



SAPIENZA
UNIVERSITÀ DI ROMA

Stima delle Forze di Collisione:
Un'Analisi Comparativa di Metodi Basati su Energia,
Velocità e Momento

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Ingegneria Informatica e Automatica

Francesco Viscione

Matricola 2084759

Relatore

Prof. Giuseppe Oriolo

Correlatore

Dr. Nicola Scianca

Anno Accademico 2024/2025

Tesi discussa il 3 Dicembre 2025
di fronte a una commissione esaminatrice composta da:
Prof. Paolo Di Giamberardino
Prof. Francesco Liberati

Stima delle Forze di Collisione: Un'Analisi Comparativa di Metodi Basati su Energia, Velocità e Momento
Tesi di Laurea Triennale. Sapienza Università di Roma

© 2025 Francesco Viscione. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Versione: 28 novembre 2025

Email dell'autore: viscione.francesco@protonmail.com

*A robot may not injure a human being or, through inaction,
allow a human being to come to harm.*

Isaac Asimov

Ringraziamenti

Il mio più sentito ringraziamento va al Prof. Giuseppe Oriolo e al Dott. Nicola Scianca: grazie per avermi guidato in questo percorso con pazienza e professionalità, offrendomi un'opportunità di crescita fondamentale.

Un grazie ai ragazzi del Robotics Lab del DIAG per la disponibilità e l'aiuto pratico dimostrato durante le sessioni di lavoro in laboratorio.

Dedico infine un pensiero a tutte le persone che ho incontrato in questi tre anni: grazie per aver reso questo capitolo della mia vita un viaggio fantastico.

Indice

1 Introduzione	1
1.1 La Nuova Era della Robotica Collaborativa	1
1.2 Sensori Propriocettivi e Esterocettivi	1
1.3 Obiettivi della Tesi	3
2 Fondamenti Teorici	4
2.1 Modello Dinamico di un Robot	4
2.2 Energia Meccanica di un Robot	5
2.3 Momento Generalizzato	5
2.4 Matrici Jacobiane e Forze Generalizzate	7
3 Osservatori Basati su Modello per il Rilevamento, l'Isolamento e l'Identificazione di Collisioni	10
3.1 L'Osservatore Basato sull'Energia	10
3.2 L'Osservatore Basato sulla Velocità	11
3.3 L'Osservatore Basato sul Momento Generalizzato	13
3.4 Rilevamento, Isolamento e Identificazione	15
3.4.1 Rilevamento	15
3.4.2 Isolamento	17
3.4.3 Identificazione	19
4 Analisi Sperimentale su Manipolatore Seriale	20
4.1 Confronto delle Prestazioni (MATLAB & Simulink)	20
4.1.1 Scenario di Test 1: Collisione a Velocità Non Nulla	20
4.1.2 Scenario di Test 2: Collisione Istantanea a Velocità Nulla	23
4.2 Validazione classe C++ <code>MomentumObserver</code> (MuJoCo & Pinocchio)	24
5 Applicazione su Robot Umanoide: Stima delle Forze di Contatto al Suolo	26
5.1 Modello Dinamico di un Robot Umanoide	26
5.2 L'Osservatore basato sul Momento Generalizzato per Robot Umanoidi	27
5.3 Analisi Sperimentale sul Robot Unitree G1	27
6 Conclusioni e Sviluppi Futuri	31
6.1 Analisi Comparativa dei Metodi	31
6.2 Estensione ai Robot Umanoidi	32
6.3 Sviluppi Futuri	32

A Implementazione della Classe C++ MomentumObserver	34
A.1 Interfaccia Pubblica (API)	34
A.1.1 Costruzione e Inizializzazione	34
A.1.2 Ciclo di Aggiornamento	35
A.1.3 Metodi di Accesso per Rilevamento, Isolamento e Identificazione	35
B Codice Sorgente	36
Bibliografia	37

Capitolo 1

Introduzione

In questo capitolo introduttivo viene delineato lo *scenario* della robotica collaborativa moderna, evidenziando la crescente necessità di garantire una coesistenza sicura tra uomo e macchina. Vengono classificati i *sensori* utilizzati in robotica analizzandone vantaggi e limitazioni nel contesto della sicurezza. Infine, vengono definiti gli *obiettivi* della tesi, che mira a confrontare metodi di rilevamento delle collisioni basati esclusivamente sulla *proprioceuzione*, con una particolare attenzione all'applicazione su *robot umanoidi*.

1.1 La Nuova Era della Robotica Collaborativa

Oggi giorno la presenza di robot nei nostri ambienti quotidiani è sempre più diffusa. Che si tratti di un ambiente lavorativo o domestico è molto probabile che ci ritroviamo a che fare con uno di loro, dal semplice robot aspirapolvere, al manipolatore industriale, arrivando fino ai robot quadrupedi (Fig. 1.1a). In questo contesto è di rilevante importanza garantire una sicura coesistenza tra uomo e macchina. In alcuni ambienti come quello industriale si sta infatti diffondendo l'utilizzo di robot collaborativi che condividono lo spazio di lavoro con operatori umani. L'obiettivo è quello di farli interagire nelle fasi di lavorazione al fine di aumentare i possibili scenari produttivi. In applicazioni come queste, il robot dovrebbe essere in grado di evitare collisioni indesiderate e gestire in maniera sicura e robusta quelle intenzionali o inevitabili.

1.2 Sensori Propriocettivi e Esterocettivi

Al fine di garantire una *physical human-robot interaction* (pHRI) sicura è quindi necessario che il robot percepisca l'ambiente circostante. Risulta quindi essenziale l'utilizzo di sensori. Essi possono essere classificati nel seguente modo:

- **Propriocettivi:** misurano variabili relative lo stato interno del robot;
- **Esterocettivi:** forniscono informazioni sull'ambiente circostante.

Per fare qualche esempio, gli *Encoder* (Fig. 1.1b) che permettono di misurare posizione e velocità dei giunti, ricadono nella categoria dei sensori propriocettivi.

Telecamere, Lidar, Pelli Tattili e Sensori di Forza-Coppia ricadono nei sensori esterocettivi. È evidente come i sensori esterocettivi siano utili alla nostra causa, vediamo alcuni esempi applicativi:

- È possibile utilizzare *Lidar* o *Stereo Camere* per determinare la posizione degli ostacoli rispetto al robot. Ciò permette di *evitare, rilevare e isolare* eventuali collisioni;
- L'utilizzo di *Pelli Tattili* permette al robot di capire *se e dove* sta toccando qualcosa;
- I *Sensori di Forza-Coppia* permettono di avere una *misura* diretta delle forze di contatto.

Bisogna però tenere in considerazione anche le loro limitazioni. Una *Stereo Camera* o un *Lidar* possono avere punti ciechi nell'inquadratura. Per esempio in una cella di produzione il robot stesso potrebbe ostacolare la visuale del sensore. Per quanto riguarda le *Pelli Tattili*, risulta estremamente complesso se non impossibile ricoprire l'intera superficie del robot. Infine i *Sensori di Forza-Coppia* sono limitati ad avere una misura di ciò che avviene a valle del loro punto d'installazione. Un altro fattore rilevante è il seguente: mentre i sensori propriocettivi sono strettamente necessari al controllo del robot e di conseguenza già forniti dai produttori, i sensori esterocettivi sono di solito aggiunti a posteriori dall'acquirente. Supponiamo ad esempio di voler acquistare un braccio robotico per effettuare operazioni di levigatura. Per poter controllare le forze impresse sul prodotto, si dovrà montare un *Sensore di Forza-Coppia* a monte dello strumento. Tale sensori hanno un prezzo che si aggira intorno a qualche migliaia di euro.



(a) Il robot quadrupede Unitree Go2 (Per gentile concessione di Unitree)



(b) Encoder Incrementale,
Foto di Joao Paulo Chagas, licenza CC BY 4.0.

Figura 1.1

1.3 Obiettivi della Tesi

Immaginate di aver perso completamente il tatto e la vista; come fareste a capire se state toccando qualcosa? Se un robot potesse rispondere a questa domanda probabilmente menzionerebbe uno dei metodi presentati in questa tesi. Nei capitoli successivi verranno esposti e messi a confronto tre metodi basati su *Osservatori* per il *Rilevamento*, l'*Isolamento* e l'*Identificazione* delle collisioni mediante l'utilizzo esclusivo di *sensori propriocettivi*. Verrà infine presentata una possibile implementazione su un robot umanoide; con l'obiettivo di stimare le forze di contatto al suolo durante la camminata. Ciò sarà utile per fini di ricerca al *Robotics Lab* del *Dipartimento di Ingegneria Informatica, Automatica e Gestionale* della *Sapienza Università di Roma*.

Capitolo 2

Fondamenti Teorici

Questo capitolo richiama gli strumenti matematici e fisici necessari alla comprensione degli algoritmi presentati nel seguito del lavoro. A partire dal *modello dinamico* di un robot rigido, vengono analizzate le *proprietà energetiche* del sistema e viene introdotto il concetto di *momento generalizzato*. La parte finale del capitolo è dedicata alla *statica* e alla *cinematica differenziale*, definendo le relazioni fondamentali tra *forze generalizzate*, *coppie d'interazione* e *matrici Jacobiane*, essenziali per la localizzazione e la stima delle forze di contatto.

2.1 Modello Dinamico di un Robot

Nel seguito considereremo solo robot rigidi con n gradi di libertà composti da una catena cinematica aperta. Come descritto nel testo di riferimento [1, p. 256, Eq (5.38)], il modello dinamico di un robot può essere descritto in forma compatta dall'equazione di movimento di Lagrange:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_m - \boldsymbol{\tau}_{\text{fric}} + \boldsymbol{\tau}_{\text{int}}, \quad (2.1)$$

dove:

- $\mathbf{q} \in \mathbb{R}^n$ è il vettore delle coordinate generalizzate;
- $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ sono rispettivamente la derivata prima e seconda di \mathbf{q} rispetto al tempo;
- $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ è la matrice d'inerzia del robot;
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ è il vettore delle coppie centrifughe e di Coriolis fattorizzato con la matrice $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ di simboli di Christoffel [1, p. 259, Eq. (5.50)];
- $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ è il vettore delle coppie gravitazionali;
- $\boldsymbol{\tau}_m \in \mathbb{R}^n$ è il vettore delle coppie prodotte dagli attuatori;
- $\boldsymbol{\tau}_{\text{fric}} \in \mathbb{R}^n$ è il vettore delle coppie dissipative dovute all'attrito nelle articolazioni;

- $\boldsymbol{\tau}_{\text{int}} \in \mathbb{R}^n$ è il vettore delle coppie generate dall'interazione con l'ambiente esterno.

Nei successivi capitoli assumeremo che il modello dinamico del robot sia noto; che le coppie di attrito $\boldsymbol{\tau}_{\text{fric}}$ siano trascurabili o compensate e che le coppie applicate dagli attuatori $\boldsymbol{\tau}_m$ corrispondano a quelle comandate. Inoltre, supporremo per semplicità che vi sia al massimo un solo link coinvolto in una collisione in un dato istante di tempo.

2.2 Energia Meccanica di un Robot

L'*energia meccanica* E di un robot è data dalla somma dell'*energia cinetica* T [1, p. 252, Eq. (5.23)] e dell'*energia potenziale* U [1, p. 254, Eq. (5.29)] dovuta alla gravità:

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}; \quad (2.2)$$

$$U(\mathbf{q}) = - \sum_{i=1}^n m_i \mathbf{g}^T \mathbf{p}_{\text{cm},i} + U_0; \quad (2.3)$$

$$E = T + U = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}} + U(\mathbf{q}), \quad (2.4)$$

dove m_i è la massa del link i -esimo; $\mathbf{g} \in \mathbb{R}^3$ è il vettore di gravità; $\mathbf{p}_{\text{cm},i} \in \mathbb{R}^3$ è la posizione del centro di massa del link i -esimo espressa nel sistema di riferimento base del robot e U_0 è una costante arbitraria. È importante notare che $\mathbf{p}_{\text{cm},i}$ dipende dalla configurazione corrente del robot \mathbf{q} .

Essendo le coppie non conservative, possiamo affermare che la variazione dell'*energia meccanica* del robot è uguale alla somma delle potenze delle coppie che agiscono su di esso [1, p. 260, Eq. (5.56)]:

$$\dot{E} = \dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{int}} + \dot{\mathbf{q}}^T \boldsymbol{\tau}_m - \dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{fric}}. \quad (2.5)$$

2.3 Momento Generalizzato

Per comprendere il concetto di *momento generalizzato* di un sistema robotico, è utile richiamare innanzitutto il suo analogo nel dominio del corpo rigido singolo. Per un corpo rigido, la quantità di moto descrive lo stato inerziale del sistema e si scomponete in due parti: la quantità di moto lineare, $\mathbf{p} = m\mathbf{v}$, legata alla traslazione del centro di massa, e la quantità di moto angolare, $\mathbf{L} = \mathbf{I}\boldsymbol{\omega}$, legata alla sua rotazione. La seconda legge di Newton afferma che la derivata temporale di queste quantità è uguale alla risultante delle forze e dei momenti esterni applicati al corpo.

In un robot, che è un sistema multibody articolato, lo stato inerziale complesso non può essere descritto da un singolo vettore di quantità di moto lineare o angolare. Tuttavia, è possibile definire una grandezza aggregata che svolge un ruolo analogo. Il *momento generalizzato* di un robot, indicato con \mathbf{p} , è un vettore n -dimensionale che condensa l'effetto combinato della quantità di moto di tutti i

link del sistema. Esso è definito come il prodotto tra la matrice d'inerzia, $\mathbf{M}(\mathbf{q})$, e il vettore delle velocità dei giunti, $\dot{\mathbf{q}}$ [1, Sez. 5.7.2]:

$$\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}. \quad (2.6)$$

Proprio come nel caso del corpo singolo, la sua derivata temporale $\dot{\mathbf{p}}$, è direttamente legata alle *forze generalizzate* che agiscono sul sistema. Sfruttando la (2.1) si ha infatti:

$$\begin{aligned} \dot{\mathbf{p}} &= \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} \\ &= \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} + \boldsymbol{\tau}_m - \boldsymbol{\tau}_{\text{fric}} + \boldsymbol{\tau}_{\text{int}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) \\ &= \boldsymbol{\tau}_m - \boldsymbol{\tau}_{\text{fric}} + \boldsymbol{\tau}_{\text{int}} + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}). \end{aligned} \quad (2.7)$$

Nell'ultimo passaggio è stata utilizzata la proprietà di antisimmetria della matrice $\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ [1, p. 259, Eq (5.51)], con la seguente proposizione.

Proposizione 2.1 (Equivalenza della proprietà di Antisimmetria). *Essendo la matrice $\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ antisimmetrica, vale la seguente identità:*

$$\dot{\mathbf{M}}(\mathbf{q}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}}). \quad (2.8)$$

Dimostrazione. Sia \mathbf{S} una matrice antisimmetrica, allora $\mathbf{S} = -\mathbf{S}^T$. Posto $\mathbf{S} = \dot{\mathbf{M}} - 2\mathbf{C}$ allora si ha:

$$\dot{\mathbf{M}} - 2\mathbf{C} = -(\dot{\mathbf{M}} - 2\mathbf{C})^T = -\dot{\mathbf{M}}^T + 2\mathbf{C}^T.$$

Poiché \mathbf{M} è simmetrica anche $\dot{\mathbf{M}}$ lo è, quindi $\dot{\mathbf{M}}^T = \dot{\mathbf{M}}$. Sostituendo otteniamo:

$$\begin{aligned} \dot{\mathbf{M}} - 2\mathbf{C} &= -\dot{\mathbf{M}} + 2\mathbf{C}^T \\ \Leftrightarrow \quad 2\dot{\mathbf{M}} &= 2\mathbf{C} + 2\mathbf{C}^T \\ \Leftrightarrow \quad \dot{\mathbf{M}} &= \mathbf{C} + \mathbf{C}^T. \end{aligned}$$

□

È infine importante notare dalla (2.7) la seguente proprietà:

$$\dot{p}_i = \tau_{m,i} - \tau_{\text{fric},i} + \tau_{\text{int},i} - \frac{1}{2}\dot{\mathbf{q}}^T \frac{\partial \mathbf{M}(\mathbf{q})}{\partial q_i} \dot{\mathbf{q}} - g_i(\mathbf{q}), \quad i = 1, \dots, n. \quad (2.9)$$

Ogni componente del momento generalizzato è quindi influenzata solo dalle rispettive componenti delle coppie non conservative. Questa proprietà di *disaccoppiamento* risalterà nel capitolo successivo quando affronteremo l'*Osservatore basato sul Momento Generalizzato* (Sez. 3.3).

2.4 Matrici Jacobiane e Forze Generalizzate

Sia \mathcal{F}_{ext} la *forza esterna generalizzata* applicata a un *punto di contatto* $\mathbf{p}_c \in \mathbb{R}^3$ del robot. Essa è espressa come:

$$\mathcal{F}_{\text{ext}} = \begin{bmatrix} \mathbf{f}_{\text{ext}} \\ \mathbf{m}_{\text{ext}} \end{bmatrix} \in \mathbb{R}^6, \quad (2.10)$$

dove $\mathbf{f}_{\text{ext}} \in \mathbb{R}^3$ è la forza e $\mathbf{m}_{\text{ext}} \in \mathbb{R}^3$ è il momento torcente, entrambi applicati al punto \mathbf{p}_c . La relazione tra la forza esterna generalizzata \mathcal{F}_{ext} e le coppie d'interazione $\boldsymbol{\tau}_{\text{int}}$, è espressa tramite lo Jacobiano (geometrico) del punto di contatto $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ [1, p. 176, Eq. (3.148)]:

$$\boldsymbol{\tau}_{\text{int}} = \mathbf{J}_c^T(\mathbf{q}) \mathcal{F}_{\text{ext}}. \quad (2.11)$$

Se il sistema non è sovradeterminato ($n \leq 6$), è possibile invertire la relazione ottenendo ¹:

$$\mathcal{F}_{\text{ext}} = (\mathbf{J}_c^T(\mathbf{q}))^\dagger \boldsymbol{\tau}_{\text{int}}. \quad (2.12)$$

In caso contrario ($n > 6$), l'equazione (2.12) è esatta solo se $\boldsymbol{\tau}_{\text{int}} \in \mathcal{R}(\mathbf{J}_c^T(\mathbf{q}))^2$. Se ciò non avviene il sistema non è invertibile e la (2.12) risulta essere una *least square solution* [2, Cap. 3, Sez. 1-2].

Anche per quanto riguarda la *velocità generalizzata* \mathbf{V}_c del punto di contatto \mathbf{p}_c , abbiamo una relazione simile [1, p. 121, Eq. (3.6)]:

$$\mathbf{V}_c = \begin{bmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{bmatrix} = \mathbf{J}_c(\mathbf{q}) \dot{\mathbf{q}} \in \mathbb{R}^6, \quad (2.13)$$

dove $\mathbf{v}_c \in \mathbb{R}^3$ è la velocità lineare e $\boldsymbol{\omega}_c \in \mathbb{R}^3$ è la velocità angolare del punto.

Consideriamo ora il link i_c -esimo sul quale è applicata la forza \mathcal{F}_{ext} e prendiamo in considerazione un frame solidale ad esso (per esempio assegnato tramite il formalismo di Denavit-Hartenberg). Possiamo esprimere la *velocità del frame*, come:

$$\mathbf{V}_{i_c} = \begin{bmatrix} \mathbf{v}_{i_c} \\ \boldsymbol{\omega}_{i_c} \end{bmatrix} = \mathbf{J}_{i_c}(\mathbf{q}) \dot{\mathbf{q}}, \quad (2.14)$$

dove $\mathbf{v}_{i_c} \in \mathbb{R}^3$ è la velocità lineare del frame; $\boldsymbol{\omega}_{i_c} \in \mathbb{R}^3$ è la velocità angolare del frame e $\mathbf{J}_{i_c}(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ è lo Jacobiano del frame solidale al link i_c -esimo. Dalla cinematica del corpo rigido, sappiamo che possiamo esprimere la velocità del punto di contatto \mathbf{p}_c usufruendo della velocità del frame solidale al link. Si ha infatti:

$$\begin{aligned} \mathbf{v}_c &= \mathbf{v}_{i_c} + \boldsymbol{\omega}_{i_c} \times {}^{i_c}\mathbf{p}_c \\ \boldsymbol{\omega}_c &= \boldsymbol{\omega}_{i_c}, \end{aligned} \quad (2.15)$$

¹Con il simbolo \dagger indichiamo la *matrice pseudoinversa di Moore-Penrose* [1, Sez. A.7, 2].

²Con $\mathcal{R}(.)$ indichiamo lo *spazio immagine* di una matrice.

dove ${}^{i_c}\mathbf{p}_c = \mathbf{p}_c - \mathbf{p}_{i_c} \in \mathbb{R}^3$ è il vettore che va dall'origine del frame solidale al link i_c -esimo \mathbf{p}_{i_c} al punto di contatto \mathbf{p}_c .

La (2.15) può essere riscritta in forma matriciale nel seguente modo:

$$\mathbf{V}_c = \begin{bmatrix} \mathbf{I} & -\mathbf{S}({}^{i_c}\mathbf{p}_c) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{i_c} \\ \boldsymbol{\omega}_{i_c} \end{bmatrix} = \mathbf{J}_{c,i_c} \mathbf{V}_{i_c}, \quad (2.16)$$

dove $\mathbf{S}(.) \in \mathbb{R}^{3 \times 3}$ è la matrice antisimmetrica associata al prodotto vettoriale e $\mathbf{J}_{c,i_c} \in \mathbb{R}^{6 \times 6}$ è una *matrice Jacobiana locale* che descrive come la velocità del punto \mathbf{p}_c è legata alla velocità del frame solidale al link i_c sul quale vive. È importante notare che, trattandosi di link rigidi il vettore ${}^{i_c}\mathbf{p}_c$ è costante.

Sostituendo la (2.14) nella (2.16) si ottiene:

$$\mathbf{V}_c = \mathbf{J}_{c,i_c} \mathbf{V}_{i_c} = \mathbf{J}_{c,i_c} \mathbf{J}_{i_c}(\mathbf{q}) \dot{\mathbf{q}}. \quad (2.17)$$

Confrontando la (2.17) con la (2.13) si arriva alla seguente relazione tra i Jacobiani [3, Eq. 60]:

$$\mathbf{J}_c(\mathbf{q}) = \mathbf{J}_{c,i_c} \mathbf{J}_{i_c}(\mathbf{q}). \quad (2.18)$$

Ritornando alle forze, possiamo esprimere una relazione simile alla (2.16) per legare la forza esterna alla sua espressione nel frame solidale al link. Si ha infatti che:

$${}^{i_c}\mathcal{F}_{\text{ext}} = \begin{bmatrix} {}^{i_c}\mathbf{f}_{\text{ext}} \\ {}^{i_c}\mathbf{m}_{\text{ext}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{S}({}^{i_c}\mathbf{p}_c) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{\text{ext}} \\ \mathbf{m}_{\text{ext}} \end{bmatrix} = \mathbf{J}_{c,i_c}^T \mathcal{F}_{\text{ext}}. \quad (2.19)$$

Vediamo ora come facendo uso di queste relazioni, è possibile collegare la forza esterna espressa nel frame solidale al link i_c -esimo con le coppie d'interazione applicate al robot. Sostituendo l'identità (2.12) nella (2.19) si ottiene:

$${}^{i_c}\mathcal{F}_{\text{ext}} = \mathbf{J}_{c,i_c}^T (\mathbf{J}_c^T(\mathbf{q}))^\dagger \boldsymbol{\tau}_{\text{int}}. \quad (2.20)$$

Proposizione 2.2. *Supponendo di avere $n \geq 6$ e che la matrice $\mathbf{J}_c^T(\mathbf{q}) \in \mathbb{R}^{n \times 6}$ abbia rango pieno per colonne, ovvero si è in una configurazione non singolare per il punto di contatto, vale la seguente identità:*

$${}^{i_c}\mathcal{F}_{\text{ext}} = (\mathbf{J}_{i_c}^T(\mathbf{q}))^\dagger \boldsymbol{\tau}_{\text{int}}. \quad (2.21)$$

Dimostrazione. Per dimostrare l'identità, partiamo dalla relazione tra i Jacobiani (2.18):

$$\mathbf{J}_c(\mathbf{q}) = \mathbf{J}_{c,i_c} \mathbf{J}_{i_c}(\mathbf{q}).$$

Essendo $\mathbf{J}_{c,i_c} \in \mathbb{R}^{6 \times 6}$ invertibile possiamo moltiplicare entrambi i membri a sinistra per la sua inversa. Otteniamo quindi:

$$\mathbf{J}_{c,i_c}^{-1} \mathbf{J}_c(\mathbf{q}) = \mathbf{J}_{i_c}(\mathbf{q}).$$

Trasponendo entrambi i membri e pseudoinvertendo si ha:

$$(\mathbf{J}_c^T(\mathbf{q}) (\mathbf{J}_{c,i_c}^{-1})^T)^\dagger = (\mathbf{J}_{i_c}^T(\mathbf{q}))^\dagger.$$

Siccome per ipotesi la matrice $\mathbf{J}_c^T(\mathbf{q})$ ha rango pieno per colonne ed essendo $(\mathbf{J}_{c,i_c}^{-1})^T$ invertibile, possiamo affermare che la matrice $\mathbf{J}_c^T(\mathbf{q})(\mathbf{J}_{c,i_c}^{-1})^T$ è una *fattorizzazione di rango pieno*. In questo caso si può dimostrare che vale la *legge dell'ordine inverso* $(AB)^\dagger = B^\dagger A^\dagger$; si veda [2, p. 42, Teo. 5, Ex. 17]. Di conseguenza si ha:

$$\left(\left(\mathbf{J}_{c,i_c}^{-1} \right)^T \right)^\dagger \left(\mathbf{J}_c^T(\mathbf{q}) \right)^\dagger = \left(\mathbf{J}_{i_c}^T(\mathbf{q}) \right)^\dagger.$$

Sapendo che l'operazione di pseudoinversione commuta con quella di trasposizione [1, Sez. A.7, 2, p. 43, Ex. 18] e che la pseudoinversa di una matrice quadrata invertibile coincide con la sua inversa, otteniamo:

$$\mathbf{J}_{c,i_c}^T \left(\mathbf{J}_c^T(\mathbf{q}) \right)^\dagger = \left(\mathbf{J}_{i_c}^T(\mathbf{q}) \right)^\dagger.$$

Sostituendo nella (2.20) otteniamo l'identità (2.21). \square

Nel capitolo successivo vedremo come questa identità sarà utile per determinare il punto di applicazione della forza esterna.

Capitolo 3

Osservatori Basati su Modello per il Rilevamento, l’Isolamento e l’Identificazione di Collisioni

Il presente capitolo entra nel vivo della trattazione analizzando tre diverse strategie per il monitoraggio delle collisioni basate sul modello dinamico: l’*Osservatore basato sull’Energia*, l’*Osservatore basato sulla Velocità* e l’*Osservatore basato sul Momento*. Per ciascun metodo vengono derivate le equazioni del *residuo* e ne viene studiata la *dinamica* evidenziandone le proprietà. Infine, viene discusso come utilizzare tali *segnali residuali* per implementare le fasi di *Rilevamento*, *Isolamento* e *Identificazione* in una *pipeline di gestione delle collisioni*.

3.1 L’Osservatore Basato sull’Energia

L’idea di base dietro questo metodo, proposto in [3, Sez. III-A], è monitorare l’*energia meccanica* del robot. Conoscendo la traiettoria desiderata, possiamo calcolare l’energia nominale che il robot dovrebbe avere in ogni istante di tempo. Intuitivamente una collisione comporterà una variazione improvvisa dell’energia meccanica. Quest’ultima può essere rilevata confrontando l’energia attuale con quella nominale della traiettoria.

Partiamo definendo il seguente *segnale residuale* scalare:

$$r(t) = k_O \left(\hat{E}(t) + \int_0^t (\dot{\boldsymbol{q}}^T \boldsymbol{\tau}_m + r) ds - \hat{E}(0) \right), \quad (3.1)$$

dove $r(0) = 0$; $\hat{E}(t)$ è una misura dell’energia meccanica del robot (2.4) mentre $k_O > 0$ è un guadagno. Il residuo $r(t)$ può essere calcolato in tempo reale utilizzando le coppie comandate agli attuatori $\boldsymbol{\tau}_m$; una misura della configurazione corrente del robot \boldsymbol{q} ottenuta tramite encoder e una misura della velocità $\dot{\boldsymbol{q}}$ la quale può essere ottenuta anche tramite derivazione numerica.

Nel caso ideale in cui l’energia meccanica del robot sia calcolata senza errori, ovvero

$\hat{E}(t) = E(t)$, sfruttando la (2.5) e assumendo $\tau_{\text{fric}} = 0$ si ha:

$$\dot{r}(t) = k_O \left(\dot{E}(t) - (\dot{\mathbf{q}}^T \boldsymbol{\tau}_m + r(t)) \right) = k_O \left(\dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{int}} - r(t) \right), \quad (3.2)$$

dove $P_{\text{ext}} = \dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{int}}$ è la potenza delle forze esterne applicate al robot.

Possiamo notare che il sistema risulta essere un *filtro passa basso* con ingresso la potenza delle forze esterne P_{ext} e uscita il residuo $r(t)$. Infatti passando la (3.2) nel dominio di Laplace si ottiene:

$$\begin{aligned} sR(s) &= k_O (P_{\text{ext}}(s) - R(s)) \\ \Leftrightarrow (s + k_O) R(s) &= k_O P_{\text{ext}}(s) \\ \Leftrightarrow R(s) &= \frac{k_O}{(s + k_O)} P_{\text{ext}}(s), \end{aligned} \quad (3.3)$$

da cui si ricava la seguente funzione di trasferimento:

$$H(s) = \frac{R(s)}{P_{\text{ext}}(s)} = \frac{k_O}{s + k_O} = \frac{1}{1 + \frac{1}{k_O}s}. \quad (3.4)$$

Il residuo $r(t)$ converge quindi a P_{ext} con una costante di tempo $\tau = \frac{1}{k_O}$.

Purtroppo con questo metodo non è possibile rilevare in ogni caso le collisioni. Se il robot è in quiete si ha che $\dot{\mathbf{q}} = 0$, di conseguenza una collisione istantanea $\boldsymbol{\tau}_{\text{int}}$ non avrà effetti su $r(t)$ fin quando il robot non inizia a muoversi, per via della collisione o non. Anche nel caso opposto $\dot{\mathbf{q}} \neq 0$, la collisione non viene rilevata a meno che la forza non produca un movimento del robot. Infatti dalla (2.11) e la (2.13) si ha:

$$P_{\text{ext}} = \dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{int}} = \dot{\mathbf{q}}^T \mathbf{J}_c^T(\mathbf{q}) \mathcal{F}_{\text{ext}} = \mathbf{V}_c^T \mathcal{F}_{\text{ext}} = 0 \Leftrightarrow \mathbf{V}_c \perp \mathcal{F}_{\text{ext}}. \quad (3.5)$$

Alla fine di questo capitolo, vedremo che essendo il residuo una grandezza scalare ed essendo una versione filtrata della potenza delle forze esterne, non sarà possibile né distinguere quale link del robot è soggetto alla collisione né ricostruirne la forza in gioco.

3.2 L'Osservatore Basato sulla Velocità

Il secondo metodo proposto, discusso in [3, Sez. III-D], si basa sulla seguente intuizione: una collisione causerà un'accelerazione improvvisa del robot. Se si riesce a stimare la velocità del robot in assenza di collisioni, è possibile rilevare un impatto confrontando la velocità stimata con quella misurata. Inoltre basandosi sulla differenza tra le due velocità, è possibile stimare le coppie d'interazione $\boldsymbol{\tau}_{\text{int}}$. Per la stima della velocità possiamo utilizzare un *osservatore di stato a dimensione ridotta* basato sul modello dinamico del robot. L'osservatore proposto ha infatti dimensione n (numero di gradi di libertà del robot) invece di $2n$ come un classico osservatore.

Per prima cosa definiamo per compattezza la quantità $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$:

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) := \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}). \quad (3.6)$$

Riscriviamo poi la dinamica del robot (2.1) isolando il termine \ddot{q} :

$$\ddot{q} = M^{-1}(q)(\tau_m - n(q, \dot{q}) + \tau_{int}). \quad (3.7)$$

Siccome non abbiamo informazioni sulle coppie d'interazione τ_{int} , la migliore ipotesi che possiamo fare per ricostruirle è considerarle costanti nel tempo:

$$\hat{\tau}_{int} = r \in \mathbb{R}^n, \quad \dot{r} = 0. \quad (3.8)$$

Utilizzando questa ipotesi, possiamo definire il seguente *osservatore di stato a dimensione ridotta*:

$$\begin{aligned} \hat{q} &= \hat{M}^{-1}(q)(\tau_m - \hat{n}(q, \dot{q}) + r) \\ \dot{r} &= K_O(\ddot{q} - \hat{q}), \end{aligned} \quad (3.9)$$

dove $K_O = \text{diag}\{k_{O,i}\} > 0$ è la *matrice dei guadagni dell'osservatore*. La Fig. 3.1 ne mostra lo schema a blocchi.

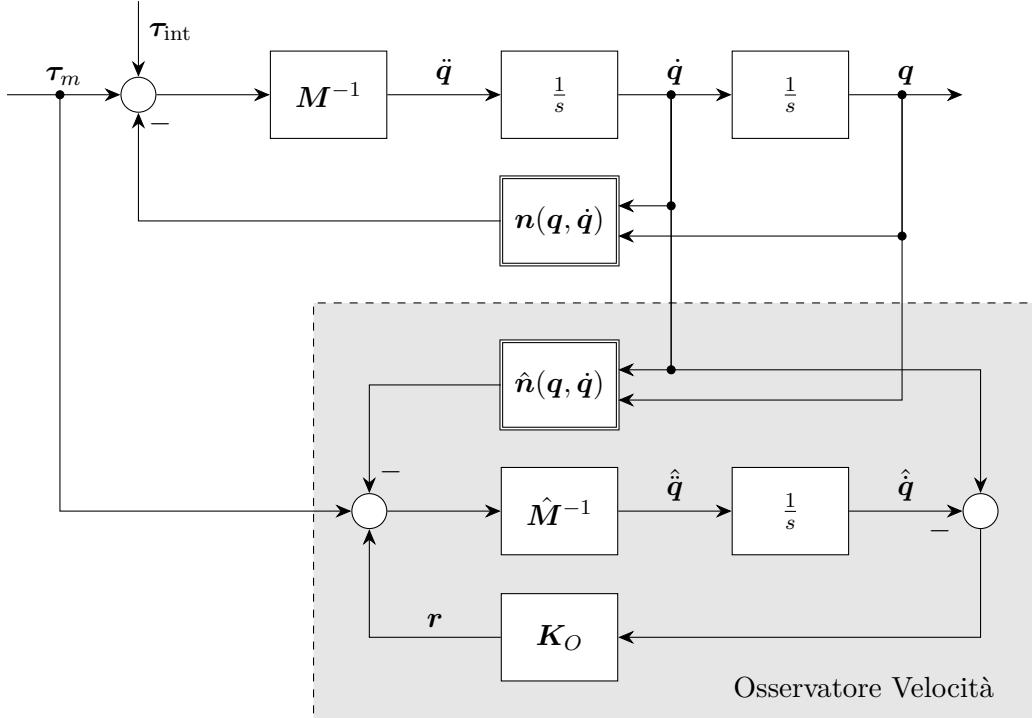


Figura 3.1. Schema a blocchi dell'osservatore basato sulla Velocità (3.9).

Dall'equazione (3.9) segue la seguente espressione per il residuo:

$$\begin{aligned} r(t) &= \int_0^t K_O(\ddot{q} - \hat{q}) ds \\ &= K_O \left(\dot{q}(t) - \dot{q}(0) - \int_0^t \hat{M}^{-1}(q)(\tau_m - \hat{n}(q, \dot{q}) + r) ds \right). \end{aligned} \quad (3.10)$$

Sostituendo l'identità (3.7) nella (3.9) si ottiene la seguente dinamica del residuo:

$$\begin{aligned} \dot{r}(t) &= K_O \left(\ddot{q}(t) - \hat{M}^{-1}(q)(\tau_m - \hat{n}(q, \dot{q}) + r) \right) \\ &= K_O \left(M^{-1}(q)(\tau_m - n(q, \dot{q}) + \tau_{int}) - \hat{M}^{-1}(q)(\tau_m - \hat{n}(q, \dot{q}) + r) \right). \end{aligned} \quad (3.11)$$

Nel caso ideale in cui si ha un modello del robot perfetto, ovvero $\hat{\mathbf{M}}(\mathbf{q}) = \mathbf{M}(\mathbf{q})$ e $\hat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})$, la (3.11) si riduce a:

$$\dot{\mathbf{r}} = \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) (\boldsymbol{\tau}_{\text{int}} - \mathbf{r}). \quad (3.12)$$

Passando la (3.12) nel dominio di Laplace si ottiene:

$$\begin{aligned} s\mathbf{R}(s) &= \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) (\boldsymbol{\tau}_{\text{int}}(s) - \mathbf{R}(s)) \\ \Leftrightarrow \quad \left(s\mathbf{I} + \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) \right) \mathbf{R}(s) &= \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) \boldsymbol{\tau}_{\text{int}}(s) \\ \Leftrightarrow \quad \mathbf{R}(s) &= \left(s\mathbf{I} + \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) \right)^{-1} \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) \boldsymbol{\tau}_{\text{int}}(s), \end{aligned} \quad (3.13)$$

da cui si ricava la seguente funzione di trasferimento matriciale:

$$\mathbf{H}(s) = \left(s\mathbf{I} + \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) \right)^{-1} \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}). \quad (3.14)$$

Dalla (3.14) possiamo affermare che il residuo $\mathbf{r}(t)$ risulta essere una versione filtrata delle coppie d'interazione $\boldsymbol{\tau}_{\text{int}}$. A causa delle non linearità di $\mathbf{M}(\mathbf{q})$ ed essendo $\mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q})$ una matrice non diagonale, si ha che la *dinamica del residuo* risulta essere *non lineare* e *accoppiata* tra le varie componenti. Nonostante ciò, possiamo affermare che il sistema rappresentato dalla (3.12) risulta essere asintoticamente stabile; ciò deriva dal fatto che le matrici \mathbf{K}_O e $\mathbf{M}(\mathbf{q})$ sono definite positive [1, p. 253, Eq. (5.26)]. Per via dell'accoppiamento ed essendo la dinamica *dipendente dalla configurazione* \mathbf{q} , la scelta dei guadagni $k_{O,i}$ risulta molto complicata. A differenza dell'*Osservatore basato sull'Energia*, in questo caso si ha un residuo vettoriale; vedremo che conoscendo il punto di contatto, ciò ci permetterà di ricostruire la forza generalizzata applicata al robot.

3.3 L'Osservatore Basato sul Momento Generalizzato

Nel capitolo precedente abbiamo introdotto il concetto di *momento generalizzato* (Sez. 2.3) e abbiamo visto come la sua variazione è influenzata dalle coppie che agiscono sul robot (2.7). Seguendo l'approccio descritto in [3, Sez. III-E], l'idea di base è quella di stimare il momento generalizzato in assenza di collisioni e confrontarlo con la misura reale per rilevare eventuali impatti.

Per compattezza definiamo la seguente quantità:

$$\begin{aligned} \beta(\mathbf{q}, \dot{\mathbf{q}}) &:= \mathbf{g}(\mathbf{q}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} \\ &= \mathbf{g}(\mathbf{q}) - \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}, \end{aligned} \quad (3.15)$$

dove nell'ultima riga abbiamo sfruttato la proprietà di antisimmetria (2.8).

Basandoci sulla dinamica del momento generalizzato (2.7) ed effettuando la medesima ipotesi sulle coppie d'interazione vista in precedenza (3.8); possiamo definire il seguente *osservatore di stato a dimensione ridotta*:

$$\begin{aligned} \dot{\hat{\mathbf{p}}} &= \boldsymbol{\tau}_m - \hat{\beta}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{r} \\ \dot{\mathbf{r}} &= \mathbf{K}_O (\dot{\mathbf{p}} - \hat{\mathbf{p}}), \end{aligned} \quad (3.16)$$

dove $\mathbf{K}_O = \text{diag} \{ k_{O,i} \} > 0$ è la *matrice dei guadagni dell'osservatore*. Integrando $\dot{\mathbf{r}}$ dalla (3.16), otteniamo la seguente espressione per il residuo:

$$\begin{aligned}\mathbf{r}(t) &= \int_0^t \mathbf{K}_O (\dot{\mathbf{p}} - \dot{\hat{\mathbf{p}}}) \, ds \\ &= \mathbf{K}_O \left(\mathbf{p}(t) - \mathbf{p}(0) - \int_0^t (\boldsymbol{\tau}_m - \hat{\beta}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{r}) \, ds \right).\end{aligned}\quad (3.17)$$

È importante notare che l'accuratezza del momento misurato dipenderà da quella del modello; si ha infatti $\mathbf{p} = \hat{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}}$. La Fig. 3.2 mostra lo schema a blocchi dell'osservatore e del robot.

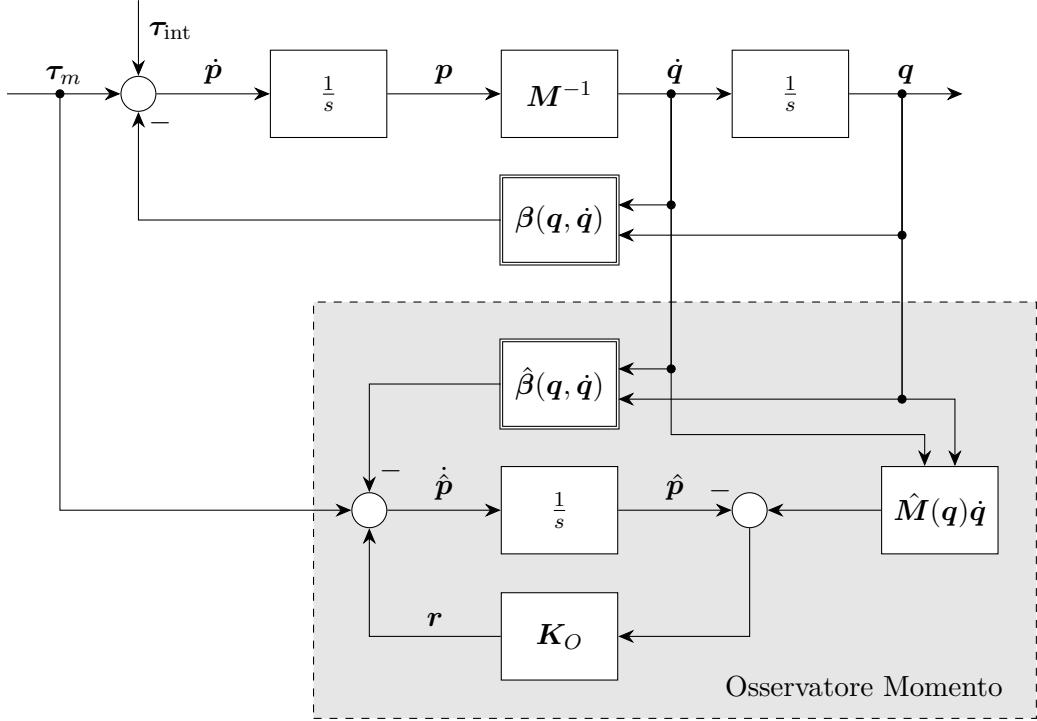


Figura 3.2. Schema a blocchi dell'osservatore basato sul Momento Generalizzato (3.16).

In condizioni ideali in cui si ha a disposizione un modello perfetto, ovvero $\hat{\mathbf{M}} = \mathbf{M}$ e $\hat{\beta} = \beta$, possiamo esplicitare la dinamica del residuo usando la (2.7) nella (3.16). Otteniamo quindi:

$$\begin{aligned}\dot{\mathbf{r}}(t) &= \mathbf{K}_O (\dot{\mathbf{p}} - \dot{\hat{\mathbf{p}}}) \\ &= \mathbf{K}_O (\boldsymbol{\tau}_m + \boldsymbol{\tau}_{\text{int}} - \beta(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\tau}_m + \hat{\beta}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{r}) \\ &= \mathbf{K}_O (\boldsymbol{\tau}_{\text{int}} - \mathbf{r}).\end{aligned}\quad (3.18)$$

Essendo \mathbf{K}_O una matrice diagonale, si ha per ogni componente la seguente equazione scalare:

$$\dot{r}_i(t) = k_{O,i} (\boldsymbol{\tau}_{\text{int},i} - r_i), \quad i = 1, \dots, n. \quad (3.19)$$

Passando nel dominio di Laplace la (3.19) otteniamo:

$$\begin{aligned} sR_i(s) &= k_{O,i}(\tau_{\text{int},i}(s) - R_i(s)) \\ \Leftrightarrow R_i(s)(s + k_{O,i}) &= k_{O,i}\tau_{\text{int},i}(s) \\ \Leftrightarrow R_i(s) &= \frac{k_{O,i}}{s + k_{O,i}}\tau_{\text{int},i}(s), \quad i = 1, \dots, n, \end{aligned} \quad (3.20)$$

da cui si ricava la seguente funzione di trasferimento scalare per ogni componente:

$$H_i(s) = \frac{R_i(s)}{\tau_{\text{int},i}(s)} = \frac{k_{O,i}}{s + k_{O,i}} = \frac{1}{1 + \frac{1}{k_{O,i}}s}, \quad i = 1, \dots, n. \quad (3.21)$$

La (3.21) rappresenta un *filtro passa basso* con costante di tempo $\tau_i = \frac{1}{k_{O,i}}$.

Si ha quindi una *dinamica disaccoppiata* in cui ogni componente del residuo $\mathbf{r}(t)$ converge alla rispettiva componente della coppia d'interazione $\boldsymbol{\tau}_{\text{int}}$. Tutto ciò accade con costanti di tempo che possono essere scelte in maniera *indipendente* l'una dall'altra. A differenza dell'*Osservatore basato sulla Velocità*, vedremo che la dinamica disaccoppiata ci permetterà di distinguere quale dei link è soggetto alla collisione. Potremmo poi ricostruire la forza d'interazione conoscendo il punto di contatto.

3.4 Rilevamento, Isolamento e Identificazione

In questa sezione, analizzeremo come sfruttare i metodi proposti per le fasi di *Rilevamento, Isolamento e Identificazione* delle collisioni, facendo riferimento alla trattazione in [3, Sez. IV].

3.4.1 Rilevamento

Il *Rilevamento* di una collisione consiste nel determinare se una collisione fisica sia presente o meno, riducendo al minimo l'occorrenza di falsi positivi o negativi. L'idea è quindi di creare una *funzione di rilevamento delle collisioni* $rc(\cdot)$ che mappi il segnale di monitoraggio prodotto dall'osservatore $\mathbf{r}(t)$ in uno dei due possibili stati *VERO* o *FALSO*

$$rc : \mathbf{r}(t) \longrightarrow \{ \text{VERO}, \text{FALSO} \}. \quad (3.22)$$

Idealmente si potrebbe effettuare la seguente classificazione:

$$rc(\mathbf{r}(t)) = \begin{cases} \text{VERO}, & \text{if } \mathbf{r}(t) \neq \mathbf{0} \\ \text{FALSO}, & \text{if } \mathbf{r}(t) = \mathbf{0}. \end{cases} \quad (3.23)$$

Purtroppo nella pratica bisogna tenere in considerazioni le seguenti non idealità:

- La coppia erogata dai motori $\boldsymbol{\tau}_m$ non è sempre uguale a quella comandata. Anche se misurata tramite sensori è comunque possibile ci sia del rumore sulle misure;
- La misura di posizione \mathbf{q} e velocità $\dot{\mathbf{q}}$ può essere soggetta a del rumore;
- Anche se piccoli, ci saranno errori di modellazione della dinamica del robot $\hat{\mathbf{D}} := \{\hat{M}(\mathbf{q}), \hat{C}(\mathbf{q}, \dot{\mathbf{q}}), \hat{g}(\mathbf{q})\}$ rispetto la dinamica reale \mathbf{D} ;

- Le coppie dissipative $\boldsymbol{\tau}_{\text{fric}}$ in generale non sono trascurabili.

Possiamo modellare tutte queste non idealità come un termine di disturbo additivo $\boldsymbol{\tau}_d(t, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}_m, \hat{\mathbf{D}})$ che si somma alle coppie d'interazione $\boldsymbol{\tau}_{\text{int}}(t)$. Riprendendo le dinamiche dei tre residui nelle condizioni ideali (3.2)(3.12)(3.18), nel caso di non collisione $\boldsymbol{\tau}_{\text{int}} = \mathbf{0}$ si ha:

$$\begin{aligned}\dot{\mathbf{r}} &= k_O \left(\dot{\mathbf{q}}^T \boldsymbol{\tau}_d - \mathbf{r} \right); \\ \dot{\mathbf{r}} &= \mathbf{K}_O \mathbf{M}^{-1}(\mathbf{q}) (\boldsymbol{\tau}_d - \mathbf{r}); \\ \dot{\mathbf{r}} &= \mathbf{K}_O (\boldsymbol{\tau}_d - \mathbf{r}).\end{aligned}\quad (3.24)$$

Grazie alla stabilità dei tre sistemi, possiamo affermare che per un ingresso $\boldsymbol{\tau}_d$ limitato verrà prodotta un'uscita \mathbf{r} limitata. In funzione di ciò è possibile definire una *soglia di tolleranza* ε_r . Un possibile approccio consiste nel definire la soglia in funzione dei valori massimi assunti dal residuo in un dato intervallo di tempo $[t - \Delta t, t]$, chiamato *intervallo d'osservazione in assenza di collisioni*. Ciò richiede la costruzione di un vettore \mathbf{r}_{\max} definito componente per componente nel seguente modo:

$$r_{\max,i} = \max_{\tau \in [t - \Delta t, t]} |r_i(\tau)|. \quad (3.25)$$

È possibile poi definire la soglia in funzione di \mathbf{r}_{\max} come:

$$\varepsilon_r = \mathbf{r}_{\max} + \varepsilon_{\text{safe}}, \quad (3.26)$$

dove $\varepsilon_{\text{safe}} > 0$ è un *margine di sicurezza* scelto per irrobustire. Possiamo quindi ridefinire la *funzione di rilevamento delle collisioni*¹:

$$rc(\mathbf{r}(t)) = \begin{cases} VERO, & \text{if } |\mathbf{r}(t)| > \varepsilon_r \\ FALSO, & \text{if } |\mathbf{r}(t)| \leq \varepsilon_r. \end{cases} \quad (3.27)$$

Questo schema di soglie limita in maniera robusta l'occorrenza di falsi positivi; allo stesso tempo può produrre dei falsi negativi per contatti lievi.

Tutti e tre i metodi proposti sono in grado di rilevare collisioni con buone prestazioni, tuttavia bisogna sottolineare alcune differenze sostanziali:

- L'*Osservatore basato sull'Energia* non rileva collisioni a velocità nulla ($\dot{\mathbf{q}} = \mathbf{0}$) e per forze ortogonali al moto ($\mathbf{V}_c \perp \mathcal{F}_{\text{ext}}$), come visto nella Sez. 3.1;
- Nell'*Osservatore basato sulla Velocità*, a causa dell'accoppiamento tramite $\mathbf{M}^{-1}(\mathbf{q})$ la "rumorosità" del residuo in condizioni di non contatto dipende dalla configurazione \mathbf{q} del robot. Come conseguenza, una possibile scelta di Δt e $\varepsilon_{\text{safe}}$ per la definizione della soglia può non essere ottimale in tutto lo spazio di lavoro.

¹La diseguaglianza tra vettori, come in $|\mathbf{r}(t)| > \varepsilon_r$, è da intendersi eseguita componente per componente. La condizione è verificata se anche una sola delle diseguaglianze scalari $|r_i(t)| > \varepsilon_{r,i}$ è vera.

3.4.2 Isolamento

Risolvere il problema dell'*Isolamento* consiste nel determinare quale link i_c del robot è entrato in collisione e il punto di contatto \mathbf{p}_c associato. Assumeremo che avvenga una singola collisione alla volta. Per semplificare la notazione considereremo $\boldsymbol{\varepsilon}_r = \mathbf{0}$.

Supponiamo che la collisione avvenga in un punto \mathbf{p}_c posto sul link i_c -esimo. Una diretta conseguenza della cinematica di un manipolatore a catena aperta è che: le ultime $n - i_c$ colonne della matrice Jacobiana del punto di contatto $\mathbf{J}_c(\mathbf{q})$ sono identicamente nulle. Si ha infatti che la velocità di un qualsiasi punto posto su un link i_c è influenzata unicamente dalle velocità dei giunti che precedono il link nella catena cinematica [1, Sez. 3.2.3]. Questa proprietà si riflette in base alla (2.11) sulle componenti del vettore $\boldsymbol{\tau}_{\text{int}}$. Si avrà infatti che le ultime $n - i_c$ componenti saranno nulle:

$$\boldsymbol{\tau}_{\text{int}} = [\tau_{\text{int},1} \quad \dots \quad \tau_{\text{int},i_c} \quad 0 \quad \dots \quad 0]^T. \quad (3.28)$$

Prendendo in considerazione l'*Osservatore basato sul Momento*, possiamo affermare grazie alla sua proprietà di disaccoppiamento (3.19) che anche le ultime $n - i_c$ componenti del residuo $\mathbf{r}(t)$ saranno nulle:

$$\mathbf{r} = [r_1 \quad \dots \quad r_{i_c} \quad 0 \quad \dots \quad 0]^T. \quad (3.29)$$

Le prime i_c componenti di quest'ultimo, risulteranno in generale diverse da zero almeno durante la collisione e inizieranno esponenzialmente a convergere a zero non appena il contatto verrà perso. È possibile quindi risalire al link che ha subito la collisione con il seguente metodo:

$$i_c = \max \{i \in \{1 \dots n\} : r_i \neq 0\}. \quad (3.30)$$

Lo stesso metodo può essere utilizzato anche per l'*Osservatore basato sulla Velocità*, bisogna però tenere in considerazione le non linearità e l'accoppiamento del residuo. Durante il regime transitorio infatti, non si ha alcuna sicurezza su quali componenti di quest'ultimo saranno diverse da zero. Va quindi trovata una strategia diversa che tenga in considerazione di ciò. Per quanto riguarda l'*Osservatore basato sull'Energia*, non si può estrarre alcuna informazione utile all'isolamento a causa della natura scalare del residuo.

Se si vuole isolare anche l'area di contatto, ovvero stimare il punto \mathbf{p}_c in cui è applicata la forza, possiamo sfruttare le relazioni viste nel capitolo precedente (Sez. 2.4). Una volta identificato il link i_c che ha subito la collisione, possiamo provare a ricostruire la forza espressa nel frame solidale al link tramite lo Jacobiano associato ad esso $\mathbf{J}_{i_c}(\mathbf{q})$. Se vengono rispettate le ipotesi della Prop. 2.2, possiamo scrivere a partire dall'equazione (2.21):

$${}^{i_c} \hat{\mathcal{F}}_{\text{ext}} = \begin{bmatrix} {}^{i_c} \hat{\mathbf{f}}_{\text{ext}} \\ {}^{i_c} \hat{\mathbf{m}}_{\text{ext}} \end{bmatrix} = (\mathbf{J}_{i_c}^T(\mathbf{q}))^\dagger \mathbf{r}. \quad (3.31)$$

È importante notare che per non avere una perdita d'informazioni nella ricostruzione della forza, la matrice $\mathbf{J}_{i_c}(\mathbf{q})$ deve avere rango completo per righe. Questo richiede

necessariamente che $i_c \geq 6$. Considerando ora l'equazione (2.19) otteniamo il seguente sistema di equazioni:

$$\begin{cases} {}^{i_c} \mathbf{f}_{\text{ext}} = \mathbf{f}_{\text{ext}} \\ {}^{i_c} \mathbf{m}_{\text{ext}} = \mathbf{S}({}^{i_c} \mathbf{p}_c) \mathbf{f}_{\text{ext}} + \mathbf{m}_{\text{ext}}. \end{cases} \quad (3.32)$$

Sostituendo la prima equazione nella seconda si ottiene:

$$\begin{aligned} {}^{i_c} \mathbf{m}_{\text{ext}} &= \mathbf{S}({}^{i_c} \mathbf{p}_c) {}^{i_c} \mathbf{f}_{\text{ext}} + \mathbf{m}_{\text{ext}} \\ &= \mathbf{S}^T({}^{i_c} \mathbf{f}_{\text{ext}}) {}^{i_c} \mathbf{p}_c + \mathbf{m}_{\text{ext}}. \end{aligned} \quad (3.33)$$

Siccome non conosciamo \mathbf{m}_{ext} non possiamo risolvere il sistema per l'incognita ${}^{i_c} \mathbf{p}_c$. Per procedere bisogna supporre che la forza di contatto non generi momenti, ovvero $\mathbf{m}_{\text{ext}} = \mathbf{0}$. L'ipotesi non è restrittiva in quanto rispecchia la situazione che si verifica maggiormente nella pratica, sia per collisioni improvvise che contatti volontari. Possiamo quindi riscrivere la precedente equazione considerando le quantità ricavate dal residuo (3.31):

$$\mathbf{S}^T({}^{i_c} \hat{\mathbf{f}}_{\text{ext}}) {}^{i_c} \mathbf{p}_c = {}^{i_c} \hat{\mathbf{m}}_{\text{ext}}. \quad (3.34)$$

La matrice $\mathbf{S}^T({}^{i_c} \hat{\mathbf{f}}_{\text{ext}})$ è una matrice singolare che ha rango due fin quando ci troviamo nel caso d'interesse ${}^{i_c} \hat{\mathbf{f}}_{\text{ext}} \neq \mathbf{0}$. Il sistema ammetterà quindi ∞^1 soluzioni. Esse descrivono geometricamente una retta nello spazio tridimensionale pari alla *linea d'azione della forza* (Fig. 3.3). È possibile trovare il punto della retta più vicino all'origine del frame solidale al link ricavando la *least square solution* del sistema [2, Cap. 3, Sez. 1-2]:

$${}^{i_c} \mathbf{p}_{\min} = (\mathbf{S}^T({}^{i_c} \hat{\mathbf{f}}_{\text{ext}}))^{\dagger} {}^{i_c} \hat{\mathbf{m}}_{\text{ext}}. \quad (3.35)$$

La pseudoinversa può essere calcolata numericamente mediante la *decomposizione a valori singolari* [2, Sez. 2 Cap. 6, Cap. 7, 1, Sez. A.8]. Una volta trovato il punto a distanza minima, possiamo esprimere gli altri punti della retta nel seguente modo:

$${}^{i_c} \mathbf{p}_c(\lambda) = {}^{i_c} \mathbf{p}_{\min} + \lambda \frac{{}^{i_c} \hat{\mathbf{f}}_{\text{ext}}}{\| {}^{i_c} \hat{\mathbf{f}}_{\text{ext}} \|}, \quad \lambda \in \mathbb{R}. \quad (3.36)$$

È importante notare che finora non siamo ancora riusciti ad isolare il punto di contatto. Per fare ciò dobbiamo assumere di conoscere la superficie del link S_{i_c} . Assumendo che i link siano corpi convessi, possiamo isolare due possibili punti d'applicazione. Essi saranno l'intersezione tra la retta d'azione della forza e la superficie S_{i_c} . Corrisponderanno quindi ai punti in cui la forza "tira" o "spinge" il robot. Infine è possibile ricavare $\mathbf{p}_c = {}^{i_c} \mathbf{p}_c + \mathbf{p}_{i_c}$, dove \mathbf{p}_{i_c} è la posizione del frame associato al link. La conoscenza di quest'ultimo ci permette di calcolare lo Jacobiano del punto di contatto $\mathbf{J}_c(\mathbf{q})$.

Il metodo d'Isolamento proposto può essere utilizzato sia per l'*Osservatore basato sul Momento* che per quello basato sulla *Velocità*. Nel secondo caso bisogna sempre tenere in considerazione le problematiche relative la dinamica del residuo discusse in precedenza.

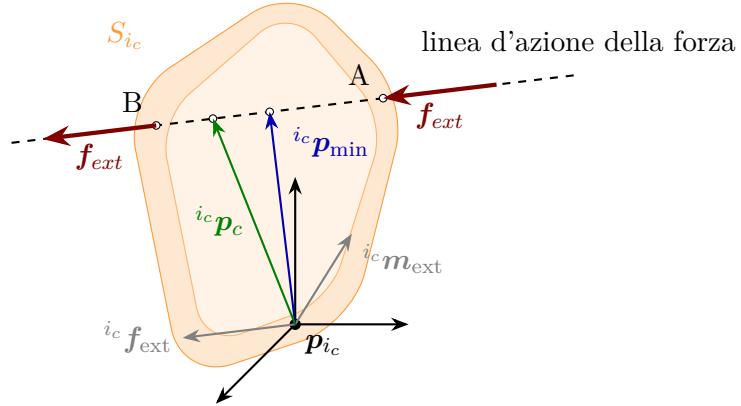


Figura 3.3. Rappresentazione grafica dell’isolamento del punto di contatto p_c tramite la retta d’azione della forza esterna f_{ext} e il punto di minima distanza $i_c p_{\text{min}}$. I punti A e B rappresentano l’intersezione tra la retta e la superficie del link S_{i_c} .

3.4.3 Identificazione

Risolvere il problema dell’*Identificazione* consiste nel determinare le coppie d’interazione τ_{int} applicate al robot e la forza esterna \mathcal{F}_{ext} che le sta generando. Abbiamo visto che gli osservatori basati sulla *Velocità* e sul *Momento* riescono a stimare senza problemi le coppie d’interazione. Per riguarda il problema di ricostruzione della forza, dobbiamo necessariamente conoscere lo Jacobiano del punto di contatto $J_c(\mathbf{q})$. Una volta nota la matrice, possiamo provare a ricostruire la forza a partire dall’equazione (2.12):

$$\hat{\mathcal{F}}_{\text{ext}} = \left(J_c^T(\mathbf{q}) \right)^\dagger \mathbf{r}(t). \quad (3.37)$$

L’equazione (3.37) fornisce una ricostruzione completa della forza di contatto solo se la matrice $J_c(\mathbf{q})$ ha rango completo per righe. In questo caso si ha la seguente espressione per la pseudoinversa [1, Sez. A.7]:

$$\left(J_c^T(\mathbf{q}) \right)^\dagger = \left(J_c(\mathbf{q}) J_c^T(\mathbf{q}) \right)^{-1} J_c(\mathbf{q}). \quad (3.38)$$

In caso contrario il robot si troverà in una configurazione singolare rispetto al punto di contatto; forze e momenti applicati in certe direzioni non saranno rilevati dall’osservatore in quanto non produrranno coppie. Possiamo conoscere lo Jacobiano del punto di contatto $J_c(\mathbf{q})$ in tre principali casi:

- Se il punto di contatto è noto a priori, come nel caso di uno strumento attaccato al robot;
- Se il punto di contatto è noto tramite sensori esterocettivi, per esempio di tipo visivo o tattile;
- Se il punto di contatto è stimato tramite l’algoritmo d’isolamento visto nella Sez. 3.4.2.

Capitolo 4

Analisi Sperimentale su Manipolatore Seriale

In questo capitolo vengono presentati i risultati sperimentali ottenuti in ambiente di simulazione per validare i metodi teorici descritti in precedenza. Vengono messi a confronto gli osservatori implementati in ambiente MATLAB [4] e Simulink [5] su un robot KUKA LBR iiwa 7 analizzandone il comportamento in scenari di collisione sia statici che dinamici. Successivamente, viene validata un'implementazione C++ dell'*Osservatore basato sul Momento* (App.A) tramite il simulatore fisico MuJoCo [6] e la libreria Pinocchio [7, 8], verificando le capacità di *Rilevamento*, *Isolamento* e *Identificazione* su un KUKA LBR iiwa 14. È possibile scaricare il codice dai repository GitHub presenti in App. B.

4.1 Confronto delle Prestazioni (MATLAB & Simulink)

Nei test di questa sezione, l'*Osservatore basato sull'Energia* è stato realizzato in codice MATLAB mentre quelli basati su *Velocità* e *Momento* sono stati realizzati su Simulink per provare un approccio diverso. Durante le simulazioni il braccio robotico è sottoposto a un *controllo basato su dinamica inversa* [1, Sez. 6.4.3]. In tutte le simulazioni lo stato iniziale del robot è sempre agganciato alla traiettoria. Va sottolineato che nella realizzazione dell'*Osservatore basato sul Momento*, la quantità $\beta(\mathbf{q}, \dot{\mathbf{q}})$ (3.15) è stata realizzata utilizzando la derivata numerica della Matrice d'Inerzia; questo a causa delle limitazioni del Robotics System Toolbox [9]. Ciò ovviamente aumenta la rumorosità del residuo. Infine sono stati scelti uguali guadagni per gli osservatori con $k_{O,i} = 100$.

4.1.1 Scenario di Test 1: Collisione a Velocità Nulla

Il primo scenario di test prevede che il robot si scontri con una persona mentre esegue una traiettoria. Il percorso che l'end effector deve seguire è una circonferenza di raggio pari a 10 cm. La velocità con il quale viene percorsa la circonferenza è costante e il tempo necessario a compiere un giro è di 5 secondi. Vengono percorsi in totale due giri e solo nel secondo avviene la collisione. La parte del robot che entra in contatto con la persona è l'end effector; per semplicità è stata considerata l'origine del

frame associato ad esso come punto di applicazione delle forze di collisione. L'essere umano è geometricamente rappresentato nello spazio come un piano che interseca la traiettoria. L'interazione tra i corpi è modellata come un *sistema molla-smorzatore* con *dinamica attiva solo in compressione*.

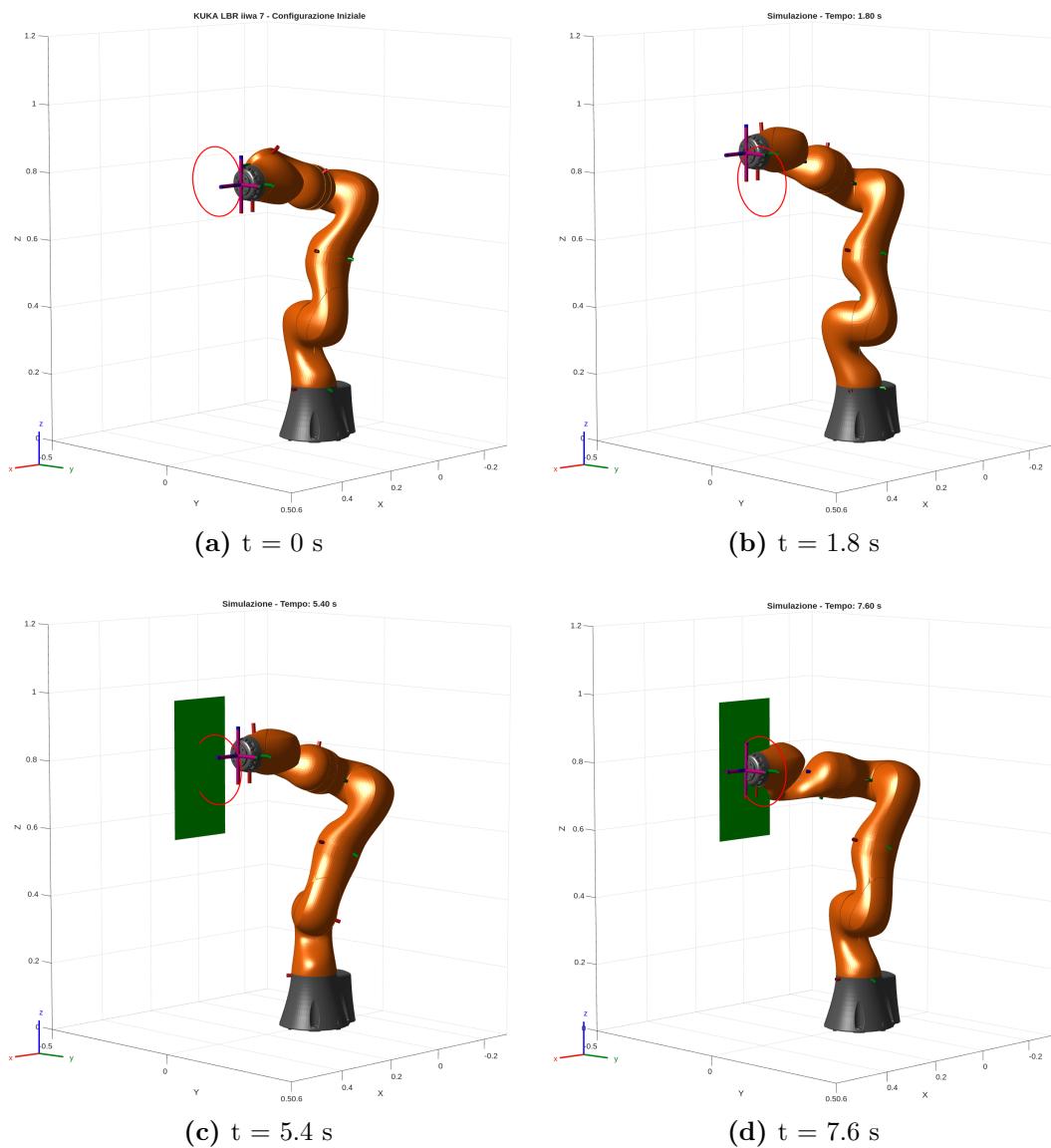


Figura 4.1. Primo scenario di test: in rosso è disegnata la traiettoria desiderata mentre in verde il piano rappresentante la persona. 4.1a Configurazione iniziale; 4.1b Primo giro pre collisione; 4.1c Secondo giro pre collisione; 4.1d Secondo giro in collisione.

Dai grafici in Fig. 4.2 si può vedere che avvengono in totale quattro collisioni; ciò accade in quanto il controllore prova a riportare il robot in traiettoria portandolo a sbattere nuovamente. All'interno del repository presente in App. B è possibile trovare un video che mostra la simulazione. Dalla figura possiamo notare come tutti e tre i metodi riescano a rilevare in tempi comparabili le collisioni.

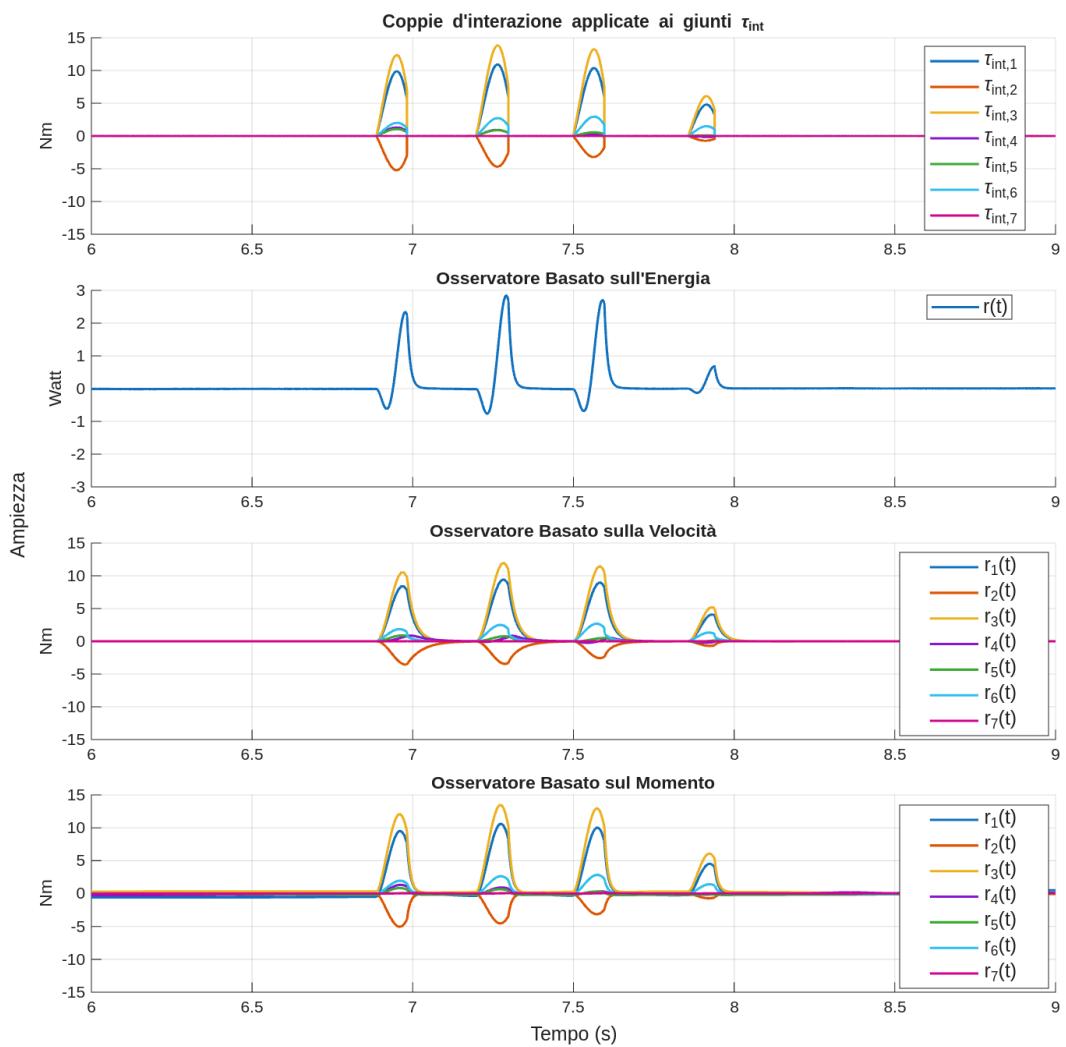


Figura 4.2. Andamento delle coppie d'interazione e dei segnali residuali durante le collisioni.

4.1.2 Scenario di Test 2: Collisione Istantanea a Velocità Nulla

Il secondo scenario di test prevede che il robot sia fermo e che avvenga una collisione istantanea. Quest'ultima ha una durata di cinque centesimi di secondo e coinvolge esclusivamente i primi tre giunti. Dai grafici in Fig. 4.3 possiamo vedere come il residuo dell'*Osservatore basato sull'Energia* non viene praticamente eccitato. In quello generato dall'*Osservatore basato sulla Velocità*, si attiva anche la quarta componente $r_4(t)$ a causa della *dinamica accoppiata*. Un'evidenza sperimentale di ciò che abbiamo visto nel precedente capitolo.

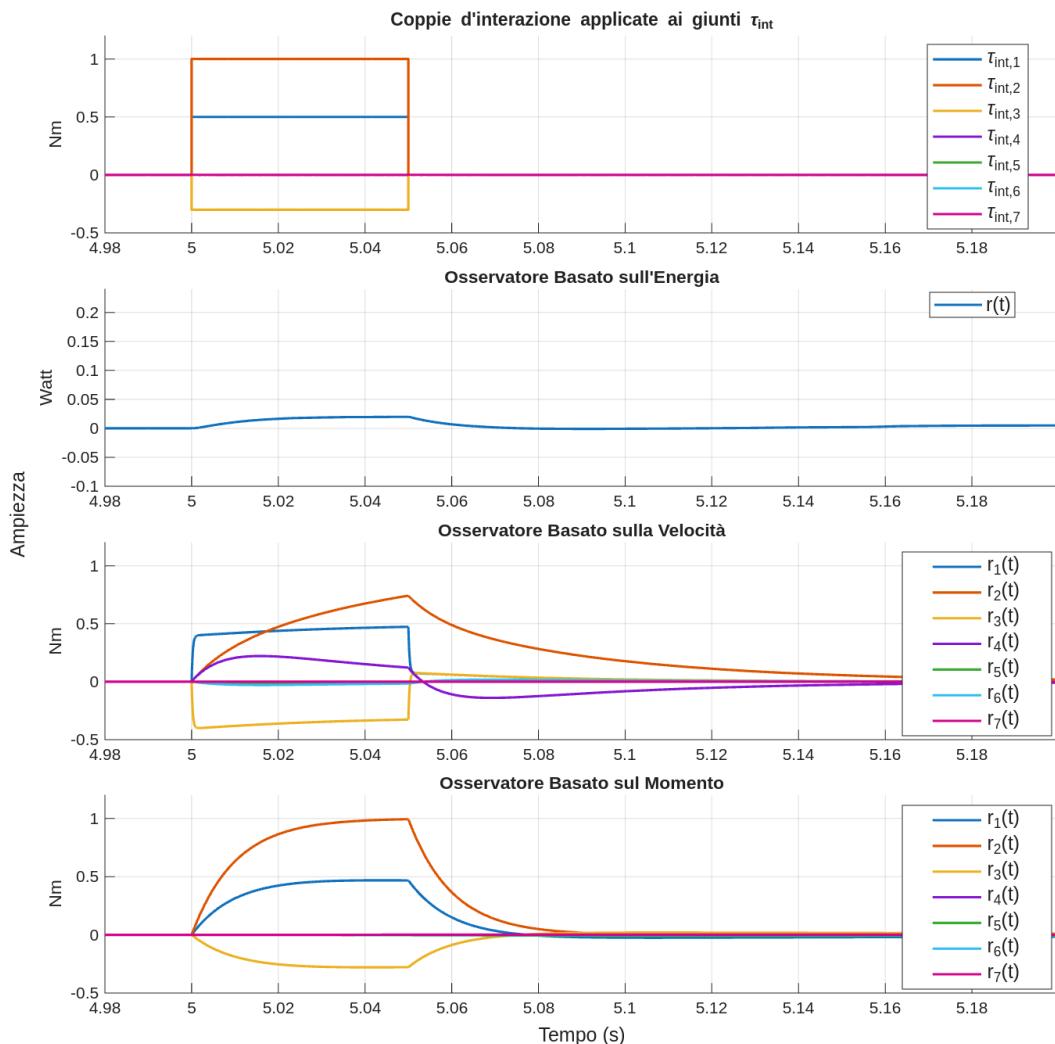


Figura 4.3. Andamento delle coppie d'interazione e dei segnali residuali durante la collisione istantanea a velocità nulla.

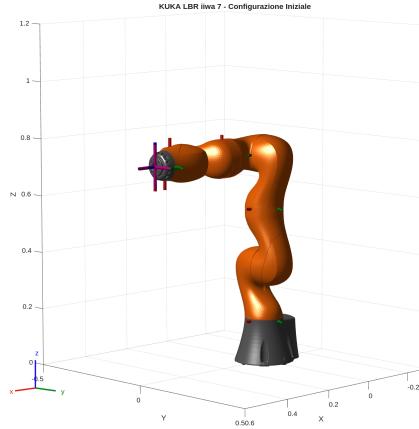


Figura 4.4. Secondo scenario di test: configurazione del robot.

4.2 Validazione classe C++ MomentumObserver (MuJoCo & Pinocchio)

In questa sezione analizziamo uno dei test effettuati per validare l’implementazione della classe presentata in App. A. Il modello del KUKA LBR iiwa 14 è basato sull’implementazione fornita da MuJoCo Menagerie [10]. Sono state fatte opportune modifiche agli attuatori in modo da poterli comandare in coppia. Il robot rimane fermo mentre da $t = 2$ s fino a $t = 5$ s viene applicata una forza all’origine del frame associato al settimo link. L’algoritmo di controllo utilizzato è basato su *linearizzazione esatta* [1, Sez. 6.4.2]. La matrice dei guadagni dell’osservatore è stata scelta con $k_{O,i} = 100$. La durata dell’intervallo d’osservazione in assenza di collisioni è stata scelta pari a $\Delta t = 0.1$ s. Il margine di sicurezza è stato impostato a $\varepsilon_{\text{safe}} = 10^{-4}$. Lo scopo di questo test era valutare contemporaneamente le capacità di *Rilevamento*, *Isolamento* e *Identificazione* della classe. Dalla Fig. 4.6 si può vedere come la forza viene ricostruita correttamente e la funzione di rilevamento delle collisioni isolati il link giusto.

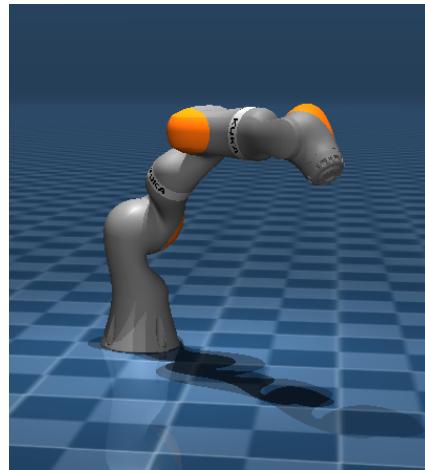


Figura 4.5. MuJoCo: configurazione del robot

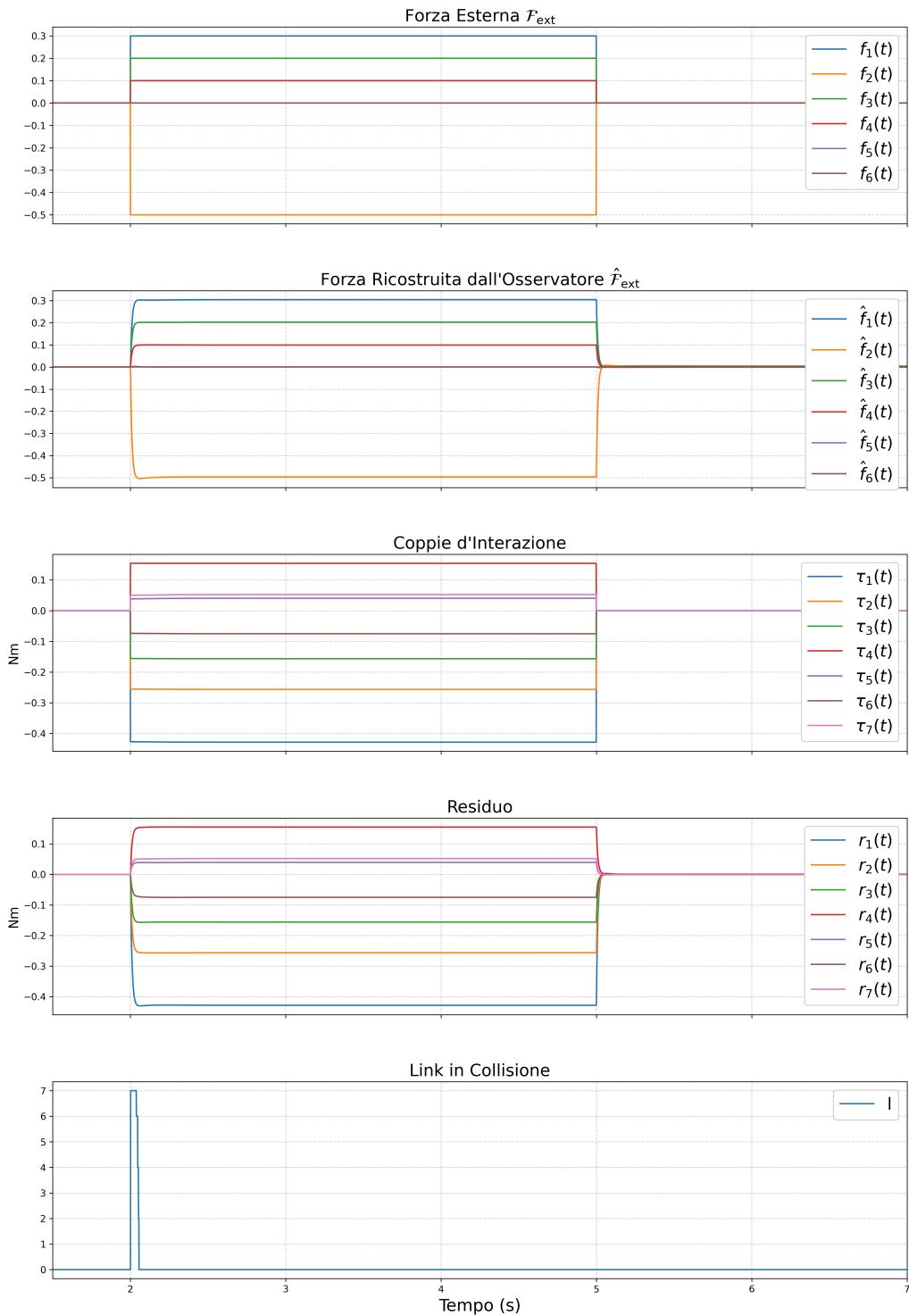


Figura 4.6. Andamento del residuo, della forza ricostruita e della funzione di rilevamento delle collisioni

Capitolo 5

Applicazione su Robot Umanoide: Stima delle Forze di Contatto al Suolo

In questo capitolo, l'analisi sperimentale viene estesa a un sistema robotico complesso: il robot umanoide *Unitree G1*. A differenza dei manipolatori seriali a base fissa, l'applicazione su un umanoide introduce le sfide legate alla dinamica della *base mobile* e alla *sotto-attuazione* del sistema. Viene quindi illustrato come l'*Osservatore basato sul Momento* possa essere utilizzato per stimare in tempo reale le *forze generalizzate d'interazione* tra i piedi e il terreno. Infine, viene mostrato un possibile utilizzo di tali ricostruzioni: il calcolo dello *Zero Moment Point* (ZMP) del robot.

5.1 Modello Dinamico di un Robot Umanoide

In un robot umanoide con n gradi di libertà, sei di questi sono legati alla traslazione e rotazione del frame di base del robot. Non potendo intervenire direttamente su tutte le variabili di stato, si ha che il sistema risulta *sotto-attuato*. Come discusso in [11, Sez. 3.2], l'equazione dinamica che descrive un robot umanoide con vincoli di contatto ai piedi è¹:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{S}\boldsymbol{\tau}_m + \mathbf{J}_l^T(\mathbf{q})\mathcal{F}_l + \mathbf{J}_r^T(\mathbf{q})\mathcal{F}_r, \quad (5.1)$$

dove:

- $\mathbf{S} \in \mathbb{R}^{n \times (n-6)}$ è la matrice d'accoppiamento degli ingressi;
- $\mathbf{J}_l, \mathbf{J}_r \in \mathbb{R}^{6 \times n}$ sono i Jacobiani associati ai frame solidali ai piedi, rispettivamente al sinistro e al destro;
- $\mathcal{F}_l, \mathcal{F}_r \in \mathbb{R}^6$ sono le forze generalizzate d'interazione con il suolo applicate ai due piedi.

Essendo il robot sotto-attuato si ha $\boldsymbol{\tau}_m \in \mathbb{R}^{n-6}$; inoltre le righe di \mathbf{S} corrispondenti alle coordinate della base mobile sono nulle.

¹Si è assunto $\boldsymbol{\tau}_{\text{fric}} = \mathbf{0}$.

5.2 L'Osservatore basato sul Momento Generalizzato per Robot Umanoidi

È possibile applicare l'*Osservatore basato sul Momento* visto in Sez. 3.3 al robot descritto dalla (5.1). Bisogna ripetere gli stessi passaggi visti nelle Sez. 2.3 e 3.3 considerando $S\tau_m \in \mathbb{R}^n$ al posto del termine τ_m . Possiamo poi supporre che le coppie d'interazione vengano generate esclusivamente dalle forze di contatto al suolo, ovvero:

$$\tau_{\text{int}} = J_l^T(\mathbf{q})\mathcal{F}_l + J_r^T(\mathbf{q})\mathcal{F}_r = [J_l^T(\mathbf{q}) \quad J_r^T(\mathbf{q})] \begin{bmatrix} \mathcal{F}_l \\ \mathcal{F}_r \end{bmatrix}. \quad (5.2)$$

Nel caso ideale la relazione (3.18) diventa quindi:

$$\dot{\mathbf{r}} = \mathbf{K}_O \left([J_l^T(\mathbf{q}) \quad J_r^T(\mathbf{q})] \begin{bmatrix} \mathcal{F}_l \\ \mathcal{F}_r \end{bmatrix} - \mathbf{r} \right). \quad (5.3)$$

Infine dal segnale residuale possiamo ricostruire le forze generalizzate di contatto nel seguente modo:

$$\begin{bmatrix} \hat{\mathcal{F}}_l \\ \hat{\mathcal{F}}_r \end{bmatrix} = [J_l^T(\mathbf{q}) \quad J_r^T(\mathbf{q})]^\dagger \mathbf{r}. \quad (5.4)$$

5.3 Analisi Sperimentale sul Robot Unitree G1

Analizziamo ora l'implementazione dell'*Osservatore basato sul Momento* vista in Sez. 5.2 su un robot umanoide *Unitree G1* (Fig. 5.1). Il codice sorgente relativo a questa sezione non è reso pubblicamente disponibile in quanto soggetto a proprietà intellettuale condivisa con il Robotics Lab. Lo scenario prevede che a partire da $t = 10$ s il robot sposti il peso a destra e a sinistra rimanendo sul posto. I test sono stati effettuati contemporaneamente sia in simulazione MuJoCo che sul robot reale (Fig. 5.2). È stata utilizzata nuovamente la classe `MomentumObserver` (App. A) con gli opportuni accorgimenti. L'osservatore è stato impostato con $k_{O,i} = 50$, $\Delta t = 0.1$ s e $\varepsilon_{\text{safe}} = 10^{-4}$. Dalla Fig. 5.3 possiamo vedere l'andamento della ricostruzione delle forze assieme a quelle calcolate da MuJoCo. L'elevata rumorosità della ricostruzione sul robot reale deriva dal rumore di misura presente nelle velocità delle coordinate generalizzate.



Figura 5.1. Il robot umanoide Unitree G1 (Per gentile concessione di Unitree)

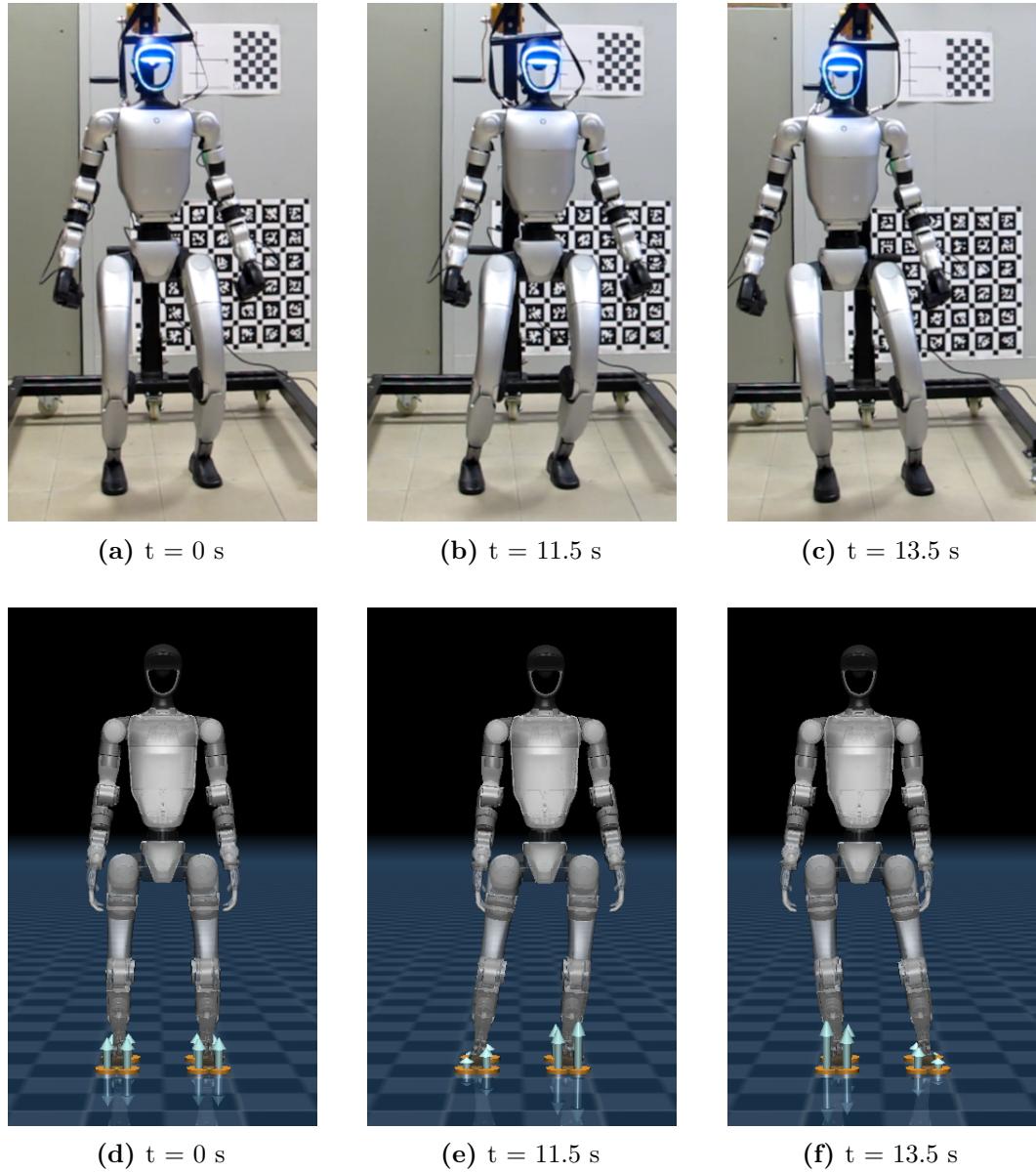


Figura 5.2. Scenario di test: 5.2a, 5.2b, 5.2c raffigurano il robot reale; 5.2d, 5.2e, 5.2f raffigurano la simulazione MuJoCo.

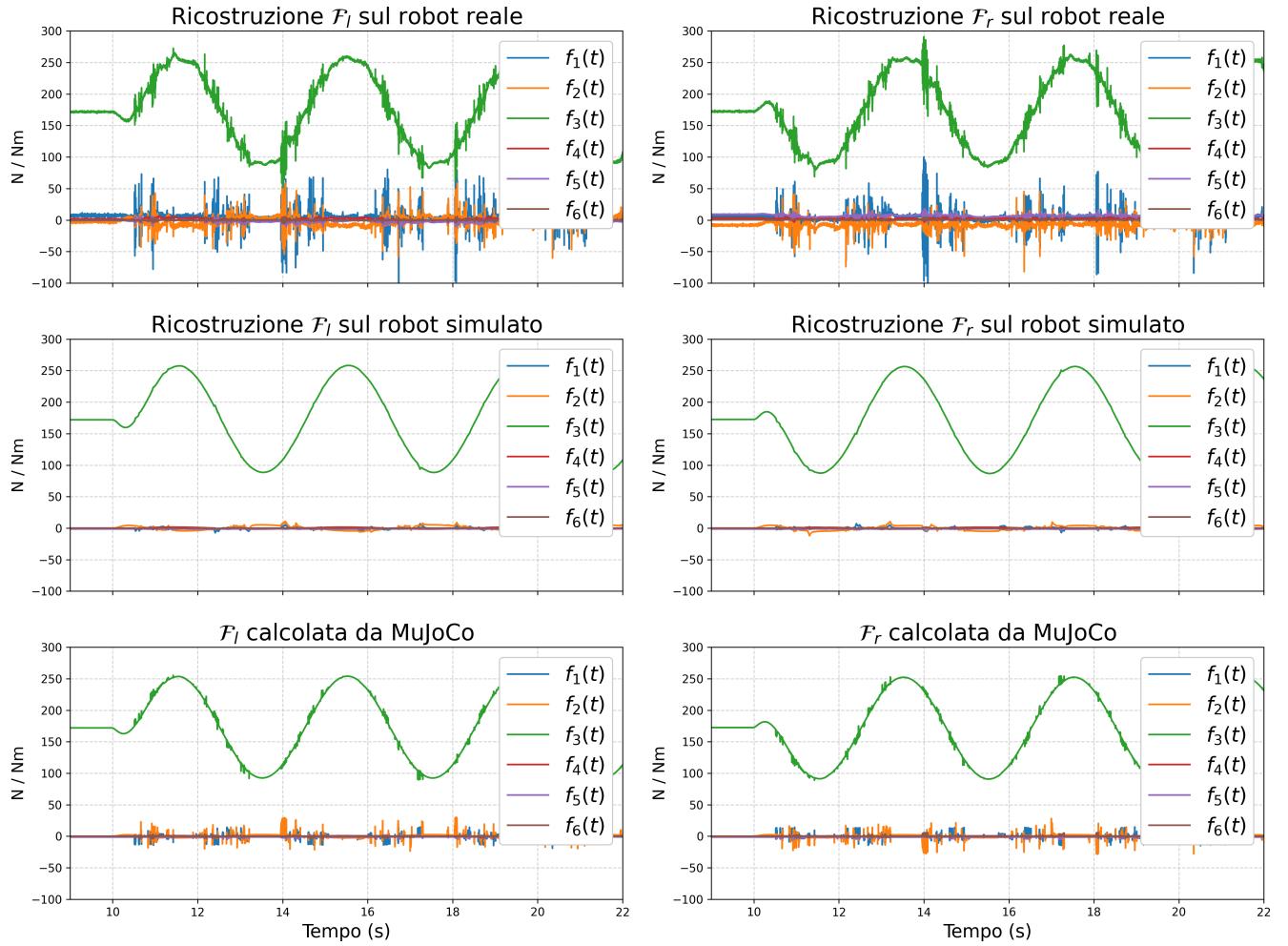


Figura 5.3. Confronto delle forze di contatto stimate. La colonna di sinistra riporta i dati relativi al piede sinistro (\mathcal{F}_l), quella di destra al piede destro (\mathcal{F}_r). Le righe mostrano, dall’alto verso il basso: la ricostruzione sul robot reale, la ricostruzione sul robot simulato e i valori di riferimento calcolati dal motore fisico MuJoCo.

Calcolo dello ZMP del robot

Le forze e i momenti ricostruiti costituiscono l'input necessario per il calcolo dello *Zero Moment Point* (ZMP), l'indice di stabilità dinamica fondamentale per garantire l'equilibrio dei robot umanoidi durante la locomozione. È stato quindi calcolato lo ZMP a partire dalle forze ricostruite. Tale calcolo è stato effettuato avvalendosi di moduli software condivisi, sviluppati all'interno del laboratorio. In Fig. 5.4 le traiettorie dello ZMP risultanti sono messe a confronto con l'evoluzione della posizione del Centro di Massa (CoM) del robot.

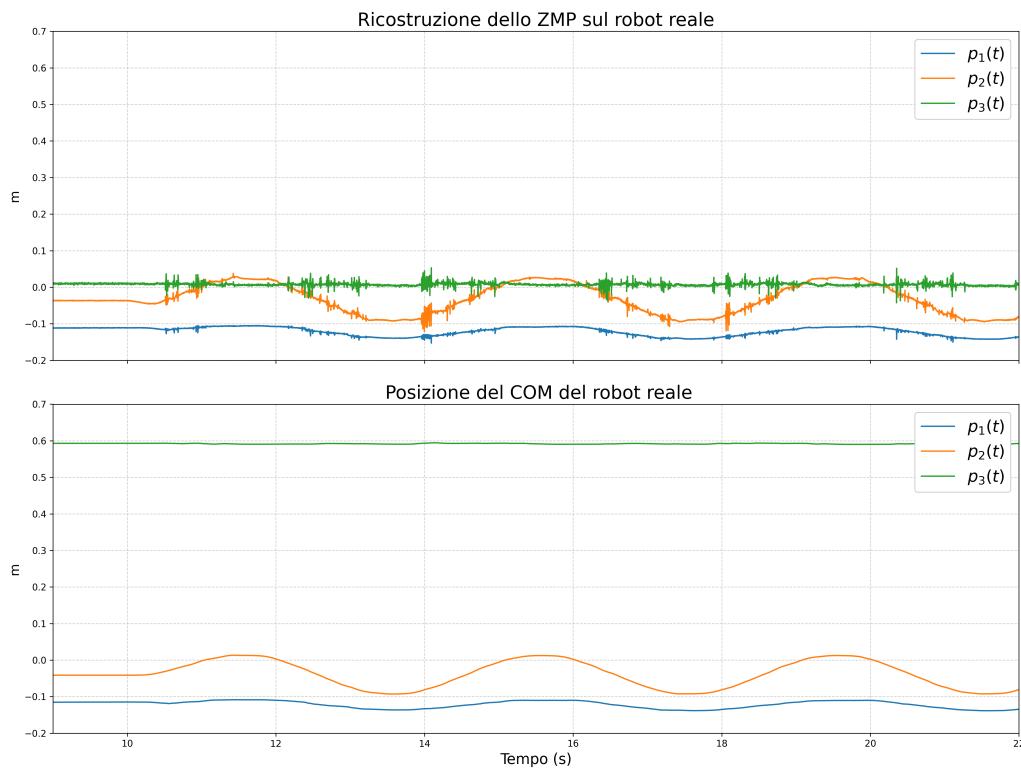


Figura 5.4. Confronto tra le traiettorie dello ZMP ricostruito e quelle del COM del robot.

Capitolo 6

Conclusioni e Sviluppi Futuri

Il lavoro svolto in questa tesi ha affrontato la problematica della gestione delle collisioni nei robot manipolatori e umanoidi, analizzando in dettaglio strategie basate esclusivamente sulla proprioceuzione. Attraverso un percorso che ha spaziato dalla modellistica dinamica all'implementazione in simulazione e su robot reali, è stato possibile valutare l'efficacia di diversi osservatori per il *Rilevamento*, l'*Isolamento* e l'*Identificazione* dei contatti.

6.1 Analisi Comparativa dei Metodi

Dall'analisi teorica e dai risultati sperimentali presentati nei capitoli precedenti, emerge un quadro chiaro riguardo le prestazioni e l'applicabilità dei tre metodi presi in esame.

- L'**Osservatore basato sull'Energia**, sebbene computazionalmente leggero e semplice da implementare, ha mostrato limiti intrinseci significativi. La natura scalare del residuo impedisce l'*Isolamento* del link in collisione e l'*Identificazione* della forza. Inoltre, l'incapacità di rilevare collisioni a robot fermo o forze ortogonali al vettore velocità lo rende inadatto per garantire una sicurezza completa in scenari non strutturati.
- L'**Osservatore basato sulla Velocità** supera i limiti del metodo energetico fornendo un residuo vettoriale. Tuttavia, la dipendenza dall'inversa della matrice d'inerzia $\mathbf{M}^{-1}(q)$ introduce un accoppiamento dinamico tra le componenti del residuo. Come evidenziato nelle simulazioni, una collisione su un singolo link eccita le componenti del residuo relative ad altri giunti, complicando notevolmente la fase d'*Isolamento* e richiedendo soglie di rilevamento più conservative e dipendenti dalla configurazione.
- L'**Osservatore basato sul Momento Generalizzato** si è dimostrato essere la soluzione superiore sotto ogni aspetto. Grazie alla proprietà di disaccoppiamento del momento generalizzato, questo osservatore permette d'isolare con precisione il link soggetto alla collisione e di ricostruire fedelmente le coppie d'interazione esterne senza richiedere l'inversione della matrice d'inerzia. La sua robustezza e la capacità di fornire informazioni direzionali sulle forze di

contatto lo rendono lo strumento ideale per la gestione completa di una *pipeline di gestione delle collisioni*.

6.2 Estensione ai Robot Umanoidi

Un contributo significativo di questo elaborato è stato dimostrare che l'applicabilità dell'*Osservatore basato sul Momento* non è confinata ai manipolatori seriali a base fissa. Come discusso nel Cap. 5, l'algoritmo è stato esteso con successo al robot umanoide *Unitree G1*, un sistema a base mobile e sotto-attuato. I test condotti, sia in simulazione MuJoCo che sull'hardware reale, hanno confermato la capacità dell'osservatore di stimare in tempo reale le forze di contatto al suolo. L'utilizzo di tali stime per il calcolo dello *Zero Moment Point* (ZMP) evidenzia come questo strumento possa fungere da "*sensore di forza virtuale*" affidabile, riducendo la necessità di costosi *sensori di forza-coppia* ai piedi e semplificando l'architettura hardware del robot.

6.3 Sviluppi Futuri

I risultati ottenuti aprono la strada a diverse linee di ricerca e miglioramenti futuri, con particolare attenzione all'applicazione su robot umanoidi:

- **Stima dei contatti multi-punto e alle mani:** Nel presente lavoro ci si è concentrati sulla stima delle forze ai piedi per l'umanoide. Uno sviluppo naturale consiste nell'estendere l'osservatore per includere i Jacobiani delle mani e di altri possibili punti di contatto (es. torso o gomiti). Ciò permetterebbe al robot di gestire scenari di *whole-body manipulation*, come appoggiarsi a una parete per mantenere l'equilibrio o manipolare oggetti bimanuali, distinguendo le forze necessarie alla locomozione da quelle di manipolazione.
- **Integrazione nel Controllo della Locomozione:** Attualmente, le forze e lo ZMP ricostruiti sono utilizzati principalmente per monitoraggio. Un passo successivo fondamentale sarà integrare queste quantità all'interno dell'anello di controllo. Le stime dell'osservatore potrebbero essere utilizzate per chiudere un loop di controllo di forza o di impedenza, permettendo al robot di adattare la camminata su terreni accidentati, reagire a spinte esterne impreviste o bilanciare disturbi dinamici in modo più reattivo rispetto all'uso dei soli sensori inerziali.
- **Fusione con Sensori Esterocettivi:** Sebbene l'approccio propriocettivo si sia rivelato robusto, presenta limitazioni nella localizzazione geometrica esatta del punto di contatto lungo il link. Un promettente sviluppo futuro riguarda la fusione sensoriale tra il residuo dell'osservatore e le informazioni provenienti da sensori esterocettivi (come *sistemi di visione*, *Lidar* o *pelli tattili*). Tale sinergia aprirebbe nuovi scenari applicativi: l'integrazione di dati eterogenei permetterebbe infatti d'isolare e ricostruire *contatti multipli simultanei*, una condizione in cui la sola propriocezione risulterebbe ambigua aggregando gli effetti in un'unica forza risultante. Inoltre, l'apporto di sensori

esterni risulterebbe determinante in prossimità di *configurazioni singolari*, dove lo Jacobiano perde rango e l'osservatore diviene strutturalmente "cieco" alle forze applicate lungo certe direzioni, garantendo così una rilevazione continua e affidabile nell'intero spazio operativo.

In conclusione, l'*Osservatore basato sul Momento Generalizzato* rappresenta una tecnologia abilitante fondamentale per la prossima generazione di robot collaborativi e umanoidi, garantendo un'interazione fisica sicura e consapevole con l'ambiente e con l'uomo.

Appendice A

Implementazione della Classe C++ MomentumObserver

Per facilitare l'integrazione dell'*Osservatore basato sul Momento Generalizzato* in diversi contesti simulativi e reali, è stata sviluppata una classe C++ modulare e autocontenuta, `MomentumObserver`. L'obiettivo del design è incapsulare la logica dell'osservatore, esponendo un'interfaccia chiara per l'*inizializzazione*, l'*aggiornamento* e l'*interrogazione* dello stato. La classe fa uso delle librerie Pinocchio [7, 8] ed Eigen [12], usate rispettivamente per effettuare i calcoli relativi la dinamica e l'algebra lineare.

A.1 Interfaccia Pubblica (API)

L'interfaccia pubblica della classe `MomentumObserver` è progettata per essere minimale e intuitiva. Essa espone i seguenti metodi principali per gestire il ciclo di vita e l'esecuzione dell'osservatore:

A.1.1 Costruzione e Inizializzazione

Il costruttore richiede il vettore dei guadagni dell'osservatore; il modello e i dati del robot caricati tramite Pinocchio; il passo di campionamento; la dimensione del buffer dei residui (in modo da definire Δt (3.25)); il margine di sicurezza ϵ_{safe} (3.26).

```

1  MomentumObserver(
2      const Eigen::VectorXd& diagKo,
3      const pinocchio::Model& robot_model,
4      const pinocchio::Data& robot_data,
5      double dt,
6      int dim_buffer_r,
7      double epsilon
8  );

```

Listing A.1. Interfaccia del costruttore.

A.1.2 Ciclo di Aggiornamento

Il metodo `update()` è il cuore dell’osservatore e deve essere chiamato a ogni ciclo di controllo. Richiede lo stato attuale del robot e le coppie applicate ai giunti. Restituisce il residuo all’istante di tempo attuale.

```

1   Eigen::VectorXd update(
2       const Eigen::VectorXd& q,
3       const Eigen::VectorXd& qdot,
4       const Eigen::VectorXd& tau
5   );

```

Listing A.2. Metodo di aggiornamento del residuo.

A.1.3 Metodi di Accesso per Rilevamento, Isolamento e Identificazione

La classe mette a disposizione una serie di metodi volti a implementare le logiche viste nella Sez. 3.4.

Il metodo `collisionDetect()` implementa la funzione di rilevamento delle collisioni. A differenza della (3.27) viene restituito l’indice del link entrato in contatto, in modo da incorporare anche la (3.30). Se non viene rilevata alcuna collisione viene restituito zero.

```

1   int collisionDetect();

```

Listing A.3. Metodo utile al Rilevamento delle collisioni e all’Isolamento del link di contatto

Il metodo `reconstructForceWrench()` ricostruisce le forze. Prende come parametro lo Jacobiano del punto rispetto al quale si vuole ricostruire la forza. Può essere utilizzato per calcolare sia la (3.37) che la (3.31).

```

1   Eigen::VectorXd reconstructForceWrench(
2       const Eigen::MatrixXd& J
3   );

```

Listing A.4. Metodo utile all’Identificazione delle forze

Il metodo `estimateContactPointInLinkReferenceFrame()` risolve il sistema (3.34) restituendo la *least square solution* (3.35). Prende come parametro la forza generalizzata espressa nel sistema di riferimento del link (2.19).

```

1   Eigen::VectorXd estimateContactPointInLinkReferenceFrame(
2       const Eigen::VectorXd& F
3   );

```

Listing A.5. Metodo utile all’Isolamento del punto di contatto

Appendice B

Codice Sorgente

Il codice sorgente sviluppato per le simulazioni presentate in questa tesi è liberamente accessibile e disponibile online nei seguenti repository GitHub.

Simulazioni su Manipolatore Seriale (MATLAB & Simulink)

Il codice utilizzato per l'analisi comparativa degli Osservatori in Sez. 4.1.

github.com/visc-1/collision-observers-comparison-matlab

Classe MomentumObserver e Ambiente di Validazione (MuJoCo & Pinocchio)

Il codice C++ che implementa l'*Osservatore basato sul Momento Generalizzato* (App.A) assieme alla simulazione vista in Sez. 4.2.

github.com/visc-1/momentum-observer-mujoco

Bibliografia

- [1] Bruno Siciliano, Luigi Villani, Giuseppe Oriolo e Alessandro De Luca. *Foundations of Robotics*. en. Advanced Textbooks in Control and Signal Processing. Cham: Springer Nature Switzerland, 2025. ISBN: 978-3-031-85522-1 978-3-031-85523-8. DOI: 10.1007/978-3-031-85523-8. URL: <https://link.springer.com/10.1007/978-3-031-85523-8> (visitato il giorno 23/10/2025).
- [2] Adi Ben-Israel e Thomas N. E. Greville. *Generalized Inverses*. en. CMS Books in Mathematics. New York: Springer-Verlag, 2003. ISBN: 978-0-387-00293-4. DOI: 10.1007/b97366. URL: <http://link.springer.com/10.1007/b97366> (visitato il giorno 08/11/2025).
- [3] Sami Haddadin, Alessandro De Luca e Alin Albu-Schäffer. “Robot collisions: A survey on detection, isolation, and identification”. In: *IEEE Transactions on Robotics* 33.6 (2017). Publisher: IEEE, pp. 1292–1312. URL: <https://ieeexplore.ieee.org/abstract/document/8059840/> (visitato il giorno 23/10/2025).
- [4] The MathWorks Inc. *MATLAB version: 25.1.0.2973910 (R2025a)*. Natick, Massachusetts, United States, 2025. URL: <https://www.mathworks.com>.
- [5] The MathWorks Inc. *Simulink version: 25.1 (R2025a)*. Natick, Massachusetts, United States, 2025. URL: <https://www.mathworks.com/products/simulink.html>.
- [6] Emanuel Todorov, Tom Erez e Yuval Tassa. “MuJoCo: A physics engine for model-based control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.
- [7] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse e N. Mansard. “The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives”. In: *International Symposium on System Integration (SII)*. 2019.
- [8] Justin Carpentier, Joseph Mirabel, Nicolas Mansard e the Pinocchio Development Team. *Pinocchio: fast forward and inverse dynamics for poly-articulated systems*. <https://stack-of-tasks.github.io/pinocchio>. 2015–2018.
- [9] The MathWorks Inc. *Robotics System Toolbox version: 25.1 (R2025a)*. Natick, Massachusetts, United States, 2025. URL: <https://www.mathworks.com/products/robotics.html>.

- [10] Kevin Zakka, Yuval Tassa e MuJoCo Menagerie Contributors. *MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo*. Set. 2022. URL: https://github.com/google-deepmind/mujoco_menagerie.
- [11] Tomomichi Sugihara e Katsu Yamane. “Reduced-Order Models”. en. In: *Humanoid Robotics: A Reference*. A cura di Ambarish Goswami e Prahlad Vaidakkepat. Dordrecht: Springer Netherlands, 2019, pp. 811–848. ISBN: 978-94-007-6046-2. DOI: 10.1007/978-94-007-6046-2_56. URL: https://doi.org/10.1007/978-94-007-6046-2_56 (visitato il giorno 25/11/2025).
- [12] Gaël Guennebaud, Benoît Jacob et al. *Eigen v5*. <http://eigen.tuxfamily.org>. 2010–2025.