

UNIVERSITÀ DEGLI STUDI DI BRESCIA

Ing. delle Tecnologie per l'Impresa Digitale (ITID)
Corso di Robotica e UX

Documentazione Echoarm

Progetto di Robotica

Gruppo di lavoro:

Amati Martina
Bellu Andrea
Dondeo Lara
Ottelli Rudy
You Calvin

Anno Accademico 2025/2026

26 November 2025

Indice

1	Robotica	3
1.1	Riepilogo del Processo UX: Dalla Ricerca alla Soluzione	3
1.1.1	Analisi del Contesto e Ricerca Utente	3
1.1.2	Definizione del Concept e Personas	3
1.1.3	Ideazione e Prioritizzazione Tecnica	3
1.1.4	Progettazione dell'Interfaccia e del Workflow	4
1.2	Analisi dei Requisiti e Validazione dell'Architettura	4
1.2.1	Sicurezza e Natura Collaborativa (Cobot)	4
1.2.2	Limitazioni delle Configurazioni Preliminari	5
1.2.3	Soluzione: Architettura Con Molteplici Ridondanze (2P-8R)	5
1.3	Modellazione 3D CAD: Definizione della Catena Cinematica	7
1.3.1	Gruppo 0: Interfaccia e Trasporto Lineare (Assi Prismatici)	7
1.3.2	Gruppo 1: Complesso di Spalla (Orientamento Prossimale)	7
1.3.3	Gruppo 2: Lower Arm Assembly (Braccio Prossimale)	8
1.3.4	Gruppo 3: Upper Arm & Elbow (Avambraccio e Gomito)	8
1.3.5	Gruppo 4: Modulo Distale (Polso Sferico)	8
1.4	Sviluppo del Tool Effector: La Sonda Multimodale	9
1.4.1	Sensore di Pressione/Forza	9
1.4.2	Sensore laser VL53L5CX	9
1.4.3	Telecamera On-Probe	10
1.4.4	Sistema Integrato di Erogazione del Gel Acustico	11
1.4.5	Scelta Progettuale e Meccanismo	11
1.5	Simulazione: Webots	12
1.5.1	Ambientazione: Allestimento dello Scenario Clinico	12
1.5.2	Ottimizzazione delle Mesh (Lettino)	12
1.5.3	Importazione e Assemblaggio Robot	13
1.5.4	Sviluppo Software e Visione	13
1.5.5	Controllo del Robot e Simulazione Ecografica	16
2	Riferimenti e Codice Sorgente	19
2.1	Documentazione Tecnica e Risorse	19

1 Robotica

In questa sezione viene descritta la progettazione tecnica di *Echoarm*, un sistema robotico collaborativo per l'assistenza alle procedure ecografiche. Il progetto nasce per rispondere a una sfida specifica della robotica medica: coniugare la precisione millimetrica di un manipolatore industriale con l'imprevedibilità e la delicatezza richieste nell'interazione con il corpo umano.

L'obiettivo primario non è la completa sostituzione del medico, ma la fornitura di un «terzo braccio» autonomo capace di posizionare la sonda su macro-aree specifiche, permettendo all'operatore di concentrarsi sull'analisi diagnostica o di operare da remoto.

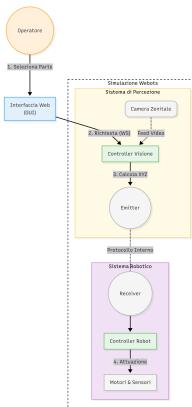


Figura 1: Schema logico

1.1 Riepilogo del Processo UX: Dalla Ricerca alla Soluzione

Il processo di User Experience Design per il progetto *Echoarm* è stato guidato dalla necessità di colmare il divario tra l'avanzamento tecnologico e le reali esigenze operative del personale medico. Il percorso si è articolato in quattro fasi metodologiche principali: Ricerca, Definizione, Ideazione e Progettazione.

1.1.1 Analisi del Contesto e Ricerca Utente

Il punto di partenza è stata l'identificazione di due criticità sistemiche nella pratica ecografica odierna: la **fatica fisica** dell'operatore durante turni prolungati e la **variabilità diagnostica** causata dall'incapacità umana di mantenere una pressione della sonda perfettamente costante.

Per validare queste ipotesi, è stata condotta una **User Research mista**:

- **Quantitativa:** Oltre 70 riscontri tramite questionari hanno confermato la necessità di ridurre lo sforzo fisico senza eliminare la supervisione umana.
- **Qualitativa:** Interviste con specialisti (neuroradiologi e geriatri) hanno evidenziato l'importanza cruciale di standardizzare le acquisizioni per i follow-up clinici.

Parallelamente, l'**Analisi dei Competitor (SWOT)** su sistemi come *Melody*, *CODI*, *ARTHUR* e *Revealan* ha rivelato un vuoto di mercato. I competitor esistenti tendono a polarizzarsi su due estremi: la tele-ecografia pura (che sacrifica l'interazione in loco) o l'automazione completa (che esclude il medico). *Echoarm* è stato quindi posizionato strategicamente come un **Cobot Assistivo**: una soluzione che non sostituisce il medico, ma ne potenzia le capacità operando direttamente al suo fianco.

1.1.2 Definizione del Concept e Personas

Sulla base dei dati raccolti, sono state elaborate le **User Personas** e gli **Scenari d'Uso** (es. il Dott. Conti, esperto ma scettico, e la Dott.ssa Martini, specializzanda) per validare il modello di interazione. È emerso chiaramente che l'utente non cerca un robot che «faccia tutto da solo», ma un «terzo braccio» intelligente. Questo ha portato a definire i tre pilastri della UX di *Echoarm*:

- **Collaborazione:** Il medico deve poter intervenire manualmente per correzioni fini.
- **Ergonomia:** Il sistema deve liberare l'operatore dal mantenimento statico della posa.
- **Controllo:** L'interfaccia deve permettere sia il controllo vocale (per avere le mani libere) sia una gestione intuitiva tramite dashboard.

1.1.3 Ideazione e Prioritizzazione Tecnica

Attraverso sessioni di *Brainwriting* e la tecnica «*How Might We?*», il team ha tradotto i bisogni astratti in requisiti tecnici specifici. Utilizzando una **Matrice Impatto/Sforzo**, sono state selezionate le funzionalità ad alto impatto che definiscono l'architettura del sistema:

- **Cinematica a 9+ DoF:** Scelta non solo tecnica ma ergonomica, necessaria per garantire che il robot possa aggirare gli ostacoli e raggiungere il paziente senza costringere il medico o il paziente a posizioni scomode.
- **Feedback Sensoriale:** L'integrazione di sensori di pressione (per il force control) e visione artificiale è stata prioritizzata per garantire la sicurezza e la fluidità dell'interazione, permettendo al robot di adattarsi ai movimenti del paziente in tempo reale.

1.1.4 Progettazione dell'Interfaccia e del Workflow

La fase finale ha visto la concretizzazione dell'esperienza utente nella **Mappa dell'Applicazione** e nella **Dashboard High-Fidelity**. L'interfaccia è stata progettata per ridurre il carico cognitivo: organizza le informazioni critiche (telemetria del robot, stato del paziente, flusso di lavoro) in un layout intuitivo che supporta sia la modalità di controllo locale che quella di tele-operazione remota.

In conclusione, la UX di *Echoarm* non si limita all'interfaccia grafica, ma permea l'intera progettazione hardware e software, garantendo che ogni movimento del robot sia finalizzato a migliorare la qualità del lavoro del medico e il comfort del paziente.

1.2 Analisi dei Requisiti e Validazione dell'Architettura

La definizione dell'architettura di *EchoArm* nasce dalla necessità di superare i limiti intrinseci dei manipolatori tradizionali nell'applicazione ecografica. Il processo di **Iterative Design** e la successiva validazione cinematica hanno evidenziato come un approccio standard non fosse sufficiente a soddisfare i vincoli operativi clinici.

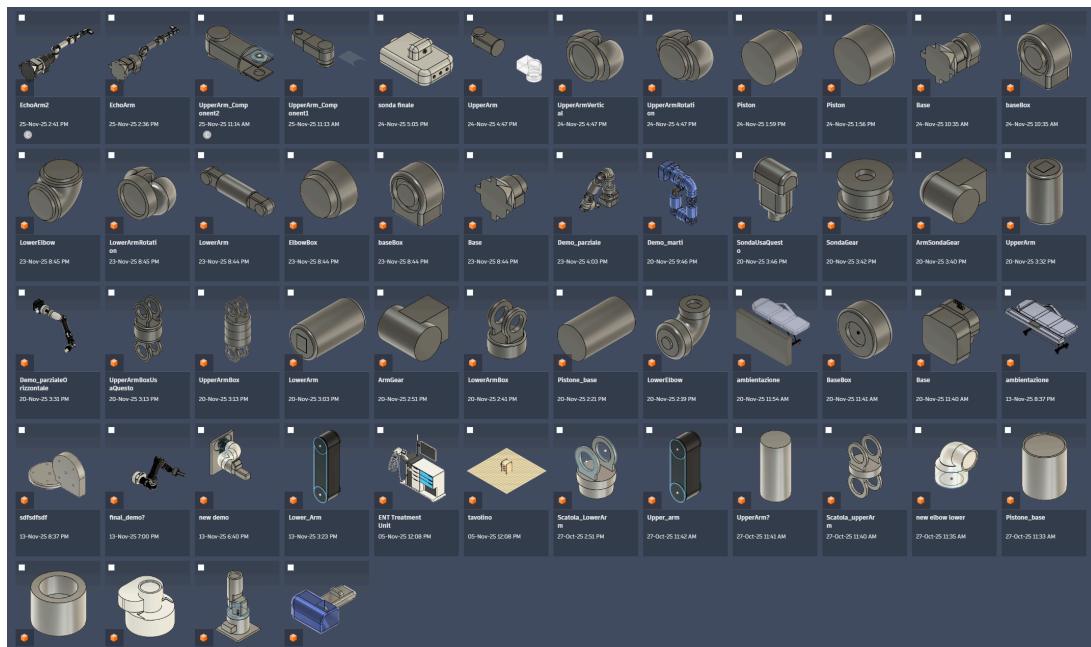


Figura 2: Iterative Design

1.2.1 Sicurezza e Natura Collaborativa (Cobot)

Il sistema è progettato nativamente come un **Cobot** (Collaborative Robot) cioè per la condivisione dello spazio di lavoro con l'operatore umano. A differenza dei robot industriali segregati in gabbie, *EchoArm* opera in un ambiente non strutturato e a contatto diretto con il paziente e il medico.

La sicurezza è garantita da un approccio a livelli:

1. **Power & Force Limiting:** I motori sono tarati per arrestarsi immediatamente se rilevano una coppia resistente anomala (es. un urto accidentale).

2. **Speed Separation Monitoring:** La velocità del braccio viene modulata in funzione della distanza dall'umano (rilevata dai sensori di visione).
3. **Design Intuitivo:** L'assenza di spigoli vivi e le carenature morbide riducono il rischio di lesioni in caso di contatto.

1.2.2 Limitazioni delle Configurazioni Preliminari

Le prime iterazioni progettuali, basate su catene cinematiche antropomorfe standard a 6 Gradi di Libertà (DoF), hanno mostrato criticità insormontabili durante le simulazioni in ambiente operativo.

1. **Singolarità e Saturazione dei Giunti:** Come noto dalla teoria dei manipolatori seriali, un braccio con $n \leq 6$ DoF possiede un numero finito di soluzioni per una data posa (posizione + orientamento) dell'end-effector. Durante l'esame ecografico, il mantenimento della sonda su un punto fisso mentre il paziente respira o si muove portava frequentemente i giunti in configurazione di singolarità, bloccando il movimento o richiedendo riconfigurazioni globali incompatibili con la sicurezza del paziente.
2. **Conflitti nello Spazio di Lavoro (Collision Avoidance):** L'assenza di ridondanza impediva di modificare la postura del braccio (es. posizione del gomito) senza alterare la posizione della sonda. Questo rendeva impossibile evitare ostacoli dinamici (come il paziente stesso o il personale medico) durante le manovre complesse.
3. **Instabilità su Grandi Sbracci:** Le architetture iniziali, pur compatte a riposo, generavano leve sfavorevoli alla massima estensione (**Reach**), compromettendo la rigidità strutturale necessaria per mantenere una pressione costante della sonda sulla cute.

1.2.3 Soluzione: Architettura Con Molteplici Ridondanze (2P-8R)

Per risolvere tali problematiche, si è optato per una configurazione con molteplici ridondanze a 10 Gradi di Libertà complessivi (**8 Dof di Manipolazione e 2 Dof di posizionamento**). Questa scelta garantisce infinite soluzioni cinematiche per la stessa posa della sonda, offrendo la flessibilità necessaria per l'evitamento degli ostacoli.

La catena cinematica è stata strutturata secondo una logica di **Disaccoppiamento Funzionale**:

- **Gruppo di Trasporto (2 DoF):** Affidato a giunti prismatici (Slitta e Sfilo) per la movimentazione macroscopica del volume di lavoro. Questo approccio elimina le singolarità di spalla tipiche dei robot antropomorfi quando devono raggiungere gli estremi del campo operatorio.
- **Gruppo di Manipolazione Ridondante (8 DoF):** Una catena seriale di giunti rotoidali che integra un asse di torsione mediale. Questo permette lo sfruttamento dei movimenti nel **Null Space**: il sistema può riconfigurare la propria struttura interna (es. ruotare il gomito) per evitare collisioni, mantenendo l'end-effector perfettamente stabile sul target.

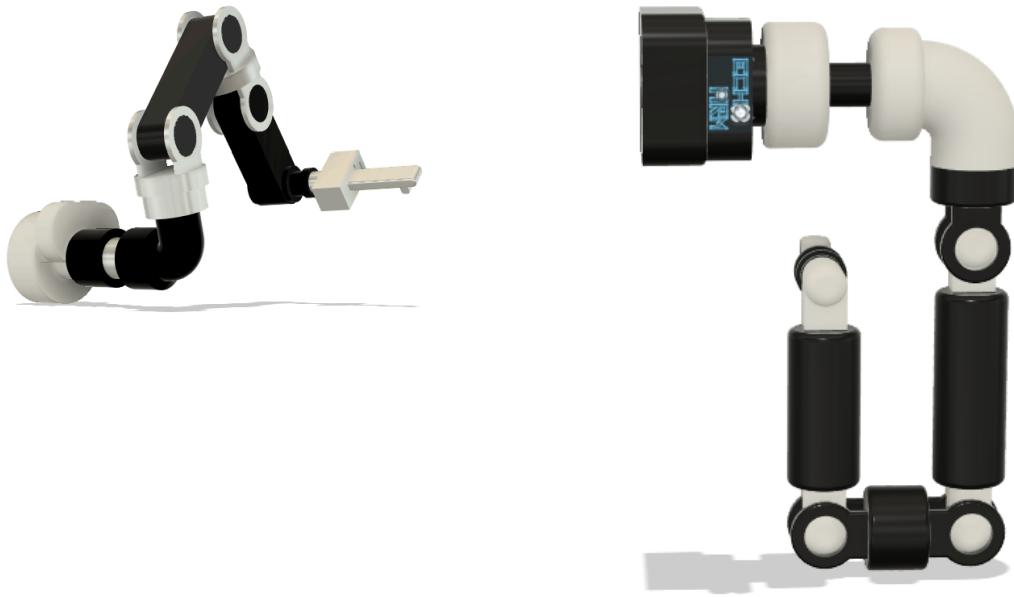


Figura 3: Esempi di iterazioni progettuali preliminari scartate a seguito della validazione cinematica per inadeguatezza strutturale o funzionale.

Strategia di Copertura del Workspace

Uno dei nodi progettuali cruciali ha riguardato la gestione del volume di lavoro allungato, tipico del lettino medico. Sebbene un braccio antropomorfo standard a 6 assi possa teoricamente raggiungere qualsiasi punto nello spazio (x, y, z) con qualsiasi orientamento, esso presenta limiti geometrici evidenti in questo contesto.

Problema: Un braccio a base fissa, posizionato centralmente rispetto al lettino, incorre in singolarità o estensioni eccessive (over-reaching) nel tentativo di raggiungere simultaneamente la testa e i piedi del paziente.

Soluzione: L'introduzione del **Gruppo di Trasporto** (Slitta + Sfilo) risolve il problema alla radice. Questa ridondanza traslazionale è strategica: mentre i 7 assi rotoidali del braccio si occupano dell'orientamento fine della sonda (garantendo la perpendicolarità alla pelle), i 2 assi lineari gestiscono il posizionamento macroscopico. In particolare, lo slider permette una copertura del 100% dell'asse longitudinale del paziente senza mai costringere il braccio in pose singolari.

Orientamento e Layout «Wall-Mounted»

La scelta del layout finale prevede il montaggio del robot su un supporto verticale a parete, con il braccio che si sviluppa orizzontalmente, parallelo al suolo.

Questa configurazione offre vantaggi ergonomici e cinematici rispetto ai robot a pavimento:

1. **Approccio «Top-Down»:** Il robot approccia il paziente lateralmente ma opera dall'alto. Questo garantisce che la sonda possa raggiungere qualsiasi distretto anatomico con un angolo di incidenza ottimale, simulando la postura naturale del braccio umano.
2. **Sicurezza Passiva:** Tale disposizione minimizza il rischio che il «gomito» del robot o i link intermedi entrino in collisione con le sponde del lettino o con le attrezzature elettromedicali circostanti.

1.3 Modellazione 3D CAD: Definizione della Catena Cinematica

L'architettura definitiva di *EchoArm* è stata consolidata nell'ambiente Autodesk Fusion 360. Il sistema si configura come un manipolatore seriale iper-ridondante a 10 Gradi di Libertà (DoF), caratterizzato dalla topologia **2P-8R** (2 Prismatici, 8 Rotatori).

Dall'analisi dell'albero di modellazione (Browser Tree), il robot è suddiviso in 5 gruppi funzionali sequenziali che trasformano il riferimento fisso (**Base**) nel punto operativo della sonda.

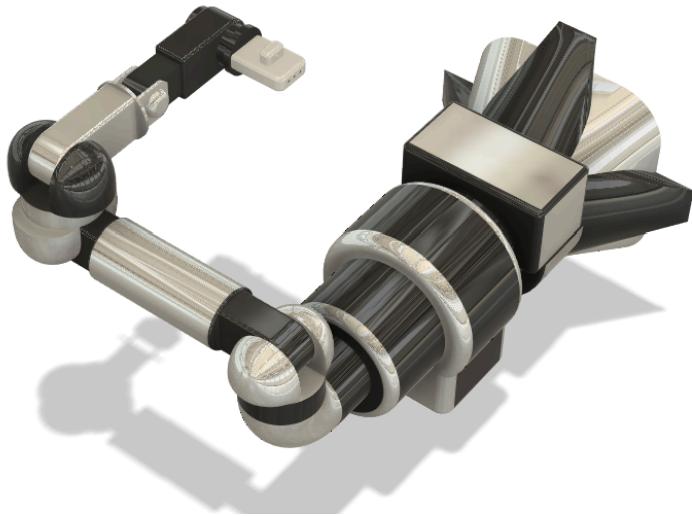


Figura 4: Echoarm

1.3.1 Gruppo 0: Interfaccia e Trasporto Lineare (Assi Prismatici)



Figura 5: Gruppo0

Questa sezione costituisce la base del robot e gestisce il posizionamento macroscopico nel piano cartesiano.

1. Interfaccia Statica (**Base : 1**): Costituisce il **Link 0** (Ground). È l'elemento strutturale fisso ancorato a parete che fornisce il riferimento inerziale per l'intero sistema.
2. Slitta Longitudinale (**baseBox : 1** - Giunto J_1): Primo giunto **Prismatico**. È il carrello che trasla rispetto alla base fissa, permettendo al robot di scorrere lungo l'asse longitudinale (guida a parete).
3. Sfilo Telescopico (**Piston : 1** - Giunto J_2): Secondo giunto **Prismatico**. È il componente che scorre internamente al **baseBox**. Definisce l'estensione radiale («Reach») del robot, permettendo di allontanare o avvicinare la struttura articolata rispetto alla parete.

1.3.2 Gruppo 1: Complesso di Spalla (Orientamento Prossimale)

Questo gruppo funge da giunto cardanico per orientare l'inizio della catena cinematica.

1. Spalla di Rollio (ShoulderBox:1 - Giunto J_3):

Giunto Rotatorio. Questo scatolato ruota attorno all'asse del pistone. La sua funzione è orientare il piano di lavoro verticale del braccio (Rollio/Roll).

2. Spalla di Elevazione (Shoulder:1 - Giunto J_4):

Giunto Rotatorio a cerniera. Connnette il blocco spalla al gruppo successivo e gestisce il sollevamento (Beccheggio/Pitch) dell'intera struttura per vincere la gravità.



Figura 6: Gruppo1

1.3.3 Gruppo 2: Lower Arm Assembly (Braccio Prossimale)

Questa è la sezione strutturale primaria che connette la spalla al gomito. La presenza di un doppio asse rotatorio qui è fondamentale per una maggiore ridondanza.



Figura 7: Gruppo2

1. Torsione Assiale (LowerArmRotation:1 - Giunto J_5): Giunto **Rotatorio** coassiale. Permette al link di ruotare sul proprio asse longitudinale (Rollio).

2. **Rotazione Trasversale Braccio** (LowerArmPitch:1 - Giunto J_6): **Giunto Rotatorio aggiuntivo.** Permette la flessione o l'inclinazione del Link Braccio sul piano verticale o orizzontale. Questo asse raddoppia la capacità del robot di evitare ostacoli e di raggiungere configurazioni complesse senza alterare l'assetto della spalla.
3. Link Strutturale (LowerArm:1 - Link 6): Il corpo rigido principale del braccio. Definisce la lunghezza della prima tratta cinematica (segmento omerale) e trasferisce il moto al gruppo distale.

1.3.4 Gruppo 3: Upper Arm & Elbow (Avambraccio e Gomito)

Questa sezione gestisce l'articolazione mediana e l'avvicinamento finale.

1. Torsione Gomito (Elbow:1 - Giunto J_7): Giunto **Rotatorio.** Permette la rotazione assiale (Torsione) del complesso del gomito. Questo asse aumenta la destrezza del manipolatore, facilitando l'orientamento dell'avambraccio in spazi ristretti.



Figura 8: Gruppo3

2. Gomito di Flessione (UpperArmVertical:1 - Giunto J_8): Giunto **Rotatorio.** È l'articolazione che permette la flessione/estensione dell'avambraccio rispetto al braccio (**Pitch**), avvicinando l'end-effector al paziente.

1.3.5 Gruppo 4: Modulo Distale (Polso Sferico)

La terminazione della catena, dedicata esclusivamente all'assetto della sonda.



Figura 9: Gruppo4

1. Polso Sferico (Giunti J_9, J_{10}): Meccanismo realizzato mediante assi incidenti (visibili come Revolute nell'albero).

Funzione: Permette di orientare la sonda ecografica (sonda finale) con movimenti di Rollio, Beccheggio e Imbardata, garantendo la perpendicolarità rispetto alla superficie anatomica

indipendentemente dalla configurazione geometrica del braccio.

1.4 Sviluppo del Tool Effector: La Sonda Multimodale

Il Tool Effector di Echoarm, ovvero l'alloggiamento della sonda ecografica, è stato progettato come un'unità multimodale e intelligente che funge da interfaccia fisica e sensoriale con il paziente. La sua configurazione è cruciale per garantire la sicurezza del contatto e la qualità diagnostica, in linea con i principi di Percezione Multimodale e Controllo di Impedenza. La punta della sonda è realizzata in gomma per migliorare la sensazione al tatto e il comfort del paziente.

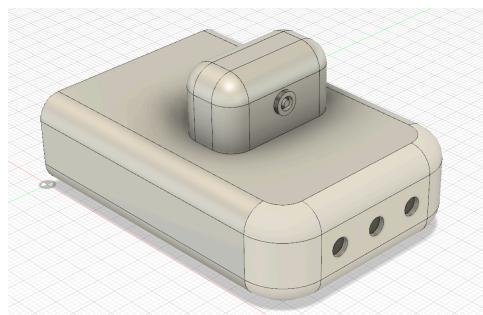


Figura 10: sonda ecografica

L'alloggiamento finale illustrato nel CAD integra quattro sensori e sistemi chiave:

1.4.1 Sensore di Pressione/Forza

Questo sensore risponde direttamente all'esigenza primaria del Controllo di Impedenza.

Scelta Progettuale: Sensore di pressione assoluta (es. TE Connectivity 325412000-00) con range operativo di 0 -> 12 bar e uscita analogica (0 -> 150 mV).

Funzione nel Sistema: Montato per misurare la pressione esercitata sulla pelle. Il *feedback* in tempo reale è l'input fondamentale per gli algoritmi del robot, consentendo di mantenere una Pressione Costante e Sicura (essenziale per la qualità acustica dell'esame) e di rilevare forze anomale o eccessive.

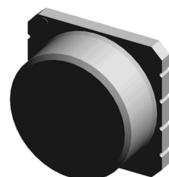


Figura 11: sensore di pressione

1.4.2 Sensore laser VL53L5CX

Il VL53L5CX è un sensore di distanza miniaturizzato che funziona come un piccolo «radar laser» a corto raggio.

Scelta progettuale: VL53L5CX Multi-Zona Time-of-Flight (ToF)

Utilizza la luce laser per misurare il Tempo di Volo (Time-of-Flight), cioè il tempo impiegato dalla luce per colpire un oggetto e tornare indietro, calcolando con precisione la distanza. La Funzione «Occhio»: A differenza dei sensori laser comuni che misurano solo la distanza di un singolo punto centrale, il VL53L5CX ha un campo visivo diviso in una griglia di 8×8 zone (64 punti di misura indipendenti). Immagina di avere 64 piccoli telemetri laser raggruppati in un'unica lente. Questo permette al sensore di vedere una piccola «mappa di profondità» dell'area immediatamente di fronte a sé.

Questa capacità multi-punto è fondamentale per il braccio robotico che deve maneggiare una sonda ecografica:

Vedere la Curvatura: Rilevando 64 distanze contemporaneamente, il robot può «vedere» la curva del corpo. Se il corpo è inclinato o curvo, i 64 punti daranno letture leggermente diverse, permettendo al robot di capire l'angolo esatto.

Atterraggio Delicato: Sfruttando le letture multiple, il robot può regolare l'orientamento della sonda in tempo reale per garantire che la sonda tocchi la pelle piatta e non con un angolo o uno spigolo, assicurando una manovra sicura e delicata.

Massima Qualità: Mantenere la sonda piatta sulla pelle è essenziale per il segnale ecografico, e questo sensore fornisce i dati necessari per questa regolazione costante. In sostanza, il VL53L5CX non dice solo «Quanto sono lontano?», ma anche «Come sono inclinato rispetto alla superficie che sto per toccare?»



Figura 12: sensore di distanza laser

1.4.3 Telecamera On-Probe

La telecamera a bordo sonda completa la percezione visiva locale, lavorando in sinergia con la visione zenitale per l'allineamento fine.

Scelta Progettuale: Modulo telecamera USB UVC (es. Arducam B0438) con risoluzione 2MP (AR0230 WDR).

Funzione nel Sistema: Fornire una visione ad alta risoluzione e corto raggio del punto di contatto. È utilizzata per affinare la posizione della sonda dopo il posizionamento macroscopico guidato da MediaPipe.



Figura 13: telecamera on-probe

L'integrazione fisica di questi tre componenti, unita al controllo cinetico e agli algoritmi di fusione sensoriale, trasforma la sonda in un Organo Terminale Intelligente, capace di navigazione autonoma, interazione sicura e acquisizione diagnostica ottimale.

1.4.4 Sistema Integrato di Erogazione del Gel Acustico

Per garantire un accoppiamento acustico ottimale e continuo tra il trasduttore ecografico e la pelle del paziente, è stato integrato un sistema di erogazione del gel direttamente sul Tool Effector.

Funzione nel Sistema (Accoppiamento Acustico)

Il gel ecografico è un mezzo fondamentale per eliminare le sacche d'aria tra la sonda e la cute, le quali rifletterebbero la quasi totalità degli ultrasuoni, impedendo la penetrazione nei tessuti. Il sistema automatizzato assolve a tre ruoli critici:

- **Massima Qualità del Segnale:** Assicura un'interfaccia a basso coefficiente di riflessione per l'ingresso delle onde sonore.
- **Automazione del Processo:** Elimina la necessità di intervento manuale per l'applicazione iniziale e la riapplicazione durante scansioni prolungate.
- **Distribuzione Controllata:** Permette di erogare la quantità minima ed essenziale di gel, riducendo gli sprechi e il disagio del paziente. Inoltre, il gel è preriscaldato a una temperatura confortevole per il paziente, contribuendo a un'esperienza meno invasiva e più rilassante durante l'esame.

1.4.5 Scelta Progettuale e Meccanismo

Meccanismo: Il sistema è basato su una micro-pompa peristaltica o a siringa (da selezionare in base alla viscosità del gel), collegata a un serbatoio autonomo e controllata dal microcontrollore dell'end-effector.

Ugello di Erogazione: L'uscita del gel è posizionata strategicamente tramite un ugello a bassa pressione vicino al bordo della superficie di contatto della sonda.

Controllo: L'attivazione dell'erogazione è gestita dall'algoritmo di controllo del robot: il gel viene rilasciato appena prima del primo contatto della sonda con la pelle o in base a un rilevamento di perdita di contatto acustico durante la scansione.

Questa integrazione assicura che il sistema robotico gestisca in modo autonomo ogni fase necessaria per un'acquisizione ecografica di alta qualità, dalla navigazione al mantenimento dell'interfaccia acustica

1.5 Simulazione: Webots

Per la simulazione è stato selezionato l'ambiente Webots. La scelta è ricaduta su questa piattaforma per la sua rapidità di configurazione e l'elevata accessibilità, permettendo l'importazione diretta di robot custom e la scrittura dei controller in linguaggio Python, standard de facto per la robotica e la visione artificiale.

1.5.1 Ambientazione: Allestimento dello Scenario Clinico

Prima di integrare il robot, è stato predisposto l'ambiente virtuale per replicare fedelmente uno studio medico, ottimizzando l'ambiente per i sistemi di visione artificiale (CV).

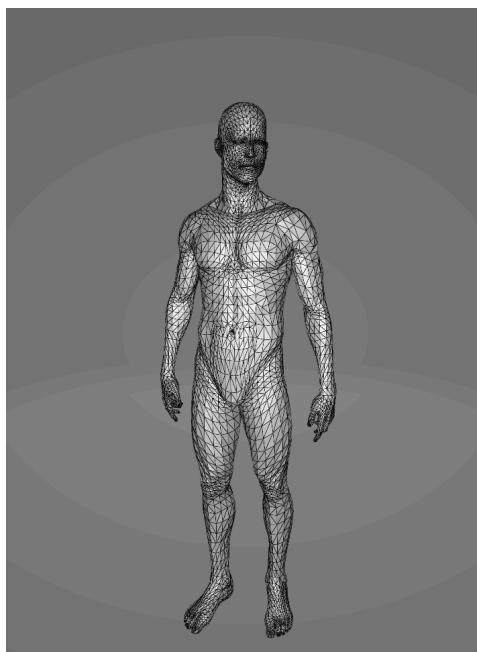


Figura 14: Mesh utilizzata per il paziente

La scena comprende:

- **Lettino:** elemento statico centrale su cui è posizionato il paziente. È dotato di una guida laterale che funge da riferimento fisico per lo scorrimento del robot.
- **Paziente virtuale:** inizialmente è stato testato l'utilizzo dei manichini standard inclusi nella libreria di Webots. Tuttavia, i test preliminari hanno evidenziato che la scarsa definizione dei tratti somatici e la mancanza di texture realistiche impedivano al framework MediaPipe di riconoscere correttamente i **landmarks corporali**.

Soluzione: È stato necessario reperire e importare un modello 3D umanoide esterno (formato .obj) caratterizzato da texture ad alta risoluzione e proporzioni corrette.

- **Setup di illuminazione:** sono state configurate fonti di luce ambientale e direzionale, focalizzate sull'area di esame, per garantire che la texture del paziente fosse visibile chiaramente dalla telecamera, minimizzando ombre nette che potrebbero confondere la visione artificiale.
- **Arredi accessori:** elementi di contorno inseriti per arricchire la scena (armadietti, monitor di supervisione, tende) e aumentare il realismo visivo della sala medica simulata.



Figura 15: Vista generale dell'ambiente simulato.

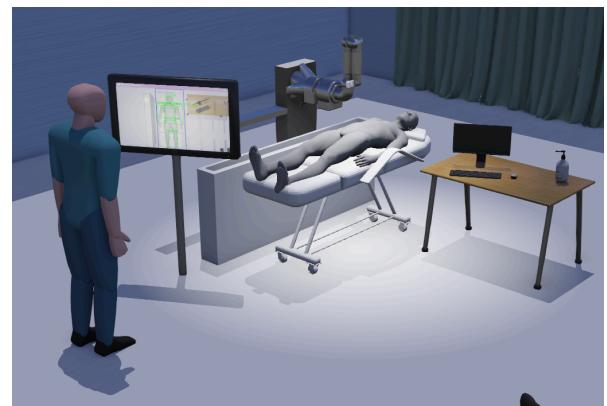


Figura 16: Dettaglio sull'area operativa.

1.5.2 Ottimizzazione delle Mesh (Lettino)

Durante l'allestimento dello scenario, è emersa una criticità specifica legata al modello del lettino. Nonostante i parametri di illuminazione fossero corretti, quest'ultimo non proiettava ombre sulla scena.

L'analisi ha evidenziato che la mesh del lettino, esportata direttamente da Fusion 360 in formato .stl ad alta risoluzione, superava il limite di vertici gestibile dal motore di rendering di Webots per il calcolo delle ombre (spesso limitato per ogni oggetto).

Soluzione: Il problema è stato risolto applicando una procedura di decimazione della mesh (Mesh Decimation) specifica per il lettino. Riducendo il numero di poligoni a circa 27.000, è stato possibile mantenere una fedeltà visiva elevata garantendo al contempo il corretto rendering delle ombre e alleggerendo il carico computazionale.

1.5.3 Importazione e Assemblaggio Robot

Il processo di importazione del modello in ambiente di simulazione ha attraversato due fasi distinte:

Tentativo preliminare (Pipeline URDF)

Inizialmente, si è tentato di seguire il flusso di lavoro standard automatizzato: esportazione dell'intero assemblaggio da CAD in formato **URDF** (Unified Robot Description Format) e successiva conversione in nodi Webots (.proto). Tuttavia, questa procedura ha evidenziato criticità bloccanti che ne hanno impedito l'utilizzo:

- **Incongruenze di Scala (Mesh vs Physics):** Il processo di conversione generava discrepanze dimensionali significative tra le mesh visive e i volumi di collisione (`boundingObject`), rendendo instabile la simulazione fisica.
- **Disallineamento dei Giunti:** I frame di riferimento dei giunti (Joint Frames) importati presentavano orientamenti errati, causando rotazioni su assi non previsti o blocchi meccanici.

Soluzione Definitiva: Ricostruzione Manuale

Di conseguenza, si è optato per un approccio di **assemblaggio manuale**. I componenti sono stati esportati singolarmente in formato .stl e ricostruiti passo dopo passo nell'albero dei nodi di Webots. Questo metodo, sebbene più oneroso in termini di tempo, ha garantito il controllo totale sulla fisica del robot.

L'assemblaggio ha seguito rigorosamente la logica della catena cinematica (Kinematic Chain), strutturando i nodi in una gerarchia padre-figlio:

1. Base Mobile (Slider): il primo giunto è stato definito come `SliderJoint`. Questo nodo vincola il movimento a un solo asse traslazionale, permettendo l'installazione di un `LinearMotor` che controlla lo scorrimento del robot lungo il binario del lettino.
2. Giunti del Braccio: i segmenti successivi del manipolatore sono stati collegati tramite nodi `HingeJoint` o `LinearJoint`. Ogni giunto è stato configurato manualmente inserendo:
 - Un `Device (Motor)`: per applicare coppia e controllare la posizione angolare.
 - Un `PositionSensor`: indispensabile per chiudere l'anello di controllo e leggere lo stato attuale del robot.
3. Fisica e Collisioni: per ogni link (`Solid`), è stata definita manualmente la mesh visiva e, separatamente, un `boundingObject` semplificato per ottimizzare il calcolo delle collisioni senza errori di scala.

1.5.4 Sviluppo Software e Visione

La parte più complessa dello sviluppo ha riguardato l'integrazione tra la percezione visiva, la gestione della profondità e la logica di interazione uomo-macchina.

Configurazione MediaPipe e Calcolo Coordinate

Per il riconoscimento delle parti anatomiche è stato configurato il framework *Google MediaPipe Pose*. Il sistema implementa una logica di **mappatura spaziale statica**: invece di inseguire il movimento in

tempo reale (che potrebbe essere pericoloso in un contesto medico), il software crea una «mappa» 3D delle articolazioni del paziente in una fase di scansione iniziale.

Il processo di calcolo delle coordinate (X, Y, Z) segue questi passaggi:

1. **Estrazione 2D:** Identificazione dei landmarks chiave per ogni parte anatomica (es. media tra spalle per il torace).
2. **Validazione ROI:** Verifica che i punti siano all'interno dell'area di taglio (Crop) definita.
3. **Acquisizione Profondità:** Calcolo della mediana dei valori di profondità in una matrice 10×10 pixel attorno al centroide, per filtrare il rumore del sensore Range Finder.
4. **Riproiezione 3D:** Conversione tramite modello **pinhole** (usando la lunghezza focale f) per ottenere le coordinate metriche nello spazio della camera.

Implementazione degli Algoritmi di Visione

Il controller Python opera secondo una macchina a stati finiti composta da due fasi distinte: **Scansione e Memoria**.

Fase di Scansione (Scanning Phase)

All'avvio, il sistema entra in modalità di acquisizione per una durata prefissata (SCAN_DURATION = 50 step). In questa fase:

- Viene visualizzato l'overlay «SCANSIONE IN CORSO» con la percentuale di completamento.
- Il sistema accumula le posizioni delle parti del corpo (Testa, Torace, Addome, ecc.).
- Le coordinate 3D vengono calcolate e memorizzate nel dizionario SCANNED_COORDS solo se la confidenza visiva e la validità della profondità sono sufficienti.
- Questa fase richiede che il paziente rimanga immobile per garantire la precisione della mappa 3D.

Modalità Memoria (Memory Mode)

Al termine della scansione, il sistema «congela» le coordinate acquisite. Questo approccio aumenta la sicurezza, prevenendo movimenti involontari del robot dovuti a falsi positivi o rumore momentaneo della telecamera.

- L'operatore vede a schermo i punti scansionati (cerchi blu) sovraimposti al video.
- L'interazione avviene tramite tastiera: premendo i tasti da **1 a 6**, l'operatore seleziona la parte anatomica desiderata.

Ottimizzazione ROI (Region of Interest)

Per focalizzare l'attenzione esclusivamente sul paziente ed escludere personale medico circostante (ed evitare che il robot venga considerato un umano), viene applicato un **Cropping** statico orizzontale (dal 35% al 60% della larghezza dell'immagine). MediaPipe elabora esclusivamente questa striscia verticale centrale.

Rendering e Feedback Visivo

L'interfaccia grafica è stata progettata per fornire un feedback chiaro sullo stato del sistema:

- **Cerchi Verdi/Blu:** Indicano i punti anatomici scansionati e disponibili in memoria.
- **Cerchio Rosso (Selezione):** Quando l'operatore preme un tasto, il punto selezionato viene evidenziato in rosso e viene mostrato il messaggio «SENDING: [PARTE]».
- **Testo di Stato:** In alto a sinistra viene sempre indicata la modalità operativa («SCANSIONE» o «MODE: MEMORY (Robot safe)»).

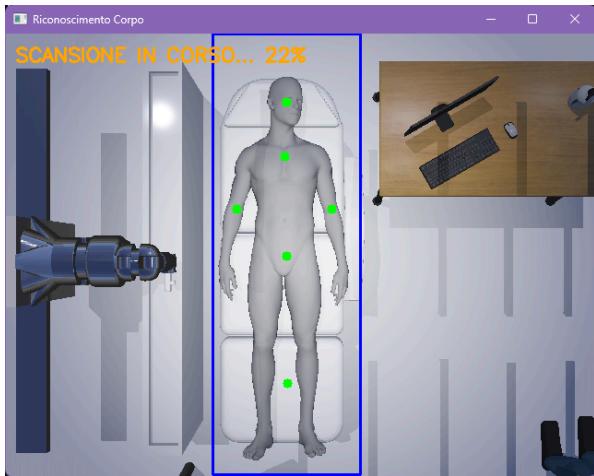


Figura 17: Fase di Scansione: Il sistema acquisisce la mappa 3D del paziente.

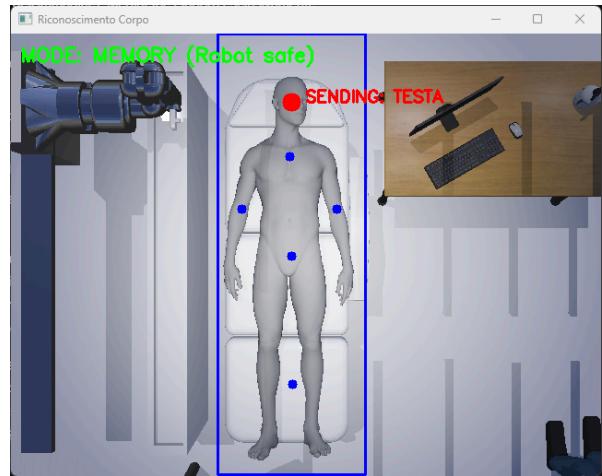


Figura 18: Modalità Memoria: L'utente seleziona «Testa» tramite tastiera.

Socket Server

Il controller della camera non si limita all'elaborazione visiva, ma agisce come un **Socket Server** non bloccante. Questo design permette di disaccoppiare la logica di simulazione (che deve girare a 32ms costanti) dagli input utente asincroni.

È stato utilizzato il modulo `select` di Python per gestire il multiplexing degli input I/O senza interrompere il loop fisico di Webots:

```
server_socket.setblocking(False)
inputs = [server_socket]

while robot.step(timestep) != -1:
    readable, _, _ = select.select(inputs, [], [], 0)
    for s in readable:
        if s is server_socket:
            # Nuova connessione dalla GUI
            conn, addr = s.accept()
            inputs.append(conn)
        else:
            # Ricezione comando (es. "BRACCIO_DX")
            data = s.recv(1024)
```

GUI

Per la fase di testing, abbiamo creato una semplice interfaccia grafica per facilitare la fase di testing.

La GUI funge da Client TCP e offre le seguenti funzionalità:

1. Mappa Anatomica Intuitiva: Pulsanti disposti spazialmente per rappresentare la topologia del paziente (Testa in alto, Braccia ai lati, Gambe in basso).
2. Feedback di Connessione: Un thread separato esegue il polling sullo stato del socket, indicando visivamente (Verde/Rosso) se la simulazione Webots è attiva e raggiungibile.
3. Log Operativo: Visualizzazione in tempo reale dei comandi inviati e degli eventuali errori di comunicazione.



Figura 19: GUI disconnessa



Figura 20: GUI connessa



Figura 21: GUI: invio dati

Comunicazione Inter-Processo (Pipeline Completa)

Il flusso dati completo per l'esecuzione di un comando segue ora questa pipeline:

1. Utente: Clicca «BRACCIO_DX» sulla GUI.
2. Socket Client: Invia la stringa «BRACCIO_DX» su `localhost:65432`.
3. Camera Controller (Server):
 - Riceve la stringa tramite select.
 - Effettua il lookup nel dizionario: `SCANNED_COORDS["BRACCIO_DX"]`.
 - Recupera le coordinate statiche (X, Y, Z).
4. Emitter: Invia al robot la stringa formattata «`BRACCIO_DX,0.35,-0.10,0.90`».
5. Robot Controller: Riceve il messaggio, ed esegue il movimento.

1.5.5 Controllo del Robot e Simulazione Ecografica

La logica di attuazione è gestita da un secondo controller indipendente (`robot_controller.py`), che agisce come nodo «ricevitore» e attuatore fisico. La progettazione di questo componente ha subito una significativa evoluzione nel corso del progetto, passando da un tentativo di controllo completamente dinamico a una strategia ibrida più robusta.

La Sfida della Cinematica Inversa

Inizialmente, l'obiettivo era implementare un controllo basato interamente sulla **Cinematica Inversa (IK)** utilizzando la libreria Python `ikpy`. L'intento era generare dinamicamente le configurazioni dei giunti (q_1, q_2, \dots, q_n) a partire dalle coordinate cartesiane (x, y, z) fornite dalla visione artificiale.

Tuttavia, l'integrazione di `ikpy` con il modello custom del robot in Webots ha evidenziato diverse problematiche bloccanti:

1. **Discrepanze nella Catena Cinematica (URDF):** L'esportazione del modello da Fusion 360 al formato URDF (necessario per `ikpy`) ha generato conflitti nella definizione dei frame di riferimento (sistemi di coordinate locali dei giunti). Questo portava il solver a calcolare soluzioni matematicamente valide ma fisicamente impossibili (es. gomiti che si piegano al contrario).
2. **Instabilità nelle Singolarità:** Nelle configurazioni estese (necessarie per raggiungere il lato opposto del lettino), il solver tendeva a fallire la convergenza o a proporre movimenti violenti per «scappare» dalle singolarità matematiche.
3. **Complessità Computazionale:** Il calcolo iterativo della IK a ogni time-step (32ms) introduceva latenze non trascurabili nella simulazione, riducendo la fluidità operativa.

A fronte di queste criticità, si è optato per un cambio di strategia architetturale.

Soluzione Implementata: Controllo Irido

La soluzione finale adotta un approccio deterministico che combina il posizionamento dinamico della base (Slider) con una libreria di pose pre-calcolate per il braccio.

Valutazione del Trade-off: Flessibilità vs Robustezza

L'adozione di questa architettura comporta, inevitabilmente, una rinuncia alla **generalità operativa**: il robot non possiede più la libertà di raggiungere coordinate spaziali arbitrarie (x, y, z), ma è vincolato a un set finito di configurazioni.

Tuttavia, questa limitazione è stata accettata consapevolmente per garantire la riuscita della **Proof of Concept**. Mentre l'approccio IK rendeva il sistema instabile, la soluzione basata su libreria di pose ha permesso di ottenere una **dimostrazione funzionante e ripetibile**. In questa fase di prototipazione, la priorità è stata assegnata alla **robustezza** dell'esecuzione: il sistema è ora in grado di completare l'intero ciclo operativo (scansione → invio comando → attuazione fisica) senza errori, validando efficacemente l'integrazione tra i sottosistemi di Visione e Robotica.

Architettura di Sicurezza: «Safe-Lift Maneuver»

Una criticità emersa durante i primi test riguardava il movimento di transizione tra due pose (es. dal Torace all'Addome). Spostando contemporaneamente lo slider (asse Y) e i giunti del braccio, il robot rischiava di trascinare la sonda sul corpo del paziente, esercitando una pressione non controllata. Per risolvere questo problema, è stata implementata una logica di controllo sequenziale a due fasi, gestita tramite un timer interno (movement_timer):

1. **Fase 1: Sollevamento di Sicurezza (Lift & Hold).** Alla ricezione del comando, il robot calcola il target dello slider ma congela il movimento del carrello (pending_slider_pos). Immediatamente, vengono attivati solo i motori verticali della spalla e del gomito (LIFT_MOTORS) per portare il braccio in una posizione di riposo alta («IDLE»).
2. **Fase 2: Posizionamento e Attuazione.** Solo dopo che il timer di sicurezza (SAFE_LIFT_DURATION) ha garantito il sollevamento dell'effettore, il controller sblocca il motore dello slider per raggiungere la coordinata Y target e, contemporaneamente, applica la posa finale richiesta (es. «ADDOME»).

Questo approccio disaccoppia il movimento di spostamento da quello di posizionamento, creando una traiettoria «a ponte» che scavalca il paziente senza mai toccarlo durante il transito.

```
# Pseudocodice della logica di sicurezza implementata
while robot.step(timestep) != -1:
    if message_received:
        # FASE 1: Calcola ma aspetta
        pending_slider = calculate_slider_y(message)
        pending_pose   = POSE_LIBRARY[target_part]

        # Alza il braccio subito (Safety Lift)
        set_motors(LIFT_MOTORS, IDLE_POSITION)

        # Avvia timer di sicurezza
        timer = SAFE_LIFT_DURATION

    # FASE 2: Esegui dopo il timer
    if timer > 0:
        timer -= 1
        if timer == 0:
            # Ora è sicuro muovere lo slider e abbassare il braccio
```

```

    slider.setPosition(pending_slider)
    set_all_motors(pending_pose)

```

Dettagli Implementativi

La logica di controllo si divide in due flussi paralleli:

- Controllo Dinamico dello Slider (Asse Y):** L'asse lineare lungo il lettino mantiene un comportamento dinamico guidato dalla visione. Il controller riceve la coordinata Y della telecamera (`raw_cam_y`) e la utilizza per pilotare direttamente il motore lineare della base. Questo garantisce che il robot «insegua» il paziente lungo il lettino in tempo reale. Sono stati implementati limiti software (`LIMIT_SLIDER = 0.8`) e funzioni di clamping per garantire che il carrello non esca mai dai binari.
- Controllo a Stati (Pose Library):** Per gli 8 gradi di libertà del braccio, è stata implementata una logica a «Look-up Table». È stato definito un dizionario `POSE_LIBRARY` che contiene configurazioni angolari sicure e validate per ogni macro-regione anatomica.

```

POSE_LIBRARY = {
    # [Pistone, Gomito, Spalla_Roll, Spalla_Pitch, ...]
    "IDLE": [0.00, 0.0, 0.0, 1.57, 1.57, 0.0, -1.57, 0.0],
    "TORACE": [0.05, -0.6, 0.0, 0.50, 1.00, 4.5, 0.00, 0.2],
    # Altre pose configurate...
}

```

Il ciclo principale del controller opera in ascolto continuo sul nodo Receiver, orchestrando i due flussi:

```

# Esempio semplificato della logica di controllo
while robot.step(timestep) != -1:
    if receiver.getQueueLength() > 0:
        message = receiver.getString()
        parts = message.split(',')
        target_part = parts[0]
        y_coord = float(parts[2])

        # 1. Attuazione Dinamica Slider (Tracking longitudinale)
        slider_cmd = clamp(y_coord, -LIMIT_SLIDER, LIMIT_SLIDER)
        slider.setPosition(slider_cmd)

        # 2. Attuazione Statica Braccio (Pose pre-validate)
        if target_part in POSE_LIBRARY:
            set_motors(POSE_LIBRARY[target_part])
        else:
            # Fallback di sicurezza
            set_motors(POSE_LIBRARY["IDLE"])

```

Questa architettura ibrida offre il miglior compromesso: garantisce la copertura totale del corpo (grazie allo slider dinamico) e la sicurezza assoluta delle posture di scansione (grazie alle pose pre-validate), eliminando i rischi di collisione dovuti a errori di calcolo della cinematica inversa.

2 Riferimenti e Codice Sorgente

Non essendo il progetto basato su una specifica letteratura accademica pregressa, i riferimenti principali riguardano la documentazione tecnica dei framework utilizzati.

Tutto il materiale di sviluppo, inclusi i file di simulazione del mondo Webots (.wbt), i controller Python, gli script dell’interfaccia grafica e i modelli 3D ottimizzati, è open source e disponibile nel repository ufficiale del progetto.



2.1 Documentazione Tecnica e Risorse

Per lo sviluppo dell’architettura software e della simulazione fisica sono state consultate le seguenti risorse ufficiali:

- **Python 3.11 Environment**

Linguaggio scelto per lo sviluppo di tutti i controller e dell’interfaccia utente.

Librerie principali utilizzate:

- *MediaPipe & OpenCV (cv2)*: Per la pipeline di Computer Vision e l’elaborazione immagini.
- *NumPy*: Per il calcolo matriciale e la gestione delle coordinate spaziali.
- *Socket & Select*: Per la gestione della comunicazione asincrona TCP/IP.
- *CustomTkinter*: Per lo sviluppo della GUI moderna e responsive.

- **Autodesk Fusion 360 (CAD file access)**

Piattaforma utilizzata per la modellazione parametrica, l’analisi delle interferenze e l’assemblaggio dei componenti. Il progetto completo è accessibile tramite il seguente hub:

→ [EchoArmCad](#)

- **Cyberbotics Webots Reference Manual**

Documentazione ufficiale per la configurazione dei nodi fisici (LinearMotor, RangeFinder) e l’API di simulazione.

→ [cyberbotics.com/doc/guide](#)

- **Google MediaPipe Solutions**

Riferimento per l’implementazione del modello *Pose Landmark Detection* e l’ottimizzazione delle performance di inferenza.

→ [developers.google.com/mediapipe](#)