

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Anja Babić, Nikola Jagodin

**HUMANOIDNI ROBOT NAO U DRUŠTVENOJ INTERAKCIJI:
ELEMENTI SUSTAVA NA PRIMJERU IGRE SKRIVAČA**

Zagreb, 2014.

Ovaj rad izrađen je u Laboratoriju za robotiku i inteligentne sustave upravljanja pod vodstvom prof.dr.sc. Zdenka Kovačića i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2013./14.

Sadržaj

1. Uvod.....	1
2. Korišteni alati.....	3
2.1. Humanoidni robot NAO.....	3
2.2. Definicije koordinatnih sustava.....	4
2.3. NAOqi radni okvir.....	5
2.4. Mobile Robot Programming Toolkit.....	6
3. Lokalni moduli.....	9
3.1. Korišteni ugrađeni moduli.....	9
3.2. Razvijeni moduli.....	14
4. Lokalizacija.....	22
4.1. Algoritam lokalizacije.....	22
4.2. Izračun globalnih koordinata.....	25
5. Pretraživanje prostora.....	28
5.1. Podjela mape.....	28
5.2. Određivanje točke cilja hodanja.....	29
5.3. Kretanje robota.....	31
6. Ostvarena funkcionalnost.....	33
6.1. Igra skrivača.....	33
6.2. NAO u socijalnoj robotici.....	36
7. Zaključak.....	39
8. Literatura.....	40

1. UVOD

Roboti nove generacije blisko će surađivati s ljudima kao društvo i pomoć kod kuće, u uslužnim djelatnostima, kao pomagači u brizi o starim i nemoćnim osobama ili osobama s invaliditetom, suradnici u terapiji i rehabilitaciji te mnogim sličnim ulogama za koje je potrebna sposobnost efikasnih, prikladnih i ugodnih interakcija s čovjekom. Za razvoj uslužnog robota koji je u stanju autonomno funkcionirati u ljudskom okruženju ključna su već visokorazvijena područja i tehnologije vezane uz percepciju, kretanje, lokalizaciju i navigaciju, no prijeko je potrebna sposobnost robota da raspozna i reproducira ljudska ponašanja. Bitan uvjet koji je potrebno ispuniti da bi robot bio društveno prihvaćen jest sposobnost da se ponaša slično ljudima [1][2]. Također je bitno osigurati da robot može samostalno dijagnosticirati probleme u vlastitom funkcioniranju, obavijestiti korisnika i u kritičnim situacijama zamoliti čovjeka za pomoć.

Cilj ovog rada jest na primjeru igre skrivača istražiti sposobnost humanoidnog robota NAO da na autonoman način sudjeluje u interakciji s ljudima. Svojom dizajnom NAO je iznimno atraktivan i prikladan za socijalnu interakciju, pogotovo s djecom. Pokazalo se da odlično zaokuplja i zadržava njihovu pažnju te da ima izniman potencijal za daljnji razvoj u tom smjeru.

U radu su izneseni detalji konkretne implementacije igre skrivača. Robot prekriva oči rukama i počinje brojati. Nakon što izbroji do deset, kreće u potragu za suigračem. Ako tijekom pretraživanja prostora uspješno uoči ljudsko lice, robot obavještava suigrača da ga je pronašao i da je sad na njemu red te čeka da ovaj sakrije oči i počne brojati. Robot provjerava da suigrač ne vara pri žmirenju (da ne „viri“) i potom bira mjesto za skrivanje u poznatom prostoru, odlazi do njega i tamo sakriven čeka da mu suigrač dodirom po glavi da do znanja da ga je pronašao.

Da bi se postigla željena funkcionalnost potrebno je ostvariti sljedeće elemente:

- Prikladne glasovne poruke
- Prikladne animacije i geste
- Prepoznavanje govora
- Prepoznavanje lica
- Prepoznavanje markera

- Lokalizacija robota na mapi
- Obrada mape i podjela prostora za pretraživanje
- Planiranje putanje hodanja

U drugom poglavlju iznose se bitne tehničke specifikacije korištenog robota i neki detalji programskog dijela implementacije, uključujući i kratki opis alata Mobile Robot Programming Toolkit. Treće poglavlje sadrži opis korištenih ugrađenih i novorazvijenih modula na robotu. U četvrtom poglavlju opisuje se algoritam lokalizacije robota u prostoru, a u petom izvedba pretraživanja prostora i planiranje putanje za hodanje robota. U šestom se poglavlju opisuje razvijena cjelina i predstavlja još nekoliko primjera korištenja robota NAO u socijalnom okruženju te se potom iznosi nekoliko zaključaka i planova za daljnji rad.

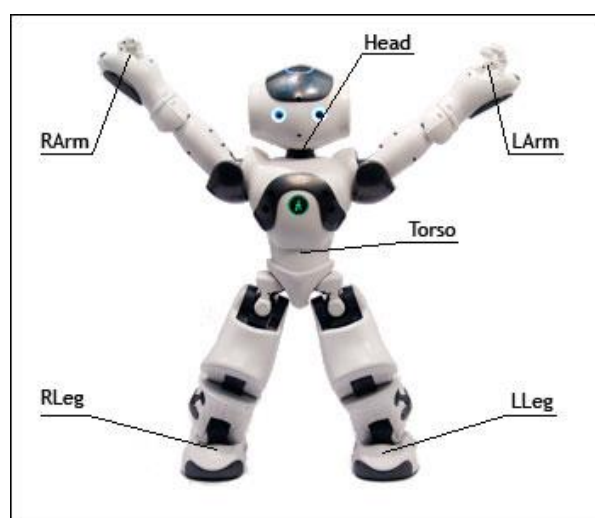
2. KORIŠTENI ALATI

2.1. Humanoidni robot NAO

NAO H25 je autonomni programabilni humanoidni robot francuske tvrtke Aldebaran Robotics [3]. Razvoj robota započet je u Parizu 2004. godine, a verzija korištena u istraživačkom laboratoriju LARICS Sveučilišta u Zagrebu jest NAO H25 ATOM iz 2012. godine.

Robot je opremljen složenim sustavom senzora koji uključuje dvije kamere, četiri mikrofona, dva ultrazvučna senzora - sonara, dva infracrvena odašiljača i prijavnika, devet senzora za dodir i osam senzora za pritisak razmještenih po tijelu. Komunikacija s računalom odvija se bežično preko Wi-Fi mreže.

Za interakciju s okolinom i korisnikom NAO ima dva zvučnika i LED indikatore. Glavni procesor koji pokreće Linux *kernel* za rad robota jest Intel ATOM CPU smješten u glavi robota, dok je drugi procesor smješten u torzu. Procesori su dovoljno snažni da omogućuju kvalitetan rad programa i modula na samom robotu, što povećava njegovu razinu autonomije i smanjuje probleme koji nastaju zbog komunikacijskog kašnjenja tijekom upravljanja robotom putem udaljenog računala.



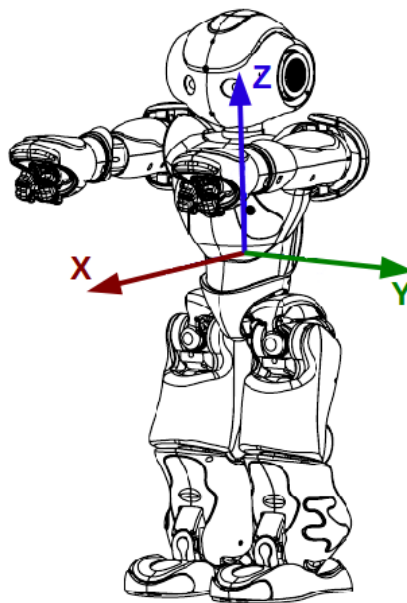
Slika 2.1. Kinematički lanci [3]

Robot ima 25 stupnjeva slobode, a efektori kojima se može upravljati prikazani su na Slici 2.1.

Ovisno o korištenju, baterija mu pruža barem 90 minuta autonomije (mjereno za najgori slučaj - neprekidno hodanje i bežično spajanje na mrežu). Robot je visok 58 cm i teži 4.3 kg.

2.2. Definicije koordinatnih sustava

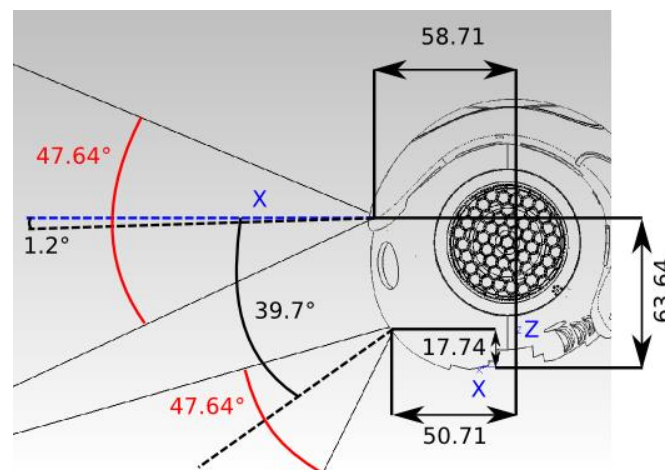
Robot NAO dolazi s već riješenom direktnom i inverznom kinematikom u tri različita prostora koji se nazivaju redom `FRAME_WORLD`, `FRAME_TORSO` i `FRAME_NAO`. `FRAME_WORLD` se definira u trenutku paljenja robota i nakon toga se ne mijenja. `FRAME_ROBOT` nalazi se uz tlo i ima ishodište u srednjoj točki između stopala robota. Ovaj se sustav okreće s robotom, no samo oko uspravne z osi, tako da njegova x os uvijek ima pozitivni smjer prema prednjoj strani robota. `FRAME_TORSO` je vezan uz određenu točku blizu sredine torza robota te se kreće i mijenja orijentaciju kako se robot miče ili naginje [3]. U ovom radu kao referentni koordinatni sustav koristi se `FRAME_ROBOT` kao najprikladniji za rad s hodom robota. Smjerovi koordinatnih osi prikazani su na Slici 2.2.



Slika 2.2. Smjerovi koordinatnih osi [3]

Preračunavanja koordinata za potrebe lokalizacije robota pomoću kamere vršit će se između dvaju sustava: koordinatnog sustava gornje kamere u kojem se dobivaju informacije o

lokacijama markera na slici i FRAME_ROBOT sustava – lokalnog koordinatnog sustava robota u kojem se zadaju ciljne točke kretanja. Prema specifikacijama robota danima u [3], koordinatni sustav gornje kamere nagnje se kako se mijenja kut glave i ima ishodište postavljeno 5.071 cm bliže prednjoj strani robota od ishodišta FRAME_ROBOT sustava. Po visini robota ishodišta su razmaknuta za 47.73 cm. Točan smještaj sustava u prostoru prikazan je na Slici 2.3.

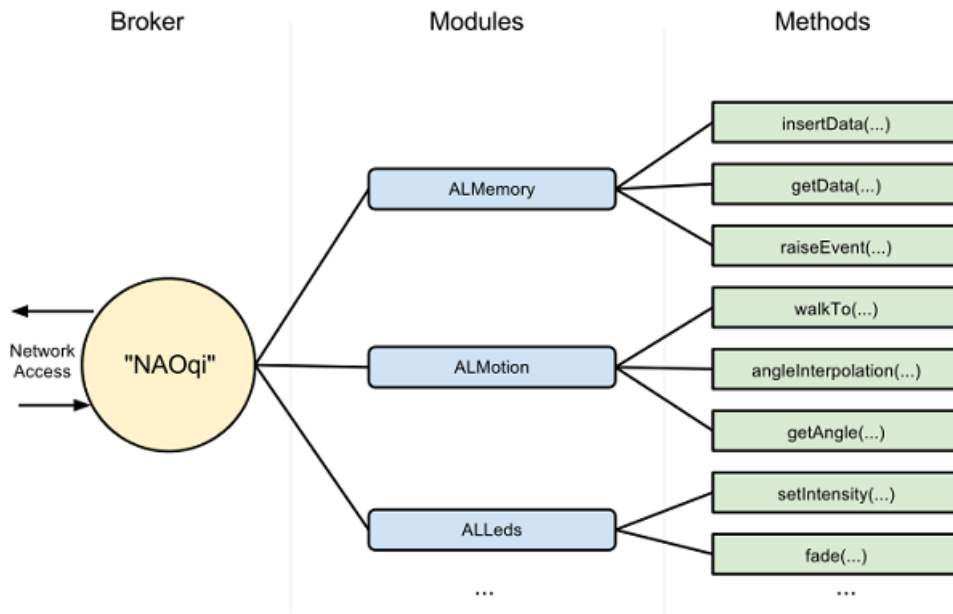


Slika 2.3. Koordinatni sustav gornje kamere [3]

2.3. NAOqi radni okvir

NAOqi radni okvir je programski okvir u kojem je isprogramiran robot. Njime je osiguran paralelizam, sinkronizacija, upravljanje resursima i upravljanje događajima (engl. *events*).

Glavni program koji se izvodi na robotu naziva se *NAOqi*. Pri paljenju robota, *NAOqi* čita postavke iz datoteke *autoload.ini* u kojoj je definiran popis biblioteka koje robot treba učitati. Svaka biblioteka sadrži jedan ili više modula koji preko *brokera* objavljuju popis svojih metoda. Taj *broker* je upravo *NAOqi*. *Broker* osigurava da svaki modul u stablu ili u mreži može dohvatiti svaku metodu koja je objavljena preko njega (Slika 2.4.).



Slika 2.4. Grafički prikaz rada *brokera* [3]

Proxy je objekt koji se ponaša kao modul kojeg predstavlja; ukoliko se stvori *proxy* prema nekom modulu, otvorit će se pristup svim metodama tog modula. *Proxy* se može stvoriti na dva načina:

1. Preko imena modula – u slučaju da se modul nalazi u istom *broker*-u. Ovo se naziva lokalni poziv.
2. Preko imena modula, te IP adrese i porta *brokera* u kojem se modul nalazi.

Moduli mogu biti lokalni ili udaljeni (engl. *remote*).

Udaljeni moduli izgrađeni su kao izvršne datoteke i ne mogu se izvoditi na robotu, već je potrebno upravljačko računalo na koje se robot spaja. Ovakvi moduli su jednostavniji za korištenje, ali su manje efikasni s obzirom na brzinu izvođenja i zauzeće memorije.

Udaljeni moduli komuniciraju preko mreže korištenjem *brokera*. Mogu komunicirati na 2 načina:

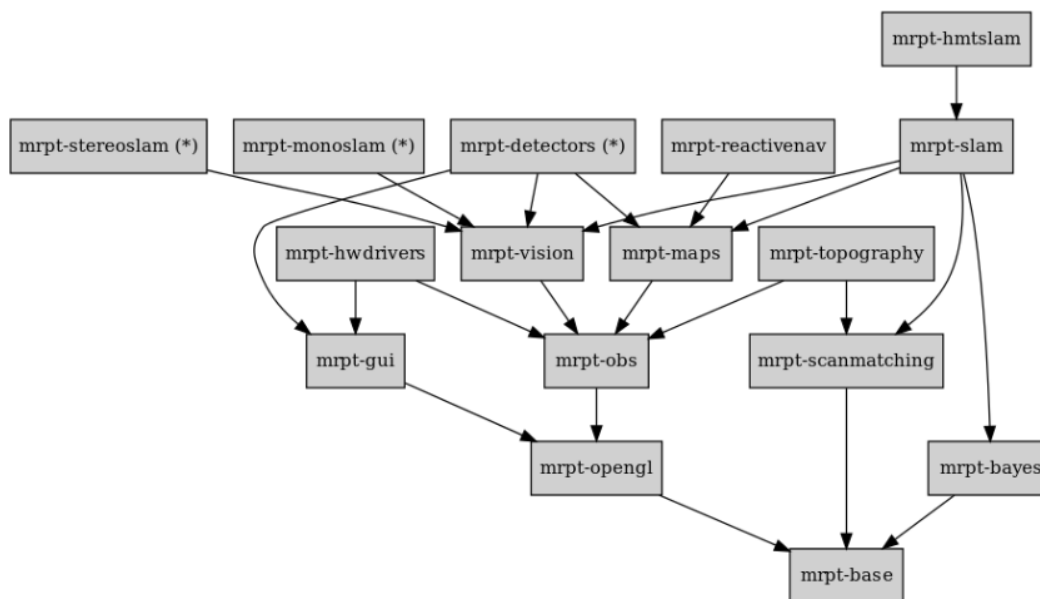
- Vezom *broker-to-broker* – gdje moduli jednog *brokera* imaju pristup svim modulima drugog *brokera*
- Vezom *proxy-to-broker* – gdje *proxy* ima pristup svim modulima *brokera*, ali ti moduli ne mogu pristupiti modulu koji je vlasnik *proxyja*

Ukoliko je modul lokalni, izgrađen je kao biblioteka i može se koristiti samo na robotu. Lokalne module čine dva ili više modula koji su pokrenuti u istom procesu. Oni komuniciraju korištenjem samo jednog *brokera*. Budući da su pokrenuti u istom procesu, oni dijele varijable i metode bez serijalizacije i korištenja mreže. Zbog toga se ovakvi moduli izvode najbrže i ključan su dio osiguravanja autonomnog funkcioniranja robota.

2.4. Mobile Robot Programming Toolkit

U nastojanju da objedini i donekle standardizira implementaciju softvera bitnog za razvoj mobilne robotike te razvije biblioteku koja bi obuhvaćala zahtjevne operacije iz linearne algebre, računski zahtjevne numeričke proračune, te pristup hardverskom dijelu rada s robotima uz optimalna vremena izvođenja, grupa MAPIR (MACHINE Perception and Intelligent Robotics) Sveučilišta u Malagi pokrenula je projekt MRPT (Mobile Robot Programming Toolkit) [4].

MRPT sadrži neke gotove aplikacije za prikupljanje, obradu i prikaz podataka tijekom rada ili simulacije rada mobilnog robota, no njegov glavni dio je skup biblioteka koje sačinjavaju okvir za programiranje s gotovim implementacijama mnogih često korištenih algoritama za izgradnju mape, SLAM (engl. *Simultaneous Localisation and Mapping* – simultana lokalizacija i mapiranje), obradu sirovih podataka dobivenih korištenjem laserskih ili ultrazvučnih senzora, specifičnih klasa razvijenih za rad s raznim vrstama mapi, te alate za izradu jednostavnih grafičkih korisničkih sučelja (Slika 2.5.).



Slika 2.5. Grafički prikaz sadržaja i međuovisnosti biblioteka unutar MRPT [5]

Alati su razvijeni u programskom jeziku C++, s time da su mnoge funkcije za koje je iznimno bitno kratko vrijeme izvođenja implementirane na asemblerskoj razini. Obrada i prikaz slika obavlja se pomoću visoko optimizirane biblioteke OpenCV, u čijem je razvoju sudjelovao i jedan od autora MRPT-a, José Luis Blanco Claraco, čime je postignut visok stupanj integracije između ovih dviju biblioteka [4].

MRPT je dostupan besplatno, pod trećom verzijom licence GND General Public License (GPL).

3. LOKALNI MODULI

3.1. Korišteni ugrađeni moduli

3.1.1. ALMemory

Ugrađeni modul *ALMemory* omogućuje spremanje i čitanje podataka iz centralizirane memorije robota. Također, ovaj modul upravlja događajima, odnosno poziva odgovarajuće metode (engl. *callback methods*) modula koji su prijavljeni na određene događaje, ako se ti događaji dogode.

3.1.2. ALBehaviorManager

Ugrađeni modul *ALBehaviorManager* koristi se za upravljanje ponašanjima (engl. *behavior*). Ponašanje je skup unaprijed definiranih zadataka koje robot izvodi (npr. ustajanje robota iz sjedećeg položaja i sl.). Modul nudi metode za dodavanje i brisanje ponašanja s robota, te za njihovo pokretanje i zaustavljanje. Neka ponašanja su ugrađena i izvode se odmah pri paljenju robota, a neka se pokreću drugim modulima ili programima (npr. pomoću alata *Choreographe*).

3.1.3. ALSensors

Ugrađeni modul *ALSensors* odgovoran je za događaje vezane uz robotove dodirne senzore na glavi i rukama, dugme na prsima, te odbojnice. Modul dohvaća podatke o sensorima iz *ALMemory*, obrađuje ih i javlja da se dogodio događaj, ukoliko je to potrebno.

3.1.4. ALTextToSpeech

Ugrađeni modul *ALTextToSpeech* koristi se za sintezu govora robota. Sadrži metode za slanje naredbi *text-to-speech* sustavu, te metode za modifikaciju robotovog glasa. Rezultat sinteze šalje se na robotove zvučnike.

Sustav za sintezu govora razvilo je poduzeće *Acapela Group* [6].

Trenutna verzija robota može proizvoditi govor samo na engleskom i francuskom jeziku, te se za potrebe ovog rada koristi engleski jezik.

3.1.5. ALSpeechRecognition

Ugrađeni modul *ALSpeechRecognition* omogućava robotu da prepozna predefinirane riječi ili rečenice na nekoliko jezika. Budući da hrvatski jezik nije podržan, u ovom radu koristi se prepoznavanje engleskog jezika.

Verzija NAO robota korištena u ovom radu koristi sustav za raspoznavanje koji je razvilo poduzeće *Nuance* [7].

Prije početka procesa raspoznavanja, potrebno je definirati listu riječi ili rečenica koje trebaju biti prepoznate. Nakon što je proces raspoznavanja pokrenut, modul počinje koristiti varijable *SpeechDetected* i *WordRecognized* u modulu *ALMemory*. Varijabla *SpeechDetected* sadrži informaciju o tome je li govor detektiran, dok varijabla *WordRecognized* sadrži listu riječi ili rečenica koje su prepoznate te pouzdanost prepoznavanja za svaku od njih.

3.1.6. ALFaceDetection

Ugrađeni modul *ALFaceDetection* omogućuje robotu detekciju lica koja se nalaze ispred njega. Dodatno, nudi i mogućnost raspoznavanja prethodno naučenih lica koja nije korištena u ovom radu.

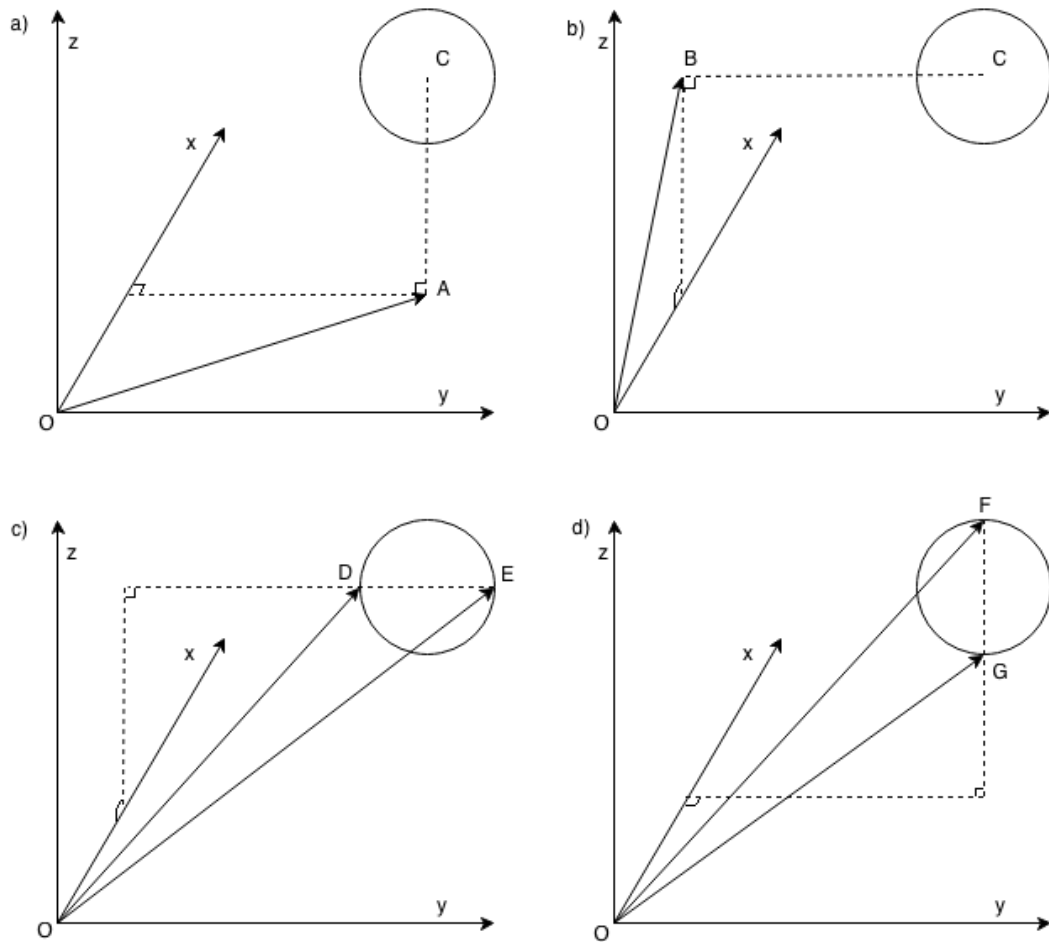
Raspoznavanje/detekcija lica temelji se na rješenju koje je razvilo poduzeće *OKI Electric Industry* [8].

Jednom kada je modul pokrenut, rezultati se spremaju u *ALMemory* u varijablu *FaceDetected*. Varijabla sadrži sljedeće informacije:

- informacije o trenutku u kojem je snimljena slika iz koje su dobiveni podaci
- informacije o licu za svako lice koje je detektirano na obrađenoj slici. Te informacije uključuju poziciju i veličinu lica mjerenu u kutovima kamere (Slika 3.1.) te dodatne informacije koje sadržavaju pouzdanost detekcije, oznaku prepoznatog lica (ako se koristi mogućnost raspoznavanja lica) i podatke poziciji i veličini očiju, nosa, ušiju i usta mjerene u kutovima kamere.
- poziciju kamere kojom je snimljena slika u koordinatnom sustavu *FRAME_TORSO*
- poziciju kamere kojom je snimljena slika u koordinatnom sustavu *FRAME_ROBOT*
- identifikacijski broj kamere kojom je snimljena obrađena slika

U ovom radu najvažnija je informacija broj detektiranih lica, a ona se koristi u modulima *LRDetekcijaOsobe* i *LRIgraSkrivaca*.

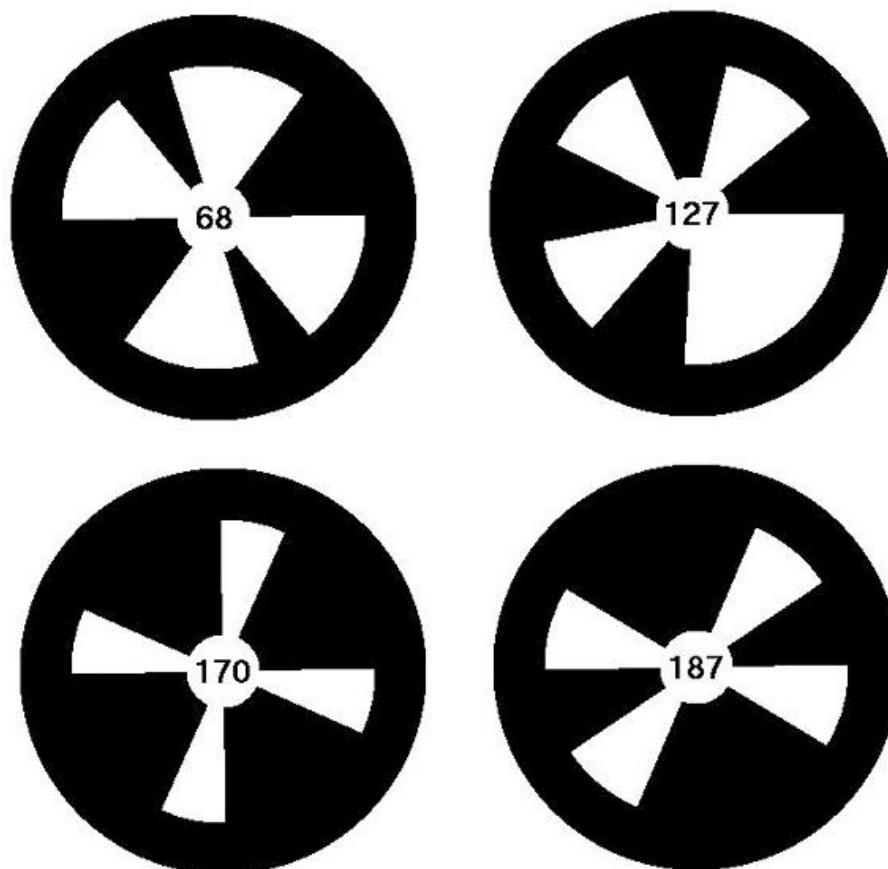
Na Slici 3.1. točka O predstavlja ishodište koordinatnog sustava kamere. Krugovi predstavljaju detektirano lice ili marker (točka C je centar). Vrijednosti su kut između osi x i vektora \overrightarrow{OA} (dio a), kut između osi x i vektora \overrightarrow{OB} (dio b), kut između vektora \overrightarrow{OD} i \overrightarrow{OE} (dio c), kut između vektora \overrightarrow{OF} i vektora \overrightarrow{OG} (dio d).



Slika 3.1. Informacije o veličinama detektiranog lica ili markera koje se nalaze u memoriji robota.

3.1.7. ALLandmarkDetection

Ugrađeni modul *ALLandmarkDetection* omogućuje robotu detekciju i raspoznavanje posebno napravljenih markera. Markeri su dizajnirani kao crni krugovi s bijelim trokutastim lepezama oko centra (Slika 3.2.), a međusobno se razlikuju po lokaciji i širini lepeza.



Slika 3.2. Primjer korištenih markera

Nakon što se proces raspoznavanja pokrene, modul će u modulu *ALMemory* osvježavati varijablu *LandmarkDetected*. Varijabla sadrži sljedeće informacije:

- informaciju o trenutku u kojem je snimljena slika iz koje su dobiveni podaci
- informacije o svakom detektiranom markeru. Te informacije uključuju lokaciju markera izraženu u kutovima kamere (Slika 3.1.), veličinu markera izraženu u kutovima kamere, orijentaciju markera s obzirom na vertikalnu os glave robota te oznaku prepoznatog markera.

3.2. Razvijeni moduli

3.2.1. LRDetekcijaOsobe

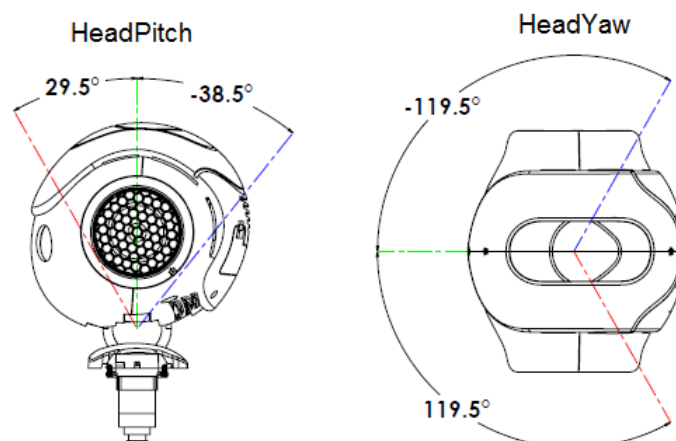
Modul *LRDetekcijaOsobe* implementira detekciju osobe (metoda *nadjiOsobu*). Metoda se poziva svaki puta kad robot dođe u poziciju u kojoj je potrebno provjeriti nalazi li se osoba u prostoru ispred robota.

Modul direktno koristi metode modula *ALMemory* i *ALMotion*, te implicitno koristi usluge modula *ALFaceDetection*. Modul *ALMotion* se koristi za upravljanje kutovima i ukočenošću zglobova *HeadYaw* i *HeadPitch* (Slika 3.3.). Metodama iz modula *ALMemory* prati se događaj *FaceDetected*. Tim događajem upravlja modul *ALFaceDetected* tijekom cijelog rada robota.

Metoda *nadjiOsobu* za detekciju osobe prima 2 parametra:

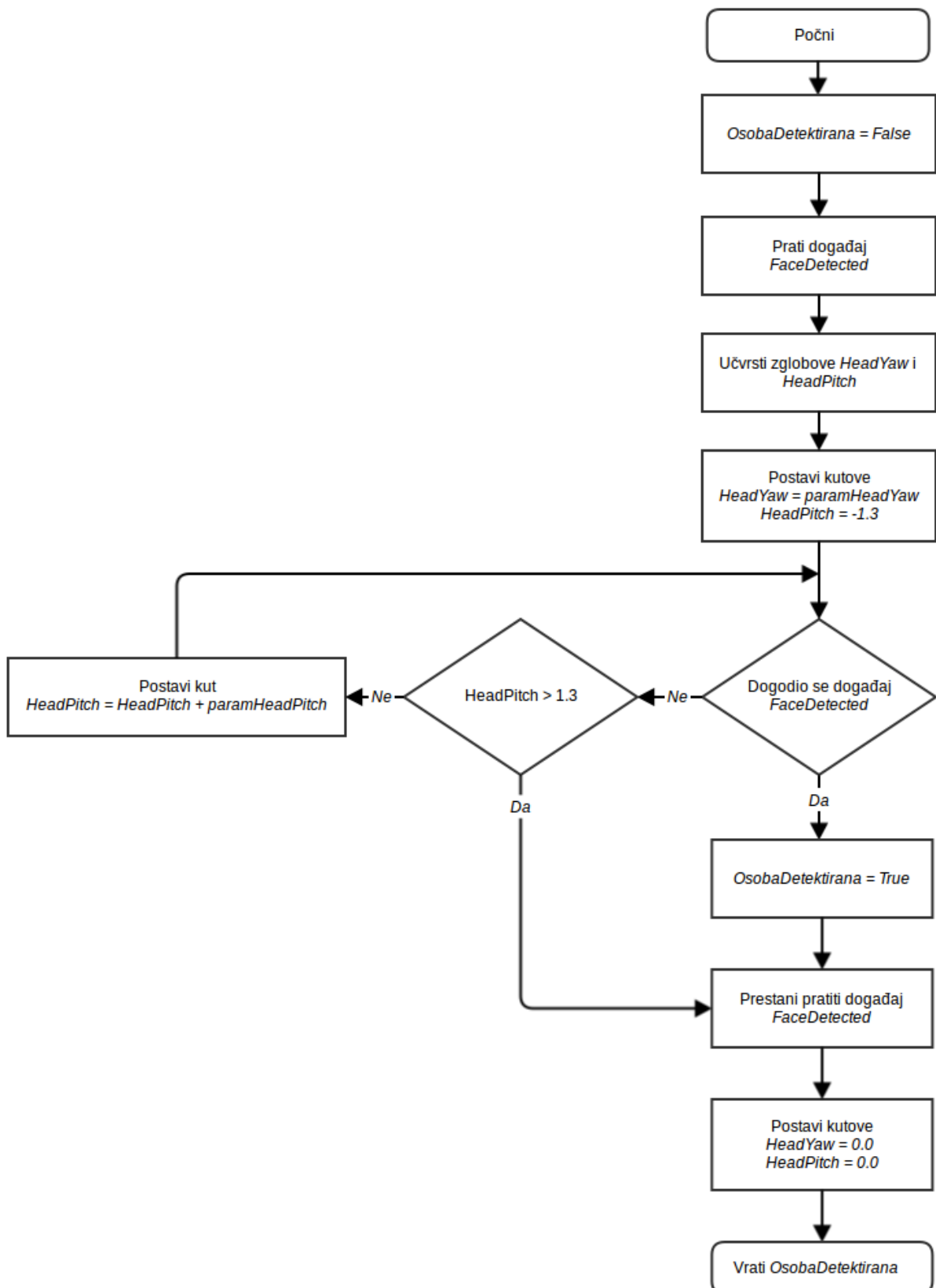
1. *paramHeadYaw* – vrijednost u koju će se postaviti kut zgloba *HeadYaw* prilikom traženja osobe
2. *paramHeadPitch* – vrijednost za koju će se uvećavati kut zgloba *HeadPitch* tijekom traženja osobe

Metoda za detekciju osobe vraća vrijednost *True* ako je osoba detektirana, te vrijednost *False* inače.



Slika 3.3. Upravljanje položajem glave robota [3]

Dijagram tîka metode za detekciju osobe prikazan je na Slici 3.4.



Slika 3.4. Dijagram tōka metode za detekciju osobe

3.2.2. LRIgraSkrivaca

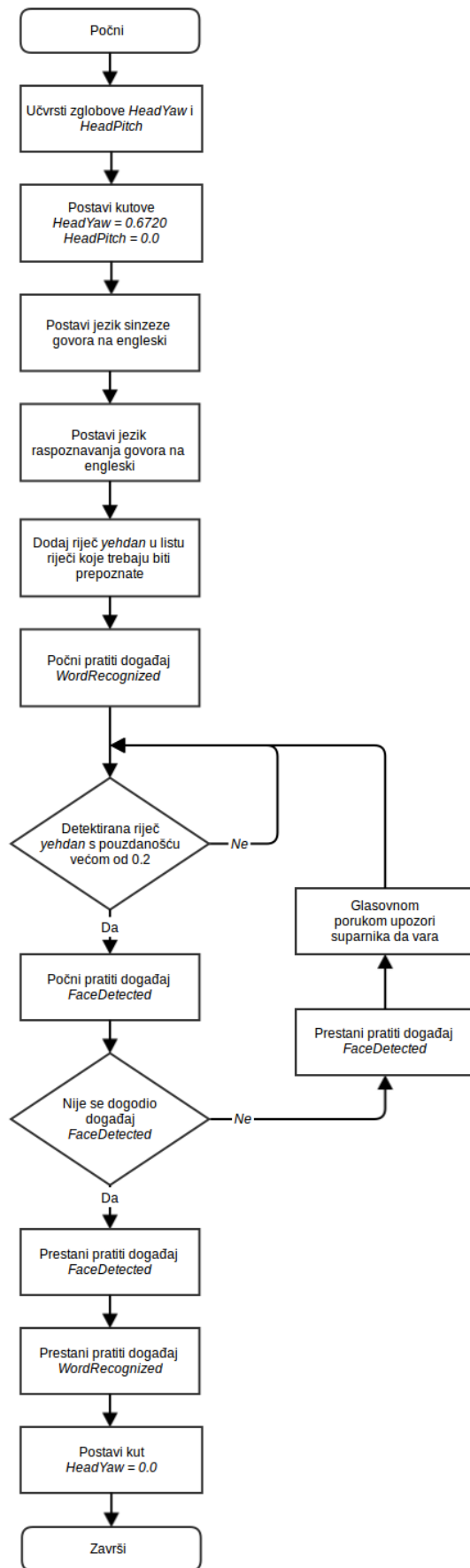
Modul *LRIgraSkrivaca* implementira neka ponašanja koja se koriste tijekom igre skrivača. Implementira ponašanje robota dok iščekuje da suigrač (čovjek) počne s brojanjem (metoda *provjeriBrojanje*), te ponašanje robota kada robot dođe u poziciju u kojoj se treba sakriti od suigrača (metoda *sakrijSe*).

Metoda *provjeriBrojanje* poziva se nakon što robot pronade suigrača i izvodi se sve dok suigrač ne počne s brojanjem. Pretpostavlja se da je suigrač počeo s brojanjem kad izgovori riječ „jedan“. Budući da robot nema implementiranu detekciju riječi na hrvatskom jeziku, obavlja se detekcija engleske riječi „yehdan“. Pokazalo se da se ovako napisana engleska riječ izgovara dovoljno slično hrvatskoj riječi „jedan“ da klasifikator vraća željene rezultate i ispravno prepoznaje izgovorenu riječ. Dodatno, robot provjerava valjanost brojanja, odnosno vara li suigrač. To radi tako da pokušava detektirati suigračovo lice u trenutku kada je detektirana riječ „yehdan“. Ako se lice vidi, pretpostavlja se da suigrač „viri“, na što ga robot upozorava i čeka na ponovni početak brojanja.

Ova metoda direktno koristi metode modula *ALMotion*, *ALTextToSpeech*, *ALMemory* i *ALSpeechRecognition*, te dodatno koristi usluge modula *ALFaceDetection*. Modul *ALMotion* se koristi za upravljanje kutovima i ukočenošću zglobova *HeadYaw* i *HeadPitch* prilikom detekcije lica suigrača. Modul *ALTextToSpeech* koristi se za upozoravanje korisnika ukoliko je njegovo lice detektirano tijekom početka odbrojavanja, odnosno za sintezu govora. Modul *ALMemory* koristi se za praćenje događaja *FaceDetected* kojim upravlja modul *ALFaceRecognition* tijekom cijelog rada robota. Modul *ALMemory* se također koristi i za praćenje događaja *WordRecognized* kojim upravlja modul *ALSpeechDetected*.

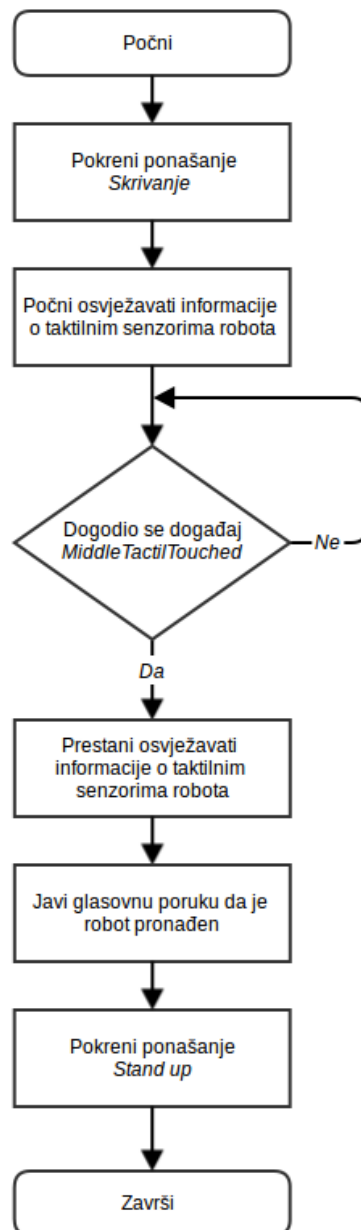
Metoda ne prima nikakve parametre, te ne vraća nikakvu povratnu vrijednost. Završava u trenutku kada je detektirano valjano odbrojavanje, odnosno kada je detektirana riječ „yehdan“, a nije detektirano lice ispred robota.

Dijagram tîka metode prikazan je na Slici 3.5.



Slika 3.5. Dijagram tōka metode *provjeriBrojanje*

Metoda *sakrijSe* implementira ponašanje robota od trenutka kada robot dođe na koordinate mjesta na kojem se planira sakriti, do trenutka u kojem suigrač pritisne srednji taktilni senzor na glavi robota. Po pozivu metode, robot zauzme položaj skrivanja, i bude u tom trenutku sve dok ga suigrač ne pronađe (pritisne srednji taktilni senzor). Kada se to dogodi, robot javi govornu poruku da je pronađen, te izlazi iz položaja skrivanja, što ujedno označava kraj metode. Položaj skrivanja je unaprijed definiran slijed pomaka robota, odnosno ponašanje *skrivanje* koje je stvoreno i instalirano na robota alatom *Choregraphe*.



Slika 3.6. Dijagram tîka metode *sakrijSe*

Ova metoda koristi metode modula *ALBehaviorManager*, *ALMemory* i *ALSensors*. Modul *ALBehaviorManager* se koristi za pokretanje instaliranog ponašanja *sakrivanje*, te ponašanja *Stand up* (ponašanje kojim robot zauzima uspravni položaj). Modul *ALMemory* koristi se za čitanje varijable *MiddleTactilTouched*, koja sadrži informaciju o tome je li pritisnut srednji taktilni senzor glave robota. Ovom varijablom upravlja modul *ALSensors*.

Ova metoda također nema nikakvih parametara, te ne vraća nikakvu povratnu vrijednost.

Dijagram tîka ove metode prikazan je na Slici 3.6.

3.2.3. LRDetekcijaMarkera

Modul *LRDetekcijaMarkera* koristi se za detekciju markera u okolini robota. Modul sadrži 3 metode:

1. *subscribeToLandmarkDetection* – podešavanje početnih parametara i početak detektiranja markera
2. *unsubscribeToLandmarkDetection* – kraj detektiranja markera
3. *getMarkerPositions* – dohvaćanje i obrada podataka o trenutno detektiranim markerima

Modul koristi metode ugrađenih modula *ALMemory*, *ALMotion* i *ALLandmarkDetection*.

Metoda *subscribeToLandmarkDetection* mora se pozvati prije korištenja metode *getMarkerPositions*. Metoda prima 5 parametara:

1. *id* – identitet pod kojim će se voditi pretplata na događaj *LandmarkDetected*
2. *period* – period u milisekundama kojim će se osvježavati varijabla *LandmarkDetected* u memoriji robota
3. *precision* – preciznost kojom će se varijabla osvježavati
4. *diameter* – unaprijed poznati promjer markera izražen u metrima
5. *sizeYToIgnore* – promjer markera u kutovima kamere ispod koje granice će se marker ignorirati

Parametri *id*, *period* i *precision* prosljeđuju se metodi *subscribe* koja pripada modulu *ALLandmarkDetection*.

Parametar *diameter* je potreban za računanje točne lokacije markera, budući da *ALMarkerDetection* radi samo s vrijednostima izraženim u kutovima kamere.

Parametar *sizeYToIgnore* se koristi zato što se pokazalo da je izračunata lokacija markera relativno netočna ukoliko je veličina detektiranog markera mala. Ovakvi markeri bi narušili konačnu točnost lokalizacije, te se zbog toga ignoriraju.

Parametri *diameter* i *sizeYToIgnore* koriste se u metodi *getMarkerPositions*.

Metoda vraća vrijednost 1 ukoliko se uspješno izvede, 0 inače.

Metoda *unsubscribeToLandmarkDetection* poziva se na kraju rada, kad više nije potrebno detektirati markere. Metoda prima samo jedan parametar: *id*, koji se prosljeđuje metodi *unsubscribe* modula *ALLandmarkDetection*. Ovaj parametar trebao bi imati istu vrijednost kao parametar *id* s kojim je pozvana metoda *subscribeToLandmarkDetection*.

Metoda vraća vrijednost 1 ukoliko se uspješno izvede, 0 inače.

Metoda *getMarkerPositions* koristi se svaki put kad je potrebno odrediti lokaciju markera na trenutnoj slici kamere. Metoda ne prima nikakve parametre, a vraća informacije o svim detektiranim markerima koji su veći od prethodno definirane vrijednosti *sizeYToIgnore*, odnosno svaki detektirani marker za koji vrijedi:

$$marker.sizeY > sizeYToIgnore$$

gdje *marker.sizeY* predstavlja veličinu markera u kutovima kamere koja je spremljena u memoriji robota (u varijabli *LandmarkDetected*).

Povratna vrijednost je lista koja sadrži:

- broj detektiranih markera koji zadovoljavaju jednadžbu
- identifikacijsku oznaku svakog detektiranog markera koji zadovoljava jednadžbu
- koordinate *x* i *y* u koordinatnom sustavu robota *FRAME_ROBOT* za svaki od detektiranih markera koji zadovoljavaju jednadžbu. Koordinata *z* se ne vraća budući da zbog 2D prirode problema nije potrebna.

Rad metode može se podijeliti na dva dijela: prvo se metodom modula *ALMemory* dohvate potrebni podaci o detektiranim markerima (iz varijable *LandmarkDetected*), a zatim se korištenjem metoda modula *ALMotion*, te funkcija biblioteke *ALMath* izračuna točna lokacija markera u koordinatnom sustavu robota, na način da se vektor koordinati markera u sustavu kamere množi s transformacijskom matricom koja povezuje sustav kamere sa sustavom robota. Ova se transformacijska matrica dobiva pomoću funkcije *getTransform* *ALMotion* modula.

4. LOKALIZACIJA

4.1. Algoritam lokalizacije

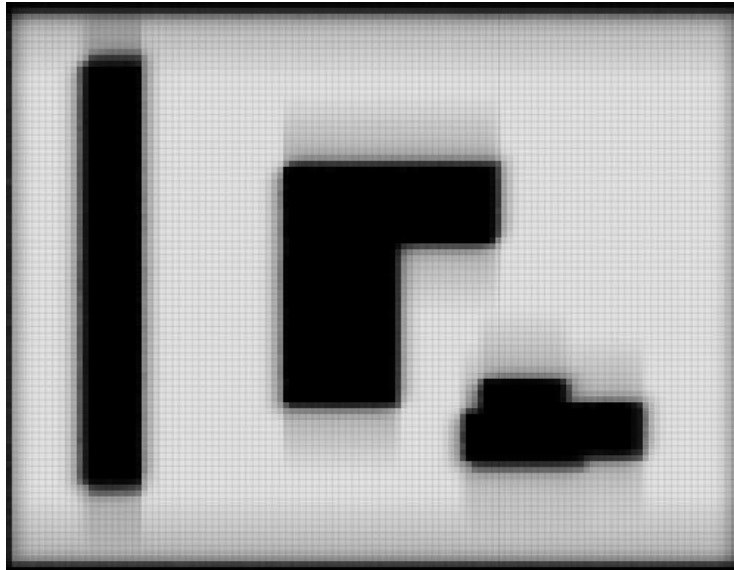
Lokalizacija mobilnog robota ključan je uvjet za postizanje ikakvog stupnja autonomije. U tipičnom slučaju robota u zatvorenom prostoru na ravnoj podlozi, lokalizacija se definira kao estimacija poze robota, tj. njegovih x i y koordinata te orijentacije dane kutom θ [9].

U ovom radu koristi se globalna lokalizacija pomoću jedinstvenih markera raspoređenih u okolini robota, koja omogućava određivanje poze robota bez poznavanja njegovog početnog položaja ili položaja u prethodnom koraku, te osigurava da autonomija robota nije ugrožena ni u tzv. *kidnapping* scenariju (od engl. *kidnapping* - otmica) u kojem mobilni robot tijekom rada biva naglo premješten na proizvoljnu lokaciju, što stvara velike poteškoće i unosi nesigurnost u većinu algoritama lokalizacije [10].

Glavna pretpostavka ostvarenog postupka lokalizacije robota je da postoji poznata mapa prostora u kojemu se odvija igra i sve planirano kretanje. U radu se za ovo koristi tzv. vjerojatnosna mrežasta karta zauzetosti, koju su razvili Elfes i Moravec sredinom 90-tih godina. Ona je metrička vrsta mape, što znači da predstavlja geometrijska svojstva prostora i to na način da prostor dijeli u mrežu polja jednake veličine te svakome od njih dodjeljuje brojčanu vrijednost koja predstavlja vjerojatnost da je taj dio prostora zauzet, tj. da predstavlja prepreku kretanju robota [11].

Primjer mrežaste karte zauzetosti dan je na Slici 4.1.

Radi efikasnije implementacije u računalu i izbjegavanja numeričkih pogrešaka nastalih zbog računanja s veoma bliskim vrijednostima vjerojatnosti u rasponu od 0 do 1, podaci se u matricu koja predstavlja mrežastu kartu zauzetosti upisuju korištenjem prirodnog logaritma tzv. *odds* funkcije, odnosno funkcije šansi – vjerojatnosti da je polje u pitanju zauzeto.

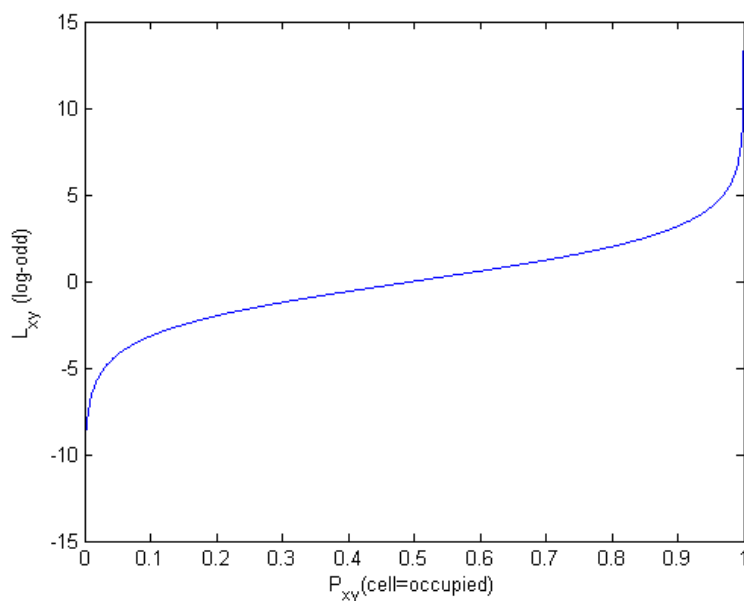


Slika 4.1. Vjerojatnosna mrežasta karta zauzetosti

U svakom trenutku iz poznatog logaritma *odds* funkcije l_k zapisanog u matrici mape moguće je dohvatiti podatak o vjerojatnosti zauzetosti polja predstavljenoj slučajnom varijablom S , koje se nalazi u retku i i stupcu j , uz jednostavnu operaciju:

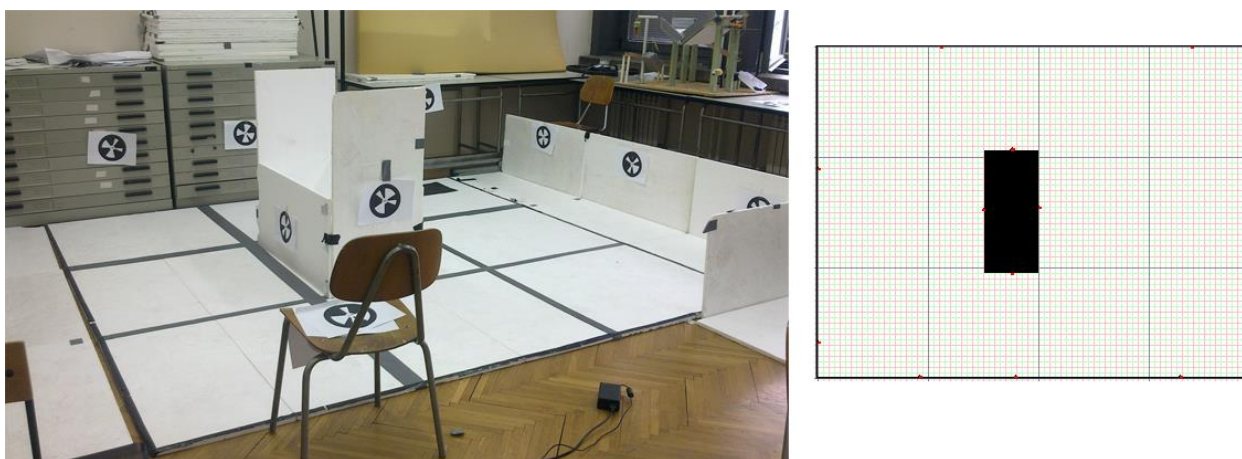
$$P(S_{ij} = \text{Zauzeto}) = \frac{1}{1 + e^{-l_k}}$$

Prikaz logaritma *odds* funkcije dan je na Slici 4.2.



Slika 4.2. Logaritam *odds* funkcije [5]

Mape korištene u ovom radu izrađene su tako da jedan piksel na slici predstavlja jedan centimetar u stvarnosti, a njegova svjetlina mjerilo je vjerojatnosti zauzetosti prostora, pri čemu potpuno crno polje znači sigurnu prepreku, a potpuno bijelo polje slobodan prostor. Primjer mape i odgovarajućeg prostora izgrađenog za kretanje robota koji ona predstavlja dan je na Slici 4.3.



Slika 4.3. Prostor za kretanje robota i pripadna mrežasta mapa zauzetosti s naznačenim položajima markera

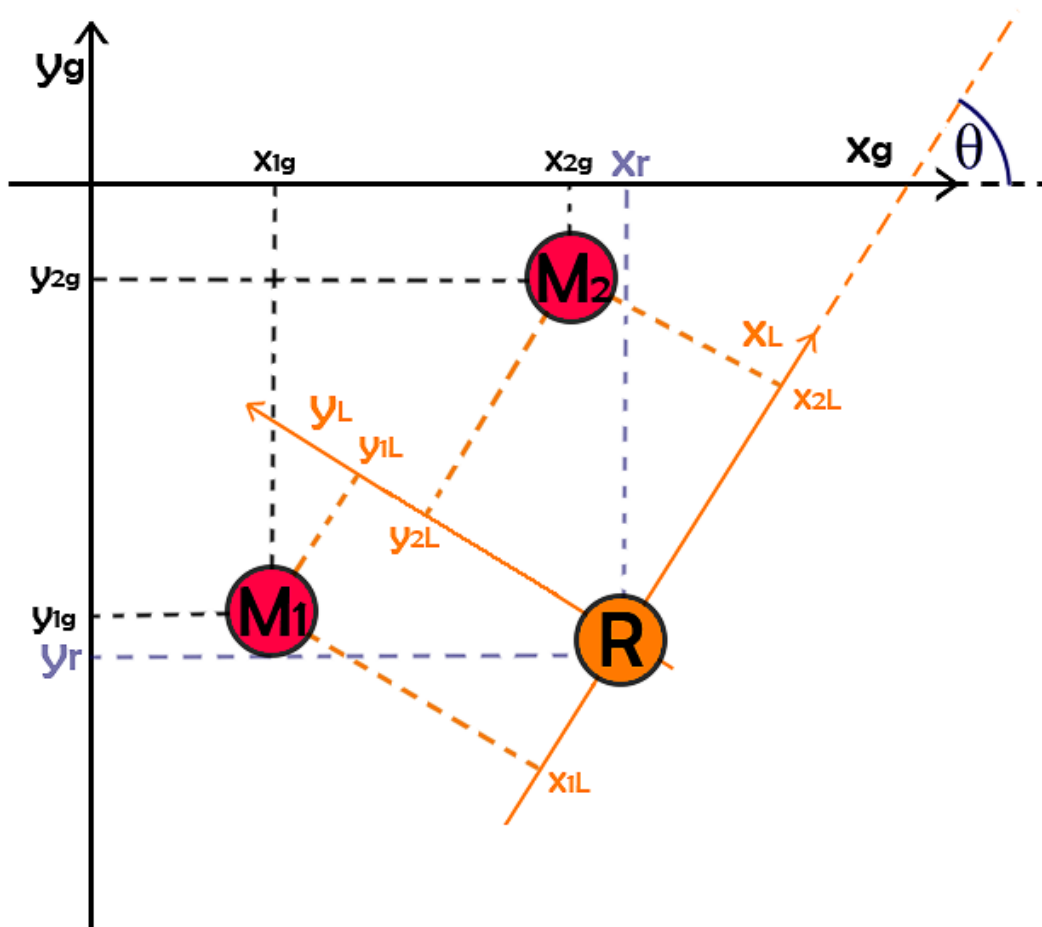
Prateći konvencije označavanja iz područja obrade slike, ishodište globalnog koordinatnog sustava nalazi se u gornjem lijevom uglu slike, te njegova y os pokazuje prema dolje.

Robotu je poznata lista markera koji se nalaze u prostoru te njihove koordinate u globalnom koordinatnom sustavu. Robot okreće glavu dok ne uoči ukupno dva markera koji se nalaze na listi poznatih markera te koristeći ranije opisani lokalni modul *LRDetekcijaMarkera* dolazi do njihovih koordinata u lokalnom koordinatnom sustavu.

4.2. Izračun globalnih koordinata

Nakon što je robot uspješno uočio dva poznata markera i odredio njihove koordinate u svojem lokalnom koordinatnom sustavu, moguće je odrediti koordinate i orijentaciju robota u globalnom koordinatnom sustavu. Promatrani slučaj je dvodimenzionalan, što pojednostavljuje dijelove proračuna.

Koordinatni sustavi prikazani su na Slici 4.4., pri čemu su (x_{1g}, y_{1g}) i (x_{2g}, y_{2g}) koordinate markera u globalnom koordinatnom sustavu, (x_{1l}, y_{1l}) i (x_{2l}, y_{2l}) koordinate markera u lokalnom koordinatnom sustavu robota, te (x_r, y_r) koordinate robota (ishodišta lokalnog koordinatnog sustava robota) u globalnom koordinatnom sustavu. Kut θ je kut između dvije x osi koordinatnih sustava i predstavlja orijentaciju robota.



Slika 4.4. Položaj robota i dvaju markera u lokalnom i globalnom koordinatnom sustavu

Između koordinatnog sustava robota i globalnog koordinatnog sustava jedine moguće transformacije su rotacija i translacija, što ih svrstava u rigidne ili neelastične geometrijske transformacije. Ove se transformacije nazivaju i euklidskima, te za njih vrijedi da čuvaju udaljenosti između bilo kojih dviju točaka [12].

Veoma vrijedno svojstvo rigidnih transformacija jest to da je za jednoznačno određivanje transformacije iz jednog koordinatnog sustava u drugi potrebno odrediti samo dva para korespondentnih točaka. Konkretno, u ovom slučaju radi se o dva pronađena markera, s poznatim koordinatama u lokalnom i globalnom sustavu.

Matrica homogene transformacije za translaciju i rotaciju koju je potrebno odrediti glasi:

$$H = \begin{bmatrix} \cos \theta & \sin \theta & t_x \\ -\sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Transformiranje koordinata između dvaju koordinatnih sustava je prema tome:

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & t_x \\ -\sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_g \\ y_g \\ 1 \end{bmatrix}$$

Uz poznate koordinate dvaju markera u globalnom i lokalnom koordinatnom sustavu, moguće je iz ovoga napisati sustav četiriju jednadžbi s četiri nepoznanice:

$$\begin{bmatrix} x_{1l} \\ y_{1l} \\ x_{2l} \\ y_{2l} \end{bmatrix} = \begin{bmatrix} x_{1g} & y_{1g} & 1 & 0 \\ y_{1g} & -x_{1g} & 0 & 1 \\ x_{2g} & y_{2g} & 1 & 0 \\ y_{2g} & -x_{2g} & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \\ t_x \\ t_y \end{bmatrix}$$

Rješavanjem ovog sustava (matričnim množenjem s lijeve strane inverzom matrice) dolazi se do vrijednosti $\cos \theta$, $\sin \theta$, t_x i t_y , te je time određena veza između dvaju koordinatnih sustava – tražena matrica H .

Želi li se odrediti položaj robota u globalnom koordinatnom sustavu, potrebno je pomoću inverza izračunati matrice transformirati lokalne koordinate (0,0).

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Za brzo računanje inverza matrice koristi se dekompozicija singularnih vrijednosti (SVD) koja vraća rezultat (pseudoinverz) čak i kad je matrica singularna ili blizu singularnoj [13].

Orijentacija robota θ dobiva se iz izraza sa sada poznatim vrijednostima

$$\tan \theta = \frac{\cos \theta}{\sin \theta}$$

uz korištenje funkcije *atan2* koja vodi računa o predznacima komponenti i prema tome kvadrantu u kojem se kut nalazi.

Na kraju, radi ranije spomenute definicije globalnog koordinatnog sustava prema slici, dobivenim globalnim y koordinatama mijenja se predznak, odnosno vrši se zrcaljenje preko x osi.

5. PRETRAŽIVANJE PROSTORA

5.1. Podjela mape

Kako bi robot mogao tražiti suigrača u igri skrivača, potrebno je definirati algoritam pretraživanja prostora koji osigurava da će čitavo područje igranja biti posjećeno i provjereno barem jednom po rundi igre.

Kao prvi korak, mapa prostora se dijeli na nekoliko područja takve veličine da robot može čitavo područje pregledati u jednom okretu. Svako polje određeno je centralnom točkom do koje robot hoda te iz koje promatra prostor oko sebe, tražeći čovjeka. Veličina područja ne smije biti premala jer bi tada za pokrivanje čitave mape bio potreban velik broj područja koje bi robot redom trebao posjetiti i pretražiti, čime bi potraga postala spora i neefikasna. U konačnici u ovom radu veličina područja postavljena je tako da odgovara radijusu pretraživanja od 80 centimetara oko robota, jer je utvrđeno da detekcija ljudskog lica dobro funkcionira na toj udaljenosti i da se tako dobiva podjela mape s prihvatljivim brojem točaka za posjećivanje.

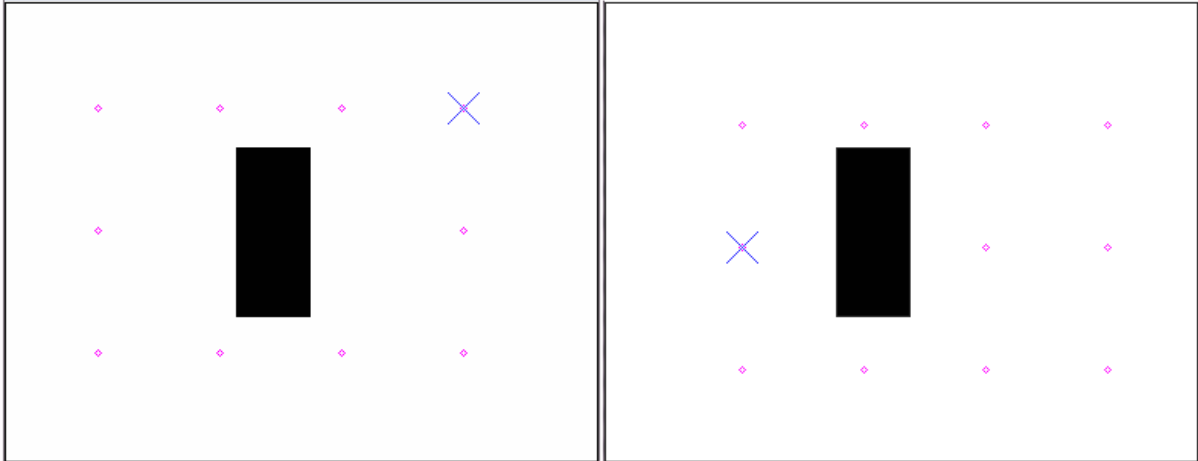
Podjela mape obavlja se tako da se trenutna pozicija robota uzima za početnu točku iz koje se pokreće pretraživanje polja mape u svim smjerovima, u intervalima definiranim pomoću ranije spomenutog radijusa pretraživanja.

Ako je polje slobodno, odnosno ako vrijedi:

$$P(S_{ij} = \text{Zauzeto}) < 0.5$$

provjerava se njegova neposredna okolina. Postavlja se zahtjev da su sva polja unutar sigurnosnog radijusa od 25 cm od razmatranog polja slobodna, odnosno da je polje dovoljno udaljeno od svake prepreke da robot sa sigurnošću može pristupiti toj poziciji.

Ako je i ovaj uvjet zadovoljen, polje se sprema na listu točaka koje će robot tijekom pretraživanja posjetiti. Primjer podjele mape za dva različita početna položaja robota dan je na Slici 5.1.



Slika 5.1. Primjeri podjele mape za dva različita početna položaja robota

5.2. Određivanje točke cilja hodanja

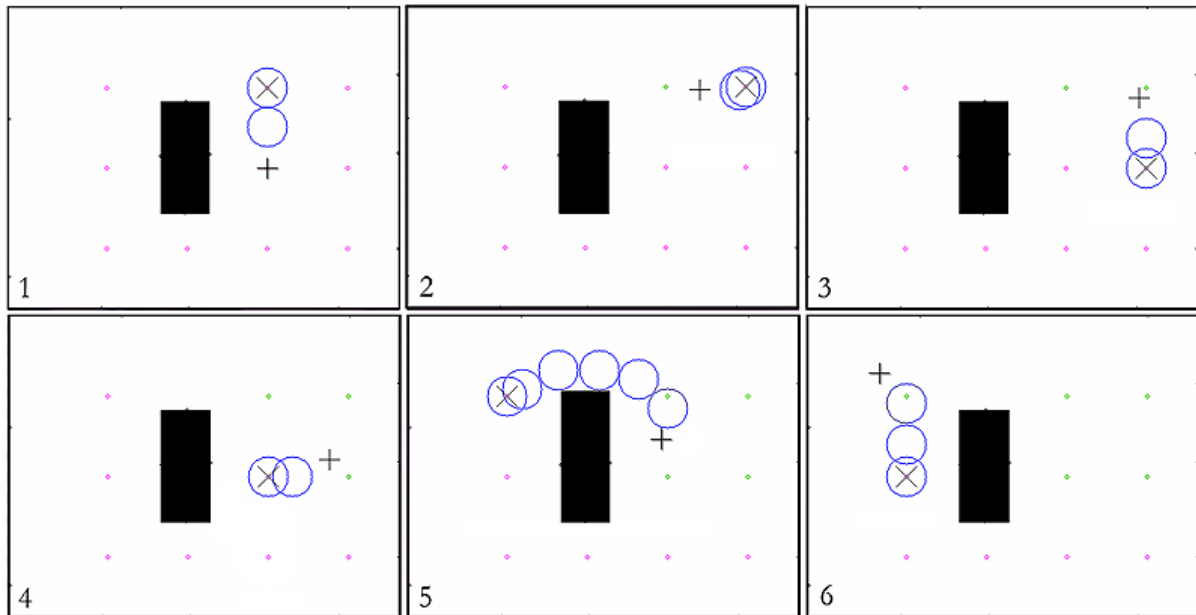
Nakon što je određen niz pozicija do kojih robot treba doći tijekom pretraživanja, potrebno je redom birati pozicije iz niza te planirati putanju kretanja robota do njih.

Za izbor iduće točke koju će robot posjetiti koristi se tzv. *greedy* algoritam pretraživanja (engl. *greedy* - pohlepan). *Greedy* algoritam oslanja se na ideju da je moguće postići u cijelosti dobro ili čak odlično ponašanje kratkovidnim biranjem trenutno najbolje i najisplativije opcije od ponuđenih u svakom pojedinom koraku, bez razmatranja utjecaja tog izbora na optimalnost cjelokupnog rješenja [14].

Robot u svakom novom koraku za određivanje bira onu točku iz niza koja je po euklidskoj udaljenosti najbliža trenutnoj poziciji robota. Po dolasku u novu poziciju, odgovarajuće područje se smatra pretraženim i točka se briše iz niza. Ovo se obavlja dokle god suigrač nije pronađen i dok niz točaka nije ispražnjen.

Korištenjem *greedy* algoritma postiže se velika brzina izbora nove točke odredišta, kao i to da će robot u svakom „koraku“ pretraživanja prelaziti relativno male udaljenosti, što je pogodno

za smanjenje mogućih navigacijskih grešaka koje se javljaju tijekom hodanja zbog proklizavanja stopala robota. Dulji intervali hodanja desit će se u situacijama kad robot mora obilaziti prepreku jer je već pretražio jednu njezinu „stranu“ ili ako se tijekom pretraživanja inzistiranjem na lokalnom minimumu „zaobide“ jedna od točaka, pa ona ostane za kraj, a robot se nalazi na poziciji relativno daleko od nje i mora se do nje vraćati.



Slika 5.2. Šest koraka postupka pretraživanja prostora i kretanja robota

Planiranje putanje obavlja se korištenjem ugrađenih metoda rada s mrežastim kartama zauzetosti alata MRPT i odvija se u dva koraka:

- prepreke predstavljene zauzetim poljima mape uvećaju se tako da veličina jednog polja odgovara sigurnosnom radijusu robota za koji se planira putanja, što osigurava da jedno slobodno polje na mapi znači da neće doći do kolizije robota s preprekama u okolini - ovom radu ta je vrijednost postavljena na 25 cm
- algoritmom iteracije vrijednosti temeljenom na Markovljevom procesu odlučivanja traži se najkraći put do zadanog odredišta, s time da je najveći dopušteni razmak između dviju točaka krajnjeg puta zadan i u ovom radu postavljen na 20 cm [5]

Niz od nekoliko koraka cjelokupnog opisanog postupka pretraživanja dan je na Slici 5.2. Križ označava trenutnu poziciju robota, a X označava odabranu destinaciju.

5.3. Kretanje robota

Za upravljanje pokretima i hodom robota koristi se ugrađeni *ALMotion* modul koji pokriva četiri glavna područja upravljanja:

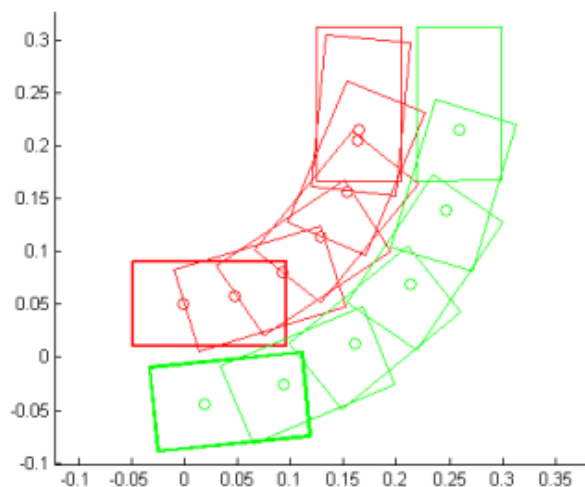
- Upravljanje momentom pojedinih motora (engl. *stiffness*)
- Upravljanje pozicijama zglobova (generiranje i zadavanje putanje, interpolacije između zadanih položaja – vrijednosti kuteva)
- Upravljanje hodanjem (upravljanje veličinom i frekvencijom koraka)
- Upravljanje efektorima u kartezijskom prostoru (ugrađeni rješavač inverzne kinematike, upravljanje čitavim tijelom, engl. *Whole Body Control*) [3]

Dinamički model hoda robota odgovara jednostavnom lineariziranom modelu obrnutog njihala predloženom u [15]. Svaki korak hoda razbijen je u dvije faze: fazu u kojoj su obje noge oslonac (engl. *double leg support phase*) i fazu u kojoj samo jedna od nogu pruža oslonac (engl. *single leg support phase*), pri čemu je robot oslonjen na obje noge barem trećinu vremena trajanja jednog koraka. Svako hodanje pozvano naredbama *ALMotion* modula počinje i završava fazom u kojoj obje noge pružaju oslonac u trajanju od 0.6 sekundi. Putanja nogu između koraka, odnosno zamah nogu, određuje se pomoću SE3 interpolacije koja slično kao interpolacija splineovima uzimanjem u obzir početnih i krajnjih brzina na segmentima trajektorije osigurava glatke prijelaze između pokreta, bez trzaja.

Stabilnost hoda robota osigurava se zatvaranjem povratne veze pomoću senzora u zglobovima robota. Ovo upravljanje izvedeno je tako da se smanje oscilacije torza koje su rezultat kretanja robota i tako da je hod robustan za relativno male vanjske poremećaje. NAO tako može prelaziti preko različitih površina dok hoda, npr. može prekoračiti rub tepiha, parketa ili pločica, što otvara mnoge mogućnosti korištenja u svakodnevnim prostorima koji nisu posebno prilagođeni za funkcioniranje robota.

Razlike u visini tla od par centimetara počinju predstavljati problem te mogu uzrokovati pad robota ako ih se ne detektira. Ovdje su korisni tzv. ugrađeni refleksi koji su upaljeni tijekom obavljanja svih pokreta robota i koji predstavljaju skup ponašanja koji se automatski aktiviraju u određenim slučajevima i po potrebi utječu na ponašanje samog robota. Bitno je spomenuti *Fall Manager* refleks koji poduzima akcije kad detektira da robot pada s ciljem minimiziranja eventualne štete. Nakon što pristignu očitavanja iz unutarnjeg akcelerometra koja su izvan sigurnosnih granica, robot podiže ruke kako bi zaštitio glavu, gasi motore i time opušta udove kako bi se što bolje amortizirao pad bez lomljenja dijelova [3].

Robot se prije početka svakog hodanja postavlja u neutralni stabilni inicijalni položaj za hodanje, tzv. *standInit*. Nakon toga se naredbom *moveTo* zadaju željene x i y koordinate i orijentacija θ u lokalnom koordinatnom sustavu robota. Ugrađene funkcije *ALMotion* modula tada obavljaju planiranje hoda prema zadanoj frekvenciji koraka. Primjer planiranja koraka dan je na Slici 5.3.



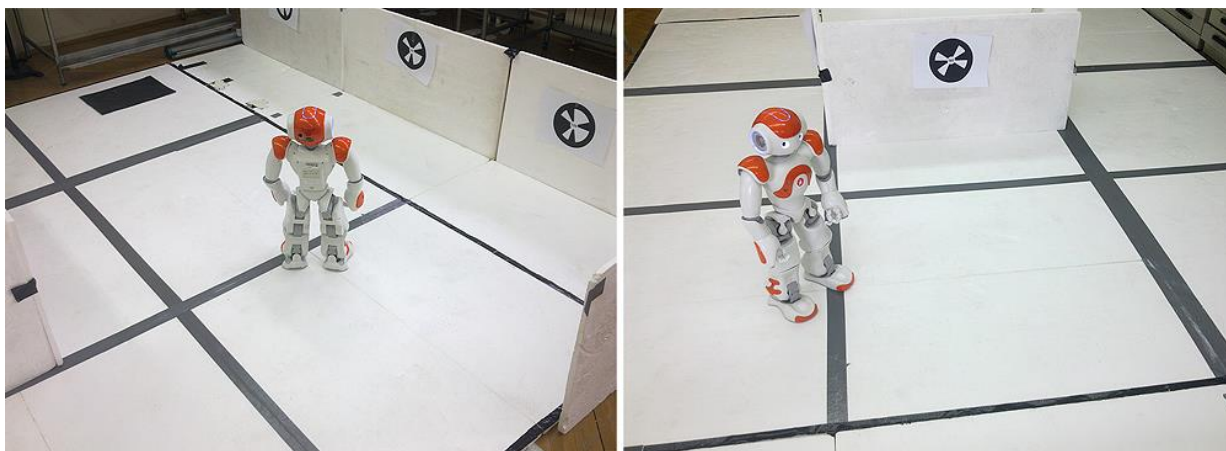
Slika 5.3. Planiranje koraka tijekom hoda robota [3]

Kako se planiranje putanje obavlja u globalnom koordinatnom sustavu, a zadavanje cilja hodanja u lokalnom koordinatnom sustavu robota, koordinate cilja izračunate tijekom planiranja putanje prebacuju se u lokalne koordinate korištenjem transformacijske matrice izračunate kako je opisano u Poglavlju 4. – Lokalizacija. Lokalizacija robota se obavlja nakon svakog uspješno obavljenog intervala hodanja, tako da se transformacijska matrica redovito osvježava i svi proračuni se obavljaju s najnovijim dostupnim podacima iz okoline.

6. OSTVARENA FUNKCIONALNOST

6.1. Igra skrivača

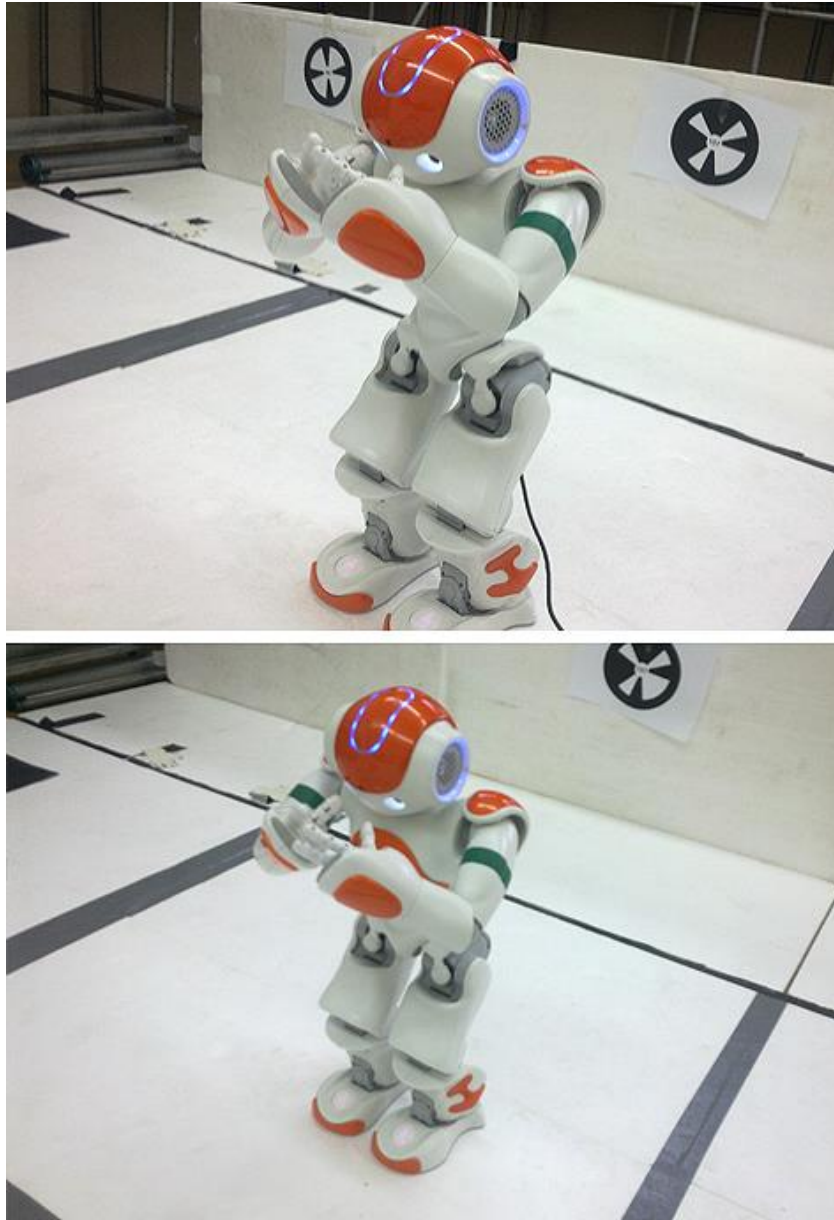
Za potrebe testiranja razvijenih modula izgrađen je poligon za kretanje i rad robota. Robot u ovom okruženju prikazan je na Slici 6.1. Slijedi sažeti opis cjeline dobivene spajanjem svih razvijenih modula.



Slika 6.1. Robot u poligonu s markerima

Robot nakon pokretanja prekriva oči rukama i počinje brojati do deset (Slika 6.2.). Po završetku brojanja, okrećući glavu traži suigračevo lice. Ako ga ne nađe, okrene se u mjestu za 180 stupnjeva te ponovno traži suigrača. Ako ni ova potraga nije uspješna, obavlja se početna lokalizacija robota u prostoru – robot gleda oko sebe dok ne uoči barem dva poznata markera, iz kojih zatim izračunava svoj položaj na mapi.

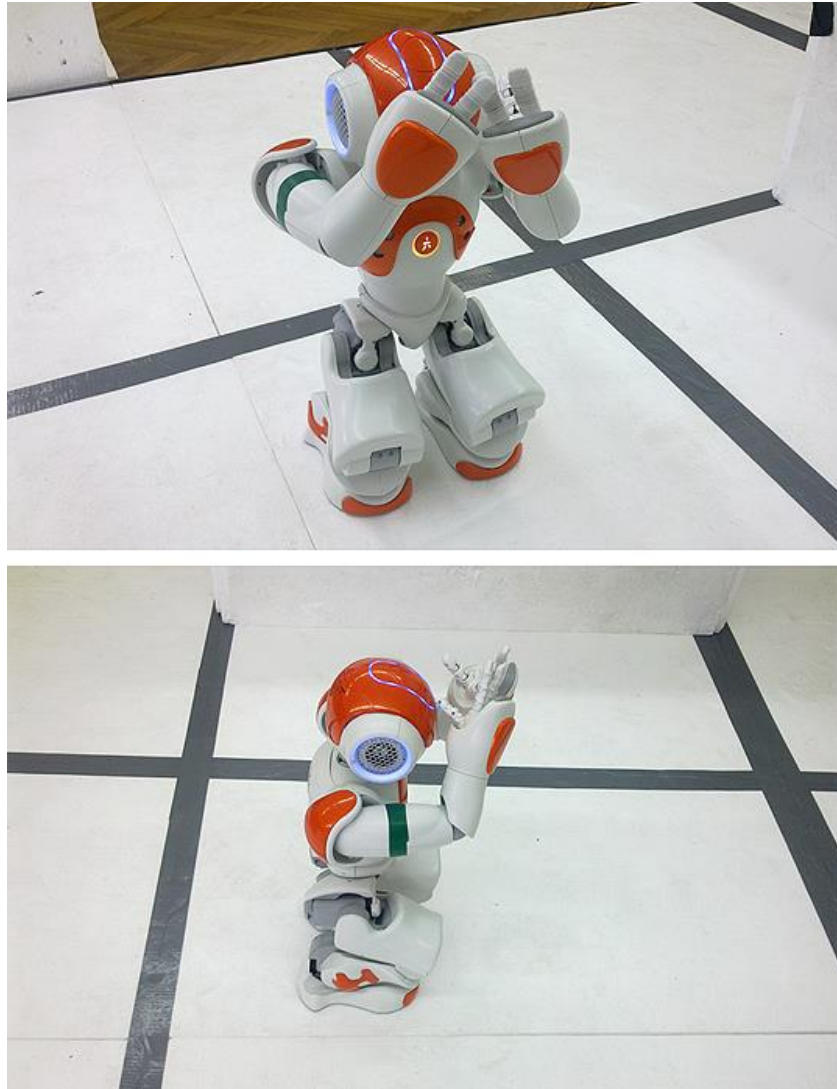
Koristeći novoizračunati položaj robota kao polazište, mapa se dijeli na jednako velika područja pretraživanja tako da se u svakom definira po jedna točka - pozicija do koje će robot hodati. Područja otprilike odgovaraju vidnom polju robota u kojem je on u stanju uspješno prepoznati ljudsko lice. Od ovako određenih točaka bira se ona koja je najbliža trenutnom položaju robota, te planira putanja hodanja do nje.



Slika 6.2. Robot u pozi brojanja

Robot između svaka dva koraka planirane putanje pretražuje prostor oko sebe kako bi našao suigračevo lice, te vrši lokalizaciju kako bi se osigurala točnost navigacije. Nakon što robot stigne do jedne od odabranih točaka pretraživanja, ta se točka i područje oko nje smatraju pretraženim te se ona briše iz niza. Ponovno se obavlja lokalizacija robota i od preostalih točaka pretraživanja odabire ona koja je najbliža trenutnoj poziciji. Ovo se ponavlja sve dok robot ne pronađe suigrača ili pretraži čitavu mapu (niz točaka pretraživanja je ispražnjen), o čemu daje odgovarajuću glasovnu poruku.

Nakon pronalaska suigrača, robot čeka da čuje početak brojanja, koristeći detekciju lica kako bi osigurao da suigrač ne „viri“. Tek nakon što sa sigurnošću začuje izgovorenu riječ „jedan“ dok lice suigrača nije vidljivo, robot će se okrenuti i krenuti sa skrivanjem.



Slika 6.3. Robot u pozi skrivanja

Odabir mjesta za skrivanje vrši se na sličan način kao i odabir točke za pretraživanje prostora, s bitnom razlikom da se ne planira putanja hodanja do najbliže točke, već do druge po redu najmanje udaljene od trenutnog položaja robota. Razlog ovome je želja da se robot udalji što je više moguće u zadanom kratkom vremenu, ograničena time da je siguran i stabilan hod robota prilično spor. Po pristizanju u željenu točku, robot čučne i prekrije se rukama, te čeka

da ga suigrač pronade (Slika 6.3.). Robotu se daje do znanja da je pronađen time da ga se dotakne po glavi, na što on ustaje i daje glasovnu poruku da je igra završena (Slika 6.4.).



Slika 6.4. Signal robotu da je pronađen

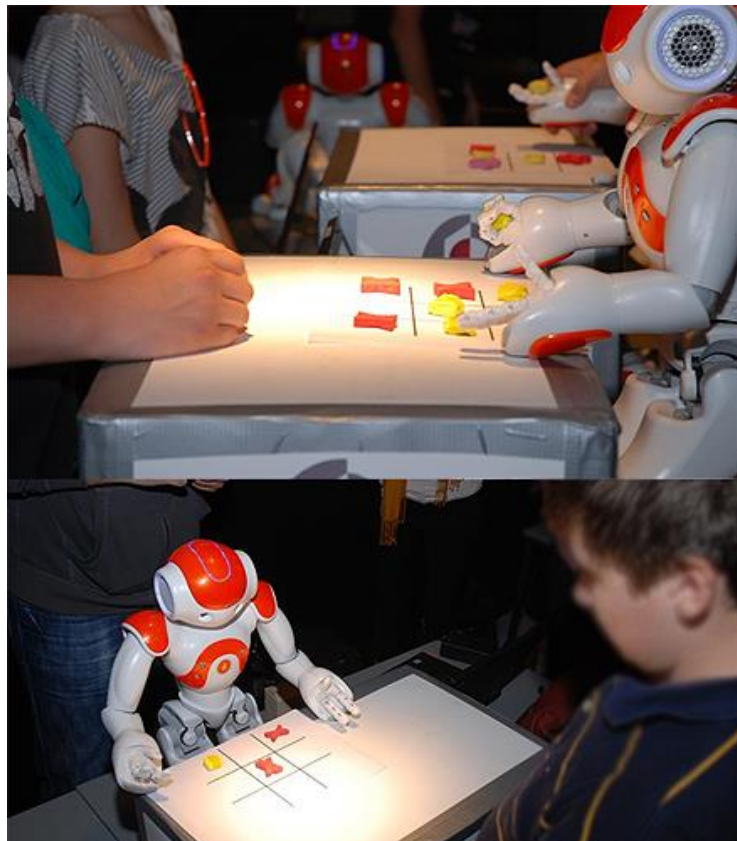
Robot se u svakom pokušaju lokalizacije pokušava ispravno lokalizirati pet puta, a ako to ne uspije (nije uspješno prepoznat dovoljan broj markera, markeri uopće nisu vidljivi, izračunate su koordinate koje se uopće ne nalaze na mapi itd.) obavještava suigrača da se izgubio te da igru treba početi ispočetka.

Video snimka ostvarenih ponašanja nalazi se na [16].

6.2. NAO u socijalnoj robotici

Na mnogobrojnim prezentacijama pokazalo se da humanoidni robot NAO svojim simpatičnim izgledom privlači djecu i da veoma uspješno zadržava njihovu pažnju. Robot nema mogućnost mijenjanja izraza lica, no promjenom boja LED indikatora oko „očiju“ i bogatim govorom tijela veoma dobro glumi ljudske emocije i ponašanje [17].

Na XI. Festivalu znanosti u Zagrebu 2013. godine prezentirano je između ostalog igranje igre križić-kružić koje su razvili Frano Petric i Damjan Miklič iz Laboratorija za robotiku i inteligentne sustave upravljanja (LARICS), prikazano na Slici 6.5. Ovo je jedna od veoma atraktivnih primjena robota NAO na kojoj je izvrsno vidljiv njegov potencijal u području socijalne robotike [18].



Slika 6.5. Igranje društvenih igara

U suradnji Fakulteta elektrotehnike i računarstva i Edukacijsko-rehabilitacijskog fakulteta Sveučilišta u Zagrebu pokrenut je projekt ADORE s ciljem uključivanja humanoidnih robota u dijagnostiku autizma kod djece predškolske dobi. Robot ispunjavanjem različitih propisanih zadataka sudjeluje kao pomoćnik u dijagnostici poremećaja autističnog spektra prema dijelovima instrumenta Autism Diagnostic Observation Schedule [19]. Prednost uključivanja robota u dijagnostički proces dvostruka je: robot uvijek sve akcije izvodi na jednak način, što osigurava konzistentnost podražaja tijekom svih ispitivanja, a svojim sensorima može zapaziti aspekte djetetova ponašanja koji mogu promaknuti ljudskom ispitivaču. Dosadašnji rezultati

interakcije robota s djecom sa i bez poremećaja iz autističnog spektra pokazali su se iznimno obećavajućima, a u tijeku su klinička ispitivanja.

Sama tvrtka Aldebaran Robotics pokrenula je projekt ASK NAO (Autism Solution for Kids) [20]. Cilj projekta je podrška učiteljima u školama pomoću razvijene biblioteke ponašanja robota za terapijski rad s djecom s autizmom te pomoć pri učenju i postizanju višeg stupnja autonomije djeteta. Ponašanja uključuju igre pogađanja emocija, šetnju s robotom, jednostavne kartaške igre i vježbe verbalne i neverbalne komunikacije. Kao neke od prednosti uvođenja robota u rad s djecom s autizmom vrijedi navesti njegovu predvidljivost u interakcijama, koja smanjuje anksioznost kod djece, jednostavnost izgleda koja smanjuje rizik od pretjerane stimulacije, mogućnost prilagodbe potrebama svakog pojedinog djeteta i odsutnost problema kao što su umor ili loše raspoloženje [21].

Jedan od glavnih nedostataka robota na trenutnom stupnju razvoja, koji ih ograničava u socijalnim ulogama, jest taj da nemaju mogućnost održavanja nekontinuiranih i dugih socijalnih interakcija, raspoređenih unutar razdoblja duljih od nekoliko sati ili čak dana, uz uzimanje u obzir i prilagođavanje ponašanja sjećanjima iz prijašnjih interakcija s istim korisnikom. Da bi interaktivna ponašanja robota bila uvjerljivo nalik ljudskima, robot mora moći korištenjem učenja i adaptivnih algoritama pružiti dojam konzistentnog odnosa s korisnikom kroz dulji niz susreta. Ovaj je aspekt interakcije čovjeka i robota veoma izazovno područje i u samim počecima razvoja [22].

7. Zaključak

U ovom je radu predstavljen razvijeni sustav sastavljen od niza zasebnih elemenata koji omogućava igranje igre skrivača s humanoidnim robotom NAO. Izrađeni su moduli koji se pokreću lokalno na robotu i koji osiguravaju osnove rada ponašanja potrebnih za uspješno igranje igre: prepoznavanje suigračeva lica, prepoznavanje izgovorene riječi, nekoliko animacija i poza te detekcija posebnih markera. Riješen je problem lokalizacije robota u prostoru i planiranja putanje za kretanje robota tijekom pretraživanja prostorije, čime mu je omogućeno autonomno gibanje u zatvorenom prostoru u skladu sa zahtjevima igre.

Za ispitivanje rada razvijenog sustava pripremljen je poligon za kretanje robota te je isprobana funkcionalnost cjeline. Najčešće se poteškoće u radu javljaju kao posljedica lošeg osvjetljenja prostorije jer robot za snalaženje u prostoru od senzora najviše koristi kameru. U takvim nepovoljnim uvjetima i detekcija lica puno je manje uspješna, uz pojavu lažno prepoznatih „lica“ koja dovode do prekida igre jer robot smatra da je pronašao suigrača iako u stvarnosti nije. Problem koji se također javlja je nastajanje većih odstupanja od zadane putanje tijekom hoda robota, pogotovo u smislu orijentacije. Stopala robota sklona su proklizavanju i ugrađena odometrija mu je veoma nepouzdana, tako da je koliko-toliko siguran hod moguć samo na malim udaljenostima prije nego što postane nužno uvesti korekcije ponavljanjem postupka lokaliziranja, što pak značajno usporava kretanje.

U daljnjem radu slijedi izrada boljeg klasifikatora za raspoznavanje potrebnih riječi koje suigrač izgovara te rad na poboljšanju hoda robota razvojem nove sekvence koraka.

Uspješnim stvaranjem jedne interaktivne cjeline od niza ponašanja humanoidnog robota pokazane su neke od njegovih brojnih mogućnosti funkcioniranja u socijalnom okruženju. Ostvaren je napredak u radu s robotom NAO jer su razvijeni moduli korisni u čitavom nizu situacija, s naglaskom na kombinaciju autonomnosti i interakcije s čovjekom.

8. Literatura

- [1] Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3), 143-166.

- [2] Müller, J., Stachniss, C., Arras, K., & Burgard, W. (2008). Socially inspired motion planning for mobile robots in populated environments. In *Proc. of International Conference on Cognitive Systems*.

- [3] NAO Software 1.12.5 documentation. Aldebaran Robotics.
<https://developer.aldebaran-robotics.com/doc/1-14/>. Datum pristupa: 29.4.2014.

- [4] Claraco, J. L. B. (2008). Development of Scientific Applications with the Mobile Robot Programming Toolkit. *The MRPT reference book. Machine Perception and Intelligent Robotics Laboratory, University of Málaga, Málaga, Spain*.

- [5] Blanco, J. L. (2011). Mobile Robot Programming Toolkit (MRPT).
<http://www.mrpt.org/>. Datum pristupa: 29.4.2014.

- [6] Acapela Group. <http://www.acapela-group.com/index.html> . Datum pristupa: 29.4.2014.

- [7] Nuance Communications, Inc. (2013). Aldebaran Robotics and Nuance Revolutionize Human – Machine Interaction. http://www.nuance.com/company/news-room/press-releases/Aldebaran_web.docx . Datum pristupa: 29.4.2014.

- [8] OKI Electric Industry. <http://www.oki.com/> . Datum pristupa: 29.4.2014.

- [9] Ivanjko, E., Petrović, I., & Vašak, M. (2004). Praćenje položaja mobilnog robota ultrazvučnim osjetilima. *AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 45(3-4), 145-154.
- [10] K. JooHyun. (2008). Place recognition and kidnapped robots - Visual Recognition and Search. *University of Texas, Austin, Texas, USA*.
http://www.cs.utexas.edu/~grauman/courses/spring2008/slides/Joohyun_place_recognition.pdf Datum pristupa: 29.4.2014.
- [11] Thrun, S. (2002). Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1-35.
- [12] Lončarić, S., & Bijmens, B. (2004). Digitalna analiza slike.
- [13] Drmac, Z., Hari, V., Marušić, M., Rogina, M., Singer, S., & Singer, S. (2003). Numerička analiza. *Internetsko izdanje (PMF-Matematički odjel), Zagreb*.
- [14] Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. (2006). *Algorithms*. McGraw-Hill, Inc..
- [15] Kajita, S., & Tani, K. (1991, April). Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* (pp. 1405-1411). IEEE.
- [16] Video snimka robotskog ponašanja pri igri skrivača (2014) dostupna na adresi: <http://larics.rasip.fer.hr/public/zkovacic/movies/naohidenseek.mp4>
- [17] Aragon, K. C. (2012). Seeing Signs: Pre-Adolescent Recognition of Social Body Language in Humanoid Robots. *CUNY Brooklyn College, Brooklyn, NY*.
- [18] Kovačić Z., Petric F., Miklić D., Babić A., & Hrvatinić K. (2014). NAO Plays a Tic-Tac-Toe Game: Intelligent Grasping and Interaction. *Laboratory for Robotics and*

*Intelligent Control Systems, Faculty of Electrical Engineering and Computing,
University of Zagreb, Zagreb, Croatia.*

- [19] Lord C., Risi S., Lambrecht L., Cook Jr E. H., Leventhal B. L., DiLavore P. C., ... & Rutter M. The Autism Diagnostic Observation Schedule—Generic: A standard measure of social and communication deficits associated with the spectrum of autism. *Journal of autism and developmental disorders.* (2000). 205-223.

- [20] Aldebaran Robotics ASK NAO Initiative. <http://asknao.aldebaran-robotics.com/> . Datum pristupa: 29.4.2014.

- [21] Shamsuddin, S., Yussof, H., Ismail, L., Hanapiah, F. A., Mohamed, S., Piah, H. A., & Ismarrubie Zahari, N. (2012, March). Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO. In *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on* (pp. 188-193). IEEE.

- [22] Baxter, P., Belpaeme, T., Canamero, L., Cosi, P., Demiris, Y., Enescu, V., ... & Wood, R. (2011). Long-term human-robot interaction with young users. In *IEEE/ACM Human-Robot Interaction 2011 Conference (Robots with Children Workshop)*.

Humanoidni robot NAO u društvenoj interakciji: elementi sustava na primjeru igre skrivača

Sažetak

Cilj rada jest istražiti sposobnost humanoidnog robota NAO-H25 da na autonoman način sudjeluje u interakciji s ljudima. Svojim dizajnom NAO se pokazao osobito prikladnim za rad s djecom te su u ovom radu osmišljeni i implementirani elementi društvene interakcije potrebni za uspješno sudjelovanje robota u igri skrivača. Dizajniran je i izgrađen prostor predviđen za odvijanje igre. Osmišljen je i implementiran algoritam lokalizacije robota te algoritam za pretraživanje prostora i kretanje robota. Implementirani su moduli za vizualnu, zvučnu i taktilnu percepciju suigrača te ponašanje robota tijekom igre korištenjem prikladnih gesti, govora tijela i glasovnih poruka. Opisan je krajnji rezultat rada i dan je kratki osvrt na ostale primjene NAO robota u socijalnoj robotici.

Ključne riječi: NAO, humanoidni robot, socijalna robotika, lokalizacija, pretraživanje 2D prostora, igra skrivača

The NAO humanoid robot in social interaction: system elements demonstrated on a game of hide and seek

Abstract

The goal is to explore the capability of the humanoid robot NAO-H25 to autonomously take part in human interaction. Due to its design, NAO has shown itself to be particularly suitable for work with children, so in this work various elements necessary for the robot to be able to take part in a game of hide and seek have been developed and implemented. A testing polygon was designed and built for the game to take place in. A localisation algorithm was developed, as well as algorithms for spatial search and robot motion. Modules for visual, audio and tactile perception of the human player were created, along with several robot behaviours required to play the game, including appropriate gestures, body language and spoken messages. The final results of the work are presented and a short review of other applications of the NAO robot in social robotics is given.

Keywords: NAO, humanoid robot, social robotics, localisation, 2D spatial search, hide and seek