



Ricerca di Sistema elettrico

Realizzazione di componenti hardware e software per facilitare l'uso del robot NAO nella Smart Home e creare servizi innovativi per l'uso efficiente ed in sicurezza delle risorse elettriche e per l'Assisted Living

Prof. Vincenzo Bonaiuto

REALIZZAZIONE DI COMPONENTI HARDWARE E SOFTWARE PER FACILITARE L'USO DEL ROBOT NAO NELLA SMART HOME E CREARE SERVIZI INNOVATIVI PER L'USO EFFICIENTE ED IN SICUREZZA DELLE RISORSE ELETTRICHE E PER L'ASSISTED LIVING

Prof. Vincenzo Bonaiuto – Dipartimento Ingegneria Industriale – Università di Roma Tor Vergata

Aprile 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - II annualità

Obiettivo: Tecnologie

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Work package: WP1 Local Energy District

Linea di attività: 1.10 - Realizzazione di componenti hardware e software per la localizzazione del robot NAO

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Claudia Meloni, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione "REALIZZAZIONE DI COMPONENTI HARDWARE E SOFTWARE PER FACILITARE L'USO DEL ROBOT NAO NELLA SMART HOME E CREARE SERVIZI INNOVATIVI PER L'USO EFFICIENTE ED IN SICUREZZA DELLE RISORSE ELETTRICHE E PER L'ASSISTED LIVING"

Responsabile scientifico ENEA: Dott. Andrea Zanela

Responsabile scientifico Università di Roma Tor Vergata: Prof. Vincenzo Bonaiuto

Indice

SOMMARIO	4
1 INTRODUZIONE	5
1.1 NAO6 – ALDEBARAN – SOFTBANK ROBOTICS CORP.....	5
1.1.1 NAO6 – SPECIFICHE TECNICHE.....	5
1.1.2 NAO6 – PROGRAMMAZIONE DEL ROBOT	7
1.1.2.1 NAOQi	7
1.1.2.2 CHOREGRAPHE	8
1.1.2.3 MONITOR	8
1.1.2.4 SIMULATORE SDK.....	9
1.2 NAVIGAZIONE AUTONOMA E SENSORI DI TIPO “RANGE FINDER”	9
1.2.1 LASER RANGE FINDER URG-04LX-UG01 (HOKUYO) – CARATTERISTICHE TECNICHE	10
1.3 INTERFACCIA SOFTWARE DEL SENSORE LASER: LIBRERIA NAOQi ALDEBARAN	12
1.3.1.1 ALPROXY.....	12
1.3.1.2 ALMEMORY	12
1.3.1.3 ALLASER	12
1.3.1.3.1 GESTIONE DELLE API NAOQi MEDIANTE SCRIPT PYTHON	12
1.4 CARATTERIZZAZIONE DEL COLLEGAMENTO NAO – SENSORE LASER LRF URG-04LX-UG01	14
1.4.1 TEST DI “PACKET LOSSES”	15
1.4.2 TEST SU STABILITÀ DELLA MISURA	18
1.5 PROGETTO DEL SUPPORTO DEL SENSORE LASER E DEI DRIVER SOFTWARE	20
1.5.1 PROGETTO E REALIZZAZIONE DELLA SCHEDA DI CARICA BATTERIA.....	20
1.5.2 PROGETTO MECCANICO DEL SUPPORTO DEL SENSORE LRF	22
1.5.3 CARATTERISTICHE FUNZIONALI DEL CASCO	23
1.5.4 CARATTERISTICHE PROGETTUALI DEL CASCO	25
1.5.5 CENTRO DI MASSA E INERZIA.	28
1.5.5.1 STUDIO MATRICE INERZIA DEL ROBOT E CALCOLO DELLA MATRICE MODIFICATA.....	29
1.5.6 PROGETTO DRIVER LRF	30
1.5.6.1 ROUTINE COLLEGAMENTO CON IL LASER.....	31
1.5.6.2 ROUTINE CONFIGURAZIONE LASER.....	31
1.5.6.3 ROUTINE ACQUISIZIONE USCITA LASER E SALVATAGGIO SU FILE	31
1.5.6.4 ROUTINE ACCENSIONE FASCIO LASER.....	32
1.5.6.5 ROUTINE SPEGNIMENTO FASCIO LASER	32
1.6 TEST DEL SISTEMA	33
1.6.1 TEST 1	33
1.6.2 TEST 2	38
1.6.3 TEST 3	41
2 CONCLUSIONI	46
3 RIFERIMENTI BIBLIOGRAFICI	47
4 ABBREVIAZIONI ED ACRONIMI	47
5 CURRICULUM SCIENTIFICO DEL GRUPPO DI LAVORO.....	48

Sommario

Le attività di ricerca svolte da parte del Dipartimento di Ingegneria Industriale dell'Università degli Studi di Roma "Tor Vergata" in questa prima annualità dell'accordo di collaborazione hanno riguardato l'integrazione di un sistema di "range finder" su un robot di tipo antropomorfo (NAO6 – SoftBank Robotics Corp, Japan) di proprietà di ENEA. Lo scopo dell'attività è quella di cercare di migliorare le capacità sensoriali e operative di tale robot nell'ambito di un più ampio progetto che riguarda l'utilizzo di tali sistemi in attività di "Assisted Living".

L'aggiunta di tale sensore permetterà al robot di ricostruire, integrando le nuove informazioni con quelle dei sensori già disponibili a bordo, lo scenario che lo circonda rendendo possibile il riconoscimento di nuovi oggetti e permettendogli di rilevare ogni ostacolo eventualmente presente sul suo cammino. L'integrazione delle nuove informazioni con quelle già a disposizione permetterà agli algoritmi di navigazione di operare in modo più efficiente, aumentando la sua autonomia di movimento all'interno della casa.

Il sensore è stato montato su un supporto appositamente progettato e posizionato sulla testa del robot e sono stati sviluppati i driver software per permetterne l'integrazione con il suo sistema di navigazione.

È possibile suddividere l'attività di ricerca in due parti:

1. Scelta del sensore e sua integrazione sul robot NAO.
 - a) Analisi dei dispositivi "range finder" disponibili sul mercato ed individuazione del dispositivo che presenti caratteristiche migliori in termini di più efficiente integrazione con il robot NAO.
 - b) Progetto meccanico per il posizionamento del sensore laser sul robot NAO
 - Disegno parti meccaniche
 - Valutazione della nuova distribuzione dei pesi sul robot.
 - c) Progetto elettrico per l'alimentazione del sensore e la trasmissione dati.
 - d) Progetto dei driver software per l'utilizzo del sensore laser con il sistema di navigazione del robot NAO.
2. Misure utili alla caratterizzazione del funzionamento del sensore laser installato sul robot NAO.

1 Introduzione

Le attività svolte in questo primo anno di attività hanno riguardato il progetto hardware (sia meccanico che elettrico) e software per l'integrazione di un sensore laser per la misura delle distanze sul robot NAO (SoftBank Robotics Corp, Japan). Lo studio ha lo scopo di migliorare le prestazioni del robot, già dotato di sensori video, di contatto e a ultrasuoni, nelle attività di navigazione autonoma in ambiente non strutturato.

L'attività è partita dalla ricerca sul mercato del prodotto migliore, in termini di prestazioni, ingombri e peso, di un sensore laser in grado di essere installato a bordo del robot. Successivamente all'individuazione del componente più adatto, è iniziata la progettazione meccanica di un supporto per tale sensore e che ha portato alla realizzazione di un casco indossabile dal robot.

L'oggetto progettato e, successivamente realizzato, è in grado di alloggiare il sensore laser e di orientarlo opportunamente. Lo studio degli assorbimenti elettrici del sensore laser ha reso necessario l'utilizzo per il sensore di una batteria aggiuntiva ed il progetto di una scheda elettrica per la sua ricarica e per la regolazione della tensione per adeguarla alla tensione di alimentazione del sensore laser. Una volta terminata la realizzazione meccanica del casco supporto del sensore, è iniziato il progetto del driver software per la sua gestione.

La presente relazione è organizzata secondo i seguenti paragrafi.

- Robot Nao6
 - Descrizione hardware
 - Descrizione software
- Laser Range Finder URG-04LX-UG01
- Progetto meccanico casco supporto sensore laser
- Progetto elettrico scheda alimentazione del sensore laser
- Progetto del driver software per la gestione del sensore laser
- Test e misure

1.1 NAO6 – Aldebaran – SoftBank Robotics Corp

Il NAO è un robot umanoide (o antropomorfo), cioè costruito per imitare il movimento e l'interazione umana, commercialmente disponibile e tra i più utilizzati in ambito della ricerca. È ampiamente utilizzato per sviluppare algoritmi innovativi di robotica sociale e creativa. L'azienda francese Aldebaran (ora SoftBank Robotics Corp) lo ha introdotto nel 2004 e sono disponibili diversi modelli che differiscono per struttura (alcuni sono a corpo intero mentre altri sono composti dalle sole parti superiori del corpo) e per numero di gradi di libertà (21 o 25 DOF – Degree Of Freedom). È una piattaforma robotica compatibile con lo standard ROS (Robot Operating System) e questo la rende particolarmente versatile e utile nel campo della ricerca scientifica. Può essere considerato, al momento, il robot umanoide più evoluto disponibile in commercio in grado di esprimersi in lingua italiana. Grazie alle sue caratteristiche, NAO può essere impiegato in diversi ambiti e, in particolare, sono disponibili in letteratura numerosi studi nei quali questo robot è stato fatto interagire con ottimi risultati con ragazzi autistici, con anziani o disabili.

1.1.1 NAO6 – Specifiche tecniche

In questo studio, è stata utilizzata la versione 6 disponibile del modello H25 che pesa 5.5 kg per 58 cm di altezza. È provvisto di due telecamere (5-megapixel OmniVision risoluzione 2560 x 1920) posizionate nella testa all'altezza degli occhi e, per consentire l'interazione con l'utente mediante comandi vocali di quattro

microfoni direzionali posizionati sulla testa, un sintetizzatore vocale e due altoparlanti ad alta fedeltà all'altezza delle orecchie. A tale scopo, il robot ha installato a bordo un sistema di riconoscimento vocale e di dialogo disponibile in venti lingue diverse.

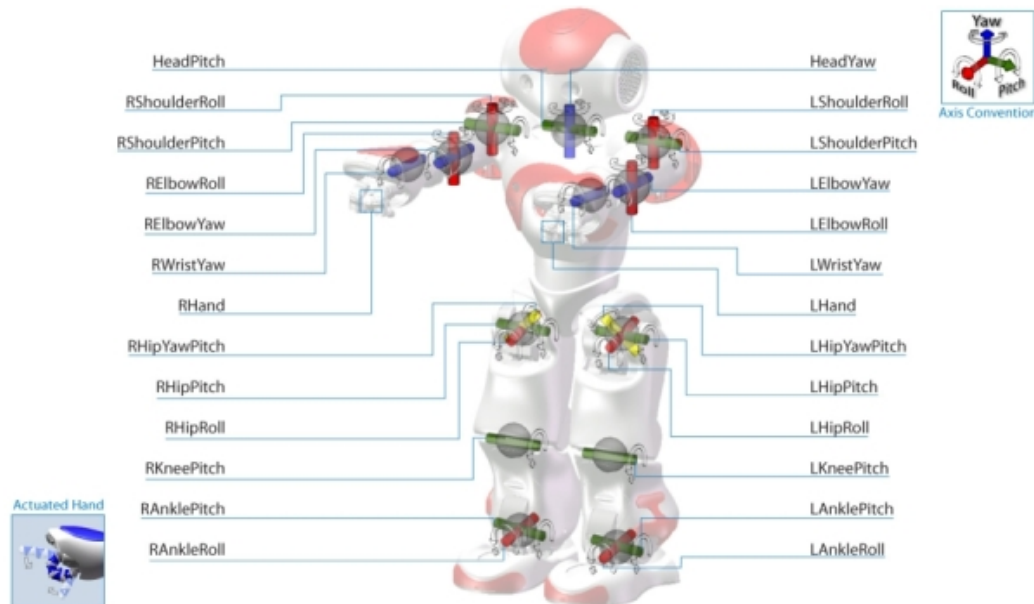


Figura 1 - NAO Joints and Actuators

Il robot è inoltre equipaggiato di numerosi altri sensori che gli consentono di ricevere feedback dall'ambiente che lo circonda. In particolare, sono presenti quattro sensori di pressione di tipo FSR (Force Sensitive Resistor) nella parte inferiore di ciascun piede per la misura della pressione applicata su di essi. Un'unità di misura inerziale (IMU – Inertial Measurement Unit) a sei gradi di libertà è posizionata al centro del tronco mentre, per la rilevazione a distanza degli ostacoli, il robot è dotato di due dispositivi ad ultrasuoni posizionati nella parte anteriore del busto a circa metà altezza. Per quello che riguarda invece i sensori di contatto posizionati sulla superficie del robot, sono presenti, per rilevare eventuali tocchi, tre sensori tattili di tipo capacitivo sia sulla testa che su ogni mano, mentre sulla parte anteriore ciascun piede, sono presenti due sensori di contatto di tipo bumper per la rilevazione delle collisioni.

Inoltre, per ciascuno dei giunti meccanici (visibili in Figura 1), il robot dispone di sensori per il controllo della posizione e degli assorbimenti elettrici. Per quello che riguarda l'elettronica a bordo necessaria per il controllo dei motori, le comunicazioni e i servizi accessori (es. riconoscimento vocale, ect.), il robot NAO6 utilizza due processori Intel quad-core Atom E3845 a 1.91 GHz in grado di indirizzare fino ad un massimo di 8 GB di memoria.

Il sistema software è basato su sistema operativo Linux Built-in (versione Gentoo). È inoltre equipaggiato con una scheda di rete che gli permette di comunicare con l'esterno mediante protocollo Ethernet (la porta è posizionata sulla parte posteriore della testa) ed un trasmettitore Wi-Fi a cui si aggiungono due ricetrasmittitori ad infrarossi posizionati sulla testa all'altezza degli occhi.

Elemento del Robot	DoF
Testa	2
Braccia	5 (x2)
Bacino	1
Gambe	5 (x2)
Mani	1 (x2)

Tabella 1: Gradi di libertà dei componenti del robot NAO

Dal punto di vista meccanico è un robot a 25 gradi di libertà come riportato in Tabella 1 e utilizza, per il movimento dei giunti, quattro tipologie di motori ciascuno dei quali presenta un differente valore di coppia (da 4.9 mNm¹ a 16.1 mNm). La batteria (21.6V/2.9Ah – LiPo – Lithium Polymer) consente un tempo di operatività che può variare dai 45 ai 120 minuti in funzione delle attività svolte.

Maggiori dettagli tecnici sulla piattaforma NAO sono disponibili su [1].

1.1.2 NAO6 – Programmazione del robot

La piattaforma NAO può essere programmata in diversi modi in funzione della specifica applicazione e delle capacità di programmazione dell'utente. In particolare, è disponibile una suite software dedicata a programmatori principianti che preferiscono un ambiente visuale senza la necessità di scrivere codice. Inoltre, il robot può essere programmato per eseguire applicazioni complesse utilizzando il framework NAOqi.

Sono inoltre disponibili una serie di simulatori nel caso in cui il robot non fosse disponibile o per effettuare test su specifici comportamenti.

1.1.2.1 NAOqi

NAOqi è il sistema operativo in esecuzione sul robot NAO e rappresenta il framework all'interno del quale è possibile interagire con i vari driver che controllano le diverse funzioni del NAO come leggere le uscite dei sensori ambientali del robot o programmarne i diversi comportamenti. Il framework risponde alle esigenze più comuni di un ambiente di programmazione robotico quale il parallelismo, la messa a disposizione delle risorse della macchina, la possibile sincronizzazione delle routine e la gestione degli eventi. NAOqi è un sistema multiplatforma, che significa che può essere utilizzato su diversi sistemi operativi come Windows, Linux e Mac OS. Inoltre, NAOqi è un sistema di tipo cross-language che consente di creare programmi in grado di girare direttamente sul robot sia utilizzando il linguaggio C++ che il Python. Il framework consente inoltre di creare programmi che possono essere eseguiti in remoto su di un computer esterno o localmente sul processore a bordo del robot. Consente inoltre ai diversi moduli del robot (come quelli relativi al movimento, all'audio e al video) di comunicare fra loro. Il sistema NAOqi viene fornito con un elenco di moduli principali e tutte le apposite interfacce di programmazione (API – Application Programming Interface) necessarie per programmare il NAO. Sono disponibili diversi kit di sviluppo software (SDK – Software Development Kit) per diversi linguaggi di programmazione, che includono Python, C++, Java, MATLAB e .Net.

Il SDK mette a disposizione una serie di API e strumenti di programmazione che supportano il programmatore nella gestione delle diverse procedure utili ad eseguire le varie attività del robot. Tuttavia, solo C++ e Python possono essere utilizzati per eseguire codice direttamente sul robot mentre altri linguaggi possono essere utilizzati solo in modalità di emulazione. L'utilizzo del linguaggio Python risulta più immediato in quanto essendo un linguaggio di tipo interpretato non richiede la compilazione del codice su un computer esterno ma può essere gestito direttamente dal processore a bordo del robot a differenza di quanto accade con il C++ dove il codice dovrà essere prima compilato e solo successivamente inviato al processore del robot per essere eseguito.

Esistono due metodi per programmare il NAO con la codifica. Il primo consiste nell'usare le API del framework NAOqi per chiamare uno qualsiasi dei moduli disponibili e organizzarlo nella struttura desiderata. Questo metodo richiede una conoscenza di programmazione comune. Il secondo modo è scrivere tutto il codice interamente da zero, creando librerie e moduli individuali, nonché un livello di astrazione hardware (HAL - Hardware Abstraction Layer) che comunica con NAOqi. Il metodo successivo presenta molti vantaggi, come una completa conoscenza e controllo dei moduli in contrasto con il primo approccio, in cui i moduli sono solo scatole nere. Il sistema operativo NAOqi viene fornito con un elenco di

¹ mNm milli Newton per metro

moduli principali già integrati all'interno del sistema ciascuno dei quali presenta un elenco di metodi predefiniti mentre è possibile installare appositi plugin per utilizzare specifici componenti esterni (es. Laser, camera, etc).

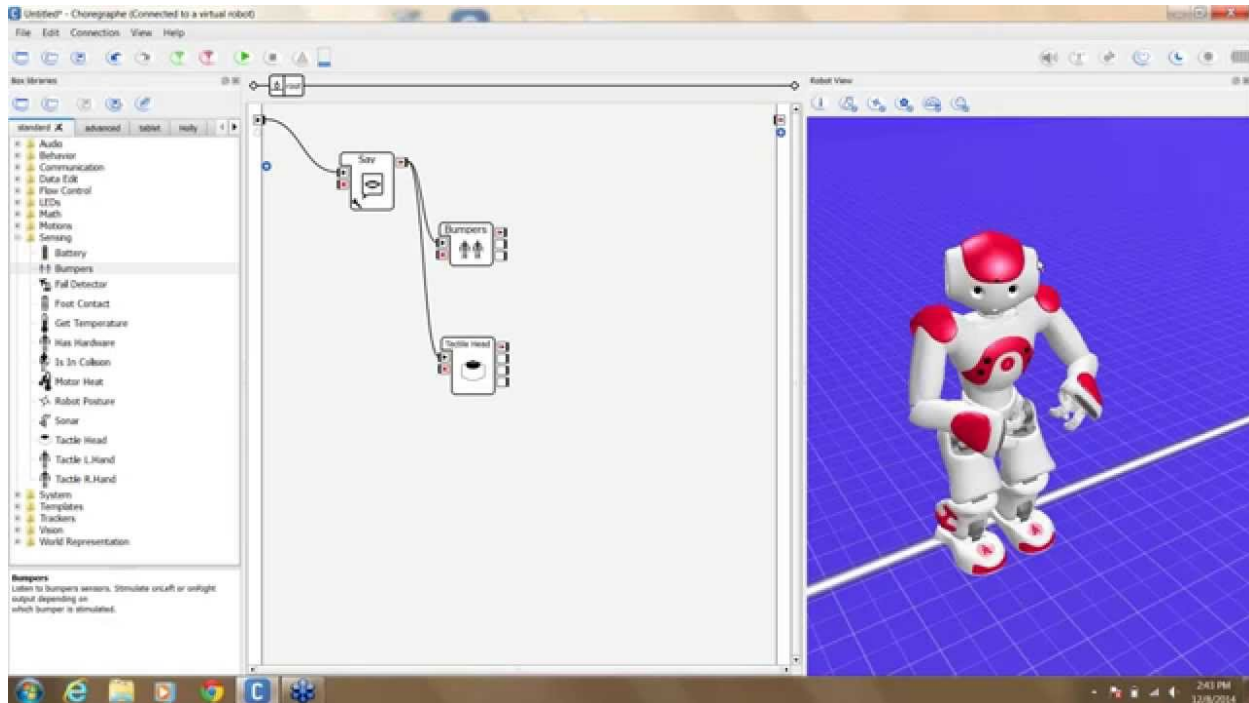


Figura 2 – Esempio di schermata del software Choregraphe

NAO è inoltre compatibile con Microsoft Robotics Studio, Cyberbotics Webots e Gostai Studio (URBI - Universal Real-Time Behavior Interface).

1.1.2.2 Choregraphe

Il software Choregraphe fa parte della suite di Aldebaran. È una GUI (Graphical User Interface) *drag and drop*, che consente di utilizzare una sequenza di moduli pre-programmati per creare animazioni, comportamenti e finestre di dialogo. In Figura 2 è mostrata la schermata principale del software.

Questi moduli possono essere organizzati in qualsiasi struttura combinatoria o sequenziale a piacere e possono essere modificati utilizzando il linguaggio Python. Il modulo Choregraphe può essere utilizzato sia collegato direttamente al robot che ad un simulatore attraverso un router, un indirizzo IP (Internet Protocol) ed una porta di rete. Il software Choregraphe è uno strumento molto versatile e potente in quanto consente, facendo uso dei moduli pre-programmati, di creare nuove applicazioni anche da parte di programmatori con poca esperienza. Il software consente infatti di realizzare applicazioni con all'interno dialoghi, servizi e comportamenti specifici con strumenti di tipo visuale e senza la necessità di dover scrivere una sola riga di codice.

1.1.2.3 Monitor

Il Monitor è un altro elemento software che fa parte della suite di Aldebaran e che consente all'utente di visualizzare il comportamento del robot in tempo reale. Con modalità analoghe al software Choregraphe, si può collegare al robot o al simulatore utilizzando un indirizzo IP ed una porta di rete del router. Come mostrato nella Figura 3a, il Monitor è in grado di fornire un feedback elementare dal robot, consentendo

all'utente di leggere direttamente i valori dei sensori, di visualizzare le immagini acquisite dalla telecamera e di leggere tutte le sezioni della memoria del robot durante l'esecuzione di qualsiasi azione eseguita da esso. È uno strumento eccellente per eseguire, in tempo reale, il debug dei programmi e osservare, a basso livello, lo stato del robot e di valutarne le prestazioni. Utilizzando appositi plugin (es. tra gli altri *3D Sensor Monitor*, *Camera viewer*, *Laser monitor*, etc.) permette la lettura delle varie zone di memoria e l'opportuna visualizzazione delle acquisizioni effettuate. Per esempio il *3D Sensor Monitor* può essere utilizzato per visualizzare l'immagine di profondità ottenuta da una telecamera 3D esterna collegata al robot, mentre il *Camera viewer* permette di controllare quello che viene ripreso dagli occhi del robot nonché regolare le impostazioni della fotocamera (es. luminosità, contrasto, etc.). Il plugin *Laser monitor* (Figura 3b), utilizzato nel corso delle attività oggetto della presente relazione per verificare l'uscita del sensore laser LRF (Laser Range Finder) per la caratterizzazione del dispositivo, permette la visualizzazione dell'uscita del sensore laser su un diagramma di tipo polare.

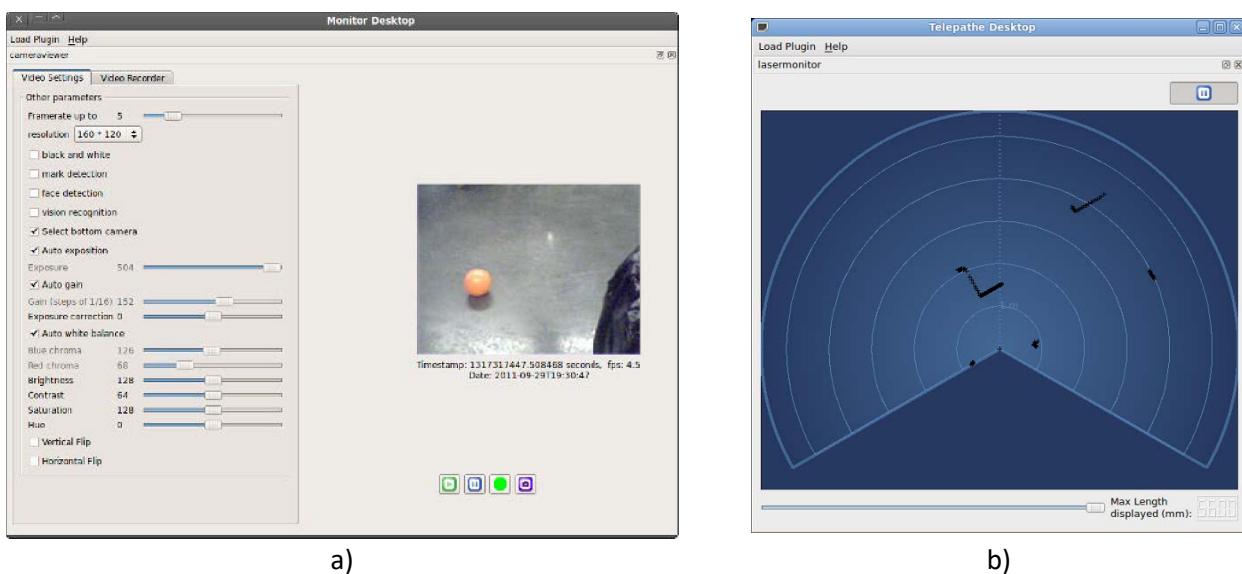


Figura 3 – Esempio di schermata del software Monitor: a) Camera Viewer b) plugin Laser Monitor

1.1.2.4 Simulatore SDK

È inoltre disponibile un pacchetto SDK (Software Development Kit) del simulatore che consente agli utenti di effettuare simulazioni di uno qualsiasi dei robot prodotti da Aldebaran all'interno di un simulatore 3D. A differenza di quanto accade quando si va a collegare l'eseguibile NAOqi ad un simulatore dove non è possibile effettuare la lettura dei parametri hardware del robot come ad esempio i valori delle grandezze lette dai sensori oppure i valori delle correnti in ingresso ai motori (queste letture produrranno dei valori nulli poiché non è effettivamente in esecuzione alcun sistema hardware). Utilizzando un apposito pacchetto SDK, fornito insieme al simulatore, è possibile per gli sviluppatori utilizzare tutte le API NAOqi disponibili per la programmazione del robot simulato e ottenere una stima dei parametri hardware del robot effettuando la lettura dei valori simulati in uscita dai diversi sensori ed i valori della stima delle correnti di alimentazione dei diversi motori.

1.2 Navigazione autonoma e sensori di tipo “range finder”

Esistono diversi elementi critici che riguardano la navigazione autonoma di un robot e, tra questi, è possibile sicuramente individuare i problemi di localizzazione della macchina e di mappatura delle zone circostanti. Queste operazioni sono generalmente realizzate mediante specifici algoritmi noti come SLAM (Simultaneous Localization And Mapping) [2]. A questo proposito, mentre le altre tipologie di sensori, tra quelli disponibili a bordo del robot, possono essere utilizzate per le procedure di misura (es. misura della

distanza da un ostacolo mediante i sensori ad ultrasuoni) e per la descrizione dell'ambiente, per la fusione del modello dinamico del robot con tali dati ambientali, è necessaria la determinazione diretta delle distanze dagli ostacoli circostanti per gestire le diverse procedure di *obstacle avoidance* [3]. Inoltre, è necessario considerare che la maggior parte degli algoritmi SLAM disponibili in letteratura è basata essenzialmente su misure ottenute mediante scansioni con telemetri laser.

Il robot NAO è dotato di un sensore di tipo sonar e che viene effettivamente utilizzato per effettuare procedure di *obstacle avoidance* ma, a causa del limitato raggio di azione e lunghi tempi di risposta del sensore, consente performance ridotte e risulta essere poco adatto a fornire informazioni utili ad algoritmi di SLAM. È stato quindi stabilito di affiancargli un telemetro laser.

Esistono due principali famiglie di telemetri laser (Laser Range Finder – LRF) che differiscono per il principio di misura utilizzato.

- Misura del tempo di volo
- Misura di sfasamento

La prima delle due famiglie è rappresentata da sistemi in cui misura di distanza è ricavata sulla base della misura del tempo di volo. È questa una classe di sensori che presenta le migliori performance sia in termini di precisione che di affidabilità ma che presenta, nella maggior parte dei casi, un ingombro e un peso tali da non rendere consigliabile il loro utilizzo su macchine di piccola taglia. Questa famiglia di sensori è invece comunemente utilizzata su robot che presentano dimensioni maggiori e che sono pensati, principalmente, per un uso di tipo outdoor.

La seconda famiglia di sensori è invece caratterizzata da tecniche per la misura della distanza che sono ricavate sulla base di misure di sfasamento. Questa seconda famiglia di sensori, presenta una struttura generalmente molto più compatta rispetto alla precedente e ha, oltre ad un consumo energetico notevolmente più basso, peso e dimensioni ridotti che la rende molto più adatta per essere utilizzata a bordo di robot di taglia inferiore principalmente destinati ad un utilizzo indoor.

1.2.1 Laser Range Finder URG-04LX-UG01 (Hokuyo) – caratteristiche tecniche

In seguito ad una ricerca effettuata sui dispositivi laser utilizzabili a bordo del robot NAO e disponibili in commercio, la scelta è caduta su di un sensore appartenente alla famiglia degli LRF basati su misure di sfasamento: LRF modello URG-04LX-UG01 (Hokuyo Automatic Co., Ltd – Giappone) (Figura. 4).

Si tratta, attualmente, del più diffuso LRF negli ambiti sia di ricerca e che industriali a causa dell'ottimo rapporto costo-prestazioni.

Le principali caratteristiche tecniche di tale sensore sono riportate in Tabella 2.



Figura 4 – Sensore laser LRF URG-04LX-UG01 (Hokuyo Automatic Co., Ltd - Japan)

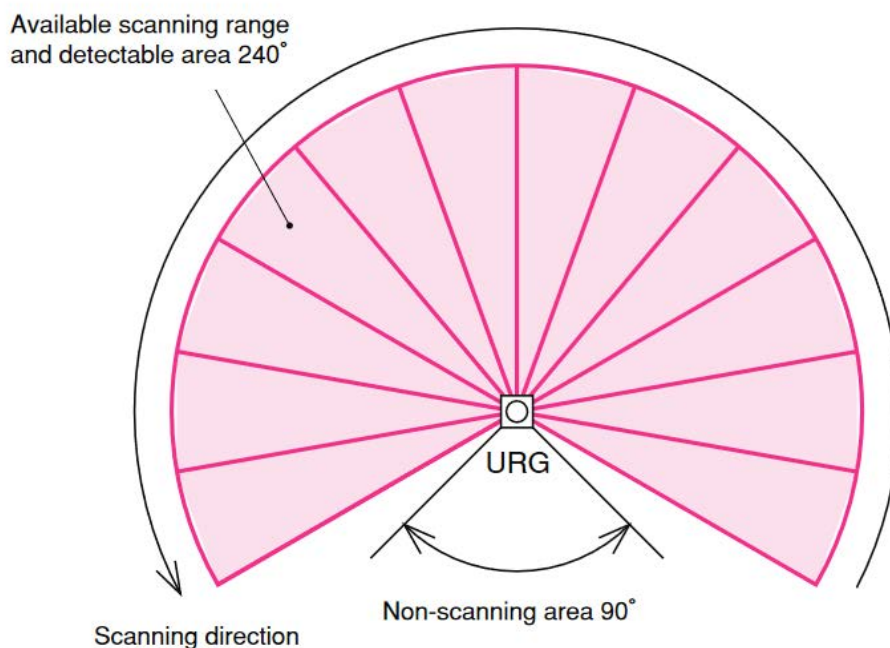


Figura 5 – Angolo massimo di visuale del laser LRF URG-04LX-UG01

Model No.	URG-04LX-UG01 (Hokuyo Automatic co., ltd)
Power source	5VDC±5%(USB Bus power)
Light source	Semiconductor laser diode ($\lambda=785\text{nm}$), Laser safety class 1
Measuring area	20 to 5600mm (white paper with 70mm×70mm), 240°
Accuracy	60 to 1,000mm : ±30 mm 1,000 to 4,095mm : ±3% of measurement
Angular resolution	Step angle : approx. 0.36°(360°/1,024 steps)
Scanning time	100ms/scan
Scanning angle	240°
Noise	25dB or less
Interface	USB2.0/1.1[Mini B](Full Speed)
Command System	SCIP Ver.2.0
Ambient illuminance	Halogen/mercury lamp: 10,000Lux or less, Florescent: 6000Lux(Max)
Sound level	25db or less (at 300mm)
Protective Structure	Optics : IP64 Case : IP40
Weight	Approx. 160g

Tabella 2: caratteristiche del sensore laser LRF URG-04LX-UG01 [4]

Il sensore laser LRF URG-04LX-UG01 è in grado di acquisire fino ad un angolo massimo pari a 240° ed un intervallo di acquisizione compreso tra 2 cm e i 5.6 m. Il sistema di comunicazione del sensore è basato sul protocollo standard SCIP2.0 (Secure Communications Interoperability Protocol) e il modulo può connettersi con il sistema host mediante protocollo URG-SCIP su un link USB (Universal Serial Bus) nativo (/dev/ttyACM0) oppure utilizzando la stessa porta USB configurata come link seriale (/dev/ttyUSB0).

All'avvio del sistema, questa porta seriale può essere configurata via software ad una velocità compresa fra i 19200 baud e i 115200 baud.

Il costruttore dichiara un consumo pari a 500 mA alla tensione di alimentazione pari a 5V. valore che rende possibile l'alimentazione del modulo mediante una batteria. Il peso del sensore laser dichiarato dal costruttore è pari a circa 160g. Considerando che questo dovrà essere montato sulla testa del robot, è possibile che questo peso aggiuntivo richieda una modifica della matrice inerziale del robot.

È un dispositivo il cui utilizzo è previsto per sole applicazioni indoor.

1.3 Interfaccia software del sensore laser: libreria NAOqi Aldebaran

La comunicazione tra il sensore laser e il sistema operativo NAOqi del robot avviene per mezzo di un file (altrimenti detto "script") scritto in linguaggio Python. Lo script utilizza la classe software ALProxy, che permette di interfacciarsi con i moduli ALLaser e ALMemory per leggere i dati acquisiti dal laser.

1.3.1.1 ALProxy

ALProxy è un oggetto che permette di accedere a tutti i metodi o al modulo a cui ci si vuole connettere secondo la seguente sintassi:

classe ALProxy (nome, ip, porta)

dove

name è il nome del modulo

ip è l'indirizzo IP del robot

port è la porta su cui il sistema operativo NAOqi è in ascolto (9559 per impostazione predefinita)

Ad esempio la seguente istruzione permette di stabilire la connessione con il robot all'indirizzo IP specificato dalla variabile *args.ip* alla porta indicata dalla variabile *args.port*

```
session.connect("tcp://" + args.ip + ":" + str(args.port))
```

1.3.1.2 ALMemory

Permette l'accesso alla quella sezione della memoria centrale del robot che viene utilizzata per memorizzare tutte le informazioni chiave relative alla configurazione hardware della macchina. In particolare, il modulo *ALMemory* permette di leggere dalla memoria del robot tutte le informazioni sullo stato corrente degli attuatori e dei sensori (mostrati in Figura 1).

1.3.1.3 ALLaser

Il modulo *ALLaser* permette di accedere ai dati inviati dalla testa laser che vengono memorizzati in una apposita variabile del modulo *ALMemory* denominata: "*Device/Laser/Value*". Le distanze misurate dal sensore laser possono successivamente essere proiettate come punti incrociati su di un grafico di tipo polare con il corretto rapporto angolare e offset.

1.3.1.3.1 Gestione delle API NaoQi mediante script Python

Mediante la libreria *ALLaser* è possibile effettuare alcune configurazioni ed inviare specifici comandi al sensore laser.

Ad esempio è possibile selezionare l'angolo di apertura del laser cioè l'ampiezza dell'angolo all'interno del quale il laser dovrà effettuare la rilevazione della presenza di ostacoli. Nell'esempio seguente il sensore è impostato per effettuare una scansione in un intervallo compreso tra i -90° e i 90° .

```
laser_service = session.service("ALLaser")

# Set the opening angle at -90/90 degrees
laser_service.setOpeningAngle(-1.570796327, 1.570796327)
```

L'angolo di apertura è compreso nell'intervallo -2.35620 rad (-135°) - 2.09235 rad (120°) e se non diversamente specificato è impostato al suo valore massimo (come è mostrato in figura 5).

È anche possibile selezionare le soglie (minima e massima) della distanza dell'oggetto rispetto al sensore laser all'interno delle quali deve essere effettuata la rilevazione. Nell'esempio seguente il sensore laser viene impostato per poter effettuare una scansione che vada a rilevare tutti gli oggetti che si trovano ad una distanza compresa tra i 50 cm e i 3.00 m rispetto alla testa del sensore laser.

```
laser_service = session.service("ALLaser")

# Set detection threshold at 500/3000 mm
laser_service.setDetectingLength(500, 3000)
```

La distanza di rilevamento del sensore è compresa nell'intervallo 20 mm – 5600 mm. Se non diversamente specificato, l'intervallo di misura è fissato al suo valore massimo.

La risoluzione angolare del laser è pari a 0.0063 rad (0.36°), una media cioè di circa 3 scansioni per ogni grado sessagesimale. Ciò significa che il sensore è in grado di effettuare, nell'intervallo compreso fra -120° e 120° (per un angolo di visuale complessivo pari a 240° - Figura 5), un numero massimo di 683 letture ciascuna delle quali è separata dalla successiva da un angolo pari a 0.36° .

Il sensore è inoltre in grado di effettuare tutte le 683 letture (angolo visuale completo) in un tempo massimo pari a 100 ms.

I seguenti comandi permettono l'accensione e lo spegnimento del laser

`ALLaserProxy::laserOFF()` – Disabilita il fascio del sensore laser mantenendo in movimento la testa del laser che continua a girare non effettuando però le misure.

`ALLaserProxy::laserON()` – Abilita il fascio del sensore laser e, di conseguenza, abilita la funzione di rilevamento.

I dati memorizzati nella variabile `Device/Laser/Value` sono nella forma di un array bidimensionale di dimensioni pari a $n \times 4$, dove il valore n è proporzionale all'angolo di apertura e che deve essere impostato insieme agli altri parametri nel modulo `ALLaser`.

Per ogni indice i vengono scritti in memoria quattro valori differenti e che rappresentano:

```
urgdata [i] [0] = coordinata polari: distanza
urgdata [i] [1] = coordinate polari: angolo
urgdata [i] [2] = coordinate cartesiane: x
urgdata [i] [3] = coordinate cartesiane: y
```

dove distanza e angolo sono la posizione dell'ostacolo espressa in coordinate polari con "distanza" che rappresenta la distanza misurata tra il laser e lo specifico ostacolo e l'angolo che presenta un valore calcolato in base all'indice e alla configurazione del dispositivo, mentre x e y sono i valori delle coordinate

dell'ostacolo nello spazio cartesiano (x: ascissa e y: ordinata). In figura 6 è mostrata la definizione dell'angolo di apertura del sensore.

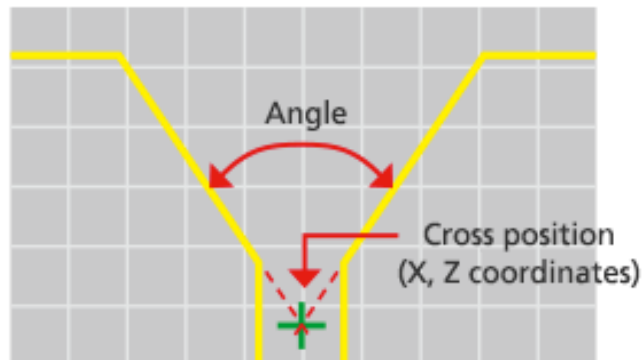


Figura 6 - Definizione dell'angolo di apertura del sensore LRF URG-04LX-UG01

1.4 Caratterizzazione del collegamento NAO – Sensore laser LRF URG-04LX-UG01

Per caratterizzare il collegamento tra il sistema software a bordo del robot NAO ed il sensore laser LRF URG-04LX-UG01 e verificare:

- la presenza di un'eventuale perdita di pacchetti di dati durante la trasmissione,
- la stabilità del dato acquisito per la misura di una stessa distanza

Il laser è stato quindi sottoposto ad alcuni test di lettura ripetuti ponendo il robot con il laser montato sul supporto sopra la testa ad alcune distanze rispetto ad un ostacolo (in particolare è stato posto di fronte ad un muro) ed è stata impostata, mediante apposito script (Figura 7), una procedura di lettura del sensore in grado di effettuare cento letture consecutive, per angolo di apertura e soglie minima e massima di rilevazione del laser impostati sia ai rispettivi valori di default che opportunamente variati.

```
#!/usr/bin/env python
import sys
from naoqi import ALProxy

ip = "192.168.1.116"
file = sys.stdout

laser = ALProxy("ALLaser", ip, 9559)
memo = ALProxy("ALMemory", ip, 9559)
i = 0

import datetime
now = datetime.datetime.now()

fileName = "LaserComTest.txt"
fileWrite = open(fileName, "w")

fileWrite.write("=====\n")
```

```

fileWrite.write("                                " + now.strftime("%Y-%m-%d %H:%M:%S"))
fileWrite.write("\n")
fileWrite.write("=====\n")

while i < 250:
    L = memo.getData("Device/Laser/Value")

    fileWrite.write(str(i) + " ")

    fileWrite.writelines(str(L))

    fileWrite.writelines("\n")

    i = i + 1

fileWrite.write("=====")
fileWrite.close()

```

Figura 7 – Esempio di Script Python per la lettura dei dati del sensore LRF

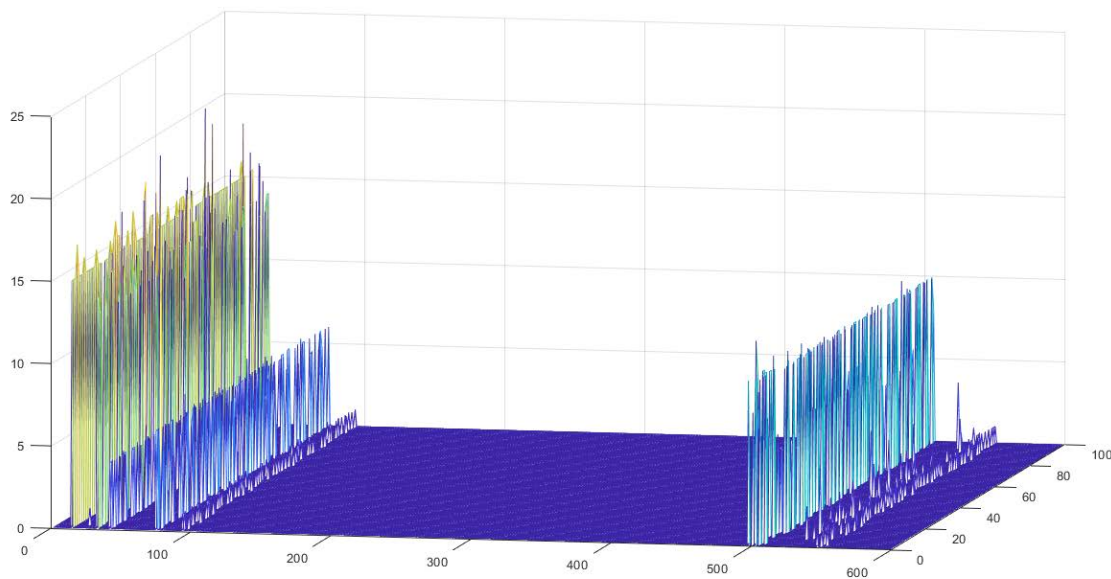
1.4.1 Test di “packet losses”

Dall’analisi effettuata sui dati acquisiti secondo la procedura descritta, si è osservato che la lettura dei dati rilevati dal sensore laser avviene correttamente, nell’analisi effettuata su 3500 letture, per il 71% dei casi come mostrato in Tabella 3.

Eventi di <i>Packet Loss</i>	N	%
None	2489	71.1%
1	6	0.17%
2-3	4	0.11%
5-10	1	0.03%
> 10	1000	28.6%

Tabella 3: Packet loss nella lettura della memoria dati scritta dal sensore LRF URG-04LX-UG01 (analisi effettuata su 3500 letture della zona di memoria)

Da tale tabella è possibile constatare come, nel circa 30% delle acquisizioni, si sia riscontrata una perdita casuale di un certo numero di pacchetti come mostrato nella Figura 9 e, in dettaglio, nella figura 8 dove è mostrato il numero degli eventi di perdita dati e l’ampiezza di tali “buchi” (asse z) nella ricezione dati in funzione dell’angolo di apertura (asse x) e del numero dell’acquisizione effettuata (asse y). Dalla figura 7 si può osservare come, nella specifica acquisizione, il fenomeno si sia verificato più o meno nelle stesse posizioni e soprattutto ai lati del diagramma di acquisizione. All’interno dell’angolo 35° – 175° non è infatti presente alcuna perdita di dati mentre all’esterno di questo intervallo si possono notare dei picchi che rappresentano il numero pacchetti non ricevuti che oscilla da minimo di 1 pacchetto fino ad un massimo di 21 per valori dell’angolo di apertura inferiori ai 35° mentre la perdita risulta essere più contenuta (il numero massimo di pacchetti persi si attesta intorno ai 10 pacchetti) per angoli superiori ai 175°.

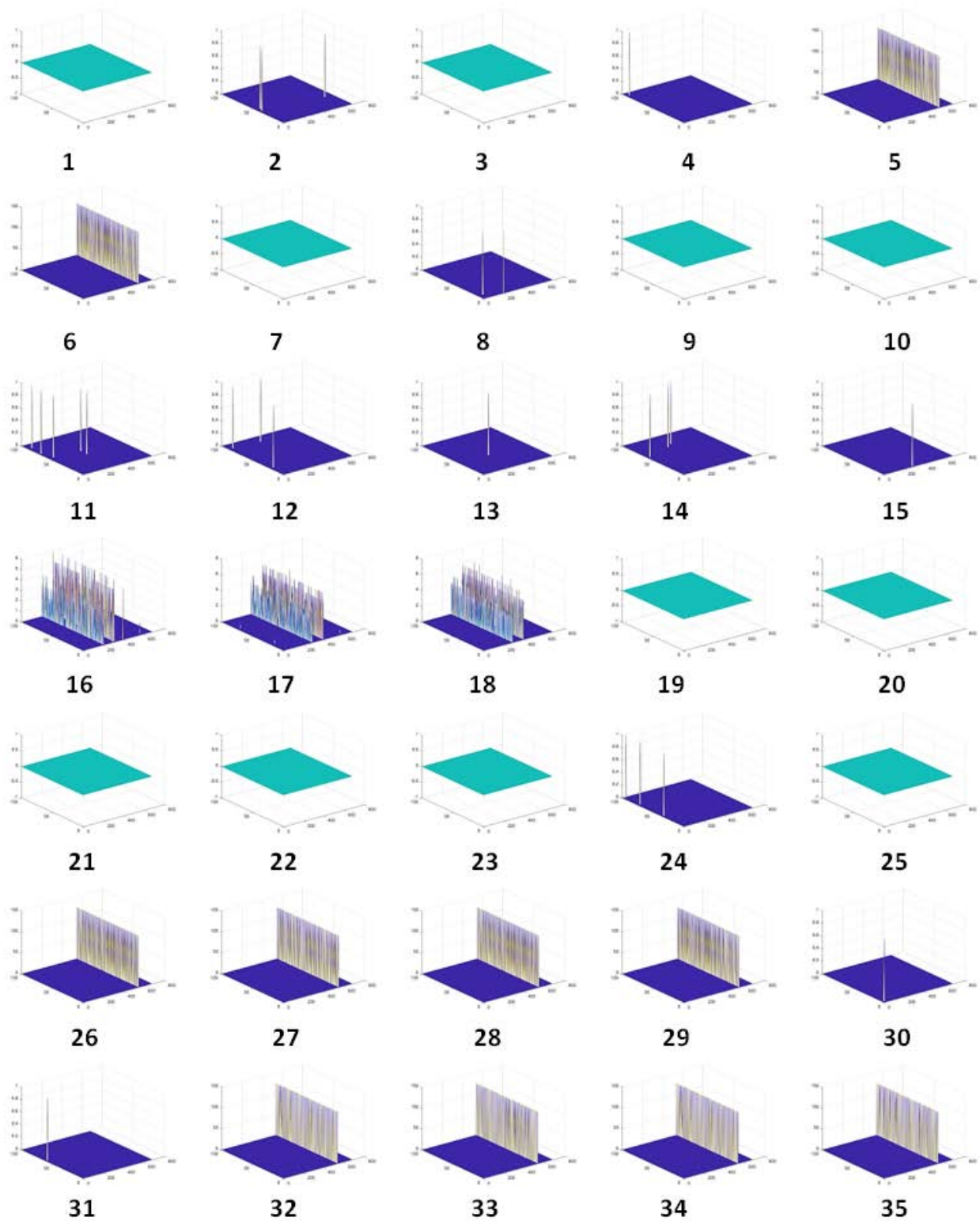


**Figura 8 – Numero pacchetti non ricevuti dal sistema in funzione dell’angolo di apertura
(test realizzato su 100 acquisizioni)**

Il problema sembra essere dovuto al conflitto che si viene a verificare fra le procedure scrittura (da parte del laser) e di lettura (lo script da noi realizzato) della zona di memoria “Device/Laser/Value”. Tale problema risulta inevitabile a causa della non possibile sincronizzazione delle due procedure software non essendo disponibile sul sensore laser un segnale di trigger che segnali la fine della procedura di scansione e memorizzazione dati. Le funzioni distribuite da Aldebaran all’interno delle API NAOqi non sembrano gestire con successo il problema (che però non si presenta utilizzando il loro software Monitor). Dopo un’attenta analisi del comportamento del sistema laser-robot si è notato come lo spegnimento del laser del sensore prima di effettuare la lettura del dato memorizzato memoria sembra riuscire a risolvere, anche se in modo un po’ drastico, il problema. È da considerare però che tale modalità operativa comporta un costo in termini di prestazioni che deve essere tenuto in conto in quanto le due procedure di spegnimento e nuova accensione del dispositivo provoca un lieve ritardo nella procedura di acquisizione dei dati.

In percentuale del tutto trascurabile (complessivamente inferiore al 1% – sempre per 3500 eventi di lettura di tutta la zona di memoria) sono stati riscontrati eventi per i quali il sistema perde, in modo casuale, qualche pacchetto di trasmissione (numero di eventi misurato è compreso fra 1 e 5). Dall’analisi effettuata si è osservato che la maggior parte di questi è limitato ad un solo evento.

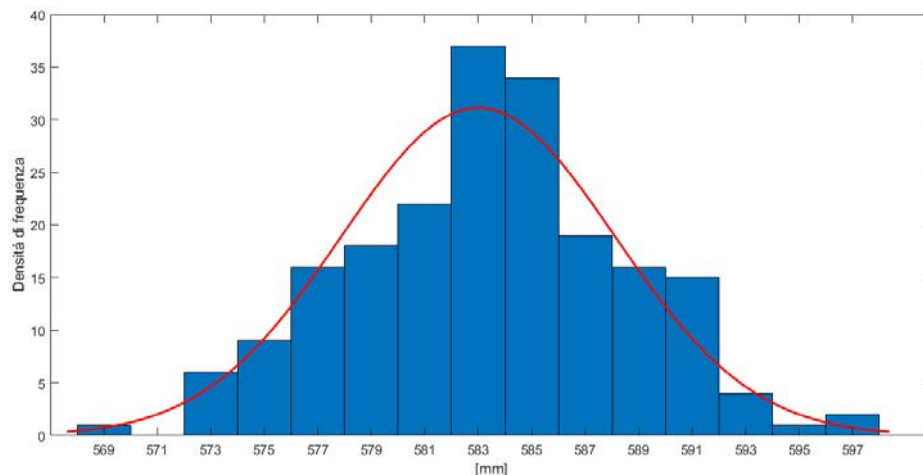
Per tenere conto di questo problema è stata realizzata una specifica procedura software che, sfruttando il valore della coordinata polare relativa all’angolo, è in grado individuare la posizione e il numero di pacchetti mancanti. Mediante una seconda procedura sarà inoltre possibile riallineare i dati acquisiti e recuperare il valore dei dati mancanti come media dei valori ricevuti in acquisizioni diverse (per la stessa coordinata angolare) oppure, in mancanza di queste, mediante metodi di interpolazione.



**Figura 9 – Numero pacchetti non ricevuti dal sistema in funzione dell'angolo di acquisizione
(ciascun test è stato realizzato su 100 acquisizioni)**

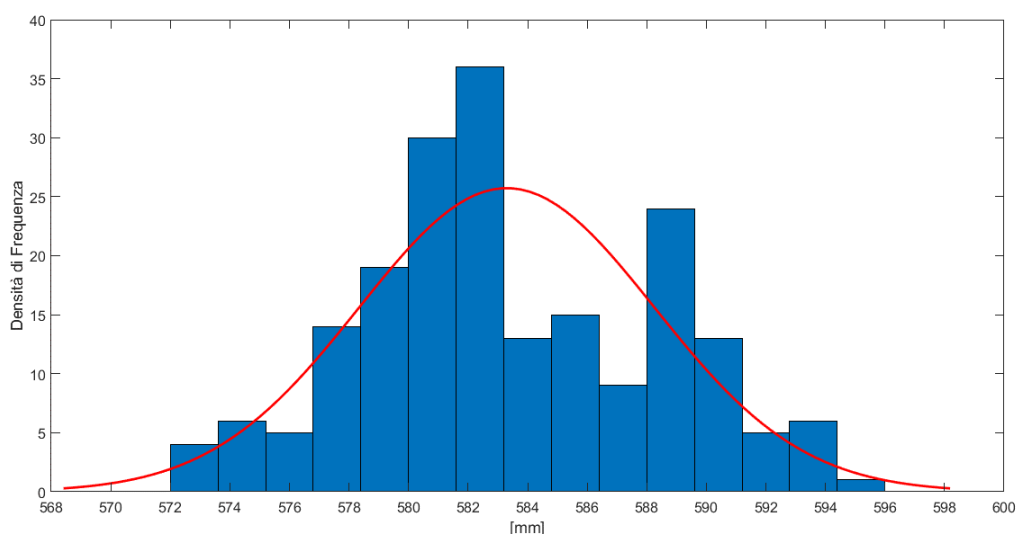
1.4.2 Test su stabilità della misura

Per poter caratterizzare il sensore laser rispetto alla stabilità delle misure rilevate con tale sistema, sono state effettuate alcune scansioni in cui il sistema sensore laser – robot è stato posizionato in un primo test alla distanza di 50 cm rispetto ad una parete (la misura è riferita all'estremità anteriore del piede sinistro del robot). il robot è stato quindi programmato per effettuare 200 scansioni consecutive che sono state effettuate avendo impostato sul laser un angolo di apertura pari a 240° (valore massimo – default). L'illuminazione dell'ambiente è stata realizzata mediante una singola lampada a fluorescenza da 60 W.



**Figura 10 – Misura di distanza a 50 cm dall'ostacolo²
(200 scansioni consecutive – angolo di apertura impostato a 240°)**

Il risultato di questo primo test (Test distanza #1) è riportato in figura 10 nella quale è mostrato l'istogramma relativo alla misura della distanza rilevata dal sensore laser per un angolo di scansione pari a 0° (direzione ortogonale al piano del robot). Il valore medio rilevato è di 58.3 cm con una deviazione standard pari a 5.13 mm. La misura rilevata dallo strumento ovviamente comprende anche la distanza fra la punta del piede del robot e il sensore laser che è pari a circa 8.5 cm. Il valore ottenuto (58.3 cm) è quindi assolutamente coerente con la misura della distanza sensore laser – parete (50 + 8.5 = 58.5 cm) e compatibile con le tolleranze nel posizionamento del robot e quelle dichiarate dal costruttore in questo range di misura (± 30 mm se < 1 m – Vedi Tabella 2).



**Figura 11 – Misura di distanza a 50 cm dall'ostacolo
(200 scansioni consecutive – angolo di apertura impostato a 90°)**

² La misura è riferita all'estremità anteriore del piede sinistro del robot

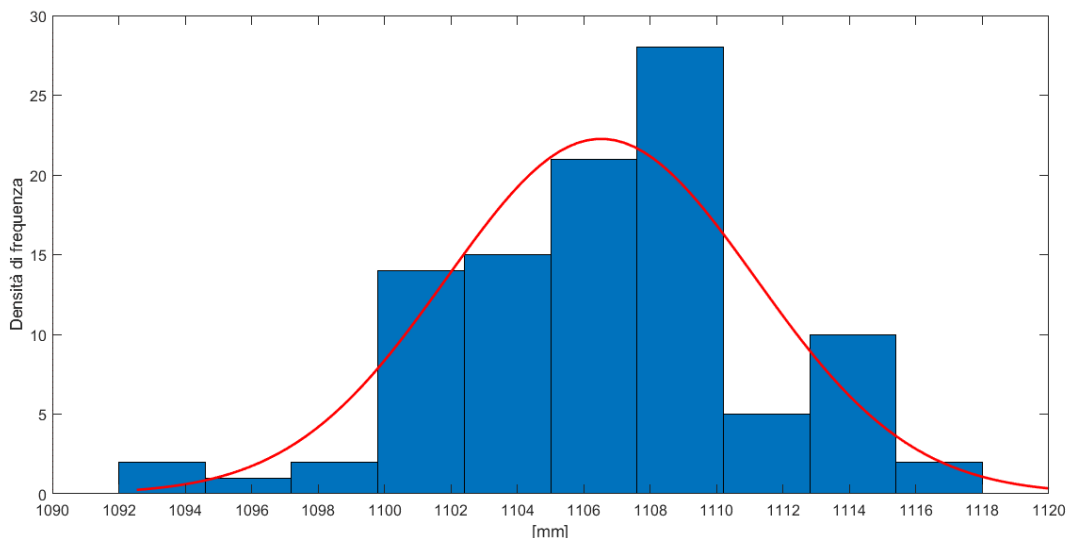
Il risultato di un secondo test (Test distanza #2) è mostrato in figura 11 dove è riportato l'istogramma delle misure realizzate con la stessa modalità del precedente test ma con un angolo di apertura impostato a 90°. Il valore medio rilevato è di 58.33 cm con una deviazione standard pari a 4.96 mm. Anche in questo caso il valore ottenuto con lo strumento è da considerarsi coerente con la misura della distanza sensore laser – parete.

1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4	3.1	3.2
597	580	585	587	577	579	579	579	1114	1101
584	577	579	582	585	591	582	582	1105	1103
583	579	581	582	578	593	584	584	1102	1109
583	588	587	581	583	588	584	584	1108	1104
583	581	577	588	591	583	578	578	1109	1113
579	575	587	582	587	582	583	583	1102	1108
575	581	581	589	591	579	572	572	1112	1099
585	572	581	592	590	580	581	581	1104	1113
584	585	587	588	585	589	582	582	1111	1102
589	576	576	579	575	583	579	579	1109	1103
578	582	584	585	584	583	578	578	1108	1102
580	582	573	580	581	581	587	587	1104	1115
578	587	590	590	579	585	588	588	1108	1108
587	583	583	590	582	590	589	589	1118	1109
591	584	578	585	592	574	583	583	1113	1099
588	587	583	576	572	587	578	578	1104	1105
587	575	580	588	591	582	585	585	1109	1109
583	583	579	588	595	585	578	578	1114	1110
588	578	589	584	579	579	585	585	1110	1105
596	584	573	585	580	588	583	583	1111	1105
589	584	587	584	583	579	581	581	1107	1101
580	578	581	587	585	576	589	589	1104	1103
590	584	577	576	585	580	588	588	1094	1108
581	588	577	593	580	591	593	593	1105	1104
588	590	577	585	576	593	574	574	1102	1106
587	578	572	585	584	582	585	585	1106	1112
584	579	583	572	580	590	594	594	1092	1114
578	582	593	580	589	584	583	583	1106	1105
579	593	581	587	588	592	587	587	1105	1114
577	582	582	582	591	580	588	588	1107	1110
578	587	580	584	582	578	588	588	1104	1111
582	576	574	584	581	581	584	584	1110	1109
590	580	576	585	584	581	589	589	1106	1103
587	583	584	584	584	587	579	579	1110	1107
587	591	577	585	581	582	576	576	1117	1101
591	581	582	585	574	572	584	584	1103	1108
575	577	587	590	585	589	585	585	1109	1114
575	583	583	577	583	583	589	589	1108	1106
583	590	575	581	583	578	584	584	1102	1097
589	568	583	595	587	580	585	585	1101	1108
583	576	585	590	589	582	588	588	1107	1104
585	583	590	575	587	578	580	580	1110	1106
585	580	587	583	587	584	579	579	1103	1101
582	575	583	578	589	580	588	588	1107	1106
584	590	589	583	582	590	581	581	1113	1110
579	578	583	573	580	581	579	579	1103	1102
583	585	585	589	592	582	590	590	1105	1100
582	583	583	584	578	579	574	574	1108	1109
585	587	581	581	583	577	583	583	1108	1105

Tabella 3: Misura della distanza dall'ostacolo rilevata (mm) (test #1, #2, #3)

In un terzo e ultimo test (Test Distanza #3, il robot è stato posizionato a 100 cm dalla stessa parete e, in figura 12, è mostrato l'istogramma relativo a 100 acquisizioni consecutive ottenute impostando l'angolo di apertura del laser al valore di 240°. Il valore medio rilevato è di 110.65 cm con una deviazione standard pari a 4.66 mm.

Anche in questo caso, la distanza misurata comprende anche la distanza piede – sensore laser di circa 8.5 cm. Il valore ottenuto dallo strumento è da considerarsi quindi coerente con la misura sensore laser-parete (108.5 cm) e compatibile con le tolleranze di posizionamento del robot e quelle dichiarate dal costruttore in questo range di misura ($\pm 3\%$ se > 1 m). In tabella 3 sono riportate tutte le misure ottenute dal sistema nelle tre prove effettuate.



**Figura 12 – Misura di distanza a 100 cm dall’ostacolo
(100 scansioni consecutive – angolo di apertura impostato a 240°)**

1.5 Progetto del supporto del sensore laser e dei driver software

Per poter essere posizionato sulla testa del robot NAO, il sensore laser LRF URG-04LX-UG01 necessita di un opportuno supporto meccanico che deve essere progettato e realizzato. Poiché inoltre tale sensore presenta assorbimenti elettrici non compatibili con quanto erogabile dalla porta USB posizionata sul retro della sua testa, si è ritenuto di dover utilizzare una batteria aggiuntiva. Questo ha richiesto il progetto e la relativa realizzazione di una scheda elettronica per la ricarica della batteria e la conversione statica della tensione per adeguarla a quella di funzionamento del sensore. Alla fine del paragrafo verrà descritta la libreria software progettata.

1.5.1 Progetto e realizzazione della scheda di carica batteria

Il sensore laser deve essere connesso alla porta USB del NAO tramite un cavo mini USB che provvede all’alimentazione e al collegamento dati. Poiché però il sensore richiede una corrente di alimentazione pari a circa 250 mA che la porta USB del NAO non è in grado di fornire, si è scelto di aggiungere una seconda batteria di supporto che verrà alloggiata, esternamente a robot, all’interno della struttura di supporto del LSR. Questa scheda di alimentazione è stata poi collegata al robot utilizzando un apposito cavo USB realizzato con doppio collegamento per garantire la corretta alimentazione al sensore.

La batteria utilizzata è di tipo LiPo (Polimeri di Litio) con tensione di uscita a 3.7 V con capacità di 1500 mAh e un peso di soli 20 g. Per quello che riguarda il relativo circuito di carica batteria, a causa delle diverse tensioni operative delle due batterie e soprattutto a causa della posizione molto distante fra le due (una è posizionata sulla testa e l’altra sul dorso del robot) si è ritenuto più conveniente progettare e realizzare una seconda scheda indipendente dai circuiti del robot.

Il circuito di carica batteria progettato è basato sul controller per la gestione della carica MCP73831T-2ACI/OT (Microchip – Giappone) [5] pensato per applicazioni portatili con carica da porta USB e conforme a tutte le specifiche relative alla carica di batterie sia a ioni che polimeri di litio.

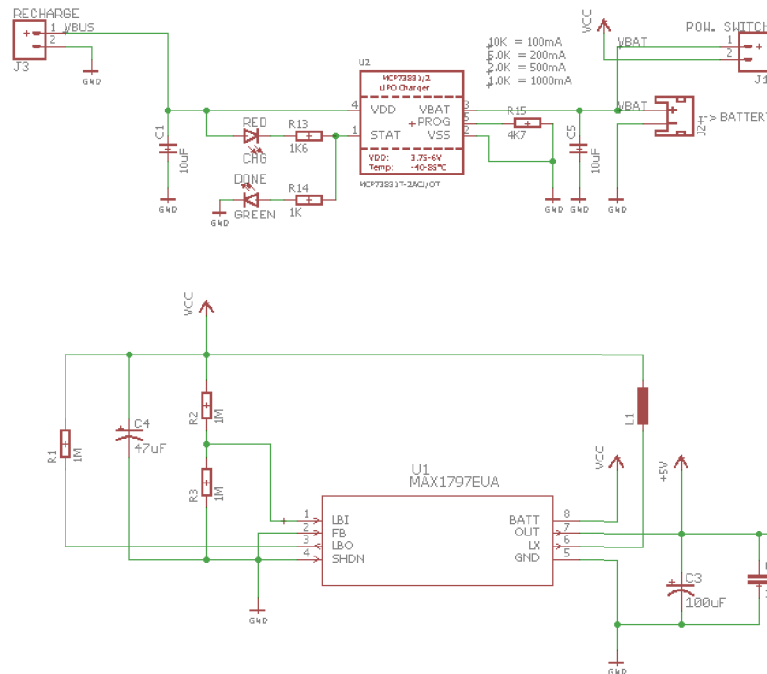


Figura 13 – Schema elettrico della scheda del carica batteria

Il controller utilizzato effettua la tipica procedura di carica utilizzata per queste tipologie di batterie che alterna periodi a corrente costante seguiti da altri a tensione costante utilizzando uno specifico algoritmo. Questo prevede la possibilità di selezionare opportune opzioni di pre-condizionamento e fine carica.

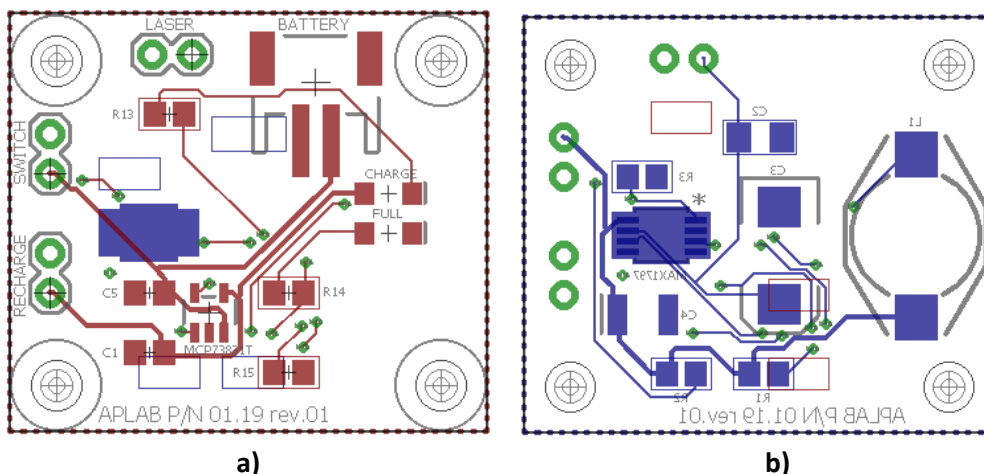


Figura 14 – Progetto PCB della scheda di ricarica della batteria del laser: vista a) fronte e b) retro

La scheda dispone di una coppia di led, uno rosso e uno verde, per indicare l'avanzamento dello stato di carica. Questa è inoltre completata da un convertitore di tensione DC-DC necessario per adeguare la tensione di uscita della batteria LiPo da 3.7V alla tensione di funzionamento a 5V del sensore. Il dispositivo utilizzato allo scopo è il MAX1797 (Maxim Integrated – USA) [6]: un convertitore di tensione tipo DC-DC step-up ad alta efficienza pensato per applicazioni all'interno di piccoli dispositivi portatili. La scheda comprende inoltre una presa micro USB per collegare la scheda alla presa USB del robot, un connettore per la batteria LiPo e un interruttore per scollegare la batteria dall'alimentazione del sensore.

In Figura 13 è mostrato lo schema elettrico di tale circuito, mentre in Figura 14 è mostrato il progetto del PCB (Printed Circuit Board) della stessa scheda (rispettivamente con vista frontale e posteriore). La Figura 15 mostra le due facce della scheda di alimentazione realizzata.

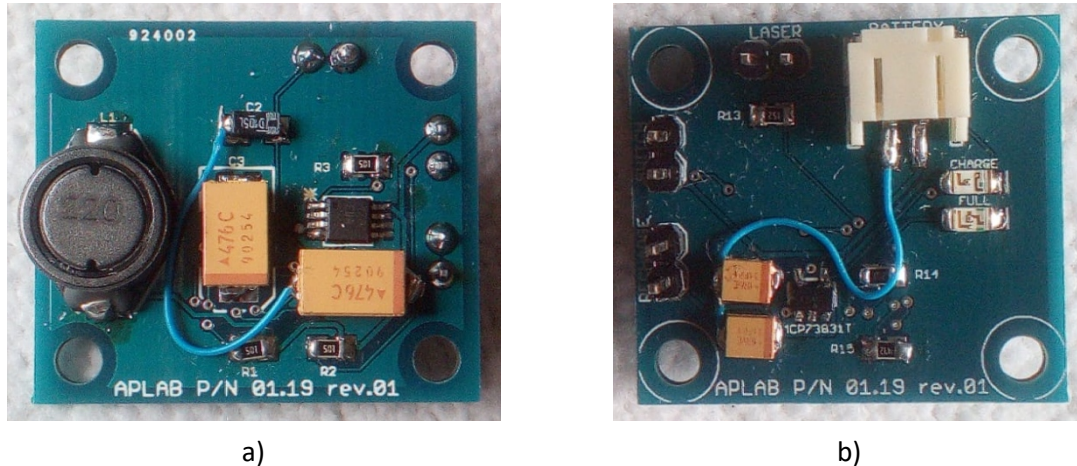


Figura 15 – Circuito di ricarica della batteria del laser: vista a) fronte e b) retro

1.5.2 Progetto meccanico del supporto del sensore LRF

Il sensore laser, la batteria e la scheda con il circuito di ricarica sono alloggiati all'interno di uno speciale casco, da noi progettato e realizzato. In figura 16 è mostrato uno screenshot del programma CAD (Computer-Aided Design) utilizzato per il progetto di tale supporto.

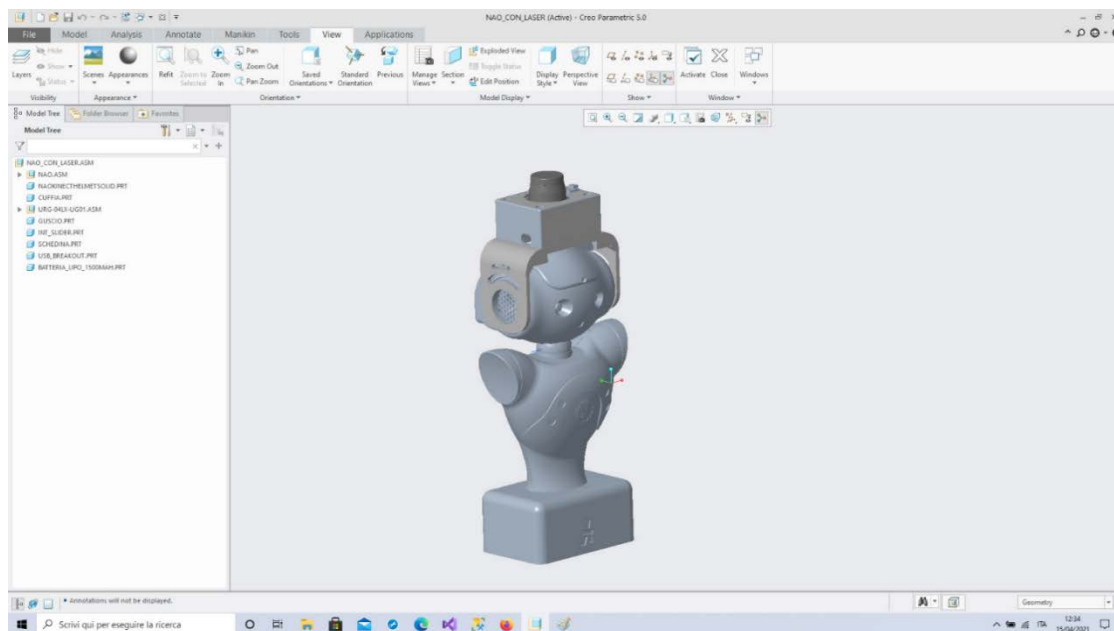


Figura 16 – Screenshot del programma CAD utilizzato per il disegno in 3D del supporto della testa laser

Vista la complessità della forma del casco, si è scelto di non farlo realizzare in officina con lavorazioni meccaniche tradizionali (preparazione di stampo in silicone, stampaggio, finitura, ecc.) a causa dei costi che sarebbero risultati eccessivi ma si è preferito utilizzare tecniche di additive manufacturing (stampa 3D) anche tenendo in considerazione la natura prototipale dell'oggetto da realizzare. Le tecniche di additive manufacturing hanno fatto notevoli progressi negli ultimi anni e rappresentano oramai una valida alternativa per la produzione di prototipi in materiale plastico. La progettazione ha comunque richiesto una

particolare attenzione, perché il casco ha una funzione di supporto strutturale e il fissaggio delle varie parti è realizzato mediante viti autofilettanti. Questo implica che il pezzo debba avere un'elevata densità. Dopo una serie di prove con diverse tipologie di materiali, si è optato per utilizzare un filamento in PLA (Acido PoliLattico), che presenta buone caratteristiche di densità, accuratezza dimensionale e sostenibilità (è infatti realizzato a partire da scarti di frumento e canna da zucchero, ed è quindi biodegradabile).

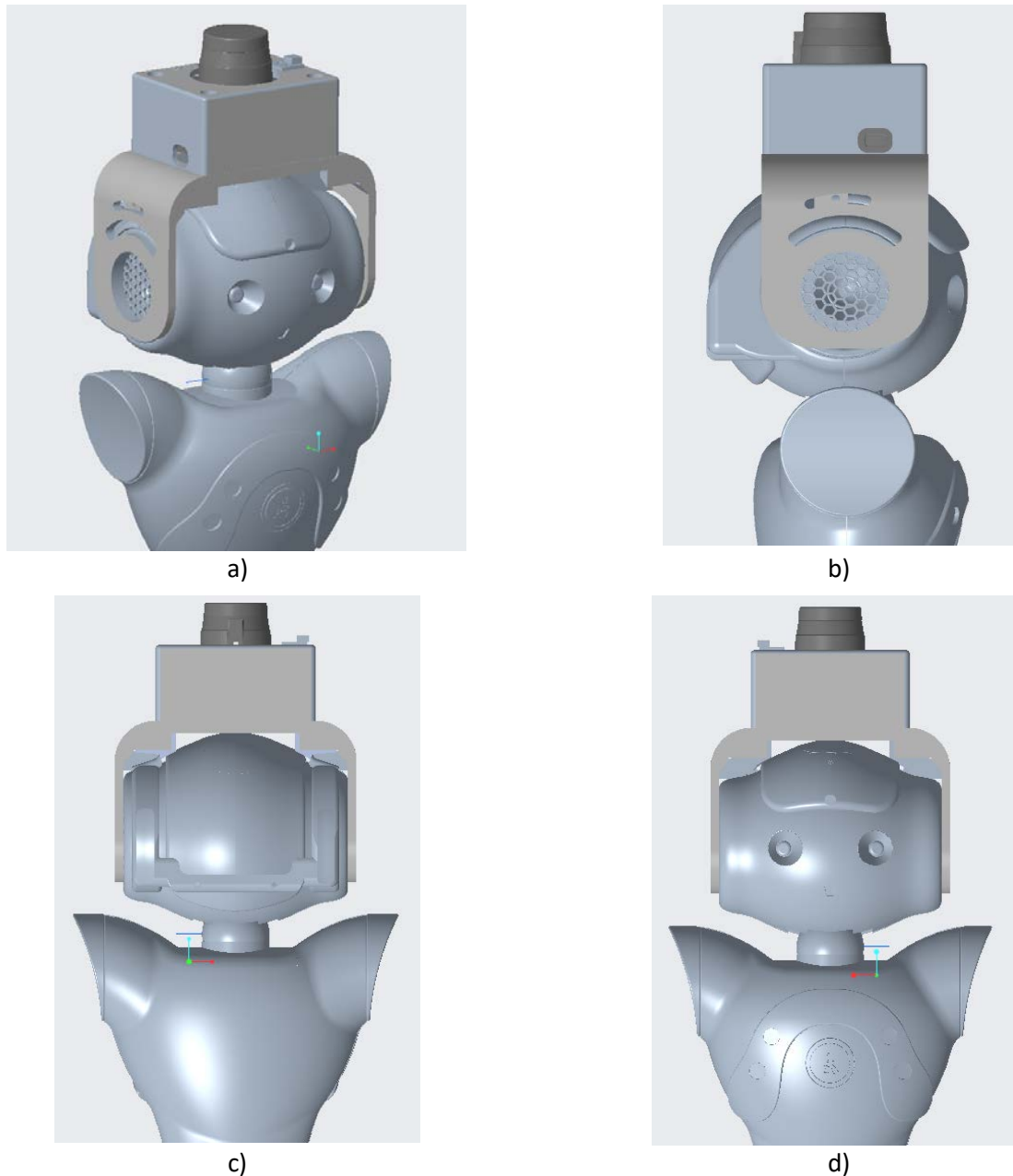


Figura 17 – Disegno in 3D del supporto della testa laser

1.5.3 Caratteristiche funzionali del casco

Dopo una prima fase di progettazione del casco con software di modellazione solida (figura 17), con relativa messa in tavola (figura 19), si è proceduto ad esportare il progetto in formato STL (Standard Tessellation Language), compatibile con la stampa in 3D.



Figura 18 – Disegno in 3D del supporto della testa laser

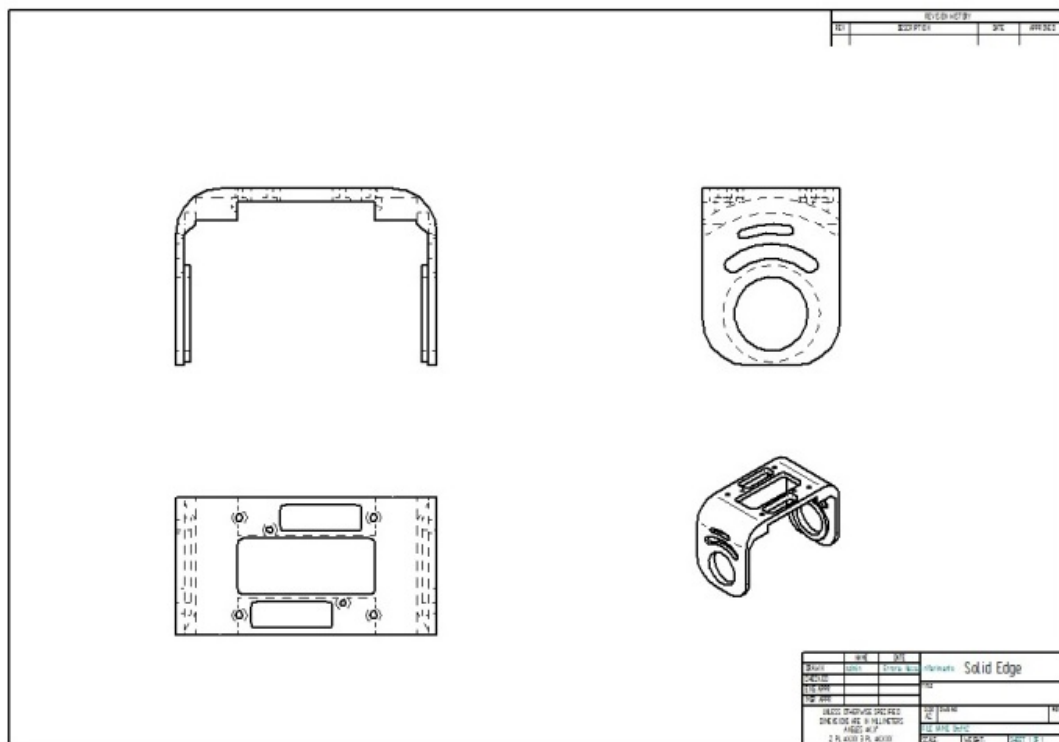


Figura 19 – Disegno in 3D e messa in tavola del casco (dettaglio)

Per la progettazione sono stati impiegati i programmi di modellazione solida Pro/ENGINEER [7] e Solid Edge [8]. Il casco rappresenta un'evoluzione di quello costruito nell'ambito delle attività precedenti, ed è formato di diverse parti separate che possono essere assemblate e smontate per facilitare l'accesso alle varie componenti del sistema. (Figura 20).

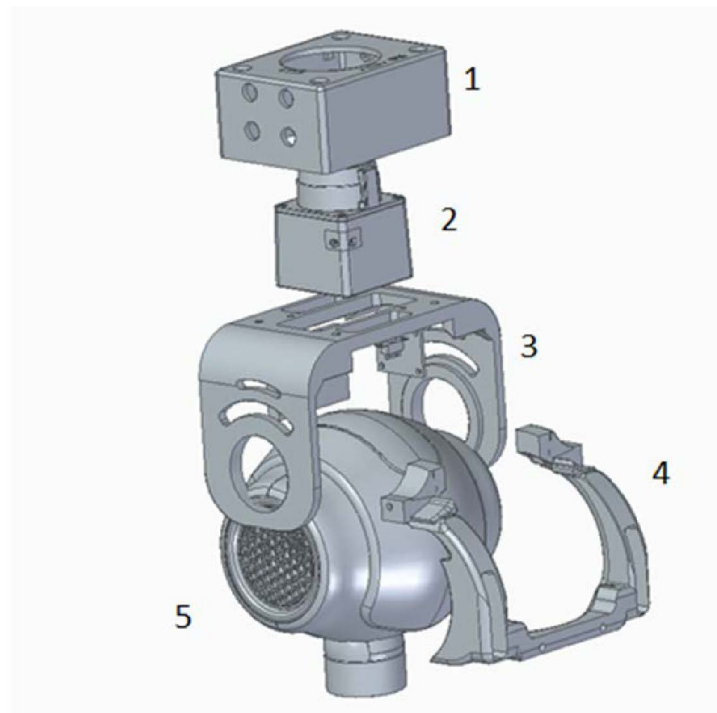


Figura 20 – Vista esplosa del casco

1.5.4 Caratteristiche progettuali del casco

Esaminiamo più da vicino le caratteristiche meccaniche del casco, evidenziando le motivazioni sottostanti alle principali scelte progettuali.

Come si vede dalla figura, partendo dall'alto si possono riconoscere le seguenti parti:

1. la scatola di protezione del sensore, che contiene anche la batteria agli ioni di litio e la relativa scheda elettronica con il circuito di ricarica (non visibili in figura)
2. il sensore laser
3. la base di appoggio del sensore, che si impegna lateralmente sulle "orecchie" del robot
4. una staffa di aggancio posteriore, che si aggancia per mezzo di due viti (non visibili) a due fori presenti posteriormente sulla testa del robot
5. la testa del robot

La forma particolare del casco è dettata dalla quasi completa assenza, sulla testa del robot, di superfici piane e di punti di aggancio.

Il cranio del robot è in sostanza un solido di rotazione, generato dalla rotazione di un profilo ad arco ribassato attorno a un asse orizzontale. Questo fa sì che un oggetto appoggiato sulla testa tenda a ruotare e a cadere al minimo movimento. Gli unici 2 agganci disponibili sono i fori di due viti M3 poste alla base dell'occipite, in prossimità del collo del robot (Figura 21).

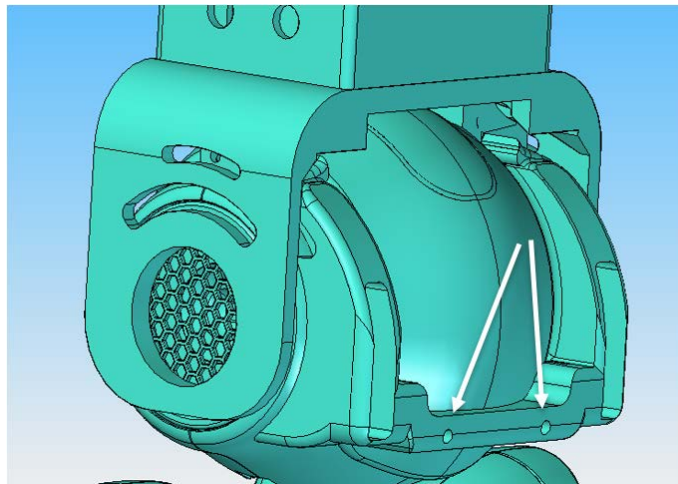


Figura 21 – Fori per viti di aggancio del casco alla testa

La parte posteriore del casco (parte 4 nella figura 20) è quindi disegnata in modo da unire la superficie di appoggio del laser (parte 3), con i punti di aggancio. Un altro requisito progettuale del casco è di consentire la possibilità di ruotare il sensore laser attorno a un asse orizzontale (movimento di tilt o basculamento).

Questo perché in determinate circostanze può essere necessario dirigere il fascio laser non orizzontalmente, ma con un'inclinazione verso il basso, per rilevare ostacoli posti a terra, o verso l'alto. Naturalmente l'inclinazione non deve variare durante il moto del robot, ma essere fissata preventivamente in base alla conformazione dell'ambiente.

Per questo motivo il casco è progettato in modo da consentire il fissaggio del laser con un angolo compreso tra $\pm 20^\circ$ rispetto alla verticale. Questo grado di libertà è realizzato come segue: la parte 3 possiede due scanalature laterali a forma di arco di cerchio (Figura 22).

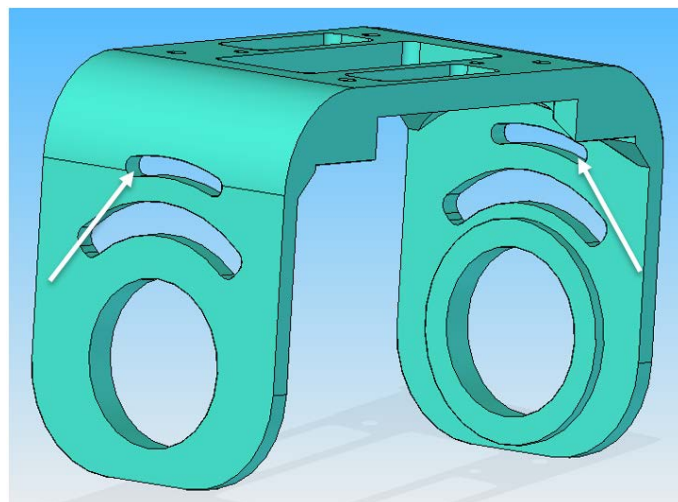
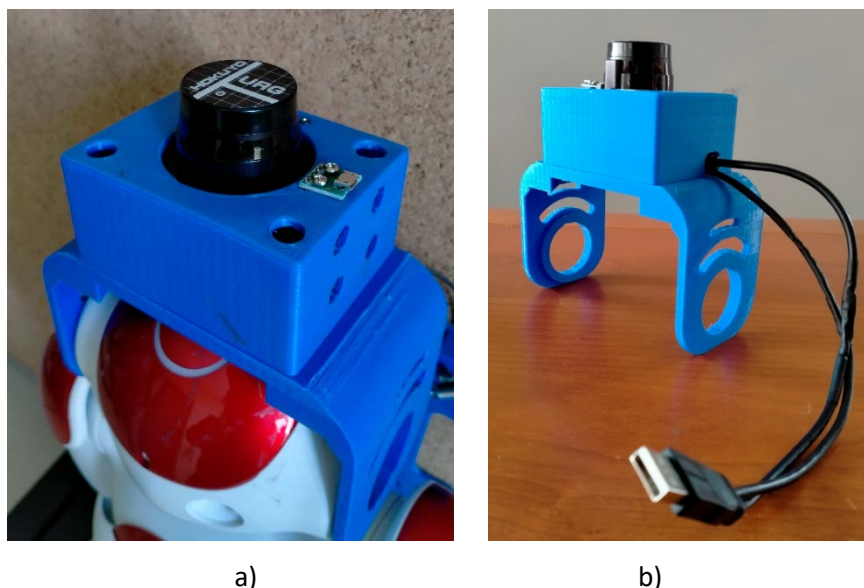


Figura 22 - Dettaglio delle scanalature per il basculamento

In queste scanalature scorrono due viti che si impegnano in due fori realizzati nella parte 4. La parte 3 può essere ruotata attorno a un asse passante per il centro delle "orecchie" del robot. Una volta regolata la posizione, è possibile serrare le viti per mantenere il casco in posizione.



**Figura 23 – a) Sensore laser URG-04LX-UG01 alloggiato sul casco
b) cavo USB doppio per alimentazione e comunicazione**

La scatola (parte 1 in figura 20 e mostrata nella foto in figura 23a) ha la funzione di contenere l'elettronica (sensore, scheda e batteria) e proteggerla dagli urti. Per assicurare un rapido smontaggio, è fissata alla parte 3 mediante quattro viti autofilettanti verticali che possono essere rimosse con un cacciavite a testa magnetica.

Dalla scatola fuoriesce lateralmente un cavo USB doppio, anch'esso progettato e realizzato su misura, che si connette alla porta USB posta nella parte occipitale della testa del robot (figura 25). Grazie a questo cavo la batteria fornisce l'alimentazione al laser, mentre lo scambio di dati avviene lungo il cavo tra laser e testa. Nelle figure 24 e 25 sono mostrate alcune foto, da diversa angolazione relative alla struttura meccanica realizzata.

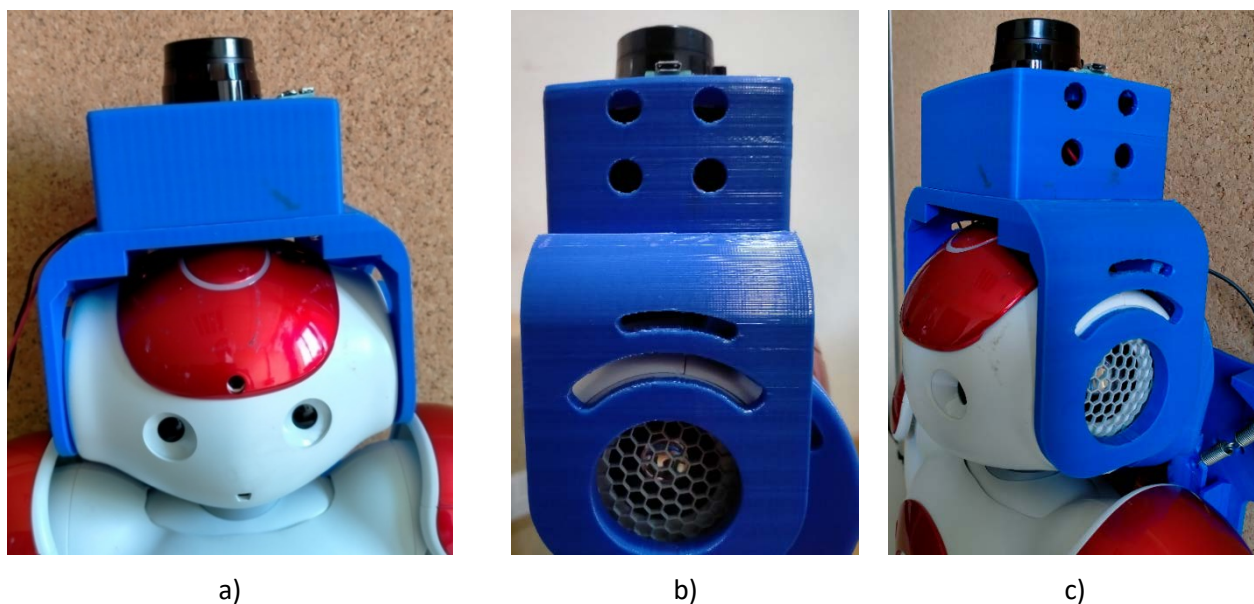


Figura 24 – Testa del robot NAO con casco supporto del sensore laser

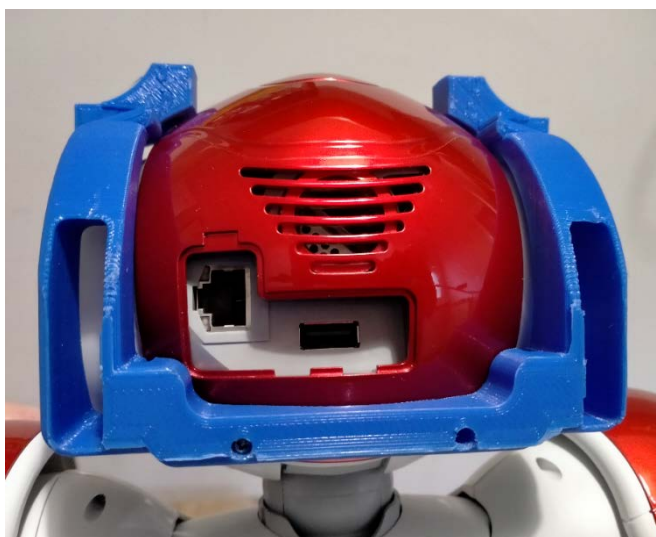


Figura 25 – Vista posteriore della testa del robot NAO dove è stato montato il solo supporto del casco è visibile sulla sinistra la porta ethernet e, al centro, la porta USB

1.5.5 Centro di massa e inerzia.

Per i nostri scopi, è obbligatorio fornire informazioni inerziali sui parametri fisici del caschetto per generare le matrici appropriate per il calcolo dei movimenti dinamici del robot NAO, sia in prove simulate che reali. Le unità della matrice inerziale I_0 , massa M_0 e centro di massa C_0 sono tutte nel sistema MKS, calcolate rispetto al sistema di coordinate mostrato in figura 26. Tutti questi parametri sono stati calcolati considerando la reale distribuzione dei componenti del caschetto.

Come è noto, il controllo dinamico dei movimenti del robot è basato sulla conoscenza della distribuzione delle masse e dei momenti di inerzia dei suoi vari segmenti. Tali parametri, che intervengono nel calcolo della dinamica diretta del robot (Figura 26), sono contenuti nella matrice di inerzia I_0 .

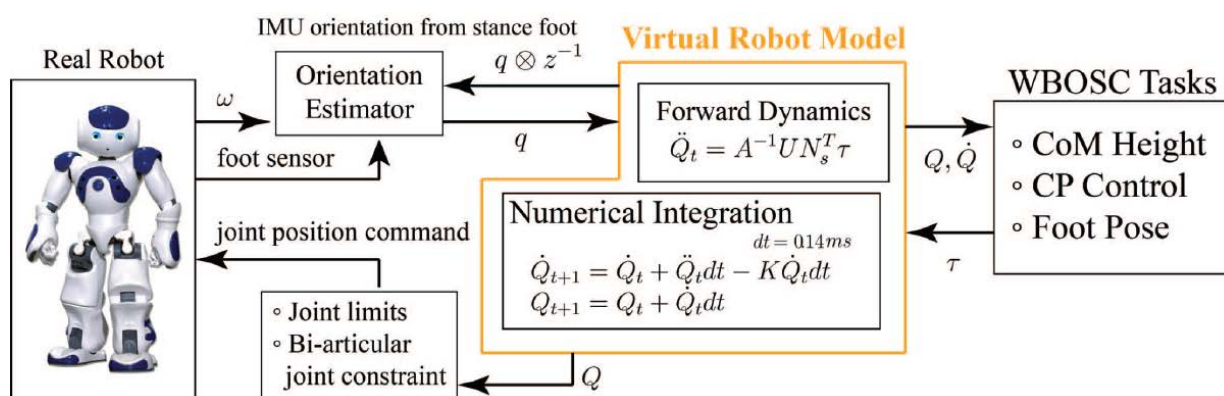


Figura 26 - Schema a blocchi della dinamica diretta del robot (in questo schema per la matrice di inerzia è utilizzata la lettera A)

L'aggiunta del sensore laser, del casco e degli accessori sopra il capo del robot ha modificato in modo significativo la distribuzione dei pesi. Poiché questa modifica può comportare una riduzione nella stabilità, l'equilibrio e i movimenti del robot, si è ritenuto opportuno studiare i possibili effetti che la nuova struttura hanno portato nella matrice di inerzia della testa del robot.

1.5.5.1 Studio matrice inerzia del robot e calcolo della matrice modificata

I valori di fabbrica della posizione del baricentro e dei momenti di inerzia sono riportati in Figura 27. Essi sono riferiti a un sistema cartesiano di riferimento che ha l'origine all'inserzione della testa sul collo del robot.

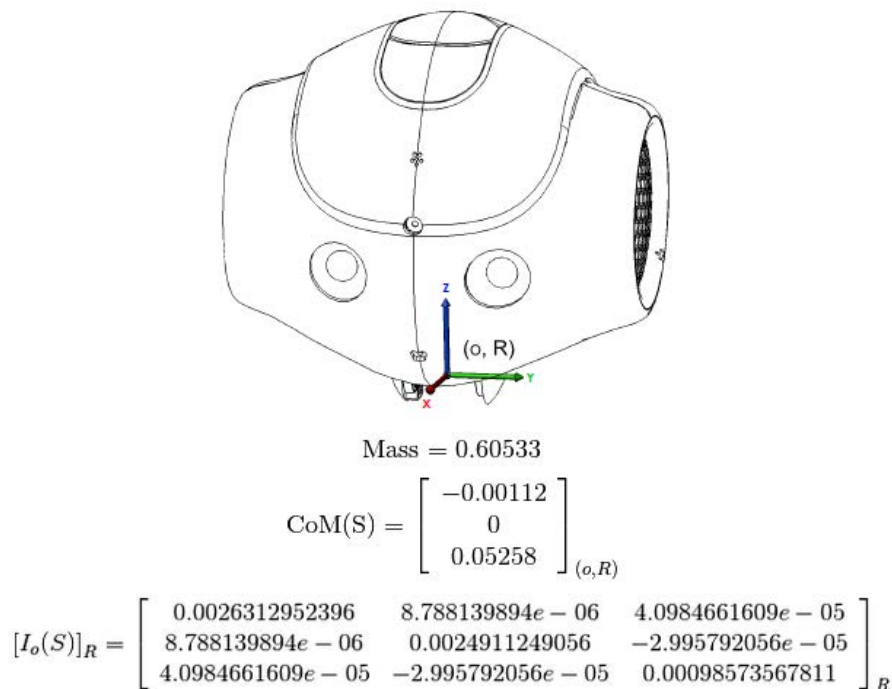


Figura 27 – Sistema di riferimento, posizione del baricentro e matrice di inerzia della testa del robot

Le masse dei componenti aggiuntivi sono state misurate con una bilancia di precisione e riportati in tabella 5, mentre i momenti di inerzia sono stati calcolati grazie ad una specifica funzionalità del programma di modellazione solida utilizzato per il progetto.

Elemento del casco	peso (g)
Scatola base e cavo	125
Supporto e viti	43
Circuito di ricarica	25
Batteria	20
Sensore Laser	160
Totale casco con laser	373

Tabella 5: Pesì dei componenti del casco laser

I nuovi valori della matrice rispetto al medesimo sistema di riferimento sono i seguenti:

Mass = 0.97 kg

$$\text{CoM}(S) = \begin{pmatrix} -0.00155 \\ 0 \\ 0.0734 \end{pmatrix}$$

$$[I_0(S)] = \begin{pmatrix} 0.00645 & 9.327e-06 & 4.632e-05 \\ 9.327e-06 & 0.00621 & -3.265e-05 \\ 4.632e-05 & -3.265e-05 & 0.000161 \end{pmatrix}$$

Una volta noti questi parametri, la matrice di inerzia del segmento relativo alla testa del robot può essere reimpostata, seguendo le procedure disponibili in letteratura.

Il robot con la nuova struttura installata è stato successivamente collaudato eseguendo una serie di movimenti, in un primo momento della sola testa e, successivamente, del robot intero. In tutti i test il robot, grazie alla nuova matrice impostata, mantiene correttamente l'equilibrio e ha risposto eseguendo senza errori i comandi impartiti sia per quello che riguarda i movimenti rotatori della testa che i movimenti di deambulazione.



Figura 28 – Robot NAO con sensore laser LRF URG-04LX-UG01 installato sul casco di supporto

1.5.6 Progetto driver LRF

Sulla base delle specifiche riportate nella documentazione delle API NAOqi, è stata realizzata una libreria software scritta in linguaggio Python³ [9-10]. Tale libreria è costituita da tre funzioni distinte: la prima di queste realizza il collegamento con il robot (ethernet oppure Wi-Fi) mentre la seconda funzione è dedicata alla configurazione del sensore laser per quello che riguarda l'impostazione dell'angolo di apertura e delle soglie inferiore e superiore della distanza di rilevamento. Infine, la terza funzione è dedicata all'acquisizione dell'uscita del sensore laser e al salvataggio su file dei dati acquisiti.

Nel seguito del paragrafo verranno descritte le caratteristiche di ciascuna delle funzioni, il formato della chiamata, la descrizione dei campi per il passaggio dei parametri ed il relativo codice Python.

³ La versione di Python raccomandata per l'uso con le API NAOqi è la 2.7 - 32 bits

1.5.6.1 Routine collegamento con il laser

Formato chiamata funzione:

```
<int>C = ConnectNaoLaser(<string>ipAdd)
```

Parametri

ipAdd = indirizzo IP del robot NAO in formato stringa

la funzione restituisce il valore 0 se la procedura di connessione è andata a buon fine

Codice:

```
def ConnectNaoLaser(ipAdd):  
    % Make the connection with NAOqi API ALLaser  
    Port = 9559          % NAOqi listening port - default value = 9559  
    laser_service = session.service("ALLaser")  
    laser = ALProxy("ALLaser", ipAdd, Port)  
    return 0
```

1.5.6.2 Routine configurazione Laser

Formato chiamata della funzione:

```
<int>C = laserConfig (<int>OpeningAngle,<int>LowerDetectLenght,<int>UpperDetectLenght)
```

Parametri

OpeningAngle = valore dell'angolo di apertura del sensore laser che verrà impostato a (-OpeningAngle, OpeningAngle)

LowerDetectLenght = soglia inferiore della distanza di rilevazione del laser

UpperDetectLenght = soglia superiore della distanza di rilevazione del laser

la funzione restituisce il valore 0 se la procedura di configurazione del dispositivo laser è andata a buon fine

Codice:

```
Def laserConfig (OpeningAngle, LowerDetectLenght, UpperDetectLenght)  
    % Configure laser device  
    laser_service = session.service("ALLaser")  
    laser.setOpeningAngle(-OpeningAngle, OpeningAngle)  
    laser.setDetectingLength(LowerDetectLenght, UpperDetectLenght)  
    return 0
```

1.5.6.3 Routine acquisizione uscita Laser e salvataggio su file

Formato chiamata della funzione:

```
<int>C= LaserRead (<int>Nread,<string>OutFileName)
```

Parametri

Nread = numero di acquisizioni

OutFileName = nome del file in uscita (se non viene indicato anche il percorso completo, viene scritto sulla cartella corrente)

la funzione restituisce il valore 0 se la procedura di acquisizione dati dal dispositivo laser e relativo salvataggio su file è andata a buon fine

Codice:

```
def LaserRead (Nread, OutFileName):
    % Read laser output and write the results in a text file
    Port = 9559 % NAOqi listening port - default value = 9559
    memo = ALProxy("ALMemory", ipAdd, Port)
    i = 0
    import datetime
    now = datetime.datetime.now()

    fileWrite = open(OutFileName, "w")
    fileWrite.write("=====")
    fileWrite.write(" "+now.strftime("%Y-%m-%d %H:%M:%S"))
    fileWrite.write("\n")
    fileWrite.write("=====")

    while i < Nread:
        L = memo.getData("Device/Laser/Value")
        fileWrite.write(str(i) + " ")
        fileWrite.writelines(str(L))
        fileWrite.writelines("\n")
        i = i + 1

    fileWrite.write("=====")
    fileWrite.close()
    return 0
```

Per ogni singola scansione del sensore laser, il sistema, come già riportato nel paragrafo 2.2.1, è in grado di rilevare un numero massimo di punti pari a 683 con una risoluzione angolare di 0.63°. Ognuna delle *Nread* acquisizioni potrà quindi essere composta fino ad un massimo di 683 gruppi di quattro valori ciascuno.

Il file scritto dalla funzione *LaserRead* è un file di tipo testo che presenta un header dove è riportata la data e l'ora in cui è stata effettuata l'acquisizione seguita da un certo numero di righe relativi ciascuna delle quali riporta le coordinate del bersaglio rilevato sia in forma polare che in forma cartesiana. Questi valori sono elencati, in senso orario, in funzione dell'angolo di acquisizione. Ogni riga del file di output riporterà quindi i valori di una singola acquisizione e la cui lunghezza (in termini di numero di punti acquisiti) sarà funzione, a meno di errori di trasmissione, dell'angolo completo di apertura che è stato precedentemente impostato.

Il formato di ciascuna delle righe di tale file di output ha la seguente forma:

```
<num acquisizione> [ [<coord.polari:distanza>,<coord.polari:angolo>,<coord.cartes.:x>,<coord.cart.:y>],
[<>,<>,<>,<>],[<>,<>,<>,<>],...,[<>,<>,<>,<>] ]
```

1.5.6.4 Routine accensione fascio Laser

Formato chiamata funzione:

```
<int>C = NaoLaserON()
```

Parametri

nessuno

la funzione restituisce il valore 0 se la procedura di accensione del fascio laser è andata a buon fine

Codice:

```
def NaoLaserON():
    % Switches ON the laser and starts the acquisition task
    ALLaserProxy::laserON()
```

1.5.6.5 Routine spegnimento fascio Laser

Formato chiamata funzione:

```
<int>C = NaoLaserOFF()
```

Parametri

nessuno

la funzione restituisce il valore 0 se la procedura di accensione del fascio laser è andata a buon fine

Codice:

```
def NaoLaserOFF():
    % Switches OFF the laser and stops the acquisition task
    ALLaserProxy::laserOFF()
```

Disabilita il fascio del sensore laser mantenendo in movimento la testa del laser che continua a girare non effettuando però le misure.

1.6 Test del sistema

Sono state effettuate tre diversi test con il proposito di caratterizzare il comportamento del sensore laser LRF URG-04LX-UG01 installato sulla testa del robot. Ciascuna acquisizione è stata ripetuta cento volte per ciascuna distanza e angolo visuale per poter effettuare una statistica sulle performance del sensore. Per quello che riguarda il posizionamento del robot rispetto ai riferimenti all'interno della stanza, tutte le misure sono state tutte considerate rispetto all'estremità anteriore del piede sinistro del robot. L'illuminazione dell'ambiente è stata realizzata mediante una singola lampada a fluorescenza da 60 W.



a)



b)

Figura 29 – Test #1: foto del robot a 50 cm dalla parete

1.6.1 Test 1

Il primo test a cui è stato sottoposto il sistema ha visto il robot posizionato, all'interno di una stanza priva di mobili, in posizione frontale e ad una distanza nota rispetto ad una delle pareti distante 190 cm rispetto alla parete destra (quella più vicina). Questa misura è stata effettuata prendendo come riferimento il lato sinistro del piede sinistro del robot. Le distanze a cui è stato posto il robot (A = 50 cm, B = 75 cm, C = 100 cm in Figura 30) sono state misurate dalla parete rispetto alla parte anteriore del piede del robot.

La figura 29 mostra due fotografie del robot posizionato all'interno della stanza di fronte alla parete e alla distanza di 50 cm da questa.

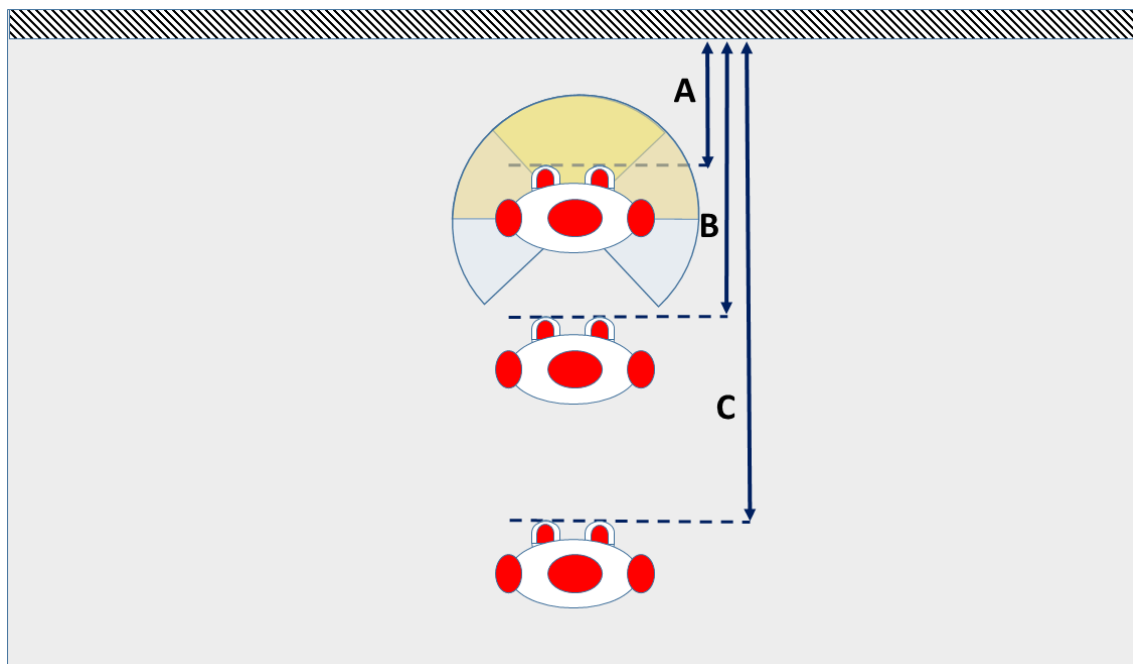
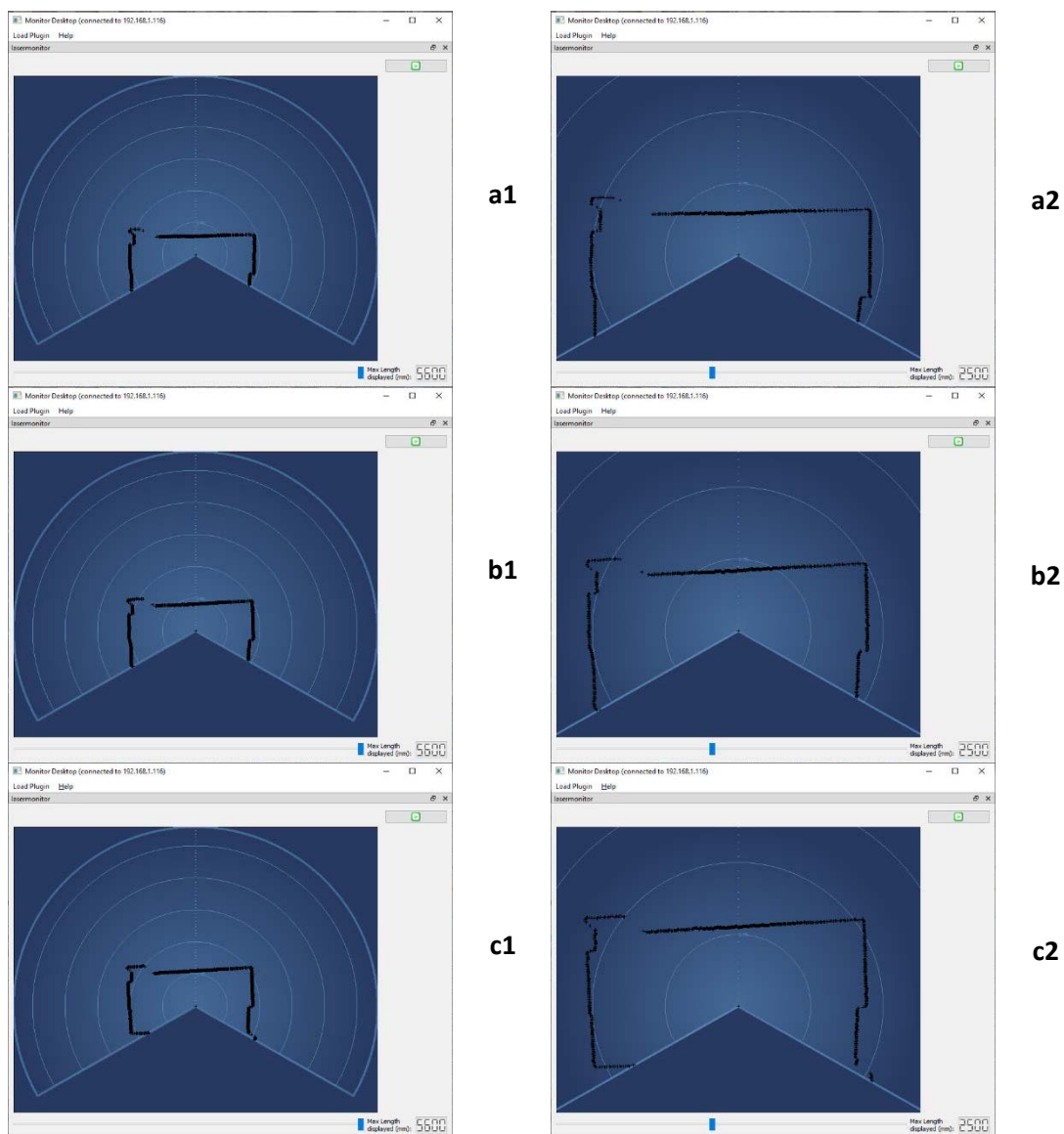


Figura 30 – Schema test #1

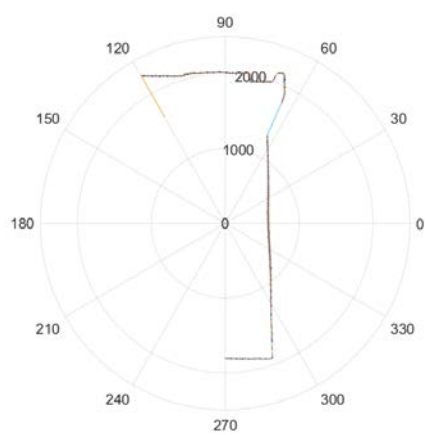
Nella figura 31 sono mostrati i risultati delle acquisizioni visualizzate attraverso il software “monitor” messo a disposizione da Aldebaran al variare della distanza rispetto alla parete e alle soglie di acquisizione (il software non permette di variare l’angolo di apertura che è predeterminato dal programma stesso). La distanza del robot rispetto alla parete è stata variata 50 cm, 75 cm e 100 cm e le acquisizioni sono state effettuate per variando sia l’angolo di apertura (Full, 180° e 90° evidenziati in figura 30 rispettivamente dai colori azzurro, marrone e giallo) che le soglie (minima e massima) di rilevamento. I test effettuati sono riportati nella seguente tabella 6.

N	Distanza fronte (cm)	Distanza dx (cm)	Angolo visuale	Range (mm)
1	50	190	Full (240°)	Max (20 – 5600)
2	75	190	Full (240°)	Max (20 – 5600)
3	75	190	90°	Max (20 – 5600)
4	75	190	180°	Max (20 – 5600)
5	100	190	Full (240°)	Max (20 – 5600)
6	100	190	90°	Max (20 – 5600)
7	100	190	180°	Max (20 – 5600)
8	100	190	180°	500 - 3000

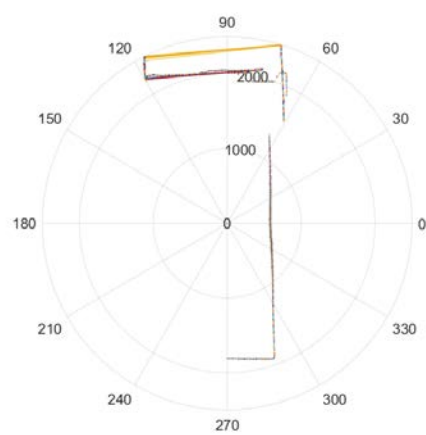
Tabella 6: Test #1



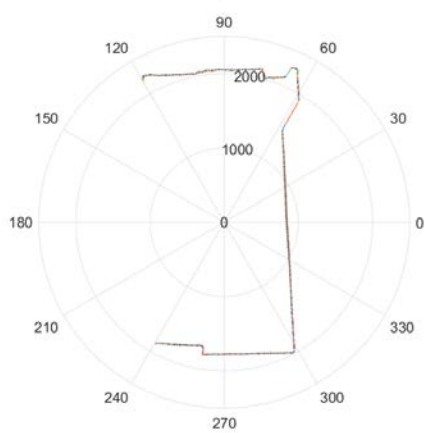
**Figura 31 – Test #1 Monitor laser con range impostato a 5600 mm (max distanza) e 2) a 3000 mm
a) 100cm, b) 750cm, c) 50 cm**



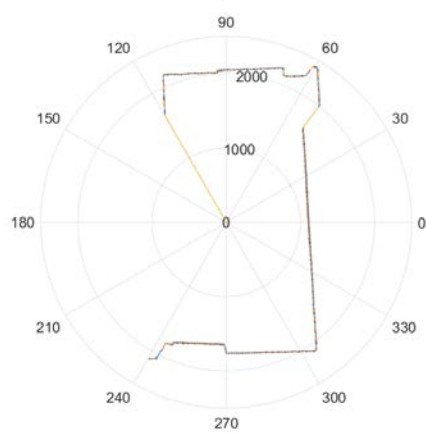
a)



b)



c)



d)

Figura 32 – Test #1 diagramma polare della memoria del laser con range impostato a 5600 mm (max distanza) e angolo di apertura a 240°: a) 50 cm, b) 50 cm, c) 75 cm, d) 100 cm

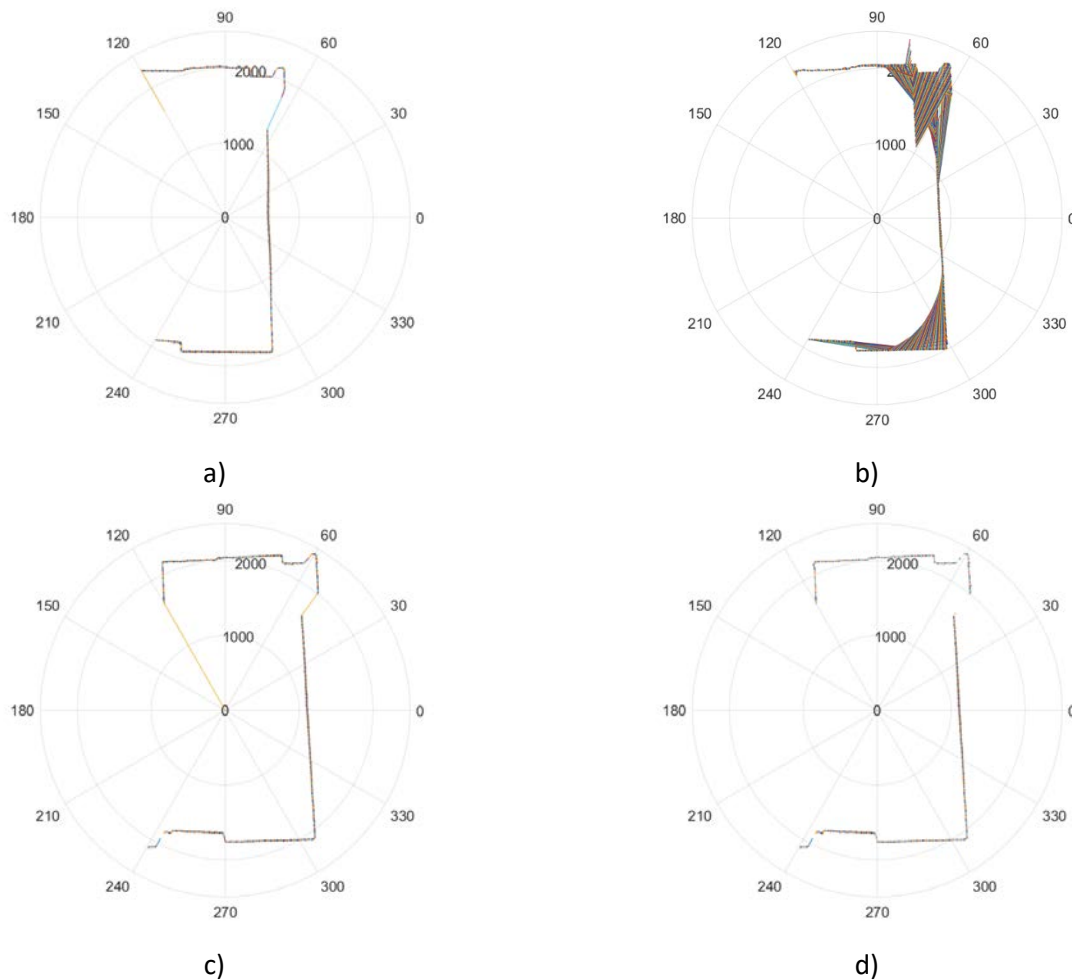


Figura 33 – Test #1 diagramma polare della memoria del laser con range impostato a 5600 mm (max distanza) e angolo di apertura a 180°: a) 50 cm, b) 75 cm, c) 100 cm, d) 100 cm (range 500-3000 mm)

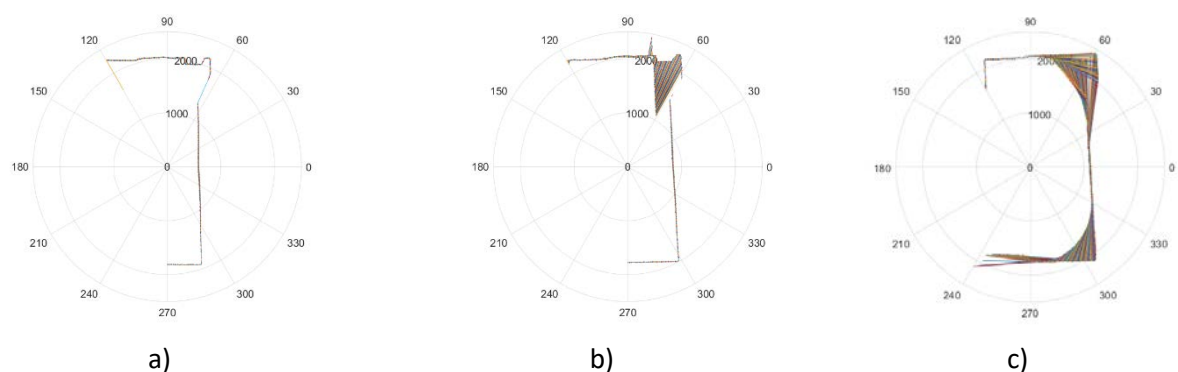


Figura 34 – Test #1 diagramma polare della memoria del laser con range impostato a 5600 mm (max distanza) e angolo di apertura a 90°: a) 50 cm, b) 75 cm, c) 100 cm

In figura 32 sono riportati i diagrammi polari ottenuti con i dati letti dalla memoria del robot al termine dell'acquisizione e misurate con un angolo di apertura di 240°. È da considerare, in queste figure che tutti i diagrammi (a differenza di come sono visualizzati dal software "monitor") sono orientati con il robot che sta guardando la direzione individuata dall'angolo 0° (asse orizzontale nella figura). La figura 32b è il risultato della ripetizione della misura con il robot posizionato a 50 cm, l'errore nella rilevazione della misura sembrerebbe dovuto ad un conflitto tra le operazioni, non sincronizzate, di scrittura da parte del laser con

quelle di lettura. Tale problema, come riportato nei precedenti paragrafi sembrerebbe essere eliminabile spegnendo il fascio laser e non permettendo, in questo modo, al laser di scrivere in memoria durante le operazioni di lettura.

In figura 33 sono mostrate le acquisizioni effettuate alle medesime distanze ma con un diverso angolo di apertura (impostato a 180°). In figura 34d è riportata la misura ancora per la distanza di 100 cm ma con un range di rilevazione impostato a con il minimo a 500 mm ed il massimo a 3000 mm.

Infine, la figura 34 mostra, ancora per le distanze di 50, 75 e 100 mm rispetto alla parete, le acquisizioni realizzate con apertura angolare di 90° . In tutte le acquisizioni si nota come il sistema sia stato in grado di rilevare tutte le pareti della stanza e, soprattutto in quelle a maggiore distanza anche la rientranza della finestra posta a sinistra del robot (e parzialmente visibile nella foto in Figura 29a).

1.6.2 Test 2

Per quello che riguarda il secondo test, il robot è stato posizionato, sempre all'interno di una stanza priva di mobili, in prossimità di un angolo della stanza (a distanza $D = 70$ cm dalla parete a destra del robot – Figura 35) e sono state effettuate letture dell'uscita del sensore laser acquisita alle medesime distanze utilizzate nel corso del test #1 ($A = 50$ cm, $B = 75$ cm, $C = 100$ cm). La misura della distanza del robot rispetto alla parete destra (distanza D) è stata considerata prendendo come riferimento il lato sinistro del suo piede sinistro. Anche in questo caso le acquisizioni (ancora per un totale di cento letture per ciascuna posizione e ciascun angolo di apertura analizzato) sono state effettuate al variare sia le ampiezze dell'angolo visuale di acquisizione che della distanza dalla parete posta di fronte al robot.

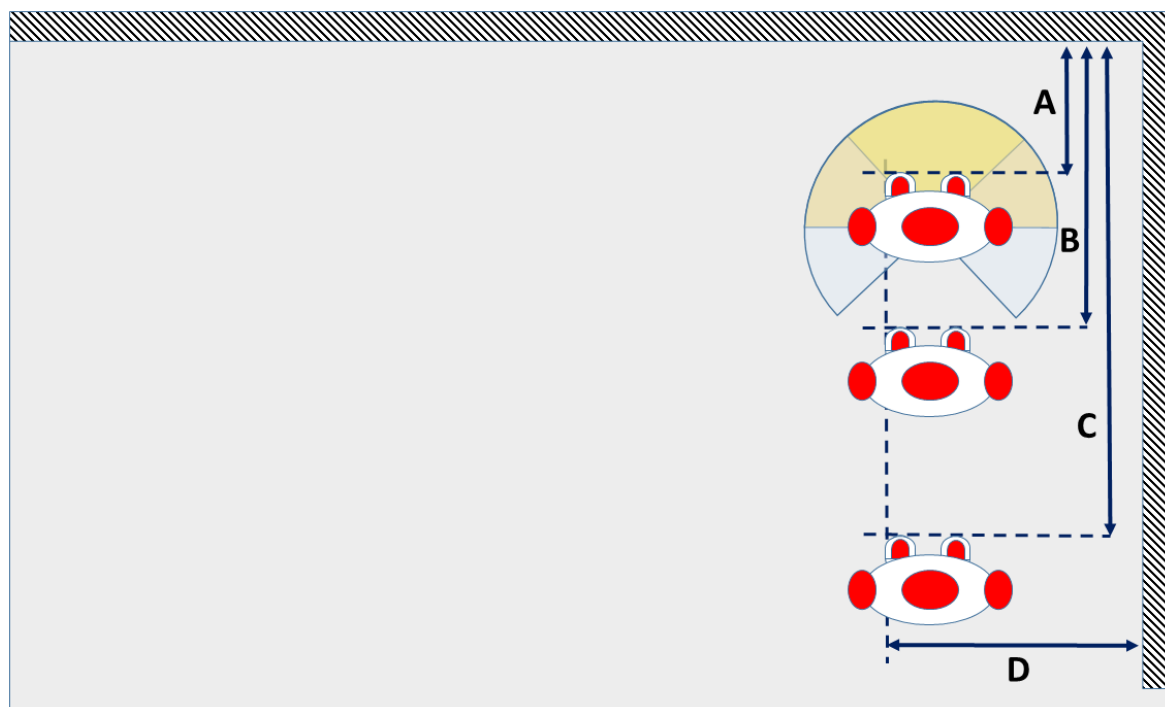
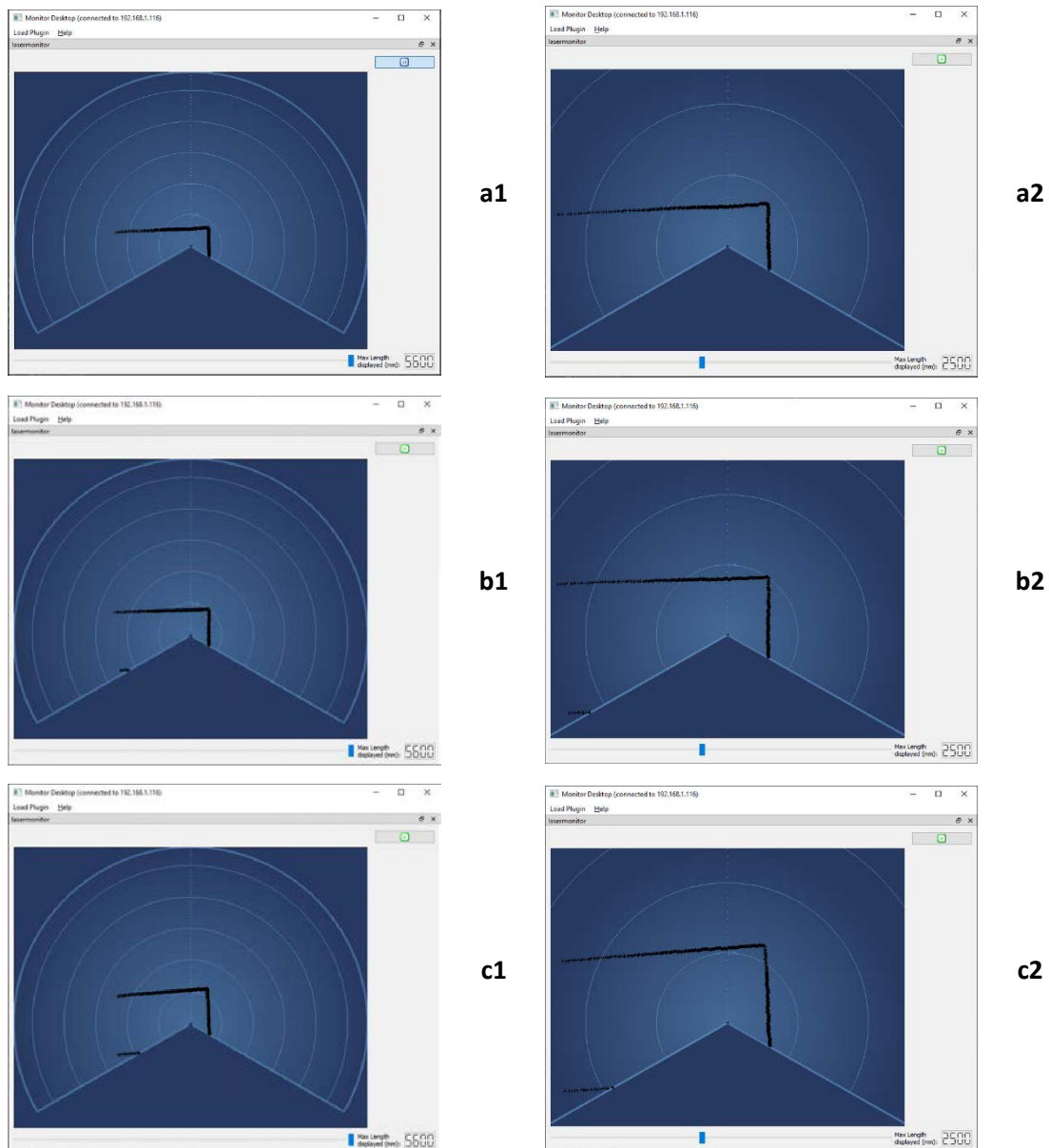


Figura 35 – Schema Test #2

Nella figura 36 sono mostrati i risultati delle acquisizioni visualizzate attraverso il software “monitor al variare della distanza rispetto alla parete e alle soglie di acquisizione (il software non permette di variare l'angolo di apertura che è predeterminato dal programma stesso).

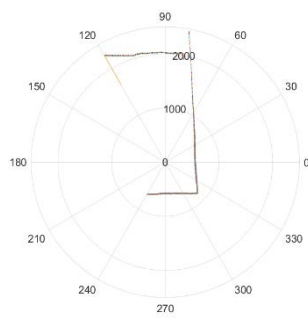
I test effettuati sono riportati in tabella 7.



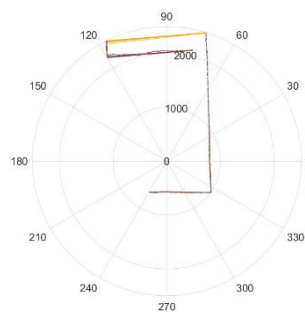
**Figura 36 – Test #2 Monitor laser con range impostato 1) a 5600 mm (max distanza) e 2) a 3000 mm
a) 200cm, b) 150cm, c) 100 cm**

N	Distanza fronte (cm)	Distanza dx (cm)	Angolo visuale	Range (mm)
1	50	70	Full (240°)	Max (20 – 5600)
2	50	70	90°	Max (20 – 5600)
3	50	70	180°	Max (20 – 5600)
4	50	70	180°	500 - 3000
5	75	70	Full (240°)	Max (20 – 5600)
6	75	70	90°	Max (20 – 5600)
7	75	70	180°	Max (20 – 5600)
8	75	70	180°	500 - 3000
9	100	70	Full (240°)	Max (20 – 5600)
10	100	70	90°	Max (20 – 5600)
11	100	70	180°	Max (20 – 5600)
12	100	70	180°	500 - 3000

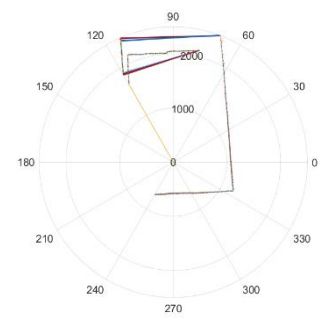
Tabella 7: Test #2



a)

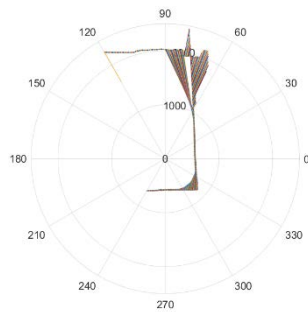


b)

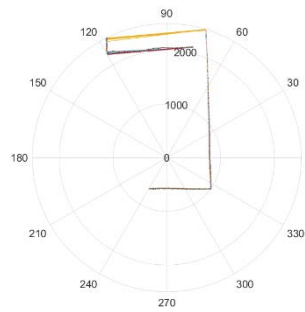


c)

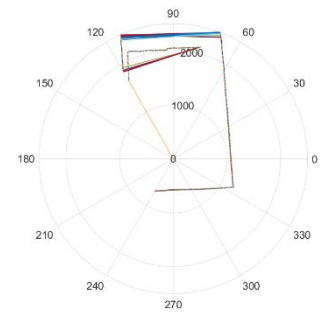
Figura 37 – Test #2 diagramma polare della memoria del laser con range impostato a 5600 mm (max distanza) e Angolo di apertura a 240°: a) 50 cm, b) 75 cm, c) 100 cm



a)

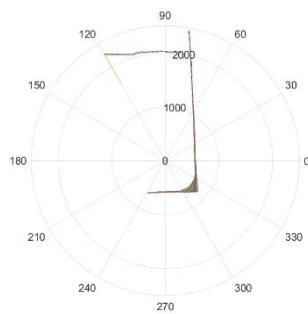


b)

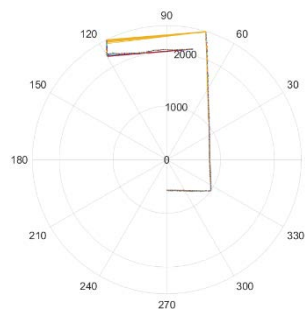


c)

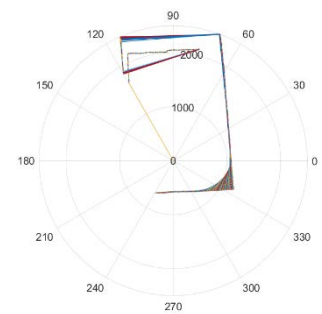
Figura 38 – Test #2 diagramma polare della memoria del laser con range impostato a 5600 mm (max distanza) e Angolo di apertura a 180°: a) 50 cm, b) 75 cm, c) 100 cm



a)



b)



c)

Figura 39 – Test #2 diagramma polare della memoria del laser con range impostato a 5600 mm (max distanza) e Angolo di apertura a 90°: a) 50 cm, b) 75 cm, c) 100 cm

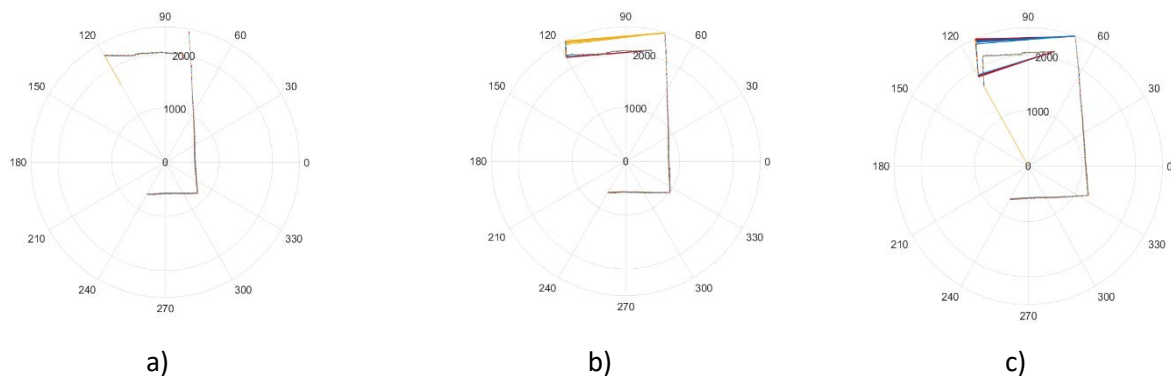


Figura 40 – Test #2 diagramma polare della memoria del laser con range impostato a 500-3000 mm e Angolo di apertura a 180°: a) 50 cm, b) 75 cm, c) 100 cm

In figura 37 sono riportate i diagrammi polari ottenute con i dati letti dalla memoria del robot a fine acquisizione con angolo di apertura e range di rilevamento impostati ai valori di default: rispettivamente 240° e 5600 mm. In figura 37b e 37c sono visibili gli errori di acquisizione già precedentemente commentati. Da tenere presente la diversa orientazione dei diagrammi rispetto a quanto mostrato dal software “monitor”.

In figura 38 e 39 sono riportate le stesse acquisizioni della figura precedente ma ottenute con angolo di apertura rispettivamente di 180° e 90°. In figura 40 è mostrato il risultato delle acquisizioni ottenute per un angolo di apertura di 180° e un range di rilevazione impostato a con il minimo a 500 mm ed il massimo a 3000 mm.

1.6.3 Test 3

In una terza sessione di misure, il robot è stato posto all’interno di una stanza, ancora senza mobilio, ed è stato posizionato un ostacolo di forma cubica (uno scatolone di altezza circa 60 cm e base 50 x 60 cm) ad una distanza pari a 200 cm dal robot (Figura 41 – posizione D) ed orientato in modo da mostrare uno degli spigoli nella direzione del robot. In figura 43 è mostrato l’ostacolo alle diverse distanze di rilevamento (immagini acquisite dalla telecamera del robot).

Sono state effettuate più acquisizioni a diverse distanze rispetto alla posizione dell’ostacolo (a = 200cm, b = 150cm, c = 100 cm, d = 60 cm) ciascuna delle quali è stata ripetuta per cento volte consecutivamente.

Le soglie dell’intervallo di acquisizione sono state impostate, in questo test con il minimo a 500 mm ed il massimo a 3000 mm.

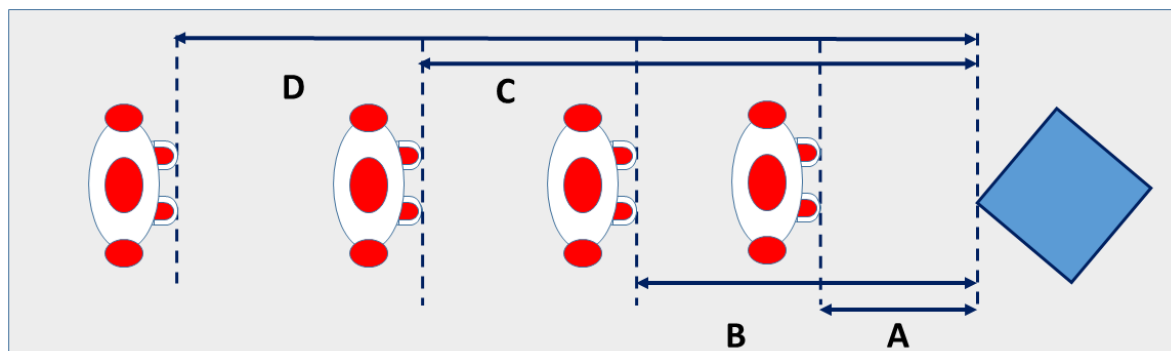
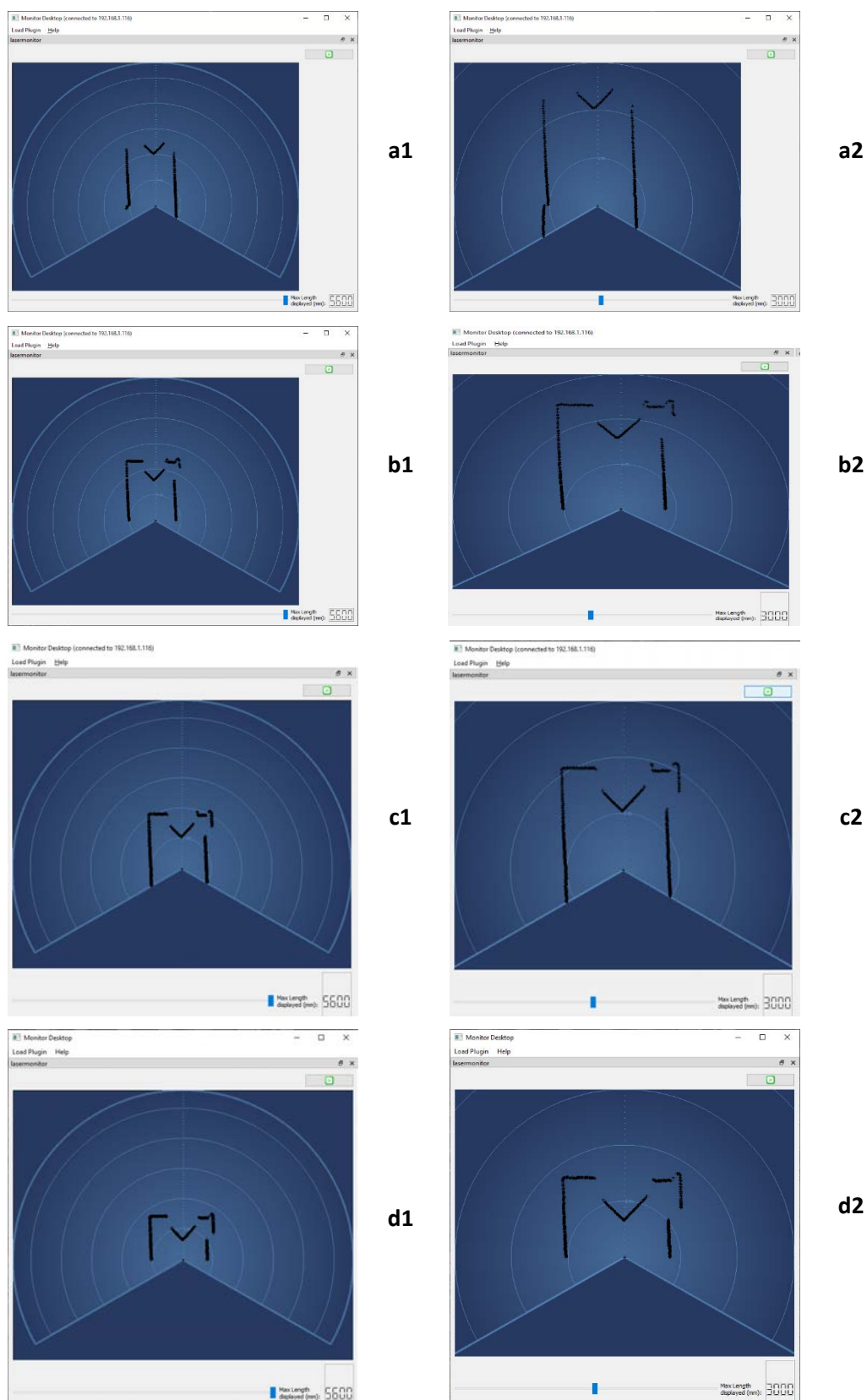


Figura 41 – Schema Test #3



**Figura 42 – Test #4: Monitor laser con range impostato 1) a 5600 mm (max distanza) e 2) a 3000 mm
a) 200cm, b) 150cm, c) 100 cm, d) 60 cm**

Nella figura 42 sono mostrati i risultati delle acquisizioni visualizzate attraverso il software “monitor al variare della distanza dall’ostacolo e per due diversi valori dell’intervallo di rilevazione (5600 mm e 3000 mm).



Figura 43 – Test #3: immagini acquisite dalla telecamera del robot prese a diversa distanza dall’ostacolo: a) 200cm, b) 150cm, c) 100 cm, d) 60 cm

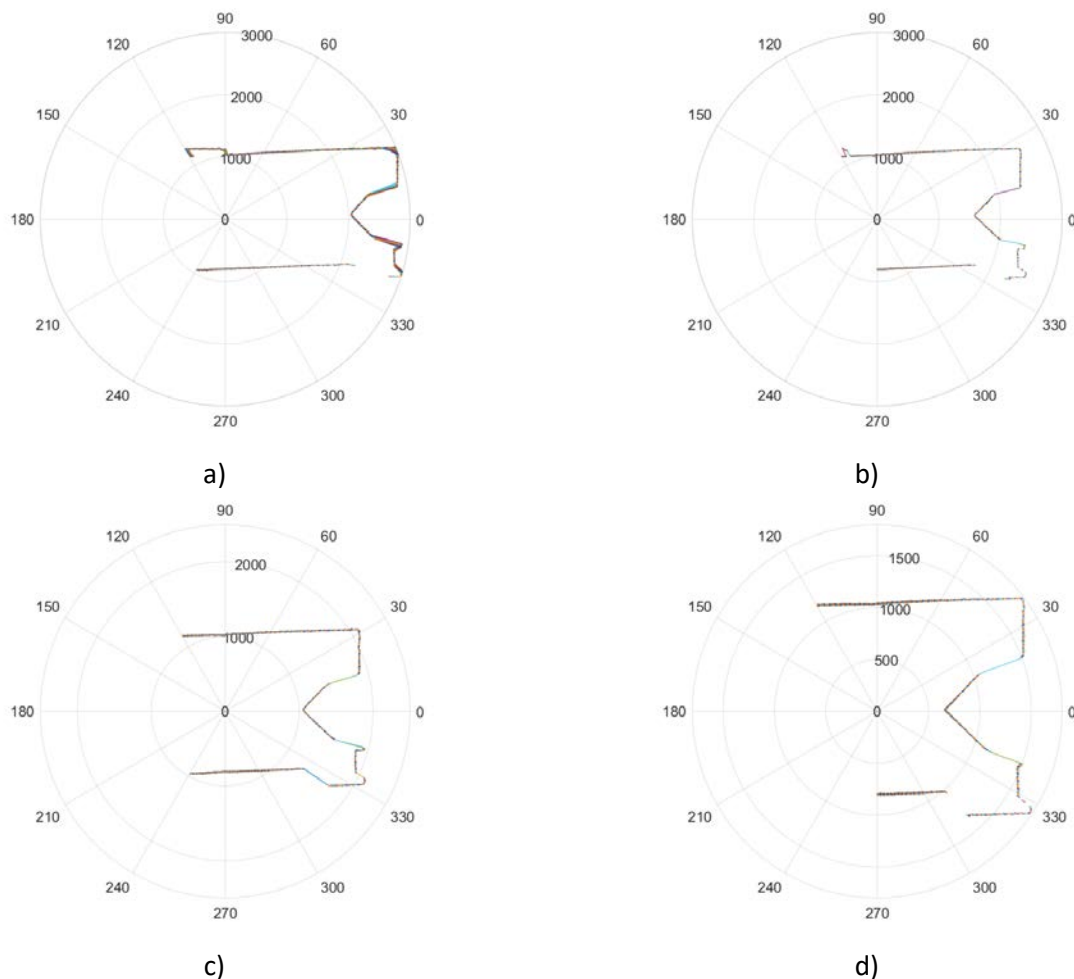


Figura 44 – Test #3 diagramma polare della memoria del laser con range impostato a 500-3000 mm e Angolo di apertura a 240°: a) 200 cm, b) 150 cm, c) 100 cm, d) 60 cm

In tabella 8 sono riportate tutte le misure effettuate nel corso del presente test

N	Distanza ostacolo (cm)	Angolo visuale	Range (mm)
1	200	Full (240°)	500 - 3000
2	150	Full (240°)	500 - 3000
3	100	Full (240°)	500 - 3000
4	60	Full (240°)	500 - 3000
5	200	180°	500 - 3000
6	150	180°	500 - 3000
7	100	180°	500 - 3000
8	60	180°	500 - 3000
9	200	90°	500 - 3000
10	150	90°	500 - 3000
11	100	90°	500 - 3000
12	60	90°	500 - 3000

Tabella 8: Test #3

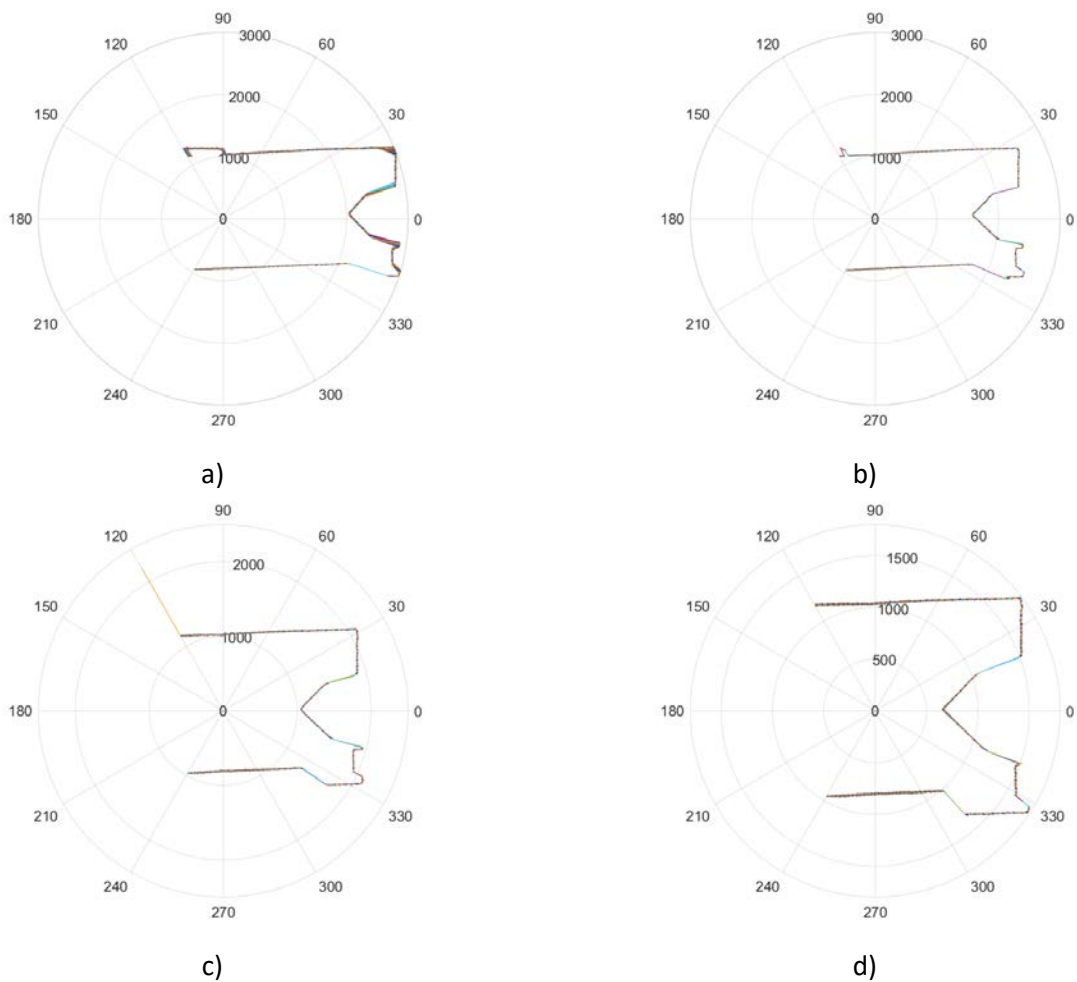


Figura 45 – Test #3 diagramma polare della memoria del laser con range impostato a 500-3000 mm e Angolo di apertura a 180°: a) 200 cm, b) 150 cm, c) 100 cm, d) 60 cm

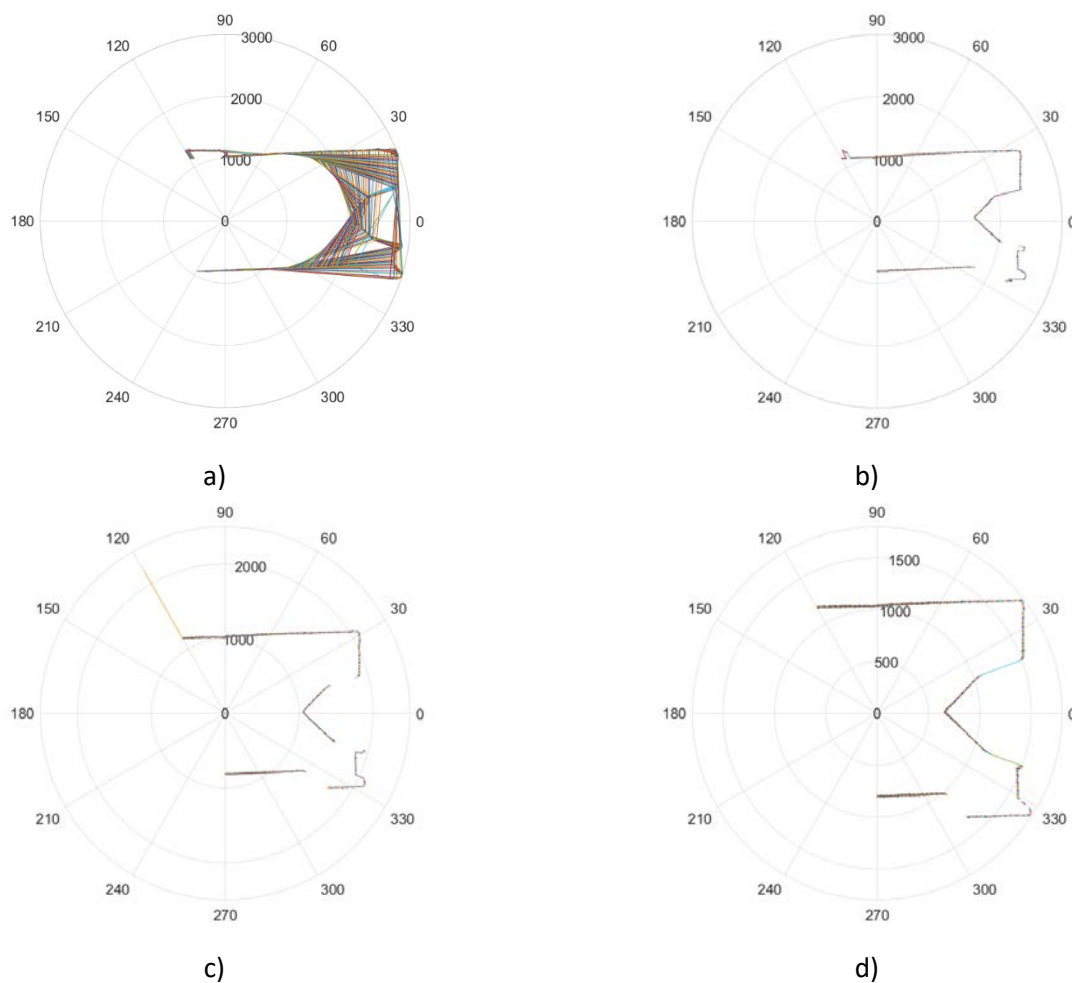


Figura 46 – Test #3 diagramma polare della memoria del laser con range impostato a 500-3000 mm e Angolo di apertura a 90°: a) 200 cm, b) 150 cm, c) 100 cm, d) 60 cm

Nelle figure 44, 45 e 46 sono riportate i diagrammi polari delle acquisizioni della memoria del robot con i dati rilevati dal sensore laser. In particolare, la figura 44 mostra, per le distanze di 200mm, 150 mm, 100 mm e 60 mm rispetto all'ostacolo ottenute con un'apertura angolare di 240°, la sagoma dell'ostacolo in avvicinamento. Nelle figure 45 e 46 sono mostrate le stesse misure ottenute, rispettivamente, per angoli di apertura di 180° e 90°. Anche in questo caso occorre considerare la diversa orientazione dei diagrammi rispetto a quanto mostrato dal software "monitor".

Infine, a parte il solito errore di acquisizione mostrato in figura 46a le cui motivazioni sono già state precedentemente discusse, è interessante notare come il sensore sia stato in grado di rilevare la presenza di una porta (figure 44c e 45c) accidentalmente lasciata aperta nel corso della specifica misura.

2 Conclusioni

L'attività di ricerca svolta in questo anno ha portato all'integrazione sul robot NAO di proprietà di ENEA di un sensore "range finder" in grado di aumentare le capacità sensoriali della macchina all'interno di ambienti non strutturati.

L'attività svolta ha riguardato il progetto meccanico, elettrico e lo sviluppo di driver software per posizionare un sensore laser sulla testa del robot ed integrarlo all'interno del suo sistema di navigazione.

L'aggiunta di tale sensore permetterà al robot di ricostruire, integrando le nuove informazioni con quelle dei sensori già disponibili a bordo, lo scenario che lo circonda rendendo possibile il riconoscimento degli oggetti circostanti e la rilevazione di ostacoli sul suo cammino. L'aggiunta di nuove informazioni ambientali garantisce una maggiore efficienza da parte degli algoritmi di navigazione, aumentando le capacità del robot di muoversi in modo autonomo all'interno della casa.

Tutti i disegni, in qualsiasi formato, tutti i file costruttivi dei pezzi per la stampante 3D e il driver per la gestione del sensore laser illustrati all'interno della presente relazione sono stati consegnati su supporto informatico al laboratorio "Robotica e intelligenza artificiale" dell'Enea CR Casaccia.

3 Riferimenti bibliografici

1. <https://www.softbankrobotics.com/emea/en/robots/nao/find-out-more-about-nao>
2. Krejsa, J. and Vechet, S. (2010) Odometry-free mobile robot localization using bearing only beacons. Proceedings of 14th International Power Electronics and Motion Control Conference (Epe-Pemc 2010). DOI 10.1109/Epepmc.2010.5606893.
3. Panigrahi, P.K. and Bisoy, S.K., 2021. Localization Strategies for Autonomous Mobile Robots: A review. *Journal of King Saud University-Computer and Information Sciences*.
4. https://hokuyo-usa.com/application/files/5115/8947/8197/URG-04LX-UG01_Specifications_Catalog.pdf<https://www.microchip.com/wwwproducts/en/en024903>
5. <https://datasheets.maximintegrated.com/en/ds/MAX1795-MAX1797.pdf>
6. <https://www.ptc.com/en/products/creo/pro-engineer>
7. <https://solidedge.siemens.com/en/>
8. G. Van Rossum, Python 2.7.10 Tutorial: An Introduction to Python, Samurai Media Limited, 2015
9. <https://www.python.org/doc/>

4 Abbreviazioni ed acronimi

API – Application Programming Interface
CAD - Computer-Aided Design
DOF - Degree Of Freedom
GUI - Graphical User Interface
FSR - Force Sensitive Resistor
HAL - Hardware Abstraction Layer
IMU – Inertial Measurement Unit
IP - Internet Protocol
LiPo – Lithium Polymer
LRF - Laser Range Finder
OS - Operating System
PCB - Printed Circuit Board
PLA - Acido PoliLattico
ROS - Robot Operating System
SCIP2.0 - Secure Communications Interoperability Protocol
SDK – Software Development Kit
SLAM - Simultaneous Localization And Mapping
STL - Standard Tessellation Language
URBI - Universal Real-Time Behavior Interface
USB - Universal Serial Bus

5 Curriculum scientifico del gruppo di lavoro

Prof. Vincenzo Bonaiuto received, in 1997, the PhD in Telecommunication and Microelectronics from University of Rome Tor Vergata. From 1996 to 2002, he was Assistant Professor in Electrical Engineering and, in March 2002, at the same university he joined as an Associate Professor.

His main research interests are in the field of the hardware design for real time signal processing by using analog as well as digital solutions (DSP, FPGA, etc.). He published more than 100 papers on international journal and conferences and he was reviewer of several projects for Italian Ministry of Research as well as for international journals and conferences.

Prof. Bonaiuto, in his research activity, has developed wide skills in the design of systems for conditioning and signal analysis and in the design, construction and testing of electrical and electronic systems.

He participated in several projects funded by Italian Research Ministry (PRIN). Since 2009, he is involved in the experiment NA62 at CERN and he is working on the project of the electronics section of Trigger and Data Acquisition System. In 2017, he founded, at the University of Rome Tor Vergata, the Sport Engineering Laboratory where his research group is developing electronics systems based on multi-sensors wearable devices suited for the monitoring of the human movement for the assessment of sport performances.