UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Gestionale

"Studio dell'architettura di controllo di un robot umanoide"

"Study of the control architecture of a humanoid robot"

Relatore: Chiar.mo
Prof. **Andrea Monteriù**

Tesi di Laurea di:
Eugenio Vecchietti

Correlatore:
Dott.ssa **Sabrina Iarlori**

A.A. **2019/2020**

Indice

1 Introduzione

Il tema della robotica è stato affrontato sin dall'antichità.

Infatti, alcune tracce di automi e umanoidi artificiali si trovano nelle varie leggende della mitologia greca. Il dio Vulcano, ad esempio, si dice avesse forgiato per sé una nutrita schiera di servitori e compagni meccanici, mentre il re Minosse si era munito di un gigante di bronzo, meccanico e alato. Era un tema particolarmente sentito anche nel periodo di Aristotele; infatti, egli parla dei robot come della soluzione definitiva alla schiavitù umana, utilizzando automi per i lavori più degradanti e pesanti. Alcuni taccuini di Leonardo da Vinci risalenti al 1495 contengono progetti piuttosto dettagliati di cavalieri meccanici in grado di alzarsi, muovere le braccia e la testa. (fastweb DigitalMegzine, 2017)

L'evolversi dell'ingegneria, poi, ha portato alla creazione di grossi meccanismi, come grandi orologi e calcolatori meccanici. All'inizio del Novecento il drammaturgo cecoslovacco Karel Capek scrive un dramma chiamato R.U.R (Rossumovi univerzlni roboti). In quest'opera compare per la prima volta la parola robot (dal ceco robota, "lavoro duro, lavoro forzato"), il quale non era un vero e proprio robot, ma un essere costruito producendo artificialmente le varie parti del corpo successivamente da assemblare. Nel romanzo si parla di una società basata sul lavoro di robot semi-umani che piano piano si ribellano e sottomettono gli umani. Anni dopo Isaac Asimov, autore di numerosi romanzi e racconti di fantascienza, nonché di volumi di divulgazione scientifica, enuncia negli anni '50 le tre leggi della robotica, le quali hanno ispirato esperti del settore e dell'intelligenza artificiale e cibernetica:

- Prima legge: un robot non può recare danno a un essere umano, né può permettere che, a causa del suo mancato intervento, un essere umano riceva danno.

- Seconda legge: un robot deve obbedire agli ordini impartiti dagli esseri umani, a meno che questi ordini non contrastino la prima legge.
- Terza legge: un robot deve salvaguardare la propria esistenza, a meno che questa autodifesa non contrasti con la prima o la seconda legge.

Nel 1954 George Devol progetta il primo robot realmente programmabile, chiamato Unimate. Pochi anni dopo venne utilizzato nella catena di montaggio della General Motors e divenne il primo robot industriale ad entrare in funzione. Da lì in poi si assiste ad una vera e propria rivoluzione, una corsa globale per progettare i miglior robot; infatti, si assiste alla nascita di bracci meccanici, robot dotati di sistemi visivi, robot bipedi, ecc. (wikipedia, s.d.)

Arrivati agli anni duemila la robotica si presenta come una scienza affermata e in crescita esponenziale, con robot inviati nello spazio, robot casalinghi come Roomba (prima generazione di aspirapolveri robotiche), la comparsa delle prime automobili a guida automatica, ecc. Un enorme passo avanti che forse neanche l'uomo si aspettava.

Il tema della robotica ha sia lati negativi che lati positivi. Se già dal 900 c'era un grosso scetticismo per la comparsa di macchine che potessero sostituire la manodopera umana, ora molti esperti del settore sostengono che lutilizzo massiccio di robot nella catena produttiva finirà con l'innalzamento del livello di disoccupazione. Si verrebbe così a creare una contrapposizione tra lavoratori e robot, cosicché qualcuno – come Bill Gates – propone di tassare l'utilizzo dei robot in ambito lavorativo, così da creare un fondo da destinare a chi resta senza occupazione per colpa dei robot. (fastweb DigitalMegzine, 2017)

Ma i lati positivi sono molteplici: infatti, come tutti i sistemi di sicurezza che oggi giorno sono presenti nelle macchine o in altri veicoli, la precisione con cui i robot compiono i vari lavori con errori quasi tendenti allo zero, i sofisticati macchinari presenti dentro gli ospedali che consentono di salvare le vite umane, l'esplorazione del mare profondo che sarebbe impossibile senza l'aiuto di appositi strumenti, l'esplorazione spaziale, aiutano e migliorano la vita quotidiana umana ogni giorno.

La parola robot racchiude tanti significati come macchina automatica, automa, persona che agisce passivamente.

È bene iniziare facendo una grossa distinzione tra due classi di robot:

- Robot non autonomi: sono classici robot utilizzati per adempiere a specifici compiti che riescono a compiere in maniera più efficace rispetto all'uomo; un esempio sono i robot utilizzati nelle fabbriche, ottenendo una produzione più precisa, veloce e a costi più ridotti; o robot utilizzati per lavorare in ambienti ostili (per esempio nello spazio). Sono "non autonomi" poiché sono guidati da un software deterministico. Dentro questa classe rientrano anche i CO-BOTS robot collaborativi in grado di lavorare insieme all'uomo nella catena di montaggio;
- Robot autonomi: sono dotati di intelligenza artificiale, caratterizzati dal fatto che operano in totale autonomia e indipendenza dall'intervento umano e sono in grado di prendere decisioni. All'interno di questa classe possiamo trovare robot umanoidi, chiamati così proprio per la loro sembianza umana. (wikipedia, s.d.)

Una peculiarità di alcuni robot umanoidi è che possono essere socialmente iterativi e assistivi, cioè sono pensati per intrattenere le persone con la comunicazione; in più hanno anche lo scopo di dare un sostegno nel lavoro o nella vita quotidiana sotto forma di robot di assistenza o compagnia. A tale scopo devono interagire e comunicare con la gente e devono essere in grado di riconoscere emozioni e di simulare modi di interagire empatici. (Bart de Witte)

Alcuni esempi di robot umanoidi socio-assistivi sono:

- Humanoid è un robot della Toyota; è un partner robot, un umanoide da compagnia che in futuro potrebbe entrare nelle nostre case con il compito di darci una mano nelle faccende quotidiane o per assistere gli anziani.
- Robina anche essa è una partner robot, così la Toyota chiama i suoi robot domestici, per aiutare in famiglia, studiato proprio per aiutare gli anziani.
- Pepper è stato realizzato dalla Aldebaran Robotics per Softbank, che da giugno 2015 ha già venduto migliaia di esemplari in Giappone, ed è

considerato uno dei migliori robot da compagnia sul mercato. (robotiko, 2016)

Fra i robot assistivi, Pepper ricopre sicuramente un posto di rilievo. In effetti, Pepper è un social robot in grado di dialogare e gesticolare tranquillamente come una persona umana. I suoi campi di applicazione sono numerosi ed infatti si possono trovare all'interno di famiglie, di ospedali, di scuole, di aziende, ecc.

La versione di Pepper per aiutare le imprese si chiama "Pepper for Biz". Questa versione possiede una serie di software preinstallati che consentono di interagire con i clienti assolvendo a compiti quali accoglienza, intrattenimento, presentazione prodotti, e raccolta dati ai fini del marketing aziendale.

Le potenzialità di questo robot sono numerose, si potrebbe trovare un nuovo impiego ogni giorno. L'obiettivo di questa tesi è quello di studiare l'architettura del robot Pepper, partendo dallo stato dell'arte nel quale si trova il punto in cui sono arrivate le ricerche riguardo all'utilizzo di Pepper (Capitolo 2), dopodiché si analizzeranno le sue caratteristiche (Capitolo 3), per poi analizzare le sue componenti hardware (Capitolo 4) e software (Capitolo 5). Si passerà quindi alla trattazione di un modulo di navigazione e orientamento (Capitolo 6), a un metodo utilizzato dal robot per rispondere alle domande (Capitolo 7) e a un procedimento di riconoscimento facciale (Capitolo 8); lo studio verrà concluso attraverso alcune considerazioni finali (Capitolo 9).

2 *Stato dell'arte*

Un robot è un oggetto complesso caratterizzato da una stretta simbiosi tra hardware e software, capace di acquisire informazioni e di operare con autonomia nell'ambiente che lo circonda. La parte hardware è composta da componenti "fisiche", le quali sono: unità di controllo, sensori e attuatori. La parte software di un robot è quella immateriale cioè i programmi e le istruzioni che danno comandi all'hardware. A fornire al robot l'automatismo operativo necessario provvede l'informatica, attraverso un'azione di programmazione e controllo dei movimenti del robot. (La robotica, s.d.)

Con l'evoluzione della tecnologia robotica, i robot sociali costituiranno uno dei più importanti settori di espansione per la robotica. Sulla base di un accelerato avanzamento in questo settore multidisciplinare con un costante aumento del numero dei casi applicativi, è possibile sostenere che i robot rivestiranno un ruolo sempre più importante nella vita quotidiana e coesisteranno con le persone conducendole verso uno stile di vita più sicuro, più salutare, felice e sicuramente più smart. Il robot Pepper, sviluppato dalla SoftBank robotics è un robot creato con questa visione e con questa finalità. Non c'è dubbio che l'utilizzo dei robot stia diventando sempre più massivo: l'impiego di dispositivi robotici di varie forme e dimensioni sta riguardando aspetti diversi del quotidiano dal campo lavorativo a quello assistivo, con la loro presenza in supermercati, ospedali, musei, stazioni ferroviarie, case di cura, scuole e abitazioni private. In questo contesto dal vasto potenziale applicativo, i robot dovrebbero diffondersi e impattare nella vita delle persone in svariati modi, sviluppando in particolare un'attitudine alla socialità e raggiungendo un buon grado di accettazione da parte dell'uomo. Pertanto, la forma, le dimensioni, l'aspetto, il comportamento e l'intelligenza di questi robot sociali

devono essere customizzati e progettati tenendo in considerazione che andranno ad interagire e a lavorare in un ambiente dove al centro c'è la sicurezza dell'utente.

Questa è stata la stessa idea dietro lo sviluppo del robot Pepper, realizzato dalla SoftBank Robotics. Sebbene Pepper sia stato inizialmente progettato con un particolare obiettivo di applicazione business-to-business utilizzato nei SoftBank store, il robot è poi diventato una piattaforma di interesse per tutto il mondo per varie applicazioni che includono: il business verso il consumatore, il business verso le università e i centri di ricerca, il business verso gli sviluppatori e una varietà di altri casi d'uso. Per esempio, il robot Pepper è attualmente impiegato in migliaia di case e scuole ed è stato selezionato come piattaforma robotica per la RobotCup@Home, competizione per la Social Standard Platform League come riportato nell'articolo (Gelin, 2018).

Inoltre, i ricercatori hanno puntato su un aspetto del robot particolarmente antropomorfo e sulle modalità di interazione che costituiscono quelle particolari caratteristiche che la maggior parte degli utenti immaginano un robot da compagnia dovrebbe avere. In aggiunta, dai robot umanoidi, ci si aspetta che questi siano supportati nella loro comunicazione dal linguaggio del corpo e da altre abilità che rispondono a segnali sociali simili a quelli che gli uomini si scambiano tra loro, in modo da rendere i robot capaci di essere altamente impiegati in queste attività. Molti degli studi che la SoftBank Robotics ha condotto sul robot NAO, una versione più piccola di robot e allo stesso modo altamente diffusa, sono stati poi riportati ed ampliati sul Pepper al fine di definire alcuni principi guida da considerare per la sua progettazione:

- Una piacevole estetica: quando parliamo di caratteristiche estetiche, parliamo di dimensioni e aspetto esterno: Pepper è stato progettato per assomigliare ad un umano, pur essendo sempre un robot. Si può notare l'influenza estetica giapponese sulla faccia del robot e nei suoi movimenti. Ha dei grandi occhi, simili a quelli di un manga, e ha la possibilità di chinarsi all'incontro con qualcuno, come la tradizione giapponese vuole. Non è definito rispetto al genere.
- Sicurezza: è considerata in vari aspetti del corpo di Pepper. Ad esempio, il robot non ha spigoli vivi, ci sono parti morbide e il centro di massa cade all'interno

della base per evitare che il robot cada. I motori di Pepper sono abbastanza potenti per muovere le articolazioni ma non così forti da ferire qualcuno con un colpo accidentale. A livello meccanico e hardware, utilizza il controllo software per verificare il comportamento di ogni giunto e rilevare se viene applicata una forza esterna sul braccio.

- Economicità: per garantire l'accessibilità economica, sono stati aggiunti solo i componenti, i sensori e le funzionalità necessarie per soddisfare le esigenze dei casi d'uso specifici. Ad esempio, la mano non è stata deliberatamente progettata per una manipolazione pesante, ma solo per essere abbastanza buona per l'iterazione espressiva.
- Interattività: è una delle caratteristiche chiave del robot Pepper. La necessità di un'interazione naturale e intuitiva è fondamentale. Quindi, Pepper ha una multimodalità di interfacce di interazione. Ciò include uno schermo touch screen, in più testa e mani tattili e diodi a emissione di luce (LED). Sono stati sviluppati diversi componenti software per facilitare le capacità di percezione necessarie a garantire una semplice interazione con l'uomo, detta Human-Robot Interaction (HRI), inclusa la capacità di riconoscere e rispondere alle emozioni umane, una libreria di gesti espressivi e comportamenti per mostrare vivacità. Per ottenere un'espressività simile a quella umana e aggraziata attraverso il linguaggio del corpo, la struttura cinematica del robot è stata accuratamente progettata con 17 articolazioni. Le tre ruote omnidirezionali aiutano a ottenere movimenti fluidi e supportano la realizzazione di piccoli spostamenti locali in modi più naturali.
- Buona autonomia: è un altro requisito importante, il robot può servire per un'intera giornata lavorativa nei negozi SoftBank senza ricarica o intervento. Pertanto, l'intero sistema è stato progettato per bilanciare i carichi software e hardware e ottenere una durata della batteria fino a 12 ore. Inoltre, è stata sviluppata una docking station appositamente progettata per la ricarica autonoma. Inoltre, ci sono moduli e app per il robot per raggiungere l'autonomia comportamentale in particolari applicazioni, riducendo la necessità di intervento umano. (Gelin, 2018)

Diversi sono gli approcci e i lavori attualmente sviluppati da gruppi di ricerca che hanno come oggetto il robot Pepper e che forniscono differenti scenari per gli applicativi sviluppati e presentano le soluzioni implementate.

In (Simone Verrasi, 2018) gli autori parlano dell'utilizzo di Pepper come strumento di screening per un potenziale deterioramento cognitivo, un fattore di rischio per la demenza e altre malattie mentali. Gli obbiettivi specifici erano: (1) verificare la correlazione tra la versione robotica e quella cartacea del test e (2) valutare l'accuratezza del punteggio robotico. Tutto questo è possibile avendo implementato un test psicometrico sul robot sociale, realizzando una valutazione cognitiva tramite Human-Robot Interaction. Lo strumento robotico è stato progettato specificatamente per sfruttare l'HRI e le interfacce all'avanguardia del robot. Le funzioni cognitive devono essere valutate attraverso modalità specifiche, quindi si è fatto affidamento ai sensori e alle interfacce utilizzate dal robot. Per esempio, per valutare le abilità visuo-costruttive dei partecipanti al test, è stato richiesto di disegnare su un foglio e poi di mostrarlo al robot, il quale scatta una foto per la sua valutazione. All'interno della relazione si testa la fattibilità della procedura e la capacità di cogliere informazioni utili per la valutazione psicologica. I risultati suggeriscono che l'obiettivo è raggiungibile con la supervisione di un professionista, la tecnologia necessita di ulteriore lavoro e affinamento per una valutazione completamente autonoma.

I ricercatori Z. Shen, A. Elibolo e Y. Chong presentano un lavoro nel quale viene valutato l'utilizzo di un nuovo framework che consente al robot Pepper di estrarre facilmente le caratteristiche visive degli utenti come lo sguardo, il movimento della testa e il movimento del corpo, nonché le caratteristiche vocali come il tono e l'energia. Gli esperimenti sono stati sviluppati sulla base dell'idea che il robot sia un individuo durante l'iterazione, quindi i dati di iterazione sono estratti senza dispositivi esterni ad eccezione del robot stesso. Il robot Pepper poneva una serie di domane all'utente e nel frattempo registrava i comportamenti abituali del destinatario, mentre i tratti di personalità venivano valutati da un questionario. I risultati sperimentali hanno presentato una

performance promettente, soprattutto nel dedurre i tratti della personalità, utilizzando solamente semplici segnagli sociali (Zhihao Shen, 2019).

Uno studio fatto da Daisy van der Putte *et al.*, mette in risalto la possibilità di utilizzo di Pepper all'interno di ospedali, per aiutare gli infermieri nella raccolta dati dei pazienti. L'esperimento è stato svolto nel reparto di medicina dell'ospedale Franciscus Gasthius e Vlietland. 35 i pazienti di età media 64.1±17.7 che hanno partecipato allo studio. È stato utilizzato il robot per condurre cinque questionari su anamnesi, defecazione, dolore, memoria e sonno. I pazienti e gli infermieri hanno trovato il robot ragionevolmente accettabile in questo ruolo. (Daisy van der Putte, 2019)

All'interno dell'articolo realizzato da Amit Kumar Pandey e da Rodolphe Gelin si elenca in vari impieghi del robot. Pepper in Giappone accoglie i clienti nei negozi SoftBank, nei sushi bar, nei negozi di abbigliamento e nelle boutique Nespresso. In Europa, le esperienze con il robot sono state eseguite con successo, è stato utilizzato nelle stazioni ferroviarie francesi, nei supermercati Carrefour, presso strutture sanitarie e di assistenza agli anziani e sulle navi Costa Crociere. Negli stati uniti, Pepper può essere trovato, nel centro commerciale Wesrfield di San Francisco. Poiché la conoscenza dei robot diventerà sempre più necessaria nei sistemi scolastici, circa 2000 robot sono stati forniti agli istituti educativi in Giappone per supportare l'insegnamento della programmazione dei robot.

Pepper è il primo nel suo genere a creare una strada per una nuova generazione di robot personali e di servizio che sono anche prodotti in serie. Tuttavia, c'è ancora molta strada da fare. Tali macchine devono comportarsi in modo socialmente accettato, hanno bisogno di una percezione robusta in ambienti reali e la necessità di interagire dinamicamente con diversi tipi di utenti (Gelin, 2018).

3 L'oggetto

Abbiamo tra le mani Pepper: un robot umanoide, creato per comunicare, per diventare amico dell'uomo, nonché per fornirgli un valido aiuto. Sono tantissimi i campi in cui può essere utilizzato: se all'inizio era stato destinato dalla Softbank Robotics, la sua casa di produzione, solo per il BUISNESS TO BUISNESS, ora è diventato una piattaforma di interesse in tutto il mondo per varie altre applicazioni; per esempio nel BUISNESS TO CONSUMER, BUISNESS TO ACCADEMICS e BUISNESS TO DEVELOPER. (Gelin, 2018). Il simpatico robot trova impiego in migliaia di case e scuole. Per comprendere le sue capacità, si pensi che Pepper è in grado di parlare 15 lingue, è in grado di riconoscere lo stato d'animo dell'interlocutore, può riconoscere le persone memorizzando il tono della loro voce, ed è in grado di comunicare e interagire col corpo ballando e gesticolando, quasi come una persona normale. Di seguito vengono elencate alcune caratteristiche del robot:

- È accattivante, grazie all'altezza realistica di 120 cm, all'aspetto e al suo comportamento umanoide, Pepper si distingue immediatamente tra la folla, attirando facilmente la completa attenzione di tutti e riesce a creare un legame emotivo con le persone;
- Mette a proprio agio, grazie ai sensori installati su di esso rileva le persone presenti nell' ambiente circostante e attira la loro attenzione iniziando a interagire con loro tramite la voce o in modo proattivo, andando verso di loro ed iniziando una conversazione;
- È personalizzabile, in quanto piattaforma programmabile avanzata, Pepper offre infinite possibilità per arricchire esperienze degli utenti, il che gli consente di essere utilizzato in diversi settori come commercio al dettaglio, banche, luoghi di lavoro, amministrazione pubblica e servizi pubblici, sanità e istruzione;

- È gentile, sicuro e spregiudicato, Pepper riesce sempre a intrattenere con una piacevole conversazione per rendere l'esperienza memorabile, il che lo rende perfettamente integrabile con le persone;
- È connesso, usando diversi servizi cloud e database che espandono le sue capacità tecniche, come ad esempio il "cognitive computing", la comunicazione e la raccolta dati in tempo reale, Pepper è in grado di fornire molteplici servizi e di destreggiarsi tra diversi ambiti (Europe)

4 Hardware

Analizziamo la parte fisica e hardware di Pepper.

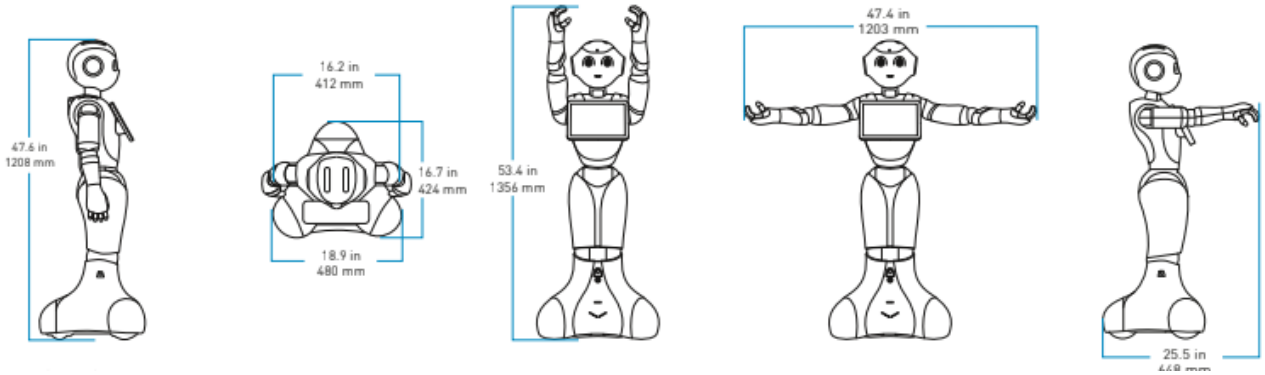


Fig. 1 – dimensioni del robot

Nella Fig. 1 sono mostrate le sue dimensioni fisiche, con un peso che è di circa 30 kg. Ha una batteria di 30 Ah, la quale permette una autonomia fino a 20 ore (minima 7 ore / media 12 ore / massima 20 ore), con la possibilità di farlo tornare alla stazione di ricarica quando è scarico. La macchina ha un processore Atom E3845 con un quad-core e una velocità di clock di 1.91 GHz. Ha 4 GB\s come velocità di trasmissione dati e una memoria flash da 32 GB, di cui 24 GB sono disponibili per gli utenti. Il robot ha 20 gradi di libertà che consentono il movimento di tutto il corpo (17 articolazioni): due in testa, due in ciascuna spalla, due in ogni gomito, una in ciascun polso, una per mano (cinque dita), due sui fianchi, una al ginocchio e tre alla base (Fig.2). Ha una navigazione omnidirezionale grazie alle tre ruote; queste consentono al robot anche di salire un gradino di 1,5 cm e di sopportare una pendenza di 5 gradi.

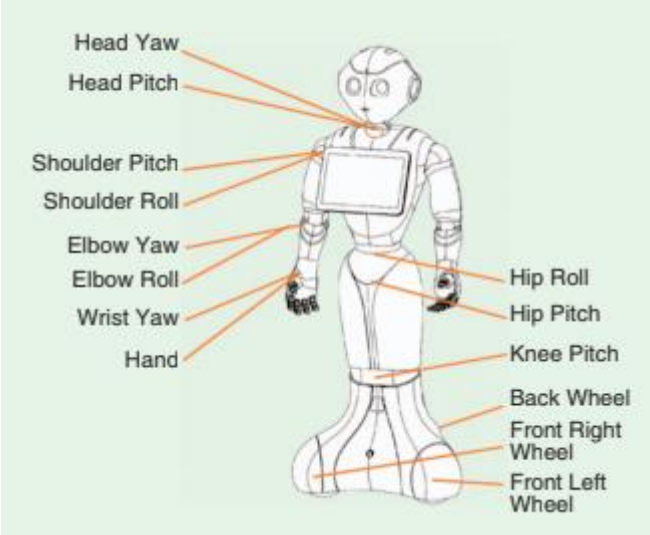


Fig. 2 – raffigurazione delle articolazioni

Pepper è fornito di una unità di misura inerziale a sei assi (IMU); è composta da un giroscopio con una velocità angolare di 500 °/s e un accelerometro a tre assi con un'accelerazione di 2g. I dati consentono una stima della velocità di base e dell'assetto (imbardata, beccheggio e rollio). Sono utilizzati motori brushless DC (corrente continua) negli arti inferiori e superiori, con sensori di posizione nei giunti a magnete, basati su encoder rotativi. Non vengono utilizzati cuscinetti a sfera nei giunti, ma boccole di plastica, più economiche, leggere e piccole. Grazie alle 3 videocamere è capace di vedere e rilevare le persone e la profondità della stanza. Pepper ha due telecamere RGB (con risoluzione di 5 MP con frequenza di fotogrammi 30 fps) montate sul fronte allineate verticalmente e un sensore 3D (con risoluzione di 320*240 pixel a 30 fps) situato dietro gli occhi. Nella Fig. 3 possiamo vedere dove sono posizionate le telecamere e il loro angolo di visione (Gelin, 2018).

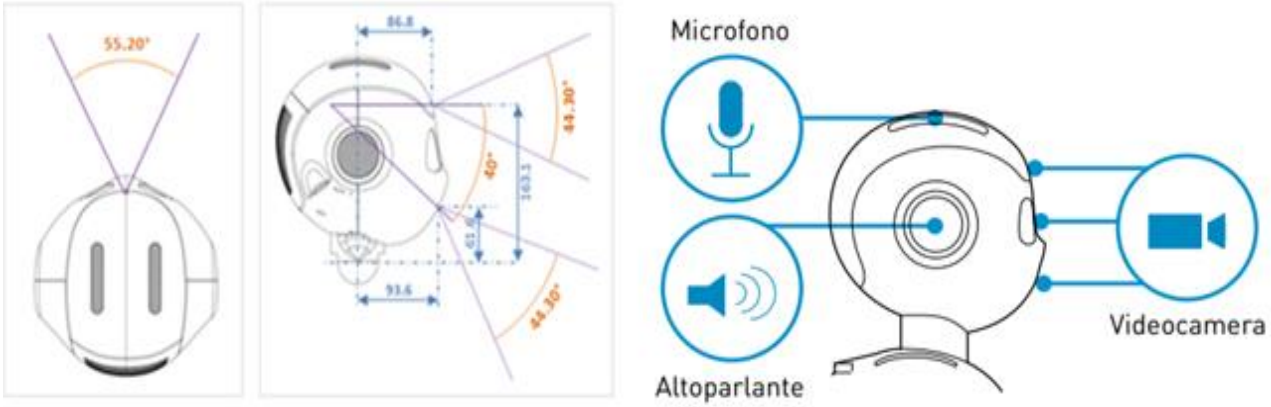


Fig. 3 – a sinistra gli angoli di vista, a destra posizione dei sensori

Pepper ha inoltre quattro microfoni in testa per valutare la corretta posizione da cui proviene la voce. Questi hanno una sensibilità di 250 Mv/Pa e una gamma di frequenza da 10 a 100 Hz. È dotato anche di due altoparlanti lateralmente posizionati a sinistra e a destra della testa, due sensori sonar, uno davanti e uno dietro, nonché di due sensori infrarossi alla base. Sono presenti sensori tattili, posizionati su testa e mani, oltre a sensori paraurti uno su ogni ruota. È dotato di un modulo di rilevamento laser composto da 6 linee laser e 3 sensori; tre linee laser sono posizionate nella parte anteriore per rilevare il terreno, i rimanenti sono nella parte inferiore del robot per percepire l'ambiente circostante. Nella Fig.4 sono elencati i sensori e viene indicato dove si trovano.

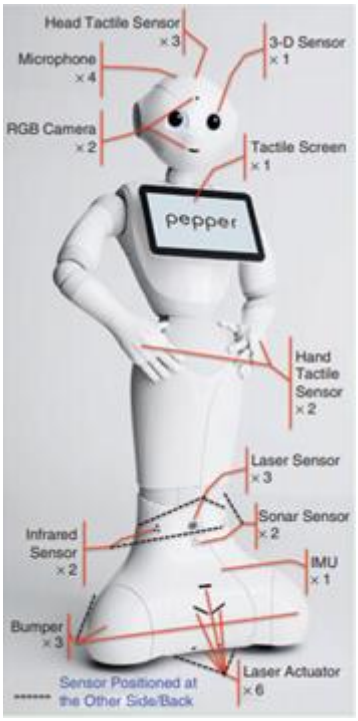


Fig. 4 posizione dei sensori

Supporta una connessione di rete Ethernet (1 x RJ-45 10/100/1000 Base-T) e Wi-Fi (IEEE 802.11). È provvisto di led posizionati sulle spalle i quali avvisano l'utilizzatore di nuove notifiche; il colore degli avvisi indicata il tipo di informazione (verde = informazione, giallo = avvertenza, rosso = errore). Sul torace è presente un tablet touch (capacitivo) da 10.1", con risoluzione di 1020x800, RAM di 1GB, capace di supportare file di tipo video, immagine e audio (Europe).

4.1 Kinematics Data

Si analizzano ora in maniera più dettagliata le dimensioni degli arti di Pepper e la cinematica dei movimenti. Nella Fig.5 osserviamo una panoramica delle grandezze del robot più dettagliata.

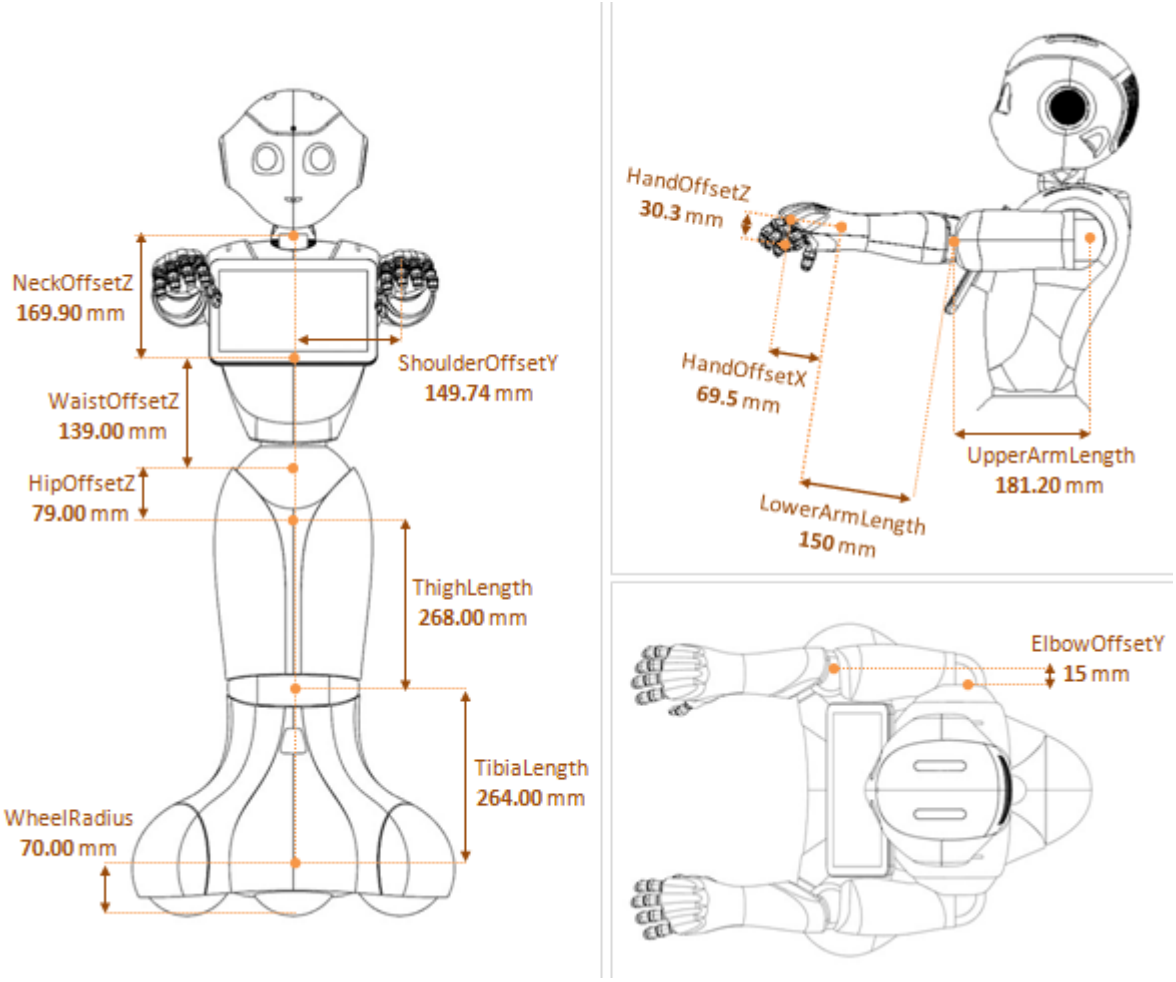


Fig. 5 - misure più dettagliate del robot

È presente un sistema di coordinate locali posizionato in HEAD, l'asse X è positivo verso la parte anteriore del robot, l'asse Y è positivo da destra verso sinistra e l'asse Z è posizionato verticalmente ed è positivo verso l'alto (Fig.6).

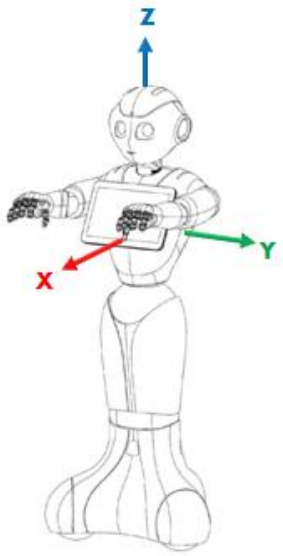


Fig. 6 – sistema di coordinate locali

Il corpo del robot è formato da elementi solidi, quali HEAD (testa), LARM (braccio sinistro), RARM (braccio destro), LEG (gamba) e TORSO (tronco).

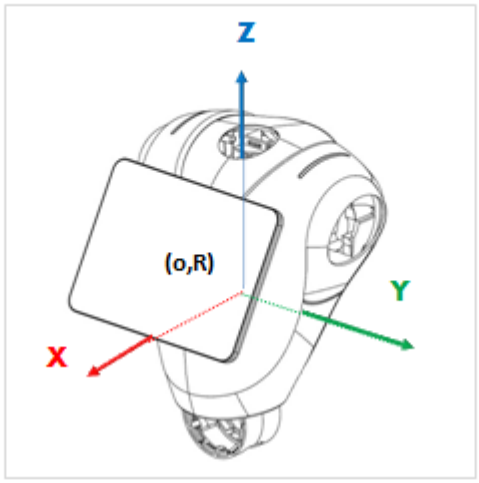
Per descrivere la massa, il centro di massa e la matrice inerziale di ogni elemento solido del robot utilizziamo questo tipo di matrici:

$$\text{Mass (kg)}$$

$$\text{CoM(S)} = \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_{(o,R)} \text{(m)}$$

$$[I_o(S)]_R = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}_R (\text{kg} * \text{m}^2)$$

La posizione dell'elemento solido e la matrice inerziale sono descritte rispetto al sistema di coordinate locali. Tutti i solidi e il sistema di coordinate locali sono descritti rispetto alla postura zero: in piedi con la gamba dritta e le braccia rivolte in avanti; un esempio di solido: TORSO (Fig.7).



$$\text{Mass} = 3.99533$$

$$\text{CoM(S)} = \begin{bmatrix} 0.00322 \\ 0 \\ -0.01029 \end{bmatrix}_{(o,R)}$$

$$[I_o(S)]_R = \begin{bmatrix} 0.0452761 & 0 & -0.00525567 \\ 0 & 0.0432702 & 0.000266886 \\ -0.00525567 & 0.000266886 & 0.0258881 \end{bmatrix}_R$$

Fig. 7 – rappresentazione di torso e delle matrici

Le articolazioni vengono definite attraverso dei nomi comuni e rispetto al punto chiamato TORSO situato a -38,00 X (mm) e 169,90 Z da HEAD.

Data un'articolazione che collega due parti del corpo del robot, la parte più vicina al tronco è considerata fissa e la parte del corpo più lontana è quella che ruota attorno all'asse articolare. Quando il robot è in posizione zero, tutti i telai dei giunti hanno lo stesso orientamento, quindi le rotazioni di rollio avvengono attorno all'asse X, le rotazioni di beccheggio attorno all'asse Y e le rotazioni di imbardata attorno all'asse Z. Certi movimenti sono limitati in modo tale che i robot non vada in collisione con se stesso. In Fig.8 si possono analizzare i possibili movimenti.

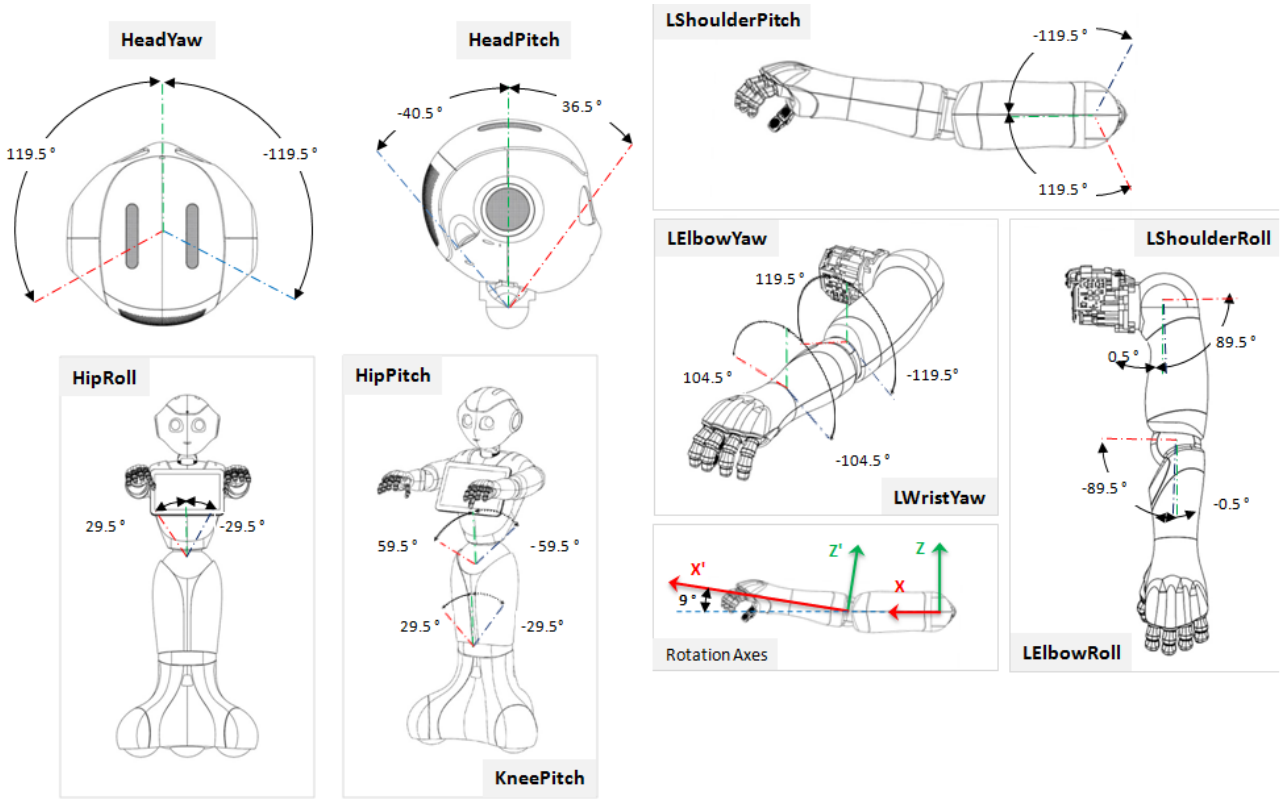


Fig. 8 – possibili movimenti degli arti

Il robot conosce delle posture predefinite come: postura zero, postura con le braccia lungo i fianchi, e riposo nella quale ha la testa rivolta verso il basso e le braccia lungo i fianchi (ALDEBARAN DOCUMENTION, s.d.).

5 Software

Il robot è controllato da un apposito sistema operativo, basato su Linux, chiamato NAOqi. Questo framework consente una comunicazione omogenea tra diversi moduli (movimento, audio, video), una programmazione omogenea e una condivisione omogenea delle informazioni; osserviamo alcune caratteristiche:

- è multipiattaforma, può essere sviluppato sia su Windows, Linux e Mac.
- è cross-language, con un API (Application Programming Interface) identica sia per C++ che per Python.
- fornisce introspezione, il che significa che il framework sa quali funzioni sono disponibili nei diversi moduli e dove.

5.1 NAOqi

NAOqi è un broker, cioè permette la cooperazione tra i sistemi; quando viene avviato, carica un file delle preferenze chiamato *autoland.ini*, che definisce le librerie da caricare. Ogni libreria contiene uno o più moduli i quali vengono utilizzati dal broker per manifestare i propri metodi.

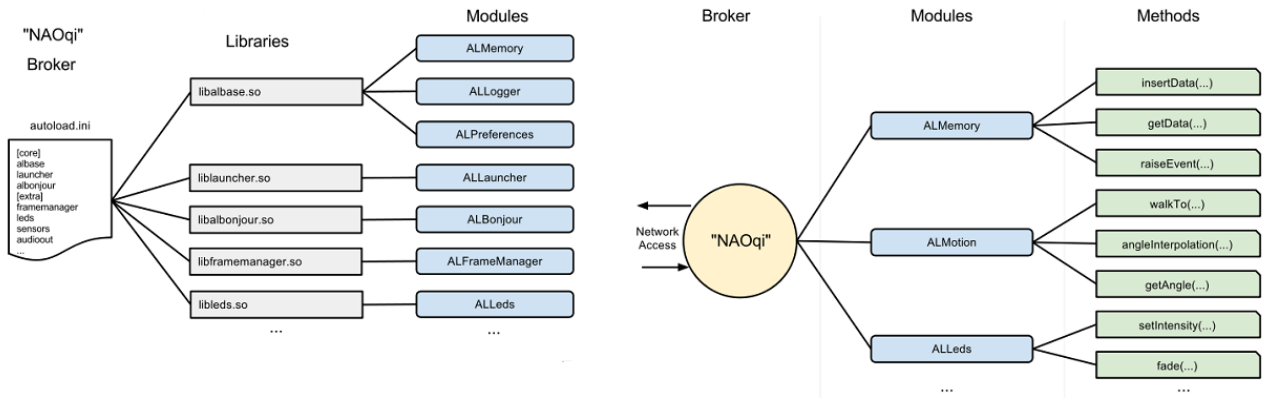


Fig. 9 – sistema a forma di albero

Il broker è un oggetto che fornisce due ruoli principali: consente di trovare moduli e metodi, consente di chiamare i metodi collegati all'esterno del processo (tramite accesso rete); fornisce servizi di ricerca in modo che qualsiasi modulo, o nell'albero o attraverso la rete, possa trovare qualsiasi metodo per essere esplicitato. Il caricamento dei moduli forma un albero di metodi collegati ai moduli, a sua volta collegati a un broker (Fig. 9). Tipicamente ogni modulo è una classe all'interno di una libreria. Quando una libreria viene caricata da *autoland.ini*, viene istanziata automaticamente la classe del modulo. I moduli più utilizzati sono:

- Modulo *ALAanimationPlayer*, responsabile della gesticolazione; migliora le espressioni del robot.
- Modulo *ALTablet*, responsabile dell'ottenimento di informazioni dell'utente da parte del tablet.
- Modulo *ALTextToSpeech*, converte un file testo in un audio dal suono naturale.
- Modulo *ALVideo*, responsabile delle informazioni visive del robot.

Un modulo può essere remoto o locale. Se è remoto viene compilato come file eseguibile e può essere eseguito all'esterno del robot; sono più facili da usare, ma meno efficienti in termini di velocità e utilizzo di memoria. Se è locale viene compilato come libreria e può essere utilizzato solo sul robot; tuttavia sono più efficienti rispetto a quelli remoti. All'interno dei moduli troviamo vari metodi; alcuni sono vincolati, il che significa che non possono essere chiamati dall'esterno del modulo (per esempio all'interno di un altro modulo) (Vittorio Pereira, 2017).

5.2 Choregraphe

La SoftBank Robotics, la società che produce Pepper, mette a disposizione come SDK (Software Development Kit) "Choregraphe", un software di programmazione a blocchi, o programmazione visuale. Dispone di una interfaccia grafica intuitiva che lo rende facilmente utilizzabile anche dai principianti. Il kit di sviluppo è compatibile con numerose piattaforme robotiche e linguaggi software: Java, C++, Python, Matlab, ecc. L'ambiente Choregraphe dovrebbe essere utilizzato per costruire applicazioni utilizzando alcuni dei blocchi/pose già esistenti, raggruppati in biblioteche. Ciò permette di creare delle pose/movimenti, che potranno essere provate e simulate, o sperimentarle direttamente sul robot reale, tutto ciò senza scrivere una sola riga di codice. È anche possibile cambiare la funzione di un blocco modificando il suo codice Python, in questo modo un utente ha pieno accesso a tutti moduli integrati e può espandere le funzionalità del robot con nuove risorse praticamente illimitate. I movimenti creati con Chorographe sono scritti nel suo linguaggio grafico specifico, di conseguenza NAOqi li interpreta e li esegue. Troviamo su questo SDK vari strumenti molto utili (Fig.10) come il pannello Monitor video, il pannello Gestione comportamento, la barra degli strumenti, la vista robot e la Timeline (ALDEBARAN DOCUMENTATION, s.d.).

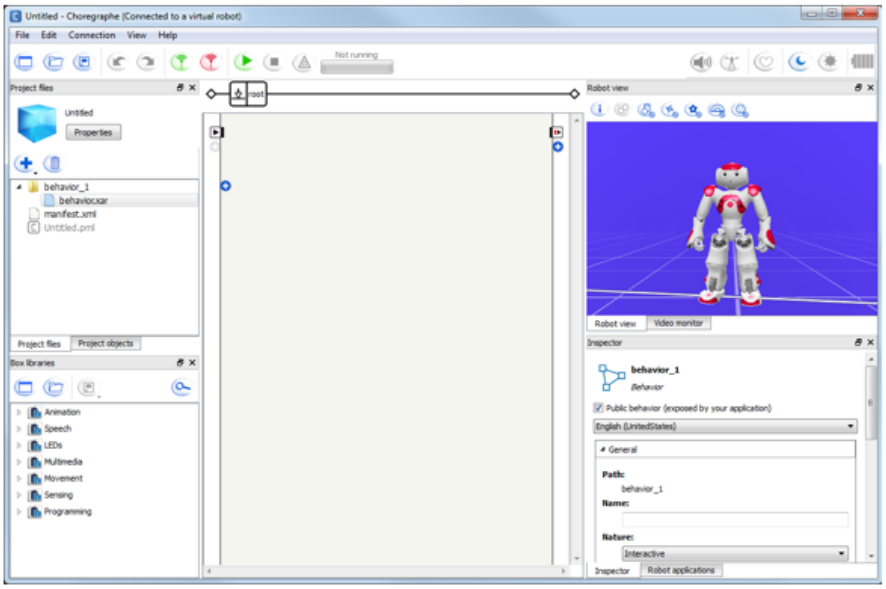


Fig. 10 – pannello Choregraphe

5.2.1 Come creare un'applicazione

Inizialmente si crea un PROJECT (progetto), che verrà salvato sul computer sotto forma di cartella, all'interno verrà creata un APPLICATION (applicazione) utilizzando:

- Behavior;
- Dialog topic;
- Planar move;
- Attached file.

Esistono delle applicazioni già create che possono essere scaricate, caricate su Choregraphe e poi istallate per testarle sul robot; oppure possono amplificare il nostro tipo di progetto aggiungendo nuove applicazioni (ALDEBARAN DOCUMENTATION, s.d.).

5.2.2 Behavior

Creare un Behavior (in italiano comportamento) è un modo semplice per far parlare, camminare, ballare, riprodurre un suono al robot. Il file contiene uno o più blocchi che vengono eseguiti in sequenza o simultaneamente. Viene salvato all'interno della cartella Project, archiviato sotto forma di file ".XAR" e situato in una sottocartella. Per crearlo si usa il "flow diagram panel" (pannello diagramma di flusso Fig. 11) che è il pannello dove andiamo a comporre l'applicazione vera e propria.

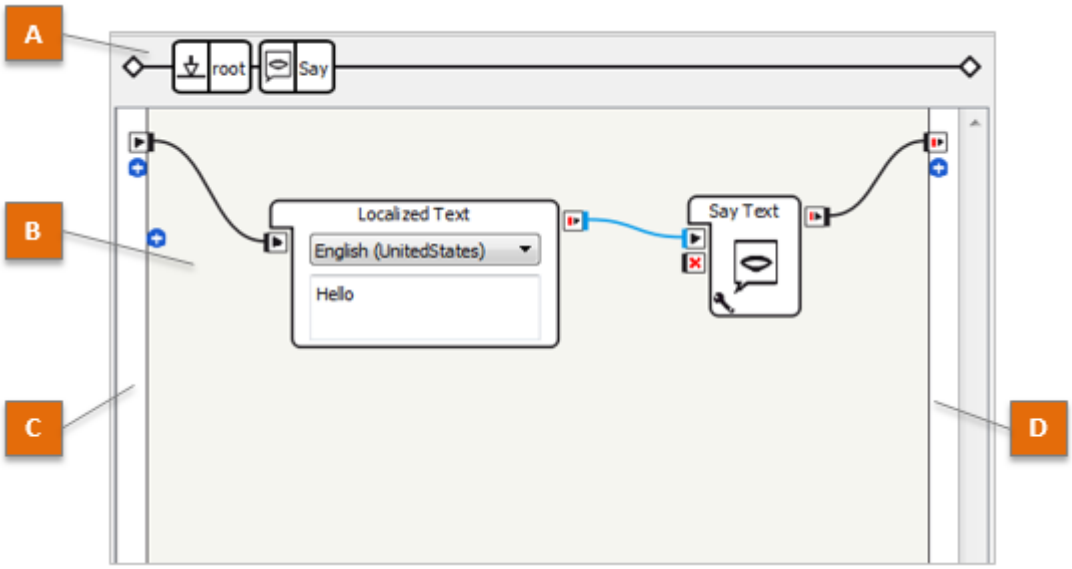


Fig. 11 – flow diagram pannel

In Fig. 11 troviamo un esempio di pannello diagramma di flusso; si analizziano ora i vari componenti:

- A ➡ Box path: si possono sfogliare i diversi livelli del diagramma, soprattutto quando un blocco contiene altri blocchi;
- B ➡ Flow diagram: crea o modifica il diagramma di flusso;
- C ➡ Input border: aggiunge, modifica o elimina gli input del blocco corrente;
- D ➡ Output border: aggiunge, modifica o elimina gli output del comportamento del box corrente.

Un diagramma di flusso è un gruppo di blocchi collegati tra loro con almeno un input.

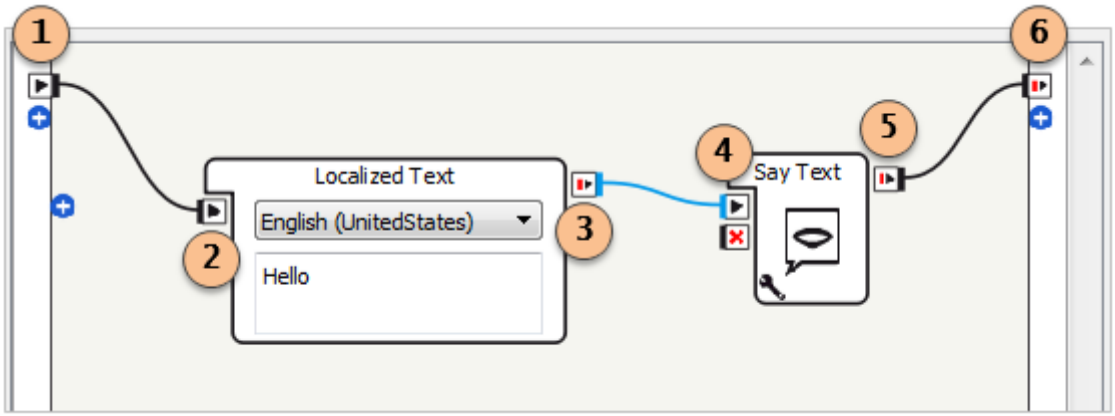


Fig. 12 – esempio diagramma di flusso

In Fig. 12 è rappresentato un esempio di diagramma di flusso, dove l'ingresso principale onStart del diagramma (1) è collegato all'ingresso onStart della casella di testo "localized text"(2); l'uscita onStopped della casella testo (3) è collegata all'input onStart della casella "Say text"(4); l'uscita onStopped della casella "Say text" (5) è collegata all'uscita principale onStopped del diagramma di flusso (6). Per inserire un blocco all'intero del diagramma basta aprire la libreria, selezionare e trascinare il blocco. Per creare i collegamenti, basta selezionare l'output onStopped-button di un blocco e trascinarlo fino all'input onStart-button di un altro blocco. Questo collegamento trasporta un semplice comando, attivando un blocco quando il blocco precedente ha terminato la sua esecuzione (ALDEBARAN DOCUMENTATION, s.d.).

5.2.3 Blocco

Un blocco è l'elemento base di Behaviors. Un box può contenere una semplice azione come "parlare", o un'azione molto complessa come "esplora la stanza".

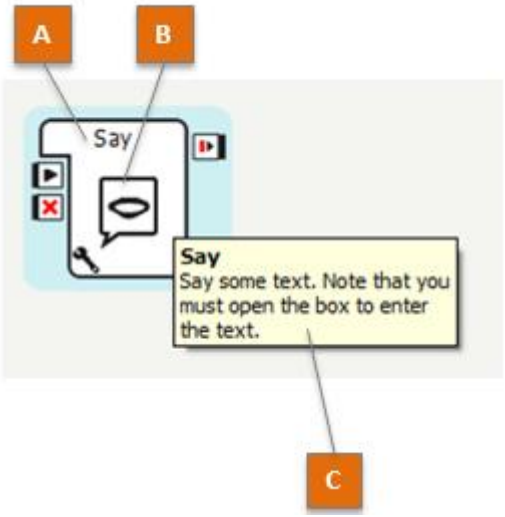


Fig. 13 – esempio di blocco

In Fig.13 si trova un esempio di box, esaminiamo le componenti:

- A ➡ nome del blocco;
- B ➡ immagine del blocco, aiuta per ricordare il tipo di blocco;
- C ➡ descrizione del blocco;

Il contenuto di un box dipende dal tipo di box, esistono:

- Python box, i quali sono scritti in Python, in questo modo è possibile utilizzare e importare qualsiasi modulo Python;
- Flow diagram box, i quali contengono un diagramma di flusso, può essere utile per creare un blocco il quale contiene a sua volta più blocchi, tutto ciò può essere collegato tra loro creando un comportamento/azione complessa; in questo caso avremo più livelli;
- Timeline box, che contengono una timeline;
- Dialog box, questi descrivono un tipo di dialogo, è possibile scegliere che cosa pronunciare e la lingua di pronuncia;

I blocchi hanno dei connettori per poter comunicare con altri blocchi, per essere avviati o arrestati. I connettori sono collegati tramite collegamenti su una logica di comunicazione basata su eventi.

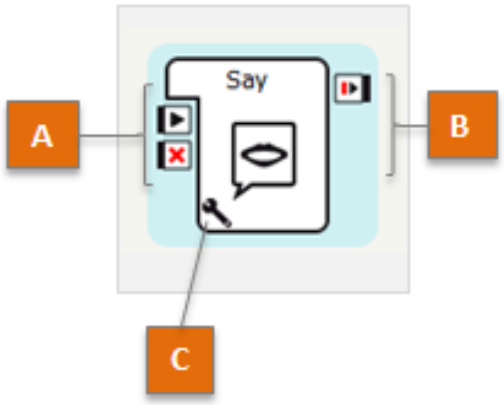


Fig. 14 – esempio di blocco

La Fig. 14 osserviamo come in (A) troviamo gli input dove si riceve eventi per avviare o fermare il box; in (B) troviamo gli output dove si invia eventi e/o dati durante l'esecuzione del box o quando l'esecuzione del box viene interrotta; in (C) si trovano i parametri/dati utilizzati dal box (ALDEBARAN DOCUMENTATION, s.d.).

5.2.4 Box input/output

Esistono diverse nature di input, alcuni sono accessibili solo dall'interno dei blocchi, alcuni solo dall'esterno e altri da entrambi. Per attivare gli input che sono disponibili dall'esterno, è necessario collegarli o all'ingresso principale dello schema o all'uscita di un blocco, gli altri vengono attivati automaticamente quando viene generato un evento speciale.

L'immagine visualizzata su un ingresso dipende dalla sua natura:

Sul blocco	All'interno del diagramma	Natura	Descrizione
		onStart	Quando questo input viene stimolato il box si avvia
		onStop	Quando questo input viene stimolato il box si arresta
		onEvent	Questo ingresso non ha alcun effetto sul box, non l'avvia né la ferma. Quando viene stimolato l'ingresso viene trasmesso allo schema del box
		ALMemory Input	Questo tipo di input speciale è visibile solo all'interno del diagramma, quindi non puoi stimolarlo dall'esterno della scatola. Viene stimolato ogni volta che vengono aggiornati i valori dei dati memorizzati in ALMemory
		onLoad	Questo tipo di input è visibile solo all'interno del diagramma e quando il box è una linea temporale.

Esistono anche diversi tipi di output che si possono stimolare dal diagramma o dalla Timeline, in questi casi è necessario collegare l'output a un'usciti del box. L'immagine visualizzata su un'uscita dipende dalla sua natura:

Sul blocco	All'intenro del diagramma	Natura	Descrizione
		onStopped	• Timeline box: quando questo output viene stimolato dal diagramma, il box viene arrestato • Dialog box: quando questo output viene stimolato dal diagramma o dallo script QiChat, l'argomento della finestra di dialogo viene disattivato • Python box: questo output non ha alcun effetto specifico, non arresta il box.
		puntctual	Questa uscita non ha alcun effetto specifico sul blocco, non l'avvia né la ferma.

La comunicazione tra i blocchi è basata sugli eventi, un semplice segnale di evento può essere inviato da una casella all'altra. Il segnalo può anche trasportare informazioni come una stringa, un numero e un array. Il colore di un input/output dipende dal tipo di dati che trasporta. In tabella troviamo tutti i tipi di I/O con il colore corrispondente:

input	Output	tipo	Descrizione
		Bang	Questo tipo non porta alcun dato, ma solo le informazioni che viene stimolato
		Numero	Rappresenta un evento che trasporta dati. Questi dati possono essere un numero(float o int) o un array di numeri
		Stringa	Trasporta dati. Questi dati possono essere una stringa o un array di stringhe
		Dynamic	Rappresenta un evento semplice, come il bang, o un evento che trasporta dati.

(ALDEBARAN DOCUMENTATION, s.d.)

5.2.5 Dialog topic

Dialog topic è un modo semplice per fornire al tuo robot la capacità di conversazione. Un Dialog topic è un set multilingue di script QIChat, che include:

- Un file DLG, che rappresenta l'argomento della finestra di dialogo e che registra le lingue supportate
- E da *uno* a *n* file TOP, ognuno contenente lo script QiChat del linguaggio supportato da Dialog topic. (ALDEBARAN DOCUMENTATION, s.d.)

5.2.6 Planner move

Un Planner move è un file che descrive i movimenti del robot in uno spazio piano. I file di spostamento hanno un'estensione in PMT. Un movimento è costituito da traiettorie e ogni traiettoria è composta da più percorsi, che collegano due pose. All'interno di una traiettoria i movimenti sono concatenati, di conseguenza continui.

Choregraphe integra tutto il necessario per poter:

- Creare un movimento su un piano;

- Modificare una mossa utilizzando l'editor movimenti;
- Integrare una mossa all'interno di un Behavior (ALDEBARAN DOCUMENTATION, s.d.).

5.3 ROS

È stato utilizzato anche ROS (Robot Operating System) come strumento open source, adatto per la realizzazione di applicazioni per Pepper. ROS è un sistema modulare che si amplifica grazie alla comunità di sviluppatori; in questo modo consente a ciascuno di utilizzare quella parte di ROS che ritiene necessaria per implementare il proprio progetto; grazie all'open source chiunque può contribuire aggiungendo pacchetti alla libreria ROS. All'interno di questo 'ecosistema' sono presenti algoritmi all'avanguardia e un quadro di comunicazione e visualizzazione tra processi e utensili. Nonostante i produttori di Pepper limitino l'accesso al suo sistema operativo, forniscono un driver che può essere utilizzato per collegare insieme NAOqi e ROS. Questo driver recupera tutti i sensori presenti e crea nodi e argomenti ROS i quali pubblicano lo stato di tutti i sensori del robot, inoltre crea argomenti per il controllo delle articolazioni del robot. Il ruolo principale del driver NAOqi è convertire i moduli NAOqi in nodi ROS (Fig.15).

Fig. 15 – esempio sistema ROS

È stato introdotto ROS per creare applicazioni molto più 'complicate' come ad esempio la navigazione autonoma all'intero di una stanza e per evitare gli ostacoli mentre si sposta.

ROS ha tre livelli concettuali: il livello del Filesystem, il livello di Computation Graph e il livello della Community. Questi livelli sono riassunti di seguito. (romero, s.d.)

5.3.1 Filesystem level

Questa parte riguarda principalmente le risorse ROS presenti sul disco, come:

- Pacchetti: sono l'unità principale per l'organizzazione del software in ROS. Un pacchetto può contenere processi runtime ROS (nodi), una libreria dipendente da ROS, set di dati, file di configurazione o qualsiasi altra cosa che sia utile. I pacchetti sono l'elemento di compilazione e rilascio più piccolo in ROS.
- Metapacchetti: sono pacchetti specializzati che servono solo a rappresentare un gruppo di altri pacchetti correlati. Più comunemente questi ultimi sono usati come sostituti temporanei retrocompatibili per gli Stack.
- Manifesti dei pacchetti: i manifesti forniscono i metadati di un pacchetto, inclusi il nome, la versione, la descrizione, le informazioni sulla licenza.
- Tipi di messaggio: le descrizioni dei messaggi definiscono le strutture di dati per i messaggi inviati in ROS (romero, s.d.).

5.3.2 Computation Graph level

Il Computation Graph è la rete peer-to-peer dei processi ROS che elaborano i dati insieme. I concetti base del grafico di calcolo sono nodi, master, server dei parametri, messaggi, servizi, argomenti e borse; analizziamo questi elementi:

- Nodes (nodi): sono processi che eseguono i calcoli. ROS è progettato per essere modulare; un sistema di controllo di un robot comprende solitamente molti nodi. Ad esempio, un nodo controlla i motori delle ruote, un nodo

esegue la localizzazione, un nodo esegue la pianificazione del percorso, un nodo fornisce una vista grafica del sistema e così via. Un nodo viene scritto con l'uso di una libreria client ROS.

- Master: fornisce la registrazione del nome e la ricerca nel resto del grafico di calcolo; senza quest'ultimo i nodi non sarebbero in grado di trovarsi, scambiarsi messaggi o richiamare servizi.
- Parameter Server: consente di memorizzare dati, fa parte del master.
- Messages (messaggi): i nodi comunicano tra loro tramite messaggi, i quali sono semplicemente una struttura di dati, composta da campi digitali; supportano boolean, integer e arrays.
- Topics (argomenti): i messaggi vengono orientati tramite un sistema di trasporto con semantica di pubblicazione. Un nodo invia un messaggio pubblicandolo su un determinato argomento. L'argomento è un nome utilzzato per identificare il contenuto del messaggio. Un nodo interessato a un certo tipo di dati si inscriverà all'argomento appropriato. Si può immaginare un argomento come un bus di messaggi fortemente tipizzato, ogni bus ha un nome e chiunque può connettersi al bus per inviare o ricevere messaggi purchè siano del tipo giusto.
- Services Servizi: domande\risposte vengono effettuare tramite servizi, definiti da una coppia di strutture di messaggio: una per le domande e una per le risposte.
- Bags (borse): le borse sono un formato per salvare e riprodurre i dati dei messaggi ROS; sono un meccanismo importante per la memorizzazione dei dati, come i dati dei sensori. Può essere difficoltoso raccogliere tutti i dati ma è necessario per lo sviluppo degli algoritmi.

Il ROS Master funge da servizio di nomi nel grafico di calcolo, memorizza argomenti e informazioni. I nodi comunicano con il Master per segnalare le proprie informazioni di registrazione. Poiché questi nodi comunichino con il master, possono ricever informazioni su altri nodi e stabilire connessioni

appropriate. Il master effettuerà dei richiami ai nodi quando le informazioni registrate variano, il che consente ai nodi di creare connessioni dinamiche man mano che vengono eseguiti nuovi nodi.

I nodi si connettono automaticamente ad altri nodi, il master fornisce solo informazioni di ricerca, proprio come un server DNS. Questa architettura consente operazioni disaccoppiate, in cui i nomi son il mezzo principale con cui è possibile costruire sistemi più grandi e complessi. I nomi hanno un ruolo molto importante in ROS: nodi, argomenti, servizi e parametri hanno tutti nomi. Ogni libreria client ROS supporta la rimappatura dei nomi dalla riga di comando, il che significa che un programma compilato può essere riconfigurato in fase di esecuzione per operare in una diversa topologia del grafico di calcolo. In Fig. 16 troviamo uno schema della comunicazione tra i nodi (romero, s.d.).

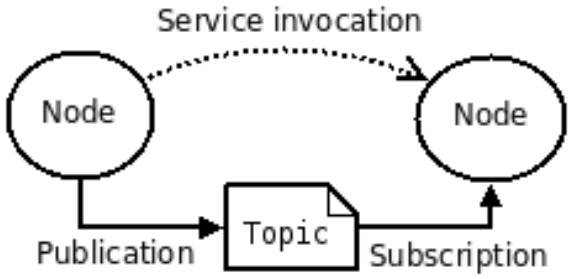


Fig. 16- esempio di comunicazione tra nodi

5.3.3 Comunity level

Questo meccanismo permette di creare una comunità ROS che consente lo scambio di software e conoscenza. È presente il ROS Wiki, il forum principale per documentare le informazioni, chiunque può registrarsi e contribuire con la propria documentazione, fornendo correzioni o aggiornamenti, scrivere tutorial o altro ancora (romero, s.d.).

6 *Navigazione e localizzazione del robot*

Dopo aver compiuto un'analisi completa del robot, possiamo osservare diversi vantaggi per l'iterazione uomo-robot, come il suo aspetto amichevole, ma anche limitazioni importanti, come la sua sensibilità ridotta e la sua modesta capacità di elaborazione. A differenza di robot personalizzati, che generalmente si affidano a costosi LIDAR (Light Detection and Ranging è una tecnica di telerilevamento laser) per la localizzazione metrica e la navigazione, che funzionano in ambienti sia interni che esterni, Pepper dispone di LIDAR a corto raggio e di telecamere che forniscono una localizzazione affidabile solo in piccoli ambienti interni, quindi incapaci di fornire informazioni utili per localizzare il robot in ambienti di grandi dimensioni. Questo è un grosso problema per Pepper, che potrebbe essere utilizzato non solo nelle case, ma anche in luoghi pubblici come ospedali, centri commerciali e scuole. Per risolvere questo problema, ci si è appoggiati sui recenti progressi del sistema visual SLAM, sviluppato su una soluzione di auto-localizzazione basata su SLAM visuale aiutata dalla ruota odometrica, che consente a Pepper di auto-localizzarsi e navigare in ambienti di grandi dimensioni. Il motivo per includere l'odometria nelle procedure di stima visiva è per recuperare la scala metrica (sconosciuta negli schemi puramente visivi tipici) e per rendere il sistema visivo più robusto (Cristopher Gomez, 2019).

6.1 visual SLAM

Visual SLAM presenta una soluzione economica per applicazioni che richiedono funzioni di localizzazione e mappatura come realtà aumentata, realtà virtuale e sistemi autonomi (come ad esempio automobili e droni di ispezione). Oggigiorno sono presenti modelli ottimizzati che eliminano problemi di filtraggio dati, i quali sono preferibili per la loro superiore precisione e perché hanno un costo computazionale simile. Gli approcci basati sull'ottimizzazione modellano il problema come un grafico di fattori che mette in

relazione in maniera probabilistica diverse variabili, come pose e punti di riferimento, dai cosiddetti fattori, che corrispondono alle misurazioni del sensore o da vincoli fisici. Un esempio di sistema visivo SLAM è mostrato in Fig.17, dove sono presenti diversi grafici di fattori relativi agli approcci di ottimizzazione. I cerchi denotano variabili come punti della mappa o fotogrammi chiave all'interno dello schema visivo; i cerchi di colore bianco sono attivi, i cerchi di colore grigio sono fissi, i quadrati indicano fattori o misurazioni.

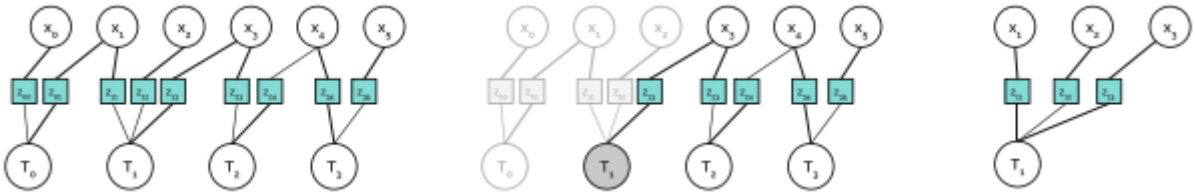


Fig. 17 – esempi di grafici di fattori

Il grafico di fattori può essere formulato come un problema non lineare dei minimi quadrati che mira a trovare gli stati $x^* = x_1, \dots, x_m$ che minimizzano l'errore tra le misurazioni Z_i e un modello osservazionale $h_i(x_i)$, che 'predice' la misura attesa dato lo stato x_i,

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} = \sum_{i=1}^{m} \|h_i(\mathcal{X}_i) \boxminus \mathcal{Z}_i\|^2_{\mathbf{\Omega}_i}$$

Lo stesso sistema vale per il caso visivo, dove gli stati corrispondono alle pose della telecamera che sono selezionate dalla traiettoria.

Per quanto riguarda i sistemi attuali, sono state sviluppate diverse soluzioni per sensori monoculari e di profondità. I sistemi SLAM visivi monoculari sono fondati su funzionalità che utilizzano solo alcune informazioni dell'immagine, mentre ORB-SLAM o metodi diretti sfruttano le informazioni complete da ogni immagine come LSD-SLAM. Il problema principale con i sistemi monoculari è che richiedono una telecamera mobile per stimare la profondità della scena, oltre a dipendere da un fattore

di scala sconosciuto che mappa gli stati stimati su dimensioni fisicamente coerenti. L'approccio tipico per risolvere il problema si basa sull'utilizzo di diverse fonti di informazione che forniscono la scala, come l'unità di misura inerziali (IMU); tuttavia, questo aumenta i requisiti di calcolo del problema di stima, poiché il numero di stati aumenta. A causa della mancanza di una scala metrica si è utilizzata una strategia per risolvere i problemi visivi di SLAM, aiutando il sistema con l'odometria delle ruote di misurazione.

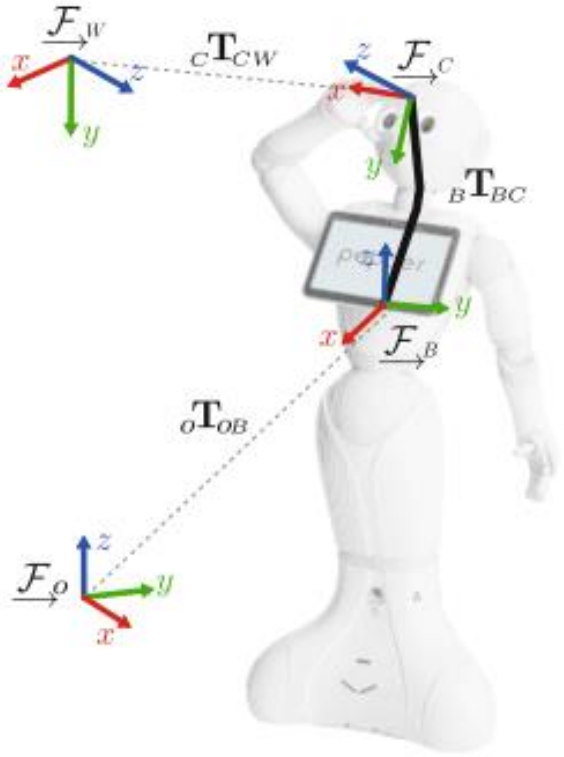


Fig. 18 – vista del robot con sistemi di riferimento

Per prevenire confusione sulla Fig. 18, seguiamo come convenzione che:

- le coordinate del frame A sono definite da F_A;
- la matrice di trasformazione omogenea (matrice di rotazione) T_{WC}, rappresenta la posa del frame F_C, rispetto al world frame, F_W, visto dal frame F_O;
- F_C, definisce il frame della camera;
- F_O, il frame dell'odometria;

- F_B, il frame del corpo.

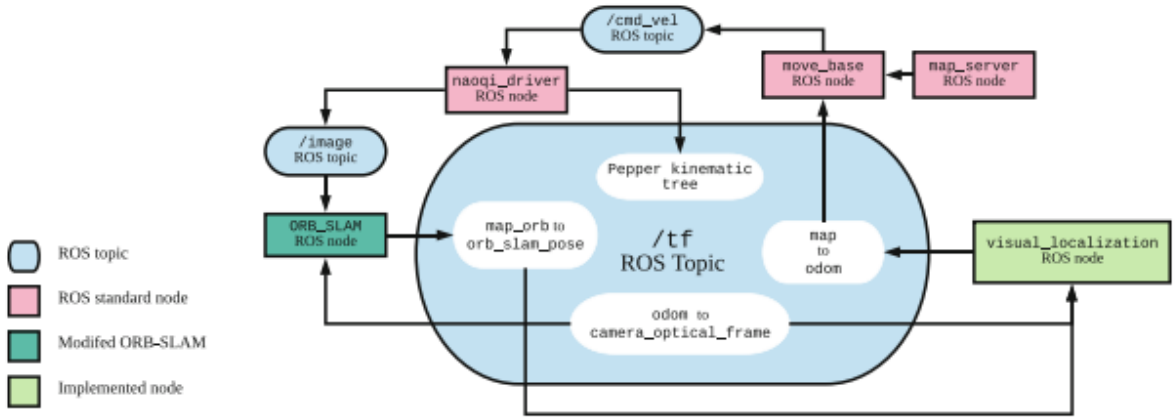


Fig.19 – esempio del sistema visual slam

In Fig. 19 troviamo la panoramica del sistema proposto. Le immagini della telecamera vengono inviate all'ORB-SISTEM-SLAM, insieme alla posizione della telecamera rispetto al telaio odometrico (odom). Viene fornita una stima della posizione della telecamera rispetto a un fotogramma arbitrario come output di ORB-SLAM. Il nodo di localizzazione visiva prende queste informazioni e le informazioni cinematiche di Pepper per calcolare una trasformazione tra il frame 'map' e 'odom' frame.

Questo sistema di localizzazione e navigazione basato su SLAM visivo, per Pepper è costituito da tre componenti principali:

1. Un sistema di localizzazione e mappatura basato su ORB-SLAM, che utilizza una singola telecamera RGB situata nella fronte di Pepper, e misurazioni odometriche calcolate dal software del robot;
2. il secondo componente corrisponde alla 'visual_localization' Nodo ROS che trasforma la stima della posa della fotocamera di ORBSLAM in una trasformazione tra la standard 'map' frame e 'odom' frame;
3. Infine, il nodo 'move_base' esegue il processo di navigazione. (Cristopher Gomez, 2019)

6.2 Localizzazione

Per il sistema di localizzazione utilizziamo la stessa architettura software già vista:

1. le immagini arrivano e vengono elaborate nel file 'tracking thread', il quale crea nuovi punti sulla mappa e fa una stima dell'attuale posizione della camera rispetto al frame globale;
2. un thread di mappatura locale che si basa sulla mappa e sui fotogrammi chiave, in pratica aggiorna le posizioni dei punti della mappa e la posizione della telecamera sui fotogrammi chiave;
3. infine, un ciclo di chiusura thread rileva in loop la traiettoria e la corregge (Cristopher Gomez, 2019).

Si noti che THREAD è una suddivisione di processo in due o più istanze o sottoprocessi che vengono eseguiti contemporaneamente da un sistema di elaborazione.

6.3 Modifiche di monitoraggio

Il 'tracking thread ' elabora non solo immagini ma anche misurazioni odometriche, ottenute direttamente da ROS. Le misurazioni dell'odometria vengono calcolate all'interno del software di Pepper e lette in ROS tramite wrapper NAOqi rispetto al frame odom. Questo wrapper richiede una misurazione dell'odometria ogni volta che viene ricevuta una nuova immagine, ottenendo una coppia sincronizzata immagine-odometria. Successivamente, ogni volta che viene creato un nuovo fotogramma chiave a seguito di un corretto rilevamento della telecamera, le informazioni sull'odometria sono incluse anche nel fotogramma chiave. Siccome ORB-SLAM smette di fornire posizioni alla telecamera quando il tracking fallisce, si imposta la stima della telecamera uguale alla previsione dell'odometria. Ciò garantisce un'ipotesi di posizione continua della telecamera per le attività di pianificazione, ma richiede che la scala metrica sia inizializzata. (Cristopher Gomez, 2019)

6.4 Inizializzazione della scala metrica

Inizialmente il sistema SLAM visivo viene inizializzato e la mappa non scalata viene costruita, dopo di che verrà calcolata la scala tra i fotogrammi chiave. Tuttavia, la mappa inziale è soggetta a importanti modifiche nelle prime fasi di mappatura. Pertanto, la correzione della scala viene eseguita dopo che è stato creato un numero fisso di *N* fotogrammi chiave, garantendo una convergenza soddisfacente e quindi una correzione della scala ragionevolmente affidabile. Il successo di questa strategia dipende solo dalle dimensioni dell'ambiente e dal movimento eseguito dal robot (Cristopher Gomez, 2019).

6.5 Mappatura locale

Ogni volta che viene creato un nuovo fotogramma chiave, 'local mapping' esegue l'ottimizzazione di un sottoinsieme della traiettoria, aggiornando sia le posizioni che la mappa. Le parti della traiettoria da ottimizzare sono formate dai fotogrammi chiave adiacenti all'ultimo fotogramma chiave aggiunto. Questa operazione garantisce un efficiente processo di ottimizzazione anche nelle mappe di grandi dimensioni. È possibile fondere le informazioni visive con le informazioni sull'odometria della ruota per evitare errori. Questo avviene aggiungendo fattori odometrici o vincoli tra i fotogrammi chiave. A tal fine la misurazione dell'odometria viene mappata dalla odometria del frame F_O, al frame F_C, utilizzando la cinematica di Pepper. Quindi calcoliamo la differenza tra le misurazioni dell'odometria *i-1* e *i*, tra tutti i fotogrammi chiave. L'errore tra le differenze è definito nella rappresentazione minima della posa (6 dimensioni) e si ottiene utilizzando la mappa logaritmica:

$$\varepsilon_{odo} = \mathrm{Log}_{\mathrm{SE}(3)} \left({}_{C}\mathbf{T}^{-1}_{C_{i-1},C_i} \mathbf{T}_{C_{i-1}W} \mathbf{T}^{-1}_{C_iW} \right).$$

(Cristopher Gomez, 2019)

6.6 Modalità di localizzazione e riutilizzo della mappa

ORB-SALM offre la possibilità di localizzare una mappa creata in precedenza. Questa localizzazione può essere eseguita in un singolo thread, quindi richiede una frazione dei

requisiti di calcolo rispetto al sistema completo. Tuttavia, è necessaria una mappa che è stata costruita nella stessa sessione in cui si voglia localizzare la mappa. Siccome è abbastanza scomoda questa strategia, si utilizza la funzionalità di autosalvataggio della mappa. Quando viene avviato con una mappa precostruita il robot Pepper deve solo rilocalizzarsi (Cristopher Gomez, 2019).

6.7 Discussione su localizzazione e navigazione

La soluzione appena proposta basata su SLAM è focalizzata sullo sviluppo di un sistema di auto-localizzazione in grado di gestire ambienti di grandi dimensioni nonostante il corto raggio dei LIDAR. Questo metodo di localizzazione e navigazione permette al robot di muoversi facilmente all'interno di stanze, ma continua ad avere problemi in vari casi: in ambienti molto aperti, in luoghi dove la luminosità è limitata e dove sono presenti superfici vetrose.

Si sta lavorando ad un'implementazione a bordo del sistema di auto-localizzazione su Pepper, inserendo altri sensori come i laser. In futuro si cercherà di migliorare la robustezza ai cambiamenti di illuminazione e ridurre il comportamento impreciso in ambienti di grandi dimensioni. (Cristopher Gomez, 2019)

7 Conversare con Pepper

Con il miglioramento del riconoscimento vocale della tecnologia robotica umanoide, i robot stanno cominciando a prendere il posto degli umani per qualche compito che richiede l'iterazione verbale, come i Robot di portineria o di accoglienza presso gli hotel e strutture qualsiasi. Un problema noto è il raggiungimento della 'naturalezza' dell'iterazione e il possesso di una base di conoscenza sufficientemente solida per portare avanti una conversazione soddisfacente (Shang Guo, 2017).

7.1 Primi metodi

I primi sistemi di dialogo intelligente agivano sulla conoscenza dichiarativa per elaborare i dati e rispondere agli esseri umani. La conoscenza dichiarativa, sotto forma di un dominio di regole, governa le risposte del robot. Si confrontano gli input/domande con le regole per far attivare delle azioni come: rispondere alle domande ed eseguire movimenti preselezionati o altri comportamenti. Tali sistemi sono stati utilizzati per robot che compiono giri all'interno di centri commerciali, robot guide nei musei e robot receptionist. All'interno del sistema basato sulle regole, l'input dell'utente doveva essere abbinato ai modelli delle regole. Questa corrispondenza limita molto l'input, cioè il modo in cui viene posta la domanda. Se la domanda dell'umano non segue il modello della regola, il robot non è in grado di rispondere alla domanda o di eseguire la procedura o il movimento (Shang Guo, 2017).

7.2 Natural Language Processing

Ora viene utilizzata l'elaborazione del linguaggio naturale (Natural Language Processing, NLP) per il ragionamento sul linguaggio umano. In questo modo si sono creati modelli di classificazione che comprendono meglio le domande o altre espressioni e mappano le domande, cioè riconoscono le domande della stessa 'classe' e permettono l'elaborazione della risposta. Esistono due approcci diversi sulla classificazione delle domande:

- gli approcci che legano la domanda alla regola, i quali risultano scomodi perché c'è la necessità di definire sempre più regole; questo metodo può funzionare male su nuovi set di dati;
- approcci basati sull'apprendimento, si esegue una classificazione estraendo le caratteristiche della domanda e utilizzando un classificatore, il quale viene creato apposta per prevedere le etichette delle classi per le nuove espressioni in input.

Messo da parte il discorso sulla classificazione del testo, la maggior parte del problema sta nel comprendere il contesto di un dialogo (Shang Guo, 2017).

7.3 Come funziona

Il sistema si attiva ricevendo un segnale acustico dall'esterno; il robot fornisce un feedback all'utente annuendo con la testa, sbattendo le palpebre e in alcuni casi dicendo "si". Questo discorso grezzo viene trasferito al servizio "speech-to-text", che restituisce una stringa di testo. Sul robot la stringa viene anche visualizzata dall'utente tramite il tablet presente sul petto. Nella domanda, per catturare l'attenzione del robot, bisogna pronunciare il nome di quest'ultimo (Fig. 20).

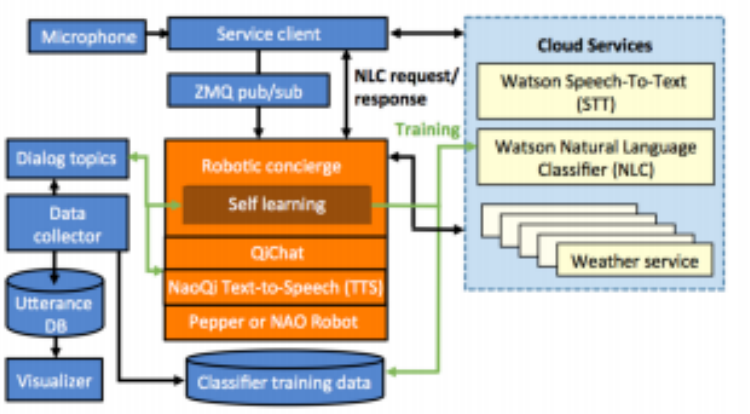


Fig. 20 - esempio di sistema di conversazione

Il messaggio appena acquisito viene passato al watson natural language classifier, che è inizializzato per il contesto specifico in cui adopera il robot. NLC raggruppa tutte le espressioni che essenzialmente dicono la stessa cosa in un'unica classe (ad esempio

"dove è il bagno" o "posso chiederti dove si trova il bagno"). In questo modo anche se l'espressione risulta diversa la risposta del robot dovrebbe essere la stessa. In generale ciò che il sistema fa è cercare la classe della domanda, poi restituisce la risposta associata a quella classe. Possono essere diverse le frasi di risposta a ciascuna classe. Alcune classi hanno modelli di risposta più complicati in cui è possibile inserire informazioni variabili. Per esempio "che ore sono?" o "qual è la temperatura di oggi", in questi casi il sistema consulta un servizio esterno.

Dopo che una risposta è stata formulata, viene passata a un Text-to-speech per generare un'uscita audio appropriata. Esistono alcune classi di richiesta che non seguono questo percorso, in particolare le richieste di cantare e ballare. In questi casi si attivano particolari suoni e animazioni sul robot.

Il robot, nel momento in cui gli viene fatta la domanda, risponde in questa maniera:

- comunica che ha ascoltato ciò che hai detto cambiando il colore degli occhi;
- comunica che ha capito ciò che hai detto facendo un cenno con la testa su e giù;
- nel caso preveda di non rispondere immediatamente in attesa di una elaborazione della risposta, dirà brevemente "mhh" o "ahh" (Shang Guo, 2017).

7.4 Autoapprendimento

Oltre a queste funzioni di base, il robot è in grado di migliorare le proprie prestazioni grazie al feedback dell'utente. Il feedback è mediato da un modulo di autoapprendimento. In questo modo forma nuove istanze di classi e permette di insegnare al sistema nuove coppie domanda/risposta, evitando la necessità di programmazione esplicita.

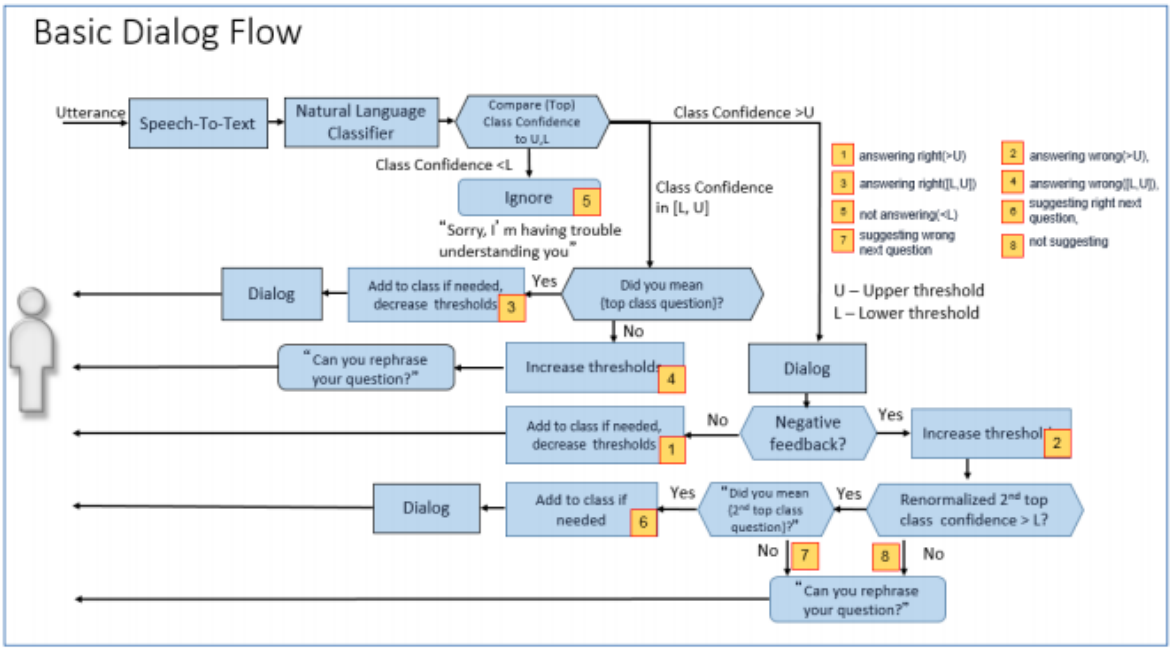


Fig. 21- base di un dialogo

In Fig. 21 troviamo una base di dialogo. Il sistema utilizza una soglia di comprensione dove *U* è la soglia superiore e *L* è la soglia inferiore. Troviamo tre casi fondamentali:

- Quando il sistema restituisce una comprensione superiore a *U*, il robot presume di aver capito la domanda e di aver trovato la risposta giusta, di conseguenza risponde;
- Quando il sistema restituisce una comprensione compresa da *U* e *L*, esso fa una domanda di conferma pe esempio "intendevi …";
- Quando il sistema restituisce una comprensione inferiore a *L*, molto probabilmente non si è in grado di rispondere alla domanda e di conseguenza la ignora.

Se il feedback dell'utente è negativo, cioè la risposta del robot non ha soddisfatto le aspettative, verrà variata la soglia, in maniera tale da non ripetere di nuovo l'errore. Se il robot riceve più volte un input che non riesce ad elaborare, a questo punto dovrà essere aggiunta manualmente una risposta al sistema (Shang Guo, 2017).

8 Riconoscimento Visivo Facciale

Come abbiamo detto sono sempre più utilizzati i robot reception. Analizziamo come funziona il modulo video per il riconoscimento visivo facciale.

Il modulo ALVideo utilizza le telecamere posizionate sulla testa del robot. Tuttavia, i movimenti della testa (causati da un algoritmo che simula i movimenti umani naturali) causano frequenti perdite di contatto con l'oggetto osservato. A causa di una potenza di calcolo relativamente bassa dell'hardware integrato, il rilevamento e il riconoscimento dei volti avviene in un lungo tempo, più o meno 10 secondi. Il problema è che una persona non può aspettare tutto questo tempo.

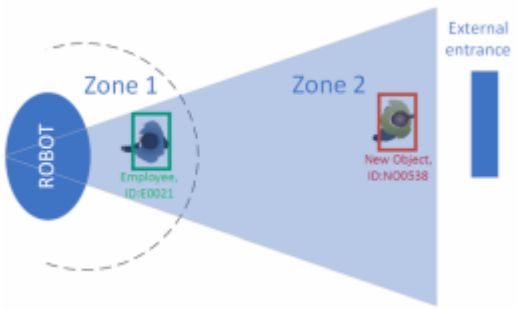


Fig. 22 - zone di rilevamento del robot

Perciò (come si può vedere in Fig.22) il compito chiave del modulo video è rilevare visivamente una persona della zona 2 in avvicinamento, per seguirlo e per identificarne le sue caratteristiche, consentendo la possibilità di identificazione o almeno di classificazione giunto nella zona 1. Questa tecnica sembra fallire se il rilevamento di una persona da parte di uno dei sensori è impedito (ad esempio robot posizionato dietro a una scrivania) (Arkadiusz Gardecki, 2018).

8.1 OpenCV

Un elemento inevitabile di un sistema di ricezione è la capacità di rilevare e riconoscere i volti, utilizzando un database interno o esterno. Pepper offre di più utilizzando la

libreria OpenCV, che consiste nell'elaborazione avanzata delle immagini; ma le funzionalità standard non sono spesso sufficienti per sviluppare una fonte di informazioni robuste. A volte, oltra ad utilizzare la fotocamera di Pepper, viene integrata una seconda telecamera esterna per rendere le immagini più chiare.

L'elaborazione video è eseguita in più fasi, riducendo al minimo la quantità di dati da elaborare, restringendo la regione di interesse e limitando o impedendo l'esecuzione di operazioni non necessarie (ad esempio, il riconoscimento facciale non viene eseguito se non è stato rilevato alcun volto).

Questa elaborazione è suddivisa in tre fasi:

1. La prima fase è progettata per rilevare il movimento dei pedoni nella zona più lontana e per restringere la regione di interesse nei fotogrammi video in un rettangolo contenente il pedone (riquadro rosso Fig. 23). Quando questo obbiettivo è raggiunto, il sistema passa alla modalità rilevamento oggetti; un numero ID viene assegnato all'oggetto/pedone, in modo che la sua identità può essere preservata per le prossime volte;
2. La seconda fase consiste nel rilevamento facciale, quando l'oggetto entra nella zona 2;
3. La terza fase inizia ogni volta che è stata rilevata una faccia (riquadro verde Fig.23). Viene eseguito un tentativo di riconoscimento del volto, nel caso fosse salvato in un database dei modelli di volti (Arkadiusz Gardecki, 2018).

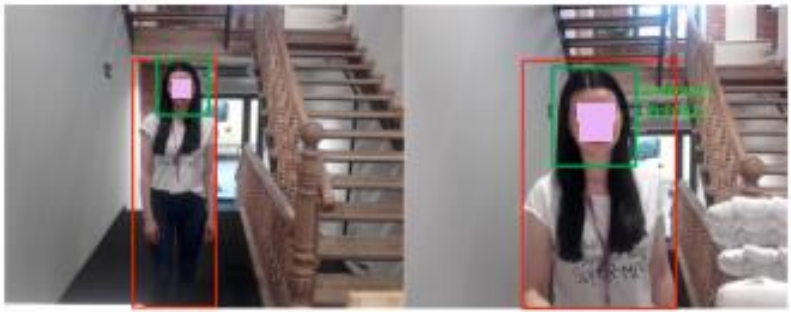


Fig. 23 - esempio di inquadratura

9 Conclusioni

I continui miglioramenti del robot Pepper presentati tengono conto del feedback degli utenti e della comunità scientifica che ne ampliano la portata per l'integrazione con le nuove tecnologie. Pepper è il primo nel suo genere a creare una strada per una nuova generazione di robot personali e di servizio prodotti anche in serie. Tuttavia, per ottenere il successo futuro di questi robot socievoli, c'è ancora molta strada da fare. Per esempio, non tutte le persone sono convinte nell'acquistare e avere un Pepper all'interno di una dimensione domestica. Infatti, il robot affronterà "un momento critico rispetto alla sua diffusione come prodotto di consumo praticabile", ha detto a PcWorld, un osservatore di robotica. Lo studioso Maisonnier afferma che Pepper diventerà più capace nel tempo, grazie agli sviluppatori che lo doteranno di nuove applicazioni. Gli utenti, poi, saranno in grado di scaricare e installare queste nuove funzionalità, così come aggiungere nuove app ai loro smartphone.

È impossibile non rendersi conto delle grandissime potenzialità di questo strumento, basti pensare all'estrema facilità di programmazione e alle sue specifiche tecniche, esso può essere utilizzato in molti settori lavorativi e nello stesso tempo come strumento educativo per far avvicinare i bambini e i ragazzi alla robotica. Utilizzando semplicemente i tre moduli che ho introdotto in questa tesi, si potrebbero creare migliaia applicazioni. In quanto il modulo della navigazione e auto-localizzazione serve al robot per muoversi in autonomia all'interno di ambienti, come centri commerciali, negozi, studi e scuole. Il modulo per la conversazione è vitale per sfruttare al meglio le sue caratteristiche da robot socio assistivo, potrebbe dare indicazioni all'interno di strutture, dialogare su un determinato argomento, ecc. In fine il modulo per il riconoscimento visivo è importante per sviluppare robot receptionist che potrebbero catturare l'utente che li utilizza. In sostanza combinando queste tre tecniche il robot è pronto per lavorare e aiutare l'utente. Uno dei campi di applicazione potrebbe essere quello della assistenza e della cura degli anziani.

In alcuni paesi stiamo assistendo ad una curiosa inversione demografica. In Giappone, ad esempio, molto presto il 30% della popolazione avrà più di 65 anni. Una situazione simile si presenterà in Cina verso il 2050, così come negli Stati Uniti, nell'America del Nord e in Europa. Secondo Brooks, ci ritroveremo in uno scenario in cui "gli anziani dovranno occuparsi di quelli più anziani". La robotica potrebbe essere la soluzione a questo problema. "Stiamo assistendo a molte prime prove di robot che aiutano gli anziani ad andare a letto, ad alzarsi dal letto, a salire le scale con la spesa, il genere di cose di cui avremo bisogno per mantenere la nostra indipendenza. Anche il veicolo automatizzato è un altro tipo di robot per la cura degli anziani perché ci permette di guidare più a lungo, di avere la nostra indipendenza più a lungo." (Bologna, 2016) Quelle descritte da Rodney Brooks sono delle innovazioni interessanti. L'esperto di robotica, che di recente si è espresso anche sui progressi dell'intelligenza artificiale, ha fornito un quadro molto chiaro dei risultati ottenuti fino ad oggi nel settore e di quelli che potremo ottenere in un futuro non molto lontano. Veicoli autonomi sempre più precisi, protesi che ci potenzieranno, robot sempre più sofisticati che condivideranno informazioni nei cloud e che potranno aiutarci in molte attività lavorative e nell'assistenza agli anziani. Un futuro in cui la robotica avrà un ruolo fondamentale e si rivelerà molto utile.

Rita Levi Montalcini in un articolo su mente e cervello di alcuni anni fa ricorda che Rodney Brooks, per capire le specialità negli esseri umani delle iterazioni intrattenute con gli altri e con il mondo esterno, per la costruzione del robot umanoide COG si è posto l'obbiettivo di riprodurre tali iterazioni artificialmente e di conciliare le scoperte della psicologia cognitiva e delle neuroscienze. Egli sostiene: "può essere che tra 40 anni non avremo ancora raggiunto questo obbiettivo, sono provato, tuttavia si potrà capire dove stanno le vere difficoltà. Fin ora tutti i robot costruiti dall'uomo si possono spegnere senza rimpianti. Nel momento di maggiore ambizione amo pensare che arriveremo a costruire un robot che non ci farà piacere di spegnere" previsione avveratasi con la costruzione del robot umanoide COG e con la costruzione di robot con capacità di interazioni come il nostro Pepper

1 Bibliografia

ALDEBARAN DOCUMENTATION. (s.d.). Tratto da Choregraphe: http://doc.aldebaran.com/2-4/software/choregraphe/index.html

ALDEBARAN DOCUMENTION. (s.d.). Tratto da kinematics data: http://doc.aldebaran.com/2-4/family/pepper_technical/kinematics_pep.html

Arkadiusz Gardecki, M. P. (2018). *The Pepper humanoid robot in front desk appliation.*

Bart de Witte, H. S. (s.d.). Il robot sociale potrebbe entrare a far parte di un team. (V. Zaslawaki, Intervistatore)

Cristopher Gomez, M. M.-d.-s. (2019). Visual SLAM-Based Localization and Navigation for Service Robots: The Pepper Case. In *Lecture Notes in Computer Science* (p. 32-44). Springer Nature Switzerland.

Daisy van der Putte, R. B. (2019). *A Social Robot for Autonomous Health Data Acquisition among Hospitalized Patients: An Exploratory Field Study.*

Europe, S. R. (s.d.). manuale d'istruzioni.

fastweb DigitalMegzine. (2017, maggio 18). Tratto da https://www.fastweb.it/web-e-digital/la-storia-dei-robot/

Gelin, A. K. (2018). *A Mass-Produced sociable humanoid robot.*

Paola Ardono, K. K. (2019). *A Hybrid Slam and Object Recognition System for pepper robot.* UK,Spain, Singapore.

robotiko. (2016). Tratto da robotiko: https://www.robotiko.it/robot-umanoidi-modelli/

romero, A. M. (s.d.). *ROS.org*. Tratto da ROS\concepts: http://wiki.ros.org/ROS/Concepts

Shang Guo, J. L. (2017). *Conversational Bootstrapping and Other Tricks of a Concierge Robot.* Vienna.

Simone Verrasi, S. D. (2018). *A Social Robot for Cognitive Assessment.*

Vittorio Pereira, T. P. (2017). *Setting Up For Atuonomous Navigation And Personalize Interaction Withe Users.* Pittsburgh.

wikipedia. (s.d.). Tratto da wikipedia: https://it.wikipedia.org/wiki/Robot

Zhihao Shen, A. E. (2019). *Nonverbal Behavior Cue for Recognizing Human Personality Traits in Human robto social interaction.* JAPAN.

10 Ringraziamenti

Ringrazio innanzi tutto il professore Andrea Monteriù per avermi dato la possibilità di lavorare su questo interessante argomento, mi ha dato la possibilità di scoprire un nuovo mondo, quello della robotica. Ringrazio la correlatrice Sabrina Iarlori per l'aiuto dato per realizzare questa tesi, e mi scuso per il tempo che le ho fatto perdere.

Ringrazio la mia famiglia la quale mi ha dato supporto morale fin dal primo anno di Ingegneria, iniziato con molto scetticismo e concluso con molta allegria. Soprattutto ringrazio mio padre Andrea e mia madre Paola per l'aiuto e per l'incitamento dato, mia sorella Vicki come figura importante della mia vita da seguire e idolatrare. Non so come ringraziarvi Flavio e Maria Grazia, i miei nonni fantastici ai quali sarò sempre debito per i mille insegnamenti e per l'aiuto dato.

Ringrazio Vero per essermi stata accanto in questi ultimi anni anche nei momenti più delicati e difficili, spero di trascorrere altro molto tempo con te.

Ringrazio quel maledetto palazzo Massetti per avermi fatto conoscere due persone importantissime per me e per il mio percorso, Matteo e Jacopo, grazie ai loro consigli, aiuti e battute ho passato tre anni senza pensieri e in più senza di loro sono quasi sicuro che non avrei tagliato questo traguardo così facilmente. Spero che rimarremo sempre uniti come questi tre anni.

Ringrazio tutti i miei amici più cari che ormai conosco da tanto tanto tempo Ska, Mex, Lucco, Simo, Mala, Riccio, Franci, Luca, Monti, Dado e tutto il resto del gruppo che hanno sempre creduto in me senza mai dubitare delle mie possibilità e ci sono sempre stati per me, ragazzi ci sarò anche io sempre per voi, vi voglio bene.

Ed infine ringrazio il "nuoto" cosa a me molto cara, mi ha insegnato molte cose , mi ha fatto conoscere tantissime persone a me care, saluto e ringrazio tutti i partecipanti di "Toppi è down" un abbraccio vi voglio bene ragazzi, ringrazio il mio allenatore Sandro per i traguardi raggiunti insieme e tutta la Vela Nuoto Ancona.

Ed infine ringrazio me stesso perché solo io so veramente i sacrifici fatti e quanto è stata dura per me essere continuativo negli studi, per me è un traguardo gigantesco.

Grazie a tutti e a tutte le persone che ho nominato sarò sempre grato.

Un abbraccio Eugenio Vecchietti.