



2022/2023

# Tesina Robotica

ALGORITMO DI INVERSIONE CINEMATICA PER UN MANIPOLATORE PLANARE

CHIARA CAPORALE : 67463 – MANUELE CAPECE : 67468

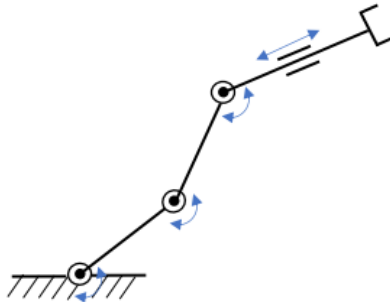
## Sommario

1.DESCRIZIONE DEL MANIPOLATORE	2
2.CINEMATICA DIRETTA	2
2.1.Procedura operativa per ogni terna:	2
2.3.Tabella parametri completa:	4
3.JACOBIANO GEOMETRICO	5
4.CARATTERIZZAZIONE DELLA RIDONDANZA	6
4.1.Vettore di spazio operativo	6
4.2.Compito assegnato	7
4.3.Configurazione iniziale	7
5.SPECIFICAZIONE DELLA LEGGE DI MOTO	9
6.ALGORITMO DI INVERSIONE CINEMATICA	12
6.1.Moti interni	13
6.2.Algoritmo di inversione cinematica	13
7.RISULTATI OTTENUTI	14
7.1.Simulazione del compito primario	15
7.1.Simulazione del compito primario e secondario	16
7.3.Confronto dei risultati delle due simulazioni	18
7.4.Conclusioni	18
8.APPENDICE	19
8.1.Algoritmo di inversione cinematica	19
8.1.Funzione per il calcolo della cinematica diretta	20
8.2.Funzione per il calcolo dello Jacobiano geometrico	20

# 1.DESCRIZIONE DEL MANIPOLATORE

In figura è rappresentata una struttura meccanica a catena cinematica aperta, una sequenza di bracci che connettono due estremi: l'estremo fisso a terra e l'end-effector. Nello specifico si tratta di un manipolatore planare formato da **tre giunti rotoidali**: che impongono movimenti di rotazione intorno ad un'asse e un **giunto prismatico**: collegato all'end-effector che genera traslazioni tra i due bracci connessi.

Ogni giunto descrive un grado di mobilità, per cui dispone di **quattro gradi di mobilità ( $n = 4$ )**.



## 2.CINEMATICA DIRETTA

L'obiettivo è di determinare posizione e orientamento dell'end-effector, rispetto alla terna base, in funzione delle variabili di giunto.

Il tutto è stato realizzato utilizzando la convenzione di **Denavit-Hartenberg** che fornisce una procedura per calcolare le matrici di trasformazione omogenea grazie all'utilizzo di quattro parametri:

- $a_i$  = distanza fra le origini  $O_{i-1}$  e  $O_i$  lungo l'asse  $x_{i-1}$
- $d_i$  = distanza fra le origini  $O_{i-1}$  e  $O_i$  lungo l'asse  $z_{i-1}$ . (variabile di giunto prismatico)
- $\alpha_i$  = angolo che ruota intorno all'asse  $x_i$  per sovrapporre l'asse  $z_{i-1}$  a  $z_i$
- $\theta_i$  = angolo che ruota intorno all'asse  $z_{i-1}$  per sovrapporre l'asse  $x_{i-1}$  a  $x_i$  (variabile di giunto rotoidale)

### 2.1.Procedura operativa per ogni terna:

L'asse  $z$  di ogni giunto rotoidale è uscente dal foglio. Ciò non accade per il giunto prismatico, in quanto effettua un moto di traslazione lungo tale asse che risulterà perpendicolare all'asse  $z$  del giunto rotoidale precedente.

$\Sigma_0 \equiv \Sigma_b$  :

Si ha ampia libertà di scelta. Per semplicità si è fissato l'origine  $O_0$  coincidente con l'asse  $z_0$  e gli assi  $x_0$  e  $y_0$  tali da completare una terna levogira.

$\Sigma_1$ :

$z_0$  e  $z_1$  sono due assi consecutivi e paralleli, la normale comune non è univocamente definita e per comodità la si è scelta allineata al braccio 1 in modo tale da poter annullare il parametro  $d_1$ .

L'origine  $O_1$  è posizionata nell'intersezione tra  $z_1$  e la normale comune agli assi  $z_0$  e  $z_1$ .

L'asse  $x_1$ , per convenzione, giace lungo la normale con verso positivo che va da dal giunto 1 al giunto 2.

Infine l'asse  $y_1$  completa la terna levogira.

Gli unici parametri definiti sono:  $a_1$  lunghezza del braccio 1 e l'angolo  $\theta_1$  che corrisponde proprio alla variabile di giunto.

Tabella parametri:

BRACCIO	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
<b>1</b>	<b><math>a_1</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_1</math></b>

**$\Sigma_2$ :**

L'origine  $O_2$ , assi  $x_2, y_2$  e i parametri sono stati definiti in maniera analogo alla  $\Sigma_1$

Tabella parametri:

BRACCIO	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
<b>1</b>	<b><math>a_1</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_1</math></b>
<b>2</b>	<b><math>a_2</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_2</math></b>

**$\Sigma_3$ :**

È individuata da un giunto prismatico. L'asse  $z_3$  è perpendicolare all'asse  $z_2$  e l'origine  $O_3$  è posizionato nell'intersezione tra  $z_3$  e la normale comune agli assi  $z_2$  e  $z_3$ .

La normale comune, su cui giace  $x_3$  con verso positivo dal giunto2 al giunto3, è scelta parallela all'asse  $x_2$  in modo tale da annullare il parametro  $a_3$ .

Infine l'asse  $y_3$  completa la terna levogira

Gli unici parametri definiti sono:  $\alpha_3 = \pi/2$  l'angolo per far sovrapporre  $z_2$  a  $z_3$  intorno a  $x_3$ , e l'angolo di rotazione  $\theta_3$  lungo  $z_2$  in modo tale da far sovrapporre  $x_2$  a  $x_3$

Tabella parametri:

BRACCIO	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
<b>1</b>	<b><math>a_1</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_1</math></b>
<b>2</b>	<b><math>a_2</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_2</math></b>
<b>3</b>	<b>0</b>	<b><math>\pi/2</math></b>	<b>0</b>	<b><math>\theta_3</math></b>

**$\Sigma_4 \equiv \Sigma_e$ :**

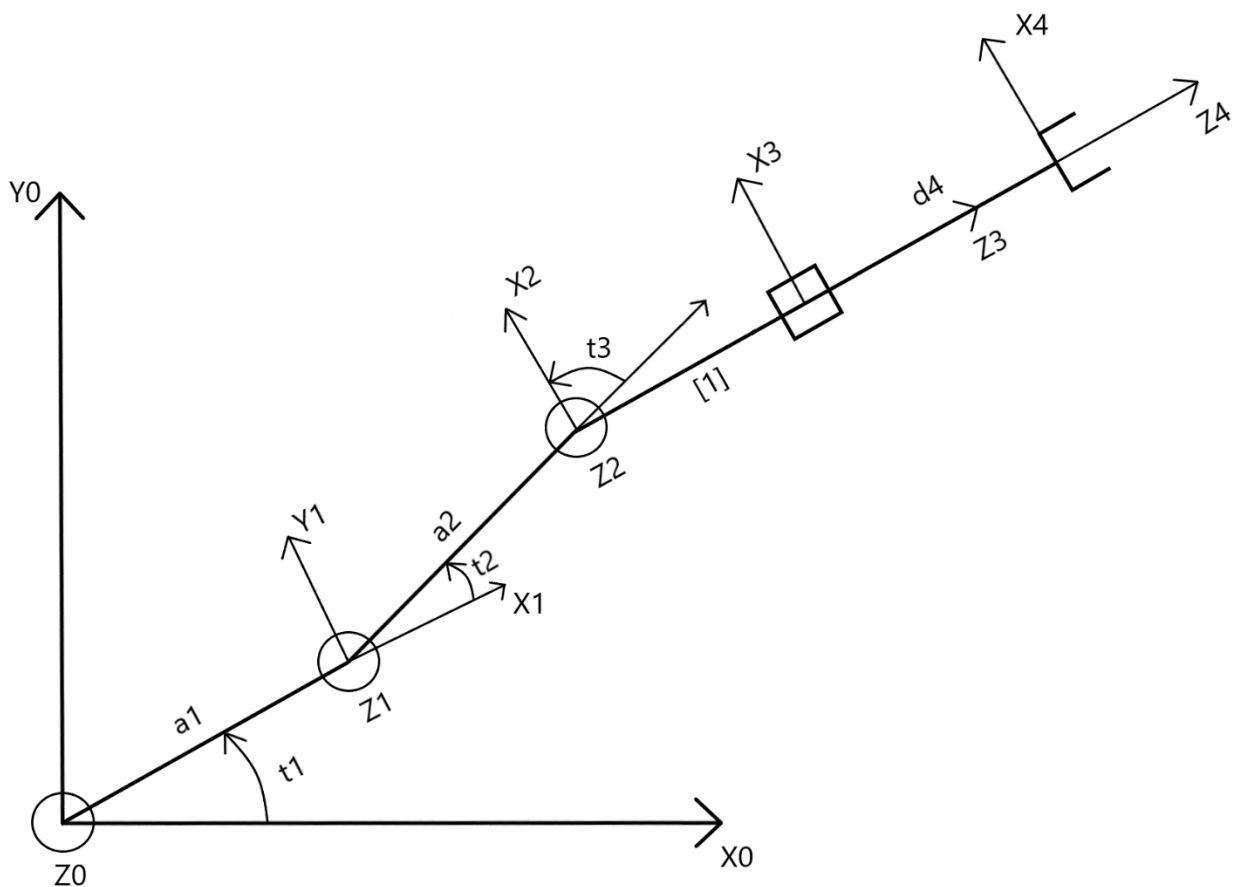
Si è scelto di posizionare l'origine  $O_4$  all'inizio della pinza, con  $z_4$  parallelo a  $z_3$  in modo tale da non annullare il parametro  $d_4$  che rappresenta proprio la variabile di giunto.

Per convenzione si sceglie  $x_4$  ortogonale a  $z_3$ , in modo tale da annullare i restanti parametri

### 2.3.Tabella parametri completa:

BRACCIO	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
<b>1</b>	<b><math>a_1</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_1</math></b>
<b>2</b>	<b><math>a_2</math></b>	<b>0</b>	<b>0</b>	<b><math>\theta_2</math></b>
<b>3</b>	<b>0</b>	<b><math>\pi/2</math></b>	<b>0</b>	<b><math>\theta_3</math></b>
<b>4</b>	<b>0</b>	<b>0</b>	<b><math>d_4</math></b>	<b>0</b>

Rappresentazione grafica delle terne:



Dopo aver costruito la tabella dei parametri di Denavit-Hartenberg, si calcolano le matrici di trasformazione omogenea  $A_i^{i-1}(q_i)$  per  $i = 1,2,3,4$

$$\begin{aligned}
\mathbf{A}_1^0 &= \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{A}_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{A}_3^2 &= \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{A}_4^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Infine si determina la funzione di cinematica diretta:  $\mathbf{T}_4^0 = \mathbf{A}_4^0 = \mathbf{A}_4^3 \mathbf{A}_3^2 \mathbf{A}_2^1 \mathbf{A}_1^0$

che fornisce posizione e orientamento della terna utensile  $\Sigma_4$  rispetto alla terna base  $\Sigma_0$ . Quindi nel nostro caso la terna base coincide con la terna 0 e la terna n coincide con la terna utensile.

$$\mathbf{T}_4^0 = \begin{bmatrix} c_{123} & 0 & s_{123} & a_1 c_1 + a_2 c_{12} + d s_{123} \\ s_{123} & 0 & -c_{123} & a_1 s_1 + a_2 s_{12} - d c_{123} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 3.JACOBIANO GEOMETRICO

Calcolare lo Jacobiano del manipolatore in questione ci permette di valutare le relazioni che intercorrono tra le variabili di giunto e le loro velocità, per ricavarlo si utilizzano le informazioni note dalla cinematica diretta.

Definiamo lo Jacobiano del manipolatore planare a 4 giunti come segue:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{z}_0 \times (\mathbf{p}_4 - \mathbf{p}_0) & \mathbf{z}_1 \times (\mathbf{p}_4 - \mathbf{p}_1) & \mathbf{z}_2 \times (\mathbf{p}_4 - \mathbf{p}_2) & \mathbf{z}_3 \\ \mathbf{z}_0 & \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{0} \end{bmatrix}$$

In cui :

$$\begin{aligned} \mathbf{z}_0 = \mathbf{z}_1 = \mathbf{z}_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \\ \mathbf{p}_3 = \mathbf{p}_2 \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix} \mathbf{z}_3 = \begin{bmatrix} s_{123} \\ -c_{123} \\ 0 \end{bmatrix} \end{aligned}$$

Svolgendo i calcoli, in particolare i prodotti tra vettori, lo Jacobiano risultante è il seguente:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - d c_{123} & a_2 c_{12} + d s_{123} & d c_{123} & s_{123} \\ a_1 c_1 + a_2 c_{12} + d s_{123} & a_2 s_{12} - d c_{123} & d s_{123} & -c_{123} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Eliminando le righe identicamente nulle lo Jacobiano di nostro interesse è quindi costituito da 4 colonne rappresentanti i 4 giunti e 3 righe. Le righe indicano le componenti di velocità posizionale e angolare. Le prime due righe sono le componenti di velocità posizionale  $\dot{p}_x$  e  $\dot{p}_y$  e l'ultima riga corrisponde alla velocità angolare lungo  $z$   $w_z$ . Essendo il manipolatore in questione un planare lo Jacobiano geometrico calcolato coincide con quello analitico. Questo perché la matrice di rotazione che esprime la rotazione elementare intorno a  $z$  coincide con l'unico angolo di Eulero corrispondente per lo Jacobiano analitico.

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - d c_{123} & a_2 c_{12} + d s_{123} & d c_{123} & s_{123} \\ a_1 c_1 + a_2 c_{12} + d s_{123} & a_2 s_{12} - d c_{123} & d s_{123} & -c_{123} \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

## 4.CARATTERIZZAZIONE DELLA RIDONDANZA

In questo capitolo esamineremo lo Jacobiano calcolato unitamente alla configurazione del manipolatore assegnato per capire quale algoritmo di inversione cinematica applicare.

### 4.1.Vettore di spazio operativo

Il vettore di spazio operativo viene ricavato prendendo le componenti di posizione dalla matrice di trasformazione  $T_4^0$  e la componente di orientamento come somma dei tre angoli di rotazione  $\theta_1$ ,  $\theta_2$  e  $\theta_3$

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} = \mathbf{k}(\mathbf{q}) = \begin{bmatrix} a_1 c_1 + a_2 c_{12} + d s_{123} \\ a_1 s_1 + a_2 s_{12} - d c_{123} \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix}$$

notiamo che  $\mathbf{x} \in R^3$  questo significa che i gradi di libertà del manipolatore sono 3.

## 4.2. Compito assegnato

Il compito assegnato consiste nel far percorrere all'organo terminale un segmento individuato da due punti estremi,  $P_i$  e  $P_f$ . Per specificare questo compito sarà necessario vincolare solo la posizione dell'organo terminale rispettivamente sull'asse  $X$  ed  $Y$ , in sostanza ci serviranno solo due gradi di libertà dei 3 disponibili.

Ricapitolando, il manipolatore assegnato è caratterizzato da 4 giunti e per specificare il compito assegnato sono necessari due gradi di libertà corrispondenti alla posizione dell'ee, quindi avremo:

- Gradi di **mobilità**  $n = 4$  (dimensione del vettore di spazio dei giunti  $\mathbf{q}$ )
- Gradi di **libertà**  $m = 3$  (dimensione del vettore di spazio operativo  $\mathbf{x}$ )
- Gradi di **libertà** del compito  $r = 2$  (gradi di libertà necessari per specificare il compito)
- Gradi di **mobilità ridondanti**  $n - r = 2$

A questo punto verrà considerata solo la parte posizionale dello Jacobiano ricavato al capitolo 3, essendo l'orientamento non di nostro interesse.

Dato lo Jacobiano  $\mathbf{J}_p$  estratto da  $\mathbf{J}(\mathbf{q})$  :

$$\mathbf{J}_P = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - d c_{123} & a_2 c_{12} + d s_{123} & d c_{123} & s_{123} \\ a_1 c_1 + a_2 c_{12} + d s_{123} & a_2 s_{12} - d c_{123} & d s_{123} & -c_{123} \end{bmatrix}$$

si può procedere alla caratterizzazione della ridondanza come segue:

1. Calcolo del rango di  $\mathbf{J}_p$ 
  - a.  $\text{rank}(\mathbf{J}_p) = 2 = r \Rightarrow$  Jacobiano di rango pieno
2. Calcolo della dimensione del Range di  $\mathbf{J}_p$ 
  - a.  $\dim(R(\mathbf{J}_p)) = r \Rightarrow 2$
3. Calcolo della dimensione del Nullo di  $\mathbf{J}_p$ 
  - a.  $\dim(N(\mathbf{J}_p)) = n - r \Rightarrow 2$

Dalle definizioni sopra riportate possiamo affermare che essendo  $N(J) \neq 0$  il manipolatore assegnato è **ridondante**.

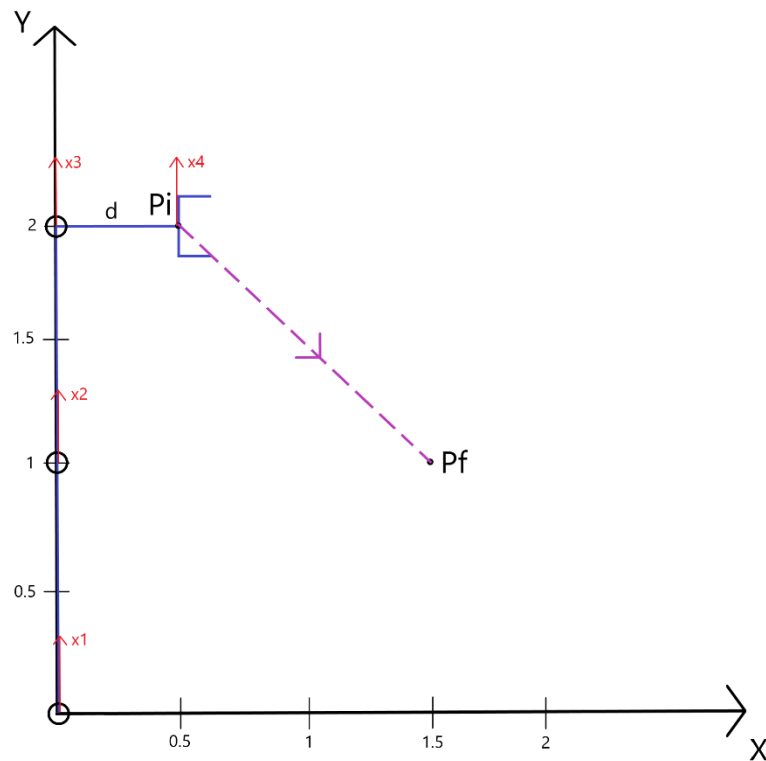
## 4.3. Configurazione iniziale

A questo punto assegniamo una configurazione iniziale al manipolatore e ripetiamo la caratterizzazione della ridondanza per la configurazione dello spazio dei giunti assegnata:



$$q(0) = \left[ \frac{\pi}{2}, 0, 0, 0.5 \right]$$

Il manipolatore si troverà nella seguente posizione:



Notiamo che la posizione dell'organo terminale coincide con la posizione del punto iniziale del segmento da percorrere.

In questa configurazione avremo:

1. Matrice di trasformazione omogenea, ricavata con Matlab tramite la funzione **[A10,A20,A30,A40]=cinematicaDiretta(a,q);**

A40 =

```
0.0000    0    1.0000    0.5000
1.0000    0   -0.0000    2.0000
0    1.0000    0    0
0    0    0    1.0000
```

a.

2. Jacobiano posizionale ricavata con Matlab tramite la funzione **[J]=jacobianoGeometrico(q,A10,A20,A30,A40);**

$J =$

$$\begin{bmatrix} -2.0000 & -1.0000 & 0 & 1.0000 \\ 0.5000 & 0.5000 & 0.5000 & -0.0000 \end{bmatrix}$$

a.

3. Calcolo del rango di  $J_p$  attraverso la funzione **rank(J)** di matlab

a.  $\text{rank}(J_p) = 2 = r \Rightarrow$  Jacobiano di rango pieno

4. Calcolo della dimensione del Range di  $J_p$

a.  $\dim(R(J_p)) = r \Rightarrow 2$

5. Calcolo della dimensione del Nullo di  $J_p$

a.  $\dim(N(J_p)) = n - r \Rightarrow 2$

Dai risultati ottenuti notiamo che il manipolatore nella configurazione assegnata non si trova in singolarità di rappresentazione.

## 5.SPECIFICAZIONE DELLA LEGGE DI MOTO

La pianificazione della traiettoria ha come obiettivo la generazione degli ingressi di riferimento per il sistema di controllo del moto. Si specifica un percorso, una primitiva geometrica, che il manipolatore deve descrivere e tramite una legge oraria si pianifica temporalmente come dovrà essere eseguito il compito assegnato.

### 5.1.Primitiva Geometrica

Per rappresentare la primitiva geometrica in maniera analitica, si utilizza la **rappresentazione parametrica di una curva nello spazio**.

Prendendo come riferimento la terna utensile una **primitiva di percorso**, caratterizzata dagli estremi  $p_i$  e  $p_f$ , è espressa tramite l'equazione:

$$p = f(s)$$

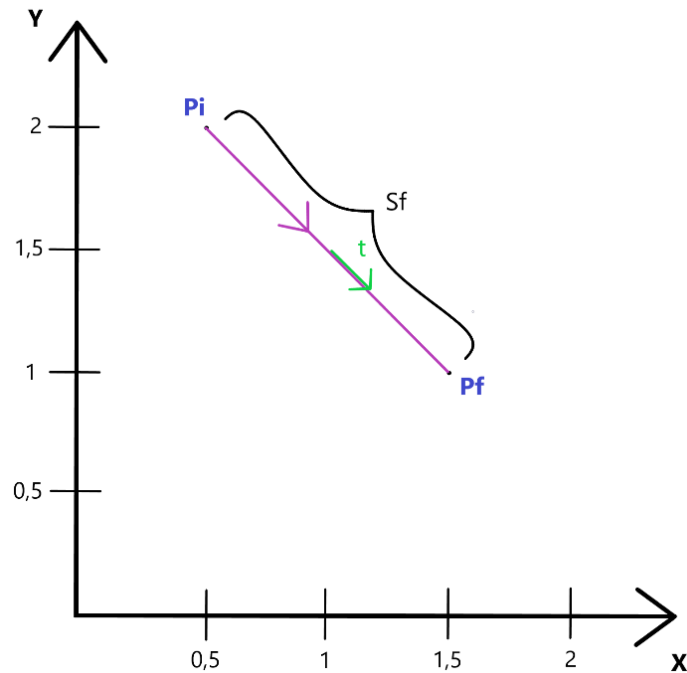
la parametrizzazione può essere effettuata rispetto **all'ascissa curvilinea s**, che rappresenta la lunghezza dell'arco di traiettoria.

In ciascun punto della curva si possono esprimere terne di versori: **tangente, normale e binormale**.

Assegnata una sequenza di configurazioni per l'organo terminale:

- **punti estremi:** posizione iniziale  $p_i = [0.5 \ 2.0]^T \text{m}$  e finale  $p_f = [1.5 \ 1.0]^T \text{m}$
- **tempo di percorrenza:**  $t = 5 \text{ s}$

Come primitiva di percorso si considera un segmento, che unisce i punti  $p_i$  e  $p_f$ . Fissato il verso di percorrenza che va da  $p_i \rightarrow p_f$  e sia la lunghezza di tale segmento pari a  $\|p_f - p_i\| = \sqrt{2}$



**l'equazione parametrica di un segmento nello spazio è:**

**Per la posizione:**

$$\mathbf{p}(s) = \mathbf{p}_i + \frac{s}{\|\mathbf{p}_f - \mathbf{p}_i\|} (\mathbf{p}_f - \mathbf{p}_i)$$

**Per la velocità:**

$$\dot{\mathbf{p}}_e = \frac{\dot{s}}{\|\mathbf{p}_f - \mathbf{p}_i\|} (\mathbf{p}_f - \mathbf{p}_i)$$

L'ascissa curvilinea deve variare tra 0 e  $s_f$  (con  $s_f = \|\mathbf{p}_f - \mathbf{p}_i\| = \sqrt{2}$ ) per cui la  $\mathbf{p}(s)$  valutata in tali valori di  $s$  sarà:

$$\mathbf{p}(0) = \mathbf{p}_i, \mathbf{p}(\|\mathbf{p}_f - \mathbf{p}_i\|) = \mathbf{p}_f$$

L'unico versore che si può definire in maniera univoca è il versore tangente, allineato al segmento e con verso positivo da  $\mathbf{p}_i \rightarrow \mathbf{p}_f$

## 5.2. Pianificazione della traiettoria

Per pianificare temporalmente la traiettoria dell'organo terminale lungo tale segmento, si è ricorsi alle **tecniche di interpolazione del moto punto-punto**.

Considerata l'ascissa curvilinea in funzione del tempo:  $\mathbf{s}(t)$ , il tutto si riduce ad **interpolare  $\mathbf{s}(t)$  lungo i valori assegnati**.

Per l'ascissa curvilinea si utilizza una legge oraria polinomiale di terzo grado

$$S(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$\dot{S}(t) = 3a_3 t^2 + 2a_2 t + a_1$$

Sono stati imposti i seguenti vincoli a  $t = 0$  e a  $t = t_f = 5$

- Posizione: valgono i vincoli descritti precedentemente per l'ascissa curvilinea

$$\begin{cases} S(t_i = 0) = 0 \\ S(t_f = 5) = \sqrt{2} \end{cases}$$

- Velocità: si è supposto l'organo terminale fermo, e quindi con velocità nulla, negli istanti di tempo iniziali e finali

$$\begin{cases} \dot{S}(t_i = 0) = 0 \\ \dot{S}(t_f = 5) = 0 \end{cases}$$

Si genera, quindi, un sistema lineare di quattro equazioni in quattro incognite, due di esse banali, con lo scopo di determinare i valori dei coefficienti del polinomio  $a_0, a_1, a_2, a_3$ .

$$\begin{cases} 0 = a_0 \\ \sqrt{2} = 125a_3 + 25a_2 + 5a_1 + a_0 \\ 0 = a_1 \\ 0 = 75a_3 + 10a_2 + a_1 \end{cases} \quad \begin{cases} a_0 = 0 \\ a_1 = 0 \\ a_2 = \frac{3\sqrt{2}}{25} \\ a_3 = -\frac{2\sqrt{2}}{125} \end{cases}$$

Da cui ne risulta il seguente polinomio:

$$S(t) = -\frac{2\sqrt{2}}{125}t^3 + \frac{3\sqrt{2}}{25}t^2$$

Effettuati i calcoli, e sostituendo i risultati ottenuti nell'equazione parametrica del segmento si ottiene la posizione e la velocità desiderata per l'organo terminale.

$$\begin{cases} p_{ex} = 0.5 + \left( \frac{-2t^3 + 15t^2}{125} \right) \\ p_{ey} = 2 - \left( \frac{-2t^3 + 15t^2}{125} \right) \\ \dot{p}_{ex} = -\frac{6}{125}t^2 + \frac{6}{25}t \\ \dot{p}_{ey} = \frac{6}{125}t^2 - \frac{6}{25}t \end{cases}$$

Le operazioni appena descritte sono state tutte effettuate nello spazio operativo. La posizione e la velocità definite per l'organo terminale saranno trasformate nelle rispettive posizioni e velocità nello spazio dei giunti tramite un'inversione cinematica, poiché il manipolatore è mosso dagli attuatori che muovono i giunti.

## 6.ALGORITMO DI INVERSIONE CINEMATICA

La cinematica differenziale inversa ci permette di ricavare la posizione dei giunti a partire dalle loro velocità. Può essere fatto in 3 principali modi.

1. Caso in cui  $n = r$ 
  - a. Manipolatore non ridondante
  - b.  $J$  quadrata e di rango pieno, manipolatore non in singolarità.
2. Caso in cui  $r < n$ 
  - a. Manipolatore ridondante
  - b.  $J(r \times n)$ , manipolatore non in singolarità.
3. Caso in cui  $\dim(R(J)) < r$ 
  - a.  $J$  non di rango pieno, manipolatore in singolarità.

Per quanto detto nel capitolo 3, il nostro problema ricade nel caso 2 quindi vedremo come risolvere la cinematica inversa per un manipolatore ridondante non in singolarità.

La relazione di cinematica differenziale del caso 2 può essere espressa come un sistema  $Ax = b$  con  $A$  rettangolare bassa, il sistema avrà più incognite che equazioni quindi ammette **infinite** soluzioni. Questo problema si può vedere come un problema di ottimizzazione vincolata in cui l'obiettivo è scegliere una tra le infinite soluzioni possibili. Significa trovare una soluzione che minimizza una funzione obiettivo soggetta al vincolo della cinematica differenziale diretta, ovvero

scelgo la soluzione a norma pesata minima che minimizza l'energia cinetica espressa come :  $\|\dot{q}\|^2$  Funzione che deve essere ottimizzata sempre rispettando il vincolo della cinematica differenziale diretta, il problema si può riassumere come:

$$\begin{cases} \min_{\dot{q}} \frac{1}{2} \dot{q}^T W \dot{q} \\ \text{s.t.} \\ v_n = J \dot{q} \end{cases}$$

Scegliendo  $W = I$ , la soluzione ottima ricavata risolvendo il problema è:

$$\dot{q} = J^T (J J^T)^{-1} v_n = J^\dagger v_n$$

## 6.1.Moti interni

Nel caso in cui si voglia esplicitamente gestire la ridondanza, quindi i gradi di libertà in più, si modifica la funzione obiettivo sostituendo al posto di  $\|\dot{q}\|^2$  la distanza tra  $\dot{q}$  e  $\dot{q}_0$  espressa come:  $\|\dot{q} - \dot{q}_0\|^2$ . Se ne ricava il seguente problema di ottimizzazione vincolata:

$$\begin{cases} \min_{\dot{q}} \frac{1}{2} (\dot{q} - \dot{q}_0)^T (\dot{q} - \dot{q}_0) \\ \text{s.t.} \\ v_n = J \dot{q} \end{cases}$$

Risolvendo questo problema attraverso i moltiplicatori di Lagrange si arriva ad una soluzione ottima:

$$\dot{q} = J^\dagger v_n + (I - J^\dagger J) \dot{q}_0$$

$I - J^\dagger J$  è un proiettore nel Nullo di  $J$  in cui possono essere “proiettati” i moti interni della struttura definiti nel vettore  $\dot{q}_0$ . In particolare, nel nostro caso, lo useremo per eseguire il task secondario del manipolatore ovvero la **misura di manipolabilità** definita come:

$$\sigma(q) = \frac{1}{2} (\sin(q(2))^2 + \sin(q(3))^2)$$

Questo task secondario, essendo proiettato nel nullo, non interferirà con il task primario e verrà quindi soddisfatto per quanto possibile dal manipolatore ma senza alcuna garanzia di successo. In particolare, questo task secondario ha una priorità più bassa di quello primario (posizione dell'organo terminale) e si cerca di soddisfarlo solo per quelle componenti che giacciono nel Nullo dello Jacobiano. Il manipolatore può soddisfarlo solo muovendosi nel Nullo, senza sfruttare tutti i moti della struttura, per non interferire con il task primario. Il compito secondario programmato è una buona approssimazione della misura di manipolabilità e valuta quanto il manipolatore si mantiene distante dalla singolarità, tanto più è massimizzata tanto più il manipolatore è lontano dalla singolarità. Il valore massimo si ha quando  $\theta_2$  e  $\theta_3$  sono uguali a  $\frac{\pi}{2}$  e quindi  $\sigma(q)$  è pari a 1.

## 6.2.Algoritmo di inversione cinematica

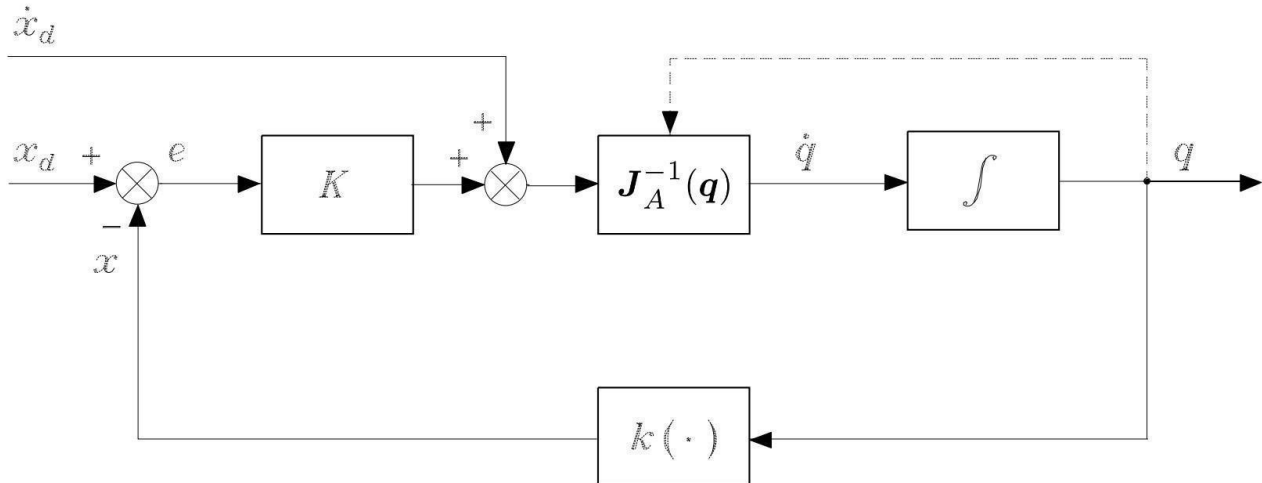
Assegnata una traiettoria desiderata per l'organo terminale, illustrata nel capitolo 5, si hanno tutti gli ingredienti per procedere con l'inversione cinematica. Per le considerazioni fatte nel capitolo 4 (manipolatore ridondante non in singolarità) si sceglie di utilizzare l'algoritmo con la pseudo inversa destra dello Jacobiano con la traiettoria desiderata specificata nello spazio operativo:

$$\dot{q} = J_A^\dagger(q) \dot{x}_d + (I - J_A^\dagger(q) J_A(q)) \dot{q}_0$$

**Principio di funzionamento dell'algoritmo:**

Una volta calcolato il vettore delle velocità ai giunti è possibile integrarlo con il metodo di Eulero utilizzando un passo di integrazione  $\Delta t$  pari a 2ms così da ricavare le corrispondenti posizioni dell'organo terminale. Per evitare i fenomeni di deriva numerica si è scelto di aggiungere un fattore correttivo  $e$  che vada a correggere l'approssimazione data dall'integrale numerico. Fattore correttivo viene moltiplicato per la matrice di guadagni  $K$  scelta diagonale pari a  $\text{diag} = \{500, 500\}$ , si pesano allo stesso modo gli errori di posizione sull'asse  $X$  e  $Y$ .

È possibile riassumere il processo di esecuzione attraverso il seguente schema:



Nel caso in esame bisogna notare che:

1.  $J_p = J_A$  che viene anche indicato come  $J$
2. Il vettore di spazio operativo  $x_d$  contiene solo le componenti di posizione lungo  $X$  ed  $Y$  essendo le sole due variabili di interesse per la specificazione del compito primario.

L'algoritmo proposto è in grado di risolvere sia il caso in cui si voglia specificare solo il compito primario sia il caso in cui si voglia gestire la ridondanza assegnando un compito secondario. Per farlo bisogna assegnare il valore corrispondente alla variabile **task** del codice Matlab. Il codice sorgente Matlab proposto per risolvere l'inversione cinematica è lasciato in calce al documento nella sezione appendice.

1. Per la risoluzione del solo compito primario viene utilizzata la seguente espressione per calcolare il vettore delle velocità ai giunti:

$$\dot{q} = J_A^{-1}(q) (\dot{x}_d + K e)$$

a. **task = 1;**

2. Per la gestione della ridondanza viene utilizzata la seguente espressione per calcolare il vettore delle velocità ai giunti:

$$\dot{q} = J_A^{\dagger}(q) (\dot{x}_d + K e) + (I - J_A^{\dagger}(q) J_A(q)) \dot{q}_0$$

a. **task = 2;**

## 7. RISULTATI OTTENUTI

A questo punto si hanno tutti gli elementi necessari per procedere alla simulazione del compito attraverso uno script Matlab che metta in pratica quanto detto nel capitolo 6.

Salvando i valori delle varie posizioni, velocità dei giunti, errore di posizione e misura di manipolabilità espressi nel tempo è possibile graficare i risultati della simulazione effettuata.

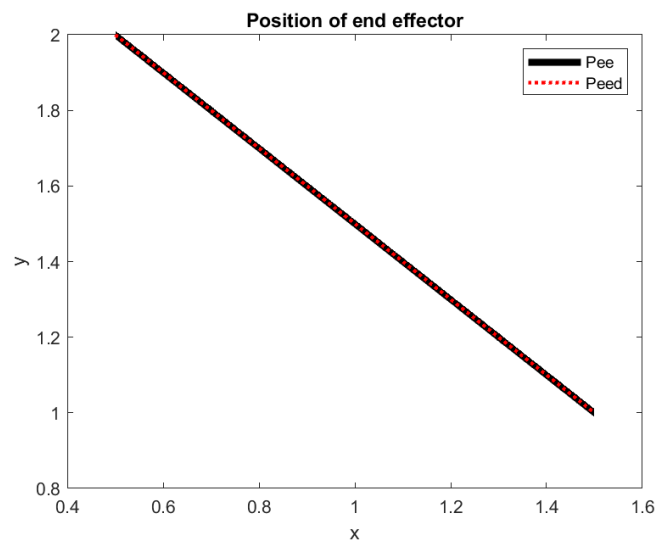
## 7.1. Simulazione del compito primario

In questo paragrafo verranno commentati i risultati relativi alla simulazione del solo task primario:

l'organo terminale deve percorrere un segmento di estremi  $P_i = [0.5, 2]$  e  $P_f = [1.5, 1]$  in un tempo pari a 5 secondi.

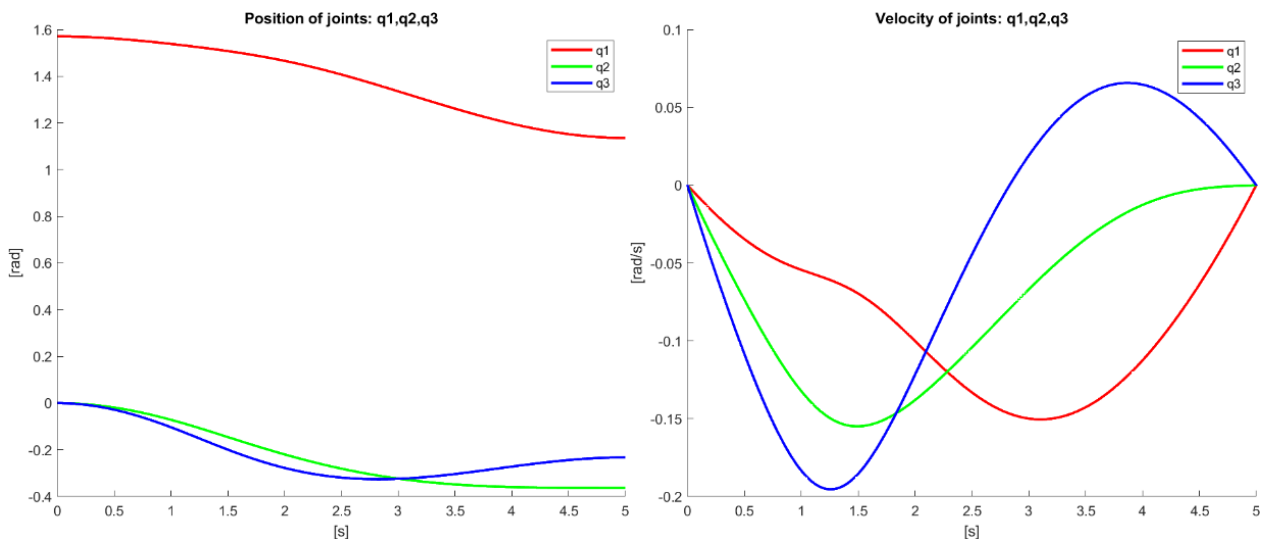
Lo script Matlab che esegue l'inversione cinematica disponibile nella sezione Appendice rispetta il compito primario previsto per il manipolatore.

In particolare settando il valore di `task = 1` verrà eseguita la simulazione corrispondente. Salvando i valori della posizione dell'organo terminale corrispondenti alle componenti  $p_x$  e  $p_y$  unitamente ai valori di posizione desiderata  $pd_x$  e  $pd_y$  dal grafico che segue possiamo notare che:

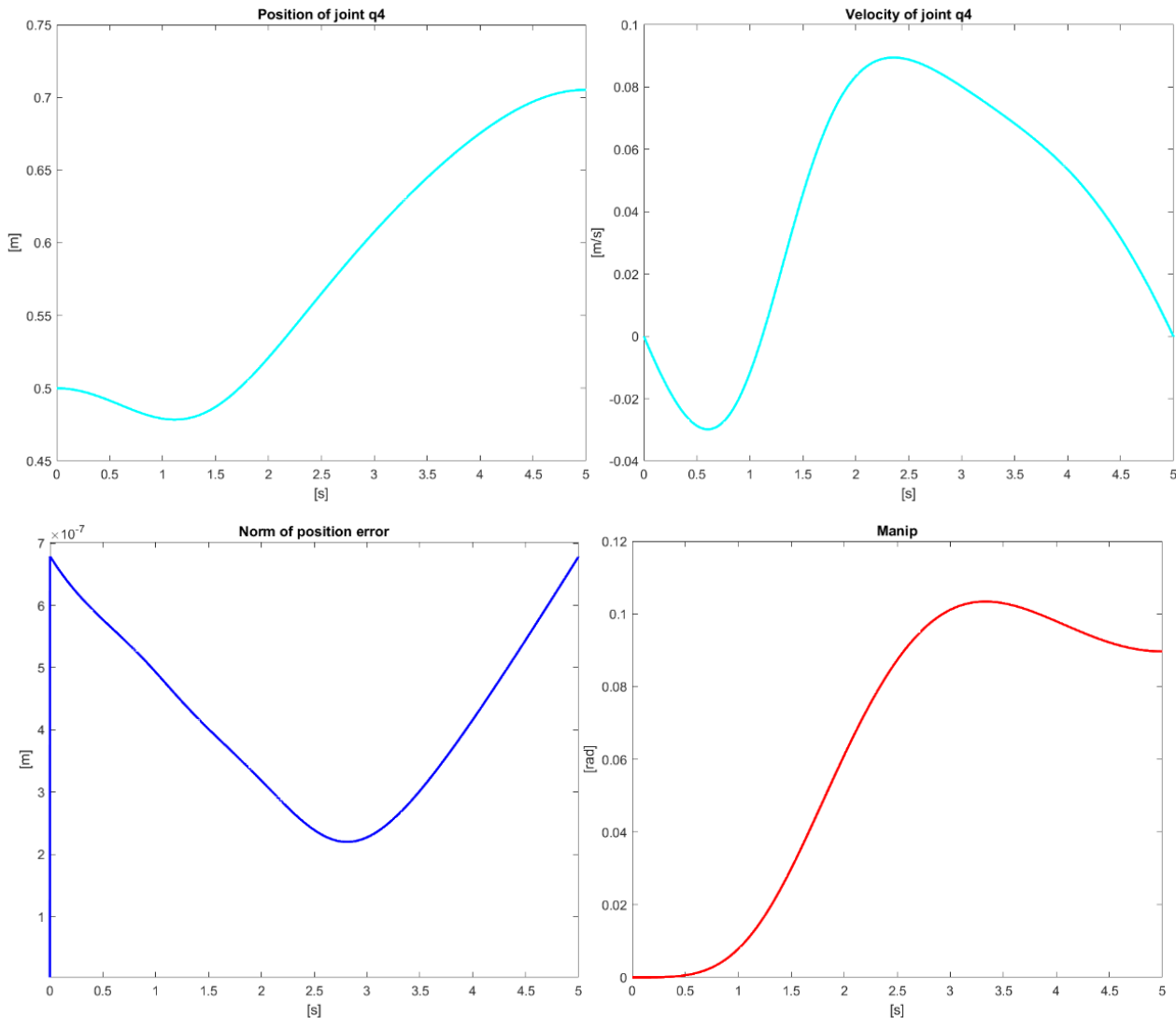


L'andamento della posizione dell'ee effettivamente eseguita dal manipolatore percorre quella pianificata, infatti i grafici di **Pee** (posizione dell'organo terminale effettiva) e di **Peed** (posizione dell'organo terminale pianificata) combaciano perfettamente.

Appurato che l'algoritmo soddisfa correttamente il compito assegnato diamo uno sguardo ai restanti grafici ricavati dall'esecuzione dell'inversione cinematica.







Come si vede dal grafico, l'errore si mantiene molto basso, nell'ordine di  $10^{-7}$  quindi l'integrazione di Eulero sta funzionando correttamente. Invece per quanto riguarda la misura di manipolabilità rimane sempre nell'intorno di 0.1 per cui si vede chiaramente che il compito secondario non essendo stato aggiunto non viene soddisfatto.

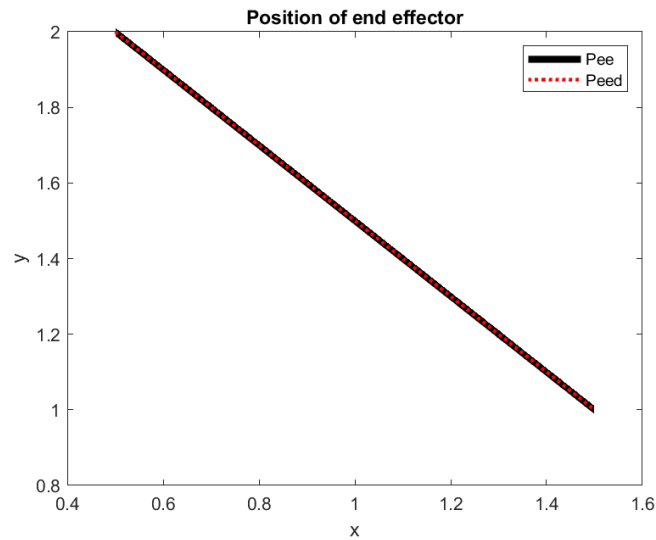
## 7.1. Simulazione del compito primario e secondario

In questo paragrafo verranno commentati i risultati relativi alla simulazione del task primario unito al task secondario a priorità più bassa ovvero la misura di manipolabilità.

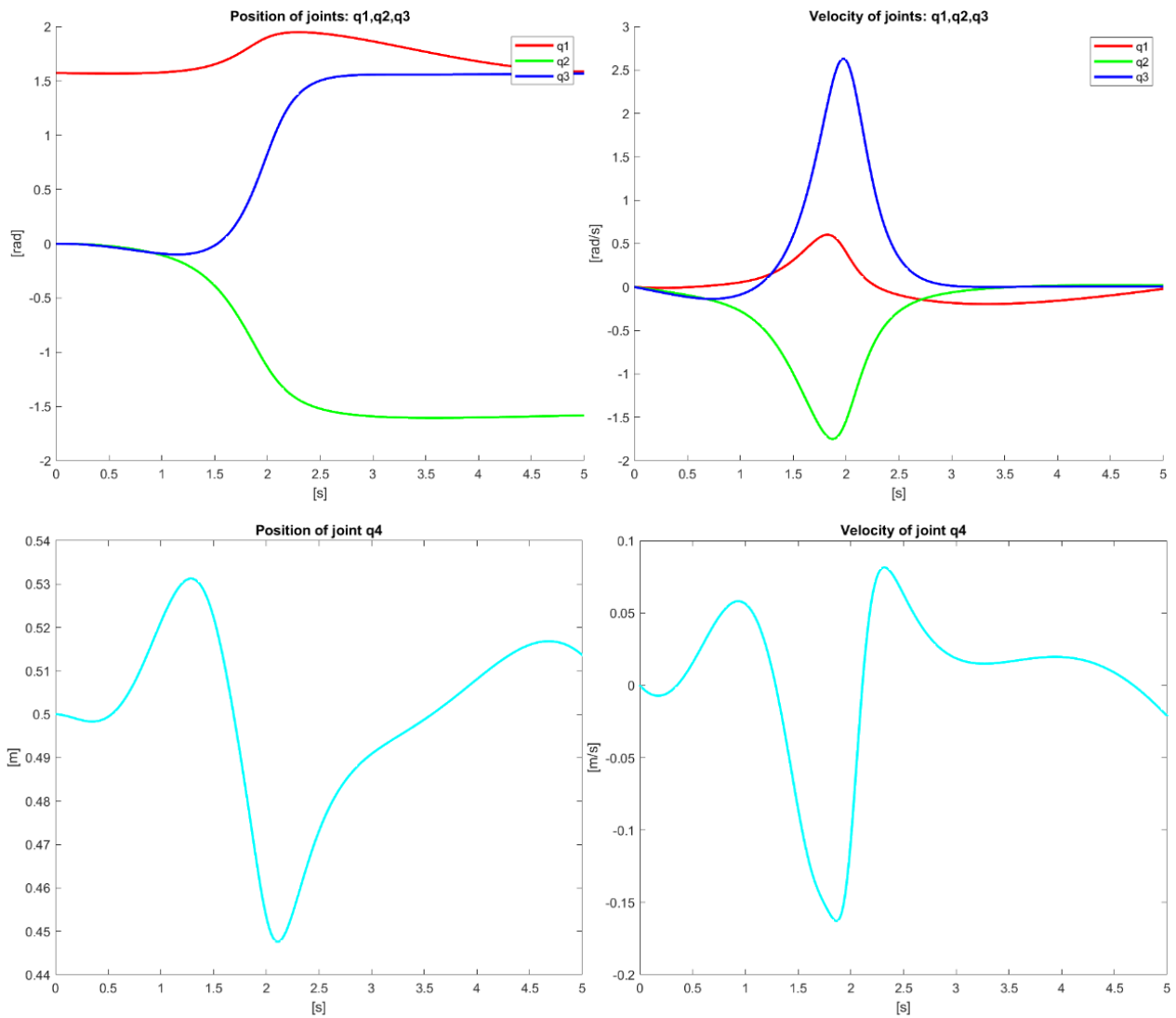
Quindi settando il valore di `task = 2` verrà eseguito il codice di inversione con proiettore nel nullo moltiplicato per  $\dot{q}_0$  (moti interni). In questo caso  $\dot{q}_0$  viene calcolato con la seguente formula:

$$\dot{q}_0 = k_0 \left( \frac{\partial \sigma_0(q)}{\partial q} \right)^T$$
, in cui  $k_0$  viene scelto pari a 5 e moltiplica il gradiente della funzione obiettivo secondaria sigma (misura di manipolabilità) discussa nel capitolo precedente. Possiamo innanzitutto notare che:

Il manipolatore soddisfa ancora pienamente il compito primario, l'aggiunta del compito secondario non interferisce minimamente con la posizione dell'organo terminale.

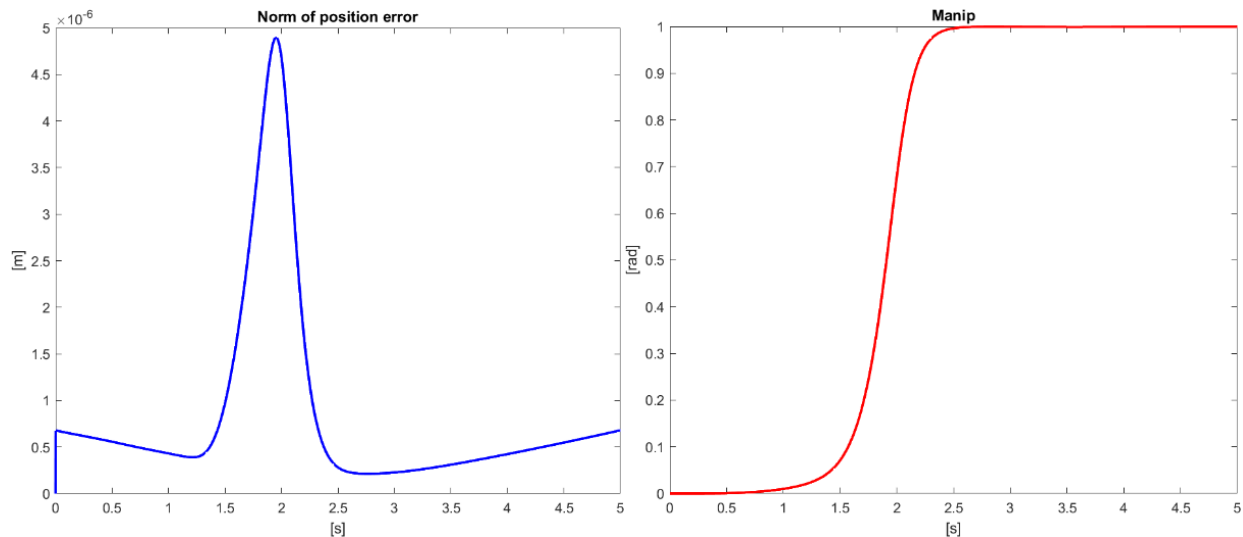


Invece gli altri grafici cambiano:



Dai grafici sopra riportati notiamo che:

- Le posizioni dei giunti q1 q2 e q3 variano rispetto alla specifica del solo task primario.
- Le velocità dei giunti q1 q2 e q3 subiscono dei picchi al tempo  $t = 2s$  circa.



Inoltre possiamo notare che l'errore di posizione rimane comunque molto contenuto e tende a 0 con l'avanzare del tempo. Infine la misura di manipolabilità raggiunge 1 nell'intorno dei due secondi e la mantiene fino alla fine dell'esecuzione. In questo caso quindi, grazie ai gradi di mobilità ridondanti, siamo in grado di specificare correttamente i due compiti prefissati.

### 7.3. Confronto dei risultati delle due simulazioni

Confrontando i risultati si nota che:

- Specificando il compito secondario le posizioni e le velocità dei giunti cambiano leggermente. Inoltre i giunti sono soggetti ad un picco di velocità nell'intorno del tempo  $t = 2s$  con accelerazioni che crescono in modo brusco. Questo è causato dal task secondario.
- La posizione dell'organo terminale non varia specificando anche il task secondario.
- Le variazioni di posizione e velocità dei giunti del task secondario rispetto a quello primario sono dovute all'approssimazione a tempo discreto che genera una piccola interferenza tra i due casi. Questa interferenza la si può notare anche se si confrontano i grafici relativi alla norma di errore, poiché nell'intorno di  $t = 2s$  l'andamento risulta diverso. Negli istanti immediatamente successivi i due grafici sono molto simili e questa similarità la si mantiene anche aumentando il tempo totale per l'esecuzione del compito.
- Il picco di velocità e l'aumento delle accelerazioni può essere contenuto andando a ridurre il termine di guadagno  $k_0$
- La misura di manipolabilità migliora di molto con la gestione della ridondanza rispetto a quando non viene specificato.

### 7.4. Conclusioni

Dai risultati delle simulazioni si può concludere che:

- Il compito primario viene sempre soddisfatto in entrambi i casi proposti.
- Il compito secondario viene soddisfatto soltanto nel caso 2. Ma comunque essendo proiettato nel nullo di  $J$  (compito a priorità più bassa) viene soddisfatto solo a partire dal tempo  $t = 2.5$  secondi circa in avanti. Non viene garantita la sua assoluta soddisfazione.
- Il picco di velocità nell'intorno di  $t = 2s$  tende ad incrementare all'aumentare del fattore di guadagno  $k_0$

## 8.APPENDICE

### 8.1.Algoritmo di inversione cinematica

```
1 %% Configurazione iniziale manipolatore
2 theta1_deg = 90; theta2_deg = 0; theta3_deg = 0;
3 theta1_rad = deg2rad(theta1_deg);
4 theta2_rad = deg2rad(theta2_deg);
5 theta3_rad = deg2rad(theta3_deg);
6 d=0.5; a=1;
7 %%-----Parametri dell'algo di inversione-----%%
8 ti = 0; tf = 5; deltaT = 0.002;
9 K = [500,0;0,500]; k0 = 5;
10 I = eye(4,4);
11 q=[theta1_rad,theta2_rad,theta3_rad,d]
12 task = 1;
13 %% Cinematica diretta
14 [A10,A20,A30,A40] = cinematicaDiretta(a,q);
15 x_pos = [A40(1,4),A40(2,4)]
16 %% JACOBIANO geometrico = analitico
17 J = jacobianoGeometrico(q,A10,A20,A30,A40);
18 J = J([1:2],:); k = rank(J); Jinv = pinv(J);
19 %% Algoritmo di inversione
20 for t = ti : deltaT : tf
21 %%-----Specificazione della traiettoria-----%%
22 PEx = 0.5 + (( -2 * (t^3) + 15 * (t^2)) / 125);
23 PEy = 2 - (( -2 * (t^3) + 15 * (t^2)) / 125);
24 PEx_der = -(6/125) * (t^2) + (6/25) * t;
25 PEy_der = (6/125) * (t^2) - (6/25) * t;
26 Pd = [PEx,PEy];
27 Pd_der = [PEx_der,PEy_der];
28 %%-----Calcolo errore-----%%
29 e = Pd-x_pos
30 %%-----Definizione del compito secondario sigma-----%%
31 sigma = (1/2)*(((sin(q(2)))^2)+((sin(q(3)))^2))
32 grad_sigma = [0,sin(q(2))*cos(q(2)),sin(q(3))*cos(q(3)),0]
33 q_der_0 = k0*(grad_sigma)'
34 %% Calcolo delle velocita' e posizioni di q
35 if(task == 1)
36     q_der = Jinv*((Pd_der')+(K*e'))
37 end
38 %%-----Con compito secondario-----%%
39 if(task == 2)
40     q_der = (Jinv*((Pd_der')+(K*e')))+((I-Jinv*J)*q_der_0)
41 end
42 q = q + (q_der * deltaT)'
43 [A10,A20,A30,A40] = cinematicaDiretta(a,q);
44 x_pos = [A40(1,4),A40(2,4)]
45 J = jacobianoGeometrico(q,A10,A20,A30,A40);
46 J = J([1:2],:)
47 Jinv = pinv(J)
48 end
```

## 8.1.Funzione per il calcolo della cinematica diretta

```
1 function [A10,A20,A30,A40] = cinematicaDiretta(a,q)
2 A10= [cos(q(1)),-sin(q(1)),0,a*cos(q(1));sin(q(1)),cos(q(1))
      ,0,a*sin(q(1));0,0,1,0;0,0,0,1];
3 A21= [cos(q(2)),-sin(q(2)),0,a*cos(q(2));sin(q(2)),cos(q(2))
      ,0,a*sin(q(2));0,0,1,0;0,0,0,1];
4 A20 = A10*A21;
5 A32= [cos(q(3)),0,sin(q(3)),0;sin(q(3)),0,-cos(q(3))
      ,0;0,1,0,0;0,0,0,1];
6 A30 = A20*A32;
7 A43= [1,0,0,0;0,1,0,0;0,0,1,q(4);0,0,0,1];
8 A40 = A30*A43
9 end
```

## 8.2.Funzione per il calcolo dello Jacobiano geometrico

```
1 function [J] = jacobianoGeometrico(q,A10,A20,A30,A40)
2 p0 = [0;0;0];
3 z0 = [0;0;1];
4 p1 =A10(1:3,end);
5 z1 =A10(1:3,3);
6 p2 =A20(1:3,end);
7 z2 =A20(1:3,3);
8 z3 =A30(1:3,3);
9 p4 =A40(1:3,end);
10 J1 = [cross(z0,(p4-p0)); z0];
11 J2 = [cross(z1,(p4-p1)); z1];
12 J3 = [cross(z2,(p4-p2)); z2];
13 J4 = [z3; p0];
14 J = [J1, J2, J3, J4];
15 end
```