

POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria dell'Automazione



**Controllo basato su ottimizzazione
vincolata per l'interazione sicura
uomo-robot**

Relatore: Prof. Paolo Rocco

Correlatore: Dott. Andrea Maria Zanchettin

Autore: Mattia Marconi

Matricola: 798525

Anno Accademico 2013-2014

Dedicato alla mia famiglia

Ex absurdo sequitur quodlibet

Anonimo

Ringraziamenti

Ringrazio il Prof. Paolo Rocco per avermi dato la possibilità di lavorare a questo progetto di tesi nell'ambito della robotica industriale, ambiente che mi ha sempre interessato. Lo ringrazio per il tempo che mi ha dedicato, soprattutto nella fase di redazione della tesi. Ringrazio il Dott. Andrea Maria Zanchettin per la grande disponibilità che mi ha riservato, anche nei momenti più inopportuni. Lo ringrazio per i preziosi consigli che mi ha offerto, fondamentali per il completamento di questo lavoro. Ringrazio il Dott. Giovanni Buizza Avanzini, il Dott. Matteo Ragaglia, il Dott. Matteo Parigi ed il Dott. Roberto Rossi per l'aiuto concreto che mi hanno dato durante questi mesi. Infine vorrei ringraziare il responsabile del laboratorio, il Dott. Vasco Previtali, per tutto il tempo che mi ha dedicato, per avermi fornito o costruito tutti gli strumenti necessari allo svolgimento del mio lavoro e soprattutto per aver reso piacevole la mia permanenza all'interno del laboratorio.

Indice

Ringraziamenti	III
Sommario	XI
Abstract	XV
1 Introduzione	1
1.1 Obbiettivi della tesi	3
1.2 Organizzazione della tesi	3
2 Stato dell'arte nella cooperazione uomo-robot	5
2.1 Tecnologie per la sicurezza passiva	6
2.2 Strategie per la sicurezza attiva	9
3 Setup sperimentale	13
3.1 Sistema robotico	14
3.2 Il sensore distribuito	16
3.2.1 Tecnologia del sensore	18
3.3 Gripper	23
3.3.1 Adattatori e flange	24
3.3.2 Circuito pneumatico	26
3.3.3 Circuito di alimentazione delle valvole	29

4	Strategia di controllo per la collision avoidance nei manipolatori industriali	33
4.1	Struttura del problema	37
4.2	Vincoli cinematici nello spazio dei giunti	37
4.3	Vincoli nello spazio di lavoro e generazione della traiettoria	39
4.4	Vincoli di safety	40
4.5	Schema di controllo	44
4.5.1	Generazione della traiettoria	44
4.5.2	Controllore reattivo	45
4.5.3	Cinematica diretta	45
4.5.4	Interfaccia sensori	45
4.6	Algoritmo di controllo	46
4.6.1	Algoritmo punto-punto con gradi di libertà ridondanti	46
4.7	Metodi risolutivi del problema di controllo in caso di robot ridondante	48
4.7.1	Problema di ottimizzazione secondario in cascata	48
4.7.2	Proiezione nello spazio nullo	49
4.7.3	Solutore dedicato	50
5	Realizzazione e validazione del sistema di controllo	53
5.1	Macchina a stati	55
5.1.1	Skill di movimento nello spazio cartesiano con ottimizzazione vincolata	56
5.1.2	Controllo del gripper	62
5.2	Prova in simulazione	63
5.2.1	Generazione ostacolo virtuale	63
5.3	Prove sperimentali	68
5.3.1	Prova senza ostacoli	68
5.3.2	Prova con ostacolo fisso che non impedisce il completamento del task	73

5.3.3	Prova con ostacolo fisso che impedisce il completamen- to del task	77
5.3.4	Prova con ostacolo mobile	82
5.4	Commenti	87
6	Conclusioni e sviluppi futuri	89
	Bibliografia	93

Elenco delle figure

1.1	Stima delle vendite di robot industriali nel mondo	1
1.2	Cella di lavoro robotizzata	2
2.1	Possibile esempio di cooperazione uomo-robot	6
2.2	Esempio di robot collaborativo della KUKA	7
2.3	Schema dell'approccio DM	8
3.1	Robot FRIDA sviluppato da ABB	14
3.2	Teach pendant di FRIDA	15
3.3	Braccio di FRIDA con relative terne di DH	16
3.4	Raffigurazione sensore distribuito	17
3.5	Sensore distribuito realizzato con la stampa 3D	18
3.6	Schema posizionamento sensori sulla shell	19
3.7	Sensore SHARP GP2Y0A21YK	19
3.8	Rappresentazione funzionamento sensore	20
3.9	Schematizzazione funzionamento PSD	20
3.10	Triangolazione	21
3.11	Curva caratteristica tensione-distanza	22
3.12	Modelli di gripper SHUNK utilizzati nel setup sperimentale (a sinistra MPZ e a destra JGP)	23
3.13	Adattatori completi	25
3.14	Flange di aggancio rapido dei gripper	26
3.15	Funzionamento fissaggio flange	26

3.16	Schema del circuito idraulico per l'alimentazione dei gripper SHUNK	27
3.17	Stati possibili della valvola	28
3.18	Schema circuitale amplificatore	30
4.1	Rappresentazione grafica del massimo insieme invariante iQ_∞ per un dato giunto i	38
4.2	Rappresentazione della relazione tra produttività e sicurezza .	40
4.3	Rappresentazione di un link del robot e di un punto di ostacolo	41
4.4	Schema a blocchi dell'algoritmo di controllo	44
4.5	Esecuzione dell'algoritmo di controllo reattivo e generazione della traiettoria	45
5.1	Schematizzazione dell'unità di governo con controllore esterno	54
5.2	Schema macchina a stati	55
5.3	Schema funzionamento del controllo durante una skill di mo- vimento nello spazio cartesiano	57
5.4	Ricostruzione di un punto di ostacolo nello spazio tramite cinematica diretta	59
5.5	Configurazione finale del braccio robotico simulato a seguito dell'interruzione del task	64
5.6	Andamento con ostacolo virtuale della coordinata X dell'end effector con relativi stati	65
5.7	Andamento con ostacolo virtuale della coordinata Y dell'end effector con relativi stati	65
5.8	Andamento con ostacolo virtuale della coordinata Z dell'end effector con relativi stati	66
5.9	Andamento con ostacolo virtuale del primo angolo di eulero dell'end effector (Z) con relativi stati	66

5.10	Andamento con ostacolo virtuale del secondo angolo di eulero dell'end effector (Y) con relativi stati	67
5.11	Andamento con ostacolo virtuale del terzo angolo di eulero dell'end effector (Z) con relativi stati	67
5.12	Foto della prova sperimentale senza ostacolo	69
5.13	Andamento senza ostacoli della coordinata X dell'end effector con relativi stati	69
5.14	Andamento senza ostacoli della coordinata Y dell'end effector con relativi stati	70
5.15	Andamento senza ostacoli della coordinata Z dell'end effector con relativi stati	70
5.16	Andamento senza ostacoli del primo angolo di eulero dell'end effector (Z) con relativi stati	71
5.17	Andamento senza ostacoli del secondo angolo di eulero dell'end effector (Y) con relativi stati	71
5.18	Andamento senza ostacoli del terzo angolo di eulero dell'end effector (Z) con relativi stati	72
5.19	Foto della prova sperimentale con ostacolo che non impedisce lo svolgimento del task	73
5.20	Andamento con ostacolo non critico della coordinata X dell'end effector con relativi stati	74
5.21	Andamento con ostacolo non critico della coordinata Y dell'end effector con relativi stati	74
5.22	Andamento con ostacolo non critico della coordinata Z dell'end effector con relativi stati	75
5.23	Andamento con ostacolo non critico del primo angolo di eulero dell'end effector (Z) con relativi stati	75
5.24	Andamento con ostacolo non critico del secondo angolo di eulero dell'end effector (Y) con relativi stati	76

5.25	Andamento con ostacolo non critico del terzo angolo di eulero dell'end effector (Z) con relativi stati	76
5.26	Foto prova con ostacolo fisso che impedisce il compimento del task	77
5.27	Andamento con ostacolo critico della coordinata X dell'end effector con relativi stati	79
5.28	Andamento con ostacolo critico della coordinata Y dell'end effector con relativi stati	79
5.29	Andamento con ostacolo critico della coordinata Z dell'end effector con relativi stati	80
5.30	Andamento con ostacolo critico del primo angolo di eulero dell'end effector (Z) con relativi stati	80
5.31	Andamento con ostacolo critico del secondo angolo di eulero dell'end effector (Y) con relativi stati	81
5.32	Andamento con ostacolo critico del terzo angolo di eulero dell'end effector (Z) con relativi stati	81
5.33	Foto prova con ostacolo in movimento	82
5.34	Andamento con ostacolo mobile della coordinata X dell'end effector con relativi stati	83
5.35	Andamento con ostacolo mobile della coordinata Y dell'end effector con relativi stati	83
5.36	Andamento con ostacolo mobile della coordinata Z dell'end effector con relativi stati	84
5.37	Andamento con ostacolo mobile del primo angolo di eulero dell'end effector (Z) con relativi stati	84
5.38	Andamento con ostacolo mobile del secondo angolo di eulero dell'end effector (Y) con relativi stati	85
5.39	Andamento con ostacolo mobile del terzo angolo di eulero dell'end effector (Z) con relativi stati	85

Elenco delle tabelle

5.1	Definizione flag	60
5.2	Modifica ΔE e Δf in casi critici	60
5.3	Assegnamento bit controllo gripper	63

Sommario

L'utilizzo dei robot in ambito industriale è diventato sempre più diffuso e capillare. Ciò ha portato alla necessità di un'attenta analisi delle problematiche relative alla sicurezza degli ambienti di lavoro, che vedono un sempre più stretto contatto tra operatori umani e robot. Come è noto, la tecnologia attuale non consente ancora una sicura collaborazione uomo-robot. Per ovviare a tale problema, le normative attualmente vigenti prevedono una netta separazione tra ambienti in cui operano i manipolatori robotici e quelli in cui lavorano gli operatori umani. Questo progetto di tesi si inserisce nell'ambito delle ricerche volte a studiare le metodologie che rendano possibile una migliore cooperazione uomo-robot. Lo scopo di questa tesi è quindi quello di implementare un algoritmo che cerchi di mediare tra i requisiti di efficienza necessari alle industrie e quelli di sicurezza degli operatori umani. Per raggiungere questo obiettivo, si utilizza un algoritmo di generazione della traiettoria da far compiere ad un manipolatore robotico all'interno di un problema di ottimizzazione vincolata. Tale problema, per come è strutturato, permette la specifica dei requisiti di sicurezza e di efficienza produttiva. Per definire i vincoli di sicurezza si utilizzano le distanze di eventuali ostacoli dal braccio robotico. Tali distanze sono ottenute integrando la tecnologia del sensore di distanza distribuito. Al fine di testare l'algoritmo, dopo aver implementato adeguate simulazioni, si sono realizzati vari esperimenti in cui vengono utilizzati anche una coppia di pinze SCHUNK applicati al braccio

robotico, così da simulare una situazione simile a quelle che realmente si incontrano in ambito industriale. Da questi esperimenti è emerso come l'algoritmo riesca a gestire correttamente tutte quelle situazioni di potenziale collisione con un ostacolo, sia fisso che mobile, adottando efficaci manovre evasive. Si conclude discutendo i possibili sviluppi di questa tecnologia, come ad esempio la suddivisione del carico computazionale tra vari calcolatori in parallelo oppure l'utilizzo di strumenti diversi dal sensore di distanza distribuito, come il Kinect, per ottenere le informazioni sull'ambiente circostante al robot, necessarie all'algoritmo.

Abstract

The employment of robots in the industry is by now widespread. This led to the need of an accurate analysis of the safety risks in working environments. This is due to the growing demand for strict interaction between industrial manipulators and human operators. However, the state of the art in robotics does not allow a safe collaboration between humans and robots. For this reason, the existing standards prescribe a separation between the human workspace and the environment where the robots are installed. This thesis belongs to the mainstream of research that develops methods to accomplish a better and safer human-robot interaction. The purpose of this work is to implement an algorithm that could find a compromise between the efficiency requirements, that are necessary to have a better industrial result, and those of the safety of human workers. This goal is achieved by combining a trajectory generation algorithm with a constraint-based optimization problem, to produce a task trajectory for the robot. In this way it is possible to jointly specify safety and production requirements. The distances between any obstacles and the robotic arm are used to define the safety constraints for the algorithm. Those distances are measured by the use of the technology of the distributed distance sensor. Together with several simulations, to test the algorithm, many experiments have been carried out also integrating some grippers from SCHUNK installed on the dual arm robot ABB FRIDA. These ones were applied to the robotic arm to emulate the situation that

can be found in the industrial environment. The results of the experiments show that the algorithm is capable of handling many dangerous situations of possible collisions with an obstacle both still and moving, adopting effective evasive actions. In the end many possible improvements are discussed. One example could be the split of the computational burden between many computers running in parallel. Another could be the use of the Kinect technology to obtain more precise information of the environment around the robot.

Capitolo 1

Introduzione

Il numero di robot utilizzati in ambito industriale è in costante crescita negli ultimi anni. I dati raccolti dall'IFR (International Federation of Robotics) (Fig. 1.1) che coprono un arco temporale che va dal 1995 al 2013 [28], identificano proprio nell'ultimo anno quello in cui si sono venduti più robot. Infatti nel solo 2013 la vendita di robot ha subito un incremento del 12%, arrivando a quasi 180,000 unità. L'utilizzo massiccio dei robot nella

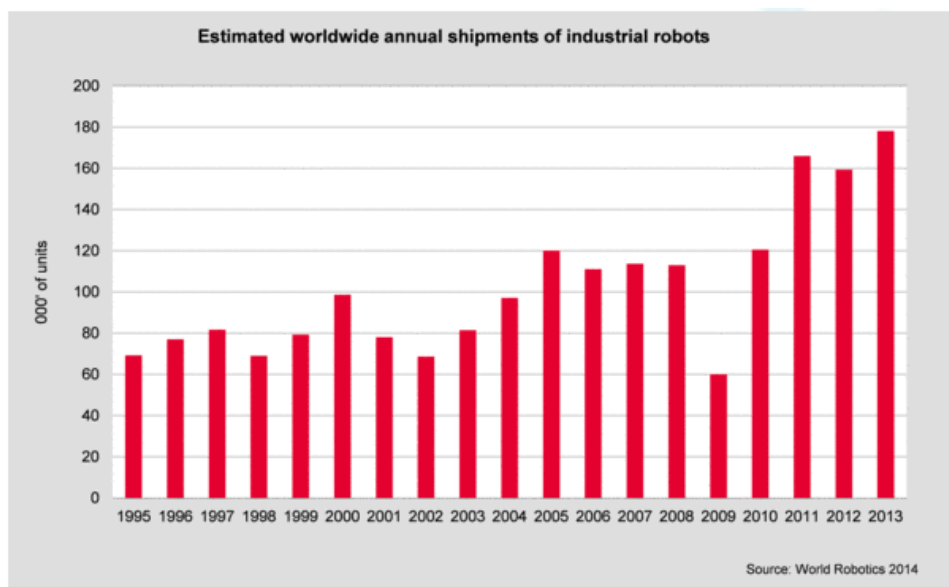


Figura 1.1: Stima delle vendite di robot industriali nel mondo

produzione industriale genera un netto aumento della produttività, facendo da traino agli investimenti in questo campo. Visto il trend di crescita, si può pensare che l'utilizzo dei robot in ambito industriale divenga sempre più diffuso e capillare. Ciò porta con sé la necessità di un'attenta analisi delle problematiche relative alla sicurezza degli ambienti di lavoro, che vedranno un sempre più stretto contatto di operatori umani e robot. Come è noto, la tecnologia attuale non consente una sicura collaborazione uomo-robot. Per ovviare a questo problema, le normative attualmente vigenti [30] prevedono una netta separazione degli ambienti in cui operano i manipolatori robotici da quelli in cui lavorano gli operatori umani, come mostrato in figura 1.2. Queste norme sono strutturate di modo da garantire al massimo la sicurezza



Figura 1.2: Cella di lavoro robotizzata

agli operatori umani nello svolgimento delle loro mansioni. A tale scopo, allo stato attuale vengono montate barriere metalliche oppure ottiche di separazione tra gli spazi accessibili agli umani e quelli in cui operano i robot. Inoltre, vengono installate varie tipologie di sensori che monitorano eventuali intrusioni nell'ambiente assegnato ai robot, in modo da bloccare l'esecuzione

del lavoro se necessario. Tutte queste restrizioni causano un considerevole rallentamento delle operazioni industriali rispetto al livello di efficienza che sarebbe ottenibile se la cooperazione uomo-robot fosse sicura.

1.1 Obiettivi della tesi

Nel quadro precedentemente delineato, l'obiettivo di questa tesi è quindi quello di concepire, realizzare e validare sperimentalmente un algoritmo orientato alla sicurezza nelle interazioni uomo-robot che cerchi di mediare tra i requisiti di efficienza che sono necessari alle industrie, e quelli di sicurezza che permetterebbero una maggiore cooperazione uomo-robot.

Per realizzare questo scopo, sarà utilizzato un algoritmo di programmazione della traiettoria da far compiere ad un manipolatore robotico all'interno di un problema di ottimizzazione vincolata. Tale problema sarà quello che permetterà la specifica dei requisiti di sicurezza e di efficienza produttiva.

Al fine di poter specificare i requisiti di sicurezza, si sceglierà di definire dei vincoli che andranno ad agire sulla distanza relativa uomo-robot, durante l'esecuzione di vari task, sia da parte del robot che dell'operatore umano. Le informazioni necessarie alla computazione di tali vincoli saranno fornite da una tecnologia realizzata in progetti antecedenti a questo dal Politecnico di Milano; ovvero da un sensore di distanza distribuito. Ulteriore scopo di questa tesi sarà quindi il tentativo di integrare questa tecnologia a basso costo in una struttura algoritmica complessa di quelle utilizzate in precedenza.

1.2 Organizzazione della tesi

La tesi è strutturata in questo modo:

- **Capitolo 1:** si discuterà dell'attuale stato dell'arte della sicurezza nell'interazione uomo-robot in ambito industriale analizzandone i requisiti. Inoltre si discuterà una possibile soluzione di compromes-

so tra necessità di maggiore sicurezza ed efficienza nella produzione industriale.

- **Capitolo 2:** si riporta la descrizione del setup sperimentale adottato per testare l'efficacia dell'algoritmo di controllo sul manipolatore dual arm FRIDA di ABB.

Saranno presentati nel dettaglio tutti i componenti ed i sottosistemi necessari al corretto svolgimento degli esperimenti.

- **Capitolo 3:** si introdurrà dapprima il concetto di sicurezza in relazione alle traiettorie compiute da un manipolatore robotico nello svolgimento dei suoi task; poi verranno discussi alcuni algoritmi che cercano di garantire tale sicurezza senza inficiare la produttività del manipolatore. Sarà infine discusso nel dettaglio l'algoritmo implementato in questo lavoro di tesi e se ne discuteranno alcuni dei possibili metodi per calcolarne la soluzione.

- **Capitolo 4:** sarà dapprima mostrato lo schema generale con il quale si è implementato l'algoritmo di controllo. Ci si soffermerà poi nell'illustrare i singoli componenti che lo compongono, ponendo particolare attenzione alla macchina a stati, che contiene l'algoritmo presentato nel capitolo 3. Si andranno poi ad illustrare e commentare i risultati ottenuti dalle simulazioni e dalle prove reali effettuate sul robot ABB FRIDA.

- **Conclusioni:** In questa sezione saranno discussi i risultati ottenuti dalle prove sperimentali. Si proporranno inoltre varie soluzioni e migliorie in relazione ai problemi incontrati durante lo svolgimento del progetto. Alcune delle soluzioni saranno facilmente attuabili da chi, eventualmente, proseguirà questo lavoro, altre invece richiederanno un'analisi prolungata nel tempo.

Capitolo 2

Stato dell'arte nella cooperazione uomo-robot

IN questo capitolo si discuterà dell'attuale stato dell'arte della sicurezza nell'interazione uomo-robot in ambito industriale analizzandone i requisiti. Inoltre si discuterà una possibile soluzione di compromesso tra necessità di maggiore sicurezza ed efficienza nella produzione industriale.

La chiave per poter sfruttare a pieno il contributo della robotica nei processi produttivi e quindi incrementare il più possibile l'efficienza di questi ultimi è, come già detto, il rendere il più naturale e sicura possibile l'interazione fra operatori umani e robot. In questa maniera si potrebbero affidare agli operatori umani tutti quei processi che richiedono un'automazione troppo complessa, mentre ai robot resterebbero destinati tutti quei compiti che necessitano maggiore rapidità di esecuzione e standard di qualità elevati.(Fig.2.1) Per rendere possibile questa strategia produttiva, bisogna individuare e progettare metodologie di controllo e supervisione tali da rendere naturale e innocua l'interazione umana con i robot. Purtroppo, a causa delle normative attuali, nell'ambito industriale non è possibile superare queste problematiche. La ricerca in questo ambito si sta quindi muovendo nella robotica di

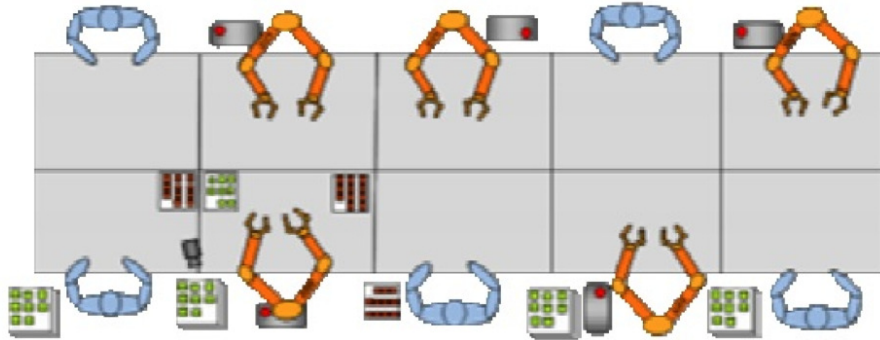


Figura 2.1: Possibile esempio di cooperazione uomo-robot

servizio. Sono stati promossi anche importanti progetti in ambito europeo per poter dare un importante contributo alla soluzione dei problemi di cooperazione uomo-robot nelle industrie. In tal senso è significativo il progetto ROSETTA [40] primo fra tutti (conclusosi nel 2013), a cui ha partecipato anche il Politecnico di Milano attraverso il gruppo Merlin. Gli approcci utilizzati per tentare di risolvere i problemi di cooperazione uomo-robot sono essenzialmente due:

- sicurezza passiva;
- sicurezza attiva.

2.1 Tecnologie per la sicurezza passiva

Tutte le metodologie volte ad aumentare la sicurezza passiva nell'interazione degli esseri umani con i robot sono essenzialmente volte a modificare la struttura e il funzionamento di questi ultimi, in modo da diminuire la probabilità di incidenti. I robot progettati seguendo questo criterio sono caratterizzati da strutture leggere e flessibili, in modo da minimizzare i danni causati da un eventuale impatto su di un essere umano e per questo sono ricoperti da materiali leggeri, come ad esempio materiali visco elastici. Le geometrie di

questi robot sono altresì rese il meno pericolose possibile, ad esempio smussandone gli spigoli. Un esempio di robot realizzato secondo questi criteri è quello rappresentato in figura 2.2. Una metodologia di progetto mirata a



Figura 2.2: Esempio di robot collaborativo della KUKA

ridurre l'inerzia ed il peso delle parti mobili di un manipolatore, spesso utilizzata in ambito industriale, è quella del Distributed Macro-Mini actuation DM[4]. Questa tecnica consiste in una diversa distribuzione degli attuatori del robot. Vicino alla base del robot vengono poste tutte le parti più pesanti,

come i motori quindi per trasmettere il moto ai giunti vengono poi utilizzati cavi e pulegge appositi. Nello specifico, questa tecnica consiste nella suddivisione dell'attuazione del robot fra due attuatori connessi in parallelo e disposti in zone differenti del manipolatore. Le componenti ad alta frequenza dell'attuazione vengono fornite da piccoli motori dotati di poca inerzia direttamente connessi ai giunti del manipolatore. Le componenti a bassa frequenza, invece, sono generate da una serie di attuatori più pesanti posti alla base del robot. Questo approccio è schematizzato in figura 2.3. Un'altra

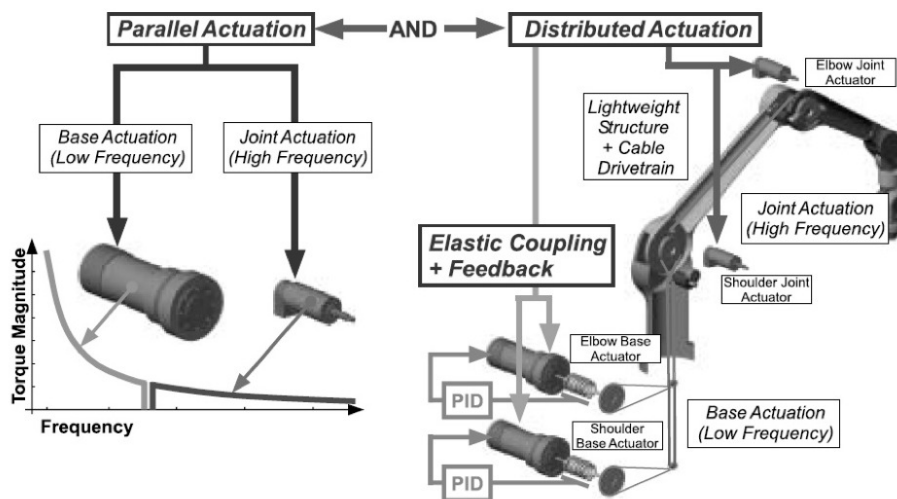


Figura 2.3: Schema dell'approccio DM

tencica volta al miglioramento della sicurezza intrinseca ai manipolatori robotici consiste nell'utilizzo di attuatori cosiddetti ad impedenza variabile o VIA[24][6]. Questi dispositivi sono in grado di variare l'impedenza meccanica, ossia la relazione tra le coppie applicate ad un punto e il relativo moto, a seconda della fase di lavoro. L'effetto pratico sarà quello di avere, nelle fasi di lavorazione ad alta velocità, un'impedenza modificata tale da ridurre la rigidità e lo smorzamento del manipolatore, così da minimizzare i danni causati da un eventuale impatto. A basse velocità invece, la modifica dell'impedenza causerà un aumento di rigidità e smorzamento, così da avere

una migliore risposta in accelerazione dei link e una minore oscillazione in fase di frenata.[1]

2.2 Strategie per la sicurezza attiva

La sicurezza attiva non è altro che lo sviluppo di particolari strategie di controllo del manipolatore che, integrando una sensoristica dedicata, permettono di sfruttare il costante monitoraggio dell'ambiente circostante il robot per modificare dinamicamente il comportamento del manipolatore in caso di situazioni potenzialmente rischiose: l'avvicinarsi di un operatore durante l'esecuzione di un task o la preparazione di un cambio pezzo.

La ricerca nell'ambito della sicurezza attiva è di particolare interesse per l'industria poiché, con il progressivo sviluppo di sensori sempre più sofisticati e a basso costo, questa permetterà di sfruttare con maggiore efficienza la presenza dei robot nelle industrie. Le tipologie di sensori esteroceettivi utilizzati normalmente negli algoritmi di controllo per la sicurezza sono due:

- Sensori che hanno la funzione di ricostruire la geometria dell'ambiente circostante il robot (telecamere a tempo di volo, scanner laser, sensori di prossimità etc.); sono i più utilizzati in ambito industriale.
- Sensori che permettono il riconoscimento di una collisione con un corpo estraneo al robot (sensori di forza); questi sono ad oggi molto utilizzati sui robot collaborativi di piccola taglia.

Le modalità d'impiego di tale sensoristica sono molteplici e variano a seconda della strategia di controllo prescelta. Ad esempio, tramite l'utilizzo di semplici videocamere di sorveglianza installate in un punto fisso esterno al robot e con il supporto di appositi algoritmi di elaborazione delle immagini, è possibile individuare le persone che eventualmente siano entrate nell'ambiente sorvegliato, seguirne gli spostamenti e addirittura fornire una stima delle loro intenzioni di percorso in real time [31]. Questi algoritmi,

se resi opportunamente robusti e certificati secondo le norme attualmente vigenti, potrebbero dare un significativo contributo alla sicurezza attiva. Un ulteriore esempio può essere dato dall'installazione direttamente sul robot di sensori di distanza, questi possono essere utilizzati per fornire informazioni direttamente al controllo riguardo la distanza di eventuali ostacoli. Questa soluzione presenta il duplice vantaggio di non necessitare calibrazioni, essendo i sensori montati direttamente sul robot e di poter totalmente evitare il problema dell'occlusione visiva, che si verifica se un oggetto o il robot stesso si frappongono fra il sensore e la zona da monitorare. Per garantire che la probabilità di fallimento dei controlli di sicurezza attiva sia quasi nulla, tutte le tecnologie utilizzate devono essere robuste a fronte di disturbi di vario genere: variazioni delle condizioni di luce, presenza di polveri in sospensione, disturbi elettromagnetici e ogni altro genere di disturbo che possa causare incertezza nella misura.

Nel corso degli anni sono state sviluppate diverse strategie di controllo attivo che hanno apportato vari miglioramenti nella sicurezza delle interazioni uomo-robot. Si può citare l'approccio di Heinzmann e Zelinsky in [18] che progettano un sistema di controllo per manipolatori robotici che limitasse i valori delle coppie in un intervallo di valori di sicurezza, il tutto all'interno di un algoritmo di controllo della posizione. Ancora si può ricordare lo schema di Zurada in [41] che fa uso di una serie di sensori per riconoscere la presenza di un operatore umano all'interno dello spazio del robot. Questo controllo impone al robot di attuare delle strategie di moto decise in precedenza (move as intended, slow down, emergency stop) a seconda della posizione dell'operatore.

Un'ulteriore metodologia di controllo che vale la pena ricordare è quella in [27][36] che si basa sul concetto di impedenza virtuale. Attorno all'ostacolo individuato viene costruita una superficie virtuale. Al robot viene poi applicata una forza che si oppone al suo moto, proporzionale alla profondità di

attraversamento della superficie. La forza virtuale viene poi tradotta in coppie antagoniste applicate ai giunti del manipolatore, di modo da costringere il robot ad allontanarsi da tale superficie.

Altra strategia, molto utilizzata in ambito industriale è quella citata in [23] che si basa sulla generazione di un indice di rischio, sulla base del quale vengono poi elaborate opportune azioni evasive che allontanino il robot dalla zona pericolosa.

Capitolo 3

Setup sperimentale

In questo capitolo si riporta la descrizione del setup sperimentale adottato per testare l'efficacia dell'algoritmo di controllo sul manipolatore dual arm dell'ABB Frida.

Saranno presentati nel dettaglio tutti i componenti ed i sottosistemi necessari al corretto svolgimento degli esperimenti.

Si è deciso di allestire un setup sperimentale che ricalcasse nella maniera più fedele possibile una tipica stazione di assemblaggio industriale. Si è scelto quindi di far eseguire una sequenza di manipolazioni di componenti ad un robot a due bracci, in uno spazio in cui fosse possibile la presenza di ostacoli in movimento ed ostacoli fissi. Andando nel dettaglio, i sistemi utilizzati negli esperimenti sono stati:

- Un manipolatore robotico a due bracci
- Gripper pneumatici
- Sensori di distanza distribuiti

3.1 Sistema robotico

Il sistema robotico scelto per compiere gli esperimenti è il prototipo di manipolatore industriale a due bracci FRIDA della ABB. La peculiarità di questo manipolatore è di avere, per ogni braccio, sette gradi di libertà. Ciò permette al robot di muovere i bracci in maniera molto simile a quella umana, rendendolo quindi particolarmente adatto ad operare in ambienti a stretto contatto con operatori umani. Per garantire una maggiore sicurezza il robot è inoltre rivestito da un'imbottitura di gomma morbida in tutte le parti che potenzialmente possano urtare qualcosa nello spazio circostante durante l'esecuzione del lavoro. Il robot è equipaggiato con un'unità di governo in-



Figura 3.1: Robot FRIDA sviluppato da ABB

tegrata (denominata IRC5) che contiene l'elettronica di potenza necessaria all'alimentazione del robot e l'elettronica di segnale indispensabile al controllo. Per consentire una movimentazione manuale, FRIDA è dotato di un teach-pendant completo di joystick e touch screen, tramite cui è possibile controllarne ogni funzione. FRIDA è un robot intrinsecamente ridondante, ovvero le dimensioni dello spazio dei giunti (7 g.d.l per ogni braccio) sono



Figura 3.2: Teach pendant di FRIDA

maggiori di quelle dello spazio operativo (6 g.d.l che definiscono posizione e orientamento nello spazio). Questa peculiarità permette di sfruttare il grado di libertà in eccesso per muovere il braccio senza dover necessariamente variare la posizione e l'orientamento dell'end-effector. Negli esperimenti, la ridondanza cinematica di FRIDA è fondamentale, poiché è proprio grazie a questa che è possibile evitare eventuali ostacoli durante l'esecuzione dei task, senza che il robot sia costretto a fermarsi necessariamente. Inoltre, sempre grazie alla ridondanza, è possibile evitare le singolarità cinematiche, ossia quelle particolari configurazioni geometriche del braccio, in corrispondenza delle quali si ha una perdita di mobilità strutturale. La presenza di possibili singolarità impone ai manipolatori industriali classici (dotati di 6 g.d.l) alcune limitazioni nella scelta delle leggi di moto da imporre all'end-effector. In FRIDA, invece, è possibile assegnare le leggi di moto in maniera molto più libera poiché, se la configurazione del robot, in un dato istante, porta un giunto alla fine della sua corsa, è ancora possibile proseguire lungo la traiettoria pianificata grazie agli altri giunti. Tutte queste caratteristiche rendono

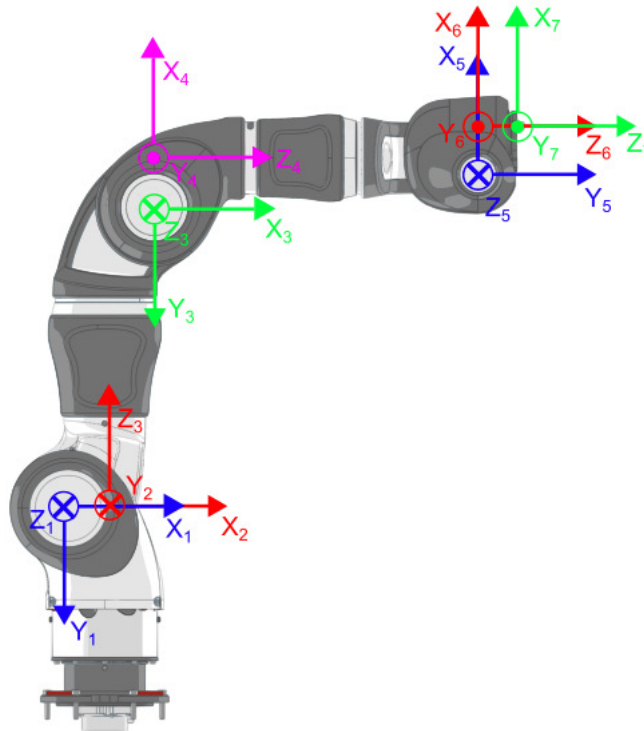


Figura 3.3: Braccio di FRIDA con relative terne di DH

FRIDA il candidato ideale per la collaborazione uomo-robot.

3.2 Il sensore distribuito

Allo scopo di favorire una interazione uomo-robot quanto più possibile sicura, è importante che la rappresentazione dello spazio circostante al robot sia il più dettagliata possibile e che sia costantemente aggiornata. In tal modo al controllo sono garantiti ad ogni istante tutti i dati necessari ad assicurare una sicura esecuzione dei task previsti. L'approccio sperimentale, in questo caso, prevede l'utilizzo di un sensore montato direttamente a bordo del braccio del robot. Questo sensore è realizzato precedentemente a questa tesi, mettendo insieme una pluralità di unità elementari, chiamate spot, che sono in grado di misurare localmente la distanza da un eventuale ostacolo[9]. Da

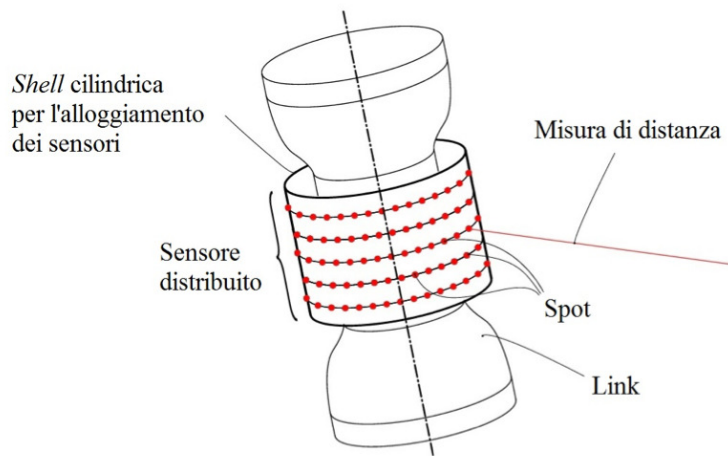


Figura 3.4: Rappresentazione sensore distribuito

un punto di vista costruttivo, il sensore di distanza distribuito è composto da una struttura pressochè cilindrica rigida denominata shell la quale ha la funzione di alloggiamento del maggior numero possibile di spot. La shell è poi realizzata in modo da poter essere fissata ad uno dei link del robot. Lo scopo di questo tipo di sensore è quello di fornire al controllore una descrizione dell'ambiente circostante il robot. Questa viene ricostruita tramite le informazioni restituite da ciascuno spot presente sulla shell, le quali non sono altro che misure di distanza fra il singolo spot e un eventuale oggetto che si trovi sulla traiettoria del raggio emesso da quest'ultimo. Per ottenere una copertura adeguata dello spazio operativo attorno al robot, il sensore distribuito dovrebbe essere costituito da un elevato numero di spot, il che lo renderebbe difficilmente realizzabile, oltre che molto costoso. Si è quindi deciso di ripiegare su di una tecnologia che potesse emulare l'idea originale di sensore distribuito, ma che fosse facilmente realizzabile e che avesse costi contenuti[34]. Il sensore distribuito che è stato poi utilizzato nel setup sperimentale in questione è stato infatti realizzato con l'alloggiamento di ventiquattro sensori di distanza facilmente reperibili sul mercato, in due shell completamente realizzate con la tecnica della stampa 3D. La prima

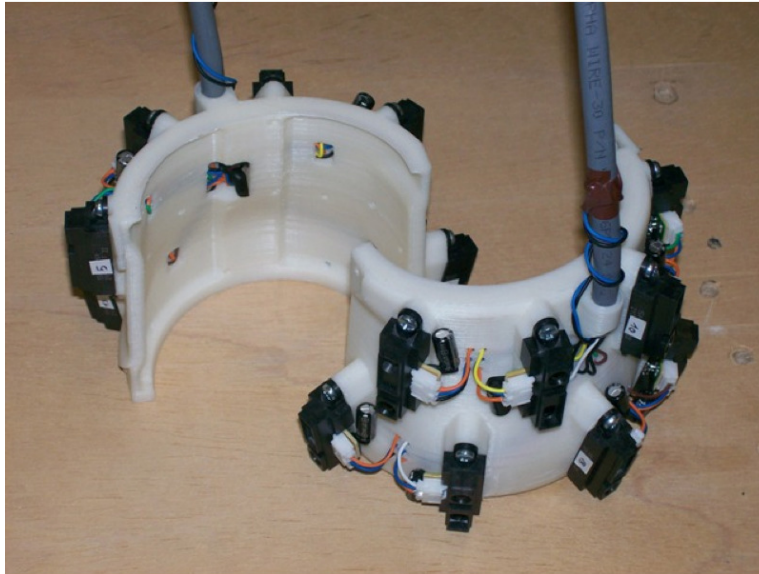


Figura 3.5: Sensore distribuito realizzato con la stampa 3D

shell, posizionata sul link 1 (parte di braccio tra la spalla e il gomito) alloggia 16 sensori suddivisi in due corone circolari, contenenti 8 sensori ciascuna e sfasate di 22.5° . In entrambe le corone i sensori sono posizionati ogni 45° , in modo da garantire la miglior copertura possibile dello spazio circostante. La seconda shell, posizionata sull'avambraccio (link dal gomito al polso di FRIDA), alloggia 8 sensori sfasati di 45° in un'unica corona circolare.

3.2.1 Tecnologia del sensore

Il tipo di sensore utilizzato nel progetto è un sensore di distanza LED (Light Emitting Diode) prodotto dalla SHARP (modello GP2Y0A21YK). Il funzionamento di questo sensore è basato sull'utilizzo di due diodi: un diodo emettitore, il LED per l'appunto, che emette un fascio di luce infrarossa adeguatamente direzionato da un sistema di lenti, e un fotodiode ricevitore chiamato PSD (Position Sensitive Device) anch'esso schermato da lenti e posizionato in modo tale da ricevere l'eventuale riflessione del fascio infrarosso emesso dal LED, nel caso cui tale fascio intercetti un ostacolo. Il

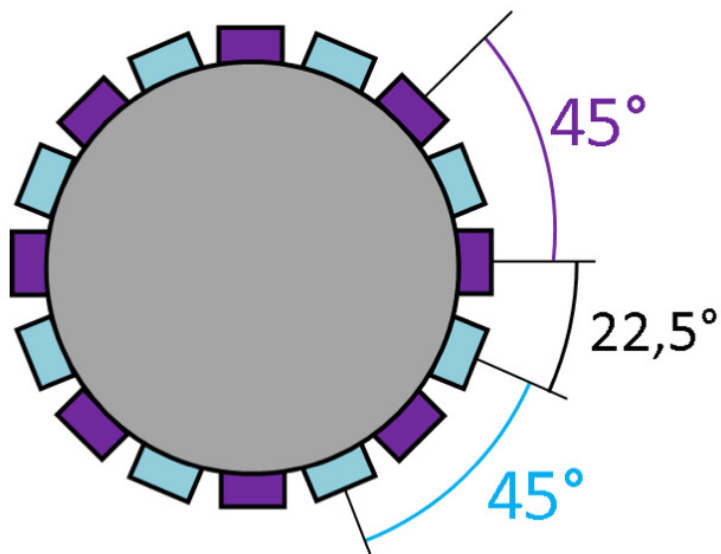


Figura 3.6: Schema posizionamento sensori sulla shell



Figura 3.7: Sensore SHARP GP2Y0A21YK

dispositivo emettitore è un LED ad infrarossi. Questo non è altro che un diodo alimentato da un ingresso in tensione che genera in uscita un'onda elettromagnetica. Appartenendo alla categoria dei diodi, questi LED sono costituiti da una giunzione P-N ottenuta da un opportuno drogaggio del materiale semiconduttore di base. Quando la tensione di polarizzazione fornita al LED è sufficientemente elevata, gli elettroni e le lacune riescono a superare la barriera di potenziale che le separa. Ricombinandosi, essi liberano energia sotto forma di fotoni appartenenti alla regione spettrale degli infrarossi, non

visibili ad occhio umano. Il dispositivo ricevente è invece il PSD. Questo



Figura 3.8: Rappresentazione funzionamento sensore

fotodiodo, se colpito da una radiazione luminosa come quella riflessa da un eventuale ostacolo, libera elettroni che generano una corrente elettrica. Tale corrente, viene ripartita in due correnti I_1 e I_2 che saranno modulate dalla

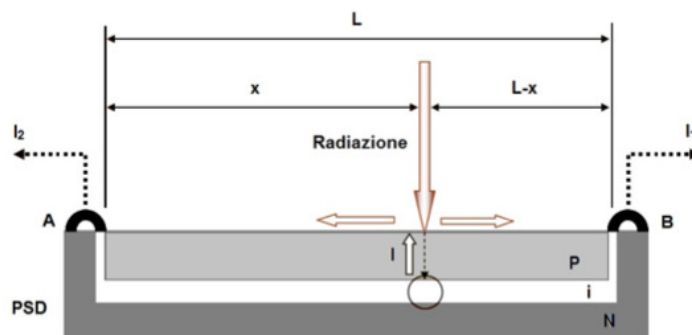


Figura 3.9: Schematizzazione funzionamento PSD

resistenza che si oppone al loro passaggio e che risulta essere proporzionale alle distanze x ed $L - x$ dagli elettrodi posti ai capi del PSD. Le relazioni sono espresse nel seguente sistema di equazioni:

$$\begin{cases} I = I_1 + I_2 \\ I_1 = \frac{L - x}{L} \cdot I \\ I_2 = \frac{x}{L} \cdot I \end{cases}$$

La tensione restituita dal sensore che è ottenuta convertendo opportunamente le due correnti I_1 e I_2 risulta essere proporzionale a x ed inversamente proporzionale alla distanza d dall'oggetto tramite la seguente relazione:

$$d = \frac{f \cdot a}{x}$$

dove f è la posizione delle lenti del sensore e a è la distanza tra il LED e il PSD. Tale relazione è derivata dalla similitudine tra triangoli mostrata in Fig. 3.10. In questa procedura conta solo il punto in cui il fascio infrarosso

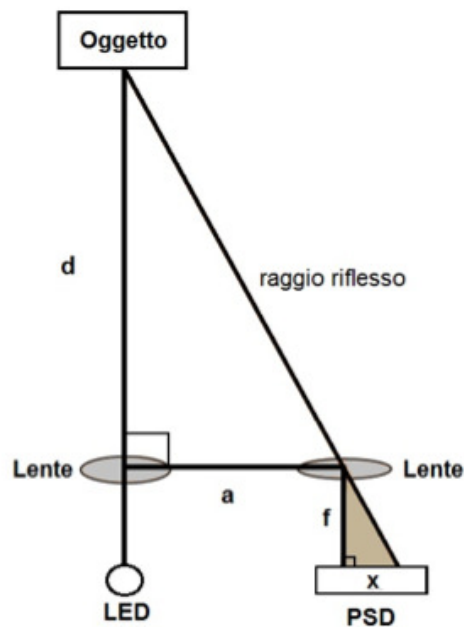


Figura 3.10: Triangolazione

viene riflesso e non l'energia restituita al sensore come radiazione luminosa. Tale caratteristica rende questo tipo di sensore adatto ad ostacoli di materiali differenti, tra cui anche i corpi di eventuali operatori umani. Per evitare eccessive oscillazioni nella misura, dovute a rumore sulla tensione, ogni sensore è stato collegato ad un condensatore da $100\mu F$ che ha la funzione di filtrare il segnale in maniera adeguata. La curva caratteristica che lega la

misura di distanza (espressa in millimetri) e la tensione ai capi del PSD è rappresentata in figura 3.11. Per poter utilizzare la misura della distanza,

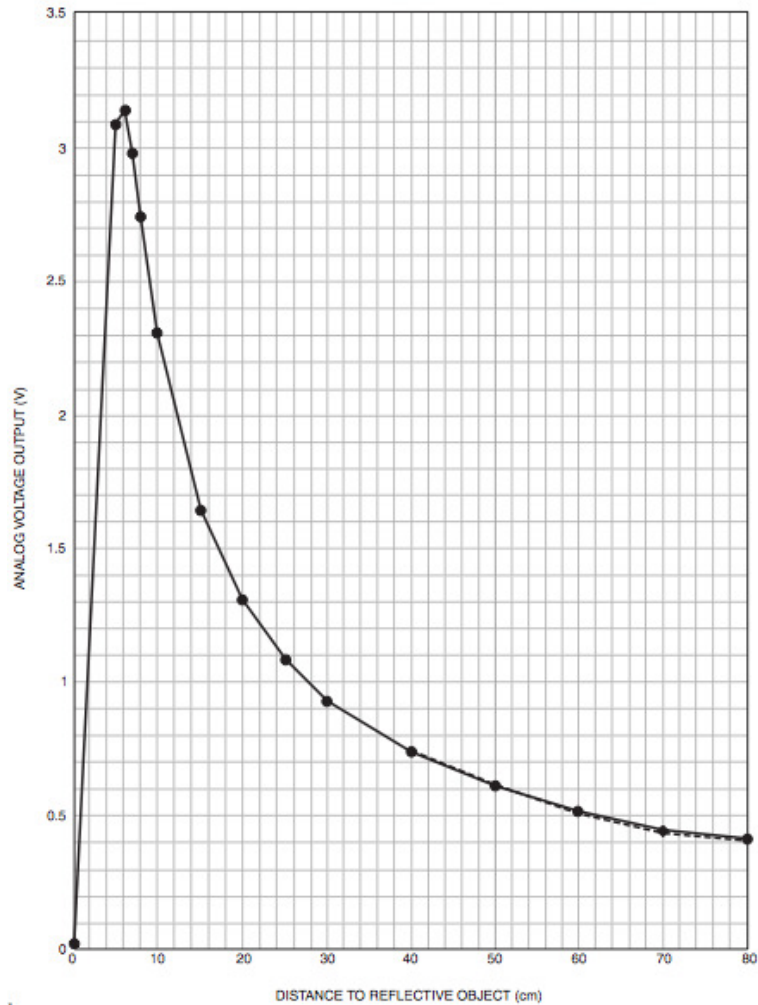


Figura 3.11: Curva caratteristica tensione-distanza

restituita dal sensore sotto forma di tensione, è necessario interpolare la curva caratteristica tramite un apposito polinomio ricavato sperimentalmente in lavori precedenti, di grado 8:

$$\begin{aligned}
 d = & \\
 & = P(1) \cdot v^8 + P(2) \cdot v^7 + P(3) \cdot v^6 + P(4) \cdot v^5 + P(5) \cdot v^4 + \quad (3.1) \\
 & + P(6) \cdot v^3 + P(7) \cdot v^2 + P(8) \cdot v + P(9)
 \end{aligned}$$

3.3 Gripper

Come già espresso, lo scopo ultimo di questo progetto è l'interazione sicura uomo-robot in un ambito industriale. Per rendere gli esperimenti realizzati in laboratorio il più realistici possibile, si è deciso di dotare FRIDA di due gripper della SCHUNK che gli permettessero la manipolazione di componenti di varie geometrie. A causa del limitato payload che FRIDA può sostenere e anche per cercare di limitare l'ingombro dell'attrezzaggio sul robot, la scelta dei gripper è caduta sul modello di gripper radiale MPZ 25 IS e sul gripper parallelo JGP 40-1-IS. Entrambi i gripper sono costruiti in modo da essere

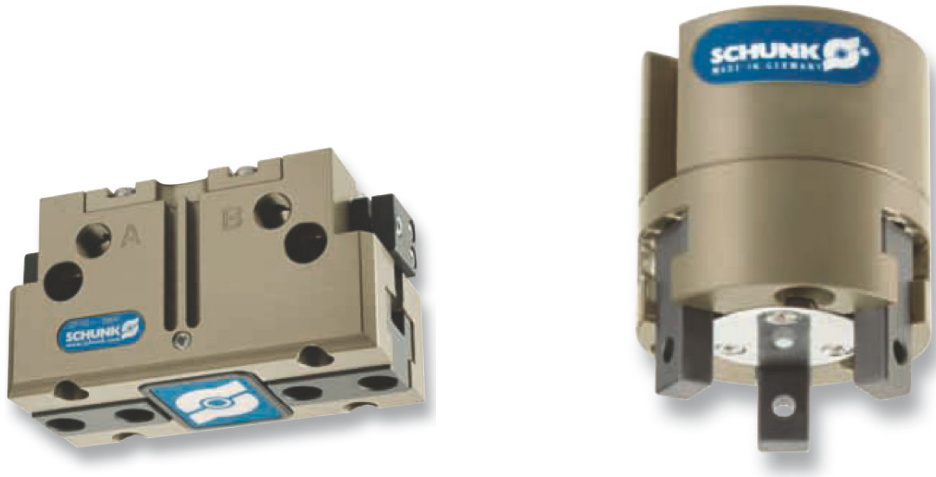


Figura 3.12: Modelli di gripper SHUNK utilizzati nel setup sperimentale (a sinistra MPZ e a destra JGP)

aperti in condizione di riposo, per poi stringersi quando viene immessa al loro interno aria in pressione. Normalmente sarebbe necessario immettere ancora aria in un'altra camera del gripper per permetterne la riapertura, tuttavia la particolarità della serie IS della SCHUNK è quella di avere all'interno del gripper una molla di ritorno la quale, non appena la pressione dell'aria diminuisce, costringe il pistone collegato alle dita del gripper a tornare in posizione di riposo. La serie IS è stata scelta proprio per questa ragione: infatti, poichè FRIDA ha un solo tubo per il trasporto dell'aria in pressione,

l'utilizzo di un normale gripper avrebbe costretto ad installare un ulteriore linea all'esterno del braccio del robot, la quale avrebbe potuto creare intralcio alle movimentazioni possibili.

3.3.1 Adattatori e flange

Essendo prodotti da aziende differenti, non era possibile connettere direttamente i gripper al polso di FRIDA. Si è quindi reso necessario progettare e poi realizzare ex-novo alcuni componenti che ne permettessero l'interconnessione rapida. Oltre a ciò si è deciso anche di utilizzare flange per il cambio rapido del gripper come elemento di raccordo tra l'adattatore sul gripper e quello sul polso. Per poter progettare tutte le componenti necessarie si è utilizzato un software CAD (Solidworks 2013) nella fase di design e la stampa 3D per una rapida prototipazione che potesse verificare direttamente i requisiti del pezzo. Come verrà spiegato in seguito, la presenza della pneumatica necessaria al funzionamento dei gripper non ha permesso di utilizzare i materiali a disposizione della stampante 3D per realizzare anche le componenti definitive, dato che questi non possedevano le adeguate caratteristiche meccaniche di resistenza. Si è quindi optato per una realizzazione manuale dei componenti in delrin, che è un materiale molto più adatto allo scopo.

Adattatori

La connessione del gripper alla flangia di sgancio rapido è effettuata semplicemente per mezzo di un componente che adatta le geometrie dei gripper (uno cilindrico, l'altro a forma di parallelepipedo) alla corona circolare che costituisce una delle due parti della flangia. Il fatto che questi adattatori siano costituiti di un materiale denso come il delrin, ha permesso la filettatura dei fori di fissaggio dei gripper e della flangia.

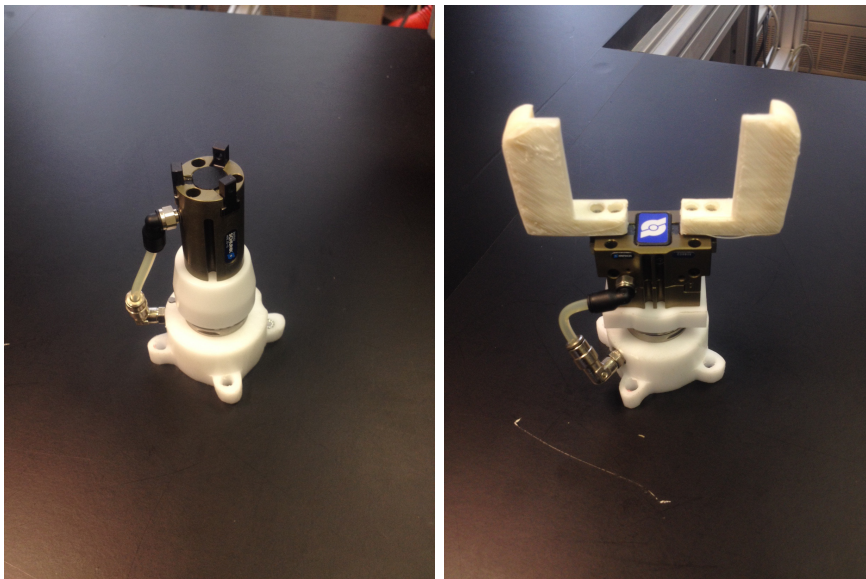


Figura 3.13: Adattatori completi

Il pezzo che connette la seconda parte della flangia al polso del robot è stato invece più complessa nella fase di progettazione: l'uscita pneumatica collocata sul polso di FRIDA infatti non è di misura standard e ciò ha reso impossibile utilizzare i connettori disponibili in commercio. Per ovviare a questo problema si è deciso di far passare il flusso d'aria in pressione direttamente all'interno dell'adattatore polso-flangia, tramite lo scavo di una canalina apposita al suo interno. La presenza di un flusso d'aria in pressione all'interno dell'adattatore (circa 8bar) è stata la principale ragione che non ha consentito l'utilizzo di componenti realizzate con la stampa 3D, oltre all'impossibilità di fissare i gripper e le flange direttamente tramite viti.

Flange

Le flange di sgancio rapido del pezzo non sono altro che due corone circolari in metallo (quattro considerando entrambi i bracci del robot), appositamente conformate per incastrarsi perfettamente tra di loro. Sulla corona circolare più esterna è situata un'ulteriore fascia metallica che, ruotando, spinge in



Figura 3.14: Flange di aggancio rapido dei gripper

direzione radiale e verso l'interno alcuni cilindretti metallici che vanno a fissare l'incastro. Le due coppie di flange installate sugli adattatori sono

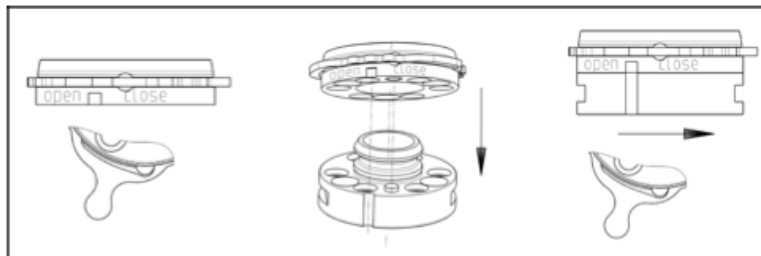


Figura 3.15: Funzionamento fissaggio flange

uguali tra di loro, permettendo in tal modo di intercambiare senza problemi i gripper.

3.3.2 Circuito pneumatico

Il corretto funzionamento dei gripper richiede che l'aria immessa al loro interno si trovi ad una pressione compresa tra i 6 e gli 8 bar. Questa viene

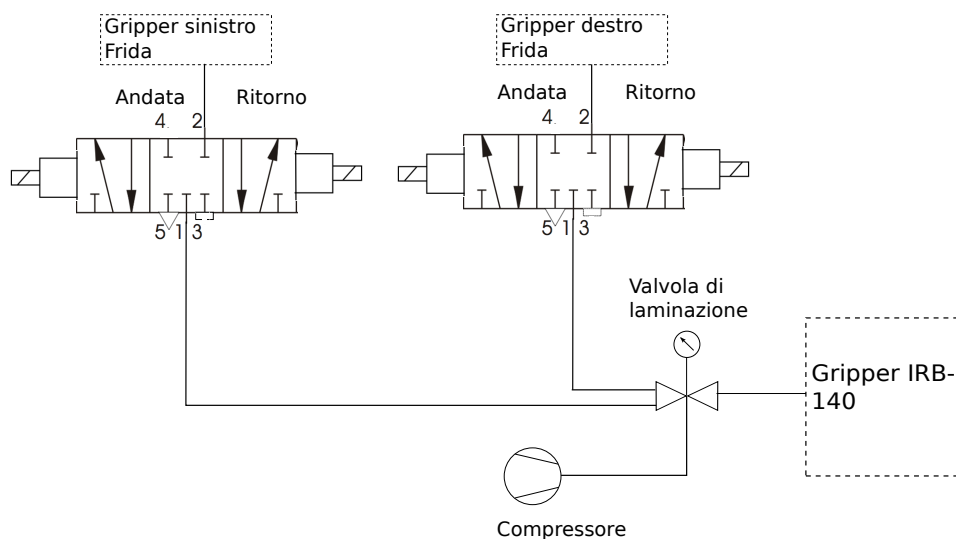


Figura 3.16: Schema del circuito idraulico per l'alimentazione dei gripper SHUNK

fornita dal compressore del laboratorio. Come già accennato, la chiusura e l'apertura dei gripper richiede prima un'immissione di fluido in pressione al loro interno ed in seguito una depressurizzazione, che consenta alla molla di ritorno di riportare il meccanismo in posizione di riposo. Avendo a disposizione solo un tubo che trasporta il fluido, si è quindi reso necessario progettare un circuito pneumatico che consentisse l'inversione del flusso al suo interno attraverso l'attivazione di apposite valvole. La scelta è ricaduta sulle valvole della FESTO già disponibili al Politecnico. Il circuito complessivo comprende l'utilizzo di due elettrovalvole FESTO e di un riduttore di pressione installato a monte di queste ultime (la pressione in arrivo dal compressore è di 10bar, mentre ai gripper sono necessari al massimo 8 bar). Lo schema idraulico è rappresentato nella figura 3.16

Specifiche valvola

Ogni valvola della FESTO è del tipo 5-3, ovvero una valvola che ha 5 possibili ingressi o uscite del fluido e 3 possibili configurazioni nelle connessioni tra questi. Ogni valvola è azionata da due bobine a induzione che, una volta

alimentate a 24V, spingono un pistone che va a modificare le connessioni tra le vie della valvola: La configurazione utilizzata negli esperimenti permette

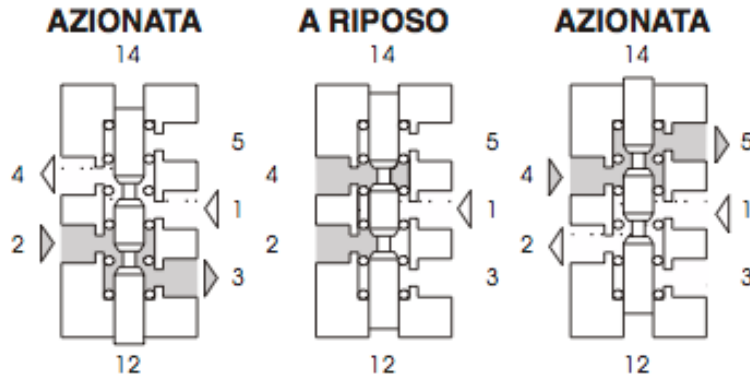


Figura 3.17: Stati possibili della valvola

il passaggio del fluido nella valvola in sole due direzioni. Le connessioni della valvola sono le seguenti:

- la via 1 è quella connessa al compressore del laboratorio;
- la via 4 è la via connessa alla camera del pistone del gripper;
- la via 5 è uno scarico verso l'ambiente esterno attraverso un filtro;
- le restanti due vie non sono di nessuna utilità al setup sperimentale quindi sono chiuse.

In condizione di riposo la via 1 non è connessa a nessun'altra. Quando la prima bobina viene attivata, il movimento del pistone apre la connessione tra le vie 1 e 2 della valvola, permettendo il fluire dell'aria in pressione verso il gripper. Quando la prima bobina viene disattivata e la seconda invece viene attivata (se entrambe le bobine venissero attivate contemporaneamente la configurazione della valvola sarebbe analoga a quella di riposo), la via 4 si connette con la 5, consentendo lo scarico della pressione all'interno del gripper (che sarà chiuso) verso l'ambiente esterno.

3.3.3 Circuito di alimentazione delle valvole

Come già osservato, le bobine di azionamento delle valvole necessitano 24V di tensione costante per essere attivate. La scheda NATIONAL INSTRUMENTS non può fornire più di 5V in output quindi, per poter utilizzare le elettrovalvole in un esperimento, si è reso necessario progettare e realizzare un circuito di potenza per la loro corretta alimentazione. Il circuito non è altro che un insieme di transistor che fungono da interruttori tra i 24V dell'alimentazione e l'uscita, collegata direttamente alle bobine delle valvole. Come si può vedere dallo schema circuitale in fig. 3.18, il circuito di potenza è costantemente alimentato con 24V continui (forniti da un alimentatore esterno). Quando il segnale di un canale in uscita dalla scheda di acquisizione assume il valore di 5V, il relativo transistor si apre, permettendo così il fluire della corrente dall'alimentazione a 24V alla bobina dell'elettrovalvola. La presenza dei diodi posti in connessione antiparallela rispetto ai transistor ha il solo scopo di evitare il reflusso di correnti parassite che potrebbero, a lungo andare, danneggiare i transistor. Sono stati inoltre aggiunti dei led, che consentono la facile visione della corretta attivazione del circuito, accendendosi quando l'uscita è eccitata dai 24V in ingresso. Questo circuito è stato realizzato in modo da poter gestire anche un'altra elettrovalvola, che però viene utilizzata in altri esperimenti, integrando così tutta l'elettronica di potenza dei setup sperimentali nel tavolo di lavoro del laboratorio.

Configurazioni delle valvole

L'attivazione dei singoli canali del circuito di potenza viene gestita tramite il sistema di controllo che si interfaccia con la scheda di acquisizione. Ogni bobina viene controllata indipendentemente dalle altre, permettendo una gestione completamente libera delle manipolazioni compiute dai gripper. Ogni canale in uscita dalla scheda di acquisizione può avere un valore in tensione continua di 0V o 5V. Quando il valore in uscita è alto, il canale corrispon-

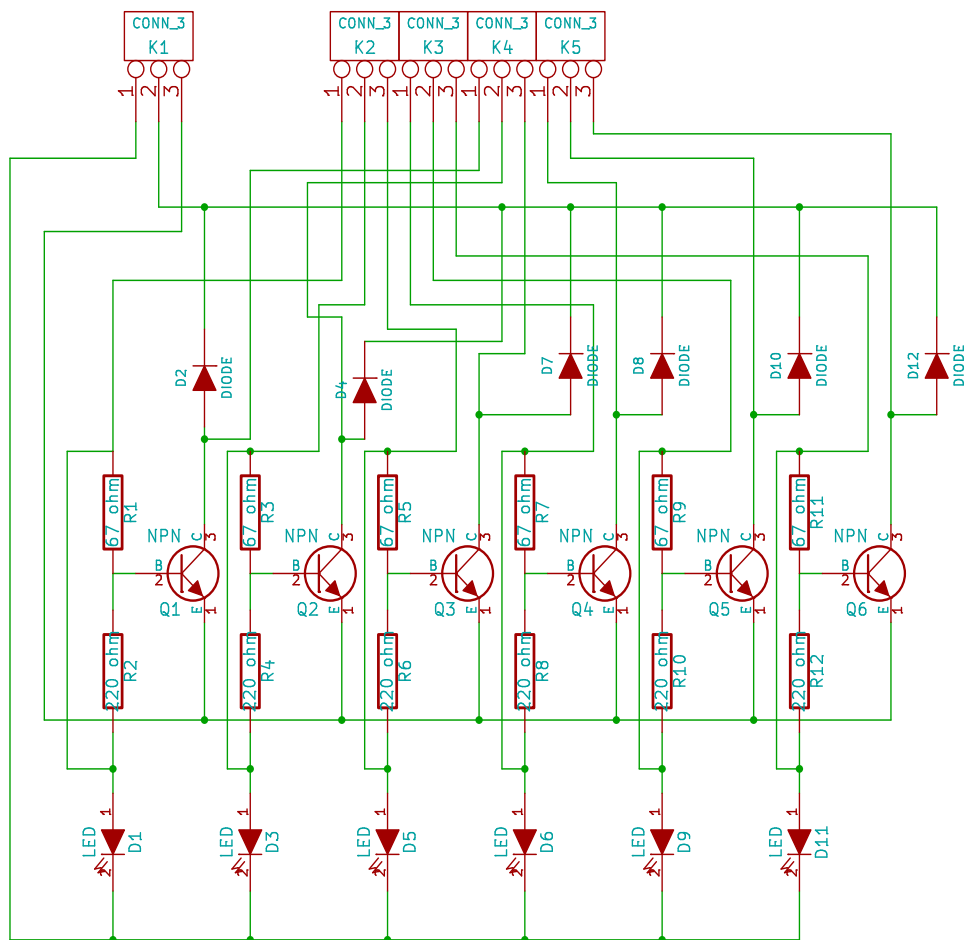


Figura 3.18: Schema circuitale amplificatore

te del circuito di potenza si accende e attiva una delle bobine delle valvole. La tabella dei valori che l'interfaccia di controllo (opcom) assegna ai canali della scheda per gestire le elettrovalvole è la seguente:

Hex	Dec	Note
0x00	0	rest position
0x01	1	Frida left close
0x02	2	Frida left open
0x03	4	Frida right close
0x04	8	Frida right open
0x10	16	IRB140 close
0x20	32	IRB140 open

In tabella sono mostrate le corrispondenze dei valori esadecimali e decimali con le rispettive funzioni. Come già detto, per chiudere un gripper (normalmente aperto) è necessario attivare l'elettrovalvola che fa fluire l'aria dal compressore alla pinza. Per consentire la riapertura è invece necessario lo scarico dell'aria in pressione all'interno del gripper con il comando opposto (es. comando 1 e poi comando 2 che prima chiudono e poi aprono il gripper sinistro di Frida).

Capitolo 4

Strategia di controllo per la collision avoidance nei manipolatori industriali

In questo capitolo si introdurrà dapprima il concetto di sicurezza in relazione alle traiettorie compiute da un manipolatore robotico nello svolgimento dei suoi task; poi verranno discussi alcuni algoritmi che cercano di garantire tale sicurezza senza inficiare la produttività del manipolatore. Sarà infine discusso nel dettaglio l'algoritmo implementato in questo lavoro di tesi e se ne discuteranno alcuni dei possibili metodi risolutivi.

Il grande sviluppo della robotica negli ultimi anni ha permesso a questa tecnologia di diventare sempre più flessibile e perciò di essere applicata in molteplici processi produttivi come la saldatura, l'assemblaggio, la colorazione, la misurazione etc.

Ad oggi tuttavia l'installazione e la progettazione di una cella robotica completamente funzionante richiede l'impiego di molti tecnici ed ingegneri esperti. Per consentire quindi una maggiore diffusione della robotica, bisognerebbe

rendere molto più sicuro il suo utilizzo, soprattutto in ambienti in cui opera anche l'uomo.

In questo senso, si può pensare di incominciare con l'integrazione nei processi produttivi di robot collaborativi di taglia ridotta, sui quali la ricerca ha già permesso l'implementazione di varie tecniche di sicurezza.

Un ulteriore ostacolo al massiccio utilizzo dei robot è dato dalla forzata contrapposizione tra requisiti produttivi e quelli di sicurezza dei lavoratori. L'industria richiederebbe, per massimizzare i profitti, massimizzazione della velocità per minimizzare il tempo di lavoro, robustezza dei manipolatori per evitare rotture da fatica, traiettorie che ottimizzino lo spazio percorso per eseguire le manipolazioni etc. La sicurezza dei lavoratori richiederebbe robot leggeri ed elastici, velocità ridotte, movimentazioni prevedibili dall'operatore umano etc. Le ultime ricerche nel campo della robotica collaborativa sono dunque mirate a trovare una mediazione fra questi due punti di vista opposti.[15]

La movimentazione di un robot è normalmente definita tramite specifiche assolute, come ad esempio la posizione e l'orientamento che l'end-effector deve avere lungo una determinata traiettoria o la velocità che deve tenere. Normalmente queste traiettorie non sono modificabili dai controllori di basso livello (i.e. quelli che regolano i giunti) poiché sono preprogrammate ad un livello di controllo più alto. Con questa logica è però molto difficile garantire un buon comportamento del robot a fronte dell'inserimento di vincoli real-time o della necessità di avere tempi di reazione molto piccoli (ad esempio in caso di pericolo di collisione con un operatore). La soluzione più ovvia a questo problema sarebbe quella di utilizzare la programmazione dinamica con vincoli.

Sono stati sviluppati vari algoritmi per risolvere questo problema anche molto diversi tra loro, ma nessuno di essi è ancora privo di inconvenienti.

Il primo algoritmo è stato il cosiddetto iTaSC (Instantaneous Task Specifica-

tion using Constraints): sviluppato in [33] ed esteso in [10] ma in grado solo di gestire vincoli applicati all'istante di tempo corrente.

Un algoritmo che permette di estendere il problema anche ai vincoli non istantanei è quello descritto in [11].

Altri algoritmi che permettono di specificare vincoli sui task sono quelli descritti in [25] e in [12]. Questi approcci hanno l'inconveniente di non poter esplicitare direttamente una traiettoria, oltre a quello di rilassare i vincoli al fine di mantenere il problema ammissibile.

Sfruttando la stessa strutturazione sono stati sviluppati gli algoritmi in [35] e [20] nei quali è anche possibile definire cifre di costo addizionali.

Un approccio che permette di includere vincoli cinematici sui giunti (posizione, velocità e accelerazione) è quello presentato in [3] e in [13].

Tutti i metodi finora descritti hanno il limite di basarsi esclusivamente su di un'eventuale riscalatura di una traiettoria predefinita. Questo fatto non solo non permette un'adeguata risposta del robot in caso di necessità, ma non garantisce neppure che il problema vincolato ammetta sempre almeno una soluzione.

Nel contesto degli algoritmi di questo tipo è bene citare quelli presentati in [5] e in [22]: questi non consentono comunque di includere le adeguate strategie di controllo, per gestire le deviazioni rispetto alle traiettorie prestabilite. Da questo rapido esame dallo stato dell'arte emergono due difficoltà. La prima è quella di fondere in un'unica struttura vincoli definiti in spazi differenti ad esempio spazio dei giunti e spazio dei task. La seconda è quella di fornire all'algoritmo di generazione della traiettoria, le adeguate capacità di reazione in caso di obbligata deviazione dalla traiettoria nominale. Sarebbe quindi ideale, al fine di garantire il giusto grado di sicurezza, fondere in un'unica struttura globale il problema di generazione della traiettoria con il conseguente problema di controllo vincolato. L'idea di fondo non è nuova ma allo stato attuale non esiste nessun algoritmo di questo tipo adatto ad

un utilizzo industriale.

Come primo passo in questa direzione, si può citare l'Elastic Strips Framework [29], rifinito poi in [8] con l'Elastic Bands Framework. Altri algoritmi real time che vale la pena ricordare sono: quello in [16], dove la reazione desiderata del manipolatore viene ottenuta tramite l'analisi dei contatti con eventuali ostacoli (reali e virtuali), l'algoritmo presentato in [21]; o ancora quello in [17] dove si presenta una metodologia per ripianificare la traiettoria a seguito di una deviazione da quella nominale.

L'obiettivo di questo lavoro di tesi è quindi quello di realizzare e testare un algoritmo di controllo che traduca i vincoli di task durante il funzionamento del robot, in modo da rendere la programmazione delle logiche di controllo a monte molto più snella permettendo così al manipolatore di reagire rapidamente in caso di forzate modifiche dei task nominali (ad esempio in caso di presenza di un ostacolo in movimento).

Il metodo presentato per raggiungere questo scopo è relativamente semplice: legare la pianificazione della traiettoria con una strategia di controllo vincolata di più basso livello. Il task assegnato al robot sarà completamente espresso tramite vincoli sia istantanei (cioè legati allo stato di moto corrente) che non (ovvero vincoli che devono essere considerati per tutti gli istanti temporali anche futuri). In questo modo si potrà assicurare un'adeguata reattività dell'algoritmo semplicemente aggiungendo o rimuovendo vincoli, direttamente a run-time. Di seguito verrà presentato il problema con i relativi vincoli che devono essere rispettati durante l'esecuzione. Questi saranno divisi in tre gruppi:

- vincoli nello spazio dei giunti;
- vincoli nello spazio di lavoro;
- vincoli di safety.

4.1 Struttura del problema

La trattazione che segue riassume una linea di ricerca sulla pianificazione del moto dei manipolatori robotici tramite ottimizzazione vincolata, in corso presso il Politecnico di Milano [26] [32].

Si consideri la cinematica di un generico manipolatore a n-gradi di libertà:

$$\begin{aligned} q_{k+1} &= q_k + T_s \dot{q}_k + 0.5 T_s^2 u_k \\ \dot{q}_{k+1} &= \dot{q}_k + T_s u_k \end{aligned} \quad (4.1)$$

dove:

- q_k è il vettore delle variabili di giunto all'istante k
- \dot{q}_k è la derivata rispetto al tempo di q_k
- T_s è il tempo di campionamento del controllo
- u_k è la derivata seconda rispetto al tempo di q_k

In questo modello si considera una legge di controllo del secondo ordine che ha come input al sistema $\ddot{q}_k = u_k$.

4.2 Vincoli cinematici nello spazio dei giunti

Il sistema descritto in (4.1) deve essere soggetto, in ogni istante di tempo k , a dei vincoli (hard) che limitano la posizione, la velocità e l'accelerazione dei giunti:

$${}_i q^{min} \leq {}_i q \leq {}_i q^{max} \quad |{}_i \dot{q}| \leq {}_i \dot{q}^{max} \quad |{}_i u| \leq {}_i \ddot{q}^{max} \quad (4.2)$$

Restringendo la configurazione dello spazio al massimo insieme invariante positivo ${}_i Q_\infty$ del sistema in (4.1) avente come input quello espresso in (4.2) si può scrivere:

$${}_i q \leq {}_i q^{max} - 0.5 \frac{{}_i \dot{q}^2}{{}_i \ddot{q}^{max}} \quad {}_i q \geq {}_i q^{min} + 0.5 \frac{{}_i \dot{q}^2}{{}_i \ddot{q}^{max}} \quad (4.3)$$

$$|{}_i \dot{q}| \leq {}_i \dot{q}^{max} \quad |{}_i u| \leq {}_i \ddot{q}^{max} \quad (4.4)$$

dove ${}_i Q_\infty$ è la più estesa regione del piano ${}_i q - {}_i \dot{q}$ di modo che ogni punto $({}_i q, {}_i \dot{q})$ ammetta almeno un input ${}_i u$ ammissibile tale per cui il sistema complessivo permanga nella stessa regione [7]. Una rappresentazione grafica di questo insieme invariante per il sistema in (4.1) (per un dato giunto i) è quella in fig 4.1. Si può osservare come, noto lo stato di moto corrente

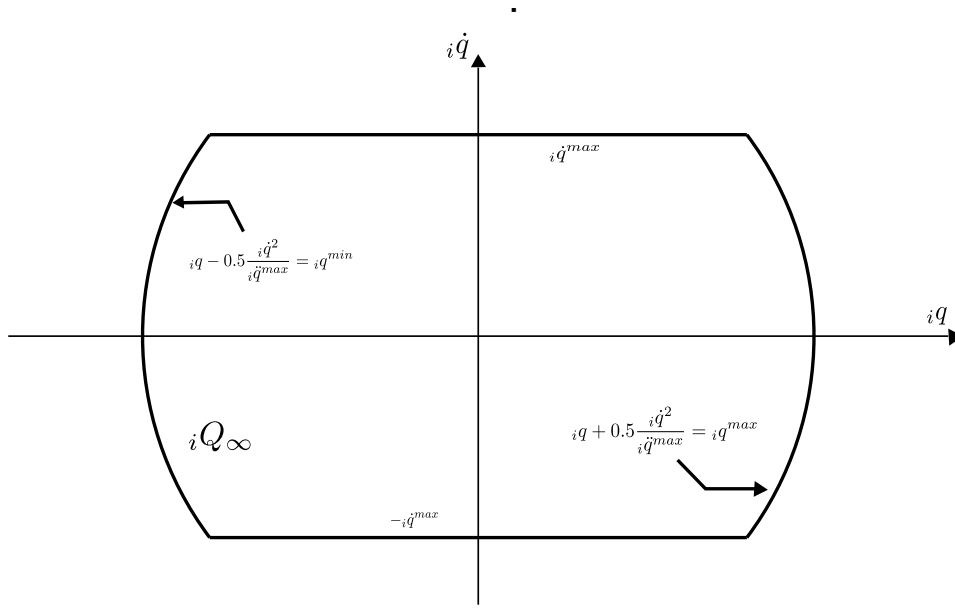


Figura 4.1: Rappresentazione grafica del massimo insieme invariante ${}_i Q_\infty$ per un dato giunto i

$({}_i q, {}_i \dot{q}) \in {}_i Q_\infty$, e note la massima e la minima accelerazione ${}_i u^{max}$ e ${}_i u^{min}$, sia garantito che lo stato di moto all'istante successivo apparterrà sempre alla regione invariante ${}_i Q_\infty$. In forma di equazioni si può scrivere:

$$-{}_i \ddot{q}^{max} \leq {}_i u_{inf} \leq {}_i u \leq {}_i u^{sup} \leq {}_i \ddot{q}^{max} \quad (4.5)$$

Per ogni istante discreto di tempo k la legge di controllo deve quindi produrre un input al sistema u_k che rispetti il vincolo espresso in (4.5).

4.3 Vincoli nello spazio di lavoro e generazione della traiettoria

Volendo definire dei vincoli anche nello spazio di lavoro è prima necessario definire un vettore x che esprima m variabili di lavoro (coordinate cartesiane, angoli di eulero[33] etc.). Dato x , come è noto, le relazioni che intercorrono tra x e il corrispondente spazio dei giunti sono:

$$x = f(q) \quad \dot{x} = J(q)\dot{q} \quad \ddot{x} = \dot{J}(q)\dot{q} + J(q)u \quad (4.6)$$

dove J è la matrice jacobiana dei task di dimensione $m \times n$, che mette in relazione istantaneamente lo spazio dei giunti a quello di lavoro. Si può notare che aggiungendo l'equazione d'uscita $x = f(q)$ al sistema in (4.1), questo abbia un grado relativo due. Da qui si rende necessaria una differenziazione del secondo ordine, per poter esplicitare la dipendenza di x dalla variabile di controllo u . A tale scopo, utilizzando una struttura simile a quella di (4.1), si può scrivere:

$$\begin{aligned} x_{k+1} &= x_k + T_s J_k \dot{q}_k + 0.5 T_s^2 (\dot{J}_k \dot{q}_k + J_k u_k) \\ \dot{x}_{k+1} &= J_k \dot{q}_k + T_s (\dot{J}_k \dot{q}_k + J_k u_k) \end{aligned} \quad (4.7)$$

dove: $J_k = J(q_k)$ e ovviamente $\dot{J}_k = \dot{J}(q_k)$.

Similmente a quanto fatto nello spazio dei giunti, si applicano i seguenti vincoli:

$$|{}_i \dot{x}| \leq {}_i \dot{x}^{max} \quad |{}_i \ddot{x}| \leq {}_i \ddot{x}^{max} \quad (4.8)$$

in cui ${}_i \dot{x}$ e ${}_i \ddot{x}$ sono i vettori dei limiti di velocità e di accelerazione massima nello spazio di lavoro.

Come già accennato, sono disponibili vari algoritmi per generare una traiettoria da un dato stato di moto (x_k, \dot{x}_k) ad uno di riferimento (x^{trg}, \dot{x}^{trg}) . In questo lavoro di tesi ne verrà utilizzato uno che restituirà, ad ogni istante di tempo k , la traiettoria in termini di posizione e velocità desiderati all'istante $k + 1$, ovvero (x_{k+1}, \dot{x}_{k+1}) .

4.4 Vincoli di safety

Come già detto, l'algoritmo presentato in questo progetto di tesi ha lo scopo di generare un'adeguata reazione del robot, nel caso in cui un ostacolo si ponesse nel suo spazio di lavoro durante l'esecuzione di un task. L'algoritmo farà allora deviare il robot dalla traiettoria nominale restituita dal generatore di traiettoria, in modo da evitare la collisione con l'ostacolo in questione. Per poter rilevare un ostacolo, è necessario utilizzare dei sensori che ne forniscano accuratamente la posizione al controllore, così da poter pianificare la manovra evasiva più adatta.

Da un punto di vista industriale, una deviazione dalla traiettoria nominale (supposta ottima per percorso e velocità) corrisponde ad una perdita di produttività, che a sua volta si traduce in una perdita economica. Bisogna quindi trovare un compromesso tra il requisito industriale che spingerebbe il robot a muoversi nella maniera più rapida possibile e il requisito di safety. Una rappresentazione grafica di questo concetto è quella in figura 4.2. In

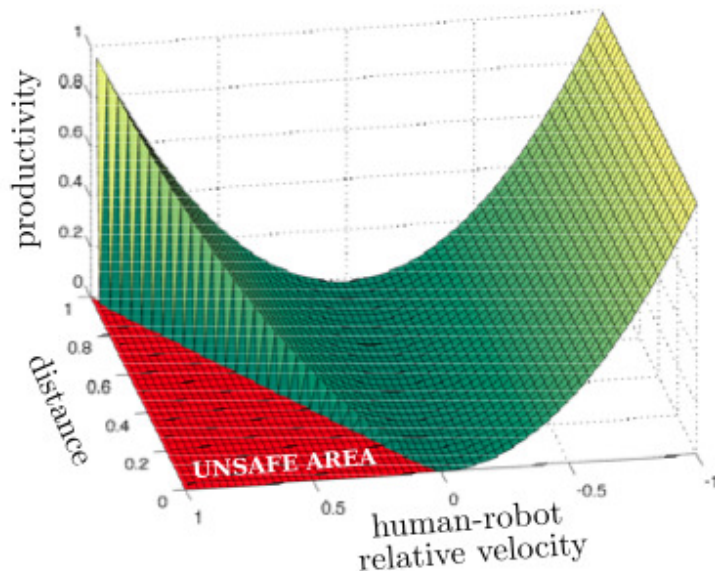


Figura 4.2: Rappresentazione della relazione tra produttività e sicurezza

questo tipo di scenario, al diminuire della distanza tra l'umano e il robot, la componente di velocità del robot, orientata verso l'ostacolo, dovrebbe diminuire in accordo con il criterio della distanza minima di separazione[2]. Questo va ovviamente ad influire negativamente sulla produttività del robot. Per esprimere questa relazione ed inserirla nell'algoritmo, è necessario imporre un vincolo hard di questo tipo:

$$\text{distanza} \geq \text{tempo di arresto} \cdot \text{velocità} \quad (4.9)$$

Nell'equazione in (4.9) la distanza è intesa come la misura della distanza che separa il braccio del robot dall'ostacolo, a cui viene sottratto un valore di sicurezza ($\text{distanza} = \text{distanza reale} - \text{valore di sicurezza}$), mentre il tempo di arresto è quello massimo del robot ad un determinato carico.

Per definire se una distanza da un ostacolo sia sicura o meno, è innanzitutto necessario definire cosa si intende per distanza da un ostacolo. Considerando

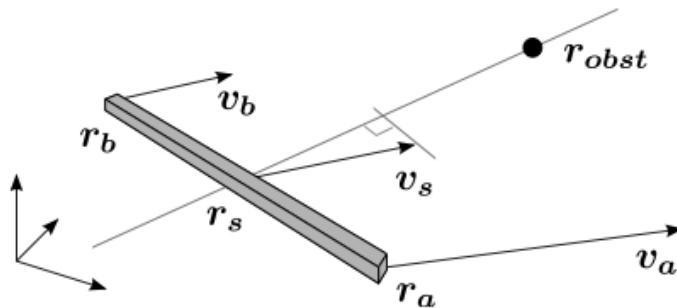


Figura 4.3: Rappresentazione di un link del robot e di un punto di ostacolo

la figura 4.3 si rappresenta il robot come una serie di link rigidi, raffigurati come segmenti (in figura ne viene mostrato uno solo). La posizione r_s di ogni punto del link e relativa velocità v_s possono essere espresse completamente in relazione agli estremi del link:

$$r_s = r_a + s(r_b - r_a) \quad v_s = v_a + s(v_b - v_a) \quad (4.10)$$

con $s \in [0, 1]$. Ora, nota la posizione nello spazio di un dato ostacolo r_{obst} , il criterio di minima distanza di separazione può essere espresso come segue:

$$v_s^T \frac{r_{obst} - r_s}{\|r_{obst} - r_s\|} T_b \leq \max(0, \|r_{obst} - r_s\|) - \Delta \quad (4.11)$$

In questa equazione Δ è il valore di sicurezza (rappresenta la distanza minima che si vuole che il robot tenga dall'ostacolo).

Rielaborando il vincolo si ottiene:

$$v_s^T (r_{obst} - r_s) T_b \leq \max(0, \|r_{obst} - r_s\|^2 - \Delta \|r_{obst} - r_s\|) \quad (4.12)$$

dove T_b è il tempo di frenata. Assumendo la parametrizzazione del link in termini di posizione e velocità come definito in (4.10), per imporre il vincolo di safety è necessario che (4.12) sia soddisfatta per ogni $s \in [0, 1]$; quindi il membro di sinistra diviene:

$$\begin{aligned} v_s^T (r_{obst} - r_s) &= v_a^T (r_{obst} - r_a) - s(v_b - v_a)^T (r_{obst} - r_a) \\ &\quad + s v_a^T (r_b - r_a) + s^2 \underbrace{(v_b - v_a)^T (r_b - r_a)}_{=0} \end{aligned} \quad (4.13)$$

Mentre per il membro destro si può notare che:

$$\max(0, \|r_{obst} - r_s\|) - \Delta \leq \max(0, \|r_{obst} - r_s\|^2 - \Delta \|r_{obst} - r_s\|) \quad (4.14)$$

Da ciò risulta che è possibile riscrivere il gruppo di disuguaglianze che esprimono il vincolo di safety in questo modo:

$$\alpha + \beta s \leq g(s) \quad \forall s \in [0, 1] \quad (4.15)$$

con:

- $\alpha = T_b v_a^T (r_{obst} - r_a)$
- $\beta = T_b (v_b - v_a)^T (r_{obst} - r_a) - T_b v_a^T (r_b - r_a)$
- $g(s) = \max[0, \|r_{obst} - r_s\|] - \Delta]^2$

Dato che il membro a sinistra è una funzione lineare in s , è possibile definire come condizione sufficiente per il vincolo di safety:

$$\max[\alpha, \alpha + \beta] \leq \min_s g(s) \quad (4.16)$$

da cui, scambiando il \min con il \max :

$$\min_s g(s) = \max[(0, \min_s \| r_{obst} - r_s \|) - \Delta]^2 \quad (4.17)$$

Riassumendo, il criterio della minima distanza di separazione può essere espresso nella forma:

$$T_b E \dot{q}_k \leq b_k \quad (4.18)$$

che è nella stessa forma di(4.9) e dove[39]:

$$E = \begin{bmatrix} (r_{obst} - r_a)^T J_a \\ (r_{obst} - r_a)^T J_b - (r_b - r_a)^T J_a \end{bmatrix} \quad (4.19)$$

$$b = \min_s g(s) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

con J_a e J_b jacobiani di posizione degli estremi del link.

Come già osservato in precedenza, il sistema in (4.1) è controllato con un input $\ddot{q}_k = u_k$, quindi bisogna esprimere il vincolo appena ricavato in modo che compaia u_k . Per fare ciò è necessario che sia E che f siano continue e derivabili in ogni istante k . Si vedrà in fase sperimentale che questa ipotesi non sempre è verificata. Applicando il lemma di Gronnwall [14] si ottiene:

$$T_b E_k \ddot{q}_k \leq -\alpha T_b E_k \dot{q}_k + \alpha b_k + \frac{\Delta b_k}{T_s} - T_b \frac{\Delta E_k}{T_s} \dot{q}_k \quad (4.20)$$

Riscritto in questa forma il vincolo può essere utilizzato nel problema di controllo ponendo $\alpha = \frac{1}{T_s}$. Nel problema studiato in questo lavoro di tesi, il vincolo è introdotto nel sistema appositamente per far fronte ad eventi rilevati dai sensori esterni al robot tramite un'adeguata scelta di u_k . Normalmente tali eventi avvengono all'istante discreto k e devono essere gestiti in $k + 1$.

4.5 Schema di controllo

Lo schema a blocchi che descrive il funzionamento dell'algorithmo di controllo è quello di figura 4.4. Questo schema comprende tre differenti moduli:

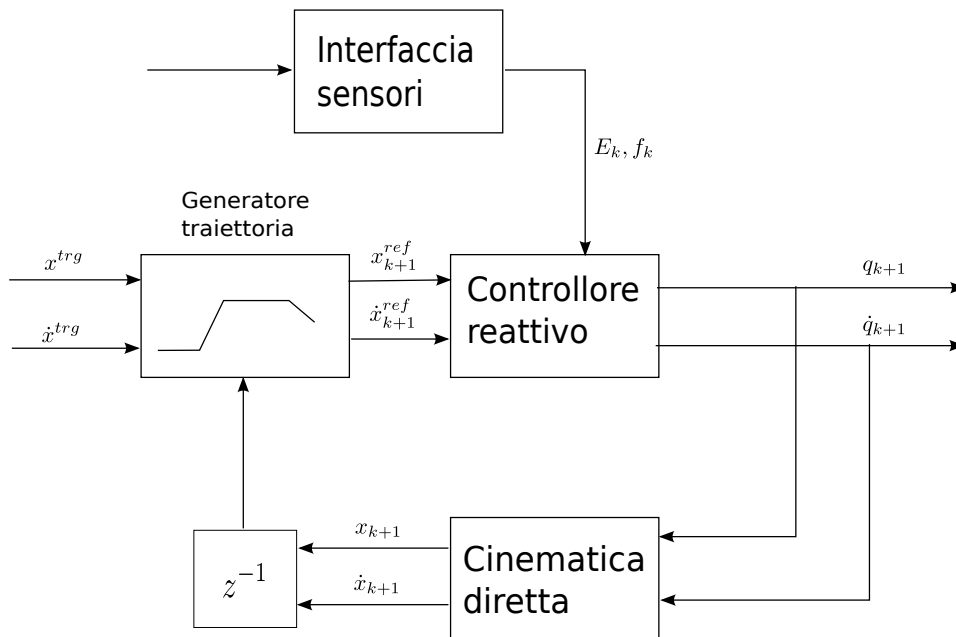


Figura 4.4: Schema a blocchi dell'algorithmo di controllo

- la generazione della traiettoria;
- il controllore reattivo;
- l'interfaccia dei sensori.

4.5.1 Generazione della traiettoria

Il blocco implementa uno qualsiasi degli algoritmi in [5]. Il modulo conosce tutti i vincoli di task, i riferimenti per il moto (x^{trg} , \dot{x}^{trg}) nonché i vincoli di velocità e di accelerazione (\dot{x}^{max} , \ddot{x}^{max}), ma ignora tutti i vincoli nello spazio dei giunti, così come i vincoli di safety che sono basati sulle informazioni trasmesse dai sensori. L'output di questo blocco è lo stato di moto (x_{k+1}^{ref} , \dot{x}_{k+1}^{ref})

nello spazio operativo che può essere raggiunto dal robot nell'intervallo di tempo discreto T_s .

4.5.2 Controllore reattivo

L'informazione restituita dal generatore della traiettoria è propagata al controllore reattivo, il quale ha il compito di cercare di seguire il riferimento di traiettoria, rispettando i vincoli di safety espressi in (4.20). Il segnale di riferimento potrebbe non essere accuratamente seguito proprio a causa di tali vincoli (la traiettoria sarebbe quindi sub-ottima).

4.5.3 Cinematica diretta

Tramite questo blocco vengono valutate le relazioni in (4.6). In tal modo si calcolerà lo stato di moto nello spazio di stato (x_{k+1}, \dot{x}_{k+1}) che sarà utilizzato come input al blocco di generazione della traiettoria al successivo ciclo di controllo per generare una traiettoria aggiornata. Questi passaggi sono raffigurati in figura 4.5.

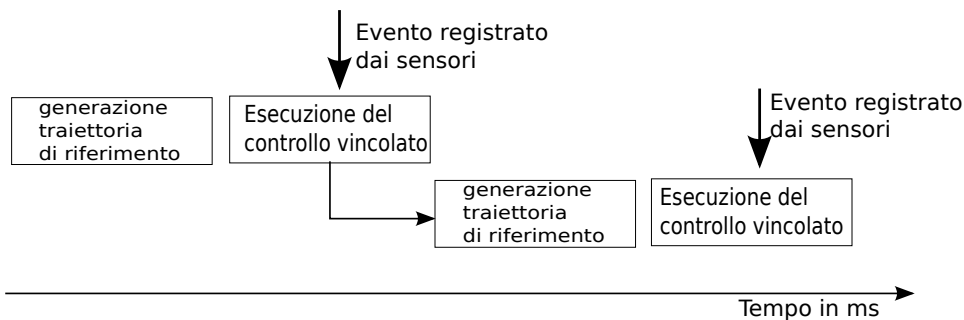


Figura 4.5: Esecuzione dell'algoritmo di controllo reattivo e generazione della traiettoria

4.5.4 Interfaccia sensori

Questo modulo ha la funzione di elaborare in real time tutti i dati inviati dai sensori per poter calcolare la matrice E_k ed il vettore f_k , necessari a definire

il vincolo di safety. Queste informazioni vengono poi passate al controllore reattivo.

4.6 Algoritmo di controllo

Sarà ora dettagliato l'algoritmo utilizzato per implementare il controllo reattivo appena descritto ed in particolare come questo:

- trasforma le variabili nello spazio di stato in corrispondenti comandi ai giunti tramite la cinematica inversa;
- gestisce vincoli di safety basati sulle misurazioni dei sensori;
- segue la traiettoria di riferimento;
- gestisce un'eventuale ridondanza cinematica nel manipolatore.

4.6.1 Algoritmo punto-punto con gradi di libertà ridondanti

Input: $q_k, \dot{q}_k, x_{k+1}^{ref}, \dot{x}_{k+1}^{ref}, \dot{x}^{max}, \ddot{x}^{max}, E_k, f_k$

Output: q_{k+1}, \dot{q}_{k+1}

L'algoritmo:

1. Calcola le quantità cinematicamente rilevanti quali jacobiani, variabili nello spazio di lavoro etc.
2. valuta i limiti di accelerazione u_{inf}, u^{sup}

3. risolve il seguente problema di ottimizzazione quadratica:

$$\min_{u_k} \left(\left\| \dot{x}_{k+1} - \dot{x}_{k+1}^{ref} \right\|_{Q_v}^2 + \left\| x_{k+1} - x_{k+1}^{ref} \right\|_{Q_p}^2 \right) \quad (4.21a)$$

$$x_{k+1} = x_k + T_s J_k \dot{q}_k + \frac{T_s^2}{2} (\dot{J}_k \dot{q}_k + J_k u_k) \quad (4.21b)$$

$$\dot{x}_{k+1} = J_k \dot{q}_k + T_s (\dot{J}_k \dot{q}_k + J_k u_k) \quad (4.21c)$$

$$u_{inf} \leq u_k \leq u^{sup} \quad (4.21d)$$

$$-\dot{x}^{max} \leq \dot{x}_{k+1} \leq \dot{x}^{max} \quad (4.21e)$$

$$-\ddot{x}^{max} \leq \dot{J}_k \dot{q}_k + J_k u_k \leq \ddot{x}^{max} \quad (4.21f)$$

$$E_k u_k \leq f_k \quad (4.21g)$$

4. aggiorna lo stato di moto come in (4.1)

$$q_{k+1} = q_k + T_s \dot{q}_k + 0.5 T_s^2 u_k \quad (4.22)$$

$$\dot{q}_{k+1} = \dot{q}_k + T_s u_k$$

La cifra di costo in (4.21a) ha la funzione di pesare la differenza tra lo stato di moto all'istante successivo (x_{k+1}, \dot{x}_{k+1}) e il suo valore di riferimento $(x_{k+1}^{ref}, \dot{x}_{k+1}^{ref})$. L'errore di inseguimento della traiettoria restituita dal generatore di traiettoria è diviso nelle due componenti di posizione e di velocità opportunamente pesate da Q_p e Q_v .

I vincoli espressi in (4.21b) e in (4.21c) mettono in correlazione le variabili nello spazio dei giunti con quelle nello spazio cartesiano di lavoro.

Il vincolo in (4.21d) serve a mantenere il successivo stato di moto nell'insieme invariante Q_∞ rappresentato in fig. 4.1.

Le equazioni in (4.21e) e in (4.21f) esprimono i vincoli sulla velocità e sull'accelerazione delle variabili nello spazio di lavoro.

Come già detto, la (4.24f) rappresenta infine il vincolo istantaneo dato dalle rilevazioni dei sensori. [12]

4.7 Metodi risolutivi del problema di controllo in caso di robot ridondante

Nel caso in cui il robot a cui viene applicato l'algoritmo sia cinematicamente ridondante (come nel caso del robot ABB FRIDA utilizzato in questo progetto di tesi) o qualora il task che questo deve eseguire lo sia, la cifra di merito espressa in (4.21a) non è più definita positiva. Questo fa sì che il problema di ottimizzazione in (4.21) ammetta infinite soluzioni, tutte parimenti ottime. Ciò è dovuto al fatto che il numero di gradi di libertà n è maggiore della dimensione dello spazio del task m . Si introduce allora, ponendo u_k^0 come un output del problema di ottimizzazione, un ulteriore vincolo di ottimalità che dovrà essere soddisfatto da tutte le u_k :

$$J_k u_k = J_k u_k^0 \quad (4.23)$$

In questo modo è possibile ottenere che ogni soluzione u_k generata dal problema di ottimizzazione che differisce da u_k^0 , lo faccia solo nello spazio nullo dello Jacobiano del task [19]. Al fine di includere nell'algoritmo di controllo anche dei requisiti secondari, è quindi conveniente sfruttare questa ridondanza. Si può procedere in tre modalità differenti:

- si aggiunge un secondo problema di ottimizzazione in cascata;
- si proietta nello spazio nullo;
- si utilizza un solutore ad hoc.

4.7.1 Problema di ottimizzazione secondario in cascata

Il secondo problema in cascata che dovrà necessariamente tenere conto di tutti i vincoli di quello precedente con l'aggiunta di quello di ottimalità

espresso in (4.24b) è esposto di seguito:

$$\min_{u_k} \frac{1}{2} u_k^T Q_u u_k + g^T u_k \quad (4.24a)$$

$$J_k u_k = J_k u_k^0 \quad (4.24b)$$

$$u_{inf} \leq u_k \leq u^{sup} \quad (4.24c)$$

$$-\dot{x}^{max} \leq \dot{x}_{k+1} \leq \dot{x}^{max} \quad (4.24d)$$

$$-\ddot{x}^{max} \leq \dot{J}_k \dot{q}_k + J_k u_k \leq \ddot{x}^{max} \quad (4.24e)$$

$$E_k u_k \leq f_k \quad (4.24f)$$

La cifra di merito espressa in (4.24a) ha lo scopo di gestire la ridondanza cinematica del robot, cercando nello specifico di minimizzare la velocità di quest'ultimo. In questa Q_u e g sono opportuni parametri di progetto. Si aggiorna poi lo stato esattamente come espresso in (4.22).

4.7.2 Proiezione nello spazio nullo

Trovandosi in una situazione di ridondanza cinematica (nello specifico utilizzando il robot FRIDA) è possibile muoversi nello spazio nullo, ossia è possibile compiere dei movimenti senza inficiare il rispetto dei vincoli del problema di ottimizzazione.

Questo si può ottenere tramite l'utilizzo della cifra di merito in (4.24a) con l'aggiunta del vincolo di ottimalità già citato in (4.24b).

Dato che FRIDA ha un solo grado di ridondanza cinematica, questo problema di ottimizzazione può essere risolto in forma chiusa ponendo:

$$J \ddot{q}_k = J \dot{q}_k^0 = n_k^0 \quad (4.25)$$

dopo avere definito la variabile n_k^0 si procede all'espressione di \ddot{q} in funzione di questa e dell'accelerazione nello spazio nullo $\ddot{\alpha}_k$ (questa è uno scalare), ottenendo:

$$\ddot{q}_k = J^+ n_k^0 + N \ddot{\alpha}_k \quad (4.26)$$

con J^+ pseudo inversa destra di J ed N una base ortonormale dello spazio nullo di J . Andando poi a sostituire la (4.26) in (4.24a) si otterrà una cifra di costo da minimizzare dipendente dalla nuova variabile $\ddot{\alpha}_k$ della forma:

$$\min_{\ddot{\alpha}_k} \frac{1}{2} \ddot{\alpha}_k^T V \ddot{\alpha}_k + f^T \ddot{\alpha}_k \quad (4.27)$$

con V ed f pesi opportuni.

Essendo una funzione scalare, per trovare il minimo è sufficiente derivare rispetto ad $\ddot{\alpha}_k$ ottenendo:

$$\ddot{\alpha}_k = -V^{-1} f \quad (4.28)$$

questo sarà il valore di accelerazione nello spazio nullo che minimizza (4.24a). L'accelerazione trovata in questo modo, andandosi ad aggiungere con le altre già calcolate dal primo livello di ottimizzazione, non va a modificare in alcun modo la posizione e l'orientamento dell'end effector, permettendo quindi l'esecuzione di manovre evasive senza andare ad inficiare il raggiungimento dell'obiettivo del task.

4.7.3 Solutore dedicato

Un ulteriore metodo per gestire la ridondanza cinematica del robot è quello basato sulla propagazione gerarchica in cascata di più problemi di ottimizzazione. Nel progetto di questa tesi si è utilizzato un solutore dedicato chiamato SOTH [38]. La sua particolarità è quella di risolvere uno Hierarchical Quadratic Problem (HQP) invece della cascata di problemi di ottimizzazione che si avrebbe utilizzando il primo metodo. In pratica, questo solutore permette l'espressione del problema di ottimizzazione completo, inclusivo della funzione di costo in(4.24a) sotto forma di insieme di vincoli (che possono essere di uguaglianza, disuguaglianza o inclusione). Il problema che sarà risolto sarà

quindi della forma:

$$\begin{aligned}l_A\{1\} &\leq A\{1\}x \leq u_A\{1\} \\ A\{2\}x &= u_A\{2\} \\ &\vdots \\ A\{n-1\}x &\geq l_A\{n-1\} \\ A\{n\}x &= u_A\{n\}\end{aligned}\tag{4.29}$$

dove x rappresenta la generica variabile di ottimizzazione, A una generica matrice di coefficienti di x . u_A e l_A sono invece i due vettori colonna che rappresentano i limiti superiori ed inferiori del problema di ottimizzazione vincolata.

Nel capitolo 5, verrà presentata la riformulazione dei vincoli in (4.21) e (4.24) in modo che questi siano scritti nella stessa forma di (4.29).

Ogni riga in (4.29) rappresenta un set di vincoli appartenenti ad una determinata tipologia. Il problema vincolato completo viene poi diviso ulteriormente in gruppi di vincoli che si vuole abbiano lo stesso livello di priorità nella soluzione, ovvero i vincoli con priorità più alta saranno ottimizzati certamente (se esiste una soluzione ammissibile), mentre i vincoli appartenenti ad un livello gerarchico inferiore verranno ottimizzati rispetto allo spazio nullo delle soluzioni di quelli gerarchicamente superiori. Questa procedura permette una computazione molto efficiente e per questo motivo viene utilizzata (come sarà spiegato in seguito) in questo progetto di tesi.

Capitolo 5

Realizzazione e validazione del sistema di controllo

In questo capitolo sarà dapprima mostrato lo schema generale con il quale si è implementato l'algoritmo di controllo. Ci si soffermerà poi nell'illustrare i singoli blocchi che lo compongono, ponendo particolare attenzione alla macchina a stati, che contiene l'algoritmo presentato nel capitolo 3. Si andranno poi ad illustrare e commentare i risultati ottenuti dalle simulazioni e dalle prove reali effettuate su FRIDA.

Per poter utilizzare un algoritmo di controllo differente da quelli precompilati su FRIDA, è necessario che quest'ultimo si interfacci con un computer esterno nel quale si trova l'eseguibile che contiene le istruzioni da svolgere.

L'interfaccia tra il robot ed il computer è costituita da un'apposita patch del sistema operativo Linux denominata Xenomai, che garantisce una connessione real-time con il robot. Tramite tale interfaccia si garantisce che l'Axis computer del robot riceva i riferimenti calcolati dallo schema di controllo esattamente ogni $4ms$.

Il linguaggio utilizzato per implementare l'algoritmo di controllo è stato quello di Matlab-Simulink. Questa scelta è stata fatta per via della capacità di tale ambiente di consentire la scrittura di algoritmi complessi in manie-

ra relativamente semplice, nonché della possibilità di sviluppare simulatori complessi per testarli. Lo schema così generato è stato poi compilato in linguaggio C dalla apposita toolbox di Matlab, Real Time Workshop.

L'esecuzione del file C così generato e la gestione della comunicazione tra l'unità di governo ed il PC esterno vengono coordinate dal programma apposito OPCOM. A seconda della modalità in cui si trova OPCOM, vengono aperte o chiuse le connessioni tra il computer principale del robot, l'axis computer ed il controllore esterno, come mostrato in fig 5.1. Le modalità disponibili

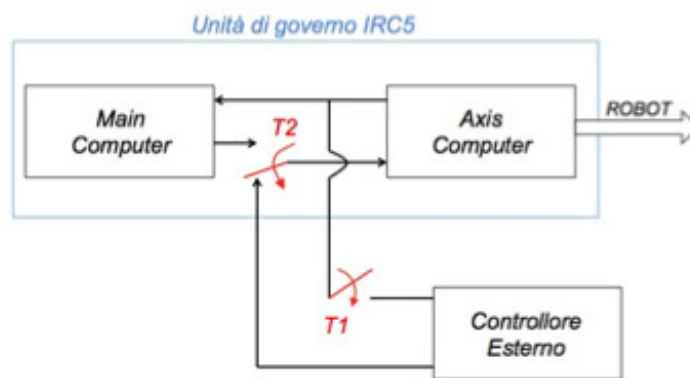


Figura 5.1: Schematizzazione dell'unità di governo con controllore esterno

sono quattro:

- **Modalità unload:** nessun programma è stato caricato.
- **Modalità load:** il codice compilato viene caricato e controllato, ma non si ha ancora nessuna comunicazione tra le due unità;
- **Modalità submit:** viene chiuso l'interruttore $T1$, il PC esterno è in grado di leggere i dati forniti dall'axis computer, ma non è ancora in grado di assegnare alcun riferimento. I riferimenti sono ancora gestiti dal main computer;

- **Modalità obtain:** viene chiuso l'interruttore $T2$, il PC esterno è in grado di assegnare i riferimenti direttamente all'axis computer e di leggere i dati da esso forniti. Ora l'axis computer viene comandato completamente dal PC esterno e non più dal main computer.

5.1 Macchina a stati

Il controllo è implementato tramite una macchina a stati, mostrata in fig.5.2, la quale suddivide il task che deve compiere il robot in più Skills. In figura

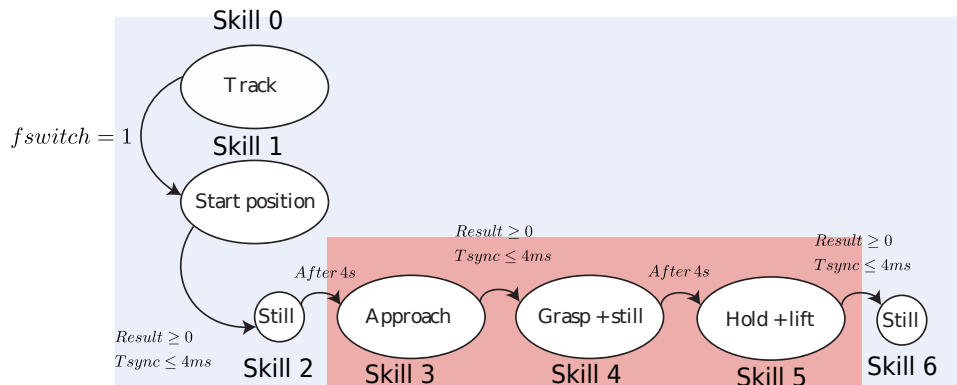


Figura 5.2: Schema macchina a stati

sono anche mostrate le condizioni di passaggio da una skill ad un'altra, che differiscono a seconda della tipologia:

- Dalla skill 0 alla skill 1 si passa quando $fswitch = 1$, questo parametro è cambiato manualmente dall'utente via OPCOM.
- Dalle skill di tipo still si esce dopo 4s passati in tale stato.
- La condizione d'uscita dalle skill di movimento è invece dipendente dal grado di completamento della traiettoria, ovvero se $result \geq 0$ significa che si è raggiunto il punto finale della skill, se $Tsync \leq 4ms$ significa invece che al completamento della skill, e quindi al raggiungimento del goal, manca al massimo un tempo di ciclo.

L'algoritmo di controllo sarà attivo solo in alcune skill del task (evidenziate in rosso). Le tipologie di skill che sono state realizzate per testare l'algoritmo di controllo precedentemente esposto sono quattro:

- **Skill di tracking:** è la prima skill della macchina a stati ed ha la funzione di assegnare alle variabili q e \dot{q} dichiarate al suo interno, i valori relativi passati da OPCOM.
- **Movimento nello spazio dei giunti:** si tratta di skill che utilizzano il solo generatore di traiettoria per muovere il braccio del robot ad una determinata configurazione desiderata (skill start position). In questa tipologia di skill non è attivo l'algoritmo di ottimizzazione.
- **Stand still:** questo tipo di skill ha la funzione di pulire la memoria occupata dal generatore di traiettoria, una volta che questo abbia terminato di portare il robot alla posizione ed all'orientamento desiderati. Per questa ragione se ne troverà una di seguito ad una skill che ha utilizzato il generatore di traiettoria. Inoltre questa skill obbliga il robot a stare nella posizione in cui si trova per 4s.
- **Movimento nello spazio cartesiano:** questa tipologia di skill è quella che sfrutta l'algoritmo di controllo precedentemente discusso e verrà approfondita in seguito (skills approach, lift).

I gripper vengono azionati durante una skill che non comporta movimento del braccio robotico, così da permettere una presa salda dell'oggetto da manipolare.

5.1.1 Skill di movimento nello spazio cartesiano con ottimizzazione vincolata

Lo schema di funzionamento del controllo, quando la macchina a stati si trova in uno stato dove il controllo è attivo è mostrato in figura 5.3. Di

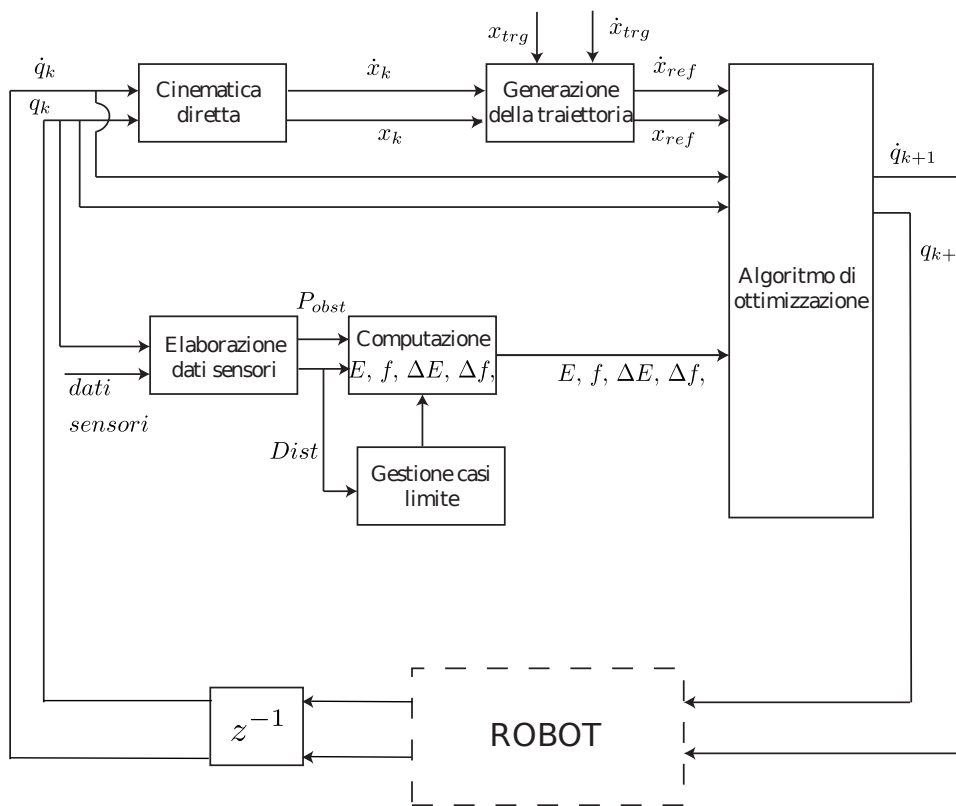


Figura 5.3: Schema funzionamento del controllo durante una skill di movimento nello spazio cartesiano

seguito sono illustrate le funzioni dei vari blocchi dello schema. Le variabili di giunto q e \dot{q} sono quelle inviate dall'axis computer del robot. Queste vengono tradotte in corrispondenti coordinate cartesiane (x_k, \dot{x}_k) tramite le relazioni della cinematica diretta.

Elaborazione dati dei sensori

Lo scopo di questa funzione è quello di ricostruire la posizione nello spazio cartesiano degli ostacoli intercettati dai sensori e di calcolare i valori di distanza da utilizzare nel vincolo di safety. Questa funzione riceve in input i valori di tensione letti dai sensori posti sulle shell e la configurazione dei giunti all'istante corrente q_k .

I valori di tensione inviati dai sensori vengono interpolati tramite un apposito polinomio, in modo da restituire la misura della distanza dagli ostacoli, espressa in metri. Queste distanze, data la particolare caratteristica del sensore, sono saturate ai loro estremi, così da evitare letture errate.

$$0.05m \leq \text{distanza} \leq 0.5m \quad (5.1)$$

Se non venissero saturate in questa maniera, i sensori potrebbero rilevare la presenza di un ostacolo anche se questo non esistesse (in caso di distanze maggiori di $0.5m$) o addirittura misurare la distanza di un ostacolo realmente esistente, molto maggiore di quella effettiva (in caso di distanza dall'ostacolo inferiore a $0.05m$) Data la configurazione dei giunti all'istante k , viene poi calcolata la cinematica diretta del braccio robotico. La posizione e l'orientamento di ogni sensore i , rispetto al giunto di riferimento (giunto 2 per i sensori sul braccio e giunto 4 per quelli sull'avambraccio) sono noti [34]. Perciò, le coordinate degli ostacoli nello spazio cartesiano $(x_{obst}, y_{obst}, z_{obst})$ rilevati dai sensori, sono calcolate sommando, per ciascun sensore, alla sua posizione (nell'istante k) la distanza misurata e direzionata lungo l'asse z della terna del sensore come mostrato, in figura 5.4.

$$P_{obst} = P_{sensore} \cdot \vec{z} \quad (5.2)$$

Una volta ottenute le coordinate degli ostacoli presenti, si procede al calcolo, per ognuno di questi, delle relative distanze minime dai tre link del robot (braccio, avambraccio, tool applicato all'end effector).

Si sottraggono i valori di sicurezza Δ , così da ottenere le distanze appropriate al calcolo del vincolo di safety.

Gestione casi critici

Come già osservato, il range di visione dei sensori è artificialmente saturato via software nel range $[0.05m \quad 0.5m]$. Questo, in aggiunta al fatto che gli

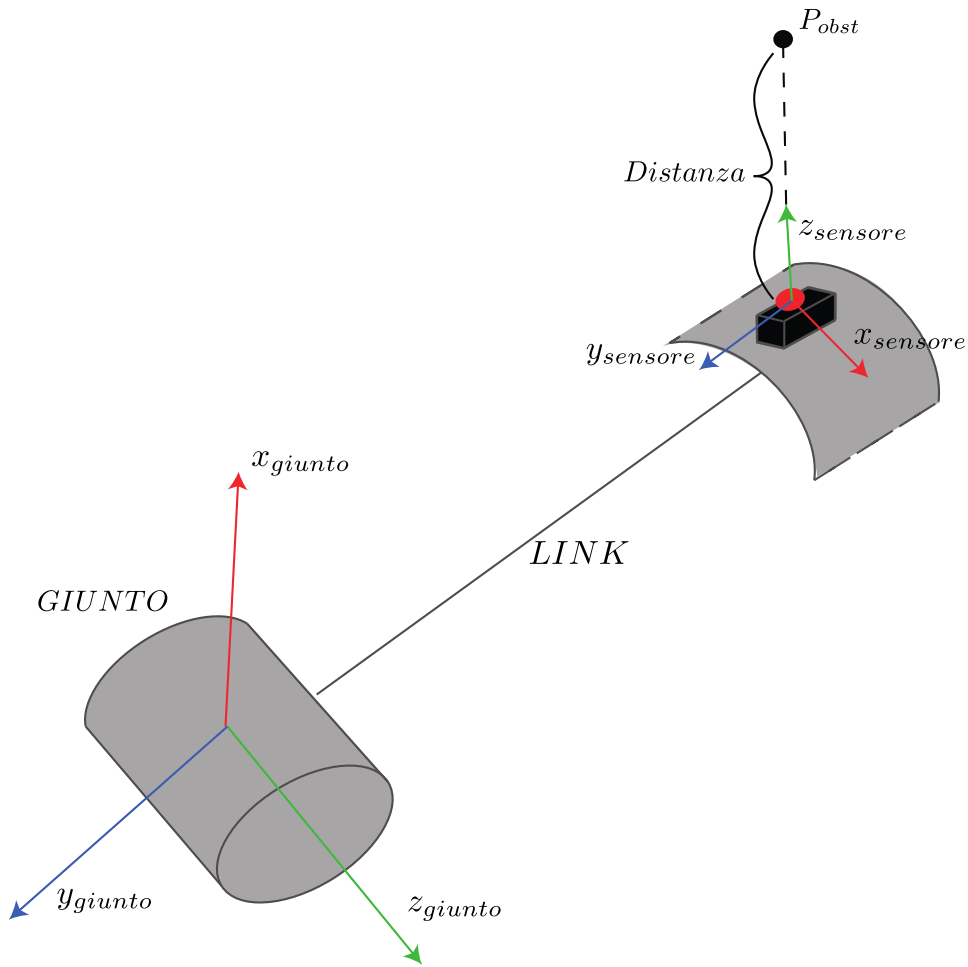


Figura 5.4: Ricostruzione di un punto di ostacolo nello spazio tramite cinematica diretta

ostacoli visti dai sensori sono puntiformi, fa sì che vi sia una variazione brusca nella distanza misurata dal sensore, quando questo comincia a rilevare un ostacolo, o smette di vederlo. Data la scrittura del vincolo di safety in 4.20, una delle ipotesi alla base del funzionamento dell'algoritmo è quella di continua derivabilità di f e di E . La discontinuità nelle letture dei sensori porta però inevitabilmente ad analoghe discontinuità nei vincoli che da queste dipendono. Per ovviare a tali criticità, dovute all'integrazione della tecnologia del sensore distribuito, si sono introdotti dei flag che vanno a modificare la righe e le colonne di ΔE e Δf , quando si presentano le situazioni

appena descritte. Si associa quindi un flag ad ogni sensore e ciascun flag può assumere i valori 0 o 1 come mostrato in tabella 5.1: dove $dist_k$ è la distanza

	flag
$dist_k < 0.5m$	1
$dist_k \geq 0.5m$	0

Tabella 5.1: Definizione flag

misurata dal sensore associato al flag all'istante corrente. Dopo aver calcolato il valore dei flag all'istante k , si memorizzano in un array le distanze misurate dai sensori.

All'istante successivo ($k + 1$) si effettuerà un controllo, che permette di escludere dal problema di ottimizzazione i casi pericolosi:

	flag=0	flag=1
$dist_{k-1} < 0.5m$	$\Delta E = 0 \quad \Delta f = 0$	nessuna modifica a ΔE e Δf
$dist_{k-1} \geq 0.5m$	nessuna modifica a ΔE e Δf	$\Delta E = 0 \quad \Delta f = 0$

Tabella 5.2: Modifica ΔE e Δf in casi critici

In tabella 5.2 si vede come, per identificare i momenti di passaggio critici dei sensori, si vadano a considerare le distanze misurate al passo precedente ($k - 1$) e quelle all'istante corrente (k), rappresentate dai valori dei flag. Si noti che, se un sensore all'istante attuale ha intercettato un ostacolo, ma al passo precedente aveva il campo di visione sgombro o se, viceversa, al passo corrente il sensore non vede alcun ostacolo, mentre ne aveva identificato uno al passo precedente, si agisce sulle variazioni che questi passaggi causano su ΔE e Δf . Si azzerano allora le righe e le colonne di ΔE e Δf relative ai sensori che si trovano in condizione critica, in modo da riuscire ad avere una continuità in E ed f tra l'istante k e l'istante $k + 1$, tale da permettere il normale funzionamento dell'algoritmo.

Generazione della traiettoria

Questo blocco utilizza una libreria chiamata *reflexxes* [37], che prende come input i limiti di velocità ed accelerazione nello spazio considerato (spazio dei giunti o spazio cartesiano).

Gli output in uscita da questa funzione sono: i profili di posizione e di velocità necessari a raggiungere la posizione e l'orientamento desiderati all'end effector, a partire dalla posizione e dall'orientamento correnti, e il tempo (T_{sync}), espresso in secondi, necessario al completamento di tali profili. Se la traiettoria è ultimata, ossia se la posizione e l'orientamento correnti sono uguali a quelli desiderati, la funzione restituisce un valore intero maggiore di zero ($Result > 0$). In questo caso, o nel caso in cui il tempo necessario al completamento della traiettoria sia inferiore ad un tempo di ciclo ($T_{sync} \leq 4ms$), si può passare allo stato successivo della macchina a stati.

Problema di ottimizzazione

Il problema di ottimizzazione viene risolto nelle prove, sia in quelle sperimentali che in quelle in simulazione, tramite l'utilizzo di un solutore numerico dedicato denominato SOTH [12] [38]. L'algoritmo completo, dato dalla risoluzione in cascata del problema in (4.21) con quello in (4.24) (presentato nel capitolo 3), viene allora riscritto sotto forma di vincoli, in modo da poter

essere risolto dall'ottimizzatore:

$$\ddot{q}_{inf} \leq \ddot{q} \leq \ddot{q}^{sup} \quad (5.3)$$

$$-\infty \leq T_s \cdot E \cdot \ddot{q} \leq -E\dot{q} - f - \Delta e\dot{q} + \Delta f \quad (5.4)$$

$$-\dot{x}^{max} - \dot{x} - \Delta J\dot{q} \leq T_s J\ddot{q} \leq \dot{x}^{max} - \dot{x} - \Delta J\dot{q} \quad (5.5)$$

$$-\ddot{x}^{max} - \frac{1}{T_s} \Delta J\dot{q} \leq J\ddot{q} \leq \ddot{x}^{max} - \frac{1}{T_s} \Delta J\dot{q} \quad (5.6)$$

$$\sqrt{Q_v}(\dot{x}_{k+1} - J\dot{q} - \Delta J\dot{q}) \leq \sqrt{Q_v} T_s J\ddot{q} \leq \sqrt{Q_v}(\dot{x}_{k+1} - J\dot{q} - \Delta J\dot{q}) \quad (5.7)$$

$$\sqrt{Q_p}(x_{k+1} - \frac{T_s}{2} \Delta J\dot{q} - \frac{1}{T_s} J\dot{q} - x) \leq \sqrt{Q_p} \frac{T_s^2}{2} J\ddot{q} \leq \sqrt{Q_p}(x_{k+1} - \frac{T_s}{2} \Delta J\dot{q} - \frac{1}{T_s} J\dot{q} - x) \quad (5.8)$$

$$-\dot{q} \leq T_s \ddot{q} \leq -\dot{q} \quad (5.9)$$

dove il vincolo in (5.3) esprime semplicemente il rispetto dei vincoli sulle accelerazioni istantanee dei giunti del braccio robotico. L'equazione (5.4) esprime il vincolo di safety. I vincoli in (5.6) e in (5.5) sono quelli relativi al task del robot e sono espressi nello spazio cartesiano. In questi vincoli compaiono la velocità e la posizione all'istante successivo (x_{k+1}, \dot{x}_{k+1}) . Le cifre di merito volte a minimizzare gli errori di posizione e di velocità sono quelle in (5.7) e in (5.8). Infine il vincolo in (5.9) è quello che cerca di minimizzare la velocità dei giunti e traduce la cifra di merito di 4.24.

L'ordine con cui sono scritti questi vincoli non è casuale: essi sono disposti in ordine di priorità gerarchica. In tal modo SOTH è libero di rilassare i vincoli a priorità più bassa (per esempio le cifre di merito), al fine di mantenere il problema ammissibile. Il primo vincolo è stato scritto in modo da essere sempre ammissibile. In questo modo non si corre il rischio di ricevere soluzioni numeriche inammissibili dal solutore.

5.1.2 Controllo del gripper

Il controllo dei gripper viene effettuato solo in fase sperimentale sul robot. I gripper sono controllati attraverso una funzione che comunica con la scheda

di acquisizione NATIONAL INSTRUMENTS. Attraverso un'opportuna maschera si settano i bit della scheda che potranno essere modificati da questa funzione. A seconda dello stato si modificheranno poi i bit necessari ad attivare la chiusura o l'apertura dei gripper come mostrato nella tabella (5.3).

	bit	
$state \geq 2$	4	gripper closing
$state = 6$	8	gripper opening

Tabella 5.3: Assegnamento bit controllo gripper

5.2 Prova in simulazione

Per verificare le prestazioni dello schema di controllo, si è proceduto a realizzare uno script apposito che permettesse la chiamata ciclica della macchina a stati e che integrasse una simulazione di ostacoli virtuali percepiti dai sensori. La prova che è stata simulata vede la generazione di un piano virtuale posizionato in modo tale da impedire al robot il completamento del task. In tal modo è possibile testare le capacità dell'algoritmo a far fronte alla necessità di un arresto completo del braccio robotico.

5.2.1 Generazione ostacolo virtuale

Concettualmente la funzione che simula la presenza di un ostacolo virtuale è molto simile a quella realmente implementata nello schema di controllo esposto in figura.5.3. Tuttavia, per motivi di complessità, si è scelto di utilizzare come ostacolo virtuale un piano fisso nello spazio. Questo ha equazione $Y = k$ con k parametro che può essere variato nelle simulazioni, per studiare i vari comportamenti dell'algoritmo.

La funzione che simula l'ostacolo sfrutta l'equazione parametrica del piano

nello spazio e quelle delle rette passanti dalle posizioni dei sensori e orientate come i relativi assi z , per trovarne i punti di intersezione. Se questi esistono, la funzione ne calcola la distanza dalle origini dei sensori, e la restituisce, depurata del valore Δ , in maniera aderente alla realtà sperimentale. In figura 5.5 si può vedere come il braccio robotico si disponga massimizzando la distanza dai punti di ostacolo individuati. In blu sono visibili i punti del piano virtuale rilevati dai sensori sul braccio. Nelle figure 5.6, 5.7, 5.8 si possono

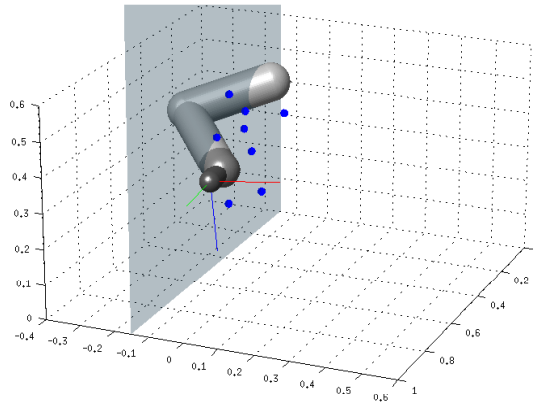


Figura 5.5: Configurazione finale del braccio robotico simulato a seguito dell'interruzione del task

vedere gli andamenti delle tre coordinate cartesiane dell'end-effector, con i relativi riferimenti. Si può notare come l'algoritmo riesca ad ottenere un buon inseguimento dei goal fino a quando non è costretto ad interrompere il moto a causa della presenza dell'ostacolo. L'orientamento dell'end-effector è mostrato nelle figure 5.9, 5.10 e 5.11. Si può notare come, correttamente, il robot non riesca a completare il task assegnatogli (resta nella skill 3), a causa della vicinanza dell'ostacolo virtuale, precludendogli la possibilità di raggiungere il punto prefissato.

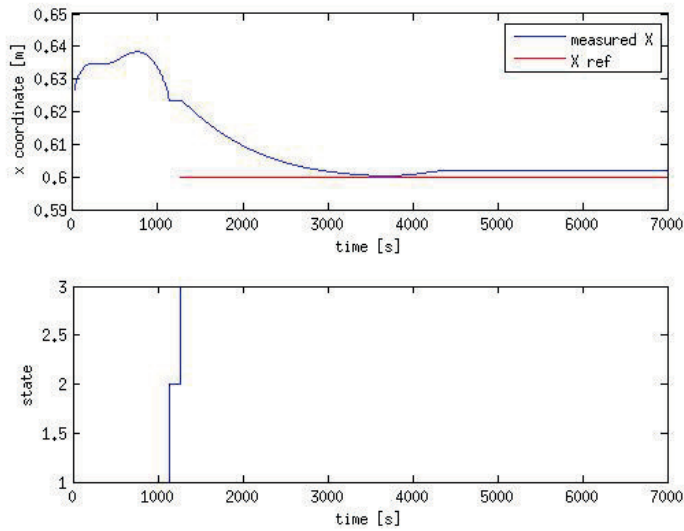


Figura 5.6: Andamento con ostacolo virtuale della coordinata X dell'end effector con relativi stati

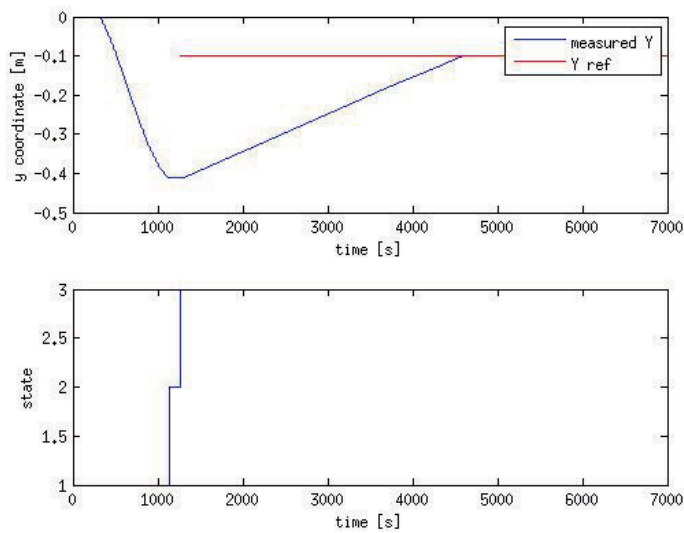


Figura 5.7: Andamento con ostacolo virtuale della coordinata Y dell'end effector con relativi stati

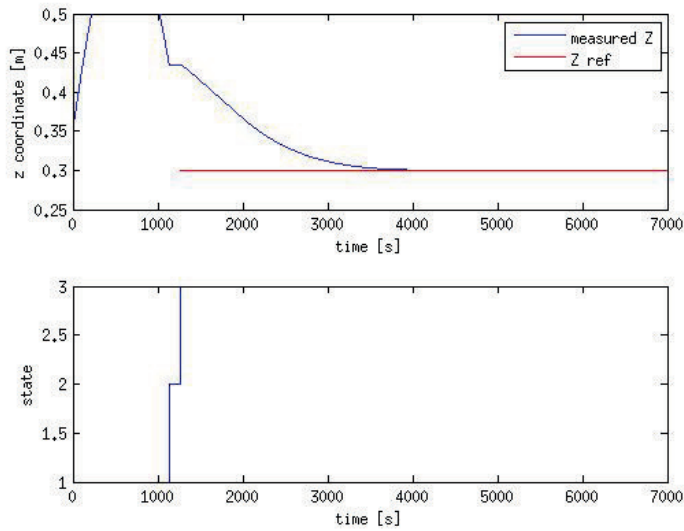


Figura 5.8: Andamento con ostacolo virtuale della coordinata Z dell'end effector con relativi stati

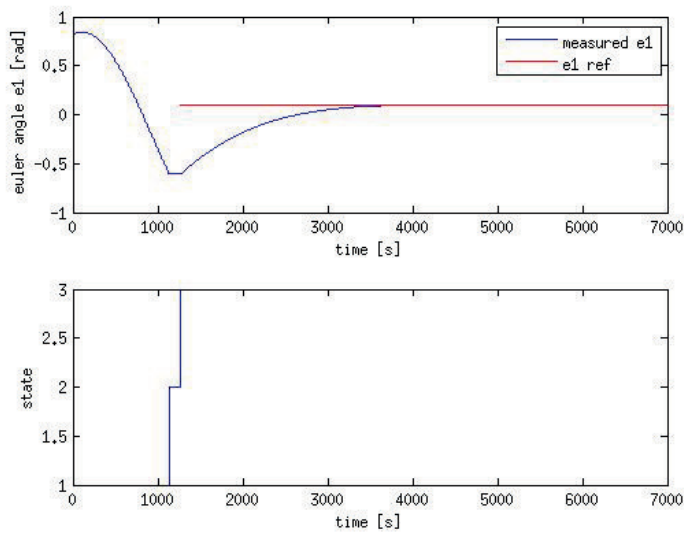


Figura 5.9: Andamento con ostacolo virtuale del primo angolo di eulero dell'end effector (Z) con relativi stati

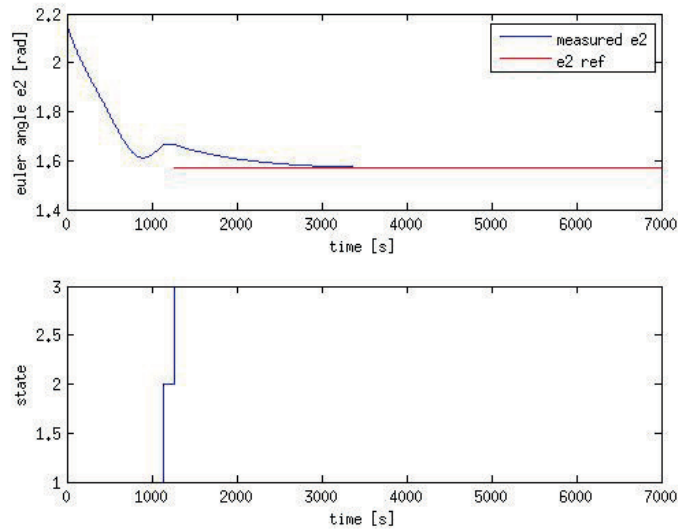


Figura 5.10: Andamento con ostacolo virtuale del secondo angolo di eulero dell'end effector (Y) con relativi stati

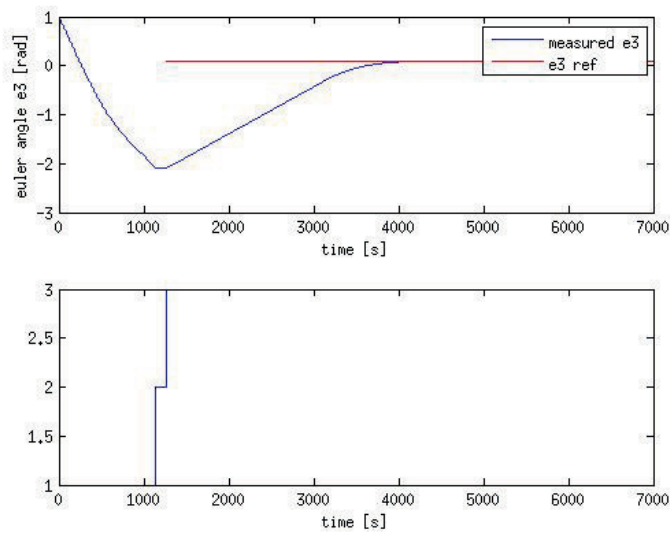


Figura 5.11: Andamento con ostacolo virtuale del terzo angolo di eulero dell'end effector (Z) con relativi stati

5.3 Prove sperimentali

Le prove sperimentali effettuate su FRIDA sono state di quattro tipologie e sono state pensate per poter adeguatamente stimolare il controllo, al fine di testarne eventuali limiti.

Per motivi di complessità computazionale, le prove reali su FRIDA sono state effettuate utilizzando solo quattro dei ventiquattro sensori disponibili sulle shell, onde evitare che la grande quantità di calcoli richiesti dalla soluzione dell'ottimizzazione vincolata potesse far superare la soglia dei $4ms$, entro i quali tassativamente è necessario inviare i riferimenti di posizione e velocità all'axis computer.

Nella sezione conclusiva eventuali accorgimenti per ovviare a questo problema verranno discussi come possibili sviluppi futuri.

5.3.1 Prova senza ostacoli

La prima prova è stata effettuata senza la presenza di ostacoli che potessero costringere il braccio robotico a deviare dalla sua traiettoria nominale o a bloccarsi completamente lungo il percorso. In questa maniera si è potuto verificare il corretto inseguimento dei riferimenti assegnati, nonché il corretto passaggio tra una skill ed un'altra della macchina a stati. Come si può notare dalla figura 5.12, il robot raggiunge il punto finale del task (definito come posizione e orientamento dell'end effector). Nelle figure sottostanti in 5.13, 5.14, 5.15, 5.16, 5.17, 5.18 sono mostrati l'andamento nel tempo delle variabili di posizione e orientamento dell'end effector, con relativi riferimenti nonché i passaggi fra le varie skill. I riferimenti di posizione e orientamento (in rosso) sono definiti dallo stato 3 in poi, poichè gli stati precedenti si basano sulla generazione del movimento nello spazio dei giunti.

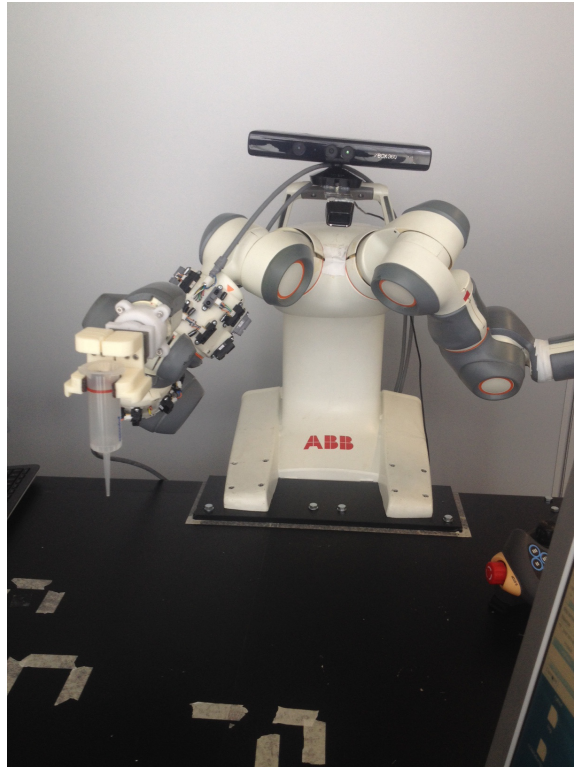


Figura 5.12: Foto della prova sperimentale senza ostacolo

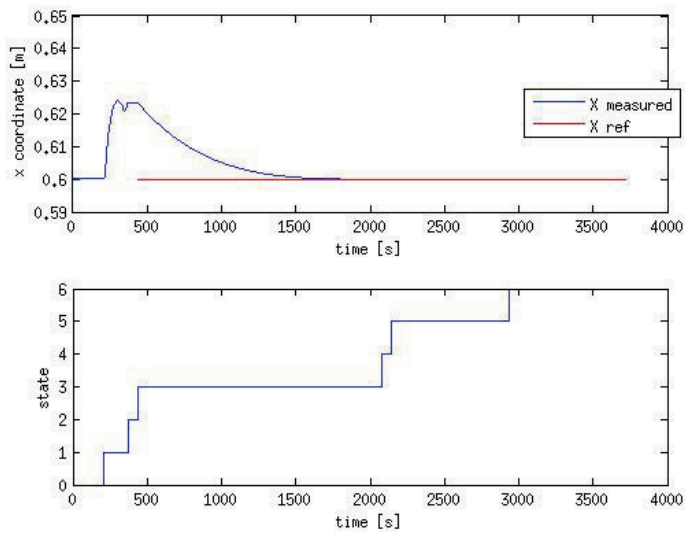


Figura 5.13: Andamento senza ostacoli della coordinata X dell'end effector con relativi stati

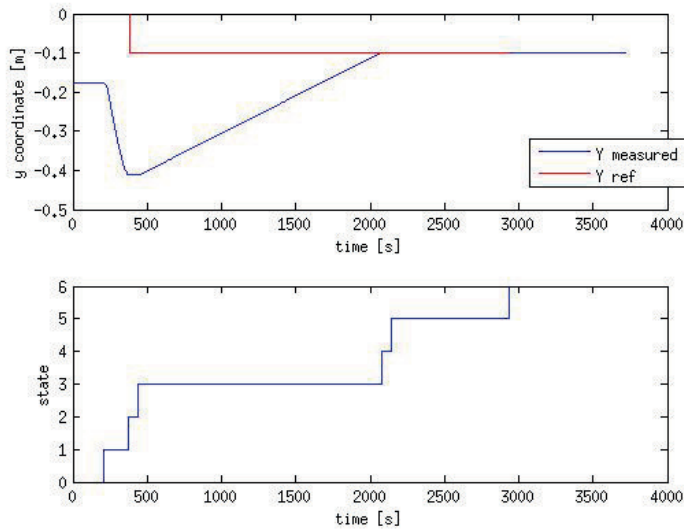


Figura 5.14: Andamento senza ostacoli della coordinata Y dell'end effector con relativi stati

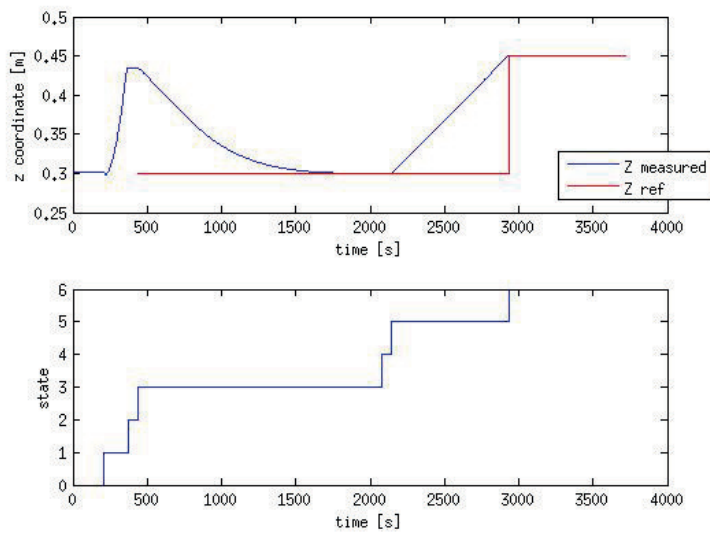


Figura 5.15: Andamento senza ostacoli della coordinata Z dell'end effector con relativi stati

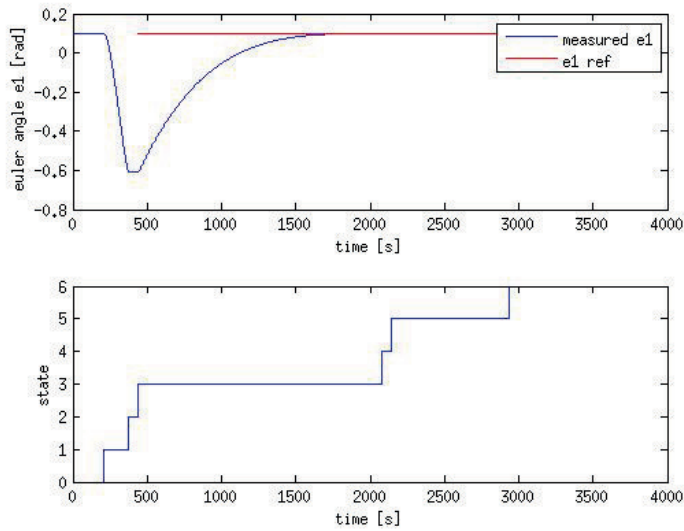


Figura 5.16: Andamento senza ostacoli del primo angolo di eulero dell'end effector (Z) con relativi stati

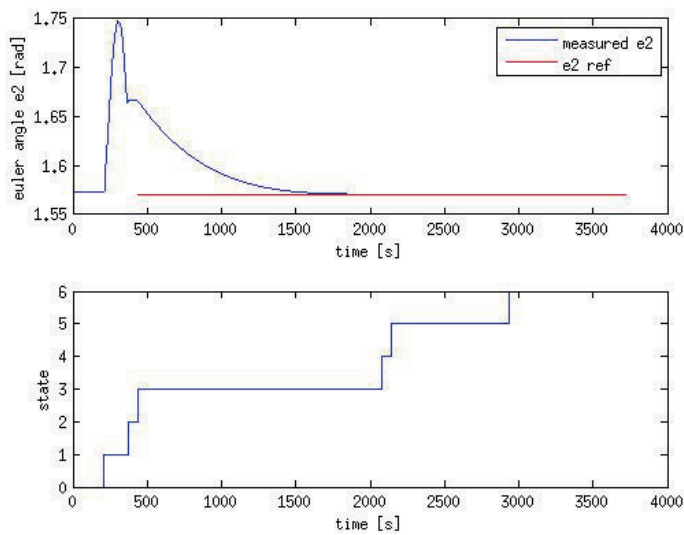


Figura 5.17: Andamento senza ostacoli del secondo angolo di eulero dell'end effector (Y) con relativi stati

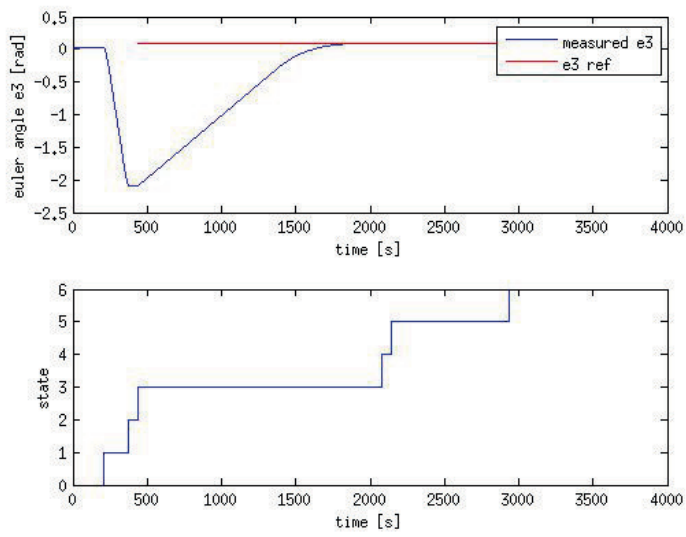


Figura 5.18: Andamento senza ostacoli del terzo angolo di eulero dell'end effector (Z) con relativi stati

5.3.2 Prova con ostacolo fisso che non impedisce il completamento del task

Questa seconda prova è stata effettuata ponendo un ostacolo fisso ben visibile dai sensori. L'ostacolo è stato posizionato in modo da non impedire totalmente lo svolgimento dell'intero task del robot, ma abbastanza vicino al braccio robotico da sollecitare il controllo. Nella foto in fig. 5.19 si nota come, nonostante la presenza di un ostacolo sulla traiettoria nominale (un pannello di polistirolo) il robot porti a termine correttamente il task, massimizzando la sua distanza dall'ostacolo tramite il grado di libertà ridondante del gomito, che si trova più in alto rispetto alla prova precedente.



Figura 5.19: Foto della prova sperimentale con ostacolo che non impedisce lo svolgimento del task

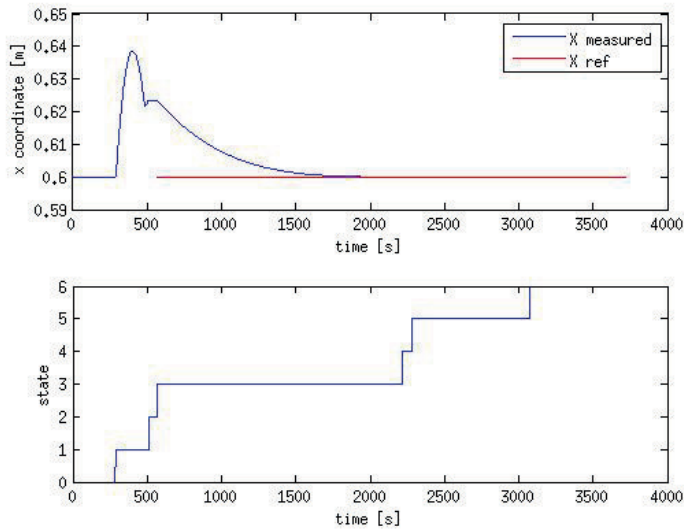


Figura 5.20: Andamento con ostacolo non critico della coordinata X dell'end effector con relativi stati

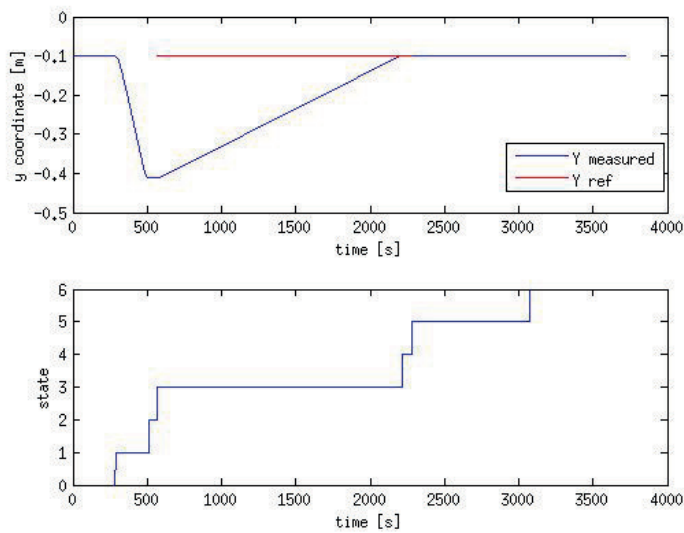


Figura 5.21: Andamento con ostacolo non critico della coordinata Y dell'end effector con relativi stati

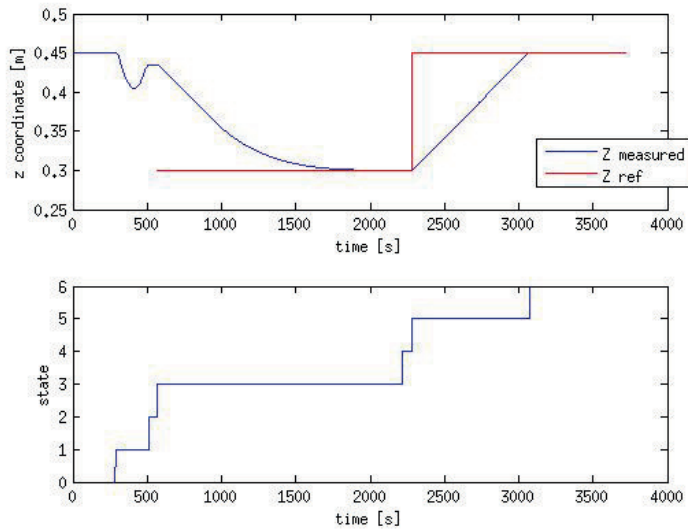


Figura 5.22: Andamento con ostacolo non critico della coordinata Z dell'end effector con relativi stati

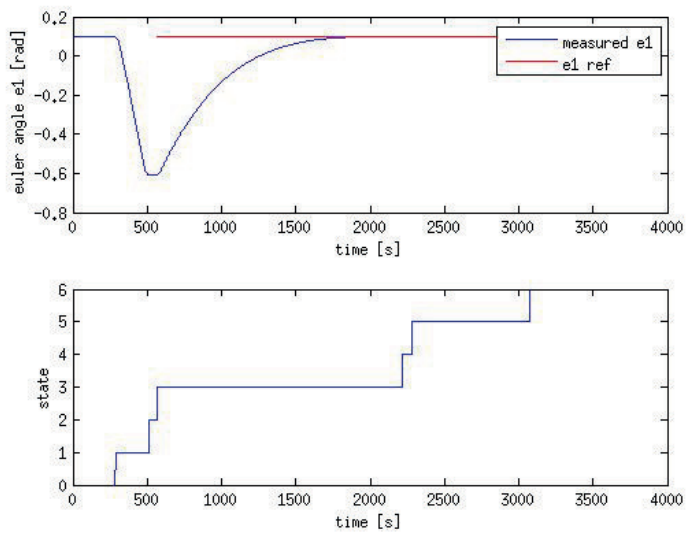


Figura 5.23: Andamento con ostacolo non critico del primo angolo di eulero dell'end effector (Z) con relativi stati

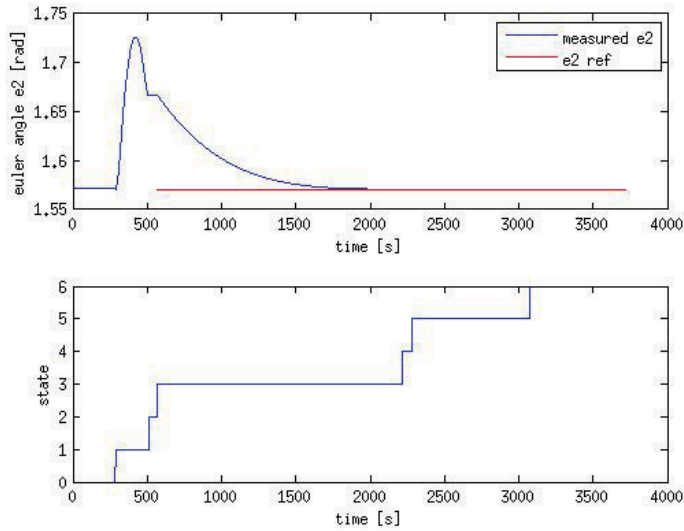


Figura 5.24: Andamento con ostacolo non critico del secondo angolo di eulero dell'end effector (Y) con relativi stati

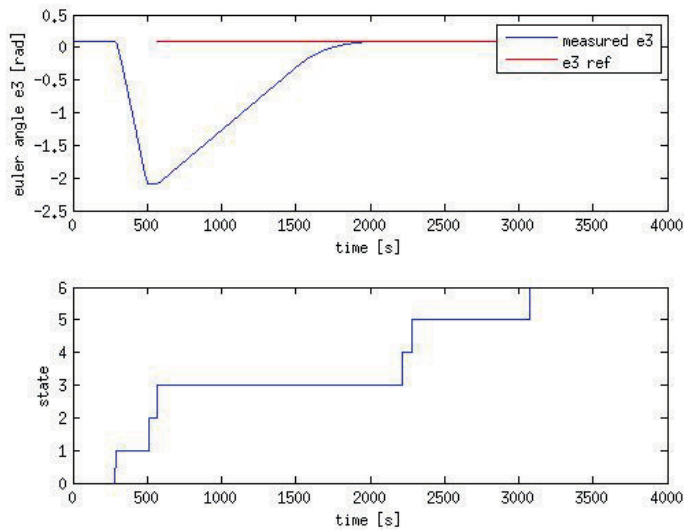


Figura 5.25: Andamento con ostacolo non critico del terzo angolo di eulero dell'end effector (Z) con relativi stati

Similmente alla prova precedente nelle figure 5.20, 5.21, 5.22 vengono mostrati gli andamenti delle tre coordinate cartesiane dell'end-effector. Si nota come questi seguano bene i riferimenti assegnati, nonostante la presenza dell'ostacolo fisso. L'orientamento dell'end effettore nelle sue tre coordinate è visibile nelle figure 5.23, 5.24 e 5.25.

5.3.3 Prova con ostacolo fisso che impedisce il completamento del task

La terza prova ha visto una configurazione simile alla precedente, con la differenza che l'ostacolo fisso è stato posizionato in modo da impedire il completamento del task, così da verificare che il controllo fosse in grado di arrestare il moto del braccio robotico prima di un possibile impatto. Come

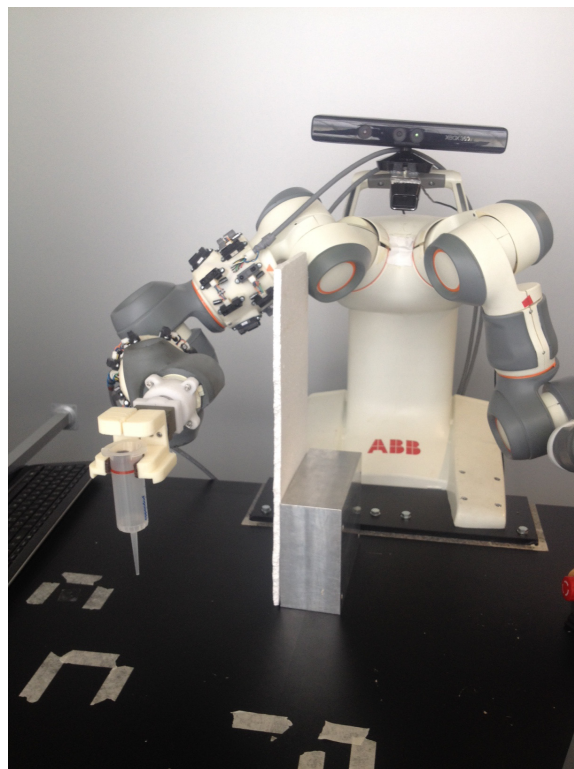


Figura 5.26: Foto prova con ostacolo fisso che impedisce il compimento del task

si può notare dalla foto in 5.26, se l'ostacolo viene posto troppo vicino al

braccio, questo non riesce a trovare una traiettoria che lo allontani a sufficienza. In questo caso il controllo costringe il braccio robotico a diminuire la propria velocità, mano a mano che la distanza dall'ostacolo diminuisce, fino a raggiungere il completo arresto, in attesa che l'ostacolo venga rimosso.

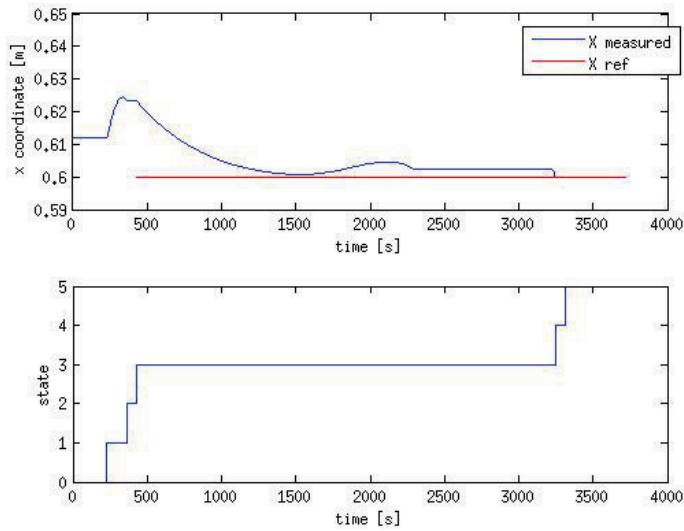


Figura 5.27: Andamento con ostacolo critico della coordinata X dell'end effector con relativi stati

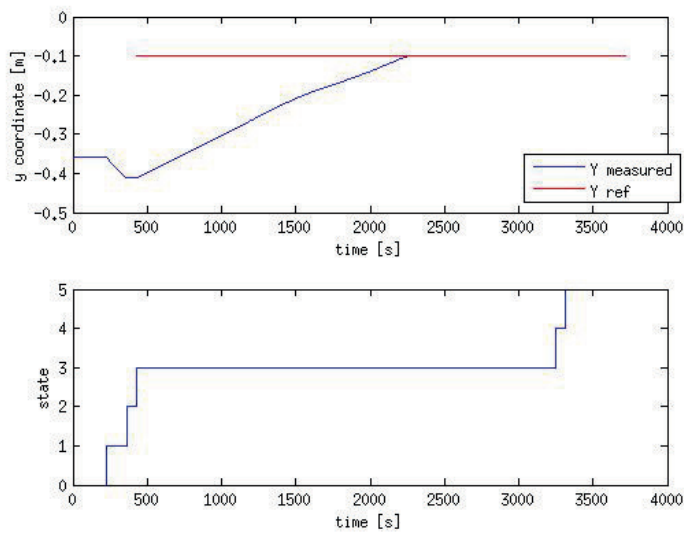


Figura 5.28: Andamento con ostacolo critico della coordinata Y dell'end effector con relativi stati

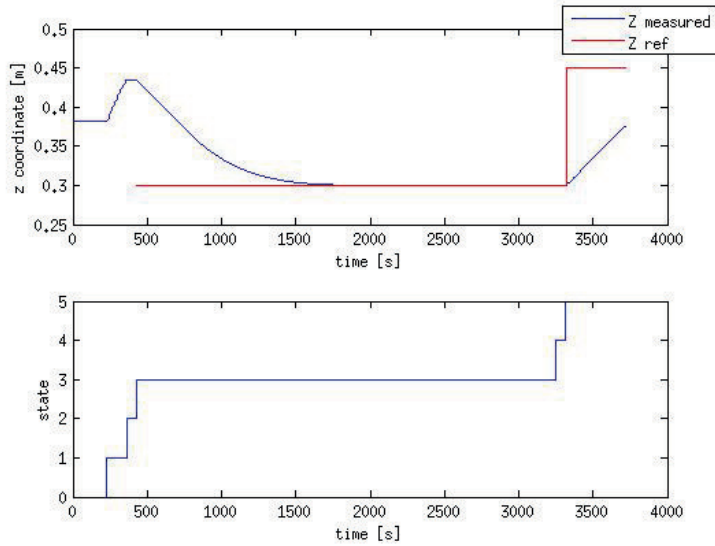


Figura 5.29: Andamento con ostacolo critico della coordinata Z dell'end effector con relativi stati

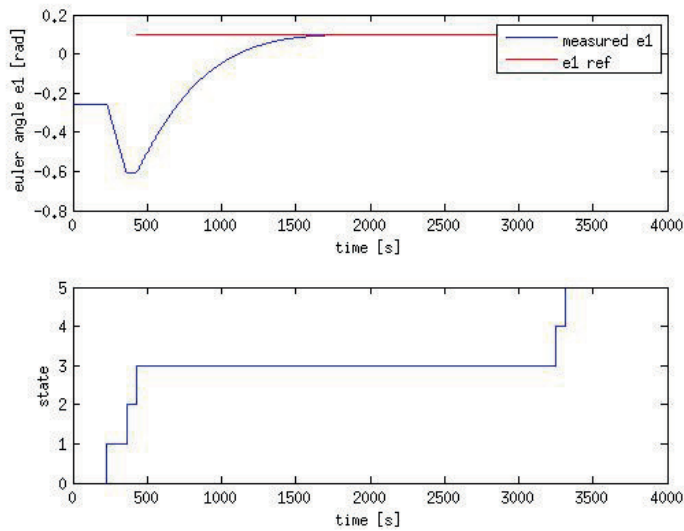


Figura 5.30: Andamento con ostacolo critico del primo angolo di euler dell'end effector (Z) con relativi stati

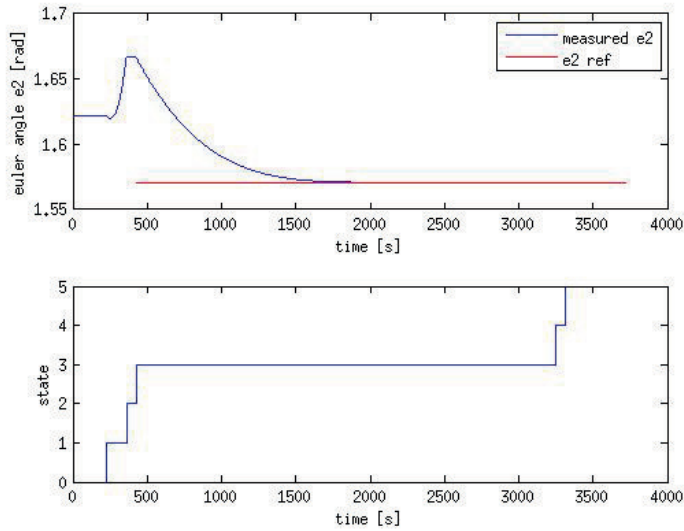


Figura 5.31: Andamento con ostacolo critico del secondo angolo di eulero dell'end effector (Y) con relativi stati

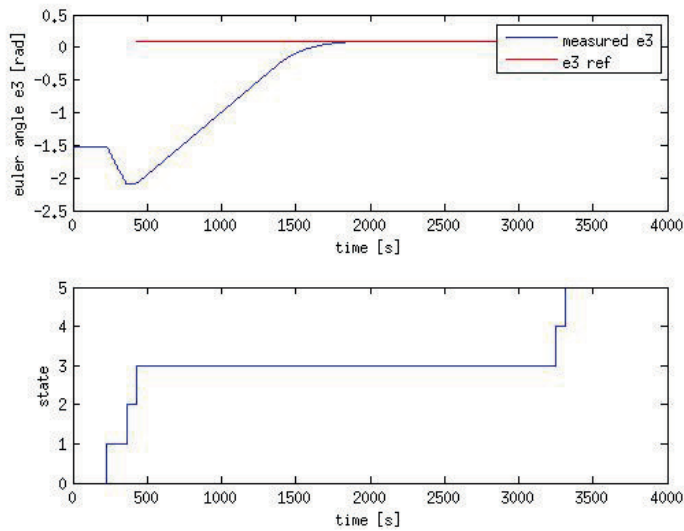


Figura 5.32: Andamento con ostacolo critico del terzo angolo di eulero dell'end effector (Z) con relativi stati

Nelle figure 5.27, 5.28, 5.29, 5.30, 5.31 e 5.32 sono mostrati gli andamenti delle variabili di interesse. In queste si osserva come l'algoritmo riesca a seguire i goal prefissati (in rosso), fino al momento dell'arresto forzato. Si può notare come in questa prova, il task venga infatti sospeso molto a lungo nello stato tre. Fino a quando l'ostacolo non è stato rimosso, il robot è correttamente rimasto fermo. Una volta che l'ostacolo è stato rimosso, si può notare come l'esecuzione del task riprenda normalmente.

5.3.4 Prova con ostacolo mobile

L'ultima prova è stata quella che ha maggiormente sollecitato il controllo, poichè sono stati avvicinati più ostacoli mobili al braccio robotico durante l'esecuzione del task.

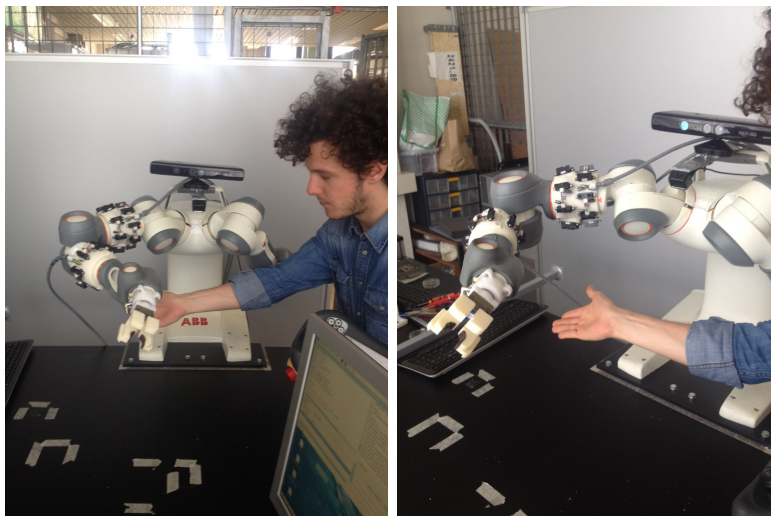


Figura 5.33: Foto prova con ostacolo in movimento

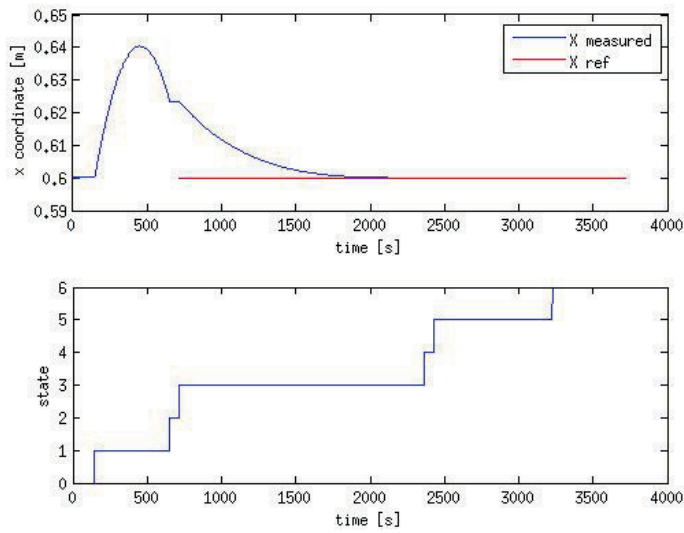


Figura 5.34: Andamento con ostacolo mobile della coordinata X dell'end effector con relativi stati

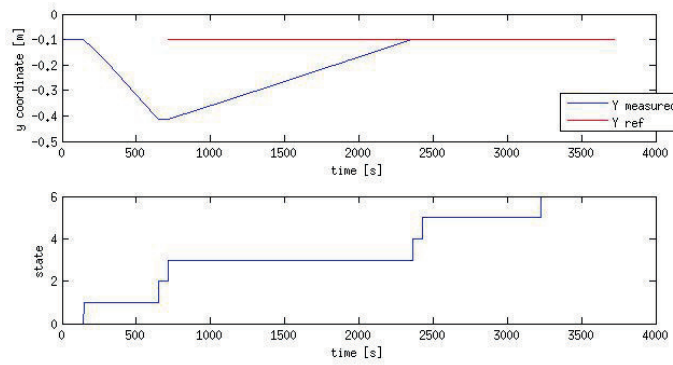


Figura 5.35: Andamento con ostacolo mobile della coordinata Y dell'end effector con relativi stati

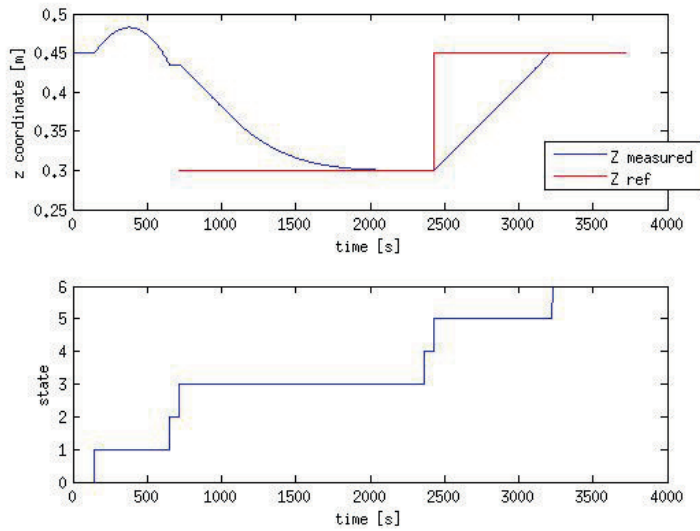


Figura 5.36: Andamento con ostacolo mobile della coordinata Z dell'end effector con relativi stati

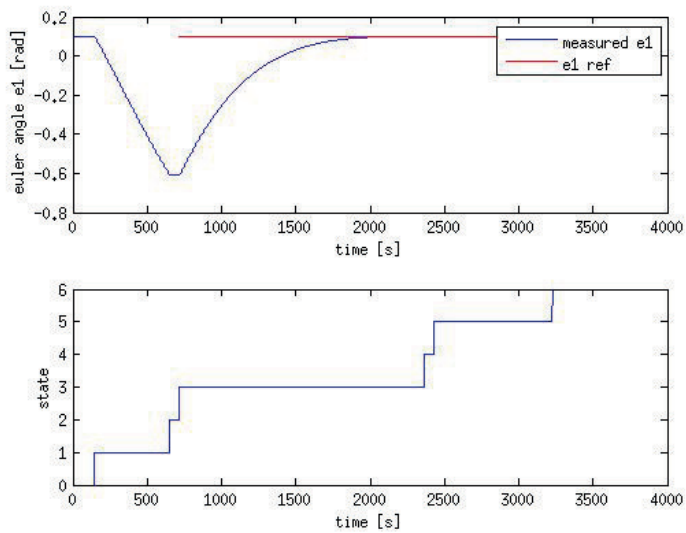


Figura 5.37: Andamento con ostacolo mobile del primo angolo di euler dell'end effector (Z) con relativi stati

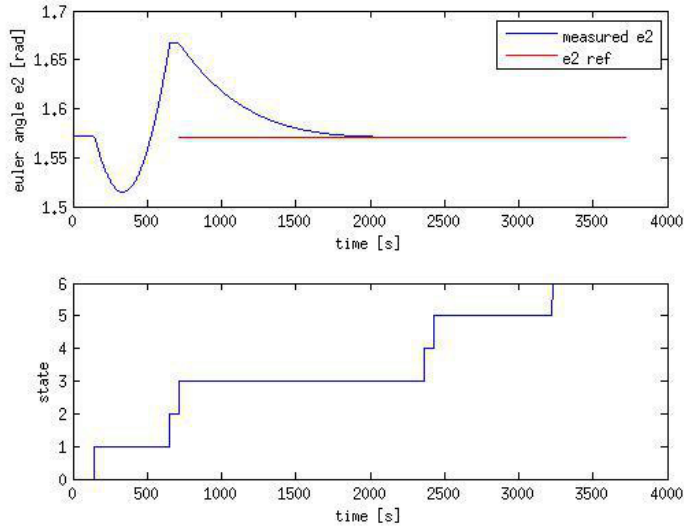


Figura 5.38: Andamento con ostacolo mobile del secondo angolo di eulero dell'end effector (Y) con relativi stati

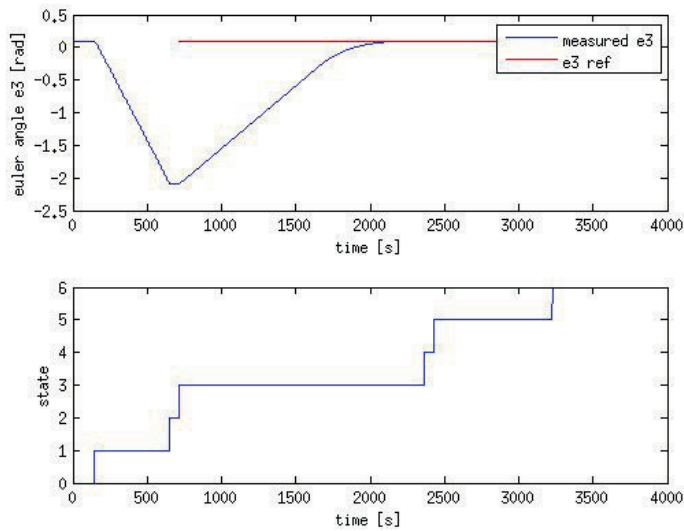


Figura 5.39: Andamento con ostacolo mobile del terzo angolo di eulero dell'end effector (Z) con relativi stati

In figura 5.33 è mostrato l'ostacolo mobile ovvero la mano e il braccio dell'operatore umano. Nell'avvicinarsi ai sensori, l'operatore costringe il controllo ad adottare manovre evasive al fine di aumentare la distanza fra il braccio e l'ostacolo, pur mantenendo inalterata la configurazione dell'end effector. Nelle figure che mostrano l'andamento dell'end effector, che sono la 5.34, 5.35, 5.36, 5.37, 5.38 e la 5.39 si può notare come il controllo, dopo aver effettuato una manovra evasiva, a seguito dell'avvicinarsi dell'operatore, riesca comunque a mantenere un buon inseguimento dei riferimenti assegnati e di completare il task.

5.4 Commenti

Dall'analisi delle prove effettuate, sia quella in simulazione, che quelle sperimentali, si è potuto evincere come l'algoritmo di controllo riesca efficacemente a seguire i riferimenti desiderati nello spazio di lavoro, anche a fronte di importanti sollecitazioni. L'algoritmo garantisce inoltre l'arresto in sicurezza del robot, in caso sia impossibile aggirare l'ostacolo rilevato, mantenendo quindi una distanza di sicurezza minima da quest'ultimo. L'instabilità numerica dovuta all'utilizzo di sensori puntiformi, che quindi forniscono informazioni molto frammentarie sulla geometria totale dell'ostacolo, è stata efficacemente compensata grazie all'introduzione di opportuni flag in grado di gestire le eventuali discontinuità. Nel complesso, pur dovendo utilizzare un numero molto inferiore di sensori, rispetto a quelli disponibili, a causa dei limiti dell'hardware, si è riusciti a testare efficacemente le potenzialità dell'algoritmo.

Capitolo 6

Conclusioni e sviluppi futuri

Nello svolgimento di questo lavoro di tesi si è integrarata la tecnologia del sensore distribuito, disponibile al laboratorio MERLIN del Politecnico di Milano, al fine di applicare una tecnologia a basso costo nella risoluzione di un problema centrale per lo sviluppo industriale, ovverosia quello della cooperazione uomo-robot.

Dopo aver effettuato varie prove, sia in simulazione, sia sul robot ABB FRIDA, si può concludere che gli obbiettivi per i quali è stato sviluppato l'algoritmo sono pienamente raggiunti. Infatti si può notare che, in presenza di vari ostacoli, sia fissi che mobili, l'algoritmo di ottimizzazione vincolata riesce a pianificare correttamente le manovre evasive necessarie ad evitare la collisione del braccio robotico con questi ultimi. Inoltre, durante l'esecuzione del task, il controllo riesce a mantenere, anche in situazioni di forte sollecitazione, un buon inseguimento dei riferimenti di posizione e orientamento dell'end-effector, garantendo così la massima produttività possibile. Durante le prove però, sono stati riscontrati problemi di varia natura:

- problemi di complessità computazionale;
- problemi di instabilità numerica, dovuti all'integrazione del sensore distribuito che non riesce a fornire in maniera accurata tutti i dati

necessari alla corretta esecuzione dell'algoritmo.

I problemi di complessità computazionale sono dovuti essenzialmente alla natura real-time dell'interazione fra il computer dove viene eseguito l'algoritmo e l'axis computer del robot. Questa comunicazione infatti necessita che i riferimenti necessari al moto del robot vengano comunicati all'axis computer, dal controllo ogni $4ms$. Questo è volto a garantire l'esecuzione in real-time dei task assegnati. Già in fase di simulazione però, si è riscontrato che la risoluzione del problema di ottimizzazione vincolato, se utilizzati tutti e ventiquattro i sensori sulle shell, aveva un tempo di computazione decisamente eccessivo. Per ovviare a questo problema e quindi garantire il funzionamento dell'algoritmo in real-time, si è quindi deciso di utilizzare nelle prove sperimentali solamente quattro sensori. Inoltre è stato riscontrato che, in alcune particolari configurazioni geometriche del braccio e quando l'algoritmo è soggetto a delle sollecitazioni eccessive date dagli ostacoli, il problema di ottimizzazione non riesce comunque a trovare una soluzione nel tempo necessario. Si è quindi riscritto il codice Matlab, prestando particolare attenzione all'efficienza computazionale, ossia precalcolando tutte le quantità possibili ed adottando strategie di compilazione orientate alla velocità di esecuzione. I problemi di instabilità numerica, sono invece dovuti alle caratteristiche intrinseche del sensore di distanza distribuito, che non permette una rilevazione completa e continuativa nel tempo degli eventuali ostacoli (gli ostacoli rilevati sono puntiformi). Per gestire i casi critici sono stati introdotti con successo degli appositi flag. Tali flag, quando attivi, vanno a modificare la forma dei vincoli di safety nell'algoritmo di ottimizzazione prima che questo venga eseguito, garantendone così la corretta esecuzione.

I rimedi adottati durante questo progetto di tesi non garantiscono però la robustezza dell'algoritmo a fronte di ogni possibile sollecitazione al quale il controllo possa essere sottoposto.

Per quanto riguarda l'eccessivo tempo computazionale, una possibile solu-

zione che potrebbe risolvere alla radice il problema consisterebbe nel far eseguire l'algoritmo di ottimizzazione su un calcolatore differente da quello che gestisce la comunicazione con il robot (via OPCOM). In questo modo si potrebbe imporre una comunicazione dei riferimenti da inviare all'axis computer ogni $4ms$. Se in tale lasso temporale l'algoritmo riuscisse a terminare la propria esecuzione, questo invierebbe i riferimenti calcolati all'istante corrente. In caso contrario verrebbe comunque fornita un'adeguata interpolazione di quelli calcolati all'istante precedente. Questa soluzione garantirebbe una comunicazione robusta tra qualunque algoritmo debba essere eseguito e l'axis computer, rendendo così inoltre la sperimentazione su FRIDA molto più user-friendly.

Per evitare invece eventuali instabilità numeriche, e poter così applicare questo algoritmo in un ambito differente da quello di laboratorio, la soluzione più ovvia sarebbe quella di utilizzare una tecnologia (il meno costosa possibile) che fornisca una ricostruzione più completa degli ostacoli. Un esempio di tale tecnologia potrebbe essere quello del sensore Kinect che, tramite l'utilizzo di algoritmi di computer vision, potrebbe fornire la completa conformazione dell'ambiente circostante al robot, inclusi eventuali ostacoli. L'applicazione di tale strategia non potrebbe però prescindere dalla previa risoluzione dei problemi di tempo d'esecuzione dell'algoritmo, poiché andrebbe ulteriormente ad appesantirne l'esecuzione.

Bibliografia

- [1] A. Albu Schaeer A. Bicchi R. Bischo R. Chatila A. D. Luca A. D. Santis G. Giralt J. Guiochet G. Hirzinger F. Ingrand V. Lippiello R. Mattone D. Powell S. Sen B. Siciliano G. Tonietti R. Alami and L. Villani. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In *IROS Workshop on pHRI - Physical Human-Robot Interaction in Anthropic Domains*, 10 2006.
- [2] ANSI/RIA. *R15 06 - Safety requirements for industrial robots and robot systems*, 1999.
- [3] G. Antonelli, S. Chiaverini, and G. Fusco. A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits. *Robotics and Automation, IEEE Transactions on*, 19(1):162–167, 2003.
- [4] O. Khatib B. Roth, M. Zinn and J.K. Salisbury. A new actuation approach for human-friendly robot design. In *International Symposium on Experimental Robotics*, 7 2002.
- [5] Luigi Biagiotti and Claudio Melchiorri. *Trajectory planning for automatic machines and robots*. Springer, 2008.
- [6] S. L. Rizzini A. Bicchi and G. Tonietti. Compliant design for intrinsic safety: general issues and preliminary design. In *IEEE Int. Symp. Intelligent Robots and Systems*, pages 1864–1869, 10 2001.

- [7] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.
- [8] Oliver Brock and Oussama Khatib. Elastic strips: A framework for motion generation in human environments. *The International Journal of Robotics Research*, 21(12):1031–1052, 2002.
- [9] Nicola Ceriani and Luca Falorni. Progetto di un sensore di distanza distribuito per il controllo dell’interazione uomo-robot in ambito industriale. Scuola di Ingegneria Industriale.
- [10] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decrè, R. Smits, E. Aertbelien, K. Claes, and H. Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *International Journal of Robotics Research*, 26(5), 2007.
- [11] W. Decrè, H. Bruyninckx, and J. De Schutter. Extending the iTaSC constraint-based robot task specification framework to time-independent trajectories and user-configurable task horizons. In *IEEE International Conference on Robotics and Automation, ICRA*, 2013.
- [12] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028, 2014.
- [13] F. Flacco and A. De Luca. Optimal redundancy resolution with task scaling under hard bounds in the robot joint space. In *IEEE International Conference on Robotics and Automation, ICRA*, 2013.
- [14] T. H. Grönwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, 1919.

- [15] S. Haddadin, A. Albu-Schaeffer, and G. Hirzinger. Requirements for safe robots: measurements, analysis and new insights. *International Journal of Robotics Research*, 28:1507–1527, 2008.
- [16] Sami Haddadin, Holger Urbanek, Sven Parusel, Darius Burschka, Jürgen Roßmann, A Albu-Schaffer, and Gerd Hirzinger. Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2010.
- [17] Kris Hauser. On responsiveness, safety, and completeness in real-time motion planning. *Autonomous Robots*, 32(1):35–48, 2012.
- [18] J. Heinzmann and A. Zelinsky. Quantitative safety guarantees for physical human-robot interaction. pages 22(7–8):479–504, 2003.
- [19] O. Kanoun, F. Lamiroux, and P. B. Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27(4):785–792, 2011.
- [20] O. Kermorgant and F. Chaumette. Dealing with constraints in sensor-based robot control. *IEEE Transactions on Robotics*, 2013.
- [21] Seyed Mohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454, 2012.
- [22] T. Kroeger and F.M. Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 26:94–111, 2010.
- [23] B. Lavecic. *Safe motion planning and control for robotic manipulators*. 2011.

- [24] G. Tonietti M. Bavaro A. Bicchi and M. Piccigallo. Variable stiness actuators for fast and safe motion control. In *International Symposium on Robotics Research*, 2003.
- [25] N. Mansard, O. Khatib, and A. Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3):670–685, 2009.
- [26] A. M. Zanchettin N. M. Ceriani P. Rocco H. Ding B. Matthias. Safety in human-robot collaborative manufacturing environments: metric and control. *IEEE Transaction on automation science and engineering*, 2015.
- [27] Y. Nakabo and M. Ishikawa. Visual impedance using 1 ms visual feedback system. 1998.
- [28] International Federation of Robotics. World robotics 2014 industrial robots.
- [29] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *IEEE International Conference on Robotics and Automation, ICRA*, 1993.
- [30] RIA/ANSI R15.06. *American national standard for industrial robots and robot systems - safety requirements*. American National Standards Institute.
- [31] M. Ragaglia, L. Bascetta, and P. Rocco. Multiple camera human detection and tracking inside a robotic cell - an approach based on image warping, computer vision, k-d trees and particle filtering. In *11th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2014)*, pages 374–381, 9 2014.

- [32] A. M. Zanchettin P. Rocco. Constraint-based control strategy and reactive motion planning for robotic manipulators. *IEEE transactions on robotics*, 2015.
- [33] Claude Samson, Bernard Espiau, and Michel Le Borgne. *Robot control: the task function approach*. Oxford University Press, 1991.
- [34] A. Serio S. Sirioni. Sistema di controllo per l’interazione sicura uomo-robot basato su un nuovo sensore di distanza distribuito. Scuola di Ingegneria Industriale e dell’informazione.
- [35] Y. Tazaki and T. Suzuki. Constraint-based prioritized trajectory planning for multibody systems. *IEEE Transactions on Robotics*, 30(5):1227–1234, 2014.
- [36] T. Tsuji and M. Kaneko. Noncontact impedance control for redundant manipulators. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 29, chapter part A. 1999.
- [37] T. Kröger, F. M. Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 2010.
- [38] A. Escande, N. Mansard, P.-B. Wieber. SOTH github webpage. <https://github.com/stack-of-tasks/soth>, 2010–2015.
- [39] Andrea Maria Zanchettin and Paolo Rocco. Path-consistent safety in mixed human-robot collaborative manufacturing environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1131–1136. IEEE, 2013.
- [40] L. Bascetta P. Bolzern G. Ferretti B. Lacevic A. Locatelli G. Magnani P. Rocco N. Schiavoni A.M. Zanchettin. Robotica industriale umano-centrica: il progetto rosetta. pages 75–79, 6 2011.

- [41] A.L. Wright J. Zurada and J.H. Graham. A neuro-fuzzy approach for robot system safety. 2 2001.