View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DSpace at Tartu University Library

TARTU ÜLIKOOL
LOODUS- JA TEHNOLOOGIATEADUSKOND
Tehnoloogiainstituut

Kristian Hunt

HUMANOIDROBOTI ALDEBARAN NAO JALGPALLITARKVARA KÄITUMISLOOGIKA ARENDAMINE

Bakalaureusetöö (12 EAP)

Juhendajad: prof. Alvo Aabloo,
Heilo Altin

Kaitsmisele lubatud .....

Juhendaja .....
allkiri, kuupäev

Tartu 2014

Sisukord

Sissejuhatus 4
1. Nao roboti tutvustus 5
1.1. NAO eesmärk ja teised kasutusalad 5
1.2. Roboti riistvara 6
1.3. Roboti tarkvara 6
1.3.1. NAOqi tarkvararaamistik 7
2. RoboCup 8
2.1. Ajalugu 8
2.2. Reeglid ja võistlustingimused 9
2.2.1. Väljak ja pall 9
2.2.2. Mängu ülesehitus 10
2.2.3. Robotitevaheline suhtlus 10
2.2.4. Peamine lõplik olekumasin 11
2.2.5. Keelatud võtted ja karistus 12
2.2.6. Reeglite ajalugu 12
2.3. *Drop-in Player* võistlus ja tehnilised väljakutsed 13
3. Töökeskkond ja -vahendid 14
4. Ülevaade kasutatava Austin Villa koodibaasi ülesehitusest 16
4.1. Arhitektuur 16
4.2. Loogikamoodulid 17
4.2.1. Nägemismoodul 17
4.2.2. Positsioneerimismoodul 18
4.2.3. Liikumismoodul 19
4.2.4. Käitumisloogika moodul 19
4.2.4.1. Initsialiseerimine ja täitmise järjekord 20
4.2.4.2. Ülesanded 20
4.3. Mälu 21
4.4. Koordinaatsüsteemid 22
5. Tehtud töö 23
5.1. Testimine 23
5.1.1. Kestvustest 23
5.1.2. Objektituvastuse test 24
5.2. Väljakuobjektidega kokkupõrgete vältimine 24
5.3. Individuaalstrateegia 24

5.3.1. Teiste võistkondade individuaalstrateegiad 25
5.3.2. Individuaalstrateegia üldreeglid 25
5.3.3. Rollispetsiifilised reeglid 26
5.3.4. Individuaalstrateegia olekumasin 27
5.4. Meeskonnastrateegia 28
5.4.1. Teiste võistkondade meeskonnastrateegiad 29
5.4.2. Meeskonnastrateegia üldreeglid 29
Kokkuvõte 31
Viited 32
Summary in English 36
Lisad 37
Lisa 1. NAO roboti elektroonikaseadmete arhitektuur 37
Lisa 2. RoboCup SPL võistluse mänguväljak 38
Lisa 3. Valminud kood 39
Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks 40

Sissejuhatus

Robootika ja intellektitehnika on mõlemad kiiresti arenevad teadusharud. Robotite jalgpall pakub väljakutset nii robootika, intellektitehnika kui ka mehhatroonika tehnika- ja teadusharudes ning on samal ajal pealtvaatajatele huvitav vaadata. Seega saab robotite jalgpallimängu abil teha laiemale publikule arusaadavaks edasiminekud sellega seotud valdkondades.

Humanoidrobot on kujult inimesetaoline ja suudab enamasti kõndida kahel jalal [1]. Sellise roboti eesmärk on jagada inimestega töökeskkonda ja mõtlemismudelit [1]. Eesmärgi täitmiseks peab robot olema võimeline tajuma seda ümbritsevat keskkonda, inimestega suhelda ja õppima kohanema muutuvate olukordadega [1]. Humanoidroboteid kasutatakse näiteks nii kosmoses inimesele ohtlike tööde tegemisel, haigete ja vanurite abilisena kui ka giidina muuseumis [2,3].

2008. aastal liitus Aldebaran NAO (edaspidi NAO) humanoidrobot jalgpallurina võistlusega RoboCup Standard Platform League (edaspidi SPL), kus kõik robotid on sama riistvaraga ja erinevad ainult tarkvaralise lahenduse poolest. Samuti on kõik robotid täielikult autonoomsed ehk nad teevad oma otsuseid ise. RoboCup võistluse (vt ptk 2) eesmärk on populariseerida robootikat ja intellektitehnikat, kuid RoboCupi kõrgem siht on arendada robotite jalgpall sellisele tasemele, et 21. sajandi keskel alistada robotitega viimase jalgpalli maailmakarika võitnud inimeste võistkond. [4,5]

Käesoleva bakalaureusetöö eesmärk on välja arendada RoboCup SPL 2014. aasta võistluse nõuetele vastav jalgpallitarkvara käitumisloogika, mida kasutatakse robotitevahelise võrguühenduse puudumisel ning mis põhineb Texase Ülikooli võistkonna UT Austin Villa (edaspidi Austin Villa) 2012. aastal avalikustatud koodil. Töö eesmärgi saavutamiseks tuleb luua robotite käitumisloogika ja muuta kood uutele reeglitele vastavaks. Töö valmib koostöös Tartu Ülikooli meeskonnaga Philosopher, kes osaleb juulis 2014 Brasiilias toimuval RoboCup võistlusel. Vastavalt võistkonna seatud eesmärkidele populariseeritakse robootikat Eestis, tehes demonstratsioone, kus tutvustatakse NAO platvormi, RoboCupi ja hetkeseisu jalgpallitarkvarast [6].

1. Nao roboti tutvustus

NAO on programmeeritav, 58 cm kõrge ja 5,2 kg kaaluv humanoidrobot, mida arendab 2005. aastal asutatud Prantsusmaa firma Aldebaran Robotics [5,7]. Käesolevas töös kasutatakse NAO neljandat versiooni.

1.1. NAO eesmärk ja teised kasutusalad

Aldebaran Robotics arendas NAO platvormi välja eesmärgiga integreerida robotid inimeste igapäevaellu. Seetõttu on robotis ka nii palju erinevaid sensoreid, et inimestega suhtlemine võimalikult lihtsaks teha (vt ptk 1.2.). Tänapäeval täidavad NAOd lisaks inimeste igapäevaelu abistamisele ka meelelahutuslikke ülesandeid, kuid on leidnud kasutust ka autismi teraapias. 2013. aastal käivitas Aldebaran Robotics *Autism Solution for Kids* programmi, mis pakub uusi õpetamisvõimalusi ja meetodeid õpetajatele, kes töötavad autismiga lastega. [5]

Autismi põdevatele lastele on oluline õpetada teisi inimesi imiteerima – oskus, mille terved lapsed omandavad loomulikult, on tihti puudulik autismi põdevatel lastel [8]. Teiste imiteerimine on aga oluline osa inimestevahelisest suhtlusest ja sotsiaalsest arengust [8]. Uuringud on näidanud, et autismi põdevaid lapsi paeluvad mehaanilised komponendid, arvutid ja robotid. Seetõttu on välja pakutud autismiga lastele imiteerimise õpetamiseks kasutada virtuaalseid keskkondi ja roboteid. [9,10]

Adriana Tapus võrdles oma uurimustöös nelja autismi põdeva lapse reaktsioone, kui nende motoorikat jäljendas inimene ja NAO robot. Tulemused olid erinevad: kaks last reageerisid samamoodi nii inimesele kui ka robotile, samas kui kaks last hoidsid robotil pikemalt pilku ja naersid tihemini, kui robot neid jäljendas. Ühele lastest osutus robot paremaks liigutuste koolitajaks. Robotile pöörasid suuremat tähelepanu kaks last, kes olid raskemate autismi sümptomitega. Seetõttu soovitati NAO roboteid kasutada pigem sügavate autismitunnustega laste teraapias. [10]

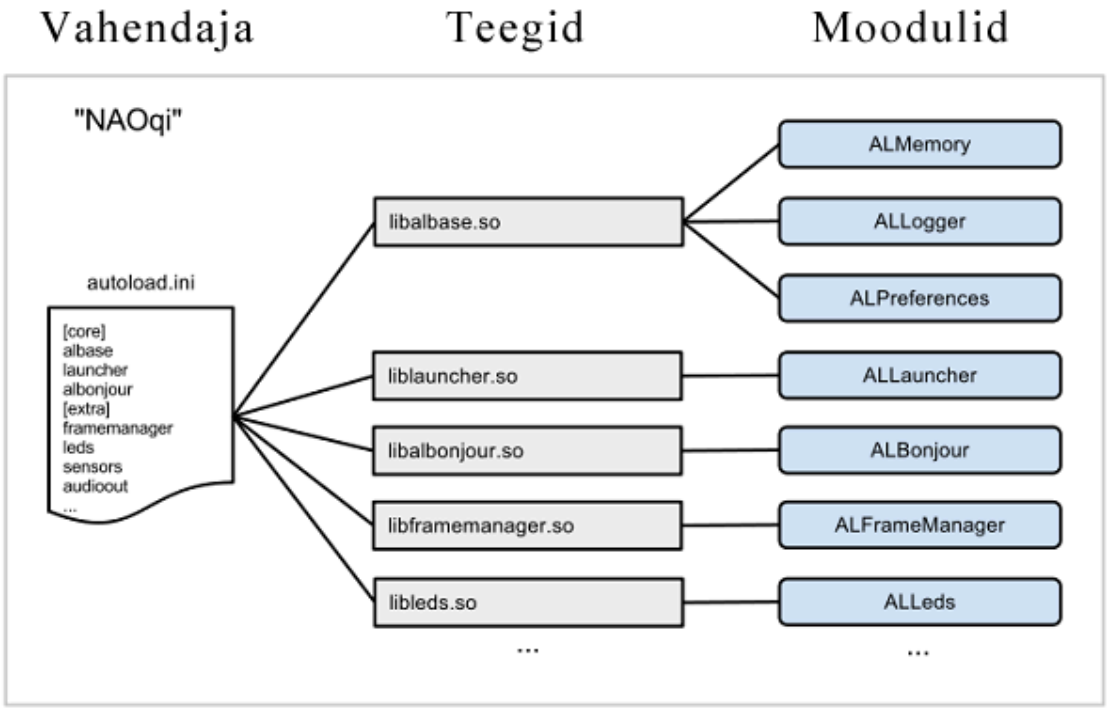
1.2. Roboti riistvara

Robotile tarkvara arendamiseks tuleb teada selle tehnilisi andmeid. Robotis on x86 arhitektuuriga Intel Atom Z530 protsessor taktsagedusega 1,6 GHz, 1 GB muutmälu ja 2 kuni 10 GB välkmälu. NAOs on 2 võrgukaarti: Gigabit Ethernet ja IEEE 802.11 b/g WiFi võrgukaart. Ümbritseva keskkonna tajumiseks on robotil kasutada kaks 960p resolutsiooniga maksimaalselt 30 kaadrit sekundis töötavat kaamerat, kaks aktiivset sonarit, infrapunakiirguse andurid, kaks güroskoopi, kiirendusandur ja 4 mikrofoni. Lisaks on roboti pea ja käte peal puutetundlik riba ning nupud jalgade otstes ja rinnus. Roboti liikumisvabadusastmete arv on 25 ja selle aku peab vastu kuni 90 minutit. Vaadake Lisa 1 täpse ülevaate saamiseks, millised elektroonikaseadmed on robotis ja kuidas nad on omavahel ühendatud. [7]

1.3. Roboti tarkvara

Operatsioonisüsteemiks robotis on OpenNAO 1.14.5 GNU/Linux, mis põhineb Gentoo distributsioonil ja on Aldebaran Roboticsi poolt välja arendatud spetsiaalselt NAOde jaoks. Operatsioonisüsteem sisaldab kõiki vajalikke juhtprogramme ning tarvikuid. Vaikimisi on operatsioonisüsteem seadistatud selliselt, et robotit saaks juhtida Aldebaran Roboticsi visuaalprogrammeerimise keskkonna Choreographe abil (vt ptk 3). Ligipääs robotisse toimub SSH (*Secure Shell)* võrguprotokolli abil ning selleks peab olema loodud ühendus robotiga ja teadma roboti IP aadressi. [11]

Roboti tööle lülitamisel käivitatakse kaks protsessi: connman ja NAOqi. Connman protsess haldab roboti võrguühendust [11]. NAOqi protsess on vahendaja, mis käivitamisel laeb autoload.ini failis defineeritud teegid ja vahendab neis leiduvaid mooduleid ja meetodeid. Näide protsessi tööst on kujutatud joonisel 1. NAOqi protsess võimaldab moodulitel leida ja käivitada meetodeid teistest moodulitest. Lisaks võimaldab vahendaja välja kutsuda laetud moodulite meetodeid teistest protsessidest. [12]



Joonis 1. Näide NAOqi vahendaja laetavatest teekidest ja moodulitest. [12]

1.3.1. NAOqi tarkvararaamistik

NAOqi Framework on tarkvararaamistik, mida kasutatakse NAO programmeerimiseks. Raamistik lubab suhtlust, programmeerimist ja infovahetust kõikide Aldebaran Roboticsi ja programmeerija enda loodud moodulite vahel. NAOqi tarkvararaamistik toetab kolme operatsioonisüsteemi: Mac OS X, Windows ja Linux. Rakendusliides on täielikult programmeeritav Python ja C++ programmeerimiskeeltes. Tarkvararaamistiku kasutamine on ühetaoline olenemata kasutatavast operatsioonisüsteemist ja programmeerimiskeelest. [12]

Andmevahetuseks roboti andurite ja täiturite vahel kasutatakse NAOqi süsteemimoodulit DCM (*Device Communications Module* – seadme kommunikatsioonimoodul, ingl. k.), mis vastutab kommunikatsiooni eest kõikide roboti elektroonikaseadmete vahel, välja arvatud heliseadmed ja kaamerad. DCM juhib peamise andmevahetuskanali – USB ühenduse roboti peas oleva Intel Atom protsessori ja *ChestBoard*i nimelise ARM protsessori vahel roboti rinnus – tööd (vt Lisa 1). [13]

2. RoboCup

RoboCup on iga-aastane rahvusvaheline teadusliku taustaga võistlus, mille eesmärk on edendada uusimaid intelligentseid roboteid ning populariseerida robootikat ja intellektitehnikat. Praegusel momendil on lisaks jalgpallile RoboCupil ülesanded ka pääste- ja koduabirobotitele. Lisaks on eraldi väljakutsed lastele. [14]

Tabel 1. Võrdlus väljakutsetest robotitele RoboCupi ülesannetes [15,16,4]

Rescue	**@Home**	**Soccer**
Mobiilsus	Mobiilsus	Mobiilsus
Ümbruse tajumine	Ümbruse tajumine	Ümbruse tajumine
Etteplaneerimine	Etteplaneerimine	Etteplaneerimine
Ümbruskonna kaardistamine	Ümbruskonna kaardistamine	Ümbruskonna kaardistamine
Ei ole autonoomsed	Autonoomsed	Autonoomsed
Meeskonnatöö robotite vahel	Robotid suhtlevad inimestega	Meeskonnatöö robotite vahel
Keskkond ette teada	Dünaamiline keskkond	Keskkond ette teada
Liikumine õhus ja roomikutel	Liikumine ratastel	Kahel jalal kõndimine ja liikumine ratastel

Dr. Tijin van der Zant Groningeni Ülikoolist on öelnud: "(Robotite) jalgpall keskendub raskete ülesannete lahendamisele lihtsustatud keskkonnas ja meie (RoboCup @Home) teeme seda teistpidi ... Meie peame lahendama lihtsaid ülesandeid keerulises keskkonnas selle asemel, et teha keerulisi ülesandeid lihtsas keskkonnas" [17]. Suurim erinevus RoboCup päästerobotite liiga ja teiste liigade vahel seisneb selles, et päästerobotid ei ole täisautonoomsed [16].

2.1. Ajalugu

Esimene kord mainiti ideed sellest, kuidas robotid mängivad jalgpalli, 1992. aastal professor Alan Mackworthi artiklis *"On Seeing Robots"*. Samal aastal toimus

Jaapanis Tokyos seminar, kus arutati võimalikke suuri väljakutseid tehisintellekti valdkonnas ning jõuti järeldusele, et kasutades väljakutsena jalgpallimängu, on võimalik tutvustada tehnoloogiat ja teadust suuremale publikule. 1993. aasta juunis algatati projekt nimega "*Robot World Cup Initiative",* lühendatult "RoboCup". Sama aasta septembris tehti esimene avalik selleteemaline teadaanne. Enne esimest ametlikku RoboCup võistlust toimus eel-võistlus 1996. aasta novembris Osakas, kus osales 8 võistkonda simulatsiooniliigas ja toimus ka demonstratsioon päris robotite jalgpallis, keskmise suurusega robotite liigas. Tegu oli maailma esimese üritusega, kus kasutati jalgpallimängu eesmärgiga propageerida teadust ja haridust. 1997. aastal toimus Jaapanis Nagoya linnas esimene ametlik RoboCup võistlus, kus osales üle 40 võistkonna. [18]

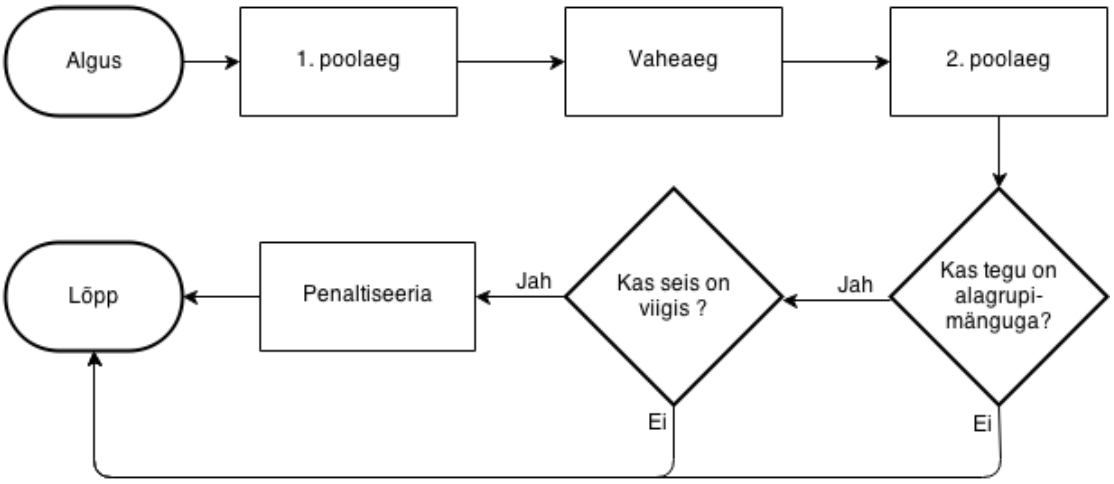
Standardse platvormiga liiga liideti võistlusele 1998. aastal ning platvormina kasutati Sony AIBO robotkoera. Liiga nimi oli "*Sony Four Legged Robot Football League*" (ingl. k. "Sony neljajalgse roboti jalgpalliliiga"). 2008. aastal jagati liiga kaheks, kuna lisandus kahejalgne platvorm, NAO, ja nimi muudeti tänapäevaseks: *Standard Platform League*. 2009. aastast alates enam Sony AIBO roboteid ei kasutatud. [19]

2.2. Reeglid ja võistlustingimused

2.2.1. Väljak ja pall

Jalgpalli mängitakse 9 m pikal ja 6 m laial väljakul. Väravad on 1,5 m laiad ja kollast värvi. Mõlemad väljakupooled on täpselt samasugused ning seetõttu on väga oluline, et robotid oskaksid enda asukohta kindlaks määrata. Detailsemat infot väljaku mõõtmete kohta saab Lisast 2. Võistkondade robotid kannavad eri värvi särke: üks võistkond on punane ja teine sinine. Igas meeskonnas on 5 mängijat, kellest 4 on väljakumängijad ja üks väravavaht, kusjuures väljakumängijatest võib korraga ainult üks viibida oma võistkonna väravavahialas. Väljakul on üks oranži värvi tänavahokipall. Lisaks võib olla võistkonnas treener-robot, kes jälgib mängu käiku väljaku äärest ning saadab lühikesi inimloetaval kujul sõnumeid tervele meeskonnale. [20]

2.2.2. Mängu ülesehitus



Joonis 2. Vooskeem mängu käigust.

Joonisel 2 on kujutatud vooskeem mängu käigust. Mängitakse kaks poolaega, mõlemad on 10 minutit pikad. Kahe poolaja vahel on 10 minuti pikkune vaheaeg, mille vältel võivad meeskonnad vahetada roboteid ja programme, ning pärast vaheaega toimub pooltevahetus. Kui tegu ei ole alagrupimänguga ja seis on pärast teist poolaega viigis, toimub penaltiseeria. Võistkond võib korra mängu jooksul mänguseisaku ajal võtta kuni viie minutise *time-out*i. [20]

2.2.3. Robotitevaheline suhtlus

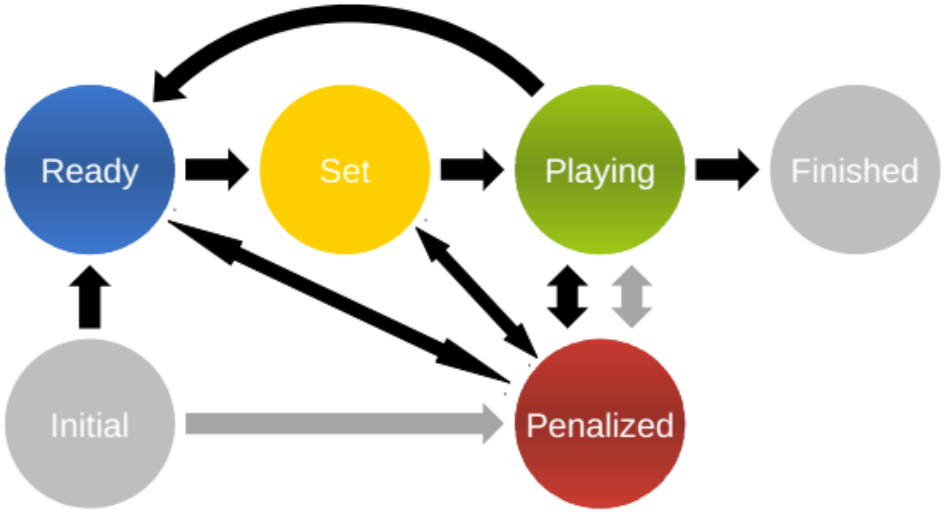
Robotid võivad saata üksteisele sõnumeid, kasutades võistkonna peale WiFi ühendust kuni 500 kb/s. Juhul kui võistkond kasutab suuremat andmevahetuskiirust mitme sekundi vältel, siis antud võistkond diskvalifitseeritakse mängult. Igale meeskonnale jagatakse staatiliste IP aadresside vahemik, mida nad saavad võistluse jaoks kasutada. Sõnumeid saadetakse kasutades UDP (*User Datagram Protocol*) protokolli ja need peavad ülesehituselt realiseerima SPL standardset sõnumipaketi päisefaili. Täiendavad piirangud on rakendatud treener-roboti ja väljakumängijate vahelisele suhtlusele: tema jaoks on eraldi standardne sõnumipakett, mis määrab sõnumi maksimaalseks suuruseks 20 baiti ja peab minema läbi GameControlleri (vt ptk 3) liidese. Sõnum peab olema inimestele loetav ja sõnumeid võib saata kõige rohkem iga 10 sekundi tagant ning GameController lisab igale paketile 3-6 sekundi pikkuse hilinemise. Akustilisi signaale võivad väljakumängijad kasutada piiramatult, kuid treenerile on need keelatud. [20]

2.2.4. Peamine lõplik olekumasin

Roboteid juhib abikohtunik, kasutades GameControlleri liidest või füüsiliselt vajutades nuppe roboti jalgadel ja rinnus. Et abikohtunik oskaks erinevate võistkondade roboteid juhtida, peavad nad realiseerima olekumasina, kus on järgmised olekud:

- *Initial* on olek, kuhu robotid lähevad nende käivitamisel. Selles olekus saab nuppude abil vahetada roboti võistkonda, igasugune liikumine peale püstitõusmise on keelatud.
- *Ready* olekus kõnnivad robotid oma algpositsioonidele.
- *Set* olekus ootavad robotid palli lahtilööki, lubatud on ainult pead liigutada.
- *Playing* olekus mängivad robotid jalgpalli.
- *Penalized* olekus on robotid, kes on saanud karistuse. Igasugune liigutamine on keelatud.
- *Finished* olekus on robotid, kui poolaeg või mäng on läbi.

Ready, Set ja *Finished* olekuid ei saa kasutada, kui roboti olekuid saab muuta ainult füüsiliselt nuppudega. Sel juhul paigutatakse robotid reeglite poolt defineeritud algpositsioonidele, mis on ebasoodus tingimus mängu alustamiseks, sest vastased teavad neid positsioone ning väga väike osa väljakust on mängijate poolt mängu alguses kaetud. [20]



Joonis 3. Peamise lõpliku olekumasina skeem. [20]

Joonisel 3 on kujutatud skeem mängu peamisest olekumasinast ja võimalikest olekutevahelistest üleminekutest, kusjuures hallide nooltega on märgitud üleminekud, mida on võimalik teha nuppudega, ning mustade nooltega üleminekud, mida saab teha

kasutades GameControlleri liidest. Vastavalt Joonisel 3 kujutatud värvidele peab roboti rinnus olev RGB LED andma tagasisidet, mis olekus robot parasjagu on. Halli värvi olekutes on LED välja lülitatud. [20]

2.2.5. Keelatud võtted ja karistus

Keelatud on igasugune liikumine, mis ei toimu kahel jalal. Lisaks on keelatud füüsilise kontakti loomine kahe roboti vahel, palli käega puutumine, enda käes palli hoidmine selliselt, et vastased ei pääse sellele ligi kauem kui 3 sekundit väljakumängijate ja 10 sekundit väravavahi puhul. Kui pikali kukkunud robot ei alusta katset püsti tõusmiseks 5 sekundi vältel, saab see samuti karistuse. Keelatud on ka võtta sisse selline poos, mis on laiem kui roboti õlgade laius, pikemaks ajaks kui 5 sekundit. [20]

Kui mängija on teinud väljakul vea, siis paneb kohtunik selle *Penalized* olekusse ja eemaldab väljakult 45 sekundiks. [20]

2.2.6. Reeglite ajalugu

Kuna SPL eesmärgiks on võistelda kunagi robotitega inimeste vastu, on iga aasta reegleid kohandatud selliselt, et need sarnaneksid inimeste jalgpalli reeglitele [4]. Käesoleva töö raames on oluline võrrelda omavahel reegleid, mis kehtisid 2012. aastal ning reegleid, mis kehtivad 2014. aasta võistlusel, et Austin Villa koodibaasi põhjal valmiv jalgpallitarkvara vastaks uuendatud reeglitele. Olulisemad muudatused, mis on muutnud reegleid sarnasemaks inimjalgpallile, võrreldes 2012. aasta võistlusega on järgmised:

- Väljaku ning keskringi mõõtmeid on suurendatud ja penalti löömise koht on toodud väravale lähemale.
- Mängijate arvu on suurendatud neljalt viiele.
- Robotite võistkondi eraldavad vööd on asendatud särkidega, milles on augud sonarite jaoks.
- Lisandus võimalus kasutada kuuendat robotit mängu jälgijana (treenerina). [20,21]

2.3. *Drop-in Player* võistlus ja tehnilised väljakutsed

Võistkond Philosopher osaleb ka *Drop-in Player* võistlusel ja plaanib osa võtta tehnilistest väljakutsetest. Siin alampeatükis kirjeldatakse lühidalt nende põhimõtet ja reegleid.

Drop-in Player võistluses moodustatakse võistkonnad erinevatest meeskondadest pärit robotitest. Võistluse eesmärk on, et RoboCupil osalevad meeskonnad arendaksid välja mängijad, kes on head võistkonnakaaslased tundmatutele robotitele. Punkte saavad robotid mängutulemuste põhjal (lõplik väravate vahe mängudest) ja kohtunike hinnatud punktide alusel. Kohtunikud hindavad positiivselt söötmist ja söödu vastuvõtmist, negatiivselt meeskonnakaaslase lükkamist ning karistuse saamist. Lisaks võivad nad korra poolajal anda boonuspunkte või ära võtta punkte, mida nad peavad põhjendama. [20]

Tehnilisi väljakutseid on 2014. aastal kolm: vaba väljakutse, suvalise koha väljakutse ja helituvastuse väljakutse. Esimese väljakutse peavad võistkonnad ise välja mõtlema, kuid see peab olema tihedalt seotud SPL peavõistlusega. Demonstratsiooniks on igal võistkonnal aega 3 minutit ja see peab toimuma reaalajas päris robotitel. Ettekannet hindavad teised võistkonnad. Suvalise koha väljakutseks on palli löömine väravasse suvalises kohas ja pinnal. Valgustingimusi ning pinnase ja ümbruskonna värve eelnevalt ei tea. Helituvastuse väljakutse ajal peavad 3 robotit, kes üksteist ei näe, tundma ära tuttavaid helisid ning nende tuvastamisel käe tõstma. Helituvastuse väljakutse eesmärgiks on järgnevatel RoboCup SPL võistlustel eemaldada WiFi suhtluse vajalikkus. [22]

3. Töökeskkond ja -vahendid

Tööks valiti XUbuntu 12.04 32-bitine virtuaalmasinas jooksev operatsioonisüsteem. Kuna vajalike teekide ülesseadmine ja konfigureerimine on töömahukas, siis võimaldab virtuaalmasin seda protseduuri läbi viia vaid korra kõikide inimeste jaoks, kes hiljem projektiga liituvad. Uute inimeste projektiga liitumisel saab neile anda koopia virtuaalmasina tõmmisest.

Käesolevas projektis kasutatakse kahte programmeerimiskeelt: C++ ja Lua. C++i kasutatakse peamiselt liikumis-, kõndimis-, positsioneerimis- ja nägemismoodulis ning Luat käitumismoodulis. Sellisel ülesehitusel on mitmeid eeliseid. Esiteks võimaldab C++ programmeerijal ise mälu hallata, mis teeb tema kasutamise madalama astme koodis kasulikuks. Teiseks on Lua interpreteeritav keel, mis teeb roboti käitumisloogika programmeerimisel muudatuste kontrollimise kiireks, kuna Lua koodi ei pea iga kord kompileerima enne robotisse või simulaatorisse saatmist. Võrreldes näiteks Pythoni keelega, kasutab Lua vähem mälu ja Lua interpretaator töötab kiiremini, mis piiratud ressurssidega projektis on mõlemad olulise tähtsusega [23].

Lua ja C++ on omavahel seotud kasutades SWIGi (*Simplified Wrapper and Interface Generator*), mis on avatud lähtekoodiga tarkvaraarenduse tööriist. SWIG ühendab C ja C++ keeltes kirjutatud programmid interpreteeritavate keeltega, nagu näiteks Perl, Python, Ruby ja Lua. SWIG võtab päisefailidest leitud deklaratsioonid ja loob neist mähiskoodi, mida interpreteeritavad keeled suudavad kasutada. [24]

GameController on Bremeni ülikooli võistkonna B-Human arendatud Java tööriist, millega saab võistlevatele robotitele korraga käske saata. GameControllerit kasutab võistlusel kohtunik ning läbi GameControlleri käib ka suhtlus treener-roboti ja meeskonna vahel. Lisaks kasutatakse GameControllerit publikule mänguseisu kuvamiseks. [20]

SimSpark on kolmemõõtmeline multiagent simulatsioonisüsteem, mille abil saab simuleerida robotitevahelist suhtlust, liikumisi ning füüsikat. SimSparki kasutatakse ka Robocup 3D simulatsiooniliigas ametliku võistluskeskkonnana. [25]

UTNaoTooli autoriks on Austin Villa võistkond. UTNaoTooli kasutatakse selleks, et luua uusi ja siluda vanu värvitabeleid ning roboti pea nurki kalibreerida, et kauguste hindamine ja objektide tuvastamine oleks võimalikult täpne. Sellega saab tellida voogesitusi roboti kaameratest ja sensoritest. UTNaoTool võimaldab oma vaate *World* (Maailm ingl. k.) abil jälgida reaalajas roboti positsioneerimisinfot, meeskonnakaaslaste UDP pakettidest saadud robotite ja palli asukohainfot ning seal on ka sisseehitatud funktsionaalsus käitumisloogika simuleerimiseks väljakul. Samuti saab UTNaoTooli abiga saata robotisse uut koodi ning vahetada roboti olekut peamises lõplikus olekumasinas.

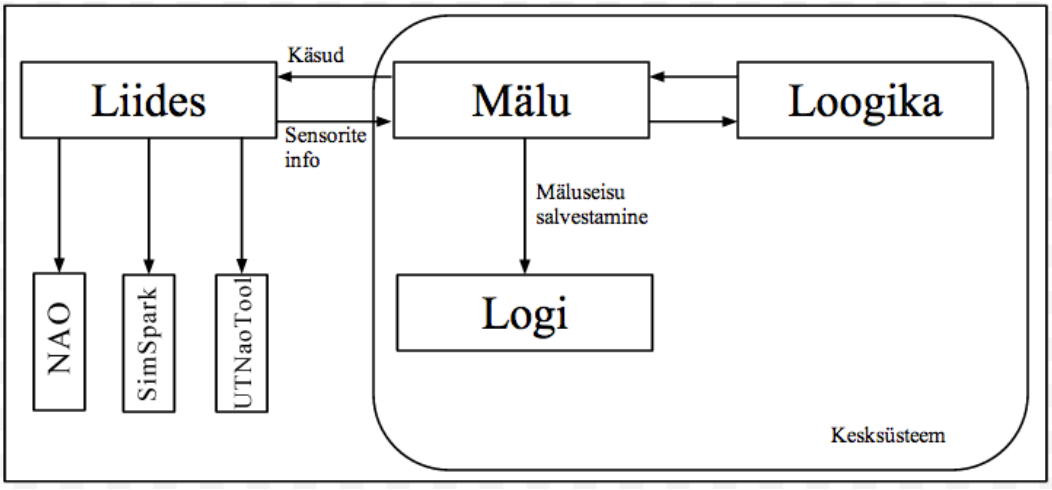
Choreographe on Aldebaran Roboticsi poolt arendatud visuaalprogrammeerimise keskkond NAOde jaoks. Selle abil saab koostada liigutusi, animatsioone ja käitumisi robotile ning testida neid simuleeritud robotil või käivitada päris robotil. Antud projekti raames kasutatakse Choreographe *Timeline* funktsiooni, mille abil saab luua animatsioone ja poose, määrates ära võtmekaadrid teatud ajahetkedel. [26]

4. Ülevaade kasutatava Austin Villa koodibaasi ülesehitusest

Texase Ülikooli võistkond Austin Villa võitis 2012. aastal RoboCup SPL võistluse ning seetõttu valiti just selle võistkonna avalikustatud koodibaas võistkonna Philosopher projekti aluseks [27].

4.1. Arhitektuur

Austin Villa võistkonna arvates üks kõige tähtsamaid aspekte, mida RoboCupi tarkvara arendamisel jälgida, on koodi testitavus ja silutavus. Seepärast on nende koodibaas üles ehitatud selliselt, et iga moodulit oleks võimalikult lihtne eraldi testida, modifitseerida ja siluda. Selleks on kood jagatud mooduliteks, kusjuures rangelt hoitakse üksteisest lahus liides, agendi mälu ja moodulite loogika. Tänu sellele on võimalik lihtsalt ühendada sama kesksüsteemi külge erinev liides, ilma et tekiksid olulised erinevused koodis (vt Joonis 4). Kuna kogu andmevahetus ja käskude saatmine käib läbi mälu, siis on võimalik iga olukorda taasmängida, salvestades selle hetke mäluseisu logi failidesse ning hiljem logifaile UTNaoToolis vaadata. [27]



Joonis 4. Tarkvara arhitektuur, kus hoitakse lahus liides, mälu ja loogika. [27]

Robotis jookseb korraga kolm uut protsessi. NAOqi-sse (vt ptk 1.3.1.) on lisatud enda teek nimega *Naointerface*, mis saadab käske läbi vahendaja ja loeb sisse uuenenud liigeste ning sensorite infot. Teiseks protsessiks on *Motion* (ingl. k. liikumine), mis loeb 100 korda sekundis liigeste olekud ja saadab uued käsud *Naointerface* kaudu

liigestele. See protsess vastutab liikumisega seotud käskude eest, näiteks kõndimine ja löögid. Viimaseks protsessiks on *Vision* (ingl. k. nägemine), mis töötleb pilti 30 kaadrit sekundis ja juhib kõrgema taseme käitumist, objektituvastust ja roboti lokaliseerimist väljakul. Iga protsess on eraldi kompileeritud ja töötab teistest sõltumatult, mis võimaldab ka liikumised hoida suuremal töösagedusel kui pilditöötlus. Andmete jagamine toimub läbi jagatud mälu. [27]

4.2. Loogikamoodulid

Joonisel 4 kujutatud loogikaplokk on jagatud neljaks suuremaks mooduliks: nägemine, positsioneerimine, käitumine ja liikumine, millest esimese kolme eest vastutab *Vision*i protsess ja viimase eest *Motion*.

4.2.1. Nägemismoodul

Nägemismoodulis toimub pilditöötlus neljas etapis, mida sooritatakse mõlemal kaameral eraldi:

1. segmenteeritakse töötlemata YUV värviruumis kaamerapilt, kasutades selleks värvitabelit, kus on eeldefineeritud segmentide YUV väärtused üks-üheses seoses segmendi värvidega. Neid väärtusi võrreldakse kaamerapildist saadud andmetega ning kokkulangevuse korral lisatakse piksel segmenti. [28]
2. Otsitakse segmenteeritud pildist horisontaalsete ja vertikaalsete piksliridade abil laike sama värvi objektidest. Horisontaalseid ridasid luuakse iga teise ja vertikaalseid iga neljanda piksli tagant. Nende otsinguridade abil moodustatakse laigud pikslitest nendes ridades ja nende ümber, millel on samad segmenteeritud värvide väärtused. Kattuvad sama värvi laigud ühendatakse üheks laiguks. [27,28]
3. Laike analüüsitakse objektituvastusalgoritmidega. Joonetuvastusalgoritmiga identifitseeritakse väljakul jooned, ringjoon ja väravad. Vertikaalsed kollased laigud on kandidaadid väravapostideks ning kõik valged vertikaalsed ning horisontaalsed laigud on kandidaadid sirgeteks joonteks ja ringjooneks. Kandidaatide leidmisel tehakse lihtsustatud eeldus, et iga joonesegmenti väljakul saab kujutada valemiga $y = ax^2 + bx + c$. Selle valemiga saab luua nii kurvi kui ka sirgjoone, kasutades ühte punkti, tõusu ja tõusu muutumise kiirust. Loodud

kurvi punkte võrreldakse laigu punktidega. Palli tuvastamiseks kasutatakse kahte heuristikut: oranži laigu sarnasust ringiga ja laigu kõrgust kolmemõõtmelises ruumis. Oranžid laigud paigutatakse kolmemõõtmelisse ruumi, arvutades palli kauguse kaamerast ja transformeeritakse väljakule. Iga objekti puhul valitakse parima kattuvusega kandidaat ning kui kattuvus on suurem kui objekti tuvastamise lävend, siis loetakse see tuvastatuks. [28]

4. Roboti kaamera asendi andmed arvutatakse roboti kaamera kõrgusest ja orientatsioonist. Nende abil transformeeritakse punktid kaamerapildist tehtud maatriksist väljakupunktide maatriksile saades teada nende kauguse robotist. [27]

4.2.2. Positsioneerimismoodul

Positsioneerimismoodulis arvutatakse kõikide robotite, sealhulgas iseenda, palli, väljakujoonte ja väravate positsioonid, orientatsioonid, liikumiskiirused ja kaugused. Asukohtade arvutamiseks võetakse nägemismoodulis tuvastatud väljakuelemendid ja antakse need sisendiks *Unscented Kalman Filter* (Lõhnastamata Kalmani filter ingl. k., edaspidi UKF) moodulile.

UKF moodulis luuakse mitmetähenduslike vaatluste või mitme võimaliku lõpptulemusega tegevuste põhjal asukohamudelite hüpoteesid, kus robotid ja pall võiksid asuda. Mitmetähenduslikuks vaatluseks loetakse üksiku väravaposti, joone või joonte ristumise nägemist ning mitme võimaliku lõpptulemusega tegevuseks loetakse näiteks palli löömist, kus tulemused võivad olla, et robot lõi pallist mööda või pallile pihta. Ühe väravaposti nägemisel on kaks võimalust, millega on tegu: kas vasakpoolse või parempoolse väravapostiga. Joone korral on erinevate võimalike objektide arv 11 ja joonte ristumistel on see arv 16 (vt Lisa 2). Iga võimaliku objekti jaoks luuakse uus asukohamudeli hüpotees ning igale hüpoteesile määratakse kaal, mis väljendab, kui tõenäoline on roboti paiknemine antud mudelis. Kaalude määramisel võetakse arvesse roboti kõndimisel kogutud odomeetria informatsiooni [29]. Ühetähendusliku vaatluse korral uuendatakse kõiki hetkel aktiivseid asukohamudeleid ning arvutatakse ümber ka nende kaalud. Kõik asukohamudelid, mille tõenäosus on alla teatud lävendi, hüljatakse kui ebatõenäolised ning kõige tõenäolisemad asukohamudelid liidetakse kokku, arvestades nende kaalusid. Kokku liidetud mudel võetakse kasutusele käitumisloogikamoodulis. [28]

4.2.3. Liikumismoodul

Liikumiseks kasutatakse Bremeni Ülikooli B-Humani meeskonna kõndimismootorit. Liidestamiseks ülejäänud projektiga, on kasutatud selle modifikatsiooni, mille autoriks on Bowdoin College Northern Bites võistkond [27]. Liikumismoodul on jagatud kaheks osaks: *Sensing* (ingl. k. tuvastamine) ja *Motion Control* (ingl. k. liikumise juhtimine). Esimene neist vastutab sensoritelt loetavate andmete eeltöötlemise eest. Teine koostab liigeste nurgad, mis saadetakse robotile täitmiseks. [30]

Sensing osa liikumismoodulist loeb kõigepealt sisse IMU (*Inertial measurement unit)* kiirendusanduri ja kahe güroskoobi andmed, seejärel loetakse sisse liigeste nurgad. Mõlemaid andmehulki kontrollitakse, et nad ei sisaldaks ootamatult suuri või väikeseid mõõtmistulemusi, mis vajavad eraldi töötlemist. Pärast seda filtreeritakse mõlemad andmehulgad. Kuna roboti liigeste nurkade info on piisavalt täpne, siis filtreerimisel kontrollitakse kõigest, et on olemas andmed igalt liigeselt. IMU-st saadud andmed sisaldavad müra ja vajavad eraldi tarkvaralist filtreerimist. Pärast filtreerimist luuakse roboti liigeste nurkade info põhjal lihtsustatud roboti asendi mudel ja arvutatakse välja roboti massikese. Kasutades nii roboti asendi mudelit kui ka filtreeritud sensorite andmeid, arvutatakse välja roboti torso orientatsioon maapinna suhtes. Jalgade rõhusensorite ja torso orientatsiooni järgi on võimalik teada saada, kas robot on pikali kukkunud ja kui on, siis millises asendis. [30]

Motion Control osa liikumismoodulist genereerib liigeste nurkade infot, mis on vajalik robotile kõndimiseks, püsti tõusmiseks ja seismiseks. Kõndimise liigutused genereeritakse dünaamiliselt inverteeritud pendli mudeli abil, et arvutada soovitud trajektoor roboti massikeskme jaoks. *Sensing* osa poolt arvutatud massikeset jälgitakse *Motion Control*is kogu aeg ning stabiilsuse tagamiseks tehakse parandusi soovitud trajektooris. Püsti tõusmise ja seismise jaoks on robotil staatilised liigeste nurkade järjestused, mis võetakse sisse ettemääratud kindlate ajavahemike tagant. [30]

4.2.4. Käitumisloogika moodul

Käitumisloogika moodulis otsustavad robotid, millist strateegiat nad mängimisel rakendavad, ja peavad selle ka realiseerima, kasutades käitumisloogikamoodulis

kirjeldatud oskusi ja poose. Robotite käitumisloogika koosneb käesolevas projektis lõplikest olekumasinatest, kusjuures iga lõpliku olekumasina olekus võib sisalduda teine lõplik olekumasin. Kõige kõrgema astme olekumasinaks antud projektis on reeglite poolt määratud peamine lõplik olekumasin (vt ptk 2.2.4.).

4.2.4.1. Initsialiseerimine ja täitmise järjekord

Käitumisloogika initsialiseerimisel luuakse ühendus C++ koodi ja Lua vahel. Seejärel initsialiseeritakse positsioneerimismooduli parameetrid ja tehakse käitumiskoodis saadavaks jagatud mäluplokid. Esimesel kaadril, kui nägemismoodul kutsub käitumismooduli välja, loetakse sisse kumba värvi särke võistkond kannab ja väärtustatakse kaamera, löömise ning kõndimise parameetrid.

Igal kaadril töödeldakse esimesena kaamerapilt ja loetakse sisse liigeste asendid. Järgmisena toimub töödeldud kaamerapildi põhjal positsioneerimine. Kõige viimasena toimub käitumisloogikas roboti edasiste tegevustega seotud otsuste tegemine ja nende otsuste avalikustamine teistele moodulitele täitmiseks.

4.2.4.2. Ülesanded

Igas olekus täidab robot mingit ülesannet, kasutades selleks eeldefineeritud oskusi ja nendevahelisi tingimusi. Igal ülesandel on kolm osa: initsialiseerimine, täitmine ja lõpetamine. Initsialiseerimisel saab muuta ülesandespetsiifilisi parameetreid, täitmisel täidetakse ülesanne ning lõpetamisel märgitakse ära tingimus, mille korral ülesanne on täidetud. Peamine erinevus funktsiooni ja ülesande vahel on asjaolu, et ülesannet täidetakse seni, kuni lõpetamise tingimus on täidetud. Seetõttu on ülesannetel ka mälu, mida funktsioonidel ei ole. Ülesanded on jagatud neljaks: FuncTask, PrimTask, ParaTask ja CompTask. [28]

FuncTask ehk funktsionaalne ülesanne on funktsioon, mis omab mälu. Funktsionaalse ülesande puhul ei ole oluline, kas ülesanne tagastab mingi väärtuse või mitte. FuncTaskiks on näiteks kõndimise parameetrite seadistamine, kus on oluline sätete mäletamine. FuncTask on lisatud käitumisloogika arhitektuuri võistkonna Philosopher poolt.

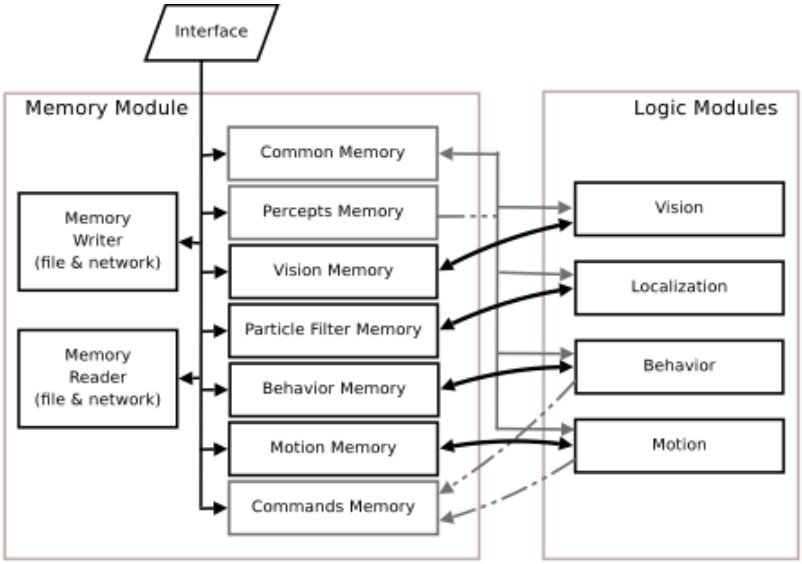
PrimTask ehk primitiivne ülesanne tegeleb vaid ühe kindla, võrdlemisi lihtsa funktsiooni täitmisega ja teisi ülesandeid välja ei kutsu. PrimTaske kasutatakse mäluplokkidesse info kirjutamiseks või madalama taseme käskude välja kutsumiseks. PrimTaskid ei tagasta täitmisel väärtusi. PrimTaskiks on näiteks kõndimine ühte kindlaksmääratud asukohta väljakul.

CompTask ehk liitülesanne koosneb mitmest alamülesandest, mis kutsutakse välja teatud tingimuste täitumisel. CompTask, erinevalt PrimTaskist, peab igal täitmisel tagastama mingi teise ülesande. CompTask on näiteks palli löömine, mille korral robot kõigepealt läheneb pallile, seejärel sihib ning viimaks lööb palli soovitud suunas.

ParaTask ehk paralleelne ülesanne on ülesanne, mille korral täidetakse mitut ülesannet paralleelselt ning pole oluline, mis tüüpi ülesannetega tegu on. ParaTaskiks on näiteks mingisse sihtkohta kõndimine ning samaaegselt pea pööramise abil roboti ümber toimuva uurimine.

4.3. Mälu

Igal moodulil on oma lokaalne mäluplokk, kus nad saavad hoida moodulisiseseks kasutamiseks mõeldud informatsiooni [27]. Moodulitevaheline andmevahetus toimub läbi ühise jagatud mälu. Tegu on niinimetatud tahvelmeetodil andmevahetuse pidamisega, mille suurim puudus on turvalisus: kuna kõikidel moodulitel on ligipääs jagatud mälule ning nad kõik sõltuvad sellest, siis kui üks moodul rikub andmed jagatud mälus, on kogu süsteemi töö häiritud.

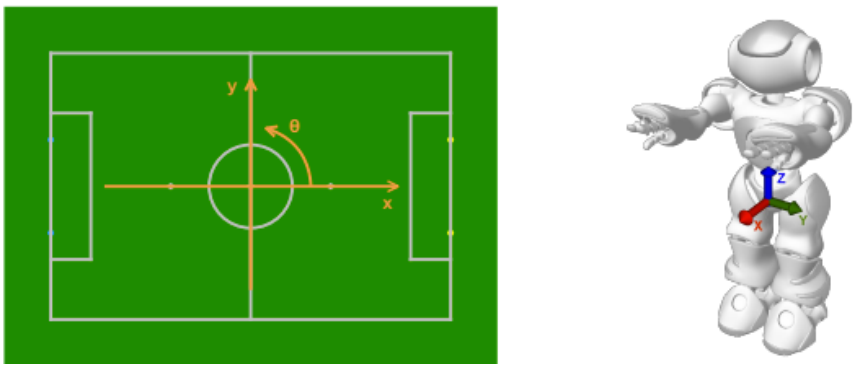


Joonis 5. Mälu mooduli disain ja suhtlus loogikamoodulitega. [27]

Joonisel 5 on kujutatud mälu mooduli disain. Hallid kastid kujutavad mäluplokke, mida loeb/kirjutab mitu loogikamoodulit samal ajal. Punktiirjoonega on ühendatud moodulid, mis on ainult kirjutatavad või loetavad. Mustad kastid kujutavad mäluplokke, mis on mõeldud igal moodulil lokaalseks kasutamiseks. [27]

4.4. Koordinaatsüsteemid

Käesolevas projektis on kasutusel kaks koordinaatsüsteemi: globaalne ja roboti asukoha suhteline. Globaalse koordinaatsüsteemi alguspunkt on väljaku keskpunkt. Abstsisstelg on suunatud vastase värava poole ning kui vaadata x-telje kulgemise suunas on ordinaattelg suunatud vasakule. Z-telg on suunatud üles. Roboti asukoha suhtelise koordinaatsüsteemi alguspunktiks on tema massikese ning x-telg on suunatud roboti vaatamise suunas, y-telg temast vasakule ning z-telg üles.



Joonis 6. Globaalne koordinaatsüsteem ja roboti asukoha suhteline koordinaatsüsteem (vastasvõistkonna väravapostid on kujutatud kollaselt). [30]

5. Tehtud töö

Käesoleva bakalaureusetöö raames uuriti Austin Villa 2012. aastal avalikustatud koodi ja selle ülesehitust. Pärast seda sooritati robotitele testid süsteemi vastupidavuse ning objektituvastuse piiride teada saamiseks. Järgmisena loodi robotitevahelise võrguühenduse katkemisel kasutatav käitumisstrateegia koos vajalike abiülesannetega. Viimasena uuriti võistkondade B-Human ja Austin Villa meeskonnastrateegiaid ning pakuti välja lahenduse idee, kuidas loodud individuaalstrateegia edasi arendada meeskondlikuks strateegiaks.

5.1. Testimine

Antud peatükis tutvustatakse robotile sooritatud kestvustesti ja objektituvastustesti metoodikat ning tehakse lühike ülevaade saadud tulemustest. Objektituvastustest sooritati koostöös võistkonnaga Philosopher.

5.1.1. Kestvustest

Kestvustest sooritati roboti riistvara vastupidavuse teada saamiseks stressiolukorras. Testi ajal kõndis robot kohapeal, pea vaatas ette ning mõõdeti roboti enda sensoritega kõige kuumema liigese temperatuuri ning akutaset. Test sooritati kolmel robotil ning tulemused on toodud tabelis 2. 70 °C temperatuuri loetakse Austin Villa koodis juba robotile ohtlikuks ning seetõttu hakatakse alates sellest temperatuurist alandama roboti mootoritesse saadetavat voolu, et vältida roboti ülekuumenemist.

Tabel 2. Kestvustesti tulemused

Aeg (min)		**0**	**5**	**10**	**15**
Robot	**Mõõdetav suurus**				
Boole (192.168.1.15)	Akutase (%)	100	100	91,9	86,7
	Temperatuur (°C)	38	47	60	74
Socrates (192.168.1.18)	Akutase (%)	100	100	100	93
	Temperatuur (°C)	39	47	61	75
Locke (192.168.1.12)	Akutase (%)	100	100	98,5	90
	Temperatuur (°C)	38	48	68	79
Keskmine	**Akutase (%)**	**100**	**100**	**96,8**	**89,9**
	Temperatuur (°C)	**38,3**	**47,3**	**63**	**76**

Pärast testi sooritamist pandi robot istuvasse asendisse ning aku laadima. Kümne minuti möödumisel oli temperatuur alanenud kõikidel robotitel 14-15 °C võrra ning akutase oli 100%.

5.1.2. Objektituvastuse test

Töö käigus sooritati täismõõtmetel väljakul objektituvastuse test, et teada saada nägemismooduli piirid. Testimise ajal seisis robot püsti, vaatas objekti poole ja ei liigutanud. Roboti pea nurk oli vaikimisi olekus. Objekt loeti tuvastatuks, kui robot tegi selle kindlaks vähemalt ühe korra kümne kaadri jooksul. Tulemused on järgnevad:

- Palli tuvastas robot kuni kuue meetri kauguselt.
- Väljaku keskringi tuvastas robot kuni kolme meetri kauguselt.
- Väravaposte tuvastas robot terve väljaku ulatuses.
- Äärejooni tuvastas robot kuni kahe meetri kauguselt.

5.2. Väljakuobjektidega kokkupõrgete vältimine

Teiste robotite ja väravapostidega kokkupõrgete vältimiseks loevad robotid kõndimise ajal roboti rinnus asuvate sonarite 10 viimase mõõtmise keskmistatud väärtusi (aritmeetiline keskmine). Tuvastades takistuse roboti ees lähemal kui 45 cm, lõpetab robot otsesuunas liikumise ning vastavalt sellele, kummast sonarist takistus tuvastati, hakatakse külg ees liikuma vastassuunas seni, kuni takistus on kaugemal kui 45 cm. Kui takistus on mõlemast sonarist tuleva info kohaselt otse roboti vastas, liigutakse väljaku keskpunkti suunas.

5.3. Individuaalstrateegia

Individuaalstrateegiaks nimetatakse mängustrateegiat, kus iga robot mängib iseseisva mängijana ja ei suhtle oma võistkonnakaaslastega. Individuaalstrateegia eesmärk on, et robotitel oleks olemas lihtne, robustne ja teistest võistkonnakaaslastest sõltumatu mänguplaan, kuhu taanduda peamiselt strateegialt juhul, kui robotitevaheline ühendus kaob või ei ole kasutamiseks piisavalt stabiilne. Kuigi reeglitekohaselt on keelatud võistluspaigas viibida 2,4 GHz sagedusel töötavate raadioseadmetega, on eelmistel aastatel RoboCup võistlusel olnud probleeme robotitevahelise ühenduse kadumisega

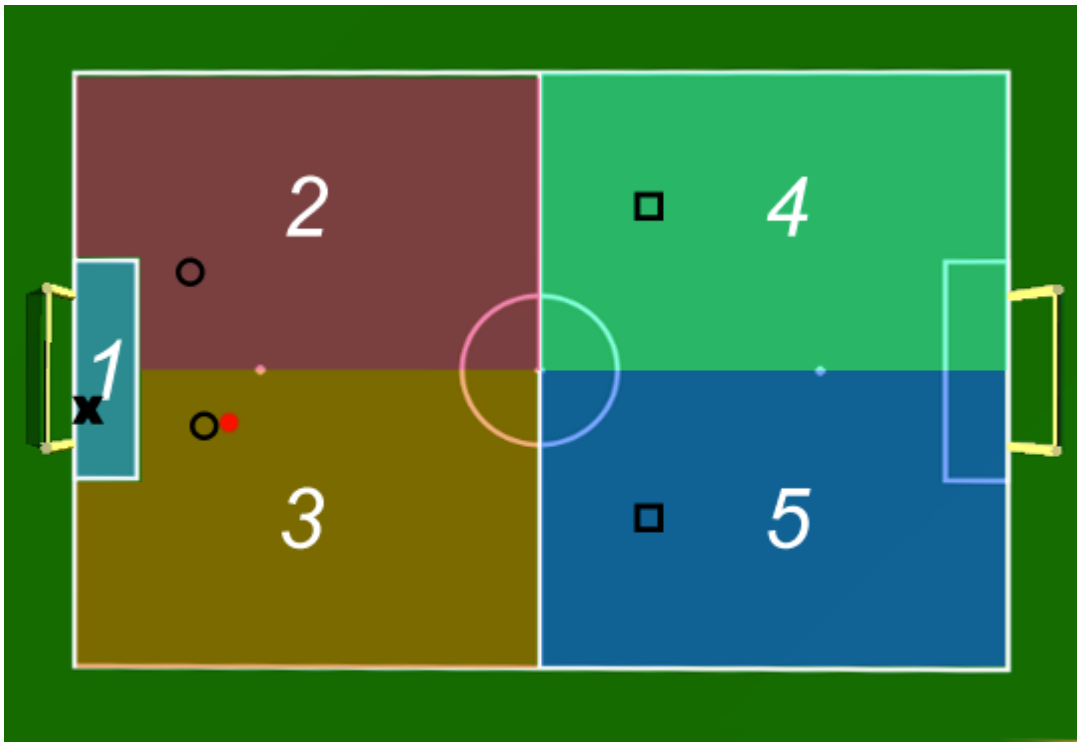
mängu ajal [27,20,31]. Individuaalstrateegia loomisel võeti arvesse objektituvastus- ja koormustestide tulemusi. Nende põhjal positsioneeritakse robot väljakul selliselt, et palli on võimalik kogu aeg tuvastada ning robot ei peaks väljakul liigset kõndima.

Tsooni klassi koostas individuaalstrateegia jaoks Philosopher võistkonna liige Siim Schults ja väravavahi käitumist aitas realiseerida võistkonna liige Viljar Puusepp.

5.3.1. Teiste võistkondade individuaalstrateegiad

Eelmistel aastatel osalenud võistkonnad on ühenduse kadumisel kasutanud mitmeid erinevaid mängustrateegiaid. Osad võistkonnad eemaldasid väljakult kõik robotid peale ühe, jäädes vabatahtlikult vastase suhtes arvulisse vähemusse. Kasutati ka väga lihtsat strateegiat – kui robot nägi palli, siis see lihtsalt jooksis palli poole ja lõi seda vastase värava suunas. Austin Villa võistkond kasutas 2012. aastal sellist mänguplaani, kus üks robot ajas palli taga, samal ajal kui teised robotid seisid liikumatult. Kui nende kaugus pallist oli piisavalt väike, võisid ka nemad palli taga ajada. [27,32]

5.3.2. Individuaalstrateegia üldreeglid



Joonis 7. Mänguväljak on jagatud viieks tsooniks. Ründajad on märgitud ruutudena, kaitsjad ringidena ja väravavaht ristina.

Väljak on jagatud viieks tsooniks (vt Joonis 7), kus esimene tsoon on väravavahi, tsoonid 2 ja 3 on kaitsjate ning 4 ja 5 ründavate robotite omad. Tsoonid on defineeritud kui ristkülikute hulgad. Igal tsooni küljel, mis külgneb teise tsooniga, on väljaku joone laiune (5 cm) kattuvus. Mänguväljakul ei ole mängijate vahel jagamata ala. Iga mängija peab alati püsima enda tsoonis ja kui pall on selle tsoonis, siis robot lööb palli vastase värava suunas. Positsioneerimisel peab robot mängides eeldama, et see on oma tsoonis, kui robot just ei tulnud *Set* või *Penalized* olekust. Kui palli ei ole robotile määratud alas, siis see peab käituma vastavalt oma rollispetsiifilistele reeglitele.

Kõndides pööravad robotid horisontaalsete liigutustega pead. Kui pall on robotile piisavalt lähedal, siis hoitakse kaamerapilt pallil.

5.3.3. Rollispetsiifilised reeglid

Väravavaht – Kui pall on vastaste poole peal, tuleb väravavaht võimalikult kaugele väravast välja ja püsib väravapostide keskel, et katta võimalikult suur ala väravast enda taga, raskendades nõnda vastaste võimalusi kaugelt värava löömiseks. Mänguolukorra jälgimiseks hoiab robot kaamerad suunatud pallile või kui palli ei ole võimalik näha, teeb peapöördeid. Juhul kui pall on oma väljakupoolel, taganeb see värava piirile ja üritab ennast positsioneerida palli ja värava vahele ning kui see näeb palli enda suunas liikumas, siis robot kukutab ennast kas vasakule või paremale vastavalt sellele, kummale poole pall tema suhtes liigub. Kui pall tuleb otse väravavahi suunas, siis robot kükitab.

Kaitsjad – Kui pall on vastase poole peal, seisavad robotid tsoonides 2 ja 3 x-teljel keskringi lõpu lähedal, y-teljel väravavahi ala lõpu juures ja jälgivad palli. Juhul kui pall on meie väljakupoolel, üritab kaitsja aidata teist kaitsjat ja väravavahti, positsioneerides ennast enda värava ja palli vahele, nagu teeb kaitsja tsoonis 2 joonisel 7.

Ründajad – Juhul kui pall on vastase poolel, püsib ründaja väljaku keskel ning hoiab end oma värava ja palli vahel, et mitte takistada teise ründaja, kelle tsoonis pall parasjagu on, värava löömise võimalust. Juhul kui pall on oma võistkonna väljakupoolel, liiguvad ründajad keskjoone lähedale joonisel 7 mustade ruutudega

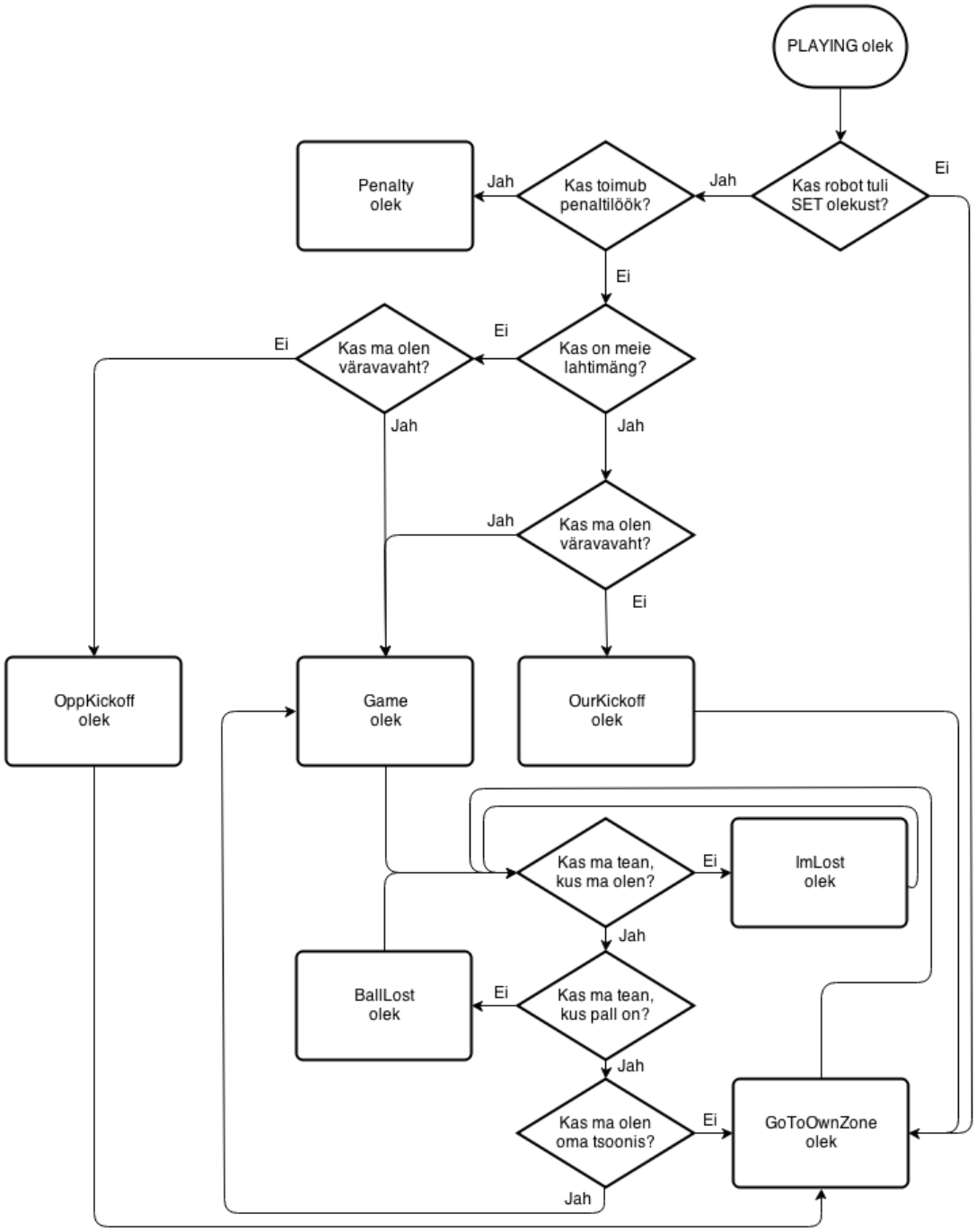
märgitud punktidesse, et hoida kaitsjatele vaba koridor väljaku keskel vastase värava suunas löömiseks ning et olla valmis kontrarünnakuks.

5.3.4. Individuaalstrateegia olekumasin

Individuaalstrateegias on defineeritud olekud *Game, ImLost, BallLost, GoToOwnZone, OurKickoff, OppKickoff* ja *PenaltyKick*.

- *Game* olekus robotid on oma tsoonis ja mängivad jalgpalli vastavalt eelpool mainitud reeglitele.
- *ImLost* olekus ei tea robot täpselt, kus ta asub, ning seisab koha peal ja pöörab pead. 10 sekundi möödudes kõnnib see oma tsooni keskpunkti, samal ajal peaga ümbrust uurides.
- *BallLost* olekus ei tea robot täpselt, kus pall asub, ning otsib seda seistes ja pead pöörates. Sarnaselt *ImLost* olekule kõnnib see 10 sekundi möödumisel oma tsooni keskpunkti, samal ajal peaga ümbrust uurides.
- *GoToOwnZone* olekus kõnnib robot oma tsooni.
- *OurKickoff* ja *OppKickoff* olekutes toimub palli lahtimäng poolaegade alguses, kusjuures esimesse olekusse lähevad robotid siis, kui on nende kord, ning teise siis, kui on vastate kord palli lahtimänguks.
- *PenaltyKick* olekus on robot väravavaht ja kaitseb või on ta ründaja ning teeb pealelööki vastase väravale olenevalt sellest, kumb võistkond ründab.

Joonisel 8 on kujutatud realiseeritud olekumasina üleminekute voodiagramm. Jooniselt on näha, et väravavaht jätab peamise olekumasina *Set* olekust tulles vahele individuaalstrateegia *GoToOwnZone* oleku. *GoToOwnZone* olek pole väravavahile vajalik, kuna kui väravavaht ei suuda kõndida väravavahi alasse *Ready* olekus, tõstab abikohtunik ise ta *Set* olekus sinna [20]. Samuti on jooniselt näha, et peamine mängutsükkel toimub *Game, GoToOwnZone, ImLost* ja *BallLost* olekute vahel.



Joonis 8. Individuaalstrateegia olekumasina olekute üleminekudiagramm.

5.4. Meeskonnastrateegia

Käesolevas alampeatükis tutvustatakse mängustrateegiat, mis erinevalt eelnevalt kirjeldatud individuaalstrateegiale kasutab robotitevahelist suhtlust ja teeb seega võimalikuks mängijatevahelise koostöö ja kokkumängu. Töö meeskonnastrateegia programmeerimiseks käib ning strateegia valmib juuliks 2014. Meeskonnastrateegia

põhineb individuaalstrateegias kirjeldatud tsoonidesüsteemil, kuid eemaldab range positsioneerimise eelduse, et robot on alati oma tsoonis. Rõhku pööratakse targale palli liigutamisele väljakul ning tühjade tsoonide, mille eest vastutav robot on karistuse saanud, elimineerimisele.

5.4.1. Teiste võistkondade meeskonnastrateegiad

Eelmise aasta RoboCup SPL võistluse võitjameeskond B-Human kasutas sama taktikat igas mängus. Mängijatel oli igaühel oma roll: väravavaht, kaitsja, toetaja, tipuründaja ja ründaja. Ründav mängija kõndis alati palli suunas kui võimalik. Toetaja kõndis ründaja taga teatud vahemaa taga. Tipuründaja oli mänguks valmis alati vastase poolel ning kaitsja oma poolel väravavahi ala lähedal. Väravavaht oli kogu aeg oma alas. Rolle vahetati dünaamiliselt vastavalt sellele, millise roboti asukoht palli või väljakupositsiooni suhtes soodus oli. [33]

Austin Villa meeskond kasutas meeskonnastrateegia loomisel dünaamilist rollijaotust koos pakkumise süsteemiga. Iga robot saatis kaaslastele pakkumise, kui väga ta sobib ründajaks, kes läheneb pallile ja lööb seda. Pakkumine tehakse võttes arvesse roboti kaugust, asendit ja nurka palli suhtes ning seda, kas pallile järgi minemine nõuaks oma värava suunas liikumist või vastase värava suunas liikumist. Ülejäänud mängijad jagatakse keskmängija, kaitsja ja tipuründaja rollidesse. Kui väljakul on vähem kui 5 mängijat, jagatakse rollid prioriteedi alusel. Rollide formatsioonide muutmisel võeti arvesse lisaks palli asukohale ka mänguseisu ja -aega. Robotid teavitasid üksteist, kui nad sööta tahtsid, et söötu vastu võttev mängija saaks selleks valmistuda. [34]

5.4.2. Meeskonnastrateegia üldreeglid

Robotid on mängu alguses jagatud tsoonidesse sarnaselt individuaalstrateegiale. Kui üks robot saab karistuse, siis temaga rolli jagav mängija (nt. teine kaitsja) laiendab oma ala tühjaks jäänud tsooni võrra karistuse kestvuse ajaks. Kui väravavaht saab karistuse, võtab üks kaitsja tema rolli karistuse ajaks enda kanda. Robotid võivad teatud tingimuste täitumisel väljuda oma tsoonist, et palli rünnata. Positsioneerimisel, selle asemel, et eeldada oma pidevat tsoonis püsimist, võrdlevad robotid oma arvamust palli asukohast. Sedasi saab lihtsalt vältida suure probleemi tekkimist: kui robotid ei tea kumba väravat rünnata.

Palli saamisel võetakse arvesse vastasrobotite kaugus ning palli võib lisaks vastase värava poole löömisele sööta ka oma meeskonnakaaslasele. Palli söötmisel meeskonnakaaslasele antakse sellest sõnumipaketis teada ja meeskonnakaaslane oskab sellega arvestada ning ennast palli vastuvõtmiseks õiges suunas pöörata.

Penalti löömisel ja peamise olekumasina kõikides teistes olekutes, peale *Playing* oleku, võib kasutada juba individuaalstrateegias välja töötatud lahendust, kuna seal meeskonnatöö ei ole oluline.

Kokkuvõte

Aldebaran Roboticsi arendatud NAO humanoidrobotit kasutatakse jalgpallurina RoboCup võistlusel Standard Platform League, kus kõik robotid on sama riistvaraga ja erinevad ainult tarkvara poolest. RoboCup võistluse eesmärk on populariseerida robootikat ja intellektitehnikat.

Käesoleva bakalaureusetöö eesmärk oli arendada välja RoboCup SPL 2014. aastal toimuva võistluse nõuetele vastav jalgpallitarkvara käitumisloogika, mis põhineb Texase Ülikooli võistkonna UT Austin Villa 2012. aastal avalikustatud koodil.

Töö käigus uuriti Austin Villa koodi ja teiste meeskondade lahendusi, sooritati testid roboti vastupidavuse ja objektituvastuse piiride teada saamiseks ning loodi 2014. aasta võistluse reeglitele vastav käitumisstrateegia, mida on robotitel kasulik kasutada siis, kui robotitevaheline ühendus on katkenud. Loodud strateegias on robotid jagatud tsoonidesse ning kui pall on roboti tsoonis, siis lüüakse see vastase värava suunas. Kui pall ei ole mängija tsoonis, siis liigub ta vastavalt palli asukohale kindlaks määratud staatilistesse punktidesse väljakul.

Töö valmis koostöös Philosopheri meeskonnaga, kes osaleb juulis 2014 Brasiilias toimuval RoboCup võistlusel. Vastavalt võistkonna eesmärkidele propageeriti robootikat Eestis ning sooritati demonstratsioone Robotexil 2013, FIRST® LEGO® League Eesti ja Läti poolfinaalis 2013 ja RoboMiku Lahingus 2014.

Töö lõpus pakuti välja lahenduse idee, kuidas loodud individuaalstrateegiat muuta meeskondlikuks strateegiaks.

Viited

1. "What is a humanoid robot?" Idaho National Laboratory [Võrgumaterjal] https://inlportal.inl.gov/portal/server.pt/community/what_is_a_humanoid_robot_ [Kasutatud: 17.05.2014].
2. Bibby, J. "Robonaut:Home" NASA [Võrgumaterjal] http://robonaut.jsc.nasa.gov/ [Kasutatud: 17.05.2014].
3. "Service & Entertainment Robots" Idaho National Laboratory [Võrgumaterjal] https://inlportal.inl.gov/portal/server.pt/community/current_research_projects/541/service___entertainment_robots/5993 [Kasutatud: 17.05.2014].
4. "Objective « RoboCup" The Robocup Federation [Võrgumaterjal] http://www.RoboCup.org/about-RoboCup/objective/ [Kasutatud 21.10.2013].
5. "Story / Vision | Aldebaran" Aldebaran-Robotics [Võrgumaterjal] http://www.aldebaran.com/en/robotics-company/history [Kasutatud 25.03.2014].
6. Schults, S., Hunt, K., Puusepp, V., Pihlakas, R., Anbarjafari, G., Aabloo, A. "Philosopher SPL Team Description Paper" Tartu Ülikool (2013) [Võrgumaterjal] http://ims.ut.ee/philosopher/docs-assets/docs/Philosopher_tdp14.pdf [Kasutatud: 26.04.2014].
7. "Aldebaran Robotics NAO Specification" Aldebaran-Robotics [Võrgumaterjal] https://community.aldebaran-robotics.com/nao/ [Kasutatud 21.10.2013].
8. Williams, I. H., Whiten, A., & Singh, T. "A systematic review of action imitation in autistic spectrum disorder." *Journal of Autism and Developmental Disorders, 34*, lk. 285-289 (2004).
9. Hart, M. "Autism/excel study" kogumikus *Proceedings of the ASSETS 2005: Seventh International ACM SIGACCESS Conference on Computers and Accessibility* (lk. 136-141). ACM.
10. Tapus, A., Peca, A. Aly, A., Pop, C. Jisa, L., Pintea, S. Rusu, A.S., David, D.O. "Children with autism social engagement in interaction with Nao, an imitative robot – A series of single case experiments" *Interaction Studies, 13 (3),* lk. 315-347 (2012).

11. “OpenNAO - NAO OS — NAO Software 1.14.5 documentation” Aldebaran-Robotics [Võrgumaterjal] https://community.aldebaran-robotics.com/doc/1-14/dev/tools/opennao.html [Kasutatud 25.03.2014].
12. “NAOqi Framework — NAO Software 1.14.5 documentation” Aldebaran-Robotics [Võrgumaterjal] https://community.aldebaran-robotics.com/doc/1-14/dev/naoqi/index.html [Kasutatud 25.03.2014].
13. “DCM Introduction — NAO Software 1.14.5 documentation” Aldebaran-Robotics [Võrgumaterjal] https://community.aldebaran-robotics.com/doc/1-14/naoqi/sensors/dcm/introduction.html [Kasutatud: 07.04.2014].
14. “About RoboCup « RoboCup” The Robocup Federation [Võrgumaterjal] http://www.RoboCup.org/about-RoboCup/objective/ [Kasutatud: 21.04.2014].
15. “Robot League” RoboCup Federation Wiki [Võrgumaterjal] http://wiki.robocup.org/wiki/Robot_League [Kasutatud: 28.04.2014].
16. “RoboCup @Home” The Robocup Federation [Võrgumaterjal] http://www.robocup.org/robocup-home/ [Kasutatud: 28.04.2014].
17. “RoboCup - @Home” YouTube [Võrgumaterjal] [Video] (2012) http://youtu.be/YpjeNa8BAYg?t=50s [Kasutatud: 28.04.2014].
18. “A Brief History of RoboCup « Robocup” The Robocup Federation [Võrgumaterjal] http://www.robocup.org/about-robocup/a-brief-history-of-robocup/ [Kasutatud 26.03.2014].
19. “Standard Platform League History” Bremeni Ülikool [Võrgumaterjal] http://www.informatik.uni-bremen.de/spl/bin/view/Website/History [Kasutatud 26.03.2014].
20. “RoboCup Standard Platform League (NAO) Rule Book” RoboCup Technical Committee (2014) [Võrgumaterjal] http://www.tzi.de/spl/pub/Website/Downloads/Rules2014.pdf [Kasutatud: 19.02.2014].
21. “RoboCup Standard Platform League (NAO) Rule Book” RoboCup Technical Committee (2013) [Võrgumaterjal] http://www.tzi.de/spl/pub/Website/Downloads/Rules2013.pdf [Kasutatud: 19.02.2014].
22. “RoboCup Standard Platform League (NAO) Technical Challenges” RoboCup Technical Committee (2014) [Võrgumaterjal] http://www.informatik.uni-

bremen.de/spl/pub/Website/Downloads/Challenges2014.pdf [Kasutatud: 19.05.2014].

23. “Lua Versus Python” Lua-users wiki [Võrgumaterjal] http://lua-users.org/wiki/LuaVersusPython [Kasutatud: 29.04.2014]
24. “Executive Summary” SWIG [Võrgumaterjal] http://www.swig.org/exec.html [Kasutatud 30.03.2014].
25. “About SimSpark” SimSpark Wiki [Võrgumaterjal] http://simspark.sourceforge.net/wiki/index.php/About_SimSpark [Kasutatud: 26.04.2014].
26. “Choregraphe overview — NAO Software 1.14.5 documentation” Aldebaran-Robotics [Võrgumaterjal] https://community.aldebaran-robotics.com/doc/1-14/software/choregraphe/choregraphe_overview.html [Kasutatud 29.03.2014].
27. Barrett, S., Genter, K., He, Y., Hester, T., Khandelwal, P., Menashe, J., Stone, P. “UT Austin Villa 2012: Standard Platform League World Champions” kogumikus *RoboCup 2012: Robot Soccer World Cup XVI*, 2013, lk. 36 – 47.
28. Barrett, S., Genter, K., Hester, T., Khandelwal, P., Quinlan, M., Stone, P. “Austin Villa 2011: Sharing is Caring: Better Awareness through Information Sharing” [Võrgumaterjal] http://apps.cs.utexas.edu/tech_reports/reports/ai/AI-2067.pdf [Kasutatud: 21.04.2014].
29. Barrett, S., Genter, K., He, Y., Hester, T., Khandelwal, P., Menashe, J., Stone, P. “The 2012 UT Austin Villa Code Release” kogumikus *Proceedings of the RoboCup International Symposium 2013*, (RoboCup, Eindhoven, 2013) [Võrgumaterjal] http://www.cs.utexas.edu/~pstone/Papers/bib2html-links/LNAI13-Barrett.pdf [Kasutatud: 21.04.2014].
30. Röfer, T., Laue, T., Müller, J., Fabisch, A., Feldpausch, F., Gillmann, K., ... & Wenk, F. “B-Human Team Report and Code Release 2011” B-Human (2011) [Võrgumaterjal] http://www.b-human.de/downloads/bhuman11_coderelease.pdf [Kasutatud: 22.04.2014].
31. Valtazanos, A., Vafeias, E., Mico, A. B., Mankowitz, D., Nardelli, N., Ramamoorthy, S., Vijayakumar, S. „Team Edinferno Description Paper for RoboCup 2013 SPL“ (2013) [Võrgumaterjal] http://www.informatik.uni-bremen.de/spl/pub/Website/Teams2013/Edinferno.pdf [Kasutatud 19.02.2014].

32. Low, C.Y., Aziz, N., Aldemir, M., Dumitrescu, R., Anacker, H., Mellado, M. "Strategy planning for collaborative humanoid soccer robots based on principle solution" *Production Engineering, 7 (1)*, lk. 23-34 (2013).
33. Röfer, T., Laue, T., Müller, J., Bartsch, M., Batram, M. J., Böckmann, A., ... & Wenk, F. "B-Human Team Report and Code Release 2013" B-Human (2013) [Võrgumaterjal] http://www.b-human.de/downloads/publications/2013/CodeRelease2013.pdf [Kasutatud: 29.04.2014].
34. Menashe, J., Barrett, S., Genter, K., & Stone, P. "UT Austin Villa 2013: Advances in Vision, Kinematics, and Strategy" kogumikus *The Eighth Workshop on Humanoid Soccer Robots at Humanoids 2013*, (HUMANOIDS13, Atlanta, GA, 2013) [Võrgumaterjal] https://www.cs.utexas.edu/~pstone/Papers/bib2html-links/HUMANOIDS13-menashe.pdf [Kasutatud: 30.04.2014].
35. "Low level architecture — NAO Software 1.14.5 documentation" Aldebaran-Robotics [Võrgumaterjal] https://community.aldebaran-robotics.com/doc/1-14/naoqi/sensors/dcm/low_level_architecture.html [Kasutatud: 07.04.2014].

Summary in English

Development of humanoid robot Aldebaran NAO's behaviour logic for soccer software

Kristian Hunt

NAO humanoid robot, developed by Aldebaran Robotics, is used as a soccer player in the RoboCup competition's Standard Platform League (SPL), where every robot has the same hardware and the players differ only by their software. The main goal of the RoboCup is to propagate robotics and artificial intellect.

The purpose of this thesis was to develop soccer software behaviour for the RoboCup SPL competition in 2014 using the University of Texas' team UT Austin Villa's code release from 2012.

During the work, the code base of UT Austin Villa and other teams' solutions were analysed and a behaviour strategy for the 2014 competition was made, which is suitable to use when the connection between team members has been lost. In this game strategy the robots are divided into zones and if the ball enters a player's zone it will kick the ball towards opponent's goal. If the ball is not in the player's zone then the robot will move according to the location of the ball to a static predefined point in the field, within it's zone.

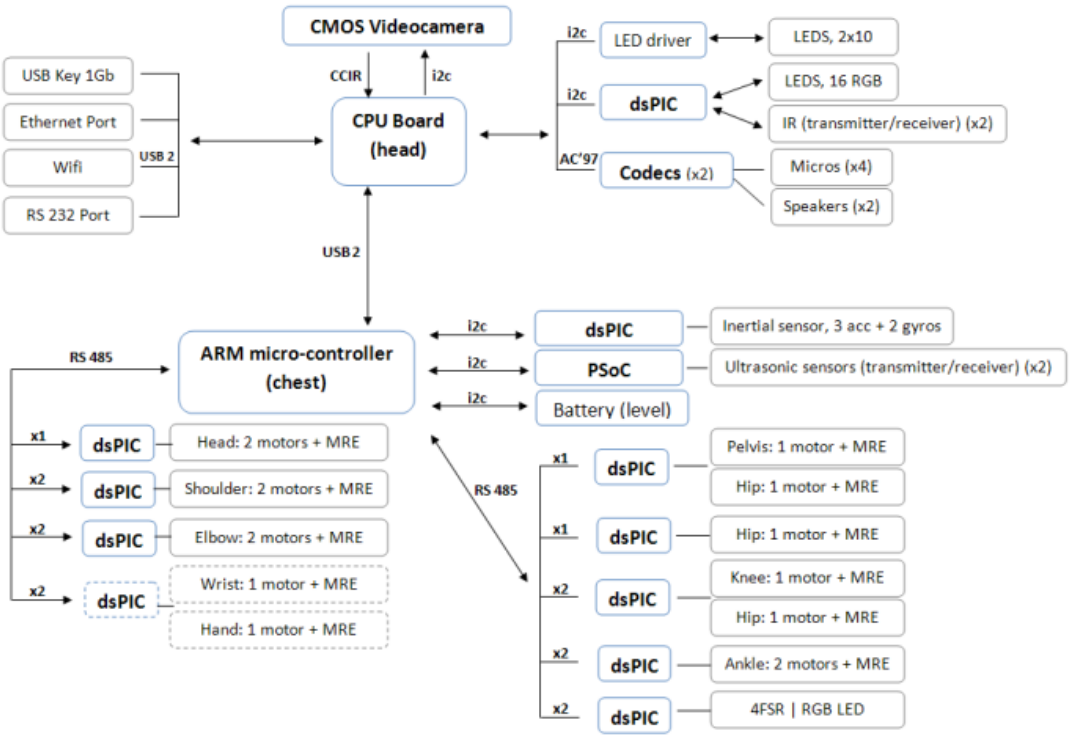
The work was made in cooperation with the team Philosopher who is participating in RoboCup in July 2014 in Brazil. According to the intentions of the team demonstrations were made in Robotex 2013, FIRST® LEGO® League Estonian and Latvian semi-final 2013 and RoboMiku Lahing 2014 to promote robotics in Estonia.

In the end of the thesis an idea of a solution of how to change the created individual strategy into a team strategy was proposed.

Lisad

Lisa 1. NAO roboti elektroonikaseadmete arhitektuur

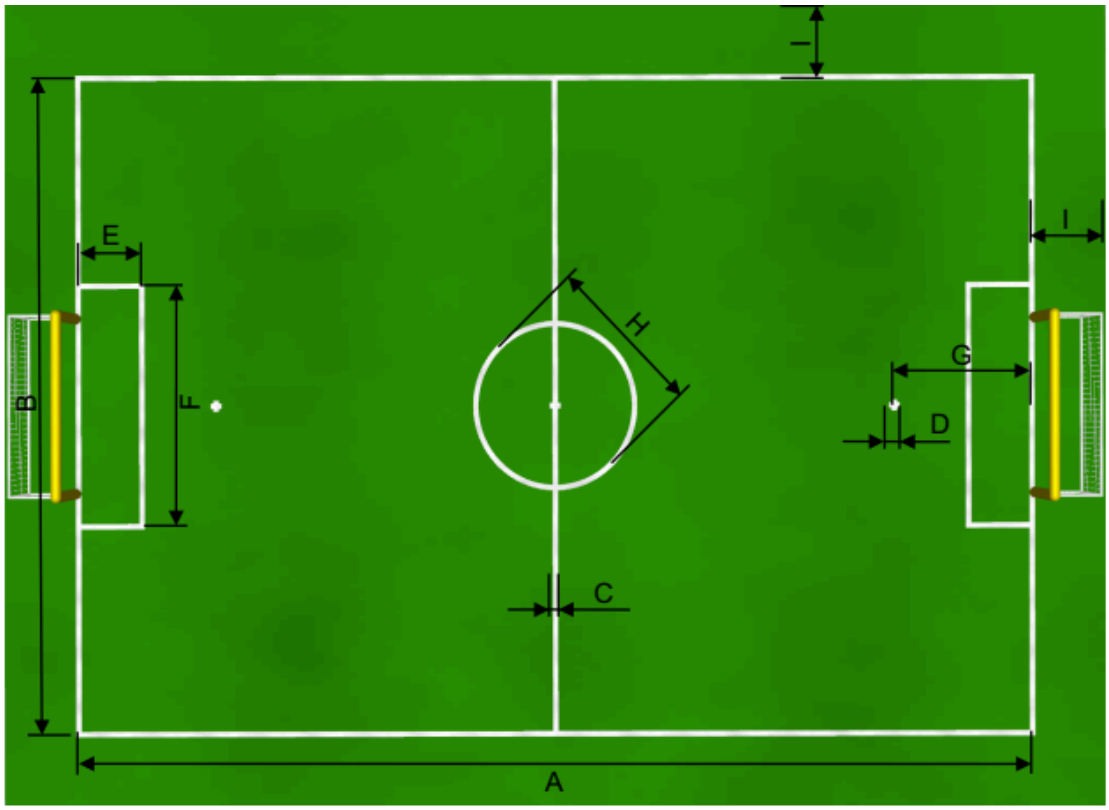
Alloleval joonisel 9 on kujutatud NAO roboti elektroonikaseadmete arhitektuur, kusjuures iga seade on defineeritud siini tüübi ja aadressi järgi antud siinil. Samuti on igal seadmel unikaalne nimi ja tüüp seadme kommunikatsioonimoodulis DCM. [35]



Joonis 9. NAO roboti elektroonikaseadmete arhitektuur. [35]

Lisa 2. RoboCup SPL võistluse mänguväljak

Alloleval joonisel 10 on kujutatud RoboCup SPL võistluse mänguväljaku koos mõõtmetega.



ID	Kirjeldus	Pikkus (mm)	ID	Kirjeldus	Pikkus (mm)
A	Mänguväljaku pikkus	9000	E	Väravavahi ala pikkus	600
B	Mänguväljaku laius	6000	F	Väravavahi ala laius	2200
C	Joone laius	50	G	Penalti märgi kaugus	1300
D	Penalti märgi suurus	100	H	Keskringi diameeter	1500
			I	Piiririba laius	700

Joonis 10. RoboCup SPL võistluse mänguväljak ja selle mõõtmed. [20]

Lisa 3. Valminud kood

Tööga kaasas oleva CD peal on failid Philosopher võistkonna repositooriumist seisuga 20.05.2014.

- /bin/ kaustas on skriptid robotite välja lülitamiseks, UTNaoTooli käivitamiseks, koodi kompileerimiseks ja robotitesse saatmiseks.
- /core/ kaust sisaldab kõikide loogikamoodulite koodi.
- /core/lua/ kaustas asuvad käitumisloogika failid ja individuaalstrateegia kutsutakse välja failist core/lua/basicSoccerBvr.lua.
- /data/ kaust sisaldab värvitabeleid, konfiguratsioonifaile ja autoload.ini faili NAOqi vahendaja jaoks.
- /docs/ kaust sisaldab Austin Villa juhendeid arvuti ja roboti ettevalmistamiseks, et nende koodibaasi kasutada, Austin Villa 2012. aasta raportit, reegleid 2014. aasta võistluse jaoks ning kiirklahvide kombinatsioone UTNaoTooli jaoks.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Kristian Hunt,
(sünnikuupäev: 08.03.1992)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose HUMANOIDROBOTI ALDEBARAN NAO JALGPALLITARKVARA KÄITUMISLOOGIKA ARENDAMINE,

mille juhendajad on Alvo Aabloo ja Heilo Altin,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 29.05.2014