



UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

Analisi di un Manipolatore RPR: Pianificazione della Traiettoria, Analisi Cinematica e Dinamica e Sistema di Controllo

Michele Ferrari, Riccardo Monti

Matricole: 719669, 719525

Dipartimento di Ingegneria Meccanica e Industriale, Università degli Studi di Brescia

Referente: Prof. Giovanni Legnani

Keywords: Trajectory Planning, Kinematics, Dynamics, Servosystems

2 Settembre 2021

Abstract Studio di un robot RPR ed elaborazione delle traiettorie da seguire e delle leggi di moto. Analisi cinematica, diretta ed inversa, e dinamica inversa tramite jacobiano e matrici omogenee su tre diversi task. Verifica della correttezza degli approcci tramite derivazione numerica. Realizzazione di un modello del manipolatore con l'utilizzo dei software Solidworks e Simscape e realizzazione in Simulink di un sistema di controllo tramite PID in configurazione a cascata.

Materiali e codice sorgente sono disponibili al seguente [link](#).

Introduzione

Questo elaborato presenta lo svolgimento del progetto finale per il corso di *Servosystems and Robotics*, tenuto dal prof. Giovanni Legnani (a.a. 2020 - 2021).

Il compito assegnato consiste nel modellizzare un manipolatore seriale che sia in grado di seguire una traiettoria assegnata che varia, insieme alla legge di moto, a seconda del task da eseguire. Il progetto delle leggi di moto deve garantire il rispetto dei limiti di spostamento, velocità e accelerazione dei giunti e i valori massimi di coppia e forza che gli attuatori possono fornire.

Nella Sez. 1 si presenta la struttura del manipolatore, le tipologie di giunti, dimensioni, masse, inerzie e limiti sugli attuatori, per i quali si è presa ispirazione da modelli di robot seriali esistenti.

Nelle Sez. 2 e 3 è riassunta l'analisi cinematica e dinamica del manipolatore per i diversi task, utilizzando un duplice approccio: jacobiano e matriciale.

La Sez. 4 descrive il passaggio ad una modellizzazione più accurata grazie all'utilizzo di Solidworks: il manipolatore è stato disegnato e dimensionato, sono stati assegnati i sistemi di riferimento coerentemente con quanto stabilito nella Sez. 2 e il software ha eseguito in modo automatico il calcolo delle posizioni dei baricentri dei *link*, le loro masse, momenti e prodotti d'inerzia. Mediante un apposito *plug-in*, il robot è stato importato automaticamente in ambiente Simscape, dove è stato assemblato uno schema di attuazione dei giunti per verificare la coerenza dell'assieme con i risultati teorici ottenuti in precedenza.

Nella Sez. 5, infine, si illustra il sistema di controllo impiegato: l'architettura complessiva, la tipologia di controllori, il *tuning* dei parametri e i risultati ottenuti. Si conclude riportando brevi cenni a possibili *improvement* per il *tuning* dei regolatori.

Il lavoro si chiude con alcune osservazioni di confronto tra i due metodi utilizzati per lo studio cinematico e dinamico, evidenziando vantaggi e svantaggi di uno e dell'altro.

1 Descrizione del Manipolatore e del Task da Eseguire

1.1 Geometria del Manipolatore

Una rappresentazione grafica del manipolatore è mostrata nelle viste multiple in Fig. 1, mentre uno schema di principio (Fig. 7) è presentato nella Sez. 2 dedicata all'analisi cinematica, dove sono evidenti le definizioni delle grandezze geometriche presentate di seguito.

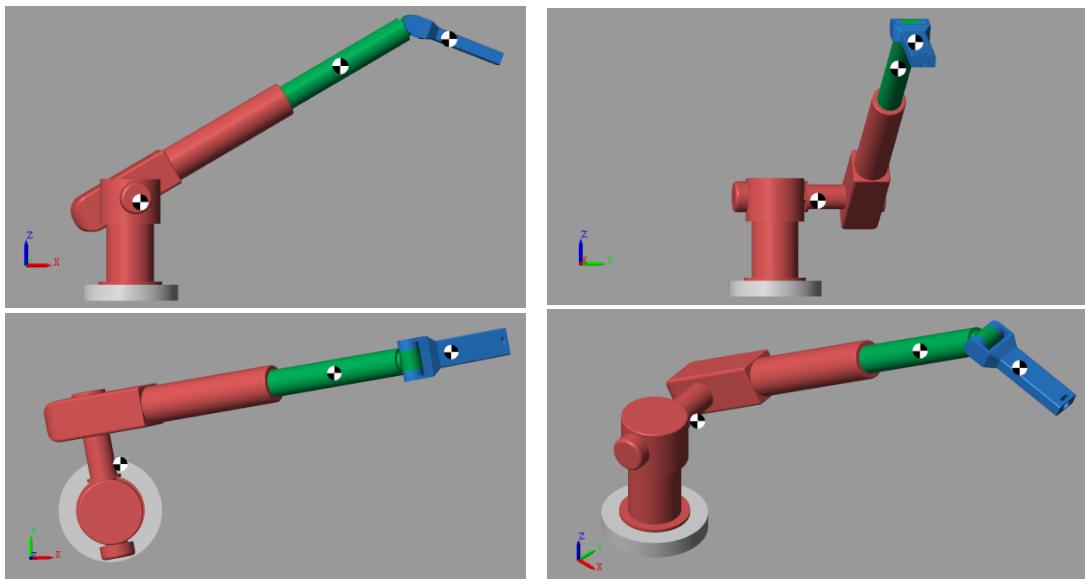


Fig. 1. Rappresentazione del manipolatore da diversi punti di vista.

Il robot in esame è costituito da 3 *link*:

- il **link 1**, rosso, ha una forma complessa e il suo baricentro non cade all'interno del volume effettivamente occupato dai componenti che lo costituiscono. È descritto dalle dimensioni $l_{1a} = 270 \text{ mm}$, $l_{1b} = 278 \text{ mm}$, $l_{1c} = 620 \text{ mm}$, $\alpha = \pi/6 \text{ rad} = 30^\circ$. Può ruotare rispetto all'asse z del riferimento posto alla sua base mediante un giunto rotoidale.
- il **link 2**, verde, è lo stelo di un pistone cilindrico, alla cui estremità è saldata l'interfaccia verso un giunto rotoidale. Il movimento è di tipo traslatorio e la lunghezza del link quando è completamente ritratto ($q_2 = 0$) è pari a $l_2 = 110 \text{ mm}$.
- il **link 3**, blu, è quello più esterno e si può interfacciare con una pinza. È collegato al *link* precedente mediante un giunto rotoidale, montato in modo tale che la sua rotazione possa avvenire in piani perpendicolari a quello della base del robot. La lunghezza del *link* è pari a $l_3 = 300 \text{ mm}$.

Le posizioni dei **baricentri** dei *link* sono riferite al sistema di riferimento posto sul *link* stesso (vedere Fig. 10) e valgono:

$$G_1 = [-0.179, -0.078, -0.040] \text{ mm},$$

$$G_2 = [0, 0.240, 0] \text{ mm},$$

$$G_3 = [-0.172, 0, 0] \text{ mm}.$$

Le masse dei *link* valgono:

$$m_1 = 117.657 \text{ kg}, \quad m_2 = 21.103 \text{ kg}, \quad m_3 = 8.522 \text{ kg}.$$

Le matrici di inerzia dei *link* sono state calcolate in automatico da Solidworks ed è stata applicata una conversione dalla convenzione adottata dal software a quella di più vasto utilizzo:

$$J_g = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \quad (1)$$

Vengono riportati di seguito i valori approssimati di momenti e prodotti di inerzia di tutti e tre i *link*, espressi in un sistema di riferimento baricentrale con assi paralleli alla terna posta sul

link:

$$J_{g_1} = \begin{bmatrix} 5.87 & -1.22 & -2.02 \\ -1.22 & 7.23 & -1.40 \\ -2.02 & -1.40 & 4.89 \end{bmatrix} \quad J_{g_2} = \begin{bmatrix} 0.52 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.52 \end{bmatrix} \quad J_{g_3} = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.07 & 0 \\ 0 & 0 & 0.08 \end{bmatrix}.$$

Sempre a livello di informazioni di tipo dinamico, sull'*end-effector* non sono applicate né forze, né coppie esterne, come del resto nessuna massa aggiuntiva è trattenuta dall'ipotetico *gripper* che andrebbe montato all'estremità del *link 3*.

1.2 Limiti degli Attuatori

Il manipolatore presenta 3 gradi di libertà, ciascuno attuato da un motore, due di tipo rotativo e uno di tipo lineare. Nella Sez. 2 è descritta rigorosamente l'analisi cinematica, mentre di seguito ci si limita ad un accenno alle variabili di giunto e alla definizione dei limiti che sono stati scelti per esse. Realisticamente, infatti, i motori presenteranno delle limitazioni di spostamento, velocità, accelerazione e forza/coppia. Di seguito sono riportati i valori assoluti dei vincoli scelti per ciascuno:

- rotazione attorno alla base: q_1 .

$$q_{1,max} = \pi \text{ rad}, \quad \dot{q}_{1,max} = \pi \text{ rad/s}, \quad \ddot{q}_{1,max} = 10\pi \text{ rad/s}^2, \quad T_{1,max} = 150 \text{ Nm};$$

- traslazione del *link 2* lungo il cilindro: q_2 .

$$q_{2,max} = 0.4 \text{ m}, \quad \dot{q}_{2,max} = 0.5 \text{ m/s}, \quad \ddot{q}_{2,max} = 5 \text{ m/s}^2, \quad F_{2,max} = 200 \text{ N};$$

- rotazione attorno al giunto del *link 3*: q_3 .

$$q_{3,max} = 0.5\pi \text{ rad}, \quad \dot{q}_{3,max} = 2\pi \text{ rad/s}, \quad \ddot{q}_{3,max} = 20\pi \text{ rad/s}^2, \quad T_{3,max} = 50 \text{ Nm};$$

1.3 Workspace secondo UNI ISO 29976

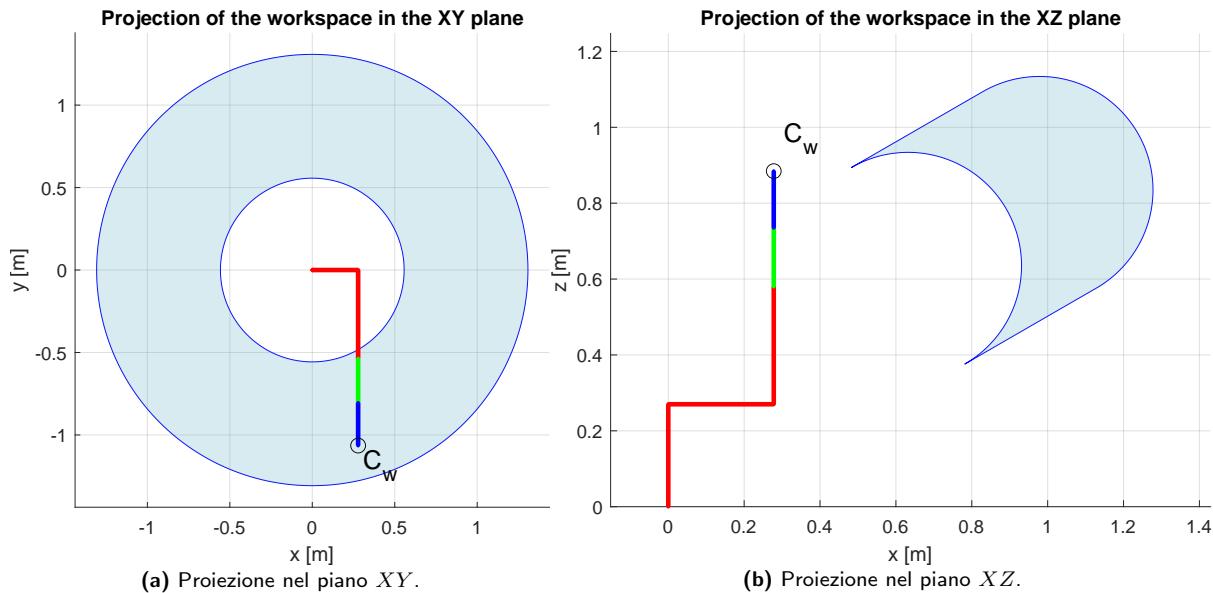


Fig. 2. Rappresentazione dello spazio di lavoro secondo la norma UNI ISO 29976.

Lo spazio di lavoro è definito come lo spazio raggiungibile dal *punto di riferimento del polso*, nel nostro caso coincidente con l'estremità dell'ultimo *link*. C_w è il *centro dello spazio di lavoro*, individuabile assegnando a tutte le coordinate di giunto il loro valore intermedio; in questo caso, pertanto, il robot viene rappresentato nella configurazione raggiunta con valori delle variabili di giunto pari a $(q_1, q_2, q_3) = (0 \text{ rad}, 0.2 \text{ m}, 0 \text{ rad})$.

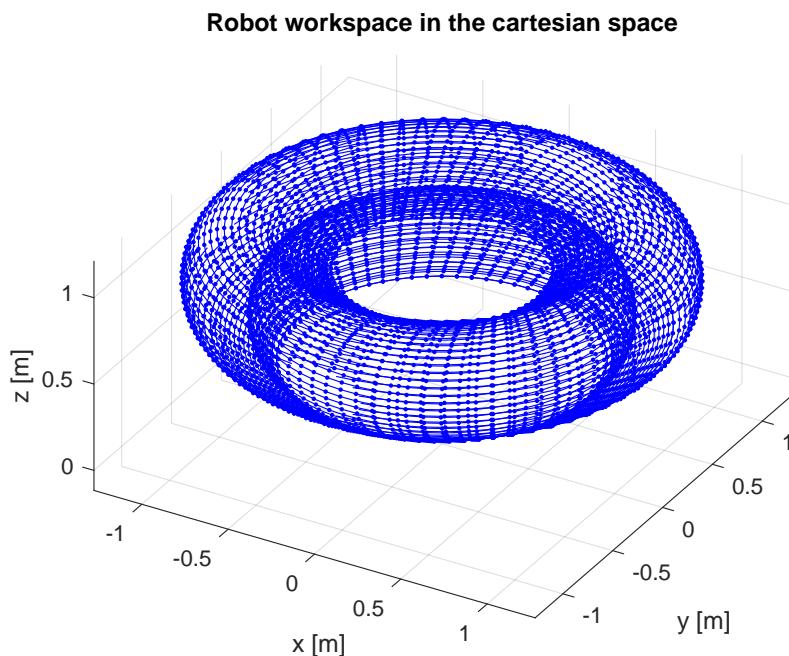


Fig. 3. Rappresentazione tridimensionale del workspace.

1.4 Task 1 e Task 3

Il compito che il manipolatore deve svolgere può essere suddiviso in tre *sotto-task* che possono essere trattati indipendentemente, a patto di garantire velocità ed accelerazione nulle nel momento in cui si passa da un compito al successivo. Si prevedono quindi due istanti di stop tra un movimento e il successivo.

Il **task 1** e il **task 3** sono del tutto analoghi: consistono nella movimentazione da un punto di partenza ad un punto di arrivo con legge di moto cicloidale e minimo tempo di attuazione nello spazio delle variabili di giunto. In particolare, il **task 1** consiste nel passare dalla posizione P_1 alla posizione P_2 (Fig. 17), punto iniziale della traiettoria da seguire nel **task 2**, mentre il **task 3** consiste nel passare dalla posizione P_3 , punto finale della stessa traiettoria, di nuovo alla posizione P_1 (Fig. 23).

Per adempiere alla richiesta di ottenere il minimo tempo di attuazione, per entrambi i *task* si è seguita la medesima strategia: calcolare il minimo tempo di attuazione per ciascun asse [1], scegliendo poi il massimo tra questi per calcolare le leggi di moto di tutti i giunti:

$$T_{min,i} = \max \left(\frac{2\Delta s}{\dot{q}_{i,max}}, \sqrt{\frac{2\pi\Delta s}{\ddot{q}_{i,max}}} \right), \quad (2)$$

dove $T_{min,i}$ è il minimo tempo di attuazione nello spazio dei giunti per il motore i – esimo, da cui

$$T_{tot} = \max (T_{min,1}, T_{min,2}, T_{min,3}). \quad (3)$$

Ci aspettiamo quindi che il giunto "collo di bottiglia" sia movimentato alla sua massima velocità e/o accelerazione, mentre per gli altri giunti la legge di moto viene "scalata" per essere completata nel tempo totale scelto in precedenza.

Da notare che nell'Eq. 3 manca il termine $\sqrt{2\pi\Delta s/|\ddot{q}_{i,min}|}$, in quanto assumiamo che i limiti imposti ai giunti siano **simmetrici**.

Nella Sez. 2 dedicata all'analisi cinematica si può vedere la legge oraria ottenuta, sia nello spazio di lavoro, sia in termini delle variabili di giunto.

1.5 Task 2

Il **task 2**, invece, risulta più complicato, in quanto bisogna innanzitutto descrivere matematicamente la traiettoria da eseguire ed in seguito trovare il modo di percorrerla senza eccedere i limiti cinematici e dinamici imposti inizialmente. Bisogna inoltre garantire continuità di velocità ed accelerazione, in modo da evitare l'insorgenza di fenomeni vibratori o di accelerazioni e forze/coppe addirittura impulsive.

1.5.1 Pianificazione della Traiettoria

La traiettoria da percorrere è stata inizialmente campionata manualmente nel piano *XY* con 17 punti (Fig. 4a), è stata espressa normalizzando i valori rispetto all'asse *x* mantenendo le proporzioni della forma e poi è stata interpolata mediante algoritmo **spline** bidimensionale (Fig. 4b). Successivamente la traiettoria ottenuta, composta da 100 punti, è stata prima scalata per giacere interamente nel *workspace* (Fig. 5) e poi traslata e ruotata nello spazio mediante una matrice di rototraslazione, in modo da ottenere i punti per i quali dovrà effettivamente passare l'*end-effector* del manipolatore (Fig. 6).

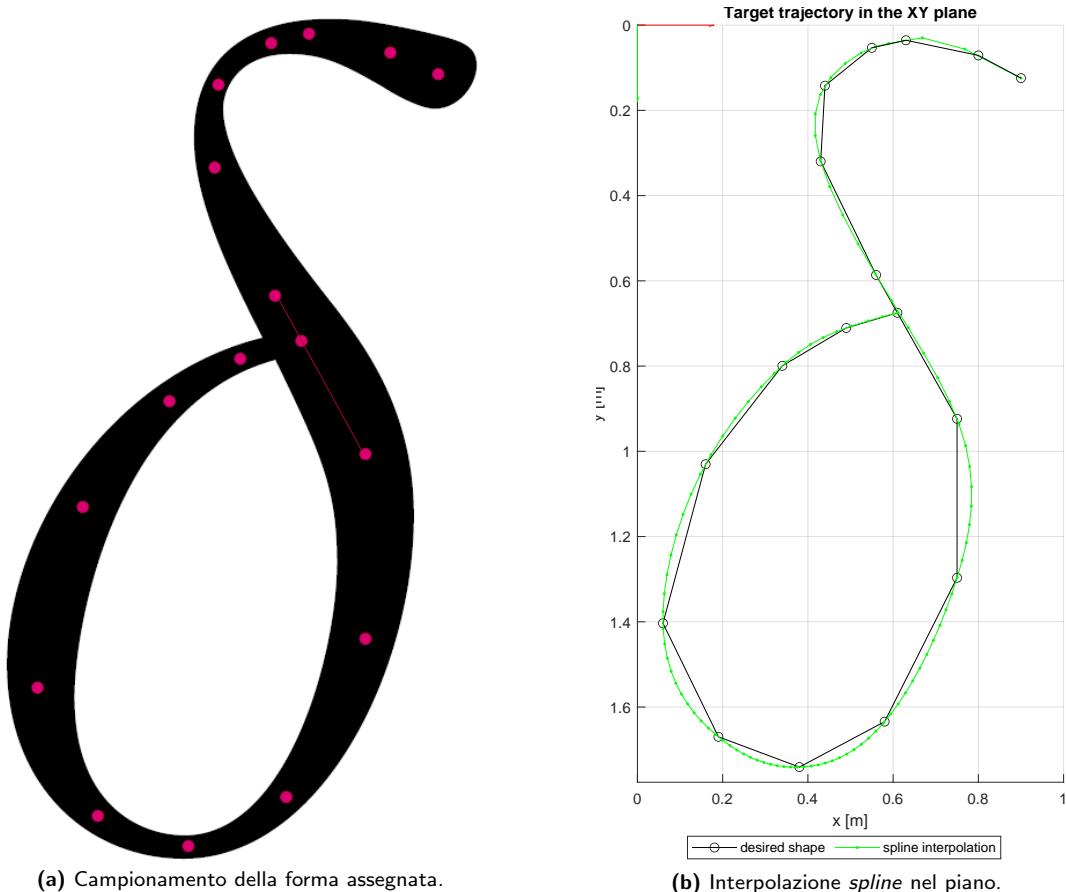


Fig. 4. Forma della traiettoria e interpolazione nel piano.

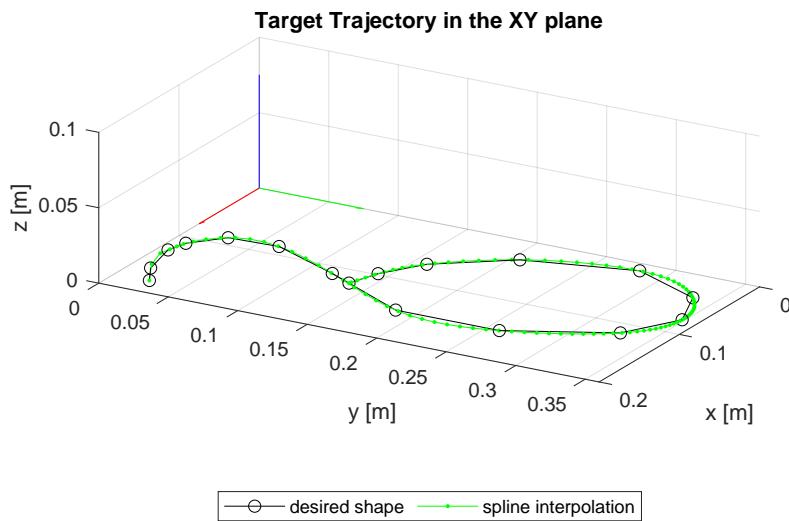


Fig. 5. Traiettoria nel piano scalata.

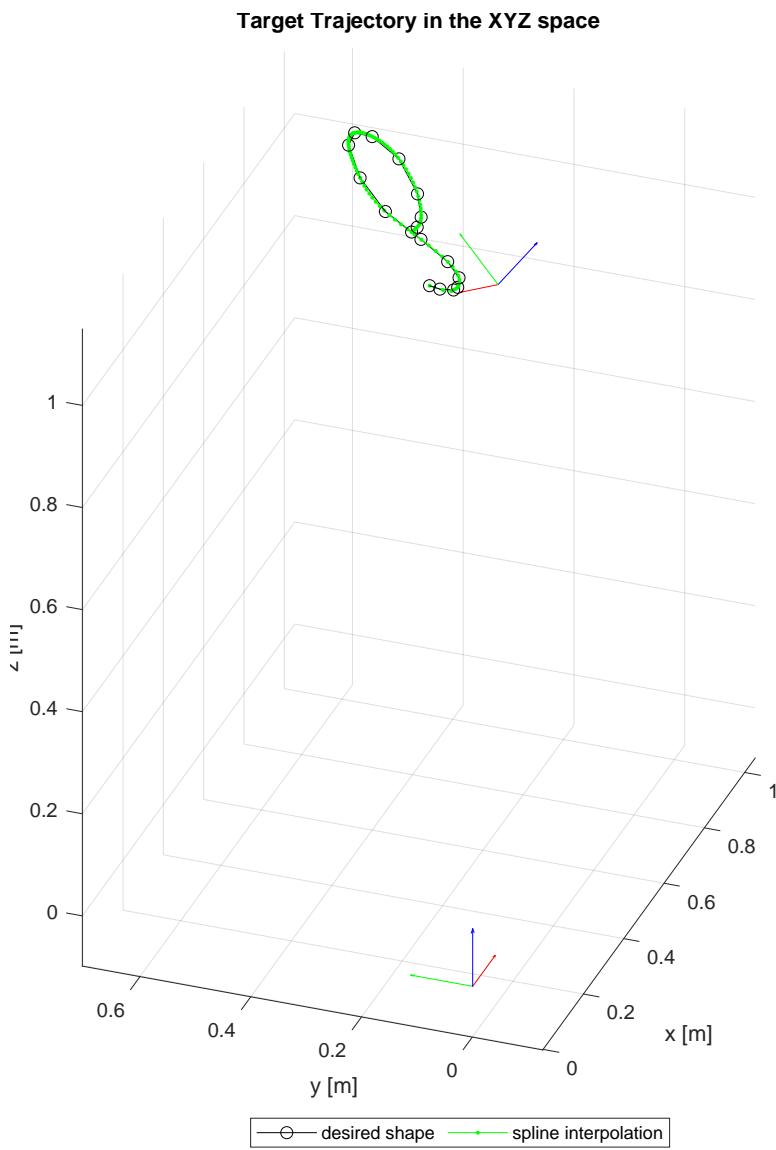


Fig. 6. Traiettoria finale nello spazio.

1.5.2 Calcolo Legge di Moto

La traiettoria è costituita da 100 punti da percorrere sequenzialmente. Ma con quale velocità ed accelerazione? Un modo efficace di determinare la legge di moto è scegliere una piccola spaziatura temporale dT tra i nodi come *first-guess* ed eseguire la seguente *routine*, che itera aumentando di volta in volta questo intervallo temporale, fino a che i limiti di velocità, accelerazione e coppia sono rispettati. All'interno del ciclo, una volta noto dT , è possibile derivare numericamente la posizione dell'*end-effector* per trovare velocità ed accelerazione, necessarie per applicare la cinematica inversa e calcolare così Q , \dot{Q} e \ddot{Q} e di conseguenza le forze/coppie richieste agli attuatori.

Di seguito il codice che implementa questo algoritmo:

```

1 % Compute sampling time in order to respect velocity and acceleration
2 % constraints of the joints
3 dT = 0.025; % Initial guess
4 Qp = [Inf; Inf; Inf]; Qpp = [Inf; Inf; Inf]; Fq = [Inf; Inf; Inf];
5 Q0 = num2cell(Q_initial_shape);
6
7 while sum([max(abs(Qp),[],2) > vel_limits([2 4 6])'; ...
8     max(abs(Qpp),[],2) > acc_limits([2 4 6])'; ...
9     max(abs(Fq),[],2) > torque_limits([2 4 6])']) > 0
10
11 xp_shape_diff = diff(S_shape(1,:),[],2)/dT;
12 yp_shape_diff = diff(S_shape(2,:),[],2)/dT;
13 zp_shape_diff = diff(S_shape(3,:),[],2)/dT;
14 Sp_shape_diff = [[0; 0; 0] [xp_shape_diff; yp_shape_diff;
zp_shape_diff]];
15
16 xpp_shape_diff = diff(Sp_shape_diff(1,:),[],2)/dT;
17 ypp_shape_diff = diff(Sp_shape_diff(2,:),[],2)/dT;
18 zpp_shape_diff = diff(Sp_shape_diff(3,:),[],2)/dT;
19 Spp_shape_diff = [[0; 0; 0] [xpp_shape_diff; ypp_shape_diff;
zpp_shape_diff]];
20
21 for i = 1:nPoints
22
23     % JOINT COORDINATES THROUGH INVERSE KINEMATICS
24     Qcell = inv_kin_numerical(num2cell(S_shape(1:3,i)),L,angle,Q0);
25     Q(:,i) = cell2mat(Qcell);
26     Q0 = num2cell(Q(:,i));
27
28     % CONVERT FROM SYMBOLIC TO NUMERICAL VALUES
29     J = double(subs(J_symb,Qsymb,[L'; angle; Q(:,i)]));
30
31     Qp(:,i) = J^-1*Sp_shape_diff(:,i);
32

```

```

33 Jp = double(subs(Jp_symb,Qp_symb,[L'; angle; Q(:,i); Qp(:,i)]));
34 ;
35 Qpp(:,i) = J^-1*(Spp_shape_diff(:,i)-Jp*Qp(:,i));
36
37 % CONVERT FROM SYMBOLIC TO NUMERICAL VALUES
38 Je = double(subs(Je_symb,Qe_symb,[L'; angle; Q(:,i); Gpos{2};
39 Gpos{3}; Gpos{4}; q1; q3]));
40 Jep = double(subs(Jep_symb,Qep_symb,[L'; angle; Q(:,i); Qp(:,i);
41 ; Gpos{2}; Gpos{3}; Gpos{4}; q1; q3]));
42
43 % ACTUATOR TORQUES
44 Fq(:,i) = (Je'*M*Je)*Qpp(:,i)+Je'*M*Jep*Qp(:,i)-Je'*Ftot;
45
46 end
47
48 if sum([max(abs(Qp),[],2) > vel_limits([2 4 6]); ...
49 max(abs(Qpp),[],2) > acc_limits([2 4 6]); ...
50 max(abs(Fq),[],2) > torque_limits([2 4 6])]) == 0
51 break
52 end
53
54 dT = 1.1*dT;
55
56 % Minimum sampling time (ROUNDED) to respect velocity and acceleration
57 % constraints of actuators
58 dT = ceil(dT*100);
59 dT = dT/100;

```

Quando il ciclo termina, il valore minimo di dT affinchè i limiti siano rispettati è ormai noto ed il ciclo vero e proprio può essere eseguito per calcolare tutte le quantità cinematiche e dinamiche a partire dai valori di S , \dot{S} e \ddot{S} .

Come per gli altri *task*, si rimanda alla Sez. 2 per vedere la legge oraria ottenuta.

2 Analisi Cinematica

Lo studio cinematico è stato svolto parallelamente seguendo due diversi approcci:

- studio della relazione tra coordinate di giunto e coordinate cartesiane mediante la matrice jacobiana e individuazione delle configurazioni singolari,
- descrizione matriciale delle relazioni tra terne di riferimento posizionate in punti opportuni del manipolatore secondo la metodologia proposta in [1].

Questa scelta ha permesso di confrontare i risultati dei due metodi, per mettere alla luce vantaggi e svantaggi di entrambi e per evidenziare eventuali errori nell'uno o nell'altro. Questa prima forma di debug dei risultati è stata accompagnata dal controllo dei risultati mediante differenziazione numerica, come descritto accuratamente al Par. 2.5.

2.1 Cinematica Diretta - Approccio Jacobiano

Lo studio della cinematica diretta del manipolatore si basa sulla scrittura di equazioni che leghino le coordinate nello spazio dei giunti a quelle nello spazio cartesiano, in modo tale che, note le prime, sia possibile ottenere la posizione del robot nello spazio. Più precisamente, è possibile scrivere le tre equazioni che descrivono la posizione del *end-effector* nello spazio lungo gli assi di riferimento del sistema posto alla base del robot, ma allo stesso tempo si può anche descrivere un numero arbitrario di punti, purché la loro posizione sia sempre riferita al medesimo sistema di riferimento.

Mediante l'utilizzo della matrice jacobiana, sarà poi possibile calcolare velocità ed accelerazione dei punti scelti.

2.1.1 Posizione

Denotando con Q il vettore delle m coordinate di giunto (i.e. i gradi di libertà del manipolatore) e con S le n coordinate nello spazio di lavoro, tali relazioni si possono esprimere mediante il sistema di equazioni:

$$S = F(Q) \quad (4)$$

dove

$$\begin{aligned} Q &= [q_1, q_2, q_j, \dots, q_m]^T \\ S &= [s_1, s_2, s_i, \dots, s_n]^T. \end{aligned}$$

Nel caso in esame, il vettore delle variabili di giunto è costituito dai seguenti elementi:

- q_1 : spostamento angolare del giunto rotoidale di base rispetto al proprio asse,
- q_2 : traslazione del giunto prismatico che collega il *link 1* al *link 2*,
- q_3 : rotazione del giunto rotoidale che collega il *link 2* al *link 3* rispetto al proprio asse,

da cui $Q = [q_1, q_2, q_3]^T$.

Il vettore delle coordinate cartesiane, invece, contiene la posizione (x, y, z) dei punti P_0 , P_{1a} , P_{1b} , P_{1c} , P_2 , P_3 evidenziati in Fig. 7:

$$S = [x_0, y_0, z_0, x_{1a}, y_{1a}, z_{1a}, x_{1b}, y_{1b}, z_{1b}, x_{1c}, y_{1c}, z_{1c}, x_2, y_2, z_2, x_3, y_3, z_3]^T,$$

dove i singoli termini sono definiti ricorsivamente per comodità:

$$\begin{aligned} P_0 : & \begin{cases} x_0 = 0 \\ y_0 = 0 \\ z_0 = 0 \end{cases} & P_{1c} : & \begin{cases} x_{1c} = x_{1b} + l_{1c} \cos(\alpha) \sin(q_1) \\ y_{1c} = y_{1b} - l_{1b} \cos(\alpha) \cos(q_1) \\ z_{1c} = z_{1b} + l_{1c} \sin(\alpha) \end{cases} \\ P_{1a} : & \begin{cases} x_{1a} = x_0 \\ y_{1a} = y_0 \\ z_{1a} = z_0 + l_{1a} \end{cases} & P_2 : & \begin{cases} x_2 = x_{1c} + (l_2 + q_2) \cos(\alpha) \sin(q_1) \\ y_2 = y_{1c} - (l_2 + q_2) \cos(\alpha) \cos(q_1) \\ z_2 = z_{1c} + (l_2 + q_2) \sin(\alpha) \end{cases} \\ P_{1b} : & \begin{cases} x_{1b} = x_{1a} + l_{1b} \cos(q_1) \\ y_{1b} = y_{1a} + l_{1b} \sin(q_1) \\ z_{1b} = z_{1a} \end{cases} & P_3 : & \begin{cases} x_3 = x_2 + l_3 \cos(\alpha + q_3) \sin(q_1) \\ y_3 = y_2 - l_3 \cos(\alpha + q_3) \cos(q_1) \\ z_3 = z_2 + l_3 \sin(\alpha + q_3) \end{cases} \end{aligned}$$

Se al posto della formulazione ricorsiva si utilizza la forma per esteso delle precedenti equazioni, si ottengono proprio le n funzioni f_i che costituiscono il sistema 4.

2.1.2 Velocità

Una volta note le espressioni delle posizioni dei punti di interesse, è possibile calcolare la loro velocità mediante la seguente formula:

$$\dot{S} = \frac{dS}{dt} = \frac{\partial F}{\partial Q} \frac{dQ}{dt} = J \dot{Q}, \quad (5)$$

dove il termine $J = \frac{\partial F}{\partial Q}$ è detto *matrice jacobiana* del sistema.

Di seguito l'espressione della matrice jacobiana nel caso del manipolatore oggetto di studio:

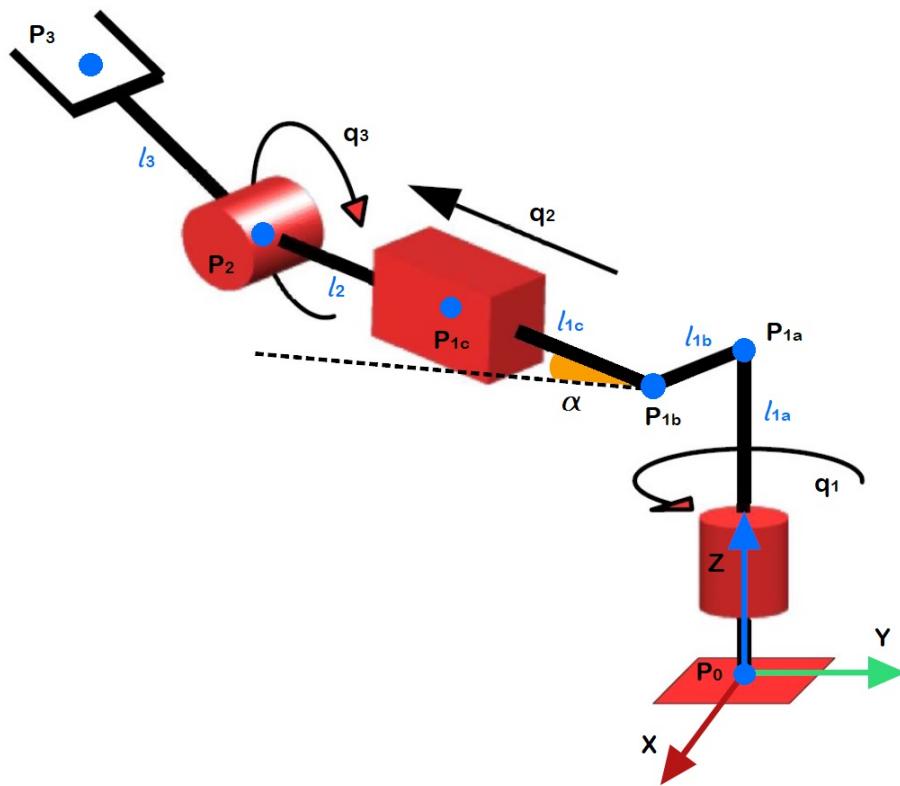


Fig. 7. Schema di principio della struttura del manipolatore ed elementi geometrici di rilievo.

$$J = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b}S_{q_1} & 0 & 0 \\ l_{1b}C_{q_1} & 0 & 0 \\ 0 & 0 & 0 \\ l_{1c}C_{q_1}C_\alpha - l_{1b}S_{q_1} & 0 & 0 \\ l_{1b}C_{q_1} + l_{1c}S_{q_1}C_\alpha & 0 & 0 \\ 0 & 0 & 0 \\ C_{q_1}C_\alpha(l_2 + q_2) - l_{1b}S_{q_1} + l_{1c}C_{q_1}C_\alpha & S_{q_1}C_\alpha & 0 \\ l_{1b}C_{q_1} + S_{q_1}C_\alpha(l_2 + q_2) + l_{1c}S_{q_1}C_\alpha & -C_{q_1}C_\alpha & 0 \\ 0 & S_\alpha & 0 \\ C_{q_1}C_\alpha(l_2 + q_2) - l_{1b}S_{q_1} + l_3C_{\alpha q_3}C_{q_1} + l_{1c}C_{q_1}C_\alpha & S_{q_1}C_\alpha & -l_3S_{\alpha q_3}S_{q_1} \\ l_{1b}C_{q_1} + S_{q_1}C_\alpha(l_2 + q_2) + l_3C_{\alpha q_3}S_{q_1} + l_{1c}S_{q_1}C_\alpha & -C_{q_1}C_\alpha & l_3S_{\alpha q_3}C_{q_1} \\ 0 & S_\alpha & l_3C_{\alpha q_3}, \end{bmatrix}$$

dove sono state adottate le seguenti abbreviazioni:

$$\begin{aligned} C_\beta &= \cos(\beta) & C_{\beta\gamma} &= \cos(\beta + \gamma) \\ S_\beta &= \sin(\beta) & S_{\beta\gamma} &= \sin(\beta + \gamma). \end{aligned} \quad (6)$$

2.1.3 Accelerazione

Per ottenere l'accelerazione dei punti è sufficiente derivare rispetto al tempo le singole velocità, ottenendo la seguente espressione:

$$\ddot{S} = \frac{d^2 S}{dt^2} = \frac{d}{dt} \frac{\partial F}{\partial Q} \frac{dQ}{dt} + \frac{\partial F}{\partial Q} \frac{d^2 Q}{dt^2} = J \ddot{Q} + \dot{J} \dot{Q}. \quad (7)$$

La derivata temporale della matrice jacobiana assume la seguente espressione nel caso in esame (avvalendosi ancora delle abbreviazioni 6):

$$\dot{J} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b} C_{q_1} \dot{q}_1 & 0 & 0 \\ -l_{1b} S_{q_1} \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b} C_{q_1} \dot{q}_1 - l_{1c} S_{q_1} C_\alpha \dot{q}_1 & 0 & 0 \\ l_{1c} C_{q_1} C_\alpha \dot{q}_1 - l_{1b} S_{q_1} \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \\ \dot{j}_{13,1} & C_{q_1} C_\alpha \dot{q}_1 & 0 \\ \dot{j}_{14,1} & S_{q_1} C_\alpha \dot{q}_1 & 0 \\ 0 & 0 & 0 \\ \dot{j}_{16,1} & C_{q_1} C_\alpha \dot{q}_1 & -l_3 S_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 C_{\alpha q_3} S_{q_1} \dot{q}_3 \\ \dot{j}_{17,1} & S_{q_1} C_\alpha \dot{q}_1 & l_3 C_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} S_{q_1} \dot{q}_1 \\ 0 & 0 & -l_3 S_{\alpha q_3} \dot{q}_3, \end{bmatrix}$$

dove

$$\begin{aligned} \dot{j}_{13,1} &= C_{q_1} C_\alpha \dot{q}_2 - l_{1b} C_{q_1} \dot{q}_1 - S_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) - l_{1c} S_{q_1} C_\alpha \dot{q}_1 \\ \dot{j}_{14,1} &= S_{q_1} C_\alpha \dot{q}_2 - l_{1b} S_{q_1} \dot{q}_1 + C_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) + l_{1c} C_{q_1} C_\alpha \dot{q}_1 \\ \dot{j}_{16,1} &= C_{q_1} C_\alpha \dot{q}_2 - l_{1b} C_{q_1} \dot{q}_1 - S_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) - l_3 C_{\alpha q_3} S_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} C_{q_1} \dot{q}_3 - l_{1c} S_{q_1} C_\alpha \dot{q}_1 \\ \dot{j}_{17,1} &= S_{q_1} C_\alpha \dot{q}_2 - l_{1b} S_{q_1} \dot{q}_1 + C_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) + l_3 C_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} S_{q_1} \dot{q}_3 + l_{1c} C_{q_1} C_\alpha \dot{q}_1. \end{aligned}$$

2.2 Cinematica Inversa - Approccio Jacobiano

Lo studio cinematico inverso consiste nel calcolo delle variabili di giunto Q che consentono di ottenere una certa posizione dell'*end-effector* $S_{grip} = [x_{grip}, y_{grip}, z_{grip}]^T$. In generale la soluzione non è unica e ciascuna di esse rappresenta una configurazione spaziale che può assumere il manipolatore.

2.2.1 Posizione

Nel caso in esame, il legame tra coordinate di giunto e cartesiane è espresso dal sistema

$$S_{grip} = F_{grip}(Q), \quad (8)$$

con

$$f_{grip,1} = l_{1b}C_{q_1} + l_{1c}C_\alpha S_{q_1} + (l_2 + q_2)C_\alpha S_{q_1} + l_3C_{\alpha q_3}S_{q_1},$$

$$f_{grip,2} = l_{1b}S_{q_1} - l_{1c}C_\alpha C_{q_1} - (l_2 + q_2)C_\alpha C_{q_1} - l_3C_{\alpha q_3}C_{q_1},$$

$$f_{grip,3} = l_{1a} + l_{1c}S_\alpha + (l_2 + q_2)S_\alpha + l_3S_{\alpha q_3},$$

che, risolto rispetto a q_1 , q_2 e q_3 , fornisce le espressioni seguenti:

$$q_1 = \pi - \arcsin \left(\frac{l_{1b}}{\sqrt{x_{grip}^2 + y_{grip}^2}} \right) - \text{atan2}(x, y)$$

$$q_{3,1} = \arcsin(A)$$

$$q_{3,2} = \begin{cases} \pi - \arcsin(A) & \text{se } A \geq 0 \\ -\pi - \arcsin(A) & \text{se } A < 0 \end{cases}$$

$$\text{con } A = \begin{cases} \frac{1}{l_3} [-x_{grip}S_\alpha + (z_{grip} - l_{1a})C_\alpha] & \text{se } q_1 = \pi/2 + 2k\pi \\ \frac{1}{l_3} [x_{grip}S_\alpha + (z_{grip} - l_{1a})C_\alpha] & \text{se } q_1 = -\pi/2 + 2k\pi \\ \frac{1}{l_3} \left[\frac{y_{grip} - l_{1b}S_{q_1}}{C_{q_1}} S_\alpha + (z_{grip} - l_{1a})C_\alpha \right] & \text{se } q_1 \neq \pm\pi/2 + k\pi \end{cases}$$

$$q_2 = \frac{z_{grip} - l_{1a} - l_3S_{\alpha q_3}}{S_\alpha} - l_{1c} - l_2.$$

I valori di $k \in \mathbb{Z}$ in generale possono essere qualunque, tuttavia in questo caso assumono valore 0 a causa del limite

$$q_1 \in [-\pi; \pi]$$

sull'attuatore del giunto di base.

Si nota infine che il manipolatore ammette **due soluzioni**, rappresentate dalle coppie $(q_{3,1}, q_2)$ e $(q_{3,2}, q_2)$. In Fig. 8 si può vedere una di queste coppie di soluzioni, che si ripetono per qualsiasi valore di q_1 . La soluzione è singola, tuttavia, nei punti in cui la coordinata di giunto q_2 raggiunge i suoi valori massimo e minimo.

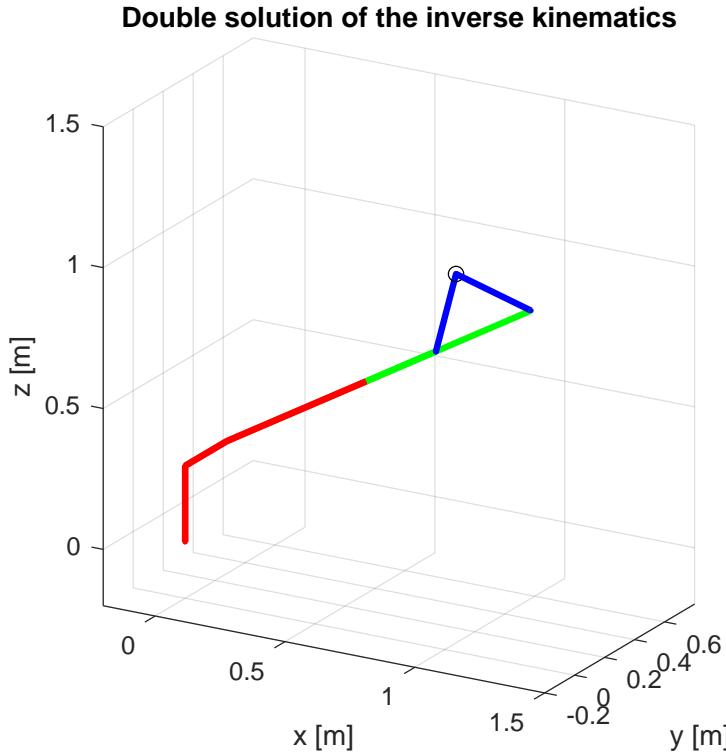


Fig. 8. Un esempio di duplice soluzione. Rosso: *link 1*, Verde: *link 2*, Blu: *link 3*.

Un approccio alternativo si basa sulla risoluzione numerica del sistema 8. Questa strategia è in genere meno complicata da implementare rispetto a svolgere analiticamente calcoli che talvolta possono essere molto complessi; va evidenziato tuttavia che la routine numerica riesce a trovare una sola soluzione, a seconda del punto iniziale da cui parte il solutore.

Diversi metodi possono essere implementati, tra cui il più tradizionale è quello di *Newton-Raphson*, che si basa sull'utilizzo della matrice jacobiana per linearizzare la relazione tra Q e S_{grip} . La scelta, tuttavia, è ricaduta su un altro metodo, implementato dal solutore *fsolve* di MATLAB. Tale *routine* è in grado di risolvere sistemi non lineari attraverso l'algoritmo *Trust-Region-Dogleg* [2], con la sola conoscenza del dato iniziale x_0 . Di seguito viene riportato il codice della funzione:

```

1 function Q = inv_kin_numerical(S,L,angle,Q0)
2
3     fun = @(x) gripper_coord(S,L,angle,x);
4     opt = optimset('Display','off');
5     Q = fsolve(fun,[Q0{:}]',opt);
6     Q = num2cell(Q);
7
8 end
9
10 function F = gripper_coord(S,L,angle,t)
11
12     l1a = L(1); l1b = L(2); l1c = L(3); l2 = L(4); l3 = L(5);
13
14     x = S{1}; y = S{2}; z = S{3};
15
16     F(1) = x-l1b*cos(t(1))-((l1c+l2+t(2))*cos(angle)+l3*cos(angle+t(3))
17     )*sin(t(1));
18     F(2) = y-l1b*sin(t(1))+((l1c+l2+t(2))*cos(angle)+l3*cos(angle+t(3))
19     )*cos(t(1));
20     F(3) = z-l1a-(l1c+l2+t(2))*sin(angle)-l3*sin(angle+t(3));
21
22 end

```

2.2.2 Velocità e Accelerazione

Il calcolo di velocità ed accelerazione per le coordinate nello spazio dei giunti è immediato:

$$\dot{Q} = J^{-1}\dot{S} \quad (9)$$

$$\ddot{Q} = J^{-1} \left(\ddot{S} - J\dot{Q} \right). \quad (10)$$

Questo calcolo richiede però che la matrice jacobiana sia invertibile, condizione vera in generale, ma non per le *configurazioni singolari*, per le quali il determinante di tale matrice è nullo. A livello numerico, anche un determinante pressoché nullo darà origine a una singolarità, dovuta a un numero di condizionamento della matrice molto elevato.

2.2.3 Configurazioni Singolari

Proprio l'annullamento del determinante della matrice jacobiana (i.e. lo *jacobiano*) è la condizione da imporre per trovare le configurazioni singolari del manipolatore. Questo metodo può essere accompagnato da ragionamenti di tipo geometrico: una configurazione è singolare quando essa è il risultato di due o più soluzioni che coincidono. In tal caso esisteranno alcune particolari direzioni nelle quali la pinza non potrà muoversi, poiché la componente della sua velocità lungo quella direzione sarà sempre nulla, qualsiasi sia la velocità dei motori. A livello

dinamico, invece, in queste direzioni il manipolatore potrà esercitare forze elevate con ridotto sforzo degli attuatori.

La struttura cinematica del manipolatore, prima ancora che l'annullamento del determinante, permette di ricavare le configurazioni singolari: come si vede in Fig. 9, sono singolari tutte le configurazioni per cui $q_3 = \pm\pi/2$, poiché in quel caso le due soluzioni del manipolatore (Fig. 8) coincidono. In tali configurazioni, per giunta, non sono possibili velocità della pinza radiali rispetto al link 3.

Examples of singular configurations of the manipulator

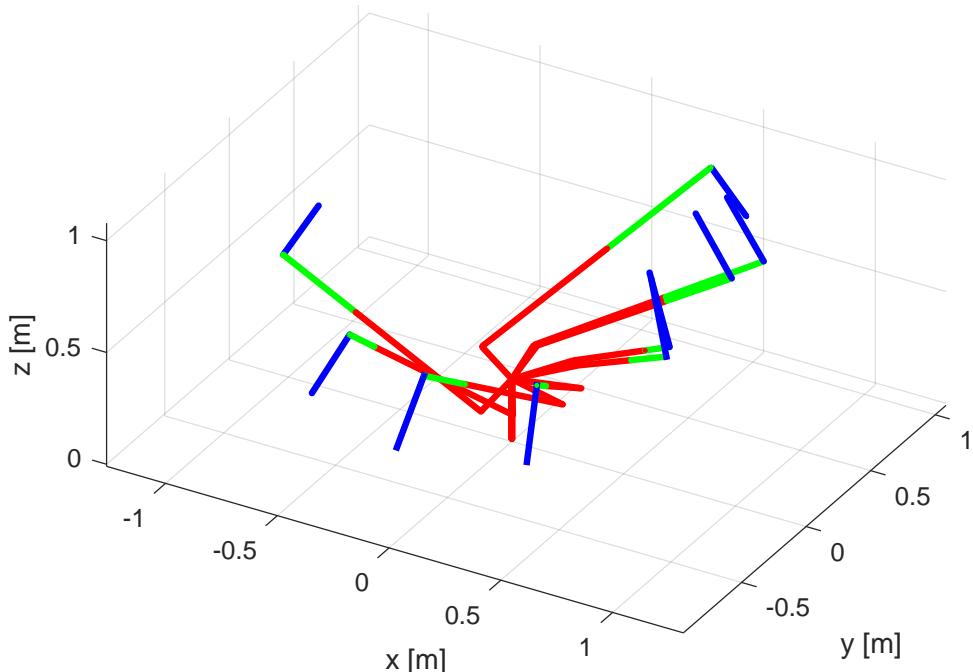


Fig. 9. Alcune configurazioni singolari del manipolatore ottenute annullando lo jacobiano.

Le configurazioni ottenute in Fig. 9 derivano dall'annullamento per via numerica dello jacobiano: trattandosi di un'equazione altamente non lineare in q_1, q_2, q_3 , si è deciso di impostare un problema di minimizzazione di una funzione di costo, (i.e. il valore assoluto dello jacobiano) per trovare i parametri ottimali ((q_1, q_2, q_3)) che facessero tendere a zero tale funzione. Accontentandosi di avere un riscontro alle considerazioni teoriche esposte poco sopra, ci si è limitati a trovare parametri localmente ottimali, facilmente individuabili da un **algoritmo genetico** con i seguenti parametri:

variabili	dimensioni popolazione	fitness function	massimo numero generazioni
3	50 individui	$\det(J)$	$100 \cdot n_{variabili}$

Dopo molteplici *run* della procedura di ottimizzazione, si è potuto riscontrare che mentre le coordinate q_1 e q_2 continuavano a cambiare valore, la coordinata q_3 era sempre pari a $\pm\pi/2$, evidenziando che quello è proprio il valore per cui lo jacobiano tende ad annullarsi, dando origine a configurazioni singolari. Questi risultati si possono vedere nella Tab. 2.2.3.

$q_1 \text{ [}\pi \text{ rad]}$	$q_2 \text{ [m]}$	$q_3 \text{ [}\pi \text{ rad]}$	$\det(J)$
0.09456	-0.65156	-0.50112	8.5042e-05
1.0584	-0.41838	-0.50028	3.1831e-05
-0.32368	-1.0318	0.49634	0.00039428
-1.9323	-0.8284	0.50141	0.00031225
-1.69	-0.60361	0.50135	5.2265e-05
-0.70431	-0.38075	0.5	2.9501e-08
-0.55538	-0.77737	0.50014	1.3874e-05
0.67569	-0.89032	-0.50705	3.2385e-05
-0.97597	-0.87133	0.49996	1.0191e-06
-1.5762	-0.75659	0.49996	4.6635e-06
-1.4744	-0.87021	-0.49997	6.9852e-07
1.3523	-0.56257	0.49622	5.0427e-06
1.0086	-0.55453	-0.49893	1.6011e-06
-1.5414	-0.75189	-0.50612	0.00072165
0.42919	-0.65052	0.49954	3.4993e-05
-0.66111	-0.66007	-0.49974	2.2124e-05
-1.3402	-0.74431	-0.50159	0.00023979
1.5259	-0.5267	-0.49873	3.0565e-05
1.9644	-0.66526	0.49964	3.1775e-05
0.077287	-0.98251	-0.49944	3.6152e-05
-1.491	-0.95863	-0.49704	0.0001281

2.3 Cinematica diretta - approccio matriciale

Parallelamente all'utilizzo dello Jacobiano, si è svolto lo studio di posizione, velocità e accelerazione del manipolatore tramite matrici omogenee 4x4. Dopo aver posizionato i sistemi di riferimento per ciascun link e dopo averne ricavato i parametri di Denavit - Hartenberg, sono state calcolate le matrici di posizione, velocità e accelerazione.

2.3.1 Sistemi di riferimento e parametri D&H

In Figura 10 sono riportati i sistemi di riferimento scelti secondo la convenzione di Denavit e Hartenberg, che permette di ottenere in modo quasi automatico le matrici M, W e H, con cui poter definire la cinematica diretta dei robot seriali.

Tale metodologia parte dal presupposto che ogni link deve avere un proprio sistema di riferimento posizionato nel giunto a valle. Per un corretto orientamento è necessario seguire i seguenti accorgimenti:

- l'asse z di ciascun *link* deve coincidere con l'asse di movimento del giunto a valle.
- l'asse x deve essere ortogonale e intersecante l'asse z del sistema di riferimento precedente.
- l'asse y deve formare una terna destra.

Inoltre, è importante sottolineare come il giunto prismatico, consenta di posizionare O_1 arretrato rispetto al giunto effettivo, in modo tale che il proprio asse x_1 intersechi l'asse z_0 .

Dai sistemi di riferimento è quindi possibile ricavare i parametri di D&H che descrivono il manipolatore, riportati in Tabella 2.3.1:

n.link	giunto	θ	d	a	ϕ
1	R	q_1	d_1	a_1	$\pi/2 - \alpha$
2	P	$\pi/2$	$q_2 + d_2$	0	$-\pi/2$
3	R	$q_3 - \pi/2$	0	a_3	$-\pi/2$

dove:

- θ (*rotazione*): angolo individuato dagli assi x_{i-1} e x .
- ϕ (*torsione*): angolo formato da z_{i-1} e z .
- d (*offset*): componente dello spostamento da O_{i-1} a O_i lungo z_{i-1} .
- a (*lunghezza dell'elemento*): componente dello spostamento da O_{i-1} a O_i lungo x_i .

2.3.2 Posizione

Sostituendo i parametri sopra calcolati nell'espressione generica per le matrici di Denavit - Hartenberg si ricavano le seguenti matrici di posizione relative tra i link:

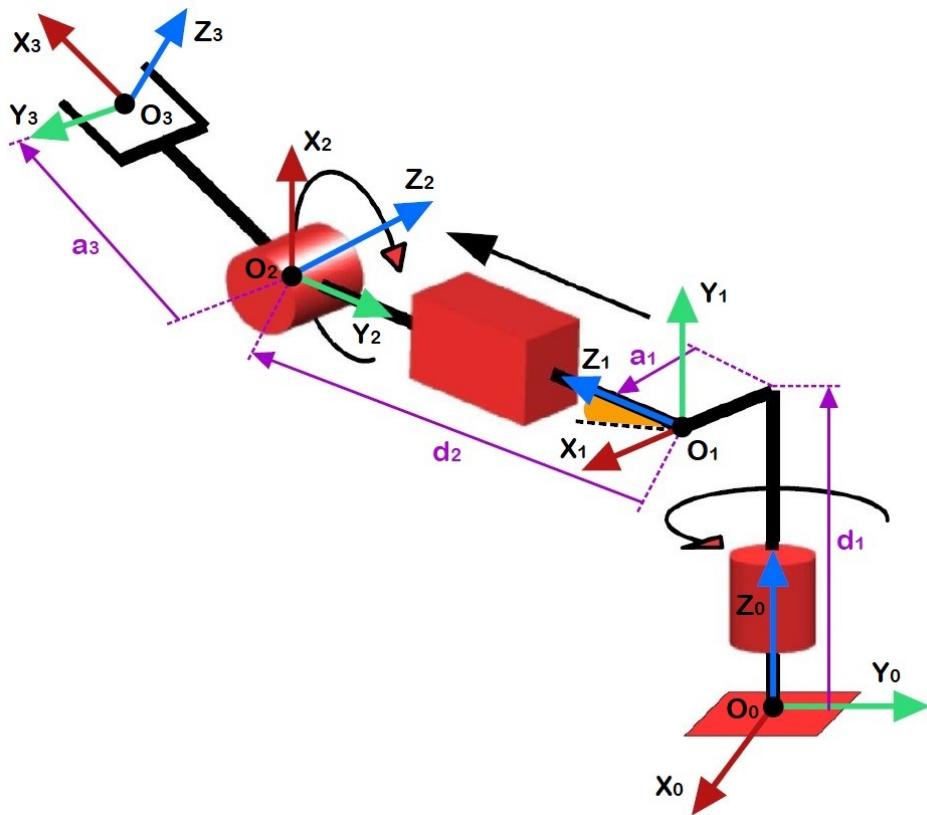


Fig. 10. Terne di riferimento posizionate secondo la convenzione di Denavit e Hartenberg

$$M_{01} = \begin{bmatrix} \cos(q_1) & -\frac{\sin(q_1)}{2} & \frac{\sqrt{3}\sin(q_1)}{2} & \frac{139\cos(q_1)}{500} \\ \sin(q_1) & \frac{\cos(q_1)}{2} & -\frac{\sqrt{3}\cos(q_1)}{2} & \frac{139\sin(q_1)}{500} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{27}{100} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & q_2 + \frac{73}{100} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{23} = \begin{bmatrix} \cos\left(q_3 - \frac{\pi}{2}\right) & 0 & -\sin\left(q_3 - \frac{\pi}{2}\right) & \frac{299\cos\left(q_3 - \frac{\pi}{2}\right)}{1000} \\ \sin\left(q_3 - \frac{\pi}{2}\right) & 0 & \cos\left(q_3 - \frac{\pi}{2}\right) & \frac{299\sin\left(q_3 - \frac{\pi}{2}\right)}{1000} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Da cui si può ottenere la matrice di posizione dal sistema di riferimento dell'*end-effector* rispetto al sistema di riferimento di base (0) tramite un semplice prodotto matriciale:

$$M_{03} = M_{01} \cdot M_{12} \cdot M_{23} \quad (11)$$

2.3.3 Velocità e accelerazioni

Per il calcolo delle matrici di velocità e accelerazione è necessario innanzitutto definire le tre matrici L , che identificano l'accoppiamento (tipo di giunto e asse di movimento).

$$L_{01_{(0)}} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad L_{12_{(1)}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad L_{23_{(2)}} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Segue poi il calcolo delle velocità nel sistema di riferimento di ciascun *link* secondo la relazione:

$$W_{i-1,i(i-1)} = L_{i-1,i(i-1)} \cdot \dot{q}_i \quad (12)$$

che vengono poi riportate al sistema di riferimento (0):

$$W_{i-1,i(0)} = M_{0,i-1} \cdot W_{i-1,i(i-1)} \cdot M_{i-1,0} = M_{0,i-1} \cdot W_{i-1,i(i-1)} \cdot M_{0,i-1}^{-1} \quad (13)$$

e sommate tra di loro per ottenere la matrice di velocità di ciascun *link* rispetto al sistema di riferimento di base secondo il *Teorema di composizione delle velocità*:

$$W_{0,i(0)} = W_{0,i-1(0)} + W_{i-1,i(0)}. \quad (14)$$

Per le accelerazioni il discorso è piuttosto simile. Si calcolano infatti le matrici H per ciascun *link* rispetto al proprio sistema di riferimento 15, ciascuna viene riportata al sistema di riferimento di base 3.2, per poi ottenere la matrice H per ciascun link rispetto al sistema di riferimento assoluto con il Teorema di Coriolis 17:

$$H_{i-1,i(i-1)} = \dot{W}_{i-1,i(i-1)} + W_{i-1,i(i-1)}^2 = L_{i-1,i(i-1)} \cdot \ddot{q}_i + L_{i-1,i(i-1)}^2 \cdot \dot{q}_i^2 \quad (15)$$

$$H_{i-1,i(0)} = M_{0,i-1} \cdot H_{i-1,i(i-1)} \cdot M_{0,i-1}^{-1} \quad (16)$$

$$H_{0,i(0)} = W_{0,i-1(0)} + W_{i-1,i(0)} + 2W_{0,i-1(0)} \cdot W_{i-1,i(0)} \quad (17)$$

Per valutare la correttezza delle matrici W ottenute usando le matrici L si sono calcolate le matrici di velocità anche tramite la relazione 18:

$$W = \dot{M} \cdot M^{-1} \quad (18)$$

ottenendo come risultato le seguenti matrici:

$$W_{01(0)} = \begin{bmatrix} 0 & -q_1 & 0 & 0 \\ q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad W_{12(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dot{q}_2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad W_{23(2)} = \begin{bmatrix} 0 & -q_3 & 0 & 0 \\ \dot{q}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2.4 Cinematica Inversa - Approccio Matriciale

2.4.1 Posizione

Per ricavare i valori delle posizioni nello spazio dei giunti (q_1, q_2, q_3) partendo dai valori di posizione nello spazio del *gripper* (x, y, z) si è utilizzata la relazione 19:

$$P_{e(0)} = M_{03} \cdot P_{e(3)} \quad (19)$$

dove $P_{e(3)}$ è il vettore posizione in coordinate omogenee del centro della pinza rispetto al sistema di riferimento e, dato che l'origine O_3 coincide con il centro della pinza, si ottiene:

$$P_{e(3)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (20)$$

Il sistema che si ottiene uguagliando riga per riga l'equazione 19 è stato risolto analiticamente in analogia con quanto riportato nella sezione 2.2.1, ottenendo i medesimi risultati.

2.4.2 Velocità

Moltiplicando la matrice $W_{03(0)}$ per la posizione dell'*end-effector* rispetto al sistema di riferimento assoluto 19 si ottengono le velocità del centro della pinza nel sistema di riferimento assoluto 23.

$$P_{0e} = \begin{bmatrix} \frac{139 \cos(q_1)}{500} - \frac{299 \sin(q_1) \sin(q_3)}{2000} + \frac{73 \sqrt{3} \sin(q_1)}{200} + \frac{\sqrt{3} q_2 \sin(q_1)}{2} + \frac{299 \sqrt{3} \cos(q_3) \sin(q_1)}{2000} \\ \frac{139 \sin(q_1)}{500} + \frac{299 \cos(q_1) \sin(q_3)}{2000} - \frac{73 \sqrt{3} \cos(q_1)}{200} - \frac{\sqrt{3} q_2 \cos(q_1)}{2} - \frac{299 \sqrt{3} \cos(q_1) \cos(q_3)}{2000} \\ \frac{q_2}{2} + \frac{299 \cos(q_3 - \frac{\pi}{3})}{1000} + \frac{127}{200} \\ 1 \end{bmatrix} \quad (21)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 0 \end{bmatrix} = W_{03(0)} \cdot P_{e(0)} \quad (22)$$

Uguagliando riga per riga e risolvendo in modo simbolico il sistema, si ricavano le espressioni delle velocità ai giunti in funzione delle posizioni dei giunti e della velocità del *gripper* nel sistema di riferimento assoluto (qui non riportate per esteso):

$$\dot{q}_1 = d(q_1, q_2, q_3, \dot{x}, \dot{y}, \dot{z})$$

$$\dot{q}_2 = g(q_1, q_2, q_3, \dot{x}, \dot{y}, \dot{z})$$

$$\dot{q}_3 = h(q_1, q_2, q_3, \dot{x}, \dot{y}, \dot{z})$$

2.4.3 Accelerazione

Un discorso simile vale per il calcolo delle accelerazioni dei giunti, note \ddot{x} , \ddot{y} e \ddot{z} . Una volta definito, infatti, il sistema di equazioni dato da:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ 0 \end{bmatrix} = H_{03(0)} \cdot P_{e(0)} \quad (23)$$

e risolvendolo simbolicamente si ottengono tre funzioni delle accelerazioni ai giunti in funzione delle posizioni e delle velocità dei giunti e delle accelerazioni del *gripper* nel sistema di riferimento assoluto:

$$\ddot{q}_1 = p(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3, \ddot{x}, \ddot{y}, \ddot{z})$$

$$\ddot{q}_2 = r(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3, \ddot{x}, \ddot{y}, \ddot{z})$$

$$\ddot{q}_3 = s(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3, \ddot{x}, \ddot{y}, \ddot{z})$$

2.5 Risultati e Debug

Di seguito vengono presentati i risultati dell'analisi cinematica: gli approcci, jacobiano e matriciale, forniscono risultati identici, indice di una probabile correttezza dei metodi. Viene tuttavia eseguita anche una verifica per via numerica, derivando le posizioni prima una e poi due volte per vedere se i valori coincidono rispettivamente con la velocità e l'accelerazione calcolate analiticamente. Questo procedimento viene ripetuto sia per le coordinate del *gripper*, sia per le variabili di giunto.

Dai grafici si può apprezzare come i limiti su escursione dei giunti, velocità e accelerazioni degli attuatori (riportati nel Par. 1.2) siano sempre rispettati in tutte e tre i task.

2.5.1 Task 1

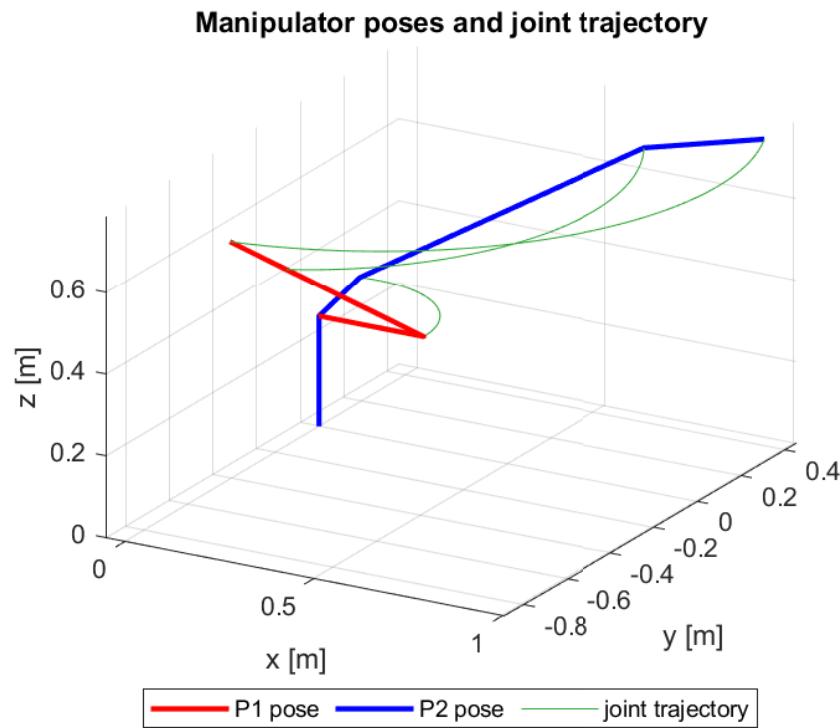


Fig. 11. Traiettoria delle estremità dei singoli link.

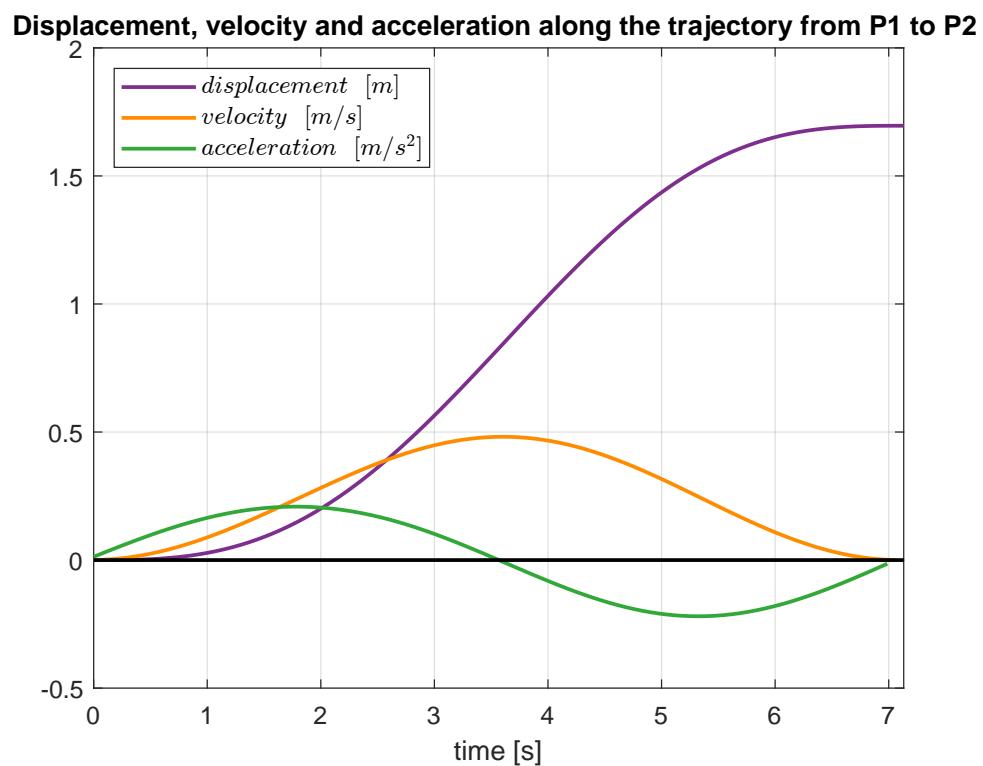


Fig. 12. Spostamento, velocità e accelerazione lungo l'ascissa curvilinea.

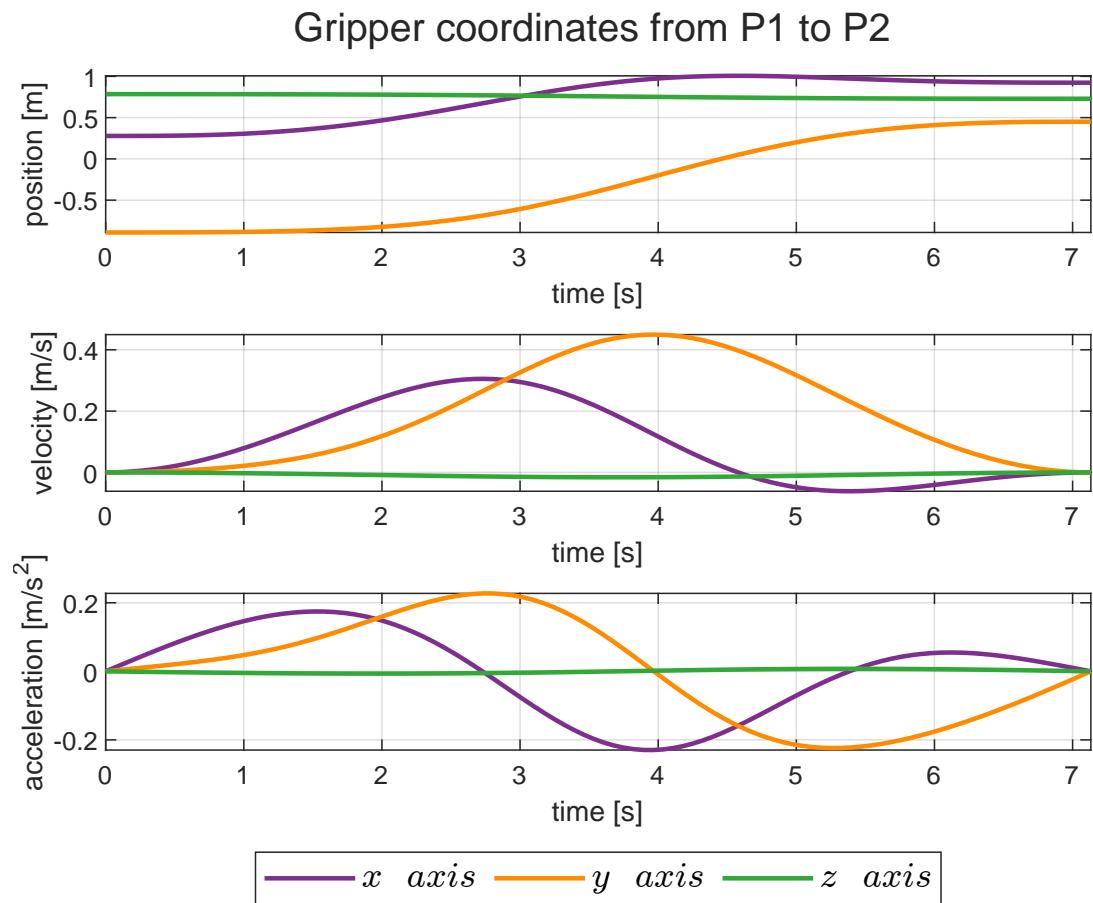


Fig. 13. Coordinate dell'end-effector.

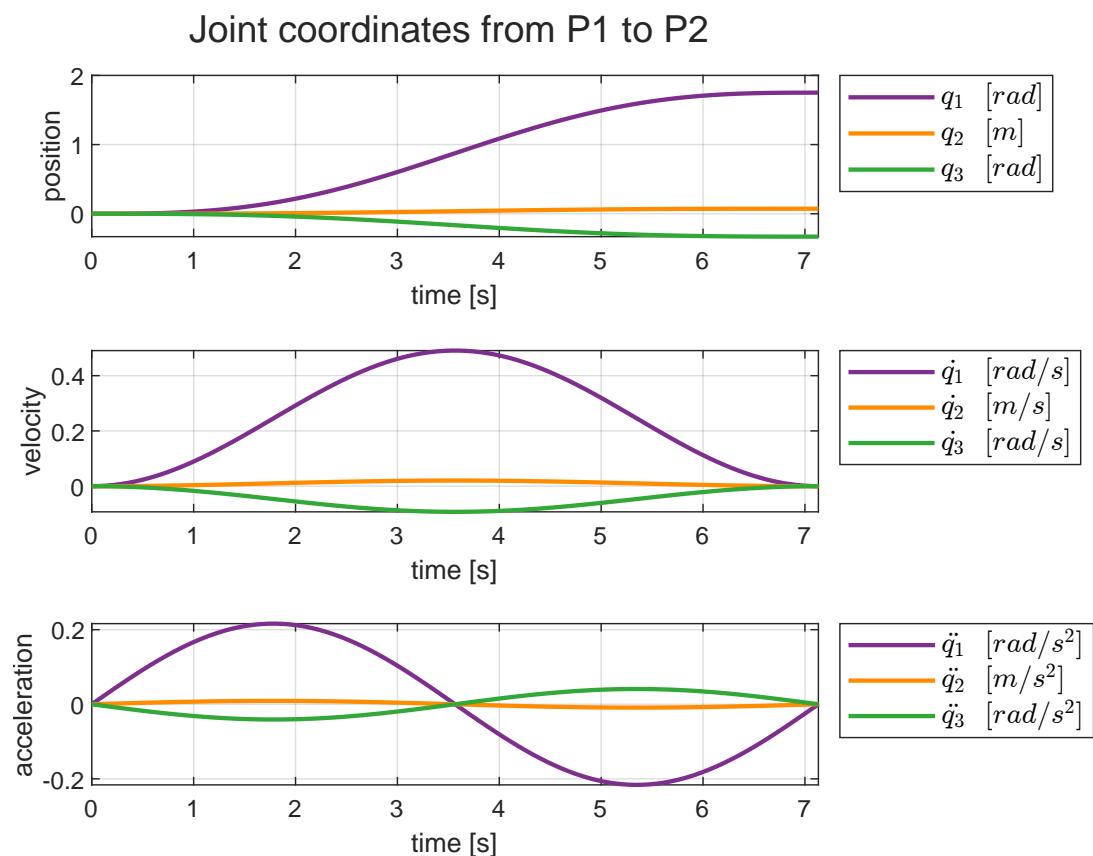


Fig. 14. Variabili di giunto.

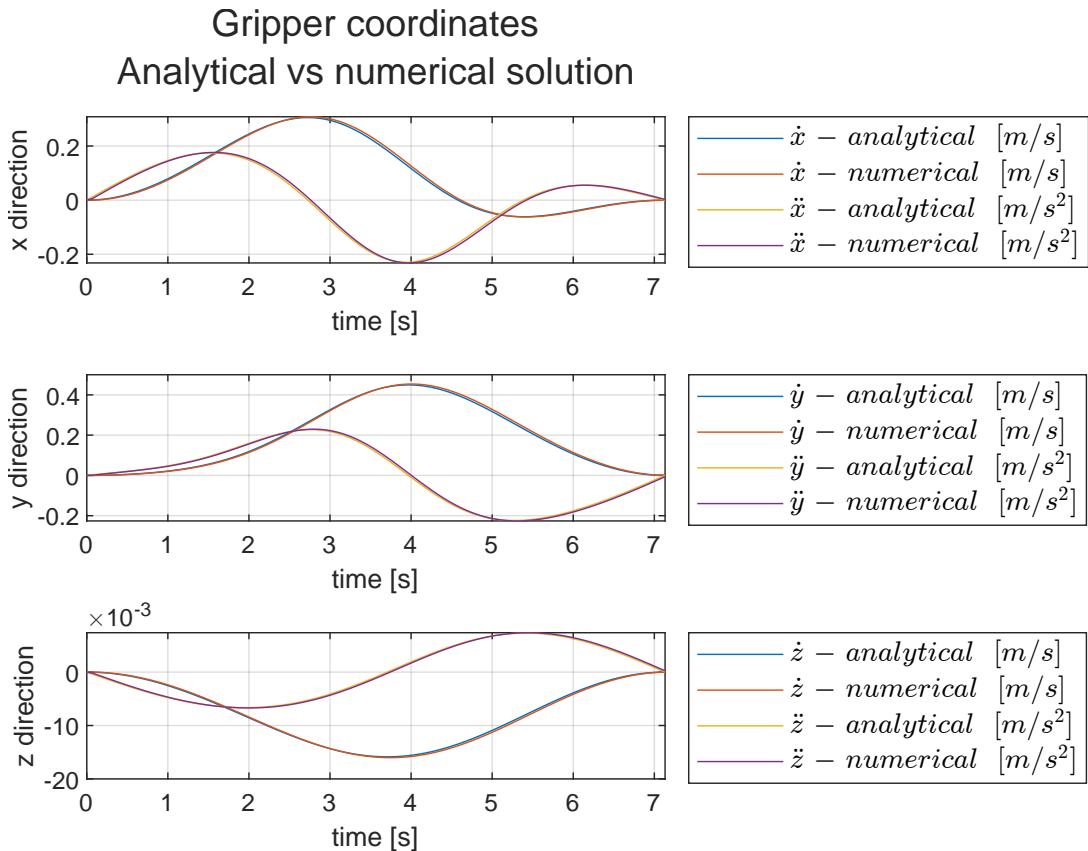


Fig. 15. Confronto tra velocità e accelerazioni dell'*end-effector* ricavate analiticamente e per derivazione.

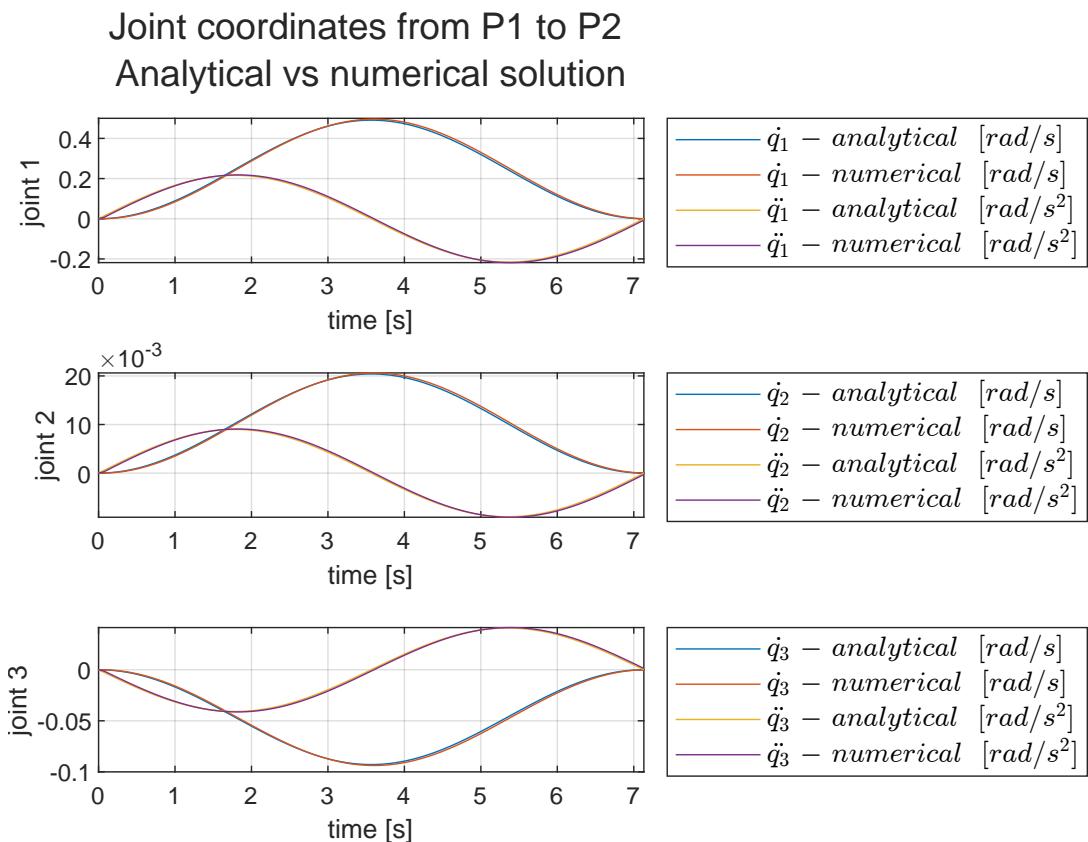


Fig. 16. Confronto tra velocità e accelerazioni dei giunti ricavate analiticamente e per derivazione.

2.5.2 Task 2

Manipulator poses and joint trajectory

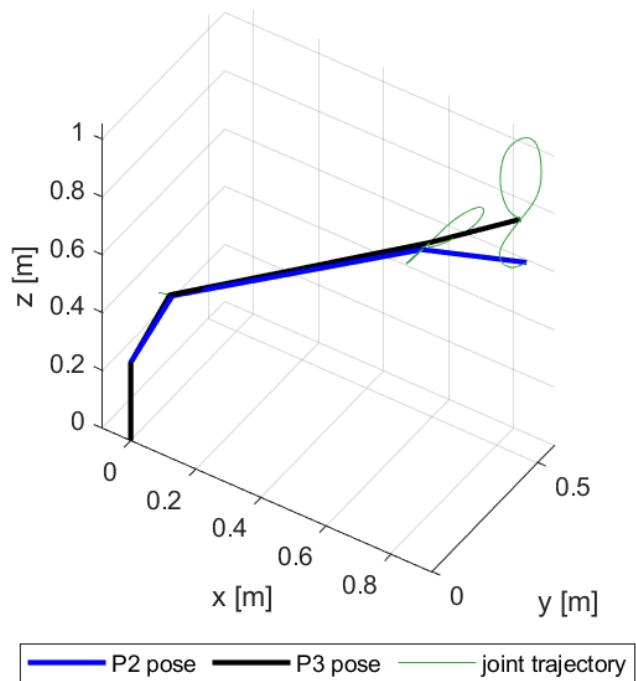


Fig. 17. Traiettoria delle estremità dei singoli link.

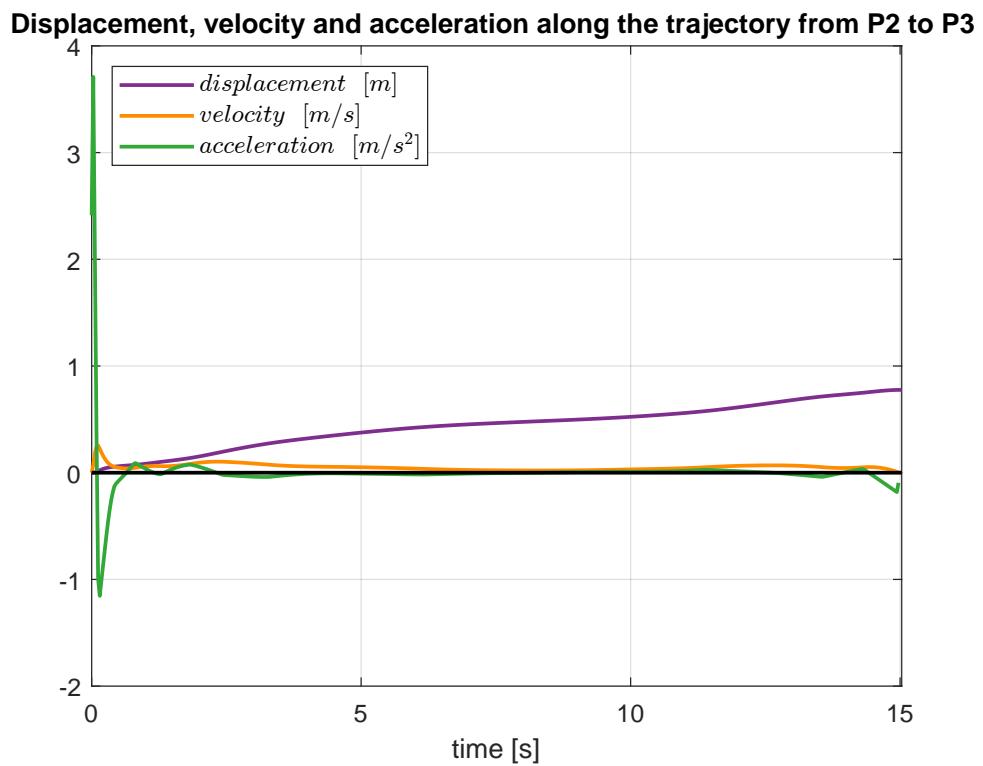


Fig. 18. Spostamento, velocità e accelerazione lungo l'ascissa curvilinea.

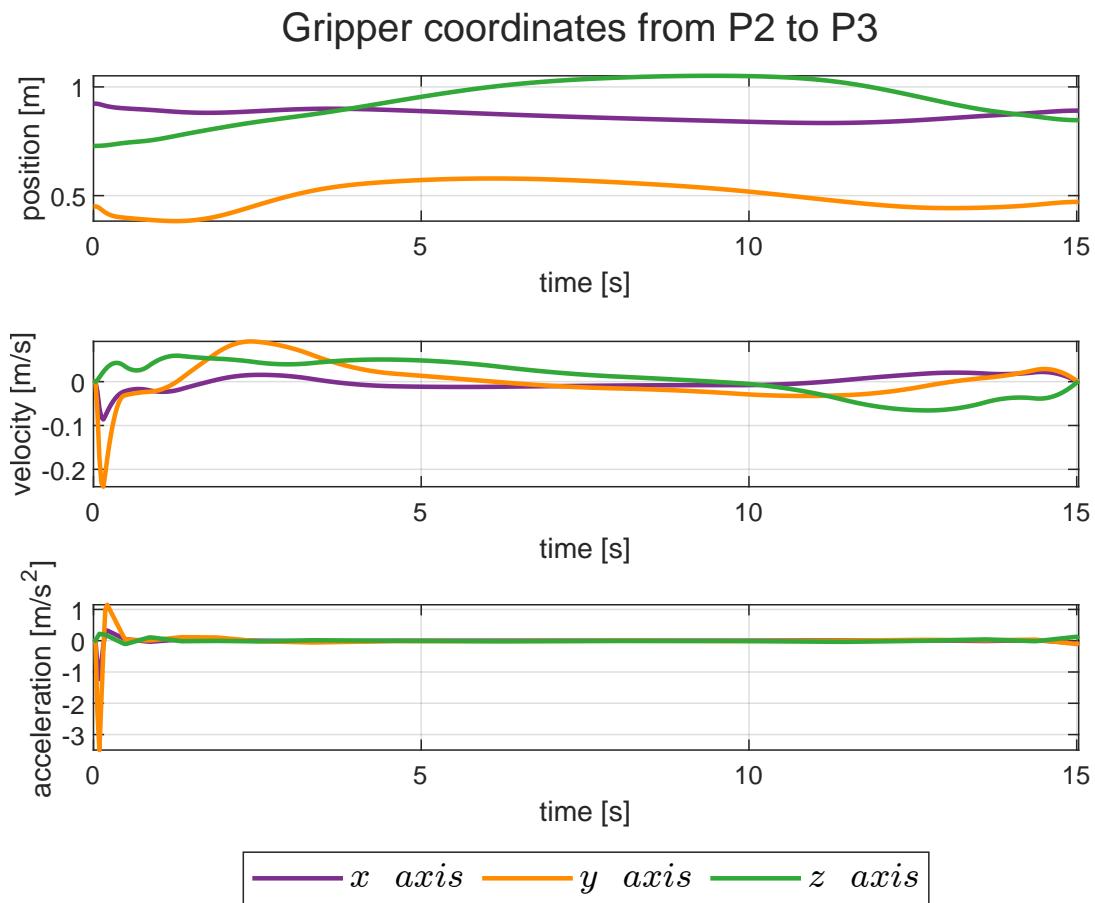


Fig. 19. Coordinate dell'end-effector.

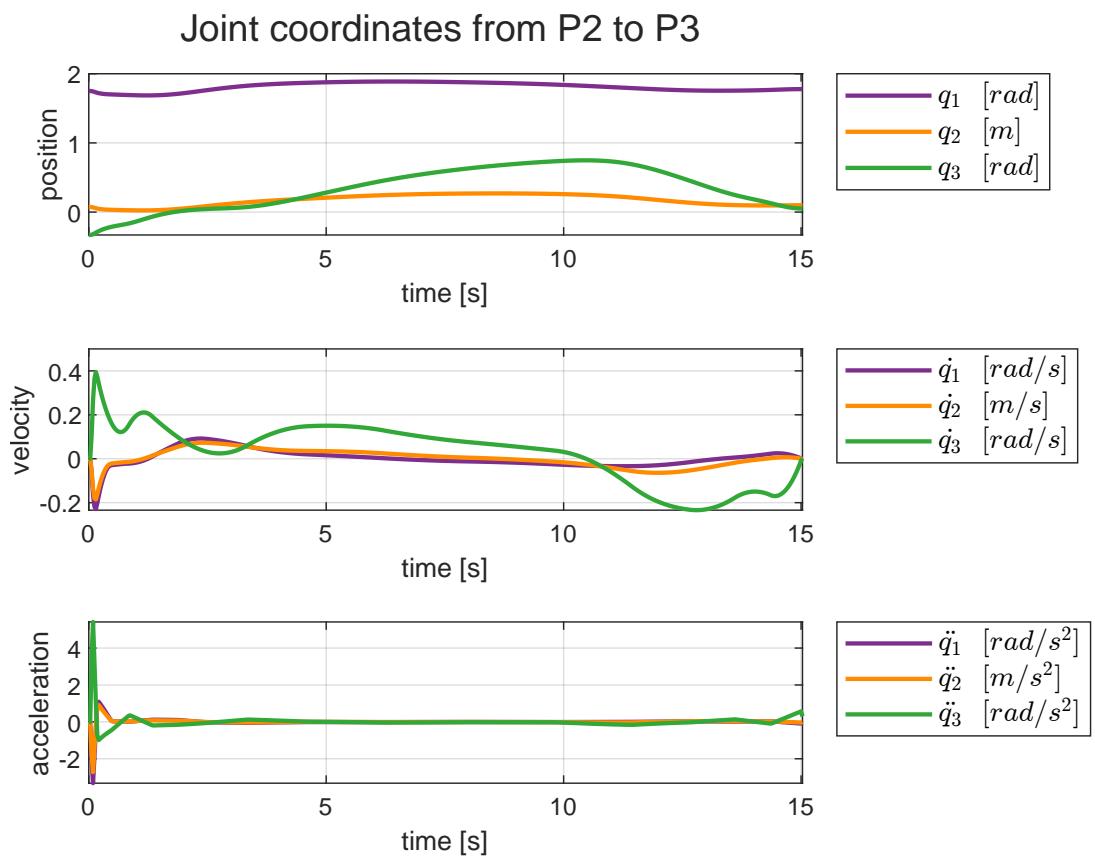


Fig. 20. Variabili di giunto.

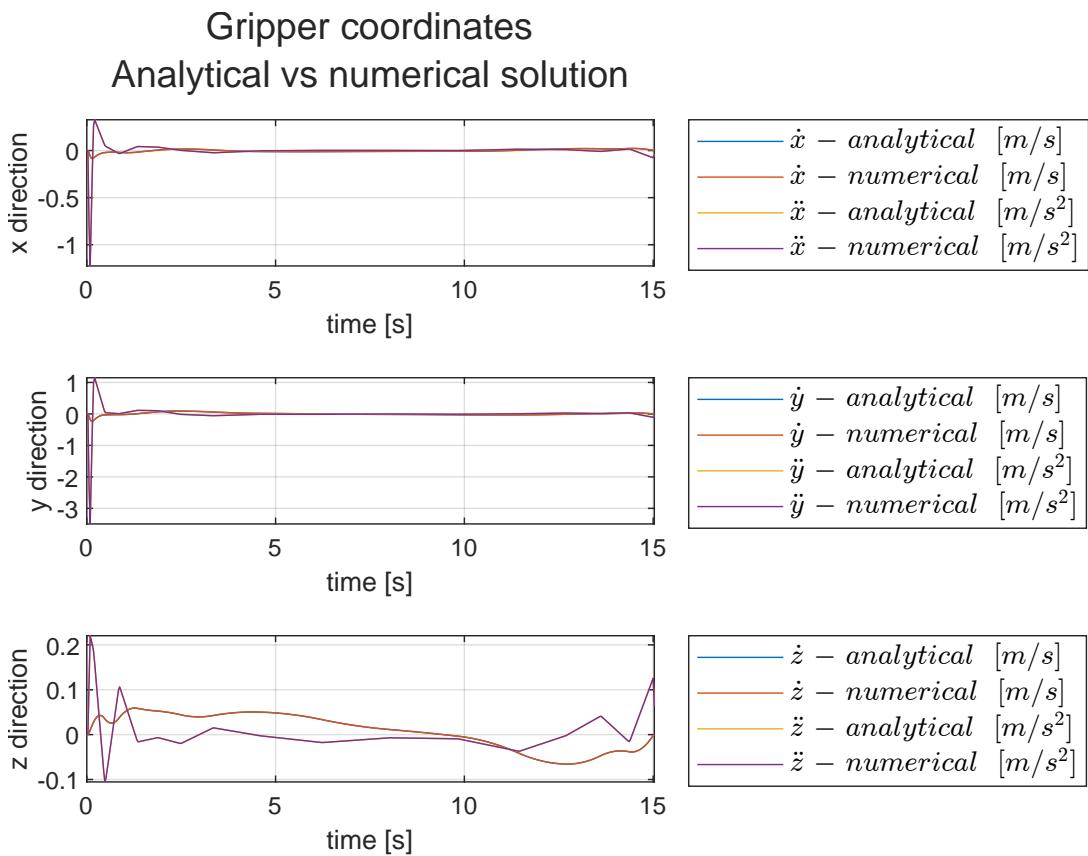


Fig. 21. Confronto tra velocità e accelerazioni dell'*end-effector* ricavate analiticamente e per derivazione.

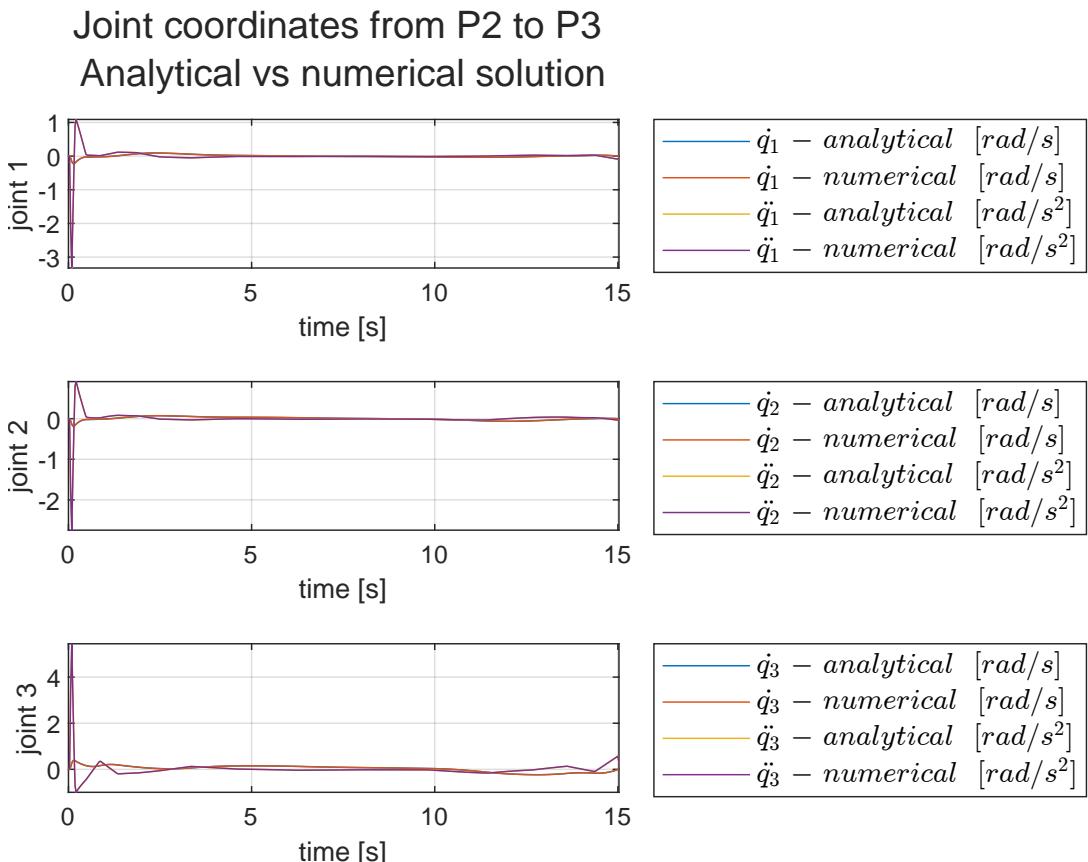


Fig. 22. Confronto tra velocità e accelerazioni dei giunti ricavate analiticamente e per derivazione.

2.5.3 Task 3

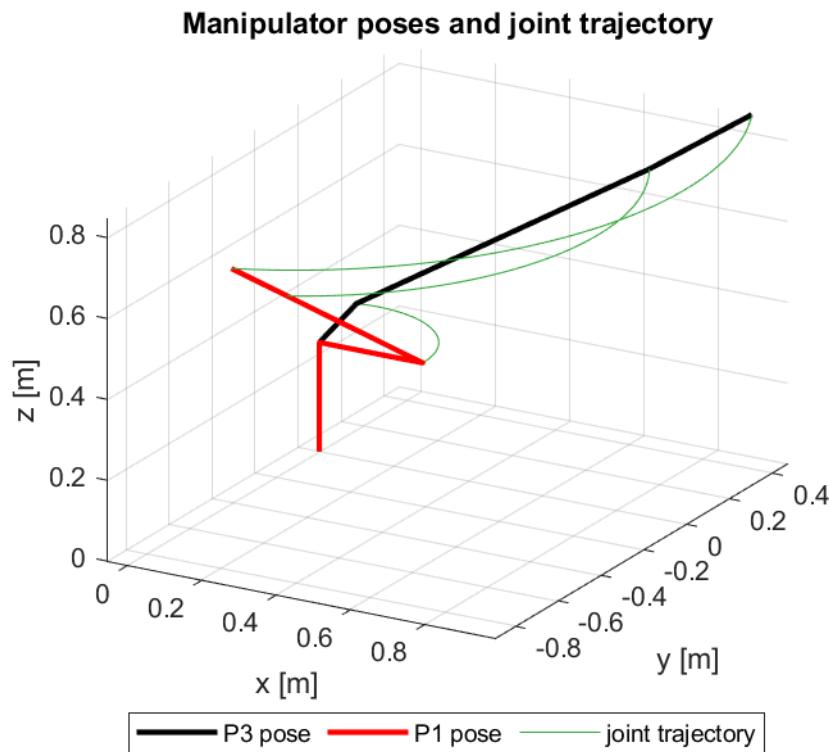


Fig. 23. Traiettoria delle estremità dei singoli link.

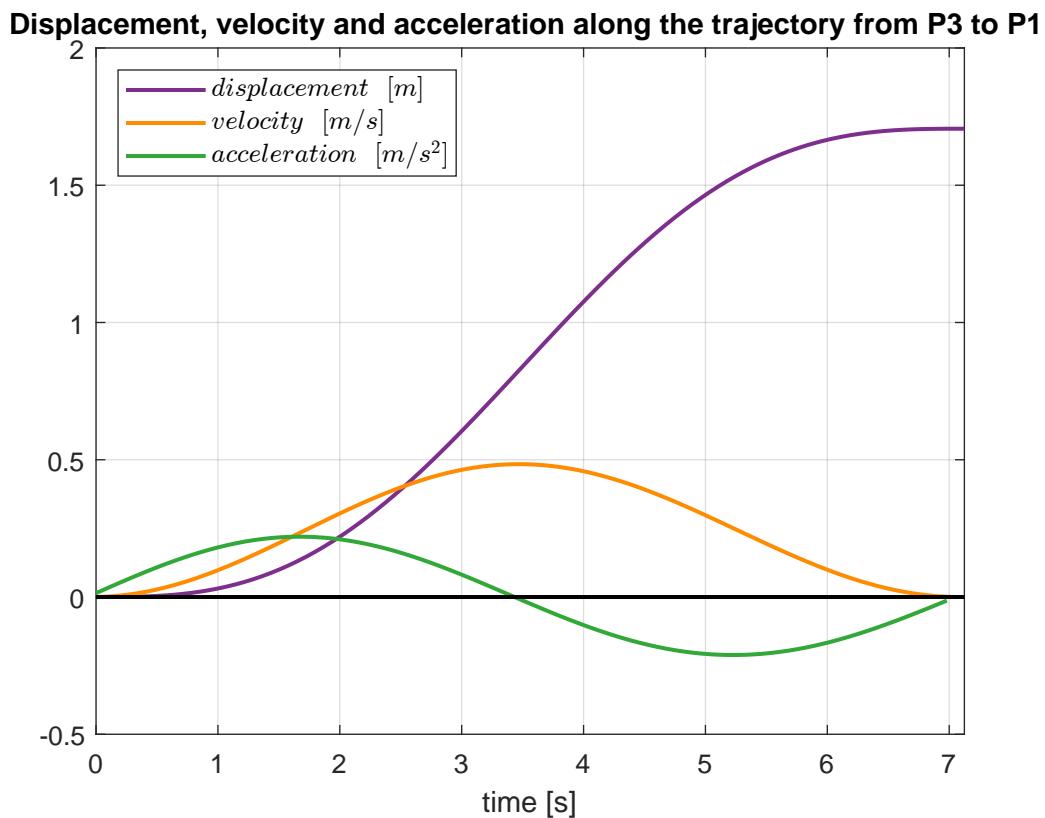
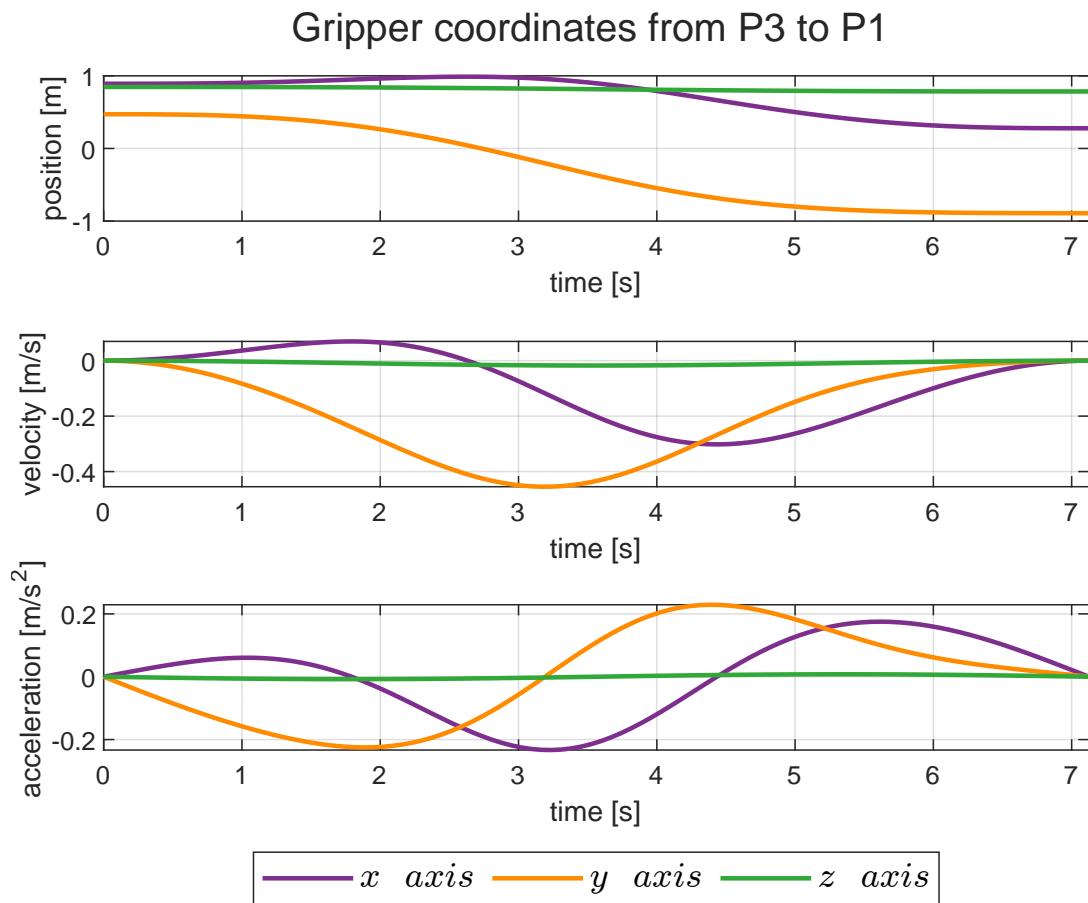
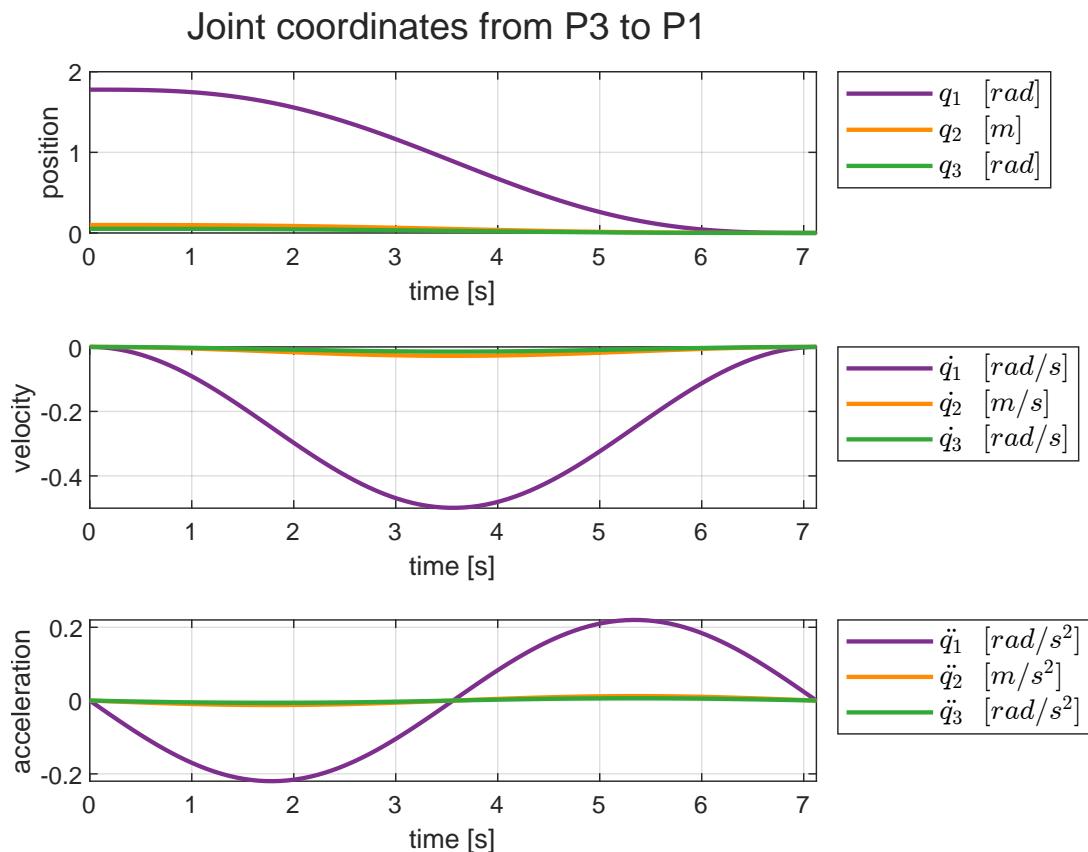


Fig. 24. Spostamento, velocità e accelerazione lungo l'ascissa curvilinea.

**Fig. 25.** Coordinate dell'end-effector.**Fig. 26.** Variabili di giunto.

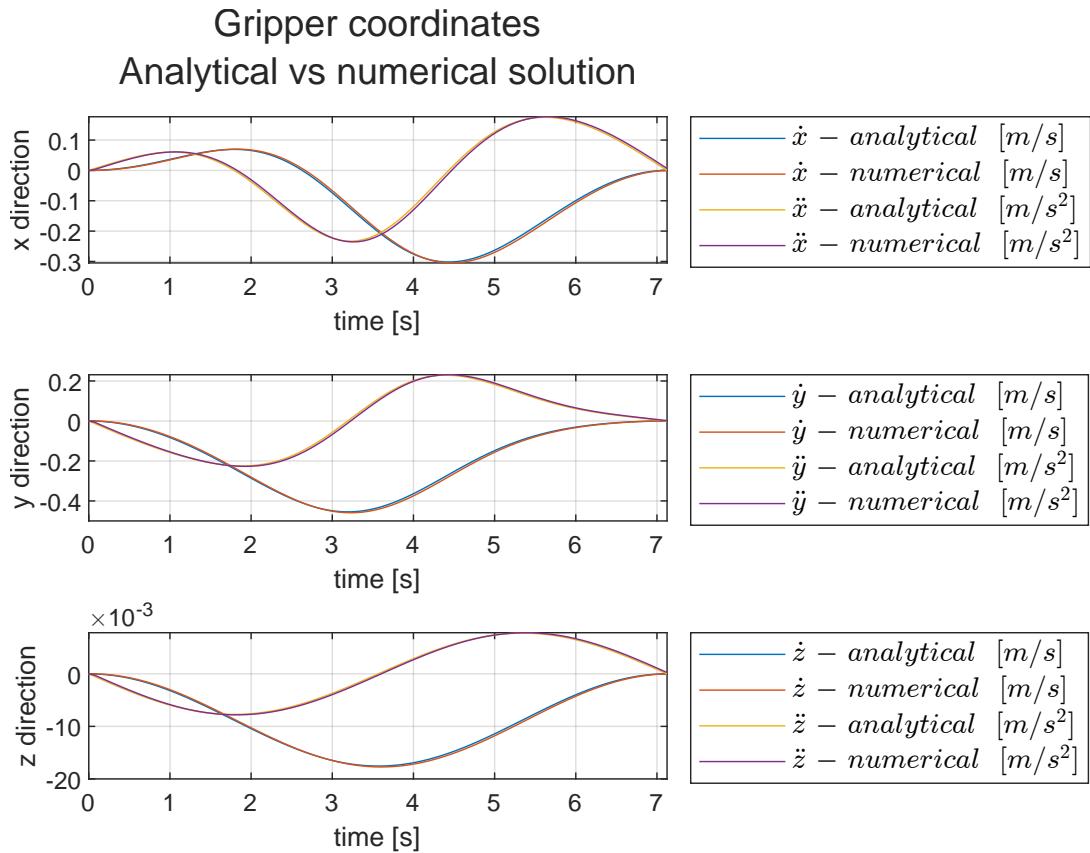


Fig. 27. Confronto tra velocità e accelerazioni dell'*end-effector* ricavate analiticamente e per derivazione.

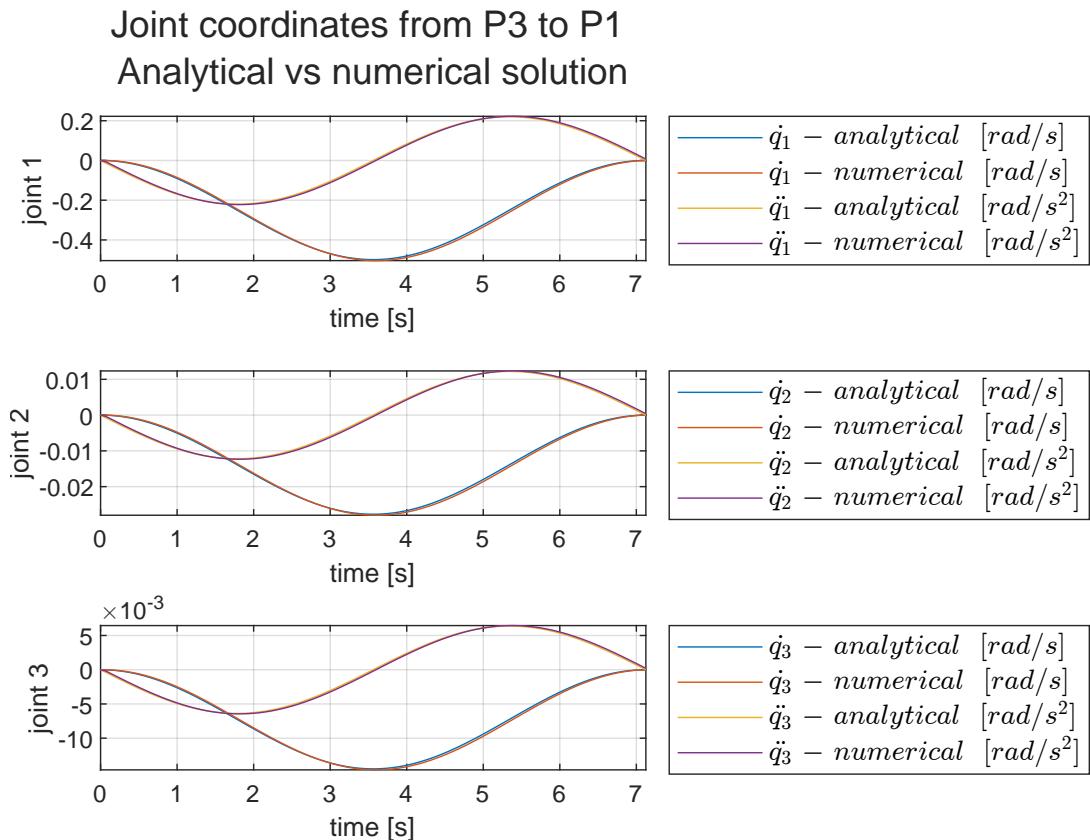


Fig. 28. Confronto tra velocità e accelerazioni dei giunti ricavate analiticamente e per derivazione.

3 Analisi Dinamica

Dati i valori di masse, posizioni dei baricentri, momenti d'inerzia baricentrali, forze e coppie esterne (nulle nel caso specifico, nonostante il codice scritto si presta anche a tenere conto di eventuali azioni esterne) per i cui valori si rimanda al Par. 4.1, si procede con la descrizione dell'analisi dinamica del manipolatore in esame.

L'analisi è di tipo **inverso**: a partire dalle azioni sul manipolatore, si vuole calcolare le forze e/o le coppie che gli attuatori devono esercitare, al fine di garantire la movimentazione imposta con l'analisi cinematica.

3.1 Approccio Jacobiano

Il primo passaggio da compiere consiste nella scrittura delle equazioni cinematiche estese, in cui non viene rappresentata solamente la posizione delle estremità degli elementi geometrici, ma anche quella dei baricentri. Il sistema 4 viene pertanto riscritto come

$$S_e = F_e(Q) \quad (24)$$

dove le espressioni dei vettori delle m variabili di giunto e delle l coordinate estese nello spazio cartesiano sono:

$$\begin{aligned} Q &= [q_1, q_2, q_j, \dots, q_m]^T \\ S_e &= [s_1, s_2, s_i, \dots, s_l]^T. \end{aligned}$$

Le coordinate S_e contengono anche le posizioni (x, y, z) dei baricentri dei link, necessarie per includere gli effetti inerziali di tipo traslazionale, e le variabili di giunto q_1 e q_2 , necessarie per tenere conto di effetti inerziali dovuti alla rotazione dei link attorno all'asse del rispettivo giunto.

Nel caso in esame, alle relazioni 2.1.1, si aggiungono quelle che seguono:

$$\begin{aligned} G_1 : \begin{cases} x_{g1} = x_{1b} + x_{g1} \\ y_{g1} = y_{1b} + y_{g1} \\ z_{g1} = z_{1b} + z_{g1} \end{cases} \quad G_2 : \begin{cases} x_{g2} = x_2 + x_{g2} \\ y_{g2} = y_2 + y_{g2} \\ z_{g2} = z_2 + z_{g2} \end{cases} \quad G_3 : \begin{cases} x_{g3} = x_3 + x_{g3} \\ y_{g3} = y_3 + y_{g3} \\ z_{g3} = z_3 + z_{g3} \end{cases} \\ q_1 = q_1 \quad q_3 = q_3, \end{aligned}$$

ottenendo quindi il seguente vettore delle coordinate estese:

$$S = [x_3, y_3, z_3, x_{g3}, y_{g3}, z_{g3}, x_2, y_2, z_2, x_{g2}, y_{g2}, z_{g2}, x_{1c}, y_{1c}, z_{1c}, x_{1b}, y_{1b}, z_{1b}, x_{g1}, y_{g1}, z_{g1}, x_{1a}, y_{1a}, z_{1a}, q_1, q_3]^T,$$

Il calcolo di velocità ed accelerazione può essere ripetuto utilizzando ancora le formule 5 e 7, avendo cura di far riferimento alle coordinate estese ($J \rightarrow J_e$, $S \rightarrow S_e$). La matrice jacobiana estesa e la sua derivata temporale assumono le seguenti espressioni:

$$J_e = \begin{bmatrix} C_{q_1} C_\alpha (l_2 + q_2) - l_{1b} S_{q_1} + l_3 C_{\alpha q_3} C_{q_1} + l_{1c} C_{q_1} C_\alpha & S_{q_1} C_\alpha & -l_3 S_{\alpha q_3} S_{q_1} \\ l_{1b} C_{q_1} + S_{q_1} C_\alpha (l_2 + q_2) + l_3 C_{\alpha q_3} S_{q_1} + l_{1c} S_{q_1} C_\alpha & -C_{q_1} C_\alpha & l_3 S_{\alpha q_3} C_{q_1} \\ 0 & S_\alpha & l_3 C_{\alpha q_3} \\ C_{q_1} C_\alpha (l_2 + q_2) - l_{1b} S_{q_1} + l_3 C_{\alpha q_3} C_{q_1} + l_{1c} C_{q_1} C_\alpha & S_{q_1} C_\alpha & -l_3 S_{\alpha q_3} S_{q_1} \\ l_{1b} C_{q_1} + S_{q_1} C_\alpha (l_2 + q_2) + l_3 C_{\alpha q_3} S_{q_1} + l_{1c} S_{q_1} C_\alpha & -C_{q_1} C_\alpha & l_3 S_{\alpha q_3} C_{q_1} \\ 0 & S_\alpha & l_3 C_{\alpha q_3} \\ C_{q_1} C_\alpha (l_2 + q_2) - l_{1b} S_{q_1} + l_{1c} C_{q_1} C_\alpha & S_{q_1} C_\alpha & 0 \\ l_{1b} C_{q_1} + S_{q_1} C_\alpha (l_2 + q_2) + l_{1c} S_{q_1} C_\alpha & -C_{q_1} C_\alpha & 0 \\ 0 & S_\alpha & 0 \\ C_{q_1} C_\alpha (l_2 + q_2) - l_{1b} S_{q_1} + l_{1c} C_{q_1} C_\alpha & S_{q_1} C_\alpha & 0 \\ l_{1b} C_{q_1} + S_{q_1} C_\alpha (l_2 + q_2) + l_{1c} S_{q_1} C_\alpha & -C_{q_1} C_\alpha & 0 \\ 0 & S_\alpha & 0 \\ l_{1c} C_{q_1} C_\alpha - l_{1b} S_{q_1} & 0 & 0 \\ l_{1b} C_{q_1} + l_{1c} S_{q_1} C_\alpha & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b} S_{q_1} & 0 & 0 \\ l_{1b} C_{q_1} & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b} S_{q_1} & 0 & 0 \\ l_{1b} C_{q_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\dot{J}_e = \begin{bmatrix} \dot{j}_{1,1} & C_{q_1} C_\alpha \dot{q}_1 & -l_3 S_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 C_{\alpha q_3} S_{q_1} \dot{q}_3 \\ \dot{j}_{2,1} & S_{q_1} C_\alpha \dot{q}_1 & l_3 C_{\alpha q_3} C_{q_1} \dot{q}_3 - l_3 S_{\alpha q_3} S_{q_1} \dot{q}_1 \\ 0 & 0 & -l_3 S_{\alpha q_3} \dot{q}_3 \\ \dot{j}_{4,1} & C_{q_1} C_\alpha \dot{q}_1 & -l_3 S_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 C_{\alpha q_3} S_{q_1} \dot{q}_3 \\ \dot{j}_{5,1} & S_{q_1} C_\alpha \dot{q}_1 & l_3 C_{\alpha q_3} C_{q_1} \dot{q}_3 - l_3 S_{\alpha q_3} S_{q_1} \dot{q}_1 \\ 0 & 0 & -l_3 S_{\alpha q_3} \dot{q}_3 \\ \dot{j}_{7,1} & C_{q_1} C_\alpha \dot{q}_1 & 0 \\ \dot{j}_{8,1} & S_{q_1} C_\alpha \dot{q}_1 & 0 \\ 0 & 0 & 0 \\ \dot{j}_{10,1} & C_{q_1} C_\alpha \dot{q}_1 & 0 \\ \dot{j}_{11,1} & S_{q_1} C_\alpha \dot{q}_1 & 0 \\ 0 & 0 & 0 \\ -l_{1b} C_{q_1} \dot{q}_1 - l_{1c} S_{q_1} C_\alpha \dot{q}_1 & 0 & 0 \\ l_{1c} C_{q_1} C_\alpha \dot{q}_1 - l_{1b} S_{q_1} \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b} C_{q_1} \dot{q}_1 & 0 & 0 \\ -l_{1b} S_{q_1} \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \\ -l_{1b} C_{q_1} \dot{q}_1 & 0 & 0 \\ -l_{1b} S_{q_1} \dot{q}_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

dove

$$j_{1,1} = C_{q_1} C_\alpha \dot{q}_2 - l_{1b} C_{q_1} \dot{q}_1 - S_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) - l_3 C_{\alpha q_3} S_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} C_{q_1} \dot{q}_3 - l_{1c} S_{q_1} C_\alpha \dot{q}_1$$

$$j_{2,1} = S_{q_1} C_\alpha \dot{q}_2 - l_{1b} S_{q_1} \dot{q}_1 + C_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) + l_3 C_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} S_{q_1} \dot{q}_3 + l_{1c} C_{q_1} C_\alpha \dot{q}_1$$

$$j_{4,1} = C_{q_1} C_\alpha \dot{q}_2 - l_{1b} C_{q_1} \dot{q}_1 - S_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) - l_3 C_{\alpha q_3} S_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} C_{q_1} \dot{q}_3 - l_{1c} S_{q_1} C_\alpha \dot{q}_1$$

$$j_{5,1} = S_{q_1} C_\alpha \dot{q}_2 - l_{1b} S_{q_1} \dot{q}_1 + C_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) + l_3 C_{\alpha q_3} C_{q_1} \dot{q}_1 - l_3 S_{\alpha q_3} S_{q_1} \dot{q}_3 + l_{1c} C_{q_1} C_\alpha \dot{q}_1$$

$$_{7,1} = C_{q_1} C_\alpha \dot{q}_2 - l_{1b} C_{q_1} \dot{q}_1 - S_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) - l_{1c} S_{q_1} C_\alpha \dot{q}_1$$

$$_{8,1} = S_{q_1} C_\alpha \dot{q}_2 - l_{1b} S_{q_1} \dot{q}_1 + C_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) + l_{1c} C_{q_1} C_\alpha \dot{q}_1$$

$$_{10,1} = C_{q_1} C_\alpha \dot{q}_2 - l_{1b} C_{q_1} \dot{q}_1 - S_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) - l_{1c} S_{q_1} C_\alpha \dot{q}_1$$

$$_{11,1} = S_{q_1} C_\alpha \dot{q}_2 - l_{1b} S_{q_1} \dot{q}_1 + C_{q_1} C_\alpha \dot{q}_1 (l_2 + q_2) + l_{1c} C_{q_1} C_\alpha \dot{q}_1.$$

Una volta definite le coordinate estese, è necessario approssimare i link che si muovono nello spazio con sistemi di masse puntiformi, in quanto l'approccio jacobiano non consente di studiare moti tridimensionali. Un ottimo compromesso tra accuratezza dell'approssimazione e semplicità implementativa è la cosiddetta *approssimazione a 3 masse puntiformi*, che è stata applicata per i *link 2 e 3* (una dimensione prevalente sulle altre due), mentre per il *link 1* si è ritenuto più opportuno utilizzare un'*approssimazione a 5 masse puntiformi*, in modo da rimanere più fedeli alla struttura del *link* stesso.

Le relazioni che permettono di ottenere 3 masse puntiformi il cui baricentro e momento d'inerzia baricentrale siano pari a quelli del *link* sono riportate in Fig. 29, mentre per l'approssimazione a 5 masse è stato necessario scrivere il sistema 25, poi risolto per trovare i valori delle 5 masse. Le posizioni di queste masse, aggiungendo quelle baricentrali, si possono vedere in Fig. 7.

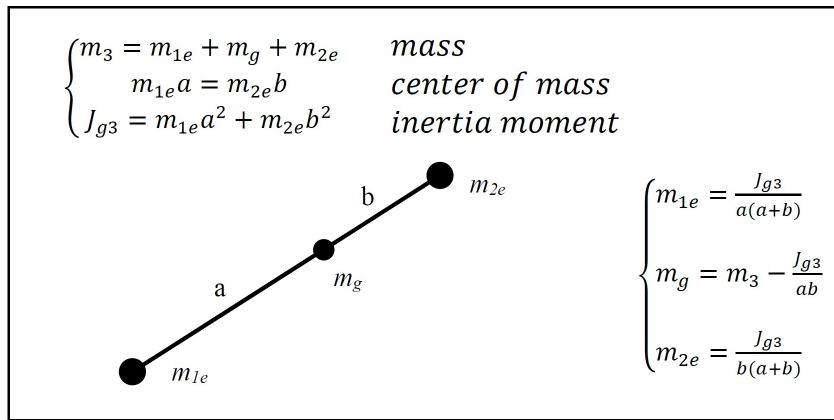


Fig. 29. Approssimazione di un *link* con 3 masse puntiformi.

$$\begin{cases} m_{1,prox} + m_{1a} + m_g + m_{1b} + m_{1,dist} = m_1 \\ m_{1,prox}P_0 + m_{1a}P_{1a} + m_gG_1 + m_{1b}P_{1b} + m_{1,dist}P_{1c} = G_1m_1 \\ m_{1,prox}d_0^2 + m_{1a}d_{1a}^2 + m_gP_g^2 + m_{1b}P_{1b} + m_{1,dist}P_{1c} = J_{1,zz} \end{cases} \quad (25)$$

I punti P_i sono gli stessi definiti per la cinematica, mentre il punto G_1 è la posizione del baricentro del *link 1* rispetto al sistema di riferimento di base. La seconda relazione, pertanto,

è di tipo vettoriale. Le distanze d_i rappresentano la perpendicolare condotta dal punto all'asse del giunto rotoidale di base.

Una volta ottenuto un insieme di masse equivalenti, bisogna definire la matrice delle masse generalizzate

$$\begin{aligned} \mathcal{M} = diag[& repelem(m_{grip} + m_{3,dist}, 3), repelem(m_{3g}, 3), repelem(m_{3,prox} + m_{2,dist}, 3), \\ & repelem(m_{2g}, 3), repelem(m_{2,prox} + m_{1,dist}, 3), repelem(m_{1b}, 3), \\ & repelem(m_{1g}, 3), repelem(m_{1a}, 3), \\ & J_{g1,zz} + J_{g2,vertical}, J_{g3,yy}], \end{aligned} \quad (26)$$

il vettore delle eventuali forze esterne (qui assenti)

$$F_{ext} = zeros(26, 1), \quad (27)$$

il vettore dell'accelerazione di gravità

$$A_g = [0, 0, -9.81, 0, 0, -9.81, 0, 0, -9.81, 0, 0, -9.81, \\ 0, 0, -9.81, 0, 0, -9.81, 0, 0, -9.81, 0, 0, -9.81, 0, 0], \quad (28)$$

per poter poi calcolare il totale delle forze esterne (forza peso inclusa):

$$F_{tot} = F_{ext} + \mathcal{M} A_g, \quad (29)$$

Per ogni punto della traiettoria sono state poi calcolate le seguenti quantità:

- $F_q = (J_e^T \mathcal{M} J_e) \ddot{Q} + (J_e^T \mathcal{M} \dot{J}_e) \dot{Q} - J_e^T F_{tot} \rightarrow$ forze generalizzate (i.e. forze e coppie) richieste agli attuatori.
- $E_k = \frac{1}{2} \dot{Q}^T J_e^T \mathcal{M} J_e \dot{Q} \rightarrow$ energia cinetica del sistema.
- $E_p = -A_g^T \mathcal{M} J_e \dot{Q} \rightarrow$ energia potenziale del sistema.
- $E_{tot} = E_k + E_p \rightarrow$ energia meccanica totale del sistema.
- $W_q = F_q^T \dot{Q} \rightarrow$ potenza richiesta ai motori posti nei giunti.
- $W_{ext} = F_{ext}^T J_e \dot{Q} \rightarrow$ potenza sviluppata dalle forze esterne.
- $W_{weight} = (\mathcal{M} A_g)^T J_e \dot{Q} \rightarrow$ potenza delle forze peso.
- $W_{tot} = W_q + W_{ext} \rightarrow$ potenza totale dovuta a forze/coppie esterne e alle azioni motrici.
- $W_{tot,w} = W_q + W_{ext} + W_{weight} \rightarrow$ potenza totale dovuta a forze/coppie esterne, azioni motrici e forza peso.

3.2 Approccio Matriciale

Come per l'approccio precedente, il codice è stato sviluppato con valori generici dei dati geometrici ed inerziali, inserendo in un secondo momento i valori corretti, forniti in automatico da Solidworks.

Per lo sviluppo dei calcoli è necessario definire diverse matrici, che permettono di modellizzare gli effetti inerziali anche per sistemi che si muovono nello spazio e non solo nel piano.

Dopo la definizione della matrice H_g come:

$$H_g = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -9.81 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

si procede con la conversione dei momenti d'inerzia baricentrali (Par. 4.1) secondo la convenzione necessaria alla definizione dello *pseudo tensore d'inerzia*:

$$J = \begin{bmatrix} \frac{-J_x + J_y + J_z}{2} & -J_{xy} & -J_{xz} & 0 \\ -J_{xy} & \frac{-J_y + J_x + J_z}{2} & -J_{yz} & 0 \\ -J_{xz} & -J_{yz} & \frac{-J_z + J_x + J_y}{2} & -9.81 \\ 0 & 0 & 0 & m \end{bmatrix}$$

e con la conversione rispetto al sistema di riferimento a valle del relativo *link* tramite la relazione:

$$J_i = M_{i,j} \cdot J_j \cdot M_{i,j}^T.$$

Nello specifico, le matrici $M_{i,j}$ a cui ci si riferisce sono matrici di posizione tra i sistemi di riferimento di ciascun *link* e i sistemi di riferimento baricentrali con assi paralleli ai sistemi di riferimento dei *link* stessi.

L'approccio adottato prevede di partire con l'analisi del *link* più esterno, calcolando le forze agenti sullo stesso e ricavando le azioni scambiate con il *link* precedente. Tramite una procedura iterativa, per ogni *link* e per ogni punto della traiettoria sono stati convertiti gli pseudo tensori d'inerzia J rispetto al sistema di riferimento di base e sono state calcolate le seguenti grandezze (tutte riferite al sistema di riferimento di base, qui sottinteso):

- $\phi_g = skew\{H_g \cdot J\} \rightarrow$ peso del *link*.
- $\phi_{in} = skew\{H \cdot J\} \rightarrow$ azioni d'inerzia del *link* corrente rispetto a (0) espresso in (0).
- $\phi_{vinc(j)} = -\phi_{in(j)} + \phi_{g(j)} + \phi_{vinc(j+1)} \rightarrow$ reazioni vincolari dal *link* corrente al precedente, con j numero del *link*. Nel caso dell'ultimo *link* $\phi_{in(j+1)}$ è sostituito dalle azioni esterne che potrebbero interagire con l'*end-effector*.
- $\phi_{act.forces} = -\phi_{vinc} \odot L \rightarrow$ forza dell'attuatore sul *link* corrente secondo il *Principio dei Lavori Virtuali*.

Da un punto di vista energetico, sono poi calcolate:

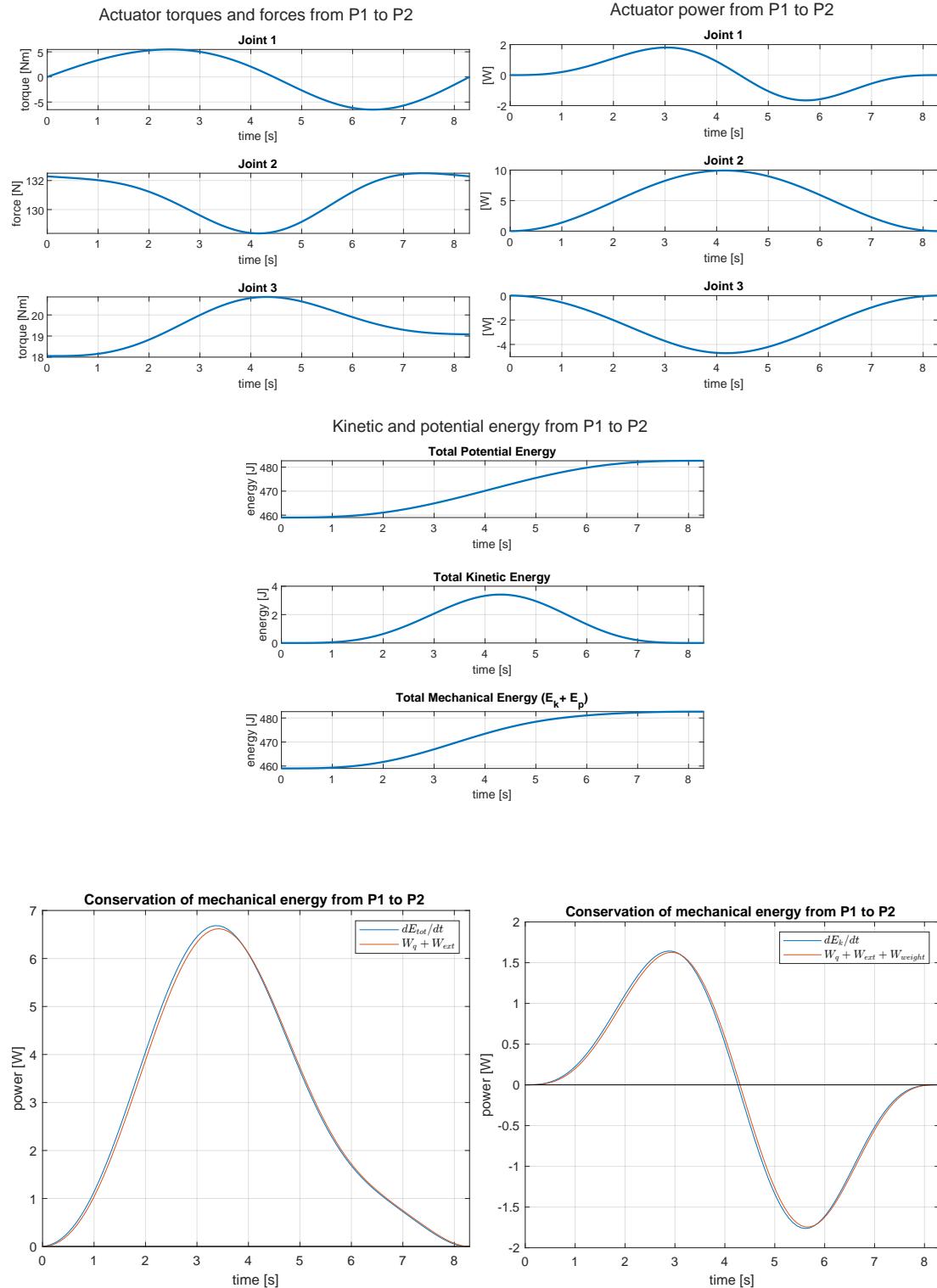
- Energia cinetica: $E_k = \text{trace}(\frac{1}{2}WJW^T)$
- Energia potenziale: $E_p = -\text{trace}(H_g J)$
- Energia meccanica totale: $E_m = E_k + E_p$

Sono infine calcolate le potenze:

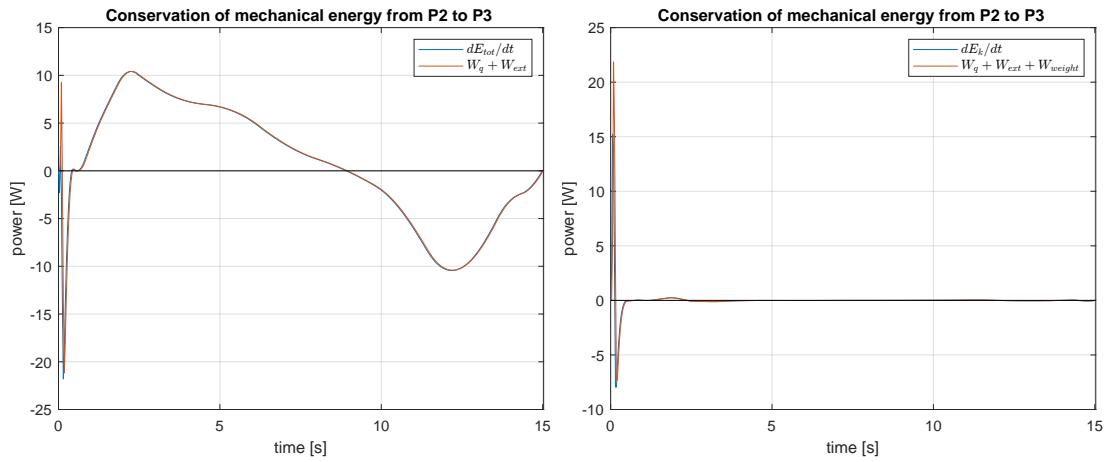
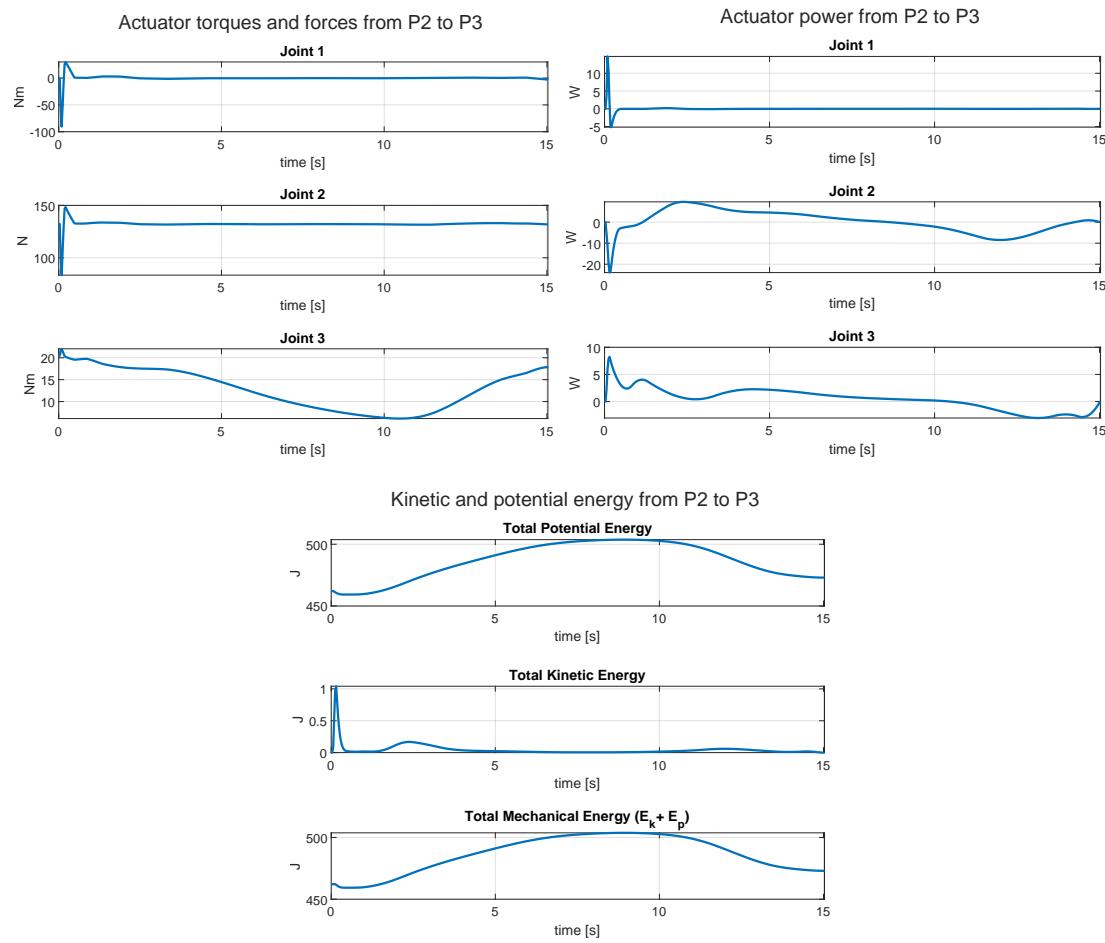
- $W_q = -\phi_{vinc} \odot W_{rel} \rightarrow$ potenza dell'attuatore nel *joint* precedente esercitata sul link corrente. W_{rel} in questo caso è la matrice di velocità relativa tra il *link* precedente e il link corrente.
- $W_g = \phi_g \odot W \rightarrow$ potenza generata per vincere la forza peso
- $W_{ext} = \phi_{ext} \odot W_{aux} \rightarrow$ potenza generata dalle forze esterne. W_{aux} è la velocità del link calcolata in un sistema di riferimento traslante (assi paralleli) rispetto sistema di riferimento di base.
- Infine si calcola la potenza totale come la somma di tutte le potenze prodotte dai motori e della potenza delle azioni esterne.

3.3 Risultati e Debug

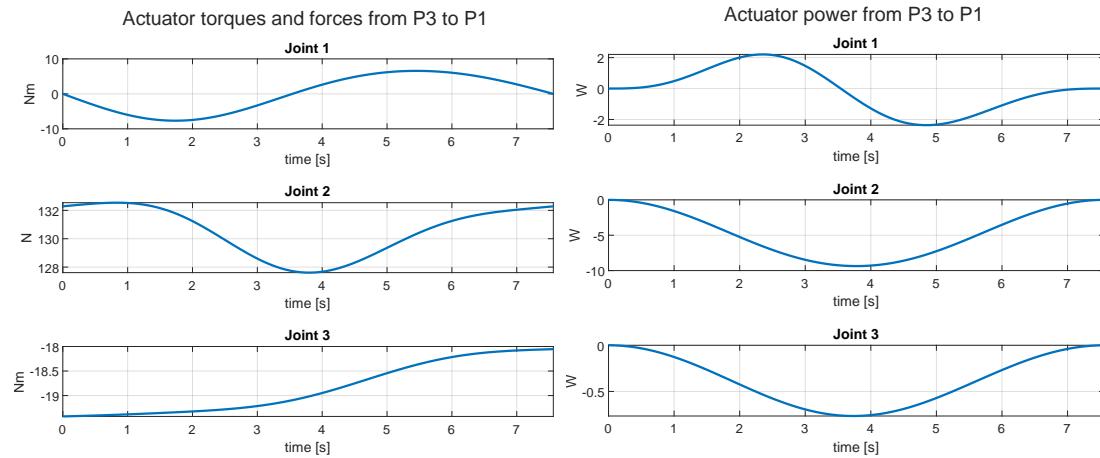
3.3.1 Task 1 - Approccio jacobiano



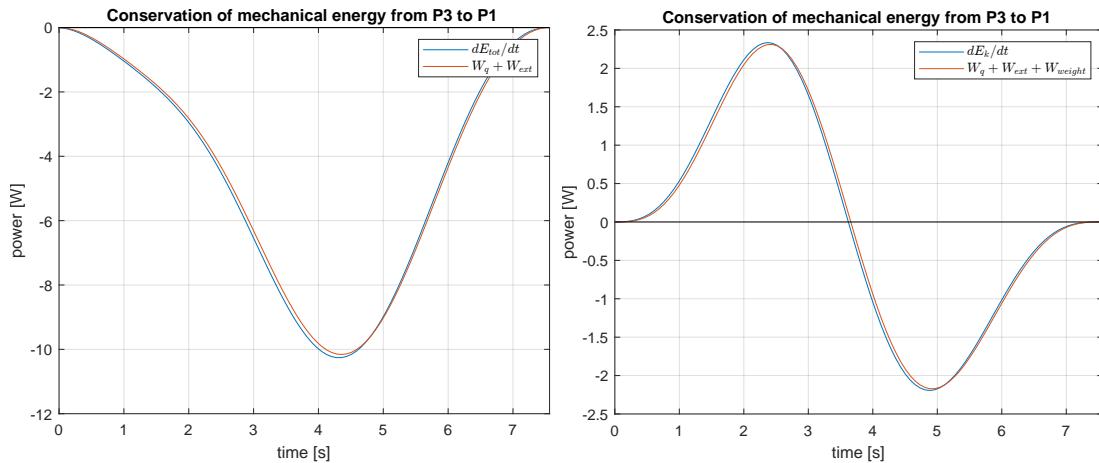
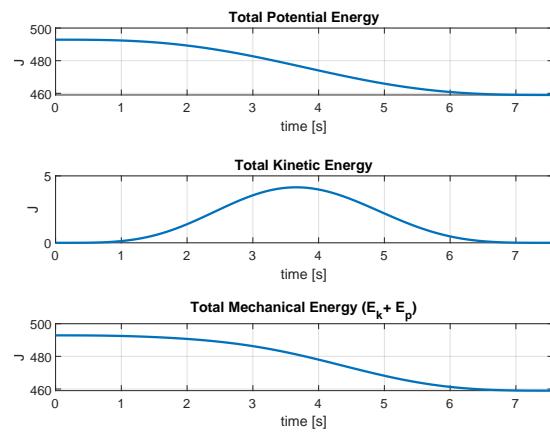
3.3.2 Task 2 - Approccio jacobiano



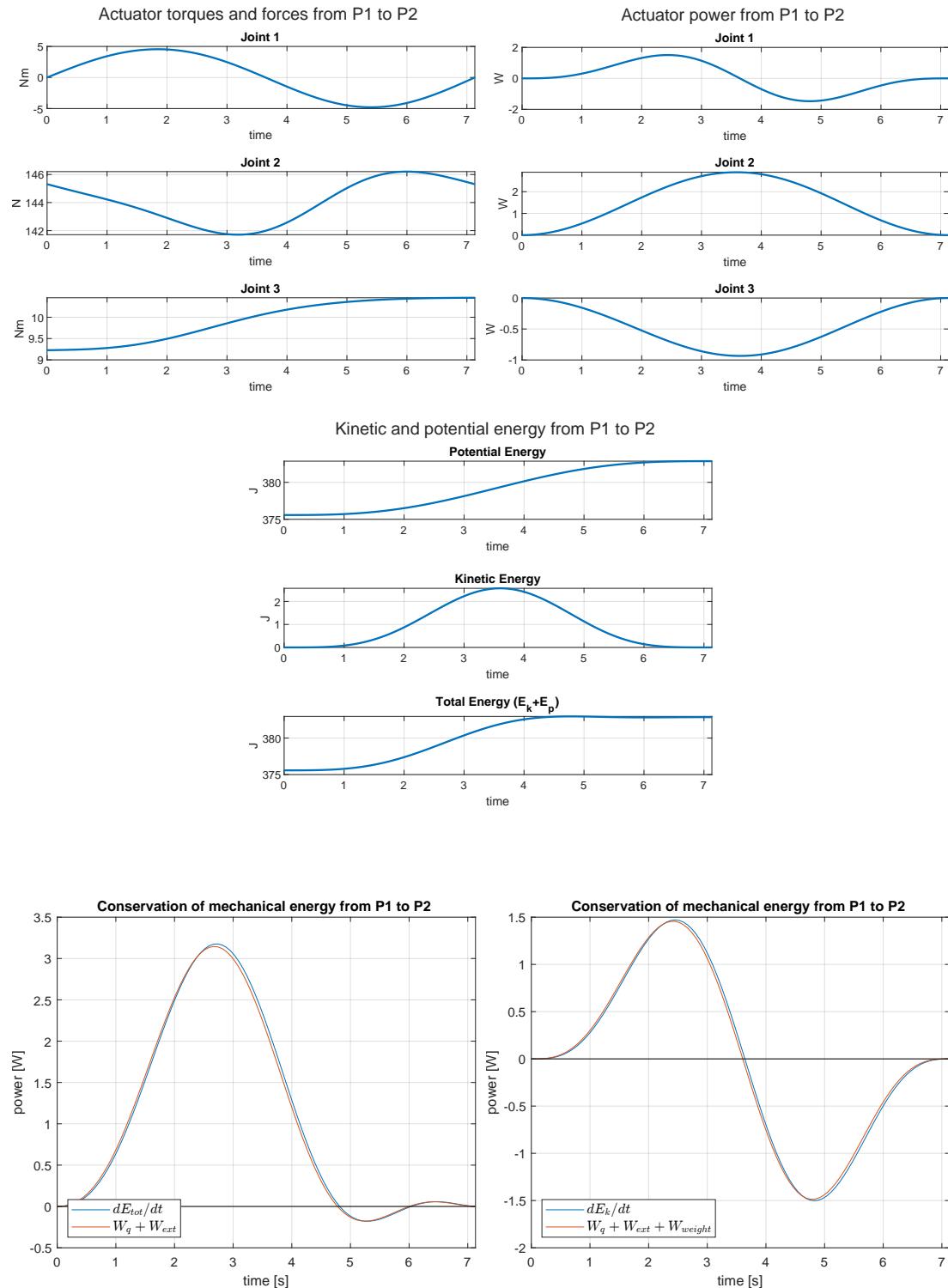
3.3.3 Task 3 - Approccio jacobiano



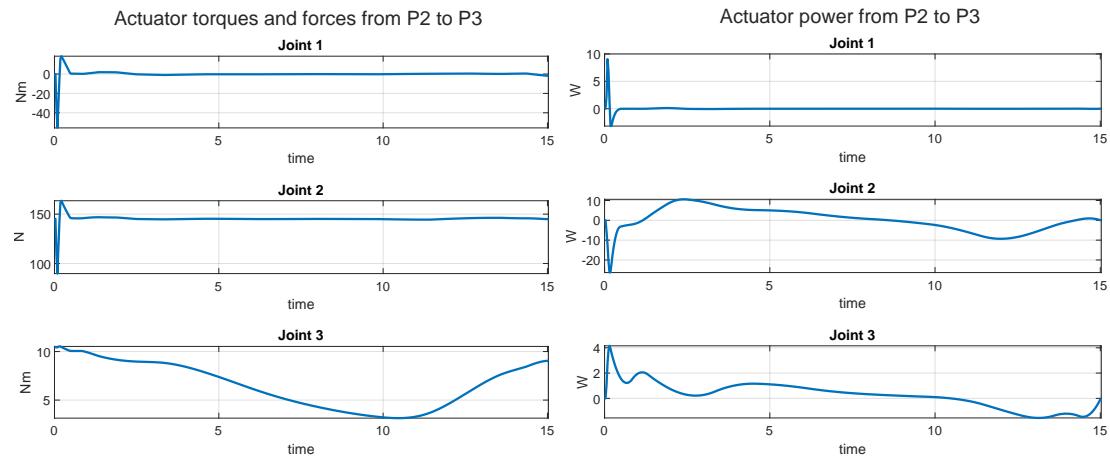
Kinetic and potential energy from P3 to P1



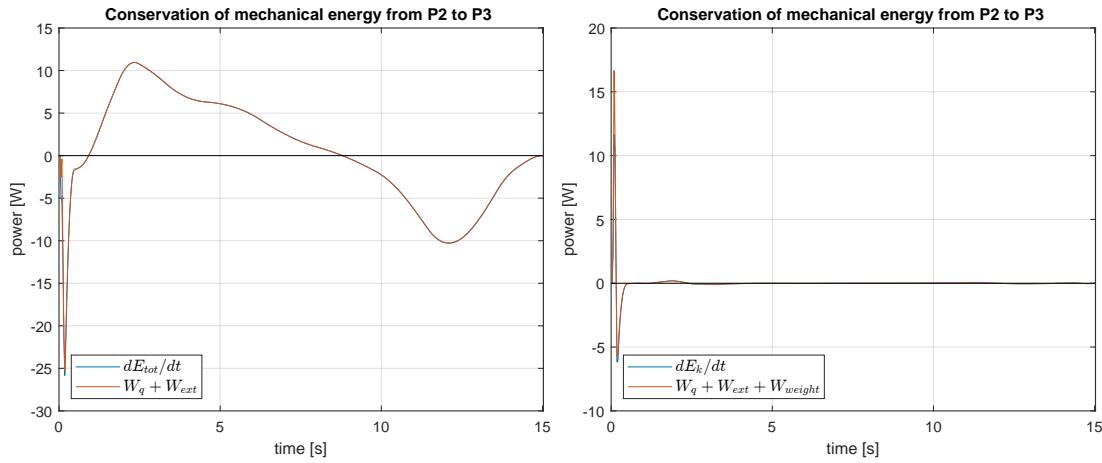
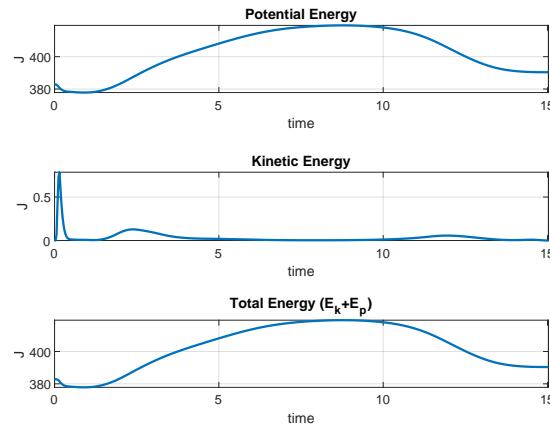
3.3.4 Task 1 - Approccio matriciale



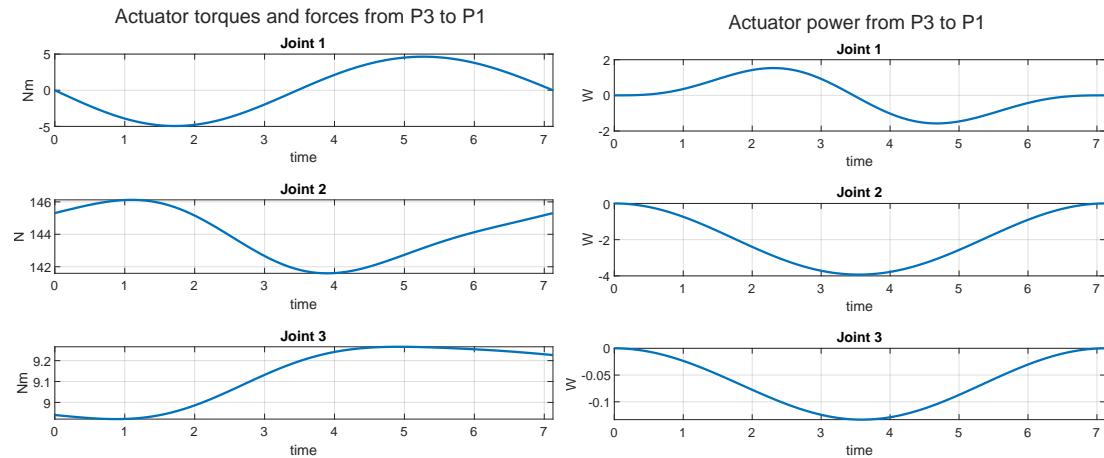
3.3.5 Task 2 - Approccio matriciale



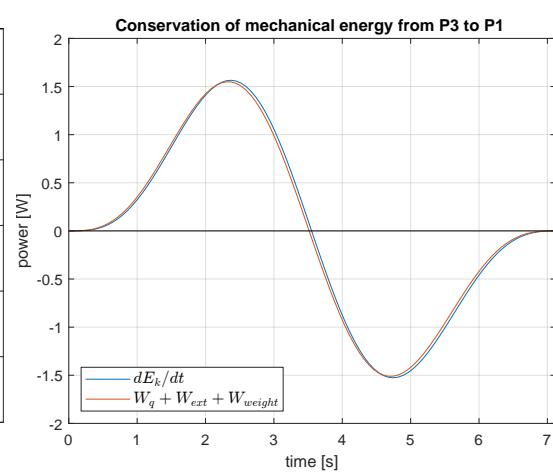
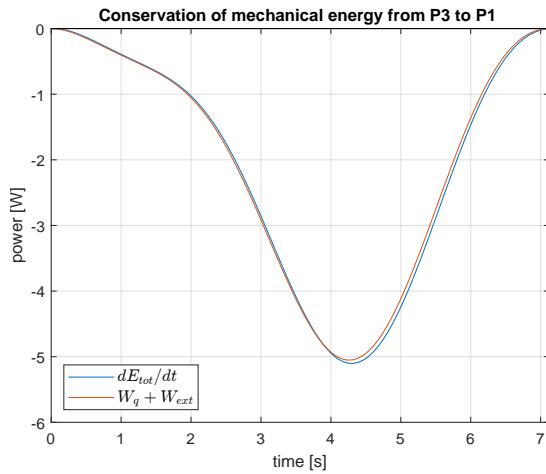
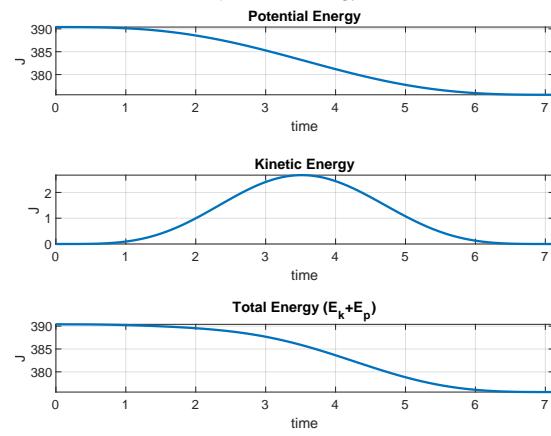
Kinetic and potential energy from P2 to P3



3.3.6 Task 3 - Approccio matriciale



Kinetic and potential energy from P3 to P1



3.3.7 Osservazioni e Confronto

I due metodi conducono a risultati perfettamente compatibili: gli andamenti delle grandezze sono analoghi e in entrambi i casi i limiti di forza/coppia sono rispettati. Si notano delle differenze nel valore assoluto delle quantità calcolate con metodi diversi in quanto l'approccio jacobiano prevede necessariamente una fase preliminare di approssimazione: simulando il comportamento di corpi rigidi con sistemi equivalenti di masse puntiformi, non si possono ottenere proprietà inerziali accurate come quelle che si possono tenere in considerazione con il metodo matriciale, che nasce per descrivere moti tridimensionali.

Oltre all'applicazione in parallelo di due metodi, un'altra strategia di debug consiste nel verificare se i risultati di un'analisi dinamica di tipo newtoniano (equilibri delle forze e coppie) coincidono con quelli di uno studio energetico del medesimo fenomeno. Viene pertanto verificata la conservazione dell'energia meccanica totale mediante le due formule seguenti, che si equivalgono:

$$\frac{dE_{tot}}{dt} = W_q + W_{ext}, \quad (30)$$

$$\frac{dE_k}{dt} = W_q + W_{ext} + W_{weight}. \quad (31)$$

4 Modellazione in Simscape Multibody

4.1 Progettazione in Solidworks

Completate l'analisi cinematica e dinamica e ottenute le coppie e forze che gli attuatori devono esercitare per garantire la percorrenza delle traiettorie assegnate, si è passati da dimensioni e dati inerziali generici a valori più realistici. Con questi dati sono stati rieseguiti i calcoli per ottenere i risultati definitivi.

La modellazione dei singoli *link* è stata fatta in ambiente Solidworks, dove, una volta assegnate le terne di riferimento ai singoli *link* e composto l'assieme, è stato possibile eseguire la valutazione automatica delle posizioni dei baricentri, delle masse e delle matrici d'inerzia dei *link*. Il vantaggio di questo approccio è la possibilità di non adottare approssimazioni che tengano conto solo dei momenti principali d'inerzia dei *link*: questo si è reso necessario per trattare il *link 1*, considerato come un unico corpo con geometria complessa, non assimilabile a un cilindro o a un parallelepipedo.

I risultati di questa analisi sono stati riportati nella Sez. , mentre in Fig. ?? si può apprezzare il livello di dettaglio che è possibile ottenere, seppure le forme siano comunque grezze e puramente indicative.

4.2 Importazione in Simscape Multibody

Grazie al *plug-in Simscape Multibody Link* di Simulink [3] è possibile importare automaticamente modelli Solidworks in ambiente Simscape: vengono generati file STEP, la cui gerarchia è definita da un file XML che può essere letto da Simulink. Questo procedimento risulta molto comodo, in quanto possono essere definite geometrie complicate a piacere e si evitano "errori di assemblaggio", quali scorrette definizioni delle matrici di trasformazione e dei giunti.

L'approccio alternativo sarebbe stata la definizione direttamente in Simscape dei solidi e delle trasformazioni rigide che li collegano, inserendo poi i blocchi *giunto* per collegare e attuare i vari *link*. Non sarebbe stato possibile, quindi, ottenere facilmente una descrizione geometrica accurata del manipolatore.

4.3 Attuazione dei giunti

Oltre allo schema a blocchi del manipolatore, è necessario definire una funzione che fornisca il riferimento di posizione, velocità ed accelerazione ai singoli assi. Dai valori teorici ottenuti dall'algoritmo di percorrenza della traiettoria, di volta in volta diversi a seconda dei vari task, questo blocco interpolatore si occupa semplicemente di infittire il campionamento dei dati stessi, in modo da renderlo compatibile con il tempo di simulazione e garantire così una

movimentazione fluida.

Per l'esecuzione della simulazione è stato scelto dapprima un solutore fixed-step, per poi orientarsi su quello di default di tipo *variable-step*, che consente di ottenere grafici lisci, privi di oscillazioni e picchi di natura puramente numerica.

4.4 Verifica del modello

Una volta ottenuto lo schema a blocchi del modello ed aver definito il blocco interpolatore, è opportuno effettuare una verifica del corretto assemblaggio, prima di passare al controllo dei motori. Si è deciso pertanto di fornire direttamente ai giunti il riferimento di posizione, velocità e accelerazione ideali (Fig. 30), controllando che la forza/coppia risultante di ciascuno (**dinamica inversa**) coincidesse con quella teorica calcolata con il metodo matriciale (si riscontrano invece differenze con il metodo jacobiano, del tutto previste in quanto si tratta di un'analisi approssimata). Questa verifica è andata a buon fine, indice di una corretta strutturazione del modello.

Se tuttavia si prova la verifica inversa, i risultati non sono confortanti: azionando i giunti con il valore teorico di forza/coppia e simulando il sistema, si ottengono valori di posizione, velocità e accelerazioni delle coordinate di giunto del tutto sbagliate (Fig. 31). Il fallimento della **dinamica diretta** potrebbe essere attribuibile ad errori numerici nella doppia integrazione che dalle forze/coppie, quindi dalle accelerazioni, consente di risalire alle posizioni. Resta da capire se sia presente qualche minima imprecisione nell'assemblaggio del modello da parte del *plug-in Simscape Multibody Link* oppure se i vari solutori testati non sono in grado di risalire accuratamente ad accelerazione, velocità e posizione.

Nonostante questa difficoltà riscontrata, è stato possibile elaborare uno schema di controllo che permette di controllare i giunti in modo da ottenere le traiettorie desiderate, come descritto nella Sez. 5.

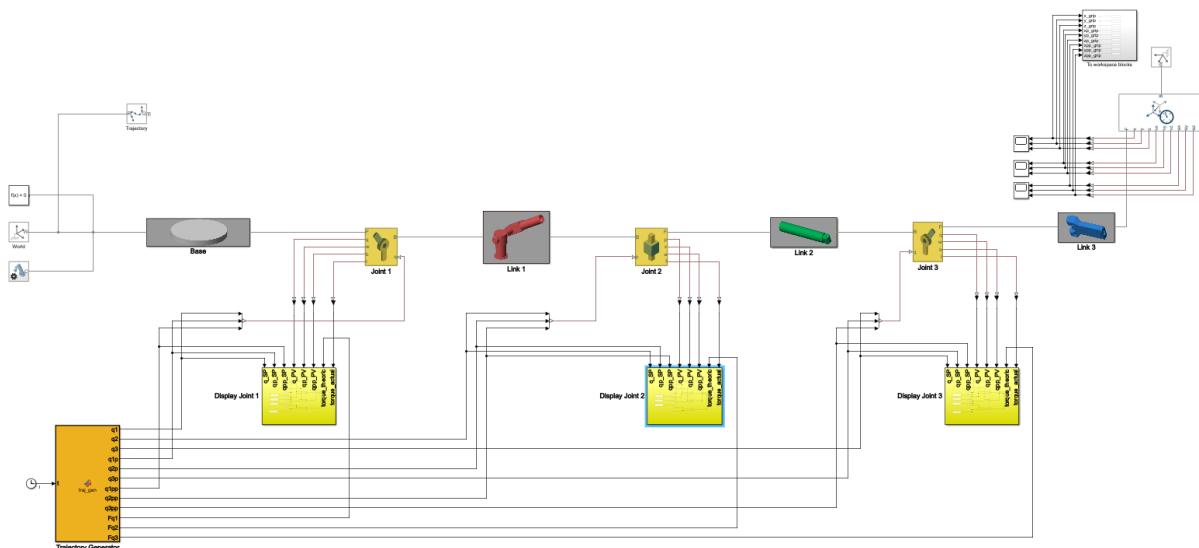


Fig. 30. Attuazione dei giunti fornendo il riferimento ideale di posizione, velocità ed accelerazione.

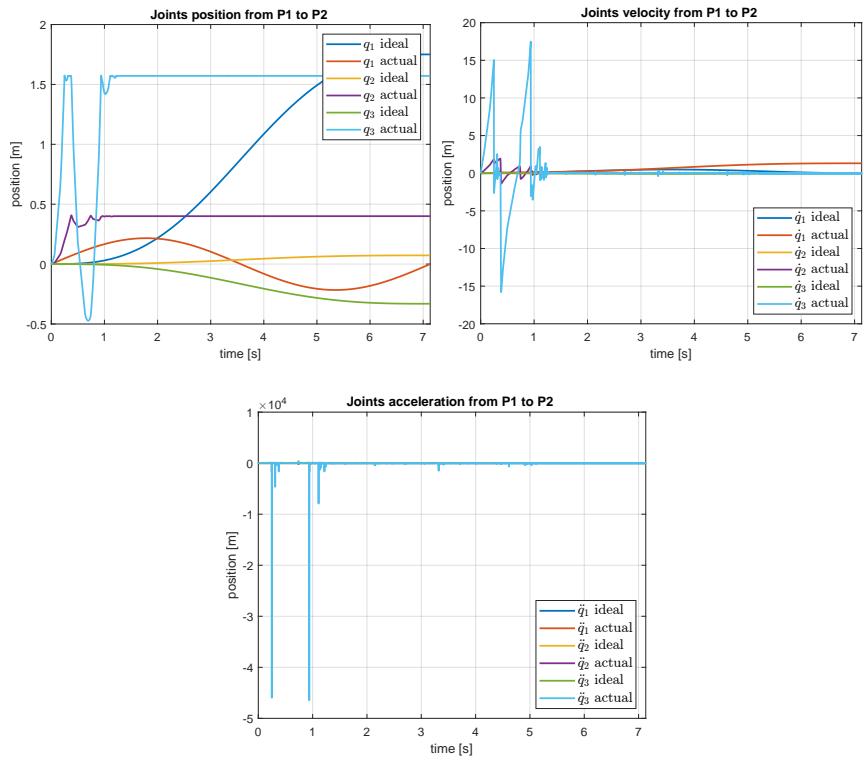


Fig. 31. Errori di integrazione da forze/copie a variabili di giunto

5 Sistema di Controllo

Dopo aver testato il modello *Simscape* del manipolatore si è passati all'elaborazione di un'architettura per il controllo dei giunti. Nonostante uno schema di controllo di tipo *centralizzato* consenta, in alcuni casi, di migliorare le prestazioni, data la capacità di tenere conto di eventuali mutue influenze tra i giunti, in questo caso si è optato per un controllo di tipo *decentralizzato*, azionando ciascun giunto singolarmente.

In Fig. 32 è riportato lo schema di controllo. Una funzione (*Trajectory generator*) invia la posizione come *input* ai regolatori che controllano i giunti fornendo loro la coppia da impiegare.

Per la realizzazione dei controllori 33 si è scelto di utilizzare uno schema a cascata (Fig. 34), tarando il loop interno in modo aggressivo per il controllo della velocità e la reiezione di eventuali disturbi e gestendo il raggiungimento del *set-point* tramite la costante integrativa del loop esterno.

Per affrontare le diverse attività del robot si è deciso di impiegare una taratura di tipo *gain scheduling*, selezionando cioè i parametri più opportuni per ogni singolo task, qui riportati:

	K_p	T_i	T_d	K_p	T_i	T_d	K_p	T_i	T_d
q_1 (Pos.)	1	0.1	0	10	1	0	1	0.1	0
q_2 (Pos.)	1	0.1	0	10	1	0	1	0.1	0
q_3 (Pos.)	1	0.1	0	10	1	0	10	1	0
q_1 (Vel.)	1000	100	0	100	100	0	1000	100	0
q_2 (Vel.)	1000	1	0	100	100	0	1000	100	0
q_3 (Vel.)	100	100	0	100	100	0	100	100	0

I valori indicati in tabella evidenziano come si è preferito utilizzare controllori molto aggressivi nel *loop* interno, consentendo un rapido inseguimento della velocità, riducendo al minimo l'effetto dell'azione integrale. Guadagno minori sono stati impiegati nei loop esterni, dove un guadagno proporzionale unitario fornisce una risposta adeguata, se coadiuvata da un'azione integrale più spinta, tale da portare la risposta a regime con errore pressoché nullo. Si è deciso di spegnere completamente l'azione derivativa, responsabile di continue oscillazioni e una diminuzione delle performance complessive.

Infine, per evitare picchi notevoli di accelerazione e forza/coppia nei primi istanti del transitorio, è stata introdotta un'azione di *feed-forward*: la forza/coppia nell'istante iniziale, nota dai calcoli sul modello ideale, è aggiunta alla variabile di controllo, in modo tale che il manipolatore sia soggetto fin da subito ad un'azione di controllo simile a quella che garantisce il movimento desiderato, senza aspettare che il controllore regoli la sua azione sulla base dello scarto tra *set-point* e variabile di processo (la posizione lineare/angolare del link controllato).

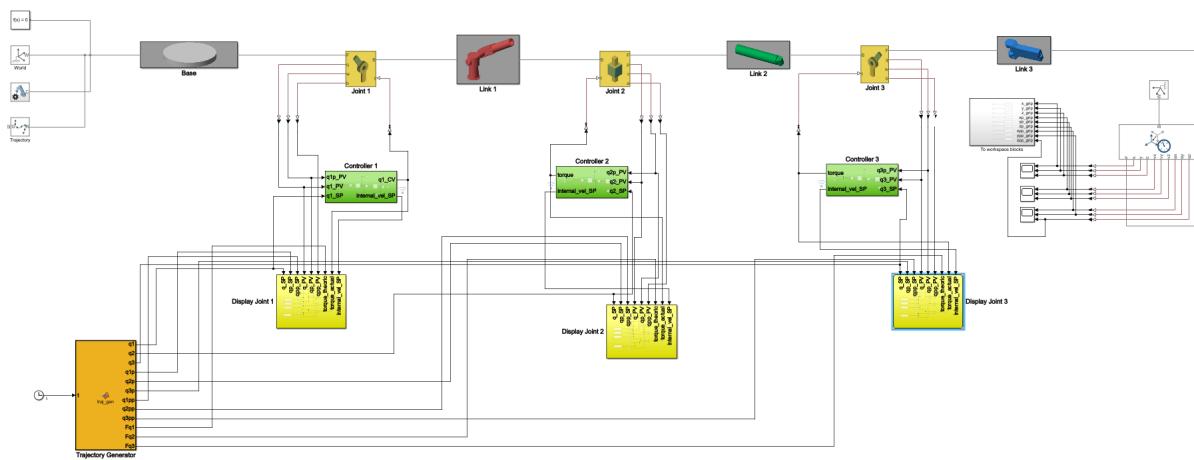


Fig. 32. Schema di controllo completo del manipolatore.

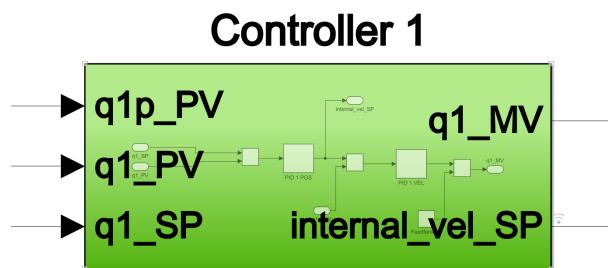


Fig. 33. Porte di input e output del controllore.

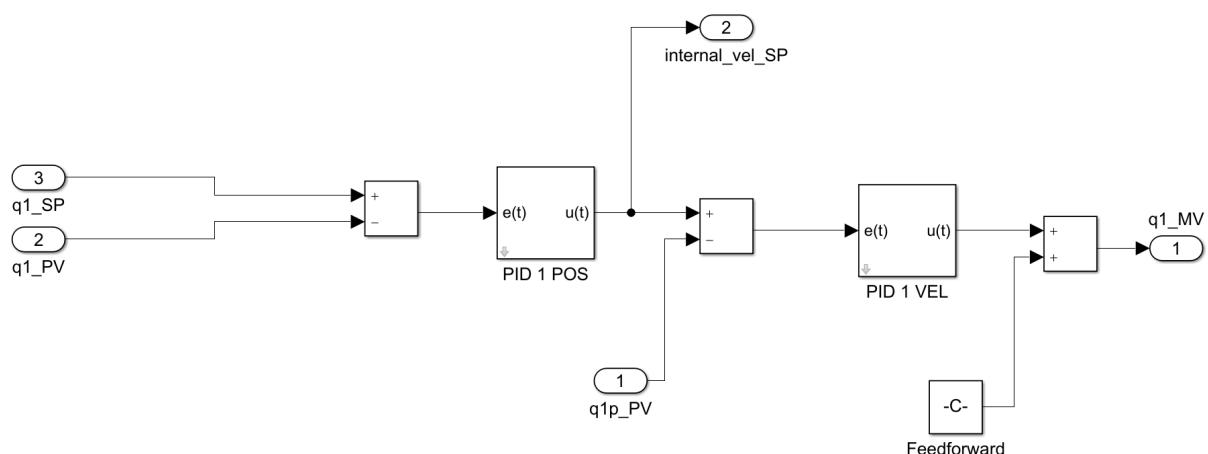


Fig. 34. Schema del tipo di controllore impiegato.

Questa soluzione ha garantito un notevole miglioramento delle prestazioni, consentendo in taluni casi di non eccedere i limiti di forza/coppia massima imposti agli attuatori.

5.1 Proposte per un Tuning Avanzato

Come ulteriore passaggio si è cercato di trovare una metodologia per tarare i parametri dei controllori secondo uno schema più rigoroso, tentando ad esempio di identificare i processi sotto controllo mediante *identificazione ai minimi quadrati* (sia la componente del processo nel loop interno, sia quella nel loop esterno). Ottenere un modello dei processi consente infatti di eseguire *offline* una *routine* di ottimizzazione dei parametri del controllore, definendo una funzione di costo che tenga in conto sia dell'errore durante il transitorio, sia dello sforzo di controllo.

Non vengono presentati risultati, in quanto l'ottimizzazione non ha portato ai vantaggi auspicati, rendendo comunque preferibile un *tuning* empirico basato sul significato dei termini proporzionale, integrale, derivativo di un controllore PID.

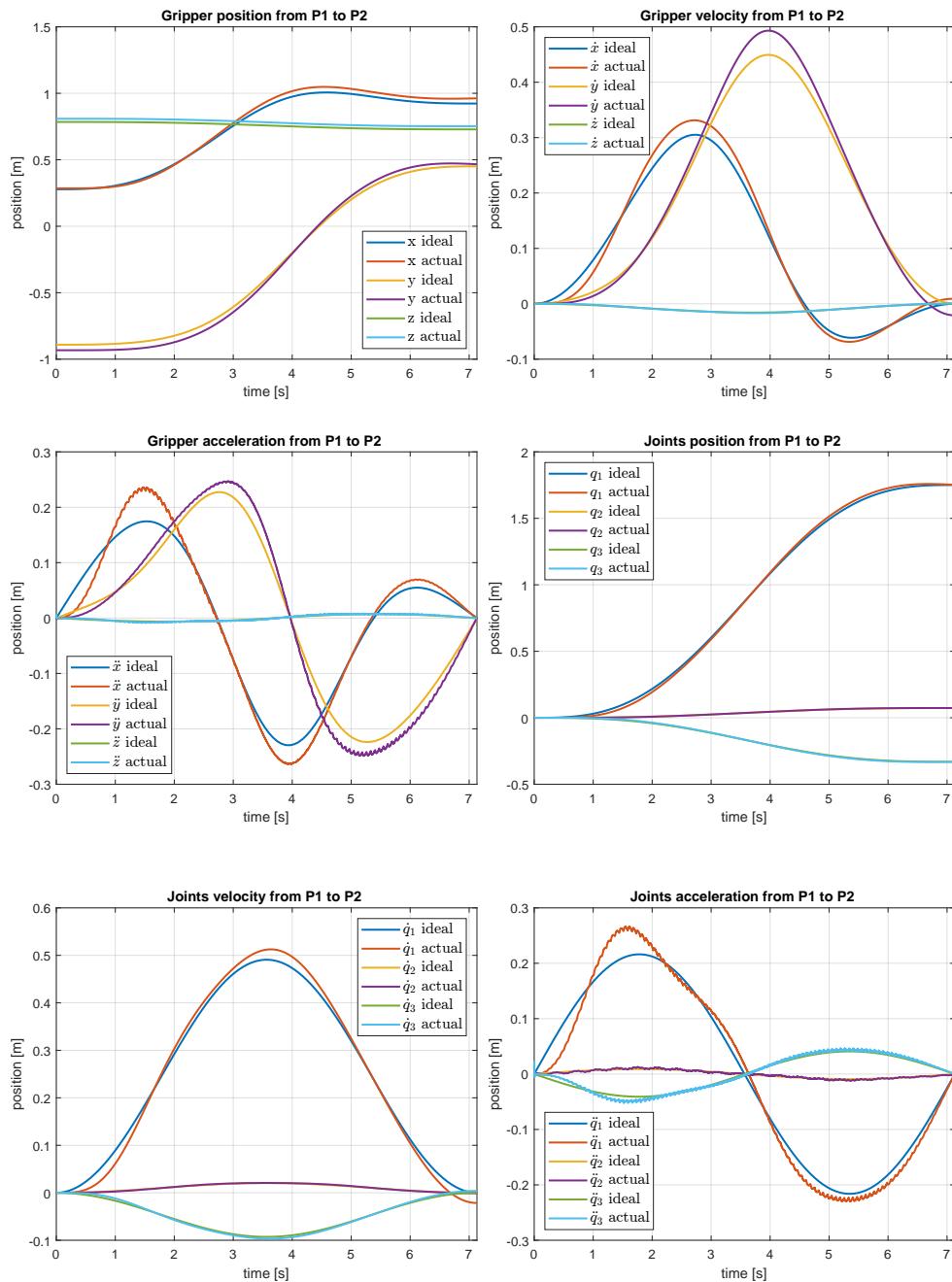
Altre soluzioni, non esplorate in quanto i risultati sono già soddisfacenti, potrebbero prevedere l'impiego di *set-point weight* per smorzare i *proportinal kick* nella risposta e magari una taratura secondo regole di *tuning* rigorose (Ziegler-Nichols, CHR, Haalman, Kapa-Tau,...), previa identificazione dei processi sotto controllo.

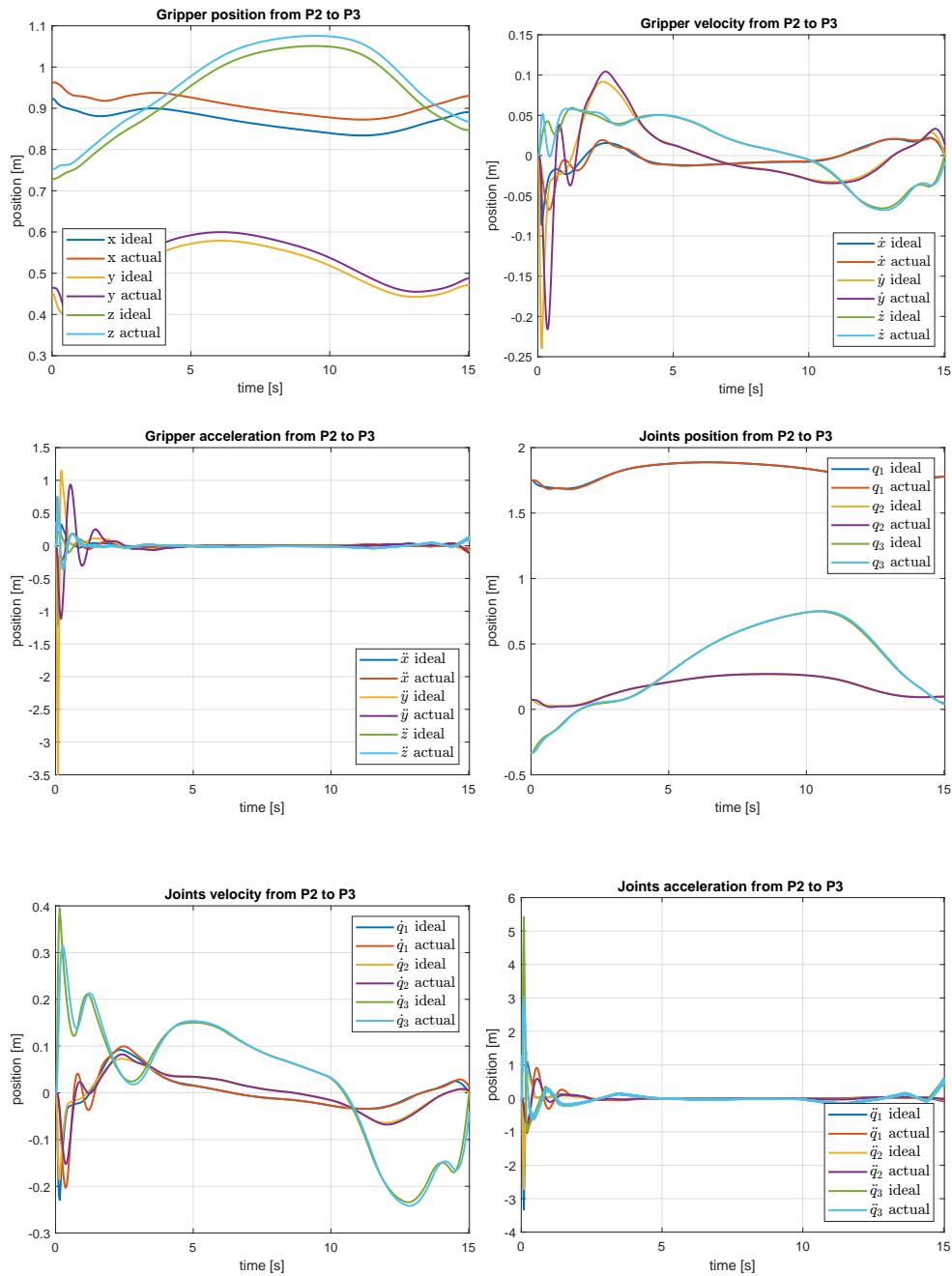
5.2 Risultati

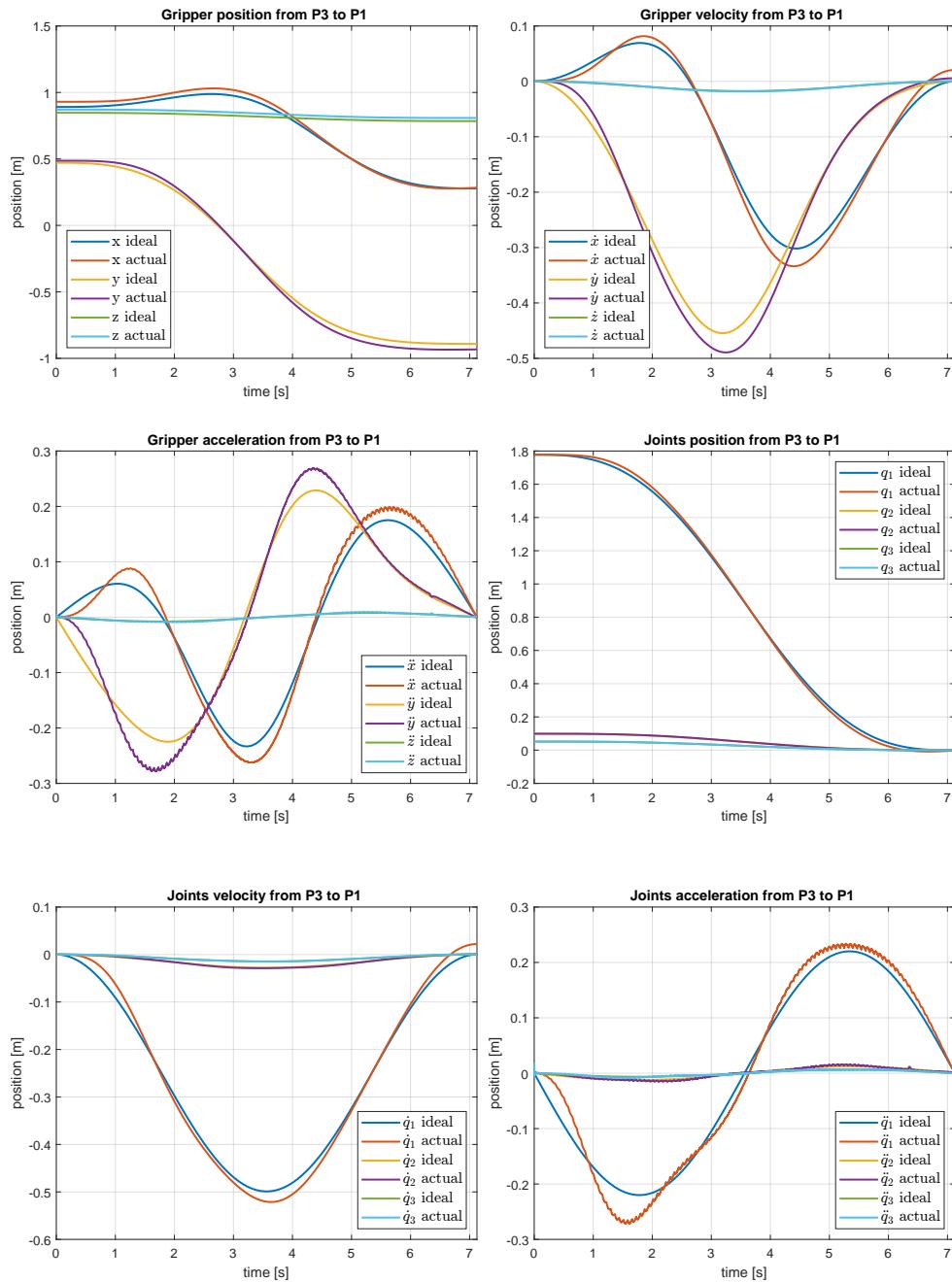
Di seguito sono riportati i risultati ottenuti controllando il movimento del robot durante l'esecuzione dei *task*. Si evince come il sistema di controllo adottato e la taratura implementata consentano un ottimo inseguimento di posizione e velocità, consentendo al robot di seguire la traiettoria desiderata. Accelerazioni e coppie risultano anch'esse coerenti con quanto atteso dallo studio teorico, sebbene vengano introdotte oscillazioni (di ampiezza relativamente bassa), imputabili alle operazioni di derivazione eseguite dal solutore.

In generale si osserva come le azioni nei giunti rispettino sempre i limiti di accelerazioni e coppie/forze riportati nella Sez. 1.2.

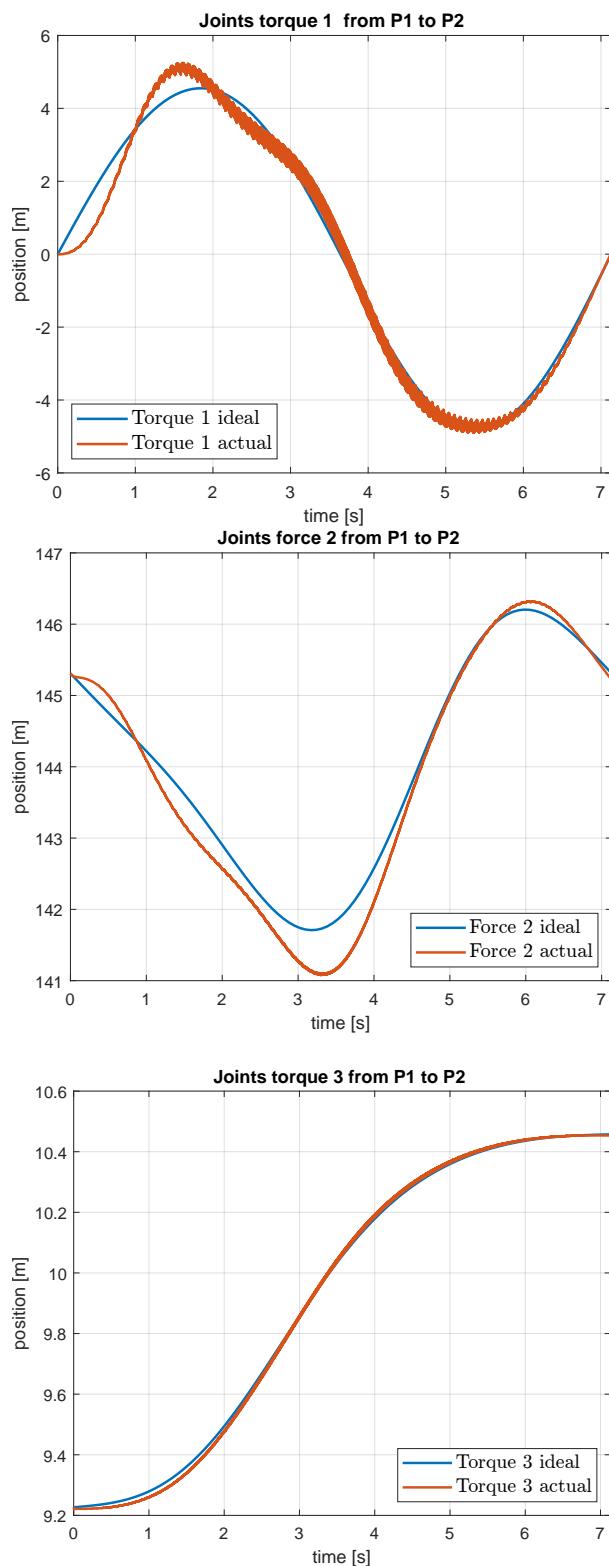
5.2.1 Posizione, velocità e accelerazione

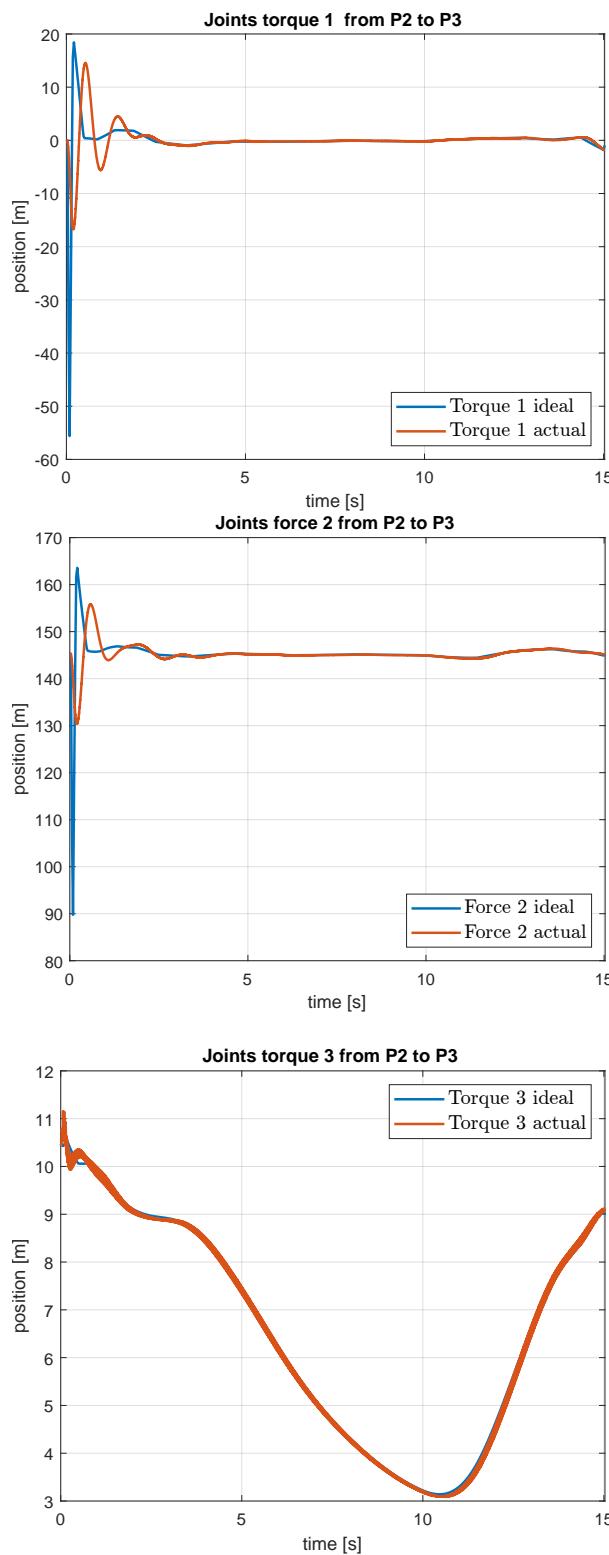


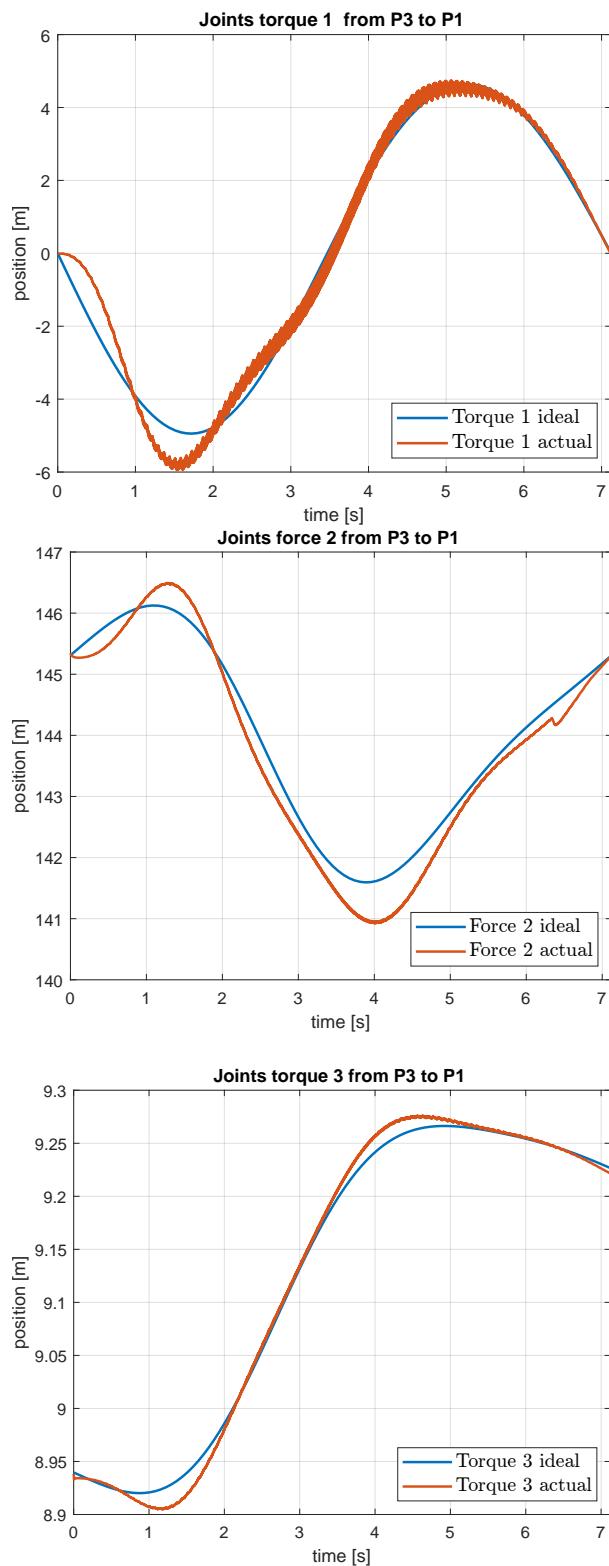




5.2.2 Azioni nei giunti







Conclusioni

In conclusione si sottolinea come il codice sviluppato nel corso di questo progetto risulti flessibile anche in presenza di variazioni sui limiti degli attuatori o interazioni con azioni esterne: modificando tali parametri si ottengono automaticamente i risultati corrispondenti. Come già presentato nella Sez. 1, nel *Task 1* e nel *Task 3* l'algoritmo calcola automaticamente il minimo tempo di azionamento e nel *Task 2* verifica autonomamente la minima spaziatura tra i nodi della traiettoria che è lecito tenere per rispettare i limiti fisici.

Per proteggere il sistema da eventuali problematiche in cui potrebbe incorrere se gli venisse assegnata una traiettoria al di fuori del *workspace*, un ulteriore *upgrade* sarebbe l'implementazione di un algoritmo di verifica della percorribilità dell'intera traiettoria, propnendo eventualmente soluzioni alternative (passaggio per opportuni punti intermedi definendo una traiettoria sub-ottimale, ad esempio).

Dal confronto tra l'approccio jacobiano e quello matriciale si evince come quest'ultimo risulti più completo, riducendo le approssimazioni e consentendo uno studio più semplice dei singoli *link* grazie all'utilizzo di strutture dati tensoriali.

Il controllo decentralizzato tramite regolatori PID in cascata si è rivelato adeguato in questa situazione, non rendendo necessari ulteriori accorgimenti quali *set-point weight* o un'identificazione dettagliata del sistema. La taratura empirica attuata consente di seguire con elevata precisione i riferimenti di traiettoria e velocità, rispettando i limiti fisici degli attuatori. Nel caso di movimentazioni più complesse l'implementazione di ulteriori *improvement*, come l'ottimizzazione dei parametri di *tuning* o un'identificazione più approfondita del sistema tempo-variante, potrebbe rivelarsi efficace per l'ottimizzazione della taratura dei controllori.

Bibliografia

1. Legnani, G. & Fassi, I. *Robotica industriale* (CittàStudi, 2019).
2. MathWorks. *Equation Solving Algorithms* <https://it.mathworks.com/help/optim/ug/equation-solving-algorithms.html>.
3. Systems, D. & MathWorks. *Simscape Multibody Link* <https://www.solidworks.com/it/partner-product/simscape-multibody-link>.