

# *Kvantitativna analiza kinematskih parametara hoda kod zdravih osoba tokom obavljanja bazičnog zadatka*

## 1. Cilj projekta

Osnovni cilj projekta je osposobljavanje studenata za kvantitativnu obradu signala ostvarenih pri hodu čoveka dobijenih sa različitih vrsta senzora, upoznavanja sa pojmom polinomijalnog fitovanja, bazne linije i njenog uklanjanja, prebacivanja sistema iz lokalnog u globalni sistem i određivanja profila brzine i predjenog puta na osnovu dobijenih rezultata koji se kasnije mogu koristiti u daljem istraživanju neuromišićnog i skeletnog sistema, kao i primer etalona za posmatranje ostvarenog hoda zdravog ispitanika i ispitanika sa nekom bolešću (Parkinsonova bolest, Parkinsonova bolest + depresija, ...). Kao hardverski deo ovog projekta poslužio je SENSY bežični senzorni sistem sa periodom odabiranja  $F_s=100\text{Hz}$ . On se sastoji od tri vrste senzora: FSR senzora, akcelerometara i žiroskopa i ostale prateće opreme. Takođe, cilj ovog projekta je i proširivanje znanja u okviru programskog jezika Python koji je, kao jedan od vodećih programskih jezika, veoma bitan za nastavak studija i posao ubuduće.

**Predznanje:** Signali i sistemi, Sistemi i signali u organizmu, Praktikum iz merno-akvizicionih sistema

**Dodatna literatura:** Diplomski radovi studenata prethodnih godina na slično odabranu temu

## 2. Obrada signala

### Korišćenje modula

Za analizu dobijenih signala biće Vam potrebne razne funkcije. U programskom jeziku Python postoje zasebne biblioteke koje sadrže odgovarajuće funkcije u zavisnosti kakvi Vam tipovi podataka trebaju. Za Vaš dalji rad potrebno je nekoliko biblioteka koje se uključuju na sledeći način na početku programskog koda, u samom zaglavlju fajla:

<code>import numpy as np</code>	za mogućnost generisanja nizova i matrica
<code>import matplotlib.pyplot as plt</code>	za prikaz grafika određenih funkcija
<code>from scipy import integrate</code>	radi integraljanjenja određenih signala
<code>from scipy.interpolate import interp1d</code>	radi interpolacije
<code>import math</code>	radi upotrebe nekih matematičkih funkcija
<code>from numpy import matrix</code>	radi pravljenja inverzne matrice potrebne u daljem razvoju zadatka

## Učitavanje signala

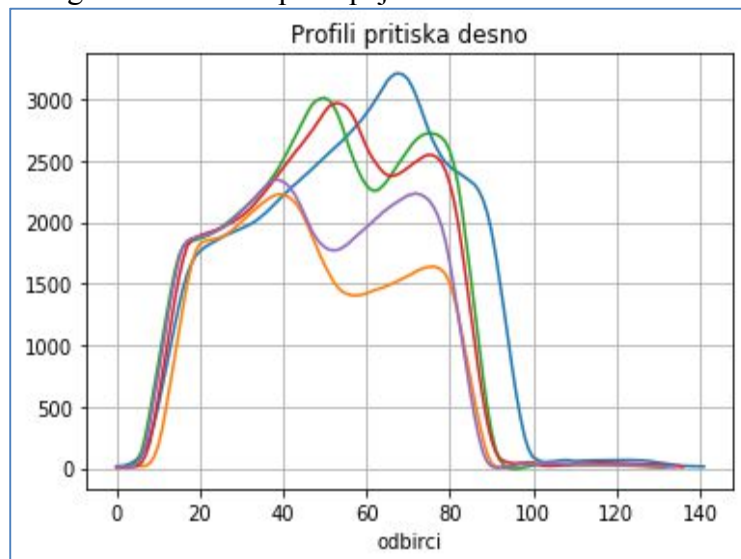
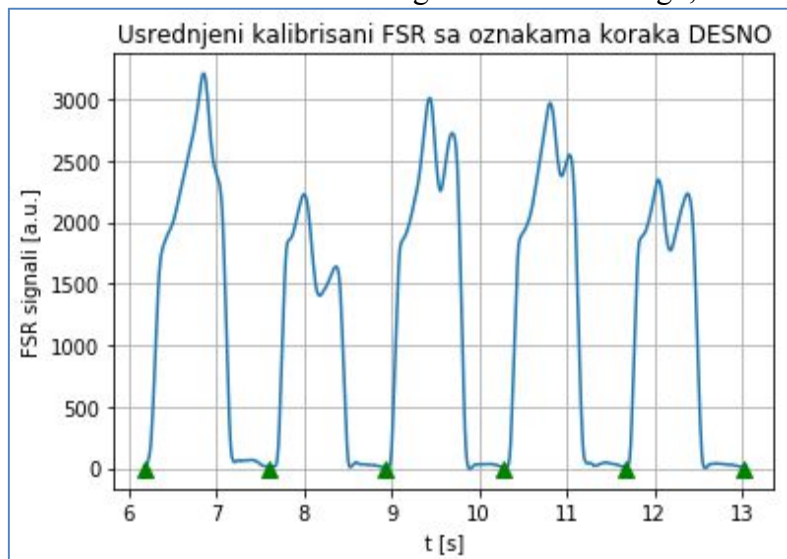
Najpre je potrebno pokrenuti program Python, odnosno otvoriti konzolu Spyder programskog okruženja i startovati novi projekat. Prethodno izmerenim signalima je uklonjeno zasićenje, izvršena kalibracija i filtriranje MA ( eng. *moving average* ) filtrom. Takvi signali su sačuvani u tekstualnim datotekama i potrebno ih je najpre pročitati kako bi se mogla vršiti dalja analiza. Učitavanja tekstualnih datoteka može se izvršiti uz pomoć funkcije ***np.loadtxt*** ("ime\_datoteke.txt") i tako pročitani signal smestiti u odgovarajuću promenljivu. Tekstualne datoteke date su u folderu gde se nalaze i ostala potrebna dokumenta za izradu zadatka.

## Obrada signala

### FSR signal

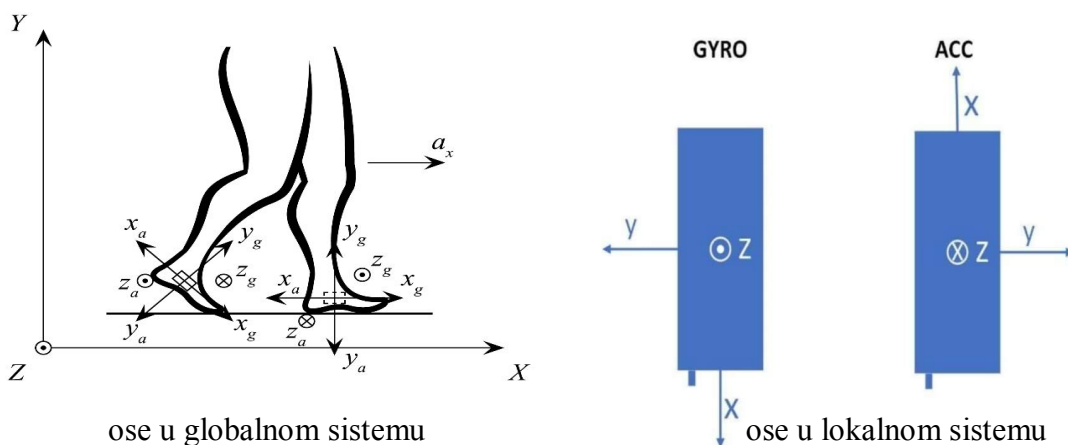
Obzirom da se tokom hoda pritisak prebacuje sa jednog dela stopala na drugi, signali dobijeni sa FSR senzora se mogu koristiti za određivanje karakterističnih događaja koji se javljaju tokom hoda, kao što su udarac petom ili odvajanje prstiju. Za određivanje ovih trenutaka potrebno je prvo usrednjiti signal za različite položaje senzora. Tako dobijene vrednosti signala treba porediti sa nekim pragom. Onaj period kada signal prelazi prag rastući predstavlja udarac petom jer se tada pritisak značajno povećava, a period kada signal prelazi preko praga u opadajućem slučaju odgovara odvajanju prstiju. Eksperimentalno je utvrđeno da se za prag uzima 20% maksimalne vrednosti signala i da ta vrednost daje pogodne rezultate. Pored toga, potrebno je odrediti i trenutke u kojima se javljaju minimumi usrednjenog signala jer su to trenuci u kojima posle udarca pete signal počinje da raste, a nakon odvajanja prstiju opada. Svaki udarac petom označava kraj jednog i početak novog koraka te je potrebno naći trenutke početaka i krajeva svakog koraka. Na osnovu trenutaka početaka i krajeva potrebno je naći trajanje svakog koraka kao i srednju vrednost trajanja koraka. Na graficima je potrebno prikazati usrednjene signale obe noge sa označenim počecima svakog koraka kao i početka i kraja samog hoda. Pored toga, potrebno je prikazati i ponašanje svakog koraka u zavisnosti od broja odbiraka i svaki korak prikazati na istom grafiku. Za izradu ovog dela zadatka koristiti funkcije ***max*** ( za određivanje maksimuma), ***potrebne petlje***, ***plt.plot( )*** i sve prateće komponente za označavanje osa grafika.(Slika 1.1)

Slika 1.1-Potrebni grafici za desnu nogu, za levu analogno treba dobiti poklapajuće rezultate

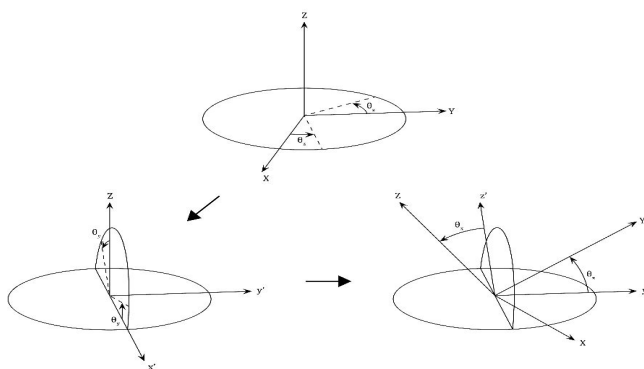


## Obrada signala sa akcelerometra i žiroskopa

Signali dobijeni sa akcelerometra i žiroskopa predstavljaju vrednosti ubrzanja i ugaone brzine u 3D lokalnom koordinatnom sistemu. Međutim, lokalni koordinatni sistem ne odgovara globalnom koordinatnom sistemu u kojem se ispitanik kreće stoga je potrebno izvršiti transformaciju koordinata iz lokalnog u globalni koordinatni sistem na koji način ubrzanje i ugaona brzina odgovaraju ubrzanju i brzini ispitanika u globalnom koordinatnom sistemu. Raspored osa u globalnom i lokalnom sistemu dat je na slici 1.2.



Slika 1.2



Potrebno je izvršiti prelazak iz lokalnog u globalni koordinatni sistem, odnosno invertovati odgovarajuće ose tako da se poklapaju sa osama globalnog koordinatnog sistema i izvršiti transformaciju uz pomoć kovarijacionih matrica za svaku osu.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \quad \text{- rotacija oko x-ose globalnog koordinatnog sistema}$$

$$R_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \quad \text{- rotacija oko y-ose globalnog koordinatnog sistema}$$

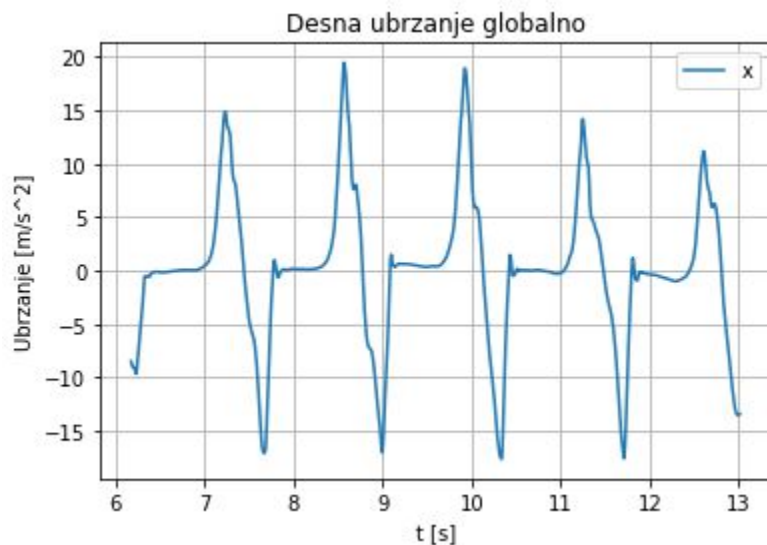
$$R_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{- rotacija oko z-ose globalnog koordinatnog sistema}$$

Najpre je potrebno integraliti ugaonu brzinu kako biste dobili ugao koji treba upotrebiti u kovarijacionim matricama. Nakon toga treba i integraliti ubrzanje kako biste dobili brzinu koja Vam je potrebna za dobijanje profila brzine i predjenog puta po x-osi. Za integraljenje koristiti funkciju **integrate.cumtrapz( )**. Medjutim, integraljenjem dolazi do pojave bazne linije koju je potrebno otkloniti. Za otklanjanje bazne linije pogodno je koristiti tehniku polinomijalnog fitovanja, odnosno treba izabrati pogodne tačke za interpolaciju. Odabir tačaka za uklanjanje bazne linije treba izvršiti tako da su u tim tačkama ubrzanje i ugaona brzina približno jednaki nuli, a to su tačke maksimuma FSR signala. Ove tačke su sasvim dovoljne kada je u pitanju ugao, medjutim, samo odabirom ovih tačaka nije moguće izvršiti interpolaciju brzine tako da se dobiju pogodni rezultati. Između već dobijenih tačaka za interpolaciju treba dodati i tačke koje odgovaraju središnjim tačkama između lokalnih maksimuma ugaone brzine čime se dobija znatno bolje rešenje. Nakon toga, potrebno je integraliti brzinu na isti način i dobiti predjeni put. Pošto je u interesu samo napredovanje ispitanika duž x-ose, treba naći profil brzine i predjenog puta samo za x-osu. Prikazati sve grafike zasebno i pogodno označiti ose.



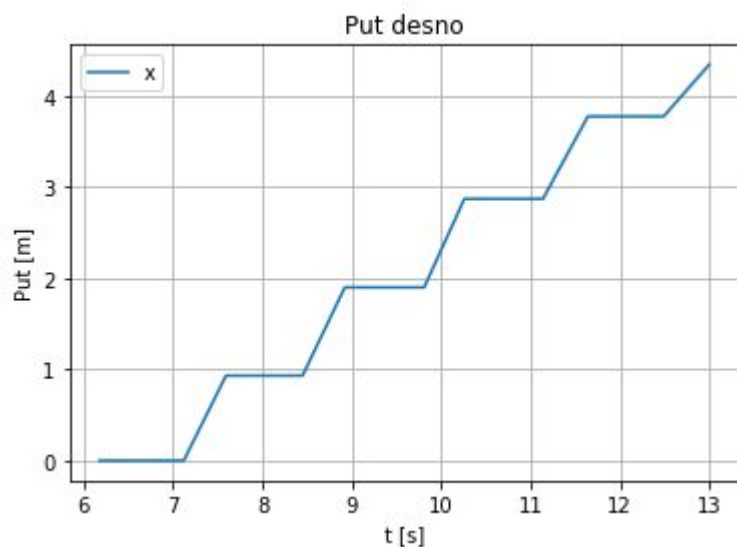
POTREBNI GRAFICI

Slika 1.3 - Nakon integracije ugaone brzine čime je dobijen ugao i otklonjena bazna linija za desnu nogu potrebno je dobiti sledeći grafik, a na sličan način treba dobiti i grafik za levu nogu



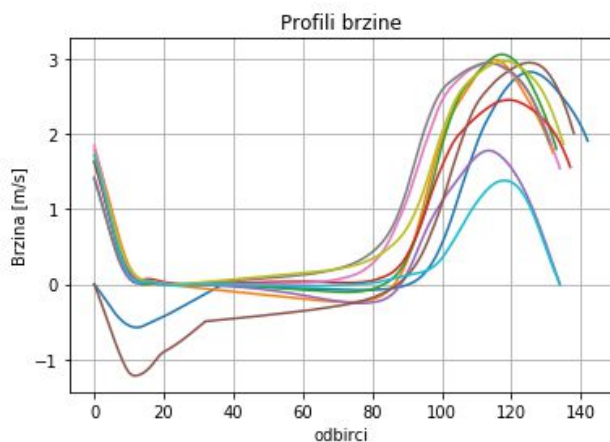
Slika 1.4 - Nakon prebacivanja ubrzanja iz lokalnog u globalni sistem uz pomoć goredobijenih uglova i kovarijacionih matrica izvršiti integraciju i otklanjanje bazne linije pri čemu za brzinu po x-osi treba dobiti sledeći grafik

Pošto treba posmatrati predjeni put ispitanika tokom celog hoda, smatrati da nakon integraljenja brzine nije potrebno skidati baznu liniju za predjeni put. Korišćenjem istih funkcija naći predjeni put i prikazati grafik predjenog puta (Slika 1.5).

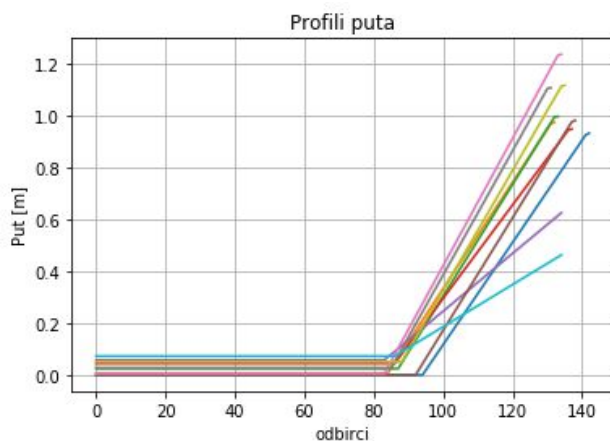


Slika 1.5 - Grafik predjenog puta za desnu nogu treba da izgleda ovako, analogno ovome naći predjeni put i za levu nogu i prikazati ga.

Na kraju zadatka potrebno je odrediti profile brzine i predjenog puta na osnovu dobijenih rezultata. Na jednom grafiku prikazati profile brzine za obe noge, a na sličan način na drugom grafiku prikazati i profile predjenog puta. Potrebno je ostvariti sledeće grafike (Slika 1.6 i 1.7) :



Slika 1.6 - Profili brzine za obe noge



Slika 1.7 - Profili predjenog puta za obe noge