

Univerza v Ljubljani  
Fakulteta za elektrotehniko

Leon Lahajnar

## **Avtonomno smučanje humanoidnega robota**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Jadran Lenarčič

Somentor: dr. Bojan Nemec

Ljubljana, 2009

## Vsebina

|  |     |
|--|-----|
| Vsebina .....                                  | i   |
| Povzetek.....                                  | iii |
| Abstract.....                                  | vi  |
| Zahvala.....                                   | ix  |
| Tabela simbolov.....                           | x   |
| 1 Uvod.....                                    | 1   |
| 1.1 Pregled ožjega znanstvenega področja ..... | 1   |
| 1.2 Cilji doktorske disertacije .....          | 4   |
| 1.3 Metodologija.....                          | 5   |
| 2 Model robota smučarja .....                  | 8   |
| 2.1 Motivacija .....                           | 8   |
| 2.2 Modeliranje človeka smučarja .....         | 9   |
| 2.3 Zgradba robota.....                        | 10  |
| 2.4 Zavijanje z nagibanjem.....                | 11  |
| 2.5 Kinematični model robota .....             | 14  |
| 2.6 Dinamični model.....                       | 17  |
| 2.7 Model okolja .....                         | 19  |
| 2.8 Elastičnost mehanizma .....                | 20  |
| 3 Stabilnost smučarja na klančini .....        | 22  |
| 3.1 Dinamično ravnotežje .....                 | 22  |
| 3.2 Točka ničelnega navora (ZMP) .....         | 22  |

|   |    |
|---|----|
| 3.3 Poenostavljen model smučarja v lateralni ravnini .....                | 27 |
| 3.4 ZMP robota na klančini .....  | 28 |
| 3.5 Zagotavljanje stabilnosti .....                                       | 32 |
| 3.6 Odločitveno vodenje .....   | 34 |
| 3.7 Analiza statične stabilnosti v odvisnosti od hitrosti in naklona..... | 37 |
| 3.8 Stabilizacija s trupom.....   | 39 |
| 4 Navigacija med vratci .....   | 41 |
| 4.1 Uvod.....   | 41 |
| 4.2 Zaznava vratc .....   | 44 |
| 4.3 Vodenje smučarja med vratci .....                                     | 46 |
| 4.4 Model človeške navigacije.....  | 47 |
| 4.5 Ocena pozicije robota glede na vratca .....                           | 52 |
| 4.6 Izvajanje poti.....   | 58 |
| 5 Implementacija realnega sistema .....                                   | 60 |
| 5.1 Navidezni robot smučar .....  | 60 |
| 5.2 Zgradba robota.....   | 62 |
| 5.3 Senzorski sistem .....  | 64 |
| 5.4 Zgradba sistema vodenja .....   | 68 |
| 6 Rezultati in ovrednotenje dela .....                                    | 70 |
| 6.1 Rolkanje .....  | 70 |
| 6.2 Smučanje.....   | 77 |
| 7 Sklep .....   | 88 |
| 7.1 Izvirni prispevki .....   | 89 |
| 8 Literatura.....   | 90 |

## **Povzetek**

Poglavitni prispevek doktorske disertacije je izdelava vodenja, ki zagotavlja avtonomno vožnjo smučarskega robota po smučišču. V delu smo predlagali tudi zgradbo robota, ki izvaja smučanje na smučkah s poudarjenim stranskim lokom na človeku podoben način. V simulaciji in na robotu smo preizkusili izdelane algoritme vodenja, ki zagotavljajo stabilno smučanje. Ob tem pa smo predlagali in implementirali še metodo za izvajanje poti po postavljeni poti.

Človek izvaja zavoje na smučkah s poudarjenim stranskim lokom na dva načina in sicer z abdukcijo in addukcijo v kolku, ter fleksijo in ekstenzijo nog. V našem primeru smo se odločili, da uporabimo pri izdelavi robota model izvedbe zavojev s fleksijo in ekstenzijo nog. Zasnovali in zgradili smo robota, ki je sestavljen iz dveh nog z eno prostostno stopnjo in trupa z možnostjo nagiba v lateralni smeri. Ta zgradba dokaj verno oponaša gibanje človeka med smučanjem. Predlagana zgradba robota nam je služila za izgradnjo dinamičnega in kinematičnega model robota. Predstavili smo tudi odnos med naklonom rola in smuča na podlago in zavijanjem. Zgrajen pa je bil še model vpliva okolja na smučarja. Na principu Newton-Eulerjeve metode smo zgradili še poenostavljen model smučarja na klančini, ki omogoča računanje dinamičnih veličin robota v realnem času.

Do sedaj predlagani algoritmi za zagotavljanje stabilnosti hodečih in mobilnih robotov delujejo v večini primerov le v strukturiranem okolju. V teh primerih se trajektorije gibanja robota izračuna vnaprej. Med samo izvedbo naloge se izvajajo zgolj manjši popravki teh gibanj. Pri smučanju pa se mora robot aktivno odzivati na zaznane ovire in konfiguracijo terena. Zato smo predlagali algoritem za zagotavljanje stabilnosti v realnem času. Kot kriterij stabilnosti robota smo uporabili točko ničelnega navora, ki se izračunava na podlagi senzorskih informacij in poenostavljenega dinamičnega modela robota. Za razliko od predhodno predlaganih rešitev, kjer se ob vnaprej točno določeni poziciji točke ničelnega

navora zagotovi tako gibanje robota, ki ohranja točko v tej poziciji, pa je pri naši metodi gibanje točke ničelnega navora omejena zgolj s področjem znotraj podpornega poligona. V tem območju se točka lahko prosto giblje, zato lahko robot izvaja še sekundarno nalogo. Algoritem temelji na regulatorju pospeška. V vsakem časovnem trenutku se preveri, ali želeni kontrolni pospeški zagotovijo ohranitev točke ničelnega navora znotraj podporne ploskve. Za robota je bila opravljena analiza, kakšni so dovoljeni nakloni na podlago, ki še zagotavljajo stabilnost robota brez lastnega gibanja robota v zavojih. Na podlagi teh dveh kriterijev je bilo zgrajeno iterativno vodenje, ki zagotavlja stabilno gibanje robota tako v zavojju, kot tudi v fazi med zavoji.

Robot pa razen zagotavljanja stabilnosti lahko izvaja še sekundarno nalogo. V našem primeru smo kot sekundarno nalogo definirali vožnjo skozi sistem vratc. Za gradnjo poti med vratci smo uporabili model človeške navigacije. Za razliko od ostalih potencialnih metod, se pri tej metodi gradi potencialno polje na novo v vsakem trenutku, glede na trenutno pozicijo ovir in ciljev. Pri tem smo predlagali nov način gradnje trenutnih ciljev. S tem smo se izognili možnosti, da bi robot zašel v lokalni minimum. Druga prednost opisane metode je, da so poti in odvodi gladki, kar je nujno potrebno za uporabo pri neholonomskih mobilnih robotih.

Iz literature je razvidno, da večina metod gradnje poti temelji na sestavljanju poti iz ravnih segmentov in krožnic okrog ovire z radijem enakim minimalnemu, ki ga je robot sposoben izvesti. V našem delu smo podali izboljšavo, ki s pomočjo virtualnega premika zagotovi pot, ki poteka bližje ovire, kot pa je minimalni radij. S tem smo zagotovili, da se ovira nahaja več časa v vidnem kotu kamere. To omogoča natančnejšo oceno dejanske pozicije vratc glede na robota.

Zaradi nezanesljivosti senzorskih, predvsem vizualnih informacij, smo implementirali tudi oceno pozicije robota glede na vratca na podlagi razširjenega Kalmanovega filtra. Predlagani model omogoča filtriranje podatkov v primeru, ko se v podatkih o poziciji vratc pojavijo napake. Ob tem pa omogoča predikcijo pozicije vratc v primerih, ko iz računalniškega vida ni podatka.

Ob koncu dela smo predstavili del eksperimentov, ki smo jih izvedli tako v simulaciji, kot tudi na robotu. Na podlagi simulacije je bilo ugotovljeno, da s pomočjo virtualnega premika vratc dosežemo bolj neposredno vožnjo skozi vratca. Preizkusi na robotu so pokazali, da je robot s predlaganimi algoritmi ohranil stabilnost med vožnjo. Medtem ko se je pri rolkanju merjeni radij zavoja ujema z modeliranimi, pa se je pri smučanju izkazalo, da je izveden radij zavoja zelo odvisen od strukture podlage. Eksperimenti so pokazali, da se je robot, z uporabo v delu opisanih algoritmov, sposoben avtonomno spustiti po smučišču in ob tem prevoziti postavljeno pot.

## **Abstract**

The main contribution of this thesis is development of control algorithms for autonomous skiing of robot on the skiing ground. In our work we also propose robot structure that allows skiing on carving skies in similar way as human does. Also some models of robot and virtual environment are produced. Developed control algorithms are tested. Also method of path execution on the track is proposed and implemented.

Human can execute turns using carving skis in two ways. First one is with adduction and abduction in the hip. The second way is with flexion and extension of the legs. In our case we concentrated on second way, because it is more commonly used. The skiing robot is composed of two legs with one degree of freedom and torso with inclination in the lateral plane. This structure represents human movement during the skiing. Based on this structure we built dynamic and kinematic models. In dynamic model also the environment influences were incorporated. Coupling between inclination to the ground and the turn radii for skateboards and carving skis are presented. Also simplified model of a robot on the skiing ground was constructed. This allowed us to calculate dynamic properties of the robot in real-time.

Previously proposed algorithms for dynamic stability of legged and mobile robots work mainly in structured environment. Those methods calculate stable motion offline in advance. During task execution only minor corrections in motion trajectories are induced. But during the skiing robot has to actively react to sensed obstacles and structure of the skiing ground. Therefore we proposed algorithm for real-time stability control. The robot stability was evaluated by zero moment point criteria, that is calculated from sensory information and simplified dynamic model of the robot. Methods proposed in literature calculate movement of the robot that

will maintain zero moment point in defined position on the ground. In our method is position of zero moment point limited only to the area of support polygon. Zero moment point can be present in any point of this area. Therefore the robot can execute also the secondary task. Algorithm is based on acceleration control. In every time sample the control accelerations are tested if those accelerations will maintain zero moment point in area of support polygon. We had to analyze which robot inclination to the surface still maintains robot stability in turns without robots self motion. According to those two criteria iterative control was built, that assures stable motion of the robot in turns and also in region between turns.

Except stability the robot can execute also secondary task. In our case this task is defined as skiing through the system of gates. A trajectory generation between the gates is based on the model of human navigation. In contrary to the other potential methods, this method generate potential field in every moment, according to temporal positions of obstacles and goals. We proposed a new way of defining goals that eliminates the possibility that the robot would be trapped in local minima. Second advantage of this method is that produced path and its derivate are smooth. Therefore produced path can be used with non-holonomic mobile robots.

Most of presented methods for optimal path generation in literature are based on building path from straight segments and circular arc with minimal radii that the robot can execute. In our work we have also some knowledge of further gates. Therefore we proposed virtual shift of the gates that allows path that is closer to the obstacle than the minimal radii. This assures that the gates remains longer in camera angle of view and consequently allows the robot to estimate more precise position of the gates.

Because of inexact sensory information, especially visual, we implemented estimation of robot position toward the gates based on extended Kalman filter. Proposed model allows to optimally estimate position of the robot in noisy environment. In cases when visual information is not available also predicted position of the gates toward the robot is estimated.



In the end of the work a part of conducted experiments is presented. Simulation experiments showed that with the usage of virtual shift closer path to the gates is achieved. Experiments on the robot showed that proposed algorithms assured stability during the ride. An executed radius during the skating is quite similar to the modeled. But at skiing radii is very dependent on structure of the snow. Built robot is capable to autonomously ski on the skiing ground and drive through the system of the gates.

## **Zahvala**

*Zahvaljujem se mentorjema prof. dr. Jadranu Lenarčiču in dr. Bojanu Nemcu za mentorstvo in koristne nasvete pri izdelavi doktorske disertacije.*

*Zahvaljujem se tudi vsem kolegom iz Odseka za avtomatiko, biokibernetiko in robotiko na Institutu Jožef Stefan, ki so mi pomagali pri delu. Prav posebna zahvala gre dr. Damirju Omrčenu in dr. Leonu Žlajpahu, ki sta mi pomagala s koristnimi nasveti.*

*Za finančno podporo podiplomskega študija se zahvaljujem Javni agenciji za raziskovalno dejavnost Republike Slovenije.*

*Nenazadnje bi se rad zahvalil vsem bližnjim za moralno podporo in spodbudo v vseh letih študija.*

## Tabela simbolov

|           |  |                   |
|-----------|--|-------------------|
| $q_i$     | koti sklepov robota  | rad( $^{\circ}$ ) |
| $l_c$     | razdalja med med ploščadjo in sredino med smučkama             | m                 |
| $l$       | dolžina segmenta noge  | m                 |
| $d$       | razdalja med nogama  | m                 |
| $R_r$     | radij zavoja rolke   | m                 |
| $\theta$  | naklon robota na podlago                                       | rad( $^{\circ}$ ) |
| $\alpha$  | kot naklona mehanizma rolke                                    | rad( $^{\circ}$ ) |
| $L_r$     | razdalja med mehanizmoma rolke                                 | m                 |
| $L$       | dolžina smučke   | m                 |
| $r$       | radij zavoja smučke  | m                 |
| $h$       | višina krožnega odseka   | m                 |
| $W_s$     | širina smučke spredaj  | m                 |
| $W_w$     | širina smučke na sredini                                       | m                 |
| $W_t$     | širina smučke na repu  | m                 |
| $d$       | globina ugreza smučke v sneg                                   | m                 |
| $c_i$     | $\cos(q_i)$  |                   |
| $s_i$     | $\sin(q_i)$  |                   |
| $J_s$     | Jacobijeva matrika robota za radij smučanja in višino smučanja |                   |
| $J_r$     | Jacobijeva matrika robota za radij rolkanja in višino rolkanja |                   |
| $a_r$     | radialni pospešek  | ms $^{-2}$        |
| $g$       | gravitacija  | ms $^{-2}$        |
| $F_v$     | sil vizkoznega trenja  | N                 |
| $F_s$     | sil statičnega trenja  | N                 |
| $m_R$     | masa robota  | kg                |
| $X$       | globalna pozicija robota v smeri osi X                         | m                 |
| $Y$       | globalna pozicija robota v smeri osi Y                         | m                 |
| $\varphi$ | globalna smer gibanja  | rad( $^{\circ}$ ) |
| $l_r$     | dolžina desne noge   | m                 |
| $l_l$     | dolžina leve noge  | m                 |
| $M_x$     | navor okrog osi x  | Nm                |

|             |  |                 |
|-------------|--|-----------------|
| $M_y$       | navor okrog osi y  | Nm              |
| $R$         | rezultanta vseh sil, s katero deluje podlaga na nogo               | N               |
| $P$         | točka na podlagi   |                 |
| $q_{sp}$    | vektor kinematičnih veličin poenostavljenega modela robota         |                 |
| $\tau$      | vektor sil in navorov poenostavljenega modela robota               |                 |
| $r_i$       | vektor težišča i-tega segmenta                                     |                 |
| $m_i$       | masa i-tega segmenta   | kg              |
| $I_i$       | vztrajnostna matrika i-tega segmenta                               |                 |
| $\omega_i$  | vektor kotnih hitrosti i-tega segmenta                             |                 |
| $M_j$       | vektor zunanjega navora  |                 |
| $F_k$       | vektor zunanje sile  |                 |
| $s_k$       | vektor prijemališča zunanje sile                                   |                 |
| $M_P$       | vektor navorov v točki P   |                 |
| $x_{ZMP}$   | točka ničelnega navora v smeri osi x                               | m               |
| $y_{ZMP}$   | točka ničelnega navora v smeri osi y                               | m               |
| $M_{i-1,i}$ | vektor navora s katerim deluje $i-1$ segment na segment $i$        |                 |
| $F_{i-1,i}$ | vektor sile s katero deluje $i-1$ segment na segment $i$           |                 |
| $\psi$      | naklon podlage   | rad( $^\circ$ ) |
| $\psi_s$    | zaznan naklon podlage  | rad( $^\circ$ ) |
| $F$         | celotna sila s katero robot deluje na podlago pravokotno na smučko | N               |
| $F_d$       | sila s katero deluje desna noga na podlago                         | N               |
| $F_l$       | sila s katero deluje desna noga na podlago                         | N               |
| $b_{sr}$    | polovica razdalje med smučkama                                     | m               |
| $\theta_d$  | želen naklon na podlago  | rad( $^\circ$ ) |
| $\theta_m$  | merjen naklon na podlago   | rad( $^\circ$ ) |
| $\varphi_o$ | kot pogleda na oviro   | rad( $^\circ$ ) |
| $d_v$       | razdalja v slikovnih elementih od navpične osi                     | piksel          |
| $K_{zk}$    | koeficient preslikave zornega kota v piksel                        | piksel          |
| $d_o$       | razdalja do ovire  | m               |
| $d_c$       | razdalja do cilja  | m               |
| $b$         | konstanta dušenja  | $s^{-1}$        |
| $k_g$       | koeficient privlačenja cilja                                       | $s^{-2}$        |
| $k_o$       | koeficient odboja ovire  | $s^{-2}$        |

|       |   |
|-------|---|
| $c_1$ | koeficient vpliva oddaljenosti cilja    |
| $c_2$ | vpliv cilja pri veliki oddaljenosti     |
| $c_3$ | koeficient vpliva kota pogleda na oviro |
| $c_4$ | koeficient vpliva oddaljenosti ovire    |

# 1 Uvod

## 1.1 Pregled ožjega znanstvenega področja

Težišče raziskav se v svetu osredotoča na humanoidno robotiko. Spekter problemov na humanoidni robotiki je zelo širok, od razvoja komponent do človeku podobnega gibanja in kognitivnega obnašanja. Veliko raziskav poteka tudi na področju t. i. dinamičnega gibanja humanoidnih robotov. Mnogo raziskovalcev se ukvarja z gradnjo in vodenjem bipedalnih robotov s sposobnostjo hoje. Večina teh raziskav je bila izvršena na ravni podlagi. Šele v zadnjem času raziskujejo hojo na nagnjenih površinah, stopnicah in hojo preko ovir, ki pa kljub vsemu predstavljajo strukturirano okolje. Glavni problem, ki ga rešujejo, je stabilnost v smeri hoje, medtem ko stransko stabilnost v mnogih primerih zanemarijo, saj je uporabljen planarni model hoje. Pogosto uporabljajo posebej oblikovana stopala, ki povzročijo, da se robot nagiba v odvisnosti od faze v koraku (Toddler [4]). Alternativni pristop je, da se s pomočjo premikov v gležnju in kolku zagotovi prenos težišča na želeno nogo, kar zagotavlja statično stabilnost. Ta pristop uporabljajo znani roboti, kot so Asimo, QRIO ...

Najpogosteje uporabljen kriterij za zagotavljanje dinamične stabilnosti bipedalnih robotov je točka ničelnega navora (ang. Zero Moment Point - ZMP) [27], ki določa točko na površini, kjer vsota vseh dinamičnih sil, delujočih na robota, ne povzroča navorov. Robot je stabilen, če se ZPM nahaja v območju podpornih površin.

Čeprav lahko stransko stabilnost pri hoji zanemarimo, pa postane pomembna pri gibanjih, kot so kolesarjenje, smučanje, rolkanje, drsanje ... Pri teh nalogah se človeku kot sistemu zmanjša okretnost, vendar pa s tem pridobi na stabilnosti in hitrosti v smeri gibanja. V literaturi so bili predstavljeni modeli avtonomnih kolesarjev. Murato-boy predstavlja kolesarja, ki svoj stranski nagib zaznava s pomočjo giroskopov, stabilnost v prečni smeri vzdržuje s pomočjo vztrajnika. Ta

način vodenja so uporabili tako za enokolesna [3] kot tudi za dvokolesna vozila [22]. Tanaka in Murakami [17] sta za kolo uporabila model inverznega nihala, stabilnost pa zagotovila s PD regulatorjem krmilnega kota. Podoben način zagotavljanja stabilnosti je uporabil tudi Jingang Yi [6] pri vodenju avtonomnega motorista, katerega pomembna nadgradnja je, da poleg stabilnosti zagotavlja tudi sledenje trajektoriji.

Gibanje, ki se izvaja na neravnih površinah, zahteva zaznavanje naklonov in oceno stabilnosti, ker bi sicer lahko prišlo do prevrnitve robota. Park [7] je v svojih raziskavah gibanja po neravnem terenu uporabil neholonomnega robota. Pri tem se je osredotočil na stabilnost v povezavi z akcijami v danem okolju. Okolje je zaznaval s pomočjo 3D skenerja. Pri robotskih vozilih se večina raziskav osredotoča na stabilnost v odvisnosti od hitrosti v zavojih, vendar pa pri tem predpostavljajo, da se struktura mehanizma in s tem točka težišča med zavojem ne spreminja [8],[9]. Tipičen primer spreminjanja točke težišča pa je na primer dvigovanje tovora med gibanjem viličarja [19].

Izdelava robota, ki je sposoben avtonomnega smučanja, zahteva tudi poznavanje osnov tehnike smučanja. S pojavom smuči s poudarjenim stranskim lokom se je smučanje korenito spremenilo. Če se pri klasičnih smučeh zavoj izvaja predvsem z oddrsavanjem, se pri smučeh s poudarjenim stranskim lokom zavoj izvede z zarezo roba smučke v sneg, kar povzroči gibanje po trajektoriji bolj ali manj podobni krožnici. Pri tem določata radij zavoja geometrija smučke in globina zarez [21]. Zarezne zavoje človek izvede na dva načina, in sicer z abdukcijo in addukcijo v kolku ter iztegom ter krčenjem kolen. Pri izvajanju zavojev pri nizkih hitrostih se uporabljata obe obliki izvedbe. Pri višjih hitrostih pa zaradi velikih radialnih sil smučar ne izvaja zavojev z abdukcijo v kolku, saj mišice v stranski ravnini niso kos velikim silam. Zato smučar izvede zavoj s krčenjem in iztegom nog (gibanje kolena), stabilnost pa zagotavlja z nagibom hrbtenice [1].

Do sedaj je bilo zgrajenih več mehanizmov, ki omogočajo smučanje. Glavna pomanjkljivost vseh teh mehanizmov je, da se ne odzivajo na okolje, saj izvajajo zgolj vnaprej predpisano gibanje. Yoneyama [33] je za oponašanje zarezne tehnike smučanja zgradil robota, ki omogoča gibanje v kolčnem sklepu. Shimizu [16] je zgradil robota za zarezno tehniko, ki uporablja gibanje kolka ali gibanje kolena. S tem je potrdil, da se pri zareznem smučanju lahko uporabljata obe obliki izvajanja

zavoja.

Meritve sil na podlago posameznih nog pri smučarju, ki izvaja zarezne zavoje, so pokazale, da človek pri smučanju vselej teži k stanju, kjer ima stik s podlago z obema nogama [32]. Razmerje med silami se spreminja v odvisnosti od faze zavoja in hitrosti. V začetku zavoja se pojavijo večje sile na zunanji smučki, v nadaljevanju pa prehaja sila tudi na notranjo smučko. Pri zaključku zavoja, ki je obenem tudi začetek naslednjega zavoja, se sile spet prenesejo na zunanjo smučko.

Smučar je v svoji osnovi neholonomni sistem, saj se, razen v primerih, ko se odlepi od tal, ne more gibati v poljubni smeri.

Pri vodenju neholonomnih sistemov se večina raziskav osredotoča na lokalizacijo v prostoru, iskanje idealne poti in izogibanje oviram. Lokalizacija in iskanje optimalne poti pogosto zahtevata planiranje na globalnem nivoju, ki upošteva celotni prostor, po katerem se robot giblje. Zaradi računske zahtevnosti globalni algoritmi ponavadi ne delujejo v realnem času. Druga slabost pri uporabi globalnih metod je, da pogosto nimamo popolnih informacij o okolju. V nasprotju z globalnimi pa lokalni algoritmi upoštevajo le informacije o bližnji okolici robota, kar sicer računsko poenostavi problem, vendar pa ne zagotavlja globalno optimalne poti. Izogibanje oviram se v mobilni robotiki pogosto izvede zgolj na podlagi lokalnih informacij. Ena najbolj uporabljenih tehnik so umetna potencialna polja, ki odbijajo robota stran od ovir, hkrati pa ga privlačijo k cilju [16]. Pri neholonomnih robotih se pojavi problem, ker se robot ne more gibati v poljubno smer in je zato potrebno originalne metode ustrezno modificirati [11], [13].

Ena izmed pogosto uporabljenih metod za izogibanja oviram je tudi polje vektorskih histogramov, ki računa enodimenzionalni polarni histogram, na osnovi katerega se izračuna prosta območja, po katerih je možna pot [14], [15]. Tako metoda potencialnih polj kot polje vektorskih histogramov se najpogosteje uporabljajo v povezavi z različnimi sonarji in panoramskimi kamerami.

Fajen [10] je za izogibanje oviram uvedel termin krmilni kot, katerega nadgradnja omogoča tudi uporabo kamer [36]. Metoda je podobna potencialnim poljem, vendar polje gradi le v okolici trenutne smeri gibanja robota. Za mero oddaljenosti pa uporablja podatke o prostorskem kotu ovire. V primeru znanih ovir pa se na podlagi velikosti in znanega kota pogleda na oviro ugotovi oddaljenost le-te.



## 1.2 Cilji doktorske disertacije

Na podlagi rezultatov predhodnih raziskav in stanja tehnike menimo, da je mogoče zgraditi robota, ki bo avtonomno smučal po klančini in pri tem oponašal gibanje in obnašanje človeka. Pri tem se bomo omejili na t. i. zarezno tehniko smučanja. Glavna izziva sta zagotavljanje stabilnosti mehanizma med smučanjem po neravnem terenu ter sprotno planiranje poti na osnovi senzorskih informacij. Namen disertacije je razviti algoritme vodenja, ki na podlagi podatkov, pridobljenih iz senzorjev, podobnih človeškim, zagotovi mehanizmu stabilno izvajanje zavojev na klančini. Algoritme bomo testirali tako s simulacijo v virtualnem okolju kot na realnem modelu smučarja, ki smo ga razvili in zgradili v okviru naloge. Mehanizem je sestavljen iz dveh nog s po eno prostostno stopnjo, ki oponaša gibanje kolena, ter trupom s sposobnostjo prečnega nagiba, ki zagotavlja lateralno stabilnost. Na mehanizmu so vgrajeni giroskopi, senzorji sile na podlago, GPS in vizualni senzor.

Vodenje mora zagotavljati stabilnost robota in obenem avtonomno gibanje robota v realnem času, pri čemer se mora izogibati statičnim in dinamičnim oviram. V splošnem ne moremo vedno zagotoviti stabilnosti in gibanja po željeni trajektoriji. V tem primeru je naša primarna naloga zagotavljanje stabilnosti. Pri tem ovire zaznavamo v realnem času s pomočjo monokularnega vida.

Algoritmi vodenja in planiranja poti morajo zagotavljati avtonomno gibanje robota po smučišču. Pri tem lahko pot določimo tudi z vratici, kot je to v navadi pri tekmah alpskega smučanja.

Podrobneje obravnavamo naslednjo problematiko:

- Razvoj dinamičnega modela smučarja v zavoju.  
Model je primeren za analizo in ovrednotenje prečne stabilnosti smučarja ob upoštevanju naklona zaznane površine, po kateri se smučar giblje. Model omogoča simulacijo gibanja smučarja, ob tem pa posreduje tudi sile, s katerimi smučar deluje na podlago. Za algoritme vodenja pa je bil zgrajen tudi poenostavljen model.
- Analiza stabilnosti mehanizma pri različnih hitrostih gibanja v zavojih.

V realnem času je potrebno pridobiti mero stabilnosti robota pri različnih naklonih podlage in hitrostih gibanja. Na podlagi teh informacij smo zgradili vodenje, ki zagotavlja stabilne zavoje tako pri nizkih kot tudi pri visokih hitrostih.

- Sinteza vodenja, ki bo zagotavljalo stabilnost robota.  
Robot ima med vožnjo določeno področje, v katerem lahko izvaja svojo nalogo. Medtem ko mora zagotoviti stabilnost, se mora na podlagi višjega sistema vodenja tudi odzivati na okolje in izvajati želeno pot.
- Razvoj navideznega okolja, ki omogoča vizualizacijo mehanizma in gibanja smučarja.  
Vizualizacija se izvaja v navideznem okolju Blender. V okolju smo zgradili model smučišča, po katerem se bo robot gibal. Na robota bo pritrjena kamera, kar bo omogočilo testiranje algoritmov za lokalno izogibanje oviram na podlagi vizualnih informacij v navideznem okolju.
- Razvoj sistema za avtonomno gibanje neholonomnega robota med vratci.  
Vodenje bo temeljilo na osnovi združevanja informacij senzorskega sistema, ki bo vseboval GPS, kamero in giroskope. Večina algoritmov za vodenje neholonomnih robotov upošteva zvezno odvisnost med zasukom koles in radijem zavoja. Smučanje z zarezno tehniko pa vsebuje nezveznost. Pri majhnih naklonih smuči glede na podlago se smučar giblje naravnost. Ko se naklon smuči na podlago poveča, rob smučke zareže v sneg in s tem preide v zavoj z določenim radijem, ki se z večanjem naklona manjša. Sistem bo moral na podlagi zaznane okolice preklapljati med ravnimi odseki in zavoji z radijem, manjšim od stranskega radija smučke.

### 1.3 Metodologija

Razvoj in izdelava robota smučarja zahteva intenzivno testiranje in prilagajanje kompleksnih algoritmov vodenja. Nerealno je pričakovati, da bomo lahko vsa testiranja opravili neposredno na smučišču. Zato je zgrajen kompleksen simulacijski sistem, ki bo temeljil na virtualnem okolju. Ta model bo omogočal tudi simulacijo

zajemanja vizualnih informacij s kamero, ki je ena ključnih problematik disertacije. Seveda pa je potrebno končno verifikacijo opraviti na realnem sistemu. V našem primeru to pomeni testiranje na smučišču. Da bi se izognili testiranju realnega sistema zgolj na smučišču, bomo za simulacijo smuči s poudarjenim stranskim lokom uporabili rolke in testiranja opravili tudi na betonski klančini.

Najprej bomo zasnovali, skonstruirali in izdelali mehanizem, ki omogoča izvajanje zavojev s pomočjo smučk s poudarjenim stranskim lokom ali z rolkami. Ta mehanizem je sposoben na podoben način kot človek izvajati zarezne zavoje s krčenjem in iztegom nog. Stabilnost mehanizma bomo zagotavljali tudi s pomočjo gibanja trupa. V nadaljevanju drugega poglavja bomo predstavili kinematični in dinamični model robota, ki smo ga uporabili za izbiro pogonov in senzorskega sistema. Dinamični model robota smučarja, razvitega v okolju SiMechanics, smo zgradili na podlagi podatkov iz orodja za računalniško podprto načrtovanje I-DEAS. V modelu smo predpostavili, da robot izvaja zgolj čiste zarezne zavoje brez oddrsavanja, ob tem pa sta smučki vedno v stiku s podlago, če je sila na podlago večja od določene meje. V model je vključeno tudi okolje, kar zajema predvsem lokalne naklone površine. Ob upoštevanju hitrosti gibanja pa na model delujejo še radialne sile.

Za namen vodenja je bilo potrebno razviti tudi poenostavljeni dinamični model smučarja v zavoju. Model v okolju SimMechanics namreč ni primeren za izdelavo algoritmov vodenja.

V tretjem poglavju opisujemo algoritme vodenja, ki zagotavljajo stabilno izvajanje zavojev na klančini. Stabilnost sistema smo ocenjevali na osnovi kriterija ZMP, ki je predstavljen v začetku poglavja. V nadaljevanju podajamo poenostavljen model robota, ki omogoča izračun ZMP-ja v realnem času, kar je potrebno za vodenje sistema. Na osnovi tega modela smo razvili algoritem, ki na podlagi zaznanega naklona in hitrosti ohranja stabilnost. Na podlagi modela se določijo meje stabilnih naklonov, ki jih lahko dosežemo z omejenimi pospeški. Ti pospeški so omejeni tako, da zagotavljajo stabilnost tudi ob upoštevanju nemodeliranih sil okolja.

V prečni smeri v zavojih delujeta poleg nemodeliranih sil okolja še komponenta gravitacije in radialna sila. Območje stabilnih naklonov smučarja, ki ga bomo ocenjevali z uporabo ZMP, je odvisno od lokalnega naklona podlage, hitrosti in pospeškov težišča robota kot tudi od gibanja robota samega. Točka ZMP se mora vedno nahajati znotraj podporne podlage znotraj smuči. Z vodenjem zagotovimo, da

se ZMP nahaja v najbolj stabilni poziciji ob upoštevanju zmožnosti realnega sistema.

V četrtem poglavju je predstavljeno vodenje robota na smučišču. V tem delu smo predstavili algoritme za sprotno generacijo trajektorij za vodenje robota po smučarski progi, ki jo določajo smučarska vratca. Lego vratc določamo s pomočjo kamere. Razvili smo algoritem, ki izračunava oddaljenost in lego vratc v realnem času na osnovi oblike in velikosti vratc. Obdelava slik temelji na podlagi iskanja robov in barvne informacije. Da bi se izognili problemom s spreminjajočo osvetlitvijo, smo vizualno zaznavanje izvedli v HSV barvnem prostoru. Pri obdelavi danih podatkov smo upoštevati še stranske naklone podlage in naklon smučarja na podlago. Lokalni naklon smučarja na podlago zaznamo s pomočjo inklinometra.

V petem poglavju je predstavljena zgradba navideznega okolja, ki vsebuje modele robota in okolja. Ob tem pa je predstavljena še vizualizacija robota in okolja. V nadaljevanju poglavja je predstavljena mehanska zgradba samega robota. Opisani so tudi uporabljeni senzorski sistemi. Na koncu pa je predstavljena še zgradba sistema vodenja, ki je bila implementirana na robota.

V predzadnjem poglavju smo predstavili rezultate simulacije in rezultate eksperimentov z robotom na klančini in smučišču. Pri poizkusih smo najprej preverili stabilnost robota pri rolkanju. V tem primeru so se podatki o stabilnosti med senzorskimi in modeliranimi vrednostmi dokaj ujemali. Izvedeni so bili tudi testi na različno postavljenih progah.

V nadaljevanju smo podali rezultate testiranja na smučišču. Analizirani smo dve tipični vožnji robota na smučišču v različnih razmerah podlage. Na podlagi te analize smo ugotovili tudi, da je radij zavoja smučk s poudarjenim stranskim lokom zelo odvisen od strukture snežne podlage in hitrosti.

## 2 Model robota smučarja

### 2.1 Motivacija

Raziskave v robotiki se v zadnjem času osredotočajo na humanoidno in servisno robotiko. Končni cilj je posnemanje človeka pri vsakdanjih opravilih, najprej pa preučevanje obnašanja človeka pri različnih nalogah. Čeprav se nekatere raziskave osredotočajo na naloge, ki so manj pomembne za uporabo robotov v vsakdanjem okolju (kot na primer kolesarjenje, drsanje, plezanje), pa so ključnega pomena pri razumevanju gibanja človeka in pri razvoju algoritmov za gibanje v nestrukturiranem okolju. Sem spada tudi oponašanje alpskega smučanja z robotom. Ne glede na prej našteto pa se pojavljajo tudi potrebe po specialnih robotih, ki bi lahko služili kot inštruktorji in pomočniki pri posameznih nalogah. Taki roboti bi bili uporabni tudi pri učenju in izvajanju posameznih športnih aktivnosti. V preteklosti so bili predstavljeni posamezni poskusi izvajanja alpskega smučanja s pomočjo raznih mehanizmov in tudi robotov. Ti primeri so se osredotočili predvsem na izvajanje zavojev s pomočjo enostavnih gibanj. Izkazalo se je, da lahko z opazovanjem teh mehanizmov lažje dojamemo bistvene aktivnosti pri smučanju. Glede na to lahko realno pričakujemo, da bi lahko robota uporabili tudi pri procesu učenja smučanja. Drug pomemben vidik uporabe robota smučarja je testiranje smučarske opreme. Ne glede na številne naprave, ki obstajajo za testiranje smučarske opreme, se večina testiranja opravi v statičnih razmerah. Te naprave omogočajo ponovljive meritve, vendar pa se pri smučanju mnoge slabosti opreme pokažejo šele pri testiranju na smučiščih v dinamičnih razmerah. Le del teh testiranj je izvedenih s pomočjo raznih senzorjev (sile, navori, zaznavanje gibanja). Velik del teh testiranj pa temelji zgolj na subjektivni oceni izkušenih smučarjev. Tudi lastnost človeka, da se hitro adaptira na različno opremo, predstavlja v fazi testiranja oviro. Robot smučar bi bil lahko pri tem bolj uspešen. Robotu lahko namreč predpišemo enako gibanje ne glede na izbrano opremo in na podlagi zabeleženih podatkov podamo bolj neodvisno oceno opreme. Možnosti uporabe robota smučarja so še druge. Omenimo samo nadzor in pregled smučišč, pomoč pri pregledovanju in

zavarovanju plazovitega terena itd. Gotovo pa so zelo pomembna in tudi neposredno in takoj uporabna znanja, ki omogočajo gibanje po smučišču in hkratno zaznavanje okolice. Na podlagi teh informacij, ki so zelo podobne človeškim, je mogoče zgraditi vodenje, ki oponaša človeško gibanje na smučišču in ob tem omogoča testiranje algoritmov za navigacijo in gradnjo zemljevidov. Ta znanja lahko neposredno uporabimo pri avtomatski navigaciji v prometu, avtomatski navigaciji invalidskih vozičkov in vodenju specialnih enokolesnih in dvokolesnih vozil.

## **2.2 Modeliranje človeka smučarja**

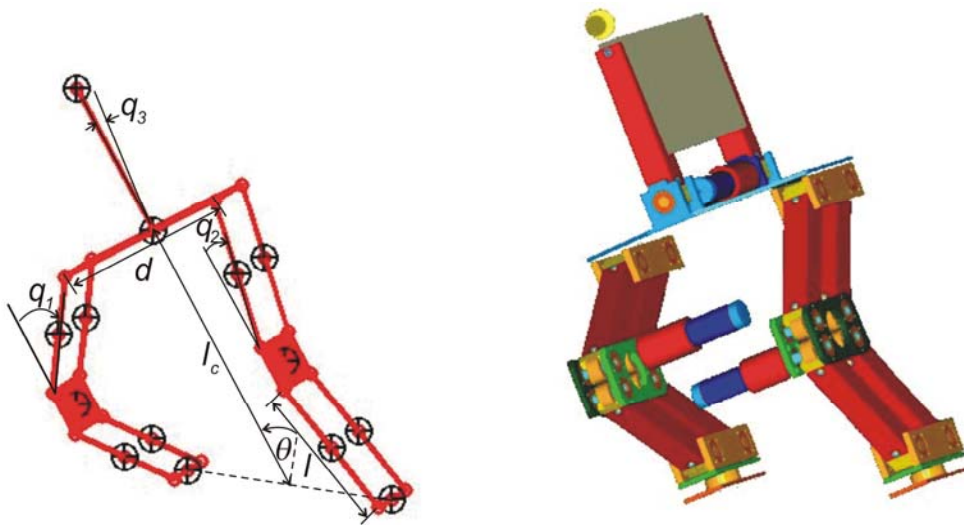
Človeška noga je kompleksen biomehanski sistem, ki ga najpogosteje modeliramo s sedmimi prostostnimi stopnjami (3 v gležnju, 1 v kolenu in 3 v kolku). Pri smučanju se gleženj nahaja v smučarskem čevlju, ki zaradi svoje zgradbe dovoljuje zgolj fleksijo in ekstenzijo gležnja. Kolk ima tri prostostne stopnje, in sicer fleksijo in ekstenzijo, abdukcijo in addukcijo ter notranjo in zunanjo rotacijo, kar človeku omogoča izvajanje smučarskih zavojev. Slednje gibanje se pri smučanju uporablja zgolj pri pluzni tehniki, medtem ko se pri smučkah s poudarjenim stranskim lokom uporabljata zgolj ostali gibanji v kolčnem sklepu. V smučarskih krogih je sprejeto prepričanje, da človek izvaja zarezne zavoje na dva načina [1]. Najpogosteje se izvajajo s fleksijo in ekstenzijo nog. V tem primeru vsi trije sklepi delujejo na način, ki povzroči izteg ali pokrčitev noge. Na ta način zavoje izvedejo mišice, ki spadajo med najmočnejše v človeškem organizmu. Zarezne zavoje se lahko izvede tudi z abdukcijo in addukcijo v kolku, vendar pa so mišice, ki skrbijo za to gibanje, šibkejše in zato se ta način uporablja zgolj pri nižjih hitrostih, ko so sile, ki delujejo na telo, manjše. Pri tej obliki izvedbe zavojev je celotna akcija, ki je potrebna za izvedbo zavoja, izvedena samo s kolčnim sklepom. V obeh primerih pa smučar stabilnost med smučanjem vzdržuje z gibanjem trupa in rok. Na podlagi teh ugotovitev smo se odločili, da bo naš robot oponašal smučanje s pomočjo fleksije in ekstenzije nog. Pri tem načinu je pomembno gibanje izteg in pokrčenje nog, medtem ko se orientacija trupa ne spreminja, saj spremembo orientacije v kolenu kompenzirata gleženj in kolčni sklep. Zato smo v našem primeru zgradili nogo v obliki dvojnega medsebojno povezanega paralelograma, kot je prikazano na sliki 2.1. S tem je zagotovljeno, da se orientacija na vrhu noge ne spremeni glede na podplat. Tako dosežemo zgolj translatoren premik v  $z$  smeri. Enako gibanje pri

smučanju izvaja tudi človek, in sicer izvede izteg ali skrčenje noge s kolkom, kolenom in gležnjem. Razlika je v dejstvu, da človek med gibom ohranja tudi silo v sagitalni ravnini v zelo ozkem področju, medtem ko je pri predlaganem robotu delovanje sile v sagitalni ravnini odvisno zgolj od lokalnega naklona.

### 2.3 Zgradba robota

V tem delu opisujemo robota, ki smo ga razvili za eksperimentalno preverjanje naše hipoteze, da je robot sposoben avtonomno smučati po smučarski progi. Robot je sestavljen iz dveh nog in trupa, ki so med sabo povezani s pomočjo ploščadi. Ploščad ima podobno funkcijo kot pri človeku medenica. Nogi sestavljata dva paralelograma, ki ohranjata kot med smučmi in ploščadjo. S tako kinematično zasnovo mehanizma lahko izvajamo dve vrsti gibov, in sicer

- dvig ploščadi na določeno višino  $l_c$ ,
- naklon nog glede na podlago  $\theta$ .



Slika 2.1: Model robota

Celotno mehansko zgradbo robota zaključuje še trup, ki je pritrjen na ploščad. Predlagana mehanska konstrukcija omogoča izvajanje zavojev s smučkami s poudarjenim stranskim lokom in tudi z rolkami. Gibanje trupa je namenjeno zagotavljanju stabilnosti na klančini. S pomočjo trupa je mogoče kompenzirati tako kratkotrajne sunke, ki se pojavijo kot posledica hitre spremembe naklona podlage, kot tudi dolgotrajne, ki so posledica radialne sile v zavoju. Predlagani mehanizem

omogoča izvedbo naklona na podlago. S tem gibanjem je robot sposoben zavijanja s smučkami s poudarjenim stranskim lokom.

## 2.4 Zavijanje z nagibanjem

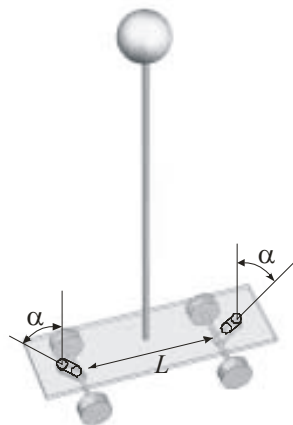
Nekateri mehanski neholonomni sistemi zavijajo s pomočjo nagibanja. Tipični primeri so kolesa, rolke in smuči s poudarjenim stranskim lokom. Robot, ki smo ga predstavili v tem poglavju, je sposoben smučanja samo z uporabo smuči s poudarjenim stranskim robom. Da smo omogočili testiranje robota tudi v poletnem času, smo namesto smuči uporabili rolke, ki omogočajo posnemanje smučanja na klančinah. V nadaljevanju bomo podali matematični model zavijanja z nagibanjem za rolke in smuči.

### *Model zavijanja rolke*

Rolke so tipični primer mehanizma, kjer zavijamo z nagibanjem. Sestavljene so iz dveh parov kolesc, ki imata naklonjeni krmilni osi, kot prikazuje slika 2.2. Osi sta postavljeni pod kotom  $\alpha$  glede na vertikalno os tako, da stranski nagib povzroči zavoj v smeri nagiba. Ob predpostavki, da so vsa kolesca v stiku s podlago, je radij zavoja v odvisnosti od  $\theta$  podan z enačbo [9]

$$R_r = \frac{L \tan(\alpha)}{2\theta}, \quad (1)$$

kjer je  $L$  razdalja med paroma kolesc prikazana na sliki 2.2.. Ta enačba velja le v primeru, če ne prihaja do stranskega zdrsa posameznih kolesc.



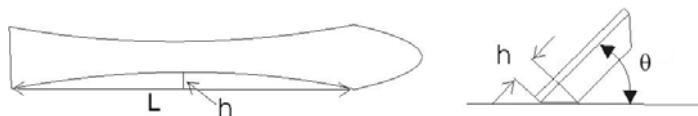


## Slika 2.2 Model rolke

V našem primeru imamo par vzporednih rolk, po eno za vsako smučko. Ker sta obe rolki naklonjeni na podlago za enak kot, izvajata enak zavoj z radijem, izračunanim po enačbi (1). Ker pa morata biti radija zunanje in notranje rolke pri zavijanju različna, prihaja med vožnjo do zdrsanja. V modelu smo zato predpostavili, da je radij zavoja definiran za točko, ki se nahaja na sredini med obema rolkama. To pomeni, da notranja rolka izvede zavoj z manjšim radijem od predpisanega, zunanja pa z večjim.

### *Model zavijanja smučī s poudarjenim stranskim lokom*

Gibanje smučarja na smučišču določajo kontaktne sile med smučmi in podlago. Medtem ko se pri klasičnih smučeh zavoj izvaja z zdrsanjem, ki ga uravnavamo tudi s pomočjo rotiranja trupa, pa se pri smučeh s poudarjenim stranskim lokom zavoj izvede zgolj z nagibom smučke na podlago. Pri tem je pomembno, da smučka zgolj zareže v podlago in ne oddrsava. V tem primeru se robnik smučke po celotni dolžini zareže v sneg. Zaradi oblike smučke, ki je del izseka krožnice, se smučka zapelje po krivulji krožnice.



Slika 2.3: Krožni odsek smučke s poudarjenim stranskim lokom

Medtem ko je tetiva krožnice enaka dolžini smučke  $L$ , pa se v primeru naklona na podlago  $\theta$  zaradi upogiba smučke na sredini poveča višina krožnega odseka na podlagi kot prikazuje slika 2.3. S spreminjanjem naklona smučke na podlago in ob predpostavki, da je sila pravokotna, na smučko, dovolj velika, da povzroči upogib smučke, je radij zavoja v odvisnosti od naklona smučke na podlago enak

$$r = \frac{L^2 \cos \theta + \frac{h^2}{\cos \theta}}{2h}, \quad (2)$$

kjer je  $L$  dolžina smuči,  $h$  pa je višina krožnega izseka in je podana z enačbo

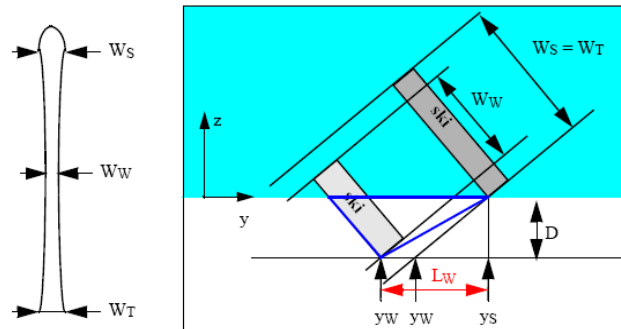
$$h = \frac{W_S + W_T - 2W_W}{2}. \quad (3)$$

Parametri  $W_S$ ,  $W_T$  in  $W_W$  predstavljajo širino smuči na sprednjem, zadnjem in srednjem delu smučke, kot jo prikazuje slika 2.4. Ta zveza velja zgolj ob pogoju, da se smučka po vsej svoji dolžini enako zareže v sneg. Predstavljeni so bili tudi modeli, ki upoštevajo še različne ugreze po dolžini smučke. Tako je Howe [21] predstavil model, ki poleg naklona smučke upošteva še globino vreza  $d$  na sredini smučke, kar je utemeljil z dejstvom, da je pritisk smučke na podlago neposredno pod smučarskim čevljem večji, kot je na sprednjem in zadnjem delu smuči. Zato smučka v tem delu globlje zareže v sneg.

$$r_{\text{Howe}} = \frac{L^2}{8 \left[ \frac{(W_S + W_T - 2W_W)}{4 \cdot \cos \theta} + d \cdot \sin \theta \right]} \quad (4)$$

Federolf [12] pa je na osnovi meritev smučk v statičnih razmerah prišel do Gauss-ove odvisnosti radija zavoja v odvisnosti od naklona. V teh meritvah je tudi pokazal, da je radij zelo odvisen od sile, s katero delujemo na podlago. Meritve pa so pokazale, da je zelo pomemben dejavnik tudi togost smuči in snežna podlaga. Vse metode izračuna radija zavoja veljajo zgolj za naklone smučk, večje od določene vrednosti, in v primeru statičnih razmer. Pri manjših naklonih se v realnosti pojavi zdrs smučke, zato so v teh primerih radiji zavojev mnogo večji, kot je izračunani stranski radij smučke. V našem primeru smo se odločili, da bomo za oceno radija zavoja uporabili metodo, ki upošteva zgolj naklon smuči na podlago, saj so ostali parametri zelo odvisni od trenutnih razmer na smučišču. Meritve v dinamičnih razmerah pa so pokazale, da pri majhnih naklonih smučke na podlago prihaja do zdrsov in smučka sploh ne izvede zavoja ali pa je radij izvedenega zavoja večji od radijev, izračunanih na podlagi enačb (2, 4). To nas je vodilo do zaključka, da je dejanski izvedeni radij na smučkah pri nizkih naklonih na podlago bolj odvisen od zunanjih dejavnikov, kot je na primer struktura snega, saj v primeru ledene proge smučka pri nizkih naklonih ne zareže v podlago. Na podlagi tega smo se odločili, da bodo za namen modeliranja radija zavoja pri majhnih naklonih smuči

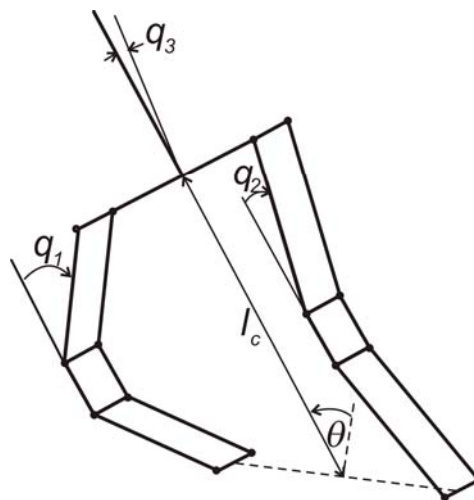
(  $|\theta| < 5^\circ$  ) na podlago uporabili obratno vrednost naklona  $z$  radijem. Pri večjih naklonih smo pri modeliranju zavijanja smučke uporabili enačbo (2).



Slika 2.4: Predstavitev smučke s poudarjenim stranskim lokom

## 2.5 Kinematični model robota

Predlagani robot ima dve nogi, ki s podlago tvorita paralelni mehanizem.



Slika 2.5: Kinematični model robota

Kot smo že omenili v poglavju 2.3, definiramo prostostni stopnji tega mehanizma kot razdaljo  $l_c$  od točke vpetja trupa do točke na podlagi med smučkama, ki je enako oddaljena od obeh smučk, in naklon mehanizma  $\theta$  na podlago. Medtem ko razdalja

$$l_c = l_d + l(c_1 + c_2) \quad (5)$$

nima vpliva na smer smučanja, ima naklon smučarja

$$\theta = \text{sign}(q_1 - q_2) \cdot \text{arc cos} \left( \frac{d}{\sqrt{(2l(c_1 - c_2))^2 + d^2}} \right) \quad (6)$$

neposreden vpliv, saj je to naš mehanizem zavijanja, kot navaja enačba (2). Pri tem smo uporabili naslednje okrajšave

$$c_1 = \cos(q_1)$$

$$c_2 = \cos(q_2),$$

ker sta  $q_1$  in  $q_2$  kota v skladu s sliko 2.5.

Radij zavoja je v tem primeru enak

$$r = \frac{\frac{L^2 d}{4\sqrt{4l^2(c_1 - c_2)^2 + d^2}} + \frac{h^2 \sqrt{4l^2(c_1 - c_2)^2 + d^2}}{d}}{2h}. \quad (7)$$

Definirajmo še Jacobijevo matriko za sistem nog

$$\begin{bmatrix} \dot{r} \\ \dot{l}_c \end{bmatrix} = J_s \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (8)$$

$$J_s = \begin{bmatrix} \frac{a-b}{2h} & \frac{a+b}{2h} \\ -ls_1 & -ls_2 \end{bmatrix}, \quad (9)$$

kjer je

$$a = \frac{L^2 dl^2 (c_1 - c_2) s_1}{(4l^2 (c_1 - c_2)^2 + d^2)^{3/2}} \quad (10)$$

in

$$b = 4 \frac{l^2 h^2 (c_1 - c_2) s_1}{d \sqrt{4l^2 (c_1 - c_2)^2 + d^2}}, \quad (11)$$

Pri tem smo spet uporabili okrajšave  $s_i = \sin(q_i)$  in  $c_i = \cos(q_i)$ .

V primeru rolk veljajo drugačne zveze. Radij zavoja za rolke v odvisnosti od kotov  $q_1$  in  $q_2$  je

$$r_r = \frac{L \tan(\alpha)}{2a \tan\left(\frac{2l(c_1 - c_2)}{d}\right)} \quad (12)$$

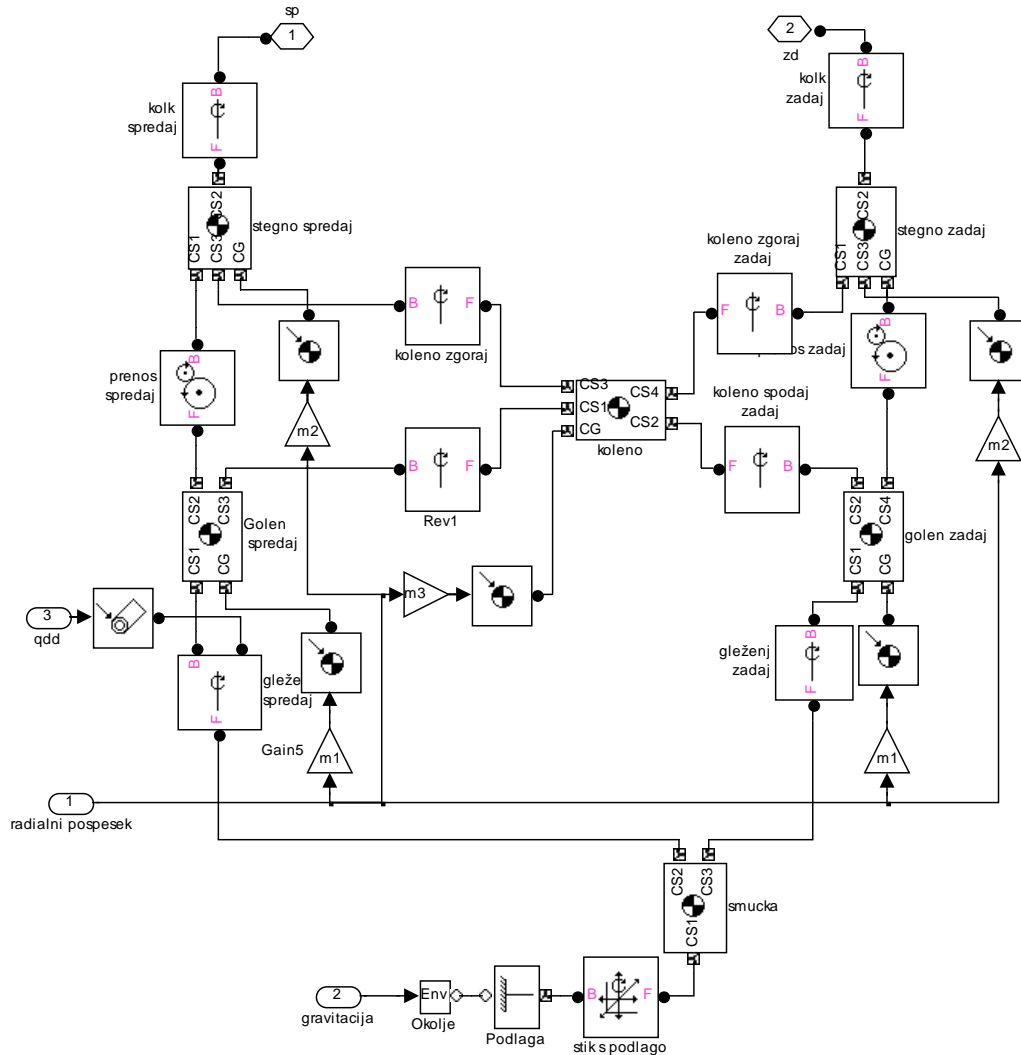
Jacobijeva matrika za rolke pa je

$$J_r = \begin{bmatrix} \frac{L_r \tan(\alpha) ds_1}{2a \tan^2\left(\frac{2l(c_1 - c_2)}{d}\right) \left(d + (2l(c_1 - c_2))^2\right)} & \frac{L_r \tan(\alpha) ds_1}{2a \tan^2\left(\frac{2l(c_1 - c_2)}{d}\right) \left(d + (2l(c_1 - c_2))^2\right)} \\ -ls_1 & -ls_2 \end{bmatrix}, \quad (13)$$

Pri tem  $L_r$  predstavlja dolžino med paroma kolesc.

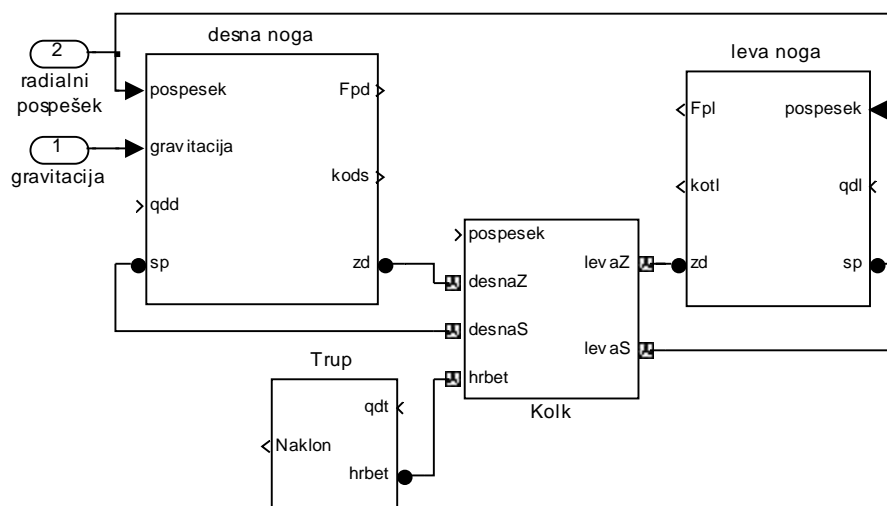
## 2.6 Dinamični model

Dinamični model robota smo zgradili v okolju Sim Mechanics-u z uporabo vgrajenih objektov za segmente in sklepe. Slika 2.6 prikazuje model ene noge v okolju SimMechanics.



Slika 2.6: Model noge, zgrajen v Matlab SimMechanics

Vhod v model so radialni pospeški, vektor gravitacije in želeno gibanje robota. Podatki o vztrajnostni matriki in masi posameznih delov noge so vgrajeni v model. Tako zgrajene posamezne dele lahko združujemo in dobimo celotno konfiguracijo robota z nogama, trupom in platformo, kot je prikazana na sliki 2.7.



Slika 2.7: Model celotnega mehanizma

Na celoten mehanizem delujeta gravitacija in radialna sila. Ker se smer in velikost teh dveh sil spreminjata v odvisnosti od naklona površine, naklona smučk in hitrosti, se vrednosti pospeškov iz modela okolja prenašajo v model smučarja. Radialna sila ima samo komponento v lateralni smeri, medtem ko se vektor gravitacije spreminja v odvisnosti od lokalnega naklona površine. Ker smo v modelu predpostavili, da ima robot obe nogi ves čas v kontaktu s podlago, se razdalja med smučkama spreminja, zato prihaja med podlago in smučkami v lateralni ravnini do zdrsov. V primeru, da je hitrost v sagitalni ravnini skoraj enaka 0, se smučka na nogi, ki izvaja izteg, zareže v sneg do določene globine. Nadaljnje iztegovanje poveča upor snega, kar povzroči, da se celoten robot premakne stran od smučke. Iz tega razloga pri nizkih hitrostih robot ne sme izvajati nagibov, ker to povzroči zelo velike sile in bi lahko privedlo do poškodb mehanizma. Med smučanjem pa smučka zareže v nov sneg in zaradi tega ne zarobi tako silovito. Želeni nagib na podlago lahko mehanizem izvede na dva načina, in sicer samo s krčenjem in iztegovanjem posamezne noge ali s sočasnim iztegovanjem in krčenjem nog. V prvem primeru se naklon smučarja na podlago spreminja okrog manj obremenjene smučke. Pri sočasnem gibanju obeh nog pa se nagib smučarja izvaja okrog osi, ki se nahaja znotraj območja med smučkama. Če se smučar giblje v sagitalni ravnini in izvaja gib  $\dot{q}_1 = -\dot{q}_2$ , se ta os nahaja točno na sredini med nogama. Tako zgrajeni model je služil za izbiro ustreznih komponent, ki so bile

uporabljene na robotu.

## 2.7 Model okolja

V prejšnjem podpoglavju smo predstavili dinamični model robota. Na ta model pa vplivajo tudi zunanje sile, ki so posledica sil, s katerimi okolje deluje na robota. Na robota delujejo sila gravitacije, radialna sila ter reakcijske sile podlage. Smer gravitacije je ob predpostavki, da so smuči v stiku s podlago, odvisna od lokalnega naklona površine. Radialna sila pa je odvisna od hitrosti in radija zavoja, ki ga robot izvede z naklonom na podlago. Robotu se hitrost spreminja zaradi zunanjih sil, ki delujejo nanj. Vpliv teh sil na pospešek v sagitalni smeri je

$$a_R = \frac{g_s + F_t + F_u}{m_R},$$

kjer je  $g_s$  dinamična komponenta gravitacije, ki deluje v sagitalni smeri robota,  $F_t$  je sila trenja med smučmi in podlago,  $F_u$  pa sila zračnega upora,  $m_R$  pa je masa celotnega robota. Sila trenja pri smučanju je odvisna od strukture podlage in smuči. Uporabili smo model, ki vključuje statično in viskozno trenje. Sila viskoznega trenja  $F_v$  je

$$F_v = k_v v,$$

pri čemer je  $k_v$  koeficient viskoznega trenja in  $v$  hitrost gibanja v sagitalni smeri.

Viskozному trenju je dodana še komponenta statičnega trenja  $F_s$ , ki je neodvisna od amplitude hitrosti, je pa odvisna od smeri le-te

$$F_s = k_s \text{sign}(v),$$

kjer je  $k_s$  koeficient statičnega trenja. V model ni vključeno lepljenje, saj so koeficienti lepljenja med smučko in snegom zanemarljivi.

Drug del modela okolja je gibanje robota po terenu. Smučar je po svojih lastnostih neholonomnen sistem. Holonomni sistemi so sposobni gibanja v vseh treh prostostnih stopnjah na ravnini, to je translacija v obeh smereh in rotacija okoli lastne osi. Gibanje neholonomnih sistemov v ravnini je bolj omejeno, saj imajo zgolj dve prostostni stopnji, ki ju opišemo na način



$$\dot{X} = v \cos(\varphi), \quad \dot{Y} = v \sin(\varphi) \quad (14)$$

z naslednjo omejitvijo [38]

$$\dot{X} \sin(\varphi) - \dot{Y} \cos(\varphi) = 0. \quad (15)$$

Enačbe (14)-(16) veljajo v ravnini XY. Ker pa se smučar giblje po razgibani podlagi, je potrebno gibanje prilagoditi podlagi, po kateri se robot giblje. Podlaga je definirana z vektorjem normale  $n$ . Kot  $\varphi$  je definiran na podlagi rotacije okrog normale in predstavlja trenutno smer gibanja. Smer gibanja je odvisna od hitrosti in radija zavoja.

$$\varphi(t) = \int \frac{v(t)}{R(t)} dt \quad (16)$$

## 2.8 Elastičnost mehanizma

Kot je bilo omenjeno pri zgradbi robota, se zaradi mehanskih prenosov navorov pojavijo med posameznimi zobniki zračnosti. Da bi ugotovili, kakšen vpliv imajo na naklon robota, smo s pomočjo inklinometra ocenili razliko med merjenim naklonom ploščadi robota in izračunanim naklonom na podlagi kinematičnega modela robota. V ta namen smo robotu predpisali določeno gibanje, katero je izvajal na ravni podlagi v sagitalni ravnini. Meritve so pokazale, da se v primeru ko sta obe nogi enako obremenjeni, meritve iz kinematičnega modela in meritve inklinometra dokaj ujemajo.

V primerih različnih sil na noge, na primer, ko je bila notranja noga bolj obremenjena (deluje večja sila), pa je opazno povečanje naklona. To nakazuje na elastičnost samega mehanizma. Ob večji obremenitvi se je bolj obremenjena noga bolj skrčila kot manj obremenjena. Na podlagi te ugotovitve smo zgradili model, ki razen kinematičnih lastnosti upošteva še obremenjenost posamezne noge, in sicer v obliki elastičnosti. Dolžina posamezne noge je v modelu enaka

$$l_i = 2l \cos(q_i) - k_e F_i \sin(q_i), \quad (17)$$

kjer je  $k_e$  koeficient elastičnosti mehanizma nog in  $F_i$  sila na posamezni nogi v smeri pravokotno na smučko. Večina elastičnosti se nahaja v prenosu med motorjem in segmenti mehanizma, kar vsebuje elastičnost v planetarnem prenosu in med zobniki. Vpliv sile na dolžino je v primeru popolnoma iztegnjene noge praktično zanemarljiv zaradi togosti same konstrukcije. S krčenjem pa se ta vpliv povečuje. Dolžino posameznih nog smo merili posredno preko naklona ploščadi. Zato smo robota postavili v posamezne pozicije, kjer so bile izmerjene sile na posamezno nogo, ter izmerili naklon ploščadi, ki povezuje obe nogi, in ga primerjali s kinematičnim modelom.

## 3 Stabilnost smučarja na klančini

### 3.1 Dinamično ravnotežje

Pri lokomociji ločimo med statičnim in dinamičnim ravnotežjem. Statično stabilni roboti vzdržujejo stabilnost z ohranjanjem težišča na področju podporne ploskve, kar pa zadostuje za stabilno delovanje zgolj v primeru, ko se robot giblje zelo počasi in so sile, povzročene z gibanjem samega robota, zanemarljive. Podporna ploskev je definirana kot sklenjena ploskev, ki jo omejujejo površine, s katerimi se robot v določenem trenutku dotika tal. Za razliko od statične stabilnosti hodečih mehanizmov pa se pri dinamičnih kriterijih upoštevajo tudi sile, ki so posledica gibanja samega mehanizma. Robot je dinamično stabilen, če so v gibanju jasno izražene faze, v katerih se projekcija težišča nahaja izven podpornega poligona. V robotiki ni konsenza o definiciji dinamičnega ravnotežja, zato se je razvilo več kriterijev za določanje le-tega. V praksi je najpogosteje uporabljen kriterij točke ničelnega navora (ang. zero moment point - ZMP), ki ga bomo predstavili v nadaljevanju.

### 3.2 Točka ničelnega navora (ZMP)

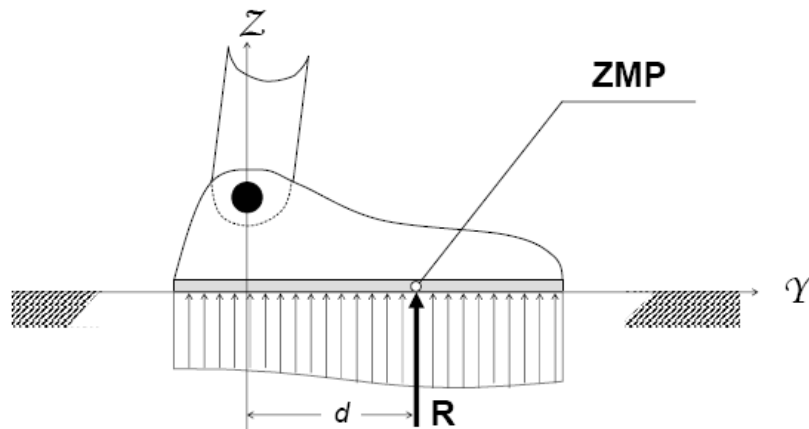
Glede na dejstvo, da je pomemben del tega dela zagotovitev dinamične stabilnosti na podlagi ZMP, je del tega poglavja posvečen kriteriju, ki je bil uporabljen v vodenju. V tem poglavju bomo predstavili koncept ZMP-ja in izpeljanke, ki so nadgradile osnovni koncept. ZMP se večinoma uporablja pri hoji, čeprav zaradi svoje splošnosti omogoča tudi uporabo pri mobilnih robotih.

#### *Definicija*

Definicijo točke ničelnega navora je podal Vukobratović [24, 25]:

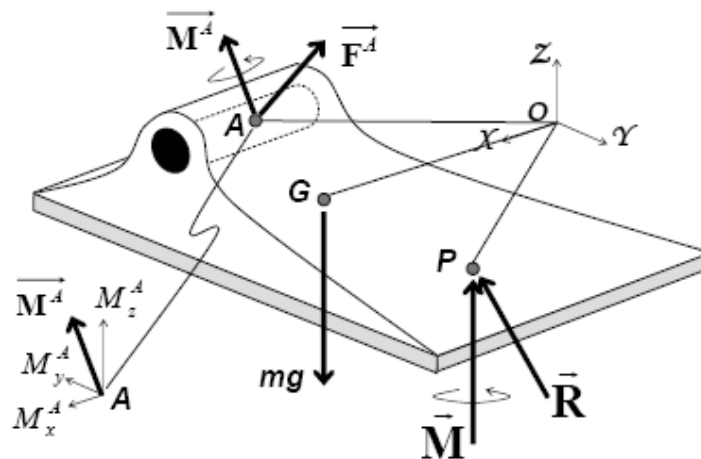
»Ko ima obremenitev enak predznak po celotni površini (slika 3.1), se jo lahko

nadomesti z rezultanto sile  $R$ , s točko delovanja, ki se nahaja znotraj mej stopala. Ta točka na površini stopala, kjer deluje rezultanta  $R$ , se označi kot točka ničelnega navora oziroma na kratko ZMP. «



Slika 3.1: ZMP - prvotna definicija [24]

Predlagan je bil kinematičen model dvonožnega mehanizma, kjer sta stopali v stiku s podlago. Za ohranitev stabilnosti podporne noge in zamah druge je nujen pogoj, da vsa sila podlage deluje zgolj na podporno nogo. Predlagani princip je pomemben zaradi naslednjih stvari. Kot prvo je na podlagi tega možna sinteza trajektorij, ki zagotavljajo prej omenjen pogoj brez upoštevanja proti akcije podlage, ker je celotna zunanja sila v dinamičnem ravnotežju s silo, ki jo proizvedejo pogoni robota. Druga prednost pa je, da se lahko gibanje gradi na podlagi kompenzacije navorov, ker mora biti navor v horizontalni ravnini okrog točke ZMP enak nič. Kasneje je Vukobratović [26] predlagal še zaprtozančno vodenje za ta model.



Slika 3.2: Delujoče sile in navori na nogo [27]

Raziskave so pokazale, da je ZMP primeren kriterij za vodenje hodečih robotov, saj omogoča razširitev tradicionalnih načinov vodenja manipulatorjev z razstavitvijo na fazo planiranja in senzorsko vodenje. Za namen predstavitve ZMP-ja je Vukobratović analiziral gibanje v fazi, ko je samo ena noga v kontaktu s podlago. Podplat te noge pa je po celotni površini v stiku s podlago. Da bi poenostavil analizo, je ločil podporno stopalo od ostalega sistema. S tem se celoten sistem zamenja z njegovim vplivom na stopalo v obliki sile  $F^A$  in  $M^A$  (slika 3.2). Sila  $R$ , ki deluje na podplat v točki  $P$ , ohranja stabilnost sistema. V tej točki deluje tudi navor  $M$  okrog osi  $z$ . Zaradi trenja in predpostavke, da stopalo miruje, je horizontalna komponenta  $R$  nasprotno enaka horizontalnima komponentama  $F^A$ . Na drugi strani vertikalni reakcijski navor  $M$  predstavlja trenje, ki nasprotuje vertikalni komponenti  $M^A$  in navoru, s katerim deluje sila  $F^A$  v točki  $P$ . Če upoštevamo, da ni prišlo do zdrsa, je statično trenje ustvarilo prej naštetih komponente sile  $R$  in navora  $M$ . Vertikalna komponenta sile podlage  $R$  je v ravnotežju z vertikalnimi silami noge. Ker je reakcijska sila podlage, povzročena z nogo, usmerjena navzgor, se horizontalne navori lahko kompenzira zgolj s premiki sile  $R$  znotraj konveksnega področja podplata, ki se pogosto poimenuje tudi podporna ploskev. V primeru, ko je podporna ploskev premajhna, da bi vsebovala prijemališče  $R$ , ki kompenzira zunanje navori, bo sila  $R$  delovala na robu podplata. Nekompenziran del navorov pa povzroči rotacijo sistema okrog roba podplata.

Na podlagi tega je potreben in zadosten pogoj za stabilnost sistema, da se točka  $P$ , v kateri velja

$$M_x=M_y=0, \quad (18)$$

nahaja znotraj podporne ploskve.

V primerih, ko se točka P nahaja izven podporne ploskve, je ta točka imenovana navidezna točka ničelnega navora FZMP (ang. »Fictitious ZMP«) [27]. V primerih, ko točka P doseže rob podpornega poligona, se ZMP ohrani zgolj v primeru, če ni nobenega navora, ki bi povzročil rotacijo. Z nadaljnjim navorom tudi v tej točki pa ZMP preide v FZMP.

Princip ZMP-ja se pogosto uporablja, vendar obstaja več različnih definicij Dasgupta in Nakamura sta ga deklarirala na sledeč način:

„ZMP je definiran kot točka navorov na podlagi, v kateri vsota momentov vztrajnostnih in gravitacijskih sil nima nobene komponente v smeri horizontalnih osi.”

Arakawa in Fukuda [28] sta podana naslednjo interpretacijo, ki jo je povzel tudi Napoleon [29]:

„ZMP je točka na podlagi, v kateri navor  $M(M_x, M_y, M_z)$ , povzročen z reakcijsko silo in reakcijskim navorom, zadosti pogojem  $M_x=0$  in  $M_y=0$ .”

Huang [30] pa se je izrazil na naslednji način:

„ZMP je definiran kot točka na podlagi, okrog katere je vsota navorov, ki so posledica aktivnih sil, enaka nič. ”

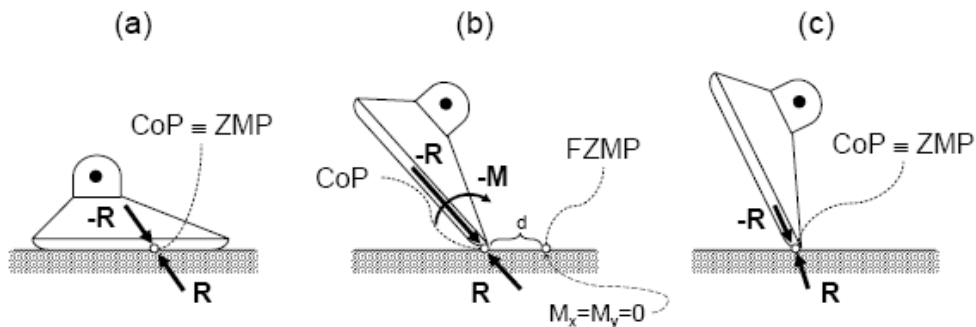
Vse te definicije predpostavljajo, da ZMP ni nujno povezana s podporno ploskvijo. Zelo pogosto se pojavi napaka v razlikovanju med ZMP in točko pritiska (ang. Centre of Pressure - CoP), zato bomo v nadaljevanju pojasnili razliko med njima.

### Točka pritiska (CoP)

CoP je Goswami [43] definiriral kot točko na podlagi, kjer rezultanta sil podlage deluje na stopalo

$$x_{CoP} = \frac{\int (x' - x) F(x') dx'}{\int F(x') dx'}, \quad (19)$$

pri čemer je  $x'$  pozicija na podlagi, kjer deluje sila podlage  $F(x')$ . Vselej, ko je ta sila v ravnovesju z aktivnimi silami (gravitacija, vztrajnost ...), je ta točka enaka ZMP. Iz tega sledi, da se nahajata ZMP in CoP v isti točki, če je robot dinamično stabilen. Razlika se pojavi v primerih, ko se ZMP nahaja zunaj podporne ploskve, saj CoP ne obstaja zunaj podporne ploskve in se rezultanta sil podlage v tem primeru pojavi na robu podporne ploskve.



Slika 3.3: Razlika med ZMP in CoP: a) dinamično stabilna lega, b) nestabilna lega, c) stabilna lega na prstih.

Da bi predstavili povezavo med ZMP in CoP, smo na sliki 3.3 prikazali tri tipične primere kontakta noge in podlage. V primeru dinamično stabilnih leg se ZMP in CoP nahajata v isti točki. Na sliki 3.3 b) je predstavljen primer, ko se ZMP in CoP ne ujemata. V tem primeru se namreč na robu podporne ploskve pojavlja še navor, ki bo povzročil prevrnitev. Slika 3.3 c) pa predstavlja primer, pri katerem se ZMP nahaja na robu podporne ploskve, in sistem je mejno stabilen.

Če povzamemo, se ZMP in CoP nahajata zmeraj v isti točki, če je robot stabilen, medtem ko se FZMP in CoP ne nahajata nikoli v isti točki, saj CoP ne obstaja izven podporne ploskve. Razlika med njima je predvsem iz kota pogleda. ZMP se

izračunava na podlagi znanega gibanja robota. CoP pa se meri na osnovi senzorjev sile, ki delujejo na podplat.

### 3.3 Poenostavljen model smučarja v lateralni ravnini

Predhodno predstavljen model, zgrajen v Matlab SimMechanics, nudi tako informacije o silah, s katerimi deluje podlaga, kot tudi podatke o potrebnih navorih motorjev, ki zagotavljajo želeno gibanje robota. Slabost pa je, da ta model ne omogoča izračuna teh vrednosti v realnem času. Zato smo za uporabo na realnem sistemu uporabili poenostavljen model smučarja, ki temelji na naslednjih predpostavkah. Glavni problem pri smučanju je stabilnost v lateralni ravnini. V sagitalni smeri ima smučar podporo v relativno dolgih smučeh in vzdržuje prijemališče sile na zelo ozkem območju pod smučarskim čevljem. V lateralni smeri pa je podporna ploskev mnogo ožja. Ob tem pa se razen gravitacije pojavijo še radialne sile in navori zaradi različnega iztega nog. Zato smo v poenostavljenem modelu upoštevali zgolj vplive, ki delujejo v lateralni ravnini. Na podlagi tega poenostavljen dinamični model smučarja zapišemo na naslednji način

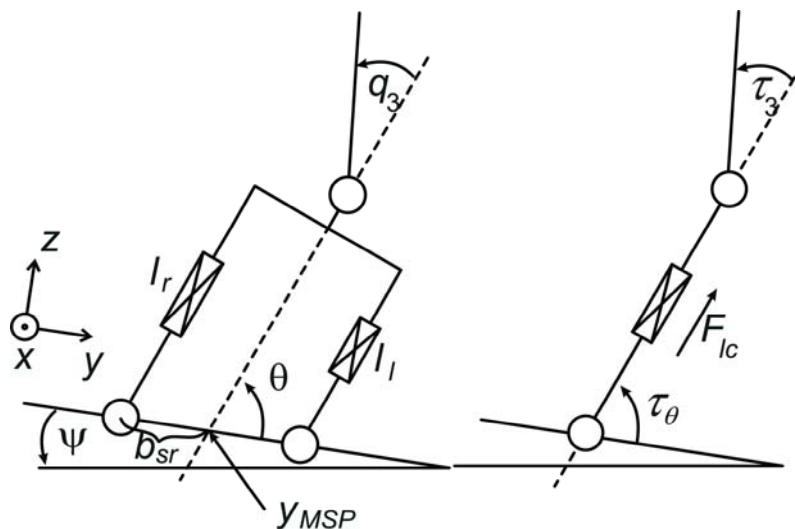
$$H(q_{sp})\ddot{q}_{sp} + C(\dot{q}_{sp}, q_{sp})\dot{q}_{sp} + G(\dot{q}_{sp}) = \tau. \quad (20)$$

Pri tem velja

$$q_{sp} = \begin{bmatrix} \theta \\ l_c \\ q_3 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_\theta \\ F_{lc} \\ \tau_3 \end{bmatrix}, \quad (21)$$

kjer je  $\tau_\theta$  navor, potreben za gibanje naklona robota na podlago,  $F_{lc}$  sila, potrebna za gibanje višine robota, in  $\tau_3$  navor za gibanje trupa.





Slika 3.4: Poenostavljen model smučarja na klančini

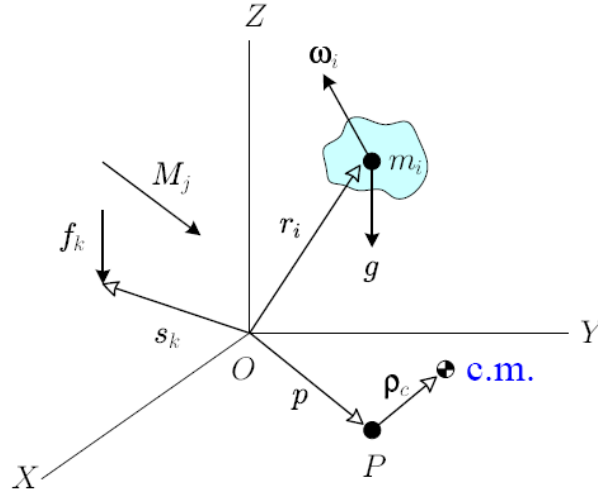
Noge v tej ravnini izvajajo zgolj translacijo v smeri iztega nog. Razdalja med smučkama je odvisna od naklona smučke na podlago. V statičnih razmerah se zaradi oblike robnika smučke zunanja smučka v primeru iztega zunanje smučke zareže v podlago do določene globine, nato notranja smučka zdrsne.

### 3.4 ZMP robota na klančini

Za predhodno predstavljen model smo zgradili dinamični model na osnovi Newton-Euler-jeve metode. Obravnavamo  $i$ -ti segment robota z maso  $m_i$ , težiščem v  $r_i$  in vztrajnostno matriko  $I_i$ . Nanj delujejo zunanje sile in navori  $F_k$  in  $M_j$ . Rotacijska enačba gibanja sistema v točki P ( $p=[x_P \ y_P \ 0]^T$ ) na podlagi  $z=0$  je enaka

$$\sum_i (r_i - p) \times m_i (\ddot{r}_i + g) + \sum_i [I_i \dot{\omega}_i + \omega_i \times I_i \omega_i] - \sum_j M_j - \sum_k (s_k - p) \times F_k = M_p. \quad (22)$$

Vektor  $s_k$  kaže v točko, kjer deluje zunanja sila  $F_k$ ,  $g$  je vektor težnostnega pospeška na površini, ki je odvisen od lokalnega naklona površine  $\psi$ , kot je prikazano na sliki 3.4.



Slika 3.5: Predstavitev za izračun analitičnega modela

Ob predpostavki, da je razen gravitacije edina zunanja sila radialna, ki deluje enako kot gravitacija v težišču, lahko enačbo navora za posamezen segment zapišemo v obliki

$$\sum_i (r_i - p) \times m_i (\ddot{r}_i + g + a_{ri}) + \sum_i [I_i \dot{\omega}_i + \omega_i \times I_i \omega_i] = M_p, \quad (23)$$

kjer je  $a_{ri}$  ( $a_{ri} = [0 \ v^2/R_i \ 0]^T$ ) radialni pospešek težišča  $i$ -tega telesa.  $R_i$  je radij zavoja za posamezno težišče segmenta. Ker je  $I_i \dot{\omega}_i + \omega_i \times I_i \omega_i$  neodvisen od točke P, uvedemo novo spremenljivko  $T_i = I_i \dot{\omega}_i + \omega_i \times I_i \omega_i$ . Z upoštevanjem definicije ZMP v točki ničelnega navora v ravnini podlage  $p_{ZMP} = [x_{ZMP} \ y_{ZMP} \ 0]$  deluje samo navor  $M_P = [0 \ 0 \ M_z]^T$ . Komponenti ZMP sta

$$x_{ZMP} = \frac{\sum_i m_i (\ddot{z}_i + g_z) x_i - \sum_i m_i (\ddot{x}_i + g_x) z_i}{\sum_i m_i (\ddot{z}_i + g_z)} - \frac{-\sum_i (T_y)_i}{\sum_i m_i (\ddot{z}_i + g_z)} \quad (24)$$

in

$$y_{ZMP} = \frac{\sum_i m_i (\ddot{z}_i + g_z) y_i - \sum_i m_i \left( \ddot{y}_i + g_y + \frac{v^2}{R_i} \right) z_i}{\sum_i m_i (\ddot{z}_i + g_z)} - \frac{-\sum_i (T_x)_i}{\sum_i m_i (\ddot{z}_i + g_z)} \quad (25)$$

Z uporabo Newton-Euler-jeve formulacije za izračun dinamičnih enačb sistema se ZMP izračuna neposredno iz gornjih enačb. Najprej izračunamo celotno gibanje robota na podlagi zahtevanega gibanja sklepov. Z upoštevanjem tako dobljenih hitrosti in pospeškov vseh segmentov začnemo z izračunavanjem sil in navorov z vrha proti prijemališču robota. Za zadnji segment veljata naslednji dinamični enačbi

$$\begin{aligned} F_{i-1,i} - m_i (\ddot{r}_i) &= 0, \\ M_{i-1,i} + r_{i-1,ci} \times F_{i-1,i} - [I_i \dot{\omega}_i + \omega_i \times I_i \omega_i] &= 0. \end{aligned} \quad (26)$$

Sedaj lahko izračunavamo sile in navore, ki delujejo v sklepih, dokler ne pridemo nazaj do prvega segmenta, kjer veljajo naslednje enačbe

$$\begin{aligned} F_{0,l} &= F_{1,2} + m_l (\ddot{r}_l + g + a_{rl}) = \sum_i m_i (\ddot{r}_i + g + a_{ri}) \\ M_{0,l} &= M_{1,2} + r_l \times m_l (\ddot{r}_l + g + a_{rl}) + [I_l \dot{\omega}_l + \omega_l \times I_l \omega_l] \\ M_{0,l} &= \sum_i [r_i \times m_i (\ddot{r}_i + g + a_{ri}) + T_i] \end{aligned} \quad (27)$$

Enačbo navorov lahko sedaj zapišemo v naslednji obliki

$$M_p = \sum_i [r_i \times m_i (\ddot{r}_i + g + a_{ri}) + T_i] - p \times \sum_i m_i (\ddot{r}_i + g + a_{ri}). \quad (28)$$

Z združitvijo enačb (23) in (28) dobimo

$$M_p = M_{0,l} - p \times F_{0,l}. \quad (29)$$

Ob upoštevanju pogoja, da navor deluje zgolj okrog osi  $z$ , ki je pravokotna na podlago, se ZMP izračuna na naslednji način

$$x_{ZMP} = \frac{(M_{0,l})_y}{(F_{0,l})_z}, y_{ZMP} = \frac{(M_{0,l})_x}{(F_{0,l})_z}. \quad (30)$$

Na ta način izračunamo tako podatke o silah, ki delujejo na podlago, kot tudi ZMP in posredno dobimo tudi izračun porazdelitve sil na posamezni nogi.

### *Ocena hitrosti*

Meritev hitrosti mobilnega robota v nestrukturiranem okolju je pogosto problematična. V primeru vozil s kolesi se pogosto uporablja odometrija, ki meri hitrost preko merjenja obodne hitrosti koles z znanim radijem. Človek hitrosti ne meri neposredno. Oceno hitrosti pridobi na podlagi vizualnih informacij. Najbolje to opiše optični tok. Vendar se v tem primeru pridobi zgolj relativno hitrost med hitrostjo gibanja in velikostjo predmeta, kateremu se približujemo. Smučar v večini primerov pozna velikosti objektov, katerim se približuje, in na podlagi tega oceni tudi absolutno hitrost gibanja. Tudi v našem primeru bi lahko uporabili tak način ocenjevanja hitrosti, vendar robot prepogosto ne vidi naslednjih vratc. Zato smo merili hitrosti s pomočjo GPS sprejemnika in pospeškov, zaznanih s pomočjo inklinometra. Ker se podatki iz GPS obnavljajo s frekvenco 4 Hz, podatki iz inklinometra pa s frekvenco 100 Hz, je potrebno zgraditi estimator hitrosti v vmesnih trenutkih, ko ni podatkov iz GPS-a, saj se hitrost v tem času lahko občutno spremeni. Zgradili smo tudi model spreminjanja hitrosti v odvisnosti od lokalnega naklona v sagitalni ravnini. Hitrost ocenjujemo na podlagi izmerjenih pospeškov v sagitalni ravnini. Vemo tudi, da ima merjena hitrost, pridobljena iz GPS-a, varianco 0,1 m/s, začetna vrednost hitrosti pa je enaka nič. Na podlagi tega smo zgradili Kalmanov filter, ki ocenjuje hitrost robota. Pri tem smo uporabili algoritem, ki je podrobneje opisan v 4. poglavju. V tem primeru med ocenjevanjem lege robota implicitno računamo tudi hitrost robota. Ocena hitrosti je zelo pomembna, saj ima neposredni vpliv na zagotavljanje stabilnosti smučarja.

### *Ocena naklona podlage*

Podatek o naklonu podlage se najpogosteje uporablja pri robotskih vozilih, ki se gibljejo po neznanih področjih. Park [7] je predstavil senzor, ki je deloval na podlagi strukturirane svetlobe. Na podlagi spremembe vzorcev je ugotavljal lokalni naklon podlage v območju pred robotom. Drug način zaznave okolice je uporaba laserskih daljinomerov [52]. V večini primerov pa se lokalni naklon

podlage oceni na podlagi senzorjev orientacije. V našem primeru smo uporabili inklinometer, vgrajen na platformo robota, ki omogoča izračun trenutnega lokalnega naklona podlage. Če sta obe nogi v stiku s podlago, je lokalni naklon površine v lateralni ravnini enak izmerjeni absolutni vrednosti naklona v tej ravnini

$$\psi = \psi_s - \theta , \quad (31)$$

kjer  $\psi$  predstavlja lokalni naklon podlage v stranski ravnini in  $\psi_s$  absolutni naklon ploščadi robota. Na podlagi merjenega in računanege naklona robota na podlago je razvidno, da se ob večji obremenitvi ene izmed nog poveča naklon v smeri bolj obremenjene noge, kot je bilo predstavljeno v predhodnem poglavju. Zato ocena trenutnega naklona na podlago  $\psi$  po enačbi (31) vsebuje napako, ki je posledica merjenega naklona robota  $\theta$ .

### 3.5 Zagotavljanje stabilnosti

Zagotavljanje stabilnosti mobilnih robotov v dinamičnih razmerah je dokaj neraziskano področje. Večina raziskav se je osredotočila na določitev stabilnih pozicij delovnih strojev v statičnih razmerah. Za te stroje se predvideva, da niso sposobni izvajati hitrih gibov, tako da so dinamične komponente gibanja v primerjavi z gravitacijo zanemarljive. Velik del raziskav na področju humanoidne robotike pa se je usmeril na področje hoje. V tem primeru so upoštewane tudi dinamične komponente, ki vplivajo na stabilnost, ob tem pa večina modelov hodi po ravnih površinah in že majhne spremembe naklona površine lahko povzročijo padec robota.

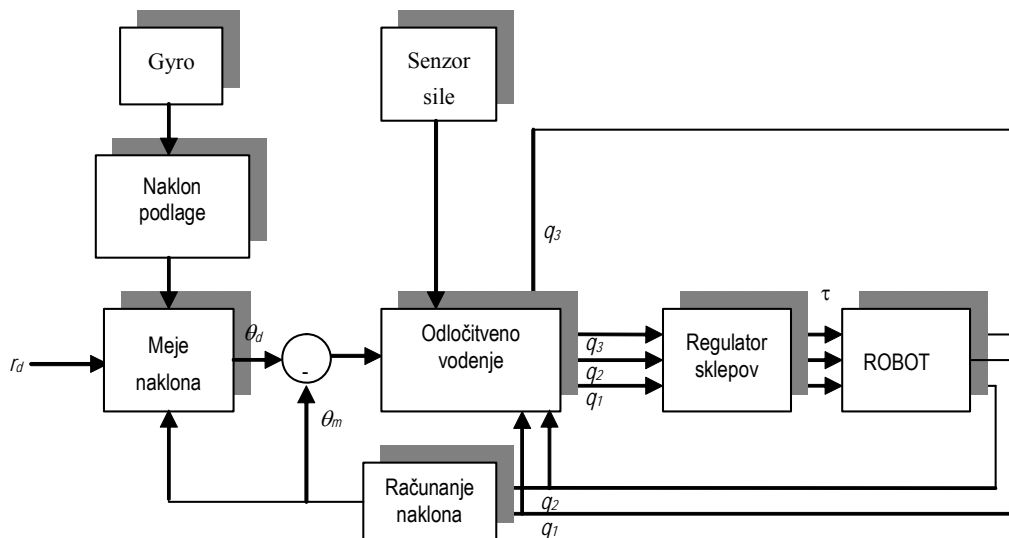
Za zagotavljanje ravnotežja sta se uveljavila dva pristopa, ki delujeta na podlagi dinamičnih modelov mehanizma. Prvi pristop temelji na izvajanju vnaprej pripravljenih stabilnih trajektorij, kar pa pri smučarskem robotu ni mogoče, ker se okolje med gibanjem preveč spreminja. Drugi pristop pa je gradnja gibanja na podlagi trenutnih senzorskih informacij. Prvi pristop vnaprej pripravi gibanje posameznih sklepov, ki se prenese na realni mehanizem z zgolj minimalnimi spremembami. Tipičen primer tega so posnetki gibanja človeka, ki se z rahlimi spremembami prenesejo na humanoidnega robota [44]. Pri tem se pojavi tudi problem, da je težko natančno zajeti gibanje človeških sklepov. Ob prenosu gibanja

na robota pa je potrebno upoštevati še različno kinematiko (dolžine segmentov, sklepi ...) in dinamične značilnosti (težišče, vztrajnosti ...) robota in človeka [45].

V drugi kategoriji pa se gibanje gradi v realnem času na podlagi senzorskih informacij. Krmiljenje in planiranje sta v tej kategoriji združena. Na podlagi modela se izračuna trenutna stabilnost robota. Ob tem se ocenjuje, kakšno gibanje samega robota v trenutnih razmerah še zagotavlja dinamično ravnotežje. Glede na želeno gibanje se potem izračuna gibanje, ki bo čim bolj sledilo zelenemu in ob tem ohranjalo dinamično ravnotežje. Pri hoji humanoidnih robotov je pogosto uporabljena rešitev, da se gibanje razdeli v dva dela, in sicer v gibanje spodnjih okončin - nog in gibanje ostalega telesa, ki zagotavlja ravnotežje [49]. Ta rešitev je možna zgolj v primeru, ko se gibanje izvaja v strukturiranem okolju. Če se naklon podlage spremeni za kot večji, od meje pri kateri se zgolj s trupom ne more zagotoviti ravnotežja, ta metoda odpove. V strogo predvidljivih razmerah, kot je na primer vožnja manipulatorja na platformi po ravni podlagi, se lahko stabilnost zagotavlja tudi s pomočjo vodenja kinematično redundantnih stopenj mehanizma [50]. Na podlagi indeksa stabilnosti mobilna platforma zagotavlja izvajanje naloge, sočasno pa se optimizira še indeks stabilnosti, ki je definiran na podlagi ZMP-ja. Večinoma pa se v primeru zagotavljanja stabilnosti med hojo in tudi vožnjo mobilnih robotov uporablja model inverznega nihala [51]. Podoben primer poenostavitve je tudi uporaba modela »cart-table«, ki ga je predstavil Kajita [53]. V tem primeru se celotnega robota predstavi kot matematično nihalo z vso maso v trupu, kar omogoči zagotovitev ravnotežnih položajev z lastnim gibanjem telesa. V dinamičnem okolju pa predlagane metode ne zagotavljajo ohranjanja ravnotežja. Pri smučanju in spuščanju na rokah se spreminjata tako lokalni naklon kot hitrost, ki je veliko višja kot pri hoji, in zato se v zavojih pojavijo velike radialne sile. Za primer robota smučarja je bilo potrebno zgraditi vodenje, ki zagotavlja, da bosta obe smučki obdržali kontakt s podlago. Ta pogoj je izpolnjen, če se točka ZMP nahaja znotraj podporne ploskve smučiči oziroma rolk. Podporna ploskev je definirana z območjem med smučmi in tvori v celotnem času pravokotnik, katerega širina se spreminja v odvisnosti od naklona robota na podlago, dolžina le-tega je določena z dolžino smučk  $L$ , kar prikazuje slika 2.3. V tem območju se mora vedno nahajati ZMP. Da bi to dosegli, smo predlagali hierarhično vodenje, ki ga predstavljamo v nadaljevanju.

### 3.6 Odločitveno vodenje

Smučar mora med smučanjem opravljati tri naloge, in sicer ohranjati ravnotežje, izvajati zavoje in ob tem še amortizirati neravnine. V tem delu se bomo osredotočili zgolj na zagotavljanje stabilnosti in reagiranju na neravnine, medtem ko sta želen radij zavoja in razdalja  $l_c$  odvisna od navigacijskega modula, ki bo predstavljen v naslednjem poglavju. Stabilnost smučarja se zagotavlja v realnem času z gibanjem nog in trupa. V primerih, ko samo s trupom ni mogoče zagotoviti stabilne lege, je potrebno to izvesti tudi z nogami. To povzroči, da ne zagotovimo predpisanega radija zavoja, ki ga predpisuje navigacijski blok, vendar pa s tem ohranimo stabilnost. Za ta namen smo razvili hierarhično odločitveno vodenje. Da bi zagotovili stabilno smučanje tudi v primeru nenadnih sunkov, ki so posledica neravnin na terenu, je potrebno zagotoviti, da se točka ZMP v vsakem trenutku nahaja znotraj podpornih ploskev smuči. V ta namen se v našem algoritmu najprej izračuna ZMP za primer, ko je pospešek naklona med podlago in trupom enak 0. Nato se ob upoštevanju maksimalnih dovoljenih pospeškov trupa izračuna, kolikšen vpliv na ZMP ima v danem trenutku lahko gibanje trupa. Na podlagi tega izračunamo, s kakšnimi pospeški se lahko spreminja naklon robota na podlago, da bo ohranil ZMP znotraj podpornega poligona. V našem primeru je dinamični model odvisen tako od hitrosti smučanja v sagitalni ravnini, lokalnega naklona podlage in naklona smuči na podlago. Na prvi dve spremenljivki nimamo neposrednega vpliva, saj sta ti posledica terena, po katerem robot smuča. Tako lahko na stabilnost vplivamo zgolj z naklonom smučarja na podlago in z gibanjem trupa.



Slika 3.6: Shema zagotavljanja stabilnosti

Na podlagi znane hitrosti in lokalnega naklona površine izračunamo naklon (meje naklona na sliki 3.6), ki med zavojem zagotovi, da se sile razdelijo na obe nogi. Tipična razdelitev sil med smučanjem v zavoju je pri človeku v razmerju 2:1. Porazdelitev sil med nogama v odvisnosti od ZMP podajata enačbi

$$F_l = \frac{F}{2} \left( 1 - \frac{y_{ZMP}}{b_{sr}} \right) \quad (32)$$

in

$$F_d = \frac{F}{2} \left( 1 + \frac{y_{ZMP}}{b_{sr}} \right), \quad (33)$$

kjer je  $F$  celotna sila, ki deluje pravokotno na smučki.  $F_l$  in  $F_d$  sta sili, s katerima robot deluje na levo in desno smučko,  $y_{ZMP}$  je točka ZMP robota,  $b_{sr}$  pa je polovica razdalje med smučkama, kot prikazuje slika 3.4.

Na začetku in koncu zavoja pa je razmerje med silama na posamezni nogi še večje. V tem primeru se razen radialne sile in gravitacije pojavi še vpliv dinamičnih sil zaradi sprememb naklona na podlago. V teh trenutkih lahko razmerje med silama na nogah pri človeku doseže tudi 3:1. Ta kriterij smo uporabili pri nastavitvi dovoljenih mej gibanja ZMP-ja, kar v primeru našega robota pomeni



$$y_{ZMP}^- = -\frac{d}{3\cos\theta} + y_{MSP} \leq y_{ZMP} \leq \frac{d}{3\cos\theta} + y_{MSP} = y_{ZMP}^+, \quad (34)$$

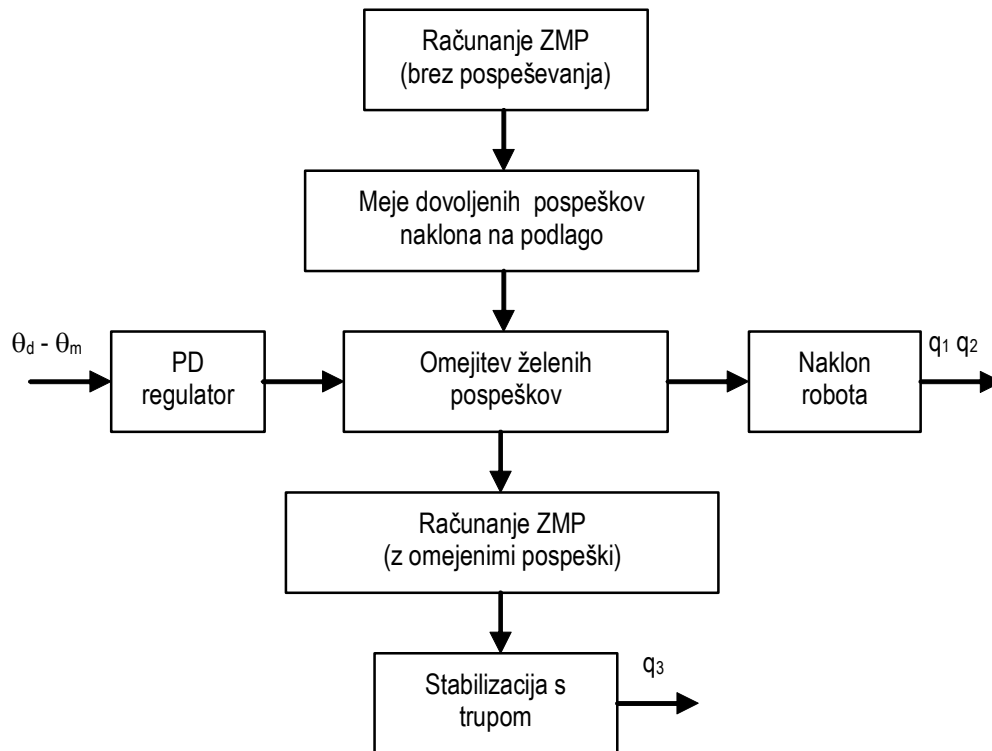
kjer je  $d$  razdalja med nogama v kolku. Vpliv pospeševanja trupa na spremembo sile na podlago izračunamo za maksimalne pospeške, ki jih je trup sposoben izvesti. Na tej podlagi izračunamo mejne pospeške naklona na podlago, ki še zagotavljajo ravnotežno gibanje. V vodenju omejimo  $\ddot{\theta}$ , in sicer v pozitivno smer

$$\ddot{\theta}^+ = -\frac{M_{10s} + y_{ZMP}^- F_{10s}}{J_{10} + y_{ZMP}^- b} \quad (35)$$

in negativno

$$\ddot{\theta}^- = -\frac{M_{10s} + y_{ZMP}^+ F_{10s}}{J_{10} + y_{ZMP}^+ b}, \quad (36)$$

kjer se  $M_{10s}$  in  $F_{10s}$  nanaša na navor in silo v statičnih razmerah ob upoštevanju gibanja maksimalnega gibanja trupa. Pri tem  $J_{10}$  predstavlja vztrajnostni moment robota okrog osi  $x$ , ki se nahaja na sredini med smučkama.



Slika 3.7: Odločitveno vodenje

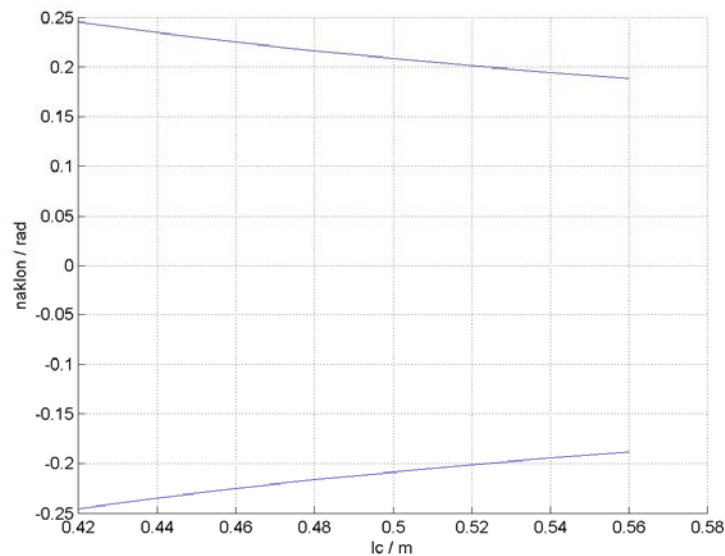
Koeficient  $b$  opisuje zvezo med  $\ddot{\theta}$  in povzročeno silo v smeri osi  $z$ . Tako izračunani maksimalni in minimalni pospeški zagotavljajo ohranjanje ZMP znotraj podporne ploskve med smučkama. S tem se zagotovi, da ne bodo nikoli preseženi nakloni, ki še omogočajo stabilno lego smučarja. Na podlagi modela se izračuna delitev neposrednega razmerja sil med posamezni nogi. V vodenju je potrebno upoštevati še dinamične vplive, ki povzročijo dokaj velike sile. Premik med različnimi zavoji se dogaja zelo hitro, in sicer v manj kot sekundi, kar pomeni, da so potrebni veliki kotni pospeški in s tem tudi dinamične komponente, ki vplivajo na stabilnost.

### 3.7 Analiza statične stabilnosti v odvisnosti od hitrosti in naklona

Človek med smučanjem ohranja stabilnost v zavoju s tem, da razporedi sile, s katerimi deluje na smučki, v območju razmerij med 2:1 do 1:2. Da bi zagotovili takšno porazdelitev sil pri robotu, je potrebno na podlagi zaznane hitrosti in lokalnega naklona podlage izračunati dovoljene naklone robota. V tem delu bomo opisali, kako se dovoljene meje naklonov spreminjajo, odvisno od trenutne

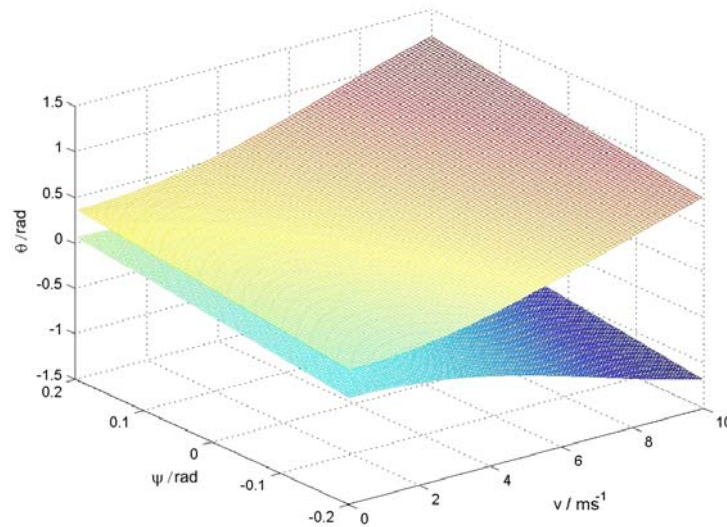
konfiguracije robota, hitrosti in lokalnega naklona površine. Medtem ko konfiguracija robota vpliva na težišče, pa hitrost vpliva samo na radialno silo, lokalni naklon površine pa vpliva na komponento gravitacije. To ponazarjata diagrama na slikah 3.8, 3.9. Pri tem smo predpostavili, da je hitrost gibanja v sagitalni ravnini enaka 2 m/s. V prvem diagramu je predstavljena odvisnost stabilnih naklonov glede na višino smučanja  $l_c$ .

Iz slike 3.8 je razvidno, da zmore robot večje naklone, če je težišče nižje. Vendar pa z nizkim težiščem onemogočimo, da bi robot lahko izvedel večje naklone na podlago, saj se nogi ne skrčita povsem, ampak imata omejitvev, kot je opisana v 5. poglavju.



Slika 3.8: Področje stabilnih naklonov v odvisnosti od višine smučanja

Izračunali smo, za koliko lahko naklon trupa spremeni lokacijo ZMP-ja. V našem primeru je ta vpliv zelo omejen, saj zaradi majhne mase trupa, tudi premik v obe skrajni legi spremeni pozicijo ZMP-ja zgolj za 0.04 m.



Slika 3.9: Meje stabilnih naklonov v odvisnosti od naklona podlage in hitrosti

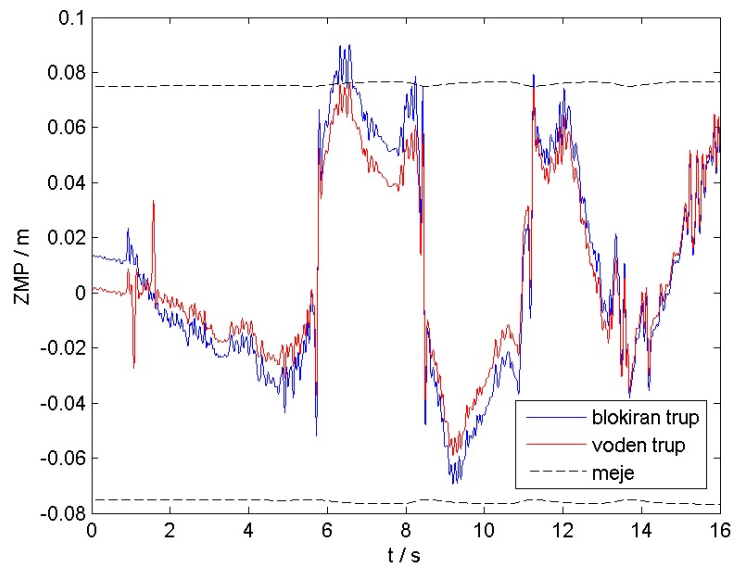
Na sliki 3.9 sta predstavljeni omejitvi dovoljenih naklonov v odvisnosti od hitrosti in naklona podlage. Že v mirovanju se lahko robot nakloni v območju 0.3 radiana. Z večanjem hitrosti pa se to območje povečuje in pri hitrosti 10 m/s se to območje poveča na skoraj 2 radiana. Lokalni naklon podlage ima pri spremembi dovoljenega naklona vpliv na premik tega območja v smeri večjih naklonov robota, nasprotnih lokalnemu naklonu podlage.

Konfiguracija robota vpliva tudi na radialno silo, vendar je vpliv le-te v primerjavi z ostalimi zanemarljiv in smo ga iz analize izpustili.

### 3.8 Stabilizacija s trupom

Z gibanjem trupa lahko stabilnost povečamo na dva načina. Prvi je s spreminjanjem težišča. Kot je razvidno iz slike 3.9, je področje stabilnih naklonov pri nizkih hitrostih zelo ozko. S spremembo težišča trupa lahko to območje občutno povečamo. Ker se naklon pri nizkih hitrostih ne spreminja sunkovito, je tudi kompenzacija stabilnosti s trupom lahko počasnejša. Testi na modelu in realnem sistemu so pokazali, da je smiselno pri nizkih hitrostih kompenzirati s trupom predvsem gravitacijo. S tem lahko dosežemo večje naklone na podlago. Pri hitrostih, manjših od 2 m/s, se s trupom skrbi zgolj za statično stabilnost.

Med izvajanjem zareznega zavoja pri večjih hitrostih (4 m/s ali več) se pojavijo velike vztrajnostne sile v trenutku zarezu smučke v sneg, kar seveda povzroči spremembo ZMP-ja. To silo kompenziramo z gibanjem trupa. Na ta način reagiramo na trenutne spremembe, ki so lahko posledica velike hitrosti in posledično hitre spremembe naklona podlage in radialne sile. V teh primerih se s hitrimi gibi na podlagi vztrajnostnega momenta trup odzove na te spremembe.



Slika 3.10: Stabilizacija s trupom

Črtkani črti na sliki 3.10 prikazujeta meje, ki zagotavljajo, da se sili porazdelita med nogi znotraj želenega razmerja. Če trupa ne premikamo, ZMP preseže želeno vrednost. S pomočjo gibanja trupa pa ZMP ostane znotraj mej.

## 4 Navigacija med vratci

### 4.1 Uvod

Iskanje poti in izogibanje oviram sta dva izmed glavnih izzivov vodenja mobilnih robotov v realnem okolju. Pri iskanju poti z neholonomnimi mobilnimi roboti se planiranje poti najpogosteje izvede vnaprej na podlagi globalnih informacij. Pogosto pa se pojavijo problemi iskanja optimalne poti med ovirami do cilja. V teh primerih zagotovimo idealno pot kot kombinacijo ravnih odsekov in izsekov krožnic okrog ovir, katerih radij je enak najmanjšemu možnemu radiju, ki ga je sposoben izvesti neholonomni sistem, oziroma najmanjšemu možnemu radiju okrog ovire, ki ga pogojuje okolje. Pri tem se pojavi problem, da je povezava med ravnim odsekom in krožnico nezvezna. Zato bi se moral neholonomni robot v točki povezave med ravnim in krožnim odsekom ustaviti oziroma v trenutku spremeniti radij zavoja iz neskončnega v določen radij, kar pa zaradi dinamike realnih mehanizmov ni možno. Pri poskusih na realnih sistemih so se zaradi vpliva dinamike pojavila odstopanja med želenimi in dejanskimi potmi. Baltes [8] je predstavil metodo, ki omogoča popraviljanje teh trajektorij v realnem času. Iskanje in izvajanje gibanja med ovirami lahko izvedemo na podlagi globalnih ali lokalnih metod. Globalne metode so bolj računsko zahtevne in se večinoma ne morejo izvajati v realnem času. Zaradi tega se tudi ne morejo uporabljati v dinamičnih okoljih. Lokalne metode pa za svoje izvajanje potrebujejo zgolj informacijo o bližnji okolici robota in določajo le trenutno najoptimalnejše gibanje. Zaradi tega so računsko manj zahtevne, kar omogoča uporabo v realnem času. Slabost lokalnih metod pa je, da ne zagotavljajo globalno optimalne rešitve.

Okolje zaznavamo z različnimi senzorji, od laserskih daljinomerov, ultrazvočnih senzorjev in kamer. V splošnem lahko pri načrtovanju poti in izogibanju oviram uporabljamo sprotne senzorske informacije samo z lokalnimi metodami. Iz literature je znanih več metod, ki jih bomo na kratko predstavili v nadaljevanju.

### *Mreže določenosti*

Metoda, ki jo je predlagal Moravec [56], zgradi zemljevid v okolici robota na podlagi verjetnosti porazdelitve ovir v prostoru. S tem se zavaruje pred nenatančnostjo senzorskih informacij. Robot na podlagi senzorskih informacij zgradi dvodimenzionalno mrežo, ki predstavlja robotsko okolico. Vsak element mreže vsebuje tudi verjetnost, da se v določenem polju nahaja ovira. Vrednost vsakega polja se obnavlja v vsakem trenutku na podlagi senzorskih informacij s pomočjo verjetnostne metode, ki upošteva karakteristiko in interval zaupanja teh informacij. Ta metoda lahko upošteva samo statične ovire, saj bi v dinamičnem okolju po določenem času zasedli vsa polja mreže z ovirami, tako da ne bi ostalo nobenih prostih področij, kamor bi se lahko robot premaknil.

### *Metoda potencialnih polj*

Ena izmed najpogosteje uporabljenih metod za lokalno navigacijo in izogibanje oviram temelji na umetnih potencialnih poljih, ki jih je uvedel Khatib [18]. Metoda je uporabna tako za mobilne robote kot tudi manipulatorje. Pri tem zgradimo ločena potencialna polja za odboj in privlačenje, ki definirajo odnos med robotom in zaznanimi objekti. Ta polja se nato združijo v enotno polje. Na podlagi gradienta tega polja dobimo gladko trajektorijo. Ovire so objekti, ki odbijajo robota stran od le-teh, medtem ko cilji privlačijo robota. Navigacija je zgrajena tako, da se robot giblje v smeri zmanjševanja potenciala. Ta metoda predpostavlja, da je oblika ovir v okolju znana. Originalna metoda je predpostavljala, da je okolje v času izvajanja naloge statično, kasneje pa so jo posplošili tudi za dinamična okolja. Potencialna polja zgradimo na podlagi zaznanih ovir in ciljev. Ta metoda deluje zgolj za en cilj. Če naloga zahteva večje število ciljev, potem najprej izberemo cilj, ki mora biti prvi izpolnjen. Ob izvršitvi prve naloge izberemo naslednji cilj in tako dalje do končnega cilja. Potencialno polje izračunamo v vsaki točki trajektorije, robot pa se giblje v smeri manjšanja potenciala, dokler ne doseže lokalnega ali globalnega minimuma. Čeprav je ta metoda ena najpogosteje uporabljenih, pa ima tudi slabosti. V analizi [54] so se kot glavne slabosti te metode izkazale:

- Lokalni minimum lahko povzroči, da se robot ujame v past in ne najde več izhoda. Na past najpogosteje naletimo v primeru ovire v obliki črke U.

- Nezmožnost prehoda med dvema bližnjima ovirama, ker se zaradi velike odbojne sile ovir izniči vpliv cilja.
- Oscilacije v primeru ozkih koridorjev.

Dodatno se pojavi še problem, ki je pogosto spregledan. Pojavi se v primeru, ko je odbojna sila večja od sile privlačenja cilja, kar povzroči gibanje robota stran od cilja in ne proti njemu [55]. Ob upoštevanju teh problemov so bile razvite različne metode, ki bi reševale te probleme.

#### *Navidezno polje sil - VFF*

Na podlagi kombinacije mrež določenosti in virtualnih potencialnih polj je Borenstein [14] predstavil metodo navideznega polja sil (Virtual Force Field - VFF). Navidezno okno določene velikosti se premika po mreži. Vsa polja, ki jih aktivno okno pokriva, se upoštevajo kot aktivna. Vsa ta polja prispevajo k sili na podoben način kot pri potencialnih poljih. Amplituda sile posameznega polja je proporcionalna meri zaupanja, da je v tem polju ovira, in obratno sorazmerna s kvadratom razdalje polja od robota. Vse aktivne navidezne sile so seštete in določijo smer gibanja robota, da se bo izognil oviram. Ker metoda temelji na potencialnih poljih, prevzame tudi slabosti le-teh. Prednost je zgolj, da upošteva negotovost senzorskih informacij. Ne glede na verjetnostni značaj metode pa ohranja dokaj hiter odziv na nove zaznane ovire.

#### *Histogram polja vektorjev VFH*

Iz VFF je bila kasneje razvita še metoda histograma polja vektorjev. Ta metoda podatke reducira v dveh stopnjah za razliko od VFF. Pri tej metodi se na podlagi podatkov sonarja zgradi diskretno enodimenzionalno polarno funkcijo gostote ovir. Pot poiščemo na podlagi minimuma prej navedene funkcije in je v vsakem trenutku usmerjena k tej točki. V naslednji nadgradnji metode VFH+ [57] so v metodo vgrajeni še podatki o povečevanju ovir, histereza, ki preprečuje oscilacijo med različnimi dolinami v histogramu, kar zagotavlja bolj gladko trajektorijo. Tretja verzija VFH\* poleg predhodnih izboljšav vsebuje vnaprejšnje (ang. »Look ahead«) preverjanje za zagotovitev, da se robot ne bi zapeljal v "past".



## *Metode na osnovi oponašanja živali*

V robotiki se pogosto uporablja algoritme, ki so nastali na osnovi opazovanja živali. Posebej pogosto se uporabljajo metode, ki oponašajo gibanje žuželk, predvsem mravelj. Večina teh algoritmov temelji na osnovi, ki sta jo predlagala Lumelsky in Stepanov [58]. Gibanje robota oziroma žuželk je sestavljeno iz dveh delov, in sicer direktnega gibanja proti cilju in izogibanja oviram. V primerih, ko pride do kontakta z oviro, se gibljemo ob oviri, vse dokler ne naletimo na črto, ki omogoča neposredno pot do cilja. V osnovi ta metoda deluje zgolj na osnovi senzorjev pozicije in senzorjev dotika. V kasnejših različicah pa so uporabili tudi druge senzorje. Te metode se trenutno zelo pogosto uporabljajo predvsem pri robotih za opravila, kot so na primer sesanje, košnja zelenic itd.

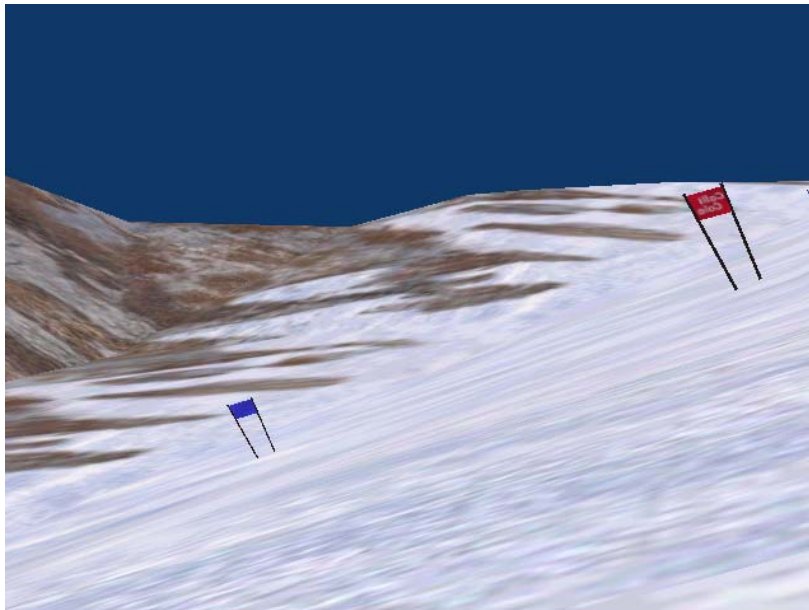
Omenimo naj, da so navigacijski algoritmi še vedno predmet številnih raziskav in se stalno spreminjajo ter dopolnjujejo. Tako se poskuša nadgraditi lokalne metode s pomočjo delnih globalnih informacij.

Zaznavanje ovir je v večini primerov izvedeno s pomočjo raznih radarjev in laserskih daljinomerov. Drug način pa je zaznavanje na podlagi vizualnih informacij, na podoben način kot to izvaja človek s pomočjo vida. Identifikacijo in lokalizacijo ovir v 3D prostoru lahko na podlagi vizualnih informacij pridobimo na podlagi stereo vida, vida gibanja in na podlagi modela opazovanih objektov. V našem primeru je bistvena predvsem lokalizacija ovir, ki je nujno potrebna za izvajanje poti med vratci. Ker je oblika ovir - vratc znana, smo lokalizacijo ovir izvedli na podlagi modela vratc.

### **4.2 Zaznava vratc**

Položaj vratc glede na smučarja smo v našem primeru določili z uporabo računalniškega vida. Vratca smo zaznavali na podlagi barve in oblike. Najprej se je potrebno odločiti, v kakšnem barvnem prostoru se bodo obdelovale slike. Najpogosteje se uporabljajo RGB prostori, vendar pa je ta barvni prostor zelo odvisen od osvetlitve. Na prostem je ta vpliv osvetlitve zelo izrazit. Ker so se poskusi izvajali v zunanjem okolju, je bilo potrebno vpliv osvetlitve zmanjšati. Da bi se izognili problemom v zvezi z osvetlitvijo, smo zajete slike pretvorili v HSV

(Hue – odtenek barve, Saturation – nasičenje, Value – vrednost) barvni prostor, ki predstavlja sliko na podoben način, kot jo zaznava človeško oko. Glede na definicijo naj bi osvetlitev vplivala zgolj na vrednost V. Sliko smo najprej upragovili v območju odtenkov in nasičenja. Tako so pridobljena področja, kjer obstaja možnost nahajanja vratc. V drugi fazi določimo, ali imajo ta področja obliko vratc. Na podlagi kalibrirane kamere, znane velikosti vratc in absolutne orientacije kamere izračunamo kot pogleda na vratca in trenutno oddaljenost le-teh. Slika je glede na položaj vratc zavrtena za določen kot, ki ga izračunamo na podlagi zasuka trupa in naklona ploščadi.



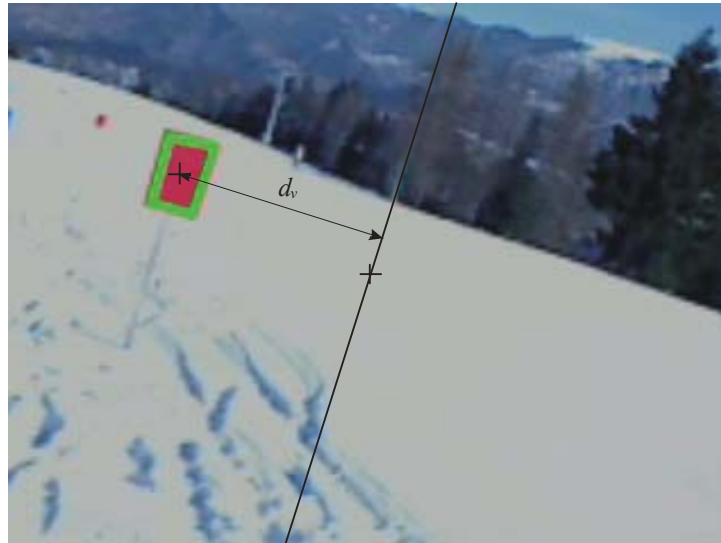
Slika 4.1: Pogled smučarja na vratca v navideznem okolju

Na sliki 4.1 je viden pogled smučarja na vratca v navideznem okolju. Na podlagi velikosti in kota pogleda je potrebno določiti razdaljo do vratc. Ob tem pa je potrebno iz slike pridobiti tudi kot pogleda na vratca. Pri tem se upošteva tudi naklon kamere.

Slika 4.2 prikazuje oddaljenost vratc od osi, ki poteka skozi optični center kamere in je zarotirana za kot naklona kamere glede na pravokotnico na podlago. Kot pogleda na oviro je določen na podlagi razdalje  $d_v$ , in sicer v naslednji obliki

$$\varphi - \varphi_o = a \tan(K_{zk} d_v), \quad (37)$$

kjer je  $K_{zk}$  koeficient preslikave zornega kota v slikovni element kamere in je odvisen od lastnosti le-te.



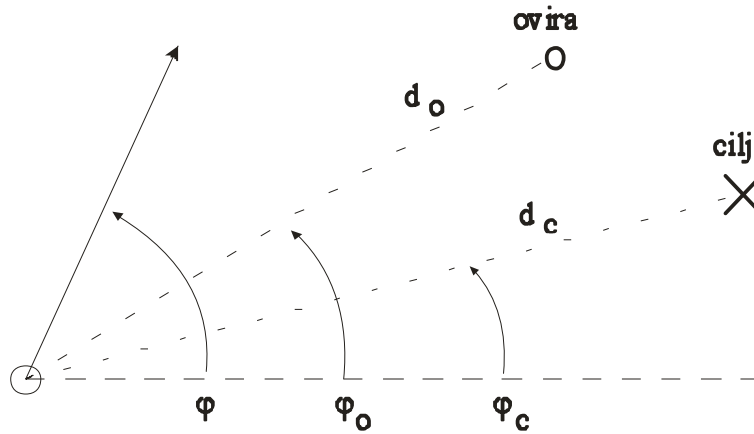
Slika 4.2: Določitev pozicije vratc

### 4.3 Vodenje smučarja med vratci

V tem delu se bomo osredotočili predvsem na realizacijo vodenja, ki omogoča smučanje med vratci na podlagi vizualnih informacij, pridobljenih s pomočjo kamere, locirane na smučarju. Celotno smučanje lahko razdelimo na dva dela. V prvem delu se po izhodu iz prejšnjega zavoja pripravimo na izvedbo naslednjega. Medtem ko se zavoj poskuša izvesti z najmanjšim možnim radijem, ki ga dopuščajo smučke, pa je potrebno izhodišče zavoja doseči s pravilnim kotom gibanja, pravo pozicijo in ob tem je nujno tudi, da se smučar že giblje s pravilno kotno hitrostjo, ki bo omogočila želen radij zavoja. V našem primeru se ta del izvede s pomočjo metode potencialnih polj, ki zagotovi, da se izhodišče zavoja prenese v pravilno pozicijo. Ker želimo, da je gibanje med smučanjem gladko in ne prihaja do velikih oscilacij, smo uporabili metodo, ki zagotavlja zvezne prehode gibanja smučarja.

#### 4.4 Model človeške navigacije

Fajen [7] je v svojem delu predlagal model človeške navigacije, ki ga je zgradil s pomočjo poskusov v navidezni resničnosti. Bistvo poskusa je bilo odzivanje ljudi na ovire, ob tem pa morajo doseči želeni cilj. Meritve so pokazale odvisnost med kotnim pospeškom ljudi in kotom pogleda na ovire in cilj ( $\varphi_o$ ,  $\varphi_c$ ) ter njihovih razdalj  $d_o$  in  $d_c$ .



Slika 4.3: Model gibanja v okolju z ovirami

Na osnovi rezultatov je zgradil model pospeškov, potrebnih za gibanje oseb v okolici ovir:

$$\ddot{\varphi} = -b\dot{\varphi} - k_g (\varphi - \varphi_c) (e^{-c_1 d_c} + c_2) + \sum_i k_{oi} (\varphi - \varphi_{oi}) (e^{-c_3 |\varphi - \varphi_{oi}|}) e^{-c_4 d_{oi}}, \quad (38)$$

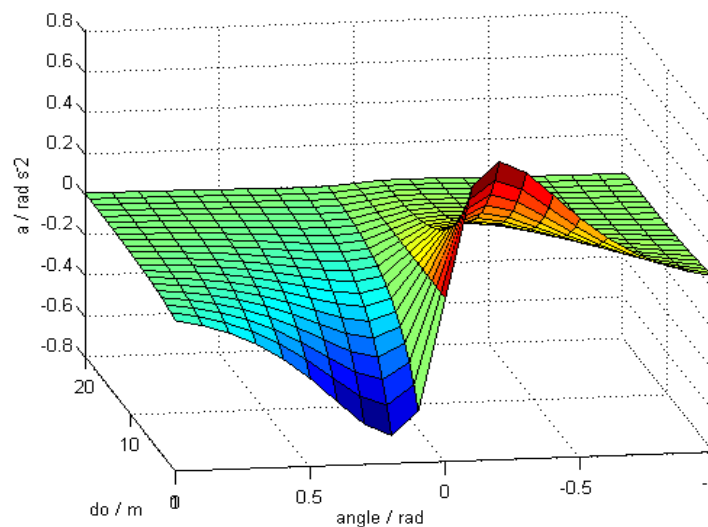
kjer  $\varphi$  predstavlja trenutni kot gibanja človeka v globalnem koordinatnem sistemu. Model je sestavljen iz treh delov:

- Dušenje, ki onemogoča spremembo kotnega pospeška sorazmerno s kotno hitrostjo gibanja osebk s konstanto dušenja  $b$ .
- Komponenta, ki privlači smer gibanja k cilju. Odvisna je od razlike med trenutno smerjo gibanja, kota pogleda na cilj in oddaljenosti od cilja. Konstanta  $c_2$  skrbi, da se tudi pri veliki oddaljenosti od cilja oseba usmerja proti cilju. Koeficient  $k_g$  pa je proporcionalno ojačanje komponente, ki privlači k cilju.

- Komponenta, ki odbija smer od bližnjih ovir. Vpliv je proporcionalen za majhne razlike  $\varphi - \varphi_{oi}$ , z večanjem pa začne eksponentno padati v odvisnosti od koeficienta  $c_3$ . Ravno tako se vpliv spreminja v odvisnosti od razdalje do ovire. Odvisnost eksponentnega padanja od razdalje do ovire je podana s konstanto  $c_4$ .

#### *Pospešek v odvisnosti od ovire*

Pospešek, ki odbija od ovire, je odvisen od oddaljenosti od ovire in razlike med trenutno smerjo gibanja in smerjo ovire. Pospešek je grafično predstavljen na sliki 4.4, kjer je normiran za hitrost gibanja smučarja 1 m/s, kajti pospeški se morajo prilagajati hitrosti, kar bo prikazano tudi v nadaljevanju. Nastavitev parametrov modela pospeškov je bila izvedena iterativno na podlagi poskusov.

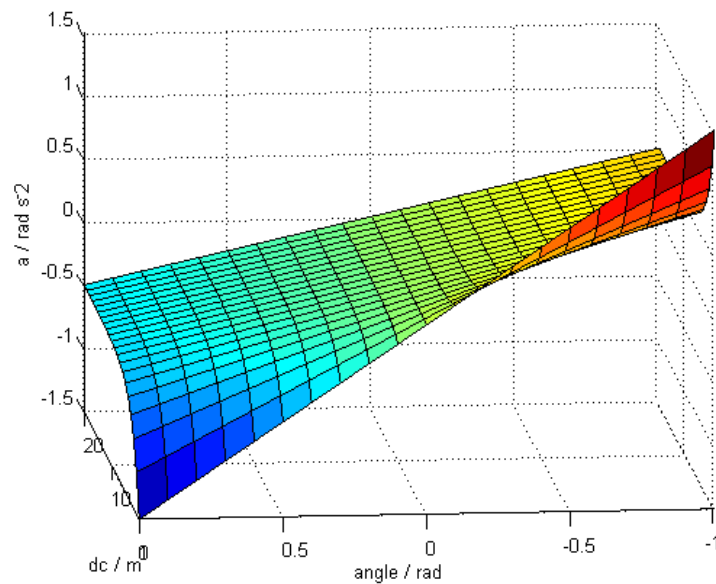


Slika 4.4: Pospešek v odvisnosti od ovire

Kot je razvidno s slike 4.4, največji pospeški delujejo v neposredni bližini ovire. Pospešek zmeraj deluje v smeri, ki odbija od ovire. V primeru, ko je trenutna smer gibanja usmerjena neposredno v oviro, pa ne deluje noben pospešek. Zato se v teh primerih zgodi, da se smer ne bo spreminjala in bo posledično prišlo do trčenja z oviro.

### *Pospešek v odvisnosti od cilja*

Medtem ko ovire odbijajo, pa cilji povzročajo pospeške, ki nas vodijo k cilju. Na sliki 4.5 so predstavljeni pospeški v odvisnosti od oddaljenosti od cilja in razlike med trenutno smerjo gibanja in kotom pogleda na cilj. Privlačni pospeški so največji v primeru neposredne bližine cilja.



Slika 4.5: Pospešek v odvisnosti od cilja

Pospeški, ki usmerjajo k cilju, pa delujejo tudi v veliki oddaljenosti od cilja. S tem se zagotovi, da se tudi v primeru zelo oddaljenih ciljev robot usmeri proti cilju.

### *Vpliv hitrosti*

Večina mobilnih robotov nadzoruje hitrost gibanja. V primeru izogibanja oviram so razvili različne algoritme za vodenje te hitrosti. V primeru smučanja pa nimamo neposrednega nadzora nad hitrostjo. Kotna hitrost smučarja je odvisna od hitrosti smučarja v sagitalni ravnini in radija zavoja  $R$ , ki ga izvaja. Izračunamo jo s pomočjo enačbe

$$\dot{\phi}(t) = \frac{v(t)}{R(t)}. \quad (39)$$

Na podlagi enačbe (39) se pri nizki hitrosti kot smučanja spreminja malo, ne glede na radij zavoja, ki ga smučar izvaja. Algoritem smo modificirali tako, da je pot, ki jo smučar izvede, neodvisna od hitrosti, s katero se vozi. V ta namen smo spremembo radija iz modela človeške navigacije pomnožili s hitrostjo, s katero se smučar giblje, kar bi vodilo k dejstvu, da se smučar vozi po enaki trajektoriji neodvisno od hitrosti, če se seveda ne pojavi omejitev zaradi zagotavljanja stabilnosti. Ob tem pa smo v algoritmu vgradili še varovalo, da se pri mirujočem naklonu robota ne spreminja, kar preprečuje poškodbo mehanizma.

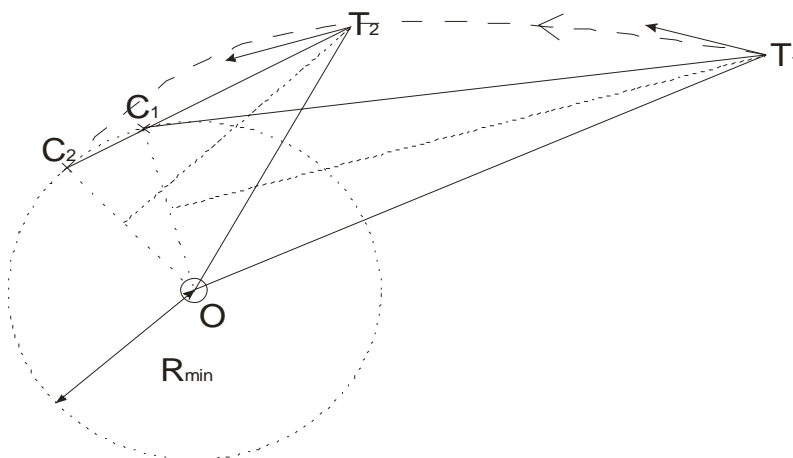
### *Planiranje poti*

Raziskave s pomočjo navidezne resničnosti so pokazale, da se človek v primeru gibanja med ovirami proti cilju giblje po določenih zakonitostih. V nadaljevanju bomo opisali novo metodo, ki omogoča iskanje poti v lokalnem prostoru in pri tem upošteva zgolj informacije o trenutnih zaznanih ovirah, na katere se sistem aktivno odziva. Vse do sedaj razvite metode so bile implementirane zgolj pri aktivno vodenih mehanizmih, ki omogočajo nadzor hitrosti gibanja. Smučar pa je za razliko od teh vozil pasiven mehanizem v sagitalni ravnini, saj nima neposrednega nadzora nad hitrostjo in zaradi tega mora svoje gibanje, ki vodi do izvajanja zavojev, prilagajati trenutni zaznani hitrosti. V našem primeru je smučanje sestavljeno iz dveh delov. Prvi del zajema pripravo na zavoj. Drugi del pa je dejanska izvedba samega zavoja z najmanjšim možnim radijem.

Na podlagi zaznanih vratc se v vsakem trenutku zgradijo potencialna polja okrog ovir in zelenih ciljev. Želena pot smučanja se generira sproti v odvisnosti od zaznane pozicije vratc in z njo povezane pozicije trenutnega cilja ter trenutne smeri gibanja, prikazane na sliki 4.6. Puščice iz točk T1 in T2 kažejo trenutno smer gibanja smučarja. V teh dveh točkah sta prikazani tudi smeri, ki kažeta proti oviri  $O$  in trenutnima ciljema  $T1$ ,  $T2$ . V T1 deluje zelo velika privlačna sila k cilju  $C_1$ , medtem ko je odbojna sila zaradi ovire – vratc majhna, ker so vratca še zelo oddaljena, pa tudi kot pogleda  $\phi_0$  je zelo velik. Predlagana metoda omogoča hitrejše prilagajanje v neposredni bližini ciljev in ovir. Zaradi tega se v primeru, ko se smučar približa radiju okrog ovire, smučar sam postavi v pozicijo, kjer izvaja

zavoj z najmanjšim možnim radijem.

V nadaljevanju, ko je smučar že zelo blizu oviri in že izvaja zavoj, preidemo v fazo izvajanja zavoja, kjer smučar zgolj drži najmanjši radij, dokler se v vidnem kotu kamere ne pojavijo naslednja vratca.



Slika 4.6: Generiranje cilja

Pri uporabi predlaganega modela planiranja poti naletimo na težavo, povezano z uporabo kamere. Pri majhni oddaljenosti od ovire nas sila cilja vodi stran od ovire in s tem se pojavi problem zaradi ozkega zornega kota kamere, tako se ovira izgubi iz vidnega področja. Da bi se izognili tem primerom, ko bi morali pozicijo vratc določati zgolj na osnovi odometrije, smo krožnico okrog ovire premaknili v smeri pravokotno glede na globalno smer smučanja.

V primeru vnaprej znane globalne smeri smučanja ali v primeru zaznave še naslednjih vratc se predlagani algoritem izboljša, saj lahko v tem primeru smučamo bolj neposredno na vratca. Za trajektorijo ni potrebno, da se nahaja v razdalji, večji od najmanjšega radija zavoja. Zato lahko vratca navidezno premaknemo pravokotno na globalno smer smučanja oziroma v smeri simetrane zavoja, če so vidna naslednja vratca. S tem premikom določimo, na kakšni razdalji od vratc se bo smučar peljal.



#### 4.5 Ocena pozicije robota glede na vratca

Pozicija vratc glede na robota je v določenem trenutku pridobljena iz podatkov kamere, ki določajo oddaljenost in kot pogleda na vratca. Pri tem se predvsem pri večjih razdaljah do vratc pojavijo veliki raztrosi meritev, saj je odvisnost med velikostjo vratc na sliki in razdaljo do njih v obratnem kvadratnem sorazmerju. Poleg tega pa pogosto te razdalje sploh ne moremo oceniti. To se na primer zgodi, ko vratca niso v vidnem polju kamere oz. se dotikajo roba vidnega področja kamere. V teh primerih bodisi ne moremo zaznati vratc bodisi ne moremo oceniti razdalje, saj ne moremo oceniti njihove velikosti.

Da bi zagotovili vztrajanje v želenih zavojih, je potrebno oceniti pozicijo vratc tudi v teh trenutkih. Na podlagi znane hitrosti in smeri gibanja robota lahko lego

$$p_{vi} = [d_{vi} \ \varphi_{vi}]^T \quad (40)$$

i-tih vratc glede na robota zapišemo

$$d_{vi} = \sqrt{d_{vi-1}^2 + (v\Delta t)^2 - 2d_{vi-1}v\Delta t\cos(\varphi - \varphi_{vi-1})}, \quad (41)$$

$$\varphi_{vi} = \varphi_{vi-1} + \text{sign}(\varphi_{vi-1} - \varphi) \arccos\left(\frac{d_{vi}^2 + d_{vi-1}^2 - (v\Delta t)^2}{2d_{vi}d_{vi-1}}\right), \quad (42)$$

kjer indeks  $_{-1}$  označuje vrednosti v prejšnjem trenutku,  $\Delta t$  čas med zajemom dveh slik,  $\varphi$  trenutno smer vožnje in  $\varphi_{vi}$  trenutno smer pogleda z robota na vratca.

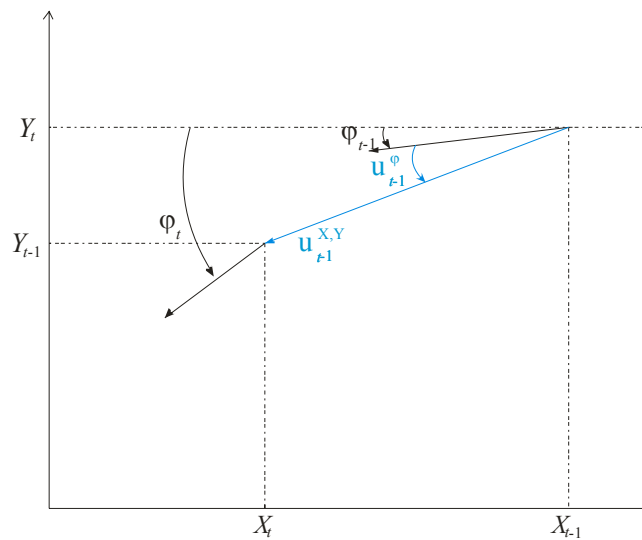
Ocena pozicije in orientacije robota je pogost problem v robotiki. Robot svojo pozicijo izračuna na podlagi merjenih oddaljenosti od znanih pozicij, merjenja kotov in lastnega gibanja. Na realnih sistemih pa vse meritve vsebujejo določeno napako in šum. Pri uporabi trigonometrije se pri izračunu pozicije pojavijo napake med posameznimi slikami. Zato se izmerjena pozicija lahko zelo spreminja in robot navidezno skače v okolici dejanske pozicije. Ena izmed najbolj uspešnih metod reševanja takih problemov je Kalmanov filter. Kalmanov filter omogoča oceno stanja na podlagi meritev in modela sistema. Ob tem pa upošteva, da podatki vsebujejo šum in je njihov vpliv na oceno odvisen od stopnje zanesljivosti le-teh.

### Razširjen Kalmanov filter

Za oceno oziroma estimacijo stanja se tudi v robotiki pogosto uporabljajo Kalmanovi filtri. Ti v svoji osnovni izvedbi omogočajo uporabo zgolj pri linearnih procesih. Za ocenjevanje stanja nelinearnih sistemov pa se uporablja razširjen Kalmanov filter (ang. Extended Kalman filter - EKF). Algoritem filtra je sestavljen iz dveh delov, in sicer iz predikcije in posodobitve. Filter se inicializira v določeni poziciji, kar sta v našem primeru znana oddaljenost in kot pogleda na oviro, z znano negotovostjo le-te. Posodobitev podatkov temelji na podlagi dveh funkcij  $f()$  in  $h()$ . Funkcija  $f()$  se nanaša na predhodno pozicijo  $p_{t-1}$ , podatke o gibanju  $u_{t-1}$  in šumu v sistemu  $w_{t-1}$

$$p_t = f(p_{t-1}, u_{t-1}, w_{t-1}) \quad (43)$$

kot je razvidno s slike 4.6.



Slika 4.7: Izračun nove pozicije na podlagi odometrije

## Predikcija

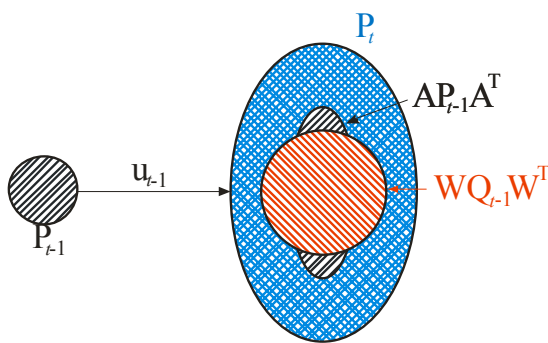
V tem koraku je izračunana pozicija  $p_t^- = [X_t^- \ Y_t^- \ \varphi_t^-]^T$  na podlagi podatkov o prejšnjem stanju in odometrije, kot jo prikazuje slika 4.6.

$$p_t^- = \begin{bmatrix} X_t \\ Y_t \\ \varphi_t \end{bmatrix} = f(p_{t-1}^-, u_{t-1}^-, w_{t-1}^-) = \begin{bmatrix} X_{t-1} \\ Y_{t-1} \\ \varphi_{t-1} \end{bmatrix} + \begin{bmatrix} (u_{t-1}^X + w_{t-1}^X) \cos \varphi_{t-1} - (u_{t-1}^Y + w_{t-1}^Y) \sin \varphi_{t-1} \\ (u_{t-1}^X + w_{t-1}^X) \sin \varphi_{t-1} + (u_{t-1}^Y + w_{t-1}^Y) \cos \varphi_{t-1} \\ u_{t-1}^\varphi + w_{t-1}^\varphi \end{bmatrix} \quad (44)$$

Ob tem izračunamo tudi kovariančno matriko napake  $P_t \in \mathbb{R}^{3 \times 3}$ . Na začetku jo nastavimo na največjo ocenjeno napako. V našem primeru je ta napaka odstopanje pozicije v smeri  $X$  in  $Y$  za 2 metra in odstopanja kota 0,1 radiana. V primeru, da bi bile te začetne vrednosti prevelike, sistem ne bi dovolj hitro konvergirala in ne bi pravočasno dobili dovolj dobre ocene pozicije robota glede na vratca.

Kovariančno matriko podaja enačba

$$P_t^- = A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T, \quad (45)$$



Slika 4. 8: Področje verjetnosti pozicije v naslednjem koraku

Pri tem  $A_t P_{t-1} A_t^T$  predstavlja predhodno nedoločenost  $P$  po vnosu gibanja,  $A_t$  je definirana kot

$$\mathbf{A}_t = \frac{\partial f}{\partial p} = \begin{bmatrix} 1 & 0 & u_{t-1}^y \cos \varphi_{t-1} - u_{t-1}^x \sin \varphi_{t-1} \\ 0 & 1 & u_{t-1}^x \cos \varphi_{t-1} - u_{t-1}^y \sin \varphi_{t-1} \\ 0 & 0 & 1 \end{bmatrix}, \quad (46)$$

medtem ko  $\mathbf{Q}_{t-1}$  predstavlja šum v fazi predikcije.  $\mathbf{W}_t \mathbf{Q}_{t-1} \mathbf{W}_t^T$  prikazuje šum  $\mathbf{Q}_{t-1}$ , dodan  $\mathbf{P}_t^-$  glede na  $\mathbf{P}_{t-1}$ . Na eksperimentalni podlagi je bilo ugotovljeno, da se informacije o šumu hitrosti in kotu lahko zapiše z normalno porazdelitvijo

$$\mathbf{Q}_{t-1} = \mathbb{E}[\mathbf{w}\mathbf{w}^T] = \begin{bmatrix} (0,2\Delta t)^2 & 0 & 0 \\ 0 & (0,2\Delta t)^2 & 0 \\ 0 & 0 & (0,02)^2 \end{bmatrix} \quad (47)$$

Šum  $\mathbf{Q}_{t-1}$  se preko matrike  $\mathbf{W}_t$

$$\mathbf{W}_t = \frac{\partial f}{\partial \mathbf{w}} = \begin{bmatrix} \cos \varphi_{t-1} & -\sin \varphi_{t-1} & 0 \\ \sin \varphi_{t-1} & \cos \varphi_{t-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (48)$$

prenese v negotovost pozicije. Tej fazi sledi posodobitev pozicije robota glede na zaznana vratca.

### *Posodobitev*

Naš senzorski model meri samo relativno pozicijo glede na trenutna najbližja vratca, ki so postavljena v koordinatno izhodišče. Na podlagi slike s kamere izračunamo vektor oddaljenosti in kota pogleda na vratca  $\mathbf{z}_t = [d_{vr} \quad \varphi_{vr}]^T$ . Ob vsaki novi sliki se posodobijo podatki o stanju sistema

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \mathbf{K}_t (\mathbf{z}_t - \hat{\mathbf{z}}_t) = \mathbf{p}_{t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{h}(\mathbf{p}_{t-1})), \quad (49)$$

kjer je

$$\hat{z}_t = h(p_{t-1}) = \begin{bmatrix} d_{vt-1} \\ \varphi_{vt-1} \end{bmatrix} = \begin{bmatrix} \sqrt{p_{t-1,X}^2 + p_{t-1,Y}^2} \\ \text{atan}\left(\frac{-p_{t-1,X}}{p_{t-1,Y}}\right) - p_{t-1,\varphi}^2 \end{bmatrix} \quad (50)$$

funkcija, ki izračuna predvideno vrednost  $\hat{z}_t$  glede na ocenjeno lego robota na vratca. Pri tem je  $K_t$  Kalmanovo ojačenje in je definirano kot

$$K_t = P_{t-1} (H_t)^T (S_t)^{-1} \quad (51)$$

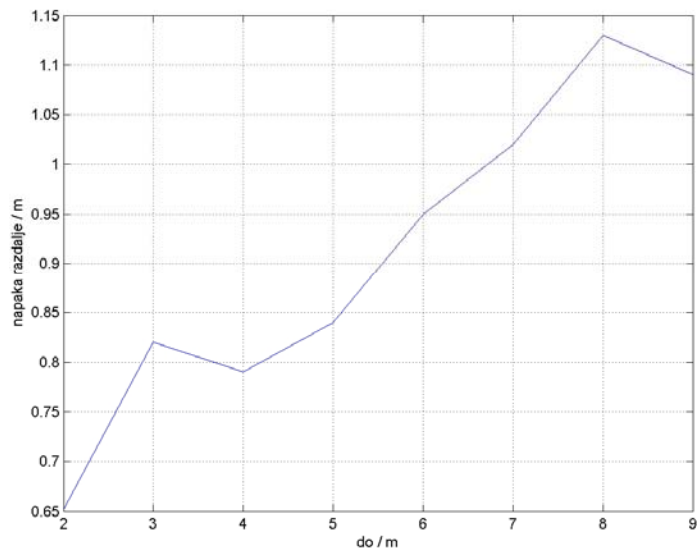
$$S_t = H_t P_{t-1} (H_t)^T (S_t)^{-1} + R_t, \quad (52)$$

$$H_t = \frac{\partial h(s_{t-1})}{\partial s_t} = \begin{bmatrix} \frac{p_{t-1,X}}{d_{vt-1}} & \frac{p_{t-1,Y}}{d_{vt-1}} & 0 \\ -\frac{p_{t-1,Y}}{d_{vt-1}^2} & \frac{p_{t-1,X}}{d_{vt-1}^2} & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad (53)$$

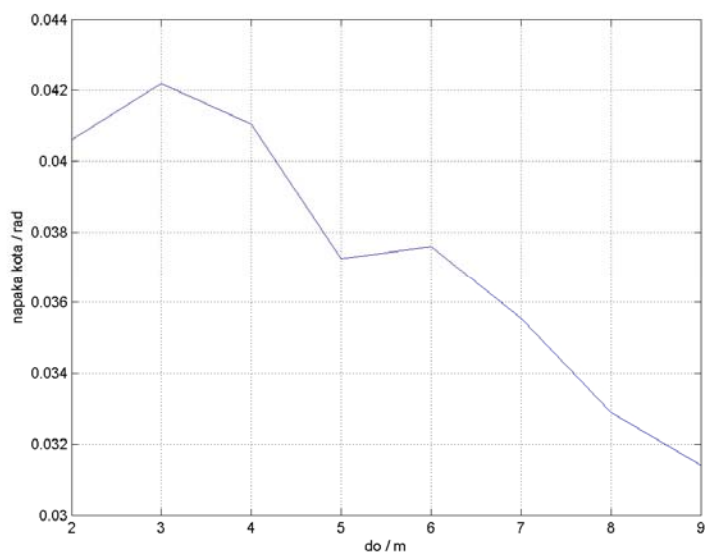
kjer je  $R_t$  šum senzorjev, ki sta prikazana za kot in razdaljo na slikah 4.9 in 4.10. V trenutkih, ko na zajetih slikah niso bila zaznana vratca in tako posredno ni bila znana pozicija vratca, smo namesto merjenih podatkov uporabili predvidene vrednosti  $\hat{z}_t$ . Vse podatke smo testirali tudi za primer prevelikega odstopanja od ostalih meritev. V primeru, da je bila vrednost

$$\delta_t = (z_t - \hat{z}_t)^T S_t (z_t - \hat{z}_t) \quad (54)$$

večja od vnaprej določene vrednosti, smo te podatke zavrgli in uporabili predvidene vrednosti, ki smo jih izračunali na podlagi predhodnih meritev in gibanja robota. Tako obdelani podatki so služili kot vhodni podatki za navigacijo robota po smučišču.



Slika 4.9: Absolutna napaka ocene razdalje

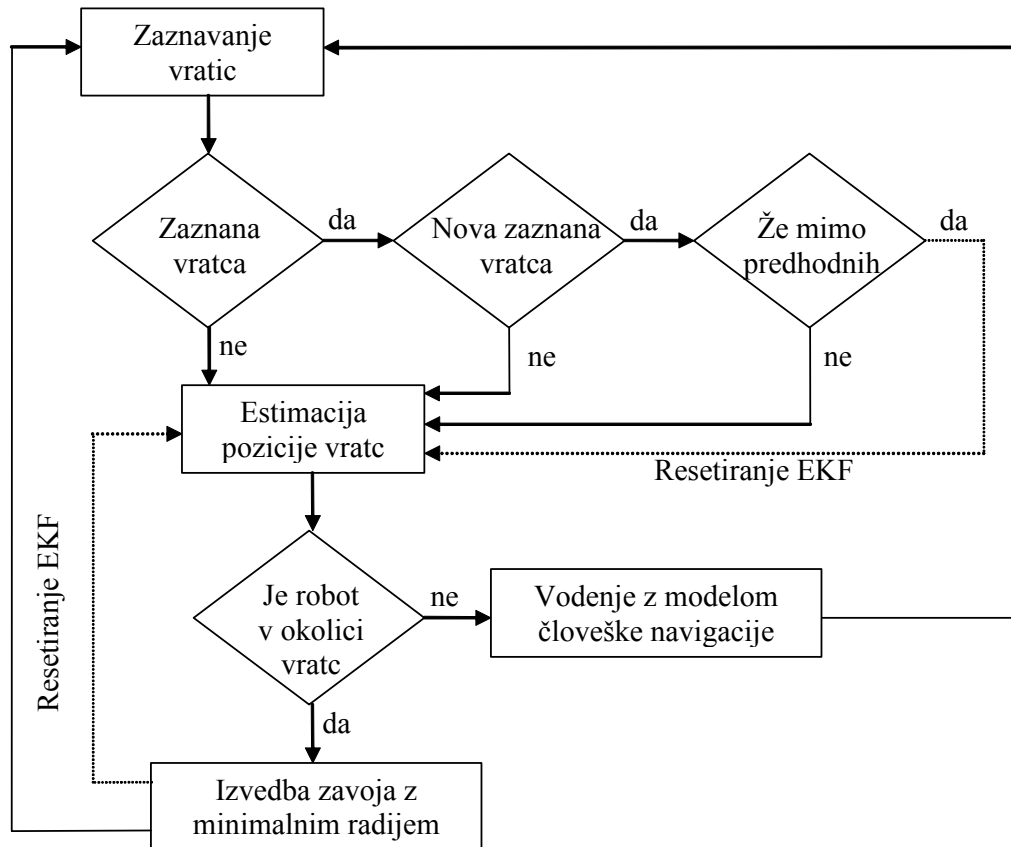


Slika 4.10: Absolutna napaka ocene kota ovire

Zbrani podatki omogočajo tudi izračun pozicije in orientacije robota med smučanjem in s tem rekonstrukcijo izvedene poti. Na enaki osnovi se gradijo tudi zemljevidi, ki se obnavljajo med izvajanjem nalog. V ta spekter spadajo naloge sočasne lokalizacije in gradnje zemljevidov. V našem primeru je bil zgrajen zemljevid zgolj pripomoček za analizo izvedenega smučanja.

## 4.6 Izvajanje poti

V našem delu je bila pot določena s pomočjo smernih vratc, ki so bila postavljena v izmeničnem zaporedju z rdečimi in modrimi vratci. Osredotočili smo se zgolj na najbližja vratca, katerih barva določa smer, s katere jih je potrebno obvoziti. Zavoj okrog vratc se izvede z najmanjšim možnim radijem, ki je odvisen tako od hitrosti kot tudi od lokalnega naklona površine in se ugotovi na podlagi zaznanega naklona površine in hitrosti gibanja v sagitalni ravnini. Celoten algoritem je predstavljen na sliki 4.10.



Slika 4.11: Algoritem za izvajanje poti

Algoritem je sledeč:

- Najprej se preveri, ali so zaznana vratca. Če so zaznana vratca nova, se še preveri, ali je robot že zapeljal mimo predhodnih.
- Na podlagi razširjenega Kalmanovega filtra se izračuna trenutno pozicijo robota glede na vratca. V primerih, ko vratca niso zaznana, se namesto senzorske informacije o poziciji vratc uporabijo predvidene vrednosti iz Kalmanovega filtra.
- Če je robot že v neposredni okolici vratc, izvede zavoj z najmanjšim možnim radijem. V drugih primerih pa se z modelom človeške navigacije vodi v bližino vratc, kjer bo potem izvedel zavoj.
- V delu, ko ni podatkov o poziciji vratc, kot tudi ni podatkov iz Kalmanovega filtra zaradi resetiranja, robot vztraja v predhodno predpisanem zavoju z najmanjšim možnim radijem. Resetiranje v našem primeru pomeni, da se na novo nastavijo parametri Kalmanovega filtra, saj podatki o prejšnjih vratcih nimajo nobenega vpliva na pozicijo trenutno zaznanih vratc.



## 5 Implementacija realnega sistema

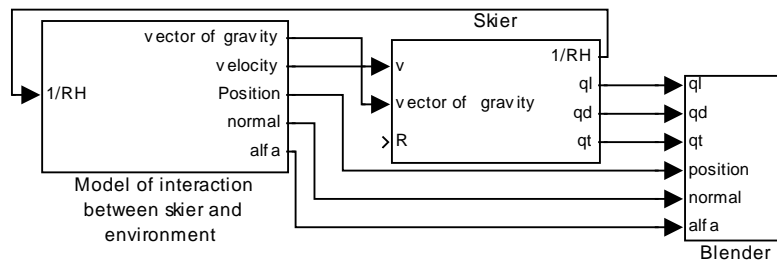
V tem poglavju opisujemo postopke implementacije predstavljenih algoritmov za vodenje robota smučarja. Algoritme smo najprej preizkusili s simulacijo v navideznem okolju, nato pa se na realnem sistemu. V nadaljevanju opisujemo tudi zgradbo robota.

### 5.1 Navidezni robot smučar

Za namen testiranja algoritmov smo modelirali robota in smučišče v navideznem okolju. Kot je splošno znano, je veliko hitreje in tudi učinkoviteje testirati algoritme s pomočjo simulacije kot na realnem smučišču, kjer ni vselej pogojev za izvedbo testiranja. Simulacijo smo uporabili tako za preverjanje stabilnosti kot tudi za testiranje algoritmov za navigacijo med vratci. Za simulacijo v navideznem okolju smo uporabili prej opisane dinamične modele. V dinamični model smučarja, predstavljen v drugem poglavju, smo vodili vektor gravitacije in hitrost, ki sta izračunana iz modela okolja. Na hitrost v sagitalni ravnini robota vplivajo gravitacija, upor zraka in trenje s podlago. Robot povzroča akcijo na smučišču zgolj z naklonom na podlago in s spreminjanjem obremenitve posameznih nog, kar povzroči spremembo radija zavoja in s tem spremembo poti.

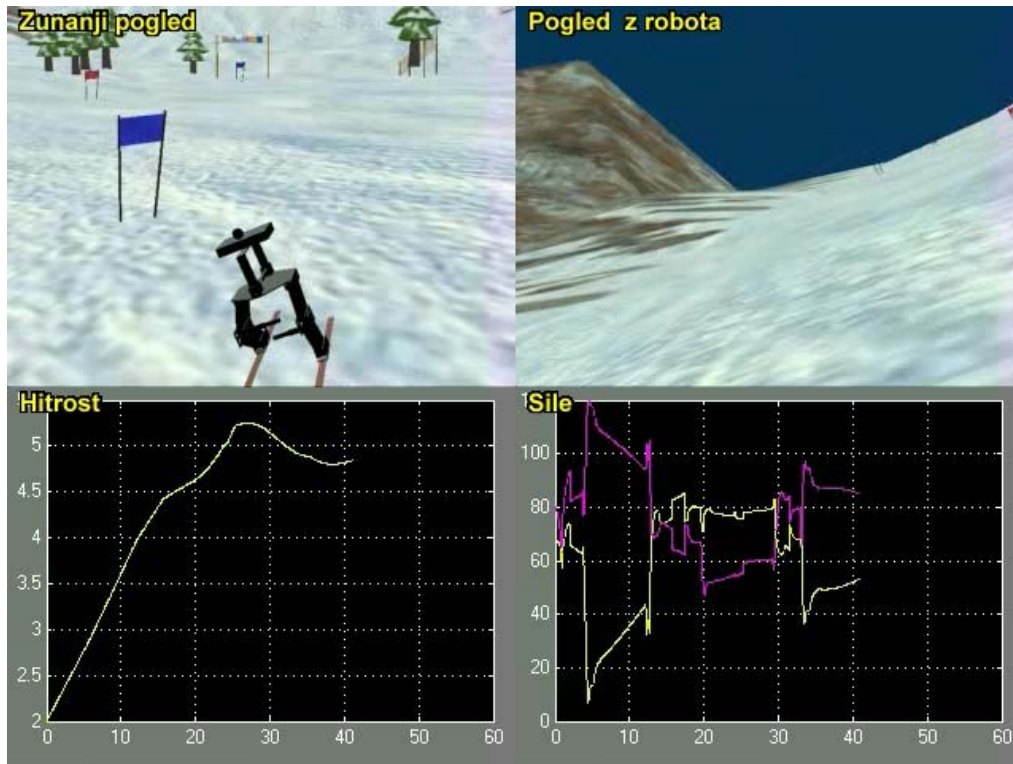
V simulacijsko okolje smo implementirali tudi algoritme vodenja, ki smo jih natančneje predstavili v 3. in 4. poglavju. Podatke o legi smučarja na smučišču smo pošiljali v virtualno okolje, ki je bilo zgrajeno v Blenderju. Programski paket Blender omogoča postavljanje poljubnega števila navideznih kamer v prostoru. Kamere se lahko tudi premikajo skupaj s posameznimi objekti navideznega okolja. V našem primeru smo namestili dve navidezni kameri. Prva je bila nameščena na robotu in je simulirala pogled iz kamere, kot ga imamo sicer tudi na realnem robotu smučarju. Druga je bila nameščena na določeno razdaljo pred robotom in se je premikala skupaj z robotom, njena orientacija pa je bila ves čas enaka. Na ta način smo imeli v vsakem trenutku na voljo dva pogleda, in sicer zunanji pogled na smučarja ter pogled iz robota. Ti vizualni podatki so služili za testiranje algoritmov za navigacijo med vratci. Okolico seveda lahko spreminjamo po želji. Obstajajo pa

tudi že zgrajeni modeli posameznih smučišč, na podlagi katerih je možno testiranje na različnih progah. Modeli podlage oziroma smučišča so podani v obliki mrež trikotnikov. Na meji med posameznimi trikotniki se pojavi trenutna sprememba normale. Naš model je predvideval le kontakt s podlago v premici, ki povezuje obe nogi, kar je v primeru večje hitrosti pomenilo, da je smučar v trenutku izgubil kontakt s podlago. V realnosti se ta kontakt porazdeli po celotni dolžini smučke in tudi v primeru hitre spremembe naklona podlage se smučka zravna in s tem obdrži kontakt s podlago, saj smučka s svojo zgradbo in obliko deluje kot dušilka. V našem primeru se s tem problemom v simulaciji nismo posebej ukvarjali, je pa Federolf [12] v disertaciji predstavil metodo na osnovi končnih elementov za modeliranje smučk, ki bi pri modeliranju omogočala bolj natančen odziv robota v odvisnosti od sil, s katerimi robot deluje na smučko. Da bi se izognili tem preskokom, smo normalo podlage filtrirali. S tem smo navidezno zgladili površino, po kateri se je spuščal virtualni robot.



Slika 5.1: Model smučanja z vizualizacijo

V navideznem okolju lahko enostavno spreminjamo tudi pozicije vratc, kar omogoča generacijo različnih trajektorij na posamezni progi. Iz navideznega okolja tako pridobimo tako podatke o odzivu smučarja na okolje kot tudi poglede na smučarja in iz smučarjeve perspektive.



Slika 5.2: Navidezno okolje

Na sliki 5.2 so podani rezultati simulacije. Robot začne smučati s hitrostjo 2 m/s. Ta hitrost se mu povečuje zaradi dinamične komponente gravitacije. V 27. sekundi se hitrost ustali, saj se dinamična komponenta gravitacije izenači s silo trenja in zračnega upora. Iz diagrama sil je razvidno, da je v 1. sekundi robot izvedel najprej razbremenitev smučk (sočasno krčenje obeh nog). Nato pa se je obremenitev smučk povečala, saj robot ustavlja svoje gibanje in se večje sile pojavijo zaradi vztrajnosti. Robot je v času simulacije izvedel dva zavoja z enakim naklonom na podlago. V začetku je radialna sila zaradi nizke hitrosti majhna, zato robot deluje z večjo silo na notranjo smučko. Z večanjem hitrosti pa se sila v zavoju enakomerno porazdeli med obe nogi.

## 5.2 Zgradba robota

Ker smo želeli preizkusiti algoritme vodenja in navigacije tudi v praksi, je bilo potrebno razviti in izdelati robota smučarja. Celoten robot je prikazan na sliki 5.3. Sestavljen je iz dveh nog in trupa, na katerem se nahajata navigacijski računalnik in kamera. Na platformo so pritrjeni akumulatorji in krmilni računalnik.

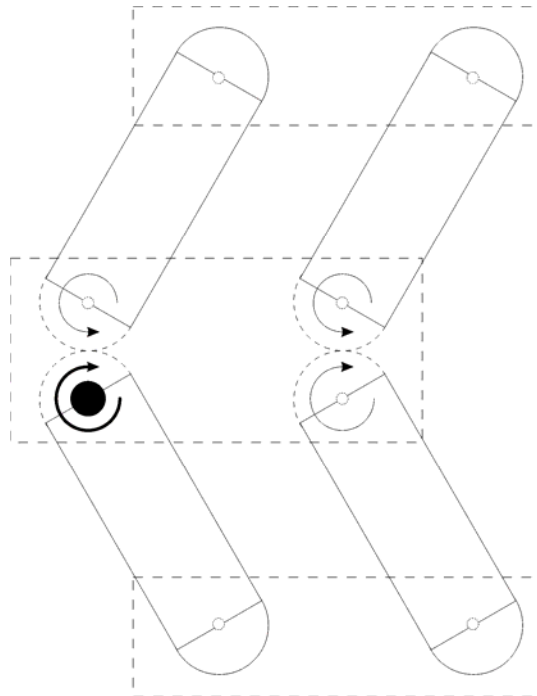
Zglobo mehanizma so gnani z motorji Maxon RE40 z maksimalnim navorom 0.6

Nm. Navor se v sklepe robota prenaša preko 3-stopenjskih planetarnih prenosov z razmerjem 156:1. Prenosi imajo na izhodni strani mrtvi hod  $1^\circ$ . Zaradi mehanskih omejitev je skrčenje posamezne noge omejeno na maksimalen kot  $58^\circ$ .



Slika 5.3: Robot smučar

Pri tem je potrebno poudariti, da moramo navor motorjev sinhrono prenašati na oba kraka paralelograma, sicer je mehanizem nevodljiv. Prenos je izveden preko zobniškega mehanizma, kot je prikazano na sliki 5.4.



Slika 5.4: Zgradba noge

S poudarjeno puščico je prikazano gibanje pogonskega sklopa. Gibanje v to smer povzroči fleksijo noge.

### 5.3 Senzorski sistem

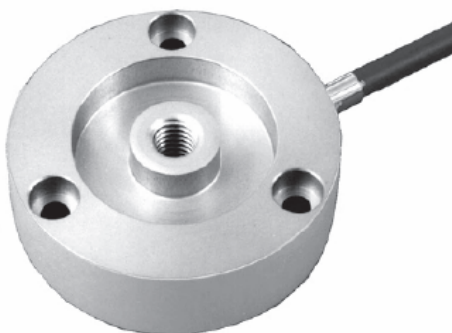
Za avtonomno vodenje robota je nujno sprotno zaznavanje okolice, še posebej v tako zahtevnem okolju, kot je smučišče. Človek med smučanjem zaznava okolico s pomočjo različnih čutil. Največji pomen pri tem imajo vid, zaznava ravnotežja in zaznane sile na noge. Pri robotu lahko te podatke pridobimo s pomočjo senzorjev, kot so senzorji sile, enkoderji na motorjih, kamera in žiroskopi.

### *Senzor sile*

Za zaznavanje reakcijskih sil podlage na robota sta med smuči in stopala robota vgrajena dva enodimenzionalna senzorja sile *Burster 8524-6001* z maksimalno silo natega ali stiska 1000 N. Ta senzor je zasnovan tako, da je praktično neobčutljiv na sile, ki ne delujejo v vertikalni smeri. Ta lastnost nam omogoča, da ga lahko pritrdimo neposredno med smuči in stopala robota brez uporabe posebnih vodil. Osnovni podatki o senzorju so navedeni v tabeli 5.1.

Tabela 5.1: Karakteristike senzorja sile

|                                |                    |
|--------------------------------|--------------------|
| Natančnost                     | $\leq 2.5\text{N}$ |
| Občutljivost                   | 1.5 mV/V           |
| Temperaturno območje delovanja | -30 ... 70°C       |
| Masa                           | 0.25 kg            |
| Maksimalna sila                | 1000N              |
| resolucija zajemanja           | 10 bitov           |
| Naravna frekvenca              | > 3 kHz            |



Slika 5.5: Senzor sile 8524-6001

### *Inklinometer*

Človek skrbi za ravnotežje s pomočjo ravnotežnega organa v srednjem ušesu. Pri robotu smučarju smo ravnotežni organ simulirali z *XSens MTI* inklinometrom. Za zaznavanje kotnih hitrosti v 3D prostoru uporablja elektronski girooskop. Ob tem pa meri še pospeške in magnetno polje, kar mu zagotavlja stabilne reference.

Tabela 5.2: Karakteristike inklinometra Xsens Mti

|                                |  |
|--------------------------------|--|
| Masa                           | 50 g   |
| Dimenzije                      | 58x58x22 mm  |
| Frekvenca zajemanja            | 500 Hz kalibrirani podatki senzorjev<br>120 Hz podatki o orientaciji |
| Temperaturno območje delovanja | -20 ... 55°C   |
| Resolucija kotov               | 0.05°  |
| Statična točnost               | <0.5°  |
| Dinamična točnost              | <2°  |

Orientacija senzorja je podana absolutno v prostoru. Ob tem senzor zagotavlja še kalibrirane informacije o pospeških v posameznih ravninah, kotne hitrosti in orientacijo zemeljskega magnetnega polja.



Slika 5.6: Inklinometer Xsens Mti

### *Kamera*

Za zaznavanje okolice smo uporabili barvno USB 2 kamero *Webcam for Notebooks*, z zornim kotom 74° in resolucijo 320x240 slikovnih elementov. Kljub temu da smo izbrali širokokotno kamero, je zorni kot naše kamere še vedno majhen v primerjavi s človeškim očesom. Človekov zorni kot namreč znaša okrog 160°. Zaradi tega bomo lahko predmet opazili šele tedaj, ko bo robot usmerjen k predmetu. To velja toliko bolj, ko smo blizu opazovanega predmeta.



Slika 5.7: USB kamera

### *GPS*

Merjenje hitrosti pri smučanju je problematično. Odometrija daje precej nezanesljive ocene hitrosti. Z našim senzorskim sistemom hitrost najenostavneje merimo s pomočjo GPS senzorjev. V našem GPS sprejemniku smo uporabili integrirano vezje Antares 4 in frekvenco zajemanja 4 Hz. Ob pogoju, da podatki iz satelitov niso moteni in da senzor v celotnem času zaznava najmanj 5 satelitov, lahko izmerimo hitrost z največjim odstopanjem  $0.1 \text{ ms}^{-1}$ .



Slika 5.8: GPS senzor

### *Enkoderji*

Za merjenje pozicije in hitrosti posameznih motorjev smo uporabili enkoderje *HEDS 5540*. Uporabljeni inkrementalni enkoder je sestavljen iz kolesca z dvema vrstama enako oddaljenih rež, razvrščenih koncentrično na kolescu. Prehod reže preko svetlobnega žarka povzroči električni pulz. Število pulzov v znani smeri ob upoštevanju znanega kota med režami prikaže kotni premik v določenem intervalu



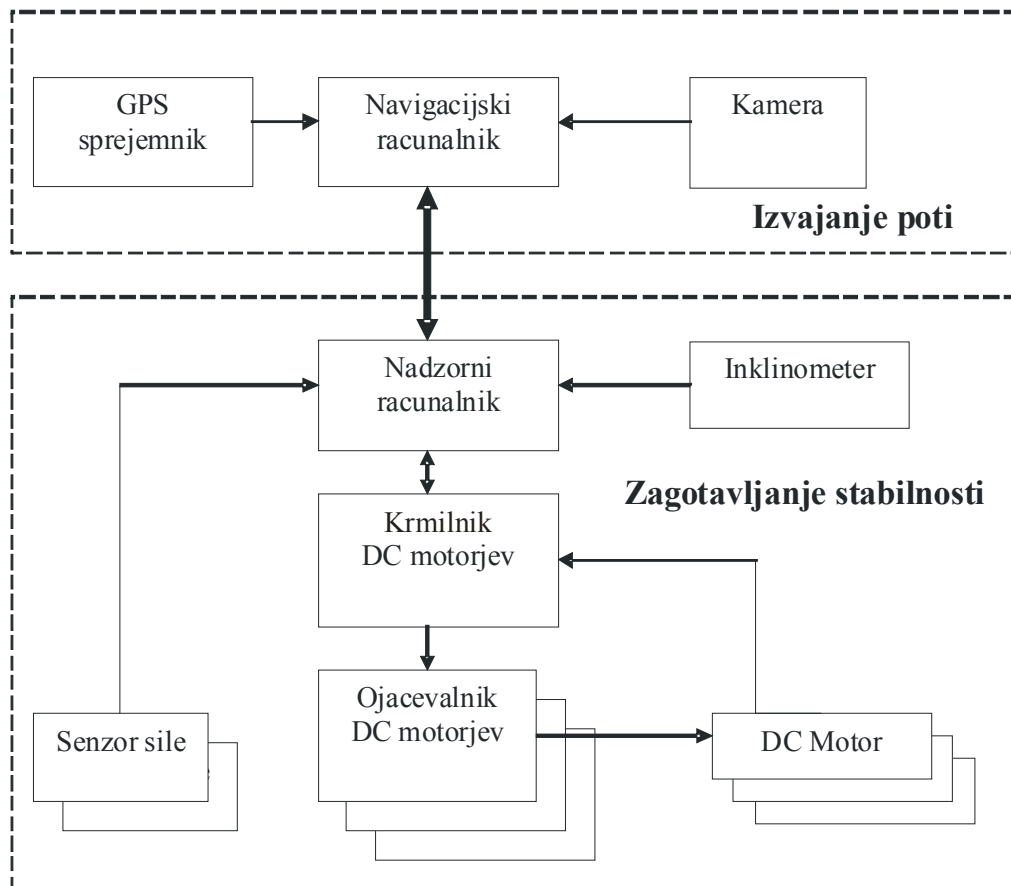
štetja. Ker so reže med vrstama zamaknjene za  $90^\circ$ , se s primerjavo med signaloma določi smer gibanja. Ker enkoder izgubi pozicijo ob izključitvi napajanja, je potrebno ob vsakem zagonu inicializirati začetno pozicijo, ki je določena z režo v tretji vrstici.

Hitrost se meri s pomočjo enkoderja na dva načina, in sicer z merjenjem časa med posameznimi impulzi ali pa s štetjem impulzov v določenem časovnem intervalu [47]. V obeh primerih se lahko meri le ena vrednost. Znano je, da imata metodi veliko nenatančnost pri visokih oziroma nizkih hitrostih. V bistvu obe metodi ocenjujeta hitrost na podlagi odvajanja diskretnih podatkov. Trenutno se uporablja množica različnih filtrov za določanje hitrosti, vendar njihova uporaba za vodenje v realnem času ni priporočljiva, saj zaradi svoje zgradbe vnašajo v meritve zakasnitve, kar neposredno vpliva na stabilnost [48]. Za oceno hitrosti pri robotu smo uporabili metodo s fiksnim časom. Pospeške realnega sistema smo pridobili iz hitrosti, ob tem pa je bil signal hitrosti filtriran z nizkopasovnim Butterworthovim filtrom tretjega reda.

#### **5.4 Zgradba sistema vodenja**

Kot smo že omenili, je naloga vodenja zagotavljanje stabilnosti in ustrezna navigacija med smučarskimi vratci. Zato je tudi vodenje zgrajeno iz dveh ločenih sistemov, in sicer sistema za zagotavljanje stabilnosti in sistema za navigacijo. Algoritem za zagotavljanje stabilnosti je implementiran v Matlab xpcTarget. XpcTarget je operacijski sistem, ki je namenjen delu v realnem času v povezavi z Matlabom. Krmilni računalnik je nadziral vodenje motorjev, branje podatkov iz inklinometra in merjenje sil. Za zagotavljanje stabilnosti je potreboval še podatke o hitrosti gibanja v sagitalni ravnini robota. Kot krmilni računalnik smo izbrali enokartični računalnik s procesorjem Geode GX1-300MHz z vodilom PC104. Na to vodilo je priključen krmilnik gibanja POSYS 1924, ki omogoča vodenje do štirih motorjev. Krmilnik omogoča tako hitrostno vodenje kot tudi vodenje navora s frekvenco 4,88 kHz. Pri robotu smo uporabili hitrostno vodenje. Za napajanje motorjev skrbijo močnostni ojačevalniki Elmo Violin. Na krmilnem računalniku teče operacijski sistem xPC Target, ki je paket programskega okolja MathWorks Matlab. Za ta sistem smo se odločili, ker omogoča hiter razvoj prototipnih aplikacij za delo v realnem času. Z opisano aparaturno opremo smo dosegli delovanje

algoritma za zagotavljanje stabilnosti s frekvenco vzorčenja 1 kHz. Krmilni računalnik pošilja podatke o naklonih smučarja navigacijskemu računalniku preko Ethernet povezave s frekvenco 100 Hz. Za komunikacijo med računalnikoma smo uporabili UDP protokol. Ta protokol omogoča uporabo v realnem času, vendar za ceno možnosti izgube posameznih podatkov. Navigacijski računalnik skrbi za sprotno navigacijo robota s pomočjo video kamere in GPS senzorja. Obdelava vida teče s frekvenco 30 Hz. S to frekvenco se tudi pošiljajo podatki o zelenem radiju in trenutni hitrosti krmilnem računalniku. Algoritem za obdelavo slik in določanje zelene poti smo implementirali v programskem jeziku C++ na operacijskem sistemu Windows XP. Na navigacijskem računalniku se poleg generiranja poti ocenjuje tudi hitrost v sagitalni ravnini.



Slika 5.9: Shema zgradbe sistema vodenja

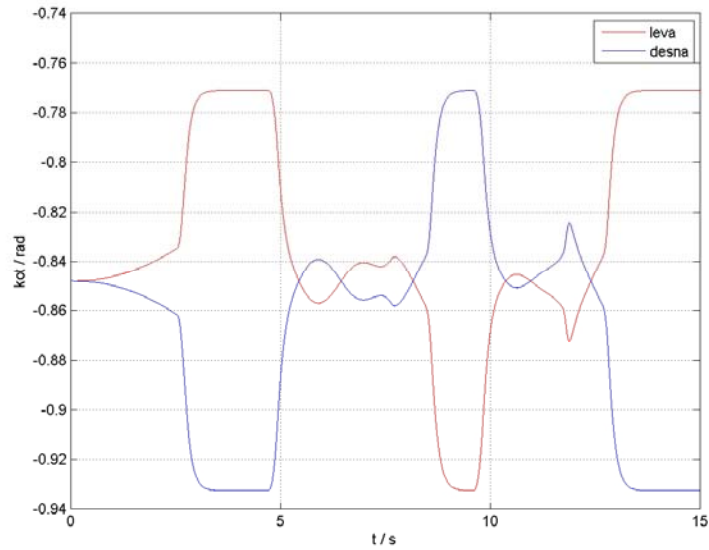
## 6 Rezultati in ovrednotenje dela

V tem poglavju bomo predstavili tako simulacijske rezultate kot tudi izvedbo na realnem sistemu. Pri tem bomo podali rezultate tako za meritve na betonski klančini z rolkami kot za meritve na snegu s smučmi s poudarjenim stranskim lokom. V prvem delu se bomo osredotočili predvsem na ovrednotenje zagotavljanja stabilnosti. V nadaljevanju poglavja pa bodo predstavljeni rezultati navigacije robota med vratci. V okviru testiranja smo izvedli okrog 30 poskusnih voženj na rolgah in približno enako število voženj na snegu. V tem poglavju bodo predstavljeni reprezentativni rezultati voženj.

### 6.1 Rolkanje

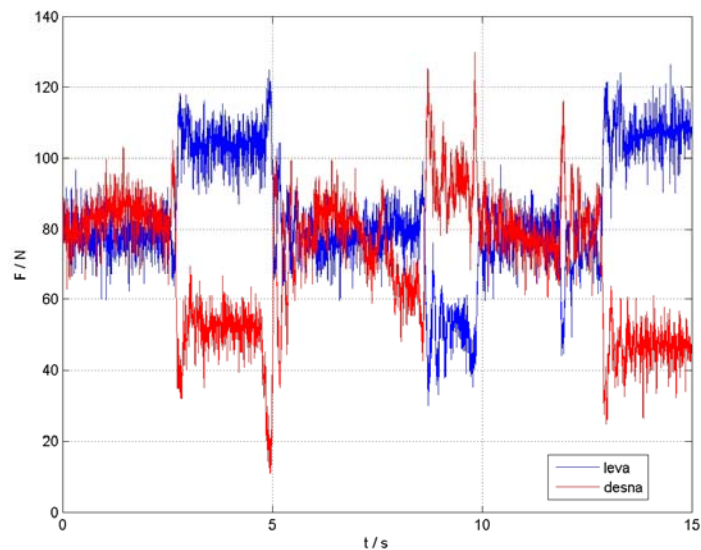
#### *Rolkanje po ravnem*

Najprej smo izvedli poskuse rolkanja po ravni površini, pri katerem se je konstantno hitrost 2 m/s zagotavljalo s pomočjo vitla. Robot je izvedel tri zavoje z določenim trajanjem. Med zavoji so bili odseki ravne vožnje. Nakloni robota so bili med vožnjo v območju med  $-0.1$  in  $0.1$  radiana, kar je pomenilo zavoje z radijem v območju med neskončnostjo in dvema metroma. Odločili smo se, da robot ne bo izvajal razbremenjevanja, zato je bila višina  $l_c$  zmeraj konstantna. V tem primeru se kota sklepov (slika 5.4) gibljeta okoli vrednosti  $-0.85$  radiana, kar zagotavlja srednjo vrednost dolžine noge med popolnoma iztegnjeno in pokrčeno.



Slika 6.1: Gibanje sklepov nog

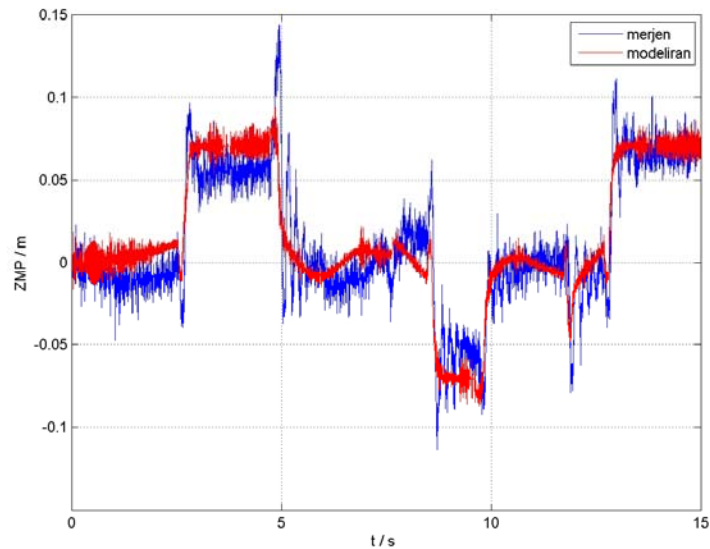
Na sliki 6.1 je predstavljeno gibanje sklepov leve in desne noge. Robot je v tem primeru izvedel tri zavoje, med temi zavoji pa so bili še ravni odseki. Iz višjega nivoja vodenja so bili podani zahtevki za zavoje ob časih 2.5 s, 8 s in 12.5 s. Zavoji so trajali različno dolgo.



Slika 6.2: Sile pravokotno na rolko

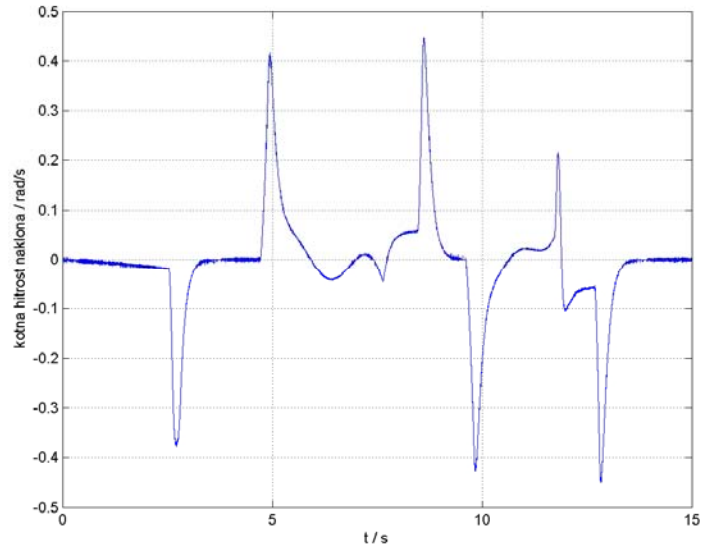
Na sliki 6.2 so prikazane merjene sile na posamezni nogi. V trenutkih, ko se spremeni naklon, se pojavijo hitre spremembe sile, ki so posledica vztrajnosti.

Zaradi tega smo v 5. sekundi dosegli tudi razmerje delitev sil 8:1 med posameznima nogama. V primerjavi z modelom, ki upošteva zgolj kinematiko nog brez elastičnosti sklepov, je to razmerje mnogo večje od modeliranega, kar je razvidno tudi s slike 6.3, saj bi na podlagi modela razmerje med silami doseglo zgolj razmerje 3:1.



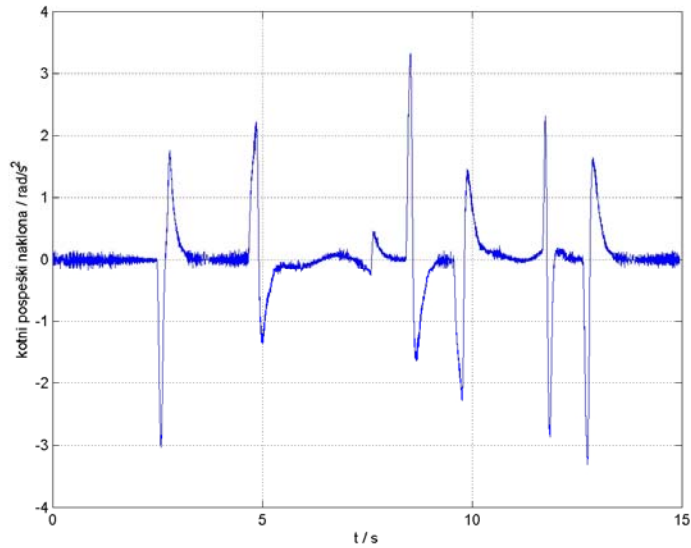
Slika 6.3: ZMP med vožnjo

Tudi v fazi, ko je robot izvajal zavoj s konstantnim radijem, je prišlo do napake v določitvi ZMP-ja, kar je posledica napake v izmeri naklona površine. Napaka med merjenim in modeliranim ZMP-jem je dokaj majhna. Večja napaka se pojavi zgolj v času trenutnih sprememb pospeška naklona na podlago. Vzrok za to napako je predvsem v mrtvem hodu sklepa, saj se je v teh trenutkih povečala sila, s katero deluje robot na podlago. To je povzročilo večje pokrčenje bolj obremenjene noge in s tem tudi povečane pospeške. Ob tem tudi opazimo nihanje robota, ki je bilo najbolj opazno v času med 5. in 6. sekundo.



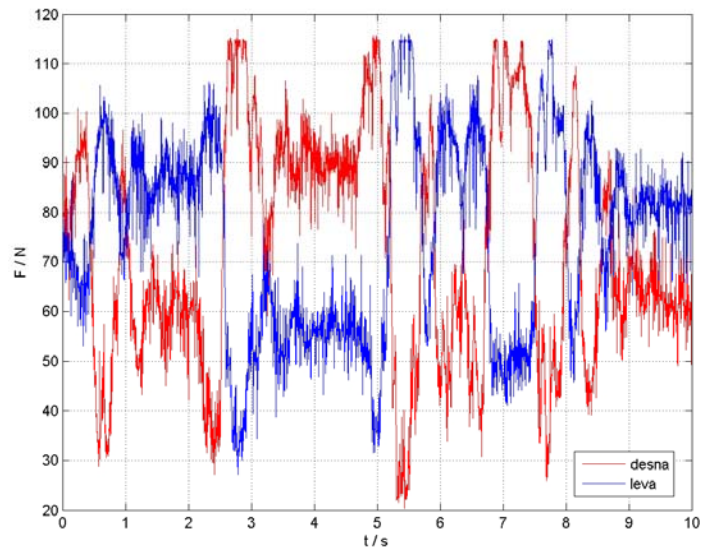
Slika 6.4:Kotna hitrost naklona na podlago

Na sliki 6.4 je prikazana hitrost naklona na podlago, ki je bila izmerjena na podlagi enkoderjev, montiranih na sklepih robota. Medtem ko na teh merjenih hitrostih ni videti nihanja, kar je razvidno tudi iz modeliranega ZMP-ja na sliki 6.3, pa je dejanska kotna hitrost nihala, kar je vidno na merjenem ZMP-ju na sliki 6.3. Prav tako tudi merjeni pospeški enkoderjev ne nihajo. Razlog za nihanje pa je lahko tudi v mehanizmu rolke. Iz rolk smo namreč odstranili blažilce, saj je robot prelahek in ne bi bil sposoben nagniti rolke.



Slika 6.5: Kotni pospešek naklona na podlago

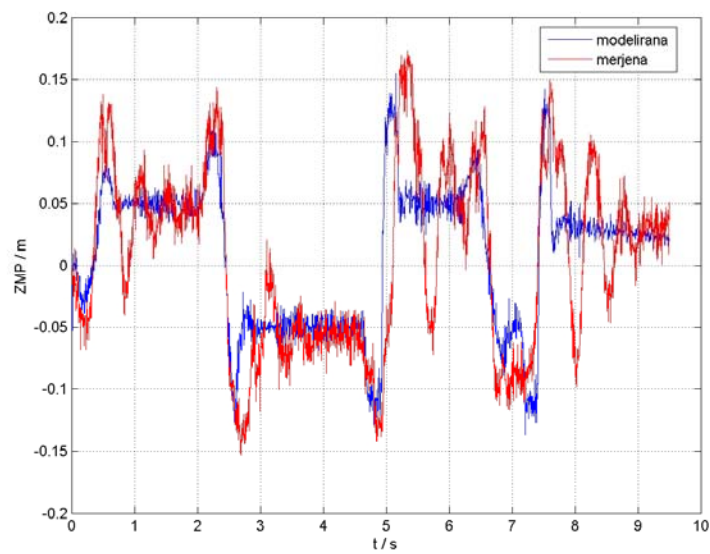
### *Rolkanje po klančini*



Slika 6.6: Sile na nogi med roljanjem po klančini

Do še večjega nihanja je prišlo v primeru spuščanja robota po klančini. Slika 6.6 prikazuje sile na posamezni nogi. Vzrok za to nihanje leži v elastičnosti celotnega mehanizma. Skozi celoten potek je vidno nihanje s frekvenco okrog 1 Hz, kar je lastna frekvenca mehanizma. Če robota namreč modeliramo kot inverzno nihalo, kot je prikazano na sliki 3.4. se lastna frekvenca takega nihala približno ujema z

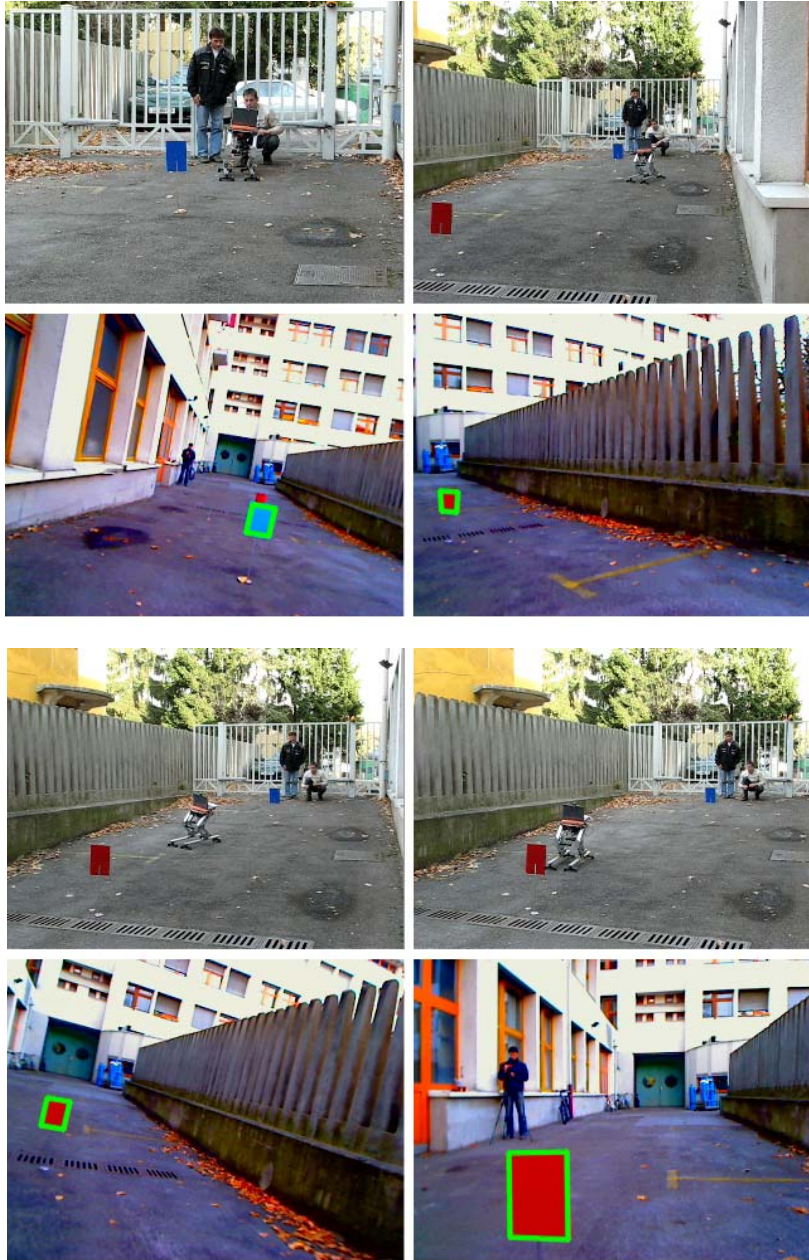
nihanjem robota.



Slika 6.7: ZMP med rolkanjem po klančini

Ker se merjeni ZMP računa na podlagi merjenih sil, je tudi s slike 6.7 razvidno nihanje. To nihanje je bilo razvidno tudi z videoposnetkov.





Slika 6.8: Zunanji in interni pogled na rolkanje na klančini

Slike 6.8 prikazujejo potek rolkanja, kjer smo na klančini med startom in ciljem postavili dvoje vratc. Ti posnetki prikažejo tipično vožnjo robota izvedeno na podlagi zaznanih vratc. Prva vrsta zgoraj kaže zunanji pogled na robota. Robot je modra vratca obvozil po levi strani, naslednja pa po desni. V drugi vrsti so prikazani pogledi iz robota v istih trenutkih. Med vožnjo so bile zajete slike, iz katerih naši algoritmi za razpoznavanje vratc le-teh niso zaznali. V teh trenutkih je bila pozicija vratc ocenjena s Kalmanovim filtrom. Vse meritve na rolgah so bile izvedene brez

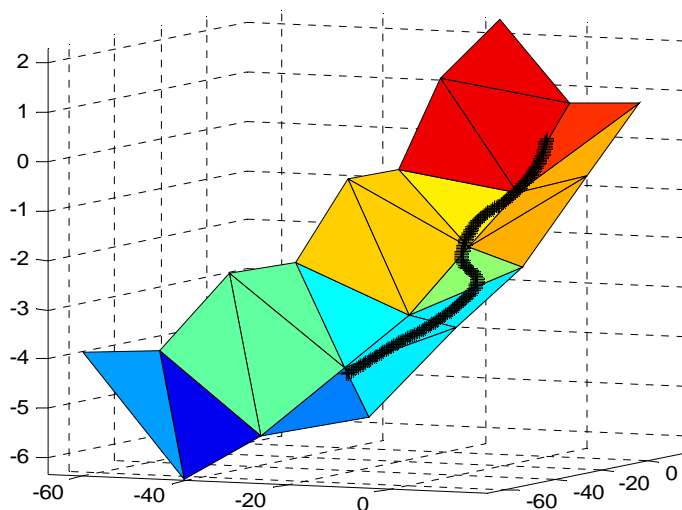
gibanja trupa.

## 6.2 Smučanje

Testiranje algoritmov za zagotavljanje stabilnosti in navigacije smo najprej izvedli v simulaciji.

### *Simulacija smučanja*

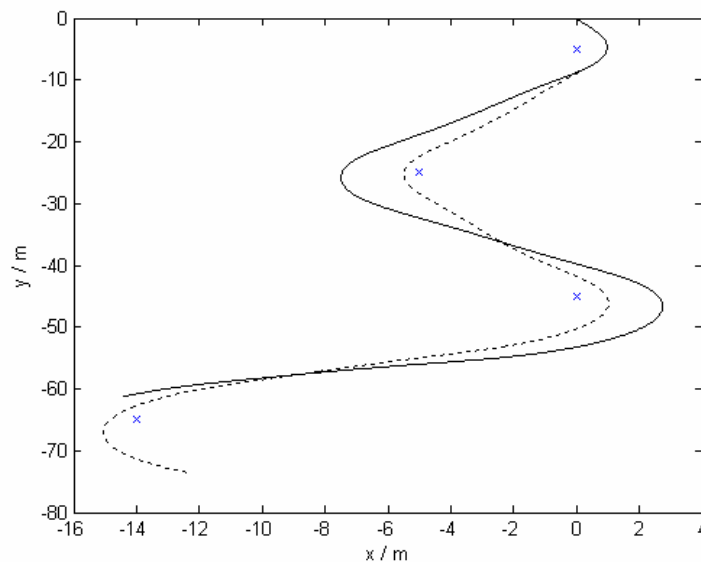
Vsi algoritmi za avtonomno gibanje smučarja so bili najprej preizkušeni v računalniški simulaciji. Na sliki 6.9 je predstavljena pot smučanja na testirani površini, izvedena s pomočjo prej opisane metode izvajanja zavojev v navideznem okolju, ob tem pa so bile upoštevane še omejitve, ki zagotavljajo stabilno smučanje.



Slika 6.9: Poligon za simulacijo smučanja

Med poskusi smo preizkusili različne postavitve vratc. Za lažjo predstavbo je naslednji diagram (slika 6.10) predstavljen v  $xy$  ravnini. Podani sta dve simulaciji izvedbe zakoličene poti. Polna črta predstavlja izvedbo ob upoštevanju, da je zavoj izveden v oddaljenosti najmanjšega možnega radija zavoja. V tem primeru vratca hitro izginejo iz vidnega kota kamere. Zato se je potrebno več časa zanašati na podatke iz predikcije filtra. Zaradi tega se je tudi v simulaciji zgodilo, da zaradi netočnega predvidevanja pozicije robot ni mogel izpeljati proge. Črtkana krivulja

prikazuje izvedbo poti pri navideznem premiku vratc za 2 metra pravokotno na globalno smer smučanja, ki je za testno postavitev usmerjena v smeri osi y. S tem smo zagotovili, da so vratca ostala dlje časa v vidnem polju kamere in je bila zato ocenjena pozicija glede na vratca natančnejša. Iz tega razloga smo tudi v realnem smučanju vratca navidežno premaknili za 3 metre pravokotno na globalno smer smučanja. V obeh primerih se sredina zavoja nahaja pod vratci, kar je posledica dejstva, da smučar ne izvede zavoja v trenutku, saj mora upoštevati pogoje, ki zagotavljajo stabilnost.



Slika 6.10: Simulirane trajektorije na izbrani progi (črtkana – navidezni premik vratc, polna – najmanjši radij okrog vratc)

V obeh primerih je simulacija trajala enak čas, vendar pa je v primeru navideznega premika vratc robot prevozil daljšo pot po progi, kar je posledica bolj direktne linije in s tem tudi višje hitrosti.

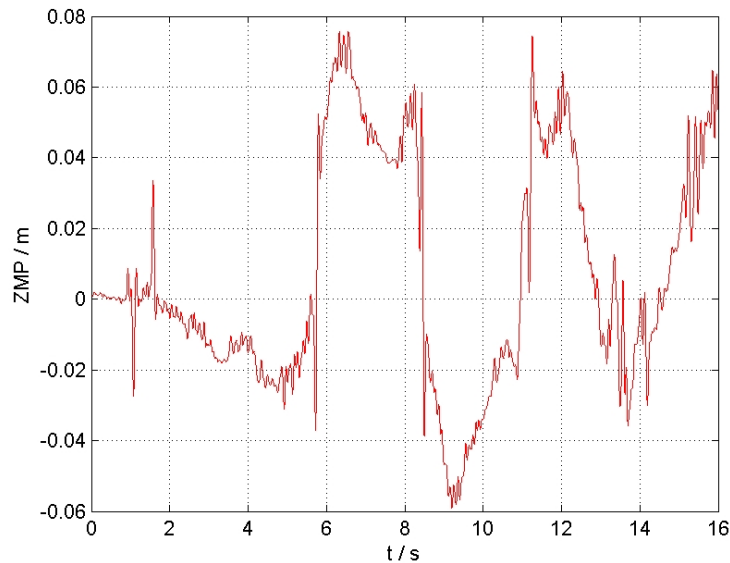
### *Testiranja na snegu*

Izvedli smo več testiranj robota, ki so bila izvedena v različnih okoliščinah. Tako je bila v prvem primeru, ki bo predstavljen v nadaljevanju, na smučišču 5 centimetrov debela plast svežega snega. V drugem primeru pa je bila snežna podlaga sveže steptana. Izkazalo se je, da ima oblika snežne podlage pomemben vpliv na

odzivanje robota. Zato bomo tipična primera voženj predstavili v ločenih podpoglavjih.

### *Svež sneg*

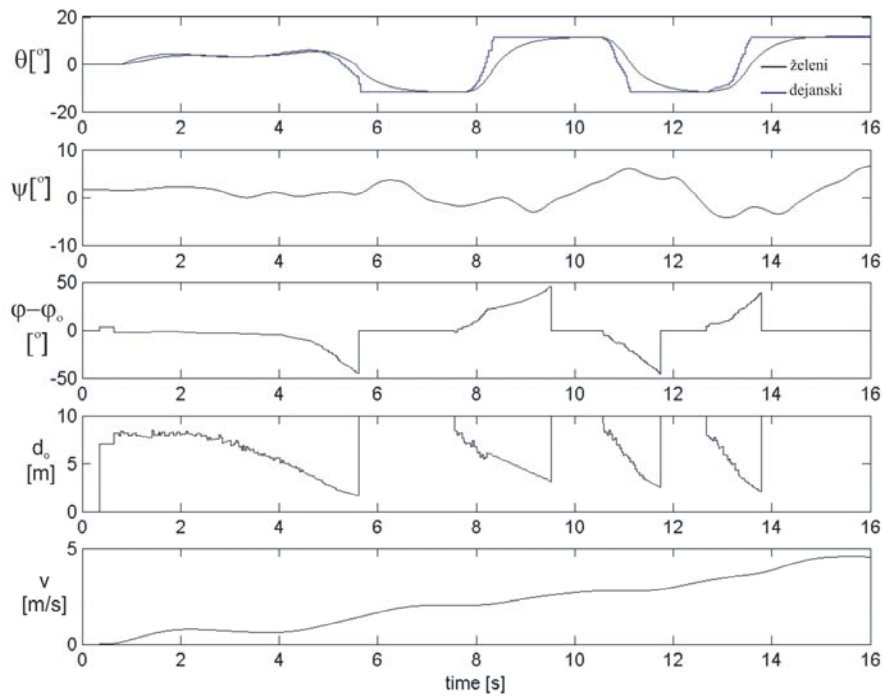
V tem primeru so bile meritve izvedene na progi s 4 vratci. Najprej se bomo osredotočili na analizo stabilnosti med izvajanjem zakoličene poti. Slika 6.11 predstavlja modeliran ZMP. V območju med 1. in 2. sekundo je bil trup v poziciji, ki enakomerno porazdeli silo med nogi. V nadaljevanju se ZMP giblje v območju, ki zagotavlja razdelitev sil na nogi v razmerju 3:1.



Slika 6.11: Modeliran ZMP

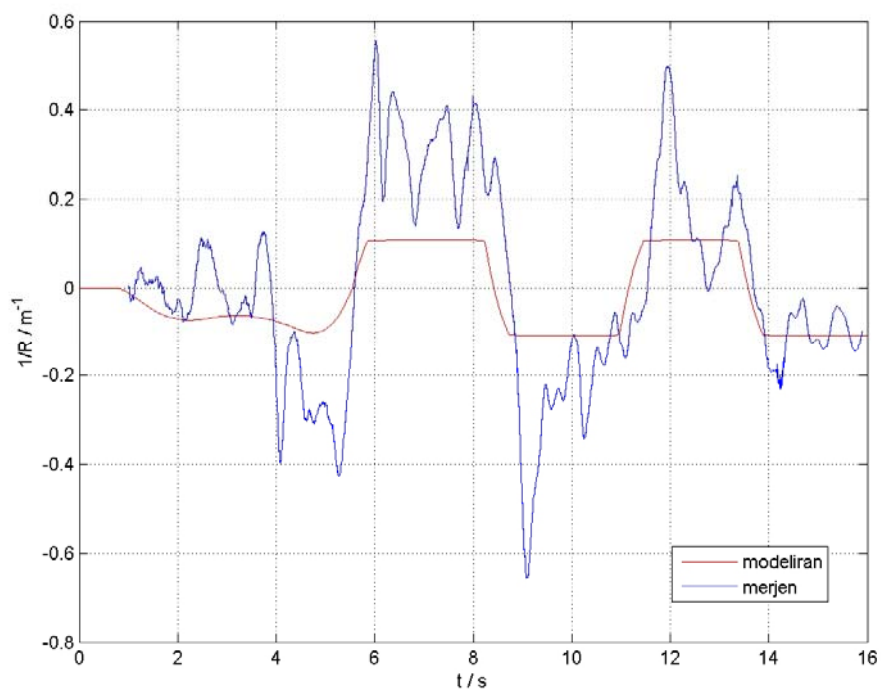
Na sliki 6.12 so predstavljeni izmerjeni navigacijski podatki, ki so bili uporabljeni za izvajanje zelene trajektorije. Opazimo, da na začetku ni nobenega hitrega gibanja, saj je hitrost spreminjanja naklona na podlago neposredno povezana s hitrostjo v sagitalni smeri. V nadaljevanju se zaradi povečanja hitrosti željeni naklon spreminja hitreje. V začetku se željeni in dejanski naklon dokaj ujemata, ko pa se hitrost poveča, se mora željeni naklon iz navigacijskega modula podrediti zahtevam za zagotavljanje stabilnosti.  $\Psi$  prikazuje lokalni naklon podlage v lateralni smeri in se spreminja v odvisnosti od smeri, v kateri se robot vozi. Največji lokalni naklon je bil zaznan v 9 sekundi. Kot pogleda na vratca ( $\varphi - \varphi_0$ ) in razdalja do vratc sta medsebojno povezana. Kot pogleda se večja, ko se robot približuje vratcem. Ko se

razdalja do vratc zmanjša pod določeno mejo, vratca izginejo izven vidnega področja kamere. V tem trenutku nimamo podatkov o kotu pogleda in razdalji, dokler se v vidnem obzorju kamere ne pojavijo nova vratca. Ker hitrost raste, je čas približevanja vratcem vedno krajši. Zaradi večje hitrosti se lahko tudi ZMP vzdržuje na sredini, saj radialna sila kompenzira silo gravitacije, ki nastopi zaradi naklona smučarja na podlago.



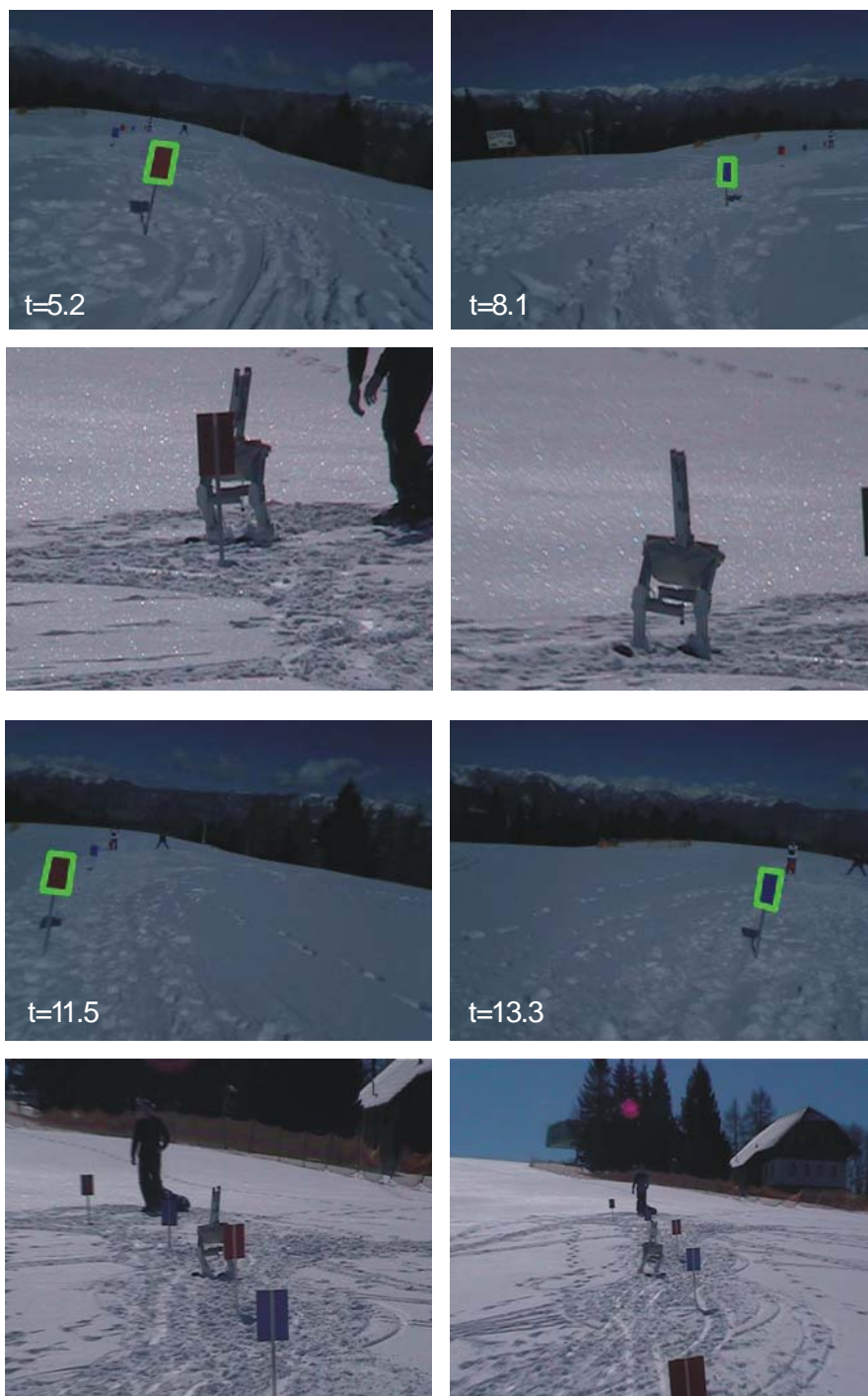
Slika 6.12: Meritve navigacijskih podatkov

Podatke o orientaciji robota in hitrosti smo uporabili za izračun izvedenih radijev zavoja, kar je prikazano na sliki 6.13.



Slika 6.13: Merjen in modeliran radij

Izkazalo se je, da se je dejanski izvedeni radij na svežem snegu zelo razlikoval od radija, ki smo ga izračunali s pomočjo modelov, znanih iz literature. Uporabljene smučke so izvedle zavoje z radijem 2.5 metra, kar bi pri modelih zahtevalo mnogo večje naklone, kot jih je robot izvedel. To lahko pojasnimo, če upoštevamo, da je prišlo na sredini smučke do globljega zarez v sneg kot v skrajnih legah smuči. Pri večji hitrosti pa so bili izvedeni radiji podobni modeliranim.



Slika 6.14: Pogledi iz robotove kamere in pogledi iz kamere ob smučišču, ki smo jih zajeli ob istem času

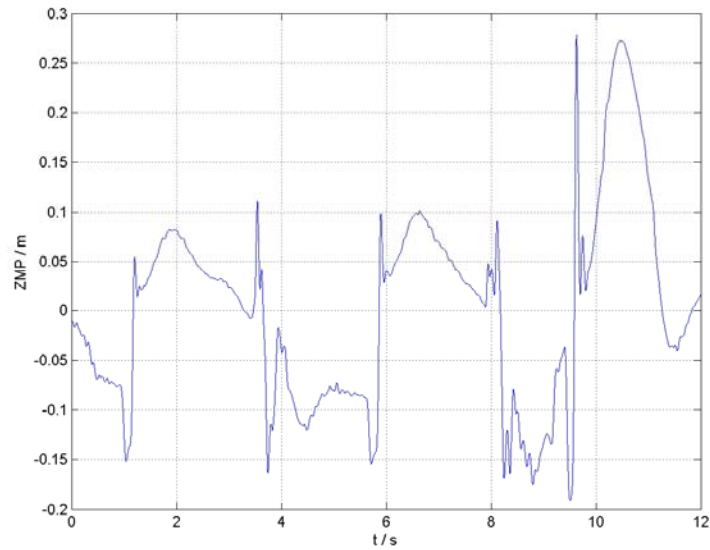
Na sliki 6.14 so predstavljene zajete slike v določenih časovnih trenutkih. Iz slik je razvidno, da je robot uspešno presmučal sistem vratc. Iz slik, ki kažejo pogled kamere na robotu, je vidno spreminjanje velikosti vratc v odvisnosti od razdalje

robota do njih. Opazimo tudi, da pri nizkih hitrostih na začetku smučanja trup skrbi za kompenzacijo komponente gravitacije.

### *Sveže steptan sneg*

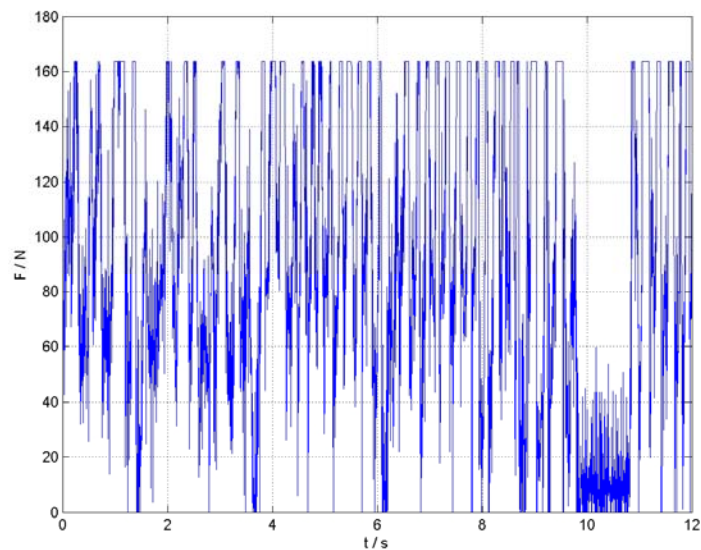
Pri testiranju na steptanem snegu se je pogosto pojavil problem, da je algoritem za zagotavljanje stabilnosti preveč spreminjal želeni radij zavoja, ki ga je generiral navigacijski algoritem. Zato se je dogajalo, da robot ni mogel izpeljati proge in je posebej na koncu proge pri večjih hitrostih odpeljal mimo vratc. Zato smo spremenili parametre algoritma za stabilnost, da je dovoljeval razdelitev sil v višjem razmerju. V tem primeru se je ZMP lahko premaknil do razdalje 2 cm od roba podporne ploskve med smučkama. Zato se je lahko skoraj celotna sila, s katero deluje robot na podlago, premaknila na eno smučko. S tako nastavljenimi parametri algoritma smo izvedli nekaj poskusov. Tudi v praksi se je izkazalo, da je s tako nastavljenimi parametri robot zelo občutljiv na reakcijske sile podlage, ki nastanejo zaradi neravnin na smučišču. V nadaljevanju je prikazan primer, ko je zaradi neravnine na progi robot smučal po eni nogi. Kot je razvidno s slike 6.15, je robot pogosto vozil na mejah stabilnih vrednosti ZMP-ja, saj v primeru, ko le ta preseže vrednost 0,17 metra, pomeni, da je praktično vsa sila zgolj na eni nogi. Ob času 9.6 sekunde je ZMP presegel to vrednost in ZMP je dosegel vrednost 0.19 metra. V tem trenutku je, če si ogledamo sliko 6.16, sila na desni nogi skoraj enaka nič, ker se je ta noga odlepila od tal. Na podlagi meritev smo želeli ugotoviti, zakaj je algoritem v tej točki odpovedal. Na podlagi merjenih zavojev se je izkazalo, da je bil naklon na podlago v tem trenutku že tak, da bi moral robot že zavijati s predpisanim radijem, ki bi stabiliziral robota. Zaradi stranskega oddrsavanja smučiči pa sta bila radij zavoja in s tem tudi radialna sila manjša od želenih, zato se je robot nagnil na notranjo smučko. Dodatno so k tej nestabilnosti prispevale tudi neravnine na podlagi.





Slika 6.15: Modeliran ZMP

Slika 6.16 prikazuje sile merjene med smučanjem. Vidimo, da so signali sil precej zašumljeni. Edina jasno razpoznavna faza je, ko noga ni v kontaktu s podlago. Deloma so vidne tudi pozicije, kjer se ZMP približuje skrajnim vrednostim.



Slika 6.16: Sila na desni nogi

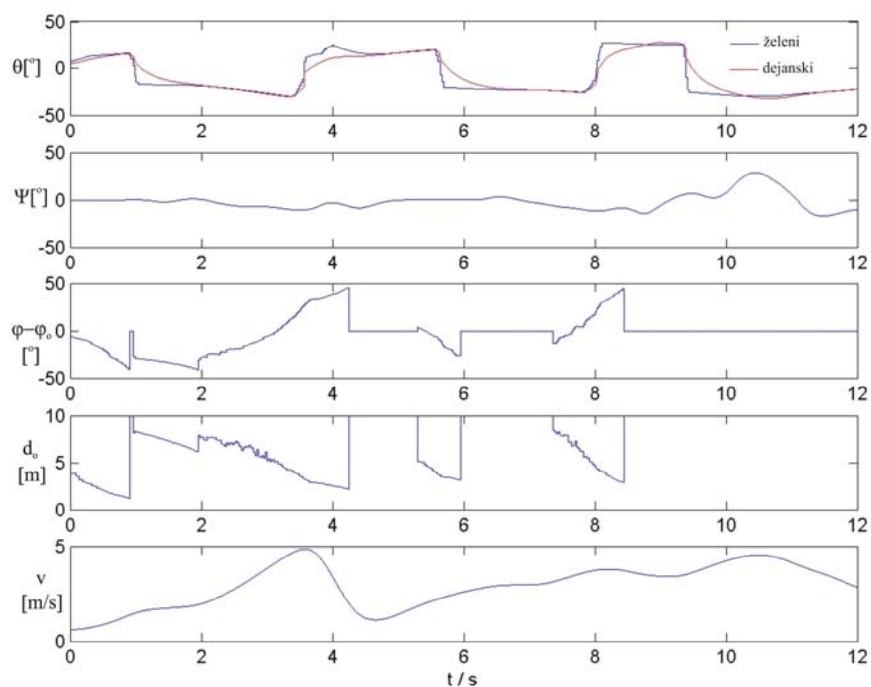
Podatki o silah, kot so na primer prikazani na sliki 6.16, so se izkazali kot neuporabni za vodenje v realnem času. Na merjenje sil je vplival tudi sneg, ki se je nabiral med smučko in ploščo na robotu, kamor je pritrjen senzor. Zato podatkov o

sili tudi nismo uporabili v vodenju.



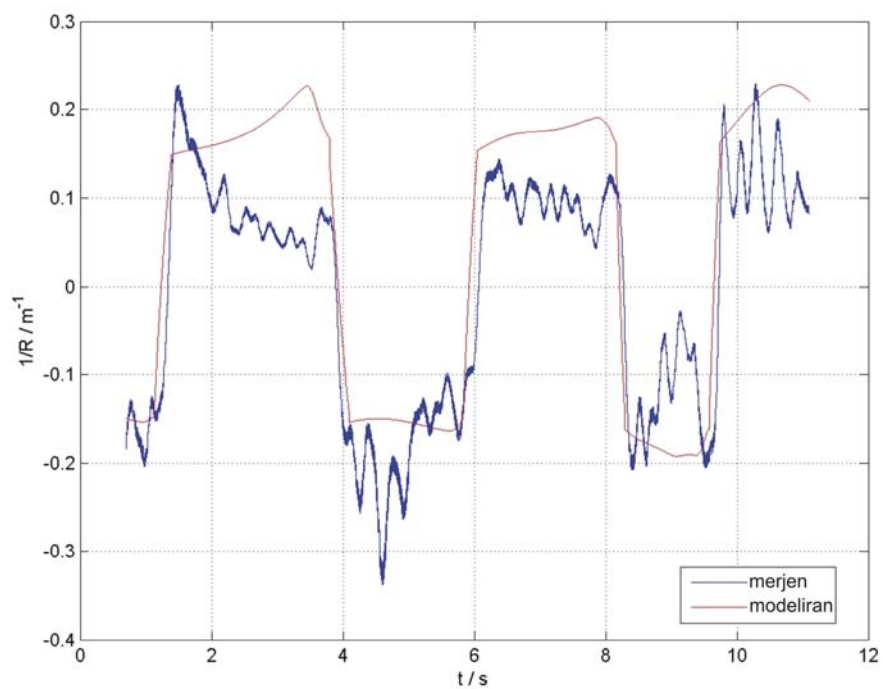
Slika 6.17: Pogled na robota v nestabilni poziciji

Na sliki 6.17 je predstavljen dogodek, ko je robot izgubil stik s podlago z eno smučko. V nadaljevanju, ko se je notranja smučka zarezala v sneg, je robot zavil in radialna sila je stabilizirala robota. Na sliki 6.18 na grafu v območju v katerem je desna noga izgubila kontakt s podlago, podatki o lokalnem naklonu površine niso verodostojni, saj je neznan lokalni naklon podlage.



Slika 6.18: Meritev navigacijskih podatkov

Če primerjamo še radije zavojev na svežem in steptanem snegu, prikazane na slikah 6.13 in 6.19, opazimo, da se na steptanem snegu mnogo bolj ujemajo z modeliranimi kot v primeru svežega snega. Radij zavojev se je v primeru svežega snega spreminjal v odvisnosti od hitrosti, saj pri nizkih hitrostih svež sneg ni nudil dovolj opore smučem neposredno pod robotom, z višjo hitrostjo pa se sila poveča po vsej dolžini smučke in je celotna smučka zarezala v bolj trdo podlago.



Slika 6.19: Merjen in modeliran radij

Iz slike 6.19 je razvidno, da je posredno računani radij v trenutku izgube stika s podlago znašal nad 20 metrov, medtem ko je modeliran znašal za uporabljeni smučko le 5 metrov.

## 7 Sklep

V delu smo razvili in predstavili robota, sposobnega avtonomne vožnje tako po klančini s pomočjo rolk kot tudi s smučmi. Robot pri tem izvaja gibanje na podoben način kot človek, ki zavija na smučkah s poudarjenim stranskim robom s fleksijo in ekstenzijo nog. Izkazalo se je, da je ključen problem v zagotavljanju stabilnosti robota med smučanjem. V ta namen smo razvili metodo zasnovano na točki ničelnega navora. Medtem ko ostale metode na tej osnovi zagotavljajo stabilnost zgolj na predhodno izračunanih trajektorijah oziroma pri neposrednem določevanju točne točke ZMP, pa se pri predlagani metodi točka ZMP giblje prosto po celotnem želenem področju podporne ploskve. Za izračun ZMP-ja smo razvili ustrezne modele smučarja. Ker je potrebno hitro izračunavanje parametrov robota za zagotavljanje delovanja v realnem času, smo razvili tudi poenostavljen model robota v lateralni ravnini.

Zgradili smo tudi virtualno okolje, ki omogoča testiranje tako metode za zagotavljanje stabilnosti kot tudi testiranje algoritma za navigacijo robota na predpisani progi določeno z vratci. Z uporabo virtualnega okolja smo preizkusili algoritme za zaznavanje vratc.

Da bi ugotovili dejanske sposobnosti mehanizma, smo na podlagi modela izvedli analizo stabilnosti mehanizma v odvisnosti od lastnega gibanja, hitrosti v sagitalni ravnini, višine smučanja in lokalnega naklona podlage. Prav tako smo prikazali, da se s pomočjo trupa lahko izvede kompenzacija stabilnosti tako za dolgotrajne spremembe kot tudi za hipne spremembe, ki so posledica trenutnih sprememb odziva smučk.

Nadalje smo zgradili algoritem, ki na podlagi modela človeške navigacije zagotavlja smučanje robota na zakoličeni progi. Metoda temelji na osnovi potencialnih polj, ki pogosto težijo k dejstvu, da se ujamejo v lokalne minimume. V predlagani izvedbi pa se temu izognemo s pomočjo sprotnega definiranja cilja v odvisnosti od ovire, kar onemogoči ujetje v lokalni minimum. Zaradi predlaganega načina gradnje cilja pa je tudi možna uporaba pri neholonomnih robotih, saj je pot vselej zvezna in odvedljiva. V algoritmu smo vgradili virtualni premik vratc, ki je zagotavljal, da so vratca ostala dlje časa v vidnem kotu kamere.

Predlagane algoritme smo implementirali na realnem sistemu, ki smo ga preizkusili tako na smučišču kot na betonski klančini. V slednjem primeru smo namesto smuč

uporabili rolke. Celotno vodenje robota je tako sestavljeno iz primarne naloge, torej zagotavljanja stabilnosti, da se robot ne prevrne. Ker zadano nalogo lahko izvede v določenem področju gibanja, je mogoče izvesti tudi sekundarno nalogo, torej navigacijo med vratci. S tem je gibanje podobno človeškemu, saj tudi človek najprej poskrbi za preprečitev padca in šele nato za smučanje po želeni trajektoriji.

Rezultati testiranj so pokazali, da je predlagani sistem vodenja zagotovil stabilnost v skoraj vseh primerih. Problem se je pojavil zgolj v primeru, ko smo želeli povečati razmerje med silama, ki delujejo na posamezni nogi. Nad določeno mejo je bilo območje dovoljenega gibanja zelo blizu nestabilnosti.

## 7.1 Izvirni prispevki

Na podlagi študij in raziskav, objavljenih v literaturi, menimo, da raziskava, opisana v doktorski disertaciji, med drugim vsebuje sledeče izvirne prispevke k znanosti:

- Realizacija mehanizma humanoidnega robota s sposobnostjo gibanja na smučišču. Zgradili smo robota s sposobnostjo avtonomnega gibanja v dinamičnem okolju.
- Metoda za zagotavljanje ravnotežja v koronalni ravnini na osnovi metode ničelnega navora in ob upoštevanju drugih vplivnih veličin. Razvita metoda omogoča stabilnost robota in ob tem omogoča še dokaj prosto odzivanje robota na sekundarno nalogo.
- Nadgradnja modela človeške navigacije pri kateri je zaznavanje cilja izvedeno s pomočjo robotskega vida. Z nadgradnjo je omogočena uporaba tudi na neholonomnih sistemih. Predlagan način gradnje ciljev in navideznega premika dovoljuje uporabo te metode tudi v primerih uporabe kamer z mnogo ožjim zornim kotom, kot ga ima človek.
- Izvedba eksperimentov z robotom v realnih dinamičnih razmerah na smučišču. Ker robot omogoča ponavljanje zelenih trajektorij, se podatki meritev lahko primerjajo za različno smučarsko opremo oziroma za različne tipe podlage.

## 8 Literatura

- [1] M. Takahashi, T. Yoneyama, Basic ski theory and acceleration during ski turn. *Science and Skiing II*, pp. 307–321, 2001.
- [2] D. Lind, S. P. Sanders. *The Physics of skiing – Second Edition*. Springer, 2004.
- [3] E. Ferreira, S. Tsai, C. Paredis and H. Brown, Control of the Gyrover: a single-wheel gyroscopically stabilized robot. *Advanced Robotics*, 14 (6): 459–475, 2000.
- [4] R. L. Tedrake, Applied Optimal control for dynamically stable legged locomotion, PhD thesis, 2004.
- [5] J. Baltes and N. Hildreth, Adaptive Path Planner for Highly Dynamic Environments. *RoboCup-2000: Robot Soccer World Cup IV*, pp. 76–85. Springer Verlag, Berlin, 2001.
- [6] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya, Trajectory tracking and balance stabilization control of autonomous motorcycles. *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, strani 2583–2589, 2006.
- [7] J. B. Park, J. H. Lee, B. H. Lee, Online Turnover-Free Control for a Mobile Agent with a Terrain Prediction Sensor. *Journal of Field Robotics*, 23(1):59–77, 2006.
- [8] N. H. Getz, J. E. Marsden, Control for an Autonomous Bicycle. *IEEE International Conference on Robotics and Automation*, May, Nagoya, 1995.
- [9] M. Wisse, Skateboards, Bicycles, and Three-dimensional Biped Walking Machines: Velocity-dependent Stability by Means of Lean-to-yaw Coupling, *The International Journal of Robotics Research*, 24(6):417–429, 2005.
- [10] B. R. Fajen, W. H. Warren, S. Termizer, L. P. Kaelbling, A dynamical model of steering, obstacle avoidance, and route selection, *International Journal of Computer Vision*, 54(1–2):13–34, 2003.
- [11] S. Sekhavat, M. Chyba, Nonholonomic deformation of a potential field for motion planning. *IEEE International Conference on Robotics and Automation*, pp. 817–822, 1999.
- [12] P. A. Federolf. *Finite element simulation of a carving snow ski*, PhD thesis, 2005.
- [13] H. G. Tanner, S. Loizou, K. J. Kyriakopoulos, Nonholonomic stabilization with collision avoidance for mobile robots, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1220–1225, 2001.

- [14] J. Borenstein, Y. Koren, "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", *IEEE Journal of Robotics and Automation*, 7(3): 278-288, 1991.
- [15] I. Ulrich, J. Borenstein, Reliable Obstacle Avoidance for Fast Mobile Robots, *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, Belgium, pp. 1572-1577, 1998.
- [16] K. Hasegawa, S. Shimizu, Masatada Yoshizawa, Robotics applied to sports engineering, *Advanced Robotics* 14 (5): 377-379, 2000.
- [17] Y. Tanaka, T. Murakami, Self Sustaining Bicycle Robot with Steering Controller, *Proceedings of 2004 IEEE Advanced Motion Control Conference*, pp. 193-197, 2004.
- [18] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5 (1): 90-98, 1986.
- [19] Diaz-Calderon, A. Kelly, On-Line Stability Margin and Attitude Estimation for Dynamic Articulating Mobile Robots, *International Journal of Robotics Research*, 24 (10): 845-866, 2005.
- [20] C. Chevallereau, E. R. Westervelt, and J. W. Grizzle, Asymptotically Stable Running for a Five-Link, Four-Actuator Bipedal Robot Re-print, *International Journal of Robotics Research*, 24 (6): 431-464, 2005.
- [21] Howe J. *The New Skiing Mechanics*. McIntire Publishing, Waterford, ME, USA. 2001.
- [22] A. Beznos, A. Formalsky, E. Gurfinkel, D. Jicharev, A. Lensky, K. Savitsky, and L. Tchesalin, Control of Autonomous Motion of Two-Wheel Bicycle with Gyroscopic Stabilisation, *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 2670-2675, 1998.
- [23] S. Tsai, E. Ferreira, and C. Paredis, Control of the Gyrover: A Single-Wheel Gyroscopically Stabilized Robot. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99)*, pp. 179-184, 1999.
- [24] M. Vukobratovic, A. A. Frank, D. Juricic, On the stability of biped locomotion. *IEEE Transactions on Biomedical Engineering*, 17 (1), 1970.
- [25] M. Vukobratovic, D. Juricic, Contribution to synthesis of biped gait, *IEEE Transactions on Biomedical Engineering*, 16(1):1-6, 1969.
- [26] M. Vukobratovic, J. Stepanenko, On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15 (1-2): 1-37, 1972.
- [27] M. Vukobratovic, B. Borovac, Zero-Moment Point – Thirty Five Years of its Life, *International Journal of Humanoid Robotics*, 1 (1): 157-173, 2004.
- [28] T. Arakawa, T. Fukuda, Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of GA, EP layers, *In*



*Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pp.211–216, 1997.

[29] Napoleon, S. Nakaura, M. Sampei, Balance control analysis of humanoid robot based on zmp feedback control, *In Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and System*, 2002.

[30] Q. Huang, K. Li, Y. Nakamura, Humanoid walk control with feedforward dynamic pattern and feedback sensory reflection, *In Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2001.

[31] J. Yamaguchi, E. Soga, S. Inoue, A. Takanishi, Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking, *In Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, 1999.

[32] P. A. Federolf, *Finite element simulation of a carving snow ski*, Phd. Thesis, Swiss federal institut of technology, Zurich, 2005.

[33] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments. *International Journal of Robotics Research*, 25 (5/6): 449–483, 2006.

[34] S. C. Peters and K. Iagnemma. An Analysis of Rollover Stability Measurement for High-Speed Mobile Robots, *In Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, pp. 3711–3716, 2006

[35] T. Geng, B. Porr, F. Wörgötter, Fast Biped Walking with A Sensor-driven Neuronal Controller and Real-time Online Learning, *The International Journal of Robotics Research*, 25 (3): 243–259, 2006.

[36] W. H. Huang, B. R. Fajen, J. R. Fink, W. H. Warren, Visual navigation and obstacle avoidance using a steering potential function, *Robotics and Autonomous Systems*, 54: 288–299, 2006.

[37] J. E. Pratt and R. Tedrake, Velocity-Based Stability Margins for Fast Bipedal Walking, *In Proceedings of the First Ruperto Carola Symposium on Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*, Heidelberg, Germany, September 2005.

[38] R. M. Murray, S. S. Sastry, Nonholonomic motion planning: Steering using sinusoids, *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[39] B. Fajen and W. Warren, Behavioral Dynamics of Steering, Obstacle Avoidance, and Route Selection, *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):343–362, 2003.

[40] T. Yoneyama, H. Kagawa, Study on the effective turn motion using a ski robot, *The Engineering of Sport*, 4: 463–469, 2002.

- [41] A. Dasgupta and Y. Nakamura, Making feasible walking motion of humanoid robots from human motion capture data, *In Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 1044–1049, Detroit, USA, 1999.
- [42] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, H. Inoue, A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomous Robots*, 12 (1):71–82, 2002.
- [43] A. Goswami, Postural stability of biped robots and the foot-rotation indicator (FRI) point, *The International Journal of Robotics Research*, 18(6):523–533, 1999.
- [44] T. Mita, T. Yamaguchi, T. Kashiwase, and T. Kawase, Realization of a high speed biped using modern control theory, *International Journal of Control*, 40(1): 107–119, 1984.
- [45] G. Capi, Y. Nasu, L. Barolli, K. Mitobe, Real time gait generation for autonomous humanoid robots: A case study for walking, *Robotics and Autonomous Systems*, 42(2):107–116, 2003.
- [46] M. Zarrugh and C. Radcliffe, Computer generation of human gait kinematics, *Journal of biomechanics*, vol. 12, no. 2, pp. 99–111, 1979.
- [47] S.-H. Lee and J.-B. Song, Acceleration estimator for low-velocity and low-acceleration regions based on encoder position data," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 1, pp. 58–64, 2001.
- [48] R. H. Brown, S. C. Schneider, and M. G. Mulligan, Analysis of algorithms for velocity estimation from discrete position versus time data," *IEEE Transactions on Industrial Electronics*, 39(1):11–19, 1992.
- [49] J. Y. Kim, I. W. Park, Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor, *Journal of Intelligent and Robotic Systems*, 48(4):457–484, 2007.
- [50] J. Kim, W. K. Chung, Y. Youm B.H. Lee, Real-time ZMP compensation method using motion for mobile manipulators, *IEEE International Conference on Robotics and Automation*, 2:1967–1972, 2002.
- [51] S. Kajita, K. Tani, Study of dynamic biped locomotion on rugged terrain derivation and application of the linear inverted pendulum mode," in *Proceedings 1991 IEEE International Conference on Robotics and Automation*, pp. 1405–1411, Sacramento, California, USA, 1991.
- [52] J.-F. Lalonde, N. Vandapel, M. Huber, and M. Hebert, Natural terrain classification using three-dimensional ladar data for ground robot mobility, *Journal of Field Robotics*, 2006.
- [53] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point," *IEEE International Conference on Robotics and Automation (ICRA)*

2003), pp. 1620–1626, May 2003.

[54] J. Borenstein, Y. Koren, “Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation”, *Proceedings of the IEEE International Conference on Robotics and Automation*, California, April 1991.

[55] S. S. Ge, Y. J. Cui, Path Planning for Mobile Robots Using New Potential Functions, *Proceedings of 3rd Asian Control Conference*, July 4–7, 2000.

[56] H. Moravec, Certainty Grids for Mobile Robots, *NASA/JPL Space Telerobotics Workshop*, pp. 307–312, January 1987.

[57] I. Ulrich, J. Borenstein, VFH+: reliable obstacle avoidance for fast mobile robots, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1572–1577, Leuven, Belgium, May 1998.

[58] V. J. Lumelsky and A.A. Stepanov, Path planning strategies for a point mobile automaton moving amidst obstacles of arbitrary shape, *Algorithmica* 2:403–430, 1987.

## **Izjava**

Izjavljam, da sem doktorsko disertacijo izdelal samostojno pod mentorstvom prof. dr. Jadrana Lenarčiča, univ. dipl. inž. el. in dr. Bojana Nemca, univ. dipl. inž. el. Izkazano pomoč ostalih sodelavcev sem navedel v zahvali.

V Ljubljani, 15. 1. 2009

Leon Lahajnar