

Univerza v Ljubljani

Fakulteta za elektrotehniko

Rok Rezar

**Posnemanje elementov človekove hoje na
humanoidnem robotu**

Magistrsko delo

Mentor: prof. dr. Matjaž Mihelj
So mentor: izr. prof. dr. Aleš Ude

Ljubljana, 2015

Zahvala

Želim se zahvaliti staršema, ki sta mi omogočila šolanje in mi vedno stala ob strani ter Barbari, za spodbudne besede in lektoriranje.

Zahvalil bi se tudi univ. dipl. ing. Roku Vugi, ki mi je pomagal z nasveti pri samem delu in so mentorju izr. prof. dr. Alešu Udetu, oba z IJS, ter mentorju s Fakultete za elektrotehniko, prof. dr. Matjažu Mihelju.

Vsebina

1 Uvod	13
2 Povezave v eksperimentalni sistem	19
2.1 Humanoidni roboti	19
2.2 Fujitsu Hoap-3	19
2.3 Microsoft Kinect.....	20
3 Posnemanje človeških gibov z upoštevanjem stabilnosti in izogibanja trkom	23
3.1 Implementacija sistema za izogibanje trkom	23
3.1.1 Funkcija za izračun najmanjše razdalje	24
3.1.2 Sklepi in segmenti robota	26
3.2 Ohranjanje stabilnosti s pomočjo točke ničelnega momenta	28
4 Posnemanje elementov človekove hoje	33
4.1 Trajektorije hoje robota	33
4.2 Prostorsko gibanje robota	34
4.2.1 Posnemanje premikanja človeka v prostoru	34
4.2.2 Posnemanje elementov hoje človeka	35
4.3 Rezultati.....	36
5 Zaključek	39
Literatura	41

Seznam uporabljenih simbolov

ZMP – točka ničelnega momenta (angl. Zero Moment Point) je točka na podlagi, v kateri je vsota vseh horizontalnih navorov enaka nič.

Podporni poligon – območje na podlagi, ki zajema kontaktne točke robota s podlago in površino med njimi. Če je projekcija težišča robota znotraj podpornega poligona, je robot statično stabilen oziroma ne bo padel, če se takoj ustavi.

Povzetek

Cilj naloge je bil razviti sistem, ki robotu omogoča posnemanje elementov človekove hoje (poglavje štiri), ki jo zajamemo z globinsko kamero Microsoft Kinect. Obravnavali smo tudi problem preprečevanja trkov med udi humanoidnega robota.

V drugem poglavju smo obravnavali glavni komponenti, ki smo ju uporabili v našem delu. To sta humanoidni robot HOAP-3 in Microsoft Kinect.

Humanoidni robot je robot, katerega zgradba je podobna človekovemu telesu. Humanoidni roboti praviloma zajemajo trup z glavo, dve roki in dve nogi. Med najpomembnejše elemente, ki opredeljujejo zmogljivost humanoidnih robotov, spadajo 1) uporabljeni senzorji in aktuatorji, 2) sposobnost zagotavljanja energije za delovanje, 3) samostojno učenje kompleksnih nalog brez ali z zunanjim posredovanjem ter s prilagajanjem strategij učenja glede na okolico in 4) sposobnost varnega delovanja v človekovem okolju brez ogrožanja varnosti ljudi v okolici.

Fujitsu HOAP-3 je humanoidni robot, ki ga odlikuje kompaktnost, nizka teža ter enostavno upravljanje. Ima 28 sklepov z električnimi servomotorji, dve kameri, ki služita za vid, mikrofona, triosni žiroskop, pospeškomer, senzorji sile na podplatih in v zapestjih, merilnik razdalje, vgrajen pa ima tudi modul za razpoznavanje in sintezo govora.

Microsoft Kinect je barvno-globinska kamera, s katero lahko zaznamo in izračunamo gibanje ljudi v prostoru. Kinect omogoča sledenje do šestim ljudem in do 25 sklepov na osebo. V našem delu smo Kinect uporabili za zajem gibanja demonstratorja.

V tretjem poglavju sledi posnemanje gibanja rok človeka, preprečevanje trkov med segmenti robota ter opis kontrole stabilnosti.

Demonstratorjeva gibanja, ki jih zaznamo s Kinectom, lahko prenesemo na humanoidne robote, saj se ti lahko gibljejo na podoben način kot ljudje. Za uspešno posnemanje pa pogosto ni dovolj, da demonstratorjeve gibe direktno prenesemo s človeka na robota, temveč moramo poskrbeti, da ne prihaja do samotrkov in da robot ostane stabilen. V tej nalogi smo razvili metodo izogibanja samotrkom s pomočjo metode prioritete nalog. Primarna naloga je preprečevanje trkov, sekundarna naloga pa je gibanje demonstratorja posneto s Kinectom.

Če želimo, da robot uspešno posnema gibanje človeka, mora med posnemanjem demonstratorjevega gibanja ostati stabilen. Robot je najbolj stabilen, ko je točka ničelnega navora (angl. zero moment point, ZMP) v centru podpornega poligona, ki obkroža nogi robota. Pri posnemanju elementov človekove hoje mora primarna naloga skrbeti za nadzor stabilnosti, medtem ko se sekundarna naloga nanaša na posnemanje gibanja. ZMP je odvisen od gibanja centra težišča robota. Vse to pa je pomembno za načrtovanje trajektorij hoje, ki jih v našem delu uporabljamo v poglavju štiri.

Ključne besede: humanoidna robotika, zaznavanje gibanja človeka, posnemanje elementov hoje, preprečevanje trkov

Abstract

The main goal of this thesis was to create a system that enables a humanoid robot to imitate human walking (chapter four), which is captured by an RGB-D camera like for example Microsoft Kinect. We also considered how to prevent collisions between the limbs of a humanoid robot during imitation.

In the second chapter we discuss the main components that are used in our work. These are humanoid robot HOAP-3 and Microsoft Kinect.

Humanoid robot is a robot whose structure is similar to the structure of a human body. Humanoid robots are typically comprised of the torso and the head, two arms, and two legs. Among the most important elements that define the capabilities of humanoid robots are: 1) sensors and actuators used to build the robot, 2) ability to provide energy for autonomous operation, 3) ability to effectively learn complex tasks with or without external teacher intervention, including movement adaptation to the environment, 4) ability to securely operate in human environments, without compromising safety of people in the area.

Fujitsu HOAP-3 is a humanoid robot distinguished by its compactness, low weight, and easy operation. It has 28 joints with electric servomotors, two cameras to provide visual information, microphone, three axis gyroscope, accelerometer, force sensors on the soles of the feet, distance meter, and a built-in module for recognition and speech synthesis.

Microsoft Kinect is an RGB-D camera that can be applied to detect and follow human motion in 3-D space. Range images are essential to enable Kinect to track up to six people and up to 25 joints per person. In our work we used Kinect to capture the motion of a human demonstrator.

In the third chapter follows the human hands movement imitation, preventing collisions between robot segments and description of stability control.

Due to its similarity to people, a humanoid robot can imitate human motion captured by Kinect. For a more effective movement imitation, two additional problems need to be solved: collision avoidance and stability of the robot. In this thesis collision avoidance was implemented using the task priority approach. In this context, the primary task is to prevent collisions between the robot body parts, The secondary task is to imitate the movement acquired by Kinect.

For effective imitation of human walking, the robot must remain stable while imitating the demonstrator's motion. The robot is most stable when the zero moment point (ZMP) is in the center of the support polygon that surrounds the legs of the robot. Thus when imitating human body motion while walking, the primary task must ensure the stability of the robot, while the secondary task refers to the imitation of the demonstrator's movement. ZMP depends on the center of gravity of the robot. All this is important for planning walking trajectories that are used in our work in chapter four.

Key words: Humanoid robotics, Human motion detection, Walking elements imitation, Collision avoidance

1 Uvod

Humanoidni roboti lahko zaradi svoje podobnosti ljudem izvajajo podobne gibe kot človek. Zato se lahko učijo novih motoričnih gibov s pomočjo opazovanja človekovega gibanja in prenosa človekovih gibov na svojo kinematiko in dinamiko. Cilj magistrskega dela je razvoj metod za posnemanje človekovega gibanja s humanoidnimi roboti, pri čemer bomo upoštevali razlike med kinematičnimi in dinamičnimi modeli človeka in robota.

Prvega humanoidnega, ki lahko hodi in se povzpne tudi po stopnicah je napravilo podjetje Honda [1]. Da bi se izognili potrebi po teleoperaciji in zaznavanju obnašanja so začeli delati na pristopu učenja. Ideja o učenju s posnemanjem se je pojavila v osemdesetih letih prejšnjega stoletja, saj naj bi bil to obetaven pristop, kako nadomestiti ročno programiranje robotov. Pomemben element pri učenju je iskanje novih dejanj oziroma možnosti, da dobimo optimalno rešitev za zadano nalogo. Posnemanje zelo zmanjša prostor, ki ga je potrebno raziskati oziroma poznati.

Samo učenje poteka tako, da se cilj najprej doseže z ročno napravljenimi gibi. Gibanje robota je pri tem razdeljeno na podcilje in glavne cilje – premike s katerimi robot te cilje doseže. Podcilji so lahko robotove orientacije in pozicije prijemala, ki jih demonstrator določi ročno. Nato dane naloge razdelijo na zaporedje le teh. Zaradi negotovosti v okolju moramo ob takšnih zaporedjih upoštevati, da je nalogo možno izvesti na več načinov [2].

Roboti, ki so ljudem namenjeni za pomoč pri vsakdanjih opravilih, morajo biti prilagodljivi, saj so okolja, v katerih mora izvesti želeno nalogo, od primera do primera drugačna. Robot se mora pri tem prilagajati različnim konfiguracijam delovnega prostora. Robot se mora biti sposoben premikati tudi po spreminjajočem

se okolju in pri tem upravljati z veliko objekti [2]. Poleg tega je pomembno, da uporabnik robota zlahka nauči, kaj mora napraviti. Pri robotskih sistemih, ki jih učimo s posnemanjem, je zaželeno, da robota naučimo opravil v enem samem poskusu, saj je za končnega uporabnika demonstracija večih podobnih uprizoritev naloge oziroma opravila lahko nadležna in zamudna.

Dillmann [3] meni, da je pristop programiranja na podlagi posnemanja primeren za hitro pridobivanje informacij za izvršitev neke naloge. Kot primer je navedel nalogo zlaganja posode v hladilnik in iz hladilnika. Za uspešno izvedbo takšnih nalog potrebuje humanoidni robot kompleksne senzorske sisteme za zaznavo in obdelavo ukazov, s katerimi bi iz opazovanja zbral čim več potrebnih informacij za uspešno izvedbo naloge. V članku je prikazano, kako se lahko robot odziva na podlagi posnemanj demonstratorja. Dillmann se osredotoča na prepoznavanje prikazanih opravil in ga pri tem ne zanima učenje informacij, kot je na primer pravilna konfiguracija sklepov za določen prijem, saj naj bi takšna znanja robotski sistem že vključeval. Učenje izvaja na ravni nalog, kot na primer, da je potrebno iz hladilnika vzeti točno določeno steklenico. V članku obravnava predvsem razumevanje nalog in človekovih namer, učenje in predstavitev prikazane naloge in izvajanje naučenih nalog.

Senzorski sistemi za opazovanje človekovih dejavnosti morajo biti sposobni zaznati operacije, kot so odpiranje vrat in sledenje predmetom. Med senzorje, ki jih lahko uporabljamo za zajemanje gibanja, spadajo podatkovne rokavice, magnetni sledilniki, strojni vid in drugo. Nato sledi preslikava podatkov iz senzorskega sistema na robota. Glavni problem pri tem so razlike v zgradbi robotskega mehanizma in človekovega telesa. Zato prenos ne more biti neposreden, temveč je potrebno napraviti ustrezno transformacijo. Sledi sklepanje o tem, katero dejanje mora biti izvedeno. Ker avtonomno pridobivanje informacij o uporabnikovi nameri ni vedno zagotovljeno, je potrebno upoštevati tudi morebitne verbalne razlage in ukaze uporabnika oziroma demonstratorja.

Za izvedbo eksperimentov o učenju gospodinjskih opravil so Asfour in kolegi [4] robotovo delovno okolje, torej kuhinjo, preuredili v center za usposabljanje kjer robot upravlja s predmeti, ki jih najdemo v kuhinji. Kuhinja deluje kot dodatna

razširitev senzorjev in aktuatorjev, tako da robot pridobi čim več informacij, preko katerih se nauči opravljati zahtevane naloge [3].

Argall in kolegi [5] so predstavili celovit pregled robotskega učenja s posnemanjem. Pri tem so poudarili, da je problem vseh takšnih pristopov preslikava med dejanjem in sistemom.

Demonstrator se mora najprej odločiti, kako bo problem prikazal robotu. Demonstrator lahko na primer prikaže nalogo, kar je tudi največkrat uporabljen pristop, lahko pa uporabi tudi pristop teleoperacije ali ročno napisane algoritme za vodenje in simulacijo. Pristop teleoperacije je redko uporabljen pri robotih z velikim številom prostostnih stopenj (humanoidni roboti), saj je njihove kompleksne gibe težko upravljati s krmilno palico. Izbira je torej odvisna od tega, kako kompleksna sta robot in naloga.

Sledi izbira strategije zagotavljanja podatkov. Ena od možnosti je, da demonstrator najprej izvede serijo operacij, učenje pa postane aktivno šele, ko je zadnja operacija zaključena. Lahko pa robot z učenjem nadgrajuje nabor podatkov med izvajanjem naloge.

Uspešnost robota, ki posnema gibanje je odvisna od kakovosti podatkov, ki jih robot pridobi od demonstratorja. V veliko primerih demonstracija ni usklajena z zmožnostmi robota, saj demonstrator na primer ne more prikazati tako natančnih in hitrih operacij, kot naj bi jih robot izvedel. Zato je v takšnih primerih potrebno vključiti tudi verbalne razlage. Sam nabor podatkov za učenje s posnemanjem je pogosto sestavljen iz parov mirovanje – izvedba, ko demonstrator prikaže želeno nalogo.

Kako te pare beležimo je odvisno od uporabljenega pristopa. Senzorji so lahko nameščeni na robota ki ga vodimo s pomočjo teleoperacije, medtem pa beleži svoja gibanja. Ta metoda je bolj splošna, vendar so izvedena gibanja manj kvalitetna kot pri direktnem zajemanju gibanja demonstratorja. Pri tem so zajeta gibanja bolj naravna, vendar pa je ta pristop bolj zapleten in drag, saj je zanj potrebna ali posebna obleka s senzorji ali pa prostor, ki je opremljen s kamerami in drugimi senzorskimi sistemi [5].

V našem delu najprej obravnavamo osnovno zgradbo humanoidnih robotov, vrste humanoidnih robotov in osnovne lastnosti le-teh. Humanoidni robot je namreč robot, katerega zgradba temelji na obliki človekovega telesa in praviloma zajemajo trup z glavo, dve roki in dve nogi. Pomembna je tudi sposobnost za zagotavljanje energije za avtonomno delovanje, saj roboti ne morejo biti stalno priključeni na izvore energije. Za uspešno delovanje v naravnih okoljih se morajo roboti samostojno učiti kompleksnih nalog. Naučena znanja morajo tudi prilagajati svoji trenutni okolici. Pri tem moramo zagotoviti varno delovanje v človekovih okoljih, saj humanoidni roboti pogosto delujejo v stiku s človekom [6].

V nadaljevanju opišemo humanoidnega robota Hoap-3 in barvno-globinsko kamero Microsoft Kinect, ki smo ju uporabili za eksperimentalni preizkus razvitih metod, to je posnemanje človeških gibov ob upoštevanju stabilnosti in izogibanja trkom. Za posnemanje gibanja demonstratorja moramo prenesti gibanje sklepov in ne na primer kartezičnih gibov dlani. Kinect poda gibanje demonstratorja v obliki pozicij in rotacij koordinatnih sistemov, ki se navezujejo na človekove sklepe. Človekovi krogelni sklepi, kot sta na primer rama in kolk, so na humanoidnih robotih modelirani s tremi zaporednimi rotacijami. Zato moramo rotacijske matrike, s katerimi Kinect izračuna gibanje krogelnih sklepov, preslikati v zaporedja Eulerjevih kotov, ki ustrezajo trem rotacijskim prostostnim stopnjam na robotu [7].

Pri posnemanju gibanja moramo rešiti naslednja dva problema: samotrki in stabilnost. V tem delu smo ju rešili z uporabo metode prioritete nalog. Ker imajo humanoidni roboti kompleksno kinematično zgradbo, se lahko med izvajanjem nalog hitro zgodi, da dva segmenta med seboj trčita, kar lahko privede do poškodb. To pride še posebej do izraza pri posnemanju človekovega gibanja. Zato smo v tej nalogi implementirali metodo izogibanja trkom na osnovi metode prioritete nalog [8]. Izogibanje trkom bi lahko napravili tudi z določanjem potencialnega polja sil, v katerem ovire na robota delujejo odbojno, cilj pa na vrh robota deluje privlačno [9]. Oviram se lahko izogibamo tudi s kinematičnim vodenjem, pri čemer metoda stremi k dodeljevanju komponent gibanja proč od ovire za vsako točko robota, ki je blizu oviri [10]. V našem delu smo predstavili celotno implementacijo izogibanja trkom na podlagi prioritete nalog od spoznavanja s samo problematiko pa vse do končne implementacije na robotu.

Nadalje smo obravnavali še enega od najpomembnejših elementov pri posnemanju gibanja človeka, to je robotova stabilnost. Različne kinematične in dinamične lastnosti človeka in robota onemogočajo natančen prenos človekovega gibanja na robota, zato nikoli ne dobimo povsem enakega gibanja. Če želimo, da robot med posnemanjem gibanja človeka ne pade, moramo med posnemanjem upoštevati stabilnost robota. Za določanje in vzdrževanje stabilnosti obstajajo različne metode, med katerimi se najpogostejše uporablja vodenje s pomočjo točke ničelnega momenta (angl. zero moment point, ZMP). Ta pristop je prvi razvil Miomir Vukobratović [11] in ga uporabil za vodenje antropomorfnih mehanizmov [12]. Takanishi in sodelavci [13] so razvili model, ki uporablja vnaprej določene trajektorije nog, medtem ko ZMP spreminjajo s kompenzacijo mase. V preteklosti je bilo razvitih več metod za stabilno vodenje humanoidnih robotov s pomočjo točke ZMP, med drugimi metodi, ki so ju predlagali Kajita s sodelavci [14] in Takanishi s sodelavci [15].

V našem delu sledimo načinu, ki ga je predlagal Sugihara s sodelavci [16]. Metoda temelji na enostavnem modelu inverznega nihala, kjer je celotna dinamika robota predstavljena z dinamiko težišča, ki ga povežemo s točko ZMP. Obstaja namreč več načinov načrtovanja trajektorij robotske hoje [13] [14] [15]. Vse te metode temeljijo na upoštevanju kriterija ZMP. V tej nalogi se nismo ukvarjali s samim načrtovanjem trajektorij hoje, saj na robotu Hoap-3 že proizvajalec zagotavlja več trajektorij, ki so stabilne in upoštevajo kriterij ZMP.

Naš cilj je bil združiti posnemanje človeka in možnost premikanja robota po prostoru, ne da bi med tem prihajalo do trkov med segmenti robota. Najbolje bi bilo, če bi lahko hkrati z gibanjem zgornjega dela telesa neposredno na robota prenesli tudi gibanje sklepov nog med hojo. Vuga in sodelavci so to že izvedli, vendar je tak način hoje neroden in se ne obnese najbolje [17] [18]. Implementacija hoje je bila namreč napravljena z neposrednim posnemanjem gibov tako kot za ostale dele telesa. V tem delu smo predstavili rešitev, pri kateri v zgornjem delu telesa robot posnema demonstratorjevo gibanje neposredno, s spodnjim delom pa mu sledi tako, da prilagaja vnaprej načrtovane vzorce hoje.

2 Povezave v eksperimentalni sistem

2.1 Humanoidni roboti

Humanoidni robot je robot, katerega zgradba temelji na obliki človekovega telesa. Humanoidni roboti praviloma zajemajo trup z glavo, dve roki in dve nogi. Nekatere posebne izvedbe humanoidnih robotov modelirajo zgolj zgornji del človekovega telesa od pasu navzgor. Taki humanoidni roboti so praviloma statično nameščeni na podstavku ali pa na mobilni ploščadi, ki zagotavlja robotovo mobilnost v prostoru. Nekatere glave humanoidnega robota so opremljene z dvema kamerama, ki služita za vizualno zaznavanje okolja in dvema ali večimi mikrofoni za zajemanje zvoka [6]. Nekateri humanoidni roboti so opremljeni tudi s senzorji vonja in usti. Humanoidni roboti, ki poleg funkcionalnosti in mehanske podobnosti modelirajo tudi estetsko plat človeka, se imenujejo androidi [19]. V nasprotju s klasičnimi industrijskimi roboti so humanoidni roboti praviloma avtonomni in se med izvajanjem zadane naloge prilagajajo spremembam v okolju. Pri humanoidnih robotih je zato pomembna sposobnost zagotavljanja energije za avtonomno delovanje, saj ne morejo biti stalno priključeni na izvore energije. Pomembno je tudi samostojno učenje kompleksnih nalog brez ali z zunanjim posredovanjem ter prilagajanje strategij učenja glede na okolico. Zagotoviti moramo tudi varno delovanje v človekovem okolju, saj humanoidni roboti pogosto delujejo v stiku s človekom [6].

2.2 Fujitsu Hoap-3

Fujitsu Hoap-3 je humanoidni robot, ki ga odlikuje kompaktnost, nizka teža ter enostavno upravljanje. Robot je visok 60 cm in težak 8.8 kg. Ima 28 sklepov z električnimi servomotorji, dve kameri, ki služita za vid, mikrofoni, triosni žiroskop,

pospeškometer, senzor sile na podplatih in v zapestju, merilnik razdalje, vgrajen pa ima tudi modul za razpoznavanje in sintezo govora. Robota lahko vodimo žično ali pa brezžično preko WLAN omrežja. Operacijski sistem robota je GNU Linux. Hoap-3 ima 28 prostostnih stopenj, šest v vsaki nogi in roki, eno v trupu in tri v vratu. V nahrbtniku humanoidnega robota se poleg servo krmilnikov in baterij nahaja tudi interni računalnik, ki skrbi za vodenje motorjev v sklepih robota in osnovno obdelavo senzorskih informacij, ki jih pridobi iz kamer, mikrofona, merilnika razdalje, žiroskopa, pospeškometra ter senzorjev sil. Robot komunicira s stacionarnim računalnikom, na katerem se vrši programiranje robota in spremljanje njegovih aktivnosti. Za zahtevnejšo obdelavo senzorskih podatkov lahko interni računalnik humanoidnega robota pošilja stacionarnemu računalniku tudi neobdelane senzorske podatke.

2.3 Microsoft Kinect

Microsoft Kinect, v nadaljevanju Kinect, je naprava, ki lahko zazna gibanje človeka v prostoru in ustvari sliko na zaslonu. Prav tako omogoča izdelavo prostoročnega krmiljenja elektronskih naprav. Je naprava za globinsko zaznavo, ki omogoča sledenje šestim ljudem in 25 sklepom na osebo. Obstajata dve različici, pri čemer smo mi uporabili novejšo, kjer so zaznane pozicije pravilnejše, stabilnejše, natančnejše, razpon sledenja pa je širši.

Kinect je sestavljen iz treh pomembnih delov, ki delujejo skupaj, da naprava zazna gibanje v prostoru in ustvari sliko na zaslonu. Ti deli so RGB barvna video kamera, senzor globine in mikrofoni. Kamera zazna rdečo, zeleno in modro komponento barve, iz česar lahko izračunamo tip postave in poteze obraza. Senzor globine je sestavljen iz infrardečega laserskega projektorja in enobarvnega CMOS senzorja, ki pomaga ustvariti 3D posnetke osvetljenega prostora (infrardeča kamera). Senzor globine izmeri razdaljo vsake točke človeka do senzorja s prenosom infrardeče svetlobe in z merjenjem pretečenega časa po tem, ko se svetloba odbije od predmetov. Mikrofoni so štirje mikrofoni, ki lahko izločijo človekov glas od zvokov iz ozadja, s tem pa omogoča zaznavo lokacije in gibanja človeka, kar omogoča dodaten nadzor. Ko te komponente združimo, lahko naprava sledi 48

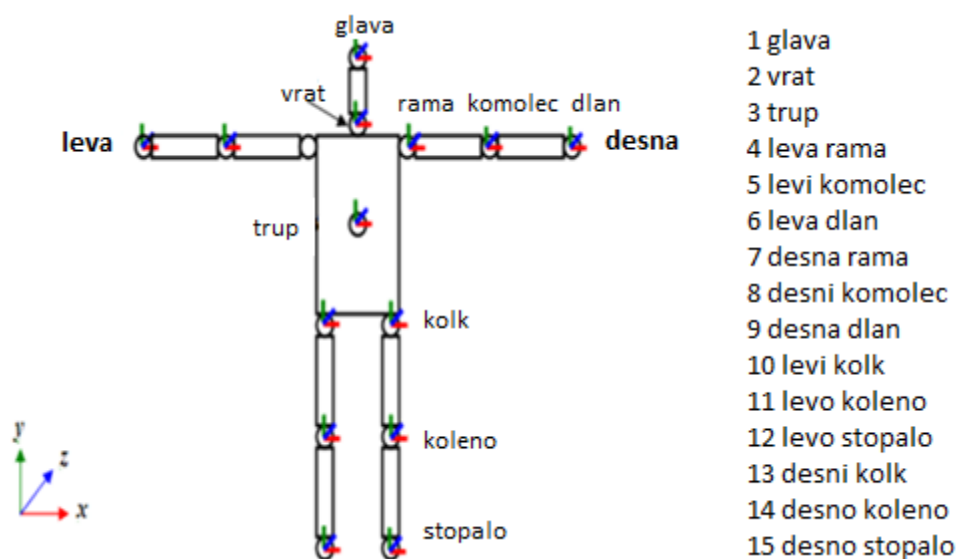
različnim točkam na vsakem človeku s frekvenco 30 Hz. Ko združimo strojni in programski del, lahko Kinect zaznava gibanje človeka v vidnem polju. Zaradi enostavnosti za uporabo in cenovne ugodnosti se za zaznavanje tridimenzionalnih slik in človekovega gibanja v robotiki vse več uporablja Kinect.

Kinect mora biti za uspešno zaznavanje gibanja človeka nameščen med 0.6 m in 1.8 m nad tlemi, medtem ko se lahko človek giblje na razdalji med 0.8 m in 3.5 m do Kinecta. Če želimo Kinect uporabiti na računalniku z operacijskim sistemom Windows, moramo k napravi dodati še Kinect Adapter [20].



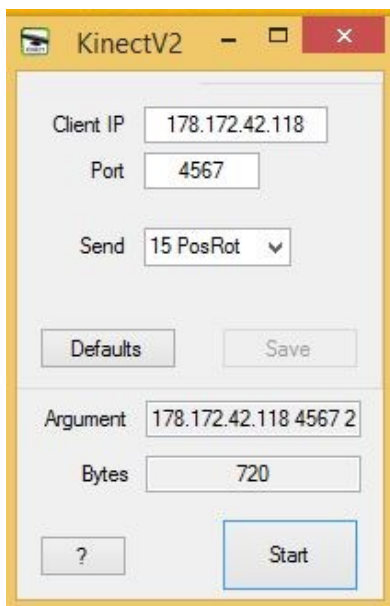
Slika 2.1: Microsoft Kinect

Kinect povežemo z računalnikom, ki zajema podatke s senzorja. Za zajem podatkov s Kinecta smo uporabili vmesnik, ki je sposoben zaznavati pozicije sklepov, ki so prikazani na Sliki 2.2. Za večino sklepov dobimo tudi pripadajoče rotacijske matrike [18].



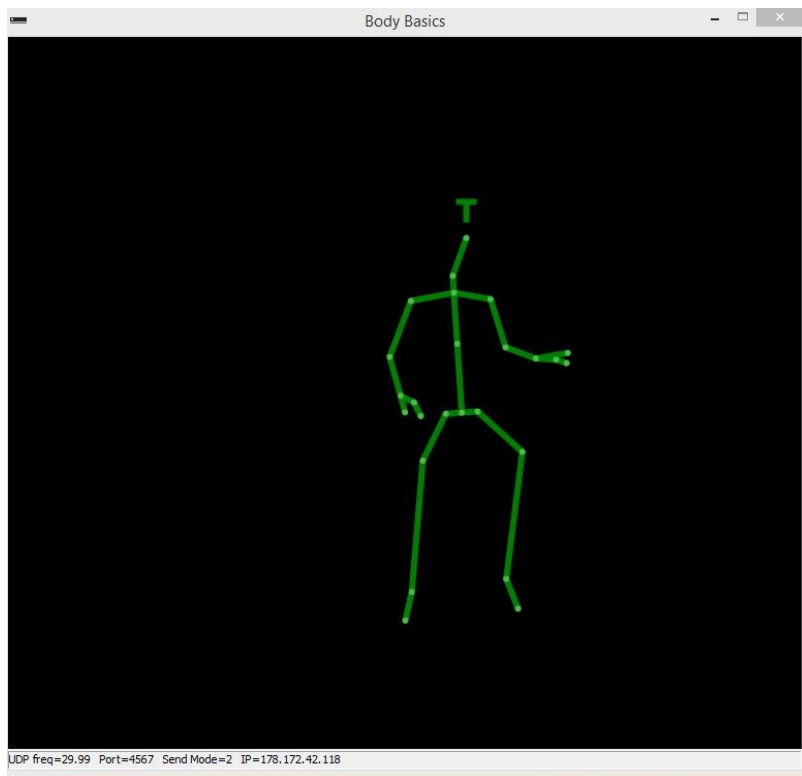
Slika 2.2: Sklepi dobljeni iz Kinecta

Ob zagonu vmesnika najprej nastavimo vrata (port) in IP številko računalnika, na katerem teče algoritem za posnemanje elementov hoje človeka (Slika 2.3).



Slika 2.3: Povezava Kinecta

Ko je programska oprema pravilno nameščena, lahko na zaslonu prikažemo telo demonstratorja in sklepe (Slika 2.4), ki jih potrebujemo za zajemanje gibanja.



Slika 2.4: Telo demonstratorja

3 Posnemanje človeških gibov z upoštevanjem stabilnosti in izogibanja trkom

Pri posnemanju gibanja demonstratorja z robotom moramo prenesti gibanje sklepov in ne na primer kartezičnih gibov dlani. Kinect poda gibanje demonstratorja v obliki pozicij in rotacij koordinatnih sistemov, ki se navezujejo na človekove sklepe (Slika 2.2). Človekovi krogelni sklepi, kot sta rama in kolk, so na humanoidnih robotih implementirani s tremi zaporednimi rotacijami. Zato moramo rotacijske matrike, s katerimi Kinect izračuna gibanje krogelnih sklepov, preslikati v zaporedja Eulerjevih kotov, ki ustrezajo trem rotacijskim prostostnim stopnjam na robotu.

Pri posnemanju gibanja moramo rešiti naslednja dva problema: samotrki in stabilnost. V tem delu smo ju rešili z uporabo metode prioritete nalog.

Izogibanje trkom lahko napravimo tudi z določanjem potencialnega polja sil, v katerem ovire na robota delujejo odbojno, cilj pa na vrh robota deluje privlačno [9]. Oviram se lahko izogibamo tudi s kinematičnim vodenjem. Metoda stremi k dodeljevanju komponent gibanja proč od ovire za vsako točko robota, ki je blizu oviri. Poudarek je na izogibanju oviram [10].

3.1 Implementacija sistema za izogibanje trkom

Humanoidni roboti imajo kompleksno kinematično zgradbo. Med izvajanjem nalog se lahko hitro zgodi, da dva segmenta trčita med seboj, kar lahko privede do poškodb. To pride še posebej do izraza pri posnemanju človekovega gibanja, saj so človekovi gibi nepredvidljivi. Zato smo v tej nalogi implementirali metodo izogibanja trkom na osnovi metode prioritete nalog [8]. Primarno nalogo, torej preprečevanje trkov, predstavlja hitrost točke na posameznem udu, ki je najbližje

oviri – nekemu drugemu segmentu. Sekundarna naloga pa je gibanje posneto s Kinectom, ki se preslika v ničelni prostor primarne naloge, kar pomeni, da nanjo ne vpliva. Ali je sekundarna naloga aktivirana ali ne, je po predlagani metodi odvisno od razdalje uda do ovire, oziroma drugega segmenta.

Vodenje po metodi prioritete nalog lahko realiziramo s pomočjo Jacobijeve matrike J_p in pripadajoče projekcijske matrike v ničelni prostor N_p . V enačbi (3.1) prvi seštevanec predstavlja primarno nalogo, drugi seštevanec pa sekundarno nalogo.

$$dq = J_p^{-1} * dp + N_p * J_s^{-1} * ds \quad (3.1)$$

Enačbo (3.1) v našem pristopu zapišemo kot enačbo (3.2). Enačbi (3.1) dodamo le faktor α , saj originalna metoda [21] tega ne vsebuje, vendar je za našo aplikacijo faktor α ključnega pomena. Če faktorja α ne bi imeli, bi bila sekundarna naloga vedno omejena, kar bi pomenilo, da nikoli ne bi mogli prenesti celotnega gibanja iz Kinecta na robota, tudi če bi bili vsi udi daleč stran drug od drugega

$$\dot{q} = \alpha * J^+ * \dot{x} + (I - \alpha * J^+ * J) * \dot{q}_s \quad (3.2)$$

V enačbi (3.2) prvi seštevanec predstavlja primarno, drugi seštevanec pa sekundarno nalogo. V enačbi (3.2) predstavlja α aktiviranost primarne ali sekundarne naloge. Če je $\alpha = 1$, to pomeni, da je na primer roka robota prišla na najmanjšo razdaljo do drugega segmenta in je vključena primarna naloga – izogibanje trkom. Z večanjem razdalje roke do drugih segmentov faktor α pada, hitrost premikanja roke pa se poveča – vključena je samo sekundarna naloga.

3.1.1 Funkcija za izračun najmanjše razdalje

Posamezne segmente robota modeliramo z daljicami. Če daljici dodamo debelino pa dobimo valj, segment robota. Zato je najprej potrebno napisati funkcijo, ki med dvema 3D daljicama izračuna najmanjšo razdaljo. Prva daljica predstavlja na primer desno podlaket, druga daljica pa predstavlja nek drug segment robota, ki se mu hočemo z desno podlaktjo izogniti.

Funkcijo napisano v jeziku C je bilo potrebno najprej prevesti v obliko primerno za Matlab [22]. Ker funkcija vrne le najmanjšo razdaljo, na konec funkcije dodamo še izračun točk, kjer sta si daljici najbližji (3.3), (3.4) ter izračun enotskega

vektorja (3.5), ki kaže proč od druge daljice, torej proč od drugega segmenta. V spodaj navedenih enačbah (3.3) in (3.4) predstavljata $\mathbf{n}_1$ in $\mathbf{n}_2$ točki na prvi in drugi daljici, kjer sta si daljici (segmenta) najbližji, $\mathbf{S}_{1a}$, $\mathbf{S}_{1b}$, $\mathbf{S}_{2a}$ in $\mathbf{S}_{2b}$ pa predstavljajo krajišča daljic. Parametra s_c in t_c sta skalarna parametra, ki določata, v katerih dveh točkah sta si daljici najbližji. Skalar s_c opisuje točko na prvi daljici, skalar t_c pa točko na drugi daljici. Spreminjata se glede na bližino enega segmenta na drugega.

$$\mathbf{n}_1 = \mathbf{S}_{1a} + s_c * (\mathbf{S}_{1b} - \mathbf{S}_{1a}) \quad (3.3)$$

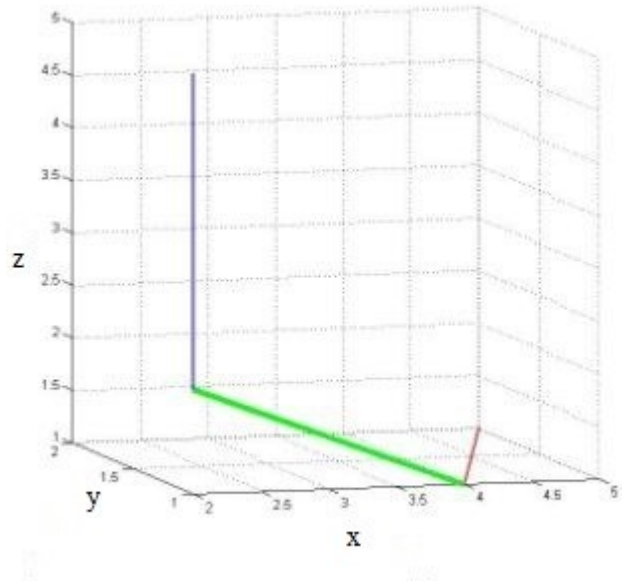
$$\mathbf{n}_2 = \mathbf{S}_{2a} + t_c * (\mathbf{S}_{2b} - \mathbf{S}_{2a}) \quad (3.4)$$

Na primer, če je $s_c = 0$ to pomeni da je točka $\mathbf{n}_1$, kar enaka točki $\mathbf{S}_{1a}$; če pa je $s_c = 1$ potem je $\mathbf{n}_1$ enako $\mathbf{S}_{1b}$. Če pa je $0 < s_c < 1$ potem je ta točka nekje vmes.

Na podlagi točk $\mathbf{n}_1$ in $\mathbf{n}_2$ izračunanih po (3.3) in (3.4), lahko izračunamo vektor $\mathbf{vec}$, ki ga bomo uporabili za izogibanje. Vektor $\mathbf{vec}$ kaže od točke $\mathbf{n}_1$ proti točki $\mathbf{n}_2$.

$$\mathbf{vec} = (\mathbf{n}_1 - \mathbf{n}_2) / \text{norm}(\mathbf{n}_1 - \mathbf{n}_2) \quad (3.5)$$

Da bi preverili delovanje funkcije vse tudi izrišemo (Slika 3.1). Z rdečo je predstavljena prva daljica oziroma segment robota, z modro pa je označena druga daljica. Z zeleno je predstavljena povezava med najbližjima točkama na obeh daljicah oziroma segmentoma.



Slika 3.1: Preizkus delovanja funkcije

Ko funkcija vrne najmanjšo razdaljo, enotski vektor, ki kaže proč od drugega segmenta, in koordinati najbližjih točk na obeh segmentih, izluščimo izmed vseh dobljenih razdalj tisto, ki je najmanjša, vmes pa shranjujemo pripadajoče indekse. To napravimo za oba segmenta obeh rok glede na vse ostale segmente. Nato indeks najmanjše razdalje uporabimo pri računanju osi za Jacobijevo matriko (3.6), ki predstavlja transformacijo hitrosti točke, ki je najbližja oviri, v sklepe za vsak segment leve in desne roke in pri računanju parametra α (3.7).

Stolpec Jacobijeve matrike, ki pripada sklepu j , se izračuna po naslednji enačbi:

$$\mathbf{J}_j = \mathbf{r}_j \times \mathbf{x}_j \quad (3.6)$$

Pri tem je $\mathbf{r}_j$ os sklepa j , $\mathbf{x}_j$ pa je vektor, ki povezuje središče sklepa s točko $\mathbf{n}_1$. Vsi predhodni stolpci so enaki kot pri običajni Jacobijevi matriki robota, medtem ko stolpce matrike, ki sledijo sklepu j , postavimo na nič, saj le-ti ne vplivajo na gibanje točke $\mathbf{n}_1$.

Za računanje parametra α pa moramo izbrati parameter d_m , ki označuje najmanjšo razdaljo, ki je še dovoljena med dvema segmentoma. Parameter α je nato definiran kot kvadrat razmerja med d_m in d_0 , pri čemer je d_0 najmanjša razdalja med katerim koli parom nasprotnim segmentom.

$$\alpha = \begin{cases} \left(\frac{d_m}{d_0}\right)^2, & d_0 > d_m \\ 1, & d_0 \leq d_m \end{cases} \quad (3.7)$$

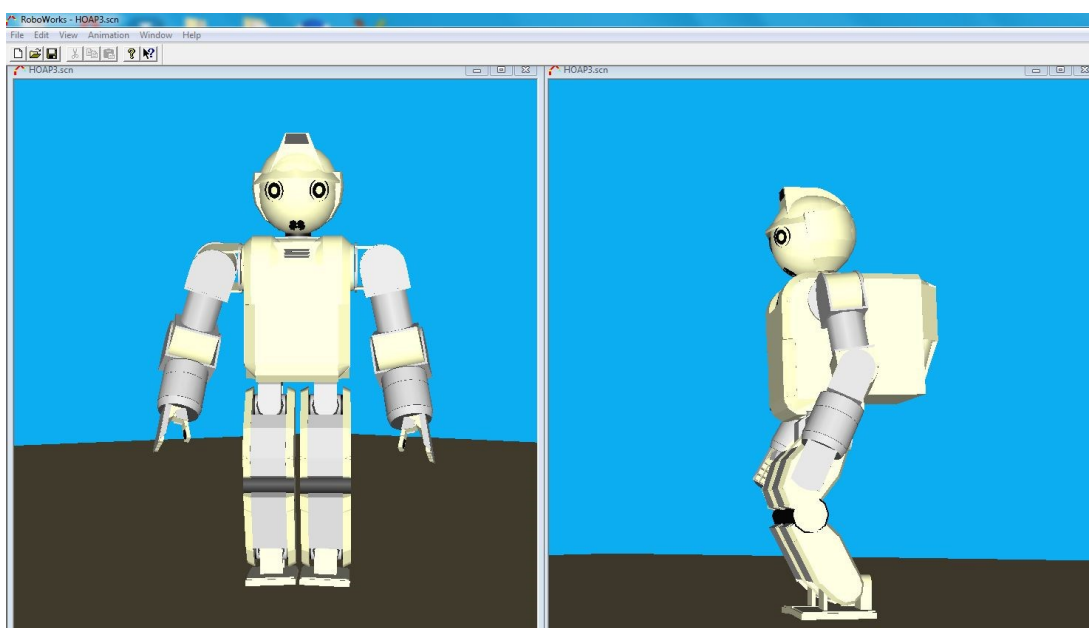
3.1.2 Sklepi in segmenti robota

Za izračun leg segmentov smo uporabili direktno kinematiko humanoidnega robota, s katero smo izračunali položaje vseh sklepov pri trenutni konfiguraciji robota. Najprej izračunamo desno nadlaket, ki jo določata ramenski del matrike in komolčni del matrike. Desna podlaket je določena s komolcem in zapestjem. Isto napravimo za levo roko. Desna stegenica je določena s kolkom in kolenom, desna golenica pa s kolenom in gležnjem. Enako napravimo za levo stran.

Sprva smo za telo vzeli segmenta, ki potekata od desne rame do desnega kolka in enako na drugi strani. Vendar pa se je pri tako določenih segmentih parameter α

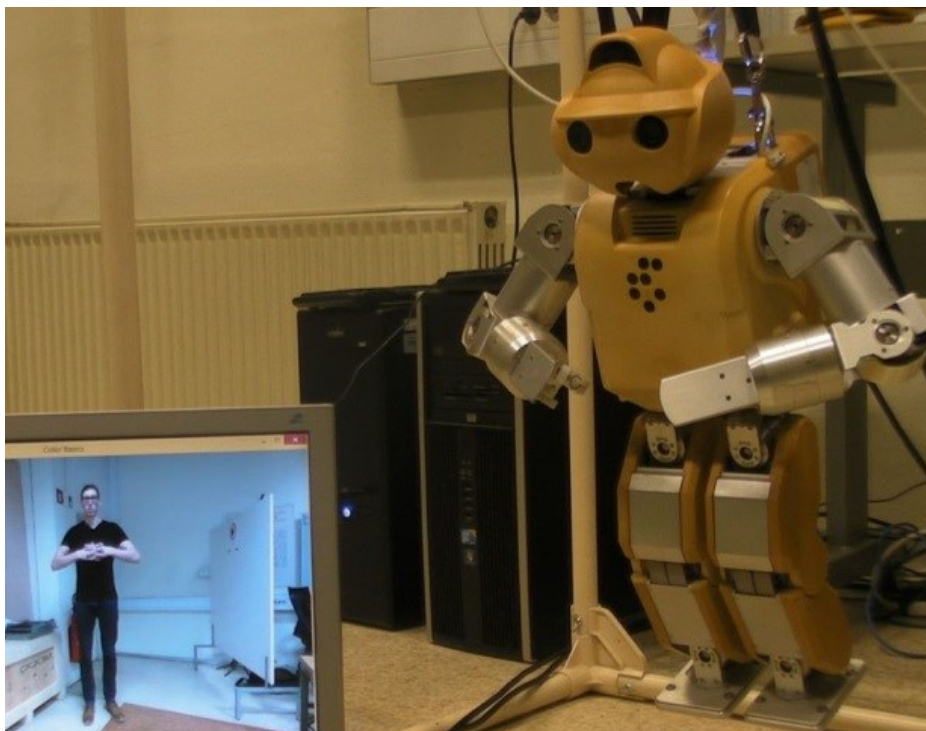
pri določenih konfiguracijah spreminjal tako, da robot ni deloval tako kot željeno – v nekaterih konfiguracijah je prišlo do trka roke in telesa. Zato smo telo določili od glave do obeh kolkov. Pri tako določenih segmentih se je robot uspešno izognil trkom med posnemanjem človekovega gibanja.

Vodenje po enačbi (3.2) smo preizkusili v simulacijskem okolju (Slika 3.2). Najprej smo preizkusili posnemanje gibanja z desno roko. Ko je izogibanje delovalo pravilno, smo preizkusili izogibanje trkom še za levo roko.



Slika 3.2: Simulacijsko okolje RoboWorks

Na koncu smo opravili še preizkus na realnem robotu (Slika 3.3). Iz slike vidimo, da kljub sklenjenima rokama demonstratorja, robot rok ne more skleniti saj bi se tako roke robota preveč približale trupu in bi prišlo do trka.



Slika 3.3: Končno preizkušanje na realnem robotu

3.2 Ohranjanje stabilnosti s pomočjo točke ničelnega momenta

Različne kinematične in dinamične lastnosti človeka in robota onemogočajo natančen prenos človekovega gibanja na robota, zato nikoli ne dobimo povsem enakega gibanja. Če želimo, da robot med posnemanjem gibanja človeka ne pade, moramo med posnemanjem upoštevati stabilnost robota. Za določanje in vzdrževanje stabilnosti obstajajo različne metode, med katerimi se najpogostejše uporablja vodenje s pomočjo točke ničelnega momenta (angl. zero moment point, ZMP). Ta pristop je prvi razvil Miomir Vukobratović [11] in ga uporabil za vodenje antropomorfnih mehanizmov [12]. Takanishi in sodelavci [13] so razvili model, ki uporablja vnaprej določene trajektorije nog, medtem ko ZMP spreminjajo s kompenzacijo mase. V preteklosti je bilo razvitih več metod za stabilno vodenje humanoidnih robotov s pomočjo točke ZMP, med drugimi metodi, ki so ju predlagali Kajita s sodelavci [14] in Takanishi s sodelavci [15].

V našem delu sledimo načinu, ki ga je predlagal Sugihara s sodelavci [16]. Metoda temelji na enostavnem modelu inverznega nihala, kjer je celotna dinamika robota predstavljena z dinamiko težišča, ki ga povežemo s točko ZMP. Gibanje

težišča oziroma centra gravitacije $\mathbf{x}_G$ lahko približno opišemo z naslednjima enačbama:

$$m\ddot{\mathbf{x}}_G = -m\mathbf{g} + \mathbf{f} \quad (3.8)$$

$$\mathbf{f} = k(\mathbf{x}_G - \mathbf{x}_{ZMP}) \quad (3.9)$$

kjer je (3.8) enačba gibanja in (3.9) enačba, ki povezuje vse zunanje sile $\mathbf{f}$, ki delujejo na robota, s centrom gravitacije in točko ZMP. Konstanta m je enaka celotni masi robota, $\mathbf{x}_G$ je pozicija težišča robota, $\mathbf{x}_{ZMP}$ je pozicija ZMP, g je gravitacijski pospešek, k pa je pozitivna konstanta. Če predpostavimo, da je $z = z_{ZMP}$, lahko izračunamo x_{ZMP} , y_{ZMP} iz naslednjih enačb

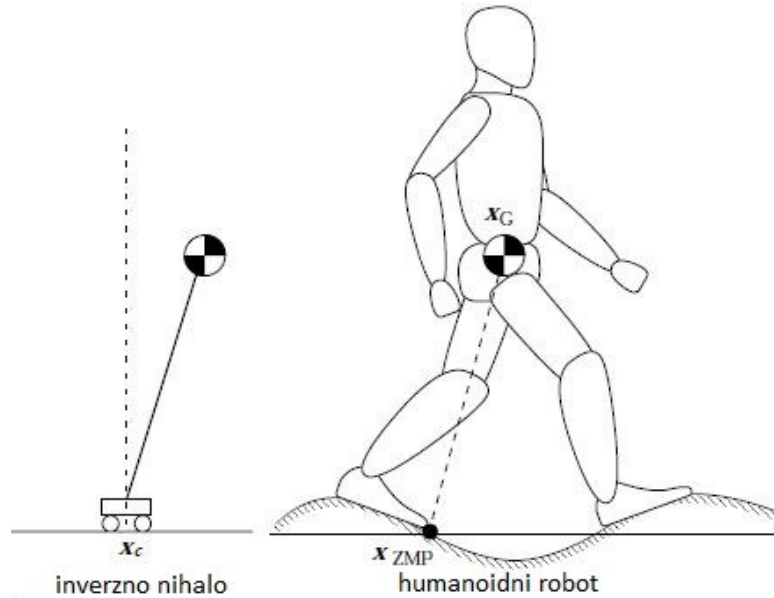
$$\ddot{x}_G = w_G^2(x_G - x_{ZMP}) \quad (3.10)$$

$$\ddot{y}_G = w_G^2(y_G - y_{ZMP}) \quad (3.11)$$

kjer je w_G določen z enačbo

$$w_G = \sqrt{\frac{\ddot{z}_G + g}{z_G - z_{ZMP}}} \quad (3.12)$$

Da je inverzno nihalo dober model, ki povezuje težišče in ZMP, pa kažejo rezultati v članku [16].



Slika 3.4: Inverzno nihalo in humanoidni robot [16]

Zveza med hitrostjo težišča robota v baznih koordinatah in hitrostmi v sklepih je dana z matriko J_{CoM} .

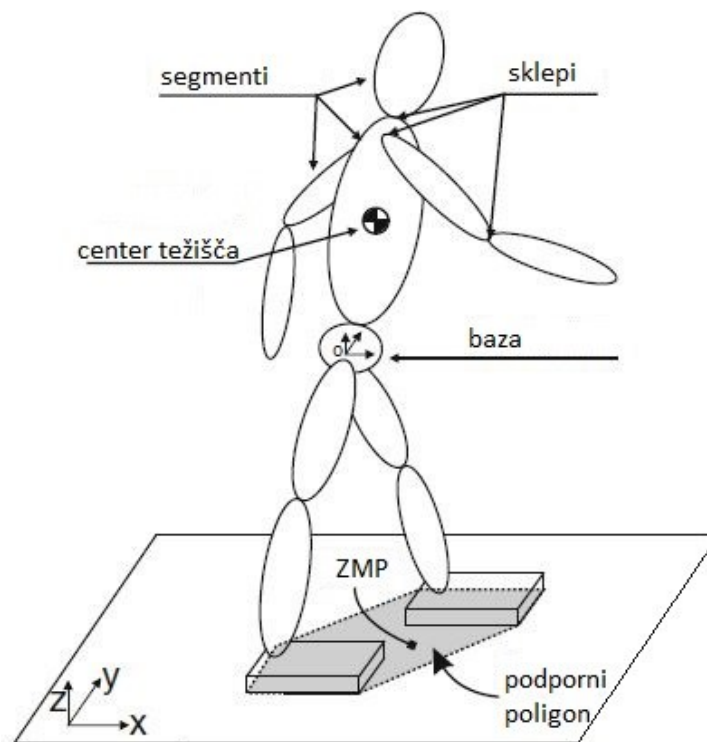
Robot je najbolj stabilen, ko je ZMP v centru podpornega poligona, ki ga določata stopali robota (Slika 3.5). Bližje kot je ZMP robu podpornega poligona, manj je robot stabilen. Ko ZMP enkrat doseže katero od točk na robu podpornega poligona, se robot prevrne. Zato je potrebno pri posnemanju elementov človekove hoje le-to napraviti na podlagi prioritete nalog. Primarna naloga skrbi za kontrolo stabilnosti, medtem ko se sekundarna naloga nanaša na posnemanje gibanja. Ko ZMP doseže določene meje, je omogočena primarna naloga, torej kontrola stabilnosti kjer je mogoče samo gibanje v sklepih v ničelnem prostoru stabilnostne matrike. Takšno gibanje ne povzroči nestabilnosti [8].

Da bi zagotovili stabilnost robota, moramo točko ZMP povezati s težiščem robota, saj lahko težišče direktno povežemo z gibanjem robota. Zato je pri vodenju potrebna Jacobijeva matrika centra težišča J_{CoM} , ki jo izračunamo po naslednjih enačbah

$$\mathbf{x}_{CoM} = \frac{\sum_{i=1}^n m_i \mathbf{x}_i}{\sum_{i=1}^n m_i} \quad (3.13)$$

$${}^b \dot{\mathbf{x}}_{CoM} = \frac{\sum_{i=1}^n m_i {}^b J_i \dot{\mathbf{q}}}{\sum_{i=1}^n m_i} = \frac{\sum_{i=1}^n m_i {}^b J_i}{\sum_{i=1}^n m_i} \dot{\mathbf{q}} = {}^b J_{CoM} \dot{\mathbf{q}} \quad (3.14)$$

kjer je ${}^b J_{CoM}$ geometrična Jacobijeva matrika centra težišča v baznih koordinatah, ${}^b J_i$ Jacobijeva matrika težišča i -tega dela telesa v baznih koordinatah za i -ti del robotskega segmenta v baznih koordinatah, z ${}^b \mathbf{x}_{CoM}$ pa označimo položaj težišča v baznih koordinatah [23].



Slika 3.5: Podporni poligon [8]

Problem stabilnosti torej lahko rešimo na enak način kot problem trkov, to je s prioriteto nalog. Spremeni se le kriterij, po katerem izračunamo parameter α .

4 Posnemanje elementov človekove hoje

V našem delu želimo združiti posnemanje človeka in možnost premikanja robota po prostoru. Najbolje bi bilo, če bi lahko hkrati z gibanjem zgornjega dela telesa neposredno na robota prenesli tudi gibanje sklepov nog med hojo. Vuga in sodelavci so to že izvedli, vendar je tak način hoje neroden in se ne obnese najbolje [17] [18]. Zato v tem delu predstavljamo rešitev, pri kateri v zgornjem delu robot posnema demonstratorjevo gibanje neposredno, s spodnjim delom pa mu sledi tako, da prilagaja vnaprej načrtovane vzorce hoje.

4.1 Trajektorije hoje robota

Obstaja več načinov načrtovanja trajektorij robotske hoje [13] [14] [15]. Vse te metode temeljijo na upoštevanju kriterija ZMP (poglavje 3). V tej nalogi se nismo ukvarjali s samim načrtovanjem trajektorij hoje, saj na robotu Hoap-3 že proizvajalec zagotavlja več trajektorij, ki upoštevajo kriterij ZMP. Za izvedbo posnemanja moramo vzorčne trajektorije hoje prilagoditi tako, da se robotova hoja ujema s človekovo.

Zaradi upoštevanja stabilnosti robota med posnemanjem gibanja rok, z gibanjem rok ne pokvarimo stabilnosti, zato robot ostaja stabilen, če je le hoja sama stabilna. V naših eksperimentih smo uporabili trajektorije hoje, ki vključujejo začetni korak, vmesni korak in končni korak. Če na primer izvršimo trajektorijo za hojo naprej, robot napravi korak naprej z desno nogo, nato z levo in priključi desno nogo. V primeru, da izvršimo trajektorijo za hojo nazaj, pa robot začne korakati nazaj z levo nogo. Trajektorijo beremo tako, da najprej preberemo vzorce začetnega koraka in nato, če korakamo naprej, beremo le še vzorce vmesnega koraka.

4.2 Prostorsko gibanje robota

4.2.1 Posnemanje premikanja človeka v prostoru

Cilj prvega dela naloge je bil, da z že obstoječimi trajektorijami hoje izvedemo gibanje robota naprej in nazaj, skladno z gibanjem človeka naprej in nazaj. Ko preberemo lego človekovega telesa iz Kinecta, kjer informacije izberemo na podlagi Slike 2.2, za določanje robotskih korakov upoštevamo samo premike trupa. Premik trupa naprej ali nazaj dobimo s primerjanjem prebranih leg v prejšnjem in tekočem merilnem času. Šele ko robot dokonča korak, je možen naslednji premik naprej (beremo trajektorijo za hojo naprej) ali nazaj (beremo trajektorijo za hojo nazaj), saj hoje ne moremo prekiniti, preden robot konča korak.

Robot torej kljub zaznavi premika človekovega trupa najprej naredi celoten korak in se šele nato odzove na nove podatke. Naš cilj pa je doseči, da robot ponovi premik človeka brez čakanja, da bi nek korak izvršili do konca. Da bi bila hoja robota čim bolj usklajena s premiki trupa človeka, moramo napraviti posnemanje hoje tako, da v vsakem obhodu našega programa sproti preverjamo pozicijo trupa človeka in nato, glede na najmanjši premik naprej ali nazaj, pošljamo vzorce trajektorije na robota enega za drugim. Za bolj gladko delovanje uporabljamo samo še trajektorijo za hojo naprej, ki jo v primeru, da se človek premika naprej beremo naprej, robot pa začne korakati z desno nogo. V primeru, da se človek začne premikati nazaj, trajektorijo beremo nazaj, robot pa začne korakati z levo nogo nazaj. Ko se človek začne premikati naprej, robotu posredujemo začetni del trajektorije hoje. V primeru, da človek nadaljuje s hojo, robotu posredujemo samo še trajektorijo vmesnega koraka in tako robot koraka naprej, dokler se premika človek. V primeru, da se demonstrator med hojo naprej premisli in začne korakati nazaj, nadaljujemo z branjem trajektorije v obratni smeri. Torej spreminjamo smer sledenja trajektoriji med izvajanjem koraka (naprej oziroma nazaj). Korake naprej oziroma nazaj izvedemo, če je zaznan premik telesa v prostoru večji od vnaprej določene mejne vrednosti. V tem delu naloge vključimo še posnemanje gibanja rok med hojo robota. Robot torej s spodnjim delom telesa ne posnema direktno gibanja človekovih nog, ampak le koraka naprej in nazaj glede na gibanje človekovega telesa v prostoru. V nasprotju s tem pa robot direktno posnema gibanje zgornjega dela telesa. Pri tem

uporabimo metodo prioritete nalog, ki je razložena v razdelku 3.2. Podatke za roke iz Kinecta na robota pošiljamo vseskozi, saj tako omogočimo posnemanje demonstratorjevih rok v vsakem trenutku.

4.2.2 Posnemanje elementov hoje človeka

Končni cilj je napraviti posnemanje hoje človeka tako, da robot sledi premikanju človeka in gre v korak z enako nogo kot človek. Ko trajektorijo za hojo beremo naprej, začne robot korakati naprej z desno nogo, ko pa jo beremo nazaj, začne korakati nazaj z levo nogo. Zato za primer, ko človek začne korak z desno nogo nazaj oziroma z levo nogo naprej, preprosto iz trajektorije za hojo naprej ustvarimo novo trajektorijo in sicer tako, da zamenjamo podatke trajektorije za levo in desno nogo, saj vsaka trajektorija vsebuje podatke za obe nogi. Zamenjanim podatkom je na robotu Hoap-3 treba zamenjati predznak, da dobimo želeno gibanje (dodatek A). Sedaj lahko začne robot korakati s poljubno nogo. Da bi lahko robot začel korak z enako nogo kot človek, se moramo najprej prepričati, da sta nogi robota postavljeni ena ob drugi, torej, da sta v osnovni poziciji. Če temu ni tako, lahko robot samo nadaljuje izvajanje koraka v isto smer kot človek, vendar s tisto nogo, ki je trenutno na vrsti. Če je na primer desna noga robota postavljena pred levo, demonstrator pa se premakne naprej z desno nogo, gre robot naprej z levo nogo. To je edina možnost med samo hojo, ko je začetni korak že napravljen. Zato je pomembno, da ob zaključku hoje, ko demonstrator postavi noge vzporedno in hojo zaključi, to napravi tudi robot, saj lahko potem robot začne korak z enako nogo kot človek. To napravimo z interpolacijo položajev nog (dodatek B).

Za izvedbo opisanega postopka, je potrebno spremljati tudi gibanje nog človeka, saj le tako lahko izračunamo, s katero nogo mora robot začeti korak. V ta namen je potrebno spremljati premikajoče se dele telesa in ustrezno definirati odzivanje robota. Zato poleg gibanja trupa spremljamo tudi premikanje kolena glede na oddaljenost od Kinecta. Spremljamo torej premik kolena v smeri koordinate z (Slika 2.2). Med posnemanjem preverjamo, če je demonstrator premaknil desno ali levo nogo in če sta nogi skupaj v nevtralnem začetnem položaju. To ugotovimo tako, da izračunamo razliko med kolenoma, ki jo dobimo glede na oddaljenost obeh kolen do Kinecta. Če človek premakne nogo naprej, se koleno približa Kinectu. Ko pa gre

človek z nogo nazaj, se koleno oddalji od Kinecta. Ob spremljanju premikov kolen spremljamo tudi smer premika trupa demonstratorja, kar nas privede do končnega delovanja.

V nadaljevanju izvajanja posnemanja faze hoje obstajajo 4 različne možnosti, glede na to, v katerem stanju se nahaja robot in kakšna je reakcija človeka.

Če dvignemo desno nogo in se s trupom premaknemo naprej, najprej preverimo, ali sta bili nogi predhodno skupaj. Če sta nogi skupaj, beremo osnovno trajektorijo, kjer napravimo korak naprej z desno nogo. Ko napravimo korak naprej še z levo nogo, preprosto beremo osnovno trajektorijo naprej, saj sta nogi ob preverjanju narazen.

Če dvignemo desno nogo in se s trupom premaknemo nazaj, najprej preverimo, ali sta bili nogi predhodno skupaj. Če sta nogi skupaj, beremo na novo napravljeno trajektorijo, kjer napravimo korak nazaj z desno nogo. Ko napravimo korak nazaj še z levo nogo, preprosto nadaljujemo z branjem trajektorije nazaj, saj sta nogi ob preverjanju narazen.

Če dvignemo levo nogo in se s trupom premaknemo naprej, najprej preverimo, ali sta bili nogi predhodno skupaj. Če sta bili nogi skupaj, beremo na novo napravljeno trajektorijo, kjer napravimo korak naprej z levo nogo. Ko napravimo korak naprej še z desno nogo, pa preprosto nadaljujemo z branjem nove trajektorije, saj sta nogi ob preverjanju narazen.

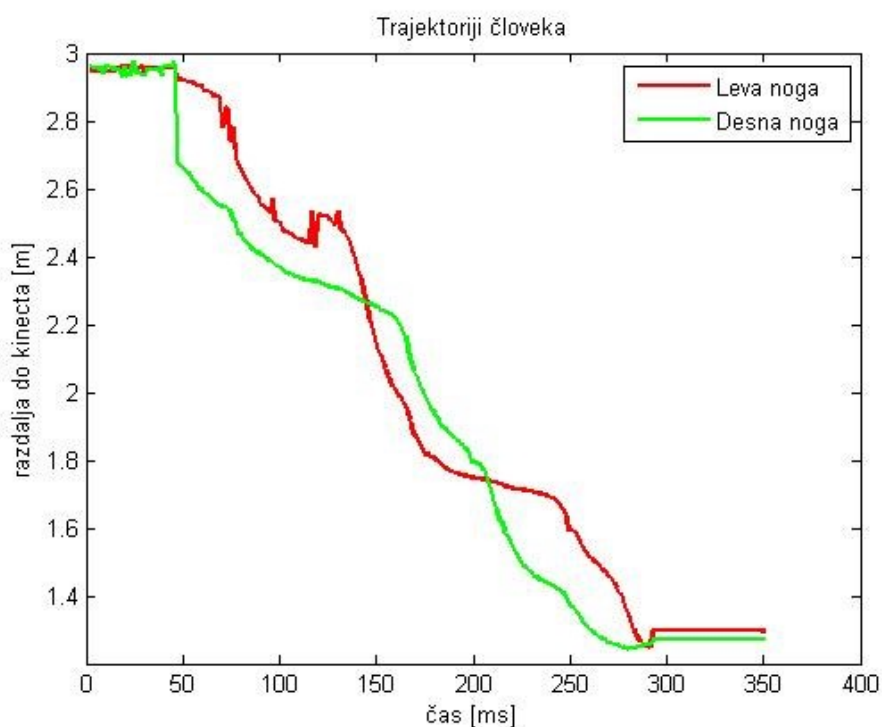
Če dvignemo levo nogo in se s trupom premaknemo nazaj najprej preverimo, ali sta bili nogi predhodno skupaj. Če sta nogi skupaj beremo osnovno trajektorijo, kjer napravimo korak nazaj z levo nogo. V tem primeru beremo trajektorijo nazaj. Ko napravimo korak nazaj še z desno nogo, preprosto nadaljujemo z branjem trajektorije nazaj saj sta nogi ob preverjanju narazen.

4.3 Rezultati

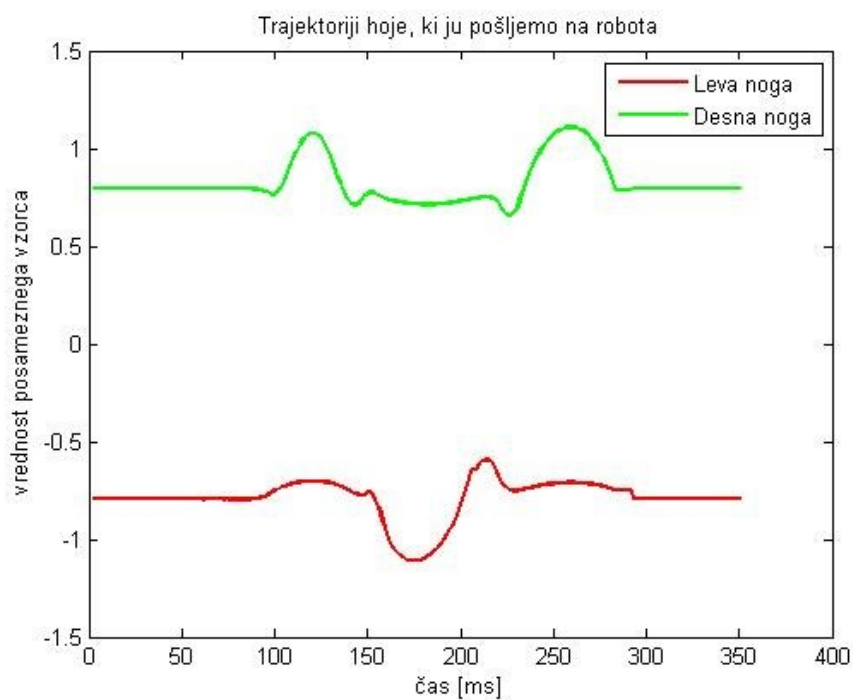
Na Sliki 4.1 vidimo hojo človeka posneto s Kinectom. Prikazani sta trajektoriji za levo in desno koleno, ko se človek giblje proti Kinectu. Iz spodnje slike vidimo, da je demonstrator najprej napravil korak naprej z desno nogo, kmalu za premikom

desne noge naprej se začne premikati leva noga in ko je desna noga že na tleh, počasi leva noga dohiti desno (~150 s) in stopi pred njo. Nato se premika desna noga in z njo stopimo pred levo, ki jo za konec koraka priključimo k desni (~300 s).

Na Sliki 4.2 je prikazana hoja robota, ko le ta posnema elemente človekove hoje, ki jih dobimo iz Kinecta. Najprej napravi robot korak naprej z desno nogo in jo z levo nogo dohiti (~150 s). Nato robot premakne levo nogo pred desno in sedaj robot z desno nogo dohiti levo nogo (~220 s), jo premakne pred levo in nato z levo zaključi korakanje tako, da levo nogo postavi k desni (~300 s). Vidimo da se gibanje nog robota po času ujema z gibanjem nog človeka.

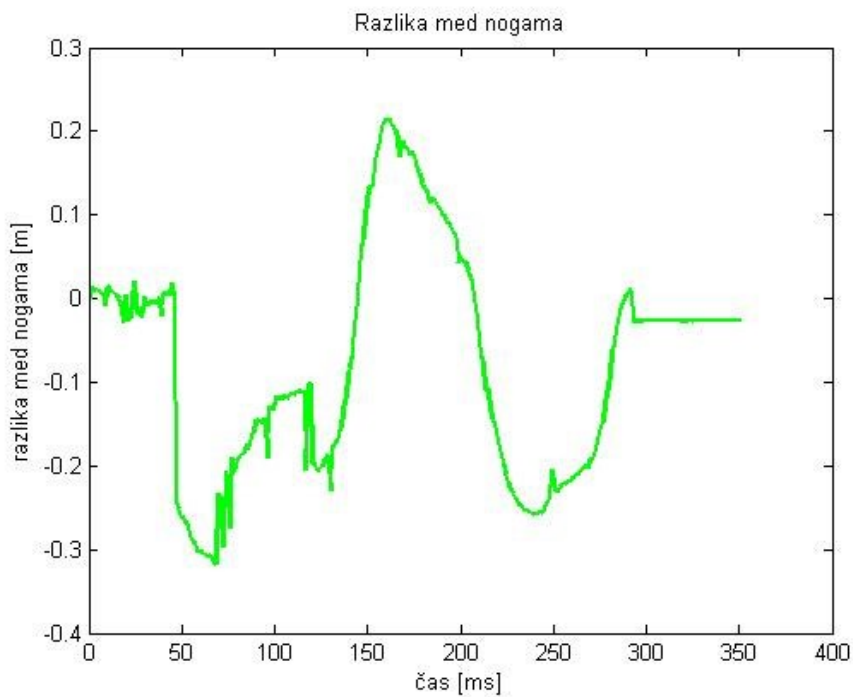


Slika 4.1: Trajektoriji 3D pozicije kolena



Slika 4.2: Trajektoriji hoje robota

Ker je gibanje robota odvisno tudi od razlike med kolenoma demonstratorja, na Sliki 4.3 prikažemo razliko med kolenoma, ki se časovno ujema s hojo robota.



Slika 4.3: Razlika med nogama oziroma kolenoma človeka

5 Zaključek

Za izvedbo naloge se je bilo potrebno najprej seznaniti s samim robotom in njegovim delovanjem, delovanjem Kinecta in branjem podatkov, teorijo prioritete nalog, preprečevanjem trkov in zagotavljanjem oziroma vzdrževanjem stabilnosti robota na podlagi ZMP. Potrebno je bilo spoznati način hoje oziroma način branja trajektorij, sestavo trajektorij, prenos le teh na robota in seveda ujeti primerno ločljivost premikanja delov človeškega telesa s pošiljanjem vzorcev trajektorij na robota.

V tem delu smo implementirali posnemanje elementov človekove hoje, kjer smo poleg stabilnosti vključili tudi preprečevanje trkov na podlagi prioritete nalog. S takšnimi metodami bodo humanoidni roboti lahko postali bolj uporabni v vsakdanjem življenju.

Literatura

- [1] M. Hanlon, „Gizmag,“ 2004. [Elektronski]. Available: <http://www.gizmag.com/go/1765/>. [Poskus dostopa 11 12 2015].
- [2] S. Schaal, „Is Imitation Learning the Route to Humanoid Robots?,“ *Trends in Cognitive Sciences*, zv. 3, pp. 233-242, 1999.
- [3] R. Dillmann, „Teaching and learning of robot tasks via observation,“ *Robotics and Autonomous Systems*, zv. 47, pp. 109-116, 2004.
- [4] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp in R. Dillmann, „An integrated humanoid platform for sensory-motor control,“ v *6th IEEE-RAS International Conference on Humanoid Robots*, Genoa, 2006.
- [5] B. D. Argall, S. Chernova, M. Veloso in B. Browning, „A survey of robot learning from demonstration,“ *Robotics and Autonomous Systems*, zv. 57, pp. 469-483, 2009.
- [6] R. Kamnik, J. Podobnik, J. Babič in T. Koritnik, *Osnove robotike - Priročnik in navodila za laboratorijske vaje*, Ljubljana: Univerza v Ljubljani, Založba FE in FRI, 2011.
- [7] K. Shoemake, „Euler angle conversion,“ *Graphic Gems IV*, pp. 222-229, 1994.
- [8] T. Petrič in L. Žlajpah, „Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem,“ *Robotics and Autonomous Systems*, zv. 61, pp. 948-959, 2013.
- [9] O. Khatib, „Real time obstacle avoidance for manipulators and mobile robots,“ v *Conference on Robotics and Automation*, 1985.
- [10] A. Maciejewski in C. Klein, „Numerical filtering for the operation of robotic manipulators through kinematically singular configurations,“ *The*

- International Journal of Robotic Research*, pp. 109-117, 3 1985.
- [11] M. Vukobratović, „International Journal of Humanoid Robotics,“ *Zero-Moment Point - Thirty five years of its life*, pp. 157-173, 1 11 2004.
- [12] M. Vukobratović in J. Stepanenko, „Mathematical Biosciences,“ *On the stability of anthropomorphic systems*, pp. 1-37, 10 1972.
- [13] A. Takanishi, M. Tochizawa, T. Takeya, H. Karaki in I. Kato, „Realization of Dynamic Biped Walking Stabilized with Trunk Motion Under Known External Force“ v *Advanced Robotics*, Springer Berlin Heidelberg, 1989, pp. 299-310.
- [14] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi in H. Hirukawa, „*Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point*“ v International Conference on Robotics & Automation, Taipei, 2003.
- [15] A. Takanishi, L. Hun-ok, M. Tsuda in I. Kato, „Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface,“ *Intelligent Robots and Systems '90. Towards a New Frontier of Applications, Proceedings. IROS '90. IEEE International Workshop on*, zv. 1, pp. 323-330, 3 7 1990.
- [16] T. Sugihara, Y. Nakamura in H. Inue, „*Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control*“ v International Conference on Robotics & Automation, Washington, 2002.
- [17] R. Vuga, M. Ogrinc, A. Gams, T. Petrič, N. Sugimoto, A. Ude in J. Morimoto, „*Motion capture and reinforcement learning of dynamically stable humanoid movement primitives*“ v Robotics and Automation (ICRA), Karlsruhe, 2013.
- [18] M. Ogrinc, *Vodenje stabilnosti humanoidnega robota z uporabo modificirane metode prioritete nalog*, Univerza v Ljubljani, 2012.
- [19] C. Becker-Asano in H. Ishiguro, „*Affective Computational Intelligence*“ v IEEE, Paris, 2011.
- [20] H. Jungong, S. Ling, X. Dong in J. Shotton, „Enhanced Computer Vision With Microsoft Kinect Sensor: A Review“, *Cybernetics, IEEE Transactions on*, zv. 43, št. 5, pp. 1318-1334, 25 6 2013.
- [21] J. Lenarčič in T. Bajd, *Robotski mehanizmi*, Ljubljana: Fakulteta za elektrotehniko, 2009.
- [22] D. Sunday, „iSurfer.org,“ softSurfer, 2012. [Elektronski]. Available:

- http://geomalgorithms.com/a07-_distance.html. [Poskus dostopa 15 10 2014].
- [23] A. Gams, J. Kieboom, F. Dzeladini in A. J. Ijspeert, „*Stable real-time full body motion imitation on the COMAN humanoid robot* “ v 22nd International Workshop on Robotics in Alpe-Adria-Danube Region, Portorož, 2013.
- [24] T. Asfour, K. Welke, P. Azad, A. Ude in R. Dillmann, „*The Karlsruhe Humanoid Head*“ v IEEE, Daejeon, 2008.

