

# Bakalauro baigiamasis darbas

## Parkinsono ligos eigos stebėjimo priemonė

Maksim Norkin

Vilniaus Gedimino technikos universitetas

Elektronikos fakultetas

Elektroninių sistemų katedra

maksim.norkin@ieee.org

### I. ĮVADAS. UŽDUOTIES ANALIZĖ

Darbo tema yra Parkinsono ligos eigos stebėjimo programa. Parkinsono liga yra dažniausiai pasitaikantis neurodegeneracinis judėjimo sutrikimas. Ankstyva ligos diagnozė ir efektyvus terapijos stebėjimas yra būtinas pacientų gydymui ir sveikatos priežiūros kainai sumažinti. Šiuo metu neegzistuoja gydytojų patvirtintos objektyvios ir vieningos vertinimo sistemos, kuri tiksliai atpažintų Parkinsono ligos simptomus. Vienas iš didžiausiai pasireiškiančių simptomų yra eisenos sutrikimas. Sutrikimo dažnumą ir svarbą patvirtina ir viešai prieinama duomenų bazė, kurioje yra pateikiami sveikų ir sergančių Parkinsono liga žmonių eisenos duomenys. Duomenų bazė vadinasi "PhysioBank".

Darbo tikslas yra sukurti programą, gebančią atpažinti jėgos jutikliais gautus signalus, priklausančius Parkinsono liga sergantiems pacientams. Programa bus paremta Matlab platforma. Tokia platforma buvo pasirinkta dėl didelių įrankių kiekių, kuris yra prieinamas Matlab aplinkoje. Taip pat sekančioje platformoje yra labai patogų ir greitą realizuoti signalų apdorojimo sistemas dėl jos architektūros - visi kintamieji yra matricos, kas yra labai patogus. Jėgos jutikliai parodo vertikalią žemės reakcijos jėgą (VŽRJ) (angl. Vertical Ground Reaction Force (vGRF)). Fizikoje, ir būtent biomechanikoje, VŽRJ nurodo kokia jėga žemė atsako į slėgiamą objektui. Kaip pavyzdžiui, stovinti žmogus slėgia žemę jėga, kuri lygi jo masei ir tuo pat metu, žemė slėgia žmogų priešinga, lygiai tokia pačia jėga. Parkinsono liga buvo pasirinkta dėl to, kad šiuo metu ji yra aktyviai tiriama, kadangi visuomenė moka labai didelius pinigus Parkinsono liga sergantiems asmenims. Per metus ši suma gali siekti 6 milijonus dolerių. Ligos rizika su amžiumi tik didėja, todėl analitikai prognozuoja, kad ateityje visuomenė mokės žymiai daugiau dėl žmonių populiacijos senėjimo.

Didžiausia ligos atpažinimo problema slypi savybių, kurios geriausiai atskiria sergantį Parkinsono liga nuo sveiko, nesergančio asmens. Darbo metu bus apžvelgtos kelios galimo savybės, kurios gali atskirti tokius pacientus. Blogiausias galimas variantas būtų nelinejinė funkcija atskirti didelių dimensijų duomenys.

Geriausias galimas variantas būtų mažos dimensijos duomenys (iki 3 dimensijų) ir liniškai atskiriami duomenys (kadangi linijinę funkciją realizuoti yra lengviausia). Turint didelių dimensijų duomenis, planuojamos sistemos aparatiniai reikalavimai automatiškai padidėja, kadangi būtina apdoroti labai daug duomenų. Tokią problemą galima išspręsti panaudojus dimensijų mažinimo algoritmus, kurių dažniausiai taikomi: Principinių komponentų analizė (angl. Principal Component Analysis (PCA)), linijinė diskriminanto analizė (angl. Linear Discriminant Analysis (LDA)). Metodai bus aptarti vėlesnėse skyriuose. Jie taip pat vadinami dimensijų praskyrimo algoritmais. Jie yra taikomi, kuomet duomenis yra atskirti netiesiškai. Duomenims, atskirtu tiesiniu dėsniu, galima taikyti paprastą klasifikavimo algoritmą. Jeigu duomenis taip ir nepavyksta atskirti tiesiškai, tenka taikyti kompleksinį klasifikatorių, ko pasekoje gali labai sumažėti klasifikavimo rezultatas. Darbe bus panaudotas Matlab platformoje įgyvendintas įrankis, kuris skirtas suprojektuoti naują dimensijų plokštumą, kurį įgyvendino Boitech Franc savo magistriniam darbe. Darbas buvo apgintas 2000 metais, Čekijos technikos universitete, Prahoje. Tiesinio sklaidimo dirbtinių neuronų klasifikatorių bus panaudotas iš Matlab "Neural Network Toolbox" įrankio. Likusi sistemos dalis bus aprašyta šiame darbe. Problema kyla pirminiam signalų apdorojimo pasirinkime: visą gaunamą signalą pilnai perduoti ar taikyti slankiojančio lango principą ar kitais, logiškai apibrėžtais būdais, apdoroti pirminį signalą.

Sukurtas produktas gebės pateikti paciento eisenos Parkinsono ligos eigos įvertinimą arba pateikti tikimybę kiek pacientas gali sirgti Parkinsono liga, neatsižvelgiant į kitus ligos simptomus: drebulys (rankų, kojų, žandikaulio, galvos), standumas (galūnių arba liemens sustingimas), bradikinezija (judesių lėtumas), pozicijos nestabilumas (arba sutrikęs balansas). Pati programa veiks nerealiu laiku. Tai reiškia, kad pirmiausiai duomenys bus surenkami, o vėliau įkeliami į programą tolimesniam apdorojimui.

Darbo tema, Parkinsono ligos eigos stebėjimo programa, reiškia, darbo rezultate bus sukurtas algoritmas, kuriuo bus parengta kompiuterinė programa. Pačiam kompiuteryje turi būti veikiantis Matlab programinis paketas. Programa bus rašoma Matlab 7.12.0 (R2011) versija su "Neural Network Toolbox" įrankiu. Eigų stebėjimas reiškia, kad visuomet egzistuoja neapibrėžtas, galimas programos netikslumas. Visiškai programa remtis, diagnozuojant Parkinsono ligą nėra galima, kadangi, kaip jau buvo minėta anksčiau - eigų sutrikimas nėra vienintelis ligos požymis. Turi būti atlikti ir kiti tyrimai, norint tiksliai diagnozuoti ligą.

Darbo objekto sudėtis yra asmeniniam kompiuteriui skirta programa, jėgos jutiklių signalų generavimo programa. Kokiam kompiuteriui programa bus rašoma, paminėta anksčiau. Jėgos jutiklių generavimo programa (modulis) bus atsakinga už signalų nuskaitymą iš duomenų bazės ir jų pateikimą sistemos algoritmui. Taip pat signalų generavimo programa (modulis) bus naudojamas programos demonstracinei versijai įgyvendinti.

Kaip buvo minėta, duomenys sistemai bus pateikiami iš "PhysioBank" duomenų bazės. Duomenys duomenų bazėje buvo surinkti diskretizuojant signalus 100 Hz diskretizavimo dažniu. Kiekvienu laiko momentu, yra įrašoma nauja eilutė į duomenų tekstinę bylą. Eilutę sudaro 19 skilčių:

- Skiltis 1 nurodo laiką (sekundėmis);
- Skiltys 2-9 nurodo kairės kojos 8 jutiklių VŽRJ, Niutonais;
- Skiltys 10-17 nurodo dešinės kojos 8 jutiklių VŽRJ, Niutonais;

- Skiltis 18 nurodo kairės kojos suminę VŽRJ, Niutonais;
- Skiltis 19 nurodo dešinės kojos suminę VŽRJ, Niutonais;

Duomenų bazės bylų pavadinimai, pavyzdžiui: "GaCo01\_02.txt" ar "JuPt03\_06.txt", sudaryti nurodant duomenų rinkimų sesijų pavadinimus: "Ga" - "Galit Yogev et al" (dual tasking in PD; Eur J Neuro, 2005), "Ju" - "Hausdoff et al" (RAS in PD; Eur J Neuro, 2007), "Si" - "Silvi Frenkel-Toledo et al" (Treadmill walking in PD; Mov Disorders, 2005). Toliau, "Co" nurodo kontrolinį subjektą arba nesergantį Parkinsono liga subjektą, "Pt" nurodo Parkinsonu sergantį subjektą. Pirmas numeris nurodo subjekto identifikacinį numerį, po brūkšnio einantis antras numeris nurodo subjekto duomenų rinkimo seką.

Programos veikimas bus vertinamas taiklumu ir jautrumu. Parametrai yra apskaičiuojami iš pasiklovimo matricos. Taiklumas apskaičiuojamas:

$$Taiklumas = \frac{T_P + T_N}{T_P + F_P + T_N + F_N}, \quad (1)$$

kur  $T_P$  - teisingai identifikuotų klasių skaičius,  $T_N$  - teisingai atmestų klasių skaičius,  $F_P$  - klaidingai identifikuotų klasių skaičius ir  $F_N$  - klaidingai atmestų klasių skaičius.

Tikslumas apskaičiuojamas:

$$Tikslumas = \frac{T_P}{T_P + F_P} \quad (2)$$

Programos veikimas grafiškai demonstruojamas kreive, kuri nurodo subjekto tikimybę sirgti Parkinsono liga.

## II. INFORMACINIŲ PARKINSONO LIGOS DIAGNOSTIKOS SISTEMŲ APŽVALGA

Šiame skyriuje bus apžvelgtos analoginės informacinės sistemos arba bandymai sukurti sistemą, kuri, remiantis įvairių jutiklių pagalba, gebėtų atpažinti Parkinsono ligą.

Pirmas darbas, kuris bus analizuojamas, yra "Biometrinė ir mobili eisenos analizavimo sistema ankstyvai Parkinsono ligos diagnozei ir terapijai".

Antras darbas, ...

Trečias darbas, ...

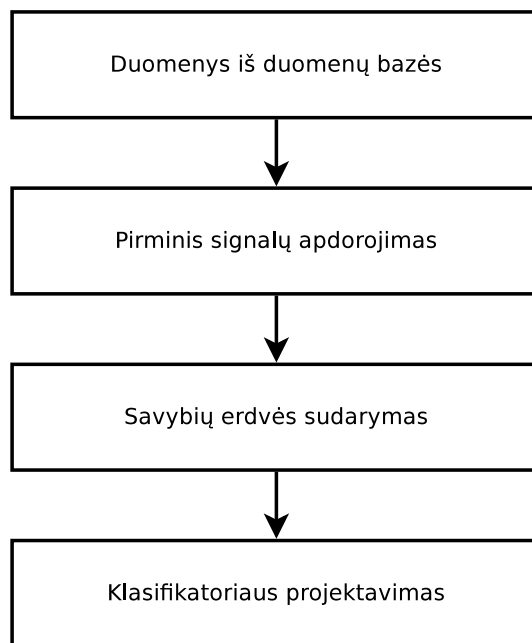
Ketvirtas darbas, ...

## III. SIGNALŲ ANALIZĖS PROGRAMOS KŪRIMAS

Šiame skyriuje apžvelgsime kylančias problemas, kuriant signalų analizės programą ir galimus jų sprendimus. Kiekvienas sprendimas bus išnagrinėtas, apžvelgta jo metodika, taikymo problemos bei rezultatai.

### A. Bendros programos struktūrinės schemos sudarymas

Priklausomai nuo taikomos metodikos, programos struktūrinės schemos skiriasi, kadangi vienas sprendimas reikalauja vieno tipo duomenų struktūros, kitas - kitos struktūros. Programos projektavimo metu buvo



1 pav.. Pirminė programos schema.

išbandyti keli programos variantai ir suprastinus veikimo schemas, kiekvienas iš programos variantų veikė pagal bendrą schemą, kurią galima apibrėžti 1 pav. Reikalaujama, kad kiekvienas iš algoritmų blokų būtų nepriklausomas nuo žemiau ar aukščiau einančio bloko, kas reiškia, kad "Pirminio signalų apdorojimo" bloko algoritmo pakeitimas neturi turėti jokios įtakos po jo einančiam "Savybių erdvės sudarymas" blokui.

Kiekvieno bloko užduotis:

- 1) "Duomenys iš duomenų bazės" bloko užduotis yra nuskaityti iš naudojamos "PhysioBank" duomenų bazės duomenis ir juos įkrauti į Matlab aplinką. Kadangi duomenys pateikiami tekstiniu pavidalu, Matlab aplinkoje yra labai patogi funkcija tokiems duomenims nuskaityti - *dlmread*.
- 2) "Pirminis signalų apdorojimas" bloko užduotis yra pradinis signalų filtravimas, nuolatinės komponentės pašalinimas. Taip pat, į šį bloką įeina ir atskirų signalų išskyrimas iš bendrai gaunamo signalo: signalo išskyrimas kuomet koja turi susilietimą su žeme, signalo išskyrimas, kuomet koja praranda susilietimą su žeme. Į bloką taip pat įeina ir slankiojančio lango algoritmo taikymas.
- 3) "Savybių erdvės sudarymas" bloko paskirtis yra pateikti pasirinktam klasifikatoriui išskirtas signalų savybes po pirminio signalo apdorojimo. Tokiomis savybės gali būti signalų dažninės komponentės (Furjė koeficientai), koreliacijos koeficientai tarp dešinės ir kairės kojos, savi-koreliacijos koeficientai dešinės ar kairės kojos. Detaliau nagrinėjamas savybes bus aptartos tolimesniuose poskyriuose. Į bloką taip pat įeina ir darbas su dimensijomis - dimensijų skaičiaus mažinimas ar naujos erdvės paieška, kurioje nagrinėjami duomenys yra geriau tiesiškai atskiriami.
- 4) "Klasifikatoriaus projektavimas" bloko paskirtis yra klasifikatoriaus apmokymas ir jo testavimas. Kla-

sifikatoriaus pasirinkimas yra labai svarbus klausimas nagrinėjamam darbe. Jis turi būti parinktas argumentuotai, pateikiant detalius klasifikavimo rezultatus, remiantis rezultato grafiku. Nepasitikėjimo matrica, bei iš jos apskaičiuotais įverčiais: taiklumu bei tikslumu. Dar svarbus klasifikatoriaus aspektas yra savybių erdvės bendro dėsnio radimas arba jo aproksimacija. Šį faktą galima apibūdinti iš klasifikavimo grafiko. Jeigu klasifikavimo rezultatas yra nestabilus, ilgą laiką neturintis pastovaus rezultato, vadinasi, galima teigti, kad klasifikatorius blogai atliko erdvės dėsnio aproksimavimą, jis nėra tinkamas nagrinėjamai savybių erdvei.

Tolimesniuose poskyriuose bus aptarti įgyvendinti blokinės struktūros elementai.

### B. Pirminio signalų apdorojimo programos kūrimas

Šiame skyriuje bus aptartas pirminio signalų apdorojimo programos kūrimas. Aptartas duomenų nuskaitymas iš duomenų bazės bylos tekstiniam pavidalė, galimas signalo išskaidymas slankiojančio lango metodu arba signalo formos nuskaitymas, priklausomai nuo žinomo fizinio poveikio, kuriuo metu signalas buvo gautas.

Programos kūrimo patogumui, buvo parinkta direktorių architektūra:

- Tėvinė direktorija
  - <programos versija, nurodyta datos formatu>
  - database
  - cache

Kadangi programos kūrimo metu yra svarbu saugoti ankstesnes programos versijas, buvo parinktas kasdienis programos versijos saugojimas: kiekvienos pradžioje darbas buvo pradedamas su ankstesnės dienos kopija. Taip buvo išsaugotas kiekvienos darbo dienos programos versija ir taip galima peržiūrėti kokių analitinių kelių buvo eita prie dabartinės programos versijos.

Pirmoji funkcija, priklausanti pirminio signalo apdorojimo programai yra *read\_data*. Funkcijos kodas pateikiamas 2 pav.

Funkcijai užtenka nurodyti tik norimos nuskaityti bylos pavadinimą, kaip pavyzdžiui "SiPt30\_01.txt" ir funkcija gražins kairės kojos duomenis. Duomenų analizėje naudojami tik vienos kojos duomenis, kadangi dešinės ir kairės kojos duomenys yra stipriai koreliuoti [1], todėl nėra prasmės naudoti dviejų kojų duomenų. Pasirinkus tokį sprendimą taip pat yra sumažinamas skaičiavimų skaičius programoje. Duomenų bazės direktorija nurodyti kintamojo vietoje yra taip pat svarbus aspektas, kadangi pakitus duomenų bazės vietai, užteks tik pakeisti vieną kintamąjį, o ne visą kodą, nors kodas ir nėra ilgas. Matlab funkcija *dlimread* gražina duomenis masyvo pavidalu, kur stulpelis nurodo įvade aptartus duomenis, o eilutė nurodo duomenų vertes tam tikru laiko momentu.

Sekantis žingsnis, po duomenų nuskaitymo, yra jų pirminis apdorojimas. Nagrinėjimas bus pradėtas nuo slankiojančio lango metodo. Programos kodas pateikiamas 3 pav. Funkcijos įvestyje pateikiamas nagrinėjamas signalas, slankiojamo lango ilgis ir slankiojamo lango perdanga. Slankiojamo lango perdanga nurodo

```

1 function data = read_data( file_name )
2 %READ_DATA Function reads the data from file
3 %
4 %   Input:
5 %       - file_name the name of the file
6 %
7 %   Output:
8 %       - data the Left foot data
9
10 database_directory = '../database/';
11
12 % Check if file exists
13 if ~exist( [ database_directory file_name ], 'file' )
14     % Problem arise
15     error( [ 'File_' file_name '_does_not_exist' ] );
16 end
17
18 data_raw = dlmread( [ database_directory file_name ] );
19
20 % We'r interested only in left foot data
21 data = data_raw(:,19);
22
23
24 end

```

2 pav.. Duomenų nuskaitymo funkcija iš tekstinės duomenų bylos.

```

1 function output = split_data( input, window_length, window_overlay )
2 %SPLIT_DATA Function to split data into windows
3
4 % Buffer to store the signal information
5 buffer = [];
6 output = [];
7
8 for i=1:length(input)
9     % if buffer reached the length of the window
10    if ( length(buffer) >= window_length )
11        % store the buffer to output signal
12        output = [ output; buffer ]; %%ok<AGROW>
13        % Do the overlay step
14        if ( window_overlay ~= 0 )
15            buffer = buffer(window_overlay:end);
16        end
17    else
18        % otherwise, populate the buffer with new entry
19        buffer = [ buffer input(i) ]; %%ok<AGROW>
20    end
21 end
22
23 end

```

3 pav.. Slankiojančio lango algoritmo pritaikymas.

reikšmių arba laiko verčių kiekį, kurį algoritmas pašalina iš spartinančiosios atminties, laukdamas naujų verčių langui užpildyti. Pavyzdžiui, jeigu lango ilgis yra 4 signalo verčių, o perdanga - 2 signalo vertės, vadinasi, kai algoritmas užpildys langą 4 signalo vertėmis, esamą langą jis perduos į išėjimo spartinančiąją atmintį, paskutines dvi vertes ištrins iš spartinančiosios atminties ir iš naujo lauks naujų dviejų reikšmių langui pilnai užpildyti.

Funkcija yra tiek lanksti, kad nėra svarbu kokio tipo duomenys yra pateikiami - ar tai konkretaus sig-

```

1      [B,A] = butter(9, 1/50, 'high');
2      [BB,AA] = butter(9, 40/50, 'low');
3      output = filter(BB, AA, filter(B, A, input));

```

4 pav.. Signalų filtravimas dviem Butterworth filtrais.

```

1  for i=1:size( Pt_t, 2 )
2      fprintf(' ');
3      signal = Pt_t{i};
4      buffer = [];
5      k = 0;
6      for j=1:size(signal, 1)
7          if ( signal(j) ~= 0 )
8              buffer = [ buffer signal(j) ]; %#ok<AGROW>
9          else
10             if ~isempty(buffer) && length(buffer) > 10 && length(buffer) < 200
11                 % Store the buffer
12                 data.Pt{ length(data.Pt)+1 } = buffer;
13                 % Clear the buffer
14                 if ( j - k > 50 && j - k < 200 )
15                     data.Pt2( length(data.Pt2)+1 ) = j - k;
16                 end
17                 k = j; % Start of stance
18                 buffer = [];
19             else
20                 buffer = [];
21             end
22         end
23     end
24 end

```

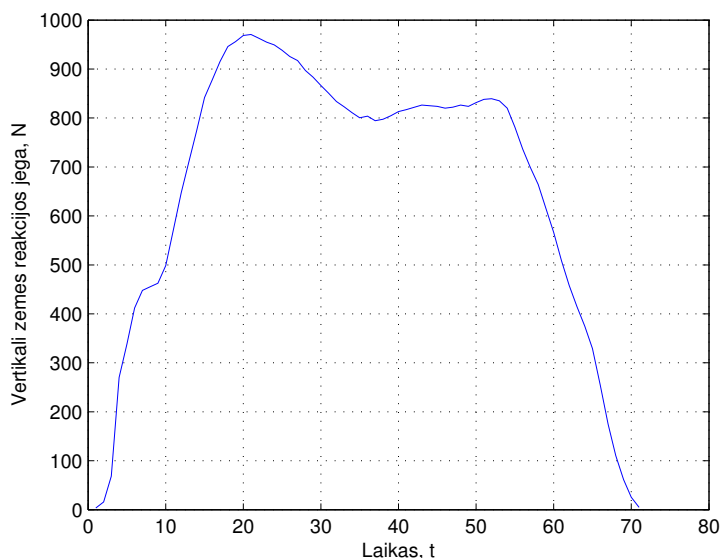
5 pav.. Kontakto su žeme signalo išskyrimo programos kodo fragmentas.

nalo vertės ar iš ankščiau pritaikyto slankiojančio lango algoritmo išskirtos signalo savybės, kuriuos buvo panaudotos formuojant naują signalą. Tokia funkcijos savybė bus labai naudinga tolimesniame darbe.

Sekanti programos dalis atlieka paprastą signalo filtravimą su dvejais *Butterworth* skaitmeniniais filtrais. Pirmasis, aukštų dažnių filtras, skirtas pašalinti signalo nuolatinę komponentę. Antras, žemų dažnių filtras, skirtas pašalinti aukšto dažnio triukšmą, kuris neneša visiškai jokios naudingos informacijos. Filtras įgyvendinamas labai paprastai. Kodo pavyzdys pateikiamas 4 pav. Abiejų filtrų eilė yra 9-ta, žemų dažnių filtro ribinis dažnis parinktas 40 Hz. Duomenys diskretizuojami 100 Hz dažniu, vadinasi didžiausias galimas signalo dažnis yra 50 Hz. Didžiausias žmogaus generuojamas dažnis ėjimo metu, remiantis šaltiniu, yra 20 Hz. Užtikrintumui buvo parinktas 40 Hz dažnis. Nuolatinė dedamoji pašalinama su aukšto dažnio filtru, kurio ribinis dažnis yra 1 Hz. Nuolatinė dedamoji neneša jokios informacijos apie eisena, kadangi ji tiktais nurodo naudojamų jutiklių jautrumą.

Toliau seka, priklausomai nuo pasirinkto pirminio signalų apdorojimo būdo, signalo išskyrimas pagal fizinę veiklą. Dominančios signalo būsenos yra, kai subjekto koja turi kontaktą su žeme ir nurodytas subjektas neturi kontakto su žeme. Kontakto su žeme signalo išskyrimui programos kodas yra pateikiamas 5 pav.

Pirmiausiai, "Pt\_t" kintamojo struktūroje yra saugoma kairės kojos Parkinsono liga sergančių subjektų duomenys. Kiekvienas signalas yra priskiriamas prie *signal* kintamojo, su kuriuo toliau yra tęsiamas ap-

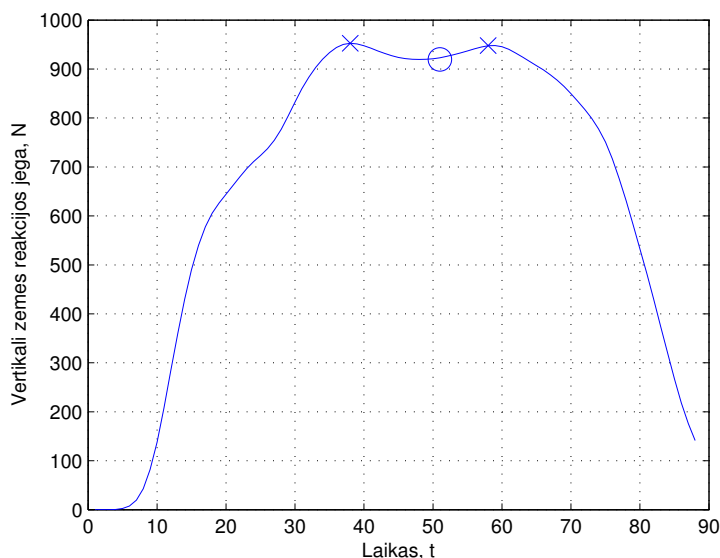


6 pav.. Susilietimo su žeme signalas.

dorojimo procesas. Jeigu signalas nėra lygus nuliui, tuomet jis įdedamas į laikinąją atmintį. Taip signalas yra tikrinamas iki tol, kol signalas tampa lygus nuliui ir tęsiamas tolimesnis apdorojimas. Apdorojimas susideda iš signalo ilgio patikros. Jeigu signalas yra trumpesnis už 10 verčių, arba turint omenyje, kad signalas buvo diskretizuojamas  $100\text{ Hz}$ , tai  $0.1\text{ s}$ , vadinasi, signalas yra tiesiog aukšto dažnio triukšmas arba blogas pavyzdys ir signalas yra atmetamas. Jeigu signalas yra ilgesnis už 200 verčių ( $2\text{ s}$ ), vadinasi, duomenų rinkimo metu įvyko klaida ir koja per tokį laiką pakelta nebuvo. Tokia klaida gali būti sukelta, kuomet subjektas ne eina, o stovi ant dviejų kojų arba tik ant kairės kojos. Jeigu visi kriterijai patenkinami, signalas yra priimamas ir kraunamas į laikinąją atmintį, pavadinimu "*data.Pt*". Signalas, kuriuo metu koja neličia žemės, yra randamas kartu su ankščiau išnagrinėtu metu. Algoritmas patikrina kiek praėjo laiko (arba kiek verčių yra priskaičiuota) nuo paskutinio užskaityto kojos ant žemės signalo ir įrašo tą signalą į laikiną atmintį, jeigu po praeito signalo nepraėjo mažiau negu 50 verčių arba  $0.5\text{ s}$  ir ne daugiau nei 200 verčių arba  $2\text{ s}$ . Kojos signalas yra pateiktas 6 pav.

Dar vienas pirminis signalo apdorojimas, kurį teko panaudoti tiriant galima bendrą erdvę - kiekvieno kojos susilietimo su žeme signalo dviejų globalių maksimumų ir vieno lokalaus minimumo paieška. Algoritmo rezultatas yra pateiktas 7 pav. Kryžiais pažymėtos globalaus maksimumo vietos, apskritimu pažymėta lokalaus minimumo signalo vieta. Išanalizavus daugumos subjektų žingsnio signalus, buvo padaryta išvada, kad kiekviename žmogaus žingsnyje egzistuoja du maksimumai. Vienas maksimumas randamas, kai subjektas yra visiškai atsirėmęs galine pėdos dalimi į žemę, antras maksimumas randamas, kai subjektas visiškai atsiremia priekine pėdos dalimi. Tarp šių dviejų maksimumų yra pereinamasis laikotarpis, kuris yra signalo lokalus minimumas. Priežastis, dėl kurios algoritmas buvo įgyvendintas, yra vienas darbas, kuriame buvo





7 pav.. Dviejų globalių maksimumų ir vieno lokalaus minimumo radimas signale.

```

1 for i=1:size(signal_window, 2)
2     buffer = [ buffer signal_window(i) ]; %#ok<AGROW>
3     if length(buffer) > 2
4         delta = buffer(i) - buffer(i-1);
5         if delta < d_pr
6             % Falls
7             buffer(i) = buffer(i-1);
8         end
9
10        if buffer(i) == buffer(i-1) && right_index == 0 && right_max == 0 && buffer(i) > min_value
11            right_index = i-1;
12            % Return the maximum in the buffer
13            right_max = max(buffer);
14        end
15    end
16 end

```

8 pav.. Dviejų globalių maksimumų ir vieno lokalaus minimumo radimo algoritmo fragmentas.

pasiūlyta naudotis būtent tokiais savybėmis atpažinti Parkinsono liga sergančius subjektus nuo nesergančių subjektų. Pačio algoritmo kodo dalis yra pateikta 8 pav.

Programos kodas krauna gaunamą signalą į laikinąją atmintį ir tikrina ar signalas pakito per užduotą dydį  $\delta$ . Delta nurodo kiek signalas turi pakisti, kad algoritmas nuspręstų, kad signalas pradėjo mažėti. Tokio tikrinimo priežastis yra ta, kad prieš globalų maksimumą taip pat yra sritis, kurioje signalas kilimas sulėtėja, o kai kuriuose pavyzdžiuose buvo pastebėta, kad signalas net pradeda mažėti. Dėl šios priežasties buvo įvesta pokyčio tikrinimo sąlyga.

Pirminio signalų apdorojimo programa baigiasi nurodytais metodais. Sekantis žingsnis yra apdorotų duomenų perdavimas tolimesnei programos pakopai - požymių išskyrimui.

### C. Požymių išskyrimo programos kūrimas

Požymių išskyrimas priklauso nuo pirminio signalo apdorojimo mechanizmo. Jeigu ankstesniame bloke pakeisime ribinį filtro dažnį ir kaip nagrinėjamą požymį bus parinktas koreliacijos koeficientai ar dažninės komponentės - požymio vektorius pakis, todėl prieš kiekvieną požymių analizavimo ciklą bus pateikta ir pirminiam signalų apdorojimo bloke naudojami metodai. Taip pat, kadangi požymių išskyrimo programos kūrimas taip pat sprendžia ir dimensijų klausimą - skyriaus pradžioje bus trumpai apžvelgti populiariausi šiame darbe panaudoti dimensijų mažinimo metodai.

1) *Dimensijų mažinimas*: Dimensijų mažinimas suteikia dideles galimybes supaprastinti klasifikavimo uždavinį. Kuomet duomenys savybių erdvėje nėra linijiškai atskirti, yra sunku nustatyti ar klasifikatorius ras apibendrintą duomenų praskyrimo funkciją. Norint užtikrinti, kad klasifikatorius veiks teisingai - yra atliekamas dimensijų praskyrimas. Metodas taip pat taikomas, kuomet norima sumažinti dimensijų skaičių. Problema iškyla, kuomet norima įgyvendinti sukurta sprendimą įterptinėje sistemoje, kurioje negalima pasiekti didelių aparatinių resursų. Turint didelių dimensijų duomenis taip pat kyla ir pačios sistemos kaina, dėl tų pačių aparatinių resursų reikalavimų. Norint išvengti tokių problemų - taikomi dimensijų mažinimo algoritmai. Populiariausi dimensijų mažinimo algoritmai yra:

- Linijinė diskriminanto analizė (angl. Linear Discriminant Analysis (LDA));
- Principinė komponentių analizė (angl. Principal Component Analysis (PCA)).

Taip pat, kiekvienas iš paminėtų būdų taip pat gali būti papildytas branduolio (angl. Kernel) funkcija. Metodų apibūdinimas bus pradėtas nuo paprasčiausios iš dviejų - PCA.

Principinė komponentių analizė yra statmena ir paprasta transformacija, kuri yra daug kur naudojama dėl paprastos matematinės pusės ir lengvo įgyvendinimo. Egzistuoja skirtingi PCA metodo įgyvendinimo algoritmai. Vienas iš jų yra vienetinės reikšmės skaidymas (angl. Singular Value Decomposition (SVD)). Literatūroje PCA ir SVD dažniausiai minimi kaip sinonimai. SVD yra greitas, tačiau didelių atminties resursų reikalaujantis metodas. Kuomet dideli atminties resursai nėra prieinami (įterptinėje sistemoje), reikia naudoti kitą įgyvendinimo variantą - naudoti tikrinių vektorių skaidymą. Toks sprendimas užima didesnius laiko resursus, lyginant su SVD, tačiau jam reikia mažesnių atminties resursų ir jis tinka nagrinėti didelių dimensijų duomenis. Toks sprendimas bus taikomas ir šiame darbe. PCA įgyvendinimas naudojant SVD ir tikrinių vektorių skaidymą yra pateiktas priede. Toliau bus išnagrinėtas PCA iš matematinės pusės.

Matematiškai, PCA apibrėžiamas kaip statmena linijinė transformacija, kuri transformuoja duomenis į naują koordinačių sistemą, kurioje didžiausią variaciją projektuojama į pirmą ašį, antroji didžiausia variacija (statmena pirmajai) projektuojama į antrą ašį. Procesas tęsiasi tol, kol yra pasiekiamas norimas dimensijų skaičius.

Apibrėžta duomenų matrica  $\mathbf{X}^T$ , su empiriniu vidurkiu, kuris lygus nuliui (empirinis vidurkis reiškia vidurkio apskaičiavimą ir jo atėmimą iš duomenų), kur kiekviena eilutė  $n$  atspindi duomenų rinkinį, o stulpelis  $m$  - turimas duomenų dimensijas. Matricos  $\mathbf{X}$  vienetinės reikšmės skaidymas išreiškiamas:

$$\mathbf{X} = \mathbf{W}\Sigma\mathbf{V}^T, \quad (3)$$

kur  $m \times n$  matrica  $\mathbf{W}$  yra matricos  $\mathbf{X}\mathbf{X}^T$  tikrinių vektorių matrica,  $\Sigma$  matrica yra  $m \times n$  stačiakampio įstrižainės matrica su realiais skaičiais įstrižainėje ir  $n \times n$  tikrinių vektorių  $\mathbf{X}^T\mathbf{X}$  matrica  $\mathbf{V}$ . PCA transformacija, kuri sukonstruoja naujas dimensijas, yra apibrėžiama:

$$\mathbf{Y}^T = \mathbf{X}^T\mathbf{W} = \mathbf{V}\Sigma^T\mathbf{W}^T\mathbf{W} = \mathbf{V}\Sigma^T. \quad (4)$$

Kadangi  $\mathbf{W}$  yra statmena matrica, kiekviena  $\mathbf{Y}^T$  eilutė yra matricos  $\mathbf{X}^T$  eilutės sukimas. Pirmasis  $\mathbf{Y}^T$  stulpelis yra pirmosios komponentės rezultatas, antras stulpelis yra antrosios komponentės rezultatas. Kiek stulpelių  $\mathbf{Y}^T$  turi, tiek ir rezultatų (dimensijų) yra po transformacijos.

Linijinė diskriminanto analizė yra dimensijų mažinimo metodas (kuris kartu yra naudojamas kaip ir klasifikatorius), yra vienas iš metodų, kuris neša kartu ir duomenų žymėjimo informaciją. Tai reiškia, kad mažinant dimensijų skaičių, metodas turi žinoti kokie duomenys priklauso kokiai duomenų klasei. Pagrindinis LDA tikslas yra didinti sekantį kriterijų:

$$\mathfrak{J}(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}, \quad (5)$$

kur  $S_B$  yra "išorinės klasės scatter matrica",  $S_W$  yra "vidinė klasės scatter matrica". Scatter matricų apibrėžimas yra:

$$S_B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T \quad (6)$$

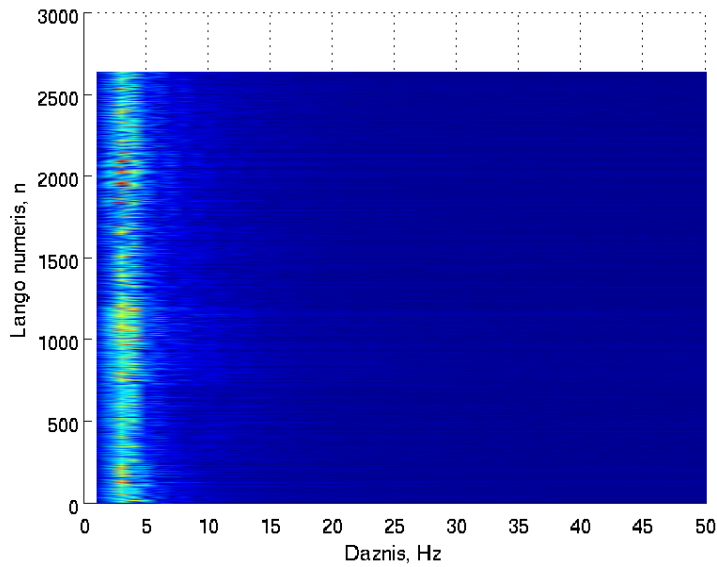
$$S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \quad (7)$$

Branduolio metodas yra labai lengvai paaiškinamas pavyzdžiu [2]. Tarkim, egzistuoja tam tikri empiriniai duomenys:

$$(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}. \quad (8)$$

Čia,  $\mathcal{X}$  yra netuščia sritis, iš kurios imami  $x_i$  duomenys. Tikslu arba žymekliu vadinami yra  $y_i \in \mathcal{Y}$  (sistemos atsakas),  $i, j \in [n]$  žymimas eilės, identifikavimo numeris, kur  $n := \{1, \dots, n\}$ .

Reikia pastebėti, kad sričiai  $\mathcal{X}$  nebuvo suteikta jokių apribojimų. Norint išspręsti apmokymo problemą reikia papildomos struktūros. Mašiniame apmokyme siekiama apibendrinti neturimus duomenis. Binarinio klasifikavimo atveju, turint naujus duomenis  $x \in \mathcal{X}$ , norima nuspėti jų žymeklį  $y \in \{\pm 1\}$ . Paprastai kalbant, norima pasirinkti tokį  $y$ , kuris labiausiai atspindėtų apmokymo metu naudotas  $(x, y)$  poras. Tam įgyvendinti



9 pav.. Kontrolinių subjektų kairės kojos žingsnių signalų dažninės komponentės

reikalingi kuo panašesnis duomenys į  $\mathcal{X}$  ir  $\mathcal{Y}$ . Pastarąjį palengvina tai, kad spėjimas gali būti arba identiškas arba atvirkščias. Formaliai yra reikalinga tokia funkcija:

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, \quad (x, x') \rightarrow k(x, x'), \quad (9)$$

kuri tenkinta tokias sąlygą visiems  $x, x' \in \mathcal{X}$ :

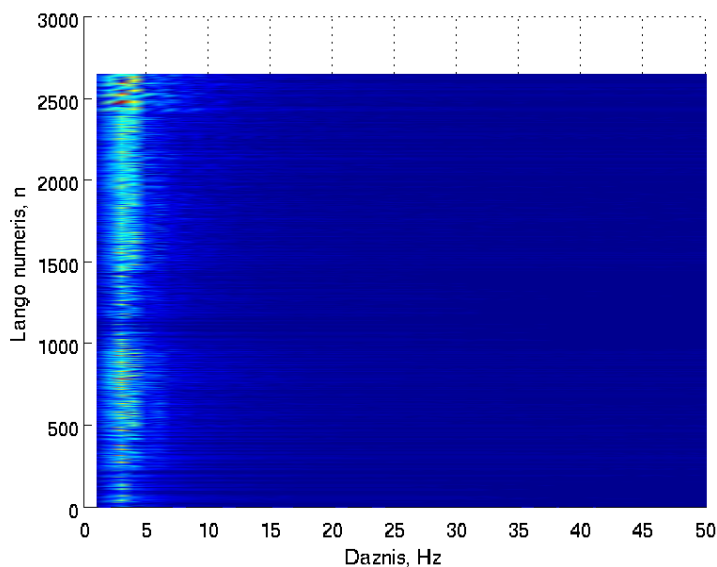
$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle, \quad (10)$$

kur  $\Phi$  projektuota taškus į Hilberto plokštumą  $\mathcal{H}$ , kai kada vadinama savybių erdvė. Vienodumo matmuo  $k$  dažniausiai vadinamas branduoliu, o  $\Phi$  vadinama savybių projekcija.

2) *Savybių tyrimas*: Šiame skyriuje bus išnagrinėtos galimos signalų savybės, pagal kurias galima atskirti Parkinsono liga sergantį subjektą nuo sveiko subjekto. Kaip buvo minėta anksčiau, ne visos savybės gali būti išskirtos, naudojantis vienu pirminio signalo apdorojimo mechanizmu, todėl kiekvienos savybės nagrinėjimo pradžioje bus paminėta ir pirminio signalo apdorojimo bloko sudėtis.

Galimų savybių analizavimas gali būti pradėtas nuo signalo dažninių komponentių (Furjė transformacijos). Kaip nurodo šaltiniai, Parkinsono liga sergančių subjektų žingsniai turi statesnius šlaitus, kas iš signalų apdorojimo srities reiškia, kad signalas turi turėti aukštas dažnines komponentes. Šiai savybei išskirti pirmiam signalo apdorojimo bloke bus panaudotas slankiojančio lango metodas. Taip kiekvienas signalas turės fiksuotą ilgį ir taip bus galima lyginti signalus tarpusavyje.

Toliau bus išanalizuoti Parkinsono liga sergančių subjektų ir kontrolinių subjektų žingsnių signalų daž-



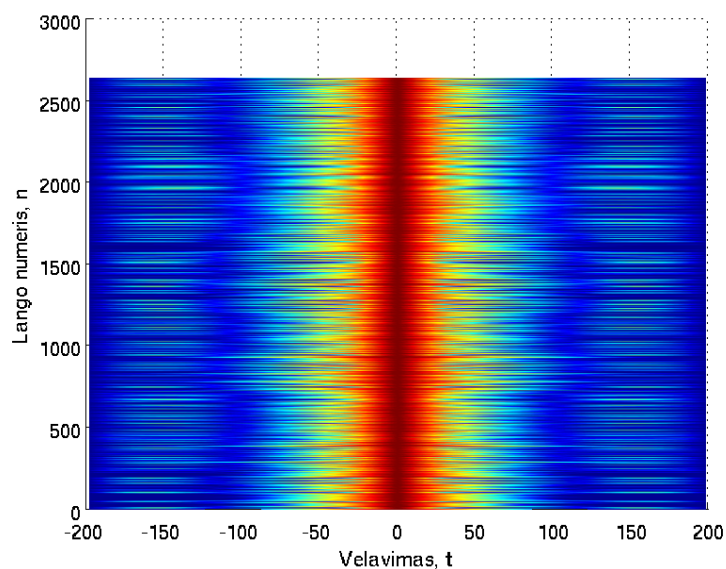
10 pav.. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų dažninės komponentės

ninės komponentės. Kontrolinių subjektų kairės kojos žingsnių dažninės komponentės yra pateiktos 9 pav. Parkinsono liga sergančių subjektų kairės kojos žingsnių dažninės komponentės yra pateiktos 10 pav. Kaip matoma iš duotų komponentių grafikų - tiek kontrolinių subjektų, tiek Parkinsono liga sergančių subjektų pagrindinės dažninės komponentės išsidėsto iki  $5\text{ Hz}$  ruože. Ties  $0\text{ Hz}$  dažninių komponentių nėra, kadangi jos buvo pašalintos filtro pagalba. Už  $10\text{ Hz}$  ribos, komponentės neneša visiškai jokios informacijos. Iš to galima padaryti išvadą, kad tiek Parkinsono liga sergančių subjektų, tiek kontrolinių subjektų eisenos yra visiškai vienodos dažnių srityje ir vien remiantis šita informacija nėra galima nustatyti ar subjektas serga Parkinsono liga ar ne.

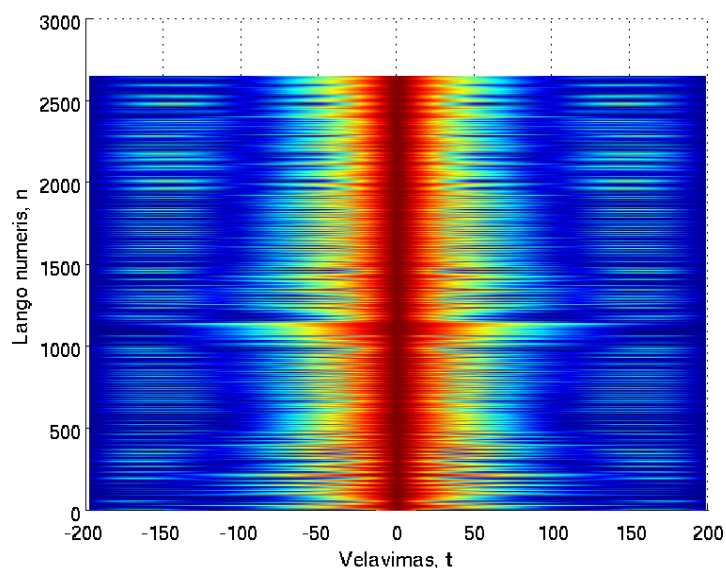
Tolimesnė analizė gali būti atlikta, remiantis koreliacijos koeficientais. Analizuojant šias savybes, galima remtis tokiu pačiu pirminiu signalo apdorojimo bloku, kaip ir analizuojant dažnines komponentes. Dėl to, kad esama tikrais kairės kojos signalais, galima taikyti tik savi-koreliacijos koeficientus. Narinėjama savybė parodė labai gerus rezultatus ankstesniame tyrime, kuriame, remiantis akselerometro ir giroskopo jutiklių parodymais, reikėjo suprojektuoti algoritmą, gebantį atskirti tokias žmogaus veiklas: stovėjimas,ėjimas,ėjimas aukštyn laiptais,ėjimas žemyn laiptais.

Kontrolinių subjektų kairės kojos savi-koreliacijos koeficientai parodyti 11 pav. Parkinsono liga sergančių subjektų kairės kojos savi-koreliacijos koeficientai parodyti 12 pav. Kaip matosi iš korelogramos, tiek sergantys, tiek kontroliniai subjektai turi panašias, o kai kuriais atvejais ir tokias pačias, koreliacijos reikšmes. Remiantis vien tik turima informacija, nustatyti ar subjektas serga Parkinsono liga ar ne, nėra įmanoma.

Sudėtingesnė analizė seka iš dviejų globalių maksimumų ir vieno lokalaus minimumo savybių erdvės. Šios trys savybės sudaro trijų dimensijų plokštumą, kurią galima lengvai pavaizduoti. Pilna savybių erdvė



11 pav.. Kontrolinių subjektų kairės kojos žingsnių signalų savi-koreliacijos koeficientai.

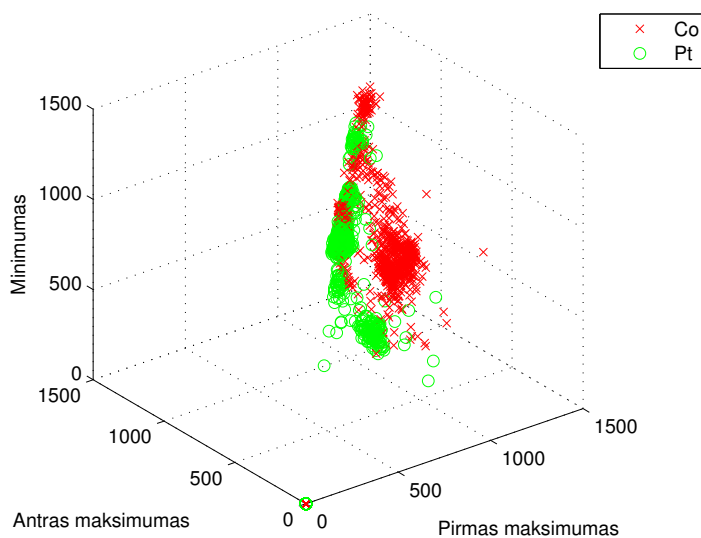


12 pav.. Parkinsono liga sergančių subjektų kairės kojos žingsnių signalų savi-koreliacijos koeficientai.

pavaizduota 13 pav. Kaip matosi iš duotos erdvės, duomenys neturi jokio koncentracijos centro. Erdvėje jie pasiskirstę pagal nežinomą dėsnį.

Tokių duomenų pateikti klasifikavimui nėra galima. Grafike "Co" taškai, pažymėti kryžiumi, parodo kontrolinį subjektą, "Pt" taškai, pažymėti apskritimu, parodo Parkinsono liga sergančius subjektus.

Sekančios savybės nagrinėjimui yra siūlomos atlikto tyrimo [1]. Nagrinėjama savybė yra žingsnio fazės



13 pav.. Dviejų globalių maksimumų ir vieno lokalaus minimumo savybių erdvė.

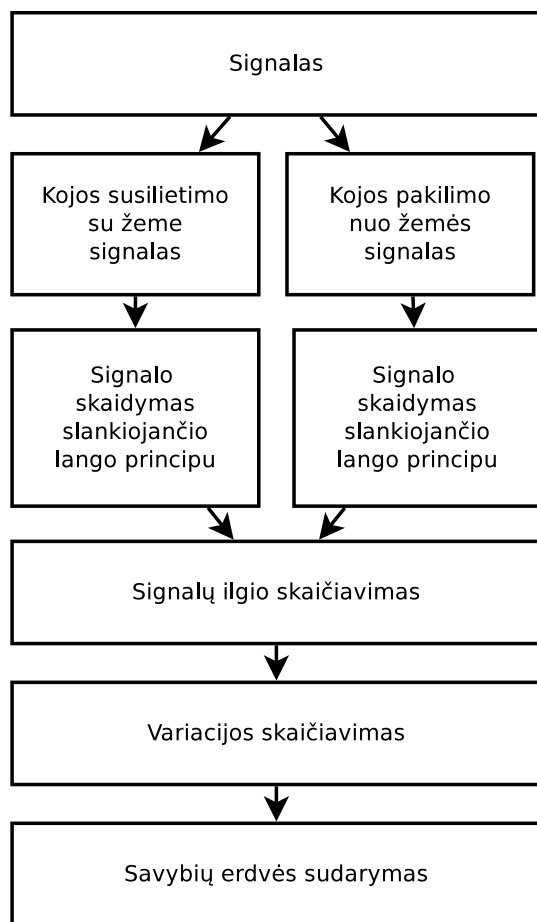
laiko variacija. Kojos susilietimo su žeme laiko variacija ir kojos pakilimo nuo žemės laiko variacija.

Išskiriant variacijos savybes iš signalo, reikalinga nuo pagrindų pakeisti pirminio signalo apdorojimo algoritmą. Kadangi kojos pakilimas nuo žemės ir kojos susilietimas su žeme gali būti vienas nuo kito nepriklausomi (subjektas sustojo ir stovi dviem kojomis ant žemės arba subjektas stovi tik ant kairės kojos), pirminis signalo apdorojimas turi būti atliekamas lygiagrečiai, t.y. tuo pat metu išskiriamas tiek kojos pakilimo nuo žemės signalas, tiek ir kojos susilietimas su žeme signalas. Tokios sistemos struktūrinis grafikas yra parodytas 14 pav. Sekantis žingsnis yra signalo išskaidymas slankiojančio lango principu. Abiejų signalų slankiojančio lango ilgis buvo parinktas 4 verčių ilgio, su 2 verčių perdanga. Kuomet tiek kojos susilietimo su žeme, tiek kojos pakilimo nuo žemės keturi signalai patenkina keliamus reikalavimus - iš jų yra apskaičiuojami signalų ilgiai. Vėliau, abiejų etapų signalams yra paskaičiuojama jų variacija ir taip yra sudaroma savybių erdvė. Toliau bus nagrinėjamos abiejų signalų variacijos vienos dimensijos plokštumoje.

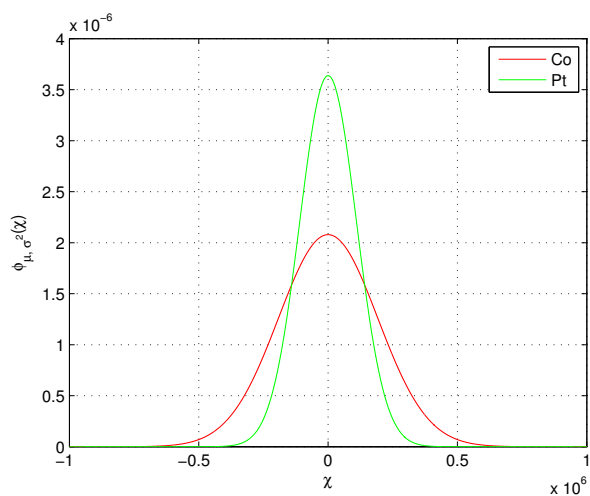
Abiejų signalų variacijų pasiskirstymai yra parodyti 15 pav. ir 16 pav. Abu pasiskirstymai turi vieną trūkumą - jų vidurkiai sutampa, tačiau gera žinia yra tai, kad jų variacijos skirtingos. Norint išspręsti iškilusią problemą, reikia taikyti dimensijų praskyrimo metodus. Darbo metu buvo išbandyti tokie dimensijų praskyrimo metodai:

- PCA;
- LDA;
- Kernel PCA (su "poly", "Gauss" branduoliais);
- Kernel LDA (su "poly", "Gauss" branduoliais);

Vienmatė linijinė PCA transformacija pavaizduota 17 pav. Kaip matosi iš grafiko - vidurkis nepasikeitė,

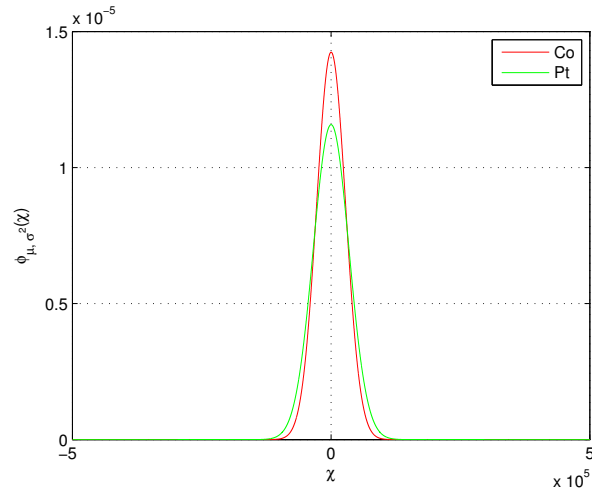


14 pav.. Pirminis signalo apdorojimas, išskiriant savybę kaip kojos susilietimo su žeme laiko variaciją ir kojos pakilimo nuo žemės laiko variaciją.

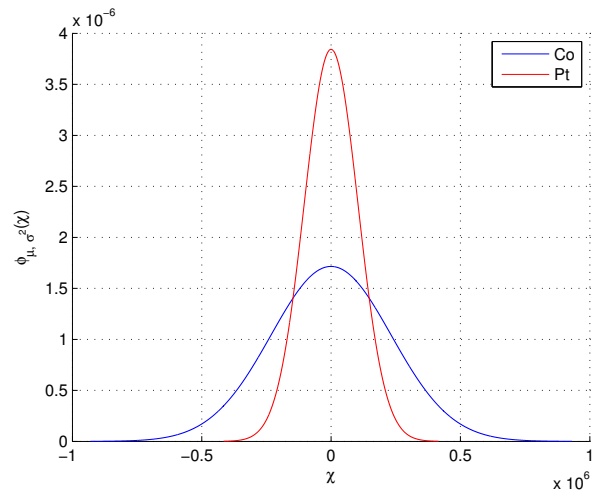


15 pav.. Kojos prisilietimo prie žemės laiko variacija.





16 pav.. Kojos pakilimo nuo žemės laiko variacija.

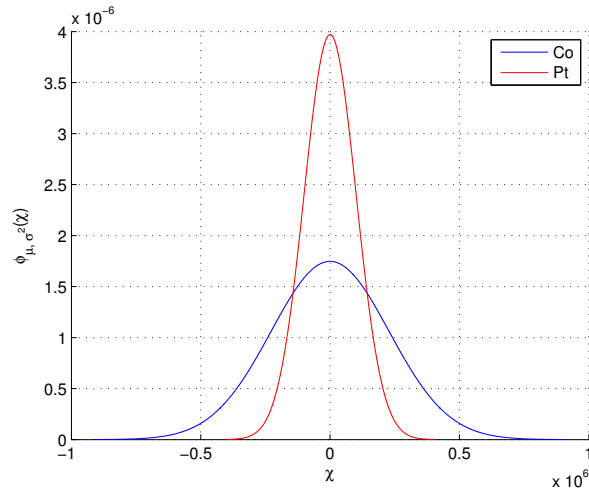


17 pav.. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos.

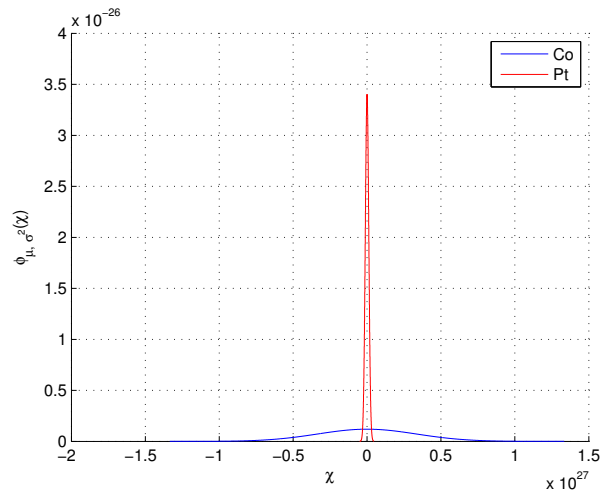
tačiau žymiai pasikeitė variacija. Vidurkio pokyčio nebuvimas pablogina situaciją, kadangi to pačio vidurkio duomenis atskirti nėra įmanoma, todėl ir teikti duomenis klasifikavimui nėra prasmės.

Sekantis metodas yra LDA. Transformacijos rezultatas yra pavaizduotas 18 pav. Kaip matosi - tiek po LDA, tiek po PCA duomenų pasiskirstymas nėra kritiškai skirtingas. Galima daryti hipotezę, kad linijinis duomenų praskirstymas šiuo atveju neveikia. Reikia taikyti branduolio metodą ir papildyti jau panaudotus transformacijos metodus polinominiu arba Gausso branduoliu.

Branduolio metodas pritaikytas PCA transformacijai yra pavaizduotas 19 pav. ir 21 pav. Branduolio metodas pritaikytas LDA transformacijai yra pavaizduotas 20 pav. ir 21 pav. Polinominis branduolys abiem atvejais buvo panaudotas su 4 branduolio argumentu.



18 pav.. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos.



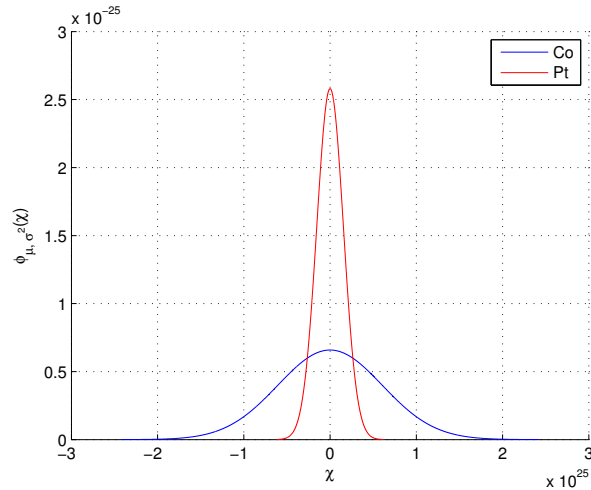
19 pav.. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant polinominį branduolį.

Kaip matosi, panaudojus polinominį branduolį, tiek PCA (19 pav.), tiek LDA (20 pav.) atveju - vidurkis iš savo vietos nepajudėjo. Abidvi transformacijos pakeitė variaciją. Tai nėra norimas tikslas, todėl toks branduolys nėra tinkamas. Panaudojus Gauso branduolį rezultatas labai pagerėja LDA transformacijos atveju (22 pav.), PCA su tokiu branduoliu (21 pav.) duomenis tik dar labiau suvienodina.

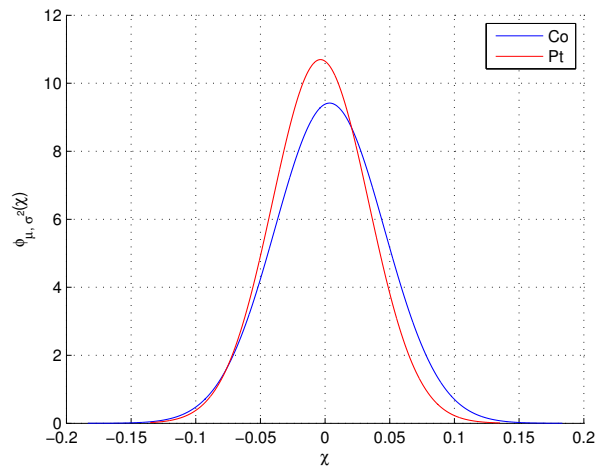
Remiantis pateikta analize galima teigti, kad geriausiai duomenis savybių erdvėje atskiria LDA su Gauso branduoliu. Tolimesniame darbe duomenis į klasifikatorių bus pateikiami būtent po tokios transformacijos.

#### D. Požymių klasifikavimo programos kūrimas

Šiame skyriuje bus išnagrinėti ir pritaikyti populiariausi šiuo metu naudojami klasifikatorių metodai:



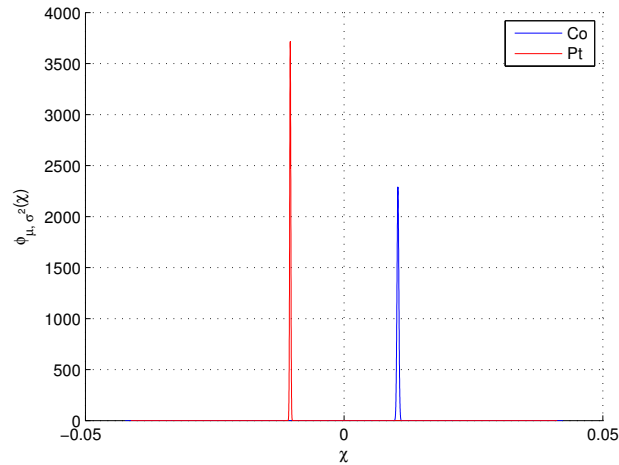
20 pav.. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos, naudojant polinominį branduolį.



21 pav.. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po PCA transformacijos, naudojant Gauso branduolį.

- Vektoriaus palaikymo mašina (angl. Support Vector Machine (SVM));
- Paslėptas Markovo modelis (angl. Hidden Markov Model (HMM));
- Naivusis Bayes klasifikatorius (angl. NaiveBayes);
- Tiesioginio sklaidimo neuronų tinklas (angl. Feed-Forward Neural Network (FFNN));

Labai svarbu yra atskirti duomenis, kuriais klasifikatorius yra apmokamas ir kuriais jis yra testuojamas. Jeigu klasifikatoriaus apmokymo duomenys yra pakankamai apibendrinti, tuomet naujus duomenis klasifikatorius turėtų gerai atpažinti. Pateikiant klasifikatoriaus testavime tokius pačius duomenis, kaip ir apmokyme - atliekamas tikrinimas ar klasifikatorius teisingai “suprato” nagrinėjamus duomenis, tačiau tai neapibrėžia



22 pav.. Kojos prisilietimo prie žemės ir pakilimo nuo žemės laiko variacijos pasiskirstymas po LDA transformacijos, naudojant Gauso branduolį.

I lentelė  
VEKTORIAUS PALAIKYMO MAŠINOS TIKSLUMAS IR TAIKLUMAS.

	C1	C2
Tikslumas	0.489	0.489
Taiklumas	0.234	0.744

kiek gerai jis apdoroja naujus duomenis.

Vektoriaus palaikymo mašina šiuo metu yra populiariausias klasifikatorius nagrinėjant netiesiškai atskiriamus duomenis. Klasifikatoriaus testavimo metu buvo naudojami tokie parametrai:

- SVM tipas - Sequential Minimal Optimization;
- Branduolio tipas - linijinis;
- Sureguliojimo konstanta - 2;
- Branduolio argumentas - 2;

Lentelėje I pateikti SVM tikslumo ir taiklumo duomenis. Kaip matosi, klasifikatorius veikia nepakankamai gerai - tikslumo koeficientas nėra didesnis negu pusę, abiem atvejais tik 0.489. Pirmos klasės atpažinimo taiklumas tėra 0.234, antros klasės atpažinimo taiklumas yra 0.744, tačiau to nepakanka. Iš padarytos klasifikatoriaus veikimo analizės galima teigti, kad klasifikatorius veikia blogai ir jo naudoti sprendime nėra galima.

Paslėptas Markovo modelis yra vienintelis iš šiame darbe nagrinėjamų klasifikatorių, kuris turi laikinę informaciją. Tokia savybė suteikia „inkaro“ galimybę - klasifikatorius gali užsilaikyti prie vienos klasės net ir tuomet, kai pagal savybių erdvę turi būti kitas klasifikavimo rezultatas. Tokia klasifikatoriaus savybė buvo pritaikyta ankstesniame darbe, sudarant žmogaus eisenos atpažinimo sprendimą [3].

II lentelė  
PASLĖPTO MARKOVO MODELIO TIKSLUMAS IR TAIKLUMAS.

	C1	C2
Tikslumas	0.500	0.500
Taiklumas	0.000	1.000

III lentelė  
NAIVAUS BAYES KLASIFIKATORIAUS TIKSLUMAS IR TAIKLUMAS.

	C1	C2
Tikslumas	0.508	0.508
Taiklumas	0.764	0.252

Lentelėje II pateikti tikslumo ir taikumo duomenys. Kaip matome iš tikslumo rezultato - klasifikatorius teisingai priskiria tik pusei reikiamų duomenų. Iš to galime teigti, kad klasifikatorius yra labai blogai apmokytas ir visiškai nesugeba apibendrinti turimų duomenų.

Naivus Bayes yra vienas iš pirmųjų statistinių metodų paremtu klasifikavimo mechanizmas. Jis veikia labai paprastai - ieškoma linijinės funkcijos, kuri geriausiai atskiria nagrinėjamus duomenis ir vieni žymenis priskiriami, jeigu duomenys yra vienoje linijos pusėje, atvirkšti žymenis priskiriami, jeigu duomenys yra kitoje linijos pusėje.

Lentelėje III pateikti tikslumo ir taikumo duomenys.

#### *E. Duomenų analizės programos kūrimas*

### IV. SIGNALŲ ANALIZĖS PROGRAMOS ĮGYVENDINIMAS

Šiame skyriuje apžvelgsime įgyvendinta programinį kodą ir jo veikimo architektūrą. Ankstesniuose poskyriuose buvo argumentuotai pažvelgti galimi analizės metodai, požymiai ir klasifikatoriai. Visi rezultatai bus panaudoti projektuojant galutinį sprendimą.

Poskyryje IV-A bus pateikta bendra programos algoritmo veikimo schema, poskyryje IV-B bus pateikta algoritmo klasifikavimo veikimo schema, poskyryje IV-C bus apžvelgta galutinė programa.

#### *A. Bendro programos algoritmo schemos sudarymas*

Ankstesniame skyriuje buvo apžvelgta bendra programos veikimo schema. Bendros schemos pavyzdys yra pateiktas 1 pav. Šiame poskyryje bus pateikta detali algoritmo schema ir aptarta kiekviena jo bloko paskirtis, bei jame naudojamas metodas.

#### *B. Požymių klasifikavimo programos algoritmo schemos sudarymas*

#### *C. Signalų analizės programos įgyvendinimas*

Šiame poskyryje

## V. SIGNALŲ ANALIZĖS PROGRAMOS PATIKRA

Šiame skyriuje bus parengtas ir įgyvendintas algoritmo patikros planas. Patikra bus atliekama su užduoties analizėje nurodytais duomenimis - ... Parkinsono liga sergančių asmenų ir ... nesergančių asmenų duomenimis.

### A. Eksperimentų plano rengimas

Detaliai ištiriant programos veikimo rezultatus, reikia nurodyti programos įvestyje kiek galima daugiau galimų signalo analizės scenarijų.

Galimi signalų apdorojimo scenarijai:

- Reikia pagalvoti;
- Reikia pagalvoti.

### B. Duomenų eksperimentams rengimas

### C. Programos patikros rezultatai

Algoritmo patikros metu buvo gauti tokie rezultatai...

## LITERATŪRA

- [1] Silvi Frenkel-Toledo, Nir Giladi, Chava Peretz, Talia Herman, Leor Gruendlinger, and Jeffrey Hausdorff. Effect of gait speed on gait rhythmicity in parkinson's disease: variability of stride time and swing time respond differently. *Journal of NeuroEngineering and Rehabilitation*, 2(1):23, 2005.
- [2] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *ArXiv Mathematics e-prints*, January 2007.
- [3] Maksim Norkin. Human activity recognition using Hidden Markov Models, September 2011.