

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tadej Vodopivec

**Identifikacija oseb ob prehodu skozi
vrata iz podatkov senzorjev pospeškov**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Peter Peer

ASISTENT: Rok Piltaver, univ. dipl. inž. rač. in inf.

Ljubljana, 2013

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00494/2013

Datum: 08.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **TADEJ VODOPIVEC**

Naslov: **IDENTIFIKACIJA OSEB OB PREHODU SKOZI VRATA IZ PODATKOV
SENZORJEV POSPEŠKOV**

**IDENTIFICATION OF PERSONS ENTERING THROUGH THE DOOR
FROM THE ACCELEROMETERS DATA**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Zasnujte in opišite postopek, ki na podlagi podatkov zajetih s senzorji pospeškov, ki so vgrajeni na vratno krilo, ob prehodu skozi vrata samodejno identificira osebo. Poskrbite za zajem podatkov, izračun atributov, definicijo faz prehoda ter testirajte različne klasifikatorje.

Mentor:


doc. dr. Peter Peer



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Tadej Vodopivec, z vpisno številko **63100368**, sem avtor diplomskega dela z naslovom:

Identifikacija oseb ob prehodu skozi vrata iz podatkov senzorjev pospeškov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Petra Peera in s pomočjo Roka Piltaverja, univ. dipl. inž. rač. in inf.,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 15. septembra 2013

Podpis avtorja:

Diploma je nastala pod mentorstvom doc. dr. Petra Peera in s pomočjo Roka Piltaverja, univ. dipl. inž. rač. in inf., ki se jima iskreno zahvaljujem za strokovno usmerjanje, pomoč in koristne nasvete pri izdelavi naloge.

Zahvaljujem se tudi Institutu "Jožef Stefan", ki mi je omogočil izvedbo diplomske naloge, staršem, ki so mi omogočili študij, za podporo med letem, Tomažu Kompari za izdelavo strojno-programске opreme in postavitev krmilnika Arduino, Borutu Grošičarju za pregled fizikalnega ozadja in izdelavo vezja za senzorje ADXL327, Jerneju Zupančiču za izračun rotacijske matrike in njeno ortogonalizacijo, Hristijanu Gjoreskemu za izdelavo strojno-programске opreme za napravo Shimmer, Pšenici Kovačič za izdelavo skic, Nataliji Krese za lektoriranje, Božidari Cvetković za pomoč pri oblikovanju besedila, sodelavcem in prijateljem, ki so sodelovali pri zajemu podatkov, ter Yvesu Lestevenu in Lucasu Draiu za pomoč pri označevanju podatkov.

Kazalo

Kazalo slik	i
Kazalo tabel	iii
Seznam uporabljenih kratic in simbolov	v
Povzetek	vii
Abstract	ix
1 Uvod	1
1.1 e-Vratar	1
1.2 Identifikacija oseb	3
1.3 Sile, ki delujejo na vratno krilo	3
2 Senzorji	5
2.1 Senzorji, priklopljeni na Arduino krmilnik	5
2.1.1 Triosni senzor pospeškov Grove	8
2.1.2 Tri-osni senzor pospeškov ADXL327	8
2.2 Senzor v tabličnem računalniku Android	10
2.3 Senzor v napravi Shimmer	11
3 Zajem podatkov	13
3.1 Zajem podatkov preko krmilnika Arduino	13

3.2	Zajem podatkov z uporabo tabličnega računalnika z operacijskim sistemom Android	17
3.3	Zajem podatkov s strojno-programsko opremo na napravi Shimmer	19
4	Rotacija izmerjenih pospeškov	21
4.1	Rotacijska matrika	24
4.2	Ortogonalizacija	25
4.3	Algoritem za rotacijo izmerjenih pospeškov	26
5	Označevanje podatkov	29
5.1	Označevanje podatkov, pridobljenih preko Arduino krmilnika	29
5.2	Označevanje podatkov, pridobljenih s tablice Android	30
6	Opis postopka zajema podatkov	33
6.1	Snemanje prehodov skozi vrata za namen analize prehoda skozi vrata	33
6.2	Snemanje prehodov skozi vrata za namen izgradnje klasifikatorja	34
6.3	Dolgoročno snemanje prehodov skozi vrata	35
7	Vizualizacija in analiza tipičnih vzorcev	37
8	Algoritem za detekcijo zanimivih dogodkov	41
9	Predobdelava signala	45
9.1	Izračun kotne hitrosti in kompenzacija šuma	47
9.2	Izračun kotnega pospeška	47
9.3	Popravek kompenzacije šuma	48
9.4	Izračun kota odprtosti vrat	48
9.5	Izračun standardnega odklona pospeška v smeri gravitacije	48
10	Faze prehoda	49
10.1	Ločevanje začetka in konca prehoda	51
10.2	Ločevanje odpiranja in zapiranja	51

KAZALO

10.3 Ločevanje delov odpiranja in zapiranja	53
10.4 Iskanje lokalnih ekstremov	54
11 Seznam značilnk in njihov pomen	55
12 Rezultati	59
12.1 Izbor značilnk in klasifikatorja	60
12.2 Ugotovitve	62
13 Sklep	69

Slike

1.1	Logotip sistema e-Vratar.	2
2.1	Vrata, na katerih so bile opravljene meritve.	6
2.2	Slika krmilnika Arduino.	7
2.3	Vežje senzorja pospeškov Grove.	8
2.4	Vežje s senzorjem ADXL327.	9
2.5	Shema vezave vezja s senzorjem ADXL327.	9
2.6	Tablični računalnik Android	10
2.7	Naprava Shimmer.	11
3.1	Shema postavitve naprav	14
3.2	GUI na tabličnem računalniku	18
4.1	Senzor pospeškov ni vzporeden z vratnim krilom	22
4.2	Os senzorja pospeškov ni vzporedna s smerjo gravitacije	22
4.3	Os rotacije vratnega krila ni vzporedna s smerjo gravitacije	23
4.4	Graf pospeškov po prvi fazi kalibracije.	27
4.5	Graf pospeškov po drugi fazi kalibracije	27
5.1	Program za telefon Android	31
6.1	Blok papirjev s kodami na mizi	35
7.1	Graf pospeškov, ko oseba z istim gibom ustavi vratno krilo in začne zapirati vrata	38

7.2	Graf pospeškov, ko oseba med prehodom preneha s silo delovati na vratno krilo	39
8.1	Primer rezultata algoritma za detekcijo zanimivih dogodkov. . .	43
9.1	Primerjava izmerjenih in izračunanih podatkov	46
10.1	Faze prehoda skozi vrata	52
11.1	Trije primeri lokalnih ekstremnih vrednosti.	57
11.2	Linearna interpolacija.	57
11.3	Integral.	58
11.4	Trajanje.	58
12.1	Vizualizacija primerov prehodov z metodo RadViz	63
12.2	Vizualizacija zgrajenega klasifikacijskega drevesa	64

Tabele

12.1	Ocene 16 najbolj ocenjenih značilk, razvrščene po meri ReliefF.	61
12.2	Ocene klasifikatorjev	65
12.3	Klasifikacijska točnost pri klasifikaciji en proti vsem	66
12.4	Uspešnost umetne nevronske mreže pri identifikaciji oseb . . .	68

Seznam uporabljenih kratic in simbolov

ARM — Advanced RISC Machine

PC — Personal Computer

USB — Universal Serial Bus

GPS — Global Positioning System

ASCII — American Standard Code for Information Interchange

RFID — Radio-frequency identification

PIN — Personal identification number

NFC — Near field communication

POSIX — Portable Operating System Interface

CSV — Comma-separated values

α — kotni pospešek

ω — kotna hitrost

SVM — Support vector machine

MARS — Multivariate adaptive regression splines

kNN — k-Nearest Neighbours

RadViz — Radial Coordinate Visualization

CA — Classification accuracy

AUC — Area under the ROC curve

IS — Information score

ROC — Receiver operating characteristic

Povzetek

V tem delu je predstavljen sistem, ki na podlagi podatkov, zajetih s senzorji pospeškov, ki so montirani na vratno krilo, ob prehodu skozi vrata avtomatično identificira človeka. Postopek za avtomatično identifikacijo sestoji iz štirih glavnih delov. V prvem koraku se izvaja zajem podatkov in rotacija izmerjenih pospeškov, ki je potrebna zato, da se iz izmerjenih pospeškov izračunajo tangencialni, radialni in gravitacijski pospešek glede na gibanje vratnega krila med odpiranjem. Rotacija koordinatnega sistema je potrebna zato, da se izmerjeni pospeški ujemaajo s komponentami pospeškov, ki delujejo na vratno krilo. Rotacijsko matriko, ki se uporablja v ta namen, je treba na podlagi za to namenjenih meritev izračunati pred izvajanjem identifikacije. V drugem koraku se na podlagi izmerjenih pospeškov izračuna kotni pospešek, kotno hitrost vratnega krila, standardni odklon pospeška v smeri gravitacijskega pospeška ter kot odprtosti vrat. V tretjem koraku se na podlagi naštetih količin izračuna točke, v katerih so meje med različnimi fazami prehoda skozi vrata. Za vsako fazo se izračuna integral, smerni koeficient linearne interpolacije in ekstremno vrednost vsake količine v vsaki fazi ter trajanje vsake faze. V četrtem koraku se vrednosti, izračunane v prejšnjem koraku, uporabijo kot značilke za klasifikacijo.

Sistem se bo uporabil v projektu e-Vratar, ki se izvaja na Institutu »Jožef Stefan«. Cilj projekta je razviti prototip elektronskega vratarja: inteligentnega sistema, ki temelji na vratih z elektromehansko ključavnico, tabličnem računalniku, krmilniku Arduino in paleti senzorjev, ki ponuja podobne storitve kot človeški vratar, izboljša varnost in povečuje udobje uporabnika.

Senzorji zbirajo podatke o dogodkih, povezanih z vrati, ki se uporabljajo za kontekstno odločanje in zavedanje, ki se doseže z metodami umetne inteligence, ki teče na tabličnem računalniku. Sistem e-Vratar je sposoben prepoznati uporabnike, odkriti nenavadne vstope in izstope, vlome in poskuse vlomov, predvidevati prisotnost uporabnikov in ponuja individualizirane storitve (nastavljiva sporočila in alarme, uporabne informacije, glasovna sporočila, pozdrave in namige) in daljinsko upravljanje s pomočjo intuitivnega grafičnega uporabniškega vmesnika ali virtualnega asistenta, ki razume naravni jezik.

Ključne besede:

identifikacija, vrata, vstop, senzorji, strojno učenje, obdelava signalov

Abstract

This thesis presents a system for automatic identification of a person who walks through a door based on the information gathered with acceleration sensors, which are mounted on the door. The procedure for automatic identification consists of four steps. In the first step, data acquisition and coordinate system rotation is performed. Coordinate system rotation is needed in order to match the direction of measured accelerations with the direction of components of the forces acting on the door. The rotational matrix needs to be computed from dedicated measurement prior to the identification procedure. In the second step, angular acceleration, angular speed, standard deviation of the acceleration in the direction of gravity and the opening angle of the door are calculated from the rotated accelerations. In the third step, the borders between various phases of transition through the door are identified. Integral, linear interpolation and extreme values are computed for each component of acceleration and duration is calculated for each phase. In the fourth step, classification is performed using values computed in the previous step.

The system will be used in the project e-Doorman, which is developed at »Jožef Stefan« Institute. The aim of the project is to develop a prototype of electronic doorman: an intelligent system based on a door with electro-mechanic lock, tablet computer, Arduino micro-controller, and an array of sensors that offers services similar to a human doorman, improves security, and increases user comfort. The sensors gather data about events related to the door, which are used for context based reasoning and awareness achieved

by artificial intelligence methods running on the tablet computer. The e-Doorman system is able to recognize the users, detect unusual entries and exits, break-ins and break-in attempts, predict user presence, offers personalized services (customizable notifications and alarms, relevant information, voice messages, greetings and tips), and remote control using intuitive graphical user interface or virtual assistant that understands natural language.

Keywords:

identification, door, entrance, sensors, machine learning, signal processing

Poglavje 1

Uvod

Človek svoja bivališča že od nekdaj opremlja z vrati, ki predstavljajo mejo med zunaj in znotraj. Glavni namen vrat je omogočanje prehoda željenih stvari in oseb in preprečevanje vdora neželenih zunanjih vplivov.

Popolno kontrolo nad prehodi skozi vrata se lahko doseže s tem, da vsak stoji pred svojimi vrati in kontrolira prehode. Tak sistem deluje dobro, a hkrati lastniku onemogoča opravljanje skoraj kakršnegakoli drugega dela. Izboljšava tega sistema je uporaba ključavnice, ki ob odsotnosti lastnika oteži prehod, ko pa je lastnik prisoten, lahko prehod nadzoruje sam. To omogoča, da se človek odmakne od vrat in dela druge stvari, prostor za vrati pa je še vedno varen pred nepooblaščenim vstopom. Po drugi strani prehod brez lastnika sploh ni mogoč tudi v primeru, ko bi lastnik to želel. Kompromis med obema rešitvama je vratar: to je oseba, ki ji lastnik zaupa in ki upravlja z vrati po pooblastilu lastnika, medtem ko lastnik ni prisoten. Ta rešitev združuje prednosti obeh predhodnih rešitev, a še vedno zahteva, da nekdo stoji pred vrati.

1.1 e-Vratar

Tu nastopi e-Vratar (slika 1.1), sistem, ki ga razvija Institut »Jožef Stefan« v sodelovanju z razvojnim centrom Intech-les, Kovinoplastiko Lož in Elgoline.



Slika 1.1: Logotip sistema e-Vratar.

Cilj projekta e-Vratar je izdelati sistem, ki simulira funkcijo vratarja, in ga tako v čim večji meri nadomestiti s sistemom, ki nikoli ne zaspi na delovnem mestu, ni podkupljiv in je vedno predvidljiv. Poglavitne funkcije so krmiljenje ključavnice, video nadzor in detekcija (poskusa) vdora z alarmiranjem, komunikacija v naravnem jeziku in upravljanje na daljavo.

Pomembno vodilo je sistem zasnovati tako, da bo cenovno dostopen manj zahtevnim uporabnikom. Hkrati bo za ugodnejšo ceno ponujal napredne, inteligentne funkcionalnosti, ki so trenutno dostopne le pri dražjih izdelkih ali sploh ne. Cilj projekta je ne samo trženje sistema v Sloveniji, ampak tudi širšemu krogu potencialnih uporabnikov v tujini.

Ena od ključnih novosti v izdelku je vključitev umetne inteligence, ki omogoča avtomatsko prilagajanje uporabniškim navadam in potrebam. To je nadgraditev običajnih sistemov, ki delujejo po načelu vnaprejšnjega nastavljanja oziroma programiranja s strani uporabnika. Druga pomembna novost je inteligentna komunikacija z uporabnikom v naravnem jeziku s pomočjo virtualnega asistenta.

1.2 Identifikacija oseb

Ljudje se med seboj razlikujemo po velikosti, moči, motoričnih sposobnostih, koordinaciji, obnašanju, teži, samozavesti itd. Vse to vpliva na naše premikanje, način hoje in način, kako ravnamo s predmeti v naši okolici.

Cilj diplomske naloge je razviti sistem, s katerim bi bilo na podlagi pospeškov vratnega krila mogoče identificirati osebo, ki stopa skozi vrata. S tem bi lahko preverili, ali je res vstopila tista oseba, za katero se izdaja, oziroma ugotovili identiteto osebe, če je le-ta vstopila ali izstopila brez predhodne identifikacije. Identifikacijo bi bilo mogoče realizirati na več načinov, v tej diplomski nalogi pa je za ta namen uporabljeno strojno učenje.

V okviru diplomske naloge je bil izdelan sistem za zajem, shranjevanje in obdelavo podatkov s treh različnih senzorjev pospeškov, pretvorbo podatkov v značilke in klasifikacijo, ki na podlagi teh značilk ugotovi, katera oseba je stopila skozi vrata. Temo dela je prispeval prof. dr. Matjaž Gams. Iz osnutka patentne prijave [9] sem uporabil idejo o uporabi pospeškomerov, ki so montirani na vratno krilo, izločanju atributov in uporabi strojnega učenja za klasifikacijo vstopajočih.

1.3 Sile, ki delujejo na vratno krilo

Gibanje vratnega krila je definirano s silami, ki delujejo nanj. Vratno krilo se krožno giblje okoli osi, ki gre skozi pante, ki vratno krilo držijo na svojem mestu. V mirovanju deluje na vratno krilo gravitacijska sila in sile v obratni smeri, ki vratno krilo podpirajo. Rezultanta vseh teh sil je enaka nič. V gibanju delujejo na vrata poleg naštetih tudi sile, ki vratno krilo pospešujejo ali zavirajo, in sile, ki se pojavijo zaradi kroženja.

Sile, ki delujejo na vratno krilo, so:

- Centrifugalna sila — sorazmerna s kvadratom kotne hitrosti in linearno odvisna od oddaljenosti od osi vrtenja — povzroča radialni pospešek.
- Sila, s katero pritiskamo na vrata — linearno odvisna od kotnega po-

speška in oddaljenosti od osi vrtenja — povzroča del tangencialnega pospeška.

- Sila gravitacije, s katero Zemlja privlači vratno krilo v smeri svojega središča, je linearno odvisna od mase vratnega krila — povzroča gravitacijski pospešek.
- Sila, s katero tečaji delujejo na vratno krilo — po velikosti enaka, po smeri pa nasprotna sili gravitacije.
- Sila trenja v tečaju vrat — linearno sorazmerna s kotno hitrostjo in linearno odvisna od koeficienta trenja v tečajih — vedno deluje v obratni smeri vektorja kotne hitrosti.
- Sila zračnega upora — vedno deluje v obratni smeri vektorja hitrosti:
 - Ko je kotna hitrost dovolj majhna, da je zračni tok laminaren, je sila zračnega upora linearno odvisna od kotne hitrosti, površine vratnega krila ter koeficienta upora.
 - Ko je kotna hitrost tako velika, da je zračni tok turbulenten, je sila zračnega upora sorazmerna s kvadratom kotne hitrosti in linearno odvisna od površine vratnega krila ter koeficienta upora.
- Sila okvirja vrat ob trku z vratnim krilom — povzroča tangencialni pospešek — v primeru, da je ta sila velika, jo zaznamo kot oscilacijo pospeškov v vseh smereh.

Izmed zgoraj naštetih sil so za potrebe identifikacije oseb ob prehodu skozi vrata zanimive predvsem centrifugalna sila, sila, s katero pritiskamo vrata, in sila okvirja vrat ob trku. Sila vrat ob trku in sila, s katero pritiskamo na vrata, delujeta v isti osi in imata podoben vpliv na vrata, zato lahko o njiju razmišljamo kot o eni sili, ki je pravokotna na ostali dve, ki sta si tudi med sabo pravokotni.

V vertikalni osi deluje gravitacijska sila ter v primeru trka z okvirjem tudi sila okvirja vrat. Ker je sila gravitacije konstantna, lahko sklepamo, da je za morebiten pojav oscilacij pospeška v vertikalni smeri odgovorna sila okvirja vrat.

Poglavje 2

Senzorji

V tem poglavju so opisani senzorji, ki so bili uporabljeni pri izdelavi sistema za identifikacijo oseb ob prehodu skozi vrata. Postavitev senzorjev na vratnem krilu je razvidna na sliki 2.1.

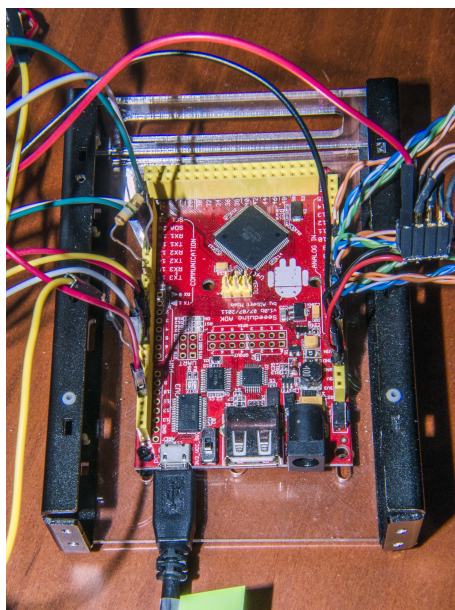
Podatki se zajemajo s triosnim senzorjem pospeškov — to je senzorjem za zajem pospeška v treh med seboj pravokotnih smereh. Prvi preizkušen senzor pospeškov, priklopljen na analogni vhod krmilnika Arduino, se je po testiranju izkazal kot premalo natančen, zato smo ga nadomestili z ustrežnejšim. Drugi preizkušeni senzor pospeškov je tudi najbolj splošno dostopen, saj nosimo podobnega že skoraj vsi s sabo v žepu. To je senzor pospeškov v telefonu ali tabličnem računalniku z operacijskim sistemom Android. Poleg tega smo preizkusili napravo Shimmer, ki vključuje paleto senzorjev, procesorsko enoto, modul Bluetooth in baterijo.

2.1 Senzorji, priklopljeni na Arduino krmilnik

Arduino (slika 2.2) je krmilnik, ki temelji na odprtokodni strojni opremi z procesorsko enoto ARM. Uporabljen krmilnik Arduino ponuja več analognih in digitalnih vhodov in izhodov ter komunikacijo preko serijskega ali USB-priključka. Deluje lahko kot samostojna naprava ali v povezavi z osebnim



Slika 2.1: Vrata, na katerih so bile opravljene meritve.



Slika 2.2: Slika krmilnika Arduino.

računalnikom ali drugo napravo Android oziroma Arduino.

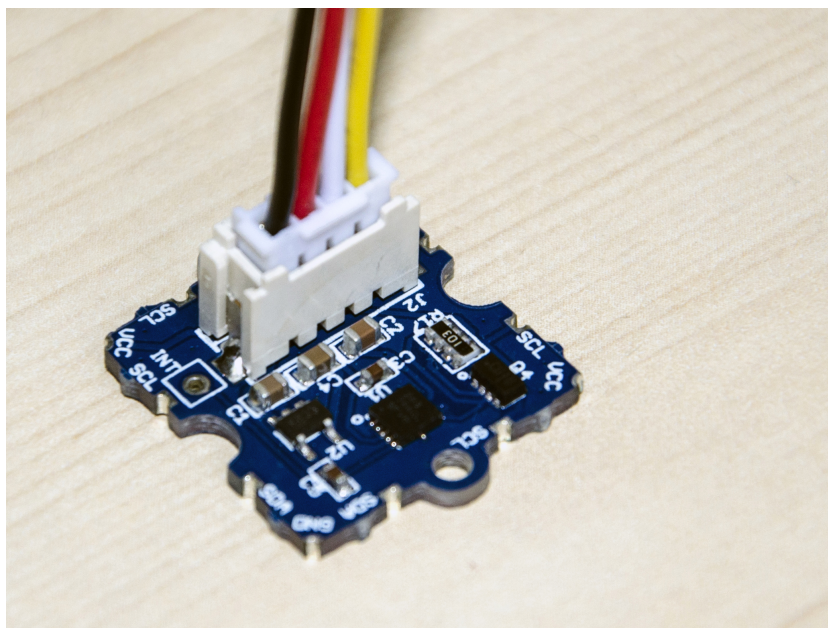
Senzorji se priklopijo na analogni vhod krmilnika Arduino, ki vhodno napetost v željenih časovnih korakih pretvori v digitalno vrednost.

Glede na frekvenco vzorčenja napetosti na analognih vhodih je treba ob senzorjih pospeška uporabiti primerne kondenzatorje. Funkcija kondenzatorjev je integracija napetosti, tako da je vrednost ob vzorčenju enaka integralu vrednosti od zadnjega vzorčenja. Pri tem je treba upoštevati, da frekvenca vzorčenja ne sme biti višja od specifikacij senzorja, saj v tem primeru podatki niso veljavni.

Formula za izračun kapacitete kondenzatorja C na podlagi frekvence zajema podatkov f :

$$C = \frac{5\mu F}{f} \quad (2.1)$$

Za frekvenco vzorčenja 200 Hz potrebujemo kondenzatorje s kapacitivnostjo $0,025 \mu F$.



Slika 2.3: Vezje senzorja pospeškov Grove.

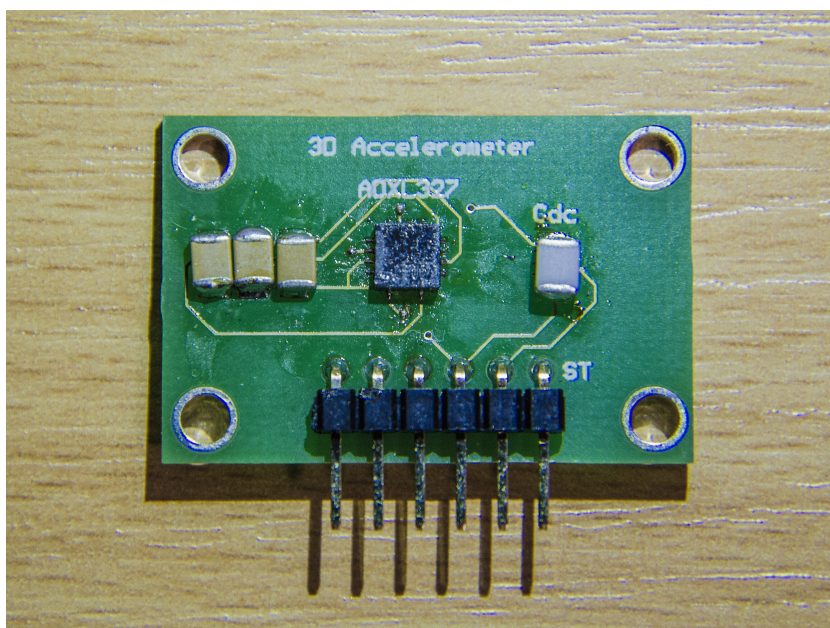
2.1.1 Triosni senzor pospeškov Grove

Prvi uporabljen senzor pospeškov je triosni senzor pospeškov Grove (slika 2.3), ki uporablja senzor pospeškov MMA7660FC. Preizkušen je bil, ker je bil na zalogi, a se je kasneje izkazalo, da ima premajhno natančnost. Rezolucija tega senzorja je $0,047\text{ g}$, šum pa 14% vrednosti, kar je za namen identifikacije oseb, ki prehajajo skozi vrata, praktično neuporabno. Zato je bil nadomeščen z natančnejšim senzorjem pospeškov.

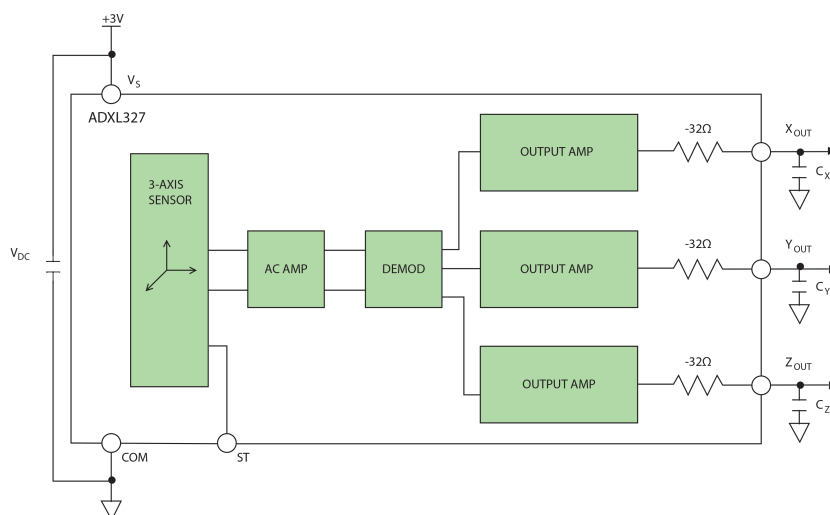
2.1.2 Tri-osni senzor pospeškov ADXL327

Senzor pospeškov ADXL327 (slika 2.4) ima natančnost $0,004\text{ g}$ in šum do 2% vrednosti pri konstantnem pospešku [5]. Senzor pospeškov je v obliki mikročipa, zato je bilo treba izdelati tudi vezje, na katerega se čip namesti (slika 2.5) in skrbi za natančne podatke pri željeni frekvenci vzorčenja (enačba (2.1)).

Na krmilnik Arduino smo povezali tri take senzorje pospeškov, ki lahko



Slika 2.4: Vezje s senzorjem ADXL327.



Slika 2.5: Shema vezave vezja s senzorjem ADXL327.



Slika 2.6: Tablični računalnik Android, nameščen v stojalo, pritrjeno na vratno krilo.

beležijo pospeške v treh oseh s frekvenco do 500 Hz v obeh horizontalnih oseh in do 400 Hz v vertikalni osi. Ker so se pojavile težave pri prenosu podatkov s krmilnika Arduino na osebni računalnik po serijski povezavi pri tako visoki frekvenci vzorčenja, smo frekvenco omejili na 200 Hz v vsaki smeri.

2.2 Senzor v tabličnem računalniku Android

Android je operacijski sistem, ki v večini primerov teče na pametnih telefonih ali tabličnih računalnikih (slika 2.6). Prilagojen je za uporabo preko zaslona na dotik. Operacijski sistem sicer podpira veliko število različnih senzorjev. Mobilni telefoni in tablični računalniki pa so v večini primerov opremljeni s senzorjem pospeškov, giroskopom, magnetometrom, GPS-sprejemnikom in senzorjem svetlosti.



Slika 2.7: Naprava Shimmer.

Operacijski sistem Android lahko iz teh podatkov izračuna še nekatere podatke, kot na primer smer gravitacijskega pospeška, velikost in smer linearnega pospeška, pozicijo in orientacijo relativno na planet Zemljo itd. Tablični računalnik, ki sem ga uporabil, ima vgrajen mikročip STMicroelectronics LIS3DH za detekcijo pospeškov z resolucijo $0,018\text{ g}$ in šumom do 2% vrednosti pri konstantnem pospešku [6].

2.3 Senzor v napravi Shimmer

Shimmer (slika 2.7) je majhna brezžična platforma s senzorji, ki je namenjena snemanju in pošiljanju fizioloških in kinematičnih podatkov v realnem času. Podpira senzorje za elektrokardiogram, elektromiogram, galvanski kožni odziv (vlažnost kože), pospeške, kotno hitrost, magnetno polje, GPS, naklon in vibracijo.

Za uporabo je treba napisati strojno-programsko opremo (angl. firm-

ware), ki se izvaja na sami napravi, in program, ki se izvaja na osebнем računalniku. Strojno-programaska oprema zajema podatke s senzorjev, jih shranjuje na pomnilniško kartico in preko povezave Bluetooth pošilja programu na računalniku. Program na računalniku sprejema podatke ter jih shranjuje na disk. Senzor pospeškov na napravi Shimmer ima resolucijo $0,001\text{ g}$ pri razponu $1,5\text{ g}$, oziroma $0,0027\text{ g}$ pri razponu 4 g in v vsakem primeru šum do 2% vrednosti [7].

Poglavje 3

Zajem podatkov

V tem poglavju so opisani postopki za prenos podatkov s senzorjev pospeškov v obliko, ki je primerna za dostop programom na osebem računalniku.

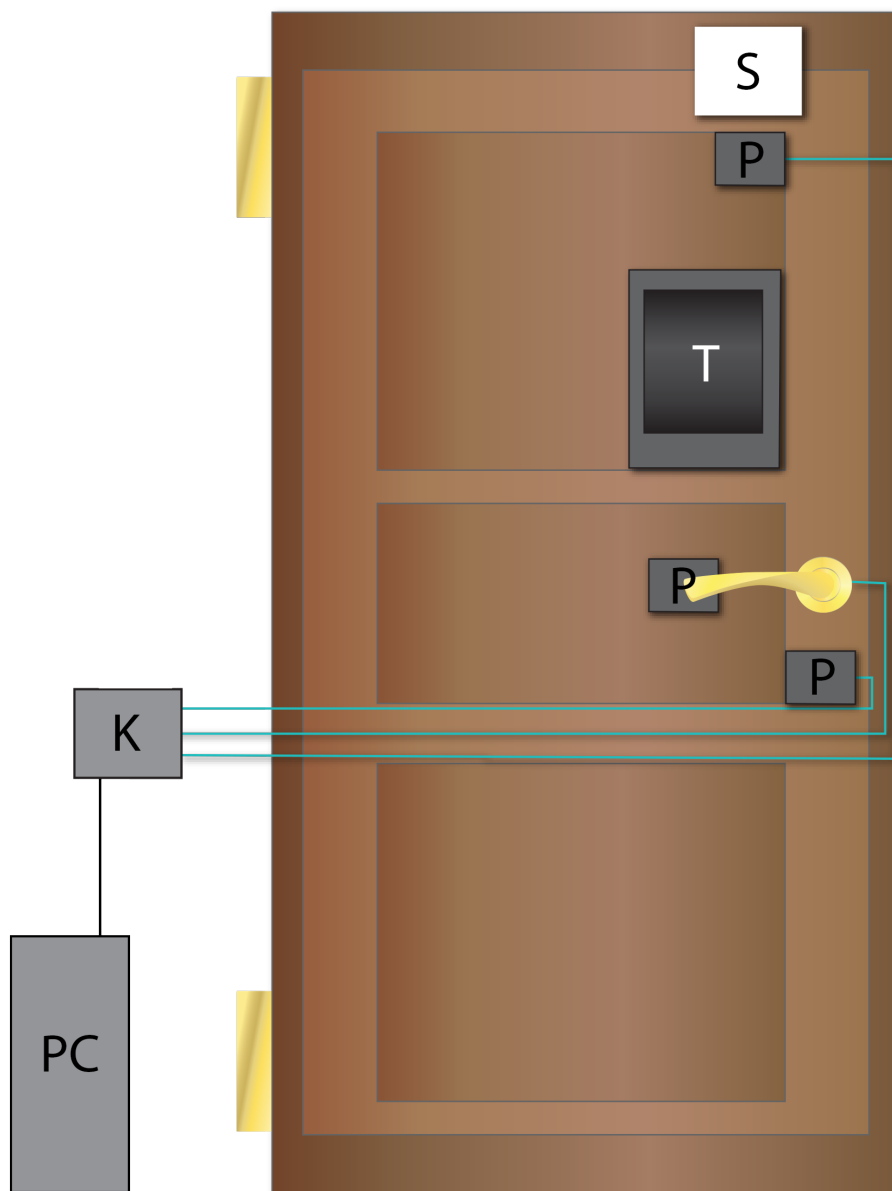
Ko so podatki s senzorjev na voljo, jih je treba zajeti in shraniti na mesto, ki omogoča trajno shranjevanje in od koder so podatki na voljo tudi drugim algoritmom. Zaradi različnih tipov uporabljenih pospeškomerov je za vsak tip potreben poseben postopek.

3.1 Zajem podatkov preko krmilnika Arduino

Krmilnik Arduino je primerna platforma za zajem podatkov, saj omogoča veliko mero svobode pri uporabi, ima pa omejitve glede procesorske moči in možnosti shranjevanja podatkov. Zato sem za sprejemanje podatkov s krmilnika Arduino napisal program za osebni računalnik, ki s krmilnikom Arduino komunicira preko serijske povezave in protokola RS232 ter s krmilnika Arduino tako pridobiva podatke o izmerjenih pospeških.

Zaradi visoke frekvence zajema podatkov vseh podatkov ni mogoče v realnem času pošiljati na osebni računalnik v surovi tekstovni obliki. V ta namen je bilo treba definirati nov protokol za prenos podatkov v paketih.

Postavitev pospeškomerov na vratnem krilu je razvidna na sliki 3.1, sen-



Slika 3.1: Shema postavitve osebnega računalnika, krmilnika Arduino, tablice, naprave Shimmer in pospeškomerov.

zorji so označeni s P, krmilnik Arduino je označen s K in osebni računalnik s PC. Večino podatkov predstavljajo vrednosti pospeškov, ki se normalno pošiljajo kot desetiška števila zakodirana z znaki ASCII. Velik preskok v prepustnosti je bil dosežen s tem, da se vrednosti namesto v tekstovni obliki pošiljajo v binarni obliki. Vrednost pospeškometera v vsaki od treh osi je zakodirana v desetbitnem številu, tako desetiško število do vrednosti 1024 namesto štirih bajtov zasede le dva bajta. Še večja prepustnost je dosežena tako, da se podatki pošiljajo v seriji 32 bitov, kjer prvih deset bitov predstavlja vrednost prve osi, med dvanajstim in enaindvajsetim bitom predstavlja vrednost druge osi, biti od triindvajsetega do dvaintridesetega vrednost tretje osi, enajsti in dvaindvajseti bit sta kontrolna bita in predstavljata tip podatka.

Podatki se s krmilnika Arduino prenašajo na osebni računalnik v paketih, ki so sestavljeni iz serij štirih tipov in se razlikujejo po dveh kontrolnih bitih:

1. Če sta oba kontrolna bita enaka 0, vsebuje serija podatke o pospeških. V paketu so zaporedno zapisane vse zajete vrednosti pospeškov od zadnjega poslanega paketa naprej.

Primer serije: `XXXXXXXX XX0YYYYY YYYYY0ZZ ZZZZZZZZ`
(X, Y in Z predstavljajo vrednosti pospeškometerov v vsaki osi).

2. Če ima prvi kontrolni bit vrednost 0, drugi pa 1, pomeni, da so bili preneseni vsi podatki s senzorjev pospeškov do tistega trenutka. Hkrati je to signal, da lahko osebni računalnik zajete podatke obdela in za tem pošlje zahtevo za prenos novega paketa podatkov.

Primer serije: `***** **0***** *****1** *****` (* predstavlja katerokoli vrednost bita).

3. Če ima prvi kontrolni bit vrednost 1, drugi pa 0, vsebuje serija informacijo o času zajema prvega podatka o pospeških v paketu. Preostalih 30 bitov je uporabljenih za zapis časa v mikrosekundah od zadnjega zagona krmilnika Arduino. Taki seriji 32 bitov sledi vsaj ena serija z vrednostmi pospeškov. Ker je frekvenca zajema podatkov znana, se lahko čas zajema vsakega naslednjega podatka izračuna iz časa zajema

prvega podatka in zaporednega števila poljubnega podatka v paketu.

Primer serije: TTTTTTTT TT1TTTTT TTTTT0TT TTTTTTTT
(T predstavlja podatek o času zajema prvega podatka o pospeških v paketu).

4. Če imata oba kontrolna bita vrednost 1, vsebuje serija podatke z drugih senzorjev. Ker imajo podatki z drugih senzorjev manj različnih vrednosti, jih je mogoče zakodirati z osmimi biti. V tem primeru predstavlja prvih osem bitov tip senzorja (senzor odprtosti vrat, senzor odklenjenosti vrat, senzor gibanja pred vrati, registracija na zunanjem čitalcu RFID, registracija na notranjem čitalcu RFID), za tem sta dva nedefinirana bita, en kontrolni bit, deset nedefiniranih bitov, drugi kontrolni bit, dva nedefinirana bita in osem bitov, ki predstavljajo vrednost, prebrano s senzorja. Pri digitalnih senzorjih je to le vrednost 0 ali 1, v primeru čitalca RFID in čitalca prstnih odtisov pa teh osem bitov predstavlja identifikacijsko številko osebe, ki se je registrirala na čitalcu.

Primer serije: IIIIIIII **1***** *****1** VVVVVVVV (I predstavlja tip senzorja, V predstavlja vrednost senzorja).

Vrednosti s senzorjev pospeškov se pošiljajo samo ob zahtevi, vrednosti z vseh ostalih senzorjev pa takoj, ko se odčitana vrednost senzorja spremeni. Zaradi tega mora imeti krmilnik Arduino predpomnilnik za shranjevanje izmerjenih vrednosti.

Vsi podatki, prebrani iz senzorjev, so shranjeni v podatkovno bazo, kjer so trajno dostopni drugim programom. V ta namen sem vzpostavil podatkovno bazo PostgreSQL, v katero se zapisujejo podatki z Arduina. Podatkovna baza je sestavljena iz štirih tabel. Tri tabele so namenjene pospeškom iz vsakega od treh pospeškomerov in so sestavljene iz petih stolpcev: čas zajema (tipa dvojna beseda (angl. long word)), komponente pospeška x, y in z (tipa število z dvojno natančnostjo (angl. double precision number)) ter oznaka besedilnega tipa (angl. text). Četrta tabela je namenjena podatkom iz ostalih senzorjev in vsebuje en stolpec za čas odčitka (tipa dvojna beseda (angl. long word)), po en stolpec za zadnjo znano vrednost vsakega senzorja

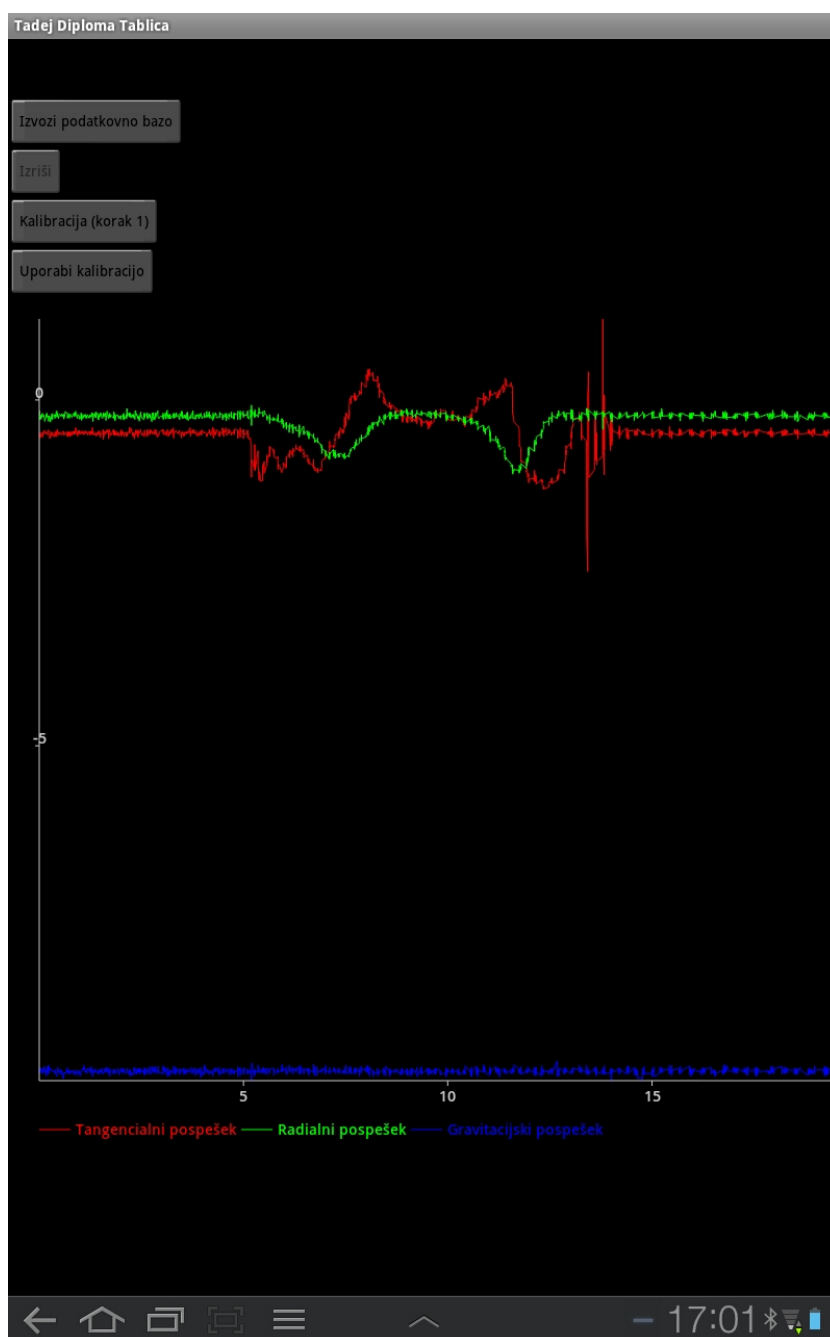
(tipa celo število (angl. integer)), po en stolpec za vsakega uporabnika, v katerem je vpisano, ali je ta uporabnik prisoten ali ne (tipa logična vrednost (angl. Boolean value)), en stolpec, v katerem je vpisano število obiskovalcev v prostoru (tipa celo število (angl. integer)), in zadnji stolpec, v katerem je vpisano, na katerem senzorju je bila odčitana spremenjena vrednost (tipa celo število (angl. integer)).

3.2 Zajem podatkov z uporabo tabličnega računalnika z operacijskim sistemom Android

Za zajemanje podatkov sem napisal program za operacijski sistem Android. Slika grafičnega uporabniškega vmesnika tega programa je na sliki 3.2. Postavitev je razvidna na sliki 3.1, tablični računalnik je lociran na desni strani vratnega krila nad kljuko in označen s T. Operacijski sistem Android omogoča zajemanje vrednosti s senzorja pospeškov v nekaj prednastavljenih frekvencah. Uporabljena je bila najvišja frekvenca, ki sicer ni konstantna, a se giblje okrog 100 Hz. Napisal sem program, ki te podatke shrani v podatkovno bazo SQLite in jih ob pritisku gumba na grafičnem uporabniškem vmesniku izvozi v datoteko .db, ki jo lahko preko USB-kabla prenesemo s tabličnega računalnika na osebni računalnik.

Program zajema podatke avtomatično, označevanje podatkov pa se izvaja ročno. Med snemanjem se lahko vklopi izris grafa, ki prikazuje graf zajetih pospeškov od trenutka pritiska gumba na uporabniškem grafičnem vmesniku do 60 sekund po tem.

Podatkovna baza SQLite je sestavljena iz ene same tabele, ki vsebuje pet stolpcev. Prvi stolpec predstavlja čas dogodka (tipa dvojna beseda (angl. long word)), sledi po en stolpec za vsako os senzorja pospeškov (tipa število z dvojno natančnostjo (angl. double precision number)), zadnji stolpec vsebuje oznako besedilnega tipa (angl. text).



Slika 3.2: Grafični uporabniški vmesnik programa za tablični računalnik Android.

V podatkovno bazo na osebnem računalniku se iz datoteke .db uvozijo vrednosti v tabelo z naslednjimi stolpci:

1. Čas zajema podatkov: v formatu POSIX; to je število pretečenih milisekund od prvega januarja 1970.
2. Velikost izmerjenega pospeška v osi x .
3. Velikost izmerjenega pospeška v osi y .
4. Velikost izmerjenega pospeška v osi z .
5. Besedilna oznaka, ki se uporablja za generiranje učne množice za strojno učenje ter za vizualizacijo podatkov. Postopek označevanja in primeri oznak so opisani v poglavju 5.

3.3 Zajem podatkov s strojno-programsko opremo na napravi Shimmer

Naprava Shimmer (postavitev je razvidna na sliki 3.1, naprava je označena s S) potrebuje namensko strojno-programsko opremo, ki upravlja s strojno opremo na najnižjem nivoju. Uporabljena je bila strojno-programsko opremo, ki zajema podatke pospeškov, jih začasno shrani in ob zahtevi prenese na osebni računalnik preko povezave Bluetooth. Frekvenca zajema pospeškov je 250 Hz v vsaki od treh osi.

Zajemanje podatkov se sproži, ko se senzor aktivira preko osebnega računalnika. Snemanje traja neprekinjeno, dokler se baterije ne iztrošijo ali se snemanje ne prekine z ukazom z osebnega računalnika. Naprava Shimmer shrani podatke na vgrajeno SD-kartico. Z nje se enkrat tedensko preko povezave Bluetooth podatke s pomočjo programa za osebni računalnik prenese na računalnik in pretvori v datoteko CSV. Le-to se nato uvozi v podatkovno bazo PostgreSQL.

Poglavje 4

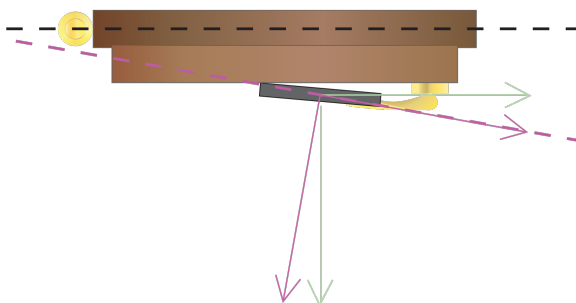
Rotacija izmerjenih pospeškov

Pri namestitvi senzorja pospeškov na vratno krilo je praktično nemogoče doseči popolno poravnanoost osi senzorja pospeškov s komponentami pospeška, ki deluje na vratno krilo. Zamik osi se pojavi, če os vrtenja vratnega krila ni poravnana s smerjo gravitacijskega pospeška, če je senzor pospeškov zaradi debeline vrat ali podstavka zamaknjen napram ravnini vratnega krila, gre skozi os vrtenja ali če senzor pospeškov ni v vseh oseh poravnan z ravnino vratnega krila.

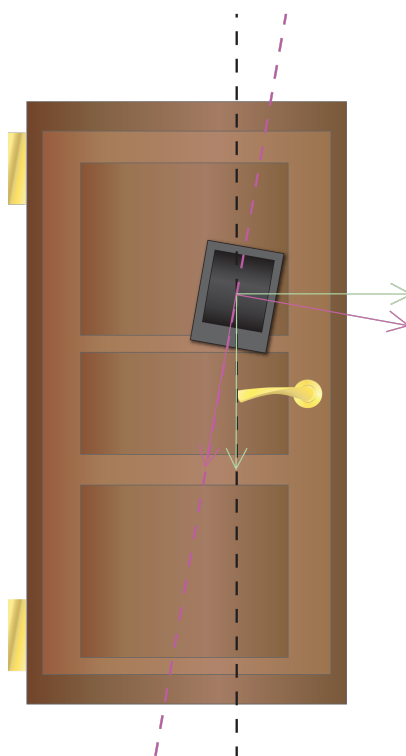
Ker se zaradi zamaknjenosti v smeri, kjer je pričakovan radialni ali tangencialni pospešek, pojavijo projekcije gravitacijskega pospeška, ki je že ob razmeroma majhnih zamikih nekajkrat večji od najvišjih pričakovanih vrednosti pospeškov v ostalih smereh, lahko izmerjene vrednosti odstopajo tudi za do 50 odstotkov glede na dejansko vrednost radialnega in tangencialnega pospeška. To napako je mogoče odpraviti s postopkom, ki je opisan v tem poglavju.

Možne so tri vrste napake pri namestitvi:

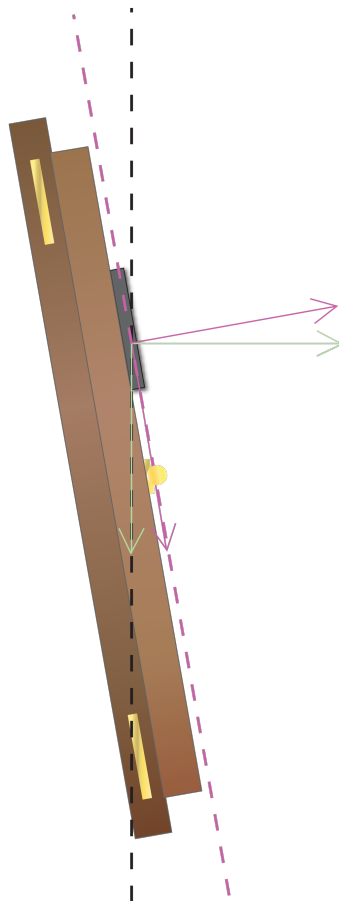
1. Senzor pospeškov ni vzporeden z vratnim krilom (slika 4.1).
2. Os senzorja pospeškov ni vzporedna s smerjo gravitacijskega pospeška (slika 4.2).
3. Os rotacije ni vzporedna s smerjo gravitacijskega pospeška (slika 4.3).



Slika 4.1: Pogled od zgoraj. Senzor pospeškov ni vzporeden z vratnim kri-lom. Komponente pospeškov delujejo v smeri zelenih puščic, izmerjeni pa so pospeški v smeri viola puščic.



Slika 4.2: Pogled od spredaj. Os sensorja pospeškov ni vzporedna s smerjo gravitacijskega pospeška. Komponente pospeškov delujejo v smeri zelenih puščic, izmerjeni pa so pospeški v smeri viola puščic.



Slika 4.3: Pogled s strani. Os rotacije vratnega krila ni vzporedna s smerjo gravitacijskega pospeška. Komponente pospeškov delujejo v smeri zelenih puščic, izmerjeni pa so pospeški v smeri viola puščic.

Vse tri vrste napake je mogoče odpraviti z rotacijo izmerjenega vektorja pospeškov v 3D-prostoru. Za to je treba določiti, za kakšen kot je potrebna rotacija v vsaki smeri.

4.1 Rotacijska matrika

Rotacijska matrika je ortogonalna matrika velikosti 3×3 , ki vektor, s katerim jo množimo, rotira okoli katerekoli osi ali več osi hkrati.

Kote, za katere je treba izmerjene pospeške rotirati v vsaki osi, se določi tako, da se v nadzorovanih okoliščinah izvedeta meritvi (slika 4.4 in 4.5), katerih vrednosti se uporabi za izračun rotacijske matrike, s katero se zmnoži izmerjeni vektor pospeškov, da se izračuna vrednosti pospeškov v želenih smereh.

Prva meritev mora biti izvedena v mirovanju vratnega krila, ko so vrata zaprta in ni zunanjih tresljajev. Pri tem izmerimo le gravitacijski pospešek. Ker vrata mirujejo in ni zunanjih tresljajev, sta radialni in tangencialni pospešek enaka nič. Izkaže se, da je v večini primerov zaradi zamaknjenosti pospeškometra izmerjen pospešek v vseh treh oseh. S to meritvijo je mogoče izračunati, kolikšen del gravitacijskega pospeška se projicira na oseh, na katerih pričakujemo le radialni in tangencialni pospešek, torej vrednost nič. To uporabimo za odpravljanje napake, ko senzor pospeškov ni vzporeden z ravnino vratnega krila in v primeru, ko os senzorja pospeškov ni vzporedna z osjo gravitacijskega pospeška.

Druga meritev je namenjena odpravi zamika, ko os senzorja pospeškov ni vzporedna s smerjo gravitacijskega pospeška. Za to je meritev treba izvesti, ko se vratno krilo giblje z enakomerno kotno hitrostjo, večjo od nič, in je zato izmerjen le radialni in gravitacijski pospešek, tangencialni pospešek pa je enak. S pomočjo te meritve se izmeri, kolikšen del radialnega pospeška se projicira v osi, na kateri je pričakovan samo tangencialni pospešek.

Rotacijska matrika je razvidna iz enačbe (4.4), ki prikazuje rotacijo izmerjenih pospeškov, velikost gravitacijskega pospeška g izračuna po enačbi (4.2)

in velikost centrifugalnega pospeška k po enačbi (4.3). Zaradi preglednosti je formula za izračun člena A zapisana ločeno, v enačbi (4.1).

$$A = \sqrt{\frac{(x_2 z_1 - x_1 z_2)^2}{(y_2^2 + z_2^2)x_1^2 - 2x_2(y_1 y_2 + z_1 z_2)x_1 + (y_2 z_1 + y_1 z_2)^2 + x_2^2(y_1^2 + z_1^2)}} \quad (4.1)$$

$$g = \sqrt{x_2^2 + y_2^2 + z_2^2} \quad (4.2)$$

$$k = \sqrt{(x_1^2 - x_2^2) + (y_1^2 - y_2^2) + (z_1^2 - z_2^2)} \quad (4.3)$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \frac{(y_1 z_2 - y_2 z_1)A}{x_2 z_1 - x_1 z_2} & \frac{x_1 - x_2}{k} & \frac{x_2}{g} \\ A & \frac{y_1 - y_2}{k} & \frac{y_2}{g} \\ \frac{(x_1 y_2 - x_2 y_1)A}{x_2 z_1 - x_1 z_2} & \frac{z_1 - z_2}{k} & \frac{z_2}{g} \end{bmatrix} \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} \quad (4.4)$$

4.2 Ortogonalizacija

Matrika, pridobljena na zgoraj opisan način, zaradi napak v meritvah in šuma pospeškomera ni ortogonalna. Ortogonalna matrika je kvadratna matrika z realnimi elementi, katere vrstice in stolpci so medsebojno pravokotni enotski vektorji (ortonormalni vektorji). Rotacijska matrika, izračunana po zgornjem postopku, je kvadratna matrika z realnimi elementi, a njene vrstice in stolpci niso enotski vektorji. Ob množenju s tako matriko se dolžine vektorja ne ohranjajo.

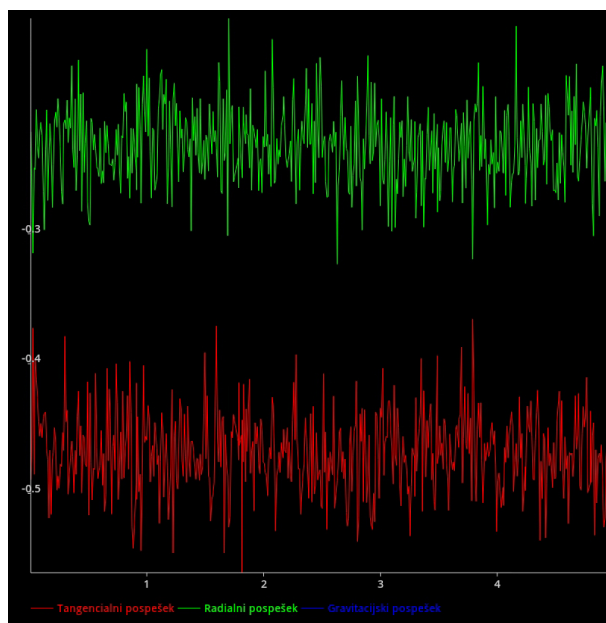
Za ortogonalizacijo se uporablja več metod, med katerimi se najpogosteje uporablja Gram-Schmidtov postopek [10], ki deluje na principu projekcije, za računalniško obdelavo pa so zaradi manjšega vpliva napake pri zaokroževanju bolj primerna Householderjeva zrcaljenja [11].

4.3 Algoritem za rotacijo izmerjenih pospeškov

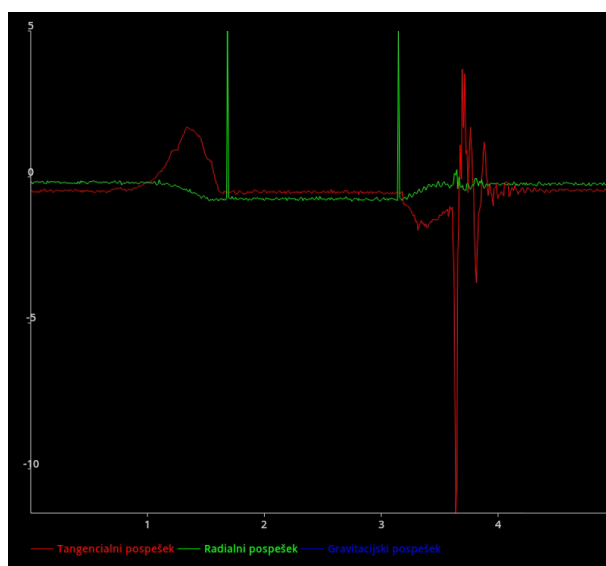
Za namen rotacije izmerjenih pospeškov sem program za zajem podatkov na tabličnem računalniku dopolnil tako, da omogoča avtomatično rotacijo izmerjenih pospeškov. Ob zagonu programa se shranjujejo vrednosti, kot jih tablični računalnik odčita z vgrajenega pospeškmera, ob pritisku na gumb za kalibracijo pa se začne postopek rotacije.

Kot je opisano zgoraj, je treba za kalibracijo izvesti dve meritvi vrat. Ob začetku postopka morajo biti vrata zaprta in brez prisotnosti tresljajev vratnega krila. Ob pritisku na gumb za kalibracijo algoritem počaka dve sekundi in s tem uporabniku omogoči, da se odmakne od vrat in tako ne vpliva na meritev. Za tem začne algoritem za pet sekund zbirati izmerjene pospeške. Ob koncu meritve prikaže zajete podatke na grafu (slika 4.4), da jih uporabnik oceni in ugotovi, ali je prišlo med meritvijo do napake ali motnje meritve. Če je meritev uspešna, se kalibracija nadaljuje tako, da uporabnik odpre vrata in s pritiskom na gumb sproži drugo meritev, ki traja pet sekund. Meritev se začne takoj po pritisku na gumb, med njo pa mora uporabnik zaloputniti vrata tako, da na začetku na vrata deluje s silo, potem pa čim prej prepusti, da se vrata z enakomerno kotno hitrostjo zaprejo, brez da bi uporabnik vplival na njih.

Če je meritev uspešna, poišče algoritem v zajetih podatkih obdobje, v katerem so se vrata premikala z enakomerno kotno hitrostjo. To je obdobje, v katerem so pospeški v vseh treh oseh konstantni. Podatke spet prikaže na grafu in označi območje, v katerem je zaznano enakomerno gibanje (slika 4.5). Iz izbranih zaporedij podatkov v mirovanju in v enakomernem gibanju se s pomočjo mediane izračuna vektorja, ki se uporabita pri izračunu rotacijske matrike.



Slika 4.4: Graf pospeškov po prvi fazi kalibracije.



Slika 4.5: Graf pospeškov po drugi fazi kalibracije. S špicama na grafu radialnega pospeška je označeno zaznano območje enakomernega gibanja.

Poglavje 5

Označevanje podatkov

Označevanje podatkov je potrebno za nadzorovano učenje (angl. supervised learning), kjer učnemu algoritmu poleg značilk (angl. attribute) podamo tudi razred (angl. class) za vsak primer. Označevanje podatkov se izvaja na podatkih zajetih s pomočjo krmilnika Arduino in na podatkih, pridobljenih s tablice Android. Podatki, ki so pridobljeni s pomočjo naprave Shimmer, se neposredno ne označijo, ker je to tehnično težje izvedljivo.

Tipične oznake so:

- Ime osebe, ki prehaja skozi vrata.
- Brez oznake — skozi vrata ne prehaja nihče oziroma nekdo, čigar prehoda ne želimo posneti.
- Napaka — v primeru, da je začetek posnetka napačno označen z imenom osebe, ki dejansko ni prešla skozi vrata, oziroma je skozi vrata prešla neka druga oseba, se podatek označi kot napaka in se kasneje ne upošteva.

5.1 Označevanje podatkov, pridobljenih preko Arduino krmilnika

Med zajemanjem pospeškov preko krmilnika Arduino se označevanje izvaja avtomatično. Ker se mora uporabnik pred prehodom skozi vrata registrirati

preko čitalca RFID oziroma čitalca prstnih odtisov, je znano, katera oseba je prešla skozi vrata, iz sensorja odprtosti pa se razbere, kdaj so se vrata odprla. Ta informacija se uporabi pri zajemu pospeškov. Z imenom osebe, ki se pred prehodom identificira na čitalcu, se označijo podatki od dveh sekund pred odprtjem vrat do dveh sekund po zaprtju. Tolikšna razlika je potrebna, ker senzor odprtosti vrat zaradi načina delovanja zazna, da so bila vrata odprta, šele ko so vrata že nekoliko odprta. Sorodno zazna zaprtost, nekoliko preden se vrata dokončno zaprejo.

Pospeške pred odprtjem se zabeleži tudi zaradi tega, ker se človek kljuge dotakne, preden se vrata odprejo, in s tem že vpliva na vrata. Podatke po zaprtju potrebujemo, ker se vrata takrat še vedno iznihujejo, če so bila zaprta sunkovito.

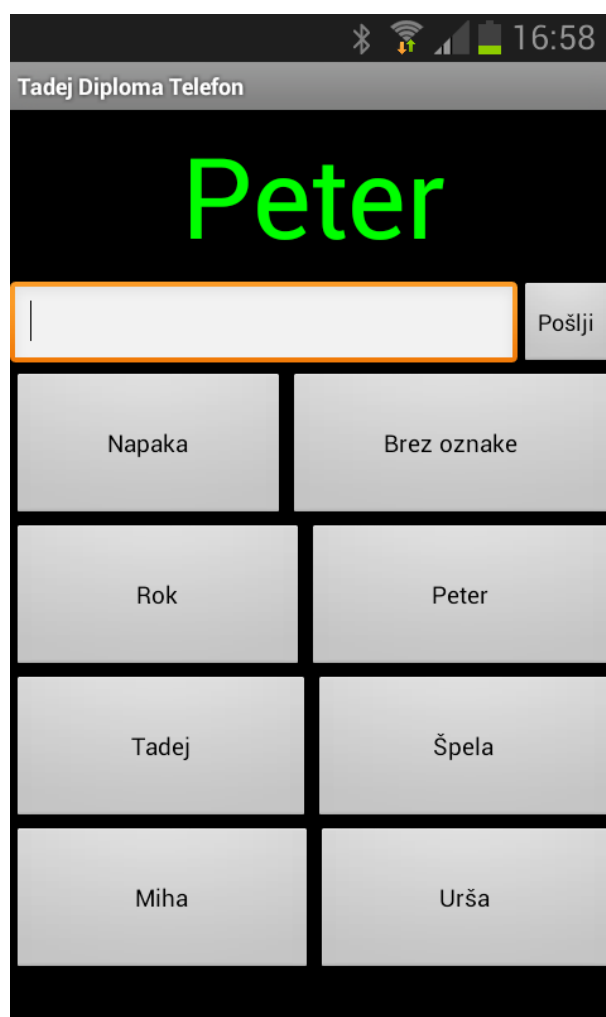
5.2 Označevanje podatkov, pridobljenih s tablice Android

Označevanje podatkov, zajetih s tablico Android, se izvaja ročno, ko je namen posneti testne ali učne podatke. Za namen zapisovanja oznak sem napisal program za telefon (slika 5.1), ki poganja operacijski sistem Android in lahko preko povezave Bluetooth komunicira s tabličnim računalnikom, ki zajema in shranjuje podatke s sensorja pospeškov.

Tablični računalnik in telefon je treba najprej spariti (angl. pair), kar omogoči komunikacijo med njima. Za tem ju je treba povezati tako, da se na enem izmed njiju odpre meni opcij, izbere novo povezavo in nato s ponujenega seznama izbere drugo napravo.

Ko je povezava ustvarjena, se na telefonu izpiše oznaka, ki je trenutno nastavljena na tabličnem računalniku. V primeru, da je povezava vzpostavljena, je ta napis zelene barve, če se povezava prekine, se napis obarva rdeče, kar opozarja na to, da nove oznake ne bodo uspešno poslane in da ni nujno, da se oznaka ne telefonu ujema z oznako na tabličnem računalniku.

Na telefonu je tekstovno polje za ročni vpis oznake, na voljo pa so tudi



Slika 5.1: Program za telefon Android, namenjen oznaevanju podatkov, zajetih s tabličnim računalnikom.

gumbi z vnaprej nastavljenimi oznakami. Novo oznako se nastavi tako, da se v tekstovno polje vpiše željeno besedilo in pritisne gumb Pošlji. Napis na telefonu se osveži šele, ko od tabličnega računalnika prejme povratno informacijo, da je bila oznaka uspešno poslana. Ta oznaka se od tega trenutka dalje dodaja vsem zajetim pospeškom.

Poglavje 6

Opis postopka zajema podatkov

Za razvoj in testiranje algoritma za identifikacijo oseb ob prehodu skozi vrata so bili zajeti podatki s sistemom za zajem le-teh, opisanem v poglavju 3. Z delujočim sistemom za zajem podatkov so se lahko začele meritve. Podatki so se zajemali v treh fazah. Prvi set podatkov (poglavje 6.1) je bil zajet za potrebe analize prehoda skozi vrata. Drugi set podatkov (poglavje 6.2) je bil zajet za potrebe izgradnje klasifikatorjev. Namen tretjega seta podatkov je pridobiti primere prehodov skozi vrata, ko se oseba, ki prehaja skozi vrata, ne zaveda snemanja (poglavje 6.3).

6.1 Snemanje prehodov skozi vrata za namen analize prehoda skozi vrata

Namen snemanja simuliranih prehodov je posneti vse možne načine prehodov. To je najhitrejši način, kako priti do podatkov, saj za to priprave niso potrebne in mogoče je v kratkem času posneti veliko število takšnih prehodov. Namen je dobiti primere vseh različnih prehodov za namen analize in razvoja algoritma za identifikacijo oseb pri prehodu skozi vrata. Na ta način je bilo posnetih 72 prehodov v 40 minutah.

Snemanje simuliranih prehodov je bilo izvedeno po vnaprej pripravljene scenariju, ki je vseboval 12 načinov prehodov po tri ponovitve v vsako smer.

Prednost tega načina zbiranja podatkov je, da omogoča pridobitev velike količine prehodov, tudi nekaj na minuto. Druga pomembna prednost je, da se lahko na ta način posname več različnih načinov prehodov. En primer razlike med prehodi je, če se vrata skoraj popolnoma ustavitvijo, preden se zaprejo, ali se zaprejo z loputanjem na koncu prehoda. Drug primer je, če se vrata ustavijo ob maksimalnem kotu odprtosti ali ne. Slabost tako pridobljenih podatkov je, da se lahko razlikujejo od podatkov, ko oseba, ki izvaja prehode, ni pozorna na sam prehod skozi vrata.

Ti podatki so bili uporabljeni za razvoj algoritma za identifikacijo oseb ob prehodu skozi vrata, na podlagi teh podatkov so bile določene faze prehoda, njihove značilnosti in postopek za določanje meje med fazami prehoda ter delitev posameznih faz v podfaze, ki so bile kasneje uporabljene kot osnovne enote za izračun značilk.

6.2 Snemanje prehodov skozi vrata za namen izgradnje klasifikatorja

Po analizi simuliranih prehodov so bili potrebni realni podatki. Ker pravočasno ni bilo mogoče pridobiti dovolj dolgoročnih podatkov, je bilo snemanje izvedeno s skupino 14 prostovoljcev, katerih naloga je bila hoditi skozi vrata, hkrati pa niso smeli vedeti, da gre v resnici za snemanje prehodov skozi vrata, da ne bi zavestno ali podzavestno vplivali na način svojih prehodov skozi vrata.

V ta namen sem pripravil seznam naključnih črk in števil, ločenih v skupine po pet znakov, in ga pustil na mizi v prostoru za vrata. Na hodniku, nekaj metrov od vrat, sem postavil blok papirjev in pisalo (slika 6.1). Prostovoljcem sem dal navodilo, naj si poskušajo zapomniti čim več znakov iz seznama in jih prepisati na blok papirja na hodniku. Medtem, ko so prostovoljci prepisovali številke, sem s pomočjo aplikacije za telefon označeval prehode. Vsaka oseba je naredila od 20 do 100 prehodov v vsako smer. Na ta način je bilo posnetih 1037 prehodov v 20 urah snemanja, razporejenih

posamezne osebe dnevno, posledično pa je frekvenca zajema bistveno nižja kot pri prehodih, opisanih v poglavjih 6.1 in 6.2. Dolgoročno snemanje je vir najkvalitetnejših podatkov, na katerih je mogoče izvesti objektivno preverjanje kvalitete delovanja različnih algoritmov za identifikacijo oseb ob prehodu skozi vrata. V času pisanja diplomske naloge ti podatki še niso na voljo.

Poglavje 7

Vizualizacija in analiza tipičnih vzorcev

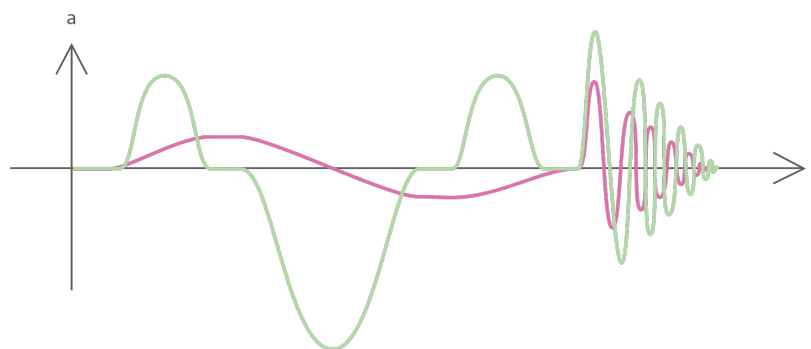
V tem poglavju je opisan postopek, ki je bil uporabljen za prikaz in analizo zbranih podatkov o prehodih.

Podatke o simuliranih prehodih sem najprej ločil v datoteke tako, da je bil vsak prehod v ločeni datoteki, ter jih prikazal na grafu. Na eni osi grafa sem prikazal čas zajema podatkov, na drugi osi pa vrednosti pospeškov posameznih smeri. Za izris grafov sem uporabil program Microsoft Excel.

Nekatere razlike med prehodi so opazne že na pogled. Med analizo sem določil lastnosti prehodov, ki imajo med primeri prehodov iste osebe podobne vrednosti, razlikujejo pa se med prehodi različnih oseb. Glavne razlike so bile v moči sunka ob zaprtju vrat, času odpiranja in zapiranja vrat ter v tem, ali so se vrata, medtem ko so bila odprta za največji kot, gibala pospešeno ali ne. Tem lastnostim sem kasneje namenil več pozornosti pri ločevanju oseb na podlagi pravil.

Primeri odpiranja, ko na vrata, medtem ko so odprta za največji kot, deluje sila, in primeri, ko sila v tem primeru ne deluje, se bistveno razlikujejo.

Tipičen primer grafa pospeškov pri prehodu, ko na vratno krilo, medtem ko so odprta za največji kot, ves čas deluje sila, je razviden na sliki 7.1. Tipičen primer grafa pospeškov pri prehodu, ko na vrata, medtem ko so

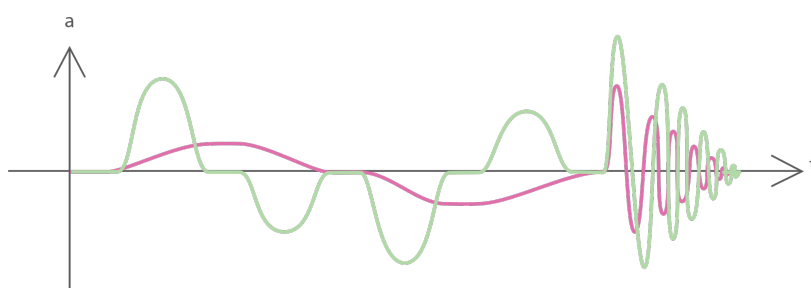


Slika 7.1: Graf pospeškov, ko oseba z istim gibom ustavi vratno krilo in začne zapirati vrata. Zelena krivulja predstavlja tangencialni pospešek, rdeča krivulja pa kotno hitrost. Na osi x je čas, označen s t , na osi y pa je velikost pospeška, označena z a .

odprta za največji kot, ne deluje sila, pa na sliki 7.2.

Vsak prehod je sestavljen iz več faz:

- Vrata mirujejo — edini izmerjeni pospešek je gravitacijski.
- Vrata se pospešeno odpirajo — kotna hitrost se povečuje, kot odprtosti se povečuje.
- Vrata se ustavljajo — kotna hitrost se zmanjšuje, kot odprtosti se povečuje.
- Vrata mirujejo — kotna hitrost je enaka nič, kot odprtosti je največji.
- Vrata se pospešeno zapirajo — kotna hitrost se povečuje v obratni smeri, kot odprtosti se zmanjšuje.
- Vrata se ustavljajo — kotna hitrost se zmanjšuje, kot odprtosti se zmanjšuje.
- Vrata se zaprejo (lahko se zaloputnejo) — kotni pospešek in kotna hitrost iznihata, kot odprtosti oscilira okrog nič.



Slika 7.2: Graf pospeškov, ko oseba med prehodom preneha s silo delovati na vratno krilo. Zelena krivulja predstavlja tangencialni pospešek, rdeča krivulja pa kotno hitrost. Na osi x je čas, označen s t , na osi y pa je velikost pospeška, označena z a .

Poglavje 8

Algoritem za detekcijo zanimivih dogodkov

Eden izmed algoritmov, ki sem jih izdelal za potrebe analize dogodkov z vrati, je tudi algoritem, ki na podlagi podatkovnega toka izmerjenih pospeškov ugotovi, ali se je v času merjenja podatkov zgodil dogodek, povezan z vrati, ali pa podatki vsebujejo le šum. To je potrebno, ker je algoritem za prepoznavanje oseb ob prehodu procesorsko zahteven in zaradi učinkovitosti ni smiselno izvajati algoritma nad podatki, pri katerih vrata mirujejo.

Algoritem deluje tako, da v časovnem oknu 0,15 sekunde okrog vsake meritve izračuna standardni odklon ter na podlagi le-tega in absolutne vrednosti označi točke, ki imajo večjo verjetnost zanimivosti. Tako označene podatke obdela s histerezo [20], ki daje prednost vrednostim, ki so označene kot zanimive.

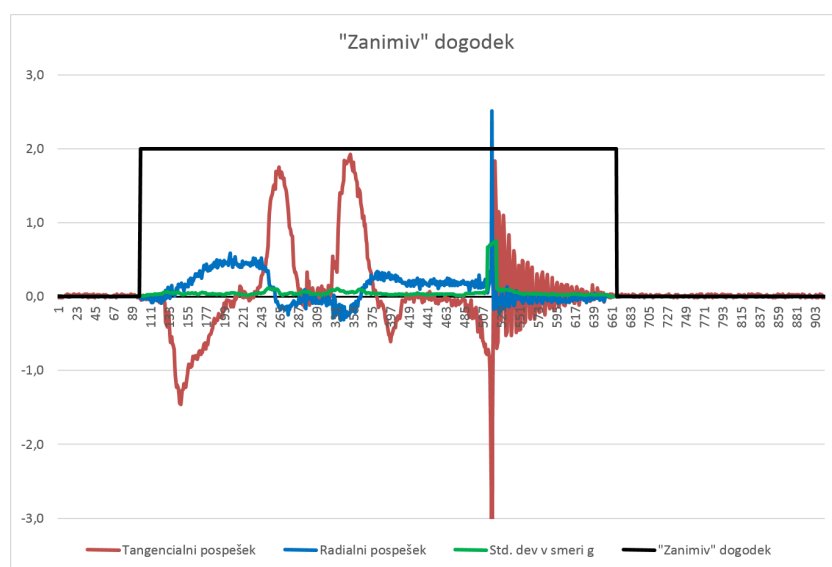
Rezultat je viden na sliki 8.1, kjer je s krivuljo rdeče barve prikazan tangencialni pospešek, s krivuljo modre barve radialni pospešek, z zeleno krivuljo pa izračunani standardni odklon kotnega pospeška. S črno barvo je označeno območje, kjer je algoritem zaznal zanimiv dogodek — v tem primeru prehod skozi vrata.

Psevdokoda algoritma za detekcijo zanimivih dogodkov:

```
nova_vrednost = šum
stara_vrednost = šum
št_zaporednih = 0
za vse točke {
    če (katerikoli_podatek > vrednost_šuma) nova_vrednost = ni_šum
    v nasprotnem primeru nova_vrednost = šum

    če (stara_vrednost != nova_vrednost) št_zaporednih = 0
    če (stara_vrednost == nova_vrednost) št_zaporednih += 1
    če (nova_vrednost == ni_šum) št_zaporednih +=2

    če (št_zaporednih >= 10) {
        stara_vrednost = nova_vrednost
        če (nova_vrednost == ni_šum) označi podatek kot zanimiv
        v nasprotnem primeru označi podatek kot nezanimiv
    }
    v nasprotnem primeru {
        če (stara_vrednost = ni_šum) označi podatek kot zanimiv
        v nasprotnem primeru označi podatek kot nezanimiv
    }
}
```



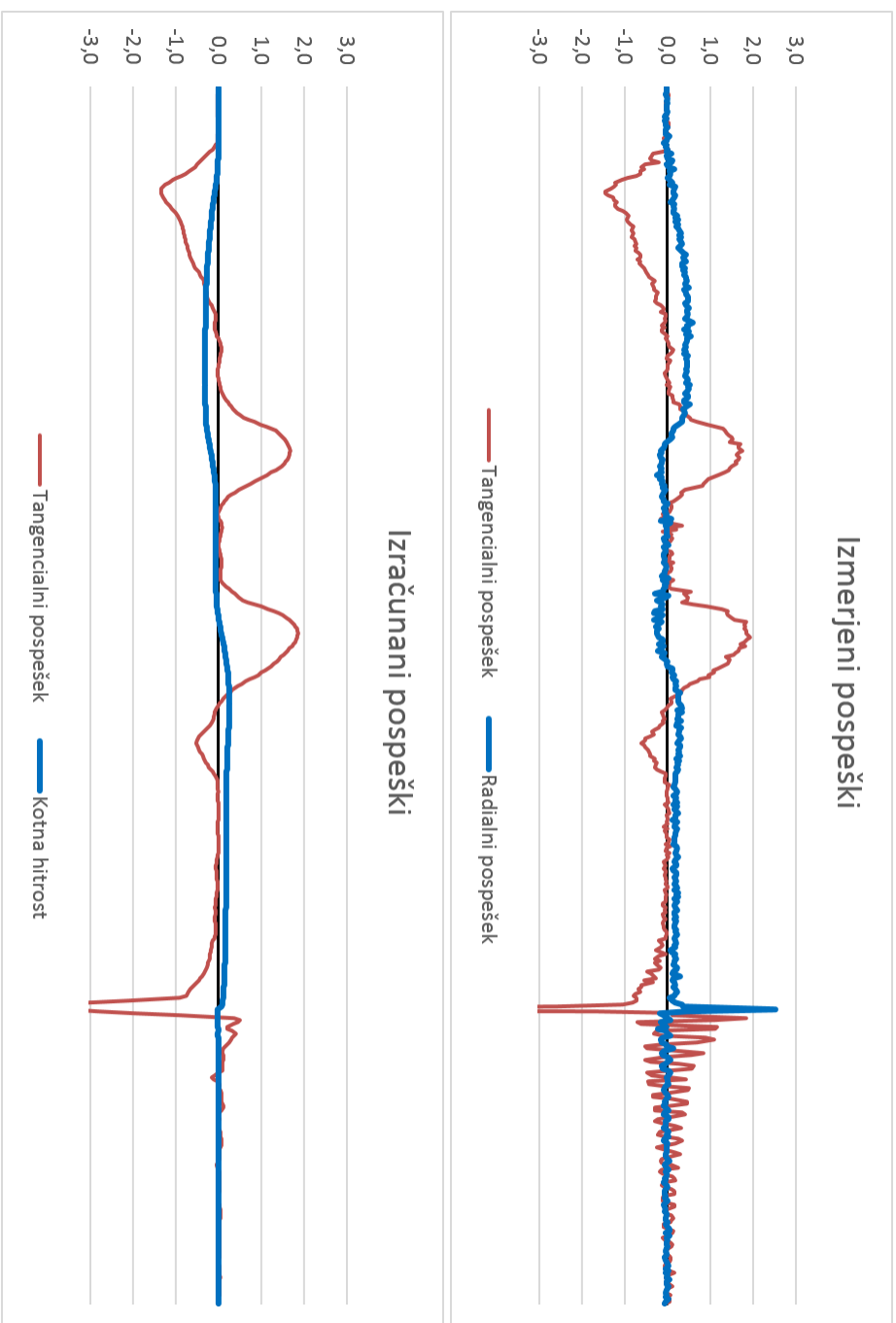
Slika 8.1: Primer rezultata algoritma za detekcijo zanimivih dogodkov.

Poglavje 9

Predobdelava signala

Podatki, zajeti s pospeškomeri, vsebujejo šum, zato jih je treba filtrirati. S pospeškomeri lahko neposredno izmerimo le vrednosti pospeškov v treh smereh in tako dobimo: gravitacijski pospešek ($a = g$), radialni pospešek ($a = \omega^2 r$) in tangencialni pospešek ($a = r\alpha$), kjer je ω kotna hitrost, α kotni pospešek in r razdalja od osi vrtenja. Algoritem za identifikacijo oseb ob prehodu skozi vrata pa potrebuje tudi kotno hitrost, kot odprtosti vrat in standardni odklon pospeška v smeri gravitacijskega pospeška, ki jih je treba izračunati iz izmerjenih pospeškov med odpiranjem vrat.

Največji pospeški, poleg gravitacijskega pospeška, so zaznani v smeri tangencialnega pospeška. Posledično ima podatek o kotnem pospešku najugodnejše razmerje med signalom in šumom ter nosi bistvene informacije o načinu odpiranja vrat. Zaradi tega temelji predobdelava signala predvsem na tem podatku. Tipičen rezultat filtriranih pospeškov v primerjavi z izmerjenimi je razviden na sliki 9.1.



Slika 9.1: Primerjava izmerjenih in izračunanih podatkov — opazna razlika v predznaku med radialnim pospeškom in kotno hitrostjo.

9.1 Izračun kotne hitrosti in kompenzacija šuma

Izmerjen pospešek v radialni smeri vsebuje informacijo o kotni hitrosti: odvisen je od oddaljenosti senzorja od vratnega tečaja in kvadrata kotne hitrosti, torej ne vsebuje informacije o smeri rotacije vratnega krila.

Kotna hitrost se lahko izračuna tudi po enačbi (9.1) kot integral kotnega pospeška, kjer t predstavlja čas. Zaradi šuma v podatkih se napake kopičijo in tako vrednost izračunane kotne hitrosti sčasoma vedno bolj odstopa od dejanske vrednosti. Ker predpostavljamo, da so ob koncu zaporedja podatkov, izločenega z algoritmom, opisanem v poglavju 8, vrata zaprta, da je kotna hitrost enaka nič in šum naključen, lahko akumulirano napako odstranimo tako, da krivuljo linearno nagnemo in s tem dobimo boljše približke. Pri tem smo zanemarili dejstvo, da je velikost šuma odvisna tudi od izmerjene vrednosti.

$$\omega = \int \alpha dt. \quad (9.1)$$

9.2 Izračun kotnega pospeška

Ko poznamo kotno hitrost, lahko iz nje izračunamo kotni pospešek. Ta se po enačbi (9.2), kjer N predstavlja dolžino okna kot število točk, izračuna kot smerni koeficient linearne interpolacije kotne hitrosti po metodi najmanjših kvadratov v oknu dolžine 0,15 sekunde. Število točk v oknu se lahko izračuna kot $N = ft$, kjer predstavlja f frekvenco zajema podatkov in t dolžino okna v sekundah.

$$\alpha_i = \frac{\sum t_i \omega_i - \frac{1}{N} \sum t_i \sum \omega_i}{\sum t_i^2 - \frac{1}{N} (\sum t_i)^2} \quad (9.2)$$

9.3 Popravek kompenzacije šuma

S pomočjo izračunane kotne hitrosti in kotnega pospeška se lahko določi območje, v katerem so prisotni večji pospeški in posledično večji šum. Na osnovi tega lahko ponovno kompenziramo šum tako, da krivuljo nagnemo v odvisnosti od velikosti kotne hitrosti in kotnega pospeška.

9.4 Izračun kota odprtosti vrat

Kot odprtosti vrat je enak prepotovani poti vratnega krila. Prepotovana pot se po enačbi (9.3) izračuna kot integral kotne hitrosti. Zaradi integracije je tudi za ta podatek potrebna kompenzacija šuma ob predpostavki, da so vrata ob koncu prehoda zaprta. Slednjo naredimo tako, da krivuljo v območju, ko so vrata odprta, linearno nagnemo tako, da je v točki zaprtja vrat kot odprtosti enak nič.

$$\phi = \int \omega dt. \quad (9.3)$$

9.5 Izračun standardnega odklona pospeška v smeri gravitacije

Zadnji signal, ki ga potrebujemo, je standardni odklon pospeška v smeri gravitacijskega pospeška. Ker v tej smeri deluje le gravitacijski pospešek, pričakujemo, da se bodo na tem signalu poznale vibracije vrat. Standardni odklon je mera razpršenosti vrednosti v okolici točke. Izračuna se kot kvadratni koren vsote kvadratov odklonov vrednosti od povprečja.

Standardni odklon v točki se izračuna po formuli (9.4), kjer a predstavlja pospešek v smeri gravitacije.

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (a_i - \bar{a})^2} \quad (9.4)$$

Poglavje 10

Faze prehoda

Za vsak prehod skozi vrata je na voljo množica podatkov, ki so sicer natančni, a je iz njih težko razbrati uporabne informacije. V ta namen lahko celoten dogodek razdelimo na več faz, ki so del prehoda in so opazne vsakokrat, ko nekdo stopi skozi vrata (slika 10.1). Nekatere faze so očitne, nekatere se skoraj ne pojavijo, nekatere pa so prisotne le pri prehodih določenih oseb. Idejo za ločevanje prehoda na odpiranje in zapiranje sem dobil v osnutku patentne prijave [9]. Osnutek patentne prijave [9] uporablja tudi podobno delitev na nivoje.

Faze smo razdelili na 4 nivoje:

- Prvi nivo (začetek in konec prehoda):
 - Začetek prehoda (točka 1a na sliki 10.1) — v tem trenutku se v podatkih zazna aktivnost, ki se razlikuje od šuma.
 - Konec prehoda (točka 1b na sliki 10.1) — v tem trenutku se v podatkih ne zazna več nobena aktivnost razen šuma.
- Drugi nivo (odpiranje in zapiranje):
 - Vrata se odpirajo (točka 2a na sliki 10.1) — v tem trenutku človek s silo deluje na vrata, povečajo se kotni pospešek, kotna hitrost in kot odprtosti vrat. Začne se lahko hkrati kot začetek prehoda s prvega nivoja ali kasneje.

- Vrata mirujejo in so maksimalno odprta (točka 2b na sliki 10.1) — v tem trenutku je kotna hitrost enaka nič, kot odprtosti vrat je največji, sila, s katero človek deluje na vrata, pa je lahko prisotna ali ne.
 - Vrata se zapirajo (točka 2c na sliki 10.1) — v tem obdobju človek s silo deluje na vrata, povečata se kotni pospešek in kotna hitrost, kot odprtosti vrat pa se začne zmanjševati. Točka je lahko enaka stanju, ko vrata mirujejo, v primeru, ko vrata v istem gibu ustavimo in zapremo.
 - Vrata so zaprta (točka 2d na sliki 10.1) — v tem trenutku se kot odprtosti vrata, kotni pospešek in kotna hitrost zmanjšajo na nič. Ta točka je lahko enaka kot točka za konec prehoda prvega nivoja ali pa se pojavi prej.
- Tretji nivo (deli odpiranja in zapiranja):
 - Vrata pospešeno odpiramo (točka 3a na sliki 10.1) — točka je enaka stanju, ko se vrata odpirajo z drugega nivoja.
 - Na vrata ne delujemo s silo (točka 3b na sliki 10.1) — v tej točki se kotni pospešek približa nič, vrata se odpirajo z enakomerno hitrostjo.
 - Vrata ustavljamo (točka 3c na sliki 10.1) — v tej točki se kotni pospešek poveča, a v obratni smeri, kot je bil prej.
 - Vrata pospešeno zapiramo (točka 3d na sliki 10.1) — točka je enaka stanju, ko se vrata zapirajo z drugega nivoja.
 - Na vrata ne delujemo s silo (točka 3e na sliki 10.1) — v tej točki se kotni pospešek približa nič, vrata se zapirajo z enakomerno hitrostjo.
 - Vrata ustavljamo (točka 3f na sliki 10.1) — v tej točki se kotni pospešek poveča, a v obratni smeri, kot je bil prej.
 - Četrty nivo (lokalni minimumi in maksimumi):

- Največji kotni pospešek med odpiranjem vrat (točka 4a na sliki 10.1).
- Največji kotni pospešek med ustavljanjem odpiranja vrat (točka 4b na sliki 10.1).
- Največji kotni pospešek med zapiranjem vrat (točka 4c na sliki 10.1).
- Največji kotni pospešek med ustavljanjem zapiranja vrat (točka 4d na sliki 10.1).
- Največji kotni pospešek med iznihovanjem (točka 4e na sliki 10.1).

10.1 Ločevanje začetka in konca prehoda

Namen prvega nivoja je odkriti, kdaj se dogaja prehod skozi vrata.

Vhodni podatki so vsi podatki od dveh sekund pred prehodom do dveh sekund po koncu prehoda. Med temi podatki iščemo prvo in zadnjo točko z vrednostjo, različno od šuma. To storimo tako, da z zanko *while* preverjamo podatke od začetka do časa, ko absolutna vrednost izračunanega kota odprtosti vrat, izračunane kotne hitrosti, izračunanega kotnega pospeška ali izračunanega standardnega odklona pospeška v smeri gravitacijskega pospeška ne preseže vrednosti meje šuma za ta tip podatka. To je točka, kjer se prehod začne.

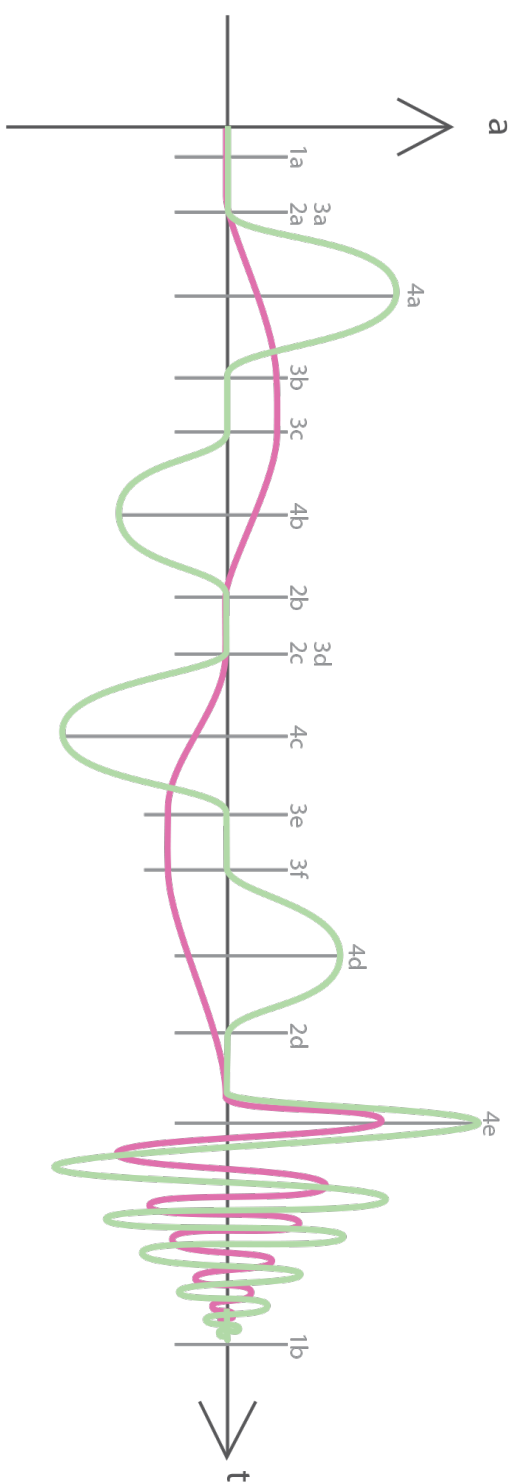
Enako ponovimo od konca proti začetku, da najdemo še zadnjo točko. V tej točki je prehoda konec.

10.2 Ločevanje odpiranja in zapiranja

Namen drugega nivoja je odkriti, kdaj se vrata odpirajo in kdaj zapirajo.

Vhodni podatki so vsi podatki med začetkom in koncem prehoda. Najprej se sprehodimo čez vse podatke ter poiščemo maksimalen kot odprtosti vrat.

Nato se še enkrat sprehodimo skozi podatke, s tem, da najprej preverjamo, če so vrata odprta. Ko zaznamo, da je kot odprtosti vrat nad mejo šuma, to točko vzamemo za začetek odpiranja vrat.



Slika 10.1: Faze prehoda skozi vrata. Zelena krivulja predstavlja tangencialni pospešek, rdeča krivulja pa kotno hitrost. Na osi x je čas, označen s t , na osi y pa je velikost pospeška, označena z a .

Ko vrata dosežejo največji kot odprtosti, so v točki, v kateri so najbolj odprta in se od tam naprej ne odpirajo več, torej na tem mestu mirujejo.

Enako ponovimo v obratni smeri. Ko se kot odprtosti vrat zmanjša od maksimalnega kota za vrednost, ki je večja od meje šuma, se ta točka izbere za začetek zapiranja vrat.

Ko se kot odprtosti vrat približa nič in je kotna hitrost pod mejo šuma, točko izberemo za stanje, ko so vrata zaprta. Po tej točki je možno izniho vanje vrat v primeru, da so bila vrata zaloputnjena.

10.3 Ločevanje delov odpiranja in zapiranja

Namen tretjega nivoja je odkriti, kdaj in kako vplivamo na vrata med prehodom. Vhodni podatki so podatki med točko, ko se vrata začnejo odpirati, dokler se ne ustavijo, in med točko, ko se vrata začnejo zapirati, dokler niso zaprta.

Pospešeno odpiranje vrat zaznamo že v prejšnji fazi, zato je prva točka tretjega nivoja enaka prvi točki drugega nivoja.

Točko, kjer na vrata med odpiranjem ne delujemo več s silo, in točko, kjer vrata ustavljamo, iščemo med točkama, kjer se vrata začnejo odpirati in kjer so vrata odprta za največji kot. Točka, kjer na vrata ne delujemo več s silo, je tam, kjer pade kotni pospešek vratnega krila pod mejo šuma. Točka, kjer vrata ustavljamo, pa tam, kjer se kotni pospešek vratnega krila spet dvigne iz meje šuma, a tokrat v obratni smeri.

Točka, kjer vrata pospešeno zapiramo, je enaka točki, kjer se vrata zapirajo iz drugega nivoja, zato to vrednost kar nastavimo na vrednost prej zaznane točke.

Točki, kjer na vrata ne delujemo več s silo in kjer vrata ustavljamo, zaznamo na enak način kot v primeru odpiranja, le da je smer kotnega pospeška in kotne hitrosti obratne vrednosti. Ti točki iščemo med točkama, ko se vrata zapirajo in ko so vrata zaprta.

10.4 Iskanje lokalnih ekstremov

Namen četrtega nivoja je ugotoviti, kdaj v vsaki fazi na vrata vplivamo z največjo silo. Vhodni podatki so podatki med odpiranjem vrat, med ustavljanjem vrat, med zapiranjem vrat, med ustavljanjem vrat in med iznihovanjem, ko na vrata s silo deluje vratni okvir.

Točke se izračuna tako, da se za vsako območje poišče ekstremno vrednost in shrani indeks točke, kjer se ta vrednost nahaja.

Poglavje 11

Seznam značilk in njihov pomen

Strojno učenje deluje na principu prepoznavanja vzorcev in iskanja pravil v učnih podatkih. Učne podatke za nadzorovano strojno učenje predstavimo s primeri, opisanimi z eno ali več značilkami, in razredom, ki mu posamezen primer pripada. Glavni namen strojnega učenja je na podlagi učnih podatkov odkriti splošno pravilo, s katerim lahko podatkom, ki niso v učni množici, določimo, kateremu razredu pripadajo. V primeru identifikacije oseb, ki prehajajo skozi vrata, so značilke značilnosti prehoda, razred pa je ime osebe.

Ko so znane meje med fazami, lahko izračunamo trajanje vsake faze. Trajanja vsake faze predstavlja po eno značilko za strojno učenje. Za uspešno klasifikacijo je potrebnih čim več značilk, ki prispevajo uporabne informacije.

Ostale značilke, ki jih lahko izračunamo, so ekstremna vrednost, smerni koeficient linearne interpolacije po metodi najmanjših kvadratov in integral za vsako fazo za izmerjeno kotno hitrost, izmerjen kotni pospešek, izračunana kotna hitrost, izračunan kotni pospešek, izračunan kot odprtosti vrat ter standardni odklon pospeška v smeri gravitacijskega pospeška. Podobno kot v diplomskem delu uporablja osnutek patentne prijave [9] ekstremno vrednost, trajanje, integral in smerni koeficient linearne interpolacije.

Skupno število izračunanih značilk je:

$14 \text{ odsekov} \cdot 6 \text{ podatkov} \cdot 3 \text{ lastnosti} + 14 \text{ trajanj odseka} = 266 \text{ značilk.}$

Odseki, za katere so bile izračunane značilke:

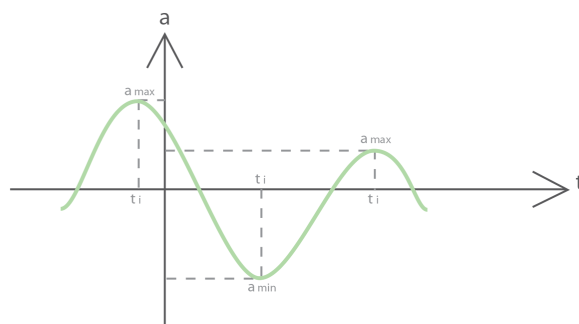
1. Začetek prehoda (od začetka prehoda do začetka odpiranja vrat).
2. Povečevanje sile, s katero oseba med odpiranjem pospešuje vratno krilo.
3. Zmanjševanje sile, s katero oseba med odpiranjem pospešuje vratno krilo.
4. Oseba na vrata med odpiranjem ne deluje s silo.
5. Povečevanje sile, s katero oseba med odpiranjem ustavlja vratno krilo.
6. Zmanjševanje sile, s katero oseba med odpiranjem ustavlja vratno krilo.
7. Oseba na vrata ne deluje s silo — tu so vrata odprta za največji kot.
8. Povečevanje sile, s katero oseba med zapiranjem pospešuje vratno krilo.
9. Zmanjševanje sile, s katero oseba med zapiranjem pospešuje vratno krilo.
10. Oseba na vrata med zapiranjem ne deluje s silo.
11. Povečevanje sile, s katero oseba med zapiranjem ustavlja vratno krilo.
12. Zmanjševanje sile, s katero oseba med zapiranjem ustavlja vratno krilo.
13. Konec prehoda (od zaprtja vrat do največje sile med iznihovanjem).
14. Iznihovanje.

Podatki, za katere so bile izračunane značilke:

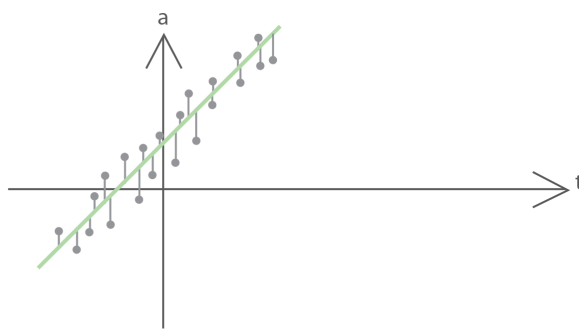
1. Izmerjena kotna hitrost.
2. Izmerjen kotni pospešek.
3. Izračunana kotna hitrost.
4. Izračunan kotni pospešek.
5. Izračunani kot odprtosti vrat.
6. Izračunani standardni odklon pospeška v smeri gravitacijskega pospeška.

Lastnosti podatkov:

1. Ekstremna vrednost.
2. Linearna interpolacija.
3. Integral.



Slika 11.1: Trije primeri lokalnih ekstremnih vrednosti.



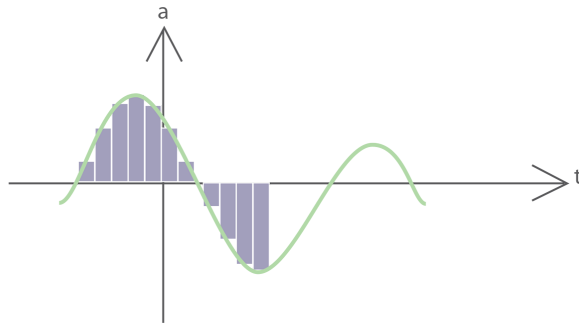
Slika 11.2: Linearna interpolacija.

Ekstremna vrednost (slika 11.1) je največja absolutna vrednost podatka v podanem oknu.

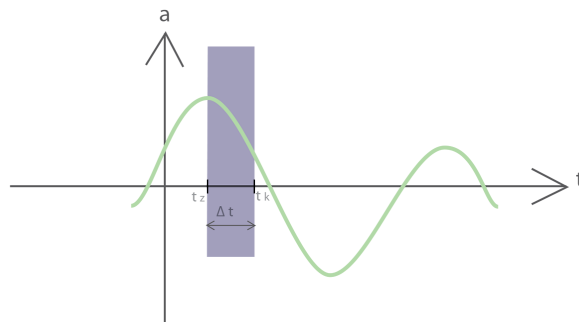
Linearna interpolacija (slika 11.2) je metoda prilagajanja krivulje podatkom. Pri linearni interpolaciji po metodi najmanjših kvadratov podatkom prilagajamo premico. Vrednost, ki nas zanima, je smerni koeficient oziroma naklon premice.

Integral (slika 11.3) je ploščina lika pod krivuljo oziroma seštevek vseh vrednosti v infinitezimalno kratkih odsekih. Ker je v našem primeru frekvenca zajema podatkov znana in konstantna, se lahko poenostavi kot seštevek vseh vrednosti.

Trajanje (slika 11.4) je razlika v času med začetkom in koncem faze. Izračuna se na podlagi indeksa točk in frekvence zajema podatkov.



Slika 11.3: Integral.



Slika 11.4: Trajanje.

Poglavje 12

Rezultati

Za potrebe diplomske naloge sem posnel 1037 primerov prehodov skozi vrata 14 različnih oseb. Prehodi so bili posneti v obdobju med 1. 7. 2013 in 31. 8. 2013 na vratih pisarne A224 na Institutu "Jožef Stefan" po metodi opisani v poglavju 6.2.

Pri dveh osebah meritve niso dale konsistentnih rezultatov, zato teh prehodov nisem vključil v množico podatkov. Pri prvi osebi se je pojavil problem med zajemom podatkov, saj je sredi izvajanja meritev oseba ugotovila pravi namen snemanja, bila za tem pozorna na sam prehod skozi vrata in je s tem vplivala na rezultat. Meritve druge osebe sem izločil zaradi tega, ker so se pogoji meritve med snemanjem bistveno spremenili. Ob začetku snemanja sva bila v prostoru samo ta oseba in jaz, med snemanjem pa je vstopilo več ljudi, ki so s svojo prisotnostjo vplivali na obnašanje te osebe, zato bi bilo za uspešno učenje algoritmov potrebno večje število prehodov, kot jih je bilo dejansko posnetih. Predvidoma bi bila z dovolj veliko količino posnetkov mogoča klasifikacija tudi oseb, katerih način prehajanja se je med meritvami spremenil. Po tako izločenih podatkih je bilo na voljo 897 prehodov 12 različnih oseb.

12.1 Izbor značilke in klasifikatorja

Ker je med izračunanimi značilkami tudi nekaj takih, ki ne vsebujejo informacije relevantne za identifikacijo oseb oziroma ob prisotnosti ostalih značilk dodatno ne pripomorejo k točnosti ločevanja med osebami, jih je bilo treba izločiti, saj lahko te značilke zmanjšajo klasifikacijsko točnost in povečajo čas učenja.

Kvaliteto vsake značilke sem izračunal s pomočjo uveljavljenih mer, ki za vsako značilko izračunajo oceno, koliko težja bi bila klasifikacija brez te značilke oziroma koliko značilka pripomore k pravilni klasifikaciji. Uporabil sem ReliefF [15, 17, 16], informacijski pribitek [12, 13] (angl. information gain), faktor informacijskega pribitka (angl. information gain ratio), gini ojačenje [16] (angl. Gini gain), MARS pomembnost [8] (angl. multivariate adaptive regression splines), uteži linearne metode podpornih vektorjev [19] (angl. linear SVM weights) in naključne gozdove [1] (angl. random forests).

Na podlagi zajetih prehodov je bila kvaliteta značilk ocenjena z vrednostmi, razvidnimi iz tabele 12.1. Ekstremnega izračunanega kotnega pospeška v fazi 6 — značilke, ki je najbolj pripomogla h klasifikacijski točnosti modela (ocenjena z oceno 4,2 po meri MARS pomembnost) — na seznamu ni, ker ima nižjo oceno po meri ReliefF. Ravno tako na seznamu ni naklona izračunanega kotnega pospeška v fazi 9, ki je po meri SVM utež ocenjen z oceno 7,798, po drugi strani pa sta na seznamu integral izračunane kotne hitrosti v fazi 9 in ekstremni izmerjeni kotni pospešek v fazi 12, ki imata dobro oceno po meri ReliefF, a sta povprečno ocenjena po ostalih merah.

Značilke sem razporedil po vsaki meri kvalitete. Eksperimentalno sem ugotovil, da so rezultati najboljši, ko sem izbral 50 najboljše ocenjenih značilk po meri linearne SVM uteži, 30 najboljše ocenjenih značilk po meri naključnih gozdov in ReliefF ter 20 najboljše ocenjenih po meri MARS pomembnost, informacijskega pribitka, faktorja informacijskega pribitka ter gini ojačenja. Skupaj sem izbral 97 od 266 izračunanih značilk.

Z izbranimi značilkami sem z uporabo orodja RadViz v programu Orange [7] na podlagi kombinacije petih značilk, ki so po rezultatu funkcije VizRank naj-

Značilka	Relieff	I. pribitek	Faktor i. p.	Gini o.	MARS p.	SVM utež	N. gozdovi
Ekstremni izm. k. pospešek v fazi 11	0,160	0,681	0,340	0,054	2,800	5,161	1,486
Ekstremni izr. k. pospešek v fazi 11	0,127	0,648	0,324	0,052	0,700	1,222	0,769
Ekstremni izm. k. pospešek v fazi 13	0,109	0,531	0,265	0,047	0,000	1,162	1,005
Ekstremna izm. k. hitrost v fazi 5	0,109	0,518	0,259	0,031	1,800	6,684	2,624
Naklon izr. k. hitrosti v fazi 8	0,106	0,780	0,390	0,056	0,000	5,319	0,867
Ekstremni izm. k. pospešek v fazi 12	0,103	0,495	0,248	0,044	0,000	0,796	0,910
Ekstremni izr. k. pospešek v fazi 13	0,100	0,537	0,268	0,046	0,000	0,990	1,105
Ekstremni izr. k. pospešek v fazi 12	0,100	0,537	0,268	0,046	0,000	0,990	0,846
Ekstremna izm. k. hitrost v fazi 6	0,099	0,573	0,286	0,033	2,800	2,959	2,731
Integral izr. k. hitrosti v fazi 9	0,095	0,375	0,187	0,024	0,000	1,742	0,224
Ekstremna izr. k. hitrost v fazi 3	0,092	0,648	0,324	0,036	0,000	0,610	0,653
Integral izm. k. hitrosti v fazi 9	0,091	0,417	0,209	0,025	0,000	1,286	0,327
Ekstremna izm. k. hitrost v fazi 7	0,091	0,533	0,267	0,032	0,500	4,685	2,883
Ekstremni std. odklon g v fazi 11	0,090	0,665	0,332	0,054	0,300	2,588	1,567
Integral izr. k. pospeska v fazi 4	0,090	0,630	0,315	0,037	0,000	0,751	0,383
Integral izr. k. hitrosti v fazi 4	0,090	0,348	0,174	0,015	0,000	1,036	0,156
Največja vrednost	0,160	0,973	0,934	0,064	4,200	7,798	4,387
Najmanjša vrednost	-0,004	0,058	0,029	0,003	0,000	0,035	-0,016
Povprečna vrednost	0,043	0,371	0,209	0,025	0,089	1,501	0,409

Tabela 12.1: Ocene 16 najbolje ocenjenih značilk, razvrščene po meri ReliefF.

boljša izbira za vizualno razločevanje razredov, izdelal sliko, kjer je mogoče opaziti vzorec, na podlagi katerega bi bilo mogoče ločevanje med osebami (slika 12.1).

Za samo identifikacijo sem v programu Orange zgradil več klasifikatorjev in med njimi izbral tistega, ki je dal najboljše rezultate. Parametri klasifikatorjev so privzeti, kot jih ponudi Orange. Tudi osnutek patentne prijave [9] predvideva uporabo strojnega učenja.

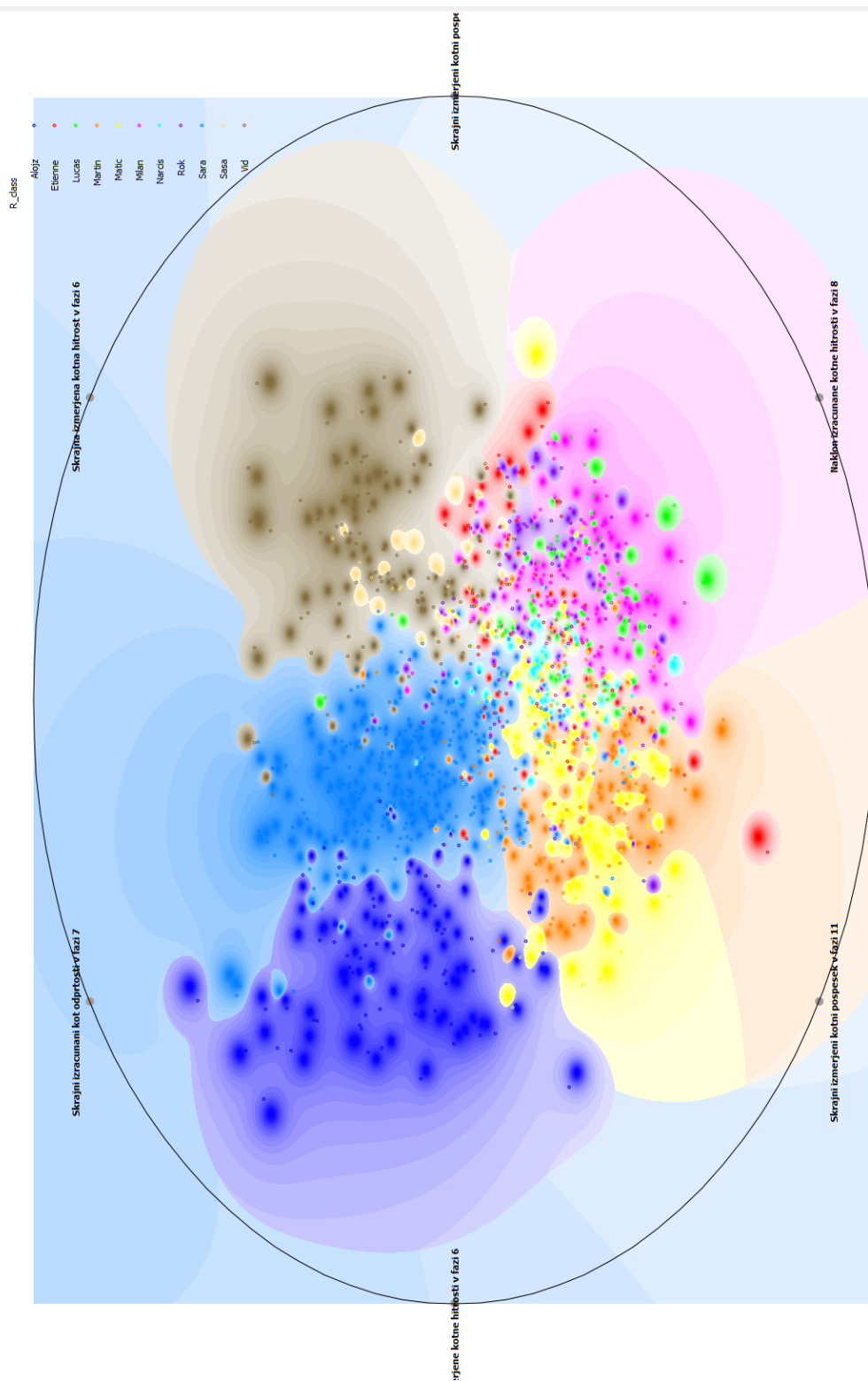
Prvi klasifikator je večinski razred (angl. majority class), ki vsem primerom določi razred, ki je med učnimi podatki predstavljen največkrat. Predpostavlja, da je v testnih podatkih podobno razmerje med številom primerov posameznega razreda kot v učnih podatkih in s tem je klasifikacija pravilna v enako ali več primerih, kot če bi vsakemu podatku določili naključni razred. Uporabnost tega klasifikatorja je za primerjavo z ostalimi klasifikatorji. Če neki klasifikator ne doseže niti tolikšnega deleža pravilnih klasifikacij, klasifikator ni uporaben.

Ostali klasifikatorji, ki sem jih uporabil, so: naivni Bayesov klasifikator [4] (angl. naive Bayes classifier), klasifikacijsko drevo [18] (angl. classification tree), k-najbližjih sosedov [6] (angl. k-nearest neighbours — kNN), metoda podpornih vektorjev [19] (angl. support vector machines — SVM), CN2 pravila [5], naključni gozd [1] (angl. random forest) in umetna nevronska mreža [21] (angl. artificial neural network). Uporabljena metoda za izračun mere uspešnosti učenja je 10-kratno prečno preverjanje (angl. 10-fold cross-validation) [2].

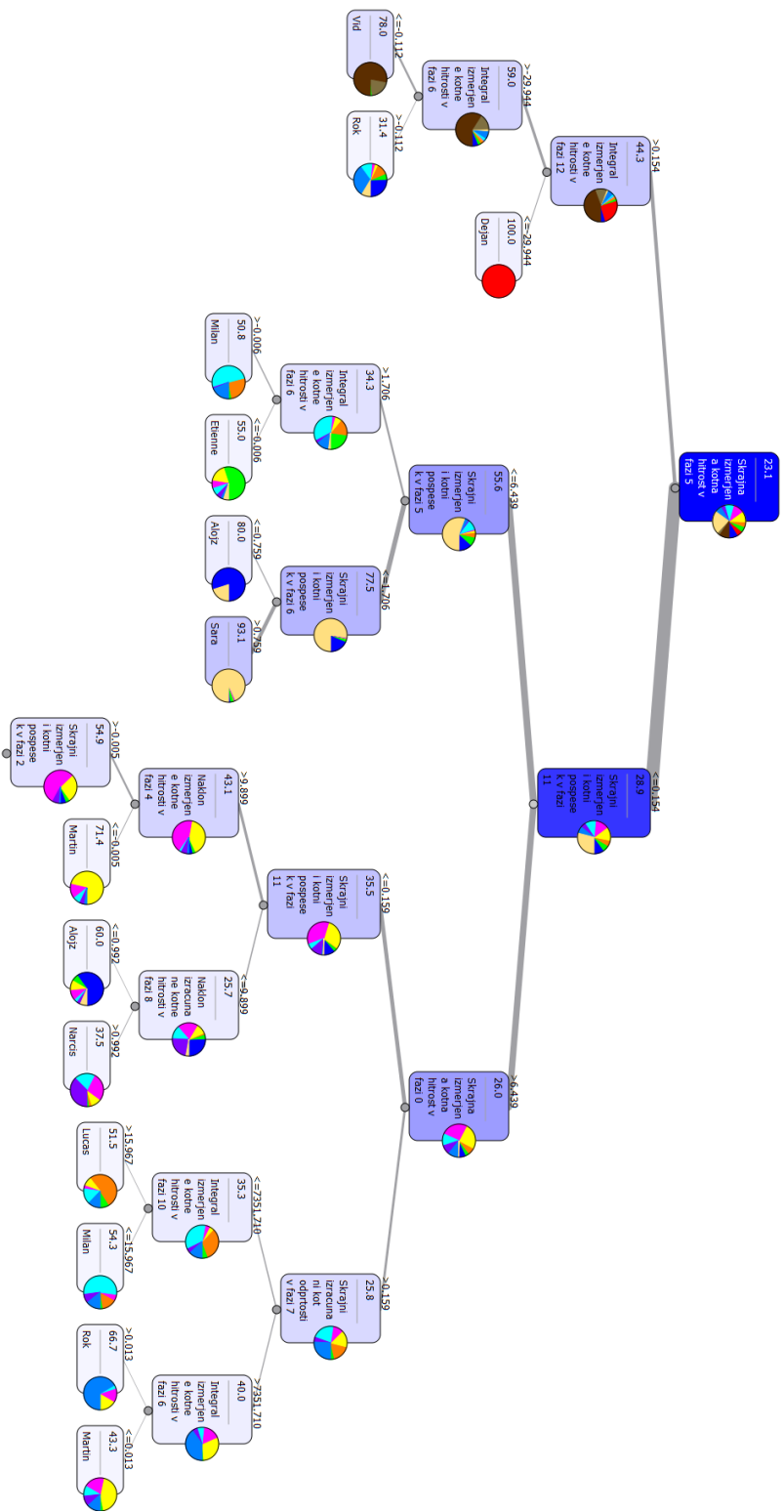
Primer zgrajenega klasifikacijskega drevesa na podatkih vstopov je viden na sliki 12.2.

12.2 Ugotovitve

V tabeli 12.2 so prikazani klasifikatorji in njihove ocene po različnih merah. Tabela 12.3 vsebuje ocene klasifikacijske točnosti za vsakega izmed klasifikatorjev pri klasifikaciji en proti vsem. Pri klasifikaciji en proti vsem poskušamo



Slika 12.1: Vizualizacija primerov prehodov na osnovi 6 značilk z metodo RadViz.



Slika 12.2: Vizualizacija zgrajenega klasifikacijskega drevesa na podatkih prehodov oseb skozi vrata. Drevo je zaradi preglednosti manjše kot drevo, uporabljeno v tabeli 12.2, in ima klasifikacijsko točnost 58,08 %.

Metoda	CA	AUC	IS	Brier
umetna nevronska mreža	90,64 %	99,67 %	2,9427	0,1433
kNN	84,18 %	98,64 %	2,8270	0,2256
naivni Bayesov klasifikator	83,85 %	98,84 %	2,7768	0,2981
naključni gozd	83,39 %	98,80 %	2,6444	0,2523
klasifikacijsko drevo	76,71 %	95,24 %	2,4969	0,3862
metoda podpornih vektorjev	75,93 %	98,31 %	2,3471	0,3518
CN2 pravila	73,02 %	96,48 %	2,3661	0,3597
večinski razred	23,08 %	50,00 %	0,0031	0,8851

Tabela 12.2: Ocene klasifikatorjev. CA je klasifikacijska točnost (angl. classification accuracy), AUC je površina pod krivuljo ROC (angl. area under the ROC curve), IS je informacijska ocena (angl. information score) in Brier je Brierjeva ocena [3] (angl. Brier score).

ločiti posamezno osebo od vseh ostalih. V praksi to pomeni, da ločujemo le na dva razreda in klasifikacijska točnost je v tem primeru višja. Večinski razred predstavlja primere vseh ostalih oseb, ki jih je v tem primeru veliko več, zato je tudi klasifikacijska točnost večinskega razreda bistveno višja. V primeru klasifikacije med dvema razredoma je klasifikacijska točnost večinskega razreda po definiciji vsaj 50 %.

Iz rezultatov v tabeli 12.2 in 12.3 je razvidno, da je bila umetna nevronska mreža najuspešnejša pri klasifikaciji prehodov skozi vrata. Njena klasifikacijsko točnost znaša 90,64 % in je višja od pričakovane. Dosežena je bila zaradi primerne nabora značilk, velikega števila primerov prehodov in kvalitativnih posnetkov prehodov. Razločevanje med do dvanajstimi osebami je v večini primerov dovolj, v primeru potrebe razločevanja med večjim številom oseb, pa bi bila klasifikacijska točnost nižja. Ob predpostavki, da je na voljo zadostno velika količina posnetkov in dovolj računske moči ter delovnega spomina, verjamem, da bi bila metoda uporabna tudi pri razločevanju nekaj velikostnih razredov več oseb.

Klasifikacijska točnost je odvisna tudi od števila oseb, ki jih želimo ločiti

Oseba	N. mreža	kNN	Bayesov k.	N. gozd	K. drevo	SVM	CN2 p.	Večinski r.
Dejan	100,00	100,00	100,00	100,00	99,89	100,00	99,78	94,65
Sara	98,89	98,77	92,98	97,88	96,55	92,75	95,66	76,93
Alojz	99,22	99,22	90,97	98,33	97,99	98,55	98,55	90,97
Etienne	99,11	98,11	97,66	96,43	96,21	97,66	96,66	94,76
Vid	98,77	97,77	96,10	97,77	97,99	97,10	96,43	90,53
Narcis	98,44	98,89	96,77	97,33	97,21	97,77	97,66	96,10
Martin	96,66	95,32	82,72	93,31	92,53	93,75	92,08	89,41
Rok	97,65	97,21	90,74	95,98	95,76	96,77	97,65	93,42
Matic	96,09	93,87	84,83	92,98	89,63	94,20	92,42	89,52
Milan	96,20	94,76	85,83	93,87	92,53	94,98	94,42	91,08
Lucas	99,00	98,33	97,33	97,10	96,77	98,89	97,55	95,10
Saša	98,78	98,33	85,76	97,77	98,44	98,22	98,11	97,55

Tabela 12.3: Klasifikacijska točnost pri klasifikaciji en proti vsem (angl. one vs. all) v %.

glede na način prehoda skozi vrata, konsistentnosti prehodov iste osebe in različnosti prehodov skozi vrata različnih oseb. V primeru manjšega števila različnih oseb je klasifikacija lažja in posledično točnejša.

Matrika zamenjav je prikazana v tabeli 12.4. Iz tabele je razvidno, s kolikšno uspešnostjo so bili primeri določene osebe klasificirani pravilno in kateremu razredu pripadajo napačne klasifikacije. Opaziti je, da je večina primerov klasificirana pravilno (vrednosti v diagonali), zamenjave pa so praviloma v parih (npr. 12,5 % Milanovih prehodov je klasifikator zamenjal z Rokovimi in 11,9 % Rokovih prehodov je klasifikator zamenjal za Milanove). Izjema je pri paru Saša in Vid, ko je 40,9 % Sašinih prehodov klasifikator zamenjal za Vidove, med tem ko je le 3,5 % Vidovih prehodov zamenjal za Sašine.

	Dejan	Sara	Alojz	Etienne	Vid	Narcis	Martin	Rok	Matic	Milan	Lucas	Saša
Dejan	100,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Sara	0,0	99,5	0,0	0,0	0,0	0,0	0,0	0,0	0,5	0,0	0,0	0,0
Alojz	0,0	2,5	97,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Etienne	0,0	2,1	0,0	93,6	0,0	0,0	2,1	0,0	0,0	0,0	2,1	0,0
Vid	0,0	1,2	0,0	2,4	92,9	0,0	0,0	0,0	0,0	0,0	0,0	3,5
Narcis	0,0	0,0	0,0	0,0	0,0	91,4	5,7	0,0	2,9	0,0	0,0	0,0
Martin	0,0	0,0	0,0	2,1	0,0	0,0	87,4	0,0	8,4	2,1	0,0	0,0
Rok	0,0	0,0	0,0	0,0	0,0	1,7	1,7	83,1	1,7	11,9	0,0	0,0
Matic	0,0	1,1	0,0	1,1	0,0	2,1	11,7	0,0	83,0	1,1	0,0	0,0
Milan	0,0	2,5	0,0	0,0	0,0	0,0	1,2	12,5	0,0	82,5	1,2	0,0
Lucas	0,0	0,0	0,0	2,3	0,0	0,0	0,0	2,3	0,0	13,6	81,8	0,0
Saša	0,0	0,0	0,0	0,0	40,9	0,0	0,0	0,0	0,0	0,0	0,0	59,1

Tabela 12.4: Uspešnost zgrajene umetne nevronske mreže pri identifikaciji posamezne osebe v %. V vrsticah so pravilni razredi, v stolpcih so klasifikacije.

Poglavje 13

Sklep

V tem delu je predstavljen sistem za avtomatično identifikacijo oseb ob prehodu skozi vrata na podlagi podatkov senzorjev pospeškov, ki so vgrajeni v oziroma na vratno krilo, in strojnega učenja. Identifikacija se izvaja samodejno in od uporabnika ne zahteva nobene dodatne akcije razen prehoda skozi vrata.

Sistem omogoča večje udobje uporabnikov vrat in nudi v primeru kombinacije s katero izmed obstoječih metod identifikacije dodatno varnost. Rezultati so dovolj natančni za vsakodnevno uporabo za namen dopolnjevanja obstoječih sistemov preverjanja identitete oziroma za uporabo v namen zabavne industrije, zanesljivost pa ni tako visoka, da bi sistem nadomestil uporabo druge metode za identifikacijo.

Praktična uporaba sistema lahko vključuje preverjanje, ali se način prehoda skozi vrata ujema z drugo identifikacijo, ki jo je dala oseba. Mogoča uporaba sistema je tudi na vratih znotraj objekta, kjer bi se samodejna identifikacija uporabljala za individualizirano nudenje storitev pametnega doma glede na prisotnost uporabnikov. V takem primeru bi lahko sistem glede na uporabnikove navade nastavljal osvetlitev in temperaturo prostora ter radio na postajo, ki jo uporabnik najpogosteje posluša.

Obstoječe metode za identifikacijo človeka ob prehodu skozi vrata vključujejo vsaj eno dodatno akcijo človeka, ki je lahko odčitek druge biometrične

lastnosti, vnos PIN-kode ali zahteva, da ima oseba med preходом ob sebi napravo, na podlagi katere se izvaja identifikacija. Najpogostejše metode so identifikacija na podlagi čitalca RFID, čitalca NFC, povezave Bluetooth z mobilnim telefonom, čitalca prstnih odtisov, prepoznavanja obraza, prepoznavanja glasu, vnosa PIN-kode. Nekatere metode so bolj varne, druge manj, a bolj udobne. Metoda opisana v tej diplomski nalogi sama po sebi ne zagotavlja varnosti, nudi pa veliko udobja, ker od uporabnika ne zahteva dodatnih akcij in pozornosti.

Sistem bi bilo mogoče izboljšati tako, da bi se vključilo sprotno učenje klasifikatorja (angl. incremental learning) na podlagi na novo posnetih prehodov v primeru, da je bila ob prehodu pravilno ugotovljena oziroma podana identifikacija. Odstranjevanje šuma iz izmerjenih pospeškov in izračun kota odprtosti vrat bi se lahko izvajalo z uporabo Kalmanovega filtra [14]. Mogoča bi bila tudi poenostavitev algoritma tako, da bi se računale le izbrane značilke, mogoča bi bila tudi optimizacija izračuna značilke in faz prehoda.

Literatura

- [1] Breiman, L. Random forests. *Machine learning*, 2001, str. 5–32.
- [2] Breiman, L. in Spector, P. Submodel selection and evaluation in regression. the x-random case. *Revue internationale de Statistique*, 1992, str. 291–319.
- [3] Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review* 78.1, 1950, str. 1–3.
- [4] Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W. in Freeman, D. AutoClass: A Bayesian Classification System, *International Conference on Machine Learning*, San Francisco: Morgan Kaufmann Publishers, 1988, str. 54–64.
- [5] Clark, P. in Niblett, T. The CN2 induction algorithm. *Machine learning* 3(4), 1989, str. 261–283.
- [6] Cover, T. in Hart, P. Nearest neighbor pattern classification. Information Theory, *IEEE Transactions* 13(1), 1967, str. 21–27.
- [7] Demšar, J., Zupan, B., Leban, G. in Curk, T. Orange: From Experimental Machine Learning to Interactive Data Mining. *Knowledge Discovery in Databases PKDD 2004*. Berlin: Springer, 2004.
- [8] Friedman, J. H. Multivariate adaptive regression splines. *The annals of statistics* 19(1), 1991, str. 1–67.

-
- [9] Gams, M., Piltaver, R. in Gjoreski, H. Postopek za identifikacijo oseb ob vstopu v prostor, osnutek patentne prijave, avgust 2013.
- [10] Björck, Å. Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT Numerical Mathematics* 7(1) 1967, str. 1–21.
- [11] Householder, Alston S. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)* 5(4), 1958, str. 339–342.
- [12] Hunt, E., Martin, J. in Stone, P. *Experiments in induction*. New York: Academic press, 1966.
- [13] Jakulin, A. *Strojno učenje na osnovi interakcij med atributi*. Doktorska disertacija. Ljubljana: Fakulteta za računalništvo in informatiko, 2005.
- [14] Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82(1), 1960, str. 35–45.
- [15] Kononenko, I. Estimating Attributes: Analysis and Extensions of RELIEF. *European Conference on Machine Learning*, 1994, str. 171–182.
- [16] Kononenko, I. *Strojno učenje*. Ljubljana: Fakulteta za računalništvo in informatiko, 2005.
- [17] Peer, P., Čargo, B. in Kononenko, I. Extension of ReliefF. *Electrotechnical Review* 64(5), 1997, str. 277–283.
- [18] Rokach, L. *Data mining with decision trees: theory and applications*. World Scientific, 2007.
- [19] Steinwart, I. in Christmann, A. *Support vector machines*. New York: Springer, 2008.
- [20] Visintin, A. *Differential models of hysteresis*. Berlin: Springer, 1994.
- [21] Yegnanarayana, B. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2004.