

Università degli studi di Modena e Reggio Emilia
Dipartimento di Ingegneria “Enzo Ferrari”
Corso di Laurea Magistrale in Ingegneria Informatica



Automazione & Robotica

PROGETTO DI CORSO

Introduzione:

Il progetto presentato espone un esempio di simulazione del movimento di un robot PUMA560 avente 6 gradi di libertà.

Il manipolatore in questione ha tutti giunti rotoidali, questo permette di comprendere meglio quali sono i parametri, variabili e fissi.

Definizione del manipolatore:

Per prima cosa è stato definito il modello del robot

```
1. function puma560_model = load_puma560_model(d, a, alpha)
2.     for i=1:6
3.         L(i) = Link('d', d(i), 'a', a(i), 'alpha', alpha(i), 'revolute');
4.     end
5.     puma560_model = SerialLink(L, 'name', 'puma560_model');
6. end
```

A questa funzione vengono passati i tre vettori reperibili dalla matrice DH. Come si può vedere, non compare il parametro ϑ all'interno della funzione *Link*, questo deriva dal fatto che si hanno solamente giunti rotoidali. La funzione *Link* serve a definire i vari bracci meccanici che compongono il manipolatore seriale.

Cinematica Diretta:

Dopo aver definito il modello del manipolatore è stato possibile calcolare la matrice di trasformazione omogenea della cinematica diretta tramite la funzione *fkine*, alla quale viene passato il modello definito come sopra, ed il vettore delle variabili di giunto.

Oltre all'utilizzo della funzione del toolbox è stato implementato il procedimento manuale, in modo da poter confrontare i risultati della funzione *fkine*.

Dalla convenzione di Denavit – Hartenberg possiamo vedere che è possibile esprimere la posizione di un giunto rispetto al precedente con solo 4 parametri.

In particolare, la posizione di un giunto i -esimo, rispetto al precedente ($i-1$) è esprimibile come:

- rotazione attorno l'asse x
- traslazione lungo l'asse x
- rotazione lungo l'asse z
- traslazione lungo l'asse z

$$T_{Rz}(\vartheta_i) = \begin{bmatrix} \cos \vartheta_i & -\sin \vartheta_i & 0 & 0 \\ \sin \vartheta_i & \cos \vartheta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{Tz}(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$T_{Rx}(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{Tx}(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(le rotazioni sono effettuate in terna corrente)

Ottenendo che la matrice omogenea complessiva che descrive la posizione del giunto i esimo rispetto al precedente si può scrivere come:

$$T_{i-1,i}(\vartheta_i, d_i, a_i, \alpha_i) = T_{Rz}(\vartheta_i)T_{Tz}(d_i)T_{Tx}(a_i)T_{Rx}(\alpha_i)$$

$$T_{i-1,i}(\vartheta_i, d_i, a_i, \alpha_i) = \begin{bmatrix} \cos \vartheta_i & -\cos \alpha_i \sin \vartheta_i & \sin \alpha_i \sin \vartheta_i & a_i \cos \vartheta_i \\ \sin \vartheta_i & \cos \alpha_i \cos \vartheta_i & -\sin \alpha_i \cos \vartheta_i & a_i \sin \vartheta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nel codice ho implementato direttamente la matrice finale di trasformazione omogenea, omettendo i calcoli delle quattro matrici riportate precedentemente.

Con il procedimento indicato da D-H, è possibile determinare un modo sistematico per associare, in posture opportune, $n+1$ sistemi di riferimento S_i (con $i=0, \dots, n$) agli $n+1$ link L_i (compreso il link base L_0) e calcolare la seguente trasformazione:

$$T_{0,n}(q) = T_{0,n}(q_1, q_2, \dots, q_n) = \prod_{i=1}^n T_{i-1,i}(q_i) = T_{0,1}(q_1)T_{1,2}(q_2) \dots T_{n-1,n}(q_n)$$

dove $q = [q_1 \quad q_1 \quad \dots \quad q_n]^T \in \mathbb{R}^n$ con

$$q_i = \begin{cases} \vartheta_i & \text{se il giunto e' rotoideale} \\ d_i & \text{se il giunto e' prismatico} \end{cases}$$

Da questo risultato è stato possibile andare a calcolare manualmente le sei matrici di trasformazione omogenea, infatti, sapendo di avere solamente giunti rotoideali, alla funzione di calcolo viene fornita la matrice D-H ed al posto del vettore ϑ viene posto come input il vettore delle variabili di giunto q_i .

Eseguendo il codice implementato in questo modo è possibile verificarne la correttezza.

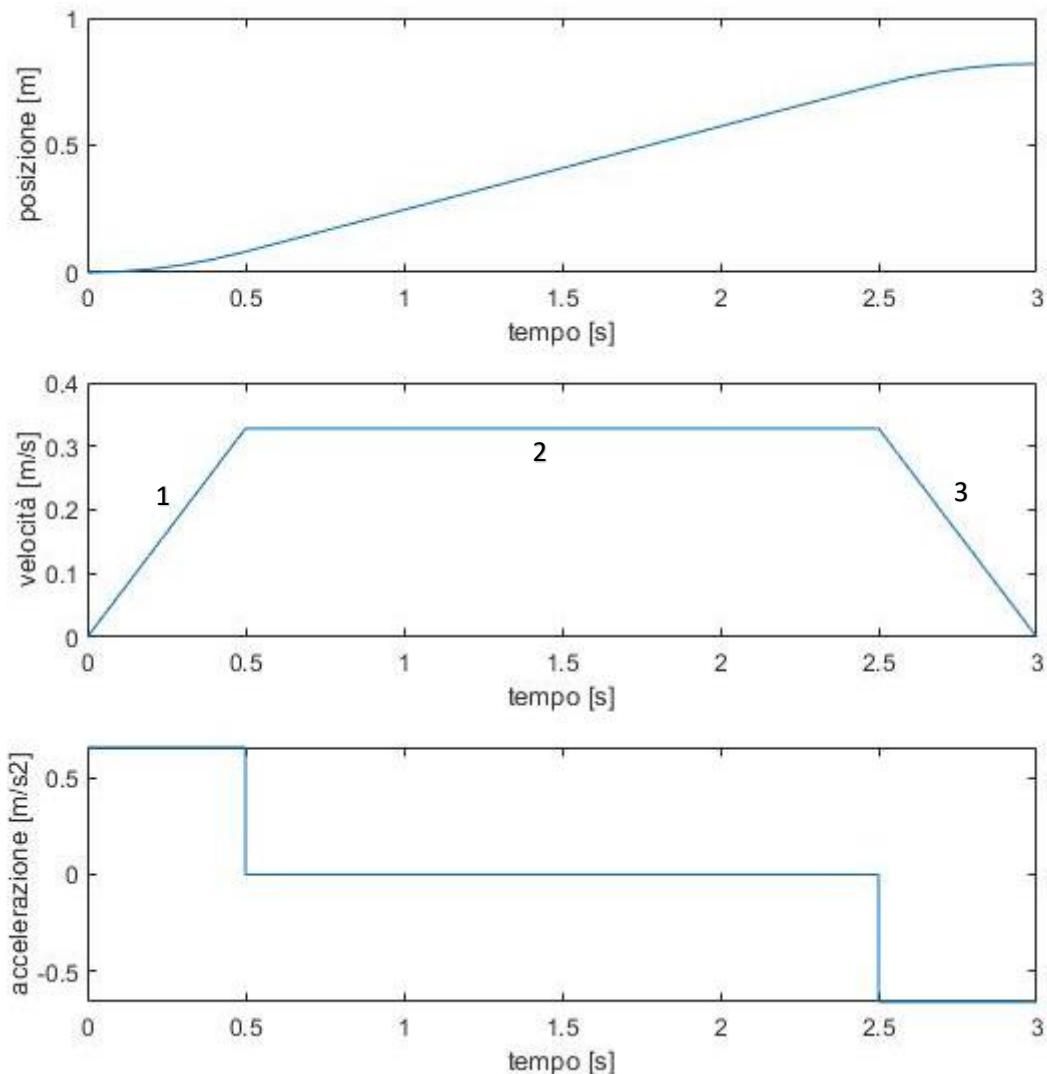
Pianificazione Della Traiettoria:

Dalle linee guida del progetto, sappiamo che la legge di moto per il movimento del manipolatore deve essere trapezoidale.

Per costruire tale legge, è stato effettuato il seguente procedimento:

sapendo che il tempo totale per compiere il movimento è suddiviso in questo modo

- 0,5 [s] per il tratto iniziale ad accelerazione costante
- 2 [s] per il tratto ad accelerazione nulla
- 0,5 [s] per il tratto iniziale a decelerazione costante
- Tempo totale per il tragitto: 3 s



Dal grafico della velocità si può calcolare che:

la distanza totale, che è possibile ricavarla tramite i dati della traccia, come

$$\text{spazio totale} = \|p_f(1 : 3) - p_i(1 : 3)\| = 0.8201 = D$$

$$D = \left(\frac{1}{2} \cdot b \cdot h\right) + (b \cdot h) + \left(\frac{1}{2} \cdot b \cdot h\right)$$

Dove:

- Il primo termine è l'area del triangolo (1)
- Il secondo termine è l'area del rettangolo (2)
- Il terzo è l'area del triangolo (3)

$$D = \left(\frac{1}{2} \cdot T_1 \cdot V_{max}\right) + (T_2 \cdot V_{max}) + \left(\frac{1}{2} \cdot T_3 \cdot V_{max}\right)$$

Ricavando V_{max} , sapendo che T_1 T_2 T_3 sono noti:

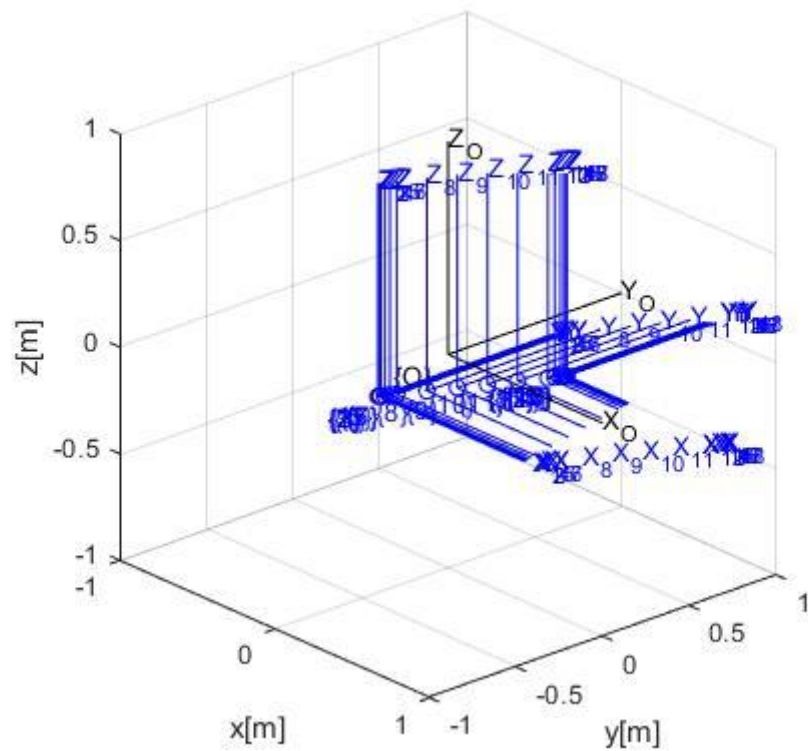
$$V_{max} = \frac{D}{2,5}$$

Per calcolare l'accelerazione, sapendo che nel tratto centrale essa è nulla,

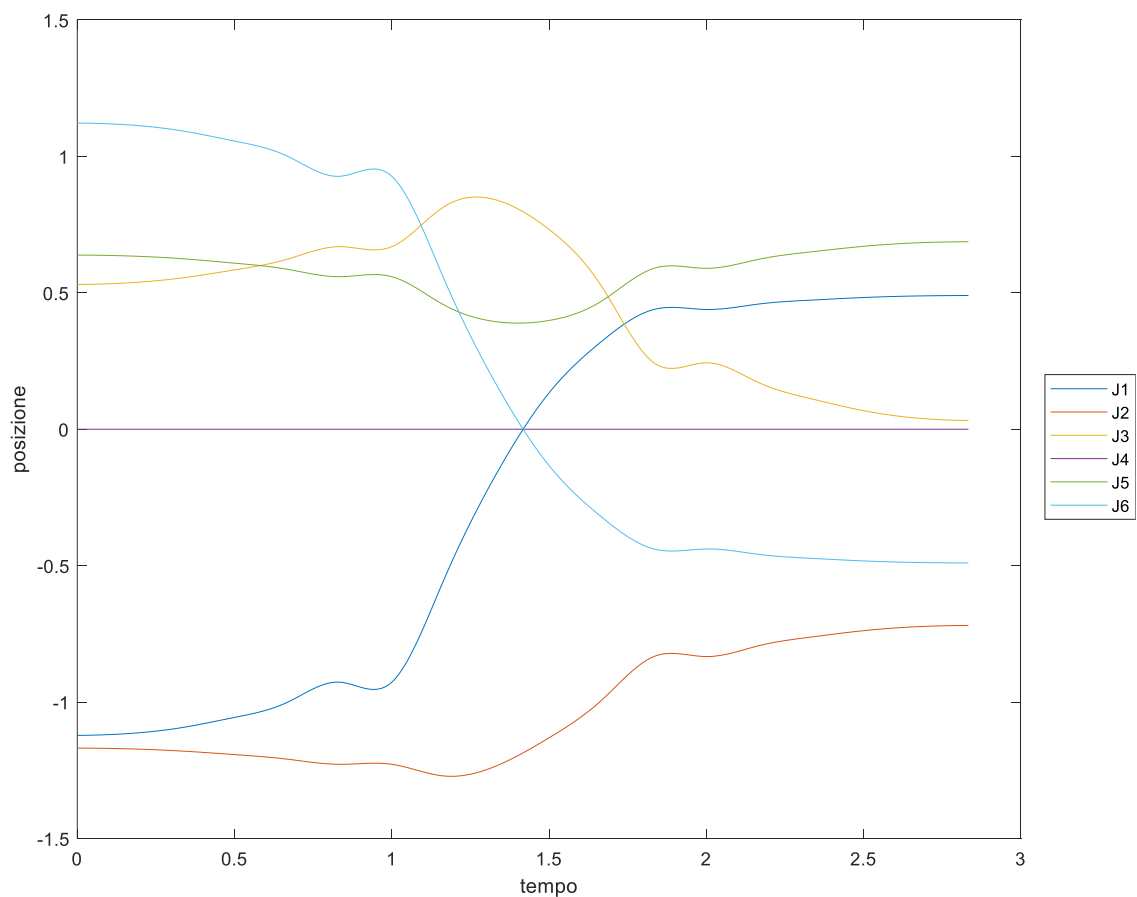
$$a = \frac{V_{max}}{T_1}$$

Trovati quindi questi parametri, è stata usata per tre volte la funzione del moto rettilineo uniformemente accelerato, nella quale però, nel tratto centrale, si impone accelerazione nulla.

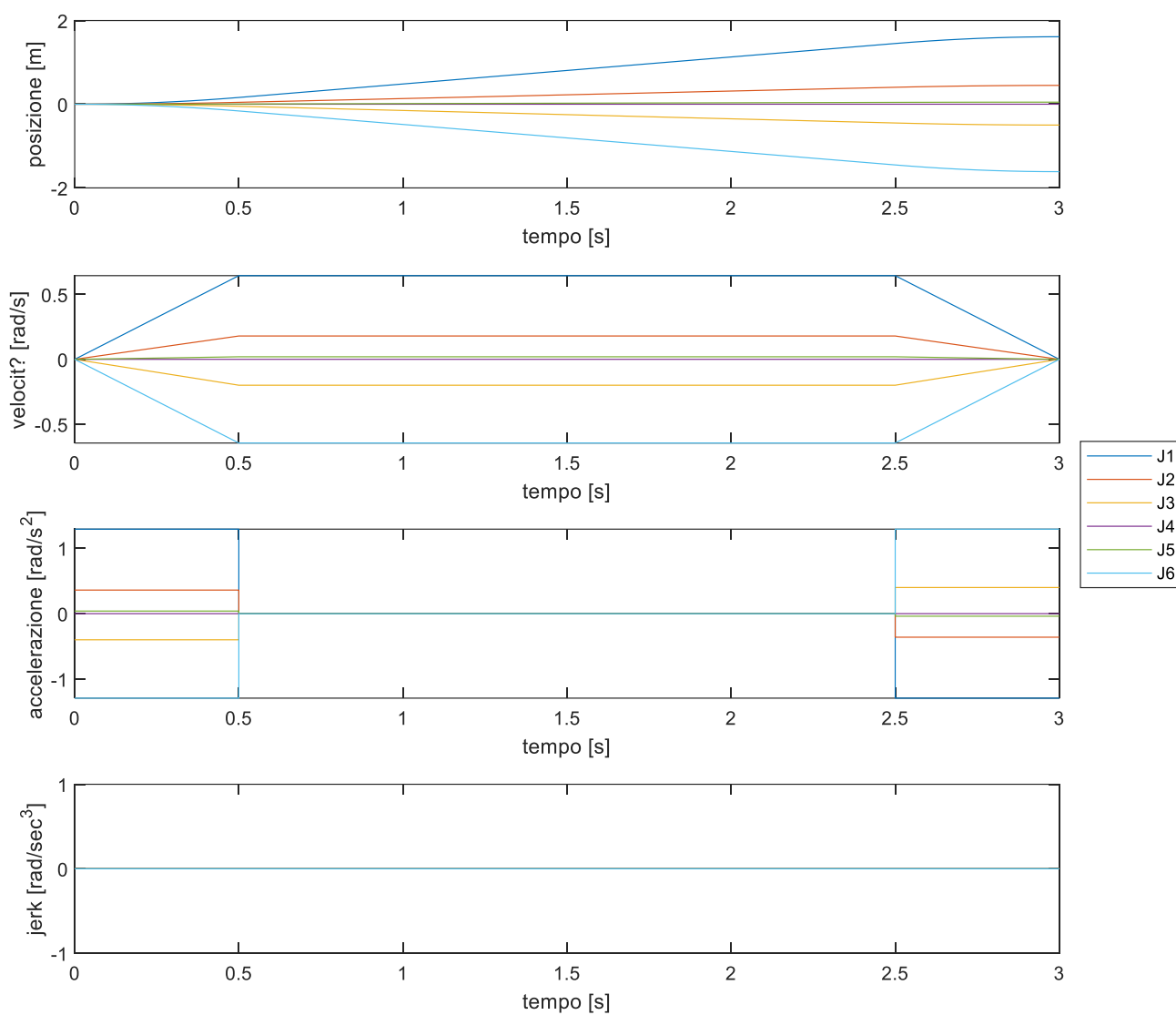
Grazie al modello del robot definito al secondo punto, nella cinematica diretta, è stato possibile tramite la funzione *ikine* di passare dallo spazio operativo a quello dei giunti e far muovere in linea retta il nostro end-effector del robot.



Definita la legge di moto è stata calcolata la traiettoria percorsa dai singoli giunti per arrivare al punto finale di arresto del movimento.



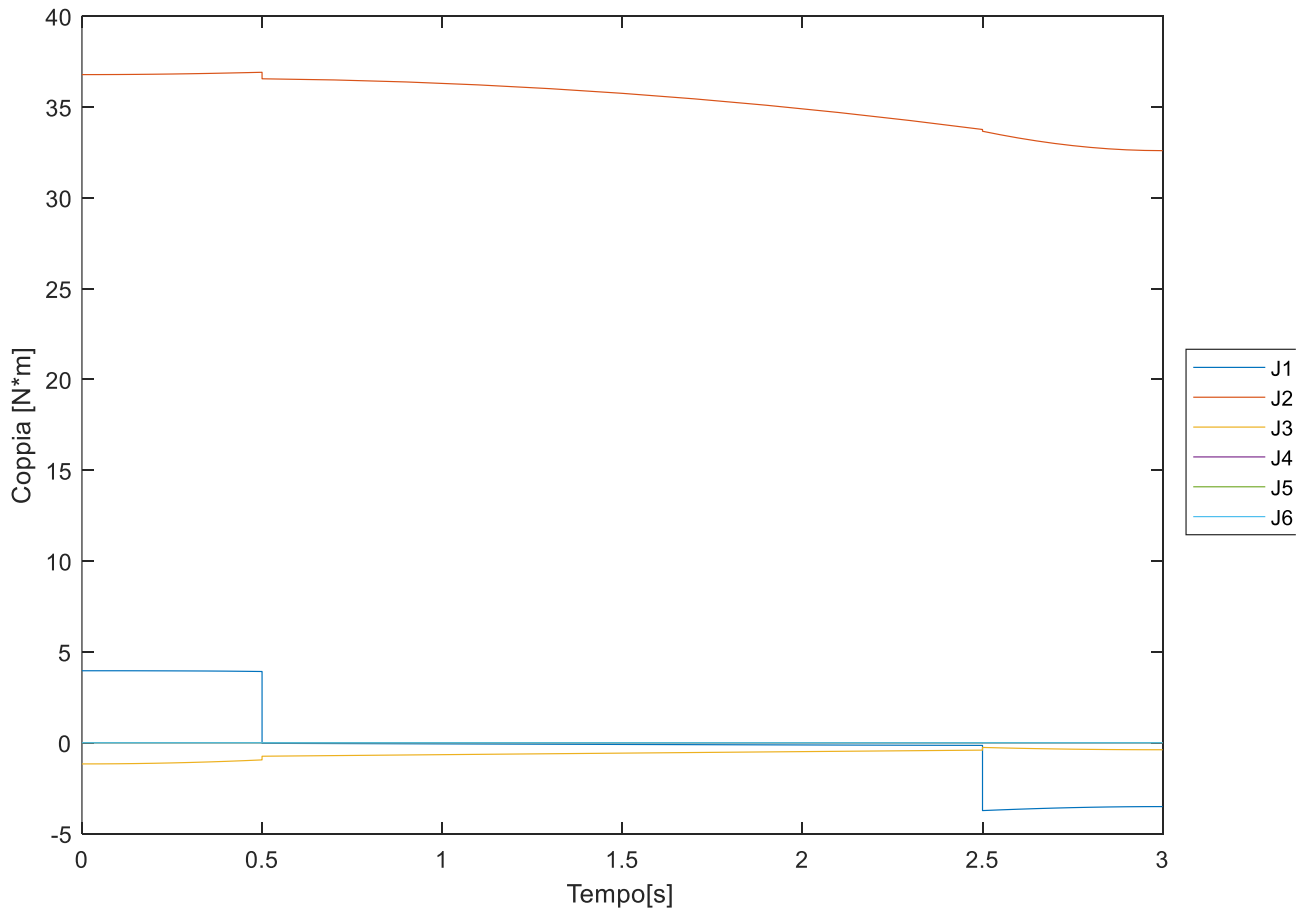
Una volta trovate le posizioni dei giunti tramite la funzione “*grafici_spazio_dei_giunti_e_coppie*” sono state calcolate le posizioni, velocità, accelerazioni e jerk dei singoli giunti.



Determinazione delle coppie necessarie per gli attuatori:

Le coppie ai giunti degli attuatori sono state calcolate grazie ai risultati del punto precedente tramite la funzione messa a disposizione da MATLAB, chiamata *SerialLink.rne Inverse dynamics*.

Grazie alla definizione e caricamento all'interno del modello del manipolatore dei parametri fisici, come momenti di inerzia e massa, è possibile andare ad estrapolare tali coppie:



Strategie per il controllo del manipolatore (*punti 5 e 6 della traccia di progetto*):

Ultimati i precedenti punti, si è passato alla stesura della strategia di controllo per il robot, in questo caso, si richiedeva due tipologie di controllo:

1. Controllore PD con compensazione di gravità
2. Controllore a dinamica inversa

Per entrambi i controllori partiamo dalla conoscenza del modello di Eulero Lagrange:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}^T(\mathbf{q})\mathbf{F}_{\text{ext}}$$

Per il primo tipo di controllo, si è ricalcato lo stesso procedimento mostrato durante l'esercitazione. Per prima cosa si è proceduto al calcolo della traiettoria da far compiere al robot tramite la funzione canonica trapezoidale. Una volta ricavata la traiettoria da dover seguire si è fatto partire il robot da una posizione errata, che però essendo fornita nello spazio operativo, tramite la funzione di cinematica inversa, si sono trovati i valori delle variabili di giunto.

Impostati i giunti su tali posizioni, è stata implementata l'azione di controllo proporzionale – derivativa con compensazione di gravità.

Purtroppo, dai grafici il controllore non agisce in modo corretto e la dinamica del manipolatore risulta totalmente errata.

Per la seconda tipologia di controllo, si è ricavato dal modello di Eulero Lagrange l'informazione sulla matrice dei termini inerziali, degli effetti di Coriolis, ed anche gli effetti introdotti dalla gravità terrestre.

Noti questi valori lo scopo è stato trovare un valore τ affinché si vada a compensare tutte le grandezze nel modello.

In questo caso gli andamenti di alcuni giunti risultano essere coerenti o molto vicini al moto desiderato ma purtroppo anche in questo caso il movimento generale che ne scaturisce non è coerente con quello desiderato e stabilito al punto 3