senzora ($f_s = 512 \text{ Hz}$)

Iz slike se vidi da postoji pomak izmjereneog signala u odnosu na nultu liniju ($0 \mu V$), te ga je potrebno kompenzirati. Ukoliko *Fourierovom* transformacijom izračunamo i prikažemo grafički spektar snage takvog nekompenziranog signala, koji je prikazan na Slici 4.2., može se uočiti da je veliki dio snage sadržan u istosmjernoj komponenti signala ($f = 0 \text{ Hz}$). Usporedbnom rezultata sa spektrom iz literature [3] (slika 4.3) dolazi do velikog odstupanja rezultata s literaturom, što dokazuje potrebu kompenzacije pomaka od nulte linije.



Slika 4.2.: Spektar snage nekompenziranog (sirovog signala)

Kao kratka diverzija, reći će se nešto o tome kako bi spektar snage trebao izgledati. Kako je već objašnjeno u poglavlju 2, EMG signal dobiva se mjerenjem akcijskog potencijala na stijenci mišića za svaki MUAP, a ukupni se signal dobije superponiranjem svih MUAP-a od kojih se sastoje sami mišić. Zbog njihovog mnoštva lako je zaključiti da ukupni signal sadrži veliki spektar amplituda i frekvencija. Na Slici 4.3. prikazano je kako bi trebao izgledati spektar snage jednog neobrađenog EMG signala prema literaturi [3].



Slika 4.3.: Spektar snage sirovog EMG signala prema literaturi [3]

Iz Slike 4.3. može se primijetiti da većinu energije s EMG signala sadrže komponente na frekvencijama od 10 do 250 Hz, s tim da najviše energije sadrže komponente na frekvencijama od 30 do 160 Hz. Vršne frekvencijske komponente kreću se u području od 50Hz do 80Hz. Na frekvencijama višim od 80 Hz snaga postepeno opada, da bi na frekvencijama oko 250 Hz poprimila vrijednosti približne nuli. Najveći problem nastaje utjecajem smetnje gradske mreže čiji se osnovni harmonik nalazi na frekvenciji 50 Hz (60 Hz u SAD-u), pa se filtriranjem signala nastoji svesti njihov utjecaj na minimum. [3]

Ako se kod kompenzacije izmjerenog oduzme njegova srednja vrijednost gledajući sveukupni signal, kompenzirani signal rezultirati će nekakvim manjim pomakom od nule te rezultati neće biti dosljedni. U našem slučaju *Shimmer3* imao je automatsku kompenzaciju pomaka od nule. No, ako to rješenje nije dostupno predloženo je slijedeće rješenje u *MatLab-u*. Funkcija za kompenzaciju pomaka „*Kompenzacija_PI_ispravljanje.m*“ od nule nalazi se u Privitku 1, dok je ovdje opisan pojednostavljeni kod (Pseudokod 4.1.). Umjesto da se uzme srednja vrijednost cijelog signala, signal se podijeli na više manjih dijelova određenih širinom prozora N . Sada se kompenzacija odvija više manjih dijelova signala što rezultira točnijom kompenzacijom.

```

kreiraj prazan stupčasti vektor koji predstavlja kompenziran signal

for segment x osi = 1: Širina prozora N: dužina signala

    odredi trenutnu poziciju prozora

    izračunaj srednju vrijednost signala unutar prozora

    korekcija = vrijednost signala unutar prozora - srednja vrijednost signala unutar prozora

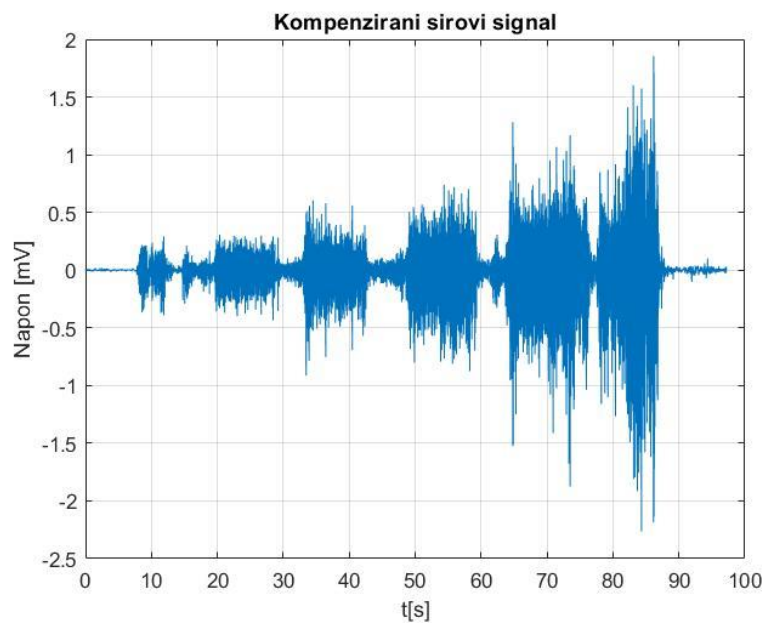
    nadodaj vrijednost signala na vektor

end

```

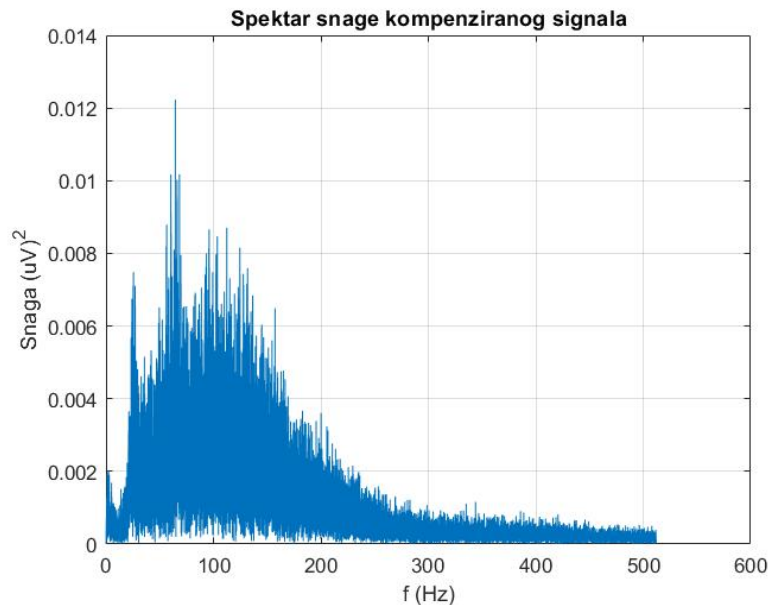
Pseudokod 4.1.: Pojednostavljeni algoritam kompenzacije

Primijenimo li gore opisani pojednostavljeni algoritam na signal sa Slike 4.1., dobije se signal prikazan u ovisnosti o vremenu na Slici 4.4. Promotri li se sada slika uočava se da ne postoji pomak od središnje linije.



Slika 4.4.: Kompenzirani sirovi EMG signal izmjeren pomoći Shimmer3 senzora ($f_s = 512$ Hz)

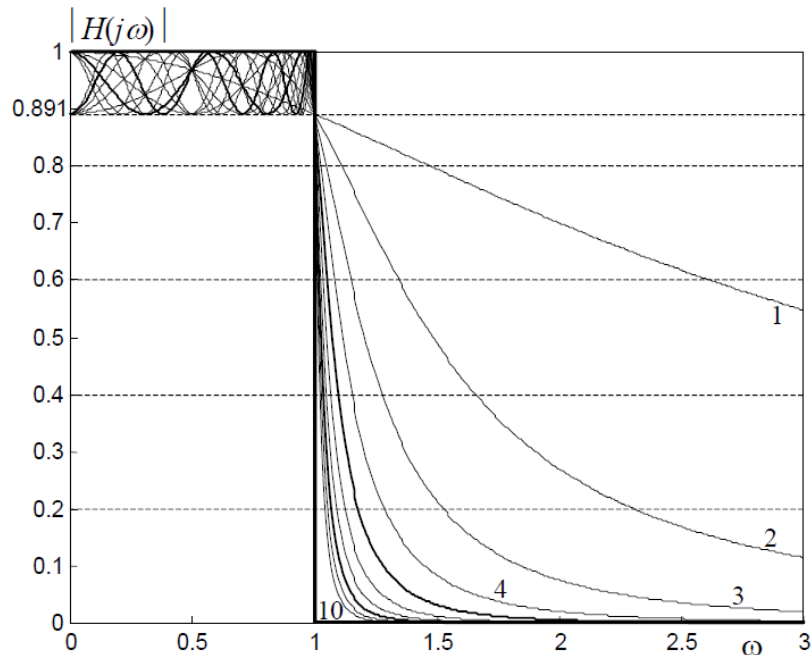
Ako se sada izaćuna i grafički prikaže spektar snage kompenziranog signala (Slika 4.5.), primjećuje se da se sada spektar poklapa s rezultatima spektra danima u literaturi [3] (Slika 4.3.). Time se zaključuje da je algoritam za kompenzaciju uspješno implementiran.



Slika 4.5.: Spektar snage sirovog kompenziranog EMG signala ($f_s = 512$ Hz)

4.2. Filtriranje

Nakon kompenzacije pomaka od nule izmjerenog EMG signala, potrebno je taj isti signal filtrirati kako bi se potisnule neželjene frekvencijske komponente EMG signala. Neželjene frekvencijske komponente bile bi: frekvencija gradske mreže (50 ili 60 Hz), više frekvencijske komponente u kojima se nalazi šum i drugo. *Shimmer3* senzor već u sebi posjeduje implementirani *Chebyshev* filter. [10] *Chebysheva* aproksimacija je aproksimacija prijenosne funkcije filtra s valovitošću u području propuštanja te graf njene normalizirane amplitudno-frekvencijske karakteristike prikazan je na Slici 4.6. [11]



Slika 4.6.: Amplitudno frekvencijska karakteristika Chebyheve aproksimacije [9]

U ovome radu koristio se već definirani filter koji je pružao *Shimmer3* modul. Implementirani filter sastoji se od slijedećih dijelova:

- visokopropusnog (VP) filtra
- niskopropusnog (NP) filtra
- pojasne brane (PB).

Prema preporukama iz [10] filter je pušten u standardnoj konfiguraciji odnosno:

- granična frekvencija VP filtra: $f_g = 5 \text{ Hz}$
- granična frekvencija NP filtra: $f_g = f_s/2$ (f_s predstavlja frekvenciju uzorkovanja)
- centralna frekvencija PB filtra: $f_c = 50 \text{ Hz}$ (uz $f_{dg} = 49 \text{ Hz}$ i $f_{gg} = 51 \text{ Hz}$)

Razlog odabira granične frekvencije NP filtra $f_g = f_s/2$ je taj što ta granična frekvencija uzrokuje ponašanje filtra kao *Antialiasing* filter, te na taj način potiskuje komponente koje su na frekvencijama više od *Nyquistove* frekvencije. Definicija parametra filtra u *MatLab-u* prikazana je u MatLab 4.1. kodu.

```

% filtering settings
fm = 50; % mains frequency [Hz]
fchp = 5; % corner frequency highpassfilter [Hz]; Shimmer recommends 5Hz to
remove DC-offset and movement artifacts
nPoles = 4; % number of poles (HPF, LPF)
pbRipple = 0.5; % pass band ripple (%)

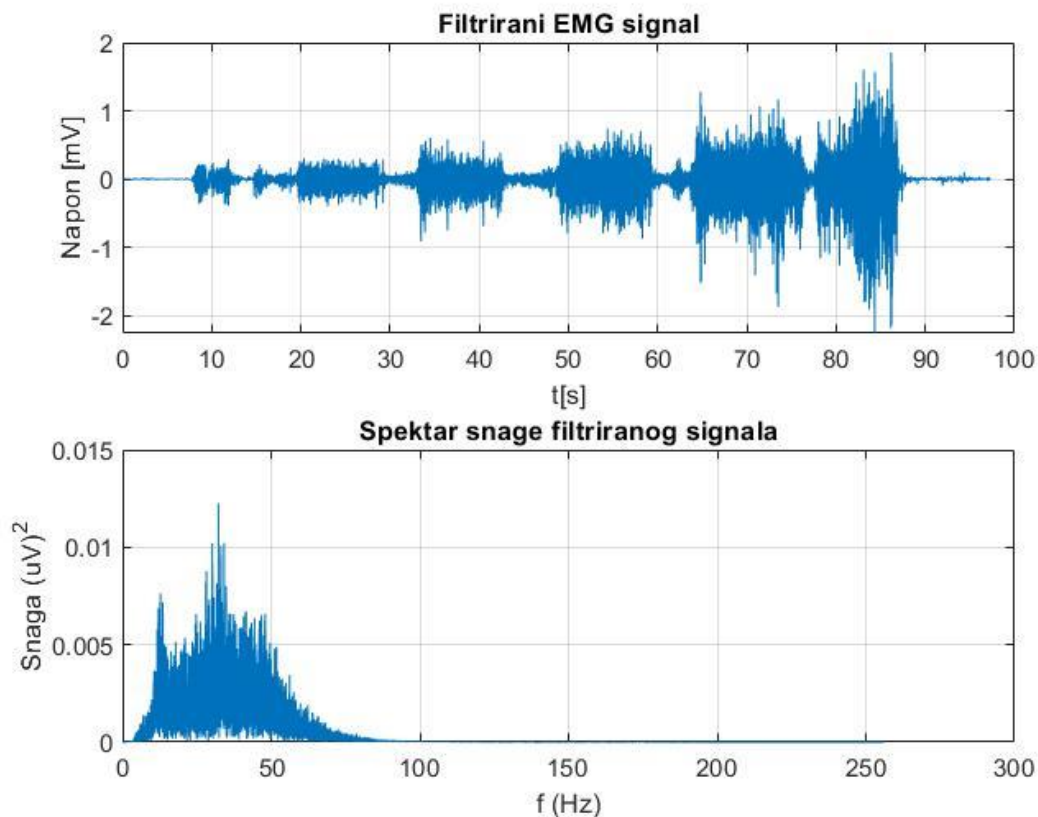
HPF = true; % enable (true) or disable (false) highpass filter
LPF = true; % enable (true) or disable (false) lowpass filter
BSF = true; % enable (true) or disable (false) bandstop filter

% highpass filters for ExG channels
if (HPF)
    hpfxg1ch1 = FilterClass(FilterClass.HPF, fs, fchp, nPoles, pbRipple);
    hpfxg1ch2 = FilterClass(FilterClass.HPF, fs, fchp, nPoles, pbRipple);
end
if (LPF)
    % lowpass filters for ExG channels
    lpfexg1ch1 = FilterClass(FilterClass.LPF, fs, fs/2-1, nPoles, pbRipple);
    lpfexg1ch2 = FilterClass(FilterClass.LPF, fs, fs/2-1, nPoles, pbRipple);
end
if (BSF)
    % bandstop filters for ExG channels;
    % cornerfrequencies at +1Hz and -1Hz from mains frequency
    bsfxg1ch1 = FilterClass(FilterClass.LPF, fs, [fm-1, fm+1], nPoles, pbRipple);
    bsfxg1ch2 = FilterClass(FilterClass.LPF, fs, [fm-1, fm+1], nPoles, pbRipple);
end

```

MatLab kod 4.1.: Definicija parametra filtra

Filtrira li se kompenzirani sneobrađeni EMG signal sa slike te izračuna njegov spektar snage dobiju se grafički prikazi sa Slike 4.8. Usporedbom spektra snage filtriranog signala sa spektrom snage nefiltriranog signala sa Slike 4.5., očito je kako su potisnute frekvencijske komponente ispod frekvencije od 5 Hz, više frekvencijske komponente koje su u sebi sadržavale šum i frekvencija gradske mreže od 50 Hz. Dok frekvencije u pojasnom propustu nisu prigušene.



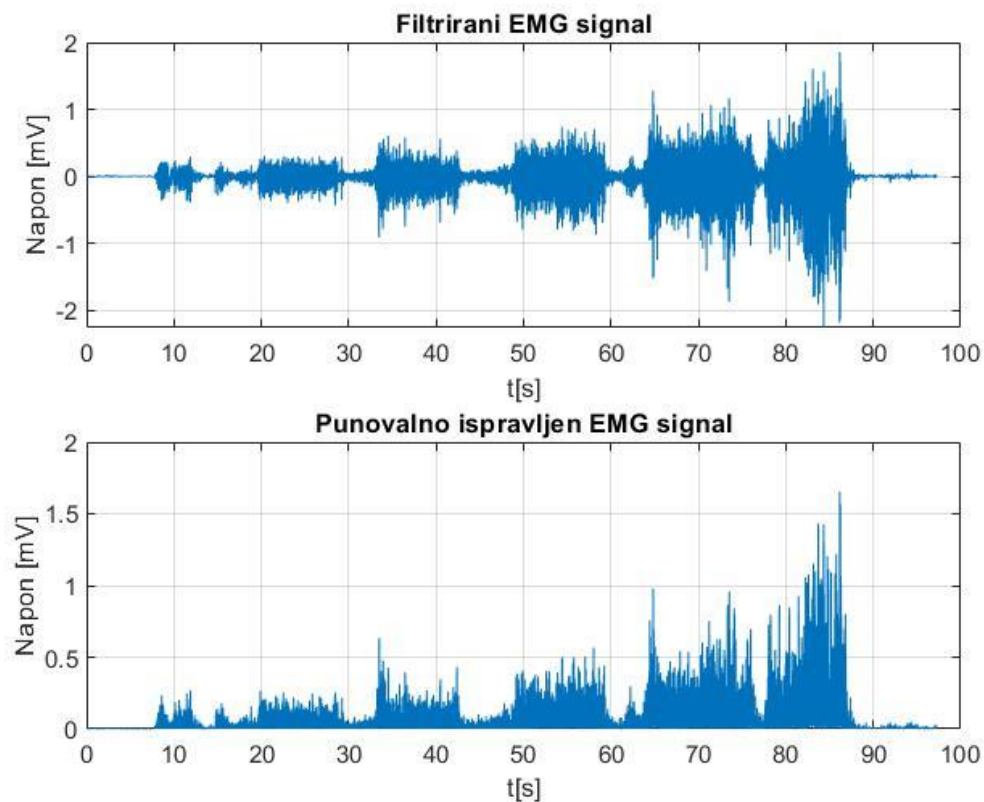
Slika 4.8.: Filtrirani EMG signal i njegov spektar snage ($f_s = 512$ Hz)

4.3. Punovalno ispravljanje

Ovim se postupkom zadržava sva energija signala, a postiže se to da mu srednja vrijednost bude različita od nule, čime se omogućuje primjena raznih postupaka usrednjavanja. Budući da se EMG signal bilježi obično s izmjenično spregnutim pojačalom, izravna primjena operacije usrednjavanja rezultirala bi nultom vrijednošću. U *MatLab-u* ovo se može postići na dva sljedeća načina:

```
1. signal_punovalno_ispravljen=abs(signal_ispravljen_offset);
2. signal_punovalno_ispravljen=signal_ispravljen_offset.^2;
```

Primjeni li se jedan od ova dva načina na prethodno filtrirani signal sa Slike 4.8., dobije se punovalno ispravljeni EMG signal čiji je valni oblik prikazan na Slici 4.9.



Slika 4.9.: Punovalno ispravljen EMG signal ($f_s = 512 \text{ Hz}$)

4.4. Izgladivanje punovalno ispravljenog signala

Budući da je EMG signal zapravo signal slučajne prirode to rezultira činjenicom da se neobrađeni (sirovi) EMG signal ne može ponovno precizno reproducirati. Kako bi se riješio taj problem provodi se postupak izgladivanja signala. U prošlosti se postupak provodio analognim putem odnosno vršilo se potiskivanje viših frekvencija, tj. niskopropusno filtriranje punovalno ispravljenog EMG signala, dok se danas to pretežito provodi digitalnim putem. Digitalizacijom se danas pretežito koriste algoritam usrednjavanja (engl. “*Moving Average*”) i algoritam srednje kvadratne vrijednosti (engl. “*Root Mean Square (RMS)*”) koji će biti opisani u daljnjem tekstu.

4.4.1. Algoritam usrednjavanja

Algoritam usrednjavanja zapravo u “digitalnom svijetu” analogan je niskopropusnom filtriranju u “analognom svijetu” (gdje širina pojasa filtra određuje stupanj izgladivanja). Provođenje algoritma može se objasniti na vrlo jednostavan način: “Uzima se srednja vrijednost signala u prozoru koji “klizi” duž signal.”

$$|\overline{m(t)}| = \frac{1}{t_j - t_i} \int_{t_i}^{t_j} |m(t)| dt \quad (4.1)$$

$T = t_j - t_i$ označuje vremenski prozor. Kad se prozor pomiče duž signala radi se o operaciji pomičnog usrednjavanja (engl. “*Moving Average*”). Razne su mogućnosti pozicioniranja prozora, prema nekim literaturama T se za potrebe kineziološke biomehaničke primjene kreće od 100 do 200 ms [3]. Ako je “prozor” u kojem se uzima srednja vrijednost signala simetričan i centriran, u tom slučaju algoritam ne utječe na fazu ili vrijeme signala. Takav filter poznat je kao “*zero-phase*” filter. Iz toga možemo vidjeti prednost ovakvog digitalnog filtriranja naspram analognom filtriranju budući da eliminira fazni pomak signala. U *MatLab-u* ovaj algoritam implementiran je preko sljedeće naredbe:

```
EMG_movavg = movmean(EMG_punovalno_ispravljen, N);
```

gdje se prvo unosi signal koji se teži filtrirati, zatim širina prozora N u broju uzoraka.

4.4.2. Algoritam srednje kvadratne vrijednosti

Alternativni način zaglađivanja EMG signala je izračunavanje srednje kvadratne vrijednosti (engl. “*Root Mean Square (RMS)*”) signala unutar prozora koji “klizi” po signalu.. Ovaj se pristup matematički samo neznatno razlikuje od pristupa usrednjavanja.

$$RMS\{m(t)\} = \sqrt{\frac{1}{T} \int_t^{t+T} m^2(t) dt} \quad (4.2)$$

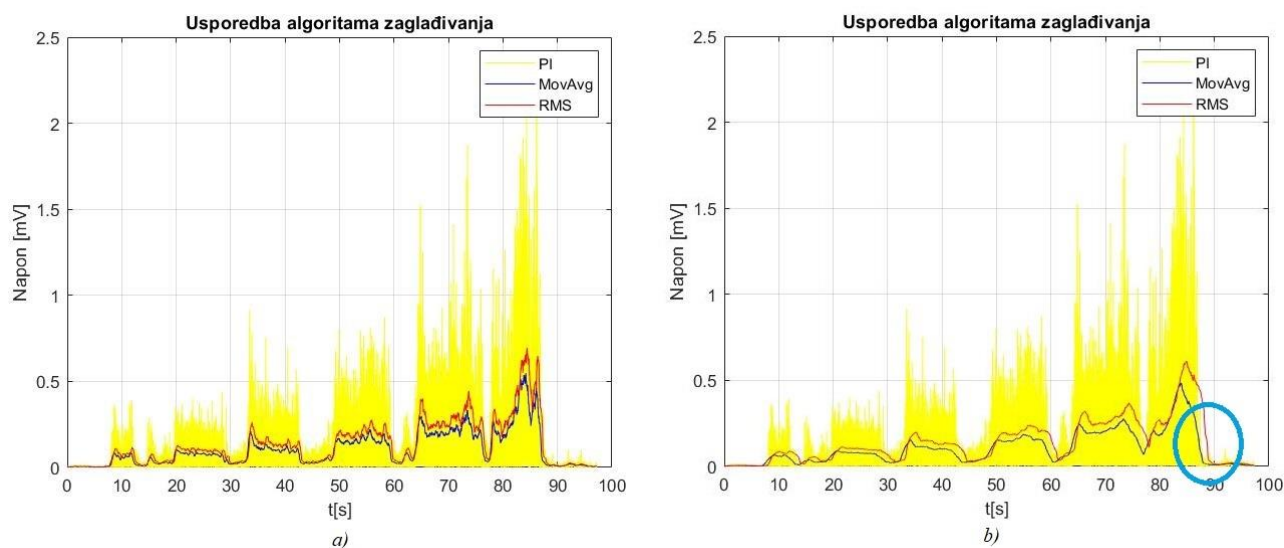
Kvadratna srednja vrijednost oznaka je za snagu signala, pa stoga za razliku od preostalih tipova obrada signala može imati fiziološko značenje, te je uz to i preporučeni algoritam za upotrebu. [3]

U *MatLab-u* ovaj algoritam implementiran je pomoću sljedeće naredbe:

```
movRMS = dsp.MovingRMS(N);  
EMG_rms = movRMS(EMG_punovalno_ispravljen);
```

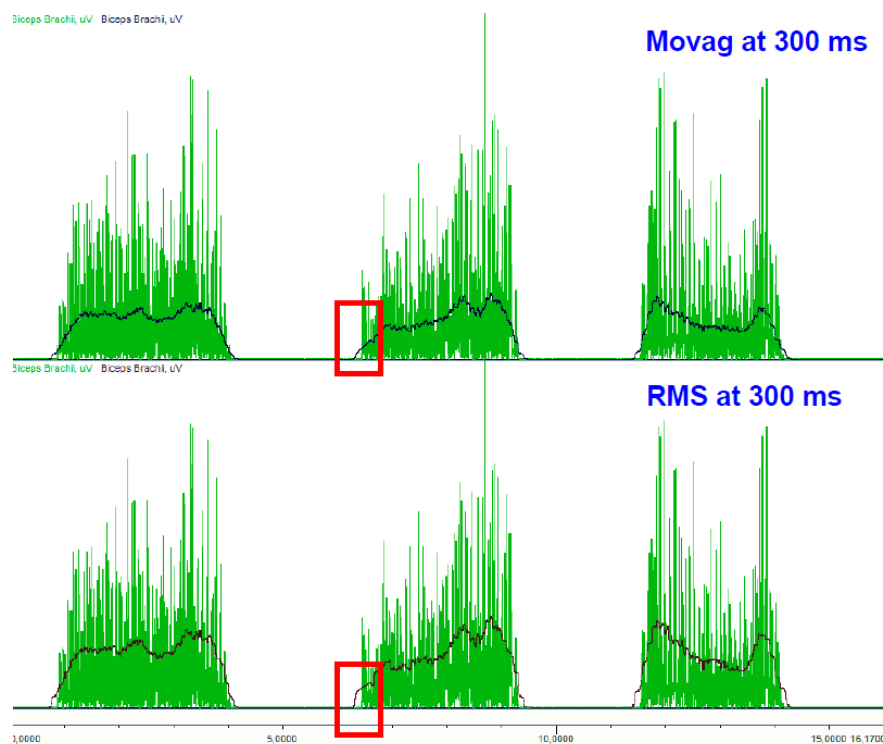
4.4.3. Utjecaj odabira širine prozora na fazno kašnjenje

Za promatranje utjecaja koji ima odabir širine prozora na fazno kašnjenje, na punovalno ispravljen signal sa Slike 4.9. primijenila su se oba algoritma zaglađivanja uz širine prozora 250 i 1000 uzoraka te su dobiveni grafički prikazi prikazani na Slici 4.10.



Slika 4.10.: Usporedba faznog kašnjenja pri a) $N = 250$ i b) $N = 1000$ ($f_s = 512$ Hz)

Iz Slike 4.9. može se primijetiti da povećanjem broja uzoraka rms algoritam ima veće fazno kašnjenje nego movavg algoritam. Prikazom usporedbe faznog kašnjenja oba algoritma iz literature [3] na slici 4.11., zapaža se isti zaključak, a to je da rms algoritam ima veće fazno kašnjenje signala.



Slika 4.11.: Usporedba Movag i RMS algoritma s istom širinom prozora iz literature [3]

4.5. MVC normalizacija i koeficijent

Kako izmjereni EMG signal ovisi o podosta čimbenika (filtarski utjecaj kože, pozicija elektroda,...) te i čak o samom ispitaniku i njegovim individualnim karakteristikama, nije moguće provoditi pouzdane usporedbe između raznih pokušaja, mišića ili ispitanika. Baš iz toga razloga javlja se potreba za normalizacijom EMG signala. Normalizirani EMG signal omogućuje valjane analize između raznih ispitanika, mišića i dr. Zajednički je konsenzus da bi "dobra" referentna vrijednost za normalizaciju EMG signala trebala imati visoku ponovljivost, posebno kod istog pokusa u istoj sesiji, i biti značajna. Odabirom referentne vrijednosti koja se ponavlja unutar pojedinca, mogu se usporediti razine dobivene iz bilo kojeg pokusa s tom referentnom vrijednošću.

Najkorišteniji postupak normalizacije je normalizacija prema maksimalnoj voljnoj kontrakciji mišića (engl. *“Maximum Voluntary Contraction (MVC)”*). Nakon što se mišić zagrije, prvo se provodi ispitivanje sile koju daje mišić dok je statički opterećen kod maksimalnog iznosa sile

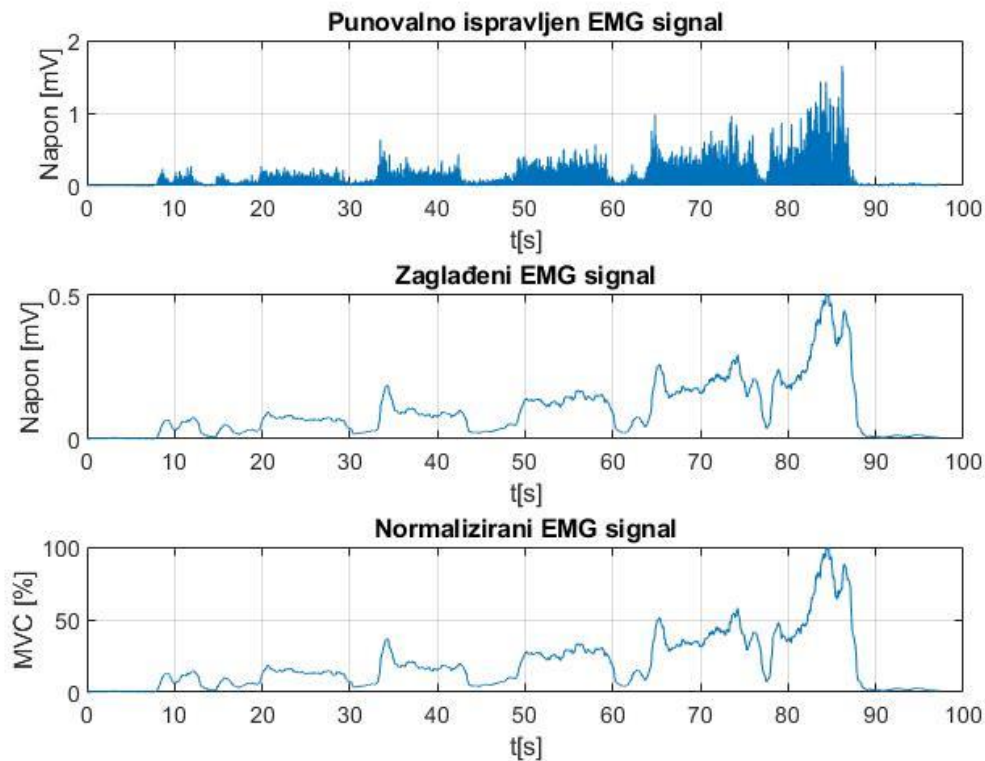
kojom može djelovati. Zatim se taj postupak ponavlja nekoliko puta ovisno o mišiću i vrsti statičkog opterećenja. Nakon toga se obavlja proces normalizacije, gdje se odabire period s najvišom amplitudom izmjerenog signala (MVC koeficijent), a sve ostale amplitude EMG signala s istog mišića prilikom drugih aktivnosti prikazuju se u postotku izmjerene maksimalne amplitude. [3]

U *MatLabu* MVC normalizacija se može implementirati na sljedeća dva načina:

```
sig = EMG_smoothed;
[MVC_coeff, X_max] = max(sig)
EMG_MVC = [];
for i = 1:length(sig)
    mvc = sig(i)/MVC_coeff*100;
    EMG_MVC = [EMG_MVC;mvc];
end
```

```
sig = EMG_smoothed;
[MVC_coeff, X_max] = max(sig)
EMG_MVC = (EMG_smoothed/ MVC_coeff) *100;
```

Za upotrebu kod obrade u stvarnom vremenu preporučeno je koristiti drugi način implementacije, jer je prvi način vremenski sporiji zbog ulaženja u *for* petlju. Primjeni li se na signal sa Slike 4.9. jedan od algoritma zaglađivanja, te jedan od gore dva navedena načina MVC normalizacije dobije se grafički prikaz signala koji su prikazani na Slici 4.12.



Slika 4.12.: Zaglađeni (RMS) i normalizirani EMG signal ($f_s = 512 \text{ Hz}$, $N = 500$, $MVC_{coeff} = 0.5041$)

4.5.1. Utjecaj odabira algoritma zaglađivanja i širine prozora na MVC koeficijent

Za promatranje utjecaja algoritma zaglađivanja uz različite širine prozora koje imaju na MVC koeficijent te fazno kašnjenje, koristit će se filtrirani i punovalno ispravljeni EMG signal sa Slike 4.9. Za širine prozora uzete su tri širine: 250, 500 i 1000 uzoraka pri frekvenciji uzorkovanja od 512 Hz. U tablici 4.1. prikazani su dobiveni MVC koeficijenti za obje vrste algoritma zaglađivanja i za različite širine prozora.

Tablica 4.1.: Vrijednost MVC koeficijenta s obzirom na širinu prozora ($f_s = 512$ Hz)

Broj uzoraka (N)	MVC (movavg)	MVC (rms)
250	0.5483	0.6957
500	0.5162	0.6525
1000	0.4846	0.6133

Iz Tablice 4.1. može se zaključiti da povećanjem širine prozora vrijednost MVC koeficijenta se smanjuje. Dodatno se može primijetiti da je kod rms algoritma vrijednost MVC koeficijenta općenito veća nego kod movavg algoritma, što je i razlog zašto se preporučuje upotreba rms algoritma.

Nakon pregleda načina kako su se implementirali postupci predobrade signala u *MatLab-u* te razmotrili njihovi utjecaji na primjerima izmjerenih signale u ovome poglavlju. U sljedećem poglavlju biti će opisani načini implementacije nekih od tih postupaka za obradu, prikaz i pohranu EMG signala u stvarnome vremenu.

5. Procjena napetosti mišića u realnom vremenu

Komunikacija računala sa *Shimmer3* senzorom vrši se preko *Bluetooth-a* pomoću *Shimmer Matlab Instrument Driver* [10] koji omogućuje sam proizvođač i koji je *OpenSource*. Prema uputama [10] potrebno je instalirati *Realterm Serial Terminal* koji služi kao komunikacija preko serijskog *Porta* i kao međuspremnik (engl. “*Buffer*”). Naime *Shimmer Matlab Instrument Driver* u sebi sadrži već gotove primjere *MatLab* skripta koje se mogu modificirati za vlastite potrebe. U tu svrhu korištena je i modificirana *Plotandwriteemgexample.m* [10] skripta iz koje su nastale tri skripte koje se koriste za obradu i pohranu podataka u stvarnome vremenu:

1. *MVC_coeff_evaluation.m* skripta – služi za određivanje MVC koeficijenta pri statičkom opterećenju mišića
2. *MVCMeasure.m* skripta - podskepta koju pokreće *MVC_coeff_evaluation.m* skripta
3. *RealTimeEMG.m* – služi za obradu i pohranu podataka u stvarnom vremenu

U sljedećem dijelu teksta ukratko će biti opisano što svaka pojedina skripta radi i koje podatke daje, te će biti opisan postupak ispitivanja i priloženi njegovi rezultati na više uzoraka mjerenja.

5.1. *MVC_coeff_evaluation.m*

Ova skripta sadrži definirane funkcije za obradu mjerenih podataka kako bi se izvršilo određivanje MVC koeficijenta koji se kasnije koristi u *RealTimeEMG.m* skripti. Prije pokretanja ove skripte potrebno je definirati globalne konfiguracijske varijable kao što su broj serijskog porta *COM Port* na koji se spaja *Shimmer3*, širina prozora *N*, vrijeme trajanja mjerenja *duration_time* te ostale varijable koje se koriste kao definicije unutar funkcija u svim skriptama. Pokretanjem ove skripte dobiju se grafički prikazi rezultata mjerenja koji omogućuje mogućnost brze procjene valjanosti mjerenog signala ili postupka mjerenja te odabira odgovarajućeg algoritma zaglađivanja i širine prozora.

Obrada podataka se odvija u sljedećim koracima:

1. učitavanje izmjerenog i filtriranog signala iz *Shimmera*
2. punovalno ispravljanje učitano signala
3. primjena oba algoritma za izgladivanje
4. MVC normalizacija za oba algoritma
5. računanje spektra izmjerenog signala
6. prikaz podataka korisniku
7. spremanje izmjerenih podataka u *.dat* datoteku (*MVC_coeff.dat*)

Na Slici 5.1. prikazan je primjer spremljenih izmjerenih podataka iz dobivene *MVC_coeff.dat* datoteke.

MVC_Coeff.dat			
	A	B	C
	MVCCoeff		
	Subject	AnaMarkot...	movavg_co...
	Number	Number	Text
1	Subject:	Ana Markot...	movavg_co...
2	Date	30-Aug-2021	rms_coeff:
3	CH1	CH2	
4	mV	mV	
5	-2.5383785...	6.55835423...	
6	-7.6801056...	-3.4047277...	
7	-5.6471200...	-0.0002353...	
8	-0.0001893...	-0.0006961...	
9	-0.0003699...	-0.0009589...	
10	-0.0003969...	0.00040434...	
11	5.75482813...	0.00542253...	
12	0.00127225...	0.01551555...	
13	0.00327680...	0.03020704...	

Slika 5.1.: Podaci iz dobivene *MVC_coeff.dat* datoteke

5.2. MVCMeasure.m

Ovo je pojednostavljena skripta koja služi kao komunikacija između *Shimmera* i računala te je pozivana od strane *MVC_coeff_evaluation.m* skripte. Ujedno, ova skripta služi za trenutni prikaz u stvarnom vremenu prilikom mjerenja MVC koeficijenta. Unutar ove skripte ne dolazi do nikakve obrade podataka, jer sve to se vrši unutar nadređene joj skripte (*MVC_coeff_evaluation.m*).

5.3. RealTimeEMG.m

Ova skripta zaslužna je za obradu i prikaz mjerenih EMG signala iz *Shimmera* u stvarnome vremenu te sve podatke povlači iz prethodnih skripti (neki podaci se definiraju, a neki se izračunavaju u navedenim skriptama). Prema uputama koje se nalaze u Pravitku 5 potrebno je jedino definirati vrstu algoritma koja će se koristiti preko varijable *smoothing* gdje 1 predstavlja *movingaverage*, a 2 *rms* algoritam. Pokretanjem skripte tako da se pozove istoimena funkcija u *MatLab-ovom Command Window-u*. Kao argumenti se upisuju broj serijskog porta na koji je spojen *Shimmer* senzor te je preporučeno staviti što duže vrijeme trajanja mjerenja. Mjereni podaci spremaju se u „*Buffer*“ koji se neprestano briše i popunjava. Nad podacima spremljenim u „*Bufferu*“ vrši se obrada podataka navedenim metodama u Poglavlju 4., zatim se ti podaci prikazuju i spremaju. Snimanje podataka zaustavlja se pritiskom na crvenu tipku „*Stop Stream*“ koja se nalazi u zasebnom figure-u. Završetkom streama podaci se spremaju u sljedeće .dat datoteke:

- *file_raw.dat* – spremljeni RAW podaci iz *Shimmera*
- *file_filtered.dat* – spremljeni filtrirani podaci
- *file_rectified.dat* – spremljeni punovalno ispravljeni podaci
- *file_smoothed.dat* – spremljeni podaci nakon algoritma zaglađivanja
- *file_mvc.dat* – spremljeni podaci nakon MVC normalizacije

5.3. Postupak testiranja s utezima

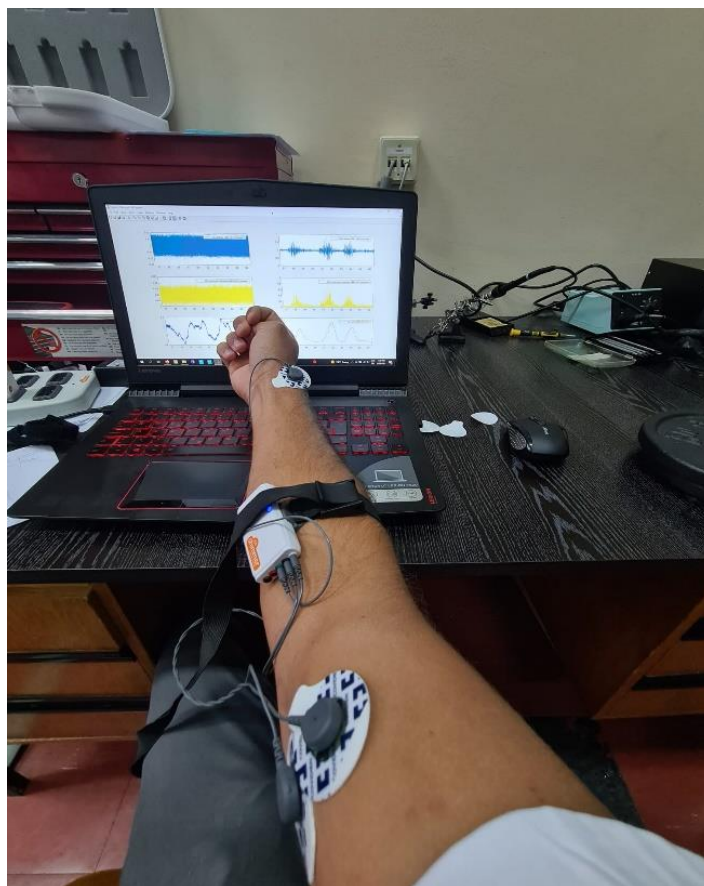
Kao postupak testiranja koristili su se metalna šipka te utezi od 1 i 2 kg. Ukupna maksimalna težina je iznosila oko 8 kg. Sva ispitivanja su se vršila s mjerenjem jednog kanala (CH2) s elektrodama pozicioniranim na bicepsu (Slika 5.2.). Ispitivanje se vršilo sljedećim redom:

1) Mjerenje MVC koeficijenta

- Pokretanje *MVC_coeff_evaluation.m* skripte
- Maksimalno naprezanje mišića uz pruženi statički otpor
- Provjera rezultata mjerenja
- Ponavljanje mjerenja ako je potrebno uz nove postavke

2) Ispitivanje pri raznim težinama utega

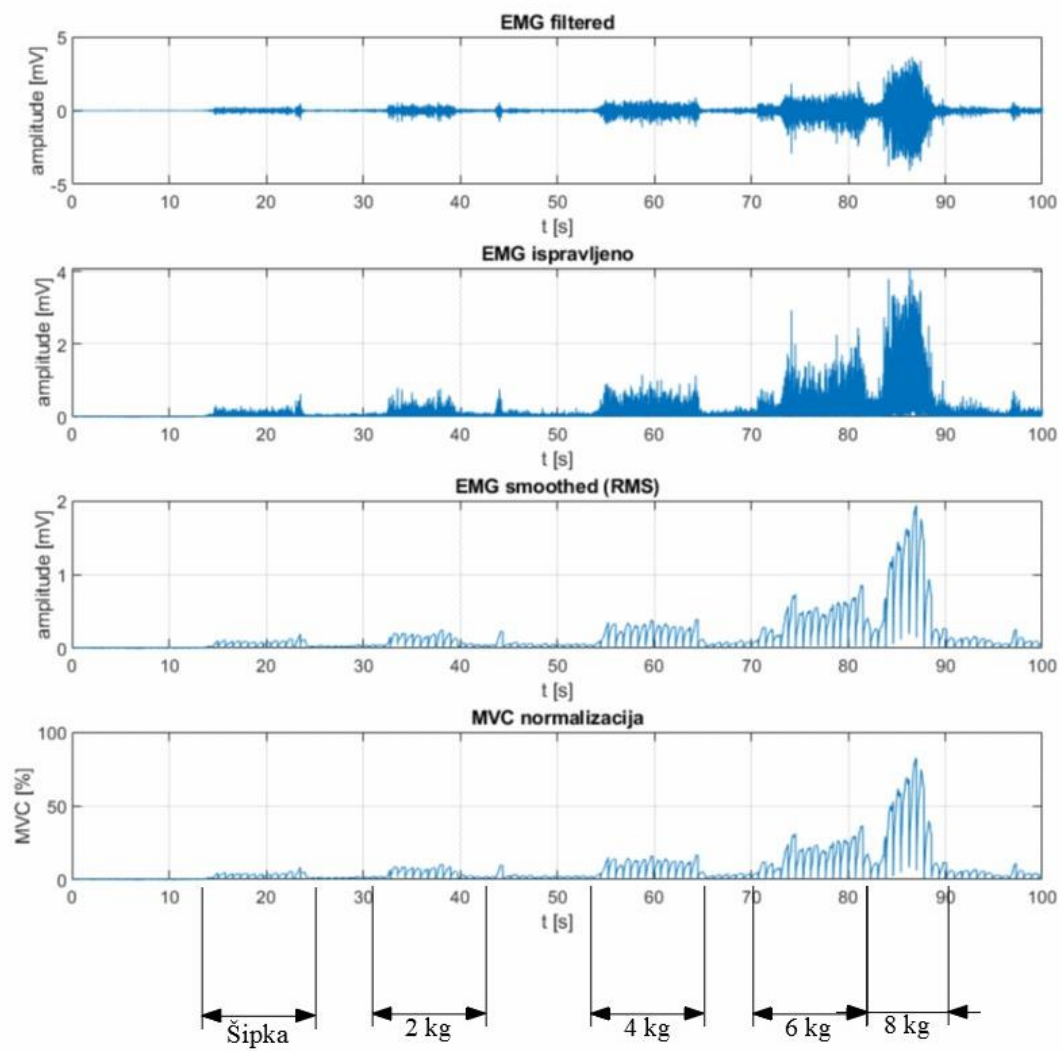
- Pokretanje *RealTimeEMG.m* skripte
- Položaj ruke u neutralnom položaju uz opušten mišić (pridržavanje šipke na koljenu)
- Polagano podizanje šipke, zadržavanje nekoliko sekundi te vraćanje u neutralni položaj
- Dodavanje utega od 2 kg te ponavljanje koraka od točke 2 do 3 pritom dodajući težinu od 2 kg pri svakom ponavljanju
- Provjera rezultata mjerenja



Slika 5.2.: Položaj elektroda na bicepsu

5.4. Rezultati mjerenja s utezima

Postupak mjerenja s utezima opisan je u prošlom potpoglavlju. Pri ispitivanju se koristio rms algoritam uz širinu prozora $N = 250$ uzoraka i frekvenciju uzorkovanja od 512 Hz. Dobiveni rezultati mjerenja prikazani su na Slici 5.3. Iz Slike 5.3. može se uočiti da porastom opterećenja mišića (težine utega) paralelno i rastu amplituda EMG signala i MVC u postotcima. Pri opterećenju gdje se nalazi samo šipka MVC iznosi oko 10 %, dok pri najvišem opterećenju od 8 kg, MVC iznosi oko 70-ak %. Iz ovog primjera se može zaključiti da je algoritam ispravno implementiran.



Slika 5.3.: Rezultati mjerenja s utezima

6. Zaključak

Open Source projekti uz dobru podršku i dokumentaciju poput *Shimmera* dobra su podloga za razvijanje vlastitih sustava. Tako korištenjem dostupnih primjera i materijala uspješno su razvijene *MatLab* programske skripte koje rade u stvarnom vremenu za prikupljanje, obradu i prikaz podataka EMG signala.

U četvrtom poglavlju pokazani su načini implementacije obrade EMG signala u *MatLab-u*, koji su zatim isprobani i na praktičnim izmjerenim signalima te su uspoređeni s rezultatima iz literature [3]. Prvo se promatrao problem pomaka signala od osnove linije. Zatim uz predloženo rješenje zaključilo se koliko dobra kompenzacija signala utječe na njegov spektar i njegovu vjerodostojnost. Kod dvaju algoritma zaglađivanja, uspoređivani su utjecaji koje ima odabir širine prozora kod faznog kašnjenja signala i na MVC koeficijent. Zaključeno je ako se želi manje fazno kašnjenje signala koristit će se algoritam usrednjavanja, uz posljedicu manje vrijednosti MVC koeficijenta. No usprkos prednostima algoritma usrednjavanja veća vrijednost MVC koeficijenta ima znatno veći plus stoga je preporučeno da se koristi srednje kvadratne vrijednosti (RMS).

U petom poglavlju opisane su razvijene *MatLab* skripte koje rade u stvarnom vremenu te je predstavljen postupak ispitivanja s utezima kako bi se provjerio konačni rezultat. Iz Slike 5.3. gdje se vidi opterećenje mišića s raznim težinama utega, moglo se primijetiti kako porastom opterećenja mišića (povećanjem težine tereta), analogno raste i napetost koja se može vidjeti na zadnjem grafu. Iz toga eksperimenta zaključeno je kako je algoritam za procjenu napetosti mišića uspješno implementiran u *MatLabu*.

Konačna svrha ovog projekta bila bi ta da se obrađeni podaci mjerenih EMG signala pomoću razvijenih skripti koriste kao povratna veza u regulaciji sa zatvorenom petljom za upravljanje aktuatora. U konačnici to bi se moglo postići na način tako da se nađe veza između *RealTerm* međuspremnik koji koristi ova skripta za privremenu pohranu i obradu podataka i sučelja koje upravlja aktuatorima. No trenutno sva obrada koja se odvija se odmah zapisuje u *.dat* datoteke te bi se trebalo naći način da se ona premjesti u nekakve vektore u *RealTerm* međuspremniku bez da ima neki prevelik utjecaj na memoriju. Drugo ograničenje su same brzine samog *MatLab software-a*.

Usprkos svemu tome ovaj projekt dobar je alat za praćenje napretka individualaca prilikom rehabilitacije zbog spremanja i prikaza podataka koji se odvija. Uz to dobra je podloga za daljnje radove u ovome području.

7. Popis literature

- [1] “EMG (elektromiografija) ruku i nogu – kako i zašto se izvodi, priprema, cijena”, s Interneta, <https://www.krenizdravo.hr/zdravlje/pretrage/emg-elektromiografija-ruku-i-nogu-kako-i-zasto-se-izvodi-priprema-cijena>, 4. rujna 2021.
- [2] “Electromyography for Ecoskeletons and Earlier Back Support Wearables”, s Interneta, <https://exoskeletonreport.com/2016/08/electromyography-exoskeletons-earlier-back-support-wearables/>, 4. rujna 2021.
- [3] Konrad, Peter: „The ABC of EMG“, Noraxon INC. USA, Verzija 1.0, travanj 2005
- [4] Weakness, the ‘Motor Unit’ and Related Testing“, s Interneta, https://www.mspca.org/angell_services/weakness-the-motor-unit-and-related-testing/, 4. rujna 2021.
- [5] The application of data analysis methods for surface electromyography in shot putting and sprinting - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-motor-unit-action-potential-MUAP-recorded-by-the-surface-EMG-The-MUAP-is-composed_fig3_316688980, 4. rujna 2021.
- [6] Winter, A. David: „Biomechanics and motor control of human movements“, Šveučilište Waterloo, Kanada, 4. Izdanje
- [7] „Analogno-digitalni pretvarač Wikipedia“, s Interneta, https://hr2.wiki/wiki/Analog-to-digital_converter#Resolution, 5. rujna 2021.
- [8] “Shimmer User Manual Revision 3p”, s Interneta, <http://www.shimmersensing.com/support/wireless-sensor-networks-documentation/>, 5. rujna 2021.
- [9] „Shimmer Matlab Instrument Driver User Manual rev2.8a“, s Interneta, <https://github.com/ShimmerEngineering/Shimmer-MATLAB-ID/blob/master/Shimmer%20Matlab%20Instrument%20Driver%20User%20Manual%20rev2.8a.pdf>, 6. rujna 2021.

- [10] „Shimmer Matlab Instrument Driver User Manual rev2.8a“, s Interneta,
<https://github.com/ShimmerEngineering/Shimmer-MATLAB-ID/blob/master/Shimmer%20Matlab%20Instrument%20Driver%20User%20Manual%20rev2.8a.pdf>, 6. rujna 2021.
- [11] Stojković, N.; Mijat, N.: „Analogna obrada signala“, Sveučilište u Rijeci, Tehnički Fakultet, siječanj 2004.

8. Tablica individualnih doprinosa

Student	JMBAG	Doprinos
Valentina Canjuga	0069072036	Teoretska podloga kod pisanja seminara, prikupljanje uzoraka u labosu
Daniel Đuranović	0069077256	MatLab programiranje i rad sa Shimmerom, testiranje i prikupljanje uzoraka u labosu, pisanje seminara.
Ana Markota	0069073332	Teoretska podloga kod pisanja seminara, prikupljanje uzoraka u labosu

Privitak 1 - Kompenzacija_PI_ispravljanje.m skripta

```
%KMS-EMG-seminar - Srednja vrijednost signala sada je na  
nuli  
%Punovovalno ispravljanje i ispravljanje DC offseta  
  
function [signal_punovalno_ispravljen,  
signal_ispravljen_offset] =  
Kompenzacija_PI_ispravljanje(signal, SirinaProzora)  
    signal_ispravljen_offset = [];  
  
    for dx = 1:SirinaProzora:length(signal)  
  
        if(length(signal)-dx < SirinaProzora) %uvjet da se  
i uzme uzorak zadnjeg dijela signala  
            prozor = signal(dx:length(signal));  
        else  
            prozor = signal(dx:dx+SirinaProzora-1);  
        end  
  
        mean_prozor = mean(prozor);  
        korekcija = prozor - mean_prozor;  
        signal_ispravljen_offset =  
[signal_ispravljen_offset; korekcija]; %spajanje 2  
stupcasta vektora  
    end
```

Privitak 2 – MVCMeasure.m skripta

```
function shimmer = MVCMeasure(comPort, captureDuration)
    % INPUT: comPort - String value defining the COM
    port number for Shimmer
    %
    % INPUT: captureDuration - Numerical value defining
    the capture time duration
    %
    % OUTPUT: shimmer - Object of the
    ShimmerHandleClass
    %
    % OUTPUT: '.dat' files of all data
    %
    % Example for Shimmer3
    addpath('./Resources/')
    % directory containing supporting functions

    close all
    %% definitions
    global subject
    global c
    global COM_Port
    global fs
    global N

    %File name definition
    file_raw = 'MVC_Coeff_Raw_data.dat';
    file_filtered = 'MVC_Coeff.dat';

    %Write file
    filename = fullfile('./', file_filtered);
    fid = fopen(file_filtered, 'wt');
    fprintf(fid,
    '%s\t%s\t%s\t\n%s\t%s\t%s\t\n%s\t%s\t\n%s\t%s\t\n',
    'Subject:', subject, 'movavg_coeff:', ...
    'Date', c, 'rms_coeff:', 'CH1', 'CH2', 'mV', 'mV');
    fclose(fid);

    % Stop loop contidition configuration (figure)
    ButtonHandle = uicontrol('Style', 'PushButton', ...
    'String', 'STOP STREAM', ...
    'Callback', 'delete(gcf)');
```



```

ButtonHandle.Position = [200 150 200 200];
ButtonHandle.BackgroundColor = 'r';

%shimer
shimmer = ShimmerHandleClass(comPort);
SensorMacros = SetEnabledSensorsMacrosClass;
% assign user friendly macros for setenabledsensors

firsttime = true;

% Note: these constants are only relevant to this
examplescript and are not used
% by the ShimmerHandle Class
NO_SAMPLES_IN_PLOT = 5000;
% Number of samples in the plot
DELAY_PERIOD = 0.2;
% Delay (in seconds) between data read operations
numSamples = 0;

%% settings

% filtering settings
fm = 50;
% mains frequency [Hz]
fchp = 5;
% corner frequency highpassfilter [Hz]; Shimmer recommends
5Hz to remove DC-offset and movement artifacts
nPoles = 4;
% number of poles (HPF, LPF)
pbRipple = 0.5;
% pass band ripple (%)

HPF = true;
% enable (true) or disable (false) highpass filter
LPF = true;
% enable (true) or disable (false) lowpass filter
BSF = true;
% enable (true) or disable (false) bandstop filter

% highpass filters for ExG channels
if (HPF)
    hpfexg1ch1 =
FilterClass(FilterClass.HPF,fs,fchp,nPoles,pbRipple);

```

```

        hpfexg1ch2 =
FilterClass(FilterClass.HPF,fs,fchp,nPoles,pbRipple);
    end
    if (LPF)
        % lowpass filters for ExG channels
        lpfexg1ch1 = FilterClass(FilterClass.LPF,fs,fs/2-
1,nPoles,pbRipple);
        lpfexg1ch2 = FilterClass(FilterClass.LPF,fs,fs/2-
1,nPoles,pbRipple);
    end
    if (BSF)
        % bandstop filters for ExG channels;
        % cornerfrequencies at +1Hz and -1Hz from mains
frequency
        bsfexg1ch1 = FilterClass(FilterClass.LPF,fs,[fm-
1, fm+1],nPoles,pbRipple);
        bsfexg1ch2 = FilterClass(FilterClass.LPF,fs,[fm-
1, fm+1],nPoles,pbRipple);
    end

    %%
    if (shimmer.connect)
        shimmer.setsamplingrate(fs);
% Select sampling rate
        shimmer.setinternalboard('EMG');
% Select internal expansion board; select 'EMG' to enable
SENSOR_EXG1
        shimmer.disableallsensors;
% Disable other sensors
        shimmer.setenabledsensors(SensorMacros.EMG,1);
% Enable SENSOR_EXG1, disable other sensors

        if (shimmer.start)
% TRUE if the shimmer starts streaming

            plotData = [];
            timeStamp = [];
            filteredplotData = [];

            h.figure1=figure('Name','Shimmer EMG signals');
% Create a handle to figure for plotting data from shimmer
            set(h.figure1, 'Position', [100, 500, 800,
400]);

```

```

        elapsedTime = 0;
% Reset to 0
        tic;
% Start timer

        while (elapsedTime < captureDuration)

            pause(DELAY_PERIOD);
% Pause for this period of time on each iteration to allow
data to arrive in the buffer

            %
[newData,signalNameArray,signalFormatArray,signalUnitArray]
= shimmer.getdata('Time Stamp','c','ECG','c'); % Read the
latest data from shimmer data buffer, signalFormatArray
defines the format of the data and signalUnitArray the unit
            % or use

[newData,signalNameArray,signalFormatArray,signalUnitArray]
= shimmer.getdata('c'); % Read the latest data from
shimmer data buffer, signalFormatArray defines the format
of the data and signalUnitArray the unit

            if (firsttime==true && isempty(newData)~=1)
                firsttime =
writeHeadersToFile(file_raw,signalNameArray,signalFormatArr
ay,signalUnitArray);
            end

            if ~isempty(newData)
% TRUE if new data has arrived

                chIndex(1) =
find(ismember(signalNameArray, 'EMG CH1'));
                chIndex(2) =
find(ismember(signalNameArray, 'EMG CH2'));

                EMGData = newData(:,chIndex);
                EMGDataFiltered = EMGData;
                % filter the data
                if HPF % filter newData with
highpassfilter to remove DC-offset

```

```

        EMGDataFiltered(:,1) =
hpfexg1ch1.filterData(EMGDataFiltered(:,1));
        EMGDataFiltered(:,2) =
hpfexg1ch2.filterData(EMGDataFiltered(:,2));
    end

    if BSF % filter highpassfiltered data
with bandstopfilter to suppress mains interference
        EMGDataFiltered(:,1) =
bsfexg1ch1.filterData(EMGDataFiltered(:,1));
        EMGDataFiltered(:,2) =
bsfexg1ch2.filterData(EMGDataFiltered(:,2));
    end

    if LPF % filter bandstopfiltered data
with lowpassfilter to avoid aliasing
        EMGDataFiltered(:,1) =
lpfexg1ch1.filterData(EMGDataFiltered(:,1));
        EMGDataFiltered(:,2) =
lpfexg1ch2.filterData(EMGDataFiltered(:,2));
    end

    plotData = [plotData; EMGData];
% Update the plotData buffer with the new ECG data
    filteredplotData = [filteredplotData;
EMGDataFiltered]; % Update the filteredplotData buffer
with the new filtered ECG data
    numPlotSamples = size(plotData,1);
    numSamples = numSamples +
size(newData,1);

    timeStampNew = newData(:,1);
% get timestamps
    timeStamp = [timeStamp; timeStampNew];

    if numSamples > NO_SAMPLES_IN_PLOT
        plotData = plotData(numPlotSamples-
NO_SAMPLES_IN_PLOT+1:end,:);
        filteredplotData =
filteredplotData(numPlotSamples-
NO_SAMPLES_IN_PLOT+1:end,:);
    end
    sampleNumber = max(numSamples-
NO_SAMPLES_IN_PLOT+1,1):numSamples;

```

```

                                %---Obrada---
                                filtered_ispravljeno =
abs (EMGDataFiltered);
                                filtered_ispravljeno_plot =
abs (filteredplotData);

                                %---Write Data---
                                dlmwrite(file_raw, newData, '-append',
'delimiter', '\t','precision',16); % Append the new data to
the file in a tab delimited format
                                dlmwrite(file_filtered,
EMGDataFiltered, '-append', 'delimiter',
'\t','precision',16);
                                %-----

                                set(0,'CurrentFigure',h.figure1);

                                % Plot Raw EMG data (CH1)
                                subplot(3,2,1);

% Create subplot
                                signalIndex = chIndex(1);
                                plot(sampleNumber,plotData(:,1));
% Plot the ecg for channel 1 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' '
signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ') '];
                                legend(legendName1);
% Add legend to plot
                                xlim([sampleNumber(1)
sampleNumber(end)]);

                                % Plot Raw EMG data (CH2)
                                subplot(3,2,2);

% Create subplot
                                signalIndex = chIndex(2);
                                plot(sampleNumber,plotData(:,2));
% Plot the ecg for channel 2 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' '
signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ') '];

```

```

                                legend(legendName1);
% Add legend to plot
                                xlim([sampleNumber(1)
sampleNumber(end)]);

                                % Plot Filtered EMG data (CH1)
                                subplot(3,2,3);
% Create subplot
                                signalIndex = chIndex(1);

plot(sampleNumber/fs,filteredplotData(:,1));                                %
Plot the filtered ecg for channel 1 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'filtered'
' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
                                legend(legendName1);
% Add legend to plot
                                xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

                                % Plot Filtered EMG data (CH2)
                                subplot(3,2,4);
% Create subplot
                                signalIndex = chIndex(2);

plot(sampleNumber/fs,filteredplotData(:,2));                                %
Plot the filtered ecg for channel 2 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'filtered'
' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
                                legend(legendName1);
% Add legend to plot
                                xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

                                %plot rectified signal (CH1)
                                subplot(3,2,5);
% Create subplot
                                signalIndex = chIndex(1);

plot(sampleNumber/fs,filtered_ispravljeno_plot(:,1), 'y');
% Plot the filtered ecg for channel 2 of SENSOR_EXG1

```

```

legendName1=[signalFormatArray{signalIndex} ' ' 'punovalno
ispravljen' ' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')']];
        legend(legendName1);
% Add legend to plot
        xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

        %Plot rectified signal (CH2)
        subplot(3,2,6);

% Create subplot
        signalIndex = chIndex(2);

plot(sampleNumber/fs,filtered_ispravljeno_plot(:,2), 'y');
% Plot the filtered ecg for channel 2 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'punovalno
ispravljen' ' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')']];
        legend(legendName1);
% Add legend to plot
        xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

        % Stop condition
        if ~ishandle(ButtonHandle)
% Ubacen stop condition
                fprintf('\n Data stream stopped by
user! \n');

                break;
        end
        pause(0.01);

        end

        elapsedTime = elapsedTime + toc;
% Update elapsedTime with the time that elapsed since
starting the timer
        tic;
% Start timer

        end

```

```

        elapsedTime = elapsedTime + toc;
% Update elapsedTime with the time that elapsed since
starting the timer
        fprintf('The percentage of received packets: %d
\n',shimmer.getpercentageofpacketsreceived(timeStamp)); %
Detect loss packets
        shimmer.stop;
% Stop data streaming

    end

end

shimmer.disconnect;

clear shimmer ;
end % MVCMeasure

```


Privitak 3 – MVC_coeff_evaluation. m skripta

```
%% Skripta za odredjivanje MSC koeficjenta (MatLab R2018a)
%
% INPUT: Subject, Shimer COM port, duration time
%
% OUTPUT: MSC koef izmjerenog signala (movavg i rms) kao
globalna varijabla
%
%% Definicija globalnih varijabli
global subject
global c
global MVC_Coeff
global COM_Port
global fs
global N

%% Stream setup
subject = 'Ana Markota2';
c = date;
COM_Port = '6';
duration_time = 45; % [s]

%% Create file and run MVCMeasure script

file_filtered = 'MVC_Coeff.dat';
MVCMeasure(COM_Port, duration_time);
%% Load data from writen file

EMG_raw = dlmread(file_filtered, '\t', 4,1);
%% Shimmer3 Setup

%sampling frequency
fs = 512; % [Hz]

%Smoothing algorithm setup
N = 250; % Window size [broj uzoraka] - (u sekundama =
N/fs)

%% Obrada podataka

%kreiranje vektora x osi za plot
max_t = size(EMG_raw)-1;
```

```

t = 0:max_t;
t = t(:);
t = t/fs; %pretvorba iz broja uzorka u vrijeme

%Punovalno ispravljanje - vec filt. sign. iz shimera
EMG_punovalno_ispravljen = EMG_raw.^2;

%MOVAVG
EMG_movavg = movmean(EMG_punovalno_ispravljen, N);

%MMS
movRMS = dsp.MovingRMS(N);
EMG_rms = movRMS(EMG_punovalno_ispravljen);

%Normalizacija
MVC_Coeff = [0;0];

%MVC Normalizacija (movavg)
sig1 = EMG_movavg;

[MVC_Coeff(1), X_max2] = max(sig1);
EMG_MVC_movavg = [];
for i = 1:length(sig1)
    mvc1 = sig1(i)/MVC_Coeff(1)*100;
    EMG_MVC_movavg = [EMG_MVC_movavg;mvc1];
end

%MVC Normalizacija (rms)
sig2 = EMG_rms;

[MVC_Coeff(2), X_max2] = max(sig2);
EMG_MVC_rms = [];
for i = 1:length(sig2)
    mvc2 = sig2(i)/MVC_Coeff(2)*100;
    EMG_MVC_rms = [EMG_MVC_rms;mvc2];
end

fprintf('MVC coeff (movAvg) = %f', MVC_Coeff(1))
fprintf('\nMVC coeff (rms) = %f', MVC_Coeff(2))

%% Plotanje
%Usporedba smoothing algoritama
figure()
plot(t,EMG_punovalno_ispravljen, 'y')

```

```

hold on
plot(t, EMG_movavg, 'b')
hold on
plot(t, EMG_rms, 'r')
legend Rectified MovAvg RMS
title('Usporedba smoothing algoritama')
xlabel('t (s)')
ylabel('Napon (mV)')
grid on

```

```

%Usporedba normalizacija
figure()
plot(t, EMG_MVC_movavg, 'b')
hold on
plot(t, EMG_MVC_rms, 'r')
legend movAvg rms
title('Usporedba Normalizacija')
xlabel('t (s)')
ylabel('MVC (%)')
grid on

```

```

%% Spektar signala
sig_fft = EMG_raw;
L=length(sig_fft);
time=L/fs;
Y = fft(sig_fft);

P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);

f = fs*(0:(L/2))/L;
figure()
plot(f,P1)
title('Spektar snage signala')
xlabel('f (Hz)')
ylabel('Snaga signala (uV)^2')
grid on

```

Privitak 4 - RealTimeEMG.m skripta

```
function shimmer = RealTimeEMG(comPort, captureDuration)
    %PLOTANDWRITEEMGEXAMPLE - Plotting ecg signal and write
to file
    %    INPUT: comPort - String value defining the COM
port number for Shimmer
    %
    %    INPUT: captureDuration - Numerical value defining
the capture duration
    %
    %    INPUT: fileName - String value defining the name
of the data file
    %
    %    OUTPUT: shimmer - Object of the
ShimmerHandleClass
    %
    % Example for Shimmer3
    addpath('./Resources/')
% directory containing supporting functions

%% Definitions
close all

global MVC_Coeff
global COM_Port
global fs
global N

%File name definition
file_raw = 'file_raw.dat';
file_filtered = "file_filtered.dat";
file_rectified = "file_rectified.dat";
file_smoothed = "file_smoothed.dat";
file_mvc = "file_mvc.dat";

% Write files loop
files = [file_filtered file_rectified file_smoothed
file_mvc];

for i = 1:4
    filename = fullfile('./', files(i));
    fid = fopen(files(i), 'wt');
    if i == 4
```

```

        fprintf(fid, '%s\t%s\t\n%s\t%s\t\n', 'CH1',
'CH2', '%', '%');
        else
            fprintf(fid, '%s\t%s\t\n%s\t%s\t\n', 'CH1',
'CH2', 'mV', 'mV');
        end
        fclose(fid);
    end

    % Stop loop contidition configuration (figure)
    ButtonHandle = uicontrol('Style', 'PushButton', ...
        'String', 'STOP STREAM', ...
        'Callback', 'delete(gcf)');

    ButtonHandle.Position = [200 150 200 200];
    ButtonHandle.BackgroundColor = 'r';

    %% Shimer Setup
    %fs = 512;
    % sample rate in [Hz]
    %MVC coefficient
    smoothing = 1;
    % 0 = movavg, 1 = rms
    %     N = 250;
    %Window size

    %shimer
    shimmer = ShimmerHandleClass(comPort);
    SensorMacros = SetEnabledSensorsMacrosClass;
    % assign user friendly macros for setenabledsensors

    firsttime = true;

    % Note: these constants are only relevant to this
    % examplescript and are not used
    % by the ShimmerHandle Class
    NO_SAMPLES_IN_PLOT = 5000;
    % Number of samples in the plot
    DELAY_PERIOD = 0.2;
    % Delay (in seconds) between data read operations
    numSamples = 0;

    %% settings

```

```

    % filtering settings
    fm = 50;
% mains frequency [Hz]
    fchp = 5;
% corner frequency highpassfilter [Hz]; Shimmer recommends
5Hz to remove DC-offset and movement artifacts
    nPoles = 4;
% number of poles (HPF, LPF)
    pbRipple = 0.5;
% pass band ripple (%)

    HPF = true;
% enable (true) or disable (false) highpass filter
    LPF = true;
% enable (true) or disable (false) lowpass filter
    BSF = true;
% enable (true) or disable (false) bandstop filter

    % highpass filters for ExG channels
    if (HPF)
        hpfexg1ch1 =
FilterClass(FilterClass.HPF,fs,fchp,nPoles,pbRipple);
        hpfexg1ch2 =
FilterClass(FilterClass.HPF,fs,fchp,nPoles,pbRipple);
    end
    if (LPF)
        % lowpass filters for ExG channels
        lpfexg1ch1 = FilterClass(FilterClass.LPF,fs,fs/2-
1,nPoles,pbRipple);
        lpfexg1ch2 = FilterClass(FilterClass.LPF,fs,fs/2-
1,nPoles,pbRipple);
    end
    if (BSF)
        % bandstop filters for ExG channels;
        % cornerfrequencies at +1Hz and -1Hz from mains
frequency
        bsfexg1ch1 = FilterClass(FilterClass.LPF,fs,[fm-
1,fm+1],nPoles,pbRipple);
        bsfexg1ch2 = FilterClass(FilterClass.LPF,fs,[fm-
1,fm+1],nPoles,pbRipple);
    end

%%

```

```

    if (shimmer.connect)
        shimmer.setsamplingrate(fs);
% Select sampling rate
        shimmer.setinternalboard('EMG');
% Select internal expansion board; select 'EMG' to enable
SENSOR_EXG1
        shimmer.disableallsensors;
% Disable other sensors
        shimmer.setenabledsensors(SensorMacros.EMG,1);
% Enable SENSOR_EXG1, disable other sensors

    if (shimmer.start)
% TRUE if the shimmer starts streaming

        plotData = [];
        timeStamp = [];
        filteredplotData = [];

        h.figure1=figure('Name','Shimmer EMG signals');
% Create a handle to figure for plotting data from shimmer
        set(h.figure1, 'Position', [100, 500, 800,
400]);

        elapsedTime = 0;
% Reset to 0
        tic;
% Start timer

        while (elapsedTime < captureDuration)

            pause(DELAY_PERIOD);
% Pause for this period of time on each iteration to allow
data to arrive in the buffer

            %
[newData,signalNameArray,signalFormatArray,signalUnitArray]
= shimmer.getdata('Time Stamp','c','ECG','c'); % Read the
latest data from shimmer data buffer, signalFormatArray
defines the format of the data and signalUnitArray the unit
            % or use

[newData,signalNameArray,signalFormatArray,signalUnitArray]
= shimmer.getdata('c'); % Read the latest data from

```

shimmer data buffer, signalFormatArray defines the format of the data and signalUnitArray the unit

```
        if (firsttime==true && isempty(newData)~=1)
            firsttime =
writeHeadersToFile(file_raw,signalNameArray,signalFormatArray,signalUnitArray);
        end

        if ~isempty(newData)
% TRUE if new data has arrived

            chIndex(1) =
find(ismember(signalNameArray, 'EMG CH1'));
            chIndex(2) =
find(ismember(signalNameArray, 'EMG CH2'));

            EMGData = newData(:,chIndex);
            EMGDataFiltered = EMGData;
            % filter the data
            if HPF % filter newData with
highpassfilter to remove DC-offset
                EMGDataFiltered(:,1) =
hpfxg1ch1.filterData(EMGDataFiltered(:,1));
                EMGDataFiltered(:,2) =
hpfxg1ch2.filterData(EMGDataFiltered(:,2));
            end

            if BSF % filter highpassfiltered data
with bandstopfilter to suppress mains interference
                EMGDataFiltered(:,1) =
bsfexg1ch1.filterData(EMGDataFiltered(:,1));
                EMGDataFiltered(:,2) =
bsfexg1ch2.filterData(EMGDataFiltered(:,2));
            end

            if LPF % filter bandstopfiltered data
with lowpassfilter to avoid aliasing
                EMGDataFiltered(:,1) =
lpfexg1ch1.filterData(EMGDataFiltered(:,1));
                EMGDataFiltered(:,2) =
lpfexg1ch2.filterData(EMGDataFiltered(:,2));
            end
        end
    end
end
```



```

        plotData = [plotData; EMGData];
% Update the plotData buffer with the new ECG data
        filteredplotData = [filteredplotData;
EMGDataFiltered]; % Update the filteredplotData buffer
with the new filtered ECG data
        numPlotSamples = size(plotData,1);
        numSamples = numSamples +
size(newData,1);

        timeStampNew = newData(:,1);
% get timestamps
        timeStamp = [timeStamp; timeStampNew];

        if numSamples > NO_SAMPLES_IN_PLOT
            plotData = plotData(numPlotSamples-
NO_SAMPLES_IN_PLOT+1:end,:);
            filteredplotData =
filteredplotData(numPlotSamples-
NO_SAMPLES_IN_PLOT+1:end,:);
        end
        sampleNumber = max(numSamples-
NO_SAMPLES_IN_PLOT+1,1):numSamples;

%% Obrada
        filtered_ispravljeno =
abs(EMGDataFiltered);
        filtered_ispravljeno_plot =
abs(filteredplotData);

        if smoothing == 0
            filtered_smoothed =
movmean(filtered_ispravljeno, N);
            filtered_smoothed_plot =
movmean(filtered_ispravljeno_plot, N);
            koef = MVC_Coeff(1);
        else if smoothing == 1
            movRMS = dsp.MovingRMS(N);
            filtered_smoothed =
movRMS(filtered_ispravljeno);
            filtered_smoothed_plot =
movRMS(filtered_ispravljeno_plot);
            koef = MVC_Coeff(2);
        end

```

```

end

MVC = (filtered_smoothed/koef*100);
MVC_plot =
(filtered_smoothed_plot/koef)*100;

%% Write data into files

dlmwrite(file_raw, newData, '-append',
'delimiter', '\t','precision',16); % Append the new data to
the file in a tab delimited format
dlmwrite(file_filtered,
EMGDataFiltered, '-append', 'delimiter',
'\t','precision',16);
dlmwrite(file_smoothed,
filtered_smoothed, '-append', 'delimiter',
'\t','precision',16);
dlmwrite(file_mvc, MVC, '-append',
'delimiter', '\t','precision',16);
dlmwrite(file_rectified,
filtered_ispravljeno, '-append', 'delimiter',
'\t','precision',16);

%% Plotting

set(0,'CurrentFigure',h.figure1);

%           % Plot Raw Signal (CH1)
%           subplot(5,2,1);
% Create subplot
%           signalIndex = chIndex(1);
%           plot(sampleNumber,plotData(:,1));
% Plot the ecg for channel 1 of SENSOR_EXG1
%
legendName1=[signalFormatArray{signalIndex} ' '
signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
%           legend(legendName1);
% Add legend to plot
%           xlim([sampleNumber(1)
sampleNumber(end)]);
%
%           % Plot Raw Signal (CH2)

```

```

%                                subplot(5,2,2);
% Create subplot
%                                signalIndex = chIndex(2);
%                                plot(sampleNumber,plotData(:,2));
% Plot the ecg for channel 2 of SENSOR_EXG1
%
legendName1=[signalFormatArray{signalIndex} ' '
signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
%                                legend(legendName1);
% Add legend to plot
%                                xlim([sampleNumber(1)
sampleNumber(end)]);

% Plot filtered Signal (CH1)
subplot(3,2,1);

% Create subplot
%                                signalIndex = chIndex(1);

plot(sampleNumber/fs,filteredplotData(:,1)); %
Plot the filtered ecg for channel 1 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'filtered'
' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
%                                legend(legendName1);
% Add legend to plot
%                                xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

% Plot filtered Signal (CH2)
subplot(3,2,2);

% Create subplot
%                                signalIndex = chIndex(2);

plot(sampleNumber/fs,filteredplotData(:,2)); %
Plot the filtered ecg for channel 2 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'filtered'
' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
%                                legend(legendName1);
% Add legend to plot

```

```

        xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

        %plot rectified signal (CH1)
        subplot(3,2,3);

% Create subplot
        signalIndex = chIndex(1);

plot(sampleNumber/fs,filtered_ispravljeno_plot(:,1), 'y');
% Plot the filtered ecg for channel 2 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'punovalno
ispravljjen' ' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
        legend(legendName1);
% Add legend to plot
        xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

        %Plot rectified signal (CH2)
        subplot(3,2,4);

% Create subplot
        signalIndex = chIndex(2);

plot(sampleNumber/fs,filtered_ispravljeno_plot(:,2), 'y');
% Plot the filtered ecg for channel 2 of SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'punovalno
ispravljjen' ' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
        legend(legendName1);
% Add legend to plot
        xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

%
        %Plot smoothed signal (CH1)
%
        subplot(5,2,7);
% Create subplot
%
        signalIndex = chIndex(1);
%
        plot(sampleNumber,
filtered_smoothed(:,1), 'r'); % Plot the
filtered ecg for channel 2 of SENSOR_EXG1

```

```

%
legendName1=[signalFormatArray{signalIndex} ' ' 'smoothed'
' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
%
legend(legendName1);
% Add legend to plot
%
xlim([sampleNumber(1)
sampleNumber(end)]);
%
%
%Plot smoothed signal (CH2)
subplot(5,2,8);
% Create subplot
%
signalIndex = chIndex(2);
%
plot(sampleNumber,
filtered_smoothed(:,2), 'r'); % Plot the
filtered ecg for channel 2 of SENSOR_EXG1
%
legendName1=[signalFormatArray{signalIndex} ' ' 'smoothed'
' ' signalNameArray{signalIndex} ' ('
signalUnitArray{signalIndex} ')'];
%
legend(legendName1);
% Add legend to plot
%
xlim([sampleNumber(1)
sampleNumber(end)]);

%Plot normalised signal(CH1)
subplot(3,2,5);
% Create subplot
signalIndex = chIndex(1);
plot(sampleNumber/fs, MVC_plot(:,1),
'b'); % Plot the filtered ecg for channel 2 of
SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'MVC' ' '
signalNameArray{signalIndex} ' (% *) '];
legend(legendName1);
% Add legend to plot
xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

%Plot normalised signal(CH2)
subplot(3,2,6);
% Create subplot
signalIndex = chIndex(2);

```

```

        plot(sampleNumber/fs, MVC_plot(:,2),
'b');          % Plot the filtered ecg for channel 2 of
SENSOR_EXG1

legendName1=[signalFormatArray{signalIndex} ' ' 'MVC' ' '
signalNameArray{signalIndex} ' (%) * '];
        legend(legendName1);
% Add legend to plot
        xlim([sampleNumber(1)/fs
sampleNumber(end)/fs]);

        % Stop condition
        if ~ishandle(ButtonHandle)
% Ubacen stop condition
                fprintf('\n Data stream stopped by
user! \n');
                break;
        end
        pause(0.01);

        end

        elapsedTime = elapsedTime + toc;
% Update elapsedTime with the time that elapsed since
starting the timer
        tic;
% Start timer

        end

        elapsedTime = elapsedTime + toc;
% Update elapsedTime with the time that elapsed since
starting the timer
        fprintf('The percentage of received packets: %d
\n',shimmer.getpercentageofpacketsreceived(timestamp)); %
Detect loss packets
        shimmer.stop;
% Stop data streaming

        end

        end

        shimmer.disconnect;

```

```
        clear shimmer ;  
end % RealTimeEMG
```

Napomena: skripta rađena u MatLab R2018a

1. Setup

1. Potrebno instalirati Realterm (link: <https://sourceforge.net/projects/realterm/>)
2. Spojiti Shimmer3 preko Bluetooth-a (PIN: 1234)
3. Provjeriti COM Port od Shimmera (Control Panel -> Devices and Printers -> Shimmer3-xxx-> Dupli klik -> Hardware Tab)

Prije pokretanja MatLab skripte provjeriti da se u istom folderu i workpathu nalaze sljedeće stvari:

- folder: „quaternion“
- folder: „realtermBuffer“
- folder: „Resources“
- FilterClass.m
- ShimmerBiophysicalProcessingLibrary_Rev_0_10
- SetEnabledSensorsMacrosClass.m
- ShimmerHandleClass.m
- MVC_coeff_evaluation.m
- MVCMeasure.m
- RealTimeEMG.m

2. Pokretanje „MVC_coeff_evaluation.m“ skripte

Ova skripta služi za određivanje MVC koeficijenta te ujedno i automatski pokreće „MVCMeasure.m“ skriptu. U skripti redom treba definirati sljedeće stvari:

1. Stream setup tab
`subject = 'Ime Prezime'` (string) – ime prezime osobe
`COM_Port = 'x'` (string) - COM port Shimmera3 iščitan u prvom dijelu
`duration_time = x` (int) - vrijeme trajanja mjerenja u sekundama
2. Shimmer3 setup tab
`fs = x` (int) – frekvencija uzorkovanja u Hz
`N = x` (int) – širina prozora (broj uzoraka)

Kao output dobiju se dva .dat filea:

- MVC_Coeff.dat – sadrži sve informacije + RAW podatke iz Shimmera
- MVC_Coeff_Raw_data.dat – sadrži samo RAW Podatke iz Shimmera



Te u figure-u se isplotaju usporedbe dvaju algoritma zaglađivanja i njihove MVC normalizacije uz spektar snage izmjerenog signala

3. Pokretanje „RealTimeEMG.m“ skripte

Ova skripta računa MVC normalizaciju u stvarnome vremenu uz prethodno izračunate koeficijente te ujedno te podatke prikazuje na grafu. U skripti potrebno definirati sljedeće stvari:

1. Shimmer3 setup Tab
`smoothing = 1 ili 2` (int) – gdje: 0 = movavg, 1 = rms

U command windowu ukucati sljedeću naredbu:

- `RealTimeEMG(comPort, x) – x` (int) - poželjno staviti što veći broj (npr 3000)

Skriptu je moguće zaustaviti pritiskom na „Stop Stream“ crvenu tipku koja se nalazi u zasebnom figure-u.

Kao output dobiju se pet .dat filea:

- `file_raw.dat` –RAW podaci iz Shimmera
- `file_filtered.dat` – filtrirani podaci
- `file_rectified.dat` – punovalno ispravljeni podaci
- `file_smoothed.dat` – podaci nakon smoothing algoritma
- `file_mvc.dat` – podaci nakon MVC normalizacije