



UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA MECCATRONICA

TESI DI LAUREA MAGISTRALE

SVILUPPO DI SISTEMI DI CONTROLLO
PER SISTEMI TELERBOTICI
CON RETROAZIONE DI FORZA
IN APPLICAZIONI DI ECOGRAFIA
SENZA CONTATTO

Relatore: Ch.mo Prof. ROBERTO OBOE

Correlatore: Ch.mo Ing. RICCARDO ANTONELLO

Laureanda: CLAUDIA VISENTIN

Matricola 620757-IMC

ANNO ACCADEMICO 2011-2012

Sommario

In questo elaborato vengono presentate l'architettura e le strategie di controllo di un sistema master/slave con feedback aptico di forza, sviluppato per applicazioni di ecografia senza contatto. In particolare si desidera fornire uno strumento efficace per la diagnosi dell'artrite reumatoide. Questa malattia, infatti, è ora facilmente diagnosticabile anche in fase precoce attraverso l'ecografia in immersione con mezzo di contrasto (CEUS). L'esame, rapido ed economico, richiede però un posizionamento preciso della sonda per ottenere dei risultati soddisfacenti.

L'architettura scelta è costituita principalmente dal dispositivo master *PHANToM Omni* e dal robot slave *Thermo CRS A465* antropomorfo a sei gradi di libertà. All'estremità di quest'ultimo è stato alloggiato un sensore ottico di prossimità, in grado di fornire la distanza dell'end effector dall'oggetto in analisi.

Così facendo l'operatore ha la possibilità di muovere agevolmente a distanza la sonda ecografica sopra l'area di indagine. Allo stesso tempo, grazie all'informazione restituita dal sensore, il sistema è in grado di orientare la sonda ortogonalmente rispetto al target e di mantenerla ad una distanza ottimale dalla superficie in analisi. Una forza visco-elastica, proporzionale alla distanza dal target, viene inoltre ritornata all'operatore. Essa determina di fatto un vincolo virtuale, oltre il quale l'end effector del robot non potrà spingersi.

I risultati sperimentali hanno dimostrato l'effettiva validità della soluzione proposta, dimostrando come essa potrebbe costituire una buona base di partenza per un ulteriore sviluppo di questo progetto.

Indice

| | |
|--|------------|
| Sommario | iii |
| Indice | v |
| Elenco delle tabelle | ix |
| Elenco delle figure | xi |
| 1 Introduzione | 1 |
| 2 Descrizione dei dispositivi del sistema | 5 |
| 2.1 Dispositivo Master: Phantom Omni | 6 |
| 2.1.1 Specifiche tecniche | 7 |
| 2.1.2 Modello cinematico e sistemi di riferimento | 8 |
| 2.2 Dispositivo Slave: Thermo CRS A465 | 8 |
| 2.2.1 Specifiche tecniche | 9 |
| 2.2.2 Controllore C500C | 11 |
| 2.2.3 Modello cinematico e sistemi di riferimento | 12 |
| 2.3 Scheda di acquisizione Quanser MultiQ-3 | 17 |
| 2.3.1 Conversione Analogico/Digitale (A/D) | 17 |
| 2.3.2 Uscite analogiche | 17 |
| 2.3.3 Ingressi provenienti dagli encoder | 17 |
| 2.3.4 Ingressi digitali | 18 |
| 2.3.5 Uscite digitali | 18 |
| 2.3.6 Clock real-time | 18 |
| 2.4 Sensore ottico di prossimità | 19 |
| 2.4.1 Caratterizzazione del sensore | 20 |
| 2.4.2 Sistemi di riferimento | 21 |
| 2.4.3 Scheda di acquisizione e condizionamento del segnale | 22 |
| 3 Integrazione dei dispositivi all'ambiente Matlab Simulink | 25 |
| 3.1 Dispositivo Master: Phantom Omni | 25 |
| 3.1.1 Blocco OmniDevice | 25 |
| 3.1.2 Codifica dei riferimenti | 26 |
| 3.1.3 Coppie di attuazione e compensazione della gravità | 28 |
| 3.2 Dispositivo Slave: Thermo CRS A465 | 28 |
| 3.2.1 Cinematica Inversa | 29 |
| 3.2.2 Switch Slow sigmoid | 29 |
| 3.2.3 Limitazione degli angoli | 30 |
| 3.2.4 Controllore | 30 |
| 3.2.5 Cinematica diretta | 32 |
| 3.3 Sensore ottico di prossimità | 32 |
| 3.3.1 Get sensor posture | 33 |
| 3.3.2 Safety block e abilitazione del sensore | 34 |

| | | |
|----------|--|-----------|
| 4 | Architettura del sistema di controllo | 37 |
| 4.1 | Architettura di controllo master/slave a due canali | 37 |
| 4.2 | Controllo automatico dell'orientazione dell'end effector | 40 |
| 4.2.1 | Criteri di commutazione del riferimento del dispositivo slave con controllo automatico dell'orientazione | 41 |
| 4.3 | Controllo automatico della distanza dell'end effector dal target | 42 |
| 4.3.1 | Criteri di commutazione del riferimento del dispositivo slave con controllo automatico della distanza minima dal piano | 44 |
| 4.4 | Generazione del feedback di forza | 45 |
| 4.4.1 | Generazione di una forza sinusoidale sovrapposta | 46 |
| 5 | Implementazione del software di controllo | 47 |
| 5.1 | Controllo automatico dell'orientazione dell'end effector | 47 |
| 5.1.1 | Criteri di commutazione del riferimento | 50 |
| 5.2 | Controllo automatico della distanza dell'end effector | 50 |
| 5.2.1 | Criteri di commutazione del riferimento | 52 |
| 5.3 | Generazione del feedback di forza | 52 |
| 5.3.1 | Forza sinusoidale sovrapposta nel controllo automatico di orientazione | 54 |
| 5.3.2 | Filtraggio del feedback di forza | 54 |
| 6 | Risultati sperimentali | 57 |
| 6.1 | Inseguimento di un punto della superficie | 58 |
| 6.2 | Inseguimento di una superficie piana | 61 |
| 6.3 | Inseguimento di una superficie curva | 66 |
| 7 | Conclusioni | 71 |
| 8 | Ringraziamenti | 73 |
| | Bibliografia | 75 |
| A | Cinematica diretta e inversa del robot Thermo CRS A465 | 77 |
| A.1 | Matrici di trasformazione | 77 |
| A.2 | Cinematica diretta | 78 |
| A.3 | Cinematica inversa | 78 |
| A.3.1 | Soluzione del giunto 1 | 78 |
| A.3.2 | Soluzione del giunto 3 | 79 |
| A.3.3 | Soluzione del giunto 2 | 79 |
| A.3.4 | Soluzione del giunto 4 | 80 |
| A.3.5 | Soluzione del giunto 5 | 80 |
| A.3.6 | Soluzione del giunto 6 | 80 |
| B | Parametrizzazione della matrice di rotazione R | 81 |
| B.1 | Codifica da notazione asse-angolo a matrice di rotazione | 81 |
| B.2 | Codifica da matrice di rotazione ad angoli di yaw - pitch - roll | 81 |
| C | Interfacciamento dei dispositivi: listati di codice | 83 |
| C.1 | Dispositivo Master Phantom Omni | 83 |
| C.1.1 | Codifica di orientazione | 83 |
| C.1.2 | Codifica di posizione | 83 |
| C.2 | Sensore ottico di prossimità: Stima della distanza media e dell'orientazione del target | 84 |

| | |
|--|-----------|
| D Controlli automatici: listati di codice | 87 |
| D.1 Controllo automatico dell'orientazione | 87 |
| D.2 Controllo automatico della distanza | 90 |
| D.3 Generazione del feedback di forza | 92 |

Elenco delle tabelle

| | | |
|-----|---|----|
| 2.1 | PHANToM Omni Device - Specifiche tecniche | 7 |
| 2.2 | Thermo CRS A465 - Caratteristiche generali | 9 |
| 2.3 | Thermo CRS A465 - Specifiche tecniche | 9 |
| 2.4 | Thermo CRS A465 - Limiti angolari dei giunti | 10 |
| 2.5 | Thermo CRS A465 - Parametri di Denavit Hartenberg | 13 |
| 2.6 | Thermo CRS A465 - Possibili soluzioni di cinematica inversa | 15 |
| 2.7 | Thermo CRS A465 - Direzione di misura positiva dei canali di uscita D/A degli encoder dei giunti | 16 |
| 2.8 | Thermo CRS A465 - Costanti di calibrazione dei giunti | 16 |
| 3.1 | Thermo CRS A465 - Saturazioni dei giunti | 30 |

Elenco delle figure

| | | |
|------|--|----|
| 2.1 | Schema di un sistema master/slave aptico per applicazioni ecografiche | 5 |
| 2.2 | Phantom Omni Device | 6 |
| 2.3 | Phantom Omni - Modello cinematico, vista laterale e vista dall'alto | 8 |
| 2.4 | Phantom Omni - Sistemi di riferimento | 8 |
| 2.5 | Thermo CRS A465 | 9 |
| 2.6 | Thermo CRS A465 - Spazio di lavoro, vista laterale, range dei giunti J_2 , J_3 , J_5 | 10 |
| 2.7 | Thermo CRS A465 - Spazio di lavoro, vista dall'alto, range del giunto J_1 . . | 10 |
| 2.8 | Thermo CRS A465 - Controllore C500C | 11 |
| 2.9 | Thermo CRS A465 - CRS/MultiQ Adapter Board (CRSMQ) | 12 |
| 2.10 | Thermo CRS A465 - Modalità di operazione in architettura aperta. MultiQ Analog and Digital I/O | 12 |
| 2.11 | Thermo CRS A465 - Diagramma di Denavit-Hartenberg | 13 |
| 2.12 | Thermo CRS A465 - Sistemi di riferimento | 14 |
| 2.13 | Thermo CRS A45 - Posizione di zero dei giunti: <i>Carldy position</i> e <i>Ready position</i> | 14 |
| 2.14 | Thermo CRS A465 - Posizioni assumibili in cinematica inversa $[1, 1, 0]$, $[1,$ $0, 1]$, $[0, 1, 0]$ | 15 |
| 2.15 | Scheda di acquisizione MultiQ3. | 17 |
| 2.16 | Scheda di acquisizione MultiQ3 - Schema del circuito | 18 |
| 2.17 | Sensore ottico di prossimità | 19 |
| 2.18 | Sensore ottico di prossimità - Diagramma di emissività del diodo LED . . . | 20 |
| 2.19 | Sensore ottico di prossimità - Diagramma di sensitività del fototransistor . | 20 |
| 2.20 | Sensore ottico di prossimità - Curve di calibrazione del sensore (caratteri- stiche I/O) | 21 |
| 2.21 | Sensore ottico di prossimità - Curva generica di calibrazione | 21 |
| 2.22 | Sensore ottico di prossimità - Sistema di riferimento collocato sul target, rispetto alla terna di riferimento del sensore | 22 |
| 2.23 | Sensore ottico di prossimità - Scheda di acquisizione e condizionamento dei segnali del sensore | 23 |
| 2.24 | Sensore ottico di prossimità - Schema di condizionamento del segnale | 23 |
| 3.1 | Phantom Omni - Blocco di interfacciamento | 25 |
| 3.2 | Phantom Omni - Codifiche di posizione ed orientazione | 26 |
| 3.3 | Phantom Omni - Codifica di orientazione | 27 |
| 3.4 | Thermo CRS A465 - Blocco di interfacciamento | 29 |
| 3.5 | Thermo CRS A465 - Blocco di cinematica inversa | 30 |
| 3.6 | Thermo CRS A465 - Blocco del controllore del robot | 31 |
| 3.7 | Thermo CRS A465 - Blocco regolatore PD | 32 |
| 3.8 | Sensore ottico di prossimità - Blocco di elaborazione del segnale | 32 |
| 3.9 | Sensore ottico di prossimità - Calcolo della distanza dell'end effector del robot dal target | 34 |
| 3.10 | Sensore ottico di prossimità - Blocco di abilitazione | 34 |
| 4.1 | Architettura generica di un sistema di controllo a due canali | 38 |
| 4.2 | Architettura generica di un sistema di controllo a due canali - Blocchi dinamici | 38 |
| 4.3 | Controllo automatico di orientazione - Macchina a stati finiti, definisce i criteri per il passaggio tra controllo manuale e automatico | 42 |

| | | |
|------|--|----|
| 4.4 | Controllo automatico della distanza - Distanza minima di avvicinamento dell'end effector a target | 43 |
| 4.5 | Controllo automatico della distanza - Macchina a stati finiti, definisce i criteri per il passaggio tra controllo manuale e automatico | 44 |
| 5.1 | Sistema di controllo master/slave implementato | 47 |
| 5.2 | Blocco di controllo automatico dell'orientazione - Ingressi e uscite | 47 |
| 5.3 | Blocco di controllo automatico della distanza - Ingressi e uscite | 50 |
| 5.4 | Blocco di generazione del feedback di forza - Ingressi e uscite | 52 |
| 6.1 | Setup sperimentale | 57 |
| 6.2 | Punto della superficie: inseguimento dei riferimenti di posizione | 58 |
| 6.3 | Punto della superficie: inseguimento dei riferimenti di orientazione | 58 |
| 6.4 | Punto della superficie: confronto fra riferimenti di posizione dei due dispositivi master e slave | 59 |
| 6.5 | Punto della superficie: confronto fra riferimenti di orientazione dei due dispositivi master e slave | 59 |
| 6.6 | Punto della superficie: andamento degli angoli di yaw e pitch misurati dal sensore | 60 |
| 6.7 | Punto della superficie: evoluzione temporale di forza di feedback e distanza media dalla superficie del target | 60 |
| 6.8 | Superficie piana: rappresentazione 3D delle traiettorie seguite | 61 |
| 6.9 | Superficie piana: proiezioni ortogonali delle traiettorie seguite | 61 |
| 6.10 | Superficie piana: traiettoria di approccio al target, piano YZ e XZ | 62 |
| 6.11 | Superficie piana: traiettoria di approccio al target, piano YX e rappresentazione 3D | 62 |
| 6.12 | Superficie piana: inseguimento dei riferimenti di posizione | 63 |
| 6.13 | Superficie piana: confronto fra riferimenti di posizione dei due dispositivi master e slave | 63 |
| 6.14 | Superficie piana: inseguimento dei riferimenti di orientazione | 64 |
| 6.15 | Superficie piana: confronto fra riferimenti di orientazione dei due dispositivi master e slave | 64 |
| 6.16 | Superficie piana: andamento degli angoli di yaw e pitch misurati dal sensore | 65 |
| 6.17 | Superficie piana: evoluzione temporale di forza di feedback e distanza media dalla superficie del target | 65 |
| 6.18 | Superficie curva: rappresentazione 3D delle traiettorie seguite | 66 |
| 6.19 | Superficie curva: proiezioni ortogonali delle traiettorie seguite | 66 |
| 6.20 | Superficie curva: traiettoria di approccio al target, piano YZ e XZ | 67 |
| 6.21 | Superficie curva: traiettoria di approccio al target, piano YX e rappresentazione 3D | 67 |
| 6.22 | Superficie curva: inseguimento dei riferimenti di posizione | 68 |
| 6.23 | Superficie curva: confronto fra riferimenti di posizione dei due dispositivi master e slave | 68 |
| 6.24 | Superficie curva: inseguimento dei riferimenti di orientazione | 69 |
| 6.25 | Superficie curva: confronto fra riferimenti di orientazione dei due dispositivi master e slave | 69 |
| 6.26 | Superficie curva: andamento degli angoli di yaw e pitch misurati dal sensore | 70 |
| 6.27 | Superficie curva: evoluzione temporale di forza di feedback e distanza media dalla superficie del target | 70 |
| B.1 | Definizione del versore \mathbf{w} e angolo θ | 81 |
| B.2 | Definizione degli angoli di roll, pitch e yaw, con i relativi versi di rotazione | 82 |

B.3 Composizione delle rotazioni degli angoli di yaw-pitch-roll 82

Introduzione

Le articolazioni della mano possono essere sede di diversi processi patologici, fra cui le varie forme di artrite. Le patologie infiammatorie reumatiche, come l'Artrite Reumatoide (AR), sono tra le principali cause di disabilità e possono portare all'inabilità al lavoro.

L'AR in particolare è una malattia cronica autoimmune che colpisce la membrana sinoviale delle articolazioni. Tale membrana reagisce all'infiammazione aumentando di volume e dando così origine al cosiddetto panno sinoviale. Quest'ultimo invade la cartilagine e ne provoca l'erosione e la graduale distruzione. Questo processo proliferativo si estende fino all'osso e l'infiammazione arriva ad interessare tutti i tessuti che circondano l'articolazione provocando in modo graduale la distruzione della stessa e l'invalidità in chi ne soffre.

Le articolazioni più precocemente e frequentemente colpite sono le metacarpofalangee (MCF) e le interfalangee prossimali (IFP). Il riscontro di alterazioni in queste articolazioni può quindi essere considerato indice di danno articolare diffuso.

La possibilità di identificare precocemente l'insorgere di questa malattia e di definirne il tipo e il grado di attività può permettere di impostare una terapia farmacologica adeguata, di monitorarla nel tempo e quindi di prevenire i danni causati dalla malattia, bloccandone l'evoluzione. Le terapie farmacologiche infatti, se avviate nelle prime fasi evolutive della malattia, consentono di ottenere una marcata riduzione dell'attività di quest'ultima, e un arresto nella progressione del danno strutturale. Nel caso in cui invece l'AR non venga diagnosticata in modo tempestivo, il paziente dovrà essere sottoposto a interventi chirurgici volti a posizionare delle protesi articolari, con gli inevitabili rischi collegati.

Di conseguenza il principale obiettivo in questo campo è il miglioramento dell'approccio clinico all'artrite, in particolare per quanto riguarda la sua diagnosi in fase precoce, la relativa valutazione prognostica e il monitoraggio dell'efficacia della terapia. A tale scopo si cercano indicatori oggettivi e ripetibili, che abbiano un notevole valore diagnostico e prognostico fin dalle prime fasi della patologia.

Negli ultimi anni è stato dimostrato che nelle fasi precoci della malattia il primo segno ad apparire nelle articolazione è la sinovite e solo successivamente si riscontra un consolidamento del danno osseo. Un altro interessante aspetto recentemente riscontrato è che la vascolarizzazione sinoviale sembra correlare con l'attività del panno reumatoide. Da qui la necessità di operazioni di scanning, in grado di registrare, evidenziare e valutare il grado e l'estensione della neo-vascolarizzazione intra-articolare.

La radiologia tradizionale, che da sempre riveste un ruolo centrale nella diagnosi e nel monitoraggio della malattia, consente tuttavia di valutare unicamente la morfologia e la struttura dei capi ossei. Essa evidenzia il danno articolare conseguente all'evoluzione della sinovite, fornendo una diagnosi tardiva della malattia. Il suo valore diagnostico risulta quindi molto limitato nelle fasi iniziali, dove le erosioni sono visibili solo nel 15% dei casi.

La risonanza magnetica (MRI) costituisce invece uno strumento di diagnosi efficace, presentando tuttavia alcuni svantaggi quali un costo e una durata elevati e notevoli controindicazioni in caso di pazienti dotati di impianti metallici, elettronici e nel caso di pazienti disabili o claustrofobici.

L'ecografia a contrasto (CEUS - *Contrast Enhanced gray scale UltraSonography*) è, invece, comunemente utilizzata per l'analisi della vascolarizzazione dei tessuti, al fine di valutare l'insorgere di tumori. Tale tecnologia risulta essere molto promettente anche per

le operazioni di scanning richieste per la diagnosi dell'AR, poiché consentirebbe di ridurre sensibilmente i tempi e i costi di scansione.

In particolare la CEUS è in grado di differenziare fra sinovie attive e inattive in modo più efficace e accurato rispetto alle altre tecniche ecografiche. Attraverso la somministrazione di un liquido di contrasto al paziente è possibile studiare il livello di vascolarizzazione delle articolazioni in real time. Le micro bolle presenti nel liquido, infatti, circolano all'interno dei vasi sanguigni per un periodo sufficientemente elevato, sono stabili e garantiscono un ottimo livello di contrasto delle zone d'interesse. L'esame in immersione consente infine una migliore focalizzazione del fascio sul campo vicino e un'ottimale risoluzione spaziale.

Per contro, la scansione in immersione richiede che la sonda resti ad una determinata distanza dalla superficie d'ispezione, rendendo in questo modo la qualità della scansione fortemente dipendente dalle movimentazioni dell'operatore e dalla sua percezione della distanza, che può essere facilmente falsata dalla rifrazione fra aria e acqua.

Risulta quindi molto vantaggioso poter disporre di uno strumento in grado di consentire un'acquisizione delle immagini accurata, rapida, ripetibile e standardizzabile, in grado di facilitare l'interpretazione dei risultati.

A tale scopo si è scelto di sviluppare un sistema master-slave robotizzato, in cui i movimenti di un robot *CRS A465* vengono comandati dall'operatore mediante un dispositivo aptico master *PHANTOM Omni*. Sull'end effector del robot vengono installate la sonda ecografica e un sensore ottico tri-assiale di prossimità, in grado di restituire l'informazione relativa alla distanza del robot dalla superficie in esame.

In questo modo l'operatore ha la possibilità di muovere agevolmente a distanza la sonda ecografica sopra l'area di indagine. Allo stesso tempo, grazie all'informazione restituita dal sensore ottico, il sistema è in grado di mantenere la sonda ad una distanza ottimale dalla superficie per l'acquisizione delle immagini, e di ritornare all'operatore una forza proporzionale alla distanza dal target, determinando in questo modo un vincolo virtuale, oltre il quale l'end effector del robot non potrà spingersi.

Si è proceduto innanzitutto all'ottimizzazione dell'interfaccia di comunicazione già esistente tra i dispositivi master e slave. Tale interfaccia è stata realizzata in ambiente MATLAB e Simulink, mentre la comunicazione fra i dispositivi è stata possibile grazie all'utilizzo della scheda di acquisizione MultiQ3.

Scopo della tesi è la realizzazione di un sistema in grado di inseguire le traiettorie imposte dall'operatore mediante il dispositivo master, ma allo stesso tempo in grado di garantire una corretta orientazione automatica della sonda rispetto alla superficie di indagine. Una volta raggiunta una determinata distanza dal target, l'operatore potrà percepire un vincolo virtuale che si estende lungo la superficie in analisi, grazie ad un ritorno di forza sul dispositivo master.

Nel presente elaborato si sofferma, inoltre, l'attenzione sulla possibilità di dotare il sistema di un controllo automatico della distanza massima dal target, operante in parallelo al controllo automatico dell'assetto della sonda.

In questo modo è possibile fissare una distanza di sicurezza o una distanza ottimale per la scansione, oltre la quale l'end effector non avanzerà. Questa soluzione garantisce una maggiore qualità delle immagini acquisite, e allo stesso tempo tutela il paziente da eventuali contatti involontari.

Nel momento in cui i dispositivi robotizzati sono destinati ad entrare a stretto contatto con le persone, infatti, è necessario che essi rispondano a determinati criteri di sicurezza, efficienza e affidabilità.

Il sistema implementato risulta notevolmente *user friendly* e, di conseguenza, facilmente utilizzabile dal personale medico, senza la necessità di possedere conoscenze tecniche

specifiche in materia.

Un sistema di questo tipo è in grado inoltre di superare i limiti operativi finora esistenti, legati principalmente all'instabilità della sonda e ai possibili errori di posizionamento che l'operatore può commettere. Altro vantaggio risiede infine nella possibilità di standardizzare i parametri di scansione e fornire immagini ripetibili, ai fini di realizzare un'analisi approfondita dell'evoluzione dei danni articolari.

Considerando l'aumento dell'aspettativa di vita della popolazione, con il conseguente prevedibile aumento delle spese mediche necessarie per ogni persona, risulta evidente come applicazioni di questo tipo, favorendo l'avvento di nuove metodologie diagnostiche e riducendo allo stesso tempo i costi del processo di diagnosi e prevenzione, possano portare ad un miglioramento della qualità del servizio sanitario.

Nel secondo capitolo vengono descritti i dispositivi che compongono il sistema master/slave proposto, focalizzando l'attenzione sulle caratteristiche costruttive del sensore di prossimità e sulle caratteristiche tecniche e cinematiche dei due dispositivi master e slave utilizzati. Il terzo capitolo presenta le modalità con cui i dispositivi sono stati integrati nell'ambiente Matlab Simulink, mentre nel capitolo 4 viene illustrata l'architettura del sistema di controllo implementato e le specifiche dello stesso. Nel capitolo 5 si descrive in modo approfondito l'implementazione dell'interfaccia software del sistema, mentre nel capitolo 6 si presentano i risultati sperimentali più significativi.

Descrizione dei dispositivi del sistema

Il sistema proposto in Fig. 2.1 implementa un'interfaccia bilaterale master/slave a due canali con feedback aptico e comprende i seguenti componenti:

- un braccio robotico (*dispositivo slave*) in grado di muovere con movimenti stabili e controllati la sonda ecografica installata sul suo end effector;
- un dispositivo aptico (*dispositivo master*), che consente all'operatore di guidare i movimenti del robot;
- un sensore ottico di prossimità, collocato a lato della sonda ecografica, in grado di misurare la distanza e l'orientazione della sonda dalla superficie ispezionata. L'informazione relativa alla distanza è usata dal dispositivo aptico per generare una forza di feedback che simuli la forza di reazione del vincolo virtuale, situato al di sopra della superficie passata in scansione. L'informazione riguardante l'orientazione, invece, viene posta in ingresso a un controllore automatico, operante a lato slave, con l'obiettivo di mantenere la sonda a una determinata orientazione rispetto alla superficie analizzata, possibilmente senza vibrazioni residue.

In questo capitolo verranno analizzati in dettaglio i diversi componenti della struttura in esame.

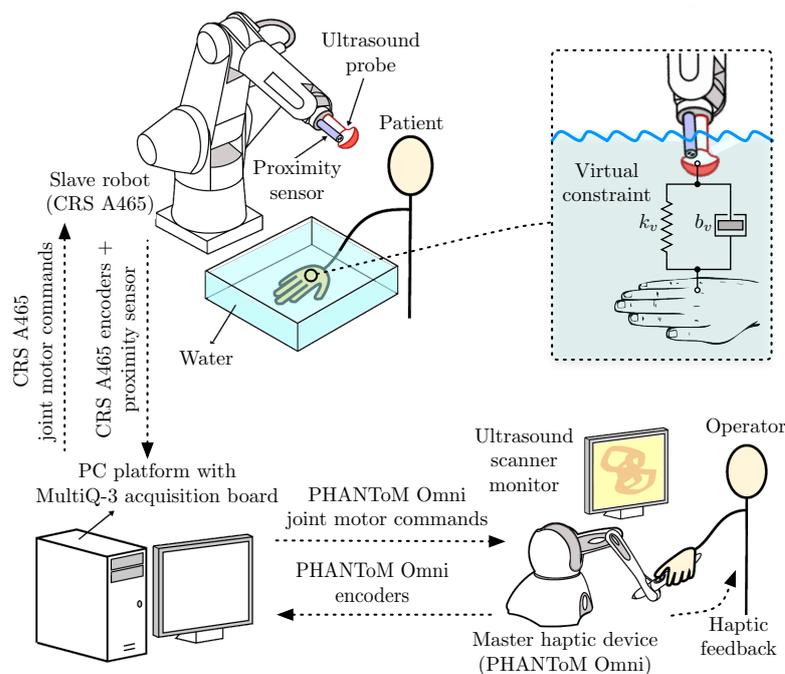


Fig. 2.1: Schema di un sistema master/slave aptico per applicazioni ecografiche

2.1 Dispositivo Master: Phantom Omni

Le interfacce aptiche sono dispositivi robotici studiati per interagire direttamente con l'operatore umano, dotati delle funzionalità necessarie a far percepire all'utente delle forze relative al contatto con un vincolo reale o virtuale simili a quelle che il mondo esterno eserciterebbe. Questi dispositivi sono stati concepiti sulla base dell'ipotesi che l'interazione fisica con l'ambiente circostante consista nella percezione di forze il cui valore è funzione della posizione degli arti dell'operatore usati per esplorare l'ambiente attraverso il dispositivo stesso.

Un dispositivo robotico può essere ritenuto idoneo alla simulazione dell'interazione fisica con l'ambiente, se possiede tre caratteristiche fondamentali:

- *Elevata trasparenza*: Durante la fase in cui non si determinano interazioni con l'ambiente circostante, l'utente non deve avvertire la presenza del dispositivo. In altri termini l'interfaccia aptica deve essere realizzata in modo da ridurre al minimo le forze di reazione necessarie al suo trascinamento, che questa esercita sull'operatore durante i suoi spostamenti nello spazio libero. Dal punto di vista tecnico, occorre ridurre al minimo le forze, indotte sull'operatore, dovute al peso proprio del dispositivo, agli attriti interni e alle masse in movimento (forze inerziali);
- *Elevata rigidità*: Durante la fase di interazione, il dispositivo deve essere in grado di indurre realisticamente la percezione del contatto con il vincolo virtuale imposto. Quest'ultimo, al limite, può essere infinitamente rigido. Il dispositivo quindi, dovrà essere in grado di mostrare all'operatore rigidità sufficientemente elevate;
- *Elevata banda passante*: Durante la transizione tra la fase di non contatto e quella di raggiungimento del vincolo virtuale, il dispositivo deve essere in grado di generare delle forze con elevata dinamica temporale (alto contenuto spettrale).



Fig. 2.2: Phantom Omni Device

PHANTOM® Omni Device è un dispositivo aptico prodotto da SensAble Technologies. Esso permette di toccare e manipolare oggetti virtuali con molta facilità, grazie ai suoi 6 gradi di libertà e all'ampio spazio di lavoro che è in grado di garantire, pur mantenendo dimensioni compatte. Il dispositivo fornisce la posizione e l'orientazione di uno stilo nello spazio cartesiano, in notazione asse-angolo o roll-pitch-yaw. Le coppie da esso generate come feedback risultano sufficientemente potenti per una chiara percezione del vincolo virtuale. Il collegamento con il sistema risulta veloce e immediato, grazie alla porta Firewire IEEE-1394a di cui è dotato.

2.1.1 Specifiche tecniche

In Tab. 2.1 vengono riportate le specifiche tecniche del dispositivo, mentre di seguito se ne elencano le principali caratteristiche:

- certificazione CE
- sei gradi di libertà nel rilevamento posizionale
- design compatto e portatile
- area di lavoro compatta, con conseguente migliore facilità d'uso
- comodo stilo in gomma strutturato per un uso a lungo termine e una presa sicura
- due interruttori integrati nello stilo per una miglior facilità d'uso e possibilità di personalizzazione da parte dell'utente finale
- Stylus-docking per la calibrazione automatica dell'area di lavoro

Tab. 2.1: PHANToM Omni Device - Specifiche tecniche

| | |
|--|---|
| Area di lavoro della forza di retroazione | 160W x 120H x 70D mm |
| Footprint (area fisica che la base del dispositivo occupa sul piano di appoggio) | 168W x 203D mm |
| Peso | 1.786 Kg |
| Range di movimento | Il movimento della mano con perno al polso |
| Risoluzione nominale di posizione | 0.055 mm |
| Attrito dei giunti | $\leq 0.26 N$ |
| Forza massima esercitabile alla posizione nominale (bracci ortogonali) | 3.3 N |
| Forza continua esercitabile (24 ore) | 0.88 N |
| Rigidezza | Asse X 1.26 N/mm Asse Y 2.31 N/mm Asse Z 1.02 N/mm |
| Inerzia (massa apparente alla punta) | 45 g |
| Forza di retroazione (feedback) | x, y, z |
| Rilevamento di posizione (Stylus cardanico) | x, y, z (encoder digitale) Pitch, roll e yaw (potenziometri con linearità $\pm 5\%$) |
| Interfaccia da 6 pin a 6 pin | Porta IEEE-1394 FireWire®: |
| Piattaforme supportate | PC Intel o AMD |

Analizzando le caratteristiche tecniche del dispositivo, esso risulta in grado di soddisfare le tre condizioni viste sopra, garantendo allo stesso tempo un ottimo rapporto qualità prezzo e risultando quindi idoneo ad essere utilizzato nell'applicazione presentata in questo elaborato.

2.1.2 Modello cinematico e sistemi di riferimento

In Fig. 2.3 viene riportato il modello cinematico del dispositivo nelle due viste, laterale e dall'alto, evidenziando i tre angoli θ_1 , θ_2 e θ_3 misurati dagli encoder digitali. In Fig. 2.4, invece, è possibile osservare la definizione del sistema di riferimento in corrispondenza dello snodo dello stilo. Grazie a dei potenziometri è possibile, infatti, ricavare l'informazione relativa a roll, pitch e yaw di quest'ultimo.

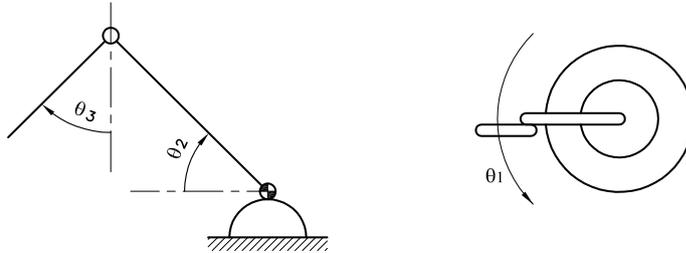


Fig. 2.3: Phantom Omni - Modello cinematico, vista laterale e vista dall'alto

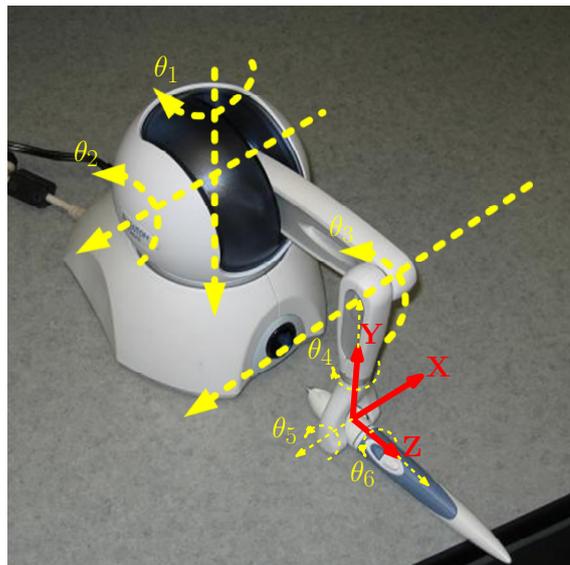


Fig. 2.4: Phantom Omni - Sistemi di riferimento

2.2 Dispositivo Slave: Thermo CRS A465

Il Thermo CRS A465 riportato in Fig. 2.5 è un robot antropomorfo a sei gradi di libertà con polso sferico e dispone quindi di 6 giunti, alimentati da 6 motori. Esso è stato progettato con lo stesso range di movimento e lo stesso carico utile del braccio umano, rendendo così ideale il suo utilizzo in applicazioni che richiedono movimenti articolati sia nel piano orizzontale, che in quello verticale. Il robot rappresenta una efficiente combinazione di automazione flessibile, elevata rapidità dei movimenti, affidabilità e facilità d'uso. Nella sezione 3.2 verrà ampiamente descritto il controllore indipendente, implementato attraverso sei controllori PD, con lo scopo di controllare la posizione e l'orientazione dei giunti del robot.



Fig. 2.5: Thermo CRS A465

2.2.1 Specifiche tecniche

Tab. 2.2: Thermo CRS A465 - Caratteristiche generali

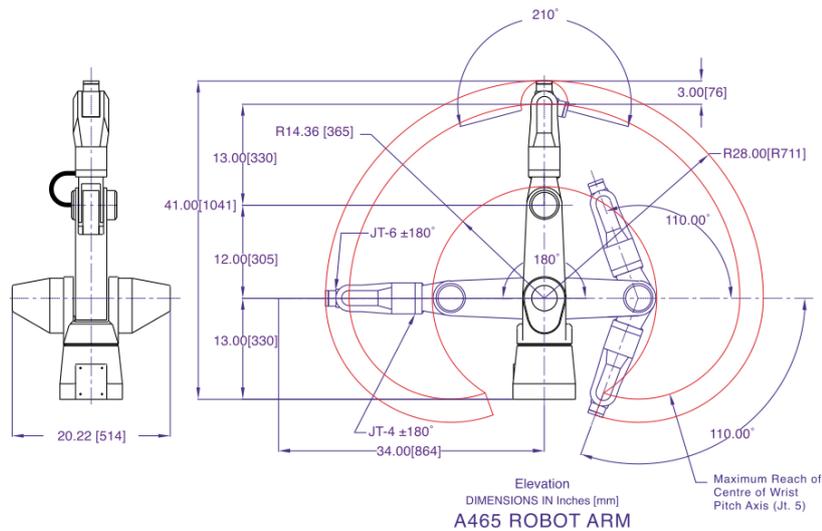
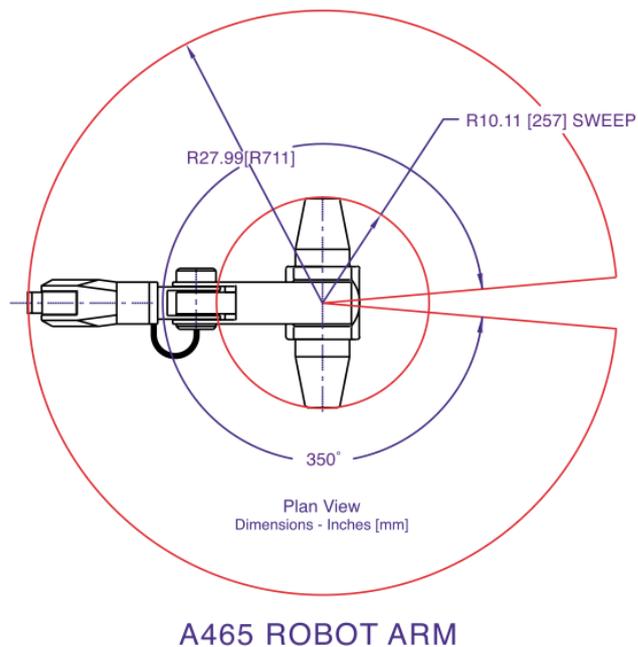
| | |
|-------------------|--|
| Giunti del robot | Articolati |
| Configurazione | 6 gradi di libertà |
| Drive | Servomotori Encoder di prossimità per ogni giunto |
| Trasmissione | Harmonic drive Cinghia di distribuzione |
| End-effector | Connettore pneumatico Servo gripper |
| C500C | Circuito integrato E-stop |
| Controller safety | Rilevazione continua dei guasti |

Tab. 2.3: Thermo CRS A465 - Specifiche tecniche

| | |
|----------------------------|----------------------|
| Carico utile nominale | 2 <i>Kg</i> |
| Estensione (senza gripper) | 711 <i>D mm</i> |
| Estensione (con gripper) | 864 <i>mm</i> |
| Ripetibilità | ± 0.05 <i>mm</i> |
| Peso | 31 <i>Kg</i> |

Tab. 2.4: Thermo CRS A465 - Limiti angolari dei giunti

| Giunto | Finecorsa meccanici [deg] | Velocità massima [deg/sec] |
|----------------|---------------------------|----------------------------|
| J_1 (vita) | ± 175 | 180 |
| J_2 (spalla) | ± 90 | 180 |
| J_3 (gomito) | ± 110 | 180 |
| J_4 (yaw) | ± 180 | 171 |
| J_5 (pitch) | ± 105 | 173 |
| J_6 (roll) | ± 180 | 171 |

Fig. 2.6: Thermo CRS A465 - Spazio di lavoro, vista laterale, range dei giunti J_2 , J_3 , J_5 Fig. 2.7: Thermo CRS A465 - Spazio di lavoro, vista dall'alto, range del giunto J_1

2.2.2 Controllore C500C



Fig. 2.8: Thermo CRS A465 - Controllore C500C

Il sistema viene dotato del controllore C500C. Esso presenta 6 controllori PID, agenti ciascuno su un motore del robot e i cui parametri non possono essere modificati. Durante il normale controllo dei giunti, tale controllore è in grado di eseguire oltre 30 processi contemporaneamente. Robcomm3 e ActiveX sono software di sviluppo che permettono di programmare il robot CRS-A465 in *close architecture* (CA) mediante l'utilizzo del linguaggio di programmazione RAPLIII. In questo modo, interfacciandosi con un PC attraverso una porta seriale oppure mediante il teach pendant in dotazione, è possibile far eseguire al robot dei compiti specifici.

In CA, tuttavia, non è possibile sviluppare software più avanzati. Per questo è possibile utilizzare il robot anche in *open architecture* (OA), modalità che è di fatto sempre stata adottata nelle prove sperimentali descritte nel presente elaborato.

Grazie all'interruttore CRS/MultiQ, è possibile passare agevolmente dalla struttura in OA, implementata mediante l'ambiente di simulazione Simulink, a quella in CA standard. Tale interruttore può anche venire controllato digitalmente dal PC, utilizzando la porta digitale 0 della scheda MultiQ, in Fig.2.10. Va inoltre evidenziato il fatto che, nel passaggio alla modalità OA, tutti i PID interni al controllore C500C vengono disattivati, lasciando il completo controllo dei singoli giunti al PC.

Si deve quindi prestare notevole attenzione in OA, in quanto un'instabilità del controllore o un comando scorretto ai giunti potrebbero portare a comportamenti incontrollati e pericolosi del robot. Per evitare pesanti danneggiamenti del robot e dell'apparecchiatura circostante, nel nostro caso sono stati predisposti dei finecorsa meccanici. In questo modo, se il giunto dovesse muoversi troppo velocemente, o dovesse avvicinarsi eccessivamente ai propri limiti meccanici, viene completamente tolta l'alimentazione al robot.

Per poter utilizzare il robot in OA, il controllore viene dotato della scheda *CRS/MultiQ Adapter Board* (CRSMQ) in Fig.2.9, posizionata sul retro del telaio del controllore e dotata di due slot per connettori, per consentire la comunicazione con la scheda MultiQ installata sul PC.

In Fig. 2.10 è possibile osservare gli ingressi e le uscite analogiche e digitali della scheda MultiQ. In particolare le linee magenta mostrano il percorso dall'uscita D/A (Digital to Analog) della scheda MultiQ agli amplificatori di potenza, e infine al motore. Questa condizione di lavoro è raggiunta quando l'interruttore è impostato in modalità manuale con SH2 o SH1 attivati, e i valori delle porte di uscita analogica e digitale sono posti a zero.



Fig. 2.9: Thermo CRS A465 - CRS/MultiQ Adapter Board (CRSMQ)

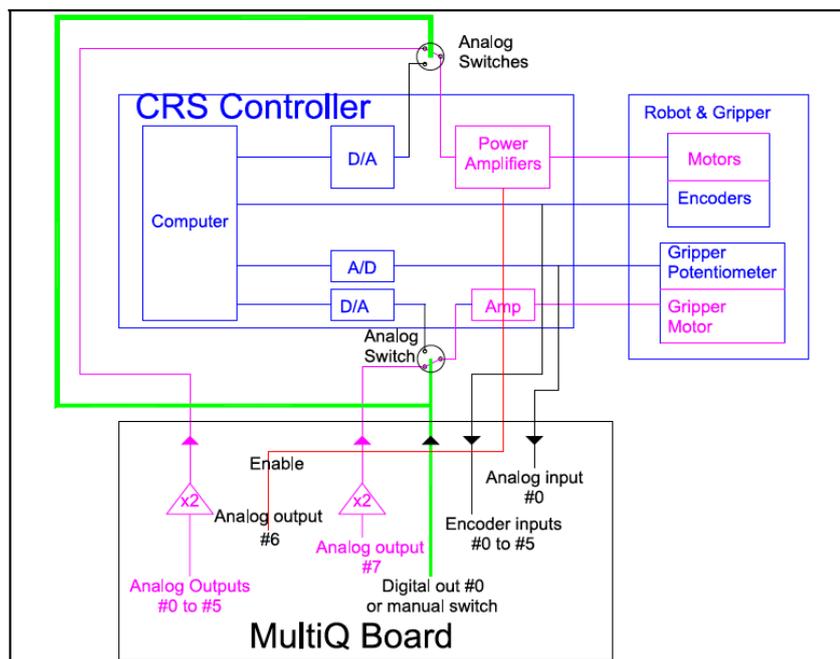


Fig. 2.10: Thermo CRS A465 - Modalità di operazione in architettura aperta. MultiQ Analog and Digital I/O

2.2.3 Modello cinematico e sistemi di riferimento

Notazione Denavit-Hartenberg

Nel determinare le formule di base che danno luogo alle trasformazioni necessarie per la cinematica diretta e inversa, è necessario descrivere matematicamente il braccio robotico. E' necessario quindi vincolare astrattamente delle terne locali ad ogni link del meccanismo, identificandole attraverso i tre versori convenzionali \mathbf{x} , \mathbf{y} , \mathbf{z} . Per fare ciò è stata utilizzata la *Notazione di Denavit-Hartenberg* (Notazione DH). Essa permette di definire le posizioni e le orientazioni delle terne di riferimento associate ai link di un robot in maniera inequivocabile e allo stesso tempo semplifica ed automatizza la procedura con cui si riesce a determinare la matrice di roto-traslazione che lega una terna a quella contigua.

Il primo passo consiste nel determinare i parametri DH, al fine di creare un diagramma che descriva in modo completo il meccanismo. Ogni sistema di riferimento può essere descritto rispetto al precedente attraverso quattro parametri denominati *parametri DH*.

In Tab. 2.5 sono riportati i parametri DH ricavati per il meccanismo in analisi, dove i vari simboli indicano rispettivamente:

- α_i : l'angolo misurato rispetto a x_i fra z_i e z_{i+1} ;
- a_i : la distanza dall'asse z_i all'asse z_{i+1} , misurata lungo l'asse x_i ;
- d_i : la distanza misurata lungo z_i fra x_{i-1} e x_i ;
- θ_i : l'angolo misurato rispetto a z_i fra x_{i-1} e x_i .

Tab. 2.5: Thermo CRS A465 - Parametri di Denavit Hartenberg

| i | α_{i-1} | a_{i-1} | d_i | θ_i |
|-----|----------------|-----------|----------|------------|
| 1 | 0 | 0 | 0 | θ_1 |
| 2 | $\pi/2$ | 0 | 0 | θ_2 |
| 3 | 0 | 304.8 mm | 0 | θ_3 |
| 4 | $-\pi/2$ | 0 | 330.2 mm | θ_4 |
| 5 | $\pi/2$ | 0 | 0 | θ_5 |
| 6 | $-\pi/2$ | 0 | 0 | θ_6 |

In Fig. 2.11 vengono riportati i sistemi di riferimento relativi a ciascun giunto secondo la notazione DH, si nota come la terna solidale all'end effector abbia la stessa orientazione rispetto alla terna base; le altre terne sono definite invece a seconda delle convenzioni adottate e descritte in Tab. 2.5.

In Fig. 2.12, infine, vengono riportati direttamente sul robot il sistema di riferimento (x, y, z) dell'end effector e i versi di rotazione positiva di ciascun giunto. Quest'ultimi sono stati definiti secondo la convenzione che il verso positivo corrisponda ad una rotazione in senso antiorario rispetto all'asse del giunto stesso.

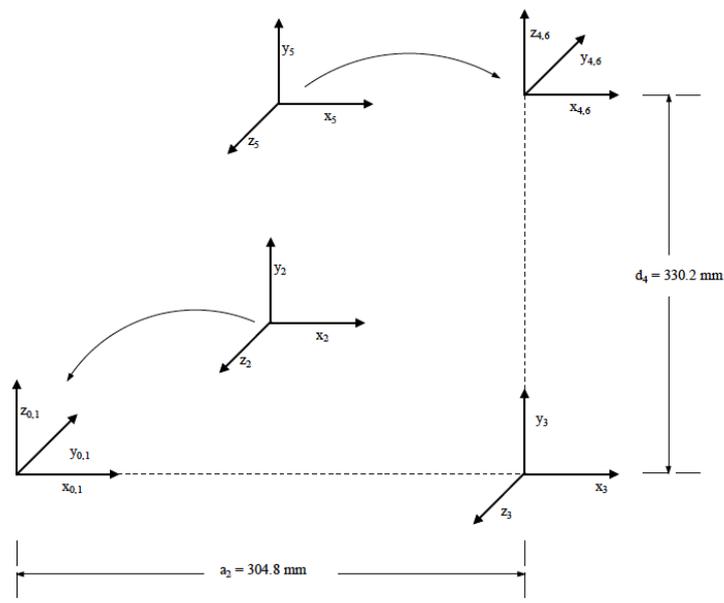


Fig. 2.11: Thermo CRS A465 - Diagramma di Denavit-Hartenberg

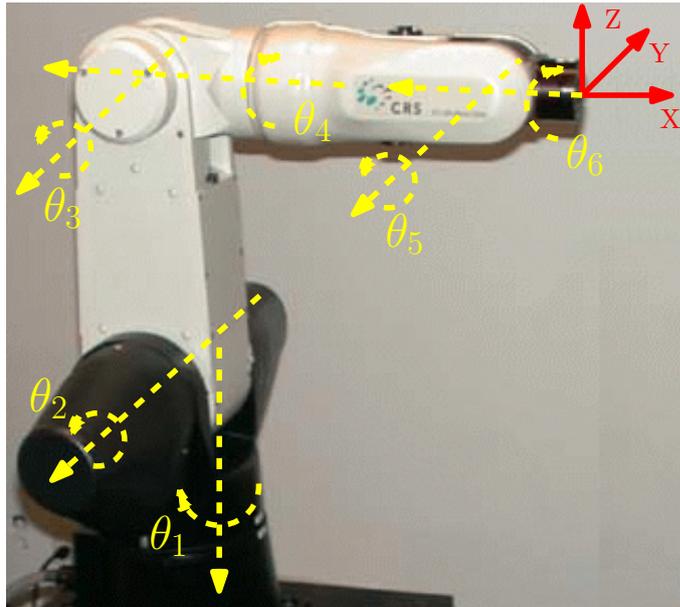


Fig. 2.12: Thermo CRS A465 - Sistemi di riferimento

Cinematica diretta ed inversa

Al fine di movimentare l'end effector del robot verso una desiderata posizione e con una certa orientazione rispetto alla terna di riferimento base, è necessario definire le equazioni che descrivono la cinematica diretta e inversa del robot stesso. Quest'ultime consentono rispettivamente di descrivere la posizione e l'orientazione dell'end effector (generalmente della flangia) dati gli angoli dei giunti, e viceversa. Posizione e orientazione relative a un dato sistema di riferimento sono spesso descritte mediante una matrice di trasformazione 4×4 . Allo scopo di non appesantire la presente trattazione, si riportano i calcoli matriciali completi necessari per ottenere le equazioni di cinematica diretta ed inversa in appendice [A](#).



Fig. 2.13: Thermo CRS A45 - Posizione di zero dei giunti: *Carldy position* e *Ready position*

Va comunque sottolineato come la soluzione delle equazioni di cinematica diretta richieda gli angoli dei vari giunti rispetto alla *Calrdy position* del robot, riportata in Fig. 2.13. Essa costituisce la 'posizione di zero' dei vari giunto nel *kinematic space*. Nell'operare in OA, ovvero lavorando nel cosiddetto *user space*, invece, la 'posizione di zero' dei giunti è identificata con la *Ready position*, sempre riportata in Fig. 2.13.

Per quanto riguarda invece la soluzione delle equazioni di cinematica inversa, è importante notare come sia possibile ottenere più soluzioni equivalenti al medesimo problema, ovvero l'end effector può raggiungere la posizione desiderata con diverse misure degli angoli dei giunti.

Il blocco di soluzione della cinematica inversa accetterà quindi un vettore di tre variabili, tali da specificare la posizione che si vuole che il robot assuma nella soluzione del problema. Le differenti possibilità sono riportate in Tab.2.6, mentre in Fig.2.14 si possono osservare le corrispondenti posizioni assunte dal robot.

Tab. 2.6: Thermo CRS A465 - Possibili soluzioni di cinematica inversa

| Stance value | 0 | 1 | -1 |
|--------------|----------|-------------|----------------------------|
| Reach | Backward | Forward | Best solution will be find |
| Elbow | Down | Up | Best solution will be find |
| Wrist | Flipped | Not Flipped | Best solution will be find |



Fig. 2.14: Thermo CRS A465 - Posizioni assumibili in cinematica inversa $[1, 1, 0]$, $[1, 0, 1]$, $[0, 1, 0]$

Encoder e acquisizione dati

A conclusione della presente sezione si ritiene opportuno precisare alcuni aspetti relativi alla modalità di acquisizione dei dati provenienti dagli encoder installati nel dispositivo slave, al fine di rendere più comprensibili alcune scelte che verranno prese nei successivi capitoli, in particolare nelle sezioni che descriveranno le interfacce dei dispositivi.

Ogni giunto del robot Thermo CRS A 465 è costituito da un motore e un riduttore. I valori di comando vengono forniti dal convertitore D/A della scheda MultiQ (operando in OA) o dal controllore C500 (operando in CA) e passati poi ad un amplificatore, in cui il range dei valori di ingresso varia tra $\pm 10V$, per comandare infine i motori.

Ciascun motore è dotato a sua volta di un encoder incrementale, il quale consente di misurare gli angoli dei giunti. In CA la misura proveniente dagli encoder viene data in ingresso a un computer, il quale esegue il programma scritto in RAPLIII e fornisce in uscita dei valori in Volt, i quali a loro volta, dopo essere passati per un convertitore D/A e attraverso gli amplificatori, commanderanno i motori.

Nel momento in cui invece si opera in OA, come nel caso del sistema in analisi, tutti i segnali vengono instradati da e verso la scheda MultiQ. In questo modo si ottiene il

pieno controllo sui motori, sulle misure dirette che provengono dagli encoder e sugli altri segnali rilevanti per il controllo dei dispositivi. In questa modalità di lavoro è quindi indispensabile tenere in considerazione tutte le caratteristiche degli encoder nello sviluppo del software di interfaccia.

Va sottolineato in primo luogo, come è necessario che all'applicazione di un voltaggio positivo al motore debba corrispondere un angolo positivo di rotazione del giunto. In Tab.2.7 si riassumono a tale scopo le direzioni positive di moto dei sei giunti e di conseguenza il segno positivo di voltaggio da applicare ai diversi motori.

Tab. 2.7: Thermo CRS A465 - Direzione di misura positiva dei canali di uscita D/A degli encoder dei giunti

| Encoder/ Joint nr. | D/A output Ch. | Direction of measurement for positive D/A volts |
|-----------------------|-------------------|--|
| 1 | 0 | +ve |
| 2 | 1 | +ve |
| 3 | 2 | -ve |
| 4 | 3 | -ve |
| 5 | 4 | -ve |
| 6 | 5 | -ve |

Gli encoder sono inoltre connessi agli *encoder channels* della scheda MultiQ mediante la CRSMA Adapter Board e in Tab. 2.8 vengono fornite le costanti di calibrazione per ciascun giunto.

Tab. 2.8: Thermo CRS A465 - Costanti di calibrazione dei giunti

| Joint | Encoder lines | Gear ratio | Calibrat. const. | Encoder Ch | Motor D/A Ch |
|-------|---------------|------------|------------------|------------|--------------|
| 1 | 1000 | 100 | -9.0e-004 | 0 | 0 |
| 2 | 1000 | 100 | -9.0e-004 | 1 | 1 |
| 3 | 1000 | 100 | 9.0e-004 | 2 | 2 |
| 4 | 500 | 100 | 0.0018 | 3 | 3 |
| 5 | 500 | 100 | 0.0018 | 4 | 4 |
| 6 | 500 | 100 | 0.0018 | 5 | 5 |

Si nota in particolare come, anche in questo caso, le costanti di calibrazione abbiano un segno tale da fornire risultati positivi per le direzioni '+ve' definite in Tab.2.7.

Un ultimo aspetto fondamentale da tenere in considerazione è il fatto che gli encoder siano di tipo incrementale. Essi vengono quindi inizializzati a zero solo al momento dell'avvio (start-up). Ne deriva che in CA è necessario assicurarsi di essere in posizione di ready (Fig. 2.13) prima di avviare il controllore. In caso contrario, la posizione in cui ci si trova verrà interpretata come posizione di zero.

In OA, invece, ogni qualvolta venga avviato il software, se non correttamente azzerati, gli encoder continueranno ad incrementare il loro valore a partire dall'ultimo valore misurato nella precedente fase di 'run' del software, determinando in questo modo un comportamento scorretto ed imprevedibile del robot.

2.3 Scheda di acquisizione Quanser MultiQ-3

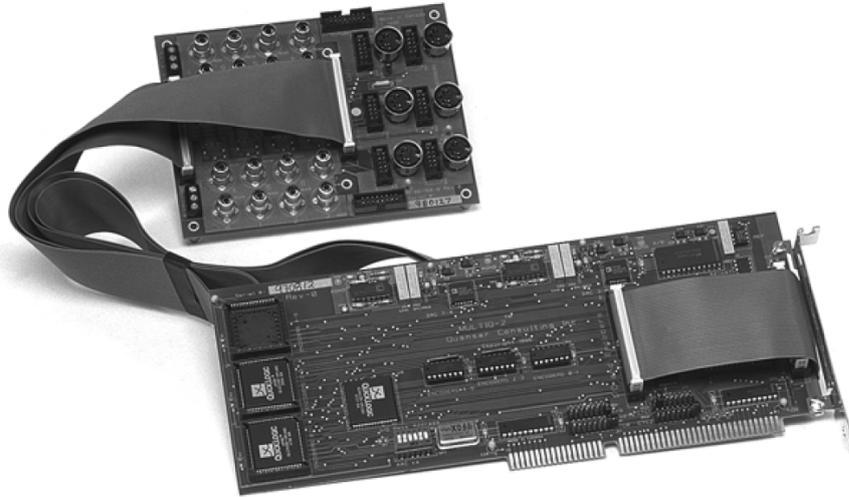


Fig. 2.15: Scheda di acquisizione MultiQ3.

MultiQ – 3 è una scheda di acquisizione e di controllo dati multiuso, che dispone di 8 ingressi analogici single ended, 8 uscite analogiche, 16 bit per gli ingressi digitali e 16 bit per le uscite digitali, 3 timer programmabili e fino a 8 ingressi per encoder, decodificati in quadratura.

Gli interrupt possono venire generati sia da uno dei tre timer, che da un ingresso digitale o dalla conclusione del processo di conversione A/D. Il sistema è accessibile attraverso il bus del PC ed è indirizzabile attraverso 16 locazioni di memoria mappate consecutive, le quali sono selezionate attraverso un interruttore DIP, collocato sulla scheda.

2.3.1 Conversione Analogico/Digitale (A/D)

La conversione A/D del MultiQ-3 è di tipo single ended bipolare a 13 bit (12 bit + 1 bit di segno). E' possibile eseguire una conversione in uno degli 8 canali semplicemente selezionando il canale e iniziando la conversione. Il dato è pronto e può essere letto quando nello STATUS REGISTER compare il bit EOC_I (end of conversion interrupt).

I dati vengono letti dall'AD DATA register e sono costituiti da due parole a 8 bit, che devono essere combinate per dare come risultato una parola a 16 bit con segno.

Il range dei valori in ingresso alla scheda è $\pm 5V$. Un ingresso di $+5V$ viene codificato con 0xFFFF, 0V con 0x0 e $-5V$ con 0xFFFF000.

Gli ingressi del multiplexer A/D sono segnali single ended nel range $\pm 5V$ e dovrebbero essere collegato ai jack RCA, etichettati come 'ingressi analogici'.

2.3.2 Uscite analogiche

I convertitori digitali-analogici (D/A) sono binari, a 12 bit senza segno. Un input di $-5V$ viene codificato con 0x000, 0V con 0x3FF e $+5V$ con 0xFFFF. I programmi creati dovranno quindi scrivere un numero a 12 bit (da 0 a 4095) nell'apposito registro. Le uscite analogiche cambiano, infine, quando i dati del registro sono aggiornati.

2.3.3 Ingressi provenienti dagli encoder

La scheda può essere dotata di un massimo di otto encoder decoder. I dati provenienti dagli encoder vengono decodificati in quadratura e incrementano o decrementano un contatore

a 2 bit. Con 24 bit è possibile ottenere fino a 16777215 conteggi. Dato un encoder lineare in quadratura, con risoluzione pari a 2000 impulsi/giro, ne deriva una risoluzione effettiva pari a 8000 impulsi/giro. Tale risoluzione può essere misurata senza incorrere in overflow dei contatori; conteggi superiori possono invece venire gestiti a livello software.

2.3.4 Ingressi digitali

La scheda può leggere 16 linee di ingresso digitali, connesse a un indirizzo I/O. Normalmente l'ingresso digitale è 'alto' (1); risulta invece 'basso' (0) quando la linea viene portata al potenziale GND. E' possibile, infine, collegare la linea di ingresso digitale 0 a un interrupt utilizzando i jumpers in dotazione.

2.3.5 Uscite digitali

La scheda è in grado di controllare 16 uscite digitali, connesse a un indirizzo I/O.

Scrivendo uno '0' sul bit appropriato si ottengono 0V (TTL LOW) in uscita, mentre scrivendo '1' l'uscita viene portata a 5V (TTL HIGH).

2.3.6 Clock real-time

La scheda è equipaggiata con tre clock (timer) indipendenti e programmabili. Ogni timer può essere programmato per lavorare a una frequenza compresa fra 2MHz e 30,52Hz. Per impostare una determinata frequenza di lavoro è sufficiente determinare un divisore (N) per il clock desiderato. Di conseguenza la frequenza si porterà ad un valore pari a $2/NMHz$. N è un valore intero a 16 bit, compreso tra 2 e 65535 (0xFFFF).

Infine, anche in questo caso l'uscita di uno dei tre clock può essere associata a una linea di interrupt, usando un jumper sulla scheda.

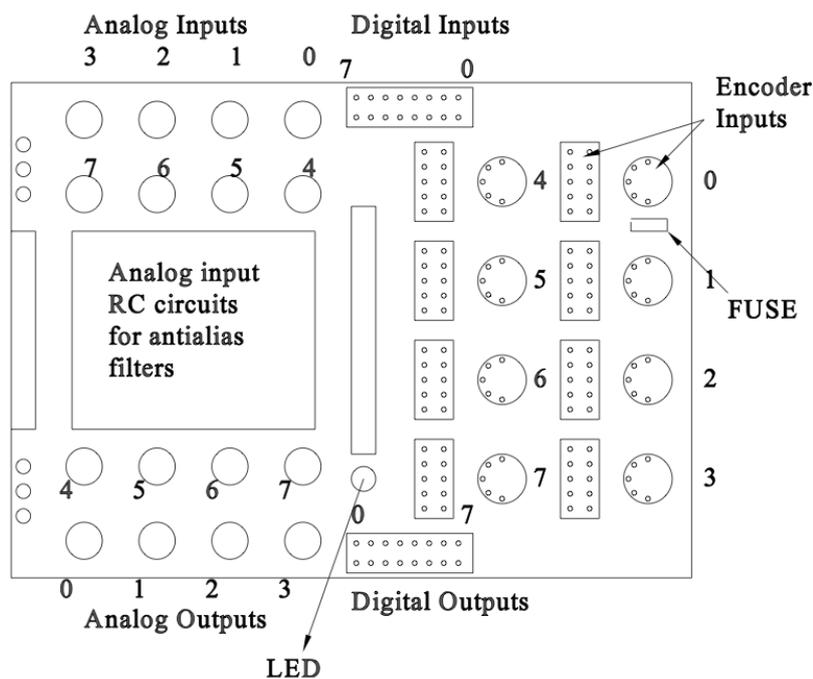


Fig. 2.16: Scheda di acquisizione MultiQ3 - Schema del circuito

2.4 Sensore ottico di prossimità

Al fine di misurare la distanza e l'orientazione dal target, sull'end effector del robot è stato installato il sensore ottico di prossimità di Fig.2.17. Esso è costituito da:

- un diodo LED infrarosso Osram LD 242 7800;
- tre fototransistor Sharp PT480F, posizionati agli angoli di un triangolo equilatero, in modo da risultare equidistanti dall'emettitore.

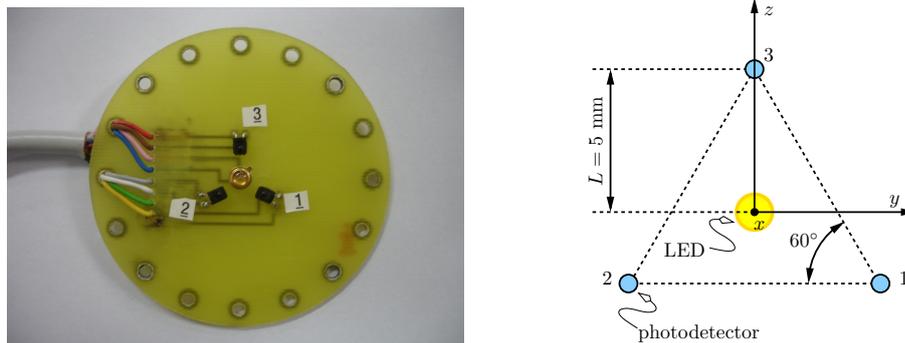


Fig. 2.17: Sensore ottico di prossimità

I sensori ottici di prossimità sono in grado di rivelare la presenza di oggetti all'interno di un determinato range, definito *portata nominale*. Il segnale analogico di tensione in uscita risulta proporzionale alla distanza fra il sensore e il corpo che riflette la luce emessa dal diodo led, generalmente secondo una dipendenza quadratica inversa.

L'utilizzo di questa tipologia di sensori ha come vantaggio la robustezza del dispositivo, il basso costo, una portata nominale appropriata per l'applicazione in esame, e la possibilità, se necessario, di lavorare anche in immersione, disponendo di un contenitore stagno.

Il principale limite, invece, risiede nel fatto che generalmente la misura del sensore viene influenzata dalla luce diffusa dell'ambiente, dalle proprietà di riflessione dell'oggetto e dall'angolo di inclinazione fra l'asse del sensore e la superficie dell'oggetto.

La proiezione di raggi luminosi intensi diretti, infine, può provocarne l'accecamento.

Alcuni accorgimenti costruttivi hanno consentito, tuttavia, di limitare in parte questi problemi. In primo luogo si è scelto un diodo fotoemettitore caratterizzato da un diagramma di emissività concentrato all'interno di un range il più limitato possibile, in modo da ridurre l'apertura del fascio luminoso ed evitare la dispersione di potenza radiante all'esterno dell'area in cui il sensore sta operando. Anche i fototransistor sono stati scelti con un diagramma di sensitività altrettanto limitato, così da poter considerare pressoché costante, al variare dell'inclinazione, la potenza assorbita, come è possibile osservare in Fig.2.18 e 2.19.

Per quanto riguarda, invece, l'influenza dell'inclinazione del piano rispetto al sensore, sono stati adottati degli accorgimenti software che verranno meglio esposti nel capitolo 3.

Infine, la cablatura dal sensore alla scheda di acquisizione è stata realizzata tramite un cavo schermato a otto fili, con schermatura a maglia stretta e doppia messa a terra sul robot e sulla messa a terra del circuito di potenza del robot, per cercare di limitare il più possibile il livello di rumore nei dati acquisiti.

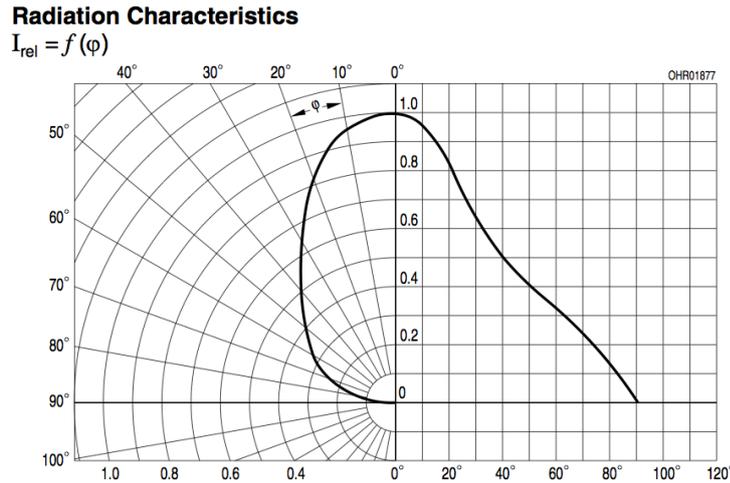


Fig. 2.18: Sensore ottico di prossimità - Diagramma di emissività del diodo LED

Fig. 8 Sensitivity Diagram ($T_a = 25^\circ\text{C}$)

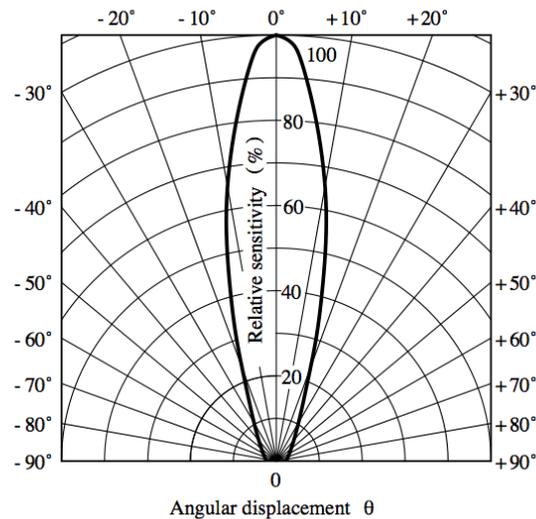


Fig. 2.19: Sensore ottico di prossimità - Diagramma di sensitività del fototransistor

2.4.1 Caratterizzazione del sensore

I sensori sperimentali come quello di Fig.2.17 necessitano di un'apposita procedura di calibrazione, al fine di poter determinare con accuratezza la portata nominale e la relazione fra la tensione in uscita dal sensore e la corrispondente distanza misurata.

Nel nostro caso, si è predisposto un banco di test dotato di un sensore laser, attraverso il quale si è misurata la distanza effettiva del target dal sensore, e un oscilloscopio, al fine di rilevare la misura di tensione in uscita dal sensore. Si sono quindi registrate un certo numero di misure di tensione per ciascun fototransistor, a differenti distanze dal target, ottenendo le curve di caratterizzazione in Fig.2.20.

In particolare si è scelto di eseguire la caratterizzazione con un target ideale, ovvero una superficie piana, liscia e bianca, in grado di riflettere tutte le componenti dello spettro luminoso. Qualsiasi altra superficie di differente colorazione o livello di ruvidezza determinerebbe una diminuzione della luce riflessa, e quindi anche una diminuzione della portata nominale. Ne deriva che, nel momento in cui il sensore dovesse essere utilizzato per rilevare la presenza o meno di un arto umano, la pigmentazione e la rugosità della

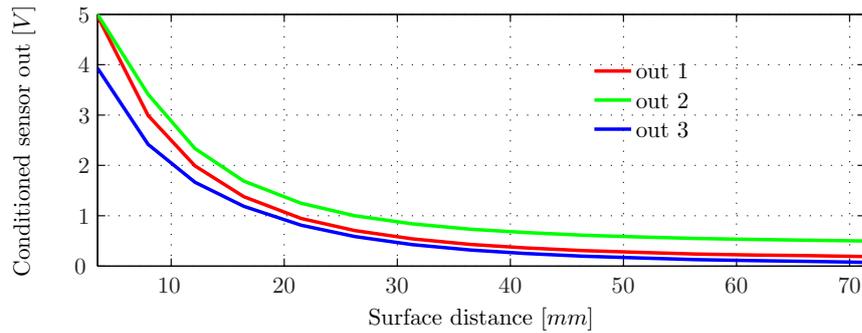


Fig. 2.20: Sensore ottico di prossimità - Curve di calibrazione del sensore (caratteristiche I/O)

pelle determinerebbero delle variazioni nel comportamento dello stesso.

Sarebbe quindi opportuno, in fasi successive di sviluppo del prodotto, effettuare anche una procedura di calibrazione su pelle umana.

Osservando la Fig.2.20, si nota che l'intensità luminosa misurata dai fototransistor è inversamente proporzionale, con relazione quadratica, alla distanza dal target. Tali curve sono state opportunamente adattate, limitando inferiormente (2 mm) i valori di distanza misurata, al fine di ottenere una relazione monotona decrescente.

La curva tipica di un sensore ottico, infatti, è riportata in Fig.2.21. Si nota come essa presenti una prima zona, definita *front slope*, monotona crescente, alla quale segue una zona più estesa, definita *back sope*. Nella pratica, così come nel nostro caso, si tende ad utilizzare unicamente la seconda zona della curva, in quanto consente di ottenere una portata nominale sufficientemente elevata e una maggiore sensibilità alle variazioni di tensione.

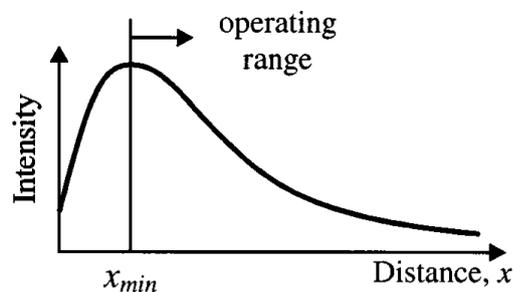


Fig. 2.21: Sensore ottico di prossimità - Curva generica di calibrazione

2.4.2 Sistemi di riferimento

La scelta di tre fototransistor è stata dettata dalla necessità di stimare posizione ed orientazione del target rispetto all'end effector. Matematicamente, infatti, è possibile determinare un piano, dati al minimo tre punti dello stesso. Di conseguenza, analizzando il fascio luminoso riflesso dalla superficie dell'oggetto e ricevuto dai fototransistor, è possibile individuare i tre punti richiesti.

In particolare, come è possibile osservare in Fig.2.22, si è assunto che il fascio luminoso riflesso dal target incida perpendicolarmente rispetto alla superficie dell'end effector, e quindi del sensore.

L'individuazione di tre punti consente di determinare due vettori sul piano e il prodotto vettoriale fra quest'ultimi determina, invece, il vettore normale al piano stesso.

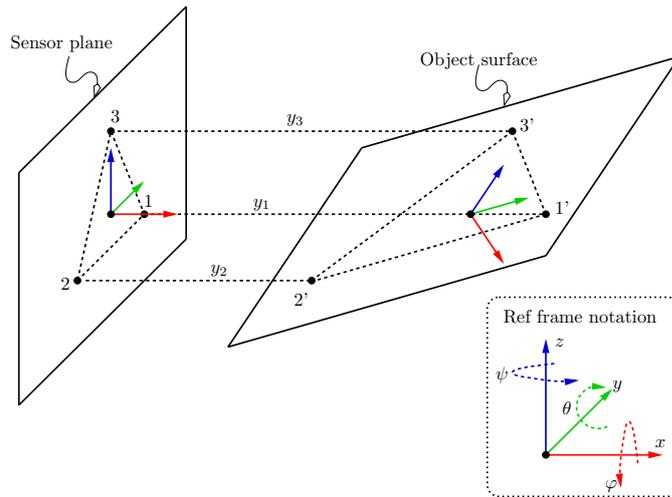


Fig. 2.22: Sensore ottico di prossimità - Sistema di riferimento collocato sul target, rispetto alla terna di riferimento del sensore

Quest'ultimo costituirà l'asse x del sistema di riferimento sul target, mentre l'asse z punterà in direzione della riflessione del terzo fototransistor, così come l'asse z nella terna solidale all'end effector era rivolto in direzione del terzo fototransistor. L'asse y sarà, infine, determinato di conseguenza.

Al fine di ridurre al minimo il carico computazionale, si sono scelte delle tecniche semplici, ma approssimate, per misurare la distanza della sonda dal piano e l'orientazione di quest'ultimo. Una volta stabilite le convenzioni secondo cui si definisce il sistema di riferimento, infatti, la distanza dal piano viene calcolata semplicemente come la media delle tre distanze restituite dai fototransistor:

$$d = \frac{y_1 + y_2 + y_3}{3} \quad (2.4.1)$$

Poiché tale stima dipende dalle curve di calibrazione, la sua accuratezza è strettamente legata alle proprietà di riflessione della superficie e dalla sua orientazione.

Quest'ultima può essere specificata in termini dei due angoli θ_s e ψ_s , rispettivamente pitch e yaw, i quali descrivono di fatto l'orientazione $\phi_s = [0, \theta_s, \psi_s]$ del sistema di riferimento collocato sull'oggetto rispetto alla terna solidale all'end effector.

Nel rispetto delle convenzioni sopra adottate, segue che :

$$\theta_s \approx \text{atan} \left[\frac{2}{3L} \left(y_3 - \frac{y_1 + y_2}{2} \right) \right] \quad (2.4.2)$$

$$\psi_s \approx \text{atan} \left(\frac{y_2 - y_1}{\sqrt{3}L} \right) \quad (2.4.3)$$

Anche se 2.4.2 e 2.4.3 rappresentano solo delle approssimazioni degli angoli pitch e yaw, essi possono essere comunque utilizzati nella procedura di allineamento dell'end effector del robot con la superficie rilevata dal sensore, integrata nel sistema di controllo e che verrà illustrata in dettaglio nei capitoli 4 e 5.

2.4.3 Scheda di acquisizione e condizionamento del segnale

L'acquisizione e il condizionamento del segnale proveniente dal sensore sono stati realizzati mediante la scheda in Fig.2.23, appositamente progettata.

Una parte del circuito, alimentata a $+5V$ direttamente dal connettore, ha lo scopo di generare un segnale ad onda quadra di ampiezza pari a $5V$ e frequenza $10kHz$, generato

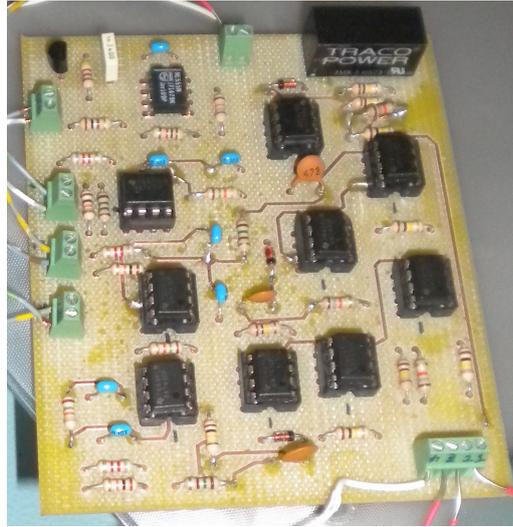


Fig. 2.23: Sensore ottico di prossimità - Scheda di acquisizione e condizionamento dei segnali del sensore

da un timer NE555, per alimentare il fotoemettitore. La frequenza dell'onda quadra è stata scelta in modo da rendere il sensore immune all'influenza della luce dell'ambiente circostante o a possibili altre fonti luminose intermittenti.

L'altra parte del circuito è dedicata, invece, all'acquisizione e al condizionamento dei segnali provenienti dai tre fototransistor, secondo il medesimo schema di principio per ciascuno di essi.

Analisi circuitale

Come è possibile osservare in Fig.2.24, la tensione V_{out_1} in uscita dal circuito di alimentazione del fototransistor viene inviata in ingresso a un filtro passa banda, centrato alla frequenza di $10kHz$. In questo modo è possibile reiettare in buona parte le componenti spurie del segnale, ricavando unicamente il segnale utile riflesso dalla superficie.

Segue quindi un rivelatore di picco, con lo scopo di estrarre l'ampiezza del segnale sinusoidale e ottenere in questo modo una tensione V_{out_3} dipendente unicamente dalle variazioni di picco del segnale. La tensione ottenuta viene infine amplificata opportunamente attraverso uno stadio di amplificazione, in modo da ottenere un'uscita pressoché costante, di un valore compreso fra $0V$ e $+5V$, e proporzionale all'intensità della luce riflessa.

In quest'ultimo stadio di condizionamento sono stati utilizzati amplificatori $ua741$, alimentati con tensione duale $\pm 15V$, fornita da un apposito alimentatore alloggiato all'interno del circuito.

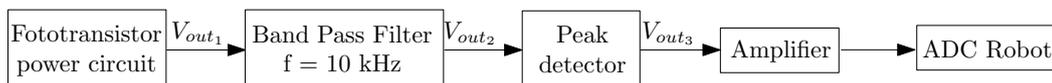


Fig. 2.24: Sensore ottico di prossimità - Schema di condizionamento del segnale

Integrazione dei dispositivi all'ambiente Matlab Simulink

In questo capitolo vengono presentati e analizzati i blocchi funzionali che hanno permesso l'interfacciamento dei dispositivi descritti nel capitolo 2 con l'ambiente Matlab Simulink. Quest'ultimo è caratterizzato da un'interfaccia semplice e modulare, che ha permesso di implementare singolarmente i diversi blocchi e di sviluppare il progetto a diversi livelli di astrazione.

In particolare, la differente architettura dei dispositivi master e slave ha reso necessarie alcune scelte preliminari, che hanno consentito di ottenere un sistema di riferimento comune e facilitato così la comunicazione fra i due. Un'altra operazione fondamentale è stata quella di scalare e adattare i due range di movimento.

3.1 Dispositivo Master: Phantom Omni

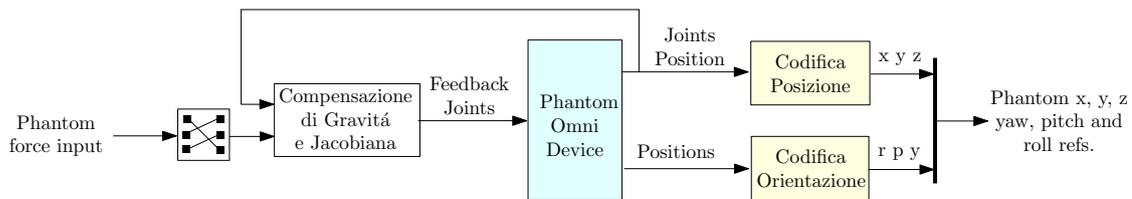


Fig. 3.1: Phantom Omni - Blocco di interfacciamento

In Fig.3.1 è presentato lo schema complessivo attraverso cui il dispositivo master Phantom Omni si interfaccia con il dispositivo slave.

In ingresso viene inviata la forza di feedback, calcolata a partire dai riferimenti del robot slave. Attraverso un selettore, essa viene adattata al sistema di riferimento del master, elaborata da un blocco di compensazione della gravità dei giunti e infine moltiplicata per la matrice Jacobiana del sistema. In questo modo è possibile ottenere in uscita le coppie di attuazione dei giunti del Phantom.

Tali coppie vengono quindi inviate in ingresso al blocco *OmniDevice*, che fornisce in uscita le posizioni angolari dei giunti, in notazione asse-angolo.

Delle funzioni di codifica di posizione e orientazione sono state scritte poi al fine di ottenere le coordinate di posizione e orientazione corrispondenti per l'end effector del robot.

3.1.1 Blocco OmniDevice

Il software Handshake proSENSE™ fornisce il blocco *OmniDevice*, il quale richiede unicamente l'impostazione del tempo di campionamento (Sample Time), nel nostro caso pari a 0.001s.

Esso riceve in ingresso un vettore di tre elementi, contenente la scomposizione del segnale di coppia per l'attuazione dei giunti lungo le tre direzioni degli assi cartesiani. Questi valori devono rimanere compresi fra -32768 e 32767 (215 bit, più il segno), che attuano rispettivamente la coppia massima negativa e positiva.

In uscita al blocco si ottengono le seguenti informazioni:

- *Joint Position*: un vettore di dimensione 3, contenente la posizione dell'end-effector espressa in termini di coordinate angolari dei giunti [*rad*];
- *Rotation*: un vettore di dimensione 4 che esprime l'orientazione dell'end-effector espressa in Axis Angles. I primi tre elementi identificano quindi le coordinate di un asse, mentre l'ultimo rappresenta l'angolo di rotazione compiuto attorno a tale asse;
- *Buttons*: un vettore di dimensione 2 che fornisce lo stato binario dei pulsanti del dispositivo. Questa uscita è stata utilizzata per fornire al sistema ulteriori finalità, che non vengono tuttavia approfondite nel presente elaborato. Per questo motivo, essa non viene riportata in Fig.3.1.

3.1.2 Codifica dei riferimenti

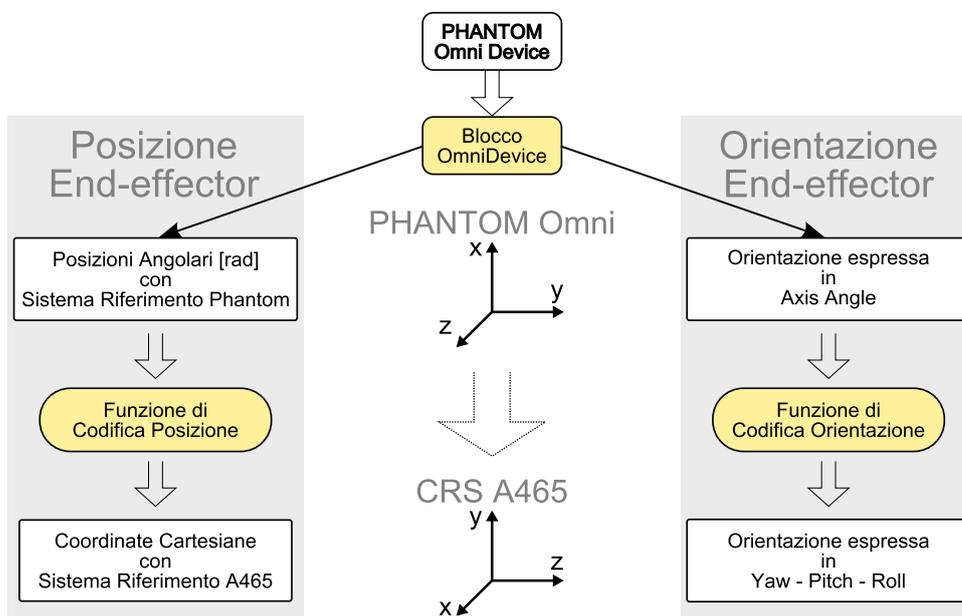


Fig. 3.2: Phantom Omni - Codifiche di posizione ed orientazione

Come anticipato, a causa delle differenti architetture dei due dispositivi è stato necessario determinare una relazione fra i due sistemi di riferimento ed effettuare una codifica delle informazioni di posizione e orientazione del Phantom, al fine di ottenere i riferimenti da fornire in ingresso al robot.

Come è possibile osservare in Fig.3.2, stabilendo come riferimento la terna base del robot CRS, al fine di uniformare i riferimenti di posizione, si rende necessario stabilire la seguente relazione fra i due sistemi di riferimento:

$$\begin{aligned} X_{crs} &= Z_{phantom} \\ Y_{crs} &= X_{phantom} \\ Z_{crs} &= Y_{phantom} \end{aligned}$$

L'informazione relativa alle posizioni angolari del master, in uscita dal blocco Omni-Device, viene elaborata inoltre da una codifica di posizione in grado di fornire in uscita le coordinate cartesiane, adattate e scalate in funzione del dispositivo slave. Analogamente, l'informazione sull'assetto espressa in axis-angles viene convertita, mediante una funzione di codifica di orientazione, in notazione yaw-pitch-roll.

Entrambe le codifiche vengono analizzate in dettaglio nelle seguenti sezioni, e riportate per esteso in appendice C.1.

Codifica di orientazione



Fig. 3.3: Phantom Omni - Codifica di orientazione

La funzione riceve in ingresso un vettore di quattro elementi, costituenti il riferimento di posizione e orientazione del Phantom in notazione asse-angolo. Di conseguenza il vettore sarà composto dai primi tre elementi, costituenti le tre componenti del versore relativo a un asse di rotazione, mentre il quarto elemento rappresenterà la misura dell'angolo di rotazione.

E' stata definita quindi la matrice di rotazione corrispondente R , calcolata in riferimento alla notazione asse-angolo:

```

1 % defines the rotation matrix w.r.t. the rgb].133,.545,.133...
  axis-angles notation ctheta = cos(theta); stheta = rgb].133,.545,.133...
  sin(theta); vtheta = 1-cos(theta);
2 R = [k_x^2 vtheta+ctheta, k_xk_yvtheta-k_zstheta, k_xk_zvtheta-k_ystheta;
3      k_xk_yvtheta+k_zstheta, k_y^2vtheta-ctheta, k_yk_zvtheta-k_xstheta;
4      k_xk_zvtheta+k_ystheta, k_yk_zvtheta+k_xstheta, k_z^2vtheta+ctheta;]
  
```

A partire dalle equazioni in appendice B.2, si è ricavato l'angolo di *pitch* e a seconda del valore di quest'ultimo, si sono poi ricavati anche yaw e roll, distinguendo fra le due casistiche possibili.

```

1 if abs( abs(pitch) - pi/2 ) < 0.001,
2     yaw = 0;
3     if pitch > 0,
4         roll = atan2(Rt(1,2), Rt(2,2));
5     else
6         roll = -atan2(Rt(1,2), Rt(2,2));
7     end
8 else
9     yaw = atan2(Rt(2,1), Rt(1,1));
10    roll = atan2(Rt(3,2), Rt(3,3));
11 end;
  
```

Codifica di posizione

Al fine di adattare i riferimenti forniti dal dispositivo master allo spazio cartesiano del CRS slave, si sono definite innanzitutto le misure dei link del robot. Gli angoli sono stati poi adattati, in modo da assumere valori compresi unicamente fra $-\pi$ e π . E' stato quindi possibile determinare le coordinate del robot come segue:

```

1 % compute cartesian coords of end-effector
2 % (i.e. T30(1:3,4), with T30 the transformation rgb].133,.545,.133...
  from frame 3 to frame 0 )
3 k7 = -link2_CRS*s23 + link1_CRS*c2;
4 x = c1*k7;
5 y = s1*k7;
6 z = (link2_CRS*c23 + link1_CRS*s2) + base_CRS;
  
```

In particolare si è fatto riferimento alla matrice di rotazione dalla terna 3 alla terna base dell'end effector, definita come $T_3^0 = T_1^0 T_2^1 T_3^2$ e calcolata come in appendice A. Dall'espressione matriciale 3.1.1 si sono infine ricavate le espressioni delle coordinate (x_0, y_0, z_0) dell'end effector del robot rispetto alla terna base.

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = T_3^0 \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} \quad (3.1.1)$$

3.1.3 Coppie di attuazione e compensazione della gravità

Il blocco di Fig.3.1 svolge due compiti fondamentali.

In primo luogo viene definita la matrice Jacobiana 3.1.2 che, trasposta e moltiplicata per il vettore di forza, consente di ottenere le coppie 3.1.3 di attuazione dei tre giunti del Phantom, da inviare ad esso attraverso il blocco OmniDevice.

$$J = \begin{bmatrix} -c\theta_1(L_2s\theta_3 + L_1c\theta_2) & L_1s\theta_1s\theta_2 & -L_2s\theta_1c\theta_3 \\ 0 & L_1c\theta_2 & L_2s\theta_3 \\ -L_2s\theta_1s\theta_3 - L_1s\theta_1c\theta_2 & -L_1c\theta_1s\theta_2 & L_2c\theta_1c\theta_3 \end{bmatrix} \quad (3.1.2)$$

$$\tau = J^T \mathbf{F} = [\tau_1, \tau_2, \tau_3] \quad (3.1.3)$$

In secondo luogo tale blocco si occupa anche di effettuare la compensazione della gravità dei giunti del dispositivo master, in modo da non influenzare con essa il feedback di forza percepito dall'operatore.

Sono state introdotte delle look up tables (LUT), in grado di fornire la coppia da attuare a partire dalla posizione angolare di ciascun giunto. A questo valore viene sommato un ulteriore contributo, calcolato come in 3.1.3, al fine di compensare le coppie determinate dal peso dei giunti. Risulta evidente come questi contributi di compensazione non siano costanti, bensì direttamente proporzionali all'angolazione di ogni link.

Non avendo tuttora a disposizione una procedura in grado di stimare in modo esatto le coppie in funzione del valore angolare dei link, le LUT descritte sopra sono state calcolate attraverso una procedura molto dispendiosa in termini di tempo, ma in grado di restituire risultati molto precisi.

Tramite un controllore si sono memorizzate le coppie di feedback che sorreggono i giunti in una determinata posizione. Una volta memorizzati tali valori per ogni posizione nel range di movimento dei primi due link, con un intervallo di discretizzazione di qualche millimetro, si sono determinate le mappe tridimensionali dei valori di coppia necessari a sostenere i link in ogni punto di lavoro.

3.2 Dispositivo Slave: Thermo CRS A465

Nella gestione del dispositivo slave va posta particolare attenzione nel passaggio del controllo da close architecture (CA) ad open architecture (OA). In questa situazione il controllore C500C rilascia tutti i controlli interni, mantenendo unicamente la funzione di amplificatore di tensione dei motori dei giunti e di collegamento tra gli encoder e la scheda di acquisizione MultiQ3.

E' stato quindi predisposto un interruttore a monte del blocco di controllo e interfacciamento del robot, con il compito di gestire la transizione tra i due diversi modi di operare. Nel momento in cui si passa in OA un blocco di memoria fornisce al robot come riferimenti la sua posizione e orientazione attuale, fino a quando un ulteriore interruttore non consente la generazione dei riferimenti ai blocchi di controllo automatico di orientazione e distanza dal target, che verranno approfonditi nei capitoli 4 e 5.

In Fig.3.4 viene presentato lo schema generale, che costituisce di fatto l'interfaccia del dispositivo slave con il sistema.

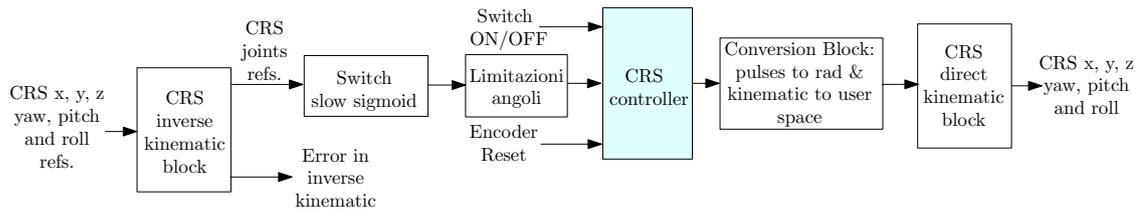


Fig. 3.4: Thermo CRS A465 - Blocco di interfacciamento

I riferimenti di posizione ed orientazione dell'end effector del robot vengono elaborati attraverso le equazioni di cinematica inversa, ottenendo in questo modo i riferimenti angolari dei sei giunti.

Il blocco slow sigmoid si occupa invece di rallentare i movimenti del robot, a seconda di un comando esterno che l'utente può modificare in tempo reale. Segue quindi la limitazione dei riferimenti angolari, fondamentale per non incorrere in fenomeni di saturazione.

I riferimenti così elaborati vengono poi introdotti in un blocco denominato CRS controller, con il compito di elaborare i segnali attraverso dei regolatori PD e fornire i segnali in tensione per comandare i giunti del robot.

Questi segnali vengono infine convertiti da impulsi a radianti, adattati allo spazio utente e convertiti nuovamente in termini di posizione e orientazione dell'end effector attraverso le equazioni di cinematica diretta.

Nelle successive sezioni viene preso in considerazione con maggiore dettaglio ciascuno dei blocchi descritti.

3.2.1 Cinematica Inversa

Il blocco in Fig.3.5 si occupa di convertire i riferimenti di posizione e orientazione dell'end effector in termini di posizioni angolari dei sei giunti del robot.

Oltre ai riferimenti, la *s-function* JointToWorld.c riceve in ingresso il *kinematic mode*, ovvero un vettore di tre elementi, tali da specificare la posizione che si vuole che il robot assuma nella soluzione del problema. Le differenti possibilità sono riportate nella sezione 2.2.3 in Tab.2.6, mentre in Fig.2.14 si possono osservare le corrispondenti posizioni assunte dal robot.

La funzione fornisce quindi due uscite: i riferimenti espressi in termini di posizioni angolari dei sei giunti del robot, e una variabile *error*, il cui valore segnala la presenza o meno di un errore nella soluzione del problema di cinematica.

In particolare, gli errori che la funzione può fornire in uscita sono:

- $0xn1$ = *giunto fuori portata*. Il giunto n ha superato il proprio limite, positivo o negativo, del range di azione;
- $0xn2$ = *trasformazione illecita*. Nessuna soluzione di cinematica inversa soddisfa le richieste;
- $0xn3$ = *errore nella cinematica inversa, posizione fuori portata*. La posizione richiesta è fuori dallo spazio di lavoro del robot e non può essere raggiunta.

3.2.2 Switch Slow sigmoid

Tramite un interruttore *Multiport Switch* a due ingressi è possibile stabilire se inviare al robot i riferimenti forniti dal blocco di cinematica inversa, o i medesimi riferimenti con accelerazioni e velocità limitati.

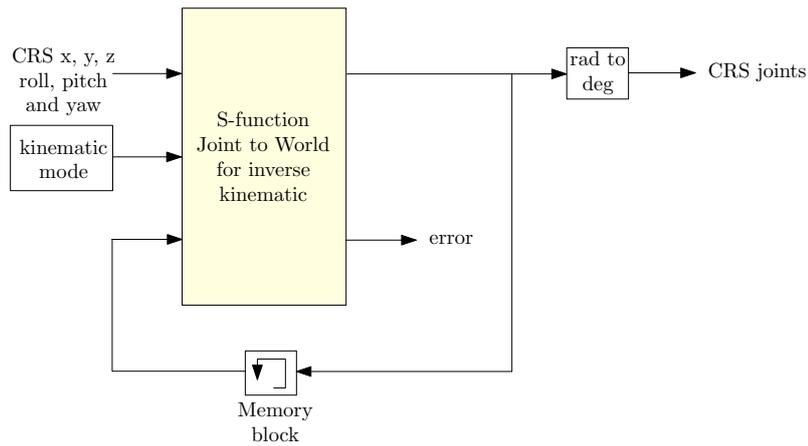


Fig. 3.5: Thermo CRS A465 - Blocco di cinematica inversa

La modalità slow sigmoid è attivabile esternamente dall'operatore tramite uno switch manuale ed è realizzabile attraverso la funzione `QSigmoid.c`. Essa genera di fatto un riferimento di posizione che cerca di inseguire nel tempo il riferimento in ingresso, eliminando tuttavia le variazioni troppo rapide. Di conseguenza, questa modalità di funzionamento si rivela vantaggiosa unicamente nelle fasi di avviamento del robot ed eventualmente anche nelle fasi transitorie di cambiamento dei riferimenti.

3.2.3 Limitazione degli angoli

Questo blocco contiene sei saturazioni, una per ciascun giunto. Il loro valore viene posto di qualche grado inferiore al limite imposto dai finecorsa meccanici riportati in Tab.2.4 introdotti nella sezione 2.2.1, in modo da evitare di incorrere in singolarità del meccanismo.

In Tab.3.1 vengono riportati i valori imposti ai blocchi di saturazione, confrontati con quelli di Tab.2.4.

Tab. 3.1: Thermo CRS A465 - Saturazioni dei giunti

| Giunto | Limite di saturazione [deg] | Finecorsa meccanico [deg] |
|----------------|-----------------------------|---------------------------|
| J_1 (vita) | $-170 < J_1 < 170$ | $-175 < J_1 < 175$ |
| J_2 (spalla) | $-85 < J_2 < 85$ | $-90 < J_2 < 90$ |
| J_3 (gomito) | $-105 < J_3 < 105$ | $-110 < J_3 < 110$ |
| J_4 (yaw) | $-175 < J_4 < 175$ | $-180 < J_4 < 180$ |
| J_5 (pitch) | $-105 < J_5 < 105$ | $-110 < J_5 < 110$ |
| J_6 (roll) | $-175 < J_6 < 175$ | $-180 < J_6 < 180$ |

3.2.4 Controllore

Come è possibile osservare in Fig.3.6 il blocco di controllo del robot CRS riceve in ingresso i riferimenti forniti dalla cinematica inversa in termini di posizioni angolari dei giunti e una variabile, denominata *encoder reset*, con il compito di azzerare la lettura degli encoder incrementali al momento del passaggio da CA ad OA, per le ragioni già esposte in 2.2.3.

I dati provenienti dagli encoder vengono acquisiti dalla funzione `sfun_multiq3_enc.c` e convertiti in termini di coordinate angolari.

Al blocco *PD position controllers* viene quindi lasciato il compito di elaborare l'errore fra i riferimenti e le posizioni angolari effettive attraverso dei regolatori PD di posizione, in modo da ottenere un segnale di tensione. Quest'ultimo servirà per comandare i motori, dopo essere stato limitato da un blocco di saturazione $\pm 5V$. Tale limitazione è necessaria in quanto i valori di uscita dei regolatori PD, dipendendo dall'errore di posizione, possono assumere valori anche di molto al di sopra del limite massimo di alimentazione.

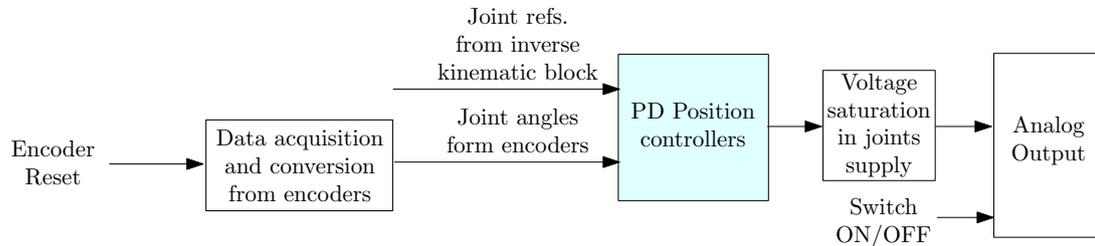


Fig. 3.6: Thermo CRS A465 - Blocco del controllore del robot

Controllori PD di posizione

Prima di venire utilizzati dal regolatore PD, i valori di riferimento espressi in coordinate angolari vengono elaborati dalla funzione *QSigmoid.c*. Come visto nella precedente sezione, essa consente di limitare velocità e accelerazione massime, adattando poi i riferimenti con i valori limitati e creando dei profili di moto normalizzati a 'S'.

In questo modo si evita che eventuali riferimenti a gradino in ingresso al regolatore PD possano generare gradini di tensione, e quindi un movimento poco fluido del robot.

I riferimenti e le effettive posizioni angolari del robot vengono confrontate ed elaborate da sei regolatori PD, come è possibile osservare nel dettaglio di 3.7. Essi sono costituiti fondamentalmente da un filtro passa basso e dai due guadagni, rispettivamente proporzionale ed integrale, K_P e K_D , prestabiliti dal costruttore.

L'uscita di ciascun regolatore viene poi opportunamente moltiplicata per una costante unitaria, con segno positivo o negativo a seconda del verso di rotazione dell'encoder. Come visto in 2.2.3, è infatti indispensabile considerare l'effettivo verso di rotazione dell'encoder, in modo da interpretare correttamente le informazioni da esso provenienti e produrre di conseguenza il giusto riferimento di tensione per l'alimentazione dei sei motori dei giunti.

Analog Output

Al fine di trasmettere un segnale analogico di tensione al controllore C500C è stato utilizzato il convertitore DA della scheda di acquisizione MultiQ3.

Poiché la scheda realizza una conversione DA a 12 bit senza segno, si è realizzato un blocco in grado di codificare un valore in tensione compreso fra $\pm 5V$ in un valore compreso fra 0 e 4095 inclusi.

Il segnale viene poi adattato sommando un offset pari a 5 e scalandolo della quantità calcolata in 3.2.1.

$$k = \frac{x'_{max} - x'_{min}}{x''_{max} - x''_{min}} = \frac{4095 - 0}{5 - (-5)} = 409,5 \quad (3.2.1)$$

Il blocco *Analog Output* presenta 8 ingressi, tanti quanti i canali gestibili dal convertitore DA. I segnali vengono elaborati dai circuiti di *Zero-Span*, i quali li incrementano, amplificano e convertono da double a word, formato compatibile per la trasmissione dei dati alla scheda.

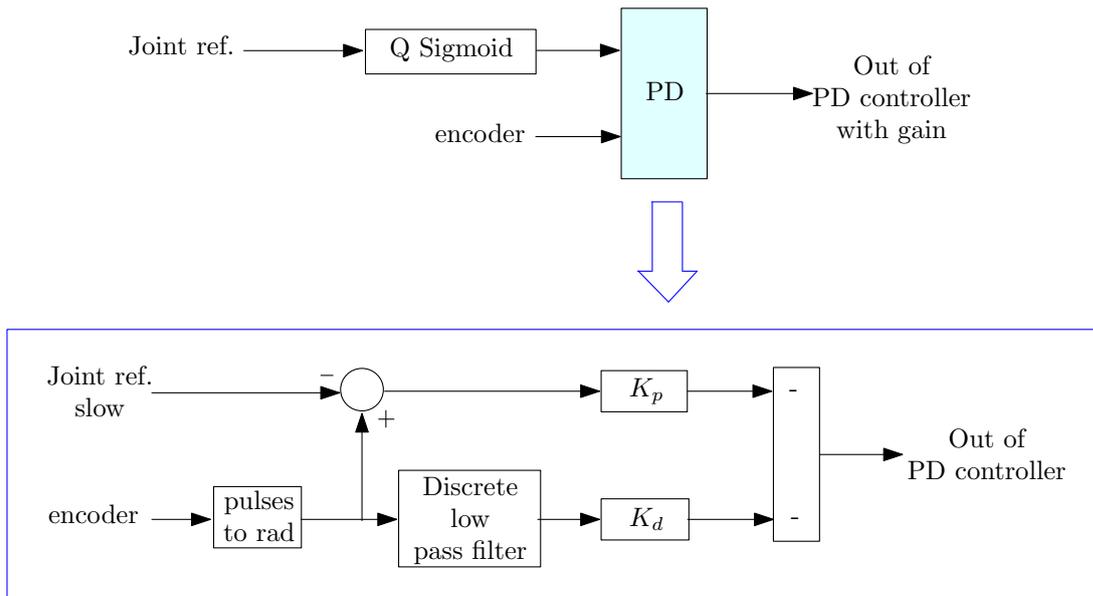


Fig. 3.7: Thermo CRS A465 - Blocco regolatore PD

La funzione Sfun_DAC.c, infine, definisce gli indirizzi di memoria ai quali inviare i dati, la frequenza del clock del convertitore DA e gestisce inoltre l'inizializzazione del canale per la trasmissione dei dati e la trasmissione stessa.

3.2.5 Cinematica diretta

Una volta elaborati i riferimenti angolari da fornire al robot, si desidera ricavare l'informazione relativa ad essi anche in termini di posizione ed orientazione dell'end effector. Questa informazione verrà utilizzata nei software di controllo di orientazione e della distanza dal target che verranno approfonditi nei prossimi capitoli.

A tale scopo si utilizza la s-function WorldToJoint.c, contenente le equazioni di cinematica diretta, in grado di restituire la posizione del robot in termini di coordinate cartesiane dell'end effector e angoli di yaw pitch e roll, a partire dalle coordinate angolari dei sei giunti.

3.3 Sensore ottico di prossimità

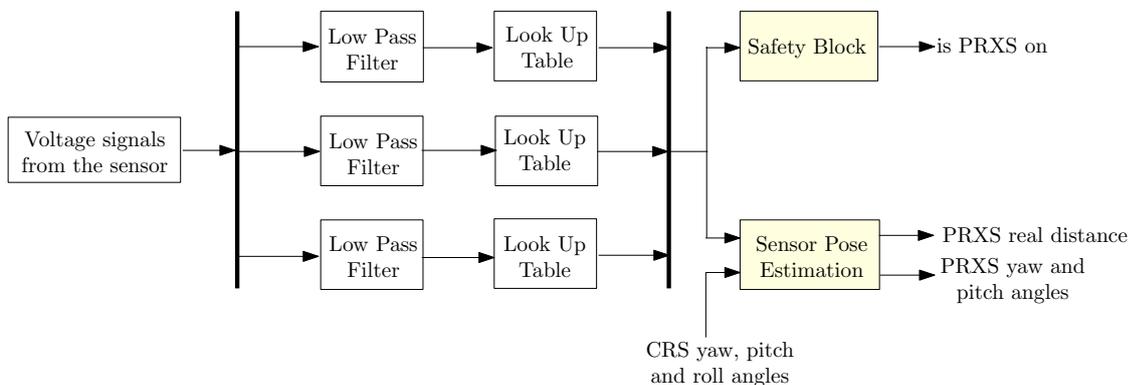


Fig. 3.8: Sensore ottico di prossimità - Blocco di elaborazione del segnale

In Fig.3.8 viene rappresentato il blocco di condizionamento ed elaborazione del segnale, che restituisce in uscita la distanza media dell'end effector dal piano, l'orientazione del piano rispetto all'end effector e l'informazione relativa all'attivazione o meno del sensore di prossimità.

I segnali di tensione provenienti dal sensore vengono acquisiti ed elaborati dalla funzione `sfun_multiq3_adc.c`, la quale restituisce le tre misure in tensione.

Ciascuna di queste misure viene filtrata attraverso un filtro di Butterworth (passa basso) di ordine 3 e frequenza $30Hz$.

La risposta in frequenza di questo tipo di filtri è quella più piatta ottenibile nella banda passante, ovvero quella maggiormente priva di ondulazioni in banda e può essere matematicamente definibile come:

$$|G(j\omega)| = \frac{1}{\sqrt{1 + \omega^{2n}}} \quad (3.3.1)$$

dove n rappresenta l'ordine del filtro, mentre ω il rapporto fra la frequenza del segnale e la frequenza di taglio che si vuole imporre con il filtro.

Nel nostro caso il filtro, essendo di ordine 3, impone un'attenuazione fuori banda di $-60dB/decade$. Esso presenta inoltre una frequenza di taglio pari a $30Hz$, determinata in modo empirico come compromesso tra la bontà dei risultati ottenibili a valle dell'azione di filtraggio e il ritardo introdotto dal filtro nella generazione dei segnali.

La necessità di filtrare opportunamente le misure di tensione acquisite deriva dal fatto che esse devono essere utilizzate come ingressi alle LUT descritte nella sezione 2.4, che forniscono una misura in [mm] della distanza del sensore dalla superficie rilevata in funzione del valore di tensione acquisito.

Si è notato come in assenza di un'azione di filtraggio le misure di tensione oscillassero in modo eccessivo, determinando un'eccessiva fluttuazione delle misure di distanza dal target, che si rivelano di fondamentale importanza nell'implementazione delle strategie di controllo che verranno approfondite nei capitoli 4 e 5.

Le misure di distanza così ottenute vengono poi utilizzate da un lato per stimare la distanza media dell'end effector dal piano, dall'altro per determinare se lo stato del sensore sia attivo oppure no, ovvero se si possano considerare affidabili le misure del sensore. Entrambi questi aspetti verranno approfonditi nelle due successive sezioni.

3.3.1 Get sensor posture

Le righe di codice riportate di seguito consentono di calcolare i valori degli angoli di *yaw* e *pitch* del piano del target rispetto alla terna di riferimento posizionata sul sensore, o equivalentemente sull'end effector.

Le formule del codice rispecchiano le equazioni 3.3.2 e 3.3.2, già esposte nella sezione 2.4 e qui riportate per chiarezza:

```

1 %get PRXS yaw and pitch
2 PRXS_YP = zeros(2,1);
3 PRXS_YP(1) = atan( (u(2)-u(1))/sqrt(3)/5 );
4 PRXS_YP(2) = atan( (u(3)-0.5*u(1)-0.5*u(2))/(3/2*5) );

```

$$\theta_s \approx \text{atan} \left[\frac{2}{3L} \left(y_3 - \frac{y_1 + y_2}{2} \right) \right] \quad (3.3.2)$$

$$\psi_s \approx \text{atan} \left(\frac{y_2 - y_1}{\sqrt{3}L} \right) \quad (3.3.3)$$

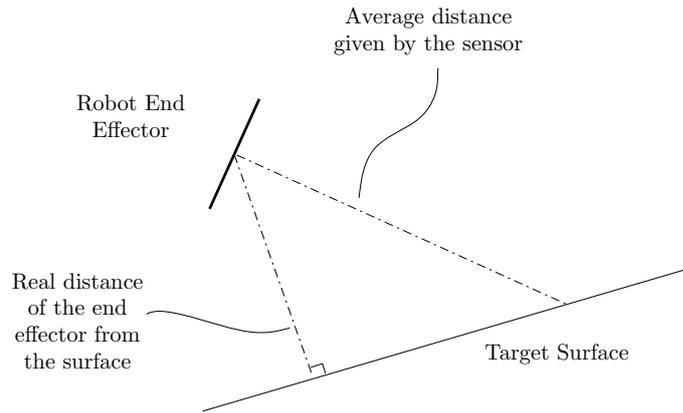


Fig. 3.9: Sensore ottico di prossimità - Calcolo della distanza dell'end effector del robot dal target

Per quanto riguarda, invece, la stima della distanza media, a partire dalle tre distanze fornite dai fototransistor, essa viene calcolata semplicemente come media delle tre misure.

Come è possibile osservare in Fig.3.9, tuttavia, tale distanza non costituisce sempre la distanza effettiva dal target, in quanto quest'ultima dipende dall'inclinazione dell'end effector stesso.

A tale scopo si è scelto di modificare il calcolo della distanza come segue:

```

1 % get avg dist
2 avg_dist = mean(u);
3 % get unit vector orthogonal to the plane
4 np = get_np(CRS_YPR, PRXS_YP);
5 % get EE approaching direction (unit vector)
6 nEE = get_nEE(CRS_YPR);
7 % get true distance of robot EE from the plane
8 PRXS_dist = avg_dist * abs(nEE'*np);

```

La distanza effettiva dell'end effector dal target si calcola quindi come il prodotto della distanza media per il valore assoluto del prodotto scalare tra il versore nEE ortogonale all'end effector e np , ortogonale al piano.

3.3.2 Safety block e abilitazione del sensore

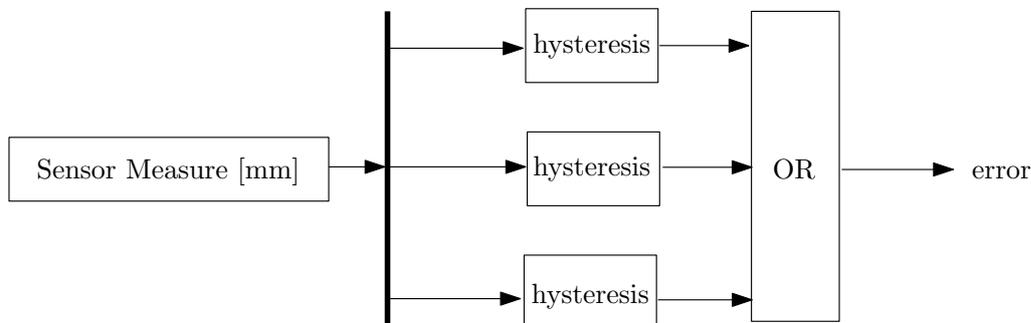


Fig. 3.10: Sensore ottico di prossimità - Blocco di abilitazione

Le misure ottenute dai tre fototransistor e opportunamente trasformate in misure di distanza attraverso le LUT calcolate sperimentalmente vengono inoltre confrontate ciascuna con un blocco di isteresi, avente:

- *Limite superiore:* $LUT_{max} - 5$;
- *Limite inferiore:* $LUT_{max} - 15$.

Le uscite dei tre blocchi di isteresi sono poste pari a 1, qualora risulti che le misure in ingresso superano il valore massimo, pari a 0 se le medesime uscite si rivelano inferiori al limite minimo. Esse vengono poi confrontate attraverso un blocco logico OR, il quale restituisce in uscita un errore, il cui valore sarà pari ad 1 se almeno un sensore sta leggendo un valore superiore al limite massimo di distanza misurabile.

In questo modo il sensore verrà attivato solo nel momento in cui l'errore assume valore 0, ovvero unicamente nel momento in cui tutti e tre i sensori leggono una misura affidabile, ragionevolmente compresa nella loro portata nominale.

Si evita così di ricadere in situazioni in cui una o più misure di distanza sono errate e determinano una stima altrettanto errata della distanza media dal target e, a un livello superiore, errori nella generazione dei riferimenti da parte dei due controlli automatici di orientazione e distanza.

Architettura del sistema di controllo

Nel presente capitolo vengono presentate l'architettura e le caratteristiche del sistema di controllo implementato, il quale si prefigge di soddisfare le seguenti specifiche:

- Si desidera che, nel momento in cui nessun oggetto rientra nel campo di visione del sensore, il dispositivo slave replichi esattamente i movimenti imposti dall'operatore attraverso il dispositivo master. In questa situazione l'operatore non deve percepire alcun vincolo nei movimenti del dispositivo master, al quale, di conseguenza, non deve venire inviata alcuna forza di feedback;
- Nel momento in cui, invece, un ostacolo rientra nel campo di misura del sensore di prossimità, si vuole che a lato slave venga replicata unicamente la posizione del master. L'orientazione dell'end effector del robot viene infatti aggiustata automaticamente da un sistema di controllo, gestito interamente a lato slave, con l'obiettivo di allineare l'end effector lungo la direzione normale alla superficie dell'oggetto. Questo risultato è ottenibile utilizzando l'informazione relativa all'inclinazione della sonda, fornita a sua volta dal sensore di prossimità.

In questa situazione viene inviato al dispositivo master un feedback di forza che simula un vincolo virtuale. L'intensità di questa forza dipende dalla distanza dell'end effector dal piano dell'ostacolo individuato, fornita dal sensore di prossimità, e dal valore dell'impedenza del vincolo virtuale che si desidera ricreare a lato master;

- Al fine di evitare il contatto con il paziente e garantire una migliore risoluzione delle immagini raccolte, si richiede la possibilità di fissare una distanza minima, oltre la quale il dispositivo slave non potrà progredire, qualunque comando giunga dal lato master.

Si viene quindi a realizzare un controllo automatico della distanza, anche in questo caso gestito interamente a lato slave, che subentra ai riferimenti di posizione imposti dal master, qualora venga superata una soglia minima di distanza dal target, definita dall'utente. Nel momento in cui l'operatore continuasse anche in questa condizione di lavoro a imporre dei movimenti al di sotto della soglia prefissata, il sistema ritornerà un feedback di forza, sommato a una vibrazione.

Le specifiche sopra esposte possono essere soddisfatte utilizzando una convenzionale architettura di controllo master/slave bilaterale a due canali. In particolare si prende in considerazione la classica architettura per sistemi telerobotici di tipo posizione/forza, nota anche come architettura di controllo con feedback di forza, apportando alcune modifiche, tali da consentire al sistema di soddisfare anche gli altri requisiti sopra enunciati.

4.1 Architettura di controllo master/slave a due canali

Il sistema di controllo implementato si basa sull'architettura bilaterale *posizione-forza* illustrata in termini generali in Fig.4.1 e, esplicitando anche tutti i blocchi della dinamica, in Fig.4.2. In essa è previsto che le posizioni del master vengano fornite come riferimenti al dispositivo slave, che le riprodurrà fedelmente e ritornerà al dispositivo master una forza di feedback.

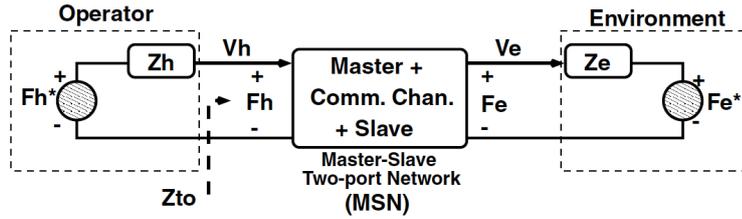


Fig. 4.1: Architettura generica di un sistema di controllo a due canali

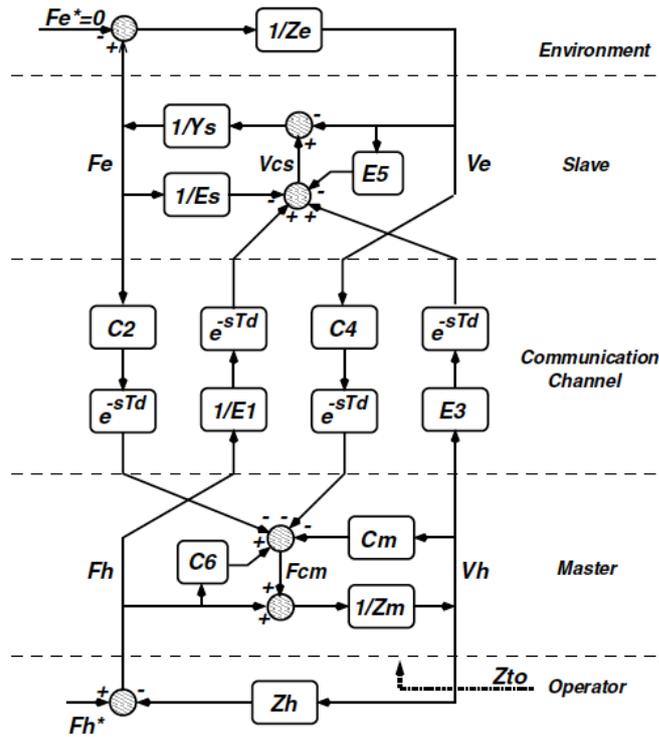


Fig. 4.2: Architettura generica di un sistema di controllo a due canali - Blocchi dinamici

Si può osservare come i modelli dei dispositivi master e slave e del canale di comunicazione, siano di fatto inseriti in un sistema lineare tempo invariante (LTI) a due canali.

Osservando Fig.4.1 in particolare, nel dominio di Laplace è possibile determinare:

$$F_h = F_h^* - Z_h V_h \quad (4.1.1)$$

$$F_e = F_e^* - Z_e V_e \quad (4.1.2)$$

La forza F_h esercitata dall'operatore sul dispositivo master è determinata, quindi, da una componente esogena F_h^* e una componente $Z_h V_h$ dovuta all'impedenza del master, dove Z_h e V_h rappresentano rispettivamente l'impedenza e la velocità del dispositivo.

Analogamente, la forza F_e a lato slave si può considerare composta da un termine esogeno F_e^* , che costituisce il contributo di forza che l'ambiente esercita sul dispositivo slave, e una componente $Z_e V_e$ dovuta all'impedenza del dispositivo slave, con Z_e e V_e rappresentanti rispettivamente impedenza e velocità del dispositivo.

Queste considerazioni risultano valide poiché l'operatore e l'ambiente esterno vengono considerati strettamente in contatto rispettivamente con i dispositivi master e slave. Di conseguenza è possibile includerli, come visto in 4.1.1 e 4.1.2, nel modello dei dispositivi, calcolato in corrispondenza di un punto di lavoro in cui operatore e ambiente esterno si trovano effettivamente in contatto con i due dispositivi master e slave.

Data la linearità del sistema, inoltre, secondo [11] è possibile ricorrere alla caratterizzazione di quest'ultimo nel dominio delle frequenze facendo uso delle trasformate di Laplace. Si ottiene così il seguente modello a matrici ibride per un sistema telerobotico a due porte:

$$\begin{bmatrix} F_h(s) \\ V_h(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} V_e \\ -F_e \end{bmatrix} \quad (4.1.3)$$

Risolvendo la 4.1.3 secondo F_h , facendo uso di 4.1.1 e 4.1.2, si ottiene la seguente espressione, da cui è possibile ricavare il valore dell'impedenza Z_t :

$$F_h = \underbrace{(H_{11} - H_{12}Z_e)(H_{21} - H_{22}Z_e)^{-1}}_{=Z_t} V_h \quad (4.1.4)$$

Anche se l'espressione 4.1.4 può notevolmente complicarsi, qualora H_{ij} presentino dinamiche significative, da essa è possibile ricavare le seguenti considerazioni di base sul sistema:

- La condizione di perfetta trasparenza $Z_t \equiv Z_e$ richiede che $H_{22} = 0$, $H_{21}Z_e = Z_e(-H_{12})$ e $H_{11} = 0$;
- Per $Z_e \rightarrow 0$, se $H_{11} \neq 0$, l'impedenza Z_t è indipendente dal valore dell'impedenza dello slave Z_e , in quanto basata sul prodotto matriciale $H_{11}H_{21}^{-1}$;
- Per $Z_e \rightarrow \infty$, l'impedenza Z_t assume il valore $H_{11}H_{22}^{-1}$, indipendente da Z_e qualora risulti $H_{22} \neq 0$.

E' importante sottolineare come i parametri H_{ij} possano venire influenzati tanto dalle dinamiche meccaniche dei dispositivi master e slave, quanto dall'architettura di controllo. Di conseguenza, non è possibile scegliere arbitrariamente i parametri ibridi, in modo da soddisfare i vincoli sopra esposti.

Secondo [10], la trasparenza nella trasmissione dei segnali, siano essi di posizione, velocità o forza, costituisce un requisito fondamentale per tali sistemi.

Idealmente, un'applicazione telerobotica non dovrebbe introdurre alcuna dinamica fra l'utente e il mondo esterno, in modo tale che l'operatore possa avere una percezione reale dell'ambiente. Si dovrebbe quindi raggiungere la condizione $Z_e = Z_h$. I due dispositivi master e slave presentano tuttavia una dinamica non trascurabile, che di fatto non sempre consente di ottenere la condizione di idealità auspicata, e che determina allo stesso tempo la necessità di condurre un'analisi accurata della stabilità.

A tale scopo, un modello LTI come 4.1.3 consente di effettuare lo studio delle performance del sistema e condurre un'analisi della stabilità, indipendentemente dalla linearità o meno dei contributi esterni dell'operatore e dell'ambiente con cui il sistema si sta interfacciando.

In [9] è stato dimostrato come, in via teorica, sia possibile raggiungere la trasparenza anche in un'architettura di questo tipo. In ogni caso, i ritardi di trasmissione, assieme alle incertezze introdotte dall'operatore e dall'ambiente esterno nella dinamica, influenzano in modo non trascurabile il comportamento del sistema e possono a volte comprometterne stabilità e performance.

Ne deriva che, in fase di progettazione e determinazione dei parametri di controllo, sarà necessario raggiungere un opportuno compromesso fra il livello di stabilità e le performance desiderate.

4.2 Controllo automatico dell'orientazione dell'end effector

La principale specifica del sistema di controllo implementato consiste nella capacità di orientare in modo automatico l'end effector del robot, nel momento in cui il sensore di prossimità individua un oggetto nel proprio campo di misura.

In queste condizioni di lavoro viene replicata solo l'informazione di posizione del dispositivo master, mentre il sistema di controllo consente l'adattamento dell'orientazione, grazie all'informazione relativa all'inclinazione della sonda, fornita dal sensore di prossimità.

Si viene di fatto a realizzare un sistema di controllo gestito a lato slave e finalizzato all'inseguimento della direzione normale al piano individuato dal sensore.

L'orientazione dell'end effector $\phi_e = [\varphi_e, \theta_e, \psi_e]$ viene specificata secondo gli angoli *yaw-pitch-roll*, dove φ_e , θ_e e ψ_e costituiscono rispettivamente gli angoli roll, pitch e yaw della terna dell'end effector rispetto al sistema di riferimento fisso alla base del robot.

Una volta che un oggetto viene individuato dal sensore di prossimità, è possibile determinare la sua orientazione ϕ rispetto alla terna base mediante gli angoli yaw, pitch e roll associati alla matrice di trasformazione $R_r^b(\phi)$, che descrive la rotazione della superficie rispetto alla terna base. Questa rotazione viene ottenuta a sua volta dalla composizione di due rotazioni: una prima rotazione della terna relativa all'end effector rispetto alla terna base, seguita dalla rotazione del sistema di riferimento collocato sulla superficie dell'oggetto rispetto a quello sull'end effector.

Si denotano rispettivamente con R_e^b e R_r^e le matrici di trasformazione relative alle due rotazioni sopra menzionate.

Da queste considerazioni segue che:

$$R_r^b(\phi^*) = R_e^b(\phi_e)R_r^e(\Delta\phi) \quad (4.2.1)$$

L'orientazione $\phi^* = [\varphi^*, \theta^*, \psi^*]^T$ costituisce il riferimento che deve essere effettivamente inseguito dal controllo automatico di orientazione dell'end effector.

$\Delta\phi = [\Delta\varphi, \Delta\theta, \Delta\psi]^T$ rappresenta invece la differenza fra l'orientazione di riferimento e quella della terna fissata sull'end effector. Quest'ultima viene a sua volta definita sulla base dei riferimenti forniti dal sistema di controllo o, in alternativa, dal dispositivo master.

Nel calcolo della matrice R_r^e , in particolare, viene utilizzata $\Delta\phi$ al fine di generare una rotazione dell'end effector tale da far convergere progressivamente la misura degli angoli di *roll-pitch-yaw* dell'end effector ϕ_e con quelli stimati dal sensore ϕ_s .

Per ottenere un'aggiustamento graduale e fluido dell'assetto con errore nullo a regime, tale riferimento ϕ^* non deve tuttavia venire passato direttamente al controllore di posizione e orientazione del robot. Il riferimento $\phi^*[k] = \phi^*(kT_s)$ da fornire al controllore viene piuttosto aggiornato in accordo con la seguente legge di controllo:

$$R_r^b(\phi^*[k+1]) = R_e^b(\phi^*[k])R_r^e(K_\phi e_\phi[k]) \quad (4.2.2)$$

dove

$$e_\phi[k] := \begin{cases} \frac{\Delta\phi[k]}{\|\Delta\phi[k]\|} & \text{if } \|\Delta\phi[k]\| \geq \varepsilon_\phi \\ \frac{1}{\varepsilon_\phi} \Delta\phi[k] & \text{if } \|\Delta\phi[k]\| < \varepsilon_\phi \end{cases} \quad (4.2.3)$$

$$\Delta\phi[k] := \begin{cases} \Delta\phi[k] = \phi_s[k] & \text{if } \text{obj. detection} = \text{TRUE} \\ \Delta\phi[k] \text{ s.t. 4.2.5} & \text{if } \text{obj. detection} = \text{FALSE} \end{cases} \quad (4.2.4)$$

con

$$R_r^e(\Delta\phi[k]) = [R_e^b(\phi_e[k])]^{-1} R_r^b(\phi^*[k]) \quad (4.2.5)$$

Nell'equazione 4.2.5, $\phi^*[k]$ rappresenta in particolare il riferimento di orientazione fornito dal dispositivo master. Si specifica infine che *obj. detection = TRUE* significa che il sensore individua un oggetto all'interno del proprio campo di misura.

La costante K_ϕ rende conto, invece, della rapidità di convergenza del sistema alla condizione $\|\Delta\phi[k]\| < \varepsilon$, ovvero alla condizione di allineamento dell'end effector del robot alla normale al piano individuato dal sensore.

L'equazione 4.2.2 può essere anche interpretata come la legge di controllo integrale di un anello esterno, la quale determina i riferimenti in ingresso all'anello più interno di controllo di orientazione e posizione del robot.

Tali riferimenti vengono generati sulla base delle informazioni fornite dal sensore di prossimità, ovvero l'orientazione ϕ_s del piano rispetto alla terna di riferimento dell'end effector, e delle informazioni provenienti dalla soluzione della cinematica diretta del robot, in particolare l'orientazione $\phi_e = \phi_e(q)$ dell'end effector rispetto alla terna base, nota la misura degli angoli dei sei giunti q .

4.2.1 Criteri di commutazione del riferimento del dispositivo slave con controllo automatico dell'orientazione

Al fine di determinare il criterio di passaggio tra controllo 'manuale', ovvero attraverso il dispositivo master, e controllo automatico dell'assetto dell'end effector si è realizzata la macchina a numero di stati finiti in Fig.4.3.

In questo modo lo stato futuro del sistema, ovvero il tipo di controllo da attuare, viene calcolato a partire dal valore degli ingressi e dallo stato presente.

Fino a quando un oggetto non viene individuato dal sensore di prossimità, la legge di controllo 4.2.2 viene utilizzata per calcolare i riferimenti per l'anello interno di regolazione di posizione ed orientazione del robot, sulla base dei riferimenti forniti dal dispositivo master Phantom Omni. Si opera quindi in una modalità di lavoro definita *controllo manuale*.

Il *controllo automatico* di orientazione viene, invece, attivato non appena tutti e tre i fototransistor rilevano una distanza inferiore a una determinata soglia di sicurezza, al di sotto della quale si ritiene affidabile il valore fornito, in quanto compreso nella portata nominale del sensore stesso.

E' importante sottolineare come i cambiamenti nell'orientazione relativa possono essere causati dal movimento dell'oggetto individuato rispetto all'end effector, o viceversa, dal movimento dell'end effector (sulla base dei movimenti del dispositivo master replicati a lato slave) rispetto ad un oggetto fermo.

Nel momento in cui, in modalità di controllo automatico, il sensore non individua più l'oggetto, il comando deve essere restituito al dispositivo master, abilitando il controllo manuale dell'orientazione dell'end effector.

Poiché dopo la procedura di allineamento automatico l'orientazione dell'end effector può risultare molto differente rispetto a quella imposta dal dispositivo master, si è scelto di utilizzare la legge di controllo 4.2.2 anche per gestire la transizione tra i due controlli ed evitare che quest'ultima risulti troppo veloce. In questo caso, tuttavia, l'orientazione ϕ^* ,

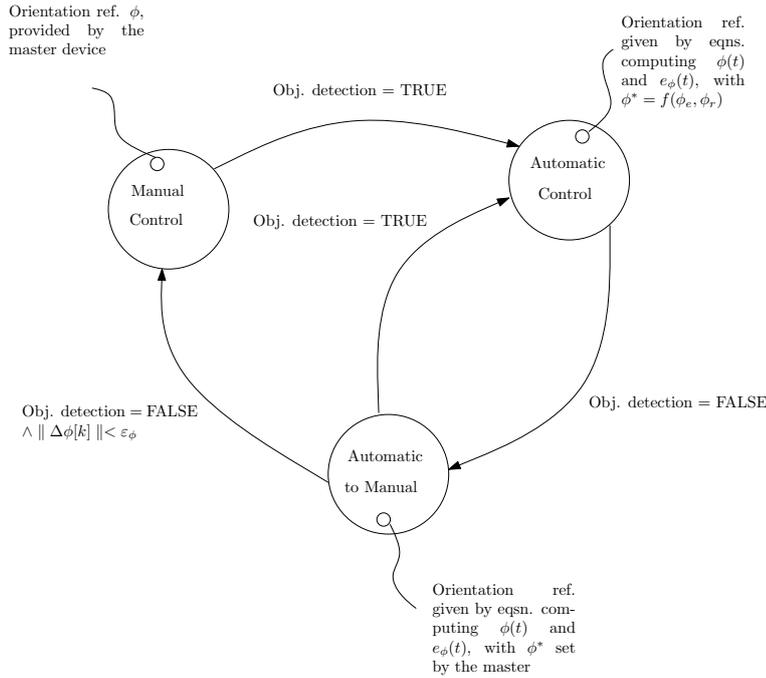


Fig. 4.3: Controllo automatico di orientazione - Macchina a stati finiti, definisce i criteri per il passaggio tra controllo manuale e automatico

che si impone come riferimento in ingresso all'anello di controllo, rappresenta l'orientazione imposta dal dispositivo master.

Infine, quando l'end effector ha raggiunto la condizione di allineamento con il riferimento imposto dal sistema di controllo, ovvero quando vale la condizione $\|\Delta\phi[k]\| < \varepsilon$, la legge di controllo 4.2.2 non viene più utilizzata e il riferimento di orientazione fornito dal master viene passato direttamente al controllore di posizione ed orientazione del robot slave, concludendo così la procedura di transizione tra le due tipologie di controllo.

4.3 Controllo automatico della distanza dell'end effector dal target

Al fine di evitare il contatto con il paziente e garantire una migliore risoluzione delle immagini raccolte, si è predisposto, parallelamente al controllo automatico dell'orientazione, anche un controllo automatico della posizione dell'end effector.

Viene data all'operatore la possibilità di scegliere una distanza minima d_{min} , al di sotto della quale non si vuole che l'end effector proceda, qualunque sia il riferimento imposto dal master.

Si denoti con $r[x, y, z]$ una generica terna di coordinate cartesiane. Siano quindi $r_P = [x_p, y_p, z_p]$ le coordinate dell'end effector e $r_R = [x_r, y_r, z_r]$ le coordinate del riferimento di posizione passato dal Phantom al dispositivo slave.

Una volta calcolato il vettore normale al piano in analisi attraverso la relazione:

$$\mathbf{n}_r = R_r^b(\phi)[-1, 0, 0] \quad (4.3.1)$$

è possibile ricavare le coordinate $r_O = [x_0, y_0, z_0] = \mathbf{r}_P + d\mathbf{n}_r$ della proiezione del punto di coordinate r_P sul piano del target, dove d rappresenta la distanza dell'end effector dal piano, come è possibile osservare in Fig.4.4.

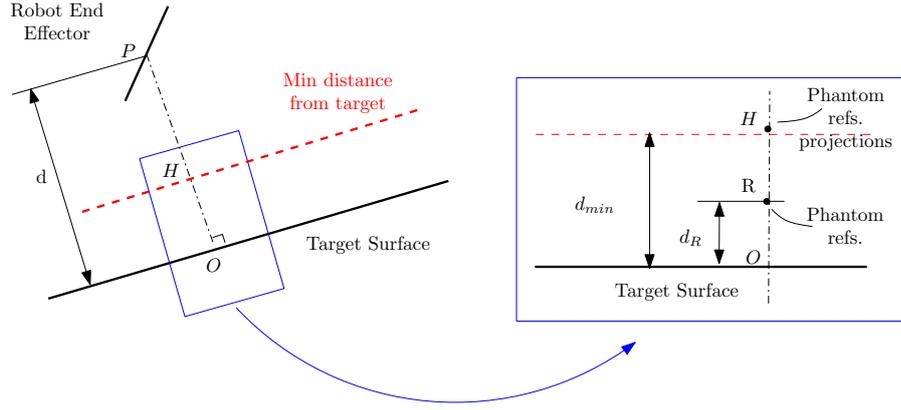


Fig. 4.4: Controllo automatico della distanza - Distanza minima di avvicinamento dell'end effector a target

La distanza del riferimento r_R dal piano si ottiene come:

$$d_R = (\mathbf{r}_P - \mathbf{r}_O)n_r \quad (4.3.2)$$

Il sistema di controllo esegue un confronto fra la distanza d_R del Phantom dal piano e il valore del vincolo d_{min} . Qualora risulti $d_R - d_{min} < 0$, l'anello di controllo forzerà il riferimento di posizione r^* del robot alle coordinate r_H , che corrispondono al punto di proiezione delle coordinate di riferimento del Phantom r_R sul piano del vincolo di distanza minima, e che viene calcolato come segue:

$$\mathbf{r}_H = r_P - (d_R - d_{min})n_r \quad (4.3.3)$$

Le coordinate r_H rappresentano il riferimento che deve essere inseguito dal controllo automatico del robot. Tuttavia, al fine di ottenere anche in questo caso un aggiustamento graduale e fluido, si è scelto di calcolare il riferimento da fornire al controllore attraverso la seguente equazione:

$$r^*[k+1] = r^*[k] + k_r e_r[k] \quad (4.3.4)$$

dove

$$e_r[k] := \begin{cases} \frac{\Delta e[k]}{\|\Delta e[k]\|} & \text{if } \|\Delta e[k]\| \geq \varepsilon_r \\ \frac{1}{\varepsilon_r} \Delta e[k] & \text{if } \|\Delta e[k]\| < \varepsilon_r \end{cases} \quad (4.3.5)$$

e

$$\Delta e[k] := \begin{cases} r_H[k] - r_P[k] & \text{if rigid. constr. detection} = \text{TRUE} \\ r_R[k] - r_P[k] & \text{if rigid. constr. detection} = \text{FALSE} \end{cases} \quad (4.3.6)$$

Rigid. constr. on = TRUE corrisponde alla situazione in cui i riferimenti r_R del master si trovano al di sotto della soglia di distanza minima d_{min} .

Va sottolineato come il controllo automatico di posizione venga attivato solo nel momento in cui il sensore rileva la presenza di un oggetto all'interno della propria portata nominale e, allo stesso tempo, la differenza fra d_R e d_{min} rimane al di sotto di una determinata soglia di isteresi.

La necessità di imporre un controllo con isteresi deriva dal fatto che le misure rilevate dal sensore sono affette da un certo livello di rumore. In assenza di soglie di isteresi, la minima oscillazione nella distanza misurata dai fototransistor causerebbe una continua commutazione fra le due modalità di controllo e, di conseguenza, un continuo cambiamento dei riferimenti di posizione passati al dispositivo slave.

L'attivazione di un controllo automatico della posizione consente, infine, di risolvere un'ulteriore problematica del sistema. Come visto nella sezione 2.4, infatti, i sensori di prossimità presentano una caratteristica completa come in Fig.2.21, che nella pratica viene ridotta solo alla seconda parte, monotona decrescente. Ne deriva che, qualora il sensore misuri una distanza inferiore al punto di inversione di pendenza della curva di calibrazione, il sistema interpreterà l'abbassamento di tensione percepito come un aumento della distanza, producendo dei riferimenti di controllo errati.

Imponendo quindi una distanza minima fra end effector e piano del target superiore al punto di inversione della pendenza, è possibile evitare di incorrere nella situazione descritta e assicurare la corretta generazione dei riferimenti.

4.3.1 Criteri di commutazione del riferimento del dispositivo slave con controllo automatico della distanza minima dal piano

Analogamente a quanto fatto per il controllo automatico di orientazione, si è realizzata la macchina a numero di stati finiti in Fig.4.5, al fine di determinare il criterio di passaggio tra controllo manuale e controllo automatico della distanza dell'end effector dal piano.

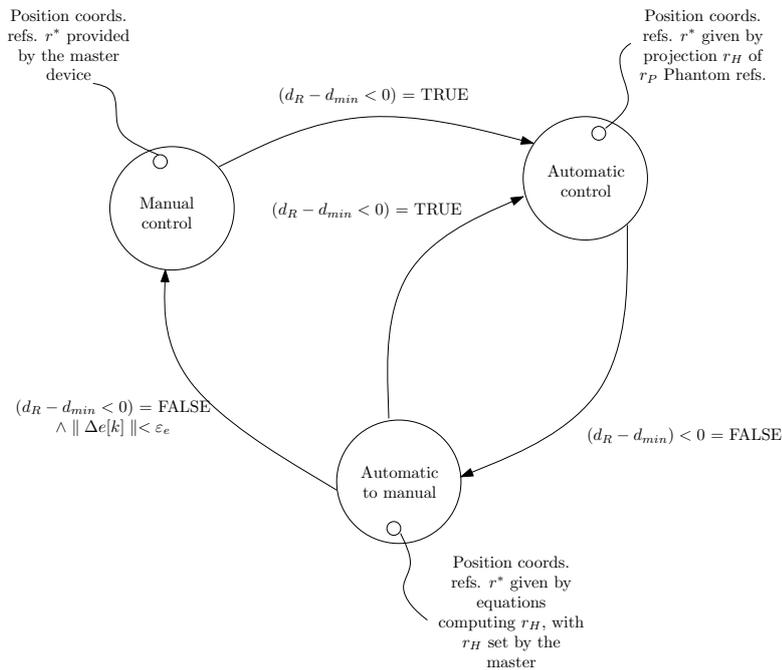


Fig. 4.5: Controllo automatico della distanza - Macchina a stati finiti, definisce i criteri per il passaggio tra controllo manuale e automatico

Anche in questo caso, fino a quando il sensore non individua un oggetto nel proprio campo di visione e i riferimenti r_R del Phantom non si trovano al di sotto della distanza minima d_{min} , si lavora in modalità di *controllo manuale*. I riferimenti di posizione r^* per l'anello di regolazione di posizione del robot vengono quindi passati direttamente dal dispositivo master.

Il *controllo automatico* di posizione viene, invece, attivato non appena tutti e tre i fototransistor rilevano una distanza inferiore a una determinata soglia di sicurezza e risulta allo stesso tempo verificata la condizione $d_R - d_{min} < 0$.

In questo caso il sistema di controllo a lato slave impone $r^* = r_H$, ovvero che i riferimenti di posizione coincidano con la proiezione dei riferimenti del Phantom sul piano di distanza minima.

Nel momento in cui, in modalità di controllo automatico, il sensore non individua più l'oggetto o i riferimenti r_P del Phantom si trovano al di sopra della distanza minima dal piano, il comando deve essere restituito al dispositivo master, abilitando il controllo manuale dell'end effector.

Poiché, anche in questo caso, dopo la procedura di controllo automatica le coordinate r_P dell'end effector possono risultare molto differenti rispetto a quelle imposte dal dispositivo master, si è scelto di gestire la transizione tra i due controlli calcolando il riferimento tramite la 4.3.4. In questo caso le coordinate di posizione r^* , che si impongono come riferimento in ingresso all'anello di controllo, rappresentano le coordinate imposte dal dispositivo master.

Infine, quando l'end effector ha raggiunto la posizione corrispondente al riferimento imposto, ovvero quando vale la condizione $\|\Delta e[k]\| < \varepsilon$, la legge di controllo 4.3.4 non viene più utilizzata e il riferimento fornito dal master viene passato direttamente al controllore di posizione ed orientazione del robot slave, concludendo così la procedura di transizione tra le due tipologie di controllo.

4.4 Generazione del feedback di forza

Quando un ostacolo si trova all'interno del campo di misura del sensore di prossimità, si desidera che venga generato un feedback di forza a lato master.

Tale forza, di natura visco-elastica repulsiva, simula la presenza di un vincolo virtuale, situato a una certa distanza dalla superficie che deve essere analizzata mediante la sonda. L'intensità della forza dipenderà dalle misure di distanza ed orientazione dell'end effector rispetto al piano fornite dal sensore e dal valore dell'impedenza del vincolo virtuale che si desidera ricreare a lato master.

Grazie a questo feedback, l'operatore è in grado di individuare in modo più agevole la distanza ottimale a cui posizionare la sonda dalla superficie, al fine di ottenere una migliore qualità delle immagini raccolte durante la scansione, e di evitare inoltre di incorrere nell'eventualità di un contatto con il paziente.

Nel sistema presentato si è scelto di generare un ritorno di forza a lato master diretto ortogonalmente rispetto all'end effector del robot e in direzione dello stesso. In questo modo l'utente percepisce in modo molto intuitivo la direzione lungo la quale muoversi per aggiustare la posizione della sonda.

Al fine di determinare la direzione della forza si è definito, quindi, il versore $\mathbf{u}_F(t)$, espresso rispetto al sistema di riferimento base del robot, e dato da:

$$\mathbf{u}_F^b(t) = R_e^b(\phi^*)\mathbf{u}_x \quad (4.4.1)$$

dove $\mathbf{u}_x = [-1, 0, 0]^T$ rappresenta il versore parallelo all'asse x del sistema di riferimento solidale all'end effector, cambiato di segno.

In base alle convenzioni definite nella sezione 2.4 per quanto riguarda i sistemi di riferimento del robot, il versore relativo all'asse x della terna solidale all'end effector risulta ortogonale e uscente da esso. Il segno negativo consente, quindi, di ottenere un versore rivolto in direzione dell'end effector, come desiderato.

L'entità della forza di feedback viene calcolata invece come segue:

$$F(t) = \begin{cases} -b_v \frac{d}{dt} \Delta d(t) - k_v \Delta d(t) & \text{se } \Delta d(t) \geq 0 \\ 0 & \text{altrimenti.} \end{cases} \quad (4.4.2)$$

$\Delta d(t)$ costituisce la profondità di penetrazione nel vincolo virtuale e si definisce come $\Delta d(t) := d^* - d(t)$, dove d^* denota la distanza del vincolo dal piano.

Le costanti k_v e b_v sono rispettivamente la costante di rigidità e il coefficiente di attrito viscoso del vincolo virtuale, essendo quest'ultimo di natura visco-elastica. La precedente relazione può essere espressa in modo alternativo anche in termini di impedenza meccanica $Z_v(s)$ del vincolo virtuale. Nel dominio di Laplace si ottiene così quanto segue:

$$F(s) = Z_v(s)V(s) = -\left(b_v + \frac{k_v}{s}\right)V(s) \quad (4.4.3)$$

dove $V(s)$ in particolare denota la trasformata di Laplace della velocità di penetrazione nel vincolo virtuale.

4.4.1 Generazione di una forza sinusoidale sovrapposta

Si è scelto, infine, di apportare alcune modifiche nella generazione del feedback di forza, qualora entri in funzione il controllo automatico di posizione del robot.

In questa modalità di controllo, infatti, l'operatore potrebbe continuare a imporre attraverso il dispositivo master dei riferimenti di posizione r_R di molto inferiori al piano di distanza minima, senza che la posizione dell'end effector ne venga influenzata, essendo questa controllata interamente a lato slave.

Tuttavia, per garantire un buon livello di stabilità del sistema, è preferibile che i riferimenti r_R provenienti dal Phantom non si mantengano di molto al di sotto della soglia minima.

Si è scelto, quindi, di sommare alla forza di feedback generata secondo la relazione 4.4.2 una forza sinusoidale, in grado di determinare una vibrazione a lato master. In questo modo l'operatore può facilmente rendersi conto della situazione di lavoro in cui si trova, e cercare di correggere la posizione, in modo da mantenere il riferimento r_R non eccessivamente al di sotto della soglia d_{min} .

Implementazione del software di controllo

Viene ora approfondita in particolare la realizzazione a livello software dell'architettura del sistema master/slave e delle diverse modalità di controllo introdotte nel capitolo 4.

Nel capitolo 3 è già stato preso in considerazione come i dispositivi che compongono il sistema si interfacciano con l'ambiente Matlab Simulink utilizzato. Si concentra quindi l'attenzione unicamente sugli algoritmi di controllo automatico dell'orientazione e della posizione dell'end effector del robot e di generazione del feedback di forza.

In particolare si vedrà come i due controlli automatici di orientazione e distanza agiscano in parallelo e siano stati realizzati mediante due implementazioni del tutto simmetriche.

Per la realizzazione di tutti questi algoritmi si è fatto uso delle Embedded Matlab Functions, messe a disposizione dalla libreria Simulink.

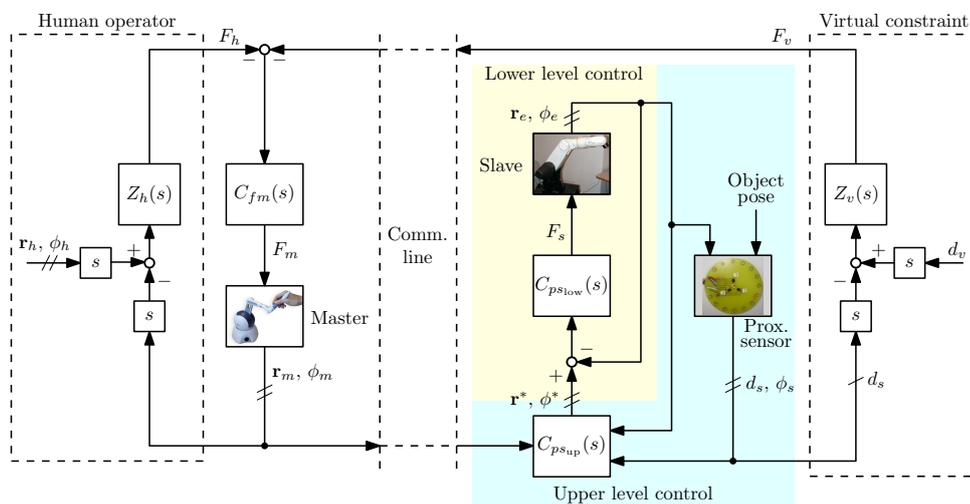


Fig. 5.1: Sistema di controllo master/slave implementato

5.1 Controllo automatico dell'orientazione dell'end effector

Il blocco di controllo automatico dell'orientazione dell'end effector riceve in ingresso i riferimenti di *yaw-pitch-roll* del dispositivo master e li elabora al fine di fornire i riferimenti da passare in ingresso al robot slave.

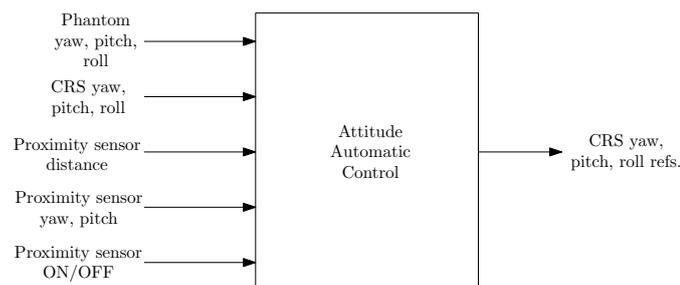


Fig. 5.2: Blocco di controllo automatico dell'orientazione - Ingressi e uscite

La procedura di orientazione automatica dell'end effector viene attivata quando risulta verificato `is_auto_ctrl_enabled = 1`. Come è possibile osservare dalla sezione di codice di seguito riportata, la variabile assume il valore 1 quando il sensore di prossimità misura una grandezza all'interno del proprio campo di misura e quando, allo stesso tempo, l'end effector ha raggiunto una determinata distanza dal target, rappresentata in questo caso dalla variabile $r = 35$.

In questo modo l'allineamento inizia solo nel momento in cui l'end effector del robot si trova nelle immediate vicinanze dell'ostacolo. Così facendo, la procedura si basa su dati misurati dal sensore meno affetti da rumore e fluttuazioni, rendendo più accurata la stima degli angoli di yaw e pitch del piano, rispetto alla terna fissa con l'end effector.

```

1 %% check distance from plane
2 r = 35;
3 e = r - PRXS_dist;
4 eL = -5;    % hysteresis threshold low
5 eH = 5;    % hysteresis threshold high
6 if (e > eH) && (y_prev == 0),
7     y = 1;
8 elseif (e < eL) && (y_prev == 1),
9     y = 0;
10 else
11     y = y_prev;
12 end;
13 %% update output
14 is_auto_ctrl_enabled = is_PRXS_on && y;

```

Data la notevole fluttuazione riscontrata nelle misure fornite dai tre fototransistor, in particolare nelle misure corrispondenti a distanze elevate dal piano, si è scelto di inserire una banda di isteresi del valore di $\pm 5mm$. In questo modo si impedisce che, in prossimità della transizione vi sia un continuo cambiamento fra controllo manuale e automatico, a cui corrisponderebbe una variazione altrettanto continua del riferimento di assetto fornito all'end effector.

La scelta dell'ampiezza della banda è stata effettuata osservando la massima ampiezza picco-picco di oscillazione della misura, in corrispondenza di una distanza minima dal target mediamente pari a $20mm$, che corrisponde alla distanza a cui viene posizionata generalmente la sonda dal paziente durante un esame CEUS.

L'uscita del controllore varia a seconda della modalità di lavoro in cui si trova il sistema. In controllo automatico, il riferimento che viene passato rappresenta l'implementazione a livello software dell'equazione 4.2.2, presentata nella precedente sezione 4.2. L'incremento definito come $\Delta\phi[k]$ viene utilizzato nel calcolo della matrice R_r^b , al fine di ricavare i valori effettivi di yaw, pitch e roll da passare al robot.

```

1 %% Determine controller output
2 CRS_DeltaYPR = [0; 0; 0];
3 if is_auto_ctrl_on,
4     if is_auto_ctrl_enabled,
5         CRS_DeltaYPR = [PRXS_YP(1); PRXS_YP(2); 0];
6     else
7         Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
8         Rbph = get_DCM(PHANToM_YPR_ref(1), rgb].133,.545,.133...
9             PHANToM_YPR_ref(2), PHANToM_YPR_ref(3));
10        Reph = inv(Rbe)*Rbph;
11        [DY, DP, DR] = get_YPR(Reph);
12        CRS_DeltaYPR = [DY; DP; DR];
13    end;

```

```

13     CRS_YPR_ref = rgb].133,.545,.133...
        get_CRS_YPR_auto_ctrl_output(CRS_YPR, rgb].133,.545,.133...
        CRS_DeltaYPR, prev_CRS_YPR_ref);
14 else
15     CRS_YPR_ref = PHANTOM_YPR_ref;
16 end;

```

La funzione `get_CRS_YPR_auto_ctrl_output`, come è possibile osservare dal seguente estratto del codice di controllo, calcola innanzitutto i valori degli errori fra gli angoli di yaw, pitch e roll attuali del robot e i riferimenti. Tali valori vengono quindi utilizzati per calcolare la matrice qui indicata con Rep , che rappresenta la matrice di rotazione dal piano del target all'end effector.

```

1 eY = sat(CRS_DeltaYPR(1), -5*pi/180, 5*pi/180)/(5*pi/180);
2 eP = sat(CRS_DeltaYPR(2), -5*pi/180, 5*pi/180)/(5*pi/180);
3 eR = sat(CRS_DeltaYPR(3), -5*pi/180, 5*pi/180)/(5*pi/180);
4 k = 0.0005;
5 Rbe = get_DCM(prev_CRS_YPR_ref(1), rgb].133,.545,.133...
        prev_CRS_YPR_ref(2), prev_CRS_YPR_ref(3));
6 Rep = get_DCM(k*eY, k*eP, k*eR);
7 Rbp = Rbe*Rep;
8 CRS_YPR_ref = zeros(3,1);
9 [CRS_YPR_ref(1), CRS_YPR_ref(2), CRS_YPR_ref(3)] = get_YPR(Rbp);
10 end

```

Qualora, invece, il sistema stesse funzionando in modalità di controllo manuale, allora i riferimenti del dispositivo slave andrebbero posti pari ai riferimenti imposti dal master.

In particolare i valori di riferimento degli angoli pitch e yaw sono stati calcolati attraverso la funzione `get_YPR`, utilizzando le formule relative alla soluzione di problemi di cinematica, mediante la parametrizzazione secondo gli angoli aeronautici, riportata in appendice [B.2](#).

```

1 function [Y, P, R] = get_YPR(DCM)
2 t1 = sqrt( DCM(1,1)^2 + DCM(2,1)^2 );
3 P = atan2(-DCM(3,1), t1);
4 if abs( abs(P) - pi/2 ) < 0.001
5     Y = 0;
6     if P > 0
7         R = atan2(DCM(1,2), DCM(2,2));
8     else
9         R = -atan2(DCM(1,2), DCM(2,2));
10    end
11 else
12    Y = atan2(DCM(2,1), DCM(1,1));
13    R = atan2(DCM(3,2), DCM(3,3));
14 end;
15 end

```

Al fine di non incorrere in singolarità del meccanismo, sia lavorando in controllo manuale che automatico, si è scelto di predisporre delle soglie di saturazione dei riferimenti da inviare al robot slave. Questo attraverso la funzione `sat`, la quale fornisce in uscita il valore massimo fra quello che si desidera fornire come riferimento e il limite minimo imposto, o il valore minimo fra i due del caso della soglia massima.

```

1 function u_sat = sat(u, umin, umax)
2 u = min([u, umax]);
3 u = max([u, umin]);
4 u_sat = u;

```

```
5 end
```

A valle della procedura di determinazione dell'uscita del controllore, vengono poste quindi le seguenti righe di codice:

```
1 CRS_Y_ref_sat = sat(CRS_YPR_ref(1), -pi, pi);
2 CRS_P_ref_sat = sat(CRS_YPR_ref(2), -pi, pi);
3 CRS_R_ref_sat = sat(CRS_YPR_ref(3), -pi, pi);
```

I livelli di saturazione sono stati scelti al fine di limitare i movimenti dei giunti alle esigenze necessarie per l'applicazione, scegliendo quindi π come limite massimo, sia in positivo che in negativo.

5.1.1 Criteri di commutazione del riferimento

Il riferimento da inviare al dispositivo slave dipende dal tipo di controllo con cui si sta lavorando, manuale o automatico.

Se si sta lavorando in modalità di controllo automatico, ma non sussistono più le condizioni viste sopra per la sua abilitazione e allo stesso tempo l'orientazione del robot corrisponde ai riferimenti imposti precedentemente, a meno di errori dell'ordine di qualche grado, allora il sistema passa in modalità di controllo manuale, portando il valore della variabile $is_auto_ctrl_on = 0$.

Se, invece, lavorando in modalità di controllo manuale dovessero verificarsi le condizioni viste sopra che determinano la condizione $is_auto_ctrl_enabled = 1$, allora la modalità di lavoro cambierebbe da manuale ad automatica.

```
1 %% Determine next controller state
2 is_aligned = 0;
3 if is_auto_ctrl_on,
4     aligned_th = 2*pi/180;
5     if (~is_auto_ctrl_enabled) && all(CRS_DeltaYPR < aligned_th),
6         is_auto_ctrl_on = 0;
7     end;
8 else
9     if is_auto_ctrl_enabled,
10        is_auto_ctrl_on = 1;
11    end;
12 end;
```

5.2 Controllo automatico della distanza dell'end effector

Si è creata innanzitutto una funzione in grado di segnalare il raggiungimento o meno da parte dell'end effector della distanza minima dal target, attraverso il valore assunto dalla variabile $is_rigid_constr_on$.

Con riferimento alla Fig.4.4 della sezione 4.3, si misura la differenza $e = d_R - d_{min}$ e la si confronta con le soglie di isteresi e_L e e_H .

Il valore di $is_rigid_constr_on$, si determina quindi sulla base del valore precedentemente assunto dalla medesima variabile e dal valore di e , come in un tipico diagramma di isteresi.

La presenza di un'isteresi nella misura dell'errore di posizione è dovuta alle stesse considerazioni esposte nella precedente sezione 5.1. Anche in questo caso la presenza di un blocco di isteresi consente che, in corrispondenza della condizione limite di transizione fra

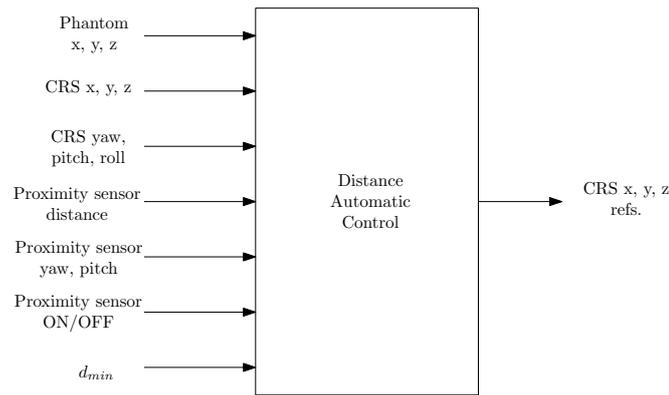


Fig. 5.3: Blocco di controllo automatico della distanza - Ingressi e uscite

le due modalità di controllo $d_R - d_{min}$ si realizzi un continuo cambiamento del riferimento di posizione dell'end effector.

```

1  if is_PRXS_on
2      eL = -5; % hysteresis threshold low
3      eH = 5; % hysteresis threshold high
4      if (e > eH) && (is_rigid_constr_on_prev == 1),
5          is_rigid_constr_on = 0;
6      elseif (e < eL) && (is_rigid_constr_on_prev == 0),
7          is_rigid_constr_on = 1;
8      else
9          is_rigid_constr_on = is_rigid_constr_on_prev;
10     end;
11 else
12     is_rigid_constr_on = 0;
13 end;
  
```

Per quanto riguarda, invece, la determinazione dell'uscita del controllore in modalità di funzionamento automatico, il riferimento da inviare al dispositivo slave CRS_XYZ_ref viene calcolato attraverso la 4.3.4, come visto nella sezione 4.3, dove la variabile k_r rende conto della rapidità di convergenza della soluzione al riferimento.

Se, invece, il controllo automatico risulta inattivo, si nota come i riferimenti di posizione vengano posti uguali a quelli inviati dal Phantom.

```

1  if is_auto_ctrl_on,
2      if norm_CRS_XYZ_Delta < 20,
3          norm_CRS_XYZ_Delta = 20;
4      end;
5      CRS_XYZ_ref = CRS_XYZ_ref_prev +
6                  k*CRS_XYZ_Delta/norm_CRS_XYZ_Delta;
7  else
8      CRS_XYZ_ref = PHANTOM_XYZ_ref;
9  end;
  
```

I riferimenti imposti allo slave, siano essi provenienti dal master o dal controllore automatico, possono portare il robot in condizioni di singolarità o saturazione dei giunti.

Si è quindi deciso di restringere lo spazio di lavoro del robot a un'area limitata, consona all'applicazione in esame, e quindi confinata alla lunghezza all'incirca di un arto umano e a un'altezza tale da consentire l'ingresso e l'uscita del braccio robotico dalla zona di controllo automatico di orientazione e posizione in modo agevole.

Questo è stato realizzato attraverso l'imposizione a livello software di alcune saturazioni al valore delle coordinate da fornire come riferimenti al robot stesso, come è possibile osservare nelle righe di codice riportate di seguito:

```

1 CRS_X_ref_sat = sat(CRS_XYZ_ref(1), 304.8-100, 304.8+200);
2 CRS_Y_ref_sat = sat(CRS_XYZ_ref(2), -100, 100);
3 CRS_Z_ref_sat = sat(CRS_XYZ_ref(3), 330.2-40, 330.2*2+40);
4 CRS_XYZ_ref = [CRS_X_ref_sat; CRS_Y_ref_sat; CRS_Z_ref_sat];

```

Sperimentalmente si è riscontrato, infine, un comportamento tendente all'instabilità da parte del robot, nel momento in cui i riferimenti del Phantom si trovano di molto al di sotto della distanza minima imposta dal controllo automatico di posizione. Ciò è dovuto in parte alla tecnica che si è scelto di utilizzare, in particolare al fatto di basarsi sulle misure fornite dal sensore, caratterizzate come noto da fluttuazioni anche elevate, per calcolare le proiezioni del master sul piano di distanza minima. Di fatto, più tali riferimenti si allontanano dal piano di distanza minima, più imprecisa diviene la stima della proiezione su tale piano, causando visibili oscillazioni dell'end effector.

In queste condizioni, l'introduzione di una forza sinusoidale sovrapposta al segnale di feedback, come verrà illustrato con maggiore dettaglio nella sezione 5.3, consente all'operatore di allinearsi il più possibile alla condizione di distanza minima anche con i riferimenti del Phantom ed evitare di incorrere nel comportamento potenzialmente instabile sopra descritto.

5.2.1 Criteri di commutazione del riferimento

Il seguente segmento di codice rappresenta di fatto la realizzazione software della macchina a stati finiti del controllo automatico di posizione di Fig.4.5 della sezione 4.3.

Il controllo automatico viene attivato nel momento in cui i riferimenti di posizione del Phantom si trovano al di sotto del piano di distanza minima dal target. Quando, invece, tali riferimenti si trovano al di sopra della soglia e la posizione dell'end effector del robot è quasi uguale a quella imposta come riferimento, allora il controllo automatico viene disattivato.

```

1 CRS_XYZ_Delta = CRS_XYZf - CRS_XYZi;
2 norm_CRS_XYZ_Delta = norm(CRS_XYZ_Delta);
3 if is_auto_ctrl_on,
4     if (¬is_rigid_constr_on) && (norm_CRS_XYZ_Delta < 30)
5         is_auto_ctrl_on = 0;
6     end;
7 else
8     if is_rigid_constr_on,
9         is_auto_ctrl_on = 1;
10    end;
11 end;

```

5.3 Generazione del feedback di forza

Il blocco in 5.4 si occupa della generazione del feedback di forza da inviare al dispositivo master. Esso riceve in ingresso la distanza misurata dal sensore di prossimità e la distanza dal target, in corrispondenza della quale si desidera iniziare a percepire un vincolo visco-elastico a lato master.

Tale funzione richiede, inoltre, in ingresso la variabile che rende conto dell'attivazione o meno del sensore e le coordinate angolari dell'end effector del robot.

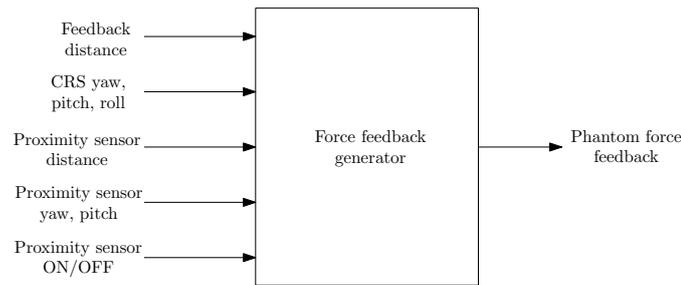


Fig. 5.4: Blocco di generazione del feedback di forza - Ingressi e uscite

La forza di feedback viene calcolata come il prodotto fra il versore della forza e la sua intensità.

```
1 F = aF*uF;
```

Come anticipato nel capitolo 4, si desidera infatti che la forza di feedback venga generata in direzione parallela all'end effector. A livello software si procede, quindi, innanzitutto con il calcolo del versore normale all'end effector, e lo si inverte poi di segno, al fine di replicare una forza diretta verso l'end effector.

```
1 uEE = get_nEE(CRS_YPR);
2 uF = -uEE;
```

Da queste prime righe di codice risulta chiara la necessità di ricevere in ingresso le misure di yaw, pitch e roll del robot, al fine di poter calcolare esattamente il versore da esso uscente, in direzione ortogonale, attraverso la matrice dei coseni direttori.

L'intensità della forza è stata quindi calcolata come uscita di un regolatore PD, al quale giunge in ingresso la differenza fra la distanza FF_dist in corrispondenza della quale si desidera iniziare a percepire un vincolo virtuale e la distanza misurata fra l'end effector e il piano del target. Di conseguenza, essa sarà proporzionale alla distanza dell'end effector dal piano, con proporzionalità inversa.

Se il valore di err_dist dovesse risultare inferiore a zero o se il sensore di prossimità non fosse abilitato, allora l'intensità della forza di feedback risulterebbe pari a zero.

```
1 err_dist = sat(FF_dist-PRXS_dist, 0, FF_dist-rigid_dist);
2 Kp = 50;
3 Kd = 0.5;
4 Ts = 0.001;
5 if is_PRXS_on,
6     aF = get_PD_out(err_dist, Kp, Kd, Ts);
7 else
8     aF = 0;
9 end;
```

Per quanto riguarda in particolare il regolatore PD, dalle righe di codice riportate di seguito si notano in particolare la definizione dei parametri del filtro passa basso e la conversione delle misure da mm a m . Il dispositivo master restituisce infatti una misura in metri, mentre l'intero sistema di controllo considera i mm come unità di misura per le distanze.

```
1 function y = get_PD_out(u, Kp, Kd, Ts)
2 %% Set LPF params
```

```

3 b0 = (1-exp(-2*pi*5*0.001));
4 a0 = exp(-2*pi*5*0.001);
5 %% mm to m conversion
6 u = 0.001*u;
7 %% Get PD out
8 yp = Kp*u;
9 yd = a0*prev_yd + b0*Kd*(u-prev_u)/Ts;
10 y = yp + yd;
11 %% Update prev vars
12 prev_u = u;
13 prev_yd = yd;
14 end

```

5.3.1 Forza sinusoidale sovrapposta nel controllo automatico di orientazione

Come anticipato nel capitolo 4, si desidera fornire un'ulteriore sensazione aptica all'operatore, qualora i riferimenti del dispositivo master superino di molto la soglia minima di controllo della distanza, in condizioni di controllo automatico della distanza.

Si è optato per una forza sinusoidale, in grado di generare una vibrazione sovrapposta a lato master, nel momento in cui vale la condizione $err_dist > FF_dist - rigid_dist$.

La forza generata in questo caso presenta ampiezza $0.5N$ e frequenza pari a $5Hz$.

```

1 if err_dist < (FF_dist-rigid_dist),
2     F = aF*uF;
3     k = 0;
4 else
5     aFsin = 0.5; % rgb].133,.545,.133...
6     fFsin = 5; % rgb].133,.545,.133...
7     NFsin = round(1/fFsin/0.001); % sin rgb].133,.545,.133...
8     Fsin = aFsin * sin(2*pi*k/NFsin);
9
10    F = (aF + Fsin)*uF;
11
12    k = mod(k+1, NFsin);
13 end;

```

5.3.2 Filtraggio del feedback di forza

A livello sperimentale si è riscontrato un andamento notevolmente rumoroso della forza di feedback, in particolare con l'insorgere di notevoli vibrazioni all'ingresso dell'end effector nella zona in cui si percepisce il vincolo di forza. Al fine di limitare tale fenomeno, si è scelto di filtrare l'intensità della forza di feedback, applicando alla grandezza aF un filtro a media mobile, calcolato dalla funzione *mov_avg_filt*.

```

1 function y = mov_avg_filt(u, N);
2 persistent u_prev;
3 if isempty(u_prev),
4     u_prev = zeros(N-1,1);
5 end;
6 y = (1/N)*sum([u; u_prev]);
7 u_prev = [u; u_prev(1:end-1)];
8 end

```

Nel momento in cui si inserisce un filtro in un sistema, si deve tenere conto tuttavia di come esso vada ad influire sulla dinamica complessiva, causando un miglioramento nell'andamento della grandezza filtrata, ma allo stesso tempo dei ritardi, a volte non trascurabili, e una diminuzione della trasparenza dell'interfaccia aptica.

Un filtro di questo tipo costituisce di fatto un filtro FIR, che esegue la media dei precedenti N campioni registrati. Al fine di non limitare eccessivamente la dinamica del sistema si è ritenuto opportuno porre $N = 16$, come compromesso fra livello di attenuazione del rumore desiderato e ritardo introdotto.

L'intensità della forza di feedback verrà quindi elaborata come segue:

```
1 aF = mov_avg_filt(aF, 16);
```

Tutti i listati di codice descritti nel presente capitolo vengono riportati interamente in appendice [D](#).

Risultati sperimentali

Al fine di validare le procedure sopra esposte sono state effettuate delle prove sperimentali. In particolare è stato valutato il comportamento del sistema nelle seguenti tre situazioni:

- L'end effector viene avvicinato alla superficie del target e viene mantenuto fermo in un punto preciso sopra di essa (tracking di un punto della superficie);
- L'end effector viene avvicinato ad una superficie piana e viene fatto scorrere sopra di essa (tracking di una superficie piana);
- L'end effector viene avvicinato ad una superficie curva e viene fatto scorrere sopra di essa (tracking di una superficie curva).



Fig. 6.1: Setup sperimentale

Per ciascuna di queste casistiche è stata valutata la capacità del sistema di inseguire i riferimenti di posizione ed orientazione del master in modalità di controllo manuale e quelli imposti dal controllo in modalità automatica. E' stata inoltre analizzata la capacità del sistema di generare una corretta azione di feedback di forza al variare della posizione dell'end effector sul piano.

In tutte le situazioni si vuole verificare che gli angoli di pitch e yaw misurati dal sensore tendano a zero dopo la procedura di orientazione automatica. Per quanto riguarda invece il controllo della distanza minima dal target, si vuole verificare che i riferimenti forniti dal sistema di controllo rimangano invariati al valore prefissato, pur forzando i riferimenti del master ad assumere valori di molto inferiori alla soglia.

Nelle seguenti figure è stata adottata una convenzione comune: le fasce in arancione che compaiono nei grafici indicano le fasi di attivazione del controllo automatico di assetto dell'end effector del robot slave. In queste condizioni deve risultare verificato che il dispositivo slave segua i riferimenti forniti dal controllo di assetto, indipendentemente da quali siano i riferimenti imposti dal master.

Allo stesso modo, le zone evidenziate in azzurro identificano le fasi di attivazione del controllo automatico della distanza minima dalla superficie del target. In corrispondenza di queste fasi il robot deve seguire il riferimento costante di distanza minima imposto dal controllo automatico, qualsiasi sia la posizione di riferimento fornita dal Phantom.

6.1 Inseguimento di un punto della superficie

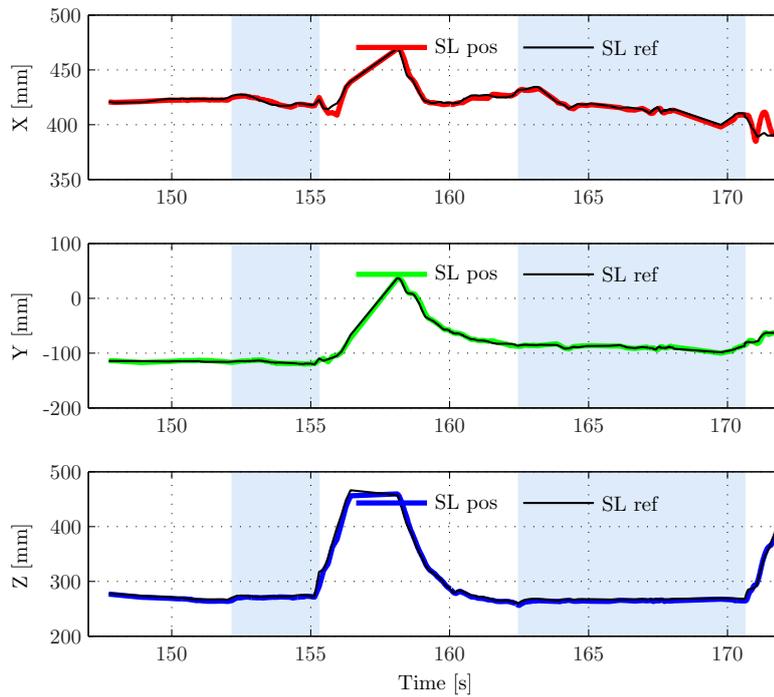


Fig. 6.2: Punto della superficie: inseguimento dei riferimenti di posizione

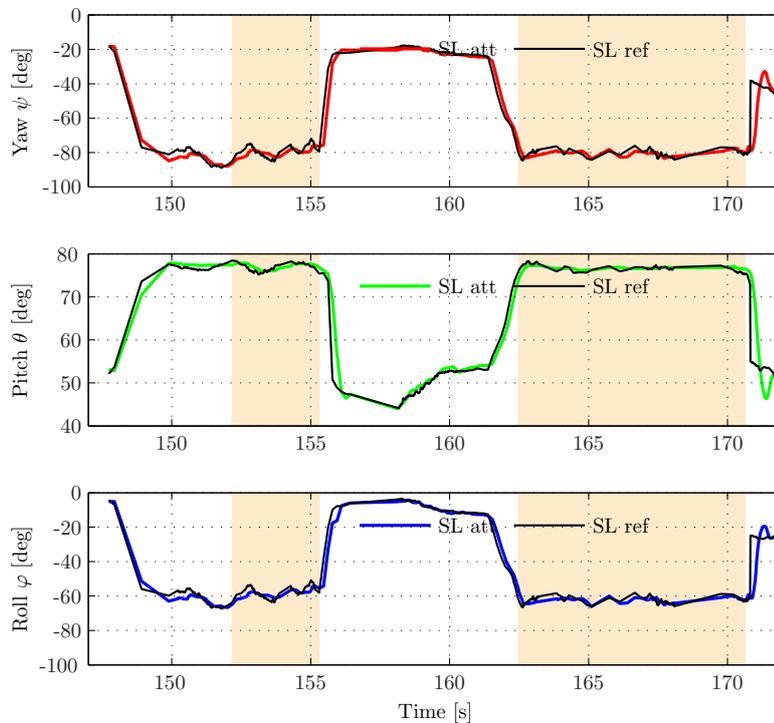


Fig. 6.3: Punto della superficie: inseguimento dei riferimenti di orientazione

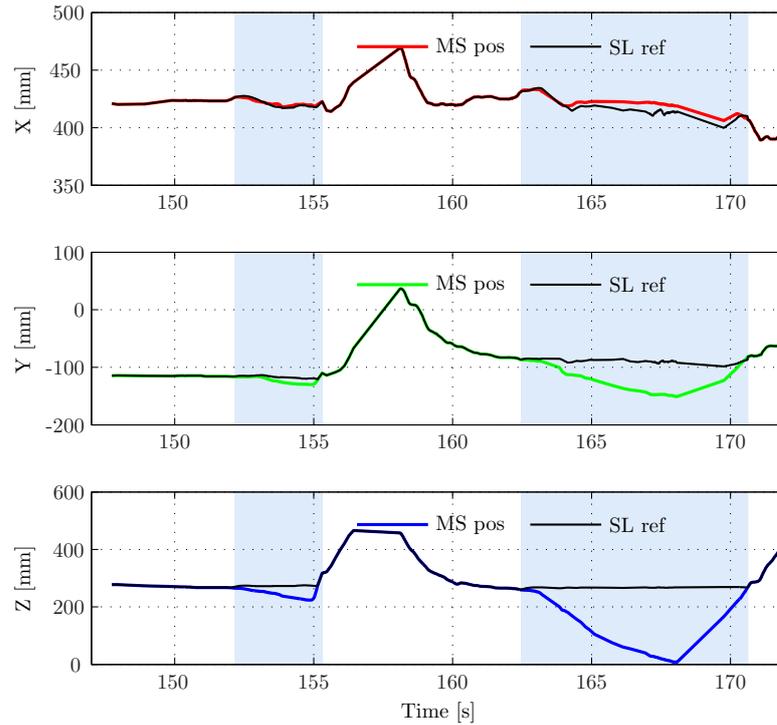


Fig. 6.4: Punto della superficie: confronto fra riferimenti di posizione dei due dispositivi master e slave

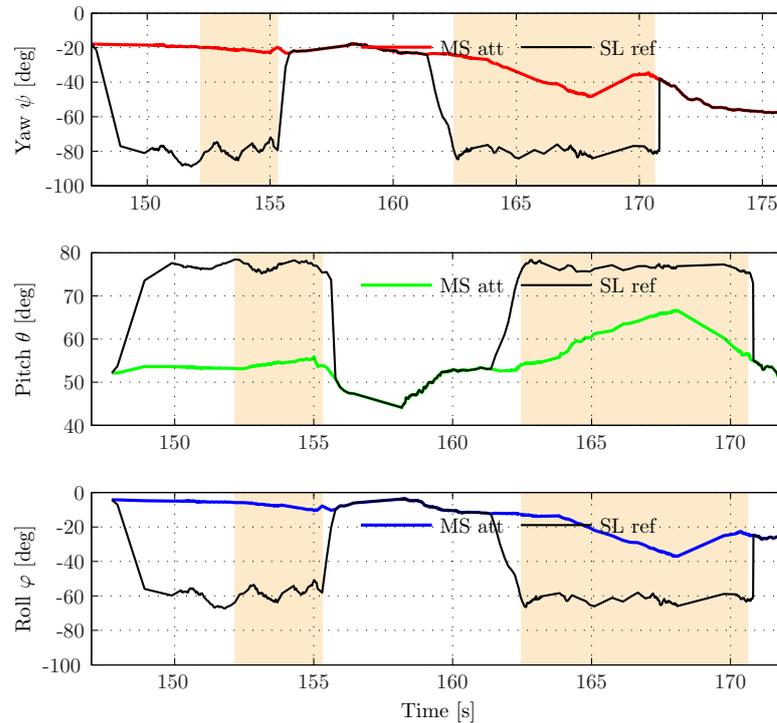


Fig. 6.5: Punto della superficie: confronto fra riferimenti di orientazione dei due dispositivi master e slave

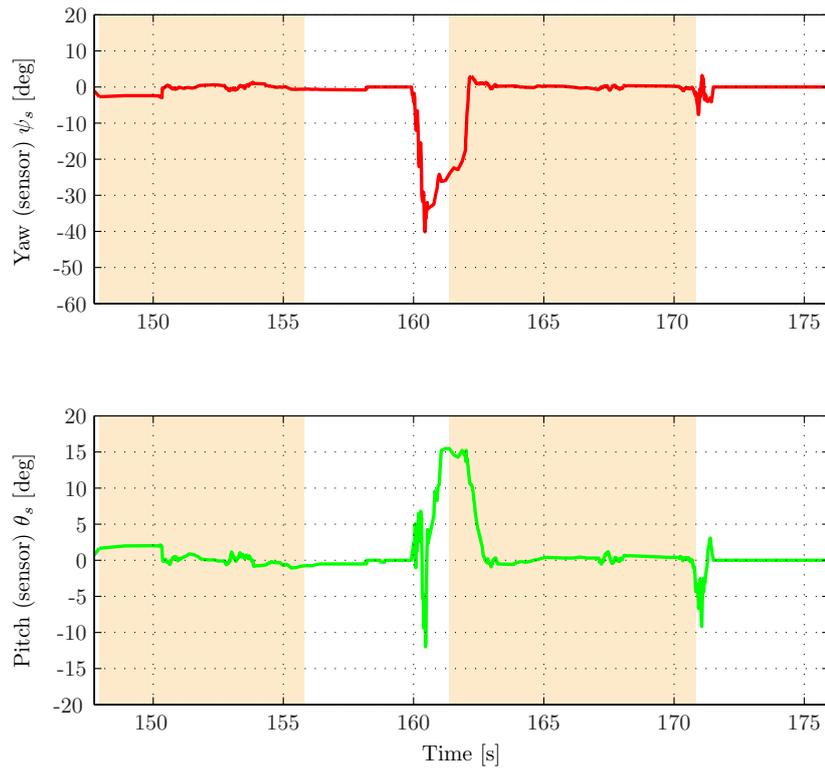


Fig. 6.6: Punto della superficie: andamento degli angoli di yaw e pitch misurati dal sensore

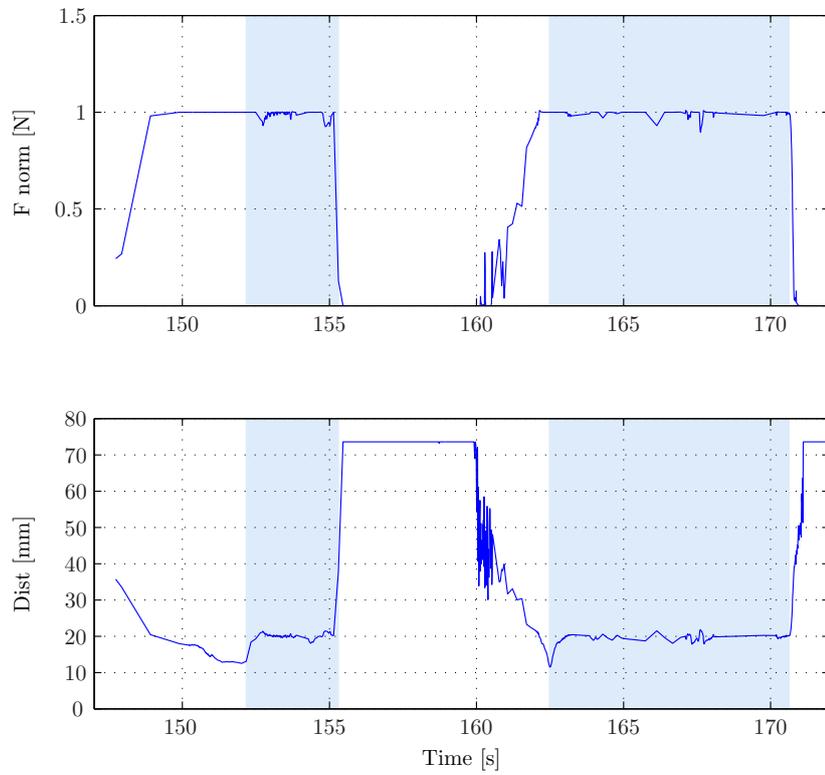


Fig. 6.7: Punto della superficie: evoluzione temporale di forza di feedback e distanza media dalla superficie del target

6.2 Inseguimento di una superficie piana

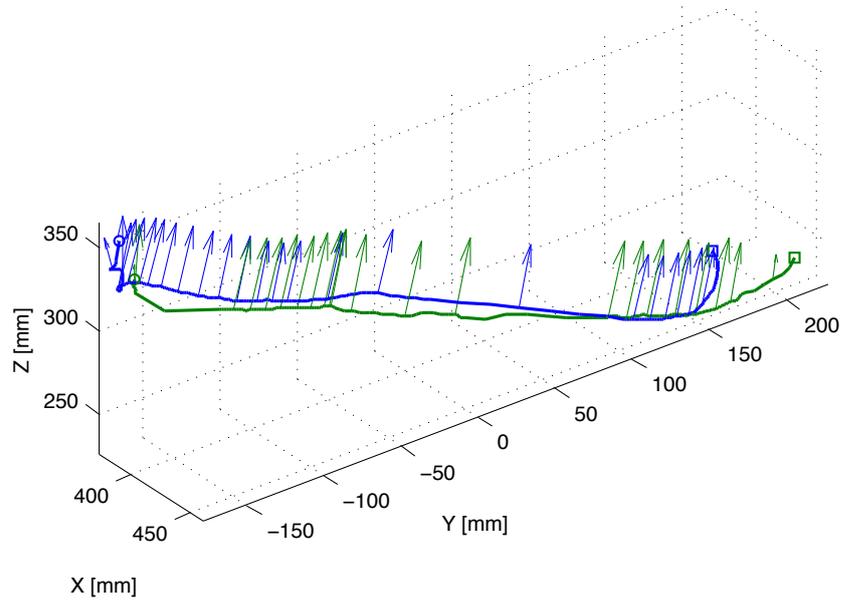


Fig. 6.8: Superficie piana: rappresentazione 3D delle traiettorie seguite

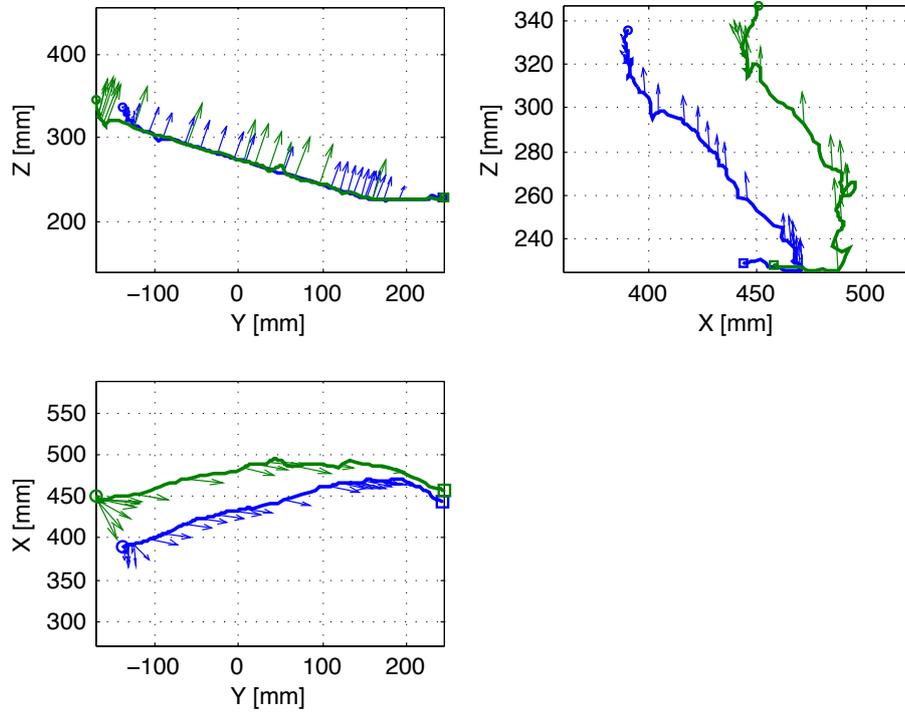


Fig. 6.9: Superficie piana: proiezioni ortogonali delle traiettorie seguite

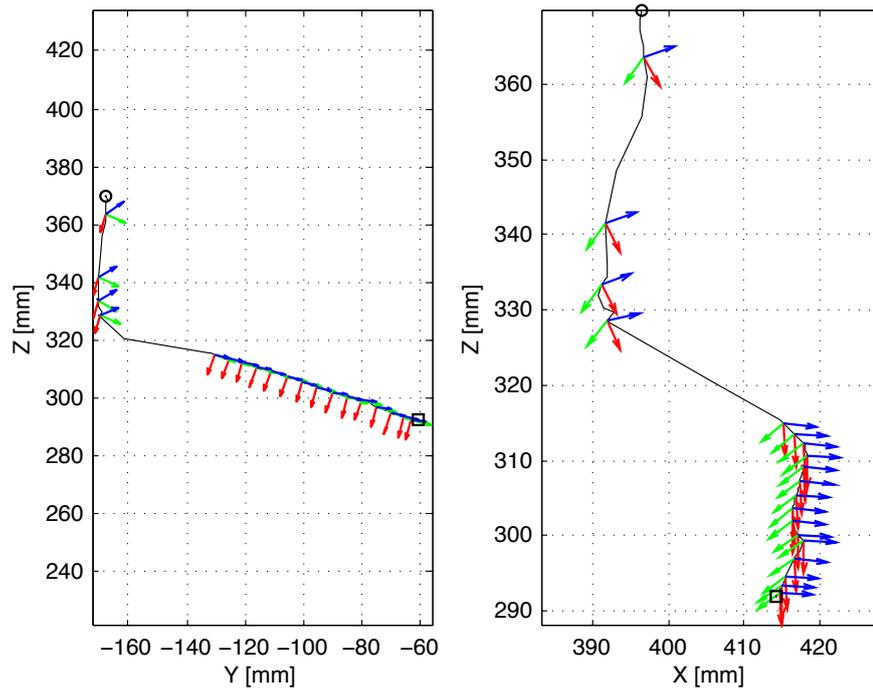


Fig. 6.10: Superficie piana: traiettoria di approccio al target, piano YZ e XZ

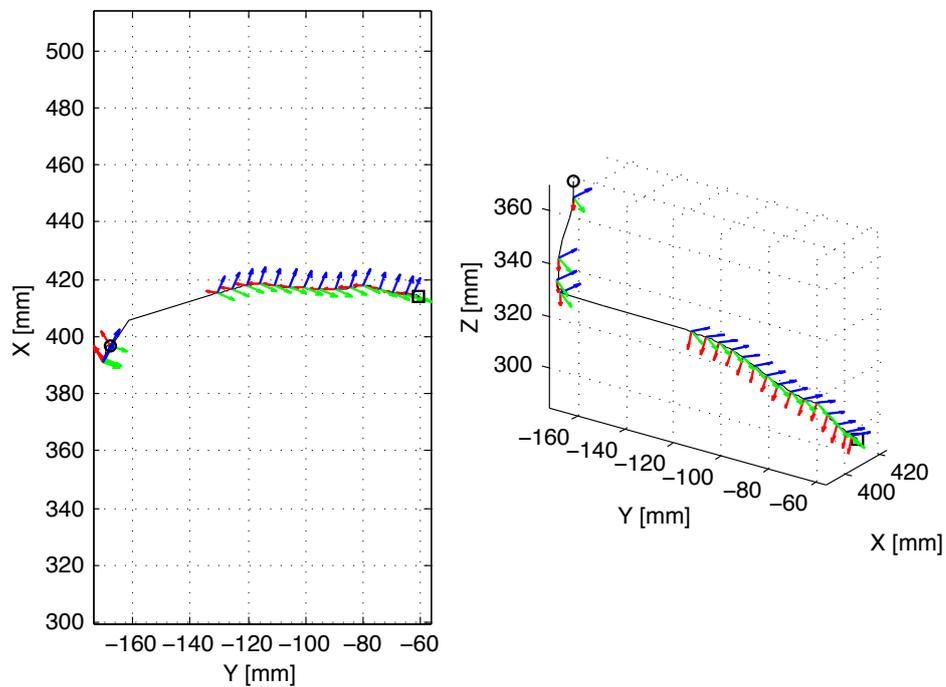


Fig. 6.11: Superficie piana: traiettoria di approccio al target, piano YX e rappresentazione $3D$

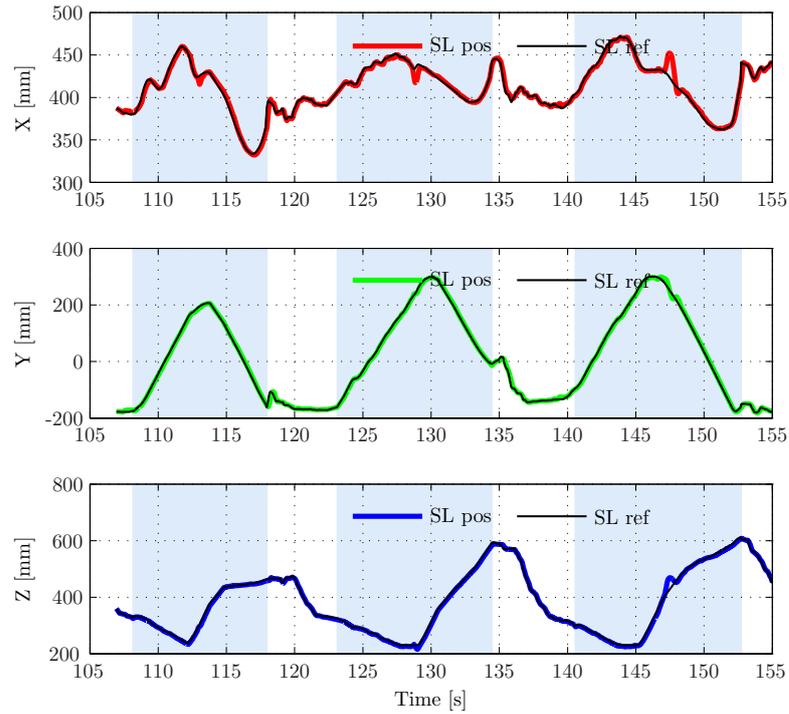


Fig. 6.12: Superficie piana: inseguimento dei riferimenti di posizione

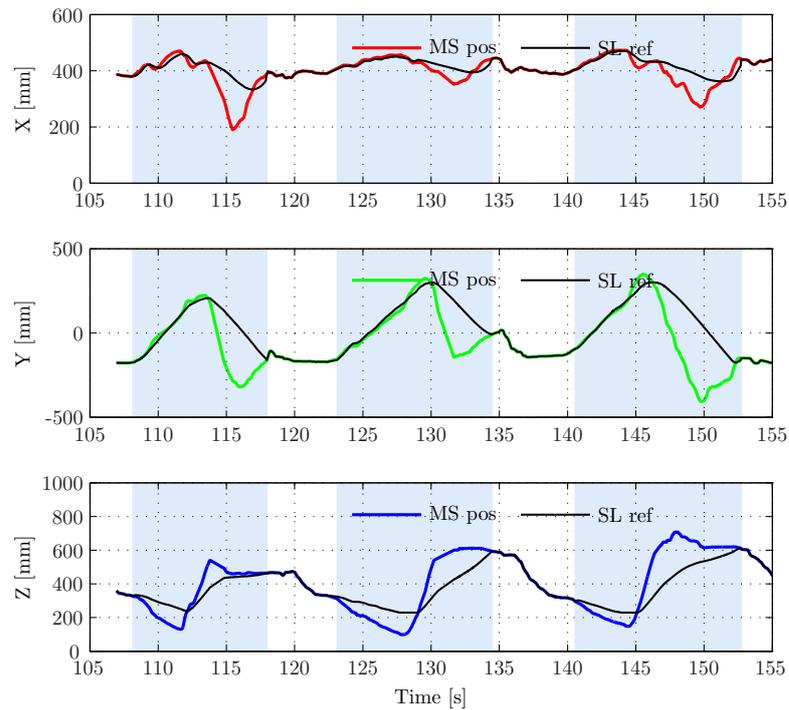


Fig. 6.13: Superficie piana: confronto fra riferimenti di posizione dei due dispositivi master e slave

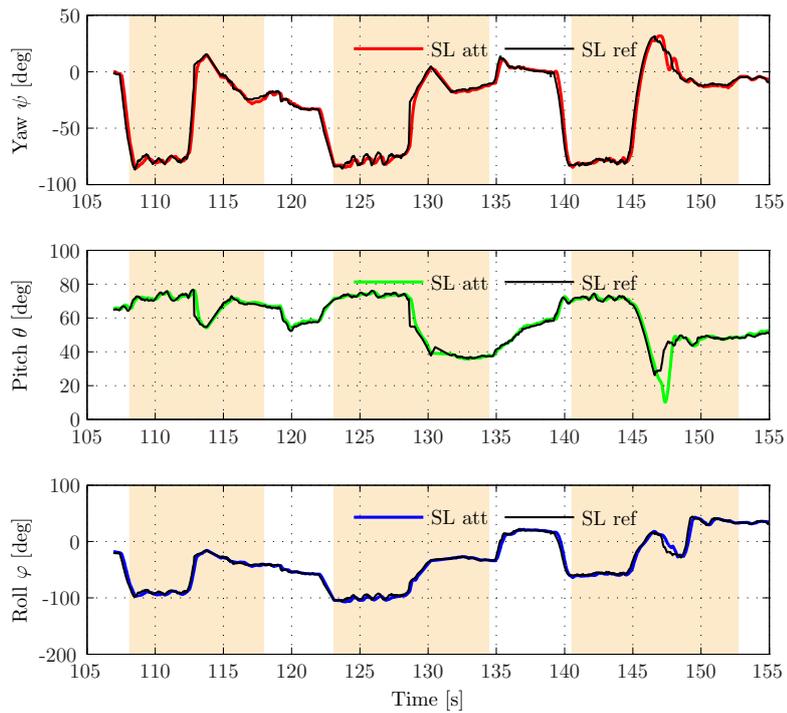


Fig. 6.14: Superficie piana: inseguimento dei riferimenti di orientazione

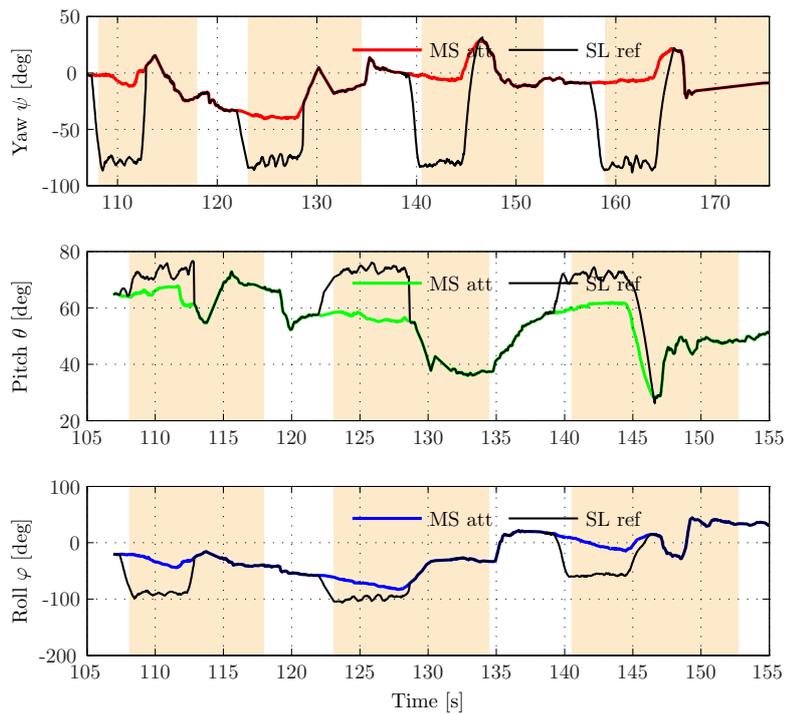


Fig. 6.15: Superficie piana: confronto fra riferimenti di orientazione dei due dispositivi master e slave

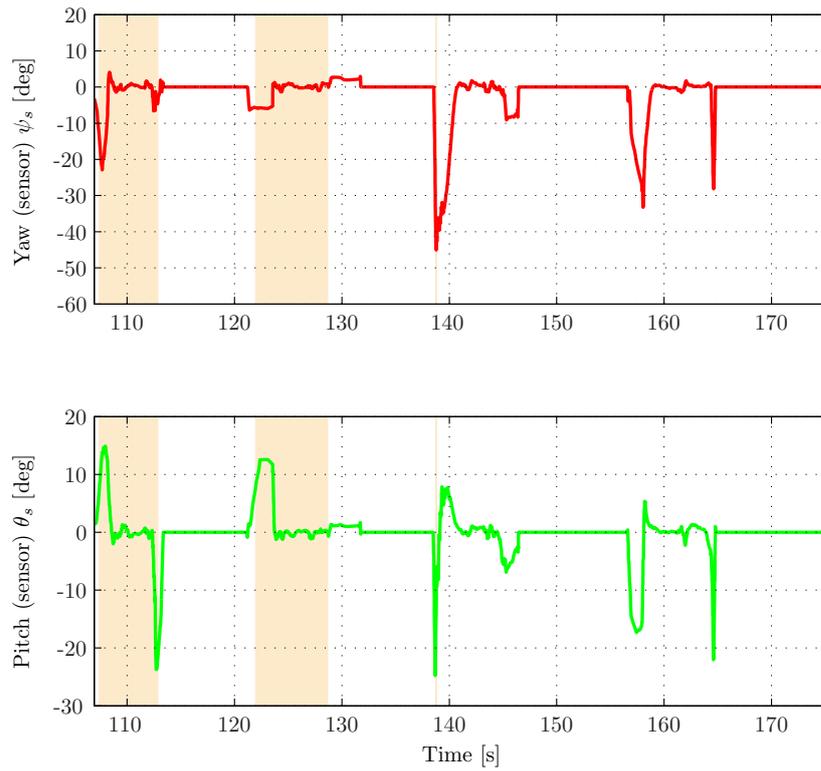


Fig. 6.16: Superficie piana: andamento degli angoli di yaw e pitch misurati dal sensore

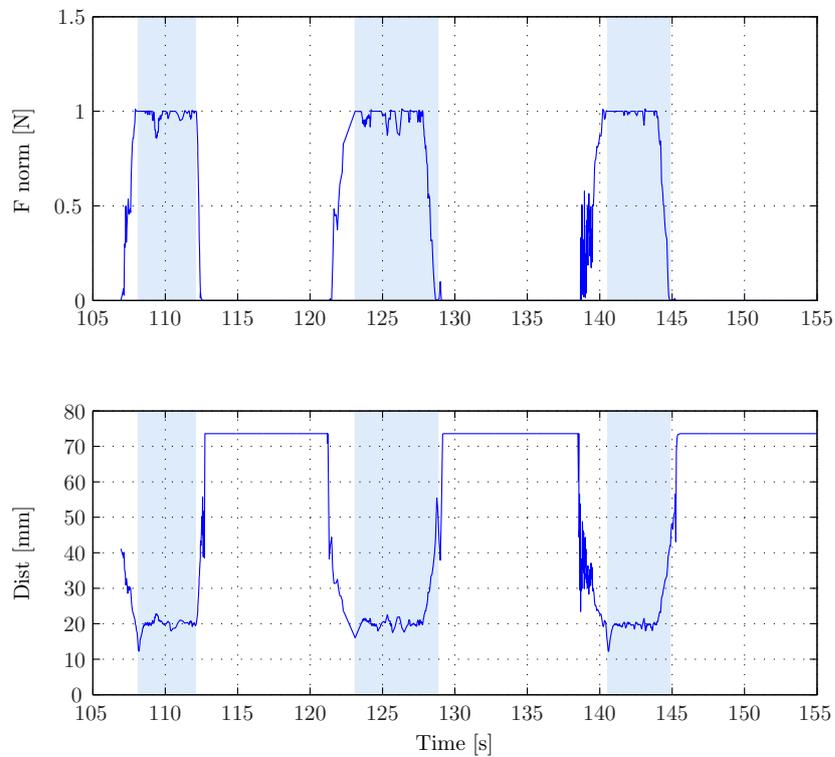


Fig. 6.17: Superficie piana: evoluzione temporale di forza di feedback e distanza media dalla superficie del target

6.3 Inseguimento di una superficie curva

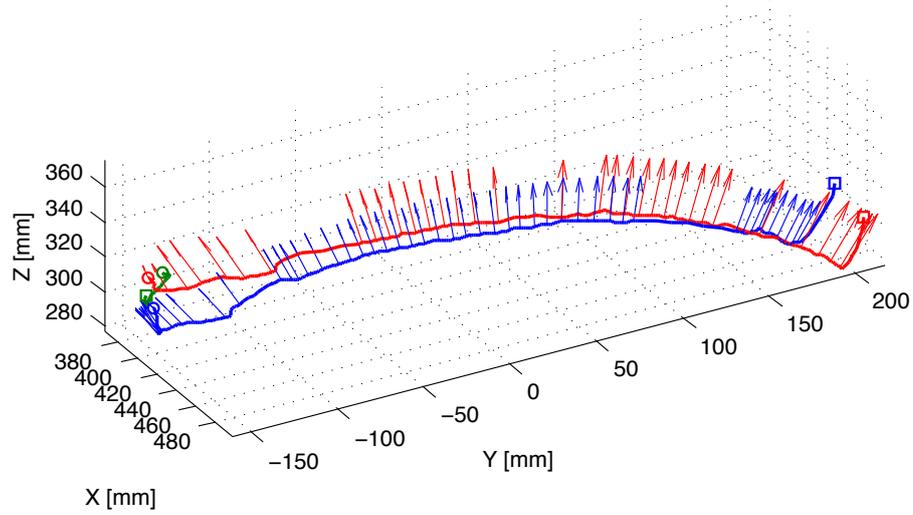


Fig. 6.18: Superficie curva: rappresentazione 3D delle traiettorie seguite

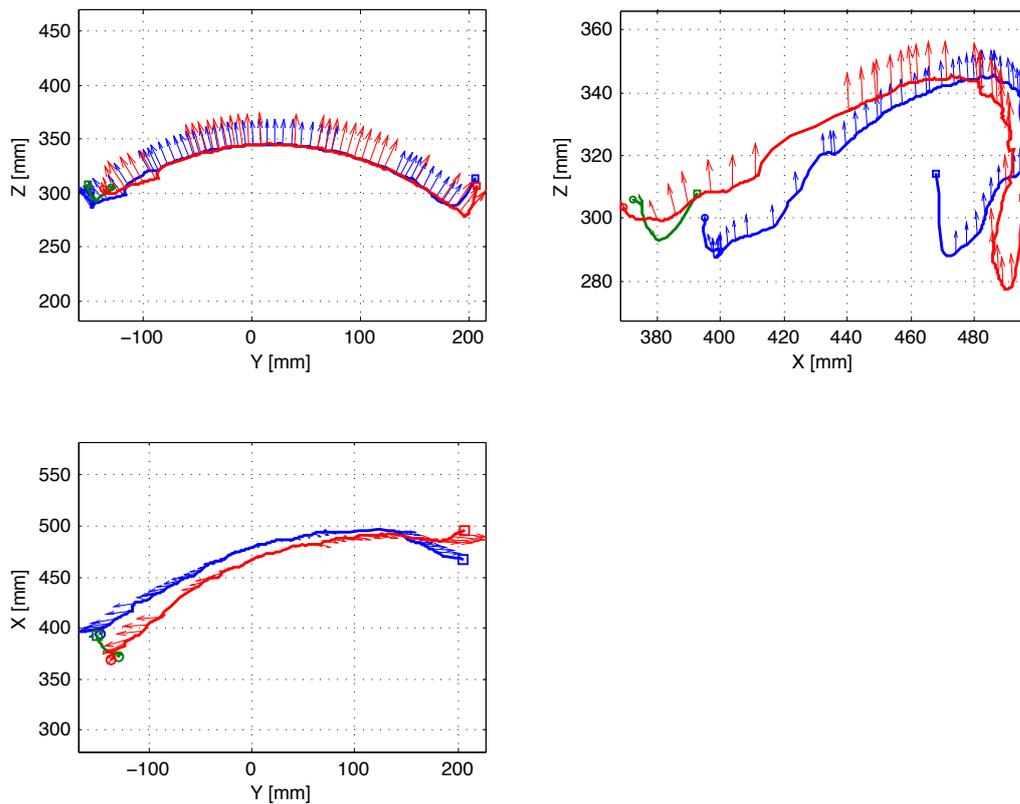
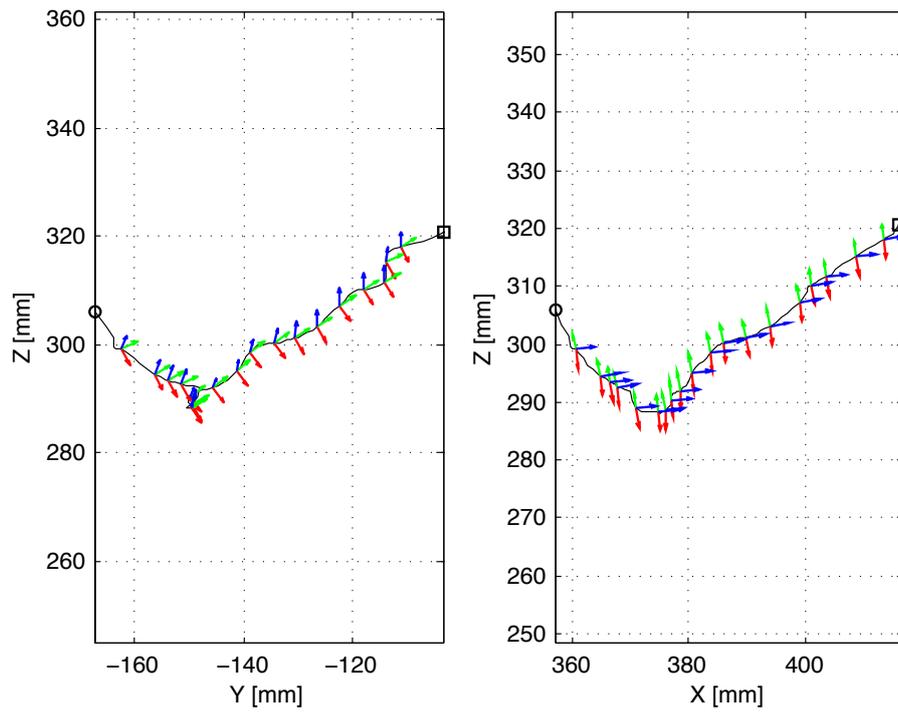
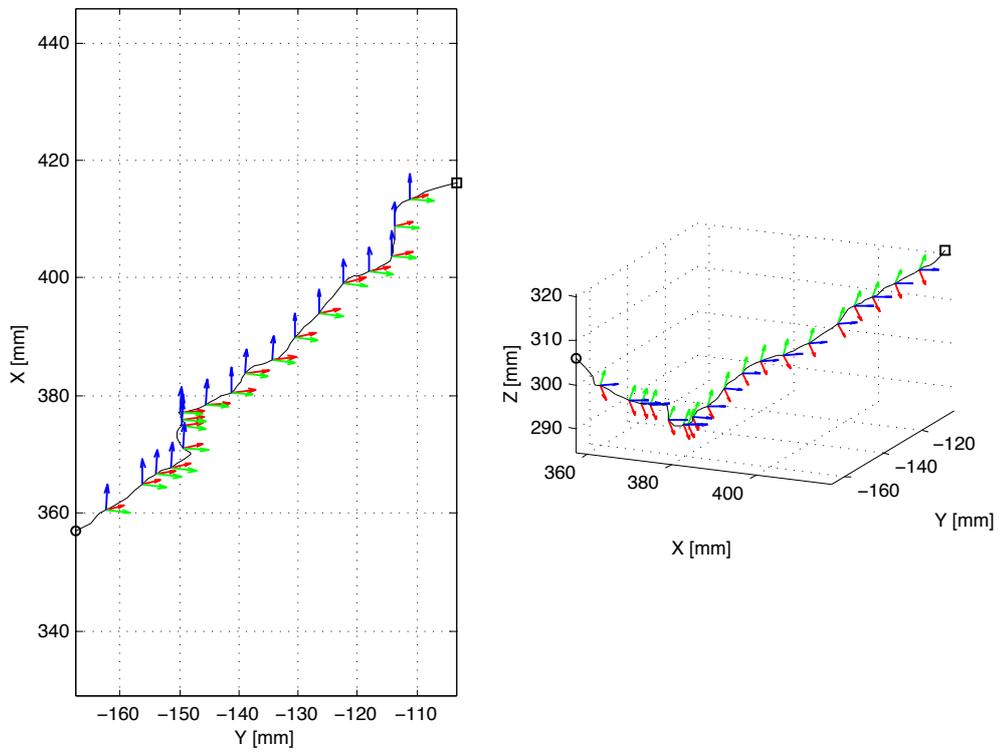


Fig. 6.19: Superficie curva: proiezioni ortogonali delle traiettorie seguite

Fig. 6.20: Superficie curva: traiettoria di approccio al target, piano YZ e XZ Fig. 6.21: Superficie curva: traiettoria di approccio al target, piano YX e rappresentazione $3D$

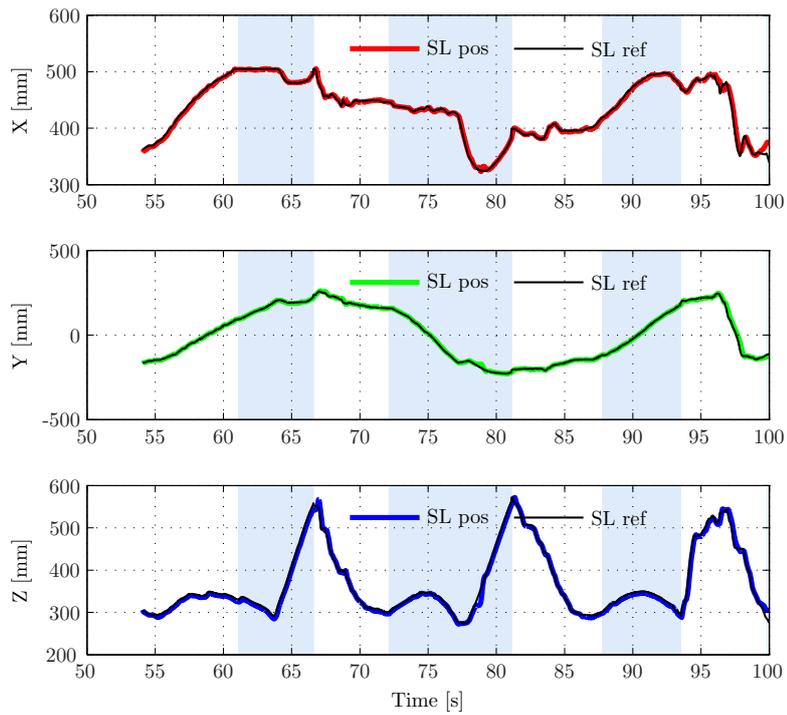


Fig. 6.22: Superficie curva: inseguimento dei riferimenti di posizione

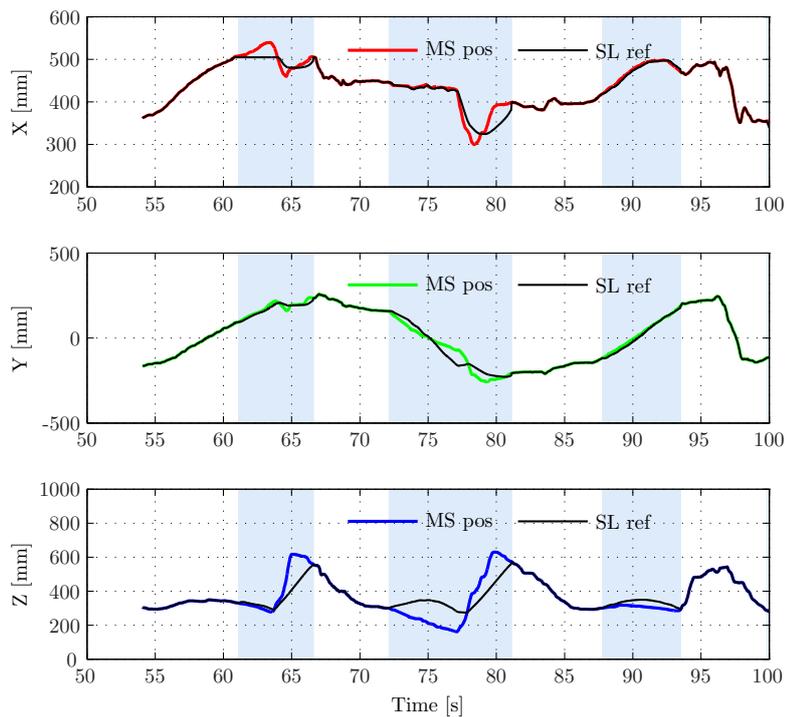


Fig. 6.23: Superficie curva: confronto fra riferimenti di posizione dei due dispositivi master e slave

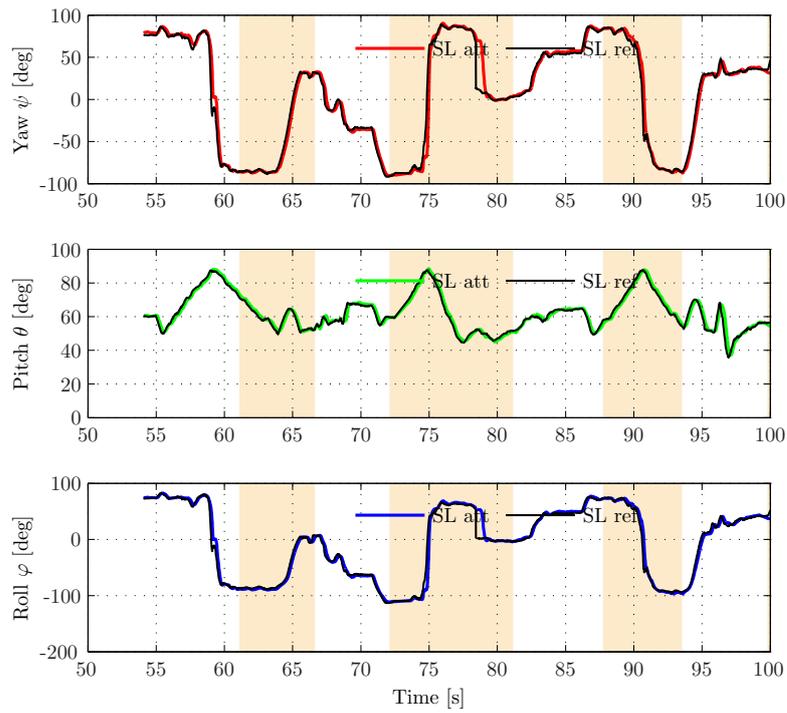


Fig. 6.24: Superficie curva: inseguimento dei riferimenti di orientazione

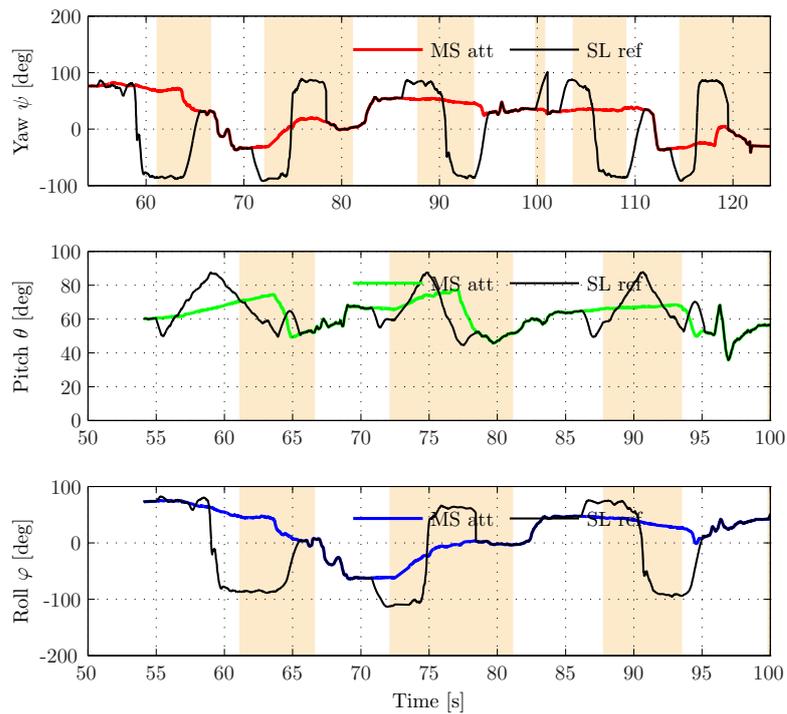


Fig. 6.25: Superficie curva: confronto fra riferimenti di orientazione dei due dispositivi master e slave

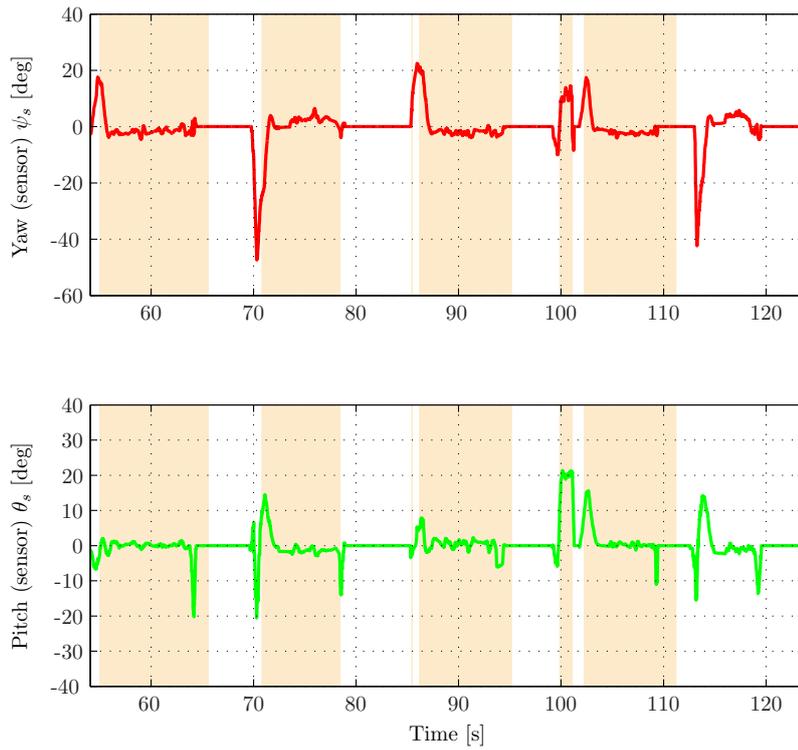


Fig. 6.26: Superficie curva: andamento degli angoli di yaw e pitch misurati dal sensore

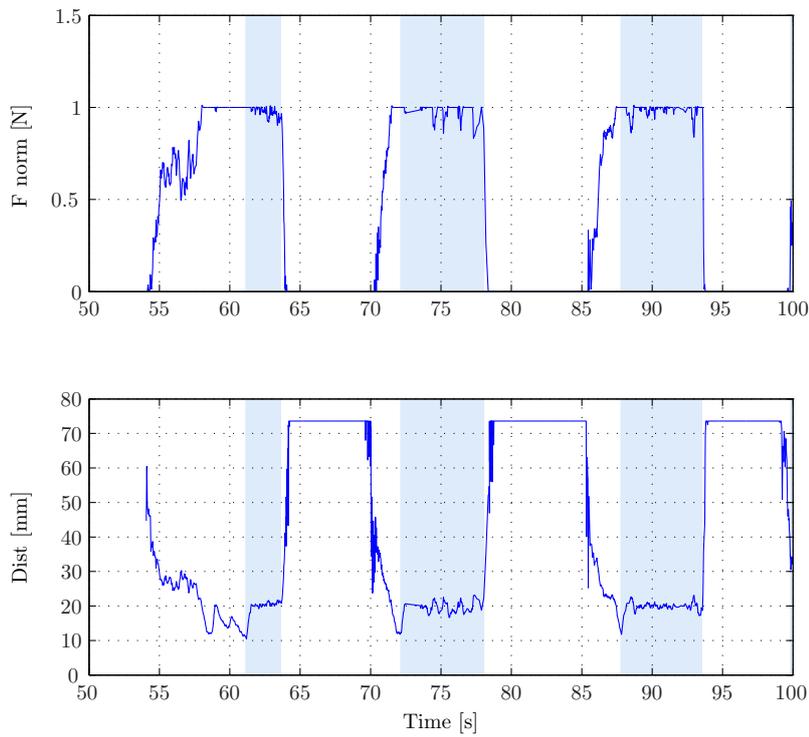


Fig. 6.27: Superficie curva: evoluzione temporale di forza di feedback e distanza media dalla superficie del target

Conclusioni

Durante lo svolgimento del progetto di tesi si sono sviluppati in particolare due sistemi di controllo, indipendenti fra loro ma simmetrici dal punto di vista della loro implementazione, che sono stati poi integrati nell'architettura bilaterale master/slave.

Il controllo automatico di assetto consente di adattare l'orientazione della sonda, in modo da mantenerla costantemente ortogonale rispetto al piano. Il controllo automatico di distanza mantiene invece la sonda a una prefissata distanza minima costante dal piano del target, in modo da evitare il contatto con il paziente e determinare una migliore acquisizione delle scansioni. Parallelamente a questi sistemi di controllo un feedback di forza viene inviato all'utente, per favorire la percezione aptica dell'ambiente con cui la sonda interagisce.

Dall'analisi dei grafici presentati nel capitolo 6 si nota l'effettiva capacità del sistema di inseguire i riferimenti, siano essi provenienti dal controllo, che dal master. Particolarmente significativi si rivelano essere i grafici che confrontano i riferimenti forniti dal master con quelli dello slave. Nel momento di attivazione delle procedure automatiche i due andamenti si discostano, rendendo conto della capacità del sistema di operare indipendentemente dal master.

I segnali del feedback di forza e dei valori degli angoli di pitch e yaw misurati dal sensore si rivelano essere ancora affetti da oscillazioni considerevoli, nonostante le azioni di filtraggio intraprese sulle misure di distanza.

Un limite che persiste tuttora nel sistema riguarda quindi la notevole fluttuazione delle misure di posizione fornite dal sensore di prossimità. L'azione di filtraggio consente tuttora di ottenere dei valori oscillanti attorno a una fascia di ampiezza sufficientemente ridotta, in modo da consentire a sua volta delle buone performance dei due controlli automatici. Si ritiene tuttavia che questo possa essere un aspetto migliorabile nel futuro, intervenendo sulle caratteristiche costruttive del sensore o analizzando con maggiore dettaglio le soluzioni migliori in termini di filtraggio e la loro ottimizzazione in funzione del sistema in analisi.

Uno dei primi possibili sviluppi futuri riguarda innanzitutto l'integrazione di una vera e propria sonda ecografica nel sistema, a cui segue l'implementazione di particolari procedure di scansione di immagini o l'individuazione e registrazione di superfici o punti di interesse dell'articolazione in esame attraverso la creazione di un reticolo.

Parallelamente a queste nuove funzionalità implementabili, si ritiene inoltre necessaria l'analisi delle norme che regolamentano l'immissione in commercio dei dispositivi medici, con particolare attenzione alle misure da intraprendere al fine di rendere l'apparecchio maggiormente sicuro in termini di incolumità del paziente sottoposto all'esame. Dovrebbero quindi essere predisposti senz'altro dei dispositivi di sicurezza indipendenti dal controllo del sistema e in grado di garantire il blocco totale dell'apparecchiatura in condizioni di pericolo per il personale e il paziente.

Si ritiene comunque di aver raggiunto risultati soddisfacenti per quanto riguarda l'architettura di controllo proposta e le funzionalità che essa è in grado di garantire. Il sistema telerobotico presentato in questo elaborato può costituire un valido punto di partenza per un ulteriore sviluppo del prodotto, con lo scopo di renderlo applicabile in campo medico.

Uno strumento di questo tipo consentirebbe infatti all'operatore di compiere agevolmente un esame economico e rapido, in grado tuttavia di fornire allo stesso tempo dei risultati attendibili per la diagnosi precoce di malattie reumatiche, fra cui l'artrite reumatoide.

Ringraziamenti

Giunta al termine di questo percorso scolastico voglio ringraziare di cuore i miei genitori e mio fratello Francesco per avermi mantenuta economicamente fino a questo momento, ma soprattutto per avermi sempre sostenuta moralmente.

Un grazie speciale a Davide, per aver camminato sempre al mio fianco in questi anni.

Ringrazio il professore Roberto Oboe per avermi dato la possibilità di sviluppare questo progetto, che mi ha permesso di ampliare le mie conoscenze teoriche e pratiche e acquisire conoscenze importanti per il mio futuro lavorativo. Grazie anche all'Ing. Riccardo Antonello, per i preziosi consigli, i suggerimenti, e la disponibilità nei miei confronti.

Grazie ai miei vicini di banco, in particolare Michele, Simone e Nicola, per essere stati dei compagni di corso sempre disponibili ad aiutare, ma soprattutto dei grandi amici.

Desidero inoltre ringraziare tutti i miei amici, i nonni e i miei parenti che hanno sempre creduto in me e senza i quali questo momento non sarebbe altrettanto speciale.

Bibliografia

- [1] Ferro, Nicola. Sviluppo di un sistema aptico per applicazioni ecografiche, Università di Padova (A.A. 2010-2011)
- [2] Riccardo Antonello, Omar A. Daud, Roberto Oboe, A telerobotic manipulation system for an immersed ultrasonic examination using haptic constraints,
- [3] A. Klauser and M. Schirmer, “Contrast-enhanced ultrasound in rheumatic joint diseases”, in *Contrast Media in Ultrasonography*, ser. Medical Radiology, E. Quaia, Ed. Springer Berlin Heidelberg, pp. 365–379
- [4] Oliver J. Sommer, et. al., *Rheumatoid Arthritis: A Practical Guide to State-of-the-Art Imaging, Image Interpretation, and Clinical Implications*, RadioGraphics 2005; 25:381–398
- [5] CEUS Padova, <http://www.ceuspadova.sistemacongressi.com>
- [6] R. E. Saad, et. al., *Proximity Sensing for Robotics*, Copyright 2000 CRC Press LLC, <http://www.engnetbase.com>
- [7] Adi Bonen, Ricardo E. Saad, et. al., A Novel Electrooptical Proximity Sensor for Robotics: Calibration and Active Sensing, *IEEE Transactions on Robotics and Automation*, Vol. 13, no. 3, June 1997
- [8] G. Petryk, M. Buehler, Robust Estimation of Pre-Contact Object Trajectories, Fifth Symposium on Robot Control (SYROCO), Nantes, France, Sept. 1997
- [9] K. Hashtrudi-Zaad and S. E. Salcudean, “Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators”, *Int. J. Robot. Res.*, Vol. 20, no. 6, pp. 419–445, 2001
- [10] Kevin B. Fite, John E. Speich, Michael Goldfarb, “Transparency and Stability Robustness in Two-Channel Bilateral Telemanipulation”, *Transactions of the ASME*, Vol. 123, September 2001
- [11] Dale A. Lawrence, Stability and Transparency in Bilateral Teleoperation, *IEEE Transactions on Robotics and Automation*. Vol. 9, no. 5. October 1993
- [12] Quanser, Quanser/CRS Model 465 Open Architecture Robot
- [13] Quanser, MultiQ-3 Programming manual
- [14] Sensable Technologies, <http://www.sensable.com>

Cinematica diretta e inversa del robot Thermo CRS A465

A.1 Matrici di trasformazione

Posizione e orientazione dell'end effector relative a un sistema di riferimento base sono generalmente descritte da una matrice di trasformazione 4x4.

La matrice A.1.1 consente di passare da un sistema di riferimento 'a' ad un sistema di riferimento 'b', mediante un opportuna matrice di rotazione e una traslazione. I termini da r_{11} a r_{33} descrivono infatti il cambio di orientazione, mentre i termini p_x , p_y e p_z descrivono la traslazione da compiere, ovvero il cambiamento di posizione.

$$T_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.1.1})$$

Una volta quindi determinati i sistemi di riferimento dei diversi link del robot, ciascuno descritto da quattro parametri DH, è possibile derivare la matrice di trasformazione che descrive la relazione fra la terna di riferimento $(i-1)$ -esima e la terna i -esima. La generica matrice di roto-traslazione è la A.1.2 (matrice DH) riportata qui di seguito:

$$T_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -\sin \alpha_{i-1} d_i \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & \cos \alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.1.2})$$

A partire da questo modello generale, conoscendo i parametri DH di ciascuna terna, è possibile derivare le matrici di trasformazione fra i vari sistemi di riferimento del robot, le quali vengono riportate di seguito:

$$T_1^0 = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2^1 = \begin{bmatrix} c2 & -s2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s2 & c2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3^2 = \begin{bmatrix} c3 & -s3 & 0 & a_2 \\ s3 & c3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} c4 & -s4 & 0 & 0 \\ 0 & 0 & -1 & d_4 \\ -s4 & -c4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5^4 = \begin{bmatrix} c5 & -s5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s5 & c5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_6^5 = \begin{bmatrix} c6 & -s6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s6 & -c6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In particolare con $c1 = \cos \theta_1$, $s1 = \sin \theta_1$, $s23 = \sin(\theta_2 + \theta_3)$ e così via.

Infine, la matrice A.1.3 descrive in modo completo posizione e orientazione dell'end effector rispetto alla base del robot e si ottiene mediante la moltiplicazione delle matrici intermedie appena ricavate qui sopra.

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \quad (\text{A.1.3})$$

A.2 Cinematica diretta

Date le misure degli angoli dei sei giunti, le equazioni della cinematica diretta sono in grado di restituire la posizione e l'orientazione dell'end effector. Si consideri la matrice A.1.3 espressa nella forma A.1.1. I valori di ciascun termine di tale matrice saranno dati da:

$$\begin{aligned}
r_{11} &= c1(c23(c4c5c6 - s4s6) - s23s5c6) - s1(s4c5c6 + c4s6) \\
r_{21} &= s1(c23(c4c5c6 - s4s6) - s23s5c6) + c4(s4c5c6 + c4s6) \\
r_{31} &= s23(c4c5c6 - s4s6) + c23s5c6 \\
r_{12} &= c1(-c23(c4c5s6 + s4c6) + s23s5s6) - s1(-s4c5s6 + c4c6) \\
r_{22} &= s1(-c23(c4c5s6 + s4c6) + s23s5s6) + c1(-s4c5s6 + c4c6) \\
r_{32} &= -s23(c4c5s6 + s4c6) - c23s5s6 \\
r_{13} &= -c1(c23c4s5 + s23c5) + s1(s4s5) \\
r_{23} &= -s1(c23c4s5 + s23c5) - c1(s4s5) \\
r_{33} &= -s23c4s5 + c23c5 \\
p_x &= c1(-d4 * s23 + a2 * c2) \\
p_y &= s1(-d4 * s23 + a2 * c2) \\
p_z &= d4 * c23 + a2 * s2
\end{aligned}$$

Dove $d4$ e $a2$ rappresentano rispettivamente le lunghezze dei link 2 e 3, secondo la notazione DH.

In questo modo sarà sufficiente moltiplicare le coordinate della base per la matrice T_6^0 così ottenuta, per ricavare la posizione dell'end effector rispetto alla terna base.

A.3 Cinematica inversa

Date le coordinate di posizione ed orientazione dell'end effector rispetto alla terna base, la cinematica inversa consente di ottenere la misura degli angoli dei sei giunti del robot. Mentre applicando le equazioni della cinematica diretta si ottiene sempre una soluzione univoca al problema, ciò non accade per la cinematica inversa.

Per un manipolatore ridondante come è in effetti il Thermo A465, ci possono essere fino a otto soluzioni valide per ogni data posizione e orientazione. L'algoritmo calcolerà quindi le otto soluzioni, per poi scegliere la migliore, in base alla configurazione attuale del robot.

A.3.1 Soluzione del giunto 1

Dato che vale $T_6^0 = T_1^0 T_6^1$, segue che $T_1^{0-1} T_6^0 = T_6^1$ o, più semplicemente, $T_0^1 T_6^0 = T_6^1$. Se si estrae il termine $(2, 4)$ da quest'ultima equazione, si ottiene $-s1p_x + c1p_y = 0$, da cui si ricavano infine le due soluzioni per il primo giunto:

$$\theta_{1F} = \text{atan2}(p_y, p_x) \quad (\text{A.3.1})$$

$$\theta_{1B} = \theta_{1F} + \pi \quad (\text{A.3.2})$$

Il pedice F indica la '*forward solution*', mentre il pedice B sta per '*backward solution*'.

A.3.2 Soluzione del giunto 3

Allo stesso modo in cui si è risolto il giunto 1 è possibile ricorrere all'equazione matriciale $T_0^1 T_6^0 = T_6^1$ per trovare la soluzione per il giunto 3. In particolare si combinano i seguenti elementi:

$$c1p_x + s1p_y = -s23d4 + a2c2 \quad (\text{A.3.3})$$

$$p_z = c23d4 + s2a2 \quad (\text{A.3.4})$$

$$-s1p_x + c1p_y = 0 \quad (\text{A.3.5})$$

Con semplici passaggi algebrici, e tenendo conto delle relazioni fondamentali della trigonometria, si ricava:

$$\theta_3 = \text{atan2}(2d4a2c3, \pm \sqrt{4d4^2a2^2 - (2d4a2c3)^2}) \quad (\text{A.3.6})$$

La [A.3.6](#), in base alle due soluzioni relative al giunto 1 trovate prima in [A.3.1](#) e [A.3.2](#) restituisce a sua volta due possibili risultati:

$$\theta_{3BUFD} = \pi/2 + \text{atan2}(-\sqrt{4d4^2a2^2 - (2d4a2c3)^2}, 2d4a2c3) \quad (\text{A.3.7})$$

$$\theta_{3FUBD} = \pi/2 - \text{atan2}(-\sqrt{4d4^2a2^2 - (2d4a2c3)^2}, 2d4a2c3) \quad (\text{A.3.8})$$

Dove il pedice *BUFD* sta per '*backward-up/forward-down*', mentre *FUBD* sta per '*forward-up/backward down*'.

A.3.3 Soluzione del giunto 2

Per risolvere il giunto 2 si considera invece l'equazione matriciale $T_0^3 T_6^0 = T_6^3$, da cui si estraggono i seguenti elementi:

$$c1c23p_x + s1c23p_y + s23p_z(-a2c3) = 0 \quad (\text{A.3.9})$$

$$-c1s23p_x - s1s23p_y + c23p_z(a2s3) = 0 \quad (\text{A.3.10})$$

$$c23(c1p_x + s1p_y) + s23p_z - a2c3 = 0 \quad (\text{A.3.11})$$

$$s23(-c1p_x - s1p_y) + c23p_z + a2s3 = d4 \quad (\text{A.3.12})$$

Combinando insieme le [A.3.9](#), [A.3.10](#), [A.3.11](#), [A.3.12](#) si ricavano dapprima i valori di $s23$ e $c23$, e successivamente:

$$\theta_2 = \text{atan2}(s23, c23) - \theta_3 \quad (\text{A.3.13})$$

A.3.4 Soluzione del giunto 4

Per trovare la soluzione relativa a tale giunto faremo riferimento alla medesima matrice utilizzata per ricavare la soluzione del giunto 2, ovvero $T_0^3 T_6^0 = T_6^3$. Da questa si estraggono i seguenti termini:

$$c1c23r_{13} + s1c23r_{23} + s23r_{33} = -c4c5 \quad (\text{A.3.14})$$

$$s1r_{13} - c1r_{23} = s4s5 \quad (\text{A.3.15})$$

Risolvendo le [A.3.14](#) e [A.3.15](#) per $s4$ e $c4$ si ottengono le due soluzioni del giunto 4:

$$\theta_{4F} = \text{atan2}[(-s1r_{13} + c1r_{23}), (c1c23r_{13} + s1c23r_{23} + s23r_{33})] \quad (\text{A.3.16})$$

$$\theta_{4N} = \theta_{4F} + \pi \quad (\text{A.3.17})$$

Dove il pedice F sta per polso '*flipped*', mentre N sta per polso '*not-flipped*'.

A.3.5 Soluzione del giunto 5

A partire dall'equazione matriciale $T_0^4 T_6^0 = T_6^4$ si ricavano le uguaglianze:

$$r_{13}(-s1s4 + c1c23c4) + c23(c1s4 + s1c23c4) + r_{33}(s23c4) = -s5 \quad (\text{A.3.18})$$

$$-c1s23r_{13} - s1s23r_{23} + c23r_{33} = c5 \quad (\text{A.3.19})$$

Da cui si arriva alla soluzione a polso '*not-flipped*':

$$\theta_{5N} = \text{atan2}(s5, c5) \quad (\text{A.3.20})$$

e alla soluzione '*flipped*':

$$\theta_{5F} = -\theta_{5N} \quad (\text{A.3.21})$$

A.3.6 Soluzione del giunto 6

Per ricavare, infine, la soluzione relativa al giunto 6, si parte dall'equazione matriciale $T_0^5 T_6^0 = T_6^5$, per estrarre poi gli elementi:

$$\begin{aligned} r_{12}(-c5s1s4 + c5c1c23c4 - s5c1s23) + \\ r_{22}(c5c1s4 + c5s1c23c4 - s5s1s23) + r_{32}(c5s23c4 + s5c23) = -s6 \end{aligned} \quad (\text{A.3.22})$$

$$r_{12}(-c1c23s4 - s1c4) + r_{22}(-s1c23s4 + c1c4) - s23s4r_{32} = c6 \quad (\text{A.3.23})$$

Le [A.3.22](#) e [A.3.23](#) restituiscono direttamente le soluzioni polso '*not-flipped*':

$$\theta_{6N} = \text{atan2}(s6, c6) \quad (\text{A.3.24})$$

e la soluzione polso '*flipped*':

$$\theta_{6F} = \theta_{6N} + \pi \quad (\text{A.3.25})$$

Parametrizzazione della matrice di rotazione R

B.1 Codifica da notazione asse-angolo a matrice di rotazione

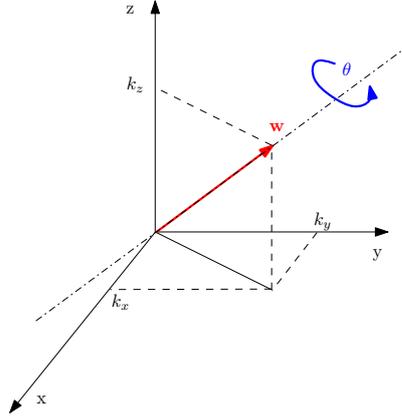


Fig. B.1: Definizione del versore \mathbf{w} e angolo θ

Secondo la notazione asse-angolo (axis-angles) l'orientazione dell'end effector si può esprimere attraverso un vettore di dimensione quattro:

$$\mathbf{w} = [k_x, k_y, k_z, \theta] \quad (\text{B.1.1})$$

I primi tre elementi del vettore indicano le componenti cartesiane di un versore, mentre l'ultimo elemento indica il valore dell'angolo di rotazione attorno all'asse identificato tale versore.

La matrice di rotazione si può ricavare a partire dagli elementi del vettore [B.1.1](#), come segue:

$$R = \begin{bmatrix} k_x^2 v \theta + c \theta & k_x k_y v \theta - k_z s \theta & k_x k_z v \theta - k_y s \theta \\ k_x k_y v \theta + k_z s \theta & k_y^2 v \theta - c \theta & k_y k_z v \theta - k_x s \theta \\ k_x k_z v \theta + k_y s \theta & k_y k_z v \theta + k_x s \theta & k_z^2 v \theta + c \theta \end{bmatrix} \quad (\text{B.1.2})$$

dove $c \theta = \cos \theta$, $s \theta = \sin \theta$ e $v \theta = 1 - \cos \theta$.

B.2 Codifica da matrice di rotazione ad angoli di yaw - pitch - roll

Si denotino con φ , θ e ψ rispettivamente gli angoli di yaw, pitch e roll, con il verso indicato in [Fig.B.2](#).

Assegnata una matrice di rotazione come [A.1.1](#) definita nell'appendice [A](#), essa può essere rappresentata equivalentemente come la composizione di tre matrici di rotazione.

Si determina, quindi, la terna (φ, θ, ψ) tale che valga:

$$R_z(\varphi)R_y(\theta)R_x(\psi) = \begin{bmatrix} c_\varphi c_\theta & -s_\varphi c_\psi + c_\varphi s_\theta s_\psi & s_\varphi s_\psi + c_\varphi s_\theta c_\psi \\ s_\varphi c_\theta & c_\varphi c_\psi + s_\varphi s_\theta s_\psi & -c_\varphi s_\psi + s_\varphi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix} \quad (\text{B.2.1})$$

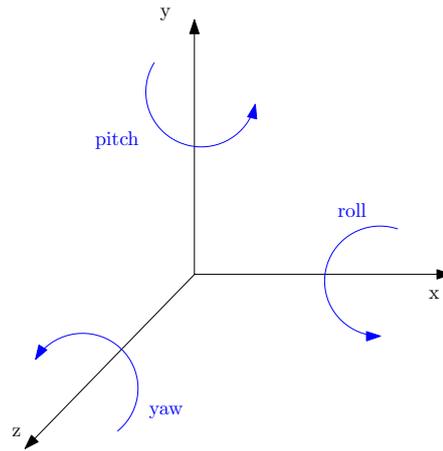


Fig. B.2: Definizione degli angoli di roll, pitch e yaw, con i relativi versi di rotazione

Se $r_{11}^2 + r_{21}^2 = \cos^2\theta \neq 0$, si possono ottenere due differenti soluzioni, a seconda del valore dell'angolo θ :

- Se $\theta \in (-\pi/2, \pi/2)$, ($\cos\theta > 0$), allora:

$$\begin{cases} \varphi = \text{atan2}(r_{21}, r_{11}) \\ \theta = \text{atan2}(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \\ \psi = \text{atan2}(r_{32}, r_{33}) \end{cases} \quad (\text{B.2.2})$$

- Se $\theta \in (\pi/2, 3\pi/2)$, ($\cos\theta < 0$), allora:

$$\begin{cases} \varphi = \text{atan2}(-r_{21}, -r_{11}) \\ \theta = \text{atan2}(-r_{31}, -\sqrt{r_{32}^2 + r_{33}^2}) \\ \psi = \text{atan2}(-r_{32}, -r_{33}) \end{cases} \quad (\text{B.2.3})$$

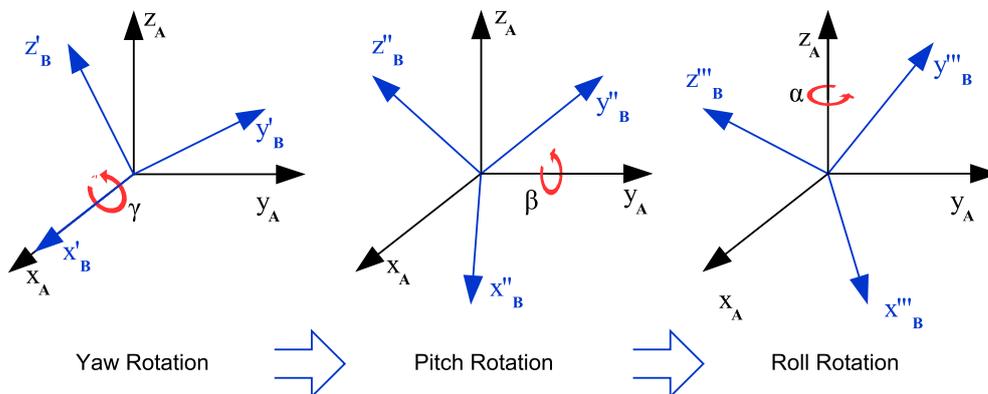


Fig. B.3: Composizione delle rotazioni degli angoli di yaw-pitch-roll

Interfacciamento dei dispositivi: listati di codice

C.1 Dispositivo Master Phantom Omni

C.1.1 Codifica di orientazione

```
1 % the input of the function is vector u
2 % the outputs yaw, pitch and roll of Phantom Device
3 function [yaw, pitch, roll] = fcn(u)
4
5 % assignment of u-vector's elements to local variables
6 kx = u(3); ky = u(1); kz = u(2); theta = u(4);
7
8 % defines the rotation matrix w.r.t. the rgb].133,.545,.133...
  axis-angles notation
9 ctheta = cos(theta); stheta = sin(theta); vtheta = 1-cos(theta);
10
11 Rt = [kx*kx*vtheta+ctheta kx*ky*vtheta-kz*stheta kx*kz*vtheta+ky*stheta;
12       kx*ky*vtheta+kz*stheta ky*ky*vtheta+ctheta ky*kz*vtheta-kx*stheta;
13       kx*kz*vtheta-ky*stheta ky*kz*vtheta+kx*stheta kz*kz*vtheta+ctheta];
14
15 t1 = sqrt( Rt(1,1)^2 + Rt(2,1)^2 );
16 pitch = atan2(-Rt(3,1), t1);
17
18 if abs( abs(pitch) - pi/2 ) < 0.001,
19     yaw = 0;
20     if pitch > 0,
21         roll = atan2(Rt(1,2), Rt(2,2));
22     else
23         roll = -atan2(Rt(1,2), Rt(2,2));
24     end
25 else
26     yaw = atan2(Rt(2,1), Rt(1,1));
27     roll = atan2(Rt(3,2), Rt(3,3));
28 end;
29
30 pitch = -pitch;
31 yaw = -yaw;
```

C.1.2 Codifica di posizione

```
1 % the input of the function is vector u
2 % the outputs are the three position coords x, y, z
3 function [x,y,z]= fcn(u)
4
5 base_CRS = 330.2;
6 link1_CRS = 304.8;
7 link2_CRS = 330.2 + 76.2 + 26.2;
8
9 % map PHANToM encoder angles to CRS (joint) rgb].133,.545,.133...
  kinematic space
10 q1 = -u(1);
11 q2 = u(2);
```

```

12 q3 = u(3)-u(2)+pi;
13
14 q = [q1; q2; q3];
15
16 for i=1:3,
17     newangle = q(i);
18
19     while newangle > pi,
20         newangle = newangle - 2*pi;
21     end
22
23     while newangle ≤ -pi,
24         newangle = newangle + 2*pi;
25     end
26
27     q(i) = newangle;
28 end
29
30 if q(2) ≤ -pi/2,
31     q(2) = q(2) + 2*pi;
32 end
33
34 if q(3) > pi/2,
35     q(3) = q(3) - 2*pi;
36 end
37
38 % pre-calculate all trigs
39 s1=sin(q(1));
40 c1=cos(q(1));
41
42 s2=sin(q(2));
43 c2=cos(q(2));
44
45 s23=sin(q(2)+q(3));
46 c23=cos(q(2)+q(3));
47
48 % compute cartesian coords of end-effector
49 % (i.e. T30(1:3,4), with T30 the transformation rgb].133,.545,.133...
50 % from frame 3 to frame 0 )
51
52 k7 = -link2_CRS*s23 + link1_CRS*c2;
53
54 x = c1*k7;
55 y = s1*k7;
56 z = (link2_CRS*c23 + link1_CRS*s2) + base_CRS;

```

C.2 Sensore ottico di prossimità: Stima della distanza media e dell'orientazione del target

```

1 function [PRXS_dist, PRXS_YP] = get_plane_posture(u, CRS_YPR)
2
3 % get avg dist
4 avg_dist = mean(u);
5
6 % get PRXS_YP
7 PRXS_YP = zeros(2,1);
8 PRXS_YP(1) = atan( (u(2)-u(1))/sqrt(3)/5 );
9 PRXS_YP(2) = atan( (u(3)-0.5*u(1)-0.5*u(2))/(3/2*5) );
10

```

```

11 % get unit vector orthogonal to the plane
12 np = get_np(CRS_YPR, PRXS_YP);
13
14 % get EE approaching direction (unit vector)
15 nEE = get_nEE(CRS_YPR);
16
17 % get true distance of robot EE from the plane
18 PRXS_dist = avg_dist * abs(nEE'*np);
19
20 end
21
22 %%%%
23
24 function nEE = get_nEE(CRS_YPR)
25
26     Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
27
28     nEE = Rbe*[1; 0; 0];
29
30 end
31
32 %%%%
33
34 function np = get_np(CRS_YPR, PRXS_YP)
35
36     Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
37     Rep = get_DCM(PRXS_YP(1), PRXS_YP(2), 0);
38     Rbp = Rbe*Rep;
39
40     np = Rbp*[-1; 0; 0];
41
42 end
43
44 %%%%
45
46 function DCM = get_DCM(Y, P, R)
47
48     cy = cos(Y);    sy = sin(Y);
49     cp = cos(P);    sp = sin(P);
50     cr = cos(R);    sr = sin(R);
51
52     DCM = zeros(3,3);
53
54     DCM(1,1) = cy*cp;
55     DCM(2,1) = sy*cp;
56     DCM(3,1) = -sp;
57
58     DCM(1,2) = cy*sp*sr - sy*cr;
59     DCM(2,2) = sy*sp*sr + cy*cr;
60     DCM(3,2) = cp*sr;
61
62     DCM(1,3) = cy*sp*cr + sy*sr;
63     DCM(2,3) = sy*sp*cr - cy*sr;
64     DCM(3,3) = cp*cr;
65
66 end

```


Controlli automatici: listati di codice

D.1 Controllo automatico dell'orientazione

```
1 function [CRS_YPR_ref, auto_ctrl] = rgb].133,.545,.133...
   CRS_YPR_controller(CRS_YPR, PRXS_dist, PRXS_YP, rgb].133,.545,.133...
   is_PRXS_on, PHANToM_YPR_ref, k)
2
3 persistent is_auto_ctrl_on;
4 persistent prev_CRS_YPR_ref;
5
6
7 %% Initialize persistent vars
8 if isempty(is_auto_ctrl_on),
9     is_auto_ctrl_on = 0;
10 end;
11
12 if isempty(prev_CRS_YPR_ref),
13     prev_CRS_YPR_ref = PHANToM_YPR_ref;
14 end;
15
16
17 %% Check auto_ctrl enable
18 is_auto_ctrl_enabled = auto_ctrl_enable(PRXS_dist, is_PRXS_on);
19
20
21 %% Determine controller output
22 CRS_DeltaYPR = [0; 0; 0];
23
24 if is_auto_ctrl_on,
25     if is_auto_ctrl_enabled,
26         CRS_DeltaYPR = [PRXS_YP(1); PRXS_YP(2); 0];
27     else
28         Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
29         Rbph = get_DCM(PHANToM_YPR_ref(1), rgb].133,.545,.133...
            PHANToM_YPR_ref(2), PHANToM_YPR_ref(3));
30         Reph = inv(Rbe)*Rbph;
31         [DY, DP, DR] = get_YPR(Reph);
32         CRS_DeltaYPR = [DY; DP; DR];
33     end;
34     CRS_YPR_ref = rgb].133,.545,.133...
        get_CRS_YPR_auto_ctrl_output(CRS_YPR, rgb].133,.545,.133...
        CRS_DeltaYPR, prev_CRS_YPR_ref);
35 else
36     CRS_YPR_ref = PHANToM_YPR_ref;
37 end;
38
39 auto_ctrl = is_auto_ctrl_on;
40
41
42 %% Determine next controller state
43 is_aligned = 0;
44
45 if is_auto_ctrl_on,
46     aligned_th = 2*pi/180;
47     if (~is_auto_ctrl_enabled) && all(CRS_DeltaYPR < aligned_th),
```

```

48     is_auto_ctrl_on = 0;
49     end;
50 else
51     if is_auto_ctrl_enabled,
52         is_auto_ctrl_on = 1;
53     end;
54 end;
55
56
57 %% Update ref integrator state
58 prev_CRS_YPR_ref = CRS_YPR_ref;
59
60 end
61
62 %%%
63
64 function CRS_YPR_ref = rgb].133,.545,.133...
        get_CRS_YPR_auto_ctrl_output(CRS_YPR, rgb].133,.545,.133...
        CRS_DeltaYPR, prev_CRS_YPR_ref);
65
66 eY = sat(CRS_DeltaYPR(1), -5*pi/180, 5*pi/180)/(5*pi/180);
67 eP = sat(CRS_DeltaYPR(2), -5*pi/180, 5*pi/180)/(5*pi/180);
68 eR = sat(CRS_DeltaYPR(3), -5*pi/180, 5*pi/180)/(5*pi/180);
69
70 k = 0.0005;
71 Rbe = get_DCM(prev_CRS_YPR_ref(1), rgb].133,.545,.133...
        prev_CRS_YPR_ref(2), prev_CRS_YPR_ref(3));
72 Rep = get_DCM(k*eY, k*eP, k*eR);
73 Rbp = Rbe*Rep;
74
75 CRS_YPR_ref = zeros(3,1);
76 [CRS_YPR_ref(1), CRS_YPR_ref(2), CRS_YPR_ref(3)] = get_YPR(Rbp);
77
78 CRS_Y_ref_sat = sat(CRS_YPR_ref(1), -pi, pi);
79 CRS_P_ref_sat = sat(CRS_YPR_ref(2), -pi, pi);
80 CRS_R_ref_sat = sat(CRS_YPR_ref(3), -pi, pi);
81 CRS_YPR_ref = [CRS_Y_ref_sat; CRS_P_ref_sat; CRS_R_ref_sat];
82
83 end
84
85 %%%
86
87 function is_auto_ctrl_enabled = rgb].133,.545,.133...
        auto_ctrl_enable(PRXS_dist, is_PRXS_on)
88
89 persistent y_prev;
90
91 %% Initialize persistent vars
92 if isempty(y_prev),
93     y_prev = 0;
94 end;
95
96 %% check distance from plane
97 r = 35;
98 e = r - PRXS_dist;
99
100 eL = -5;    % hysteresis threshold low
101 eH = 5;    % hysteresis threshold high
102 if (e > eH) && (y_prev == 0),
103     y = 1;
104 elseif (e < eL) && (y_prev == 1),
105     y = 0;
106 else

```

```
107     y = y_prev;
108 end;
109
110 %% update output
111 is_auto_ctrl_enabled = is_PRXS.on && y;
112
113 % update prev vars
114 y_prev = y;
115
116 end
117
118 %%%
119
120 function DCM = get_DCM(Y, P, R)
121
122 cy = cos(Y);    sy = sin(Y);
123 cp = cos(P);    sp = sin(P);
124 cr = cos(R);    sr = sin(R);
125
126 DCM = zeros(3,3);
127
128 DCM(1,1) = cy*cp;
129 DCM(2,1) = sy*cp;
130 DCM(3,1) = -sp;
131
132 DCM(1,2) = cy*sp*sr - sy*cr;
133 DCM(2,2) = sy*sp*sr + cy*cr;
134 DCM(3,2) = cp*sr;
135
136 DCM(1,3) = cy*sp*cr + sy*sr;
137 DCM(2,3) = sy*sp*cr - cy*sr;
138 DCM(3,3) = cp*cr;
139
140 end
141
142 %%%
143
144 function [Y, P, R] = get_YPR(DCM)
145
146 t1 = sqrt( DCM(1,1)^2 + DCM(2,1)^2 );
147 P = atan2(-DCM(3,1), t1);
148
149 if abs( abs(P) - pi/2 ) < 0.001
150     Y = 0;
151     if P > 0
152         R = atan2(DCM(1,2), DCM(2,2));
153     else
154         R = -atan2(DCM(1,2), DCM(2,2));
155     end
156 else
157     Y = atan2(DCM(2,1), DCM(1,1));
158     R = atan2(DCM(3,2), DCM(3,3));
159 end;
160
161 end
162
163 %%%
164
165 function u_sat = sat(u, umin, umax)
166
167 u = min([u, umax]);
168 u = max([u, umin]);
169
```

```

170 u_sat = u;
171
172 end

```

D.2 Controllo automatico della distanza

```

1 function [CRS_XYZ_ref, is_rigid_constr_on] = rgb].133,.545,.133...
   CRS_dist_controller(CRS_XYZ, CRS_YPR, PRXS_YP, rgb].133,.545,.133...
   PHANToM_XYZ_ref, d_rigid, PRXS_dist, is_PRXS_on, k)
2
3 persistent is_auto_ctrl_on;
4 persistent CRS_XYZ_ref_prev;
5
6 %   init persistent vars
7 if isempty(is_auto_ctrl_on),
8     is_auto_ctrl_on = 0;
9 end;
10
11 if isempty(CRS_XYZ_ref_prev),
12     CRS_XYZ_ref_prev = PHANToM_XYZ_ref;
13 end;
14
15
16 %%   Set initial XYZ coords
17 CRS_XYZi = CRS_XYZ;
18
19
20 %%   Rigid constraint detection
21 [H, is_rigid_constr_on] = get_proj(PHANToM_XYZ_ref, rgb].133,.545,.133...
   CRS_XYZ, CRS_YPR, is_PRXS_on, PRXS_YP, PRXS_dist, d_rigid);
22
23 %%   Set target XYZ coords to reach
24 if is_rigid_constr_on,
25     CRS_XYZf = H;
26 else
27     CRS_XYZf = PHANToM_XYZ_ref;
28 end;
29
30
31
32 %%   Determine next controller state
33 CRS_XYZ_Delta = CRS_XYZf - CRS_XYZi;
34 norm_CRS_XYZ_Delta = norm(CRS_XYZ_Delta);
35
36 if is_auto_ctrl_on,
37     if (~is_rigid_constr_on) && (norm_CRS_XYZ_Delta < 30)
38         is_auto_ctrl_on = 0;
39     end;
40 else
41     if is_rigid_constr_on,
42         is_auto_ctrl_on = 1;
43     end;
44 end;
45
46
47 %%   Determine controller output
48 if is_auto_ctrl_on,
49     if norm_CRS_XYZ_Delta < 20,
50         norm_CRS_XYZ_Delta = 20;
51     end;

```

```

52
53     CRS_XYZ_ref = CRS_XYZ_ref_prev + k * CRS_XYZ_Delta/norm_CRS_XYZ_Delta;
54 else
55     CRS_XYZ_ref = PHANToM_XYZ_ref;
56 end;
57
58 CRS_X_ref_sat = sat(CRS_XYZ_ref(1), 304.8-100, 304.8+200);
59 CRS_Y_ref_sat = sat(CRS_XYZ_ref(2), -100, 100);
60 CRS_Z_ref_sat = sat(CRS_XYZ_ref(3), 330.2-150, 330.2*2+40);
61 CRS_XYZ_ref = [CRS_X_ref_sat; CRS_Y_ref_sat; CRS_Z_ref_sat];
62
63 %% Update ref integrator state
64 CRS_XYZ_ref_prev = CRS_XYZ_ref;
65
66
67 end
68
69 %%%
70
71 function [H, is_rigid_constr_on] = rgb].133,.545,.133...
    get_proj(PHANToM_XYZ_ref, CRS_XYZ, CRS_YPR, rgb].133,.545,.133...
    is_PRXS_on, PRXS_YP, PRXS_dist, d_rigid)
72
73 persistent is_rigid_constr_on_prev;
74
75 % init persistent vars
76 if isempty(is_rigid_constr_on_prev),
77     is_rigid_constr_on_prev = 0;
78 end;
79
80 if is_PRXS_on,
81     % get unit vector orthogonal to the plane
82     np = get_np(CRS_YPR, PRXS_YP);
83
84     % get orth proj of EE frame origin on the plane
85     O = CRS_XYZ - PRXS_dist*np;
86
87     % get true distance of PHANToM ref from the plane
88     d_phantom = (PHANToM_XYZ_ref-O)'*np;
89
90     % get orth proj of PHANToM ref on the plane of rgb].133,.545,.133...
    the rigid constraint
91     e = d_phantom - d_rigid;
92     H = PHANToM_XYZ_ref - e*np;
93
94     % rigid constr detection
95     eL = -5; % hysteresis threshold low
96     eH = 5; % hysteresis threshold high
97     if (e > eH) && (is_rigid_constr_on_prev == 1),
98         is_rigid_constr_on = 0;
99     elseif (e < eL) && (is_rigid_constr_on_prev == 0),
100         is_rigid_constr_on = 1;
101     else
102         is_rigid_constr_on = is_rigid_constr_on_prev;
103     end;
104 else
105     is_rigid_constr_on = 0;
106     H = PHANToM_XYZ_ref;
107 end;
108
109 % update prev vars
110 is_rigid_constr_on_prev = is_rigid_constr_on;
111

```

```
112 end
113
114 %%%
115
116 function nEE = get_nEE(CRS_YPR)
117
118 Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
119
120 nEE = Rbe*[1; 0; 0];
121
122 end
123
124 %%%
125 function np = get_np(CRS_YPR, PRXS_YP)
126
127 Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
128 Rep = get_DCM(PRXS_YP(1), PRXS_YP(2), 0);
129 Rbp = Rbe*Rep;
130
131 np = Rbp*[-1; 0; 0];
132
133 end
134
135 %%%
136
137 function DCM = get_DCM(Y, P, R)
138
139 cy = cos(Y);    sy = sin(Y);
140 cp = cos(P);    sp = sin(P);
141 cr = cos(R);    sr = sin(R);
142
143 DCM = zeros(3,3);
144
145 DCM(1,1) = cy*cp;
146 DCM(2,1) = sy*cp;
147 DCM(3,1) = -sp;
148
149 DCM(1,2) = cy*sp*sr - sy*cr;
150 DCM(2,2) = sy*sp*sr + cy*cr;
151 DCM(3,2) = cp*sr;
152
153 DCM(1,3) = cy*sp*cr + sy*sr;
154 DCM(2,3) = sy*sp*cr - cy*sr;
155 DCM(3,3) = cp*cr;
156
157 end
158
159 %%%
160
161 function u_sat = sat(u, umin, umax)
162
163 u = min([u, umax]);
164 u = max([u, umin]);
165
166 u_sat = u;
167
168 end
```

D.3 Generazione del feedback di forza

```

1 function F = force_feedback_controller(CRS_YPR, rgb].133,.545,.133...
    PRXS_dist, FF_dist, rigid_dist, is_PRXS_on)
2
3 persistent k;
4
5 %% Init persistent vars
6 if isempty(k),
7     k = 0;
8 end;
9
10 %% Get force direction (unit vector)
11 uEE = get_nEE(CRS_YPR);
12 uF = -uEE;
13
14 %% Get force intensity
15 err_dist = sat(FF_dist-PRXS_dist, 0, FF_dist-rigid_dist);
16
17 Kp = 50;
18 Kd = 0.5;
19 Ts = 0.001;
20
21 if is_PRXS_on,
22     aF = get_PD_out(err_dist, Kp, Kd, Ts);
23 else
24     aF = 0;
25 end;
26
27 aF = mov_avg_filt(aF, 16);
28
29 %% Return force feedback
30 if err_dist < (FF_dist-rigid_dist),
31     F = aF*uF;
32     k = 0;
33 else
34     aFsin = 0.5; % rgb].133,.545,.133...
35     fFsin = 5; % rgb].133,.545,.133...
36     NFsin = round(1/fFsin/0.001); % sin samples/period
37     Fsin = aFsin * sin(2*pi*k/NFsin);
38
39     F = (aF + Fsin)*uF;
40
41     k = mod(k+1, NFsin);
42 end;
43 F = aF*uF;
44
45 end
46
47 %%%
48
49 function u_sat = sat(u, umin, umax)
50
51 u = min([u, umax]);
52 u = max([u, umin]);
53
54 u_sat = u;
55
56 end
57
58 %%%
59
60 function DCM = get_DCM(Y, P, R)

```

```

61
62 cy = cos(Y);      sy = sin(Y);
63 cp = cos(P);      sp = sin(P);
64 cr = cos(R);      sr = sin(R);
65
66 DCM = zeros(3,3);
67
68 DCM(1,1) = cy*cp;
69 DCM(2,1) = sy*cp;
70 DCM(3,1) = -sp;
71
72 DCM(1,2) = cy*sp*sr - sy*cr;
73 DCM(2,2) = sy*sp*sr + cy*cr;
74 DCM(3,2) = cp*sr;
75
76 DCM(1,3) = cy*sp*cr + sy*sr;
77 DCM(2,3) = sy*sp*cr - cy*sr;
78 DCM(3,3) = cp*cr;
79
80 end
81
82 %%%
83
84 function y = get_PD_out(u, Kp, Kd, Ts)
85
86 persistent prev_u prev_yd;
87
88 %% Initialize persistent vars
89 if isempty(prev_u),
90     prev_u = 0;
91 end;
92
93 if isempty(prev_yd),
94     prev_yd = 0;
95 end;
96
97 %% Set LPF params
98 b0 = (1-exp(-2*pi*5*0.001));
99 a0 = exp(-2*pi*5*0.001);
100
101 %% mm to m conversion (distanza misurata in ingresso)
102 u = 0.001*u;
103
104 %% Get PD out
105 yp = Kp*u;
106 yd = a0*prev_yd + b0*Kd*(u-prev_u)/Ts;
107 y = yp + yd;
108
109 %% Update prev vars
110 prev_u = u;
111 prev_yd = yd;
112
113 end
114
115 %%%
116
117 function nEE = get_nEE(CRS_YPR)
118
119 Rbe = get_DCM(CRS_YPR(1), CRS_YPR(2), CRS_YPR(3));
120
121 nEE = Rbe*[1; 0; 0];
122
123

```

```
124 end
125
126 %%%
127
128 function y = hyst_cmp(u, umin, umax);
129
130 persistent y_prev;
131
132 %% Initialize persistent vars
133 if isempty(y_prev),
134     y_prev = 0;
135 end;
136
137 if (u > umax) && (y_prev == 0),
138     y = 1;
139 elseif (u < umin) && (y_prev == 1),
140     y = 0;
141 else
142     y = y_prev;
143 end;
144
145 y_prev = y;
146
147 end
148
149 %%%
150
151 function y = mov_avg_filt(u, N);
152
153 persistent u_prev;
154
155 if isempty(u_prev),
156     u_prev = zeros(N-1,1);
157 end;
158
159 y = (1/N)*sum([u; u_prev]);
160
161 u_prev = [u; u_prev(1:end-1)];
162
163 end
```