

**POLITECNICO DI MILANO**

**Scuola di Ingegneria Industriale e dell'Informazione**

**Corso di Laurea Magistrale in Ingegneria dell'Automazione**



**Generazione di Traiettorie  
con Controllo di Forza Sensorless  
per un Manipolatore Robotico a Due Bracci**

Relatore: Prof. Paolo Rocco

Correlatore: Dr. Andrea M. Zanchettin

Autore: Davide Nicolis

Matricola: 798886

Anno Accademico 2013-2014



# Sommario

Negli ultimi anni, la crescente domanda per l'impiego di soluzioni robotizzate da parte di piccole e medie imprese ha stimolato il mondo della ricerca alla formulazione di nuove e innovative strategie di controllo, caratterizzate da un'elevata flessibilità e generalità, che permettano una veloce ed efficiente messa a punto dei processi di produzione.

Particolare attenzione è stata posta su algoritmi di generazione della traiettoria basati sulla formalizzazione di vincoli, con lo scopo di tradurre gli specifici requisiti produttivi in segnali di riferimento per il robot attraverso una loro definizione di più alto livello.

In questa tesi si farà uso di uno di questi algoritmi, i cui vantaggi risiedono nella possibilità di essere eseguito in tempo reale e nella capacità di elaborare informazioni sensoriali provenienti dall'esterno. In quest'ottica di interazione con l'ambiente, l'algoritmo verrà esteso per includere compiti che richiedano il controllo della forza di contatto esercitata dal robot.

Le proprietà di stabilità e robustezza di tale controllo saranno analizzate nel dettaglio, con l'obiettivo di fornire indicazioni utili per la sintesi in applicazioni reali. La convalida della strategia adottata verrà effettuata sia in simulazione, sia sperimentalmente sul robot ABB FRIDA, sfruttando l'elevato numero di gradi di libertà della struttura a doppio braccio attraverso particolari metodi di risoluzione della ridondanza. L'approccio verrà inoltre generalizzato per implementazioni sensorless della misura di forza.





# Abstract

In recent years, the growing demand for robotized solutions in small and medium enterprises has encouraged the research of new and innovative control strategies, characterized by a high degree of flexibility and generality, which allow for a swift and efficient set-up of production processes.

Particular attention has been devoted to constraint-based trajectory generation algorithms, with the aim of translating manufacturing specifications, defined at a higher level through tasks, into reference signals for the robot.

In this thesis one of these algorithms will be used, whose advantages reside in its real-time formulation and in its ability to process external feedback signals. In this perspective of interaction with the environment, the algorithm will be extended to include tasks which require some kind of control of interaction forces.

Stability and robustness properties of the synthesized control will be analyzed in detail, with the objective of giving valuable guidelines on its application on real systems. Validation of the approach effectiveness will be given both in simulations and with experiments on the ABB robot FRIDA, whose high number of degrees of freedom and dual-arm structure will be exploited with several redundancy resolution methods. The control algorithm will also be expanded to include sensorless implementations of the force measurements.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Obiettivi della Tesi . . . . .	2
1.2	Struttura della Tesi . . . . .	4
<b>2</b>	<b>Generazione della traiettoria tramite ottimizzazione vincolata</b>	<b>7</b>
2.1	Algoritmo reattivo di pianificazione del moto . . . . .	8
2.2	Applicazione alla Tesi . . . . .	12
<b>3</b>	<b>Inversione cinematica e risoluzione della ridondanza</b>	<b>13</b>
3.1	Proiezione nello spazio nullo . . . . .	14
3.2	Solutore QP allo stato dell'arte (qpOASES) . . . . .	16
3.3	Solutore QP gerarchico dedicato (SOTH) . . . . .	20
<b>4</b>	<b>Formalizzazione dei vincoli</b>	<b>23</b>
4.1	Definizione del vincolo di forza . . . . .	24
4.1.1	Relazione tra forza e variabili di giunto . . . . .	24
4.1.2	Esplicitazione del vincolo di forza . . . . .	27
4.1.3	Analisi di stabilità e robustezza . . . . .	29
4.2	Definizione dei vincoli complementari . . . . .	36
4.2.1	Limiti sulle variabili di giunto . . . . .	36
4.2.2	Cinematica inversa . . . . .	38
4.2.3	Limiti di velocità e accelerazione . . . . .	39
4.2.4	Minimizzazione delle velocità dei giunti . . . . .	40

<b>5</b>	<b>Simulazioni</b>	<b>43</b>
5.1	Descrizione del simulatore . . . . .	44
5.1.1	Valutazione cinematica e simulazione della forza . . . . .	45
5.1.2	Movimento nello spazio dei giunti . . . . .	47
5.1.3	Avvicinamento alla superficie . . . . .	48
5.1.4	Path-following con controllo di forza . . . . .	49
5.2	SOTH1: Limiti di forza . . . . .	53
5.3	SOTH2: Riferimento di forza . . . . .	58
<b>6</b>	<b>Stima delle forze esterne</b>	<b>69</b>
6.1	Osservatore delle coppie esterne ai giunti . . . . .	69
6.2	Modellizzazione e identificazione dell'attrito . . . . .	72
6.3	Condizionamento dei segnali di coppia e forza . . . . .	75
<b>7</b>	<b>Esperimenti</b>	<b>79</b>
7.1	Descrizione dell'apparato sperimentale . . . . .	79
7.2	Scelta dei parametri . . . . .	82
7.2.1	Stima delle forze esterne . . . . .	82
7.2.2	Vincoli di forza . . . . .	83
7.2.3	Cinematica inversa . . . . .	85
7.2.4	Inseguimento dei riferimenti . . . . .	86
7.3	Risultati e correzione della stima di forza . . . . .	86
7.4	Applicazione dual-arm su ABB FRIDA . . . . .	94
<b>8</b>	<b>Robustezza dell'algorithm a disturbi geometrici</b>	<b>99</b>
8.1	Comportamento in presenza di disturbi . . . . .	99
8.2	Limitazione dinamica della velocità . . . . .	101
8.3	Risultati . . . . .	103
<b>9</b>	<b>Conclusioni</b>	<b>107</b>
	<b>Bibliografia</b>	<b>109</b>

# Elenco delle figure

1.1	ReThink Robotics Sawyer e Baxter. . . . .	3
1.2	KUKA LBR iiwa. . . . .	4
2.1	Schema a blocchi dell'algoritmo reattivo di generazione della tra- iettorie. . . . .	9
2.2	Schema a blocchi dell'algoritmo per task di path-following. . . . .	11
4.1	Rappresentazione grafica delle terne introdotte. . . . .	26
4.2	Contorno delle radici al variare di $G$ per $R = 1$ . . . . .	32
4.3	Contorno delle radici al variare di $R$ per $G = 25$ . . . . .	33
4.4	Stabilità del sistema retroazionato al variare di $R$ e $G$ . . . . .	34
4.5	Banda passante del sistema retroazionato al variare di $R$ e $G$ . . . . .	34
4.6	Risposta in frequenza al disturbo $\varepsilon_{z_0}$ agente sull'uscita. . . . .	35
4.7	Rappresentazione degli stati ammissibili per un giunto. . . . .	37
5.1	Rappresentazione grafica dell'apparato di simulazione. . . . .	45
5.2	Rappresentazione grafica delle terne utilizzate in simulazione. . . . .	50
5.3	SOTH1: Posizione relativa del tool rispetto alla superficie in con- dizioni ideali. . . . .	55
5.4	SOTH1: Forza di contatto in condizioni ideali. . . . .	55
5.5	SOTH1: Posizione relativa del tool rispetto alla superficie con incertezza geometrica. . . . .	57
5.6	SOTH1: Forza di contatto con incertezza geometrica. . . . .	57
5.7	SOTH2: Posizione relativa del tool rispetto alla superficie in con- dizioni ideali. . . . .	60

5.8	SOTH2: Forza di contatto in condizioni ideali. . . . .	60
5.9	SOTH2: Posizione relativa del tool rispetto alla superficie con incertezza geometrica. . . . .	61
5.10	SOTH2: Forza di contatto con incertezza geometrica. . . . .	61
5.11	SOTH2: Posizione relativa del tool rispetto alla superficie al va- riare di $K = K_{amb}$ , con incertezza geometrica. . . . .	62
5.12	SOTH2: Forza di contatto al variare di $K = K_{amb}$ , con incertezza geometrica. . . . .	62
5.13	SOTH2: Forza di contatto al variare della stima di $K$ , in condi- zioni ideali. . . . .	64
5.14	SOTH2: Forza di contatto al variare della stima di $K$ , con incer- tezza geometrica. . . . .	64
5.15	SOTH2: Forza di contatto al variare della stima di $K$ , con incer- tezza geometrica. . . . .	65
5.16	Zoom di Fig. 5.15. . . . .	65
5.17	SOTH2: Forza di contatto al variare di $\dot{s}_{max}$ , con incertezza geometrica. . . . .	67
6.1	Modello della forza di attrito. . . . .	73
6.2	Errore di identificazione per il giunto 2. . . . .	74
6.3	Comparazione tra coppie misurate e ricostruite per il giunto 2. . . . .	75
6.4	Forze misurate in assenza di contatto per un movimento rettilineo verticale a $0.01 \frac{m}{s}$ . . . . .	76
6.5	Rappresentazione vettoriale delle deadzone utilizzate. . . . .	77
6.6	Schema a blocchi del processo di condizionamento dei residui. . . . .	78
7.1	Robot dual-arm ABB FRIDA. . . . .	80
7.2	Apparato e configurazione utilizzati negli esperimenti. . . . .	81
7.3	SOTH2: Forza di contatto normale alla superficie. . . . .	87
7.4	SOTH2: Forze tangenziali alla superficie. . . . .	88
7.5	Rappresentazione vettoriale della correzione della stima di forza. . . . .	90
7.6	SOTH2: Posizione relativa del tool rispetto alla superficie. . . . .	91

7.7	SOTH2: Forza di contatto dopo aver effettuato la correzione. . .	91
7.8	SOTH2: Effetto della correzione sulla direzione della forza controllata. . . . .	92
7.9	SOTH2: Posizione relativa del tool rispetto alla superficie di nylon.	93
7.10	SOTH2: Forza di contatto relativa alla superficie di nylon. . . . .	93
7.11	Piastra di Petri e gripper utilizzati nell'esperimento. . . . .	94
7.12	Configurazione del braccio destro per l'esperimento dual-arm. . .	95
7.13	SOTH2 dual-arm: Posizione relativa del tool rispetto alla piastra di Petri. . . . .	96
7.14	SOTH2 dual-arm: Forza di contatto relativa alla piastra di Petri.	96
7.15	Robot a contatto durante l'esperimento dual-arm. . . . .	97
8.1	Apparato e configurazione utilizzati negli esperimenti per la robustezza. . . . .	100
8.2	SOTH2: Posizione relativa del tool rispetto alla terna della superficie non ruotata. . . . .	102
8.3	SOTH2: Forza di contatto normale alla superficie non ruotata. .	102
8.4	SOTH2: Forza di contatto normale alla superficie non ruotata. .	105
8.5	SOTH2: Velocità dell'ascissa curvilinea. . . . .	105
8.6	SOTH2: Forza di contatto normale alla superficie non ruotata (segmento). . . . .	106
8.7	SOTH2: Velocità dell'ascissa curvilinea (segmento). . . . .	106





# Elenco delle tabelle

7.1 Parametri identificati del modello di attrito per il robot ABB	
FRIDA. . . . .	83



# Capitolo 1

## Introduzione

Il numero di settori interessati dalla robotica ha subito nell'ultimo decennio un improvviso aumento. Le aree coinvolte da questo fenomeno non sono più le sole industrie manifatturiere, l'adozione di soluzioni robotizzate si è infatti estesa a un ampio spettro di nuovi contesti, che vanno dal campo medicale a quello dell'intrattenimento, dalla riabilitazione all'esplorazione spaziale e all'utilizzo, sia per scopi militari che civili, in zone ad alto rischio per operatori umani (ricognizione, esplorazione in caso di crolli, operazioni di salvataggio).

Vista la necessità di adattare i manipolatori a nuovi ambienti e applicazioni, la ricerca si è attivata per produrre soluzioni ai problemi introdotti con strutture di controllo apposite, dedicate a specifiche operazioni. Raramente, però, gli schemi sono sufficientemente flessibili da poter includere l'inserimento di nuovi obiettivi, o di considerare segnali esterni diversi da quelli per cui il controllo è progettato. Il risultato di questo tipo di approcci è stata la produzione di algoritmi estremamente efficienti e in grado di assolvere efficacemente compiti specifici, tuttavia complessi da riprogrammare in presenza di nuove condizioni dell'ambiente che circonda il robot e sprovvisti di una visione di insieme del sistema.

A questo problema di approccio risolutivo si aggiunge quello dovuto al fatto che l'inserimento di queste strategie in schemi di controllo proprietari è difficoltoso e richiede personale altamente specializzato. Queste condizioni tendono a

scoraggiare l'utilizzo di sistemi robotizzati, soprattutto nelle piccole e medie industrie, dove è necessario un veloce adattamento dei processi produttivi in base alle richieste di mercato. Per superare queste difficoltà di installazione e flessibilità, negli ultimi anni il mondo della ricerca si è messo in moto per individuare strategie innovative che non prevedano di modificare il controllo interno del robot, ma che agiscano ad un livello più esterno di pianificazione del moto e allo stesso tempo siano in grado di utilizzare le informazioni provenienti dai sensori per adattare opportunamente il movimento.

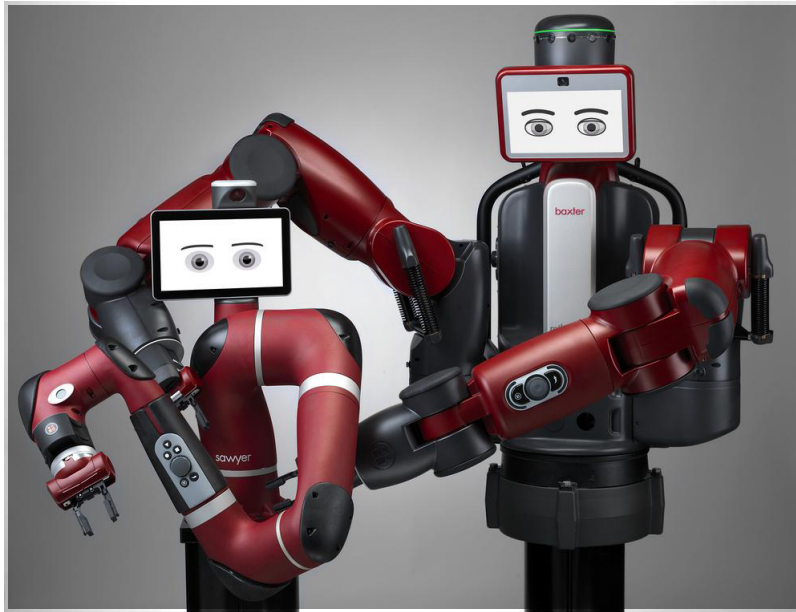
Questo tipo di approccio non è del tutto nuovo: una prima trattazione fu già proposta nel 1993, con l'introduzione delle così dette *bande elastiche* [1], poi riformulate nel 2002 [2], metodi più sistematici e di più semplice implementazione sono stati presentati solo recentemente con l'*iTaSC framework* [3] e la formulazione presentata in [4]. Queste metodologie mirano a fornire nella definizione della traiettoria il numero minimo di informazioni e di rielaborarla solo durante l'esecuzione tramite l'imposizione di vincoli pre-programmati legati all'attività specifica cui è chiamato a operare il manipolatore.

Tali metodi sono inoltre particolarmente adatti alla risoluzione della ridondanza di robot con un numero elevato di gradi di libertà e capaci di eseguire contemporaneamente diversi compiti o soddisfare obiettivi multipli.

A questo cambiamento di paradigma di programmazione si sono aggiunti il rapido aumento produttivo e diffusione di manipolatori, altamente ridondanti, orientati alla cooperazione, che per loro stessa natura necessitano di relazionarsi con l'ambiente e l'operatore umano; soprattutto in questo caso, fare ricorso a formulazioni che facilitino l'installazione e la messa a punto risulta particolarmente critico (Fig. 1.1).

## 1.1 Obiettivi della Tesi

L'adattamento di funzionalità e metodi di controllo già affermati a queste nuove circostanze è di notevole interesse. In particolare la possibilità di controllare la forza di interazione tra il robot e l'ambiente è senza dubbio fondamentale



**Figura 1.1:** *Sawyer e Baxter, i robot collaborativi di Rethink Robotics.*

e le possibili applicazioni che ne possono trarre vantaggio sono molteplici: dall'assemblaggio, all'interazione con l'uomo o altri robot, a più o meno complessi compiti di manipolazione.

L'inclusione del controllo di forza in algoritmi di generazione della traiettoria è stato recentemente affrontato su robot leggeri e votati alla cooperazione (Fig. 1.2), ma facendo sempre affidamento su schemi di controllo aggiuntivi (nello specifico controllori indiretti di forza IFC: [5], [6]), che certamente semplificano il problema, ma che mancano di generalità, non essendo applicabili indipendentemente dall'hardware e dal software di cui il robot è costituito.

Con questa tesi si vuole pertanto cercare di fornire su questo argomento una risposta sufficientemente generale da poter essere applicata all'interno di algoritmi di pianificazione del moto task-based, e tale da essere considerata autonoma, non esigendo la presenza di ulteriori strutture di controllo. Per rafforzare questo concetto di autosufficienza, il controllo di forza sarà realizzato anche in assenza di strumenti di misurazione diretta montati sul manipolatore utilizzato per gli esperimenti.

Ci si propone dunque di sintetizzare un opportuno vincolo nella pianificazio-



**Figura 1.2:** *KUKA LBR iiwa.*

ne del moto che esprima un obiettivo sul valore o sull'intervallo desiderato di forza. Attraverso studi teorici, di simulazione e sperimentali, si valuteranno stabilità e robustezza del sistema risultante, validando i risultati ottenuti con un'applicazione sul robot dual-arm ABB FRIDA.

## 1.2 Struttura della Tesi

La tesi è così suddivisa e organizzata.

Nel capitolo 2 sarà presentato l'ambiente di lavoro entro il quale si sviluppa la tesi e la struttura dell'algoritmo di pianificazione del moto utilizzato, illustrandone le principali peculiarità e vantaggi. Saranno inoltre addotte alcune motivazioni che giustificano l'ampliamento dell'algoritmo all'inclusione di un controllo di forza.

Nel capitolo 3 saranno considerati alcuni metodi di inversione cinematica attraverso i quali implementare la componente reattiva dell'algoritmo; per ogni formulazione saranno messi in evidenza pregi e limiti.

Nel capitolo 4 verranno sintetizzati alcuni vincoli utili nella definizione del blocco reattivo di generazione della traiettoria e sarà ricavato il vincolo che andrà ad implementare il controllo di forza, mostrando come la forza di contatto possa

essere relazionata alle variabili di giunto del robot. Infine, per quest'ultimo, sarà affrontata un'approfondita analisi di stabilità e robustezza rispetto ai parametri di progetto e ad eventuali incertezze.

Nel capitolo 5 i vincoli formulati in precedenza saranno verificati in simulazione e le conclusioni fatte a livello teorico su stabilità e robustezza saranno qui confermate, effettuando prove per ogni parametro e in molteplici condizioni applicative rilevanti per il controllo di forza.

Nel capitolo 6 verrà descritta una tecnica che permette l'adattamento del controllo di forza sintetizzato anche a robot non equipaggiati della sensoristica necessaria, attraverso l'opportuna identificazione del modello del robot e dell'impiego di un osservatore che rilevi le coppie ai giunti dovute al contatto con l'ambiente.

Nel capitolo 7 sarà presentato l'apparato sperimentale e il criterio di scelta dei parametri per gli esperimenti, i risultati di questi saranno quindi utilizzati per convalidare gli argomenti avanzati in teoria e simulazione. Verrà poi mostrata una possibile applicazione dual-arm sul robot ABB FRIDA.

Nel capitolo 8 saranno condotti ulteriori esperimenti per evidenziare il comportamento del controllo a fronte di disturbi dovuti alla geometria del contatto. Sarà ricavata una relazione tra velocità del task ed errore di inseguimento del riferimento di forza, che permetterà di attenuare l'effetto dei disturbi.

Nel capitolo 9 saranno tratte alcune conclusioni riepilogative sul lavoro condotto e i risultati ottenuti, offrendo qualche spunto per sviluppi futuri di miglioramento.





## Capitolo 2

# Generazione della traiettoria tramite ottimizzazione vincolata

Il paradigma di programmazione dei robot si sta spostando negli ultimi anni da un modello imperativo, in cui il programmatore ha il compito di tradurre gli obiettivi di un task direttamente in una traiettoria di riferimento espressa mediante una sequenza di comandi di moto (*move to*), a un modello dichiarativo in cui gli obiettivi sono tradotti in movimento solo a tempo di esecuzione. In questo modo sarebbe possibile includere nella generazione del moto anche la gestione di eventi o il soddisfacimento di vincoli dipendenti dalle condizioni operative e non pianificabili a priori.

Nato in ambiente di ricerca, questo approccio inizia a suscitare interesse anche in ambiente industriale, dove permetterebbe una sintesi e un'implementazione molto più veloce dei processi produttivi. A ciò va aggiunto il problema introdotto dall'architettura, solitamente chiusa, del controllo di basso livello, per cui l'introduzione di schemi che realizzino specifici comportamenti in base al feedback fornito da sensori risulta inattuabile. In queste condizioni diventa particolarmente interessante spostare la definizione degli obiettivi ad un livello più elevato tramite la creazione e l'utilizzo di vincoli direttamente in fase di pianificazione

del moto.

L'inclusione in questa struttura di segnali provenienti da sensori che possano adattare la traiettoria durante l'esecuzione secondo logiche desiderate, rende necessaria un'implementazione real-time; d'altro canto rimane l'esigenza di poter definire traiettorie in qualche misura ottimizzate, che il robot deve eseguire nel caso in cui i feedback non provochino l'attivazione di vincoli che modificherebbero il movimento.

Questo problema di generazione reattiva e dinamica è stato già investigato facendo ricorso a elementi tipici del controllo predittivo, tuttavia molte delle formulazioni proposte (e.g. [7] e [8]) pur soddisfacendo questi requisiti, risultano difficilmente attuabili in contesti online a causa della complessità computazionale, oppure prevedono la definizione di problemi di ottimizzazione non lineari includendo la dinamica del robot.

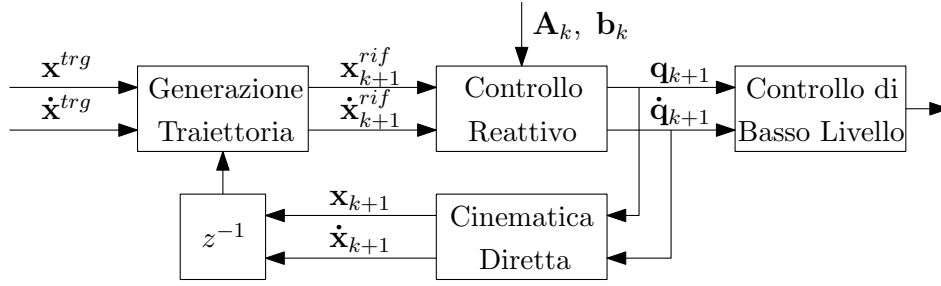
## 2.1 Algoritmo reattivo di pianificazione del moto

L'algoritmo proposto da Zanchettin e Rocco in [9], e poi perfezionato sperimentalmente in [4], offre un'efficace soluzione ai problemi presentati, caratterizzata dalla semplicità con cui il movimento del robot può essere pianificato e specifici vincoli possono essere inclusi per rielaborare dinamicamente la traiettoria di riferimento.

Il metodo in esame prevede di dividere logicamente e temporalmente l'algoritmo in due momenti: il primo, in cui viene generata nello spazio operativo una traiettoria ideale di riferimento, che, a partire dallo stato corrente del moto, garantisca l'esistenza di una soluzione che minimizzi il tempo di movimento in base ai limiti di velocità e accelerazione del manipolatore; il secondo, in cui i riferimenti da impartire al controllo di basso livello sono aggiornati in base alle informazioni prodotte dai sensori e dai particolari vincoli, istantanei e non, definiti in fase di programmazione.

Queste due funzioni sono eseguite da altrettanti blocchi funzionali ben distinti:

- Generatore della Traiettoria ideale.



**Figura 2.1:** Schema a blocchi dell'algoritmo reattivo di generazione della traiettoria.

- Controllo Reattivo.

In figura 2.1 è mostrato lo schema a blocchi complessivo dell'algoritmo.

Il primo modulo, come anticipato, ha il compito di pianificare una traiettoria che colleghi lo stato dell'istante  $k$  corrente del robot nello spazio operativo  $(\mathbf{x}_k, \dot{\mathbf{x}}_k)$  a uno stato finale specificato dall'operatore  $(\mathbf{x}^{trg}, \dot{\mathbf{x}}^{trg})$ , entrambi descritti da un insieme delle variabili di task e delle loro derivate. Il percorso generato è tale da rispettare le limitazioni imposte su velocità e accelerazioni  $(\dot{\mathbf{x}}_{max}, \ddot{\mathbf{x}}_{max})$ , e fornisce in uscita i riferimenti nello spazio operativo  $(\mathbf{x}_{k+1}^{rif}, \dot{\mathbf{x}}_{k+1}^{rif})$  che dovranno essere raggiunti all'istante successivo nel caso il blocco di Controllo Reattivo non richieda alcuna modifica del moto.

Questa operazione può essere implementata facendo uso di generatori allo stato dell'arte caratterizzati da una bassa complessità computazionale. È ad esempio questo il caso della libreria Reflexxes Type II creata da T. Kröger [10], in grado di generare online profili trapezoidali di velocità e che verrà utilizzata nell'implementazione dell'algoritmo nel resto della presente tesi.

I riferimenti cartesiani prodotti sono utilizzati in ingresso dal blocco di Controllo Reattivo, il quale, effettuando un'inversione cinematica, cerca di inseguirli pur rispettando i vincoli imposti in fase di progettazione del task e in accordo ai segnali prodotti dalla sensoristica. I risultati saranno quindi dei riferimenti aggiornati, espressi nelle velocità e posizioni di giunto  $(\mathbf{q}_{k+1}, \dot{\mathbf{q}}_{k+1})$ , che rispettano ancora le limitazioni cinematiche definite precedentemente e che saranno utilizzati dal controllo d'asse di più basso livello.

Un task generico, sia esso influenzato o meno dalle letture dei sensori, può essere

solitamente scritto in forma di disequazione (o equazione, se vale il solo simbolo di uguaglianza):

$$\mathbf{A}_k \mathbf{u}_k \leq \mathbf{b}_k \quad (2.1)$$

dove  $\mathbf{A}_k$  e  $\mathbf{b}_k$  sono quantità a priori tempo-varianti che definiscono il task e  $\mathbf{u}_k$  l'incognita rispetto alla quale il task va risolto. Nel caso dell'algoritmo in esame, poiché sono generati riferimenti in posizione e velocità, le variabili rispetto a cui effettuare l'inversione sono le accelerazioni di giunto  $\ddot{\mathbf{q}}$ , dalle quali è poi possibile ottenere i riferimenti attraverso la dinamica a tempo discreto:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + T_s \dot{\mathbf{q}}_k + \frac{T_s^2}{2} \ddot{\mathbf{q}}_k \quad (2.2)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + T_s \ddot{\mathbf{q}}_k \quad (2.3)$$

dove  $T_s$  è il tempo di campionamento del sistema di controllo real-time.

Fatte queste premesse, risulta naturale definire il blocco di Controllo Reattivo attraverso un algoritmo di inversione cinematica, per il quale esistono molteplici soluzioni realizzative che verranno approfondite nel capitolo 3. La formulazione più semplice, che permette l'utilizzo di un'ampia gamma di vincoli, è quella definita attraverso un problema di programmazione quadratica (QP):

$$\min_{\ddot{\mathbf{q}}_k} \|\dot{\mathbf{x}}_{k+1} - \dot{\mathbf{x}}_{k+1}^{rif}\|_{\mathbf{Q}_v}^2 + \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{rif}\|_{\mathbf{Q}_p}^2 \quad (2.4a)$$

soggetto a

$$\text{Vincoli cinematici: } \dot{\mathbf{q}}_{max}, \ddot{\mathbf{q}}_{max}, \dot{\mathbf{x}}_{max}, \ddot{\mathbf{x}}_{max} \quad (2.4b)$$

$$\text{Vincoli di progetto: } \mathbf{A}_k \ddot{\mathbf{q}}_k \leq \mathbf{b}_k$$

dove le  $\mathbf{Q}$  sono matrici di peso che mirano a favorire l'inseguimento di posizione o velocità fornite dal Generatore di Traiettoria.

Nota la relazione di cinematica diretta tra variabili di giunto e cartesiane è quindi possibile determinare il vettore ottimo di accelerazioni di giunto  $\ddot{\mathbf{q}}_k$ , che soddisfa i vincoli imposti e insegue al meglio i riferimenti. Questo è poi utilizzato per l'aggiornamento di posizioni e velocità da fornire al controllo d'asse attraverso la dinamica (2.2).

I valori di riferimento aggiornati saranno poi impiegati nel ciclo successivo per valutare, attraverso la cinematica diretta, lo stato corrente del robot ( $\mathbf{x}_k, \dot{\mathbf{x}}_k$ )

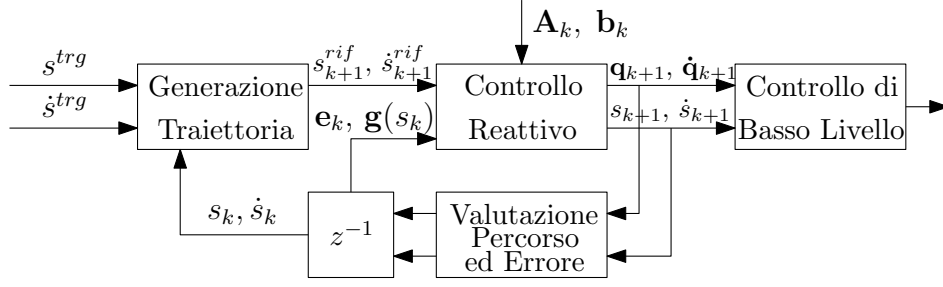


Figura 2.2: Schema a blocchi dell'algoritmo per task di path-following.

e pianificare una nuova traiettoria che colleghi questo stato a quello finale determinato in fase di progetto: così facendo si chiude il ciclo di generazione e adattamento reattivo dei riferimenti. Si osservi che la traiettoria che porta allo stato  $(\mathbf{x}^{trg}, \dot{\mathbf{x}}^{trg})$  viene ripianificata ad ogni istante di tempo.

Nel caso in cui il moto preveda un task di path-following la formulazione rimane la medesima, con la differenza che, in questo caso, la traiettoria generata sarà quella dell'ascissa curvilinea  $s$  che individua la posizione sul percorso e  $\ddot{s}$  si aggiungerà come nuova variabile di ottimizzazione. Oltre a ciò, diventa necessaria l'introduzione di un vincolo di cinematica inversa in anello chiuso che espliciti la relazione tra le variabili cartesiane  $\mathbf{x}$  e  $s$  (si veda il capitolo 4.2.2). Il nuovo blocco di Controllo Reattivo assume, dunque, forma simile a quello della (2.4):

$$\min_{\dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k} \|\dot{s}_{k+1} - \dot{s}_{k+1}^{rif}\|_{Q_v}^2 + \|s_{k+1} - s_{k+1}^{rif}\|_{Q_p}^2 \quad (2.5a)$$

$$\text{Vincoli cinematici: } \dot{\mathbf{q}}_{max}, \ddot{\mathbf{q}}_{max}, \dot{\mathbf{x}}_{max}, \ddot{\mathbf{x}}_{max}, \dot{s}_{max}, \ddot{s}_{max}$$

$$\text{soggetto a } \text{Vincolo di cinematica inversa: } \ddot{\mathbf{e}}_k + \mathbf{K}_d \dot{\mathbf{e}}_k + \mathbf{K}_p \mathbf{e}_k = 0 \quad (2.5b)$$

$$\text{Vincoli di progetto: } \mathbf{A}_k \ddot{\mathbf{q}}_k \leq \mathbf{b}_k$$

dove  $\mathbf{e}_k = \mathbf{x}_k^{rif} - \mathbf{x}_k = \mathbf{g}(s_k) - \mathbf{x}_k$  è l'errore di inseguimento, mentre  $\mathbf{g}(s_k)$  è una funzione che valuta lo stato di riferimento in base al valore dell'ascissa curvilinea. In figura 2.2 è mostrato lo schema a blocchi per compiti di path-following.

## 2.2 Applicazione alla Tesi

Data la natura reattiva e real-time, l'algoritmo descritto si presta ad essere utilizzato anche in applicazioni che prevedano il controllo della forza esercitata dal robot sull'ambiente.

Come si vedrà, è possibile introdurre all'interno del blocco di Controllo Reattivo un vincolo che, se rispettato, permette di controllare in forza una o più direzioni cartesiane del robot, senza che venga sintetizzato un controllo di basso livello che agisca tenendo conto dell'intera dinamica del robot e dei singoli giunti. Questa caratteristica è particolarmente rilevante dal punto di vista della robotica industriale, dove spesso non è possibile modificare l'algoritmo proprietario di controllo, ed è solamente lecito agire a livello di riferimento degli anelli interni. Per questa ragione la formulazione appena esaminata ha la peculiarità di avere carattere generale, indipendentemente dall'hardware su cui deve essere applicata. Estendere l'algoritmo per includere il controllo di forza oltre che di posizione sembra quindi una naturale conseguenza.

Detto questo, bisogna sottolineare come controllare in forza una direzione escluda automaticamente il controllo di posizione lungo di essa; risulterà però evidente come l'ampiezza di manovra concessa da questo metodo, permetta di definire vincoli di forza tali da permettere in una certa misura la precisazione di posizioni desiderate. Inoltre, si cercherà di capire come sfruttare all'interno di questo ambito la relazione esistente tra forza misurata e posizioni e velocità in presenza di incertezze nel caso di task con path-following.

## Capitolo 3

# Inversione cinematica e risoluzione della ridondanza

Nel capitolo 2 si è visto come il blocco di Controllo Reattivo svolga un ruolo di primaria importanza per imporre a livello di giunto i requisiti di traiettoria definiti nello spazio operativo, o di altri compiti derivanti dall'interazione del sistema con i segnali provenienti dai sensori.

La funzione svolta da questo elemento dell'algoritmo è sostanzialmente quella di produrre congrui riferimenti nello spazio dei giunti e quindi di effettuare l'inversione cinematica e la risoluzione di eventuali ridondanze. I riferimenti generati sono frutto della descrizione in termini di task del comportamento desiderato del robot, questi vengono solitamente espressi a livello di velocità o accelerazione tramite la sintesi di opportune equazioni o disequazioni.

Questo processo risulta tanto più critico quanto maggiore è il numero di compiti da eseguire e tanto più sono disponibili gradi di ridondanza. Infatti, può essere necessario instaurare una rigida gerarchia tra i vari obiettivi da conseguire e dare la precedenza a particolari finalità piuttosto che ad altre qualora si rivelassero contrastanti, ad esempio si potrebbero voler imporre dei limiti di giunto che devono essere soddisfatti ad ogni costo, indipendentemente da qualsiasi altro proposito.

Nel resto del capitolo saranno presentati alcuni metodi di inversione cine-

matica gerarchici, utilizzabili nell’algoritmo presentato all’interno del blocco di Controllo Reattivo e se ne descriveranno caratteristiche, vantaggi e svantaggi. Alcuni di questi sono stati pensati avendo in mente una risoluzione a livello di velocità di giunto, tuttavia sono facilmente riconducibili anche a tecniche operanti a livello di accelerazione [11], come nel caso della presente tesi.

### 3.1 Proiezione nello spazio nullo

Il metodo di risoluzione della ridondanza più comunemente adottato è quello proposto da Siciliano e Slotine [12], esso consiste nell’utilizzare per la risoluzione di un task a priorità inferiore i gradi di libertà rimanenti dalla soluzione ottenuta per quelli a priorità maggiore.

In generale un task viene definito attraverso un sistema lineare del tipo:

$$\mathbf{A}_1 \mathbf{x} = \mathbf{b}_1 \quad (3.1)$$

dove  $\mathbf{A}_1$  rappresenta lo jacobiano del task e  $\mathbf{x}$  il vettore di variabili rispetto alle quali si vuole effettuare la risoluzione cinematica: comunemente in campo robotico si tratta delle velocità o accelerazioni di giunto ( $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$ ). Un esempio è dato dall’espressione della cinematica diretta a livello di velocità:  $\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\mathbf{x}}$ , dove  $\mathbf{J}(\mathbf{q})$  è lo jacobiano geometrico del robot, e  $\dot{\mathbf{q}}$  e  $\dot{\mathbf{x}}$  le velocità dei giunti e nello spazio operativo rispettivamente (da non confondere con la  $\mathbf{x}$  della (3.1), che ha un altro significato).

Sia nel caso in cui il sistema (3.1) sia sovra-vincolato che sotto-vincolato l’insieme delle soluzioni è dato da quelle a norma minima:

$$\mathbf{x}_1 = \mathbf{A}_1^\dagger \mathbf{b}_1 + \mathbf{P}_1 \tilde{\mathbf{x}}_2 \quad (3.2)$$

dove  $\mathbf{A}_1^\dagger$  indica la pseudo-inversa di Moore-Penrose di  $\mathbf{A}_1$ ,  $\tilde{\mathbf{x}}_2$  un vettore che può essere usato per soddisfare obiettivi a priorità inferiore, e  $\mathbf{P}_1$  la matrice che proietta  $\tilde{\mathbf{x}}_2$  nello spazio nullo di  $\mathbf{A}_1$ , in modo da non modificare la soluzione ottima ottenuta per il primo task e utilizzare solo i gradi di libertà rimanenti. Ovviamente, nel caso in cui (3.1) sia sovra-vincolato, lo spazio nullo di  $\mathbf{A}_1$  ha



dimensione 0 e pertanto non possono essere soddisfatti altri compiti.

Volendo introdurre un ulteriore vincolo  $\mathbf{A}_2\mathbf{x} = \mathbf{b}_2$  e sostituendovi la soluzione trovata per il primo task, si ottiene:

$$\mathbf{A}_2\mathbf{P}_1\tilde{\mathbf{x}}_2 = \mathbf{b}_2 - \mathbf{A}_2\mathbf{A}_1^\dagger\mathbf{b}_1 \quad (3.3)$$

Il sistema ha ancora come soluzione quella ottenuta attraverso la pseudo-inversa:

$$\tilde{\mathbf{x}}_2 = (\mathbf{A}_2\mathbf{P}_1)^\dagger(\mathbf{b}_2 - \mathbf{A}_2\mathbf{A}_1^\dagger\mathbf{b}_1) + \tilde{\mathbf{P}}_2\tilde{\mathbf{x}}_3 \quad (3.4)$$

dove  $\tilde{\mathbf{P}}_2$  è il proiettore nello spazio nullo di  $\mathbf{A}_2\mathbf{P}_1$ .

La soluzione completa che risolve entrambi i task, sostituendo l'equazione precedente nella (3.2), è data quindi da:

$$\mathbf{x}_2 = \mathbf{A}_1^\dagger\mathbf{b}_1 + (\mathbf{A}_2\mathbf{P}_1)^\dagger(\mathbf{b}_2 - \mathbf{A}_2\mathbf{A}_1^\dagger\mathbf{b}_1) + \mathbf{P}_2\tilde{\mathbf{x}}_3 \quad (3.5)$$

dove  $\mathbf{P}_2 = \mathbf{P}_1\tilde{\mathbf{P}}_2$  è la matrice che proietta  $\tilde{\mathbf{x}}_3$  nello spazio nullo della matrice complessiva dei task:

$$\underline{\mathbf{A}}_2 = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \quad (3.6)$$

Chiaramente, qualora i task non siano in conflitto tra loro, la soluzione è più semplicemente data dalla pseudo-inversa di  $\underline{\mathbf{A}}_2$ :

$$\mathbf{x}_2 = \underline{\mathbf{A}}_2^\dagger\mathbf{b}_2 \quad (3.7)$$

Il ragionamento descritto può essere esteso a un numero arbitrario di task e quindi di livelli di priorità. Dati  $p$  vincoli, la soluzione può essere scritta ricorsivamente:

$$\mathbf{x}_p = \sum_{k=1}^p (\mathbf{A}_k\mathbf{P}_{k-1})^\dagger(\mathbf{b}_k - \mathbf{A}_k\mathbf{x}_{k-1}) + \mathbf{P}_p\tilde{\mathbf{x}}_{p+1} \quad (3.8)$$

dove  $\mathbf{P}_0 = \mathbf{I}$ ,  $\mathbf{x}_0 = \mathbf{0}$  e  $\mathbf{P}_k$  il proiettore nello spazio nullo di  $\underline{\mathbf{A}}_k$ .

Come è intuibile, l'algoritmo adottato necessita di calcolare l'inversa di alcune matrici, anche se quelle di proiezione possono essere ricavate in modo incrementale. Dal punto di vista implementativo questa operazione è solitamente effettuata facendo ricorso ad una decomposizione ai valori singolari, che, pur essendo robusta dal punto di vista numerico, risulta avere una complessità numerica tutt'altro che trascurabile.

Con lo scopo di snellire questo calcolo, in [13] si è fatto ricorso ad una decomposizione completa ortogonale che permette di essere effettuata in minor tempo computazionale anche se leggermente meno robusta. Da questa decomposizione si ottiene anche una base  $\mathbf{Z}_k$  dello spazio nullo di  $\mathbf{A}_k$ , il cui proiettore può essere scritto come:

$$\mathbf{P}_k = \mathbf{Z}_k \mathbf{Z}_k^T \quad (3.9)$$

Questo accorgimento permette di riscrivere la (3.8) in modo più efficiente, senza passare per il calcolo di  $\mathbf{P}_k$ :

$$\mathbf{x}_p = \sum_{k=1}^p \mathbf{Z}_{k-1} (\mathbf{A}_k \mathbf{Z}_{k-1})^\dagger (\mathbf{b}_k - \mathbf{A}_k \mathbf{x}_{k-1}) + \mathbf{Z}_p \mathbf{z}_{p+1} \quad (3.10)$$

dove  $\mathbf{z}_{p+1}$  è un vettore della dimensione dello spazio nullo di  $\mathbf{A}_k$ .

Le formulazioni appena descritte permettono di instaurare una rigida gerarchia tra i task da eseguire.

Pur trattandosi di procedimenti sistematici, indipendenti dal numero di vincoli che si desidera imporre al robot, rimangono delle forti limitazioni, nonostante sia possibile in una certa misura l'ottimizzazione dell'algoritmo. Infatti, rimane inattuabile la definizione di task che si possano più facilmente esprimere tramite disequazioni, come possono essere ad esempio limiti di campo visivo, o aggiramento di ostacoli; nel caso di quest'ultimo si fa solitamente ricorso a una formulazione come campo potenziale che riconduce il problema alla sintesi di un'equazione. Questo stratagemma risulta però inapplicabile quando il vincolo sotto forma di disequazione va implementato a livelli con priorità più elevata, come può accadere per la definizione di limiti di giunto.

Queste tecniche risultano dunque troppo limitanti per i requisiti che si vorrebbero imporre attraverso il Controllo Reattivo descritto nel capitolo 2.1.

## 3.2 Solutore QP allo stato dell'arte (qpOASES)

Un modo per eliminare le limitazioni imposte dal metodo di proiezione nello spazio nullo è stato proposto in [14] ed è qui discusso in relazione all'ambito di applicazione della tesi.

L'inclusione nell'insieme dei task di vincoli descritti da disuguaglianze impedisce, come già osservato, l'utilizzo dell'approccio precedente. Infatti le soluzioni ottenute da questo tipo di sistemi non descrivono un sottospazio affine a quello individuato dalle variabili di ottimizzazione, come nel caso di sole uguaglianze, delineano invece un politopo convesso, non rappresentabile attraverso pseudo-inverse. Poiché l'approccio tramite l'utilizzo di pseudo-inverse risulta difficoltoso, è quindi necessario fare ricorso a veri e propri solutori di programmazione quadratica (QP), come ad esempio quello allo stato dell'arte *qpOASES* [15].

Lo scopo è quello di risolvere una successione di problemi di ottimizzazione a partire dal primo task, propagandone la soluzione nella formulazione del problema QP successivo, che dovrà ottenere una nuova soluzione che non infici l'ottimalità della prima e allo stesso tempo soddisfi nel migliore dei modi il nuovo compito.

Di seguito verrà descritto il metodo che consente la propagazione delle soluzioni nella cascata di chiamate al solutore QP.

Si supponga di dover risolvere l'inversione cinematica per un primo task definito dall'uguaglianza  $\mathbf{A}_i \mathbf{x} = \mathbf{b}_i$ , e soggetto ad alcuni vincoli, siano essi uguaglianze o disuguaglianze, il problema che il solutore deve risolvere può essere formalizzato nel seguente modo:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_i\|^2 \quad (3.11a)$$

$$\text{soggetto a} \quad \begin{aligned} \bar{\mathbf{A}}_{i-1} \mathbf{x} &= \bar{\mathbf{b}}_{i-1} \\ \bar{\mathbf{C}}_{i-1} \mathbf{x} &\leq \bar{\mathbf{d}}_{i-1} \end{aligned} \quad (3.11b)$$

dove  $(\bar{\mathbf{A}}_{i-1}, \bar{\mathbf{b}}_{i-1})$  e  $(\bar{\mathbf{C}}_{i-1}, \bar{\mathbf{d}}_{i-1})$  sono sistemi di vincoli di equazioni e disequazioni derivanti dalla propagazione delle soluzioni di livelli a priorità più elevata. Data la soluzione ottimale del problema  $\mathbf{x}_i$ , qualora si dovesse risolvere un altro task con priorità inferiore,  $\mathbf{A}_{i+1} \mathbf{x} = \mathbf{b}_{i+1}$  (ma può essere anche una disequazione), per mantenere la gerarchia è necessario allargare il sistema di vincoli  $\bar{\mathbf{A}}_{i-1} \mathbf{x} = \bar{\mathbf{b}}_{i-1}$  con quello ulteriore dato da  $\mathbf{A}_i \mathbf{x} = \mathbf{A}_i \mathbf{x}_i$ , ottenendo il nuovo

problema complessivo da risolvere:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}_{i+1}\mathbf{x} - \mathbf{b}_{i+1}\|^2 \quad (3.12a)$$

$$\text{soggetto a} \quad \bar{\mathbf{A}}_i \mathbf{x} = \begin{bmatrix} \bar{\mathbf{A}}_{i-1} \\ \mathbf{A}_i \end{bmatrix} \mathbf{x} = \begin{bmatrix} \bar{\mathbf{b}}_{i-1} \\ \mathbf{A}_i \mathbf{x}_i \end{bmatrix} = \bar{\mathbf{b}}_i \quad (3.12b)$$

$$\bar{\mathbf{C}}_i \mathbf{x} = \bar{\mathbf{C}}_{i-1} \mathbf{x} \leq \bar{\mathbf{d}}_{i-1} = \bar{\mathbf{d}}_i$$

La soluzione di questa seconda ottimizzazione  $\mathbf{x}_{i+1}$ , è tale da mantenere l'ottimo ottenuto con il risultato precedente e da soddisfare per quanto possibile il secondo task. Iterando questo procedimento per  $p$  task, si ottiene la soluzione che risolve in modo gerarchico i  $p$  vincoli imposti.

Se la funzione desiderata per cui effettuare l'ottimizzazione è definita da una disequazione, il problema è così riformulato:

$$\min_{\mathbf{x}, \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \quad (3.13a)$$

$$\begin{aligned} & \bar{\mathbf{A}}_{i-1} \mathbf{x} = \bar{\mathbf{b}}_{i-1} \\ \text{soggetto a} \quad & \bar{\mathbf{C}}_{i-1} \mathbf{x} \leq \bar{\mathbf{d}}_{i-1} \\ & \mathbf{C}_i \mathbf{x} \leq \mathbf{d}_i + \mathbf{w} \end{aligned} \quad (3.13b)$$

dove  $\mathbf{w}$  è un vettore di variabili di rilassamento di dimensione pari a quella del task che definisce la disequazione  $i$ .

In questo caso, data la soluzione ottimale  $\mathbf{x}_i$  e volendo soddisfare un ulteriore task a priorità inferiore  $\mathbf{A}_{i+1}\mathbf{x} = \mathbf{b}_{i+1}$  (ancora una volta può essere una disequazione, per cui ci si riporta alla formulazione (3.13)), è necessario propagare la soluzione precedente nel nuovo problema.

Se il risultato  $\mathbf{x}_i$  è tale per cui qualche elemento di  $\mathbf{w}$  è non nullo, è necessario aumentare il sistema di vincoli  $\bar{\mathbf{A}}_{i-1}\mathbf{x} = \bar{\mathbf{b}}_{i-1}$  con le seguenti equazioni, derivanti dalle rispettive righe  $j$  del task per cui  $w^j \neq 0$ :

$$\mathbf{c}_i^j \mathbf{x} = \mathbf{c}_i^j \mathbf{x}_1 \quad (3.14)$$

dove  $\mathbf{c}_i^j$  indica la riga  $j$  del vincolo rappresentato da  $\mathbf{C}_i$ .

Per le righe di  $\mathbf{C}_i$  per cui  $w^j$  è nulla, è invece sufficiente allargare  $\bar{\mathbf{C}}_{i-1}\mathbf{x} \leq \bar{\mathbf{d}}_{i-1}$

con le disequazioni:

$$\mathbf{c}_i^j \mathbf{x} \leq d_i^j \quad (3.15)$$

Il procedimento di propagazione delle soluzioni di livello in livello, iterato su un numero desiderato di task, permette di ottenere la soluzione globale gerarchica dall'ultima ottimizzazione effettuata.

Qualora su uno stesso livello fossero presenti sia equazioni che disequazioni la propagazione segue il criterio descritto individualmente per i due tipi di vincolo.

La tecnica di risoluzione cinematica appena descritta permette di superare il limite imposto dall'approccio proposto in [12], consentendo la definizione e l'utilizzo di compiti più semplicemente descritti per mezzo di disequazioni e mantenendo la gerarchia desiderata. Tuttavia, anche in questo caso sono presenti alcuni svantaggi: il più facilmente intuibile è quello dato dal costo computazionale, infatti la risoluzione di un problema di programmazione quadratica prevede solitamente l'impiego di algoritmi di active-search dal costo computazionale rilevante. A ciò si aggiunge il fatto che, dati  $p$  livelli di priorità, devono essere risolti  $p$  problemi di ottimizzazione, la cui complessità totale risulta ad essi proporzionale, rendendo estremamente inefficiente l'algoritmo. Questa carenza è anche riscontrabile nel fatto che ad ogni livello di ottimizzazione viene risolto da capo lo stesso problema con l'aggiunta di pochi vincoli.

Questi difetti, sommati, rendono proibitivo l'utilizzo di questa procedura per applicazioni real-time, come quelle richieste dal nostro algoritmo. Una soluzione dalle prestazioni migliori è stata proposta in [16], ciononostante permette di definire task in forma di disequazione soltanto per il primo livello di ottimizzazione, mancando quindi di generalità.

Per aggirare questa problematica, può essere il caso di ridurre il numero di risoluzioni in cascata e applicare dei pesi ai vari task presenti nelle cifre di merito:

$$\min_{\mathbf{x}, \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|_{\mathbf{Q}_w}^2 + \frac{1}{2} \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_i\|_{\mathbf{Q}_i}^2 + \dots + \frac{1}{2} \|\mathbf{A}_k \mathbf{x} - \mathbf{b}_k\|_{\mathbf{Q}_k}^2 \quad (3.16)$$

dove le  $\mathbf{Q}$  sono matrici di peso che conferiscono diversa priorità a ogni task. Purtroppo questo approccio non è in grado di applicare una stretta gerarchia, visto che l'aggiunta dei pesi implica solo un trade-off tra i vari obiettivi.

### 3.3 Solutore QP gerarchico dedicato (SOTH)

Come si è visto, i limiti delle precedenti formulazioni rendono necessaria la realizzazione di un solutore dedicato che sia in grado di:

- Trattare task definiti sia attraverso equazioni che disequazioni.
- Produrre una soluzione che rispetti la gerarchia di obiettivi desiderata.
- Sia adatto ad una implementazione online, con tempi computazionali nell'ordine del millisecondo.

La risposta a queste richieste è stata fornita da A. Escande, N. Mansard e P.-B. Wieber in [17] con il loro solutore *SOTH* [18].

L'algoritmo prevede la definizione di uno *Stack of Tasks*, cioè di definire ogni task sotto forma di vincolo, sia esso di disuguaglianza o uguaglianza, e di ordinarli secondo i livelli di priorità assegnati.

Un algoritmo di active-search viene eseguito in un loop interno: ad ogni sua iterazione un vincolo dell'intero stack viene attivato, cioè viene aggiunto al set di vincoli per cui è valida l'uguaglianza, e ne viene calcolata la soluzione ottima; la soluzione dell'esecuzione precedente viene quindi aggiornata effettuando un passo verso quella ottima appena ottenuta. La lunghezza del passo è tale per cui la soluzione aggiornata non violi nessun vincolo che non fosse già violato in precedenza, ma tale da provocare l'attivazione di almeno uno di essi. Affinché questa si verifichi è necessario che esso rimanga violato o sia saturato (ovvero sia valido il segno di uguaglianza).

Quando tutti i vincoli sono soddisfatti o attivi, un loop esterno, che cicla sui  $p$  livelli della pila di task, controlla che i moltiplicatori di Lagrange del livello per cui si sta eseguendo il loop interno siano non negativi. Se ciò avviene si passa al livello successivo nello stack e, se i moltiplicatori sono strettamente positivi, i corrispondenti vincoli vengono trasformati in uguaglianze e quindi permanentemente inseriti nell'active set. Qualora almeno uno di essi sia strettamente negativo, il vincolo relativo a quello inferiore viene disattivato e inizia un nuovo ciclo interno.

L'algoritmo giunge al termine quando il loop esterno ha esaminato tutti i livelli dello Stack of Tasks e i moltiplicatori di Lagrange di tutti i task sono non negativi, la soluzione ottenuta è quella ai minimi quadrati che rispetta la gerarchia definita in partenza.

L'efficienza dell'algoritmo è da ricercare nella decomposizione che viene fatta della matrice  $\underline{\mathbf{A}}_p$ , cioè della matrice allargata comprendente gli jacobiani  $\mathbf{A}_i$  di tutti i vincoli dell'active set. Infatti, similmente a quanto descritto in [13] per ottenere la (3.10), la decomposizione ortogonale completa concepita è tale per cui può essere aggiornata con ridotto costo computazionale ogni qual volta un vincolo viene aggiunto o eliminato dall'active set ed è richiesto il calcolo di un nuovo ottimo dal loop interno; inoltre, essa mantiene al suo interno le informazioni relative alla priorità dei task, prendendo il nome di decomposizione ortogonale completa gerarchica (HCOD). Da questa è possibile ricavare la matrice gerarchica inversa  $\underline{\mathbf{A}}_p^\dagger$ , che permette il calcolo della soluzione ottima analogamente alla pseudo-inversa:

$$\mathbf{x}_p = \underline{\mathbf{A}}_p^\dagger \mathbf{b}_p \quad (3.17)$$

dove  $\mathbf{b}_p$  è il vettore dei termini noti dei task appartenenti all'active set. Nel caso in cui i vincoli non siano tra loro conflittuali, la matrice si riconduce semplicemente alla classica pseudo-inversa di Moore-Penrose. Questo permette di ottenere tempi computazionali fino a 10 volte più brevi rispetto al metodo descritto nella sezione 3.1.

Rispetto alla formulazione in cascata eseguibile con solutori tradizionali come qpOASES, quella appena descritta non richiede di ricalcolare, dall'inizio e per più volte, lo stesso problema di volta in volta allargato, poiché la gerarchia è tenuta in considerazione sin dall'inizio. Pertanto la complessità dell'algoritmo risulta indipendente dal numero di livelli di priorità anziché esserne proporzionale, e l'effettivo costo rimane di fatto dipendente dal solo numero di attivazioni e disattivazioni. La quantità di queste ultime può essere fatta ulteriormente diminuire in contesti online facendo ricorso a tecniche di warm start, cioè utilizzando come stato iniziale dell'algoritmo quello finale generato per la risoluzione cinematica precedente. Se il sistema è nel frattempo cambiato di poco, sarà sufficiente

un numero esiguo di attivazioni per trovare la nuova soluzione ottima globale. Queste considerazioni permettono l'utilizzo del metodo in esame per la generazione della traiettoria di robot all'interno di ambienti real-time; inoltre, vista anche la semplicità di definizione dello Stack of Tasks in fase implementativa, esso è particolarmente adatto al contesto di questa tesi e verrà utilizzato all'interno di tutte le successive simulazioni ed esperimenti.



## Capitolo 4

# Formalizzazione dei vincoli

L'utilizzo di solutori di problemi quadratici rende necessaria la formalizzazione del problema di generazione della traiettoria, in modo che il solutore sia in grado di interpretare correttamente i requisiti che si intende imporre.

Sia solutori allo stato dell'arte come qpOASES, che algoritmi di risoluzione ad hoc come SOTH, prevedono di descrivere il problema in termini di vincoli, la cui dipendenza dalle variabili di ottimizzazione, in questo caso le accelerazioni dei giunti, è espressa linearmente. Qualora si debba definire una cifra di merito, come per qpOASES, la dipendenza deve essere invece quadratica.

L'implementazione di un vincolo di forza direttamente all'interno dell'algoritmo di ottimizzazione permette di unificare la logica di generazione della traiettoria e quella di controllo di forza in un unico contesto, e di trattare il problema esattamente come uno di generazione di cammino nello spazio operativo, in una sorta di controllo ibrido forza/velocità. Inoltre, la definizione del vincolo risulta formalmente analoga a quella che si effettuerebbe per altri tipi di requisiti, ad esempio di sicurezza o di visione.

Nel seguito verrà illustrata la derivazione del vincolo di forza e di vincoli complementari, che risultano utili nella definizione generale di task.

## 4.1 Definizione del vincolo di forza

La sintesi del vincolo di forza deve essere tale da legare le forze di contatto all'end effector con le variabili di giunto del robot considerato, in modo da istituire una dipendenza diretta tra forza e  $\mathbf{q}$ . Fatto ciò, è necessario manipolare quanto ottenuto per definire effettivamente il vincolo in modo che sia compatibile con la formalizzazione richiesta dal solutore, ed esplicitare il tutto rispetto alle variabili di ottimizzazione.

### 4.1.1 Relazione tra forza e variabili di giunto

Al fine di ricavare il legame tra forza e variabili di giunto, risulta naturale relazionare la forza con le variabili geometriche dello spazio operativo attraverso un opportuno modello. In particolare, è stato utilizzato un modello puramente elastico. Nel resto della trattazione verranno considerate solo forze, ma la sintesi per i momenti che nascono a contatto è la medesima:

$$\mathbf{f} = \mathbf{K}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) = \mathbf{K}\boldsymbol{\Delta}\boldsymbol{\varepsilon} \quad (4.1)$$

dove  $\mathbf{f}$  è il vettore delle forze di contatto,  $\mathbf{K}$  la matrice di rigidità diagonale e definita positiva, e  $\boldsymbol{\Delta}\boldsymbol{\varepsilon}$  il vettore che rappresenta la posizione relativa tra i due oggetti interagenti ( $\boldsymbol{\varepsilon}$ ,  $\boldsymbol{\varepsilon}_0$ ) espressa nel punto di contatto.

In genere le variabili  $\boldsymbol{\Delta}\boldsymbol{\varepsilon}$  dipendono sia dalle variabili di giunto che dalle proprietà geometriche delle superfici tra cui avviene il contatto. Nonostante dipenda dalla particolare applicazione, queste ultime possono solitamente considerarsi costanti nel tempo, per cui la relazione può essere riscritta attraverso la derivata, ricavando un legame lineare con le velocità di giunto:

$$\boldsymbol{\Delta}\boldsymbol{\varepsilon} = h(\mathbf{q}, \mathbf{x}_{geom}) \quad \rightarrow \quad \dot{\boldsymbol{\Delta}}\boldsymbol{\varepsilon} = \frac{\partial h}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_\Delta \dot{\mathbf{q}} \quad (4.2)$$

dove  $\mathbf{J}_\Delta$  rappresenta lo jacobiano che lega la derivata del vettore delle distanze relative ( $\dot{\boldsymbol{\Delta}}\boldsymbol{\varepsilon}$ ) alle velocità dei giunti del manipolatore ( $\dot{\mathbf{q}}$ ). In sostanza ogni riga di  $\mathbf{J}_\Delta$  corrisponde a una relazione sulle velocità relative tra superficie e end-effector del robot nello spazio operativo. L'obiettivo è quindi quello di ricavare

questa matrice a partire dalla conoscenza del modello cinematico del robot, della superficie e del tipo di contatto, che nel seguito verrà considerato puntiforme. Come si vedrà nei capitoli successivi, la conoscenza geometrica della superficie di contatto gioca un ruolo rilevante nelle prestazioni ottenibili con la formalizzazione utilizzata, importanti incertezze geometriche di questo tipo possono infatti compromettere la bontà dei risultati del controllo di forza.

L'iTaSC framework proposto da De Schutter et al. [3] [19] [7] suggerisce, a questo scopo, di introdurre delle terne ausiliarie che facilitino la scrittura delle velocità relative. Questo tipo di approccio è chiaramente non univoco, e due diverse implementazioni possono portare ai medesimi risultati, tuttavia il merito del metodo è da ricercare nella versatilità e nella varietà di applicazioni per cui può essere utilizzato, oltre che nella sua adattabilità e rigore logico.

Nel seguito la superficie di contatto sarà considerata planare, ma il procedimento è facilmente estendibile a superfici generiche di geometria nota, per come sono definite le terne ausiliarie; sono inoltre trascurati fenomeni di attrito.

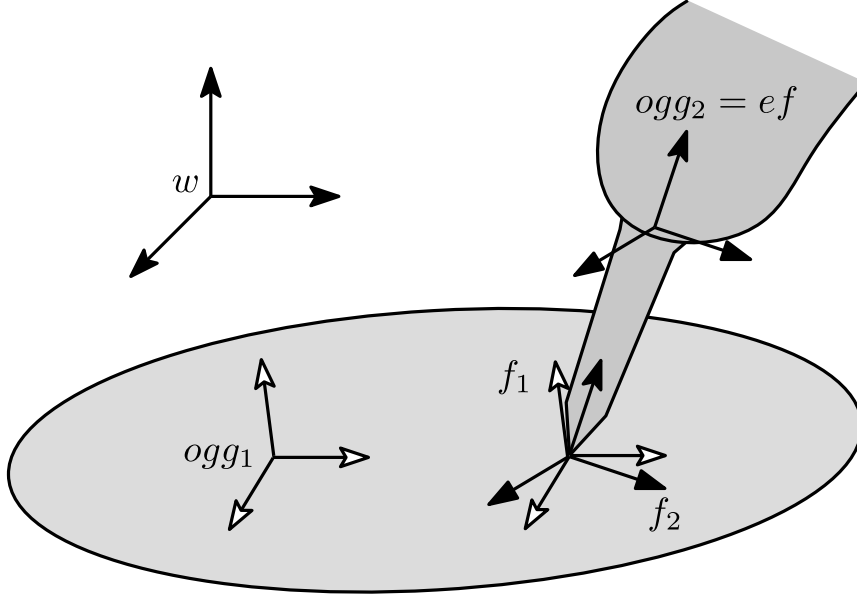
Nel caso in esame sono state introdotte le seguenti terne (Fig. 4.1):

- $ogg_1$  - Solidale alla superficie e ad essa perpendicolare con uno dei suoi assi
- $ogg_2$  - Coincidente con la terna dell'end-effector ( $ef$ )
- $f_1$  - Parallela a  $ogg_1$ , ma posizionata nel punto di contatto
- $f_2$  - Parallela a  $ogg_2$ , ma posizionata nel punto di contatto
- $w$  - Terna base del robot o terna mondo

La relazione tra le posizioni relative di superficie ed end-effector può essere ricavata facendo ricorso alle matrici di trasformazione tra le terne introdotte, tuttavia risulta più immediato effettuare il calcolo a livello di velocità attraverso gli jacobiani, da cui poi ricavare  $\mathbf{J}_\Delta$ .

L'equazione di chiusura utilizzata è la stessa riportata in [20]:

$$\mathbf{v}_{ef,w} + \mathbf{v}_{ogg_2,ef} + \mathbf{v}_{f_2,ogg_2} + \mathbf{v}_{f_1,f_2} = \mathbf{v}_{ogg_1,w} + \mathbf{v}_{f_1,ogg_1} \quad (4.3)$$



**Figura 4.1:** *Rappresentazione grafica delle terne introdotte.*

dove  $\mathbf{v}_{i,j}$  rappresenta le velocità lineari e angolari della terna  $i$  rispetto alla terna  $j$ .

Dovendo ricavare  $\dot{\Delta}\boldsymbol{\varepsilon}$ , risulta naturale esplicitare l'equazione (4.3) rispetto a  $\mathbf{v}_{f_1,ogg_1}$ , che infatti rappresenta la velocità della punta dell'end-effector rispetto alla superficie di contatto.

Viste le scelte fatte per le terne ausiliarie, poichè  $ogg_2$  ed  $ef$  sono coincidenti,  $\mathbf{v}_{ogg_2,ef}$  è il vettore nullo, così come  $\mathbf{v}_{ogg_1,w}$  se si assume che la superficie sia in posizione fissa rispetto alla base del robot, ugualmente, se la punta del tool non ha possibilità di movimento relativo rispetto all'end-effector del robot, anche  $\mathbf{v}_{f_2,ogg_2}$  è nullo. In definitiva si ottiene:

$$\mathbf{v}_{f_1,ogg_1} = \mathbf{v}_{ef,w} + \mathbf{v}_{f_1,f_2} \quad (4.4)$$

Esplicitando questa equazione rispetto alle variabili di giunto, e riferendo le grandezze all'origine di  $f_1$  con l'orientamento della terna  $ogg_1$ , l'equazione diventa:

$$\begin{bmatrix} \dot{\Delta}\boldsymbol{\varepsilon} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} = \begin{bmatrix} \dot{\Delta}\varepsilon_x \\ \dot{\Delta}\varepsilon_y \\ \dot{\Delta}\varepsilon_z \\ \mathbf{0}_{3 \times 1} \end{bmatrix} = \mathbf{P}_{f_1}^w \mathbf{M}_{f_1}^{ef} \mathbf{J}_{ef} \dot{\mathbf{q}} + \mathbf{P}_{f_1}^{f_2} \mathbf{M}_{f_1}^{f_1} \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \boldsymbol{\omega}_{f_1,f_2} \end{bmatrix} \quad (4.5)$$

dove:

$$\mathbf{P}_j^i = \begin{bmatrix} \mathbf{R}_j^i & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R}_j^i \end{bmatrix} \quad \mathbf{M}_j^i = \begin{bmatrix} \mathbf{I}_3 & \begin{bmatrix} 0 & -p_z^{j,i} & p_y^{j,i} \\ p_z^{j,i} & 0 & -p_x^{j,i} \\ -p_y^{j,i} & p_x^{j,i} & 0 \end{bmatrix} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \quad (4.6)$$

sono le matrici di cambio della terna di riferimento ( $ogg_1$ ) e del punto di riferimento ( $O_{f_1}$ ),  $\mathbf{J}_{ef}$  è invece lo jacobiano geometrico dell'end-effector del robot.

Nel caso della (4.5),  $\mathbf{M}_{f_1}^{f_1}$  è un'identità, pertanto il secondo termine non fornisce contributo a  $\Delta \dot{\boldsymbol{\varepsilon}}$ ,  $\mathbf{P}_{f_1}^w$  e  $\mathbf{M}_{f_1}^{ef}$  sono calcolabili noti l'orientamento della superficie di contatto e i parametri geometrici del tool. È da notare che gli ultimi tre termini di  $\mathbf{v}_{f_1,ogg_1}$  sono nulli per come sono state scelte le terne.

Qualora si volesse determinare la relazione tra momenti e variabili di giunto, si dovrebbe esplicitare l'equazione di chiusura (4.3) rispetto a  $\mathbf{v}_{f_1,f_2}$  e ricavare  $\Delta \dot{\boldsymbol{\Phi}}$ , dopo aver trasformato le velocità angolari in opportune derivate degli angoli di Eulero.

Riassumendo, si ottiene:

$$\mathbf{J}_\Delta = \mathbf{P}_{f_1}^w \mathbf{M}_{f_1}^{ef} \mathbf{J}_{ef} \quad (4.7)$$

Ipotizzando che la matrice di rigidità  $\mathbf{K}$  sia costante, è quindi possibile definire il legame tra la derivata della forza e le velocità di giunto:

$$\dot{\mathbf{f}} = \mathbf{K} \Delta \dot{\boldsymbol{\varepsilon}} = \mathbf{K} \mathbf{J}_\Delta \dot{\mathbf{q}} \quad (4.8)$$

A questo punto è bene sottolineare che la relazione (4.8) è già espressa nella terna  $ogg_1$  ed è quindi riferita alla superficie di contatto, inoltre, nel caso in cui si volesse applicare il controllo di forza lungo un solo asse cartesiano, sarebbe sufficiente estrarre le righe di  $\mathbf{K} \mathbf{J}_\Delta$  corrispondenti alla direzione richiesta (ad esempio la terza riga se il controllo va effettuato in direzione  $z$ ), o eventualmente effettuare la rotazione in una terna ulteriore.

#### 4.1.2 Esplicitazione del vincolo di forza

Il vincolo che si desidera applicare è del tipo:

$$\mathbf{f} = \mathbf{f}_{rif} \quad oppure \quad \mathbf{f}_{lb} \leq \mathbf{f} \leq \mathbf{f}_{ub} \quad (4.9)$$

dove, nel secondo caso, la disuguaglianza è da intendere elemento per elemento dei vettori  $\mathbf{f}$  e  $\mathbf{f}_{lb}$ ,  $\mathbf{f}_{ub}$ . La relazione ricavata nella sezione precedente è tuttavia definita solo a livello di velocità, mentre per come è stata modellizzata,  $\mathbf{f}$  è una funzione della sola  $\mathbf{q}$ . Inoltre il solutore e l'algoritmo complessivo di generazione della traiettoria necessitano di un vincolo definito a livello di accelerazione  $\ddot{\mathbf{q}}$ .

Per ottenere questo risultato è possibile riscrivere il vincolo attraverso le sue derivate, in modo che converga esponenzialmente a quello richiesto. Il processo è il medesimo sia in caso di vincolo di uguaglianza che disuguaglianza, nel secondo caso la convergenza è garantita dal lemma di Grönwall [21], mentre nel primo risulta evidente dalla risoluzione dell'equazione differenziale.

Nel seguito i calcoli saranno effettuati per vincoli di uguaglianza senza perdere comunque di generalità.

Considerato il vincolo:

$$g(\mathbf{q}, t) = \mathbf{f} - \mathbf{f}_{rif} = 0 \quad (4.10)$$

Si mostra facilmente che:

$$\delta_1 \frac{\partial g}{\partial t} = -g(\mathbf{q}, t) \quad (4.11)$$

converge esponenzialmente alla (4.10) con costante di tempo  $\delta_1$ . Ipotizzando un riferimento costante, si riscrive l'equazione nella forma:

$$\delta_1 \dot{\mathbf{f}} + \mathbf{f} = \mathbf{f}_{rif} \quad (4.12)$$

A questo punto il vincolo ottenuto dipende dalle velocità ai giunti, ma non ancora dalle accelerazioni, è quindi necessario ripetere il procedimento, imponendo:

$$g(\mathbf{q}, \dot{\mathbf{q}}, t) = \delta_1 \dot{\mathbf{f}} + \mathbf{f} - \mathbf{f}_{rif} = 0 \quad (4.13)$$

Si ottiene:

$$\delta_2 \delta_1 \ddot{\mathbf{f}} + (\delta_1 + \delta_2) \dot{\mathbf{f}} + \mathbf{f} = \mathbf{f}_{rif} \quad (4.14)$$

Ricordando l'equazione (4.8) e sostituendola in quella precedente si ricava:

$$\delta_2 \delta_1 (\mathbf{KJ}_\Delta \ddot{\mathbf{q}} + \mathbf{KJ}_\Delta \dot{\mathbf{q}}) + (\delta_2 + \delta_1) \mathbf{KJ}_\Delta \dot{\mathbf{q}} + \mathbf{f} = \mathbf{f}_{rif} \quad (4.15)$$

la quale, tenendo conto della notazione a tempo discreto utilizzata per l'algoritmo, diventa:

$$\delta_2 \delta_1 \mathbf{K} \mathbf{J}_\Delta \ddot{\mathbf{q}}_k = \mathbf{f}_{rif} - \mathbf{f}_{mis} - \mathbf{K}(\delta_2 \mathbf{J}_\Delta + \delta_2 \delta_1 \dot{\mathbf{J}}_\Delta + \delta_1 \mathbf{J}_\Delta) \dot{\mathbf{q}}_k \quad (4.16)$$

La forma ottenuta nella (4.16) è quella richiesta dagli algoritmi di ottimizzazione e converge al vincolo richiesto con costanti di tempo  $\delta_1$  e  $\delta_2$ , i termini a secondo membro sono completamente noti e  $\mathbf{f}_{mis}$  rappresenta la forza misurata o altrimenti stimata. Il vincolo può quindi essere espresso come:

$$\mathbf{A}_{forza} \ddot{\mathbf{q}}_k = \mathbf{b}_{forza} \quad (4.17)$$

dove:

$$\mathbf{A}_{forza} = \delta_2 \delta_1 \mathbf{K} \mathbf{J}_\Delta \quad (4.18)$$

$$\mathbf{b}_{forza} = \mathbf{f}_{rif} - \mathbf{f}_{mis} - \mathbf{K}(\delta_2 \mathbf{J}_\Delta + \delta_2 \delta_1 \dot{\mathbf{J}}_\Delta + \delta_1 \mathbf{J}_\Delta) \dot{\mathbf{q}}_k \quad (4.19)$$

$\delta_1$  e  $\delta_2$  sono parametri di progetto e dipendono dallo specifico manipolatore e dal task da realizzare.

Con la stessa formulazione possono essere espressi sia vincoli di uguaglianza, che permettono l'inseguimento di un riferimento di forza in una o più direzioni cartesiane, sia vincoli di disuguaglianza, che permettono di preservare un certo grado di libertà, pur mantenendo le forze di contatto limitate. Così facendo, il controllo di forza è integrato dal punto di vista logico a livello di generazione della traiettoria, evitando di dover sintetizzare un controllo di più basso livello. Ovviamente questo approccio presenta degli svantaggi, ad esempio rimane il problema di stima della rigidità di contatto. Inoltre la scelta dei parametri per ottenere prestazioni desiderate è più complessa e merita una disamina più approfondita, come si vedrà nel seguito.

### 4.1.3 Analisi di stabilità e robustezza

Per validare il vincolo sintetizzato, si è cercato di analizzarne la stabilità e la robustezza a fronte di variazioni dei parametri di progetto utilizzati  $\mathbf{K}$ ,  $\delta_1$  e  $\delta_2$  e l'effetto di disturbi agenti sulla variabile controllata di forza.

Per semplificare la trattazione verrà analizzato il caso di controllo di forza in una sola direzione. Questo assunto non è tuttavia limitante se si assume diagonale la matrice di rigidità, in questo modo, infatti, le direzioni cartesiane sono disaccoppiate.

Il vincolo definito nella (4.12) rappresenta un semplice controllo proporzionale, che mira a imporre la derivata della forza uguale all'errore di inseguimento:

$$\dot{f} = \frac{f_{rif} - f}{\delta_1} \quad (4.20)$$

È perciò di particolare interesse studiare come la scelta della costante di tempo influisca sulla stabilità.

Richiamando il modello della forza (4.1) e ipotizzando la superficie fissa e perfettamente piana ( $\varepsilon_{z_0} = costante$ ), si ottengono le seguenti espressioni per le derivate:

$$f = K(\varepsilon_z - \varepsilon_{z_0}) \quad (4.21)$$

$$\dot{f} = K\dot{\varepsilon}_z \quad (4.22)$$

$$\ddot{f} = K\ddot{\varepsilon}_z \quad (4.23)$$

Il sistema controllato può essere rappresentato in prima approssimazione da un doppio integratore, avente come stati la forza e la sua derivata prima:

$$\begin{cases} f_{k+1} = f_k + T_s \dot{f}_k + \frac{T_s^2}{2} \ddot{f}_k \\ \dot{f}_{k+1} = \dot{f}_k + T_s \ddot{f}_k \end{cases} \quad (4.24)$$

dove  $T_s$  è il tempo di campionamento del sistema.

La variabile di controllo che si è in grado di imporre attraverso la (4.14) è l'accelerazione cartesiana  $\ddot{\varepsilon}_z$ , e non la derivata seconda della forza che dipende dalla rigidità reale  $K_{amb}$ . Discretizzando il vincolo e sostituendovi le espressioni (4.22) e (4.23), si ottiene la seguente legge di controllo:

$$\ddot{\varepsilon}_{z,k} = \frac{f_{rif} - f_k - (\delta_1 + \delta_2)K\dot{\varepsilon}_{z,k}}{K\delta_1\delta_2} \quad (4.25)$$

Sostituendola insieme alle (4.22) e (4.23) nel sistema (4.24), in cui si è fatta emergere la dipendenza dalla rigidità  $K_{amb}$ , si ottiene il sistema in anello chiuso



avente come variabili di stato la forza  $f_k$  e la velocità  $\dot{\epsilon}_{z,k}$ , in cui si evidenzia il legame con la rigidezza di progetto  $K$  e la costante di tempo  $\delta_1$ . Si è inoltre assunto  $\delta_2 = T_s$ : questa semplificazione, come si vedrà nel capitolo 7.2.2, non è così restrittiva, visto che un limite inferiore prestazionale è sicuramente dato dal tempo di campionamento.

$$\begin{cases} f_{k+1} = \left(1 - \frac{T_s}{2\delta_1} \frac{K_{amb}}{K}\right) f_k + \left(T_s K_{amb} - \frac{T_s}{2\delta_1} K_{amb}(\delta_1 + T_s)\right) \dot{\epsilon}_{z,k} + \frac{T_s}{2\delta_1} \frac{K_{amb}}{K} f_{rif} \\ \dot{\epsilon}_{z,k+1} = -\frac{1}{\delta_1 K} f_k - \frac{T_s}{\delta_1} \dot{\epsilon}_{z,k} + \frac{1}{\delta_1 K} f_{rif} \end{cases} \quad (4.26)$$

Per valutarne la stabilità è stato quindi calcolato il polinomio caratteristico del sistema a tempo discreto. Evidenziando con  $G = \frac{\delta_1}{T_s}$  e  $R = \frac{K}{K_{amb}}$  i rapporti tra le costanti di tempo e le rigidezze di progetto e reale rispettivamente, si ottiene:

$$\Phi(z) = z^2 + z \frac{2R - 2GR + 1}{2GR} + \frac{1 - 2R}{2GR} \quad (4.27)$$

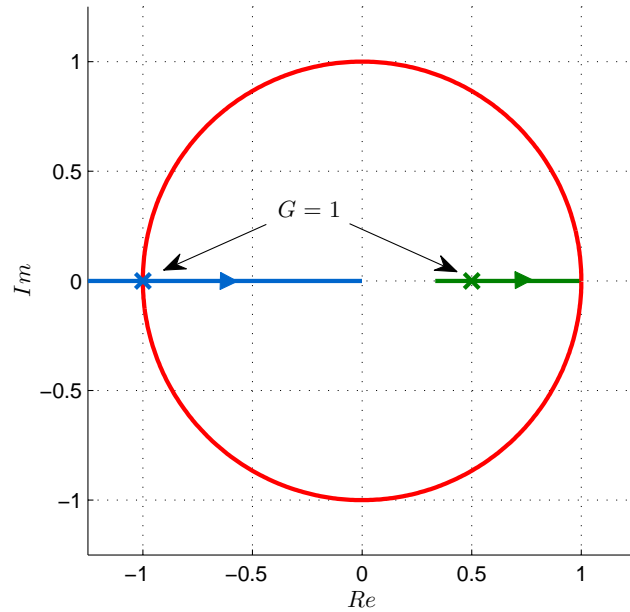
In caso di stima perfetta della rigidezza  $K = K_{amb}$  ( $R = 1$ ), l'equazione si semplifica:

$$\Phi(z) = z^2 + z \frac{3 - 2G}{2G} - \frac{1}{2G} \quad (4.28)$$

Le radici di tale polinomio sono sempre reali, tuttavia l'asintotica stabilità è garantita solo per  $\delta_1 > T_s$  ( $G > 1$ ), infatti per  $G = 1$  si arriva al limite di stabilità e si ottengono le radici  $z_1 = -1$ ,  $z_2 = \frac{1}{2}$ , come si può vedere dal contorno delle radici in figura 4.2; ciononostante soddisfare questa condizione non risulta particolarmente limitante. Al contrario, per valori di  $G$  molto elevati il sistema rimane stabile, ma il controllo diventa molto lento, visto che per  $G \rightarrow \infty$ ,  $z_1 \rightarrow 0$  e  $z_2 \rightarrow 1$ .

Sfortunatamente raggiungere una stima perfetta della rigidezza è alquanto difficile, e di conseguenza diventa estremamente importante determinare la robustezza del controllo a fronte di errori di stima.

In figura 4.3 è mostrato il contorno delle radici al variare del rapporto di stima  $R$ , fissato  $G = 25$ . In caso di severa sovrastima ( $R \gg 1$ ) il sistema rimane asintoticamente stabile, anche se si verificano cali prestazionali dovuti a uno dei poli che tende a 1 per  $R \rightarrow \infty$ . Per  $R < 1$  il sistema rimane asintoticamente



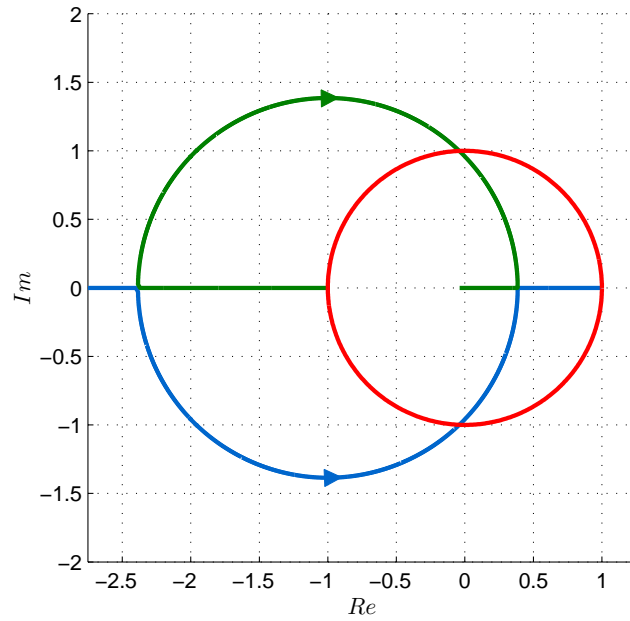
**Figura 4.2:** Contorno delle radici al variare di  $G$  per  $R = 1$ .

stabile quando la sottostima di  $K$  è leggera, anzi, si verifica un aumento delle prestazioni dovuto alla diminuzione del modulo del polo dominante. Per sottostime importanti, i poli prima di raggiungere l'instabilità diventano complessi coniugati introducendo oscillazioni nella risposta del sistema.

Se è possibile stimare in modo sufficientemente accurato  $K_{amb}$  può essere conveniente scegliere i parametri  $K$  e  $\delta_1$  in modo congiunto, così da raggiungere migliori prestazioni rispetto a quelle ottenibili operando solo sul valore di  $\delta_1$  e scegliendo  $K$  quanto più simile a  $K_{amb}$ . Si potrebbe quindi selezionare  $\delta_1$  in modo da ottenere la banda di controllo desiderata, mentre la rigidità sarebbe calibrata in modo da sopprimere eventuali oscillazioni date da poli complessi o ulteriormente migliorare la velocità di risposta.

Il grafico 4.4 mostra come cambia la stabilità del sistema al variare della scelta della coppia di parametri  $(G, R)$ .

Si può vedere come per  $G$  elevate la stabilità sia garantita per un intervallo più ampio di valori di  $R < 1$ , mentre per sovra-stime di  $K_{amb}$  la stabilità è garantita per tutte le  $G > 1$ . È possibile inoltre vedere come la presenza di poli complessi



**Figura 4.3:** Contorno delle radici al variare di  $R$  per  $G = 25$ .

segnali l'avvicinamento alla zona di instabilità (si veda di nuovo Fig. 4.3).

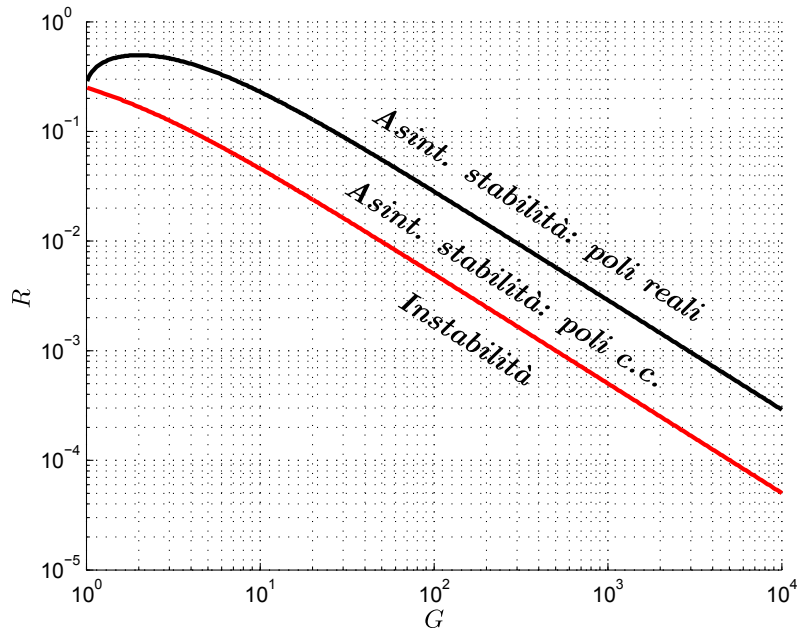
Come accennato, esiste almeno una coppia  $(G, R)$  tale da rendere asintoticamente stabile il sistema solo per  $\delta_1 > T_s$ , per valori di  $G$  minori di 1, infatti, i rami circolari descritti dai poli nella figura 4.3 si riducono a un punto posizionato in  $(-1; 0)$ , pertanto sarà sempre presente almeno un polo instabile o al limite della stabilità.

Dal sistema in anello chiuso 4.26 è possibile ricavare la funzione di sensitività complementare:

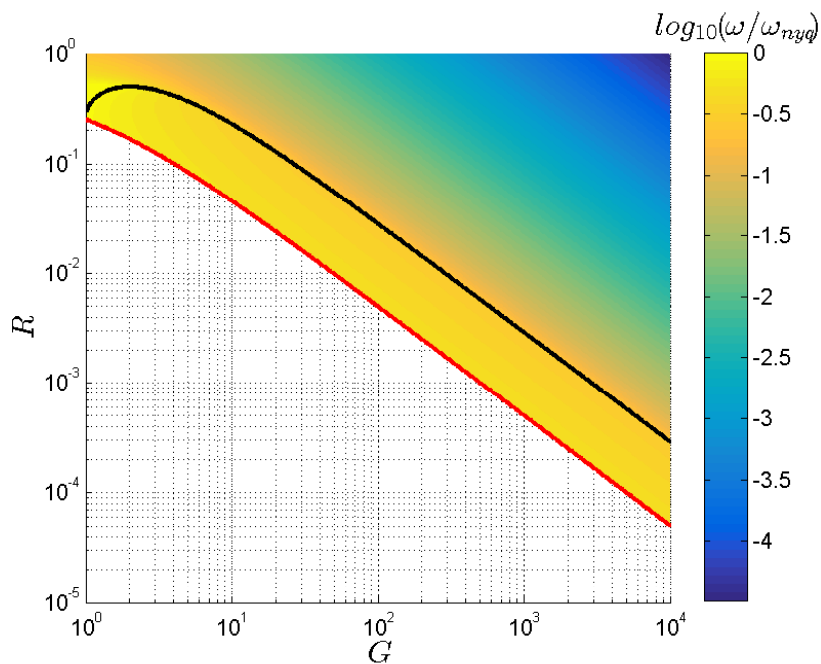
$$F(z) = \frac{f(z)}{f_{rif}(z)} = \frac{z + 1}{2GRz^2 + (2R - 2GR + 1)z + (1 - 2R)} \quad (4.29)$$

Coerentemente si ottiene guadagno statico unitario. Effettuando il calcolo della banda passante si vede come questa cresca al diminuire di  $G$  e di  $R$ , in accordo con quanto visto; la zona cui i poli sono complessi permette di ottenere bande maggiori, ma al prezzo di un forte aumento delle oscillazioni in transitorio (Fig. 4.5).

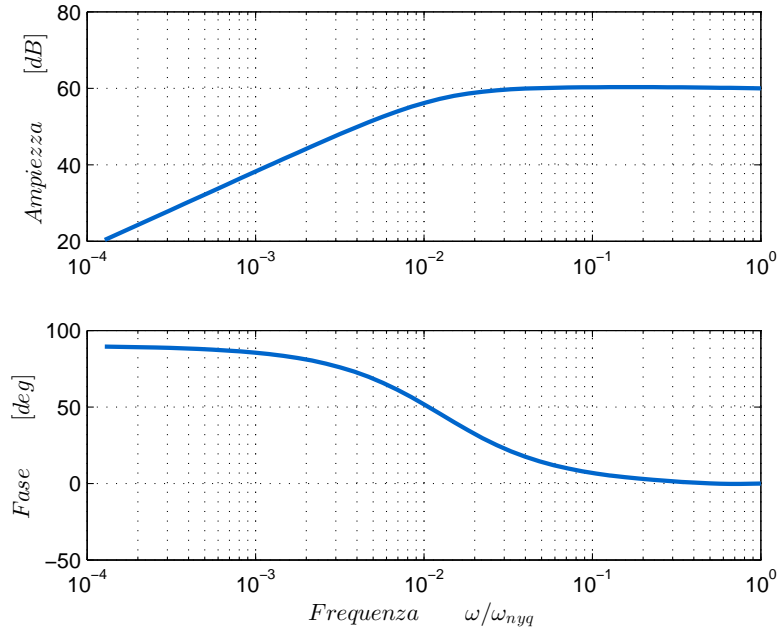
Finora si è supposta superficie piana e assenza di disturbi sull'uscita, im-



**Figura 4.4:** Stabilità del sistema retroazionato al variare di  $R$  e  $G$ .



**Figura 4.5:** Banda passante del sistema retroazionato al variare di  $R$  e  $G$ . La pulsazione sull'asse  $z$  è espressa relativamente a quella di Nyquist ( $\frac{\omega}{\omega_{nyq}}$ ) in scala logaritmica.



**Figura 4.6:** Risposta in frequenza al disturbo  $\varepsilon_{z_0}$  agente sull'uscita. La pulsazione è espressa relativamente a quella di Nyquist ( $\frac{\omega}{\omega_{nyq}}$ ). Si è ipotizzato  $K_{amb} = 1000 \frac{N}{m}$ , come mostrano i 60dB raggiunti per  $\omega \rightarrow \infty$ .  $G = 25$ ,  $R = 1$ .

ponendo nella (4.21)  $\varepsilon_{z_0} = costante$ . Volendo analizzare il comportamento del controllo a fronte di disturbi derivanti ad esempio dalla geometria di contatto, è indispensabile rilassare questa ipotesi. Poiché  $\varepsilon_{z_0}$  agisce direttamente sull'uscita a meno di una costante moltiplicativa dovuta alla rigidità dell'ambiente, la funzione di trasferimento è data semplicemente da quella di sensibilità moltiplicata per  $K_{amb}$ :

$$S(z) = \frac{f(z)}{\varepsilon_{z_0}(z)} = K_{amb} \frac{2R(z-1)(Gz+1)}{2GRz^2 + (2R - 2GR + 1)z + (1 - 2R)} \quad (4.30)$$

In figura 4.6 ne è mostrato il diagramma di Bode. La presenza dello zero in 1 permette la reiezione di disturbi a scalino, mentre disturbi a rampa, in cui si potrebbe incorrere per via di errori di orientamento, generano un errore costante sull'inseguimento del riferimento.

## 4.2 Definizione dei vincoli complementari

Nell'implementazione di un task che preveda controllo di forza in una o più direzioni, ma più in generale in un problema di generazione di traiettoria, è spesso indispensabile inserire vincoli aggiuntivi che, ad esempio, tengano conto di limiti cinematici, o che impongano l'inseguimento di una traiettoria nelle direzioni non vincolate in forza.

Nel seguito alcuni di questi vincoli sono formalizzati: essi verranno poi utilizzati in simulazione e nella validazione sperimentale finale.

### 4.2.1 Limiti sulle variabili di giunto

Alcuni limiti che è necessario considerare, qualsiasi sia il task da eseguire, sono sicuramente quelli sulle variabili di giunto, in particolare posizione minima e massima, e velocità e accelerazioni massime:

$$\mathbf{q}_{inf} \leq \mathbf{q} \leq \mathbf{q}_{sup} \quad (4.31)$$

$$-\dot{\mathbf{q}}_{max} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max} \quad (4.32)$$

$$-\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}} \leq \ddot{\mathbf{q}}_{max} \quad (4.33)$$

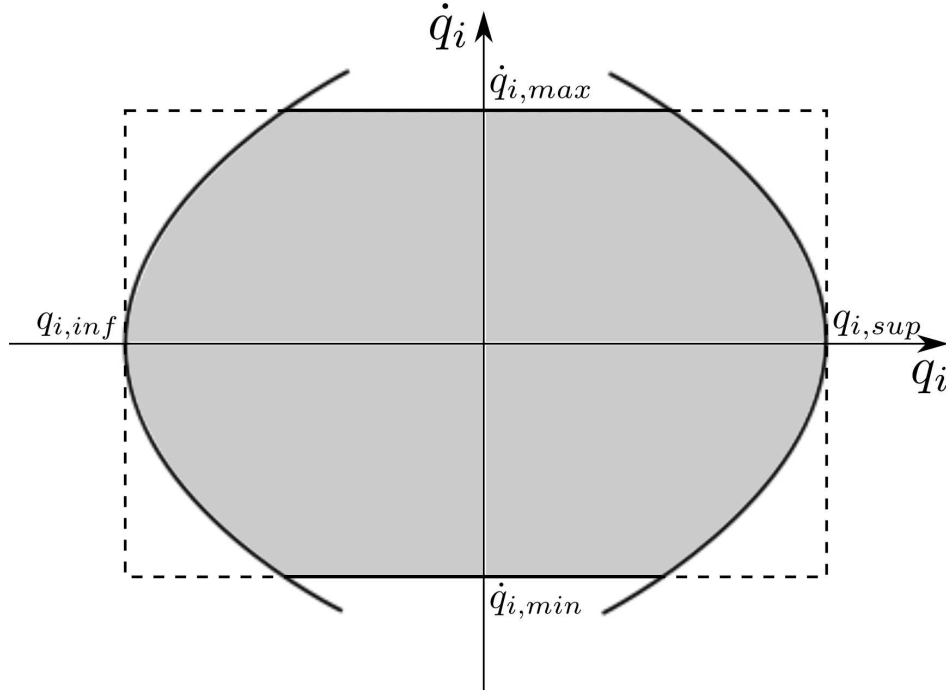
Il rispetto contemporaneo di questi vincoli assicura la fattibilità del task solo per l'istante corrente di ottimizzazione, mentre non è detto che questo sia soddisfatto all'istante successivo in cui viene risolto il problema QP. Infatti, avendo a disposizione accelerazioni finite non è possibile annullare istantaneamente le velocità, impedendo che nel tempo che intercorre tra due ottimizzazioni i limiti di giunto vengano violati.

Le relazioni che intercorrono tra posizioni, velocità e accelerazioni sono date, a tempo discreto, dalle seguenti equazioni alle differenze, dove  $T_s$  è il tempo di campionamento del controllore del robot:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + T_s \dot{\mathbf{q}}_k + \frac{T_s^2}{2} \ddot{\mathbf{q}}_k \quad (4.34)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + T_s \ddot{\mathbf{q}}_k \quad (4.35)$$

Data l'accelerazione massima, ci sono quindi istanze di  $(\mathbf{q}, \dot{\mathbf{q}})$  non accettabili, che causerebbero cioè la violazione dei limiti di posizione o velocità per il



**Figura 4.7:** Rappresentazione degli stati ammissibili per un giunto.

problema di ottimizzazione successivo, nonostante siano ammissibili prese singolarmente, dati i vincoli (4.31) e (4.32).

Facendo uso di queste relazioni, bisogna perciò sintetizzare un vincolo unico non-istantaneo, espresso nelle accelerazioni, che impedisca di violare i limiti di giunto all'istante successivo.

Date posizione e velocità correnti,  $\mathbf{q}$  deve essere limitata, in modo tale che, applicando la massima decelerazione, non vengano violati  $\mathbf{q}_{inf}$  e  $\mathbf{q}_{sup}$  al successivo istante. L'insieme di stati ammissibili è dunque delimitato da due parabole (Fig. 4.7):

$$\mathbf{q} \leq \mathbf{q}_{sup} - \frac{1}{2} \frac{\dot{\mathbf{q}}^2}{\ddot{\mathbf{q}}_{max}} \quad (4.36)$$

$$\mathbf{q} \geq \mathbf{q}_{inf} + \frac{1}{2} \frac{\dot{\mathbf{q}}^2}{\ddot{\mathbf{q}}_{max}} \quad (4.37)$$

Tenendo conto di queste e delle equazioni (4.34), (4.35), a tempo discreto i

vincoli da soddisfare sono i seguenti:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + T_s \dot{\mathbf{q}}_k + \frac{T_s^2}{2} \ddot{\mathbf{q}}_k \leq \mathbf{q}_{sup} - \frac{1}{2} \frac{(\dot{\mathbf{q}}_k + T_s \ddot{\mathbf{q}}_k)^2}{\ddot{\mathbf{q}}_{max}} \quad (4.38)$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k + T_s \dot{\mathbf{q}}_k + \frac{T_s^2}{2} \ddot{\mathbf{q}}_k \geq \mathbf{q}_{inf} + \frac{1}{2} \frac{(\dot{\mathbf{q}}_k + T_s \ddot{\mathbf{q}}_k)^2}{\ddot{\mathbf{q}}_{max}} \quad (4.39)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + T_s \ddot{\mathbf{q}}_k \leq \dot{\mathbf{q}}_{max} \quad (4.40)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + T_s \ddot{\mathbf{q}}_k \geq -\dot{\mathbf{q}}_{max} \quad (4.41)$$

$$-\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{q}}_{max} \quad (4.42)$$

Esprese le relazioni rispetto alle variabili di ottimizzazione  $\ddot{\mathbf{q}}_k$ :

$$\frac{T_s^2}{2\ddot{\mathbf{q}}_{max}} \ddot{\mathbf{q}}_k^2 + \left( \frac{T_s \dot{\mathbf{q}}_k}{\ddot{\mathbf{q}}_{max}} + \frac{T_s^2}{2} \right) \ddot{\mathbf{q}}_k + \mathbf{q}_k - \mathbf{q}_{sup} + T_s \dot{\mathbf{q}}_k + \frac{\dot{\mathbf{q}}_k^2}{2\ddot{\mathbf{q}}_{max}} \leq \mathbf{0} \quad (4.43)$$

$$\frac{T_s^2}{2\ddot{\mathbf{q}}_{max}} \ddot{\mathbf{q}}_k^2 + \left( \frac{T_s \dot{\mathbf{q}}_k}{\ddot{\mathbf{q}}_{max}} - \frac{T_s^2}{2} \right) \ddot{\mathbf{q}}_k - \mathbf{q}_k + \mathbf{q}_{inf} - T_s \dot{\mathbf{q}}_k + \frac{\dot{\mathbf{q}}_k^2}{2\ddot{\mathbf{q}}_{max}} \leq \mathbf{0} \quad (4.44)$$

$$\ddot{\mathbf{q}}_k \leq \frac{\dot{\mathbf{q}}_{max} - \dot{\mathbf{q}}_k}{T_s} = \mathbf{ub}_3 \quad (4.45)$$

$$\ddot{\mathbf{q}}_k \geq -\frac{\dot{\mathbf{q}}_{max} + \dot{\mathbf{q}}_k}{T_s} = \mathbf{lb}_3 \quad (4.46)$$

$$\mathbf{lb}_4 = -\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{q}}_{max} = \mathbf{ub}_4 \quad (4.47)$$

Risolvendo le (4.43) e (4.44), si ottengono rispettivamente  $(\mathbf{lb}_1, \mathbf{ub}_1)$  e  $(\mathbf{lb}_2, \mathbf{ub}_2)$ .

Il vincolo finale richiesto è dato quindi da:

$$\max_i \{\mathbf{lb}_i\} \leq \ddot{\mathbf{q}}_k \leq \min_i \{\mathbf{ub}_i\} \quad (4.48)$$

Il rispetto di questi vincoli garantisce di non violare alcun limite cinematico del robot per il successivo ciclo di ottimizzazione, pur generandola online.

#### 4.2.2 Cinematica inversa

In talune applicazioni potrebbe essere conveniente definire completamente il cammino percorribile dall'end-effector nello spazio operativo. Questo comportamento è solitamente definito introducendo una parametrizzazione della traiettoria:

$$\mathbf{x}_{rif} = \mathbf{g}(s) \quad (4.49)$$

dove la variabile  $s$  è un'ascissa curvilinea che rappresenta la parametrizzazione.



Velocità e accelerazioni di riferimento si possono ricavare derivando:

$$\dot{\mathbf{x}}_{rif} = \frac{\partial \mathbf{g}}{\partial s} \dot{s} = \mathbf{G}(s) \dot{s} \quad (4.50)$$

$$\ddot{\mathbf{x}}_{rif} = \dot{\mathbf{G}}(s) \dot{s} + \mathbf{G}(s) \ddot{s} \quad (4.51)$$

Con questa formulazione è necessario inserire nell'algoritmo in esame un vincolo che rappresenti l'inseguimento del riferimento e allo stesso tempo leghi alle variabili di giunto l'andamento di  $s$ . Così facendo,  $\ddot{s}$  diventa un'ulteriore variabile di ottimizzazione, che può essere scelta per controllare la posizione lungo il percorso e la velocità di percorrenza.

L'adozione di un algoritmo di cinematica inversa in anello chiuso del secondo ordine soddisfa questi requisiti. Fissati  $\mathbf{e}(\mathbf{q}) = \mathbf{x}_{rif} - \mathbf{x}(\mathbf{q})$ , e  $\mathbf{K}_p$  e  $\mathbf{K}_d$  matrici di guadagni definite positive, si ha:

$$\ddot{\mathbf{e}}(\mathbf{q}) + \mathbf{K}_d \dot{\mathbf{e}}(\mathbf{q}) + \mathbf{K}_p \mathbf{e}(\mathbf{q}) = 0 \quad (4.52)$$

Rendendo esplicita la dipendenza dalle accelerazioni:

$$\mathbf{J}_A \ddot{\mathbf{q}}_k - \mathbf{G} \ddot{s}_k = \dot{\mathbf{G}} \dot{s}_k - \dot{\mathbf{J}}_A \dot{\mathbf{q}}_k + \mathbf{K}_d (\mathbf{G} \dot{s}_k - \mathbf{J}_A \dot{\mathbf{q}}_k) + \mathbf{K}_p \mathbf{e}_k \quad (4.53)$$

Il vincolo è già nella forma richiesta, del tipo:

$$\mathbf{A}_{CLIK} \begin{bmatrix} \ddot{\mathbf{q}}_k \\ \ddot{s}_k \end{bmatrix} = \mathbf{b}_{CLIK} \quad (4.54)$$

dove:

$$\mathbf{A}_{CLIK} = \begin{bmatrix} \mathbf{J}_A & -\mathbf{G} \end{bmatrix} \quad (4.55)$$

$$\mathbf{b}_{CLIK} = \dot{\mathbf{G}} \dot{s}_k - \dot{\mathbf{J}}_A \dot{\mathbf{q}}_k + \mathbf{K}_d (\mathbf{G} \dot{s}_k - \mathbf{J}_A \dot{\mathbf{q}}_k) + \mathbf{K}_p \mathbf{e}_k \quad (4.56)$$

### 4.2.3 Limiti di velocità e accelerazione

Quando si esprime la traiettoria del robot nello spazio operativo, può essere utile affiancare ai limiti su velocità e accelerazioni dei giunti, anche delle limitazioni nello spazio cartesiano, o sull'ascissa curvilinea  $s$  nel caso il task preveda

path-following. I vincoli sono solitamente del tipo:

$$-\dot{\mathbf{x}}_{max} \leq \dot{\mathbf{x}}_{k+1} \leq \dot{\mathbf{x}}_{max} \quad (4.57)$$

$$-\ddot{\mathbf{x}}_{max} \leq \ddot{\mathbf{x}}_k \leq \ddot{\mathbf{x}}_{max} \quad (4.58)$$

$$0 \leq \dot{s}_{k+1} \leq \dot{s}_{max} \quad (4.59)$$

$$-\ddot{s}_{max} \leq \ddot{s}_k \leq \ddot{s}_{max} \quad (4.60)$$

Per esprimere i vincoli in termini di  $\ddot{\mathbf{q}}$  si richiama la dinamica a tempo discreto:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T_s \dot{\mathbf{x}}_k + \frac{T_s^2}{2} \ddot{\mathbf{x}}_k = \mathbf{x}_k + T_s \mathbf{J} \dot{\mathbf{q}}_k + \frac{T_s^2}{2} (\dot{\mathbf{J}} \dot{\mathbf{q}}_k + \mathbf{J} \ddot{\mathbf{q}}_k) \quad (4.61)$$

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + T_s \ddot{\mathbf{x}}_k = \mathbf{J} \dot{\mathbf{q}}_k + T_s (\dot{\mathbf{J}} \dot{\mathbf{q}}_k + \mathbf{J} \ddot{\mathbf{q}}_k) \quad (4.62)$$

$$s_{k+1} = s_k + T_s \dot{s}_k + \frac{T_s^2}{2} \ddot{s}_k \quad (4.63)$$

$$\dot{s}_{k+1} = \dot{s}_k + T_s \ddot{s}_k \quad (4.64)$$

Sostituendo si ricava:

$$\mathbf{lb}_1 = \frac{-\dot{\mathbf{x}}_{max} - (\mathbf{J} + T_s \dot{\mathbf{J}}) \dot{\mathbf{q}}_k}{T_s} \leq \mathbf{J} \ddot{\mathbf{q}}_k \leq \frac{\dot{\mathbf{x}}_{max} - (\mathbf{J} + T_s \dot{\mathbf{J}}) \dot{\mathbf{q}}_k}{T_s} = \mathbf{ub}_1 \quad (4.65)$$

$$\mathbf{lb}_2 = -\ddot{\mathbf{x}}_{max} - \dot{\mathbf{J}} \dot{\mathbf{q}}_k \leq \mathbf{J} \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{x}}_{max} - \dot{\mathbf{J}} \dot{\mathbf{q}}_k = \mathbf{ub}_2 \quad (4.66)$$

$$lb_3 = -\frac{\dot{s}_k}{T_s} \leq \ddot{s}_k \leq \frac{\dot{s}_{max} - \dot{s}_k}{T_s} = ub_3 \quad (4.67)$$

$$lb_4 = -\ddot{s}_{max} \leq \ddot{s}_k \leq \ddot{s}_{max} = ub_4 \quad (4.68)$$

I vincoli possono quindi essere condensati in:

$$\max\{\mathbf{lb}_1, \mathbf{lb}_2\} \leq \mathbf{J} \ddot{\mathbf{q}}_k \leq \min\{\mathbf{ub}_1, \mathbf{ub}_2\} \quad (4.69)$$

$$\max\{lb_3, lb_4\} \leq \ddot{s}_k \leq \min\{ub_3, ub_4\} \quad (4.70)$$

#### 4.2.4 Minimizzazione delle velocità dei giunti

Tipicamente, nel caso di manipolatori ridondanti, si è soliti introdurre delle condizioni aggiuntive relative ai gradi di libertà rimanenti tali da rendere deterministico il comportamento del robot durante il movimento. Un modo comune di ottenere ciò, quando si usano solutori come qpOASES, è quello di minimizzare le velocità dei giunti in un ulteriore passo di ottimizzazione. Nel caso di solutori

come SOTH, invece, anche questo livello aggiuntivo è trattato come un vincolo a priorità minore:

$$\min_{\ddot{\mathbf{q}}_k} \left( \frac{1}{2} \dot{\mathbf{q}}_{k+1}^T \dot{\mathbf{q}}_{k+1} \right) \rightarrow \dot{\mathbf{q}}_{k+1} = 0 \quad (4.71)$$

Sostituendo l'equazione (4.34) si ottiene il vincolo:

$$T_s \ddot{\mathbf{q}}_k = -\dot{\mathbf{q}}_k \quad (4.72)$$

Per altri metodi di risoluzione della ridondanza il procedimento è il medesimo, e consiste nel riscrivere la cifra da ottimizzare come vincolo.



## Capitolo 5

# Simulazioni

L'adeguatezza dei vincoli nel riprodurre i comportamenti definiti nel capitolo 4 è stata verificata all'interno dell'algoritmo sviluppato da Zanchettin e Rocco in [9], [4] e [22]. Nei test è stata inoltre messa alla prova sia la validità dell'intero algoritmo di ottimizzazione vincolata, sia la robustezza a fronte di variazioni dei parametri, in particolare del valore della rigidità di contatto.

A questo scopo sono stati definiti due task specifici:

- il robot simulato esegue un compito di path-following su una superficie, cercando di mantenere un riferimento di posizione anche in direzione normale, ma considerando dei limiti sulla forza di contatto.
- il robot simulato esegue un compito di path-following su una superficie, mantenendo un riferimento di forza in direzione normale al piano definito dal percorso.

Ci si aspetta che la traiettoria ideale di riferimento generata dall'algoritmo venga progressivamente corretta in base ai feedback ottenuti per la misura di forza, al fine di soddisfare contemporaneamente requisiti sulla posizione e sulla forza.

Sono state adottate alcune ipotesi semplificative, in particolare il contatto è stato ipotizzato puramente elastico e il robot perfettamente rigido, inoltre si è completamente trascurata la modellizzazione degli attriti dovuti allo strisciamento.

Si è assunta la superficie di contatto come perfettamente piana, e sono state poi eseguite prove di robustezza introducendo ondulazioni e imperfezioni superficiali che simulassero incertezza nella conoscenza della sua geometria.

Nel seguito del capitolo verrà illustrata la struttura del codice di simulazione impiegato, dopodiché saranno mostrati i risultati delle simulazioni relativi ai due task definiti.

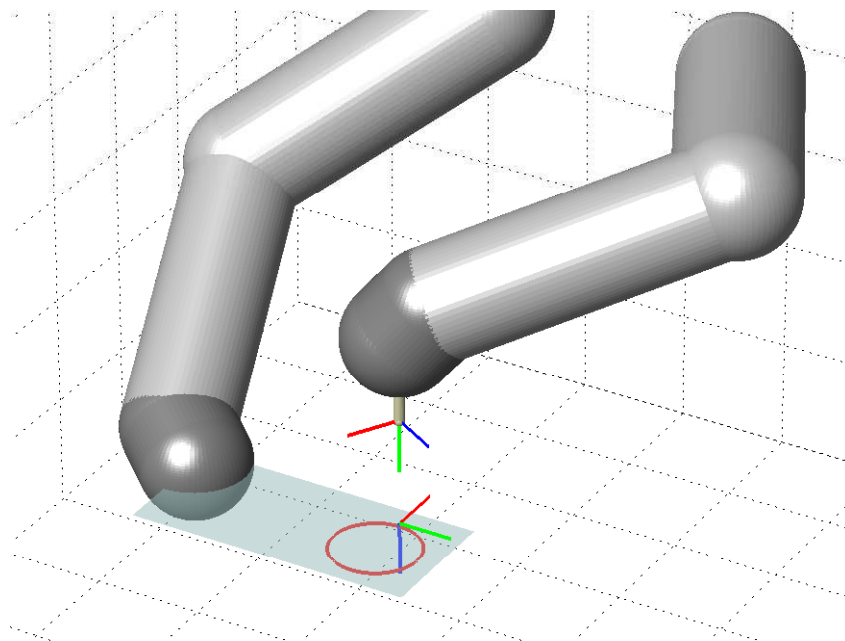
## 5.1 Descrizione del simulatore

Per la simulazione è stato utilizzato il modello cinematico del robot dual-arm ABB FRIDA (7 gradi di libertà per braccio). Per entrambi i task precedentemente descritti, la superficie su cui è eseguito il compito di path-following è sostenuta dal braccio destro del robot, mentre il braccio sinistro esegue il task vero e proprio. L'attività comprende l'uso di entrambe le braccia del robot, tuttavia, poiché in simulazione il robot è assunto perfettamente rigido e il braccio destro è mantenuto fermo e non soggetto al processo di ottimizzazione, l'applicazione può essere considerata single-arm, cioè come se si avesse un solo braccio robotico agente su una superficie non in movimento, caratterizzata dalla sua rigidità.

La forza di contatto è stata simulata solo nella direzione interessata dal controllo di forza, cioè quella normale alla superficie definita, la misura è stata ricavata dalle posizione relative di superficie e end-effector del robot, cioè dal prodotto dell'affondamento del tool nella superficie per la rigidità, come modellizzato dall'equazione (4.1).

Nella figura 5.1 è mostrata una rappresentazione grafica dell'apparato di simulazione, con le corrispondenti terne dei tool e del percorso da seguire per il braccio sinistro.

Il codice di simulazione è impostato come una macchina a stati sequenziale: a ogni stato corrisponde un sub-task che viene eseguito dal robot, il passaggio di stato avviene ogni qual volta il sub-task è stato completato o sono soddisfatte alcune condizioni di contatto.



**Figura 5.1:** *Rappresentazione grafica dell'apparato di simulazione.* I segmenti blu, rossi e verdi rappresentano gli assi  $x$ ,  $y$ ,  $z$  delle terne dei tool. In rosso è segnato il percorso descritto dal tool del braccio sinistro durante il task.

Il codice è scritto in modo tale da riprodurre quello che sarebbe il funzionamento dell'algoritmo sul robot reale, quindi a tempo discreto e col tempo di campionamento proprio del controllore del robot: ad ogni ciclo viene rieseguito il codice dello stato corrente e calcolati i nuovi riferimenti per i giunti, i cui valori di posizione e velocità vengono poi aggiornati e riutilizzati nel ciclo successivo per il calcolo della nuova traiettoria che soddisfi i requisiti inseriti in forma di vincoli nel problema di ottimizzazione; al verificarsi di particolari condizioni (adempimento del sub-task o raggiungimento di una soglia di forza), avviene il cambio di stato e il ciclo si ripete come in precedenza.

Sono ora descritti gli stati principali e le quantità che vengono calcolate e utilizzate.

### 5.1.1 Valutazione cinematica e simulazione della forza

A ogni iterazione viene valutata la cinematica diretta del robot, note le variabili di giunto, indipendentemente dallo stato in cui si trova la macchina. Per

l'applicazione in esame è stato ipotizzato l'utilizzo di un tool di lunghezza pari a 8 cm montato sul braccio sinistro, la cui terna ha lo stesso orientamento della terna dell'end-effector (lo stesso è stato effettuato per il braccio destro, dove il tool è invece dato dalla superficie su cui avverrà il task). Complessivamente la matrice di trasformazione omogenea dalla terna end-effector alla terna tool è costante ed è la seguente:

$$\mathbf{A}_{l_{tool}}^{ef} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.08 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Il calcolo dello jacobiano geometrico rispetto alla terna base è lo stesso che si avrebbe sostituendo la posizione dell'end-effector con quella del tool data dalla quarta colonna di  $\mathbf{A}_{tool}^0 = \mathbf{A}_{ef}^0 \mathbf{A}_{tool}^{ef}$ :

$$\mathbf{J}_{tool} = \begin{bmatrix} [\mathbf{z}_0 \times (\mathbf{p}_{tool} - \mathbf{p}_0)]^T & \mathbf{z}_0^T \\ [\mathbf{z}_1 \times (\mathbf{p}_{tool} - \mathbf{p}_1)]^T & \mathbf{z}_1^T \\ [\mathbf{z}_2 \times (\mathbf{p}_{tool} - \mathbf{p}_2)]^T & \mathbf{z}_2^T \\ [\mathbf{z}_3 \times (\mathbf{p}_{tool} - \mathbf{p}_3)]^T & \mathbf{z}_3^T \\ [\mathbf{z}_4 \times (\mathbf{p}_{tool} - \mathbf{p}_4)]^T & \mathbf{z}_4^T \\ [\mathbf{z}_5 \times (\mathbf{p}_{tool} - \mathbf{p}_5)]^T & \mathbf{z}_5^T \\ [\mathbf{z}_6 \times (\mathbf{p}_{tool} - \mathbf{p}_6)]^T & \mathbf{z}_6^T \end{bmatrix}^T \quad (5.2)$$

Per esprimere l'orientamento sono stati usati, così come nel resto della trattazione, gli angoli di Eulero  $XZX$  ( $\varphi, \theta, \psi$ ), a cui corrisponde la seguente matrice di trasformazione per il passaggio da jacobiano geometrico ad analitico:

$$\mathbf{T} = \begin{bmatrix} 1 & -\frac{c_\varphi c_\theta}{s_\theta} & -\frac{s_\varphi c_\theta}{s_\theta} \\ 0 & -s_\varphi & c_\varphi \\ 0 & \frac{c_\varphi}{s_\theta} & \frac{s_\varphi}{s_\theta} \end{bmatrix} \quad (5.3)$$

$$\mathbf{J}_{A,tool} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{T} \end{bmatrix} \mathbf{J}_{tool} \quad (5.4)$$

dove con  $c$  ed  $s$  si intendono coseno e seno degli angoli posti a pedice.



La simulazione della misura di forza è effettuata utilizzando come superficie di contatto il piano individuato dagli assi  $y$ ,  $z$  del tool del braccio destro (terna  $ogg_1$ ), su cui il braccio sinistro eseguirà il task; la direzione di interesse per la misura è quella lungo l'asse  $x$  di  $ogg_1$ . Usando i pedici  $r$  ed  $l$  per indicare il braccio destro e il sinistro, si ha che la posizione relativa di  $tool_l$  (terna  $ogg_2$ ) rispetto alla terna  $ogg_1$  è:

$$\tilde{\mathbf{x}}_{l_{rel}} = \mathbf{A}_{r_{tool}}^0{}^{-1} \tilde{\mathbf{x}}_l \quad (5.5)$$

dove  $\sim$  indica un vettore in coordinate omogenee  $[x \ y \ z \ 1]^T$ . La forza simulata è quindi:

$$\begin{cases} f_x = K_{amb} x_{l_{rel}}, & x_{l_{rel}} \geq 0 \\ f_x = 0, & x_{l_{rel}} < 0 \end{cases} \quad (5.6)$$

dove con  $K_{amb}$  si intende, da qui in avanti, la rigidità utilizzata per effettuare la simulazione della misurazione di forza, e rappresenta nella realtà la rigidità dell'ambiente, cioè della superficie di contatto, a priori ignota.

### 5.1.2 Movimento nello spazio dei giunti

In questo stato l'algoritmo porta il robot da una configurazione qualsiasi ad una predeterminata, compatibile con il resto del task. Ad ogni iterazione viene generata la traiettoria necessaria nello spazio dei giunti, e i corrispettivi valori delle accelerazioni da applicare, che portino le due braccia nelle posizioni fissate; questa operazione è effettuata attraverso la libreria Reflexxes Type II creata da Thorsten Kröger [10], ed è tale per cui tutti i 14 giunti del robot terminino il loro movimento nello stesso istante e nel minor tempo complessivo, in base ai valori massimi di velocità e accelerazione.

Una volta completato questo sub-task, il braccio destro si trova nella posizione in cui rimarrà per tutto il resto della simulazione, mentre il sinistro è posizionato al di sopra della superficie e col tool ad essa perpendicolare (Fig. 5.1), si passa quindi allo stato che prevede l'avvicinamento del tool alla superficie tramite un movimento rettilineo.

Questo è l'unico stato che prevede un movimento del braccio destro, nel seguito

si farà riferimento solo a movimenti del braccio sinistro, poiché, come accennato, l'applicazione è riconducibile a un compito eseguibile da un solo braccio.

### 5.1.3 Avvicinamento alla superficie

L'avvicinamento è eseguito descrivendo la traiettoria nello spazio operativo, facendo uso dell'ascissa curvilinea  $s$ : in particolare viene eseguito un segmento, mantenendo fisso l'orientamento iniziale, che vada dalla posizione iniziale della terna del tool del braccio sinistro a quella finale individuata dall'origine della terna del tool del braccio destro, che determina la superficie di contatto. Con lo scopo di simulare in qualche modo anche l'impatto tra i due tool, la posizione finale è stata corretta in modo che il braccio sinistro sia ancora in movimento nell'istante in cui avviene il contatto.

L'algoritmo di generazione della traiettoria Reflexxes opera in questo caso sull'ascissa curvilinea  $s$ , fornendo in output  $\ddot{s}_k$  e le posizioni e velocità teoriche del ciclo successivo ( $s_{k+1_{rif}}$  e  $\dot{s}_{k+1_{rif}}$ ). Per implementare il vincolo di path-following è però necessario esplicitare la relazione tra  $s$  e le variabili nello spazio operativo ( $\mathbf{p}_f$  e  $\mathbf{p}_{in}$  indicano le posizioni iniziali e finali del tool):

$$\mathbf{x}_{l,k_{rif}} = \mathbf{g}(s_k) = \begin{bmatrix} \mathbf{p}_{in} + \frac{s_k}{\|\mathbf{p}_f - \mathbf{p}_{in}\|} (\mathbf{p}_f - \mathbf{p}_{in}) \\ \Phi_{3 \times 1} \end{bmatrix} \quad (5.7)$$

$$\dot{\mathbf{x}}_{l,k_{rif}} = \mathbf{G} \dot{s}_k = \begin{bmatrix} \frac{\mathbf{p}_f - \mathbf{p}_{in}}{\|\mathbf{p}_f - \mathbf{p}_{in}\|} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \dot{s}_k \quad (5.8)$$

Gli output dati da Reflexxes sono poi utilizzati per definire il problema di ottimizzazione, che per questo sub-task dovrà soddisfare:

- Limiti cinematici di giunti e ascissa curvilinea
- Vincolo di cinematica inversa
- Inseguimento dei riferimenti  $s_{k+1_{rif}}$  e  $\dot{s}_{k+1_{rif}}$
- Minimizzazione delle velocità dei giunti

Complessivamente lo Stack of Tasks, con i relativi livelli di priorità, è definito come:

$$\begin{aligned}
 & \mathbf{q}_{inf} \leq \mathbf{q}_{k+1} \leq \mathbf{q}_{sup} \\
 1. \quad & -\dot{\mathbf{q}}_{max} \leq \dot{\mathbf{q}}_{k+1} \leq \dot{\mathbf{q}}_{max} && \text{Limiti giunti, 7 vincoli} \\
 & -\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{q}}_{max} \\
 2. \quad & \ddot{\mathbf{e}}_k + \mathbf{K}_d \dot{\mathbf{e}}_k + \mathbf{K}_p \mathbf{e}_k = 0 && \text{Cinematica inversa, 6 vincoli} \\
 3. \quad & 0 \leq \dot{s}_{k+1} \leq \dot{s}_{max} && \text{Limiti ascissa curvilinea, 1 vincolo} \\
 & -\ddot{s}_{max} \leq \ddot{s}_k \leq \ddot{s}_{max} \\
 4. \quad & \sqrt{Q_v} \dot{s}_{k+1} = \sqrt{Q_v} \dot{s}_{k+1_{rif}} && \text{Inseguimento riferimenti, 2 vincoli} \\
 & \sqrt{Q_p} s_{k+1} = \sqrt{Q_p} s_{k+1_{rif}} \\
 5. \quad & \dot{\mathbf{q}}_{k+1} = 0 && \text{Minimizzazione } \dot{\mathbf{q}}, 7 \text{ vincoli}
 \end{aligned} \tag{5.9}$$

$Q_v$  e  $Q_p$  sono scalari positivi che permettono di pesare differentemente vincoli ad uno stesso livello di priorità, come si è soliti fare con algoritmi di ottimizzazione non gerarchici. Operativamente, le derivate di matrici sono effettuate numericamente ad ogni iterazione.

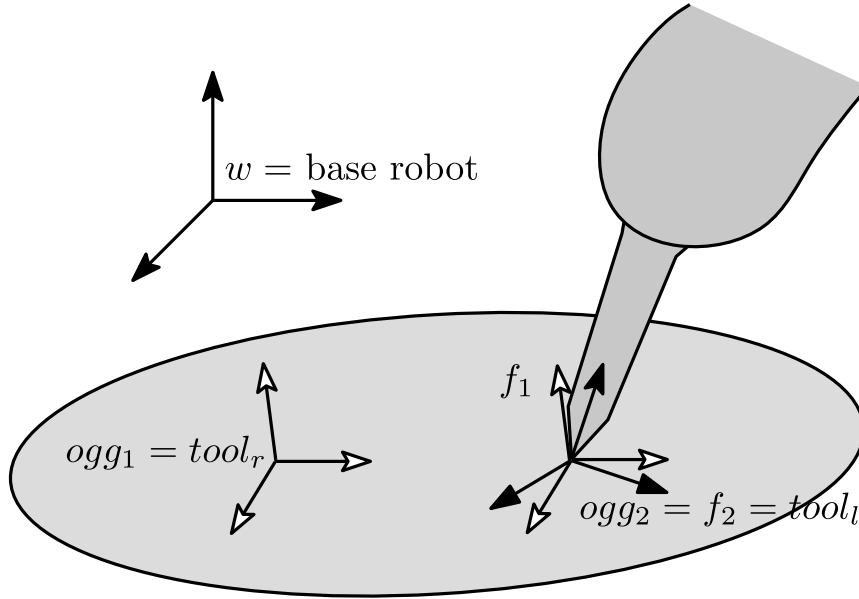
L'ottimizzazione produce come risultato le accelerazioni  $\ddot{\mathbf{q}}_k$  e  $\ddot{s}_k$ , che sono poi utilizzate per aggiornare  $\mathbf{q}_k$ ,  $\dot{\mathbf{q}}_k$ ,  $s_k$  e  $\dot{s}_k$  secondo le dinamiche presentate nelle (4.34) e (4.35), e serviranno come input al ciclo successivo dell'algoritmo.

Quando viene rilevato il contatto ( $f_x > 0$ ) lo stato cambia e si passa al task vero e proprio di controllo di forza e path-following.

#### 5.1.4 Path-following con controllo di forza

In questo stato vengono inizialmente calcolate le quantità necessarie all'implementazione del vincolo di forza e di path-following, già descritte nel capitolo 4.

Avendo già implementato nella cinematica del robot le terne dei tool (note quindi le relazioni tra terne end-effector e terne tool) e ricordando la definizione data per le terne ausiliarie di pagina 25, si può semplificare ulteriormente il problema definendo  $ogg_2$  e  $f_2$  coincidenti con la terna  $tool_1$ , avendola posizionata



**Figura 5.2:** Rappresentazione grafica delle terne utilizzate in simulazione.

nel punto in cui si realizza il contatto, e  $ogg_1$  coincidente con la terna  $tool_r$  solidale alla superficie, ottenendo complessivamente:

$$ogg_2 = f_2 = tool_l \quad \text{e} \quad ogg_1 = tool_r \quad (5.10)$$

A fronte di queste considerazioni, in figura 5.2 sono rappresentate le terne per maggiore chiarezza. Ricordando inoltre che  $f_1$  ha lo stesso punto di applicazione di  $f_2 = tool_l$ , la matrice  $M_{f_1}^{ef}$  della 4.5 diventa la  $M_{f_1}^{ogg_2} = M_{f_1}^{f_2} = M_{f_1}^{tool_l}$  e risulta essere:

$$M_{f_1}^{tool_l} = \begin{bmatrix} \mathbf{I}_3 & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} = \mathbf{I}_6 \quad (5.11)$$

Prendendo inoltre la terna base del robot come world frame si ha  $P_{f_1}^w = P_{f_1}^0 = P_{tool_r}^0$ :

$$\mathbf{P}_{tool_r}^0 = \begin{bmatrix} \mathbf{R}_{tool_r}^0 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R}_{tool_r}^0 \end{bmatrix} \quad (5.12)$$

Lo jacobiano del task è quindi semplicemente la proiezione dello jacobiano geometrico del tool del braccio sinistro nella terna della superficie di contatto:

$$\mathbf{J}_\Delta = \mathbf{P}_{tool_r}^0 \mathbf{M}_{f_1}^{tool_l} \mathbf{J}_{tool_l} = \mathbf{P}_{tool_r}^0 \mathbf{J}_{tool_l} \quad (5.13)$$

Essendo il controllo di forza effettuato solo lungo la direzione  $x$  della terna della superficie, è sufficiente estrarre la prima riga di  $\mathbf{J}_\Delta$ .

Il percorso che il tool deve compiere a contatto con la superficie è più semplicemente definito nella terna  $ogg_2$  stessa. Per rendere il vincolo di path-following consistente con questa definizione è necessario ricalcolare lo jacobiano analitico nel sistema di riferimento della superficie e riportare la posizione del tool nella stessa terna come visto nell'equazione (5.5).

A questo scopo si estraggono gli angoli di Eulero dalla matrice di rotazione della terna  $tool_l$  rispetto alla terna  $tool_r$  così calcolata:

$$\mathbf{R}_{tool_r}^{tool_l} = \mathbf{R}_{tool_r}^0 \mathbf{R}_0^{tool_l} \quad \rightarrow \quad \Phi_{rel} \quad (5.14)$$

Noti gli angoli di Eulero, lo jacobiano analitico relativo è così definito:

$$\mathbf{J}_{A,tool_l,rel} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{T}_{\Phi_{rel}} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{tool_r}^0 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R}_{tool_r}^0 \end{bmatrix} \mathbf{J}_{tool_l} \quad (5.15)$$

Una volta calcolate queste quantità cinematiche, l'algoritmo Reflexxes genera la traiettoria ideale per l'ascissa curvilinea  $s$  che individua il percorso; la relazione tra  $s$  e le variabili nello spazio operativo è la seguente:

$$\mathbf{x}_{l_{rel},k_{rif}} = \mathbf{g}(s_k) = \begin{bmatrix} 0 \\ A \cos(ks_k) + y_0 \\ A \sin(ks_k) + z_0 \\ \Phi_{rel,3 \times 1} \end{bmatrix} \quad (5.16)$$

$$\dot{\mathbf{x}}_{l_{rel},k_{rif}} = \mathbf{G} \dot{s}_k = \begin{bmatrix} 0 \\ -Ak \sin(ks_k) \\ Ak \cos(ks_k) \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \dot{s}_k \quad (5.17)$$

In questo caso si è imposto un percorso circolare di raggio  $A$ , mantenendo nuovamente costante l'orientamento. Il riferimento lungo  $x$  in realtà ha importanza relativa, poiché quando in quella direzione il controllo di forza impone un riferimento, quello in posizione è completamente trascurato.

In simulazione sono stati implementati due differenti Stack of Tasks per l'ottimizzazione: nel primo la forza viene limitata entro un certo intervallo, nel secondo viene imposto un riferimento per  $f_x$ . Anche in questo caso le accelerazioni fornite in output dal solutore sono usate per aggiornare posizioni e velocità che verranno impiegate dal ciclo di ottimizzazione successivo.

Quando il sub-task giunge al termine la macchina passa in uno stato che esegue il distacco dalla superficie in modo analogo al moto di avvicinamento, completando il movimento.

Nei successivi paragrafi sono presentati nel dettaglio gli Stack of Tasks utilizzati e i risultati ottenuti nelle simulazioni al variare di alcuni parametri.

## 5.2 SOTH1: Limiti di forza

La prima simulazione è stata effettuata inserendo solamente dei limiti inferiori e superiori sulla forza di contatto con priorità superiore al vincolo di path-following in direzione  $x$ . Quello che ci si aspetta è che il tool del braccio sinistro cerchi di inseguire al meglio il riferimento di posizione definito dalla (5.16), mantenendosi però all'interno dell'intervallo di forza ammissibile.

Nella (5.18) è illustrato lo Stack of Tasks utilizzato, dove al secondo livello di priorità, in particolare, si è fatto uso del vincolo formalizzato nella (4.16).

$$\begin{aligned}
 & \mathbf{q}_{inf} \leq \mathbf{q}_{k+1} \leq \mathbf{q}_{sup} \\
 1. \quad & -\dot{\mathbf{q}}_{max} \leq \dot{\mathbf{q}}_{k+1} \leq \dot{\mathbf{q}}_{max} && \text{Limiti giunti, 7 vincoli} \\
 & -\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{q}}_{max} \\
 2. \quad & f_{x_{inf}} \leq f_{x_{k+1}} \leq f_{x_{sup}} && \text{Limiti di forza e cinematica inversa,} \\
 & \ddot{\mathbf{e}}_k + \mathbf{K}_d \dot{\mathbf{e}}_k + \mathbf{K}_p \mathbf{e}_k = 0 && \text{1 + 5 vincoli} \\
 3. \quad & 0 \leq \dot{s}_{k+1} \leq \dot{s}_{max} && \text{Limiti ascissa curvilinea, 1 vincolo} \\
 & -\ddot{s}_{max} \leq \ddot{s}_k \leq \ddot{s}_{max} \\
 4. \quad & \ddot{e}_{x_k} + K_d \dot{e}_{x_k} + K_p e_{x_k} = 0 && \text{Riferimento di posizione} \\
 & && \text{in direzione perpendicolare, 1 vincolo} \\
 5. \quad & \sqrt{Q_v} \dot{s}_{k+1} = \sqrt{Q_v} \dot{s}_{k+1_{rif}} && \text{Inseguimento riferimenti, 2 vincoli} \\
 & \sqrt{Q_p} s_{k+1} = \sqrt{Q_p} s_{k+1_{rif}} \\
 6. \quad & \dot{\mathbf{q}}_{k+1} = 0 && \text{Minimizzazione } \dot{\mathbf{q}}, 7 \text{ vincoli}
 \end{aligned} \tag{5.18}$$

L'inserimento al livello 2 di priorità sia di vincoli di posizione che di forza potrebbe teoricamente portare a soluzioni tra loro contrastanti, il cui risultato è in realtà un trade-off delle soluzioni prese singolarmente. Ciò non accade poiché con la scelta fatta per le terne di riferimento, la direzione controllata in forza e quelle controllate in posizione sono disaccoppiate.

Al contrario, il vincolo al livello 4 impone l'inseguimento di un riferimento di posizione nella stessa direzione del vincolo di forza, tuttavia dato che è ammessa

la possibilità di variazione della forza di contatto, l'algoritmo soddisferà questo vincolo al meglio, rispettando quelli a priorità superiore.

Nel seguito sono presentati i risultati di alcune simulazioni, rappresentanti l'evoluzione dell'algoritmo di generazione della traiettoria in termini di posizione cartesiana relativa e di forza misurata, sia in condizioni ideali di completa conoscenza del sistema (geometria e rigidità della superficie) che in presenza di disturbi. Per queste prove e le successive, se non espressamente indicato, sono stati scelti i seguenti parametri per il vincolo di forza (4.16):

$$K = K_{amb} = 1000 \frac{N}{m} \quad (5.19)$$

$$\delta_1 = 0.03s \quad - \quad 0.3s \quad (5.20)$$

$$\delta_2 = 0.004s \quad (5.21)$$

$$f_{x_{inf}} = 1N \quad (5.22)$$

$$f_{x_{sup}} = 5N \quad (5.23)$$

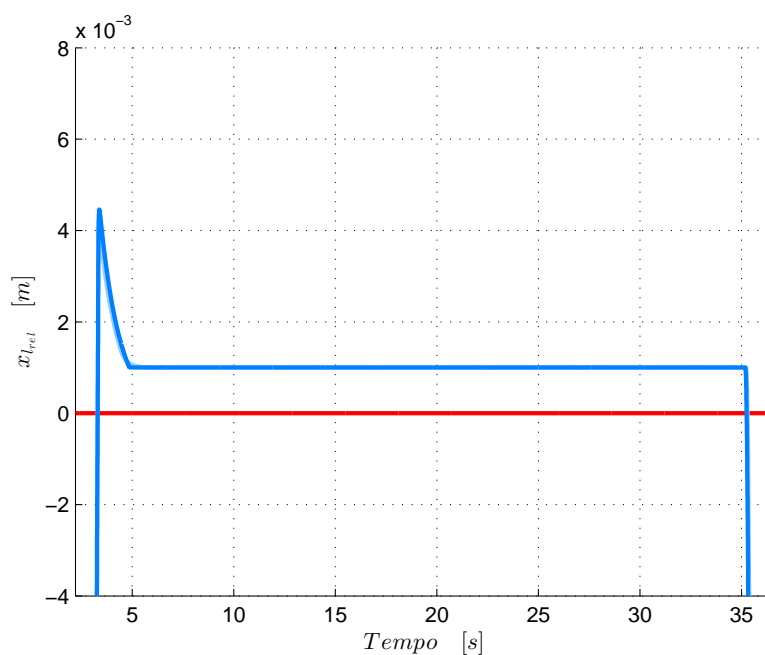
Nelle figure 5.3 e 5.4 sono mostrati i risultati ottenuti in termini di posizione e forza in direzione normale alla superficie di contatto.

La scelta di  $\delta_2$  in realtà non incide sul risultato, ammesso che sia inferiore a  $\delta_1$ , poiché il tempo di convergenza del vincolo scritto a quello ideale dipende in prima approssimazione dalla costante di tempo più elevata.

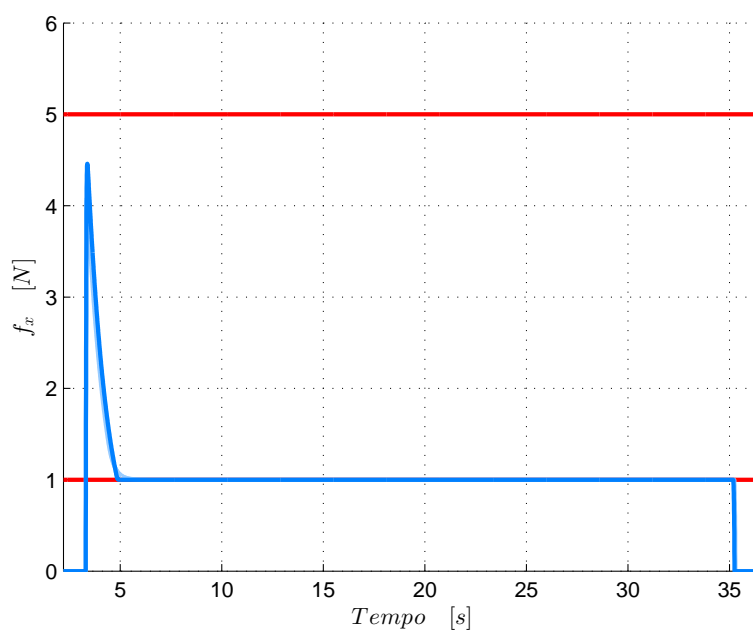
In condizioni nominali, e quindi di totale assenza di incertezze geometriche e sulla rigidità, i grafici ottenuti per i due diversi valori di  $\delta_1$  si sovrappongono, poiché la velocità di convergenza al valore di riferimento di posizione non è influenzata da questa variabile. Come ci si aspetta, si verifica inizialmente un picco di forza dovuto al contatto tra tool e superficie, dopodiché la forza misurata si assesta al livello del bound che permette un errore inferiore di posizione in direzione  $x$ , in questo modo il contatto è comunque mantenuto e la posizione inseguita nel modo migliore possibile.

In una seconda prova è stata simulata la presenza di disturbi, derivanti dalla non perfetta planarità della superficie. Per fare ciò è stato modificato il processo di simulazione della forza misurata introducendo delle ondulazioni sulla





**Figura 5.3:** *SOTH1: Posizione relativa del tool rispetto alla superficie in condizioni ideali.* Rosso: riferimento di posizione - Azzurro: andamento della posizione con  $\delta_1 = 0.3s$  - Blu: andamento della posizione con  $\delta_1 = 0.03s$ .



**Figura 5.4:** *SOTH1: Forza di contatto in condizioni ideali.* Rosso: limiti del valore della forza di contatto - Azzurro: andamento della forza con  $\delta_1 = 0.3s$  - Blu: andamento della forza con  $\delta_1 = 0.03s$ .

superficie, dipendenti dalla posizione sul percorso del tool  $s_k$ :

$$\Delta = A \cos(k s_k) \sin(s_k) = 0.005 \cos(2 s_k) \sin(s_k) \quad (5.24)$$

La misura di forza è modificata di conseguenza:

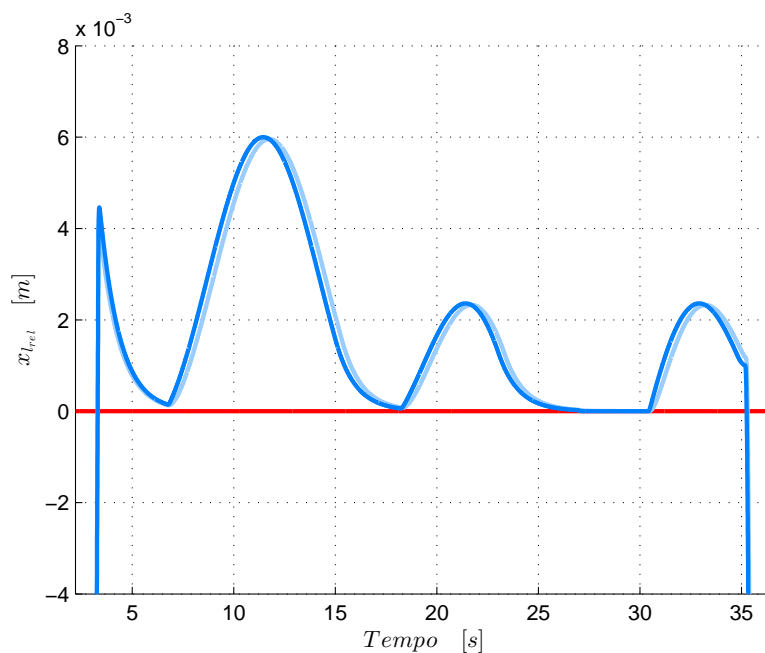
$$\begin{cases} f_x = K_{amb}(x_{l_{rel}} + \Delta), & x_{l_{rel}} + \Delta \geq 0 \\ f_x = 0, & x_{l_{rel}} + \Delta < 0 \end{cases} \quad (5.25)$$

Posizione e forza risultanti sono illustrate nelle figure 5.5 e 5.6 per i due diversi valori di  $\delta_1$ .

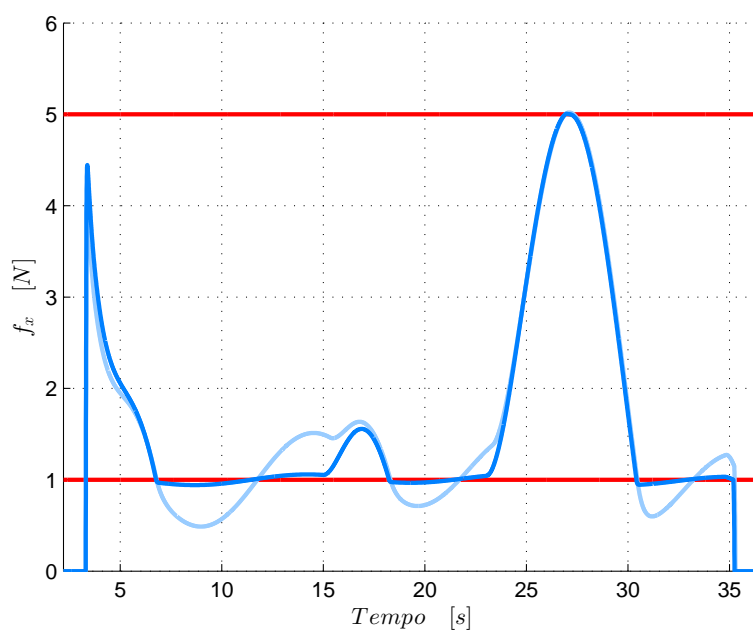
Quando la forza misurata è all'interno dei limiti la posizione converge correttamente al riferimento come si vede nell'intervallo 25 – 30s, mentre quando l'inseguimento di posizione comporterebbe una violazione dei limiti di forza si dà la precedenza al rispetto di questi ultimi.

L'introduzione di incertezza geometrica fa emergere la differenza di prestazioni ottenibili variando  $\delta_1$ . Come è intuibile, una costante di tempo più bassa permette di avere migliori prestazioni, poiché il vincolo converge a quello ideale più velocemente, permettendo di attenuare il disturbo rappresentato dalla variazione di altezza della superficie rispetto a quella ideale piana. L'utilizzo di una costante di tempo elevata provoca l'aumento di oscillazioni della forza quando il vincolo viene attivato e quindi una violazione più o meno importante del vincolo ideale.

Questi risultati sono concordi con lo studio effettuato nel capitolo 4.1.3: è auspicabile l'utilizzo di una  $\delta_1$  più piccola possibile, che aumenti la banda di controllo attenuando più efficacemente i disturbi. La scelta del particolare robot introdurrà limiti inferiori per questa variabile nell'applicazione sperimentale, che verranno analizzati in seguito.



**Figura 5.5:** *SOTH1: Posizione relativa del tool rispetto alla superficie con incertezza geometrica.* Rosso: riferimento di posizione - Azzurro: andamento della posizione con  $\delta_1 = 0.3s$  - Blu: andamento della posizione con  $\delta_1 = 0.03s$ .



**Figura 5.6:** *SOTH1: Forza di contatto con incertezza geometrica.* Rosso: limiti del valore della forza di contatto - Azzurro: andamento della forza con  $\delta_1 = 0.3s$  - Blu: andamento della forza con  $\delta_1 = 0.03s$ .

### 5.3 SOTH2: Riferimento di forza

Nella seconda simulazione il riferimento di posizione in direzione normale è stato sostituito con un vincolo di forza per inseguire un riferimento costante. Lo Stack of Tasks è definito nella (5.26), si può vedere come vari solo il livello 4 di priorità rispetto al caso precedente, è stato infatti qui introdotto il riferimento di forza in accordo alla (4.16).

$$\begin{aligned}
 & \mathbf{q}_{inf} \leq \mathbf{q}_{k+1} \leq \mathbf{q}_{sup} \\
 1. \quad & -\dot{\mathbf{q}}_{max} \leq \dot{\mathbf{q}}_{k+1} \leq \dot{\mathbf{q}}_{max} && \text{Limiti giunti, 7 vincoli} \\
 & -\ddot{\mathbf{q}}_{max} \leq \ddot{\mathbf{q}}_k \leq \ddot{\mathbf{q}}_{max} \\
 2. \quad & f_{x_{inf}} \leq f_{x_{k+1}} \leq f_{x_{sup}} && \text{Limiti di forza e cinematica inversa,} \\
 & \ddot{\mathbf{e}}_k + \mathbf{K}_d \dot{\mathbf{e}}_k + \mathbf{K}_p \mathbf{e}_k = 0 && \text{1 + 5 vincoli} \\
 3. \quad & 0 \leq \dot{s}_{k+1} \leq \dot{s}_{max} && \text{Limiti ascissa curvilinea, 1 vincolo} \\
 & -\ddot{s}_{max} \leq \ddot{s}_k \leq \ddot{s}_{max} \\
 4. \quad & f_{x_{k+1}} = f_{x_{rif}} && \text{Riferimento di forza} \\
 & && \text{in direzione perpendicolare, 1 vincolo} \\
 5. \quad & \sqrt{Q_v} \dot{s}_{k+1} = \sqrt{Q_v} \dot{s}_{k+1_{rif}} && \text{Inseguimento riferimenti, 2 vincoli} \\
 & \sqrt{Q_p} s_{k+1} = \sqrt{Q_p} s_{k+1_{rif}} \\
 6. \quad & \dot{\mathbf{q}}_{k+1} = 0 && \text{Minimizzazione } \dot{\mathbf{q}}, 7 \text{ vincoli}
 \end{aligned} \tag{5.26}$$

Anche in questo caso, se non espressamente scritto, sono stati utilizzati i seguenti parametri:

$$K = K_{amb} = 1000 \frac{N}{m} \tag{5.27}$$

$$\delta_1 = 0.03s \quad - \quad 0.3s \quad - \quad 3s \tag{5.28}$$

$$\delta_2 = 0.004s \tag{5.29}$$

$$f_{x_{inf}} = 1N \tag{5.30}$$

$$f_{x_{sup}} = 5N \tag{5.31}$$

$$f_{x_{rif}} = 3N \tag{5.32}$$

Ancora una volta, essendo le direzioni controllate in posizione e quelle controllate in forza disaccoppiate, dato il sistema di riferimento utilizzato, non nascono conflitti dalla definizione dei vincoli, pertanto ci si aspetta che il riferimento di forza venga seguito senza alcun problema.

Nelle figure 5.7 e 5.8 è mostrato l'andamento di posizione e forza in condizioni nominali di superficie perfettamente piana al variare di  $\delta_1$ . La posizione è automaticamente fissata dato il riferimento di forza, e pertanto non c'è più un margine entro cui la posizione possa convergere al valore di riferimento.

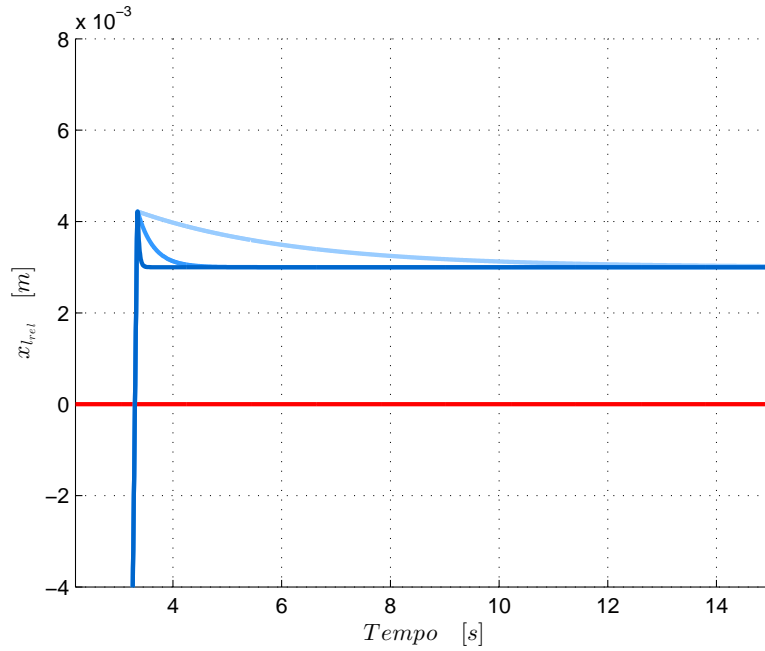
Come si poteva prevedere, una diminuzione della costante di tempo  $\delta_1$  aumenta la velocità con cui la forza raggiunge il riferimento assegnato, evidenziando un netto aumento di prestazioni.

Introducendo nuovamente incertezza geometrica sulla superficie come effettuato per il task SOTH1 con la (5.24), dalle figure 5.9 e 5.10, ci si rende conto di come il degrado delle prestazioni aumenti velocemente con l'aumentare della costante di tempo del vincolo di forza, che non è in grado di fronteggiare i repentini cambiamenti della concavità della superficie, introducendo oscillazioni sempre più ampie, fino a rendere il risultato totalmente inaccettabile, seppur congruente con l'analisi effettuata sull'effetto dei disturbi.

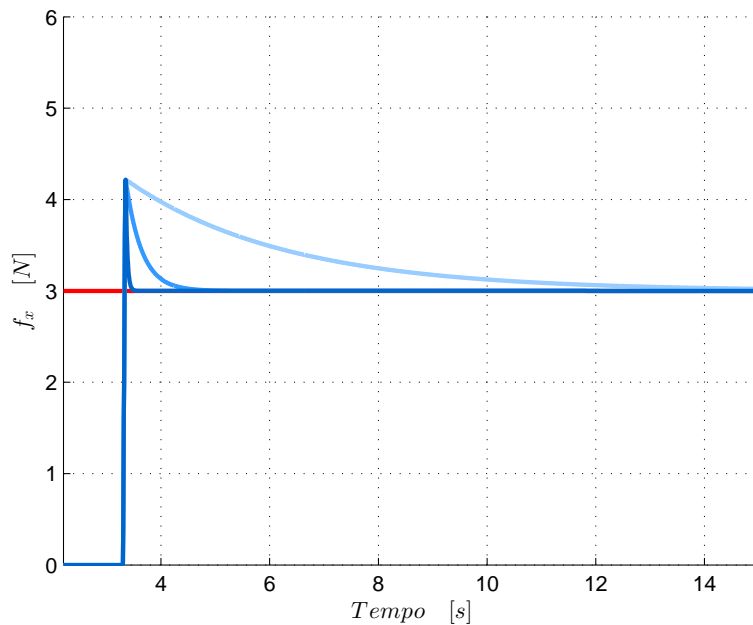
Il comportamento del vincolo di forza è stato poi confrontato utilizzando diversi valori di rigidità, stimati sempre correttamente, cioè tali per cui la rigidità utilizzata per la misura di forza e quella utilizzata per la definizione del corrispondente vincolo siano identiche (Fig. 5.11 e Fig. 5.12).

L'aumento di rigidità ovviamente amplifica le oscillazioni di forza quando è presente incertezza: dato che la funzione di sensitività è moltiplicata per  $K_{amb}$ , un piccolo errore in posizione porta ad un errore maggiore in termini di forza più la rigidità dell'ambiente aumenta. Si nota, inoltre, come il picco in fase di instaurazione del contatto sia ben più elevato, sempre per lo stesso motivo. Per attenuare questi fenomeni è quindi richiesto l'utilizzo di  $\delta_1$  inferiori e di diminuire la velocità di avvicinamento in prossimità della superficie.

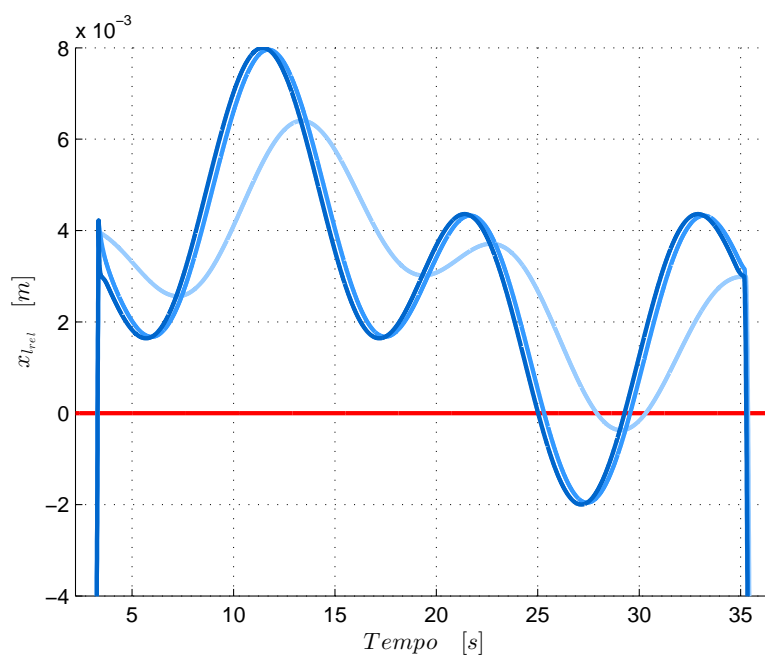
Una delle criticità nella definizione del vincolo di forza è la necessità di dover stimare la rigidità del contatto, che dipende non soltanto dall'ambiente, ma



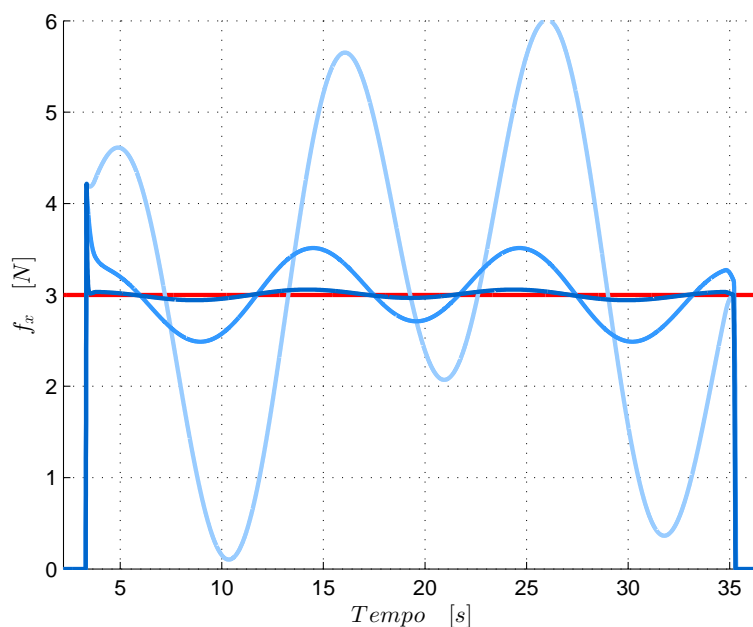
**Figura 5.7:** *SOTH2: Posizione relativa del tool rispetto alla superficie in condizioni ideali.* Rosso: riferimento di posizione - Azzurro: andamento della posizione con  $\delta_1 = 3s$  - Blu: andamento della posizione con  $\delta_1 = 0.3s$  - Blu scuro: andamento della posizione con  $\delta_1 = 0.03s$ .



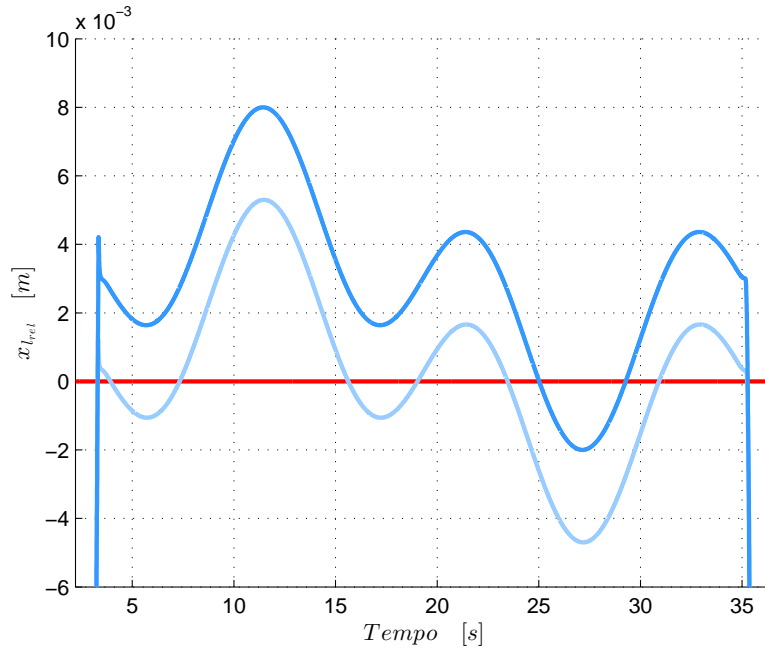
**Figura 5.8:** *SOTH2: Forza di contatto in condizioni ideali.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $\delta_1 = 3s$  - Blu: andamento della forza con  $\delta_1 = 0.3s$ , - Blu scuro: andamento della forza con  $\delta_1 = 0.03s$ .



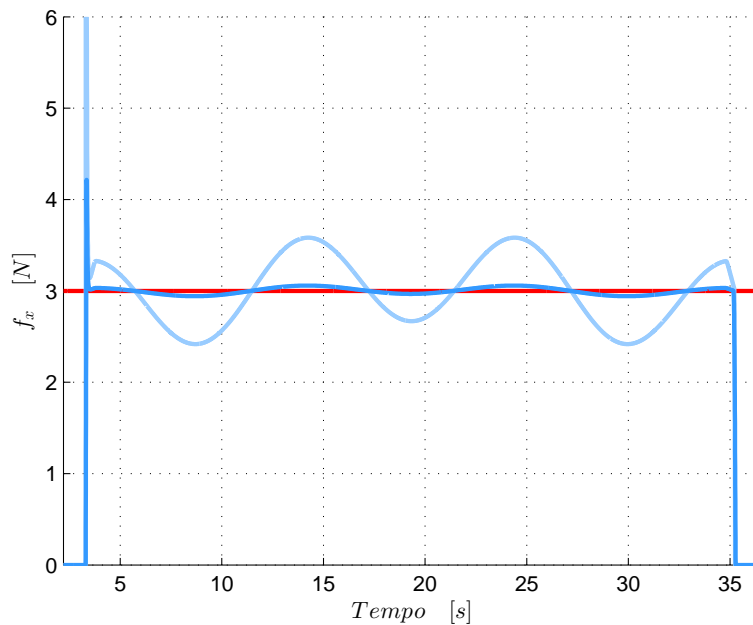
**Figura 5.9:** *SOTH2: Posizione relativa del tool rispetto alla superficie con incertezza geometrica.* Rosso: riferimento di posizione - Azzurro: andamento della posizione con  $\delta_1 = 3s$  - Blu: andamento della posizione con  $\delta_1 = 0.3s$  - Blu scuro: andamento della posizione con  $\delta_1 = 0.03s$ .



**Figura 5.10:** *SOTH2: Forza di contatto con incertezza geometrica.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $\delta_1 = 3s$  - Blu: andamento della forza con  $\delta_1 = 0.3s$  - Blu scuro: andamento della forza con  $\delta_1 = 0.03s$ .



**Figura 5.11:** *SOTH2: Posizione relativa del tool rispetto alla superficie al variare di  $K = K_{amb}$ , con incertezza geometrica.* Rosso: riferimento di posizione - Azzurro: andamento della posizione con  $K = K_{amb} = 10000 \frac{N}{m}$  - Blu: andamento della posizione con  $K = K_{amb} = 1000 \frac{N}{m}$ .  $\delta_1 = 0.03s$ .



**Figura 5.12:** *SOTH2: Forza di contatto al variare di  $K = K_{amb}$ , con incertezza geometrica.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $K = K_{amb} = 10000 \frac{N}{m}$  - Blu: andamento della forza con  $K = K_{amb} = 1000 \frac{N}{m}$ .  $\delta_1 = 0.03s$ .

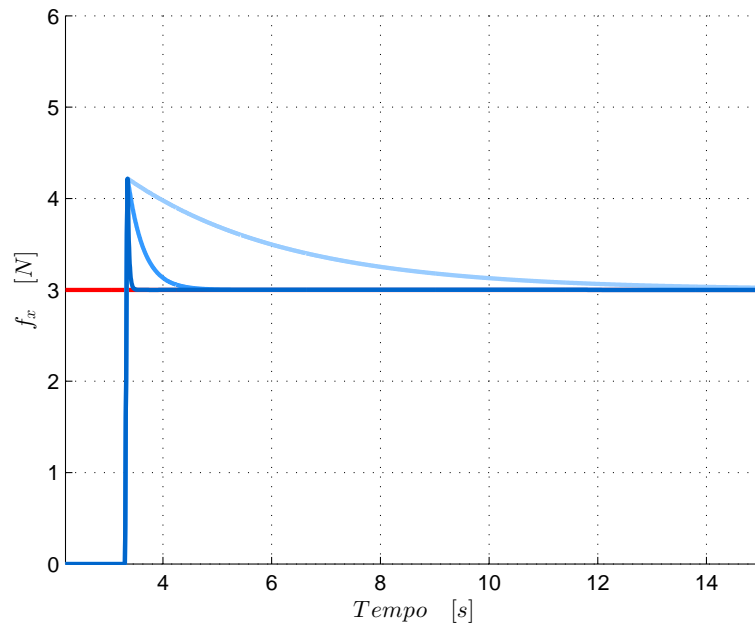


anche dalla rigidità dei giunti dello stesso robot. Una stima perfetta della rigidità risulta molto difficile, mentre è solitamente più facile determinarne almeno l'ordine di grandezza. È perciò interessante valutare anche in simulazione la robustezza dell'algoritmo a fronte di possibili sovra-stime o sotto-stime di questo parametro.

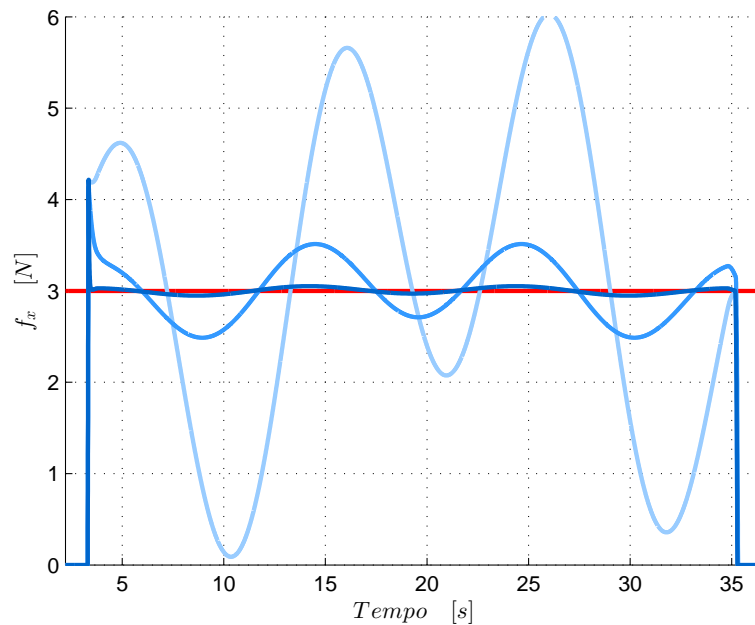
Nelle figure delle prove 5.13 e 5.14, sono rappresentati gli andamenti della forza a fronte di variazioni nella stima del valore di rigidità reale utilizzato nella simulazione della misura ( $K_{amb} = 1000 \frac{N}{m}$ ), sia in assenza che in presenza di incertezza geometrica. Come si può vedere sia dalle figure che dall'equazione (4.16) e dall'analisi effettuata per la figura 4.5, una variazione nella stima di  $K$  ha lo stesso effetto di una variazione della costante di tempo  $\delta_1$ . Una sottostima di  $K$  porterà in genere a prestazioni migliori: questo è spiegabile col fatto che per compensare un errore nell'inseguimento del riferimento si cercherà di compiere un movimento più ampio nello spazio cartesiano, e quindi, se la rigidità reale è maggiore, ciò risulterà in un comportamento più reattivo della forza. Al contrario, una sovrastima di  $K$  genererà un comportamento più conservativo, visto che una piccola variazione di posizione genererebbe una grande variazione di forza.

Tuttavia una sotto-stima troppo accentuata della rigidità accoppiata ad una scelta della costante di tempo in grado di fornire elevate prestazioni, può dar luogo ad effetti indesiderati. Infatti si potrebbe incorrere in ampie oscillazioni ad alta frequenza durante la fase di transitorio del contatto tra tool e superficie o in caso di elevata rugosità del materiale (Fig. 5.16); in casi estremi si potrebbe verificare un continuo alternarsi di contatto e distacco che porterebbe a instabilità. Tutto ciò avviene indipendentemente dalla planarità della superficie o dalla presenza di altri disturbi di tipo geometrico. Le oscillazioni a più bassa frequenza (Fig. 5.15) sono invece dovute alla presenza di incertezze geometriche non modellizzate, come nei casi precedenti.

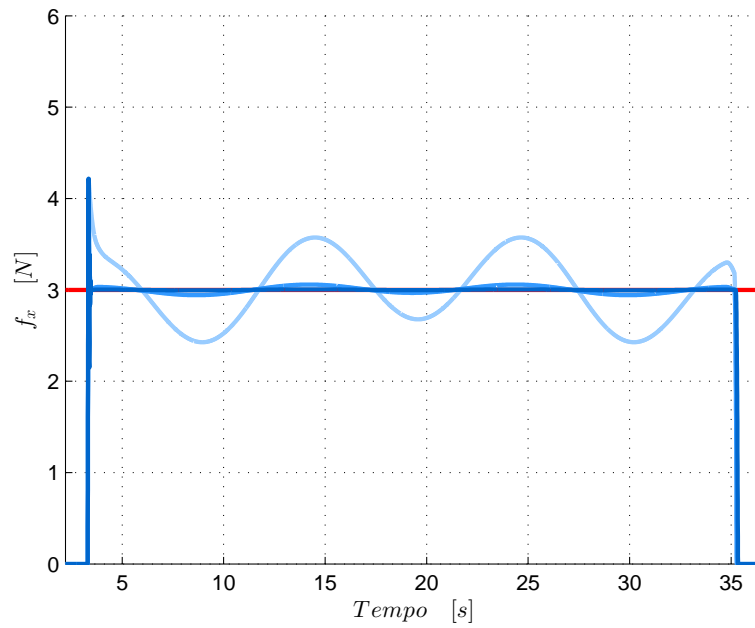
Un'ultima prova è stata effettuata cercando di evidenziare la relazione tra il valore di velocità massima dell'ascissa curvilinea e le prestazioni in caso di incertezza geometrica.



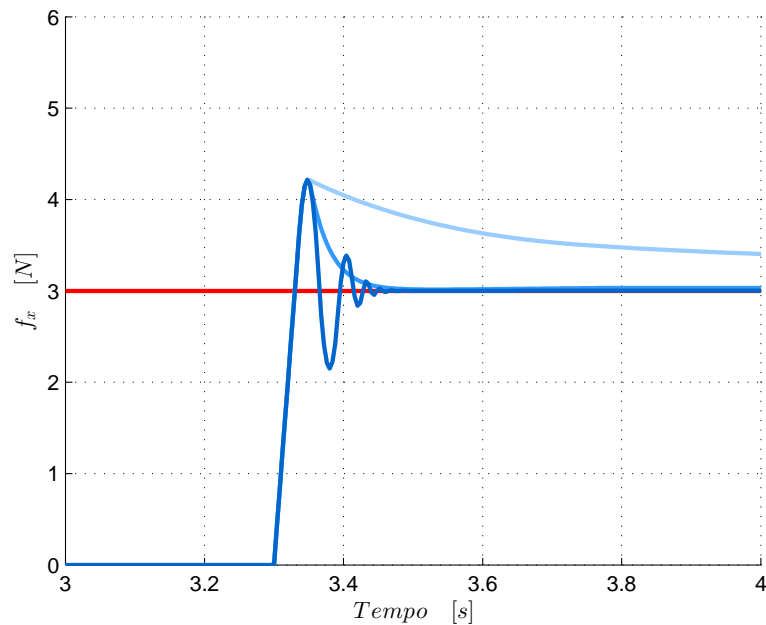
**Figura 5.13:** *SOTH2: Forza di contatto al variare della stima di  $K$ , in condizioni ideali.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $K = 10K_{amb}$  - Blu: andamento della forza con  $K = K_{amb}$  - Blu scuro: andamento della forza con  $K = 0.1K_{amb}$ .  $K_{amb} = 1000$ ,  $\delta_1 = 0.3s$ .



**Figura 5.14:** *SOTH2: Forza di contatto al variare della stima di  $K$ , con incertezza geometrica.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $K = 10K_{amb}$  - Blu: andamento della forza con  $K = K_{amb}$  - Blu scuro: andamento della forza con  $K = 0.1K_{amb}$ .  $K_{amb} = 1000$ ,  $\delta_1 = 0.3s$ .



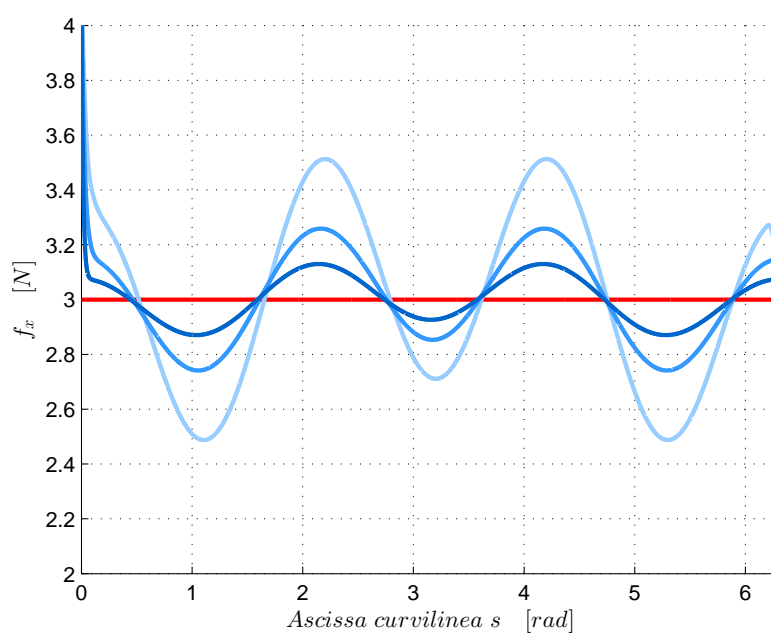
**Figura 5.15:** *SOTH2: Forza di contatto al variare della stima di  $K$ , con incertezza geometrica.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $K = 10K_{amb}$  - Blu: andamento della forza con  $K = K_{amb}$  - Blu scuro: andamento della forza con  $K = 0.1K_{amb}$ .  $K_{amb} = 1000$ ,  $\delta_1 = 0.03s$ .



**Figura 5.16:** *Zoom di Fig. 5.15.* Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $K = 10K_{amb}$  - Blu: andamento della forza con  $K = K_{amb}$  - Blu scuro: andamento della forza con  $K = 0.1K_{amb}$ .  $K_{amb} = 1000$ ,  $\delta_1 = 0.03s$ .

Nel grafico 5.17 sono illustrate le varie prove: una diminuzione di  $\dot{s}_{max}$  comporta un aumento del tempo necessario al completamento del task, pertanto gli andamenti della forza sono descritti in base al valore dell'ascissa curvilinea, anziché del tempo. Le prestazioni sono sensibilmente peggiori per elevati valori della velocità di percorrenza, questo perché un aumento di velocità equivale, in caso di irregolarità geometriche non modellizzate, a un aumento della frequenza di variazione della superficie, più difficilmente compensabile dal vincolo sintetizzato (si veda nuovamente la Fig. 4.6). Anche la velocità dell'ascissa curvilinea diventa quindi un importante parametro di progetto per questo tipo di applicazione.

Complessivamente la scelta dei parametri dipende dal singolo caso in esame, dal tipo di contatto, dal tipo di superficie e dalla sua incertezza, nonché dal robot stesso. Tuttavia l'analisi effettuata nel capitolo 4.1.3 fornisce importanti informazioni progettuali: si può dire che l'utilizzo di costanti di tempo più piccole, una leggera sottostima della rigidità, e velocità di percorso non elevate portano a migliori prestazioni nel tracking; ciononostante esistono limiti inferiori che non permettono di ottenere prestazioni arbitrarie, le quali dipendono soprattutto dalle performance del robot.



**Figura 5.17:** SOTH2: Forza di contatto al variare di  $\dot{s}_{max}$ , con incertezza geometrica. Rosso: riferimento della forza di contatto - Azzurro: andamento della forza con  $\dot{s}_{max} = 0.2 \frac{rad}{s}$  - Blu: andamento della forza con  $\dot{s}_{max} = 0.1 \frac{rad}{s}$ , - Blu scuro: andamento della forza con  $\dot{s}_{max} = 0.05 \frac{rad}{s}$ .  $\delta_1 = 0.3s$ .



## Capitolo 6

# Stima delle forze esterne

Nei capitoli precedenti è stato sintetizzato il vincolo necessario all'implementazione del controllo di forza e successivamente ne è stata verificata l'efficacia in simulazione.

Per la stessa natura della struttura dell'algoritmo considerato, è imprescindibile la presenza di un qualche tipo di misura della forza che si vuole controllare (si veda la (4.16)). In questo senso, possono insorgere problemi quando non è possibile l'installazione di opportuni sensori. Per esempio è questo il caso di manipolatori di piccola taglia, per cui i carichi utili sono limitati, ma che ciononostante necessitano di lavorare a contatto con l'ambiente o con operatori umani. In particolare può non essere possibile nemmeno montare il sensore di forza su un supporto rigido, come nel caso di applicazioni dual-arm.

In questi casi si deve fare ricorso a opportuni stimatori, che permettano una ricostruzione sufficientemente accurata del segnale di forza anche in assenza di sensori. Chiaramente, questo tipo di approccio introduce problematiche ulteriori derivanti dalla modellizzazione del robot e dal condizionamento dei segnali misurati e ricostruiti.

### 6.1 Osservatore delle coppie esterne ai giunti

La definizione di osservatori in grado di fornire una stima delle coppie esterne ai giunti è già nota in letteratura, e principalmente utilizzata per applicazioni di

sicurezza e individuazione di guasti.

L'osservatore, basato sul calcolo dei residui di coppia, ideato da De Luca e Mattone [23], è stato utilizzato con successo per identificare il guasto di uno o più giunti e per rilevare la collisione dei link del robot con l'ambiente circostante, tuttavia l'applicazione a schemi di controllo ibrido è stata affrontata solo in ambiente di simulazione [24].

Applicazioni sperimentali sono state condotte da Zanchettin e Rocco [4] nell'ambito di interazione uomo-robot, in particolare per la definizione di filtri di ammettenza. In questo caso, a differenza dei precedenti, un'accurata stima delle forze e quindi di identificazione del modello dinamico del robot è consigliabile.

La ricostruzione delle coppie esterne ai giunti è effettuata di seguito, facendo ricorso al modello già proposto in [23].

Si consideri il ben noto modello dinamico di un manipolatore generico:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{fr}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \quad (6.1)$$

e i momenti generalizzati:

$$\mathbf{p} = \mathbf{B}(\mathbf{q})\dot{\mathbf{q}} \quad (6.2)$$

Derivando  $\mathbf{p}$  rispetto al tempo, e sostituendo  $\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}}$  con il modello (6.1), si ottiene:

$$\dot{\mathbf{p}} = \dot{\mathbf{B}}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \mathbf{fr}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \quad (6.3)$$

Si può inoltre dimostrare che:

$$\dot{\mathbf{B}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})^T \quad (6.4)$$

Infatti, definiti i simboli di Christoffel del primo tipo  $c_{ijk} = \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right)$  e sfruttando la simmetria della matrice di inerzia, si ha:

$$\begin{aligned} \dot{b}_{ij} - c_{ij} &= \sum_k \frac{\partial b_{ij}}{\partial q_k} \dot{q}_k - \sum_k c_{ijk} \dot{q}_k = \\ &= \sum_k \left[ \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \right] \dot{q}_k = \\ &= \sum_k \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{jk}}{\partial q_i} - \frac{\partial b_{ik}}{\partial q_j} \right) \dot{q}_k = \sum_k c_{jik} \dot{q}_k = c_{ji} \end{aligned}$$



L'espressione finale di  $\dot{\mathbf{p}}$  risulta essere:

$$\dot{\mathbf{p}} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})^T \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \mathbf{fr}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} = \boldsymbol{\alpha}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_{ext} \quad (6.5)$$

Noto  $\boldsymbol{\alpha}(\mathbf{q}, \dot{\mathbf{q}})$  è quindi possibile definire il residuo, a cui corrisponde la seguente dinamica:

$$\mathbf{r} = \mathbf{K}_{oss} \left[ \mathbf{p} - \int_0^{t_k} (\boldsymbol{\alpha}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{r}) dt \right] \quad (6.6)$$

$$\dot{\mathbf{r}} = \mathbf{K}_{oss} (\boldsymbol{\tau}_{ext} - \mathbf{r}) \quad (6.7)$$

Il vettore residuo  $\mathbf{r}$  converge asintoticamente al vettore delle coppie esterne ai giunti  $\boldsymbol{\tau}_{ext}$  se la matrice diagonale  $\mathbf{K}_{oss}$  è definita positiva, infatti per ogni componente di  $\mathbf{r}$  si ottiene la funzione di trasferimento:

$$\frac{r_i(s)}{\tau_{i,ext}(s)} = \frac{K_{oss,i}}{s + K_{oss,i}} \quad (6.8)$$

dove  $K_{oss,i}$  può essere scelto per determinare la velocità di convergenza. Il vantaggio di questa formulazione è che non necessita delle accelerazioni ai giunti, ma solo di velocità e posizioni, ovviamente a patto di essere a conoscenza del modello dinamico del robot.

Dalla letteratura è nota la relazione esistente tra le coppie esterne ai giunti ( $\boldsymbol{\tau}_{ext}$ ) e il vettore di forze e momenti agenti all'end-effector ( $\mathbf{f}$ ), espresso nella terna base del robot, attraverso lo jacobiano geometrico dell'end effector, espresso anch'esso in terna base:

$$\mathbf{J}(\mathbf{q})^T \mathbf{f} = \boldsymbol{\tau}_{ext} \simeq \mathbf{r} \quad (6.9)$$

Per ricavare forze e momenti rappresentate dal vettore  $\mathbf{f}$  è necessario risolvere questo sistema di equazioni, la cui soluzione è data dall'inversa del trasposto dello jacobiano solo se il sistema è quadrato, cioè solo se il robot non è ridondante. Nel caso di manipolatore ridondante il sistema è sovra-vincolato, pertanto la soluzione si ottiene dalla pseudo-inversa di  $\mathbf{J}(\mathbf{q})^T$ , ovvero la soluzione ai minimi quadrati:

$$\mathbf{f} = \mathbf{J}(\mathbf{q})^{T\dagger} \boldsymbol{\tau}_{ext} \quad (6.10)$$

Teoricamente, con un manipolatore con  $n$  gradi di libertà, e  $m$  componenti del vettore  $\mathbf{f}$  che si intende determinare, sarebbe possibile effettuare la stima pur

non avendo a disposizione al massimo  $n - m$  degli  $n$  residui delle coppie esterne ai giunti, se si assume che la forza possa agire soltanto sull'end effector; il degrado della stima risulterebbe tuttavia maggiore, più sono eccitati i giunti trascurati.

## 6.2 Modellizzazione e identificazione dell'attrito

Lo stimatore presentato necessita della conoscenza completa del modello dinamico del manipolatore. Mentre i parametri relativi alle matrici di inerzia, Coriolis e gravitazionale possono essere calcolati facilmente e con buona accuratezza noti i dati forniti dal produttore del robot, o con tecniche consolidate di identificazione, risulta invece di particolare interesse la modellizzazione delle componenti di attrito.

Le prestazioni di diversi modelli, statici e dinamici, sono state studiate in letteratura [25]: le questioni principali di modellizzazione si riscontrano nella zona di funzionamento a bassa velocità, dove insorgono problemi di stick-slip. Modelli dinamici come quelli di LuGre e GMS (Generalized Maxwell-Slip), offrono ovviamente stime migliori nelle zone critiche, a patto di accettare una maggiore complessità, mentre formulazioni classiche come quella coulombiana mostrano i propri limiti in queste condizioni di funzionamento.

Nel caso in esame, la stima dell'attrito ai giunti sarebbe una questione critica, qualora si volessero stimare le coppie esterne in tutte le regioni di velocità; visto il tipo di applicazione, si assumerà nel seguito che il manipolatore lavori per lo più a velocità sufficientemente elevate da poter trascurare il fenomeno di stick-slip, in questo caso un aumento di complessità del processo di identificazione non porta a miglioramenti sostanziali.

Con queste premesse è stato utilizzato un modello misto, dato da una parte di attrito costante e una viscosa, proporzionale alla velocità di rotazione del giunto; sono stati inoltre determinati valori di attrito indipendenti per velocità di rotazione positive e negative. Per quanto riguarda la stima a basse velocità, tra  $-0.01$  e  $0.01 \frac{rad}{s}$  è stata utilizzata un'approssimazione lineare per rendere continuo il modello (Fig. 6.1). Per effettuare l'identificazione sono state eseguite

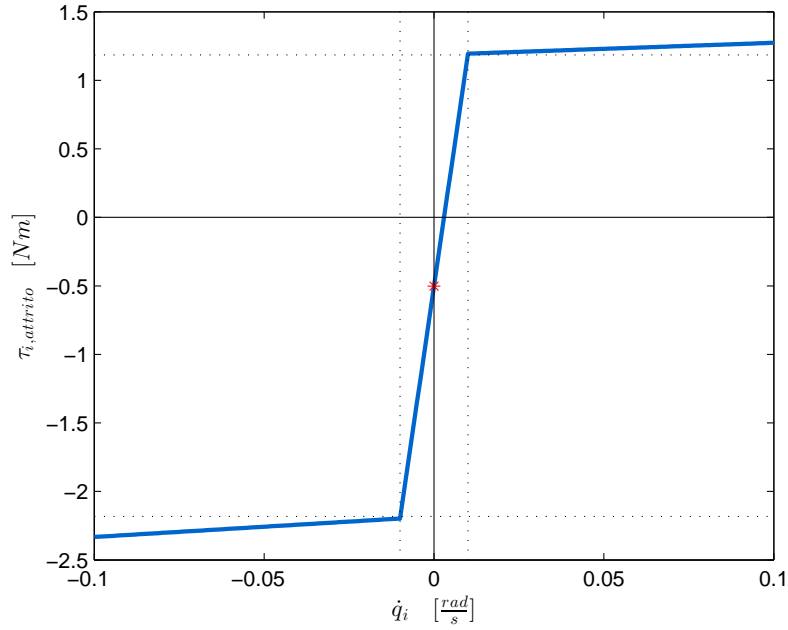


Figura 6.1: Modello della forza di attrito.

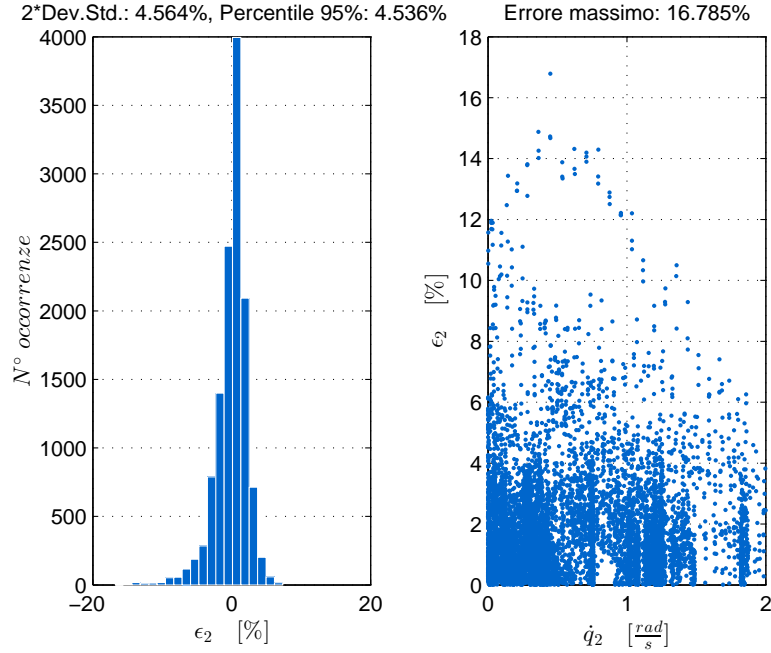
delle prove sul robot senza alcun carico, distribuite nell'intero spazio operativo, sono state quindi registrate accelerazioni, velocità, posizioni e coppie ai giunti.

Riprendendo il modello dinamico classico del manipolatore e ipotizzando che le matrici  $\mathbf{B}(\mathbf{q})$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ ,  $\mathbf{g}(\mathbf{q})$  del modello corrispondano a quelle reali, si ottiene il modello dell'errore che contiene i parametri da stimare:

$$\begin{aligned}
 \boldsymbol{\tau}_{mis} - \boldsymbol{\tau}_{model} = & (\mathbf{d}_p \dot{\mathbf{q}} + \mathbf{t}_{cp})(\dot{\mathbf{q}} > 0.01) + (\mathbf{d}_n \dot{\mathbf{q}} - \mathbf{t}_{cn})(\dot{\mathbf{q}} < -0.01) + \\
 & + \left[ \left( \frac{\dot{\mathbf{q}}}{2} + \frac{\mathbf{1}}{200} \right) \mathbf{d}_p + \left( \frac{\dot{\mathbf{q}}}{0.02} + \frac{\mathbf{1}}{2} \right) \mathbf{t}_{cp} + \left( \frac{\dot{\mathbf{q}}}{2} - \frac{\mathbf{1}}{200} \right) \mathbf{d}_n + \right. \\
 & \left. + \left( \frac{\dot{\mathbf{q}}}{0.02} - \frac{\mathbf{1}}{2} \right) \mathbf{t}_{cn} \right] (-0.01 \leq \dot{\mathbf{q}} \leq 0.01) \quad (6.11)
 \end{aligned}$$

dove  $\boldsymbol{\tau}_{model}$  sono le coppie ai giunti calcolate applicando al modello (6.1), posizioni, velocità e accelerazioni misurate.

Il problema di identificazione è quindi ricondotto alla risoluzione di un sistema



**Figura 6.2:** Errore di identificazione per il giunto 2. Sinistra: Istogramma con il numero di occorrenze per errore percentuale - Destra: Errore percentuale rispetto alla velocità di giunto.

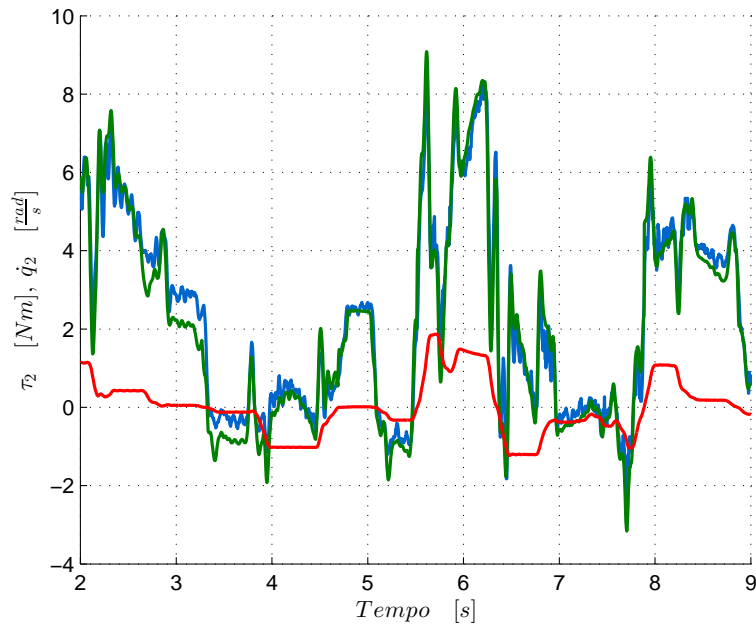
di equazioni sovra-vincolato, e la soluzione ottenuta tramite minimi quadrati:

$$\begin{bmatrix} \dot{\mathbf{q}}_{mis,p} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dot{\mathbf{q}}_{mis,n} & -\mathbf{1} \\ \frac{\dot{\mathbf{q}}_{mis,z}}{2} + \frac{\mathbf{1}}{200} & \frac{\dot{\mathbf{q}}_{mis,z}}{0.02} + \frac{\mathbf{1}}{2} & \frac{\dot{\mathbf{q}}_{mis,z}}{2} - \frac{\mathbf{1}}{200} & \frac{\dot{\mathbf{q}}_{mis,z}}{0.02} - \frac{\mathbf{1}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{d}_p \\ \mathbf{t}_{cp} \\ \mathbf{d}_n \\ \mathbf{t}_{cn} \end{bmatrix} = \begin{bmatrix} \tau_{mis,p} - \tau_{model,p} \\ \tau_{mis,n} - \tau_{model,n} \\ \tau_{mis,z} - \tau_{model,z} \end{bmatrix} \quad (6.12)$$

Le coppie ai giunti dovute all'attrito sono state poi calcolate con i parametri ricavati e confrontate con  $\tau_{mis} - \tau_{model}$  relativamente alla coppia massima erogabile da ogni giunto, ottenendo una stima dell'errore di identificazione attraverso la deviazione standard campionaria (Fig. 6.2):

$$\epsilon_{i,rel} = \frac{\tau_{i,mis} - \tau_{i,model} - fr_i(\mathbf{q}, \dot{\mathbf{q}})}{\tau_{i,max}} \quad (6.13)$$

In Fig. 6.3 è mostrato il confronto fra coppie misurate e coppie ricostruite con l'uso del modello per uno dei giunti, utilizzando gli stessi valori di posizione,



**Figura 6.3:** *Comparazione tra coppie misurate e ricostruite per il giunto 2. Blu:  $\tau_2$  misurata - Verde:  $\tau_2$  ricostruita col modello identificato - Rosso: Velocità di giunto.*

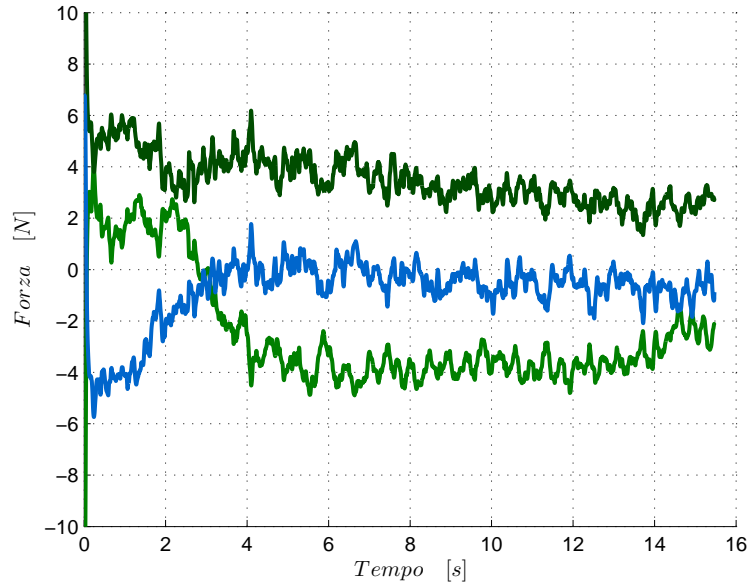
velocità e accelerazione. Si evidenzia come in genere la ricostruzione sia peggiore in prossimità della velocità di giunto nulla (3 – 4s).

Effettuando una prova preliminare sul manipolatore, le carenze del modello utilizzato sono evidenti: soprattutto quando uno o più giunti si muovono a velocità basse, il residuo rileva coppie esterne anche in assenza di contatto o interazione con l'ambiente (Fig. 6.4).

### 6.3 Condizionamento dei segnali di coppia e forza

L'errore commesso in fase di identificazione comporta la rilevazione di falsi positivi quando il manipolatore è in movimento, di conseguenza vengono calcolate forze a livello del tool in realtà assenti. Ciò può generare comportamenti indesiderati nel momento in cui sia necessario rilevare se il contatto è avvenuto o meno.

La varianza dell'errore percentuale (6.13) fornisce una stima della variabilità



**Figura 6.4:** Forze misurate in assenza di contatto per un movimento rettilineo verticale a  $0.01 \frac{m}{s}$ . In verde chiaro, verde scuro e blu rispettivamente, le componenti di forza lungo gli assi  $x$ ,  $y$ ,  $z$  della terna base del manipolatore.

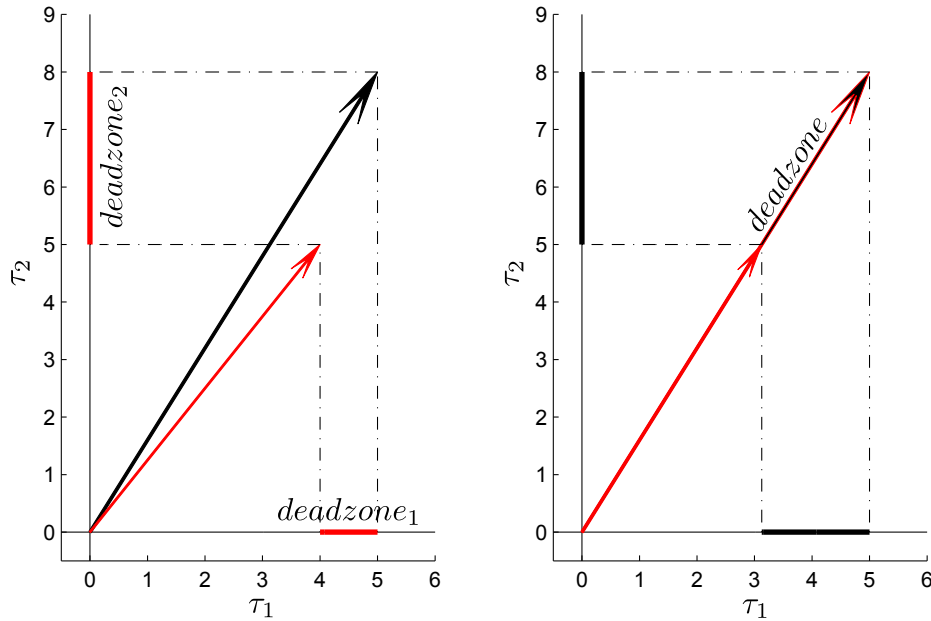
della misura dei residui, relativamente alle coppie massime. Con lo scopo di rimuovere il fenomeno descritto, questo valore è stato utilizzato per filtrare le componenti di coppia, in pratica si applica una deadzone (zona morta) ad ogni componente, in modo da aumentare la soglia oltre la quale viene rilevata una coppia. Così facendo si introduce, tuttavia, un errore sistematico costante di misura, pertanto l'errore relativo sarà meno significativo più elevati saranno i valori stimati di  $\tau_{ext}$ , ovvero più elevata sarà la forza di contatto.

Nell'applicazione della deadzone sono stati utilizzati due differenti approcci:

- Una deadzone è applicata ad ogni singola componente di  $\mathbf{r}$  in base alla corrispondente varianza dell'errore.
- Un'unica deadzone è applicata direttamente al modulo del vettore  $\mathbf{r}$

Con il primo approccio, ad ogni  $r_i$  è applicata la deadzone:

$$deadzone(r_i, 2\sigma_{\epsilon_{i,rel}} \cdot \tau_{i,max}) \quad (6.14)$$



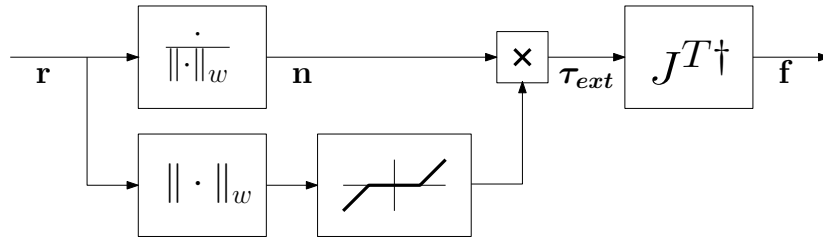
**Figura 6.5:** Rappresentazione vettoriale delle deadzone utilizzate. Sinistra: Deadzone non proporzionali alle coppie - Destra: Deadzone proporzionali.

Così facendo ogni componente è filtrata quanto basta per evitare di includere contributi inesistenti. Ciononostante poiché le deadzone non sono proporzionali in ugual modo a ogni  $r_i$ , il vettore risultante sarà ruotato rispetto a  $\mathbf{r}$  (Fig. 6.5), di conseguenza anche il vettore calcolato delle forze di contatto avrà un orientamento errato, presentando componenti di forza in realtà assenti e rendendo necessaria un'ulteriore manipolazione del segnale.

Per aggirare questo problema è possibile preservare la direzione di  $\mathbf{r}$ , applicando la deadzone direttamente al modulo del vettore stesso. Per fare ciò si introduce la norma pesata di  $\mathbf{r}$  sulle coppie massime ai giunti, cosicché possa assumere al massimo valore unitario:

$$\|\mathbf{r}\|_w = \sqrt{\sum_1^n \left(\frac{r_i}{\tau_{i,max}}\right)^2 / n} \quad (6.15)$$

Utilizzare questa norma permette di pesare le  $r_i$  in base alle coppie massime, evitando di penalizzare con la deadzone quelle componenti per cui  $\tau_{i,max}$  è piccolo, quando il vettore  $\mathbf{r}$  è principalmente diretto lungo di esse.



**Figura 6.6:** Schema a blocchi del processo di condizionamento dei residui. Secondo approccio.

Il valore della deadzone è ricavato allo stesso modo da quelli ottenuti per le singole componenti. In generale la deadzone complessiva è più ampia di quella che si calcolerebbe col primo approccio, ottenendo un comportamento più conservativo, ma mantenendo la direzione del vettore:

$$deadzone_w = \sqrt{\sum_1^n (2\sigma_{\epsilon_{i,rel}})^2 / n} \quad (6.16)$$

Applicando la deadzone a  $\|\mathbf{r}\|_w$  e poi moltiplicando per la direzione ottenuta da  $\frac{\mathbf{r}}{\|\mathbf{r}\|_w}$ , si ottiene il vettore finale delle coppie esterne da cui ricavare la stima delle forze di contatto (Fig. 6.6).

Una volta disponibile il vettore  $\mathbf{f}$  nella terna base è poi possibile proiettarlo nella terna di riferimento richiesta senza ulteriori manipolazioni.



# Capitolo 7

## Esperimenti

Con lo scopo di accertare i risultati ottenuti in simulazione, l'algoritmo con i vincoli sintetizzati è stato verificato sperimentalmente.

Per questi esperimenti è stato utilizzato il robot ABB FRIDA, il cui modello cinematico è già stato utilizzato in fase di simulazione.

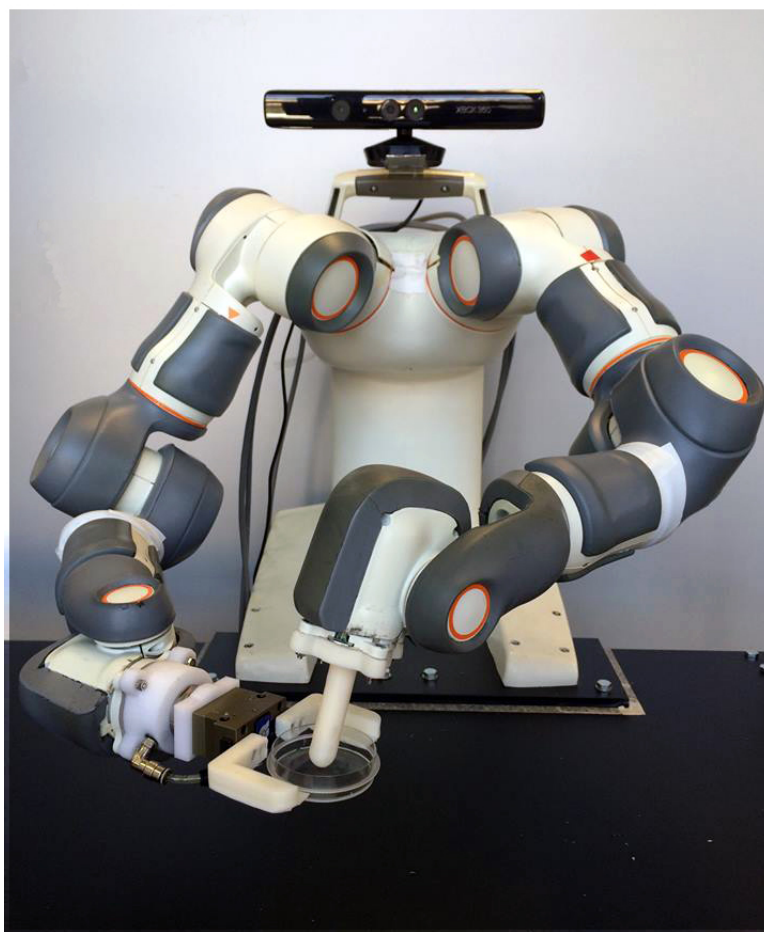
Nel resto del capitolo verrà presentato il robot utilizzato e gli esperimenti eseguiti, dopodiché verranno affrontate alcune questioni relative alla scelta dei parametri, già accennate in precedenza. Successivamente sarà trattato un problema nato in fase sperimentale relativo alla corretta stima della misura della forza di contatto.

Infine saranno presentati i risultati ottenuti in un esperimento dual-arm finale, comprendente tutte le modifiche e gli accorgimenti effettuati.

### 7.1 Descrizione dell'apparato sperimentale

Nell'esperimento si è fatto uso del robot ABB FRIDA (Fig. 7.1).

Il robot in questione è un prototipo di ABB concepito all'interno del percorso di ricerca e sviluppo che ha portato alla realizzazione del robot YuMi [26]. Si tratta di un robot a due bracci (dual-arm) leggero, con bassi payload, e pensato soprattutto per essere utilizzato in cooperazione con l'uomo. Per questi motivi è un manipolatore che presta particolare attenzione alla sicurezza, con l'obiettivo di eliminare barriere che ne renderebbero difficoltosa l'interazione.

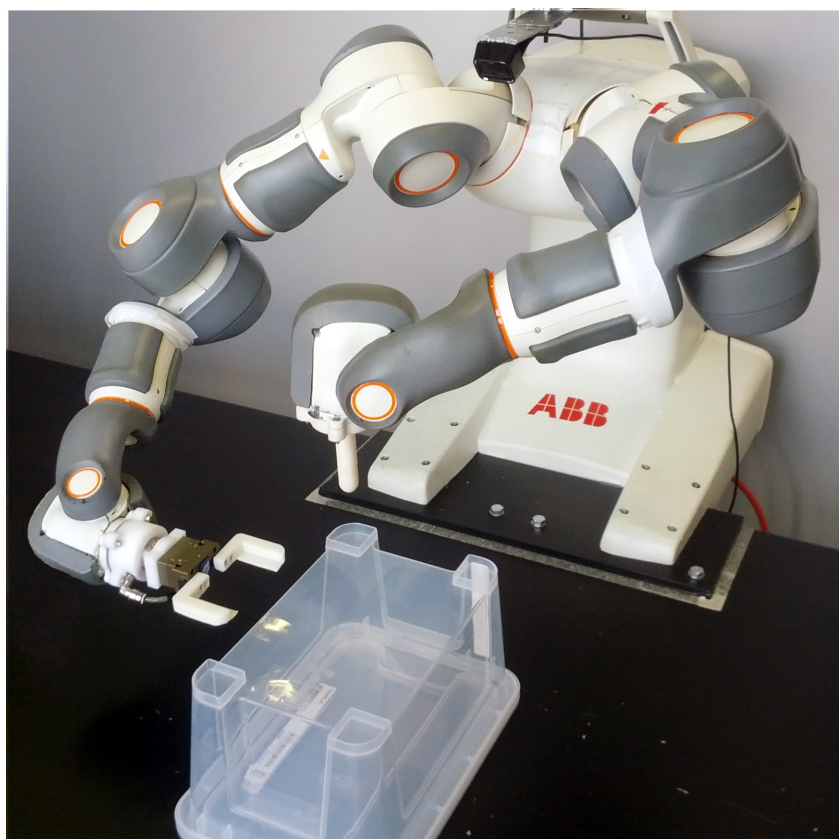


**Figura 7.1:** *Robot dual-arm ABB FRIDA.*

La configurazione del robot prevede due manipolatori montati su un corpo centrale, ogni braccio possiede 7 gradi di libertà, rendendolo intrinsecamente ridondante, per un totale di 14 gradi di libertà per l'intero sistema. Questa elevata destrezza lo rende particolarmente efficace per applicazioni di assemblaggio e maneggiamento di piccole parti, con l'ulteriore vantaggio di necessitare di spazi ridotti per il suo utilizzo.

Poiché la stessa natura del robot prevede l'interazione con l'ambiente esterno, può risultare utile il controllo della forza di contatto, tuttavia, come accennato nel capitolo 6, su robot con limitati payload risulta difficoltosa l'installazione di opportuni sensori, per cui si rende necessaria la stima in altro modo.

Come nelle simulazioni, il sistema è stato dapprima semplificato in modo da



**Figura 7.2:** *Apparato e configurazione utilizzati negli esperimenti.*

essere riportato ad un'applicazione single-arm. Il braccio sinistro del manipolatore esegue il task descritto da SOTH2 (5.26), tuttavia la superficie di lavoro non è più sorretta dal braccio destro, ma è fissa e appoggiata sul piano di lavoro che sostiene lo stesso robot. Questo accorgimento permette di eliminare in questi esperimenti le componenti non modellizzate in fase di simulazione, come l'elasticità del braccio destro, e di concentrarsi sull'effettiva verifica del funzionamento del vincolo di forza e dell'algoritmo complessivo.

Ancora una volta, il braccio sinistro ha il compito di eseguire un percorso circolare su una superficie individuata da un box di materiale plastico, caratterizzato da una bassa rigidezza, e da un box di nylon con rigidezza più elevata, mantenendo il riferimento impostato sulla forza di contatto normale alla superficie (ancora diretta lungo l'asse  $x$  della terna relativa al box, Fig. 7.2).

In termini realizzativi rimane valido il codice implementato in Matlab per la

simulazione, apportando opportune correzioni ai parametri dei vincoli e utilizzando i dati tecnici riportati dal produttore per quanto riguarda velocità, accelerazioni, coppie massime e limiti di giunto. Queste informazioni non vengono riportate per questioni di riservatezza, tuttavia è importante sapere che i giunti di spalla e gomito (cioè i primi quattro) sono in grado di erogare coppie maggiori rispetto a quelli di polso (gli ultimi tre). Tramite il software di controllo esterno fornito da ABB è possibile utilizzare direttamente il codice scritto anche per l'esperimento real-time.

Il movimento è nuovamente suddiviso nei seguenti passi:

1. Movimento nello spazio dei giunti da una posizione casuale a una predefinita per entrambe le braccia.
2. Avvicinamento del tool del braccio sinistro alla superficie fino a che non viene rilevato il contatto.
3. Esecuzione del task di path-following con riferimento di forza.
4. Allontanamento del tool dalla superficie e stop all'esecuzione.

## 7.2 Scelta dei parametri

Il primo problema da affrontare nell'implementazione sperimentale è la scelta dei parametri utilizzati nella definizione dei vincoli e per la stima della forza di contatto. La loro scelta influenza le prestazioni nella realizzazione del task programmato, e particolarmente critici sono i valori dei parametri per il controllo di forza, vista anche la variabilità introdotta dalla rigidità.

Nel seguito si descrive il ragionamento effettuato per la loro messa a punto.

### 7.2.1 Stima delle forze esterne

Nella (6.8) si è visto come la scelta della matrice dei guadagni  $\mathbf{K}_{oss}$  influenzi l'ampiezza della banda correttamente stimata delle coppie esterne ai giunti. Non esistono particolari limiti sulla scelta, se non quelli imposti dal tempo di

campionamento del robot, pertanto si può semplicemente scegliere una matrice diagonale di guadagni positivi uguali tra loro. Nello specifico si è utilizzato:

$$\mathbf{K}_{oss} = 50\mathbf{I}_7 \quad (7.1)$$

Per quanto riguarda le deadzone (6.13) e i parametri di attrito (6.12), sono stati ottenuti dal processo di identificazione i seguenti valori:

Giunto	$2\sigma_{\epsilon_{rel}}$ [%]	$d_p$ [ $\frac{Nmms}{rad}$ ]	$t_{cp}$ [ $Nm$ ]	$d_n$ [ $\frac{Nmms}{rad}$ ]	$t_{cn}$ [ $Nm$ ]
$q_1$	6.1866	0.8836	1.1831	1.4956	2.1853
$q_2$	4.5636	1.0068	0.2711	1.5461	1.9341
$q_3$	5.3296	0.5000	1.1824	0.6275	0.2882
$q_4$	3.7623	0.6551	0.3625	0.4895	0.9251
$q_5$	5.5676	0.0765	0.2828	0.0662	0.3848
$q_6$	5.4584	0.0528	0.2544	0.0656	0.2443
$q_7$	4.4474	0.0609	0.2756	0.0491	0.3649

**Tabella 7.1:** Parametri identificati del modello di attrito per il robot ABB FRIDA.

Dalla seconda colonna della tabella si ricava la deadzone complessiva (6.16) da applicare alla norma pesata (6.15) dei residui:

$$deadzone_w = \sqrt{\sum_1^7 (2\sigma_{\epsilon_{i,rel}})^2 / 7} = 5.6442\% \quad (7.2)$$

### 7.2.2 Vincoli di forza

L'equazione (4.16) prevede l'introduzione di due costanti di tempo e la conoscenza della rigidità di contatto per la particolare applicazione, tutte e tre sono parametri di progetto nella definizione del vincolo di forza e devono quindi essere accuratamente scelte in accordo con i limiti del sistema per garantire certi comportamenti.

Le costanti di tempo  $\delta_1$  e  $\delta_2$  indicano quanto velocemente il vincolo (4.16) converge a quello ideale:

$$\mathbf{f} = \mathbf{f}_{rif} \quad oppure \quad \mathbf{f}_{lb} \leq \mathbf{f} \leq \mathbf{f}_{ub} \quad (7.3)$$

Come si è visto dall'analisi di stabilità e confermato in simulazione, esse agiscono sulle performance durante l'esecuzione del task, in particolar modo quando sono presenti incertezze e disturbi. Tendenzialmente le si vorrebbe scegliere le più piccole possibili, al fine di evitare oscillazioni ed errori nell'inseguimento del riferimento, tuttavia un limite inferiore è rappresentato dalle specifiche tecniche del robot.

Si trascuri per il momento  $\delta_2$ , si ricordi la prima derivazione effettuata del vincolo (4.12) e se ne effettui la discretizzazione:

$$\mathbf{f}(k + \delta_1) \simeq \delta_1 \dot{\mathbf{f}}(k) + \mathbf{f}(k) = \mathbf{f}_{rif} \quad (7.4)$$

Con questa dicitura si impone che il valore di forza dopo  $\delta_1$  secondi sia uguale al riferimento, avendo ipotizzato che la derivata della forza si mantenga costante nel periodo e che la superficie di contatto sia piana o completamente nota.

Avendo modellizzato la forza come puramente elastica e quindi con derivata proporzionale alla velocità relativa tra superficie e tool, ovvero alle velocità di giunto (4.8), si nota come, qualora la forza sia molto vicina al riferimento o bound e la sua derivata elevata all'istante  $k$ , sia necessario avere a disposizione sufficiente tempo ( $\delta_1$ ) da poter azzerare la velocità relativa ed avere una forza coincidente con il riferimento e derivata nulla.

Il tempo minimo necessario ad ottenere ciò è dato dal valore massimo dei rapporti tra velocità e accelerazioni massime di giunto del robot, cioè dal tempo necessario a fermare un giunto in rotazione alla massima velocità:

$$\delta_1 = \max_i \left\{ \frac{\dot{q}_{i,max}}{\ddot{q}_{i,max}} \right\} \quad (7.5)$$

Per  $\delta_2$  è valido lo stesso ragionamento utilizzando invece del rapporto tra velocità e accelerazioni quello tra accelerazioni e jerk massimi:

$$\delta_2 = \max_i \left\{ \frac{\ddot{q}_{i,max}}{\overset{\cdot\cdot}{q}_{i,max}} \right\} \quad (7.6)$$

Poiché l'accelerazione è la variabile di ottimizzazione/controllo, si può supporre che i jerk massimi siano infiniti, per cui non si avrebbe un limite inferiore sul valore di  $\delta_2$ , che quindi può semplicemente essere scelto pari al tempo di ciclo. Il limite superiore è invece rappresentato indicativamente dal valore scelto per  $\delta_1$ .

Per quanto riguarda la rigidezza  $\mathbf{K}$ , idealmente si vorrebbe utilizzare una stima quanto più possibile prossima al valore reale, risultante dall'accoppiamento tra ambiente e rigidezza di giunti e link del robot, col fine di rendere la sintesi il più possibile indipendente da questo parametro. La sua determinazione può, però, risultare più o meno difficoltosa.

Come visto in simulazione una sottostima rende l'algoritmo più reattivo, ma la combinazione con valori particolarmente performanti di  $\delta_1$  può risultare deleteria, generando fenomeni di oscillazione ad alta frequenza ed eventualmente di repentini distacchi. Una sovrastima, invece, rende il vincolo più conservativo, ma meno performante e capace di attenuare i disturbi. Nella maggior parte dei casi va quindi valutato il trade-off prestazionale: per le prove sperimentali si è optato per una più o meno accentuata sottostima, in base alla superficie.

Fatte queste considerazioni, per gli esperimenti sono stati utilizzati i seguenti parametri:

$$K = 1000 \frac{N}{m} \quad (7.7)$$

$$\delta_1 = \max_i \left\{ \frac{\dot{q}_{i,max}}{\ddot{q}_{i,max}} \right\} = 0.583s \quad (7.8)$$

$$\delta_2 = T_s = 0.004s \quad (7.9)$$

dove  $T_s$  è il tempo di campionamento del controllo del robot.

### 7.2.3 Cinematica inversa

Il vincolo in esame è quello definito dall'equazione (4.52):

$$\ddot{\mathbf{e}}(\mathbf{q}) + \mathbf{K}_d \dot{\mathbf{e}}(\mathbf{q}) + \mathbf{K}_p \mathbf{e}(\mathbf{q}) = 0 \quad (7.10)$$

Affinché l'equazione abbia una dinamica asintoticamente stabile,  $\mathbf{K}_p$  e  $\mathbf{K}_d$  devono essere definite positive, nel caso più semplice possono essere prese diagonali.

Mentre quanto detto è valido a tempo continuo, nell'applicazione il vincolo è definito a tempo discreto, pertanto come suggeriscono P. Falco e C. Natale [27] esistono delle limitazioni sui valori assegnabili alle matrici. Questi limiti derivano chiaramente dal tempo di ciclo utilizzato ( $4ms$ ), ma come sottolineato dagli stessi

autori, almeno riguardo algoritmi del primo ordine, sono molto laschi per cicli molto brevi. Per gli esperimenti sono quindi state scelte le seguenti matrici:

$$\mathbf{K}_p = 50\mathbf{I}_6 \quad (7.11)$$

$$\mathbf{K}_d = 50\mathbf{I}_6 \quad (7.12)$$

#### 7.2.4 Inseguimento dei riferimenti

Nella formulazione classica del problema di ottimizzazione quadratica, l'inseguimento dei riferimenti generati dall'algoritmo Reflexxes è inserito come cifra di merito del problema:

$$\min \|s_{k+1} - s_{k+1_{rif}}\|_{Q_p}^2 + \|\dot{s}_{k+1} - \dot{s}_{k+1_{rif}}\|_{Q_v}^2 \quad (7.13)$$

$Q_p$  e  $Q_v$  rappresentano, nel caso generale, matrici diagonali definite positive di pesi che, in caso di incompatibilità nella minimizzazione di entrambe le componenti della cifra di merito, favoriscono maggiormente quella con peso superiore.

Nella formulazione gerarchica utilizzata, inserendo i due vincoli che si ottengono dalla cifra di merito in uno stesso livello di priorità, è possibile ottenere lo stesso comportamento definendo  $\sqrt{Q_v}$  e  $\sqrt{Q_p}$  come le matrici diagonali i cui elementi sono le radici quadrate dei singoli elementi di  $Q_v$  e  $Q_p$ :

$$\begin{cases} \sqrt{Q_v}\dot{s}_{k+1} = \sqrt{Q_v}\dot{s}_{k+1_{rif}} \\ \sqrt{Q_p}s_{k+1} = \sqrt{Q_p}s_{k+1_{rif}} \end{cases} \quad (7.14)$$

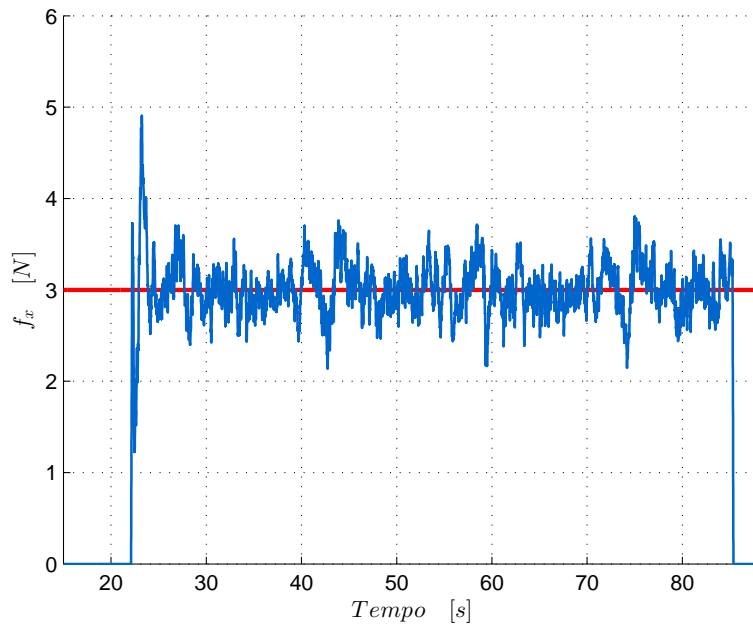
Nel caso in esame le matrici si riducono a scalari e si è deciso di privilegiare l'inseguimento del riferimento di velocità, pertanto si è scelto:

$$Q_p = 0.01 \quad Q_v = 1 \quad (7.15)$$

### 7.3 Risultati e correzione della stima di forza

In figura 7.3 è mostrato l'andamento della forza, avendo posto un riferimento di 3N. Essendo la superficie sufficientemente piana la forza misurata converge



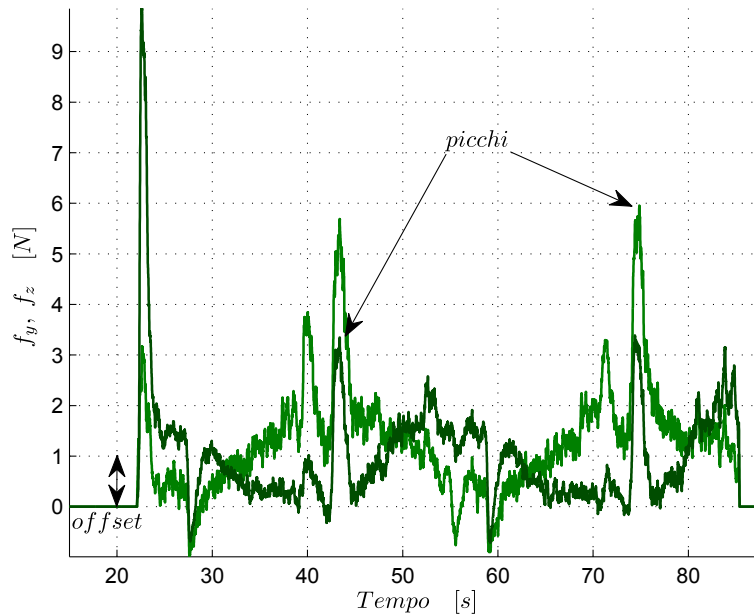


**Figura 7.3:** *SOTH2: Forza di contatto normale alla superficie.* Rosso: riferimento della forza di contatto - Blu: andamento della forza (direzione  $x$ ).

rapidamente al valore di riferimento in accordo col risultato ottenuto in simulazione (Fig. 5.8). Inizialmente si può notare il picco che segnala la realizzazione del contatto. Le visibili oscillazioni ad alta frequenza sono da imputare al calcolo per la stima delle coppie esterne ai giunti, che fa uso di segnali di coppia rumorosi.

Il fatto che la forza converga al riferimento dimostra l'efficacia della sintesi del vincolo, tuttavia andando ad analizzare le forze nelle direzioni tangenziali al piano si rileva una discrepanza dai valori attesi. Essendo il contatto sufficientemente approssimabile a uno di tipo puntiforme, ed effettuando un percorso circolare, ci si aspetterebbe un andamento sinusoidale a media nulla delle due forze dovute all'attrito, tra loro sfasate di  $90^\circ$ .

Gli andamenti ottenuti (Fig. 7.4) hanno forma sinusoidale, ma sono presenti picchi elevati, periodici, indipendenti dalla rugosità della superficie, e degli offset costanti sia su  $f_y$  che  $f_z$ , che indicano una componente costante di attrito in quelle direzioni, incompatibile con il percorso circolare definito.



**Figura 7.4:** *SOTH2: Forze tangenziali alla superficie.* Verde chiaro: andamento della forza in direzione  $y$  - Verde scuro: andamento della forza in direzione  $z$ .

I picchi sono imputabili al gioco e all'elasticità dei giunti del robot e alla particolare postura, pertanto compensabili solo effettuandone un'analisi completa. La misura in direzione normale non risente di questi effetti poiché la loro presenza è mascherata dal controllo, ma effettivamente si giunge a una ricostruzione errata della forza di contatto rispetto alla realtà.

L'esistenza degli offset induce un ulteriore errore di ricostruzione, il vettore complessivo di forza è in realtà ruotato rispetto a quello reale. Infatti effettuando la prova per diversi valori di riferimento si ottengono offset proporzionali alla forza normale registrata, a segnalare il fatto che parte di forza di contatto è in realtà calcolata lungo un'altra direzione.

Per compensare questa inesattezza sono state effettuate prove a diversa velocità e con diversi riferimenti e sono stati estratti due coefficienti di proporzionalità che legano gli offset alla forza di contatto misurata, e validi in un intervallo di

circa 2 – 7N:

$$offset_y = \alpha_y f_x = 0.3774 f_x \quad (7.16)$$

$$offset_z = \alpha_z f_x = 0.3120 f_x \quad (7.17)$$

Ad ogni iterazione dell'algoritmo vengono calcolati gli offset corrispondenti al valore di forza misurato e viene definito il vettore  $\mathbf{f}_{old}$  avente come componenti la forza normale misurata e gli offset calcolati:

$$\mathbf{f}_{old} = \begin{bmatrix} f_x \\ \alpha_y f_x \\ \alpha_z f_x \end{bmatrix} \quad (7.18)$$

Successivamente si ruota questo vettore in modo da sovrapporlo al versore dell'asse  $x$  della superficie, ovvero alla direzione interessata dal controllo di forza, ottenendo  $\mathbf{f}_{new}$ , di cui solo la prima componente è non nulla e rappresenta la nuova forza misurata dopo aver riportato la componente di forza dovuta agli offset nella direzione corretta:

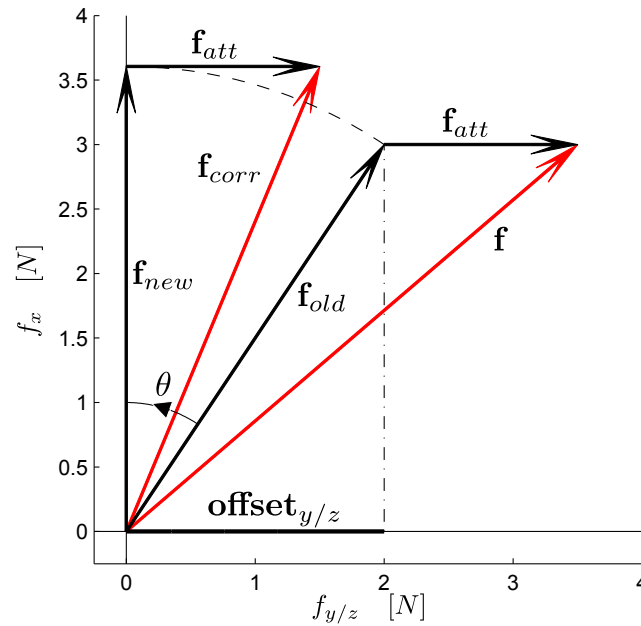
$$\mathbf{f}_{new} = \begin{bmatrix} f_{x_{corr}} \\ 0 \\ 0 \end{bmatrix} = \cos \theta \mathbf{f}_{old} - \sin \theta (\boldsymbol{\omega} \times \mathbf{f}_{old}) + (1 - \cos \theta) (\boldsymbol{\omega} \cdot \mathbf{f}_{old}) \boldsymbol{\omega} \quad (7.19)$$

Nell'equazione precedente si è fatto uso della formula di Rodrigues [28] per la rotazione del vettore, dove  $\boldsymbol{\omega}$  e  $\theta$  rappresentano l'asse e l'angolo di rotazione, facilmente ricavabili noti il versore  $x$  della terna della superficie e il vettore da ruotare  $\mathbf{f}_{old}$ .

Il vettore di misura corretto complessivo è quindi ottenuto da  $f_{x_{corr}}$  e le componenti  $f_y$  e  $f_z$  depurate dai termini di offset:

$$\mathbf{f}_{corr} = \begin{bmatrix} f_{x_{corr}} \\ f_y - \alpha_y f_x \\ f_z - \alpha_z f_x \end{bmatrix} \quad (7.20)$$

In figura 7.5 è rappresentato schematicamente, in due dimensioni, il procedimento adottato, dove  $\mathbf{f}_{att}$  è il vettore di attrito complessivo.

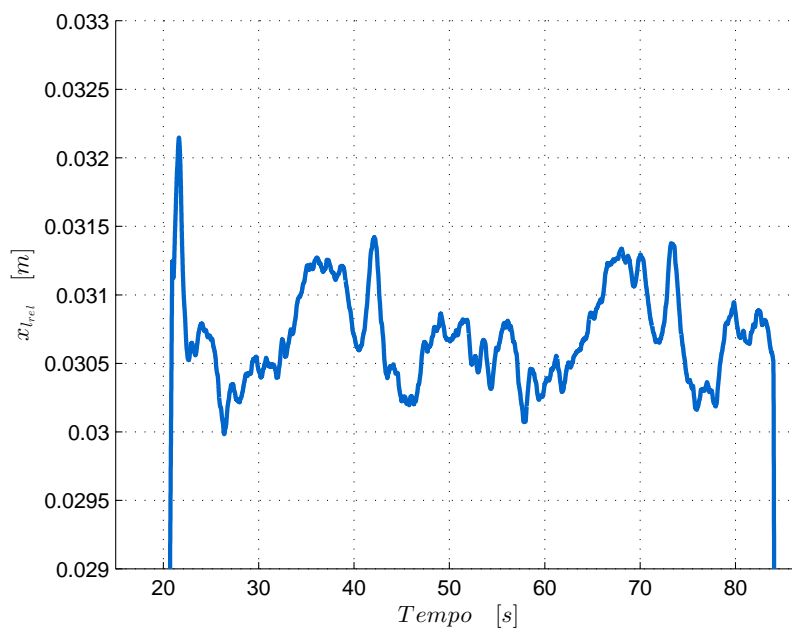


**Figura 7.5:** *Rappresentazione vettoriale della correzione della stima di forza.*

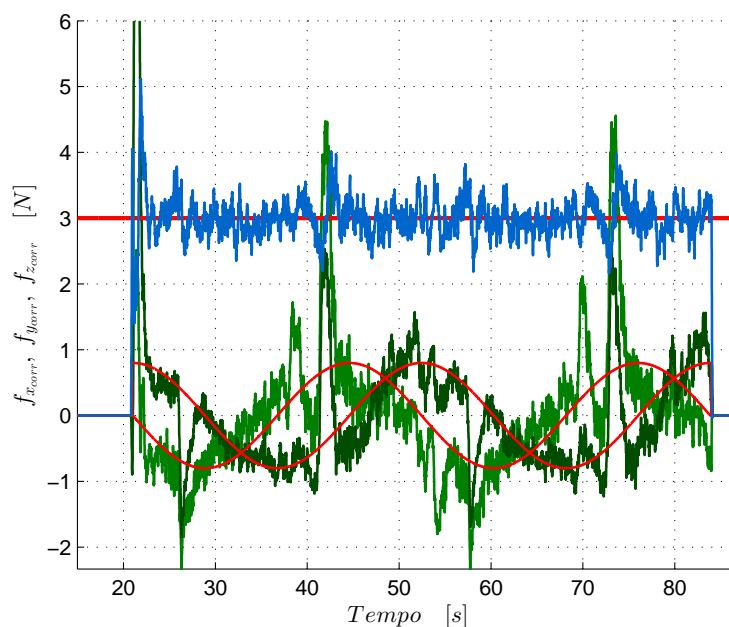
Ripetendo l'esperimento con questo accorgimento si riesce ad apprezzare come ora l'andamento delle componenti tangenziali di forza sia effettivamente sinusoidale, e come sia contemporaneamente mantenuto il riferimento di forza normale, è inoltre riportata la posizione del tool  $x_{l_{rel}}$  rispetto alla superficie (Fig. 7.6 e Fig. 7.7), confrontando  $f_x$  con  $f_{x_{corr}}$  si osserva la leggera differenza in valore assoluto dovuta alla correzione effettuata (Fig. 7.8).

Lo stesso risultato si ottiene anche nel caso di superfici più rigide, come nel caso del box di nylon (Fig. 7.9 e Fig. 7.10).

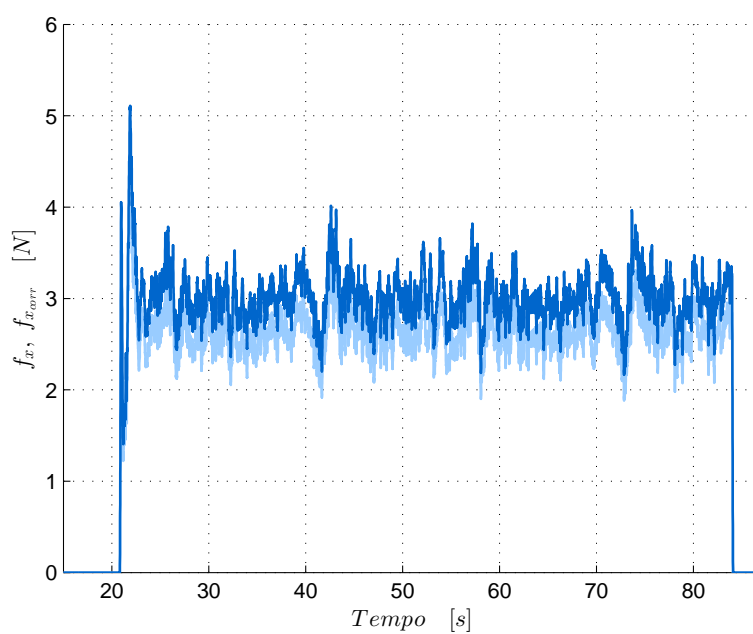
È possibile vedere, per entrambe le superfici, come in concomitanza ai picchi che si registrano nelle direzioni tangenziali di forza, si rilevino anche degli improvvisi affondamenti maggiori del tool nella superficie, a segnalare che in queste occasioni la forza non è correttamente ricostruita. La restante variabilità della posizione è invece attribuibile alle piccole variazioni di rigidità riscontrabili nel materiale, la ripetibilità delle prove è verificata notando che, effettuando due giri del percorso circolare di riferimento, gli andamenti di forze e posizioni risultano periodici.



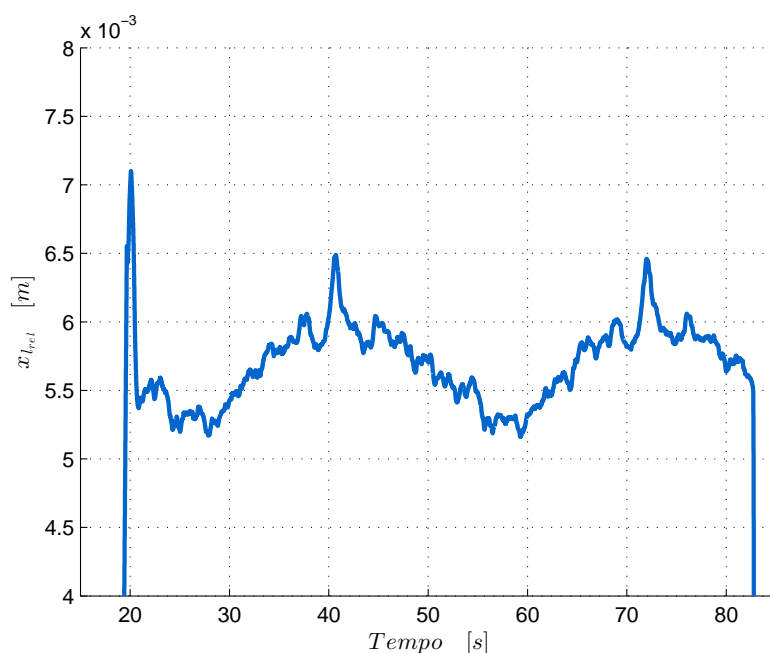
**Figura 7.6:** *SOTH2: Posizione relativa del tool rispetto alla superficie.* Blu: andamento della posizione (direzione  $x$ ). N.B. Non è importante il valore assoluto della posizione, ma la sua variazione ai fini del controllo di forza, poiché la terna della superficie può anche non essere posizionata sulla superficie stessa, per come è stato definito il vincolo.



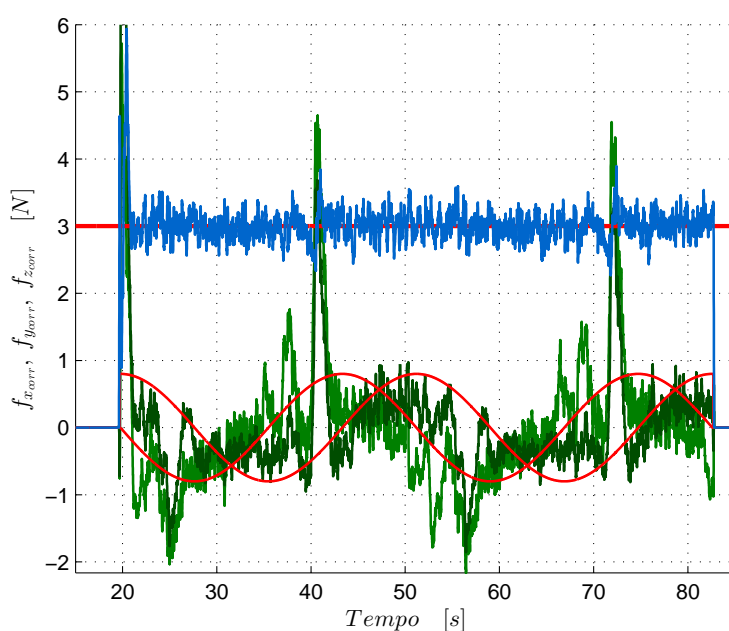
**Figura 7.7:** *SOTH2: Forza di contatto dopo aver effettuato la correzione.* Rosso: riferimento della forza di contatto e andamenti teorici delle forze di attrito - Blu: andamento della forza normale (direzione  $x$ ) - Verde chiaro: andamento della forza in direzione  $y$  - Verde scuro: andamento della forza in direzione  $z$ .



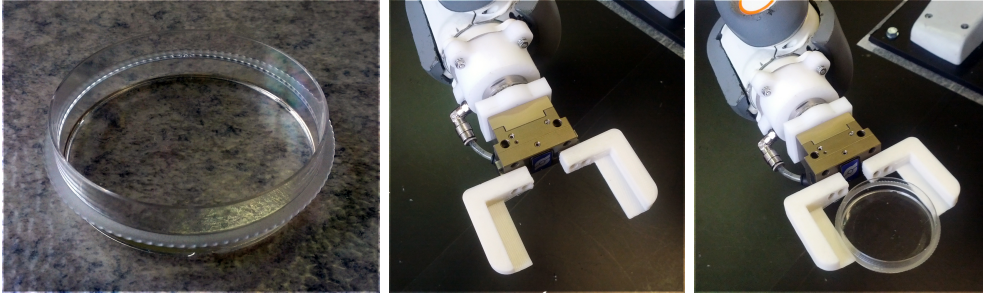
**Figura 7.8:** *SOTH2: Effetto della correzione sulla direzione della forza controllata.*  
Azzurro: andamento della forza normale pre-correzione - Blu: andamento della forza normale post-correzione (direzione  $x$ ).



**Figura 7.9:** *SOTH2: Posizione relativa del tool rispetto alla superficie di nylon.* Blu: andamento della posizione (direzione  $x$ ). N.B. Non è importante il valore assoluto della posizione, ma la sua variazione ai fini del controllo di forza, poiché la terna della superficie può anche non essere posizionata sulla superficie stessa, per come è stato definito il vincolo.



**Figura 7.10:** *SOTH2: Forza di contatto relativa alla superficie di nylon.* Rosso: riferimento della forza di contatto e andamenti teorici delle forze di attrito - Blu: andamento della forza normale (direzione  $x$ ) - Verde chiaro: andamento della forza in direzione  $y$  - Verde scuro: andamento della forza in direzione  $z$ .



**Figura 7.11:** *Piastra di Petri e gripper utilizzati nell'esperimento.*

## 7.4 Applicazione dual-arm su ABB FRIDA

Come verifica finale dell'algoritmo proposto, l'esperimento di controllo di forza è stato eseguito sul robot sfruttando le potenzialità date dalla struttura dual-arm.

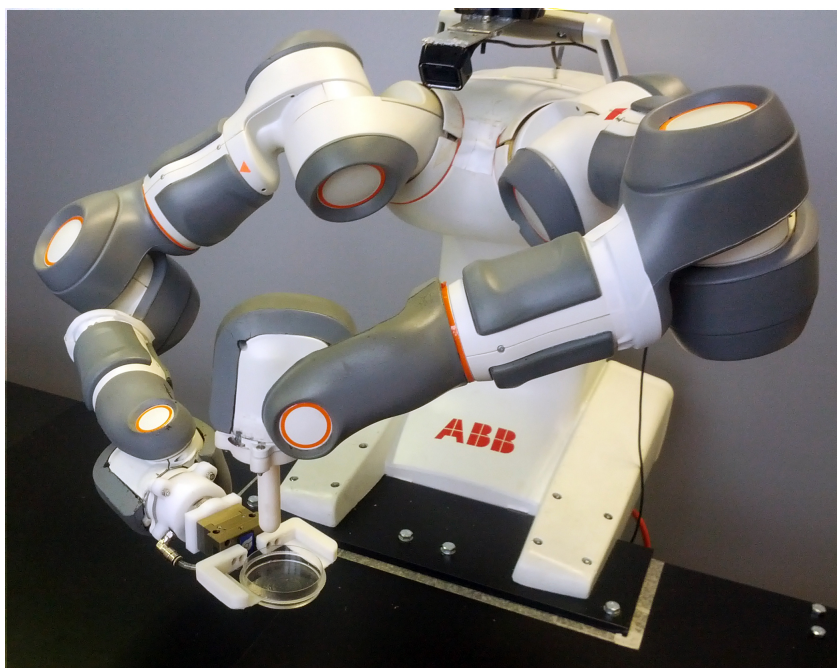
Come già descritto, il braccio sinistro deve eseguire il task di path-following e controllo di forza SOTH2 (5.26) su una superficie, in questo caso sorretta dal braccio destro. Per l'esperimento è stata utilizzata una piastra di Petri tenuta saldamente dal gripper montato sull'end-effector del braccio in esame (Fig. 7.11). L'applicazione è simile alle spatolature che vengono effettuate in campo biologico e medicale su campioni di tessuto o colture cellulari.

Nell'applicazione dual-arm risulta critica la postura delle due braccia, poiché in talune configurazioni potrebbe accadere che la forza si scarichi quasi interamente su uno dei giunti, richiedendo al robot di erogare una coppia più elevata di quella massima. Per aggirare questo problema il braccio destro è stato posto in un configurazione che distribuisse la forza sui primi quattro giunti, capaci di erogare coppie maggiori rispetto a quelli del polso (Fig. 7.12).

Viste le ridotte dimensioni del supporto, la traiettoria circolare è necessariamente ridotta rispetto a quella utilizzata nei precedenti esperimenti. In particolare essa ha un diametro di 2 cm e viene eseguita 4 volte.

Nelle figure 7.13 e 7.14 sono mostrate posizione relativa e forze di contatto misurate. Come si può vedere, il riferimento di forza è correttamente seguito, anche se rispetto agli esperimenti single-arm la misura è più disturbata a causa

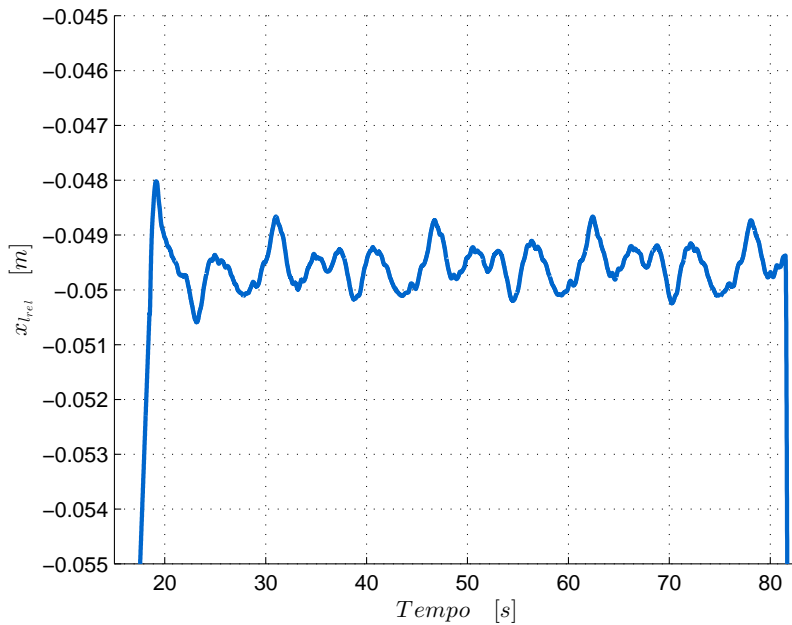




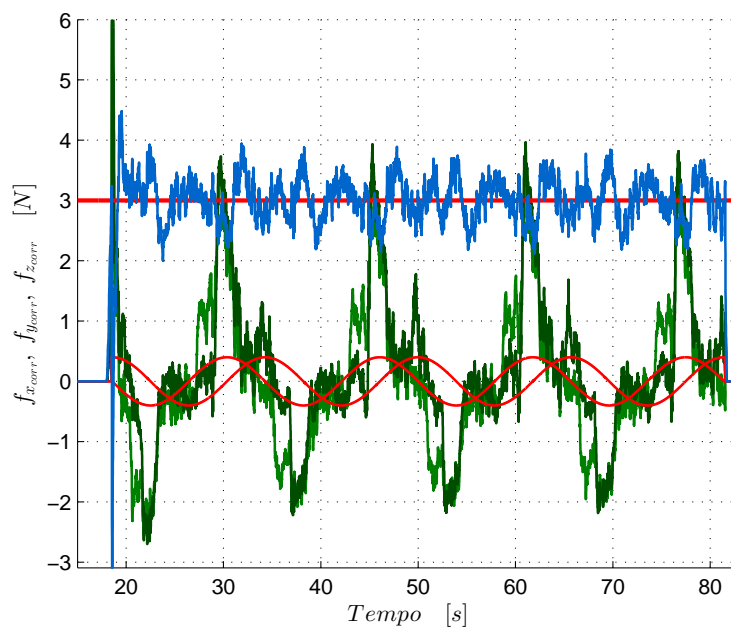
**Figura 7.12:** *Configurazione del braccio destro per l'esperimento dual-arm.*

dell'elevata cedevolezza del robot. Quest'ultima genera, inoltre, alcuni fenomeni di stick-slip vista la ridotta velocità di percorrenza, i quali si ripercuotono anche sull'andamento della forza nelle direzioni tangenziali. Lo scostamento da quello sinusoidale teorico è pertanto maggiore rispetto alle prove precedenti.

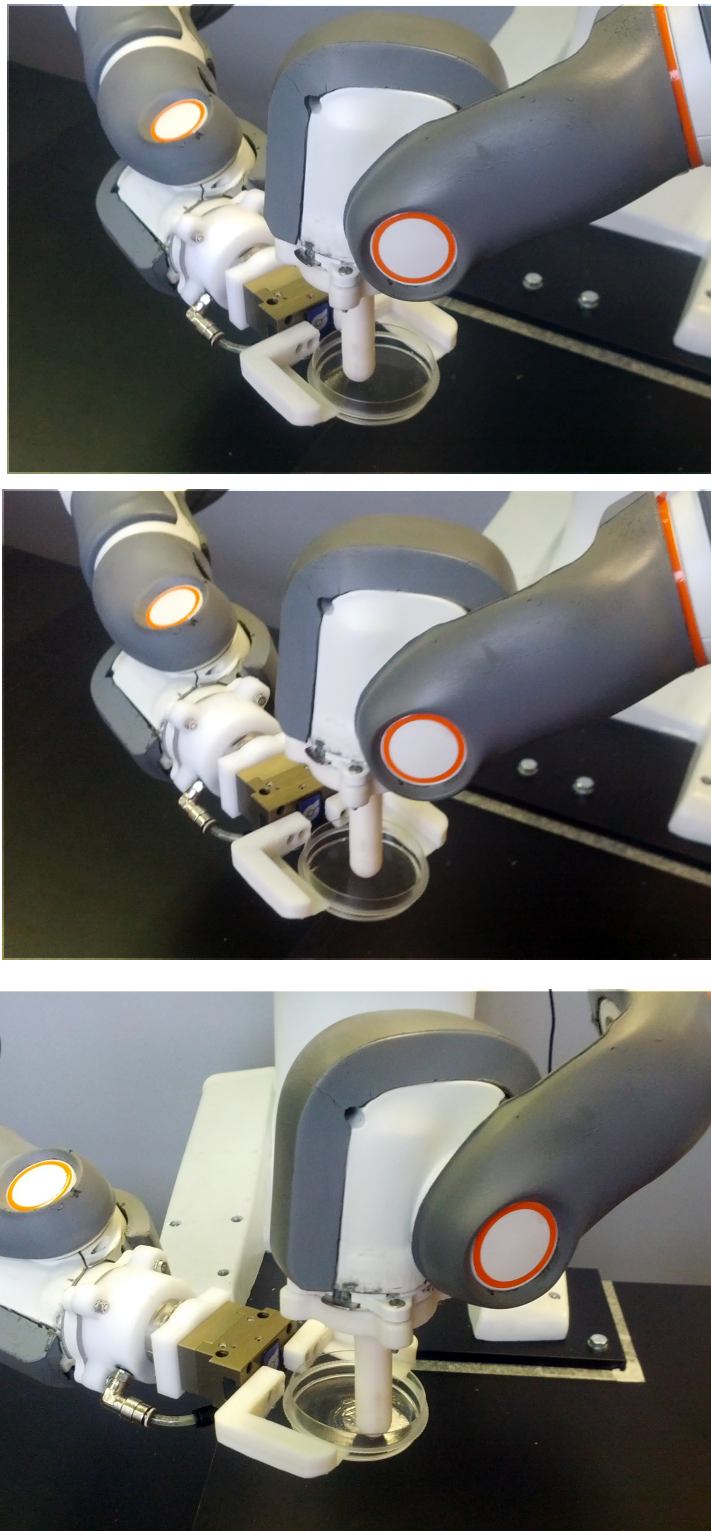
In Fig. 7.15 sono mostrate alcune fasi della prova effettuata.



**Figura 7.13:** *SOTH2 dual-arm: Posizione relativa del tool rispetto alla piastra di Petri.* Blu: andamento della posizione (direzione  $x$ ). N.B. Non è importante il valore assoluto della posizione, ma la sua variazione ai fini del controllo di forza, poiché la terna della superficie può anche non essere posizionata sulla superficie stessa, per come è stato definito il vincolo.



**Figura 7.14:** *SOTH2 dual-arm: Forza di contatto relativa alla piastra di Petri.* Rosso: riferimento della forza di contatto e andamenti teorici delle forze di attrito - Blu: andamento della forza normale (direzione  $x$ ) - Verde chiaro: andamento della forza in direzione  $y$  - Verde scuro: andamento della forza in direzione  $z$ .



**Figura 7.15:** *Robot a contatto durante l'esperimento dual-arm.* Le prime due figure mostrano il momento in cui avviene il contatto, data l'elevata compliance, il braccio si flette leggermente. Nella terza figura è mostrato un momento dell'esperimento durante la fase di controllo di forza.



## Capitolo 8

# Robustezza dell'algoritmo a disturbi geometrici

Nel capitolo 7 è stata verificata l'efficacia dei vincoli sintetizzati e la congruenza rispetto a quanto ottenuto in simulazione.

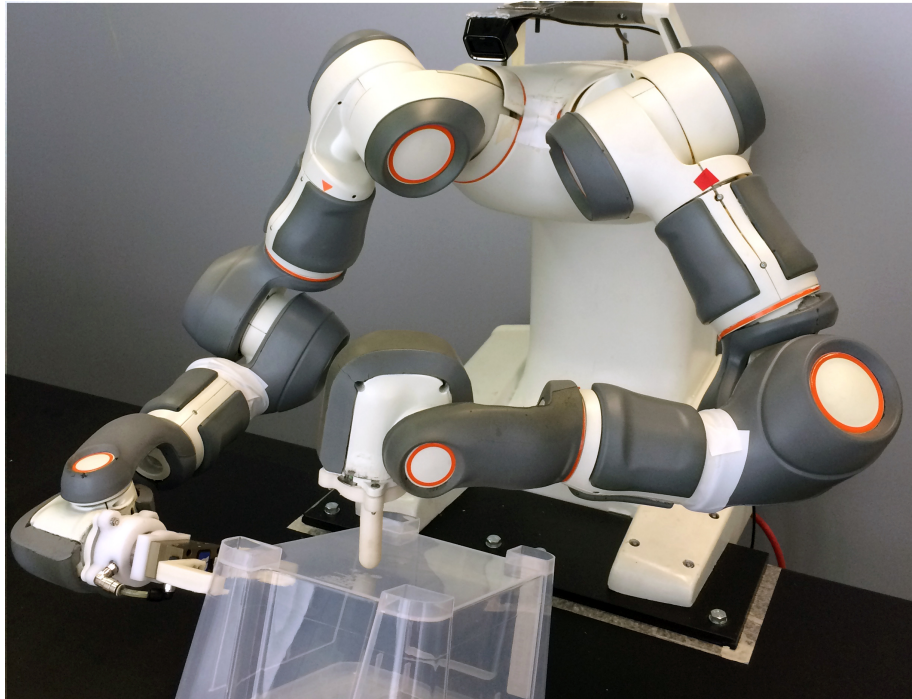
Le prove sono state effettuate in condizioni pressoché ideali, in cui orientamento e curvatura della superficie sono noti, ovvero con superficie piana.

Nel resto del capitolo verrà analizzato il comportamento dell'algoritmo in presenza di incertezze di tipo geometrico già estensivamente considerate in fase di sintesi del vincolo di forza e in simulazione, si cercherà poi di specificare una possibile soluzione ai problemi introdotti da questi disturbi.

### 8.1 Comportamento in presenza di disturbi

Per verificare la robustezza dell'algoritmo gli esperimenti sono stati ripetuti introducendo un errore nell'orientamento della superficie attraverso una rotazione negativa lungo il suo asse  $z$ , in tal modo si può apprezzare il comportamento sia a fronte di errori di orientamento che di disturbi relativi alla non planarità della superficie e dovuti a variazioni di curvatura non note (Fig. 8.1).

I test sono stati effettuati utilizzando come riferimento per il path-following un percorso circolare e un segmento orientato lungo l'asse  $y$  della superficie per-



**Figura 8.1:** *Apparato e configurazione utilizzati negli esperimenti per la robustezza.*

corso in entrambe le direzioni.

Nella figure 8.2 e 8.3 è mostrato l'andamento di posizione e forza espresse nella terna della superficie nel caso di percorso circolare. A causa del valore elevato di  $\delta_1$ , del valore di rigidezza utilizzato, e della rilevante entità dell'errore introdotto sull'orientamento per renderne ben visibili gli effetti, il controllo non riesce a mantenere il riferimento, ma come evidenziato in simulazione, insorgono delle oscillazioni tanto più ampie quanto più elevati sono il disturbo e la velocità di percorrenza.

Bisogna precisare che non si tratta di un comportamento al limite di stabilità, il controllo cerca di mantenere il riferimento, ma le repentine variazioni dell'altezza della superficie introducono le oscillazioni; è anche da sottolineare che l'utilizzo all'interno dell'algoritmo di più elevati valori di rigidezza diminuisce le capacità di attenuare questi disturbi.

Questo comportamento è conforme a quanto analizzato in precedenza in presenza di disturbi: una velocità di percorrenza più elevata si traduce in una frequenza a cui agisce il disturbo più elevata, e quindi meno attenuata dal controllo, come



evidenziato dalla funzione di trasferimento di sensitività (4.30).

Nel caso di percorso rettilineo i risultati sono simili, nella prima parte di moto il tool vede una superficie con pendenza positiva, pertanto si registra una forza maggiore del riferimento, al contrario, durante il movimento di ritorno, la pendenza è negativa e la forza misurata inferiore al riferimento.

## 8.2 Limitazione dinamica della velocità

In simulazione si è visto come una più elevata velocità di attraversamento del percorso accentui i disturbi dovuti a geometrie della superficie non considerate, al contrario, un rallentamento della velocità permette al controllo di filtrarli e assicurare un miglior tracking della forza, a dispetto di un tempo di esecuzione maggiore (Fig. 5.17).

Si può quindi inserire un ulteriore livello di priorità nell'equazione (5.26), subito al di sotto di quello relativo al riferimento di forza, che descriva un limite superiore variabile per  $\dot{s}$ , dipendente dal valore assoluto dell'errore di tracking della forza a quell'istante di tempo.

Il vincolo può essere scritto in termini relativi, noti la velocità massima ammissibile per  $s$  ( $\dot{s}_{max}$ ) e il riferimento di forza ( $f_{rif}$ ):

$$\frac{\dot{s}}{\dot{s}_{max}} \leq 1 - \alpha \frac{|e_{fx}|}{f_{rif}} \quad (8.1)$$

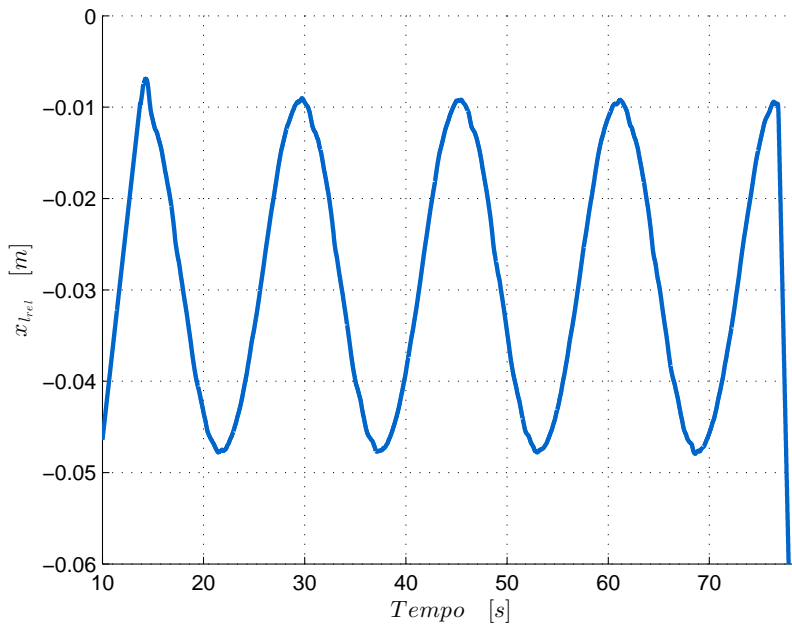
Applicando il lemma di Grönwall [21] si ottiene una relazione che dipende dall'accelerazione  $\ddot{s}$ , cioè dalla variabile di ottimizzazione:

$$\delta_s \ddot{s} \leq \dot{s}_{max} - \dot{s} - \alpha \frac{\dot{s}_{max}}{f_{rif}} \left( |e_{fx}| + \delta_s |e_{fx}| \right) \quad (8.2)$$

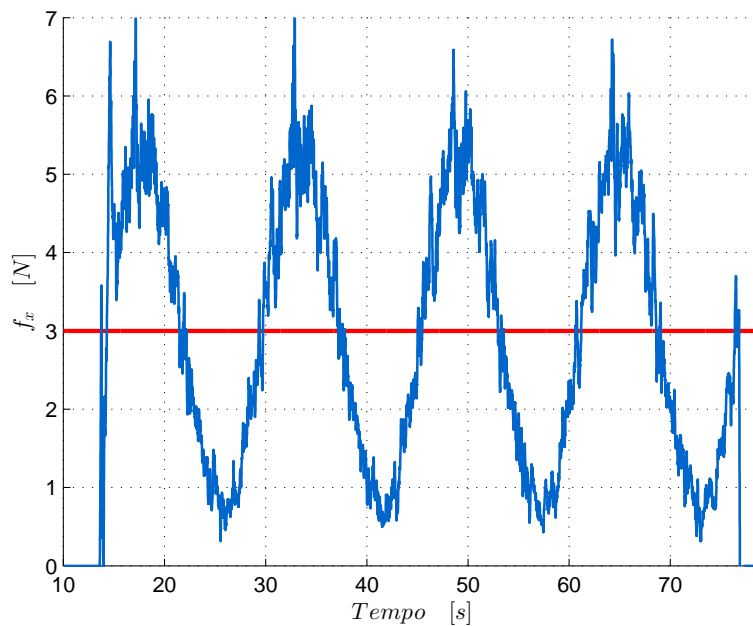
Nonostante la non differenziabilità in 0, la presenza del valore assoluto nel vincolo non genera particolari problemi per il calcolo di  $|e_{fx}|_k$ , poiché la derivata è effettuata numericamente e assunta nulla in caso  $|e_{fx}|_k = 0$  per due istanti consecutivi.

Con la discretizzazione usata fino ad ora si ottiene:

$$\delta_s \ddot{s}_k \leq \dot{s}_{max} - \dot{s}_k - \alpha \frac{\dot{s}_{max}}{f_{rif}} \left( |e_{fx}|_k + \delta_s |e_{fx}|_k \right) \quad (8.3)$$



**Figura 8.2:** *SOTH2: Posizione relativa del tool rispetto alla terna della superficie non ruotata.* Blu: andamento della posizione (direzione  $x$ ). N.B. Non è importante il valore assoluto della posizione, ma la sua variazione ai fini del controllo di forza, poiché la terna della superficie può anche non essere posizionata sulla superficie stessa, per come è stato definito il vincolo.



**Figura 8.3:** *SOTH2: Forza di contatto normale alla superficie non ruotata.* Rosso: riferimento della forza di contatto - Blu: andamento della forza normale (direzione  $x$ ).



$\delta_s$  e  $\alpha$  sono parametri di progetto. Per la scelta del primo può essere adottata la stessa strategia utilizzata per  $\delta_1$ :

$$\delta_s = \frac{\dot{s}_{max}}{\ddot{s}_{max}} \quad (8.4)$$

Per ciò che riguarda  $\alpha$ , esso indica quanto incide l'errore di inseguimento sulla diminuzione di  $\dot{s}$ , utilizzare un  $\alpha$  elevato imporrà un limite superiore alla velocità molto stringente, mentre sarà più blando se minore; se troppo elevato, si potrebbero verificare bruschi e frequenti momenti di stop e ripartenza del moto.

L'equazione (8.3) è nella forma utilizzata dagli algoritmi di ottimizzazione:

$$A_s \ddot{s}_k \leq b_s \quad (8.5)$$

dove:

$$A_s = \delta_s \quad (8.6)$$

$$b_s = \dot{s}_{max} - \dot{s}_k - \alpha \frac{\dot{s}_{max}}{f_{rif}} \left( |e_{f_x}|_k + \delta_s |e_{\dot{f}_x}|_k \right) \quad (8.7)$$

Per il calcolo del vincolo a tempo di esecuzione è ovviamente necessario utilizzare la misura di forza, la quale è inevitabilmente soggetta a rumore, al fine di rendere la variazione di velocità più morbida quando il vincolo (8.3) è attivo e quindi ottenere un movimento più omogeneo, la forza va opportunamente filtrata. In questo caso è stato utilizzato un filtro del primo ordine:

$$f_{x_{filt,k+1}} = (1 - \lambda)f_{x_{filt,k}} + \lambda f_{x_k} \quad (8.8)$$

## 8.3 Risultati

Per l'esperimento sono stati utilizzati i seguenti parametri:

$$\dot{s}_{max} = 0.1 \frac{rad}{s} \quad (8.9)$$

$$\ddot{s}_{max} = 0.5 \frac{rad}{s^2} \quad (8.10)$$

$$\delta_s = 0.2s \quad (8.11)$$

$$\alpha = 2.5 \quad (8.12)$$

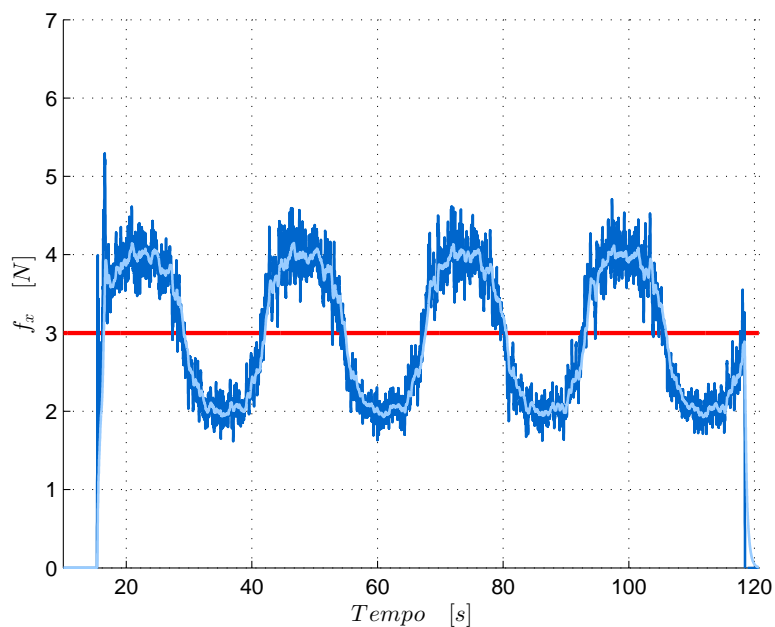
$$\lambda = 0.01 \quad (8.13)$$

Nelle figure 8.4 e 8.5 sono mostrati il nuovo andamento della forza e quello della velocità dell'ascissa curvilinea.

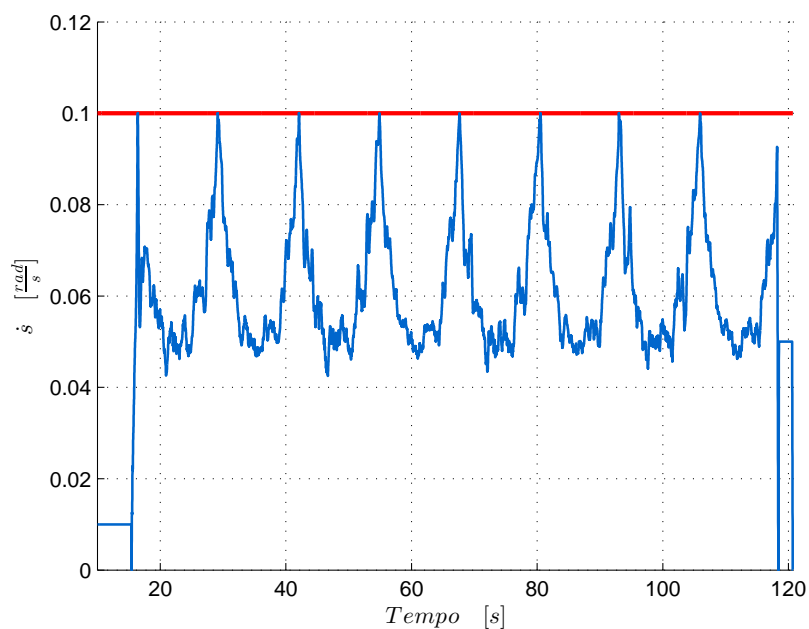
Rispetto al caso in assenza di questo vincolo aggiuntivo (Fig. 8.3) si ha una riduzione dell'errore di circa il 50%. Dal grafico di  $\dot{s}$  si vede come quando l'errore di forza è elevato la velocità raggiunga il suo minimo, mentre per errore vicino a zero è libera di assumere il valore definito da  $\dot{s}_{max}$ .

Simili risultati si ottengono fornendo come percorso di riferimento il segmento (Fig. 8.6 e Fig. 8.7): l'errore di inseguimento si mantiene costante, questo è dovuto al fatto che, nel caso del segmento e dell'errore di orientamento introdotto, il segnale di disturbo agente sulla forza corrisponde da una rampa. Ricordando la (4.30) e il corrispondente diagramma di Bode 4.6, si vede come sia presente un unico zero in 1, il quale è in grado di annullare l'errore di inseguimento a transitorio esaurito, al più per disturbi a scalino.

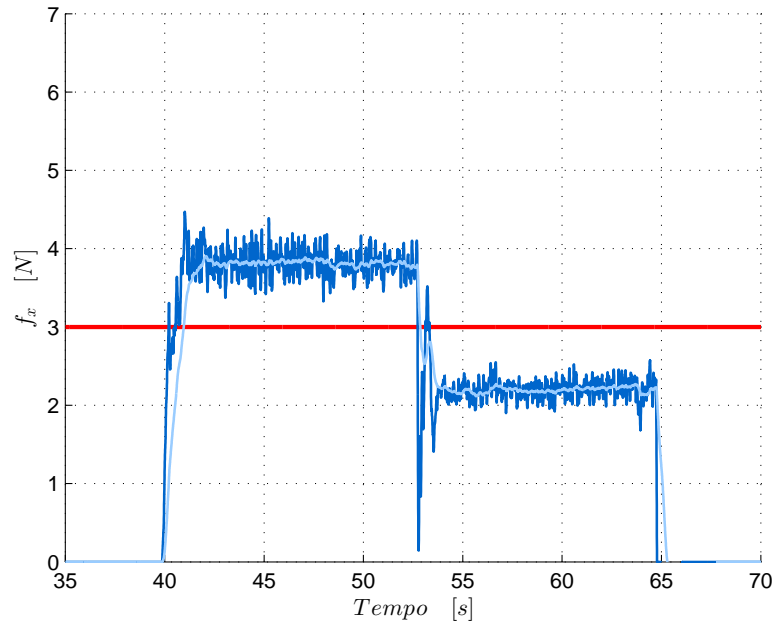
In entrambi i casi permane un inevitabile errore di inseguimento dovuto alle scelte di  $\delta_1$  e  $\alpha$ , la prima dovrebbe essere abbassata per ottenere migliori prestazioni, mentre la seconda aumentata per limitare ulteriormente la velocità quando è presente l'errore sul tracking di forza.



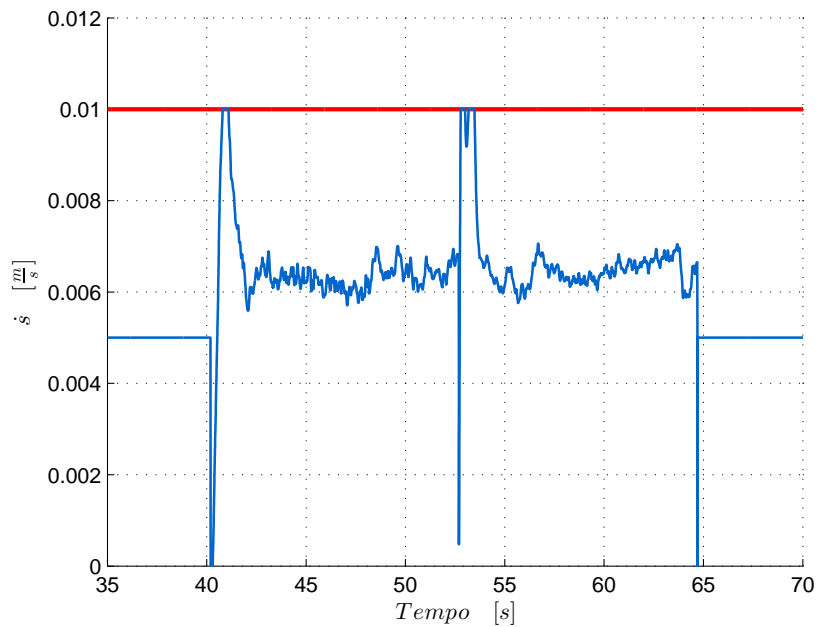
**Figura 8.4:** *SOTH2: Forza di contatto normale alla superficie non ruotata.* Rosso: riferimento della forza di contatto - Blu: andamento della forza normale (direzione  $x$ ) - Azzurro: forza filtrata utilizzata nella definizione del vincolo (8.3).



**Figura 8.5:** *SOTH2: Velocità dell'ascissa curvilinea.* Rosso: velocità massima consentita ( $\dot{s}_{max}$ ) - Blu: andamento della velocità dell'ascissa curvilinea.



**Figura 8.6:** *SOTH2: Forza di contatto normale alla superficie non ruotata (segmento).* Rosso: riferimento della forza di contatto - Blu: andamento della forza normale (direzione  $x$ ) - Azzurro: forza filtrata utilizzata nella definizione del vincolo (8.3).



**Figura 8.7:** *SOTH2: Velocità dell'ascissa curvilinea (segmento).* Rosso: velocità massima consentita ( $\dot{s}_{max}$ ) - Blu: andamento della velocità dell'ascissa curvilinea.

## Capitolo 9

# Conclusioni

Nella presente tesi si è visto come l'utilizzo di un algoritmo reattivo di generazione della traiettoria sia in grado di fornire un alto livello di flessibilità nella definizione dei movimenti da impartire al robot. La fase di programmazione del moto risulta semplificata, infatti non è più richiesta la traduzione diretta degli obiettivi produttivi in riferimenti di moto, che sono invece definiti tramite vincoli e interpretati in riferimenti per il robot solo a tempo di esecuzione. Questo approccio permette inoltre l'impiego di segnali esterni per meglio adattare online la traiettoria ideale definita in fase di progetto.

Sono state investigate varie soluzioni nell'implementazione del blocco di Controllo Reattivo, valutandone pro e contro: la varietà di task che è possibile definire e l'elevata efficienza computazionale hanno fatto ricadere la scelta su un algoritmo di ottimizzazione gerarchica dedicato, tuttavia le altre tecniche rimangono sicuramente valide per la definizione di compiti più semplici e meno esigenti dal punto di vista del calcolo numerico.

L'algoritmo è stato poi esteso per includere task che prevedano il controllo di forza, arrivando alla formalizzazione di un vincolo che consenta sia l'inseguimento di un riferimento di forza, sia l'introduzione di limiti meno stringenti sul suo valore, permettendo il controllo contemporaneo di forza in talune direzioni e posizione in altre. La stabilità e la robustezza del vincolo sintetizzato sono state approfondite, evidenziando il contributo fornito dai parametri di progetto,

in particolare quello di rigidezza; è stato quindi possibile fornire delle linee guida per la loro scelta nel caso di applicazioni reali.

I risultati ottenuti in simulazione hanno permesso di confermare quanto desunto dai risultati teorici, attraverso l'elaborazione di prove ad hoc. Essi hanno inoltre fatto emergere come le incertezze sulla geometria di contatto possano influire negativamente sulle prestazioni.

Gli esperimenti eseguiti sul robot ABB FRIDA hanno convalidato la formulazione adottata. L'assenza di sensori che misurassero direttamente la forza di contatto ha richiesto la modellizzazione e identificazione delle coppie di attrito ai giunti, con lo scopo di implementare un adeguato osservatore; da questo punto di vista, benché il modello sia sufficientemente fedele per l'applicazione sperimentale eseguita, le prestazioni ottenibili a basse velocità sono ancora carenti e in futuro potrebbe essere opportuna l'adozione di modelli dell'attrito più complessi.

Il rumore sulla misura delle coppie ai giunti, i giochi e l'elevata compliance del robot hanno reso necessaria la correzione della stima di forza: per meglio compensare questi fenomeni potrebbe perciò essere interessante realizzarne una modellizzazione completa, sia per migliorare la qualità della stima che del controllo nel suo complesso.

Esperimenti sono stati effettuati anche per determinare come la relazione tra la velocità di movimento dell'end-effector e i disturbi di natura geometrica agenti sulla forza influenzi le prestazioni: è stato quindi ricavato un particolare vincolo che permette, in parte, di migliorare i risultati.

Come è stato più volte fatto presente, la scelta del parametro di rigidezza rispetto a quella reale è estremamente rilevante per conferire prestazioni desiderate (Fig. 4.4 e Fig. 4.5), una stima sufficientemente corretta della rigidezza dell'ambiente consentirebbe di poter imporre al robot la banda di controllo desiderata: ciò si potrebbe ottenere mediante opportuni stimatori online [29], che permetterebbero di raggiungere prestazioni costanti anche in caso di variazioni della rigidezza.

# Bibliografia

- [1] S. Quinlan, O. Khatib. Elastic bands: Connecting path planning and control. *1993 IEEE International Conference on Robotics and Automation*, 1993.
- [2] O. Brock, O. Khatib. Elastic strips: A framework for motion generation in human environments. *International Journal of Robotics Research*, 2002.
- [3] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, H. Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 2007.
- [4] A. M. Zanchettin, P. Rocco. Reactive motion planning and control for compliant and constraint-based task execution. *2015 IEEE International Conference on Robotics and Automation*, 2015.
- [5] E. Lutscher, G. Cheng. A practical approach to generalized hierarchical task specification for indirect force controlled robots. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.
- [6] E. Lutscher, G. Cheng. Hierarchical inequality task specification for indirect force controlled robots using quadratic programming. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.
- [7] J. De Schutter, W. Decré, R. Smits, H. Bruyninckx. Extending the iTaSC constraint-based robot task specification framework to time-independent

- trajectories and user-configurable task horizons. *2013 IEEE International Conference on Robotics and Automation*, 2013.
- [8] A. Del Prete, F. Romano, L. Natale, G. Metta, G. Sandini, F. Nori. Prioritized optimal control. *2014 IEEE International Conference on Robotics and Automation*, 2014.
- [9] A. M. Zanchettin, P. Rocco. Near time-optimal and sensor-based motion planning for robotic manipulators. *52nd IEEE Conference on Decision and Control*, 2013.
- [10] T. Kröger, F. M. Wahl. Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Transactions on Robotics*, 2010.
- [11] A. De Luca, G. Oriolo, B. Siciliano. Robot redundancy resolution at the acceleration level. *Laboratory Robotics and Automation*, 1992.
- [12] B. Siciliano, J.-J. E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. *IEEE International Conference on Advanced Robotics*, 1991.
- [13] A. Escande, N. Mansard, P.-B. Wieber. Fast resolution of hierarchized inverse kinematics with inequality constraints. *2010 IEEE International Conference on Robotics and Automation*, 2010.
- [14] O. Kanoun, F. Lamiroux, P.-B. Wieber. Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 2011.
- [15] H. J. Ferreau, A. Potschka, C. Kirches. qpOASES webpage. <http://www.qpOASES.org/>, 2007–2014.
- [16] M. De Lasa, I. Mordatch, A. Hertzmann. Feature-based locomotion controllers. *ACM SIGGRAPH '10*, 2010.



- 
- [17] A. Escande, N. Mansard, P.-B. Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 2014.
- [18] A. Escande, N. Mansard, P.-B. Wieber. SOTH github webpage. <https://github.com/stack-of-tasks/soth>, 2010–2015.
- [19] J. De Schutter, W. Decré, R. Smits, H. Bruyninckx. Extending iTaSC to support inequality constraints and non-instantaneous task specification. *2009 IEEE International Conference on Robotics and Automation*, 2009.
- [20] J. De Schutter, T. De Laet, J. Rutgeerts, E. Aertbeliën, F. de Groote, T. Lefebvre, W. Verdonck, H. Bruyninckx. Unified constraint-based task specification for complex sensor-based robot systems. *2005 IEEE International Conference on Robotics and Automation*, 2005.
- [21] 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.
- [22] A. M. Zanchettin, P. Rocco. Constraint-based control strategy for reactive motion planning in robotic manipulators. *Draft*, 2014.
- [23] A. De Luca, R. Mattone. Actuator failure detection and isolation using generalized momenta. *2003 IEEE International Conference on Robotics and Automation*, 2003.
- [24] A. De Luca, R. Mattone. Sensorless robot collision detection and hybrid force/motion control. *2005 IEEE International Conference on Robotics and Automation*, 2005.
- [25] T. Tjahjowidodo, F. Al-Bender, H. Van Brussel. Friction identification and compensation in a DC motor. *International Federation of Automatic Control 2005*, 2005.
- [26] ABB Ltd. ABB YuMi press release. <http://goo.gl/KJHxbm>, 2014–2015.

- [27] P. Falco, C. Natale. On the stability of closed-loop inverse kinematics algorithms for redundant robots. *IEEE Transactions on Robotics*, 2011.
- [28] O. Rodrigues. Des lois géométriques qui regissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérées indépendantes des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, 1840.
- [29] F. Coutinho, R. Cortesão. Force-based stiffness estimation for robotic tasks. *51st IEEE Conference on Decision and Control*, 2012.