

II. RAPORTARE ȘTIINȚIFICĂ

FAZA DE EXECUȚIE NR. I

TITLUL: *STUDIUL DETALIAT AL ROBOȚILOR TIAGO SI BAXTER SI PEPPER*

Avizat,

Coordonator

Universitatea Politehnică din București

Reprezentant Legal

Mihnea COSTOIU



Director Proiect

Prof. dr. ing. Adina Magda FLOREA

Agent Economic

Centrul IT pentru Știință și Tehnologie S.R.L.

Reprezentant Legal

Dr. Ing. Bogdan Cramariuc



Responsabil de proiect,

Dr. Ing. Bogdan Cramariuc

SPARC

Servicii Performante de Asistență a Clienților prin Platforme Robotice

cod: PN-III-P2-2.1-BG-2016-0425

Grant in cadrul Programului Național PNIII - Programul 2 - Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare, Transfer de cunoaștere la agentul economic „Bridge Grant”

Raport inițial Tiago versus Baxter versus Pepper

Denumire Etapa:

Studiul detaliat al roboților Tiago si Baxter si Pepper

Rezultate Etapa:

Raport inițial Tiago versus Baxter versus Pepper

Durata proiectului:

2 ani cu începere 30.09.2016

Rezumat

Principalul obiectiv al proiectului este proiectarea și implementarea unei platforme pentru realizarea integrată a planificării sarcinilor roboților de interes în proiect, comanda execuției planurilor definite și monitorizarea acțiunilor acestora. Platforma trebuie să permită personalizarea facilă a programului robotului dar și crearea unui comportament adaptiv la anumite condiții neprevăzute.

Agentul economic care face obiectul actualei propuneri de proiect, Centrul IT pentru Știință și Tehnologie SRL (CITST) este un IMM orientat spre cercetare și dezvoltare de produse inovatoare. Tehnologia disponibilă la CITST pe care se bazează proiectul este robotul TIAGO produs de Pal Robotics. CITST vrea să exploateze mai bine tehnologia robotică disponibilă în companie (Tiago) oferind-o ca serviciu (închiriere, leasing) pentru potențialii clienți. Scenariile de utilizare avute în vedere se referă la promovare de produse și/sau asistență în locuri publice: centre comerciale, muzee, bănci, expoziții. CITST are în vedere și achiziționarea unui robot Pepper în același scop. Tehnologia disponibilă la UPB este reprezentată de roboții Baxter de la Rethink Robotics și Pepper de la Aldebaran Robotics.

Obiectivul etapei curente este realizarea unui studiu detaliat al celor trei roboți menționați, cu accent pe capacitățile acestor roboți de a realiza scenariile din proiect, și de a vedea mai departe cum vom proiecta și implementa platforma robotică avută în vedere, compatibilă cu toți cei trei roboți dacă se poate, nu numai cu robotul Tiago.

Colectivul proiectului este format din membrii grupului de cercetare AIMAS de la Facultatea de Automatică și Calculatoare din UPB și de studenți din cadrul programului de masterat Artificial Intelligence, tot de la Facultatea de Automatică și Calculatoare.

Cuprins

1. Introducere și scopul raportului.....	2
2. Caracteristicile hardware ale roboților	4
2.1. Robotul Tiago	4
2.2. Robotul Baxter	5
2.3. Robotul Pepper	6
2.4. Comparație	7
2.4.1. Caracteristici fizice	7
2.4.2. Analiză	7
3. Caracteristicile sistemelor de operare ale roboților	7
3.1. Sistemul de operare ROS	8
3.2. Sistemul de operare NAOqi	9
3.3. Comparație	9
4. Caracteristici de vedere artificială	10
4.1. Sistemul de operare ROS	10
4.2. Sistemul de operare NAOqi	11
4.3. Comparație	12
5. Caracteristici de navigare.....	12
5.1. Sistemul de operare ROS	12
5.2. Sistemul de operare NAOqi	13
5.3. Comparație	14
6. Caracteristici de interacțiune	14
6.1. Capacități de interacțiune fizică.....	14
6.1.1. Robotul Tiago.....	14
6.1.2. Robotul Baxter	14
6.1.3. Robotul Pepper	15
6.2. Capacități de interacțiune în limbaj natural.....	16
6.2.1. Sistemul de operare ROS	16
6.2.2. Sistemul de operare NAOqi.....	17
6.3. Comparație	17
7. Concluzii.....	18
Bibliografie	19

1. Introducere și scopul raportului

Roboții încep să exercite un impact considerabil în economie și societate, prin reducerea costurilor, îmbunătățirea calității și a condițiilor de lucru, asistarea persoanelor cu nevoi speciale sau, recent, deservirea și asistarea clienților în diferite domenii de activitate. Veniturile actuale la nivel mondial în domeniul producerii și vânzării de roboți, de diferite tipuri, sunt la nivelul a 22 miliarde euro și se așteaptă ca aceste venituri să crească la un nivel situat între 50 și 62 de miliarde de euro până în anul 2020.

În acest context, rolul roboților în asistența clienților, fie a roboților software fie a celor hardware, a devenit din ce în ce mai important și de dorit. În ultimii ani, au apărut din ce în ce mai mulți roboți (hardware) umanoizi sau cu capacități umanoide care nu mai sunt strict un produs de cercetare ci sunt comercializați pentru a reprezenta o prezență interesantă, utilă și avantajoasă în interacțiunea cu un număr mare de clienți. Acești roboți, dotați cu software bazat pe tehnici de inteligență artificială, au început să fie programați astfel încât să fie capabili să promoveze produse, să răspundă personalizat la întrebări sau să ghideze clienții în magazine, expoziții, sau pentru participarea la evenimente.

Există deja numeroși furnizori care oferă soluții robotizate, care sunt potrivite pentru categoriile de aplicații enumerate mai sus, așa cum se menționează și în documentul propunerii de proiect. Printre aceștia se numără și roboți folosiți în acest proiect: Tiago de la PAL Robotics¹, Baxter de la Rethink Robotics² și Pepper, proiectat de gigantul japonez Softbank, în colaborare cu Aldebaran Robotics³.

Chiar dacă există o mare varietate de platforme robotizate, este încă o provocare adaptarea rapidă a comportamentului unui robot asistent de la un scenariu la altul, în funcție de cerințele clientului. Principala provocare constă în timpul necesar pentru a programa/reprograma comportamentul robotului având în vedere noile specificații ale unei aplicații.

În acest context, principalul obiectiv al proiectului este proiectarea și implementarea unei platforme pentru realizarea integrată a planificării sarcinilor robotului, comanda execuției planurilor definite și monitorizarea acțiunilor robotului. Activitatea robotului, pe baza acestei platforme, trebuie să fie ușor configurabilă la nivelul scopurilor de realizat ca o compunere de comportamente de bază (goal level programming based on simple behaviors). O astfel de platformă permite personalizarea facilă a programului robotului dar și crearea unui comportament adaptiv la anumite condiții neprevăzute.

Abordarea propusă, de tip „goal level programming”, ajută pe dezvoltator să determine cu ușurință acțiunile întreprinse de un robot, folosind structura comportamentelor de bază, de exemplu: detectarea unui utilizator, răspunsul la întrebările utilizatorului, urmărirea utilizatorului, navigarea către o anumită locație, detectarea unui obiect, punctarea spre un anumit obiect.

Provocările acestui demers cuprind două aspecte:

- Definirea și implementarea unui set de comportamente de bază de sine stătătoare, din care orice scenariu pentru una din categoriile de aplicații menționate poate fi exprimat cu ușurință. Aceste comportamente de bază trebuie să îndeplinească condiții ca:
 - Siguranța în interacțiune cu utilizatorii umani
 - Funcționarea corectă și robustă (de exemplu: toleranța la schimbările neașteptate de mediu, gradul scăzut de erori, eroare de manipulare)

1 <http://tiago.pal-robotics.com/>

2 <http://www.rethinkrobotics.com/baxter/>

3 <http://www.smashingrobotics.com/aldebaran-unveils-pepper-personal-emotional-robot/>

- Definirea și implementarea unei planificări dinamice bazate pe scop și realizarea unui cadru capabil să combine în mod automat comportamentele de bază într-un plan de acțiune coerent care să realizeze obiectivul de utilizare dorit. Modelul dezvoltat trebuie să fie în măsură să răspundă în mod continuu la evenimentele date de interacțiunea cu utilizatorul și schimbările mediului înconjurător. Acestea vor acționa ca mecanisme declanșatoare ale unei proceduri replanificare, care va actualiza scopurile / subscopurile robotului în mod corespunzător.

Primul pas către obiectivele de mai sus este analiza și comparația capacităților oferite de cele trei platforme robotizate disponibile în cadrul proiectului: Tiago, Baxter și noua achiziție Pepper. Analiza și comparația care se face în acest raport are în vedere categoriile de aplicații prezentate anterior. Un set de comportamente de bază ale robotului trebuie să fie activat, în scopul de a realiza suportul de asistență descris:

- Comportamentele legate de vedere artificială:
 - Capacitatea de a detecta și a contoriza numărul de persoane din raza de acțiune a robotului
 - Abilitatea de a recunoaște (identifica) un set de persoane distincte (de exemplu: vânzătorii dintr-un magazin, personalul dintr-un muzeu) sau recunoașterea oamenilor recentți întâlniți
 - Capacitatea de a detecta un obiect și poziția relativă a robotului față de un obiect perceput de către camerele de luat vederi ale acestuia
 - Abilitatea de a recunoaște markeri vizuali, în scopul de a calibra mișcarea sau poziția în raport cu obiectele vizate
- Comportamente legate de navigare:
 - Capacitatea de a învăța, memora și determina poziționarea într-un mediu static (de exemplu: în cazul în care nici un om din jurul lui nu se deplasează)
 - Capacitatea de a construi și a urma un plan de mișcare dintr-un punct în altul într-un mediu static
 - Capacitatea de a ține cont de prezența persoanelor la planificarea unui traseu
 - Capacitatea de a urmări o țintă umană (adică să însoțească persoana de-a lungul unei rute)
- Comportamente legate de comunicare:
 - Capacitatea de a detecta când o persoană dorește să interacționeze cu robotul
 - Capacitatea de a participa printr-o simplă comandă la interacțiuni în limbaj natural
 - Abilitatea de a crea și dialoguri specifice din domeniu (de exemplu: formule de salut, să ceară direcții, solicite locația curentă, să discute despre un articol din prezentare, să răspundă la întrebări cu privire la un anumit articol din prezentare)

În mod evident, comportamentele de mai sus trebuie să fie însoțite de un mecanism de planificare la nivelul robotului, care:

- Reacționează la evenimentele care vin de la cererile utilizatorului sau schimbările de mediu și să le transforme în obiective
- Construiește planuri de execuție și de rezervă corespunzătoare (prin structura comportamentelor de bază) pentru a atinge obiectivele și care să permită replanificarea în caz de eșec
- Monitorizează execuția planului executat de robot pentru a crea un comportament coerent al robotului

Luând în considerare ideile discutate mai sus, analiza și comparația celor trei platforme robotizate disponibile în proiect va avea loc după cum urmează. În capitolul 2 raportul descrie hardware-ul (senzori și dispozitiv de acționare) caracteristic fiecărui robot, în special acele elemente care sunt relevante pentru tipurile de comportamente de bază enumerate anterior. Capitolul 3 continuă cu o prezentare a caracteristicilor generale și de funcționare ale roboților (de exemplu NAOqi, Sistemul de operare al robotului Pepper) care se execută pe fiecare robot

și permit programarea aplicațiilor. Capitolele de la 4 până la 6, aruncă o privire mai atentă asupra bibliotecilor, care pot fi utilizate pentru dezvoltarea comportamentelor de bază menționate într-un mod uniform și transferabil (adică utilizabil pe alți roboți care folosesc aceleași sisteme de operare). Ele se concentrează pe caracteristicile de viziune, de mișcare și de interacțiune (limbaj fizic și natural de bază). Raportul se încheie cu capitolul 7 destinat concluziilor.

2. Caracteristicile hardware ale roboților

Acest capitol face o trecere în revistă a platformelor robotizate puse la dispoziție în cadrul proiectului: Tiago (de la CITST), Baxter și Pepper (de la UPB). Sunt discutate scopurile propuse ale roboților (așa cum sunt date de producător), precum și specificațiile hardware-ul. După prezentarea platformelor robot, se face o comparație, evidențiind gradul de aliniere și adecvare al fiecărui robot cu obiectivele proiectului.

2.1. Robotul Tiago

Tiago este fabricat de compania spaniolă PAL Robotics⁴. Acesta este conceput ca un robot de cercetare cu capacități de manipulare a obiectelor (dat fiind brațul său extensibil care se termină cu un element de prindere – vezi Figura 1). Dimensiunea sa, capacitățile de detectare și de mișcare îl fac potrivit pentru tipul de scenarii de asistare preconizate în cadrul proiectului (de exemplu: ghid asistent la expoziții, ghid de muzeu, vânzător).

Tiago măsoară 146 mm înălțime, cu o bază de 54cm diametru și o greutate de 70 kg. Placa de baza a robotului rulează un procesor Intel 4.40 GHz i7 Haswell (CPU), cu 16 GB de memorie RAM și 256 GB SSD pentru stocare. Robotul suporta 3 tipuri de conectivitate, Ethernet (2 porturi Gigabit Ethernet), WiFi (802.11 a / b / g / n / ac) și Bluetooth (Bluetooth 4.0).

Tiago are mai mulți senzori care îl ajuta să interacționeze cu mediul înconjurător: un LIDAR (light detection and ranging), trei sonare în spate, o unitate de măsurare inerțială (pentru a monitoriza forțele de inerție și postura robotului).



Fig. 1 Vedere generală a robotului Tiago

⁴ <http://tiago.pal-robotics.com/>

Pentru interacțiunea cu utilizatorii, capul lui Tiago este echipat cu un mecanism de panoramare – înclinare (en. pan-tilt), două camere DoF, un microfon stereo, un difuzor și o cameră RGB-D. Pe partea de sus a capului există o suprafață plană, cu puncte de fixare ce permit utilizatorului să adauge senzori sau echipamente noi. În plus, Tiago are un suport de laptop, o suprafață plană (33 cm x 27,5 cm) plasată în partea de sus a trunchiului, chiar în spatele capului său.

Brațul lui Tiago are 7 articulații. El are o greutate de 10 kg și poate duce o sarcină utilă de până la 2,8 kg. Pe brațul lui Tiago este fixat un element de prindere paralel, capabil să suporte o sarcină utilă de până la 2 kg.

Utilizatorii pot controla robotul Tiago în mod direct cu ajutorul unui joystick, care poate fi folosit pentru a muta baza robotului, trunchiul acestuia, capul, brațul manipulator și cleștele de prindere al brațului.

2.2. Robotul Baxter

Baxter este un robot umanoid dezvoltat de Rethink Robotics⁵, proiectat să ocupe un spațiu relativ mic și pentru a realiza o mulțime variată de joburi de producție: de la linia de încărcare și mașina care încarcă, la ambalarea și manipularea materialelor. În timp ce este considerat în primul rând un robot industrial, capacitățile lui Baxter (atât hardware cât și software) permit roboților să fie utilizați ca baze experimentale pentru tipul de scenarii luate în considerare în cadrul proiectului. Specificațiile hardware sunt prezentate mai jos.

Robotul Baxter are două brațe robotizate cu șapte grade de libertate. Dimensiunile sale sunt de 177,8 cm înălțime, cu un soclu reglabil și are o rază maximă de 121.0 cm. CPU de la bordul lui Baxter este un Intel Core i7-3770. Robotul are 4GB de memorie RAM și o placă grafică integrată de tip Intel HD4000. Stocarea pe termen lung se face prin intermediul unei memorii flash eMMC de 8GB și a unui SSD de 128 GB.

Brațele lui Baxter au 7 articulații (3 de îndoire și 4 de rotație), care permit un grad mare de flexibilitate în mișcare. Fiecare braț cântărește 21,3 kg și are o forță de prindere de 35N și o sarcină utilă maximă de 2,2 kg.



Fig. 2 Vedere generală a robotului Baxter cu bază mobilă adițională

⁵ <http://www.rethinkrobotics.com/baxter/>

Baxter poate interacționa în mod suplimentar cu utilizatorii prin informație vizuală ce poate fi afișată pe un ecran încorporat cu o rezoluție de 1024 x 600 pixeli.

2.3. Robotul Pepper

Pepper este un robot creat de SoftBank Robotics și este primul robot proiectat pentru a trăi împreună cu oamenii. Aspectuos și prietenos, Pepper este mult mai mult decât un robot, este un companion capabil să comunice cu un utilizator prin intermediul unor interfețe intuitive: voce, atingeri și emoții (vezi Figura 3).

Dimensiunile robotului Pepper sunt 1210 mm înălțime, cu o bază de 480 x 425 mm. Folosindu-și mâinile poate ajunge la 1335 mm înălțime, lățime de 1196.9 mm și o lungime 648,2 mm. Placa de bază a lui Pepper conține un procesor Intel Atom E3845, 4GB de memorie RAM și o placă grafică integrată de tip Intel HD Graphics. Pentru stocarea de date pe termen lung sunt folosite o memorie flash eMMC de 8GB și un card SDHC de 16 GB.

Robotul permite conectivitate prin Ethernet și Wifi. Dispune, de asemenea, de o tableta cu o rezoluție de 1280x800 pixeli ce poate rula aplicații independente pe sistemul de operare Android. Pepper are o multitudine de senzori ce îi permit să interacționeze cu mediul: o unitate inerțială⁶, LIDARe⁷, sonare (pentru estimarea distanței la obstacole din mediu)⁸ și senzori de infra-roșu⁹.

Pentru interacțiunea cu utilizatorii, Pepper dispune de senzori tactili (pe cap și pe mâini)¹⁰, precum și de 3 camere montate pe cap: 2 camere 2D¹¹ și una 3D (cu senzor de adâncime)¹². Pepper poate percepe și produce sunete prin patru microfoane¹³ și două difuzoare¹⁴. Din punct de vedere al mișcărilor, și implicit motoare, Pepper încearcă să semene cu cât mai mult posibil cu oamenii, permițând mișcări ale capului, mâinilor și șoldurilor¹⁵.



Fig. 3 Vedere generală a robotului Pepper

6 http://doc.aldebaran.com/2-4/family/pepper_technical/inertial_pep.html

7 http://doc.aldebaran.com/2-4/family/pepper_technical/laser_pep.html

8 http://doc.aldebaran.com/2-4/family/pepper_technical/sonar_pep.html

9 http://doc.aldebaran.com/2-4/family/pepper_technical/irspot_pep.html

10 http://doc.aldebaran.com/2-4/family/pepper_technical/contact-sensors_pep.html

11 http://doc.aldebaran.com/2-4/family/pepper_technical/video_pep.html

12 http://doc.aldebaran.com/2-4/family/pepper_technical/video_3D_pep.html

13 http://doc.aldebaran.com/2-4/family/pepper_technical/microphone_pep.html

14 http://doc.aldebaran.com/2-4/family/pepper_technical/loudspeaker_pep.html

15 http://doc.aldebaran.com/2-4/family/pepper_technical/motors_pep.html

2.4. Comparație

În această secțiune se realizează o comparație între diferitele componente hardware ale celor trei roboți.

2.4.1. Caracteristici fizice

Din punct de vedere fizic, dimensiunile roboților Pepper și Tiago îi fac pe aceștia mai potriviți ca unități de asistență (raportat la greutate și mărime).

Capabilitățile de calcul sunt semnificativ mai bune pentru Tiago și Baxter (procesoare Intel i7 și memorie RAM suficientă). Cu toate acestea, în cadrul proiectului se așteaptă ca puterea de calcul să nu fie o problemă majoră, permițând de asemenea utilizarea unui robot precum Pepper. Abilitățile de vedere artificială sunt în mod clar mai puternice pentru Pepper și Tiago, care au fost concepuți pentru a interacționa cu oamenii, pentru a detecta obiecte și a naviga în împrejurimile lor.

Tiago și Pepper au, de asemenea, o mai bună durată de viață a bateriei (~ 8h fiecare la funcționalitate normală), în timp ce Baxter are necesități de putere crescute (care este în concordanță cu designul său ca un robot industrial).

Din punctul de vedere al detecției, senzorii roboților Tiago și Pepper sunt adaptați pentru recunoașterea oamenilor și a obiectelor (camere RGB și de adâncime), precum și pentru navigație (lasere, sonare și senzori infraroșu). Senzorii Baxter servesc în principal capacităților sale de manipulare. Baza mobilă de pe Baxter permite echipamente suplimentare de navigare, dar adăugarea acestora este în prezent un obiectiv în curs de pregătire pentru unitatea disponibilă în proiect. În cele din urmă, din perspectiva de afișare a informațiilor, Pepper este cel mai bine adaptat, dispunând de o tabletă care poate fi programată separat și care este capabilă să ruleze aplicații bazate pe Android.

2.4.2. Analiză

Din partea de prezentare generală devine evident faptul că dimensiunile robotului Pepper sunt mai potrivite pentru tipul de scenarii luate în considerare în cadrul proiectului decât cele ale lui Baxter și TIAGo.

Timpul de funcționare al bateriei Baxter este limitat la o oră, o perioadă scurtă de timp pentru un robot mobil. Mai mult, robotul Baxter care este disponibil în proiect nu este echipat cu senzori de navigare (de exemplu găsirea gamei de lasere, sonarele). Din această cauză cartografierea mediului, planificarea traseului și evitarea obstacolelor vor fi mai dificile dacă nu sunt cumpărați și instalați senzori suplimentari. Baxter nu are un adaptor WIFI, ceea ce înseamnă că accesul la serviciile online, care nu rulează pe robot, este împiedicat când robotul nu este staționar.

Pe de altă parte, roboții Pepper și TIAGo conțin tipul de senzori și dispozitive de conectare care lipsesc la robotul Baxter, ceea ce îi face mai potriviți pentru dezvoltarea aplicațiilor propuse în proiect.

3. Caracteristicile sistemelor de operare ale roboților

Pentru o bună programare a roboților cât și pentru dezvoltarea unei platforme de aplicații care să poată să fie adaptată la toate cele trei tipuri de roboți este necesară investigarea caracteristicilor sistemelor de operare disponibile pe roboți.

Luând în considerare tipul de roboți utilizați în cadrul proiectului, sunt discutate în continuare două sisteme de operare: sistemul de operare ROS (Robot Operating System) disponibil pe roboții Tiago și Baxter și sistemul de operare NAOqi disponibil pe robotul Pepper.

Sistemul ROS este realizat ca un efort de colaborare a numeroase laboratoare de robotică cu scopul de a crea un cadru flexibil și deschis pentru dezvoltarea de software și aplicații pentru roboții care suportă ROS. Este un ansamblu de instrumente, biblioteci și convenții care au ca scop simplificarea sarcinii de a crea un comportament complex și robust al unui robotului pe o varietate largă de platforme robotizate.

Această compatibilitate extinsă este de asemenea una dintre principalele puncte forte ale ROS. Baxter și Tiago sunt compatibili implicit cu ROS, în timp ce pentru robotul Pepper există posibilitatea integrării între NAOqi și ROS.

NAOqi este un sistem de operare care permite dezvoltarea de aplicații pentru suita de platforme robotizate dezvoltate de Softbank Robotics și Aldebaran (Pepper, Nao și Romeo). Este în esență o colecție de module care permit interacțiuni la nivel înalt (de exemplu: merge mai mult de un metru, detectează numărul de persoane din fața robotului) cu senzori și elemente de acționare prezente pe acești roboți. Există o comunitate puternică de suport pentru sistemul de operare NAOqi, dar sistemul este disponibil numai pentru roboții realizați de companiile menționate mai sus.

3.1. Sistemul de operare ROS

Sistemul de operare ROS (Robot Operating System) este o colecție de peste o mie de module open-source pentru dezvoltarea software-ului robotului. Acesta oferă un cadru flexibil pentru scrierea de software pentru roboți și încurajează colaborarea pentru dezvoltarea de module software pentru roboții compatibili ROS. Comunitatea ROS este compusă din 15.000 de utilizatori activi, cu mai mult de 1,4 milioane de vizualizări de pagini în acest an.¹⁶

Acesta este un sistem de operare la nivel meta prin design, oferind abstractizarea hardware-ului, control al dispozitivelor de nivel scăzut, transmiterea de mesaje între procese și de administrarea pachetelor. De asemenea prevede instrumente și biblioteci de comunicare între mai multe calculatoare sau roboți care rulează ROS.

Rulând ca un framework distribuit de procese, numite Noduri, sistemul permite nodurilor slab conectate să comunice prin intermediul transmiterii de mesaje. Fiecare nod poate fi inclus într-un pachet sau într-o stivă care poate fi partajat sau distribuit prin intermediul sistemului de administrare a pachetelor ROS.

Procesele în ROS comunică folosind "Computation Graph", o rețea de tip peer-to-peer. Conceptele de bază ale Computation Graph sunt noduri, Master, Serverul Parameter, mesaje, servicii, subiecte și bags.

Nodurile sunt de obicei procese care permit comunicarea cu anumite module, cum ar fi lasere, rangeri, camere video RGB și de adâncime, trimițând date sau primind comenzi de la nodul Master. Procesul Master oferă înregistrarea numelui și caută în Computation Graph și rutează datele în acest graf. Serverul Parameter permite ca datele să fie stocate într-o locație centrală și implementat în interiorul nodului Master. Mesajele sunt o structură de date compuse dintr-un set de primitive și vectori de tipuri primitive (integer, floating point, boolean, char). Mesajele sunt dirijate printr-un sistem de transport bazat pe publicare / subscriere semantică. Un nod transmite un mesaj prin publicarea acestuia la un anumit topic. În cazul în care un nod este interesat de mesajele legate de anumit topic, trebuie să se aboneze la acest topic. Mai multe noduri pot publica într-un anumit topic, în timp ce mulți alții se pot abona la unul, permițând astfel crearea multor arhitecturi many-to-many. ROS oferă de asemenea o interacțiune cerere-răspuns cu serviciile. Un nod poate oferi un serviciu sub un nume și un nod client poate utiliza serviciul prin trimiterea unui mesaj de solicitare și așteptând un răspuns. Bags sunt utilizați pentru salvarea și redarea mesajelor de date.

¹⁶

<http://wiki.ros.org/Metrics>

3.2. Sistemul de operare NAOqi

NAOqi este software-ul care ruleaza pe roboții Aldebaran și îi controlează. NAOqi este dezvoltat de SoftBank Robotics și este destinat strict pentru roboții companiei Aldebaran. Roboții NAOqi, Pepper în cazul nostru, vine cu NAOqi preinstalat ca sistem de operare principal pe care programatorii îl pot accesa și de la distanță. Framework-ul rulează peste o distribuție GNU / Linux bazată pe Gentoo. Este o distribuție GNU / Linux embedded, dezvoltată special pentru a se potrivi nevoilor robotului Aldebaran.

NAOqi oferă un nivel înalt de abstractizare a senzorilor și a elementelor de acționare ale robotului utilizate pentru programarea de aplicații. Framework-ul răspunde nevoilor comune de robotică, cum ar fi paralelism, resurse, sincronizare, evenimente. Acest sistem de operare permite comunicarea omogenă între module diferite (de mișcare, audio, video), programare omogenă și partajarea de informații omogenă.

Dezvoltatorii pot folosi acest framework pe Windows, Linux sau MacOS, scriind cod în mai multe limbaje de programare (e.g. Python, C++, Java), rulând fie nativ, fie utilizând un API de acces la distanță. O practică des întâlnită este de a dezvolta comportamente ale robotului în Python și de a scrie servicii de bază în C++.

Executabilul NAOqi care ruleaza pe robot este un broker. Atunci când pornește, se încarcă un fișier de preferințe care definește care bibliotecă ar trebui să se încarce, iar fiecare bibliotecă conține una sau mai multe module care utilizează brokerul să facă publicitate metodelor. În mod obișnuit fiecare modul este o clasă dintr-o bibliotecă. Când este încărcată biblioteca aceasta automat va reprezenta clasa modulului. Fiecare modul conține diferite metode. Modulele pot fi fie la distanță sau locale și interacționează cu alte module prin conectarea brokerilor lor la alte module de brokeri folosind un proxy.

3.3. Comparație

Ambele sisteme de operare NAOqi ROS permit un control fin asupra senzorilor și elementelor de acționare ale roboților. Ele oferă în continuare multe nivele înalte de abstractizare care să permită programarea aplicațiilor (de exemplu navigarea într-o linie dreaptă de 2 m, numărarea numărului de fețe umane pe care le vede robotul). Mulțimea de caracteristici de nivel înalt care sunt utile pentru aplicațiile din proiectul vor fi enumerate în secțiunile care urmează.

Ambele sisteme de operare, NAOqi și ROS lucrează prin construirea unor module de control asupra diferiților senzori și elemente de acționare. Modulele sunt capabile să comunice unele cu altele prin intermediul unor evenimente și mesaje. În ambele cazuri, în funcție de capacitățile robotului, aplicațiile dezvoltate folosind cele două sisteme pot rula fie pe robot fie la distanță, pe o mașină de control care se poate conecta printr-o rețea locală la robot.

Astfel, diferența dintre cele două sisteme constă în esență în cea mai mare parte în filozofia de proiectare și în suportului comunității de dezvoltatori (mai degrabă decât în diferențele privind capacitățile de procesare).

Cadrul ROS este un efort colaborativ, în care diferite laboratoare de robotică publică rezultate pe care le-au obținut în activitatea de cercetare. Astfel, laboratoarele care au experiență în vedere artificială pot completa activitatea celor care au experiență în navigarea și controlul robotului. De exemplu, există peste 200 de pachete software disponibile pentru a fi utilizate gratuit pentru distribuția ROS utilizată în cadrul proiectului (ROS Indigo).

Ceea ce este esențial este faptul că odată ce este creat un pachet ROS, acesta va rula la fel pe toate platformele care sunt compatibile ROS (adică au configurat fișierele și au formatat mesajele pentru a avea acces la diverși senzori și elemente de acționare în conformitate cu specificațiile ROS).

În timp ce NAOqi se bucură de asemenea de o comunitate de dezvoltatori, aceasta este mai puțin populară decât cea a ROS. Mai mult, aplicațiile NAOqi pot rula numai pe roboți produși de Aldebaran și Softbank: Nao, Pepper și Romeo.

Este interesant și demn de remarcat faptul că există în prezent un pachet wrapper¹⁷ ROS peste NAOqi SDK. Prin urmare, în timp ce Pepper nu este compatibil ROS utilizarea acestui wrapper permite robotului rularea de aplicații bazate pe ROS.

În introducere am menționat faptul că unul dintre obiectivele proiectului este implementarea unui set de comportamente de bază ale robotului, cu accent pe transferabilitatea lor (comportamentul funcționează în același mod pe diferite platforme robotice). Această cerință, impune mai mult sau mai puțin utilizarea unui middleware care să poată oferi abstractizari puternice pentru dezvoltarea de aplicații.

Din acest punct de vedere, devine evident de ce se ia în considerare capacitățile ROS, ca primă alegere atunci când dorim implementarea comportamentelor de bază.

4. Caracteristici de vedere artificială

Modulul de viziune computerizată va fi responsabil pentru a automatiza sarcinile pe care sistemul vizual uman le poate face și care vor servi scopurilor acestui proiect.

Modulul va trebui să analizeze datele legate de vedere, cum ar fi flux video provenit de la camere 3D, și să fie capabil să detecteze și să recunoască oamenii și obiectele din câmpul vizual al robotului.

Pornind de la aceste analize, modulul trebuie să gestioneze un proces de învățare, prin care să rețină informații pe termen scurt / lung și prin care să poată transpune obiectele și persoanele detectate în sistemul de coordonate al robotului. În acest capitol sunt prezentate pachetele disponibile în prezent, care se pot ocupa de prelucrarea datelor de vedere artificială ale robotului, organizate conform comportamentelor de bază considerate în proiect.

4.1. Sistemul de operare ROS

Acest capitol descrie pe scurt diferite pachete care sunt integrate în ROS și sunt legate de obiectivele computer vision. De asemenea, vom descrie aici software-ul care utilizează senzori 3D pentru a rezolva diferite probleme. Luând în considerare faptul că ROS este un proiect open source menit să sprijine mai mulți roboți și plăci, există o mulțime de software util open-source legat de probleme de computer vision ce pot fi integrate

Având în vedere obiectivele proiectului am listat pachetele care pot fi utile pentru rezolvarea problemelor de viziune:

- Detectare persoane
 - Stiva "People": Stiva People deține algoritmi pentru a detecta oameni prin mai mulți senzori.
 - Pachetul "People_tracking_filter": O colecție de instrumente de filtrare pentru urmărirea diferitelor locații ale oamenilor.
 - Pachetul "People_velocity_tracker": Urmărește ieșirea leg_detector pentru a indica viteza persoanei.
 - Pachetul "Leg_detector": Leg Detector utilizează o abordare de învățare tip mașină pentru a găsi modele tip picior cum ar fi modele de cititoare scanar cu laser.
 - "Spencer_people_tracking" open-source: Multi-modal ROS - bazat pe detectarea oamenilor și grupurilor de oameni & cadrul de urmărire pentru roboți mobili dezvoltat în cadrul EU FP7 pentru proiectul SPENCER.
 - Aplicații OpenCV: Modele opencv antrenate folosind clasificatori de tip Haar Cascade pentru detecția persoanelor (corp complet, partea superioară și inferioară a corpului)

¹⁷

<http://wiki.ros.org/pepper/>

- Detectarea feței și recunoașterea ei
 - Pachetul “Face_detector” : Detectarea feței din imagini.
 - Pachetul “Face_recognition”: Un pachet ROS pentru recunoașterea feței in flux video. Recunoașterea feței este realizată cu ajutorul Eigenfaces (denumită de asemenea "Principal Component Analysis" sau PCA) folosind sursa cod C++ dezvoltată de Shervin Emami (<http://www.shervinemami.info/faceRecognition.html>).
 - Pachetul “Face_detection_tracking”: Aceste noduri ROS sunt făcute să detecteze și să urmărească fețele din diferite poze și provin dintr-un subiect de imagine ROS.
 - Aplicații Opencv: Detectează fețe in utilizând clasificatori de tip cascadă pentru analiza imaginilor preluate prin intermediu ROS sensor_msgs/Image
- Detectarea obiectului:
 - Pachetul “Find_object_2d”: Simpla interfață Qt pentru a implementa OpenCV a SIFT, SURF, FAST, BRIEF și alte detectoare de funcții și descriptori. Cu ajutorul unei camere web, obiectele pot fi detectate și publicate pe un subiect ROS cu ID-ul și poziția ei (pixeli în imagine). Acest pachet este o integrare ROS a aplicației Găsește-Obiectul.
 - Pachetul “Object_recognition”: Pachetele cu libs și nodurile ROS asigura recunoașterea obiectului bazat pe transformarea Hough a grupării SURF.
- Marker detectare vizuală:
 - Pachetul “Ar_pose”: Sporirea estimării reale a markerului de poze folosind ARToolkit.
 - Pachetul “Ar_sys”: Estimare poze 3D pachetul ROS folosind panouri de marcare ArUco.
 - Pachetul “Ar_kinect”: Acest pachet extinde pachetul ar_pose să se ocupe de puncte cloud și imagini generate de Kinect pentru îmbunătățirea localizării markerului AR.

Pachetul general important ROS este folositor când folosește senzori ‘vision’ (camera 2D, senzori de adâncime):

- Pachete de mesaje Bridge. Acesta este un pachet util deoarece permite utilizarea tuturor software-lor specifice dezvoltate pentru un robot. De exemplu, puteți utiliza toate caracteristicile vision oferite de către robot Pepper.
- Pachetul Opencv. Acest modul permite accesul la o bibliotecă extinsă de soluții informatice vision.

4.2. Sistemul de operare NAOqi

NAOqi oferă unele module integrate de vedere artificială și funcții care pot fi utilizate în proiectul nostru. Având în vedere obiectivele proiectului vom enumera pachetele utile care pot fi utilizate pentru rezolvarea problemelor vision.

În primul rând, funcțiile de bază de care avem nevoie pentru proiect sunt cele pentru realizarea fotografiilor și înregistrarea. Acestea sunt **ALPhotoCapture**, **ALVideoDevice** and **ALVideoRecorder**.

Proiectul presupune capacități de percepție a unor oameni și NAOqi are unele funcții importante pentru percepția oamenilor:

- **ALEngagementZones** – clasifică persoanele detectate și / sau mișcările acestora folosind pozițiile lor în spațiu
- **ALFaceCharacteristics** - actualizări ale fiecărei persoane cu unele informații suplimentare, cum ar fi o estimare de vârstă și sex.
- **ALFaceDetection** – detectează și opțional clasifică fețele detectate

- **ALGazeAnalysis** – analizează direcția privirii unei persoane detectate pentru a ști dacă acea persoană se uită la robot
- **ALPeoplePerception** – menține lista persoanelor detectate în jurul robotului și furnizează informații stocate despre acestea
- **ALSittingPeopleDetection** – actualizează starea unei persoane detectate furnizând informație dacă acea persoană este așezată sau în picioare

O parte posibilă a modului de vedere vision este recunoașterea spațiului și a obiectelor. Funcțiile importante pe care le oferă NAOqi pentru detectarea obiectelor sunt:

- **ALBarcodeReader** – scanează o imagine căutând după un cod de bare
- **ALLandMarkDetection** – recunoaște diferite repere pre-învățate folosind diferite șabloane
- **ALMovementDetection** – detectează mișcare în câmpul vizual al robotului
- **ALRedBallDetection** – furnizează un modul de detecție și urmărire al unei mingi roșii, pe bază de informație vizuală
- **ALVisionRecognition** – modul ce încearcă recunoașterea de imagini, obiecte sau chiar locuri învățate anterior

Referitor la vedere artificială combinată cu navigația, NAOqi are 2 funcții care oferă unele informații spațiale. **ALVisualCompass** furnizează unghiuri de rotație ale robotului față de o imagine de referință, iar modulul **ALLocalization** permite localizarea robotului într-un mediu închis (in-door)

4.3. Comparație

În ceea ce privește comportamentele vision stabilite ca obiective ale acestui proiect, atât cadrele NAOqi și ROS conțin mai multe soluții pentru a gestiona toate obiectivele proiectului. Ambele au pachete disponibile pentru a face față analizei umane și obiectului vizual. Modulele care permit accesul direct la camera și sensorul de date 3D pot fi manipulate numai prin intermediul cadrului NAOqi și poate fi utilizat prin intermediul nodurilor ROS.

O diferență importantă este faptul că toate pachetele NAOqi sunt tip sursă-închisă și pot fi depășite din punct de vedere al performanței. Prin comparație, ROS este un cadru tip sursă-deschisă care are avantajul de a avea acces la tehnici mai diversificate și publice, precum și capacitatea de a personaliza punerea în aplicare (de exemplu Intrare / ieșire / evenimente ridicate), cu respectarea obiectivelor proiectului.

Prin urmare, eforturile viitoare se vor concentra asupra analizei aprofundate a pachetelor vision ROS identificate.

5. Caracteristici de navigare

Modulul de navigare preia datele de intrare de la laser, adâncimea și distanța senzorilor, în scopul de a extrage caracteristici vizuale și de a construi o reprezentare a mediului în care robotul trebuie să se localizeze cu succes. Mai mult decât atât robotul trebuie să identifice în mod corect componentele dinamice, cum ar fi oameni sau uși, de care el trebuie să țină cont pentru a fi în măsură să se deplaseze între diferite puncte de referință.

5.1. Sistemul de operare ROS

În ceea ce privește obiectivele proiectului din punct de vedere al navigației, cadrul ROS oferă diverse pachete și stive utilizate pentru navigație:

- Cel mai important și folosit de obicei este pachetul gmapping pentru SLAM (localizare simultană și cartografiere), care utilizează mesaje de la senzorul laser pentru a crea o hartă de ocupare stil grilă 2D.

Pachetul ROS numit gmapping este un wrapper peste OpenSlam Gmapping¹⁸. Pentru a-l utiliza este nevoie ca robotul să dispună de un senzor LIDAR. Folosind datele de la LIDAR și cele de odometrie, precum și transformatele între poziția senzorului LIDAR și ale bazei robotului, acesta poate furniza o hartă 2D a zonei mapate. Datele obținute prin acest pachet conțin referiri la lățime, înălțime, rezoluția hărții (metri / celulă), punctul de origine al hărții create și o matrice cu actualizări continue ale hărții. Figura



Fig. 4. Pachetul ROS gmapping pe Pepper. Harta scoasă de pachet, vizualizată în rviz, este afișată în partea stângă, corespunzând unui mediu real de laborator vizibil în partea dreaptă a figurii.

- **Google Cartographer**¹⁹ este un modul ROS open source lansat recent de Google, care oferă servicii simultane de localizare în timp real și cartografiere (SLAM), în 2D și 3D pe mai multe platforme și configurații de senzori. Se pot folosi scanere laser rotative ce produc mesaje de tip LaserScanner sau senzorii 3D ce produc mesaje PointCloud2. Pachetul furnizează ca ieșiri o hartă de ocupare și o listă de sub-hărți.

5.2. Sistemul de operare NAOqi

Proiectul presupune deplasarea robotului și pentru că NAOqi oferă unele funcții pentru a se deplasa și pentru animație. Cu toate acestea, NAOqi nu are o funcție pentru navigație autonomă, cum ar fi SLAM.

Modulele care permit deplasarea sunt:

- **ALNavigation** – permite controlul programatic al robotului, asigurând mișcări sigure (e.g. evită ciocniri în mod automat)
- **ALTracker** – permite robotului să urmărească diferite ținte (o minge roșie, o față, un reper, etc)
- **ALMotion** – conține diverse metode prin care se pot face mișcări ale robotului (atât de deplasare, cât și de postură)

Mai mult decât atât, NAOqi permite o coregrafie de gesturi mai complexă folosind modulele de animație ale robotului. Modulul **ALAnimationPlayer** permite rularea de animații, iar **ALRobotPosture** folosește la încărcarea de posturi predefinite.

¹⁸ <http://wiki.ros.org/gmapping>

¹⁹ <https://opensource.googleblog.com/2016/10/introducing-cartographer.html>

5.3. Comparație

În ceea ce privește navigația robotului, cadrul ROS conține pachete care sunt direct capabile să se ocupe de sarcini, cum ar fi cele descrise în obiectivele proiectului (de exemplu: cartografierea și navigația pe o hartă), capacități care îi lipsesc lui NAOqi. Mai mult decât atât, chiar dacă NAOqi conține module care permit controlul direct al poziției, mișcării, posturii și gesturilor robotului, acestea pot fi ușor accesate și utilizate prin intermediul nodurilor ROS, care pot cuprinde funcționalitatea nodurilor NAOqi.

În ceea ce privește obiectivul de a fi capabil de a naviga într-un mediu dinamic, evitând obstacolele aflate în mișcare și capacitatea de a urmări oameni în mișcare, ar putea fi necesar să se efectueze o dezvoltare suplimentară. Pentru acest lucru este nevoie de o integrare cu modulul computer vision, indiferent de cadrul de bază.

6. Caracteristici de interacțiune

Caracteristicile de interacțiune se referă la capacitatea platformelor robotizate luate în considerare în cadrul proiectului de a întâmpina și de a comunica cu utilizatorul. Acest raport se referă la o prezentare generală a ambelor mijloace de interacțiune: cea fizică (e.g. butoane, senzori tactili, capabilități de semnalizare vizuală ale robotului) și cea bazată pe comunicare în limbaj natural.

6.1. Capacități de interacțiune fizică

Această secțiune se referă la interacțiunea fizică ceea ce înseamnă că este disponibilă pentru fiecare tip de robot.

6.1.1. Robotul Tiago

Capacitățile fizice de interacțiune ale lui Tiago sunt limitate deoarece acesta nu are nici un buton sau ecran tactil prin care să interacționeze direct cu utilizatorul. Cu toate acestea Tiago are un Joystick de teleoperare ce permite utilizatorilor să controleze baza sa, trunchiul, capul și extremitățile.



Fig. 5. Joystick-ul de teleoperare pentru Tiago

6.1.2. Robotul Baxter

Capacitățile fizice de interacțiune ale lui Baxter se axează în cea mai mare parte pe brațele sale. Cele 7 articulații (3 pentru îndoire și 4 pentru torsiune) permit un grad mare de libertate al membrelor robotului Baxter, astfel încât acesta să poată avea un comportament complex de manipulare.

Cleștele de prindere de la capătul brațului permite robotului să manipuleze obiecte, fie apucându-le sau împingându-le pe o suprafață.

Un utilizator poate folosi în continuare unul din cele 4 butoane manșetă prezentate în figura 6 pentru a-i schimba starea într-o aplicație dată. Baxter are 4 astfel de grupuri pe butoane poziționate pe manșetele fiecărui braț și pe spatele său, chiar sub articulația umărului.



Fig 6. Elemente ale unui buton manșetă pentru robotul Baxter.

În afara de interacțiuni bazate pe atingere, Baxter poate comunica cu utilizatorul cu ajutorul semnalelor vizuale. Afișajul de pe capul robotului Baxter este un ecran LCD 1024 x 600 SVGA capabil să se rotească la 180°. Aceasta permite robotului să afișeze informații specifice. Capul lui Baxter are o articulație de rotire ce îi permite o acțiune de „încuviințare”, fiind astfel în măsură să confirme o acțiune sau o cerere a utilizatorului. Pe capul robotului există, de asemenea, un inel cu LED-uri roșii, un inel cu LED-uri verde și 12 LED-uri galbene individuale. Aceste inele și LED-uri pot fi controlate independent, permițând astfel unui potențial programator să semnaleze diferite stări ale aplicațiilor interne (de exemplu o confirmare vizuală, avertizare, eroare) pentru utilizator, în funcție de setările de culoare selectate.

6.1.3. Robotul Pepper

Din punct de vedere al interacțiunii fizice, Pepper poate fi operat prin intermediul a 2 butoane, unul în spatele tabletei și unul între umeri, în spatele gâtului.

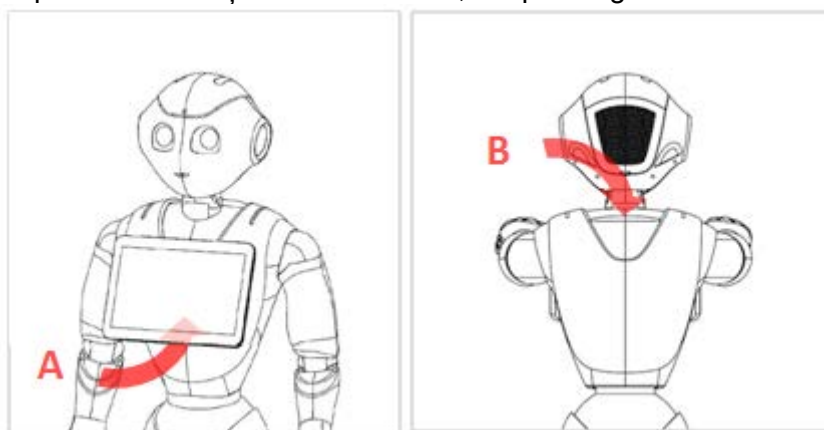


Fig. 7. Poziția butoanelor robotului Pepper.

Pepper reacționează de asemenea atunci când oamenii îi ating capul sau mâinile, declanșând evenimente ce pot fi interpretate corespunzător de programator pentru a declanșa comportamentul dorit.

Mai mult decât atât, Pepper dă unele feedback-uri utilizatorului prin intermediul culorii ochilor și urechilor sale. Diferite culori înseamnă diferite notificări. O culoare roșie a ochilor și urechilor înseamnă că a apărut o eroare sau una dintre funcțiile lui Pepper a devenit inutilizabilă, sau că operațiunea în curs de desfășurare a eșuat. În cazul în care culoarea este verde, înseamnă că operațiunea curentă a reușit și că Pepper a revenit la starea inițială. Culoarea galbenă anunță că a apărut o problemă care necesită atenție și / sau acțiune. Acest lucru înseamnă că Pepper este încă utilizabil, dar poate rula fie într-un mod de avarie în care unele caracteristici pot să nu fie disponibile sau poate deveni în curând inutilizabil dacă nu se rezolvă problema. Culoarea albastră a ochilor și urechilor înseamnă că Pepper ascultă și așteaptă intrarea audio de la utilizator.

O altă modalitate prin care Pepper poate interacționa cu un utilizator este prin intermediul tabletei sale. Se pot afișa diferite tipuri de informații sau poate fi utilizat pentru datele introduse de utilizator. Domeniul său de aplicare este de a extinde capacitățile lui Pepper având un dispozitiv suplimentar de interacțiune umană.

Tableta are un sistem de operare Android și se pot dezvolta aplicații Android care pot fi independente sau corelate cu acțiunile robotului. Prin folosirea ei, se pot arăta utilizatorilor tot felul de informații relevante, cum ar fi harta clădirii, informații despre un produs, descrierea unui artefact, grafice, texte etc.

Una dintre cele mai importante utilizări ale acestei tablete ar fi interacțiunea cu persoanele cu deficiențe de vorbire sau de auz. Se poate afișa pe ea textul pe care Pepper îl spune simultan, iar după aceea se poate utiliza o interfață grafică pentru input din partea utilizatorului.

6.2. Capacități de interacțiune în limbaj natural

În timp ce platformele robotizate prezentate permit un număr semnificativ de interacțiuni fizice și semnale vizuale, principalul mijloc dezirabil de comunicare om - robot rămâne cel prin intermediul limbajului natural.

Secțiunile care urmează raportează sprijinul disponibil pentru crearea de metode de comandă și control pe bază de voce, precum și de implicare în mai multe dialoguri, cu ajutorul framework-urilor menționate: ROS și NAOqi.

6.2.1. Sistemul de operare ROS

Această secțiune descrie pe scurt diferite pachete care sunt integrate în ROS și sunt legate de obiectivele de comunicare: Capacitatea robotului de a se angaja cu utilizatorul în comenzi simple.

● Abilitatea de a avea interacțiuni simple de comandă și control

- **Pocketsphinx**²⁰: Pocketsphinx este un simplu wrapper peste motorul de recunoaștere a vorbirii cu același nume, care folosește gstreamer și o interfață în Python. Gstreamer este folosit pentru a sparge inputul audio în propoziții distincte, spre a fi recunoscute. Pentru sinteza de voce poate fi combinat cu pachetul ROS **Sound_play**²¹.
- **Rospeex**²²: este un pachet ROS de comunicare multilingvă, ce oferă suport pentru recunoașterea vorbirii și sinteză de voce în engleză, japoneză, coreană și chineză.
- **Cyborg's Speech Recognition**²³: Modulul primește intrare de la un microfon și efectuează convertirea vorbirii în text utilizând una din următoarele două platforme: API-ul Google²⁴ și platforma Wit.ai²⁵.
- **NTNU Cyborg Communicational Abilities**²⁶: Modulul este responsabil de generarea unui răspuns și a unui mod "comportamental", dat fiind un input generat de modulul de recunoaștere a vorbirii k. Cyborg's Speech Recognition Solution. Modul comportamental (e.g. bucuros, supărat, uimit) este însoțit de un text dat ca răspuns la propoziția utilizatorului și este convertit în vorbire folosind pachetul ROS "sound_play".

²⁰ <http://wiki.ros.org/pocketsphinx>

²¹ http://wiki.ros.org/sound_play

²² <http://rospeex.org/top/>

²³ <https://www.ntnu.no/wiki/display/cyborg/k.+Cyborg's+Speech+Recognition+Solution>

²⁴ <https://cloud.google.com/speech/>

²⁵ <https://wit.ai/>

²⁶ <https://www.ntnu.no/wiki/display/cyborg/j.+Giving+NTNU+Cyborg+Communicational+Abilities>

- **Abilitatea de a modela dialoguri generale sau specifice unui anumit domeniu**
 - **Iwaki Interaction Manager**²⁷: Iwaki Interaction Manager este un manager de dialog open-source care nu se limitează doar la dialog. Acesta este strâns legat de teoria Sharedplan a discursului și punerea în aplicare a acestuia. Managerul de interacțiune Iwaki a fost inițial dezvoltat pentru a controla interacțiunea dintre roboții sociali la Institutul de Robotică de la Universitatea Carnegie Mellon.

6.2.2. Sistemul de operare NAOqi

SDK-ul NAOqi vine cu un număr mare de module, care facilitează capacitățile de limbaj natural, de la detectarea simplă a sunetului până la capacitatea de a înțelege și de a răspunde la enunțuri ale utilizatorilor. Prezentăm caracteristicile de procesare ale limbajului natural (NLP) disponibile în NAOqi cu referire la capacitățile de comunicare prezentate în capitolul 1.

În ceea ce privește înțelegerea de bază a vorbirii, NAOqi oferă un set de module ce pot fi utilizate pentru implementarea de comenzi simple sau secvențe de interacțiune de tip întrebare – răspuns. Modulele dau robotului capacitatea de a recunoaște cuvinte sau fraze predefinite în mai multe limbi (**ALSpeechRecognition**), precum și capacitatea de a vorbi, chiar și într-un mod expresiv (**ALTextToSpeech**, **ALAnimatedSpeech**).

În ceea ce privește capacitățile de conversație la nivel superior (dialog structurat), SDK-ul NAOqi pune la dispoziție modulul **ALDialog**. Modulul folosește o listă de reguli scrise, în scopul de a gestiona fluxul conversației dintre om și robot. Aceste reguli sunt de două tipuri: reguli de utilizator, care leagă un anumit input de la utilizator la o ieșire posibilă dată de robot, și reguli de propunere care declanșează o ieșire specifică a robotului, fără nici o intrare de la utilizator în prealabil.

În afară de modulele care se adresează obiectivelor comportamentale specificate în introducere, SDK-ul NAOqi conține pachete care pot îmbunătăți experiența utilizatorului în interacțiunea pe bază de voce cu robotul.

Pachetele sunt utile pentru a detecta sunete semnificative și direcția din care provin (**ALSoundDetection**, **ALSoundLocalization**), pentru a face robotul să stabilească și să păstreze contactul vizual cu oamenii (**ALBasicAwareness**), pentru a identifica emoția exprimată prin vocea vorbitorului (**ALVoiceEmotionAnalysis**), sau pentru a permite mișcări autonome atunci când robotul ascultă sau vorbește (**ALListeningMovement**, **ALSpeakingMovement**).

6.3. Comparație

Acest capitol a investigat capacitățile de interacțiune ale roboților utilizați în cadrul proiectului, în ceea ce privește modul în care utilizatorii pot intra în contact fizic cu roboții, cât și folosirea limbajului natural.

În ceea ce privește interacțiunea fizică, robotul Pepper iese în avantaj prin design, din moment ce acesta este conceput ca un robot de companie. Deși se află în imposibilitatea de a efectua orice sarcini de manipulare (spre deosebire de Baxter sau Tiago), robotul poate furniza informații utilizatorului prin intermediul tabletei sale, precum și prin diferite setări de culori pentru LED-urile situate în ochi, urechi și umeri.

Modulele de interacțiune în limbaj natural sunt mai numeroase și mai variate în cadrul ROS, după cum era de așteptat, deoarece acestea sunt contribuții ale diferitelor laboratoare de cercetare.

Pentru interacțiunile prin comandă simplă sau prin mecanism de întrebare-răspuns, lista de pachete enumerate la punctul 6.2.1 oferă suficientă flexibilitate pentru punerea în aplicare a acestora.

Avantajul principal (de exemplu dacă s-ar utiliza pachete ROS, cum ar fi Pocketsphinx sau Rospeex) este potențialul de a antrena motoarele de recunoaștere a vorbirii, astfel încât să se recunoască comenzi în mai multe limbi, prin crearea de modele lingvistice explicite.

²⁷ <http://wiki.ros.org/iwaki>

SDK-ul lui NAOqi vine cu un mecanism de recunoaștere incorporat pentru cuvinte și fraze și este mai ușor de utilizat. Cu toate acestea, SDK-ul permite doar două modele de limbi active la un moment dat, dintre care unul este în mod necesar cel pentru limba engleză.

Un dezavantaj principal al SDK-ului NAOqi este că nu permite accesul la analiza cuvântului sau a expresiei care a fost spusă de către utilizator. Modulul ALSpeechRecognition permite unui dezvoltator să creeze un vocabular de cuvinte și expresii pe care robotul le poate recunoaște. Cu toate acestea, modulul nu va returna alte cuvinte sau fraze în afara acestui vocabular, chiar dacă motorul intern de recunoaștere a vocii le-ar putea detecta de fapt. Acest lucru înseamnă că un dezvoltator este limitat la capacitățile SDK-ului NAOqi, fără posibilitatea de a integra cu ușurință alte soluții mai complexe de procesare a limbajului natural.

În ceea ce privește managementul de dialog, modulul ALDialog din NAOqi oferă o soluție completă de scriere a unui dialog. Deși limitată la limbile active pe un robot compatibil NAOqi (limba engleză și a doua limbă), modulul este suficient de capabil pentru a fi considerat un candidat la soluția inițială de management de dialog în cadrul proiectului.

În plus, așteptarea este că sistemul de evenimente care stă la baza NAOqi, și care poate fi declanșat de modulul ALDialog, va permite integrarea cu alte module bazate pe ROS (e.g. pentru planificare sau navigație, care devin ascultători pentru evenimentul generat de modulul de dialog).

Pachetul managerului de interacțiune Iwaki și ROS vor fi investigate mai atent numai după efectuarea mai multor teste cuprinzătoare cu modulul ALDialog din NAOqi.

7. Concluzii

Prezentul raport oferă o privire de ansamblu introductivă și comparativă a platformelor robotice disponibile în proiect și gradul de adecvare al acestora la scenariile și obiectivele identificate.

Prezentarea și analiza acoperă atât partea hardware cât și cea software (modulele și pachetele existente) caracteristice celor trei roboți: Tiago, Baxter și Pepper. Scenariile roboticii de asistare prezentate în capitolul 1 sunt exprimate printr-un set de comportamente de bază care trebuie să fie compuse în mod coerent prin intermediul unui modul de planificare. La rândul său, setul de comportamente de bază dictează tipul capabilităților necesare, văzute dintr-o serie de perspective funcționale: navigare, vedere artificială și comunicare cu utilizatorul.

Fiecare capitol din acest raport a analizat o astfel de perspectivă în detaliu, a prezentat soluții hardware sau software existente și a sugerat linii suplimentare de investigare prin intermediul secțiunii corespunzătoare de comparație. Concluzia acestui raport rezumă rezultatele comparațiilor individuale și scoate în evidență următorii pași pentru fiecare direcție de implementare și testare.

Din punct de vedere al hardware-ului, analiza sugerează că roboții Pepper și Tiago sunt mai potriviți pentru obiectivele finale ale proiectului. Dimensiunile lor, capacitățile de mișcare și de detectare îi fac mai atrăgători pentru punerea în aplicare a setului de comportamente impuse.

Comparația dintre sistemele de operare ROS și NAOqi sugerează ca ROS oferă o mai mare flexibilitate pentru dezvoltare. Prin urmare, decizia este de a acorda prioritate implementării comportamentelor de bază pornind de la modulele și pachetele existente în ROS. Mai mult, dezvoltarea se va concentra de asemenea pe implementarea nodurilor ROS care înglobează funcționalitatea relevantă și utilă a modulelor din NAOqi.

Luând fiecare perspectivă de dezvoltare la rând concluziile sunt după cum urmează:

- Vedere artificială:
 - Concentrarea atenției asupra dezvoltării bazate pe pachetele ROS. Prezentarea generală a pachetelor prezintă un potențial de acoperire a tuturor obiectivelor proiectului legate de aspectele de vedere artificială a robotului. Mai mult decât

atât, natura pachetelor de vedere din ROS permite lucrul cu mai multe tehnici actuale, precum și o mai mare flexibilitate în particularizarea implementării.

- Navigare:
 - Utilizarea pachetelor ROS GMapping sau Google Cartographer pentru a pune în aplicare cartografierea mediului și planificarea traiectoriei
 - Combină pachetele ROS de navigație și de vedere artificială pentru a implementa funcționalitatea de evitare și de urmărire a utilizatorilor umani
 - Implementarea de noduri ROS peste modulele NAOqi care permit setarea posturii robotului și controlul animației (exemplu ALAnimationPlayer, ALRobotPosture)
- Interacțiune:
 - Semnalizările fizice (e.g. luminoase) vor fi utilizate de la caz la caz. Un modul ROS de planificare va fi responsabil de tratarea evenimentelor de apăsare a butoanelor sau cele de atingere ca declanșatoare pentru un comportament de bază (de exemplu începe o interacțiune bazată pe voce)
 - Pentru comenzi simple și interacțiuni întrebare-răspuns pachetele ROS identificate la punctul 6.2.1 vor avea prioritate, având în vedere flexibilitatea în manipulare a mai multor limbi și accesul de granularitate mai fină la textul recunoscut.
 - Pentru managementul de dialog, capacitățile modulului ALDialog din NAOqi vor fi testate pentru robotul Pepper. Se ia în considerare crearea unei încapsulări ca nod ROS a funcționalității din ALDialog, în cazul în care capacitățile de scriptare a dialogului sunt satisfăcătoare. Pentru robotul Tiago și Baxter, care rulează nativ ROS, infrastructuri alternative de management de dialog vor fi căutate, pornind de la o examinare mai atentă a managerului de interacțiune Iwaki²⁸.

Pașii de mai sus au stabilit foaia de parcurs pentru designul arhitecturii particularizate a sistemului care permite definirea de comportamente de bază și compoziția acestora. În consecință afirmăm faptul că actuala etapă de implementare a proiectului SPARC a îndeplinit integral obiectivele propuse.

Bibliografie

- [1] http://www.robotis.com/x/darwin_en
- [2] <http://www.smashingrobotics.com/aldebaran-unveils-pepper-personal-emotional-robot/>
- [3] <https://www.ald.softbankrobotics.com/en/cool-robots/romeo>
- [4] http://hubolab.kaist.ac.kr/p_hubo2p
- [5] <http://www.dongburobot.com/jsp/cms/view.jsp?code=100796>
- [6] <https://www.engineeredarts.co.uk/robothespian/>
- [7] <https://www.willowgarage.com/pages/pr2/overview>
- [8] <http://world.honda.com/ASIMO/>
- [9] <http://tiago.pal-robotics.com/>
- [10] <http://www.rethinkrobotics.com/baxter/>
- [11] <https://www.ald.softbankrobotics.com/en/cool-robots/nao>
- [12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. "ROS: an opensource Robot Operating System." În ICRA workshop on open source software, vol. 3, no. 3.2. 2009.
- [13] <https://svrobo.org/wp-content/uploads/2015/05/Service-Robotics-Case-Studies.pdf>

- [14] http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Our-Autonomous-Future-with-Service-Robots/content_id/4925
- [15] I. Okhtay and D. S. Nau. A General Approach to Synthesize Problem-Specific Planners. Technical Report CS-TR-4597, UMIACS-TR-2004-40, University of Maryland, October 2003.
- [16] J. Bohren, and S. Cousins. The SMACH High-Level Executive. In: IEEE Robotics Automation Magazine 17 (2010), Nr. 4, p. 18–20.
- [17] V. Pradeep, E. Marder-Eppstein. Actionlib Documentation. Online at <http://www.ros.org/wiki/actionlib>, July 2013
- [18] <https://eu-robotics.net/membership/list-of-members/index.html>
- [19] <http://dataspeedinc.com/assets/Dataspeed-BaxterMobilityBaserevBMedia.pdf>
- [20] <http://erric.eu/ambient-intelligence-lab>