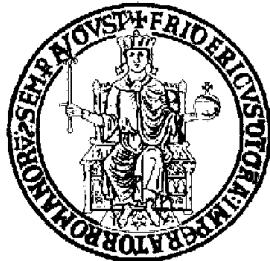


UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II



SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
Dipartimento di Ingegneria Elettrica e delle Tecnologie
dell'Informazione

Tesina di Fondamenti di Robotica

**Analisi e controllo
di un robot SCARA**

Prof. Bruno Siciliano

Carlo Cianflone
P38000165

Anno Accademico 2023/2024

Indice

1. Cinematica	3
2. Analisi di manipolabilità	5
3. Generazione delle traiettorie	9
4. Inversione cinematica	11
4.1. Algoritmo CLIK con inversa dello Jacobiano	11
4.2. Algoritmo CLIK con trasposta dello Jacobiano	14
4.3. Algoritmo CLIK con pseudo-inversa dello Jacobiano	16
4.4. Algoritmo CLIK del secondo ordine	18
5. Controllo del moto	22
5.1. Modello dinamico	22
5.2. Controllo robusto	24
5.3. Controllo adattativo	28
5.4. Controllo a dinamica inversa nello spazio operativo	32

1. Cinematica

La denominazione SCARA (Selective Compliance Assembly Robot Arm) sta ad indicare una categoria di robot a catena cinematica aperta dotati di due giunti rotoidali, uno prismatico ed un ultimo giunto rotoidale agente da polso. Gli assi dei giunti risultano essere tutti paralleli tra loro, e la struttura nel complesso offre elevata rigidezza verticale e buona cedevolezza nel piano orizzontale.

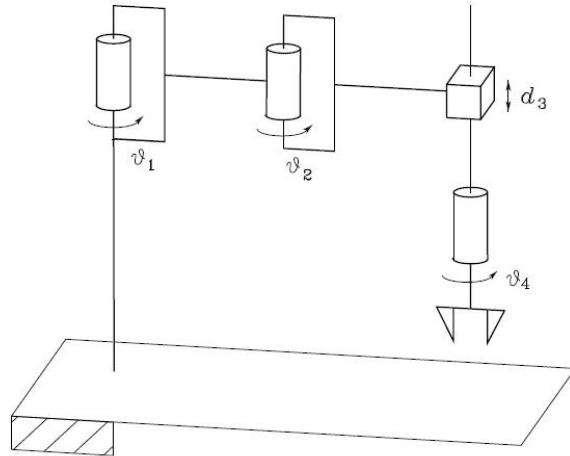


Fig. 1.1: Schema del robot SCARA in esame con variabili di giunto.

Sia $q \in \mathbb{R}^4$ il vettore delle variabili di giunto del robot, definito come:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ d_3 \\ \vartheta_4 \end{bmatrix} \quad (1.1)$$

Si procede col disporre le terne di riferimento secondo la convenzione di Denavit-Hartenberg. Gli assi y delle terne sono sottintesi per semplificare l'interpretazione della figura 1.2, le terne sono levogire.

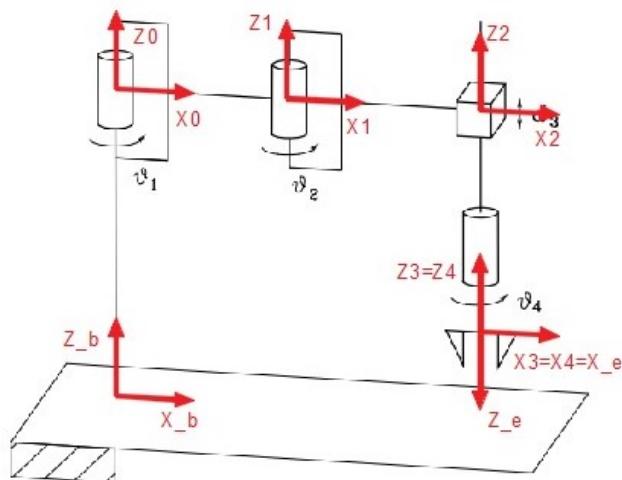


Fig. 1.2: Robot SCARA con terne di riferimento.

A valle di ciò la tabella dei parametri di Denavit-Hartenberg risulta costituita come segue:

LINK	a_i	α_i	d_i	θ_i
1	$a_1 = 0.5 \text{ m}$	0	0	ϑ_1
2	$a_2 = 0.5 \text{ m}$	0	0	ϑ_2
3	0	0	d_3	0
4	0	0	0	ϑ_4

Tabella 1.1: Parametri di Denavit-Hartenberg.

nota la quale è possibile ricavare la cinematica diretta del robot.

$$T_e^b = T_0^b T_4^0 T_e^4 \quad (1.2)$$

Osservo che:

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 \quad (1.3)$$

$$T_0^b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

$$T_e^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Le matrici T_0^b e T_e^4 sono matrici di trasformazione omogenea costanti a differenza della matrice omogenea riportata nella (1.3) e stanno a rappresentare rispettivamente la traslazione tra terna base e terna 0 e la rotazione tra terna 4 e terna solidale con l'end effector (EE).

Nonostante sia possibile procedere a partire dalla matrice cinematica diretta per giungere alla funzione cinematica diretta del robot in esame, la sua struttura è sufficientemente semplice da permettere di procedere in maniera puramente geometrica.

$$x_e = \begin{bmatrix} a_1 \cos(\vartheta_1) + a_2 \cos(\vartheta_1 + \vartheta_2) \\ a_1 \sin(\vartheta_1) + a_2 \sin(\vartheta_1 + \vartheta_2) \\ d_0 + d_3 \\ \vartheta_1 + \vartheta_2 + \vartheta_4 \end{bmatrix} \quad (1.6)$$

Al fine di rendere possibili le operazioni nei capitoli successivi risulta necessario calcolare lo Jacobiano del robot. Un primo approccio al problema ha portato al calcolo dello Jacobiano geometrico a partire dalle terne disposte in figura 1.2, valendo:

$$\begin{bmatrix} Jp_i \\ Jo_i \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} \quad i - \text{esimo giunto prismatico} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} \quad i - \text{esimo giunto rotoidale} \end{cases} \quad (1.7)$$

Lo Jacobiano ottenuto presenta sei righe di cui due nulle. Ciò è riconducibile al fatto che la struttura in esame ha un unico grado di libertà legato all'orientamento: l'EE non può compiere rotazioni attorno agli assi x e y. Eliminando le righe nulle (quarta e quinta) si ottiene il seguente Jacobiano:

$$J = \begin{bmatrix} -a_1 \sin(\vartheta_1) - a_2 \sin(\vartheta_1 + \vartheta_2) & -a_2 \sin(\vartheta_1 + \vartheta_2) & 0 & 0 \\ a_1 \cos(\vartheta_1) + a_2 \cos(\vartheta_1 + \vartheta_2) & a_2 \cos(\vartheta_1 + \vartheta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (1.8)$$

Si osserva che la matrice ottenuta è quadrata, con numero di giunti uguale a quello delle variabili necessarie a descrivere il moto nello spazio operativo. Ne consegue che il manipolatore non è intrinsecamente ridondante, ma può divenire funzionalmente tale se si ignora una delle componenti di posizione o orientamento nello spazio operativo. Inoltre, calcolando il determinante della matrice si può notare come il rango sia pieno a patto che sia verificata la condizione $\sin(\vartheta_2) \neq 0$: lo SCARA può essere soggetto a singolarità di gomito, condizione che va opportunamente evitata.

Per quanto riguarda lo Jacobiano analitico, poiché l'orientamento del manipolatore in questo caso specifico è descritto da una sola variabile, si può verificare che la matrice di trasformazione tra i due Jacobiani è sostanzialmente una matrice identità: i due Jacobiani sono identici. Sarebbe stato dunque possibile calcolare lo Jacobiano direttamente in maniera analitica secondo:

$$J = J_A = \frac{dx_e}{dq} \quad (1.9)$$

ottenendo lo stesso risultato.

2. Analisi di manipolabilità

Al fine di condurre un'analisi relativamente alla manipolabilità del robot in esame ricorriamo allo studio degli ellisoidi in velocità e forza.

Definita una sfera di raggio unitario nello spazio delle velocità ai giunti secondo:

$$\dot{q}^T \dot{q} = 1 \quad (2.1)$$

essendo per la cinematica differenziale verificata la relazione:

$$v_e = J(q)\dot{q} \quad (2.2)$$

sostituendo la (2.2) nella (2.1) si perviene alla formula del seguente ellissoide:

$$v_e^T (J(q)J^T(q))^{-1} v_e = 1 \quad (2.3)$$

a partire dal quale è possibile ricavare le velocità generabili nello spazio operativo per un dato vettore di velocità ai giunti, con il manipolatore in una postura nota.

Sfruttando la dualità cinetostatica è possibile pervenire a un'espressione analoga partendo da una sfera di raggio unitario nello spazio delle coppie generalizzate ai giunti:

$$\tau^T \tau = \gamma_e^T (J(q)J^T(q)) \gamma_e = 1 \quad (2.4)$$

Tale ellissoide descrive le forze generabili all'EE a partire da un dato vettore di coppie generalizzate ai giunti.

Gli ellisoidi in questo caso specifico sono ridotti a delle ellissi bidimensionali, data la dipendenza dello Jacobiano, e conseguentemente della manipolabilità, dalle uniche variabili ϑ_1 e ϑ_2 . Gli autovettori di $J(q)J^T(q)$ definiscono le direzioni degli assi principali degli ellisoidi, mentre i valori singolari dello Jacobiano forniscono le dimensioni degli stessi. Osserviamo che il nucleo di una forma quadratica è l'inverso dell'altra: i due ellisoidi avranno gli assi lungo le stesse direzioni, ma quelli dell'ellissoide di forza avranno dimensioni pari agli inversi dei valori singolari dello Jacobiano.

È possibile definire una misura di manipolabilità a partire dalla misura del volume dell'ellissoide stesso, secondo:

$$w(q) = \sqrt{\det(J(q)J^T(q))} \quad (2.5)$$

Facendo riferimento al caso particolare di manipolatore planare a due bracci:

$$w(q) = |\det(J(q))| = a_1 a_2 |\sin(\vartheta_2)| \quad (2.6)$$

Sulla base di quanto detto, è stata analizzata la manipolabilità della struttura portante del robot in velocità e forza in quattro distinte configurazioni:

$$q_A = \begin{bmatrix} \frac{\pi}{6} \\ -\frac{\pi}{2} \\ 0 \\ 0 \end{bmatrix}; q_B = \begin{bmatrix} \frac{\pi}{2} - \frac{\pi}{16} \\ -\pi + \frac{\pi}{16} \\ 0 \\ 0 \end{bmatrix}; q_C = \begin{bmatrix} -\frac{\pi}{8} \\ \frac{\pi}{4} \\ 0 \\ 0 \end{bmatrix}; q_D = \begin{bmatrix} \frac{\pi}{16} \\ -\frac{\pi}{16} \\ 0 \\ 0 \end{bmatrix} \quad (2.7)$$

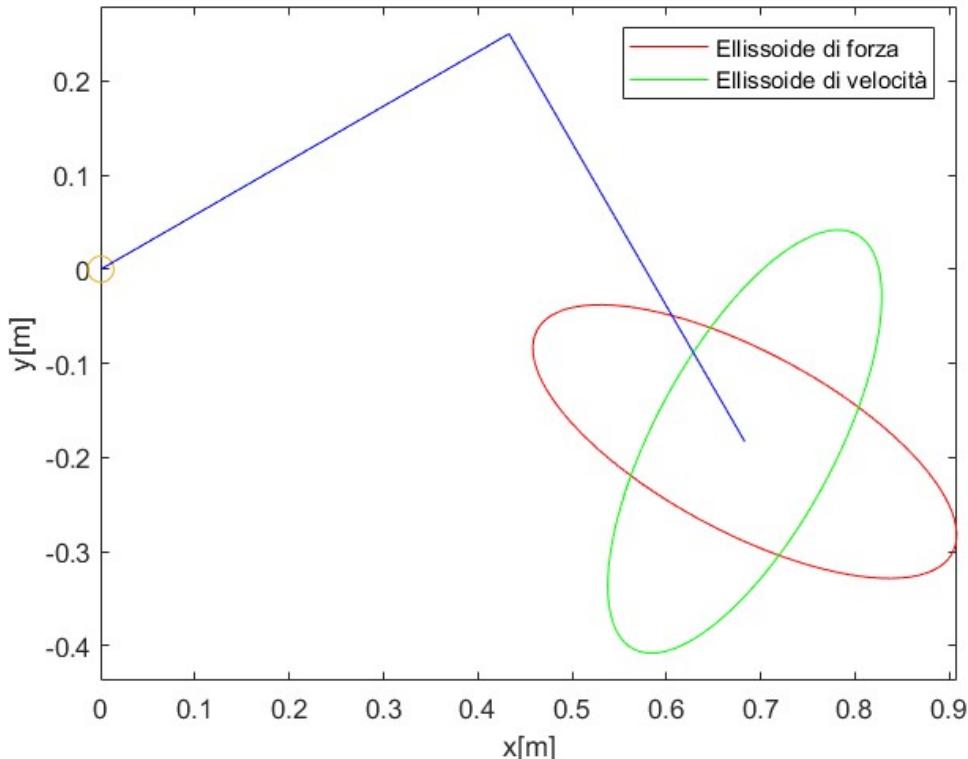


Fig. 2.1: Ellisoidi di velocità e forza per q_A ($w(q_A) = 0.25$).

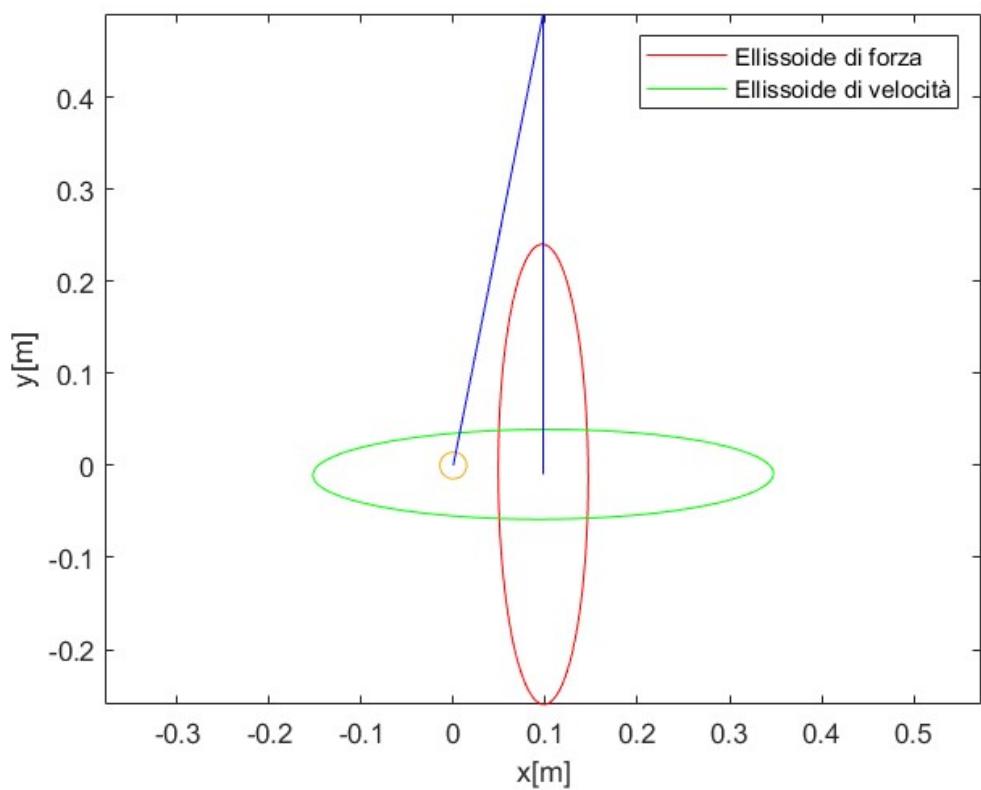


Fig. 2.2: Ellissoidi di velocità e forza per q_B ($w(q_B) = 0.0488$).

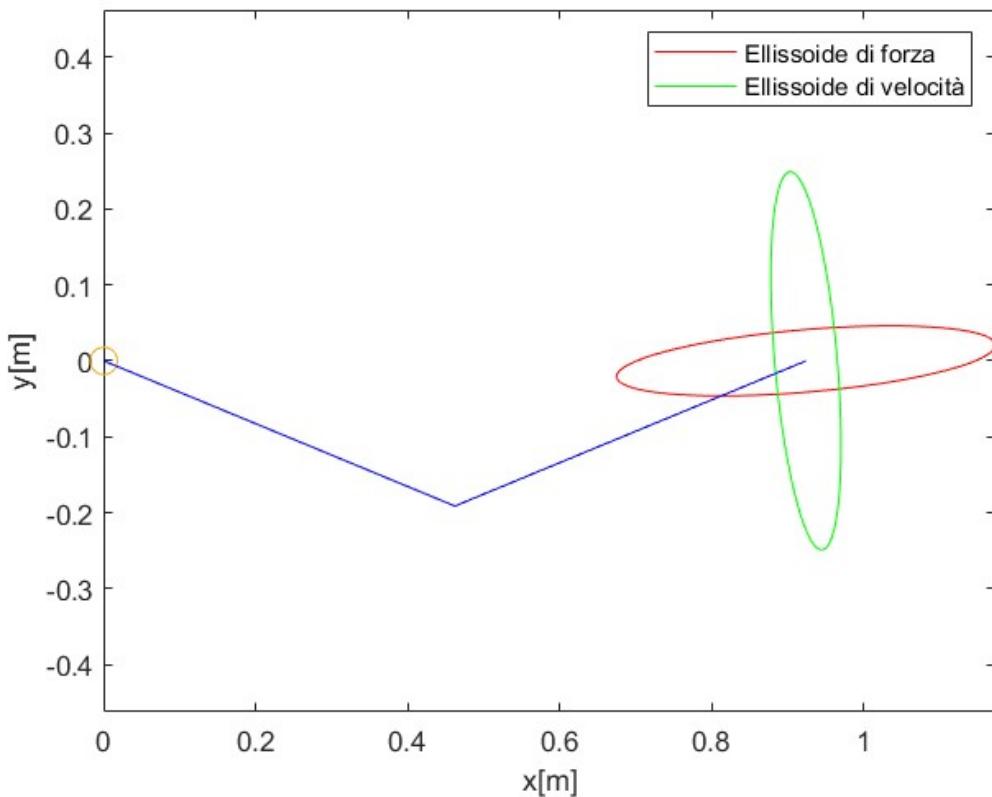


Fig. 2.3: Ellissoidi di velocità e forza per q_C ($w(q_C) = 0.1768$).

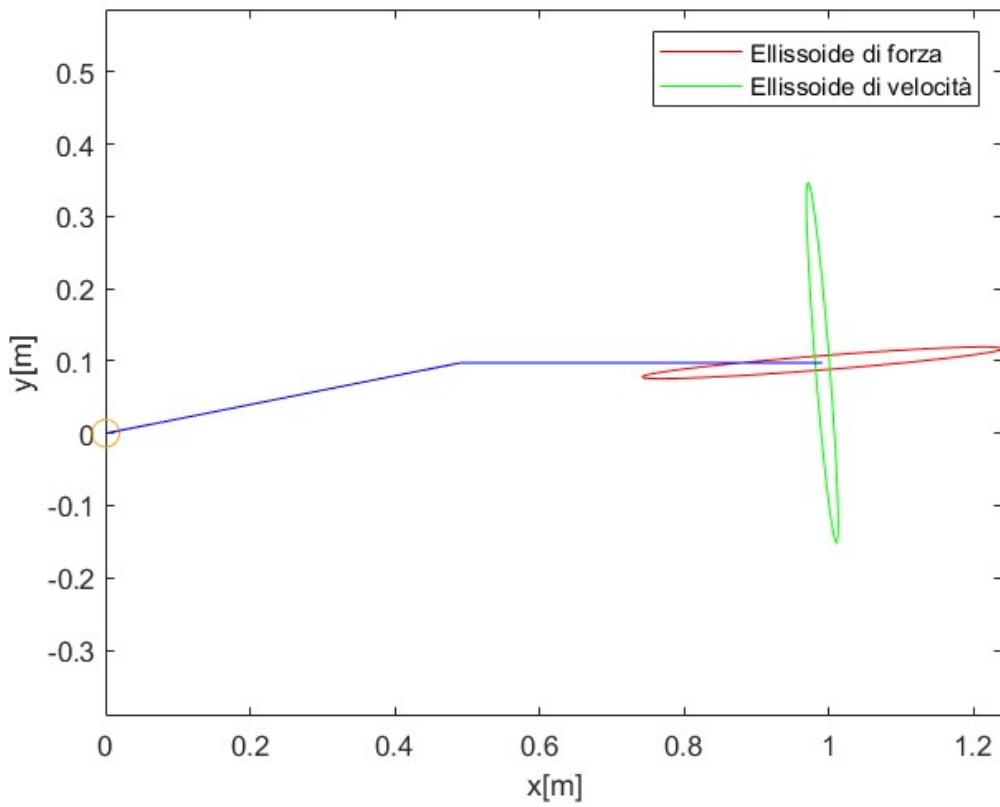


Fig. 2.4: Ellissoidi di velocità e forza per q_D ($w(q_D) = 0.0488$).

La visualizzazione degli ellissoidi è semplificata dalla normalizzazione delle scale di rappresentazione. Diversamente, gli assi maggiori dei due ellissoidi sarebbero differenti.

È possibile verificare come la misura di manipolabilità risulti massima nella figura 2.1, e in generale nel caso in cui $\vartheta_2 = \pm \frac{\pi}{2}$, risultato deducibile facilmente dalla (2.6). Diversamente, per ϑ_2 prossimo a $0 \pm k\pi$, ossia nell'intorno delle configurazioni singolari (figure 2.2 e 2.4), ho velocità lungo la direzione radiale praticamente nulla e valori di forza molto elevati.

3. Generazione delle traiettorie

Si desidera pianificare la traiettoria dell'EE lungo un percorso geometrico caratterizzato tramite una sequenza di 11 pose. Il percorso geometrico scelto è chiuso, con posa iniziale identica a quella finale. Sono presenti tre punti di via, rispettivamente P2, P4 e P10, mentre i punti P6 e P7, così come i punti P7 e P8, sono uniti tramite due semicirconference. Tutti gli altri tratti del percorso sono rettilinei.

Indice	P_X	P_Y	P_Z	Ψ
1	0.5	0.5	0.75	$-\frac{\pi}{2}$
2	0.5	0	0.5	0
3	0.5	-0.5	0.75	0
4	0	-0.5	0.5	$\frac{\pi}{2}$
5	-0.5	-0.5	0.75	$\frac{\pi}{2}$
6	-0.6	-0.5	0.25	π
7	-0.6	0	0.25	π
8	-0.6	0.5	0.25	π
9	-0.5	0.5	0.75	$-\frac{\pi}{2}$
10	0	0.5	0.5	$-\frac{\pi}{2}$
11	0.5	0.5	0.75	$-\frac{\pi}{2}$

Tabella 3.1: Pose di riferimento per la pianificazione di traiettoria.

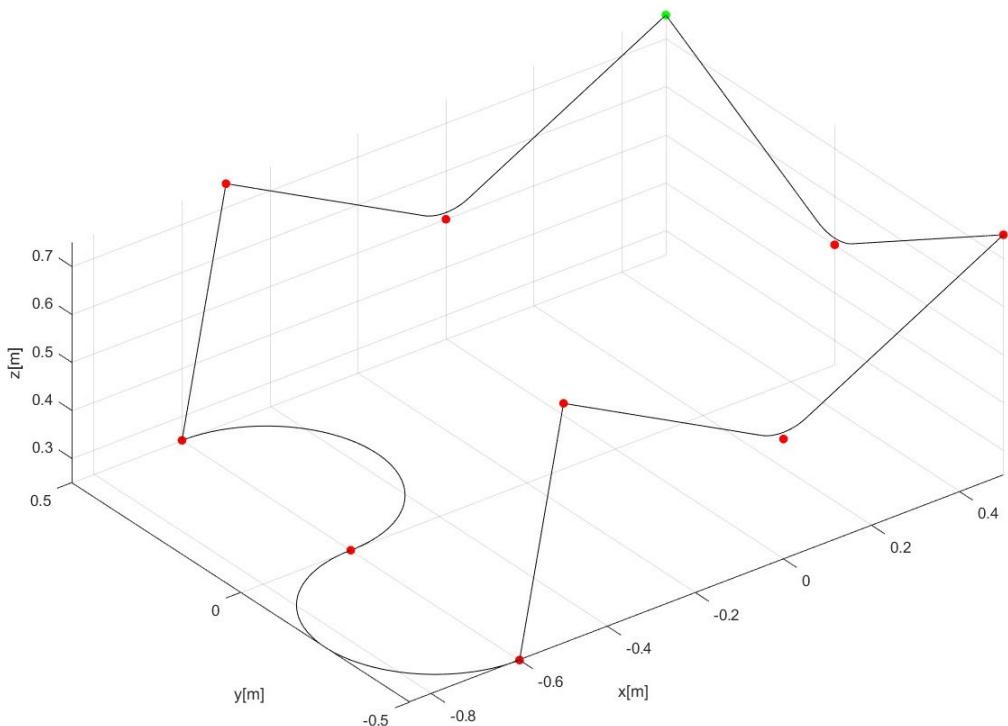


Fig. 3.1: Percorso geometrico desiderato per l'EE. Il punto verde indica il punto di partenza e arrivo, mentre i punti rossi sono punti di cammino.

Noto il percorso geometrico, si è scelto di costruire la traiettoria mediante polinomi del terzo ordine: questo ha permesso di assegnare i valori desiderati per posizione e velocità all'inizio e alla fine di ogni singolo tratto. La conseguenza principalmente osservabile di tale scelta è la discontinuità dell'accelerazione in corrispondenza dei punti di cammino (figura 3.4).

Il tempo richiesto per passare da P_i a P_{i+1} è stato posto pari a 2 secondi, mentre l'entità dell'anticipo temporale associato ai punti di via è stato scelto come pari a 0.4 secondi. Di seguito sono riportati gli andamenti temporali delle componenti di posa, velocità e accelerazione ottenuti per l'EE.

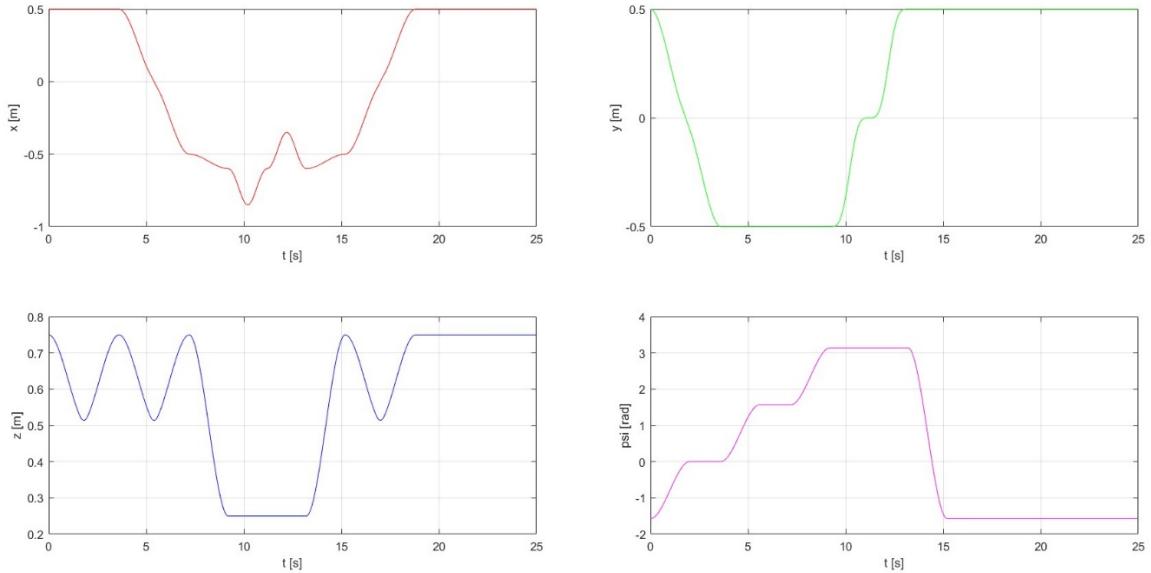


Fig. 3.2: Andamento delle componenti della posa dell'EE.

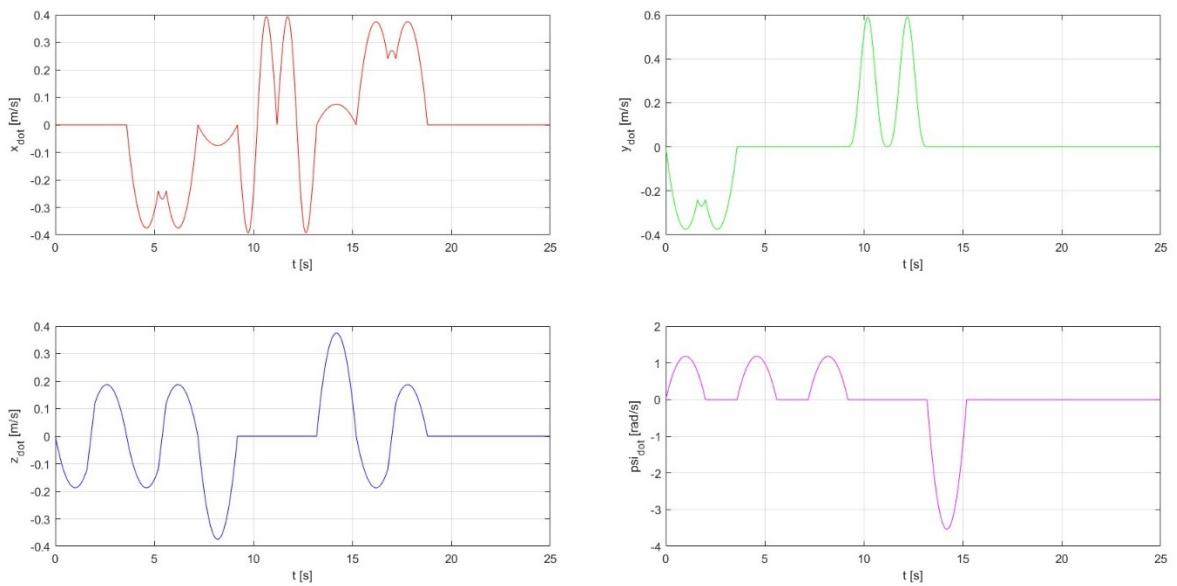


Fig. 3.3: Andamento delle componenti della velocità dell'EE.

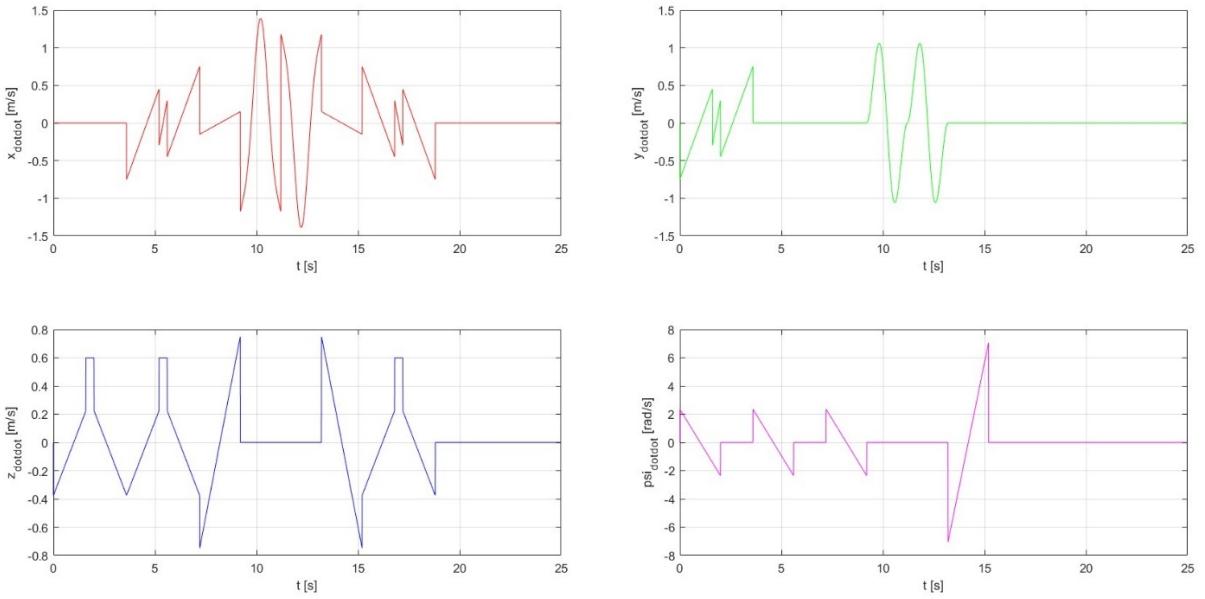


Fig. 3.4: Andamento delle componenti dell'accelerazione dell'EE.

È possibile osservare dalla figura 3.3 come le velocità risultino essere continue in corrispondenza dei punti di cammino, come desiderato, e come esse non si annullino in corrispondenza dei punti di via.

4. Inversione cinematica

La traiettoria per il nostro manipolatore è stata pianificata nello spazio operativo. Al fine di rendere possibile la sua esecuzione tramite un controllo nello spazio dei giunti si rende necessaria un'inversione cinematica che, a partire dalle variabili di riferimento nello spazio operativo, calcoli i riferimenti per le variabili di giunto.

Gli algoritmi di inversione cinematica che ci apprestiamo ad analizzare si fondano sull'utilizzo dello schema CLIK (Closed Loop Inverted Kinematics), in cui l'operazione di inversione avviene in un ciclo chiuso: ciò garantisce errore nullo tra posa di riferimento in ingresso e posa calcolata tramite cinematica diretta a partire dai riferimenti ai giunti ottenuti. Normalmente tali algoritmi richiederebbero l'utilizzo dello Jacobiano analitico, ma per quanto visto nel capitolo 1 tale Jacobiano coincide con quello geometrico in questo caso specifico.

In Simulink verrà adottato per le simulazioni il solver ode1 (Eulero in avanti) con tempo d'integrazione di 1 millisecondo.

4.1. Algoritmo CLIK con inversa dello Jacobiano

L'algoritmo CLIK con inversa dello Jacobiano si fonda sulla relazione:

$$\dot{q} = J_A^{-1}(q)(\dot{x}_d + K e) \quad (4.1.1)$$

ove:

$$e = x_d - x_e \quad (4.1.2)$$

con x_d posa desiderata per l'EE e x_e la posa effettiva del manipolatore, ricostruita a partire dalle variabili di giunto. La (4.1.1) porta alla seguente equazione d'errore:

$$\dot{e} + Ke = 0 \quad (4.1.3)$$

convergente a zero per K definita positiva.

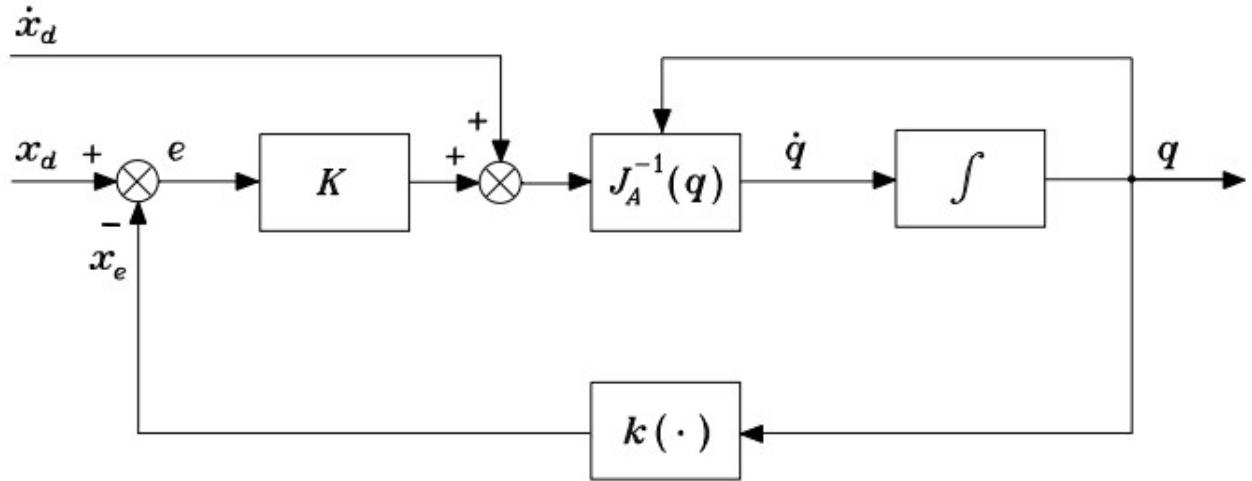


Fig. 4.1.1: Schema di principio dell'algoritmo CLIK con inversa dello Jacobiano.

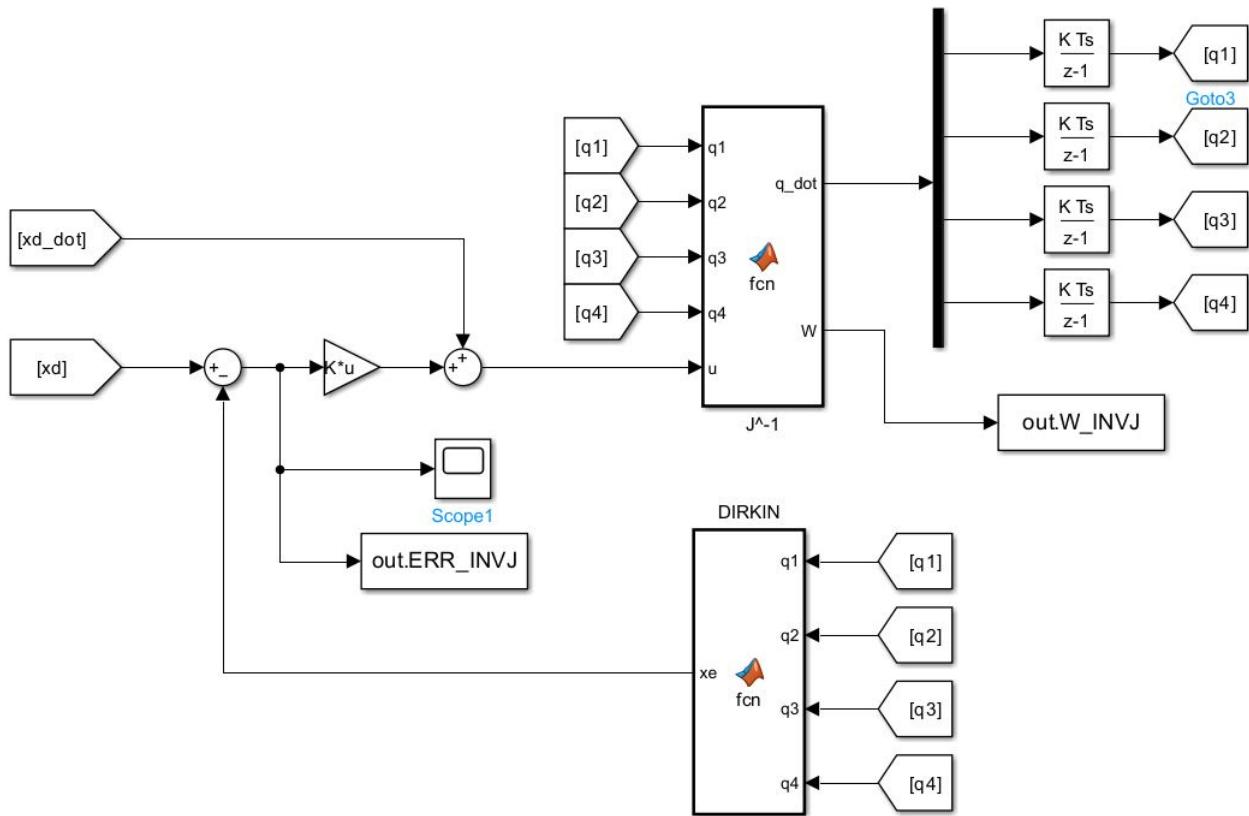


Fig. 4.1.2: Schema Simulink dell'algoritmo CLIK con inversa dello Jacobiano.

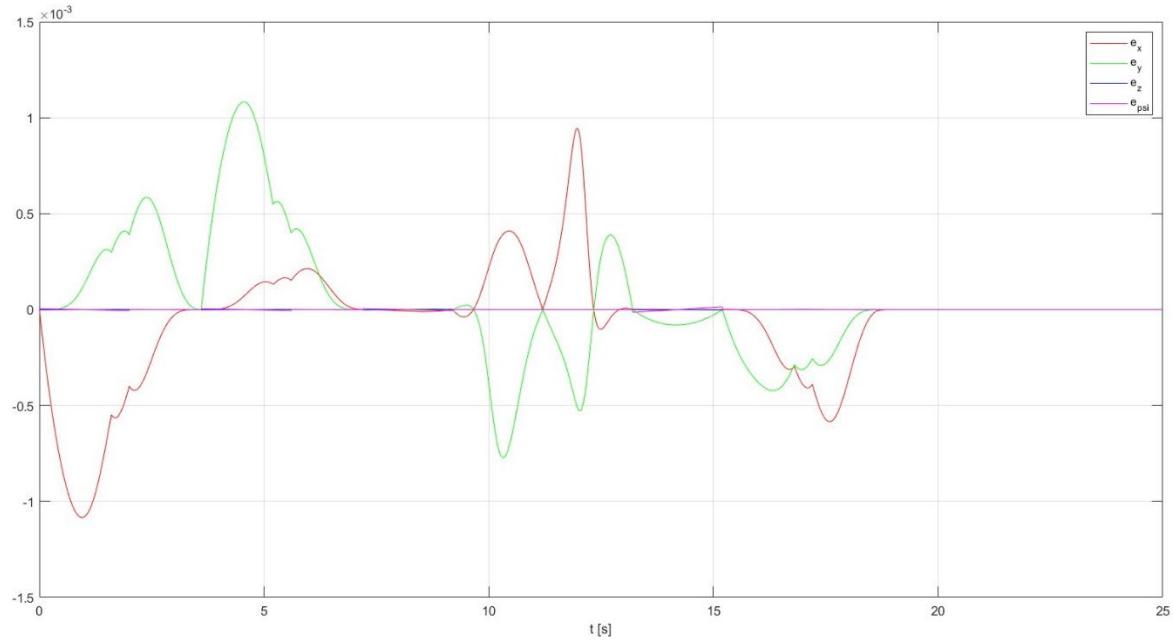


Fig. 4.1.3: Andamento dell'errore per la posa dell'EE con CLIK con inversa dello Jacobiano.

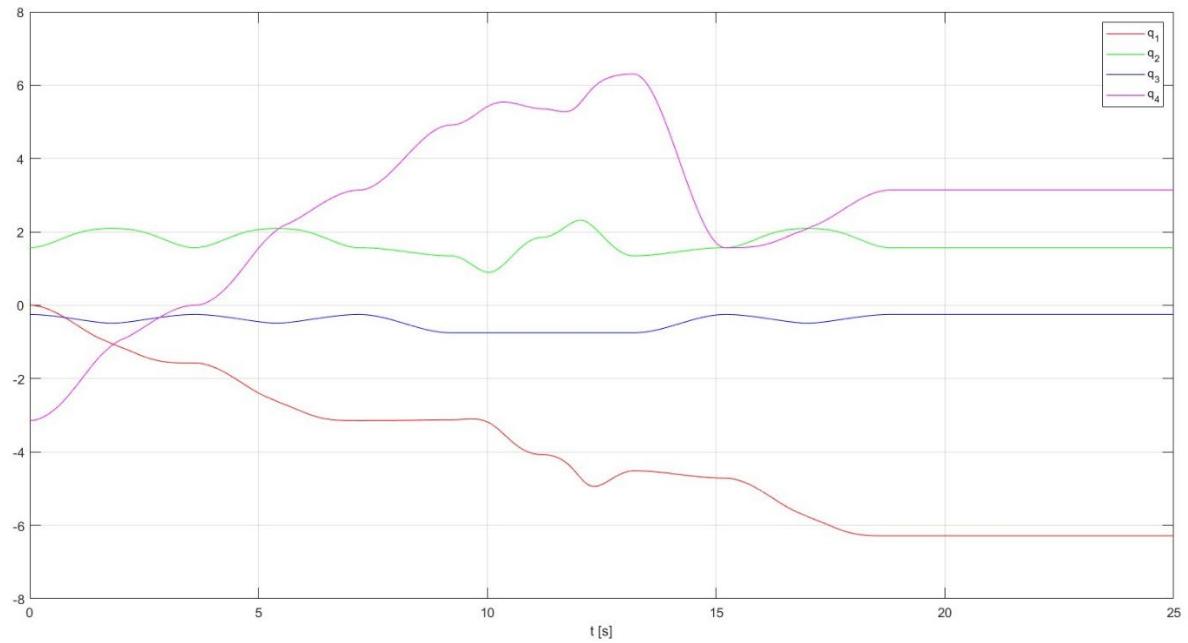


Fig. 4.1.4: Andamento delle variabili di giunto con CLIK con inversa dello Jacobiano.

Scegliendo quale matrice dei guadagni $K = 500 * I$, con $I \in \mathbb{R}^{4 \times 4}$ matrice identità, si ottengono gli andamenti dell'errore nello spazio operativo e i riferimenti per le variabili di giunto necessari al compimento della traiettoria di riferimento, rispettivamente riportati nelle figure 4.1.3 e 4.1.4. È possibile osservare che l'errore risulta essere estremamente contenuto in transitorio e convergente a zero a regime, il ché è un risultato soddisfacente.

4.2. Algoritmo CLIK con trasposta dello Jacobiano

L'inversione dello Jacobiano è un'operazione azzardata qualora il robot sia in condizione di adottare configurazioni singolari, condizione nel quale l'operazione non è possibile a causa del determinante nullo dello Jacobiano. Ciò è verificabile inizializzando a 0 gli integratori nello schema in figura 4.1.2: ciò corrisponde a posizionare il robot inizialmente tutto steso, il ché porta a una singolarità di gomito e all'impossibilità di eseguire la simulazione.

Al fine di irrobustire lo schema nei confronti delle singolarità e diminuire la complessità computazionale dell'algoritmo è possibile sostituire l'inversa dello Jacobiano con la sua trasposta. Di contro, l'errore sulle variabili nello spazio operativo tenderà asintoticamente a zero solamente nel caso di $\dot{x}_d = 0$ e nel caso in cui $Ke \notin N(J_A^T(q))$.

L'algoritmo CLIK con trasposta dello Jacobiano si fonda sulla relazione:

$$\dot{q} = J_A^T(q)Ke \quad (4.2.1)$$

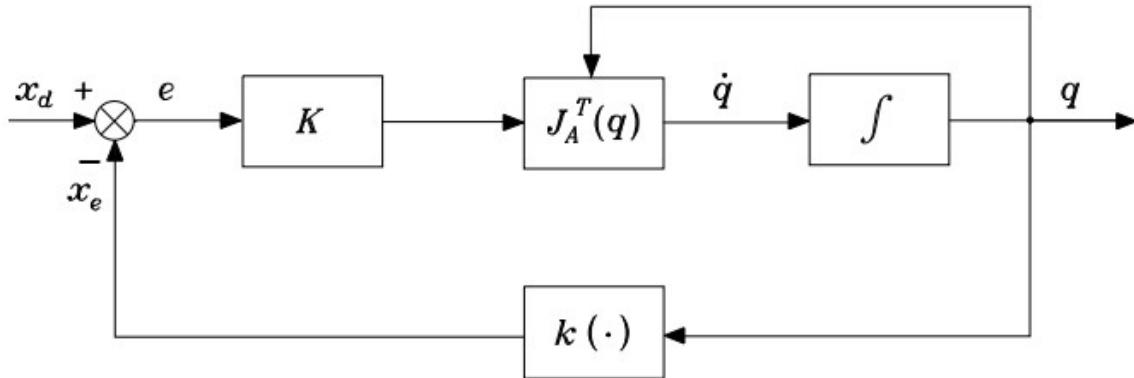


Fig. 4.2.1: Schema di principio dell'algoritmo CLIK con trasposta dello Jacobiano.

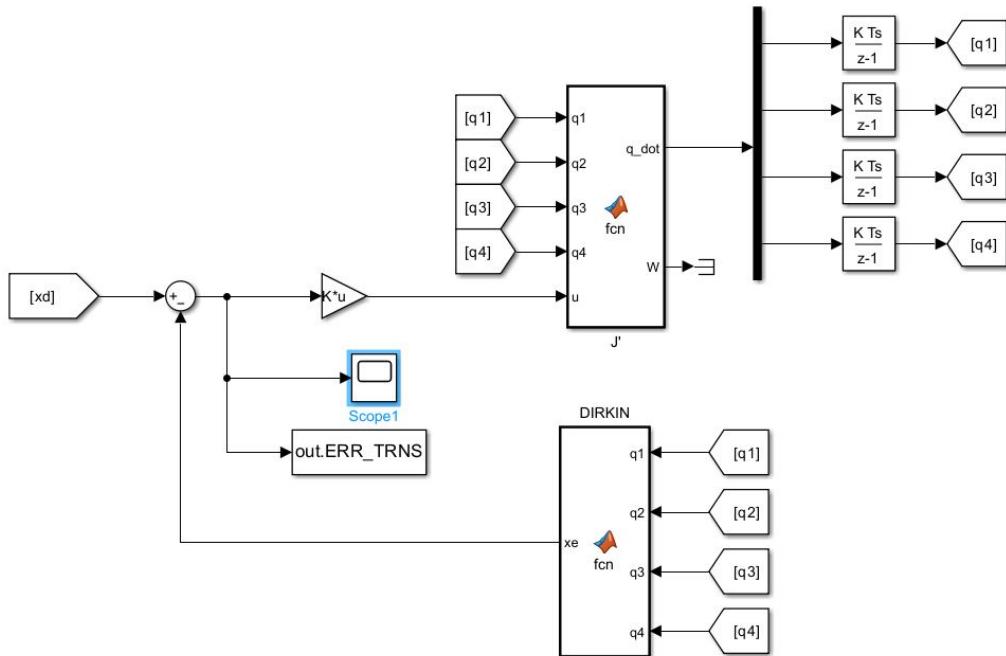


Fig. 4.2.2: Schema Simulink dell'algoritmo CLIK con trasposta dello Jacobiano.

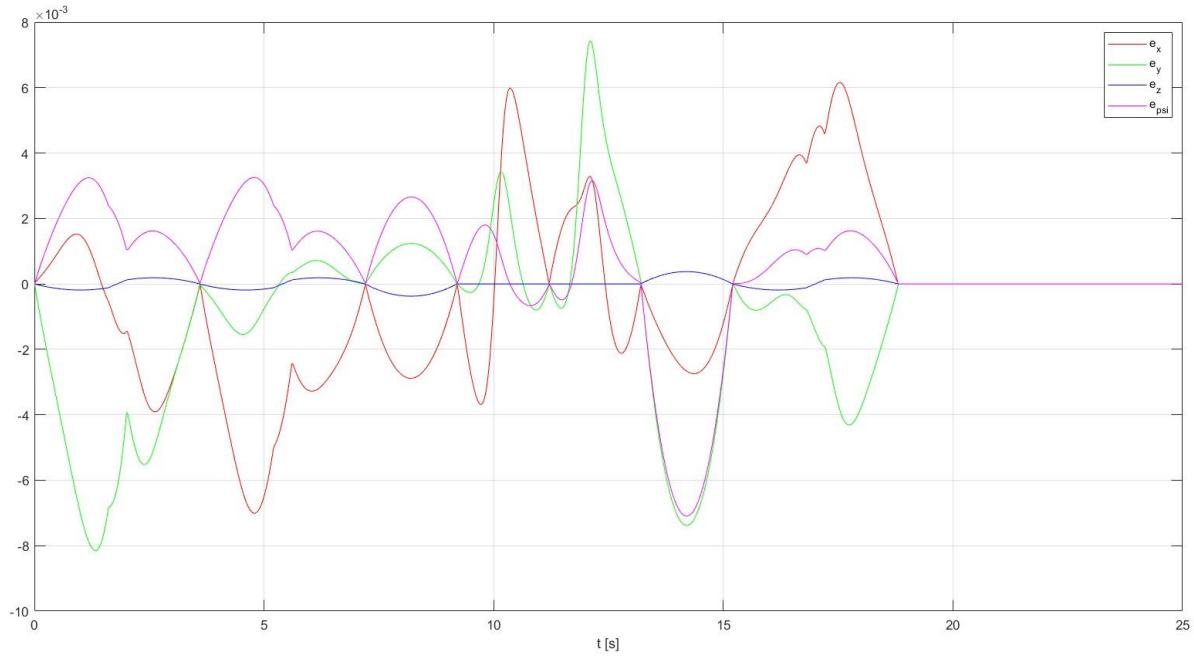


Fig. 4.2.3: Andamento dell'errore per la posa dell'EE con CLIK con trasposta dello Jacobiano.

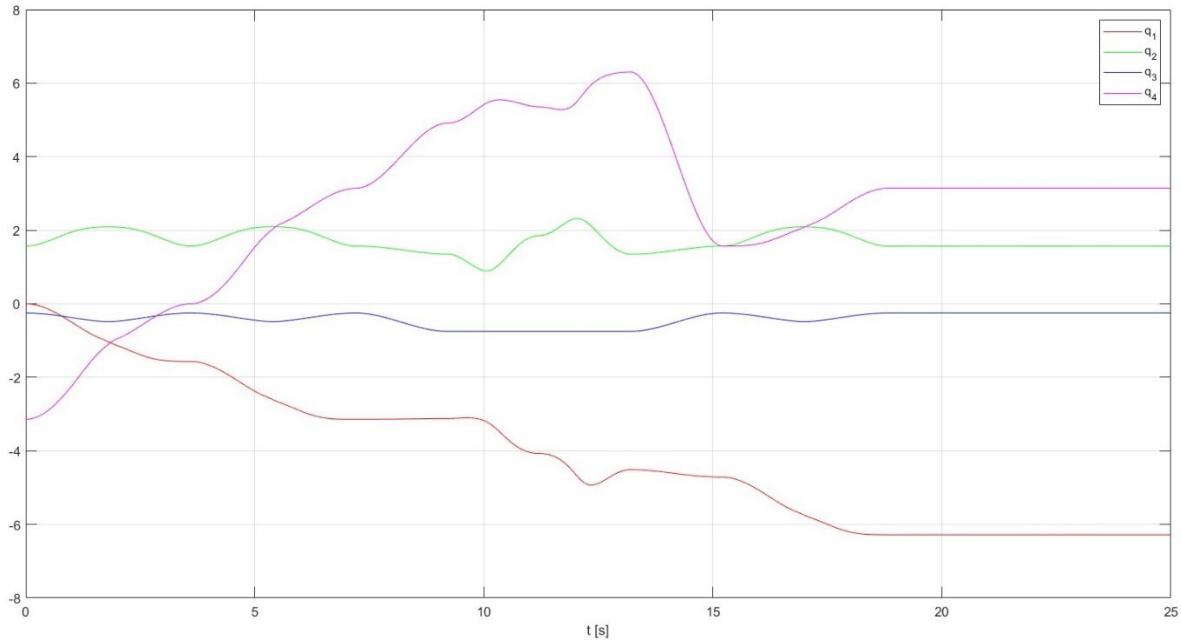


Fig. 4.2.4: Andamento delle variabili di giunto con CLIK con trasposta dello Jacobiano.

La matrice dei guadagni K è stata scelta pari a:

$$K = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 500 \end{bmatrix} \quad (4.2.2)$$

È possibile osservare come, nonostante dei guadagni maggiori, gli errori in transitorio relativamente alle variabili nello spazio operativo risultino essere molto maggiori di quelli ottenuti adottando lo schema con inversa dello Jacobiano. Al termine della traiettoria il manipolatore è fermo ed è possibile osservare la convergenza a zero dell'errore.

4.3. Algoritmo CLIK con pseudo-inversa dello Jacobiano

Il robot in esame non risulta intrinsecamente ridondante, tuttavia è possibile renderlo tale rilassando un vincolo su una delle componenti della posa dell'EE. In tal caso, si verifica che il nullo dello Jacobiano risulta avere dimensione non più nulla. In particolare:

$$\dim(N(J)) = n - r \quad (4.3.1)$$

con n numero di giunti e r numero di gradi di libertà necessari a descrivere il task. Ne consegue che a valle del rilassamento il nullo dello Jacobiano avrà dimensione pari a uno. È possibile sfruttare tale condizione per assegnare un task secondario, l'esecuzione del quale non andrà a interferire con quella del task primario tramite un adeguato proiettore nel nullo dello Jacobiano del primo. Quale task secondario, sceglieremo di ottimizzare la misura di manipolabilità, per la cui espressione facciamo riferimento alle equazioni 2.5 e 2.6.

l'espressione di \dot{q} sarà:

$$\dot{q} = J_A^\dagger(q)(x_d + Ke) + (I - J_A^\dagger(q)J_A(q))\dot{q}_0 \quad (4.3.2)$$

ove:

$$J_A^\dagger(q) = J_A^T(q)(J_A(q)J_A^T(q))^{-1} \quad (4.3.3)$$

$$\dot{q}_0 = K_0 \left(\frac{\partial w(q)}{\partial q} \right) \quad (4.3.4)$$

La velocità \dot{q}_0 sarà proiettata nel nullo dello Jacobiano, dunque non provocherà variazioni nel moto dell'EE. Tuttavia imponendola secondo la (4.3.4) possiamo massimizzare la misura di destrezza. Nelle analisi successive il manipolatore è stato reso ridondante rilassando il vincolo sulla coordinata x .

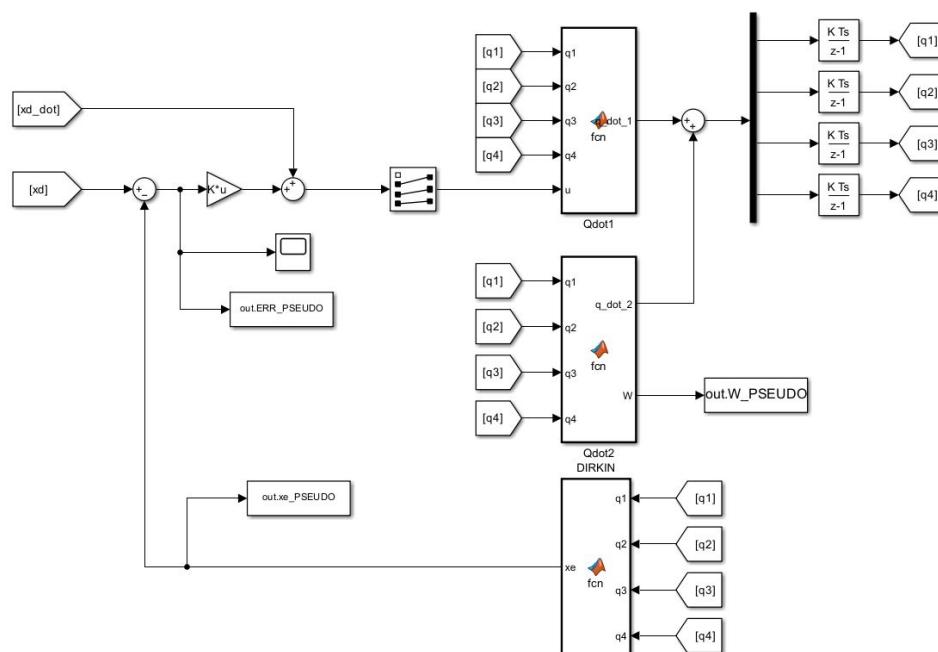


Fig. 4.3.1: Schema Simulink dell'algoritmo CLIK con pseudo-inversa dello Jacobiano.

La matrice dei guadagni K è stata scelta pari alla (4.2.2), mentre $K_0 = 100 * I$, con $I \in \mathbb{R}^{4 \times 4}$ matrice identità. Operando un confronto tra le misure di manipolabilità ottenute dai due schemi CLIK con inversa e con pseudo-inversa dello Jacobiano (figure 4.3.4 e 4.3.5) è possibile osservare come quella ottenuta dal secondo schema, eccezion fatta per un crollo tra i secondi 10 e 11, risulti generalmente maggiore rispetto alla prima, mantenendosi in generale al massimo valore possibile.

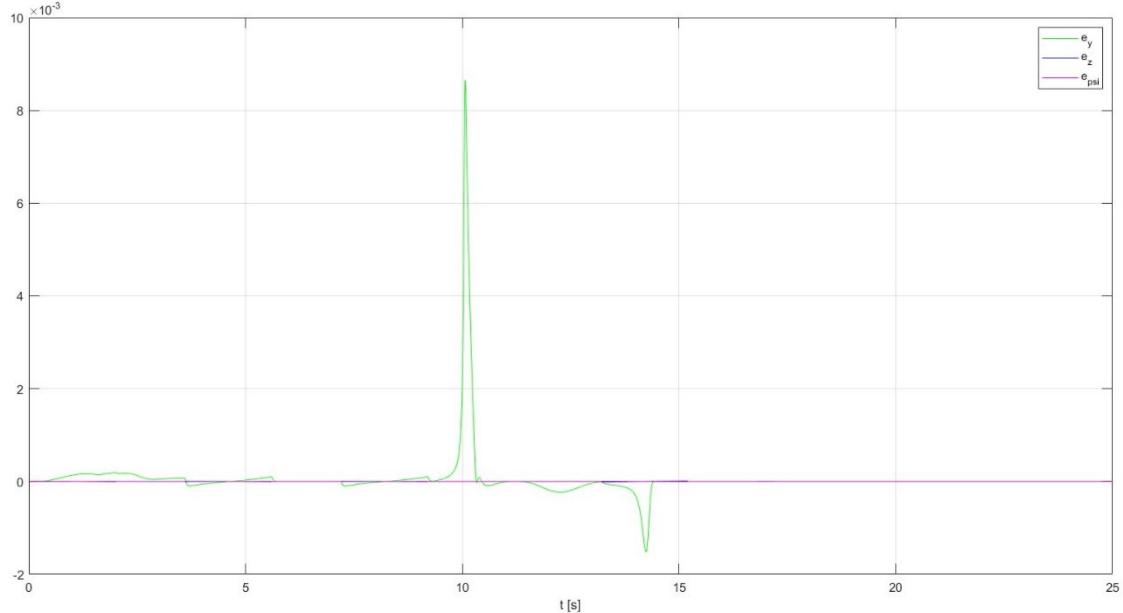


Fig. 4.3.2: Andamento dell'errore per la posa dell'EE con CLIK con pseudo-inversa dello Jacobiano.

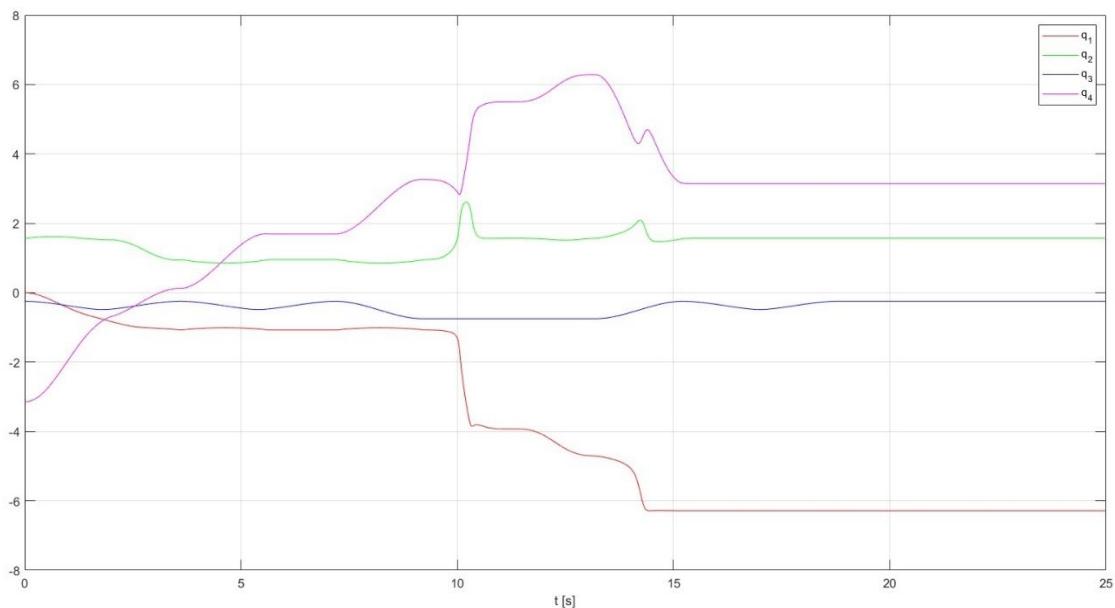


Fig. 4.3.3: Andamento delle variabili di giunto con CLIK con trasposta dello Jacobiano.

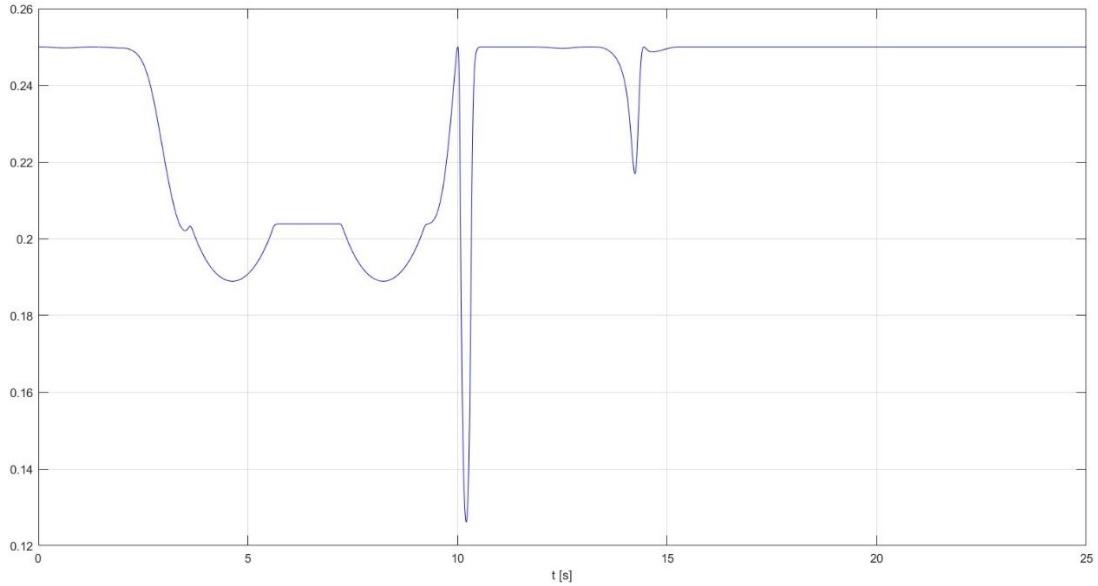


Fig. 4.3.4: Misura di manipolabilità con CLIK con pseudo-inversa dello Jacobiano.

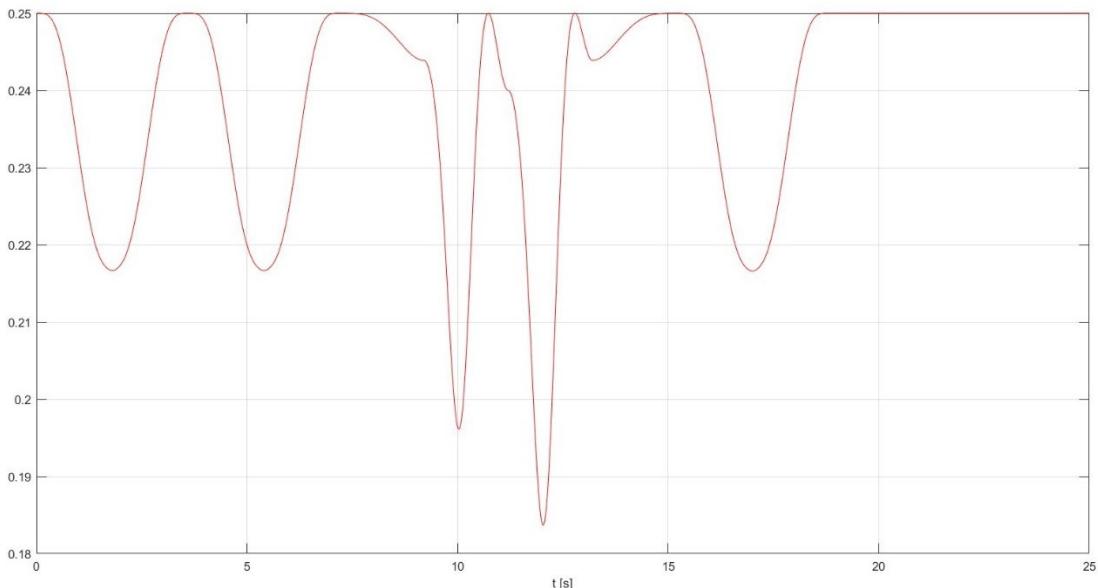


Fig. 4.3.5: Misura di manipolabilità con CLIK con inversa dello Jacobiano.

4.4. Algoritmo CLIK del secondo ordine

Gli schemi mostrati finora permettono di inseguire un riferimento di posizione con errore nullo a regime. Tuttavia vi sono riferimenti di velocità e accelerazione che tali schemi non sono in grado di inseguire. A tal fine è necessario adottare un algoritmo di inversione cinematica che lavori sulla derivata seconda dell'errore: si parla di algoritmo CLIK del secondo ordine.

Tale algoritmo si fonda sulla relazione:

$$\ddot{q} = J_A^{-1}(q)(\ddot{x}_d - j_A(q, \dot{q})\dot{q} + K_D \dot{e} + K_P e) \quad (4.4.1)$$

dalla quale si ricava il sistema d'errore:

$$\ddot{e} + K_D \dot{e} + K_P e = 0 \quad (4.4.2)$$

il quale risulta asintoticamente stabile per K_D e K_P definite positive. Nel caso in esame, tali matrici sono state scelte come segue:

$$K_P = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 5000 & 0 & 0 \\ 0 & 0 & 5000 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix} \quad (4.4.3)$$

$$K_D = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad (4.4.4)$$

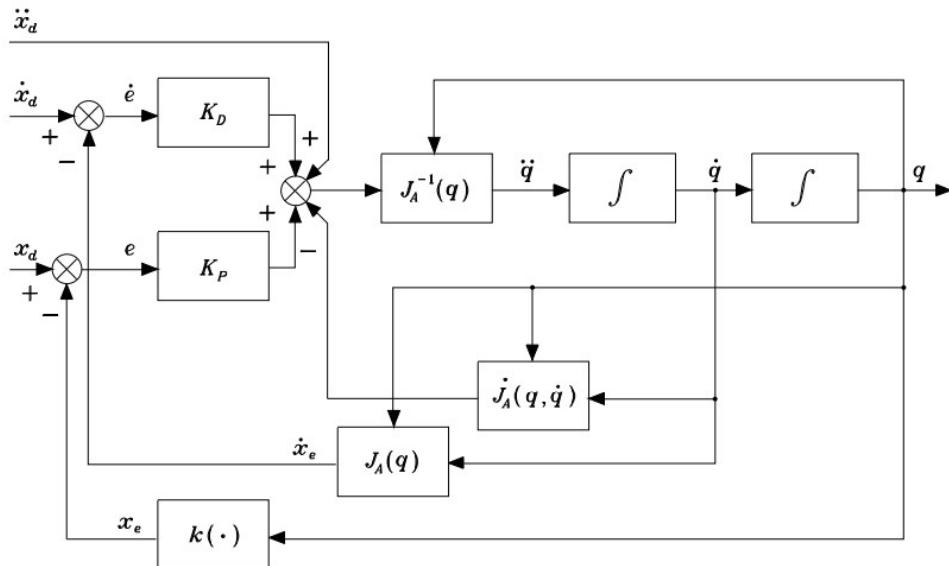


Fig. 4.4.1: Schema di principio dell'algoritmo CLIK del secondo ordine.

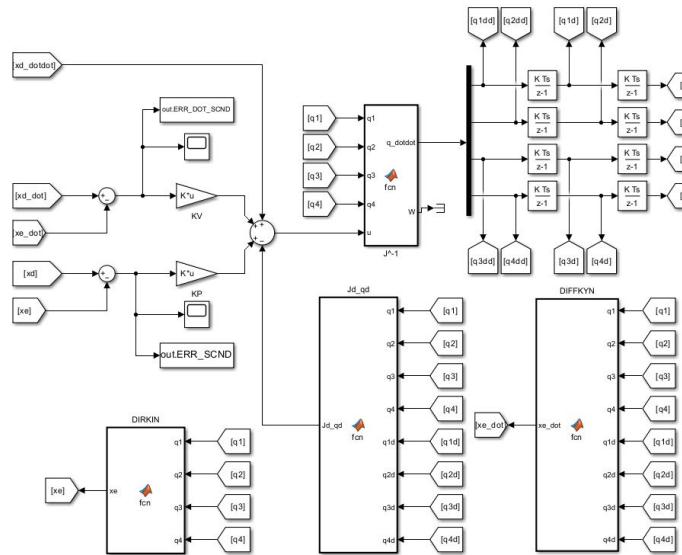


Fig. 4.4.2: Schema Simulink dell'algoritmo CLIK del secondo ordine.

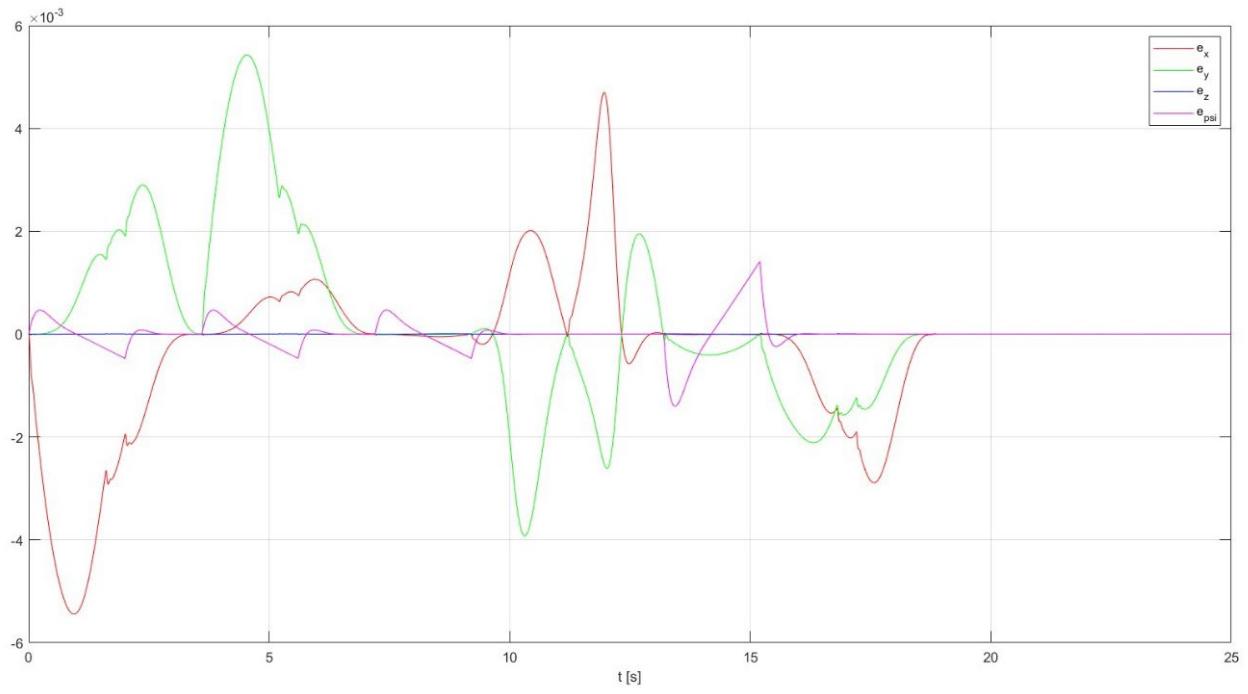


Fig. 4.4.3: Andamento dell'errore per la posa dell'EE con CLIK del secondo ordine.

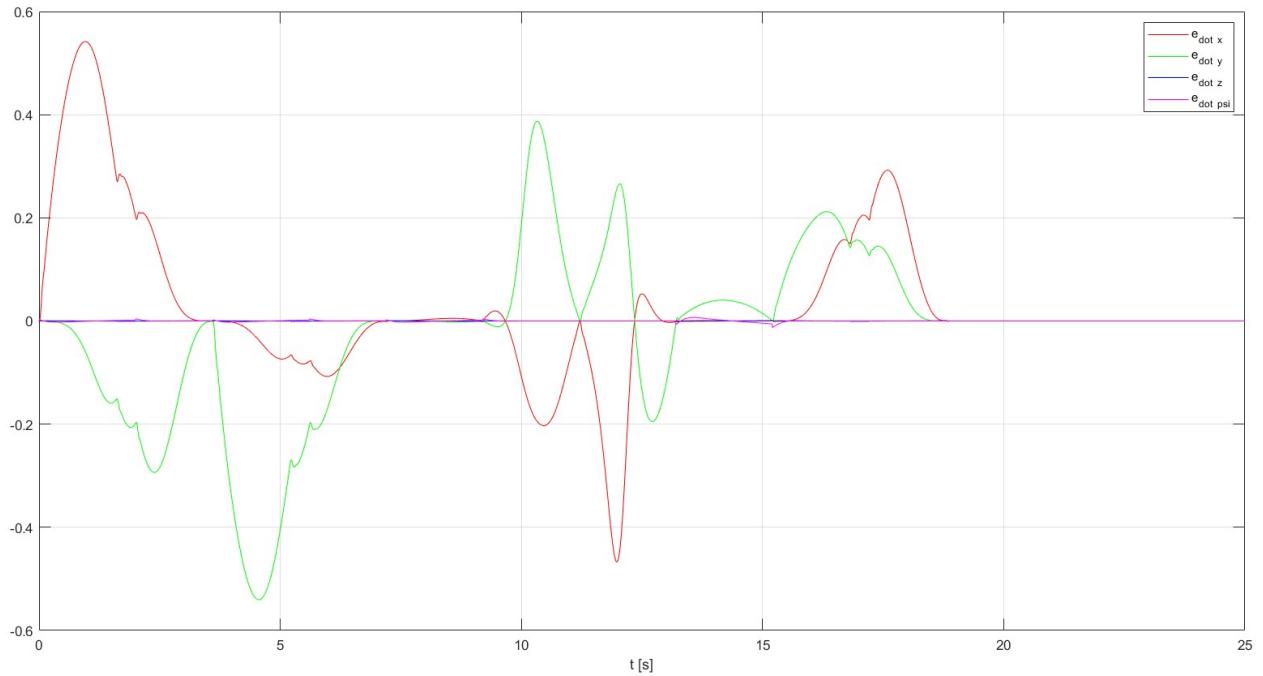


Fig. 4.4.4: Andamento dell'errore per la velocità dell'EE con CLIK del secondo ordine.

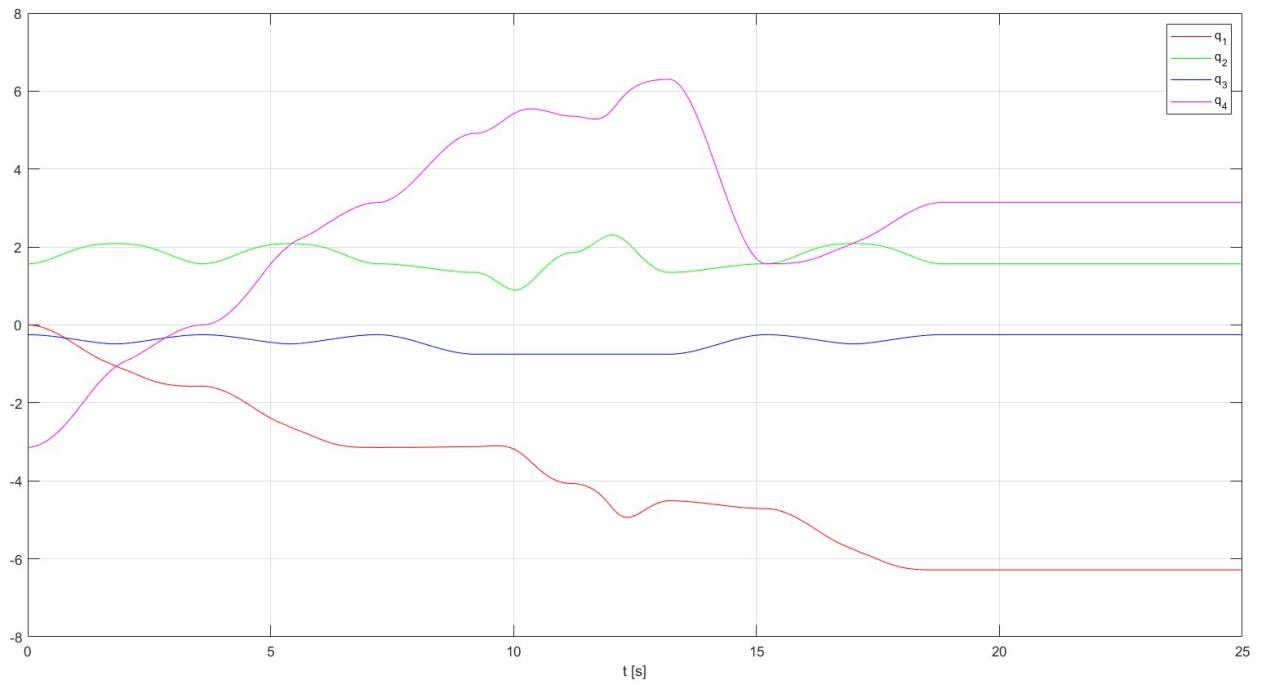


Fig. 4.4.5: Andamento delle variabili di giunto con CLIK del secondo ordine.

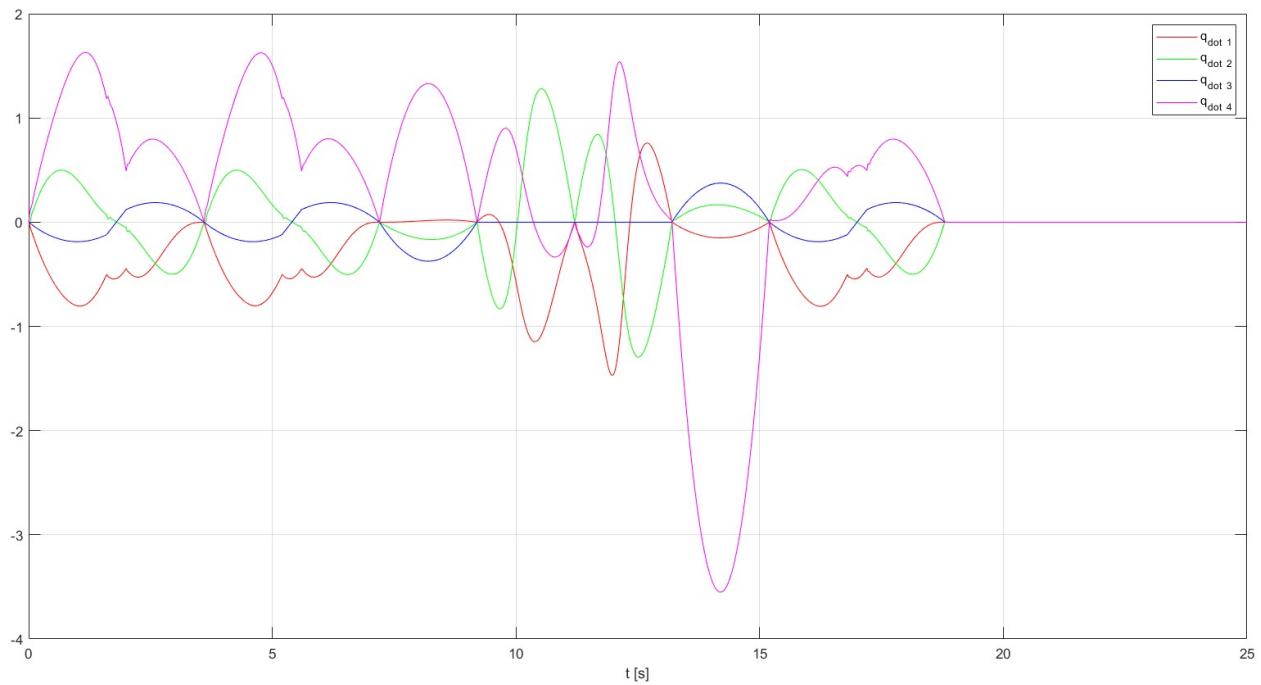


Fig. 4.4.6: Andamento delle velocità ai giunti con CLIK del secondo ordine.

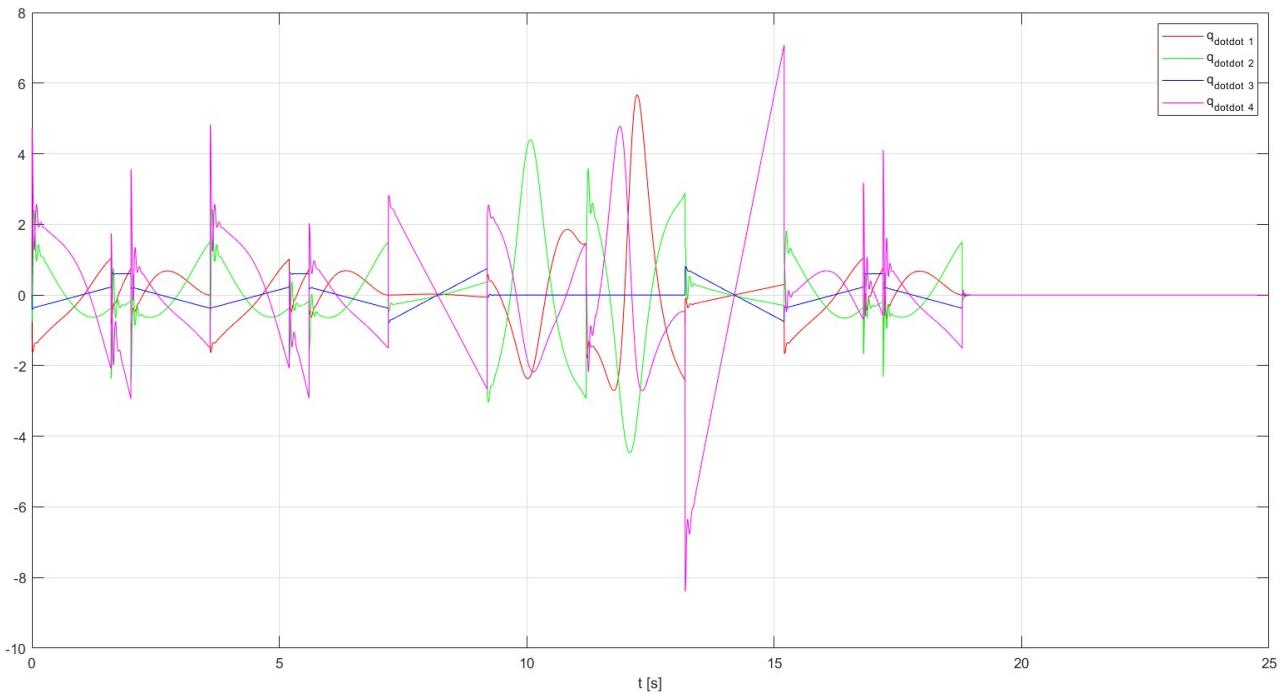


Fig. 4.4.7: Andamento delle accelerazioni ai giunti con CLIK del secondo ordine.

5. Controllo del moto

In questo capitolo viene affrontato il problema del controllo del robot sulla base dei riferimenti prodotti finora. A valle del calcolo del modello dinamico del manipolatore analizzeremo tre differenti schemi di controllo:

- controllo robusto;
- controllo adattativo;
- controllo a dinamica inversa nello spazio operativo.

5.1. Modello dinamico

Il modello dinamico del robot è stato ricavato tramite il metodo di Lagrange. Ipotizzando l'assenza di attrito statico e di forze d'interazione, si vuole giungere a una formulazione del tipo:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_V\dot{q} + g(q) = \tau \quad (5.1.1)$$

ove:

- $B(q)$ è la matrice d'inerzia definita come:

$$B(q) = \sum_{i=1}^n \left(m_{l_i} J_P^{(l_i)T} J_P^{(l_i)} + J_O^{(l_i)T} R_i I_{l_i}^i R_i^T J_O^{(l_i)} + m_{m_i} J_P^{(m_i)T} J_P^{(m_i)} + J_O^{(m_i)T} R_{m_i} I_{m_i}^i R_{m_i}^T J_O^{(m_i)} \right) \quad (5.1.2)$$

Nella (5.1.2):

- m_{l_i} è la massa dell'i-esimo braccio;

- $J_P^{l_i}$ è lo Jacobiano di posizione dell'i-esimo braccio;
- $J_O^{l_i}$ è lo Jacobiano di orientamento dell'i-esimo braccio;
- $I_{l_i}^i$ è il tensore d'inerzia dell'i-esimo braccio rispetto alla terna i-esima;
- m_{m_i} è la massa dell'i-esimo motore;
- $J_P^{m_i}$ è lo Jacobiano di posizione dell'i-esimo motore;
- $J_O^{m_i}$ è lo Jacobiano di orientamento dell'i-esimo motore;
- $I_{m_i}^i$ è il tensore d'inerzia dell'i-esimo motore rispetto alla terna i-esima.

- $C(q, \dot{q})$ è la matrice che tiene conto degli effetti delle forze di Coriolis e delle forze centrifughe. Utilizzando i simboli di Christoffel del primo tipo è possibile scriverne gli elementi come:

$$c_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_k \quad (5.1.3)$$

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \quad (5.1.4)$$

- F_V è la matrice che tiene conto dell'effetto delle forze d'attrito viscoso, definita come:

$$F_V = \text{diag}(k_{r_1}^2 F_{m_1}, k_{r_2}^2 F_{m_2}, k_{r_3}^2 F_{m_3}, k_{r_4}^2 F_{m_4}) \quad (5.1.5)$$

ove:

- k_{r_i} è il rapporto di trasmissione del giunto i-esimo;
 - F_{m_i} è il coefficiente di attrito viscoso del motore i-esimo.
- $g(q)$ è il termine che esprime l'effetto della forza di gravità. Nel caso in esame essa risulta pari a:

$$g(q) = \begin{bmatrix} 0 \\ 0 \\ -(m_{l_3} + m_{load})g \\ 0 \end{bmatrix} \quad (5.1.6)$$

- τ è il vettore delle coppie di attuazione ai giunti.

Tutti i parametri necessari sono noti e definiti come segue:

$$m_{l_1} = m_{l_2} = 20 \text{ kg}; m_{l_3} = 10 \text{ kg}$$

$$I_{l_1} = I_{l_2} = 4 \text{ kg} \cdot m^2; I_{l_4} = 1 \text{ kg} \cdot m^2$$

$$k_{r1} = k_{r2} = 1; k_{r3} = 50 \text{ rad/m}; k_{r4} = 20$$

$$I_{m_1} = I_{m_2} = 0.01 \text{ kg} \cdot m^2; I_{m_3} = 0.005 \text{ kg} \cdot m^2; I_{m_4} = 0.001 \text{ kg} \cdot m^2$$

$$F_{m1} = F_{m2} = 0.00005 \text{ N} \cdot m \cdot s/rad; F_{m3} = 0.01 \text{ N} \cdot m \cdot s/rad; F_{m4} = 0.005 \text{ N} \cdot m \cdot s/rad$$

5.2. Controllo robusto

Il funzionamento dello schema di controllo robusto fonda le proprie basi su una feedback linearization dei termini non lineari del modello, soggetti a una cancellazione imperfetta. In questo modo si inserisce un'azione robustificante che garantisce buone prestazioni anche in presenza di una conoscenza imperfetta del modello.

L'azione di controllo risulta definita come:

$$\tau = \hat{B}(q)y + \hat{n}(q, \dot{q}) \quad (5.2.1)$$

con $\hat{B}(q)$ e $\hat{n}(q)$ rispettivamente stime della matrice d'inerzia e di:

$$n(q, \dot{q}) = C(q, \dot{q})\dot{q} + F_V\dot{q} + g(q) \quad (5.2.2)$$

Occorre definire la y in (5.2.1). A tal fine, definiamo:

$$\xi = \begin{bmatrix} \tilde{q} \\ \tilde{\dot{q}} \end{bmatrix} = \begin{bmatrix} q - q_d \\ \dot{q} - \dot{q}_d \end{bmatrix} \quad (5.2.3)$$

a partire dalla quale introduciamo:

$$z = D^T Q \xi \quad (5.2.4)$$

ove Q è una matrice di pesi per gli errori di posizione e velocità, definita positiva, mentre:

$$D = \begin{bmatrix} 0 \\ I^4 \end{bmatrix} \quad (5.2.5)$$

A questo punto è possibile definire y come un'azione di controllo che presenta un'azione di feedback e una di feedforward, secondo:

$$y = \ddot{q}_d + K_D \dot{\tilde{q}} + K_P \tilde{q} + w \quad (5.2.6)$$

ove K_D e K_P sono matrici definite positive e w è una legge di controllo a versore:

$$w = \begin{cases} \frac{\rho}{\|z\|} z & se \|z\| \geq \varepsilon \\ \frac{\rho}{\varepsilon} z & se \|z\| < \varepsilon \end{cases} \quad (5.2.7)$$

con ρ costante di guadagno. Una legge di controllo a versore semplice porterebbe a un controllo sliding soggetto a commutazione di frequenza infinita: per evitare che ciò accada si ricorre alla formulazione della (5.2.7) con ε ampiezza della regione all'interno della quale vogliamo rimanere durante la convergenza. Si parla di approccio boundary layer: il chattering è eliminato al costo di un errore al più limitato in norma piuttosto che convergente a zero.

Nel caso in esame, sono state fatte le seguenti scelte:

$$K_P = \begin{bmatrix} 500 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 500 \end{bmatrix}; \quad (5.2.8)$$

$$K_D = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix}; \quad (5.2.9)$$

$$D^T Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.1 \end{bmatrix}; \quad (5.2.10)$$

$$\rho = 1; \varepsilon = 10^{-3} \quad (5.2.11)$$

Al fine di mostrare l'efficacia del controllo è stato ipotizzato un carico $m_{load} = 4 \text{ kg}$ in punta, sconosciuto al controllo. Ne consegue che nelle stime $\hat{B}(q)$ e $\hat{n}(q, \dot{q})$ tale massa non è stata considerata. I riferimenti q_d, \dot{q}_d e \ddot{q}_d sono stati prodotti tramite l'algoritmo CLIK del secondo ordine.

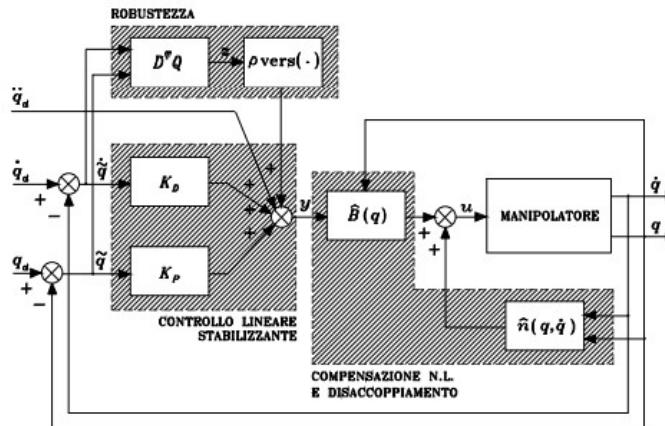


Fig. 5.2.1: Schema di principio dell'algoritmo di controllo robusto.

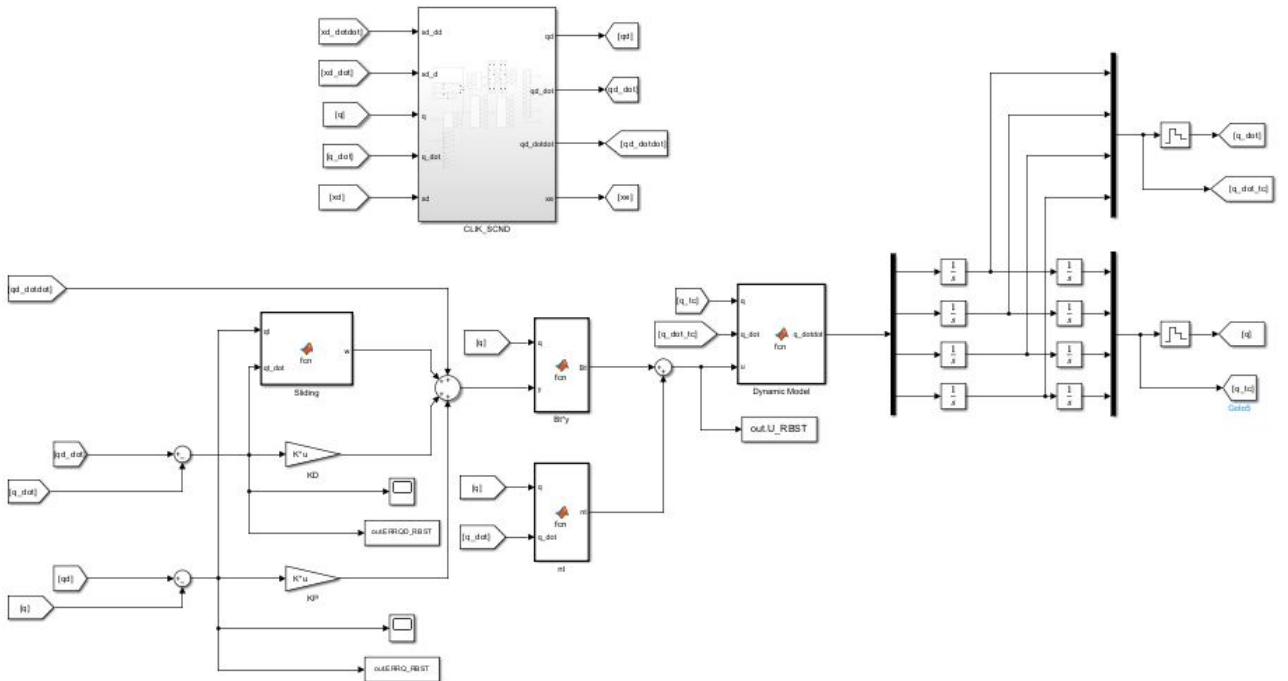


Fig. 5.2.2: Schema Simulink dell'algoritmo di controllo robusto.

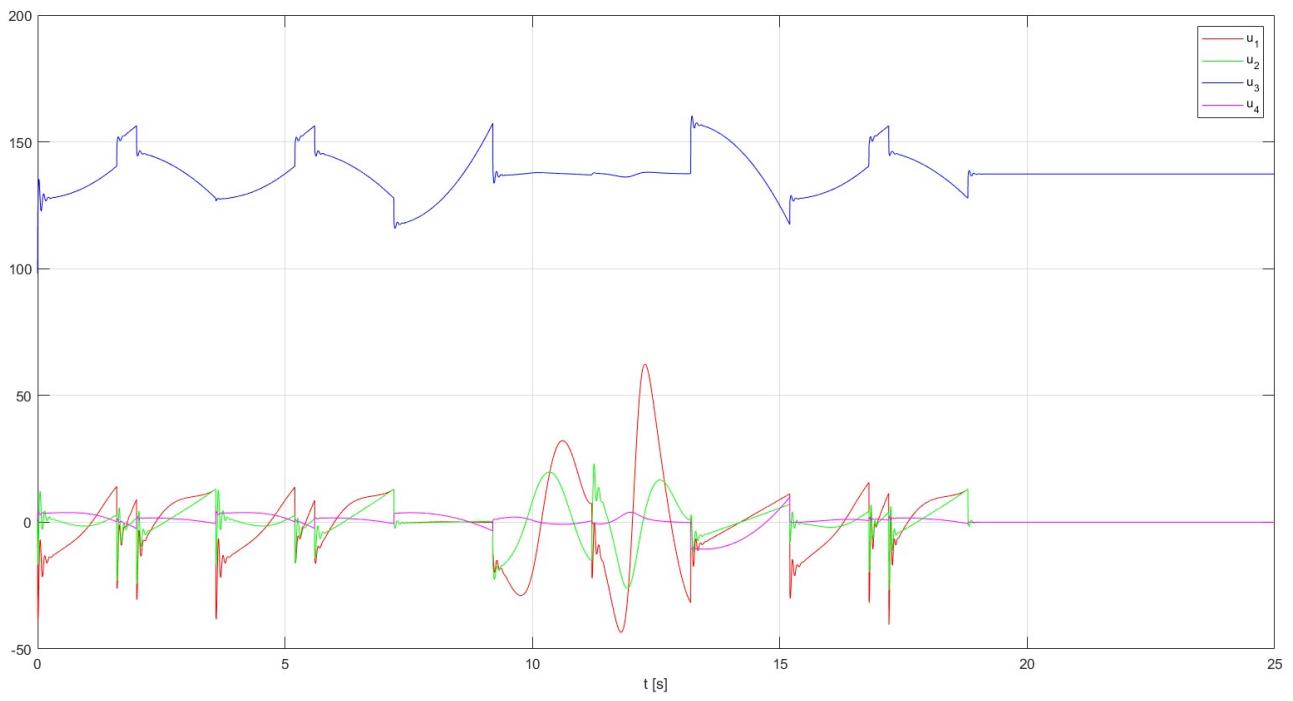


Fig. 5.2.3: Azioni di controllo ai giunti con algoritmo di controllo robusto.

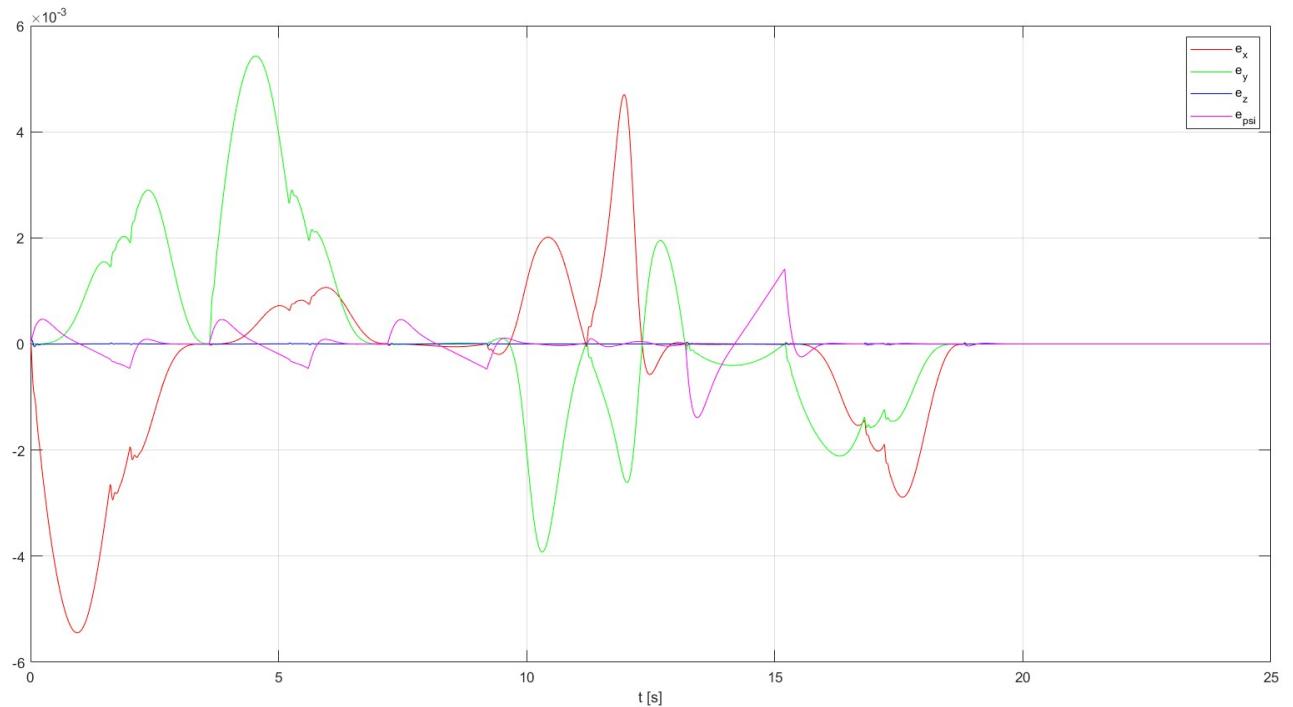


Fig. 5.2.4: Andamento dell'errore per la posa dell'EE con algoritmo di controllo robusto.

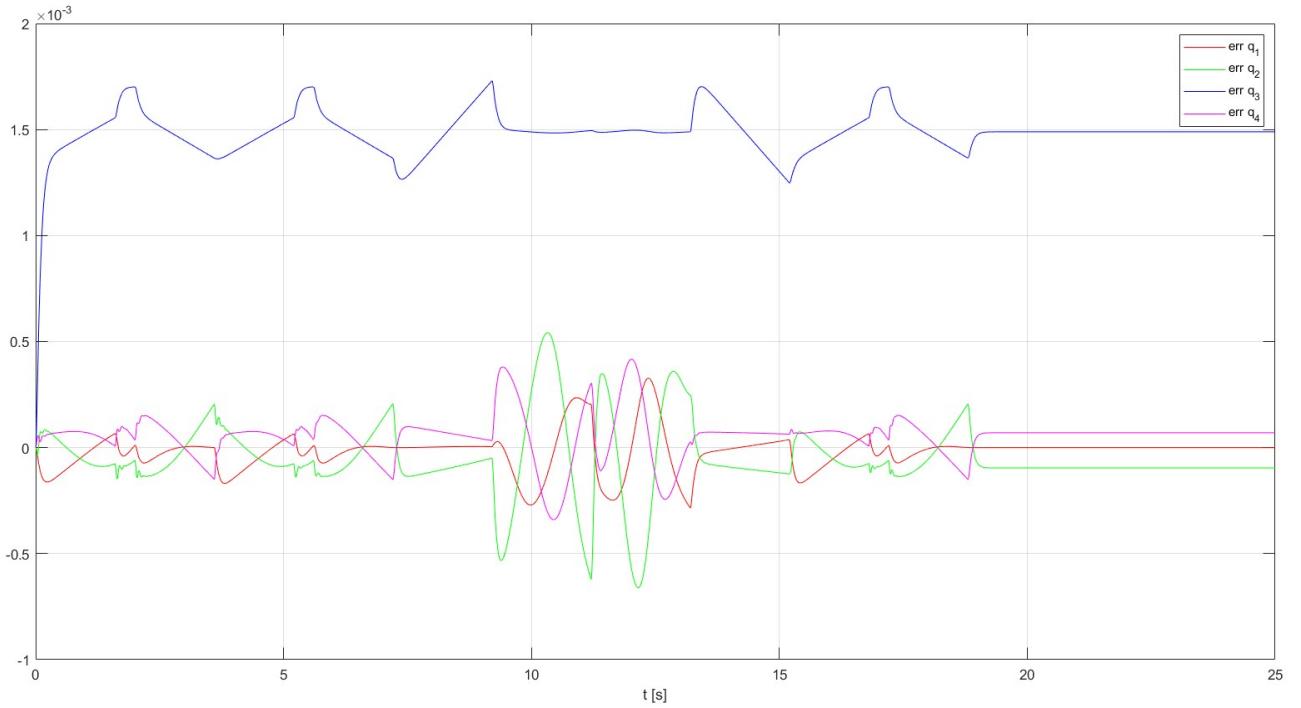


Fig. 5.2.5: Andamento dell'errore per q con algoritmo di controllo robusto.

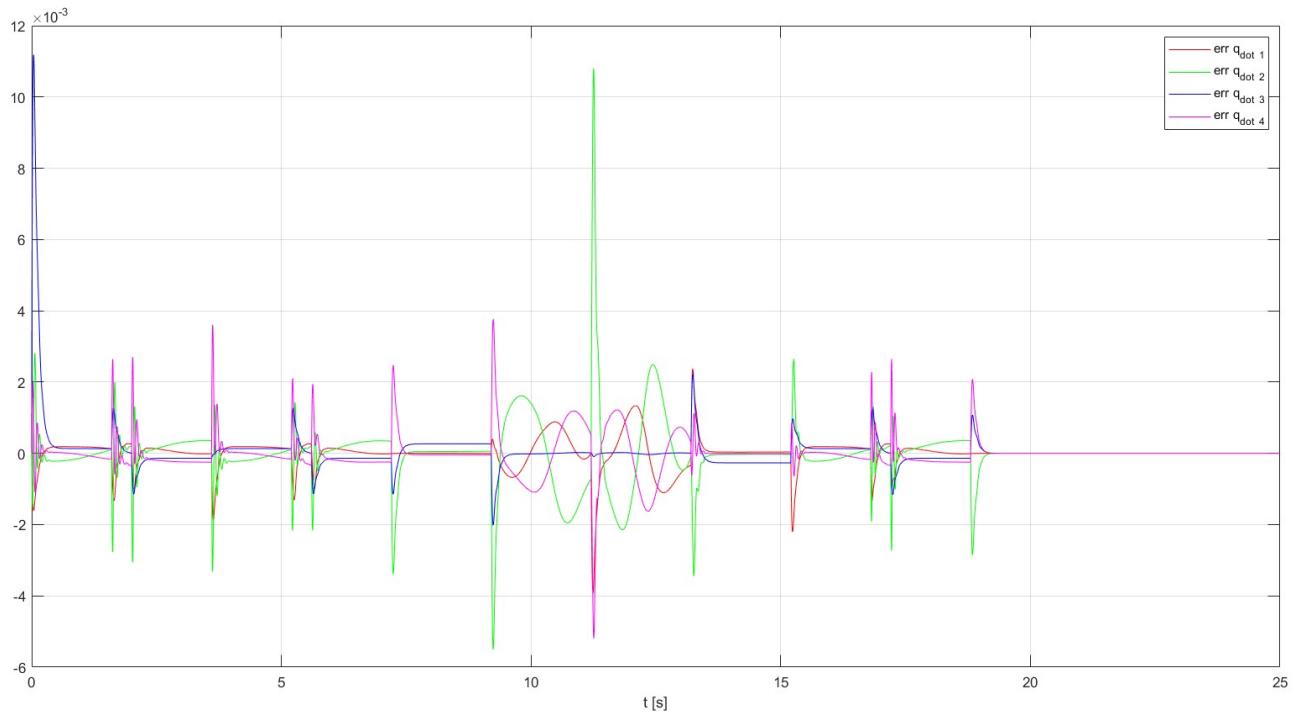


Fig. 5.2.6: Andamento dell'errore per \dot{q} con algoritmo di controllo robusto.

Il controllo offre buone prestazioni, pur non essendo in grado di garantire errore nullo a regime per le variabili di giunto, essendo ε non nullo (figura 5.2.5). Ciò comunque non comporta conseguenze percepibili nello spazio operativo, dove l'errore rimane entro valori accettabili (figura 5.2.4).

5.3. Controllo adattativo

Sia nota la proprietà secondo la quale un modello dinamico può essere espresso come una combinazione lineare di parametri dinamici π secondo una matrice regressore Y:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_V\dot{q} + g(q) = \tau = Y(q, \dot{q}, \ddot{q})\pi \quad (5.3.1)$$

Il controllo adattativo adotta una legge di controllo del tipo:

$$u = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\hat{\pi} + K_D\sigma \quad (5.3.2)$$

ove:

- $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ è la matrice regressore calcolata secondo:

$$Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r) = B(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + F_V\dot{q}_r + g(q) \quad (5.3.3)$$

ove:

$$\dot{q}_r = \dot{q}_d + \Lambda\tilde{q} \quad (5.3.4)$$

$$\ddot{q}_r = \ddot{q}_d + \Lambda\dot{\tilde{q}} \quad (5.3.5)$$

con \tilde{q} definito come alla (5.2.3);

- $\hat{\pi}$ è la stima dei parametri dinamici. In questo caso tali parametri risultano essere stati scelti pari a:

$$\pi = [m_{l_1}l_1^2, m_{l_2}, m_{l_2}l_2^2, m_{l_3}, m_{load}, I_{l_1}, I_{l_2}, I_{l_4}, I_{m_1}, I_{m_2}, I_{m_3}, I_{m_4}, F_{m_1}, F_{m_2}, F_{m_3}, F_{m_4}]^T \quad (5.3.6)$$

- σ è una combinazione lineare degli errori di posizione e velocità ai giunti, pari a:

$$\sigma = \dot{q}_r - \dot{q} = \dot{\tilde{q}} + \Lambda\tilde{q} \quad (5.3.7)$$

È possibile stimare i parametri tramite una legge di adattamento:

$$\dot{\hat{\pi}} = K_\pi^{-1}Y^T(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\sigma \quad (5.3.8)$$

di modo da far tendere l'errore di posizione e velocità ai giunti a zero.

Nel caso in esame sono state fatte le seguenti scelte:

$$\Lambda = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}; \quad (5.3.9)$$

$$K_D = \begin{bmatrix} 800 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 \\ 0 & 0 & 800 & 0 \\ 0 & 0 & 0 & 800 \end{bmatrix}; \quad (5.3.10)$$

Per quanto riguarda la matrice dei guadagni della legge di adattamento K_π , questa è stata presa diagonale con elementi ij pari a 100 eccetto che per $i = j = 6$, indici cui corrisponde la massa in punta. Tale elemento della matrice è stato posto pari a 0.01 di modo da agire più decisivamente sul parametro non noto.

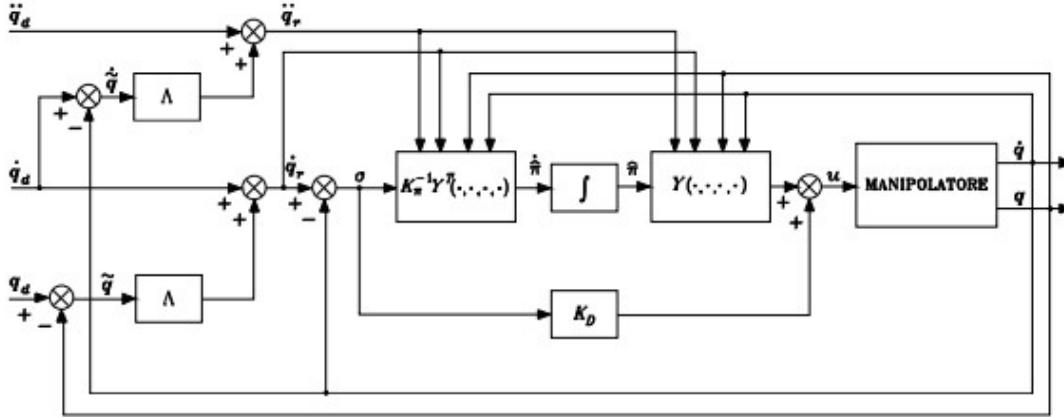


Fig. 5.3.1: Schema di principio dell'algoritmo di controllo adattativo.

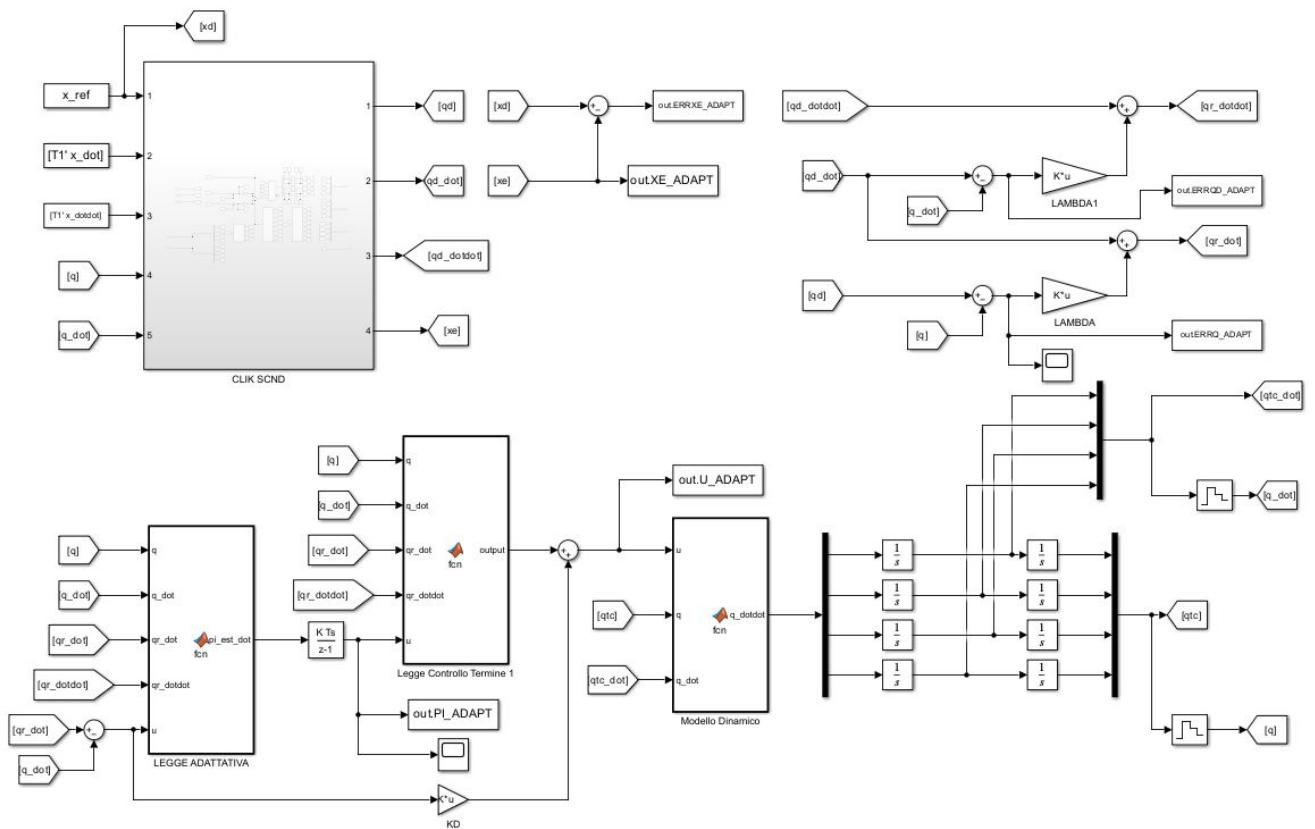


Fig. 5.3.2: Schema Simulink dell'algoritmo di controllo adattativo.

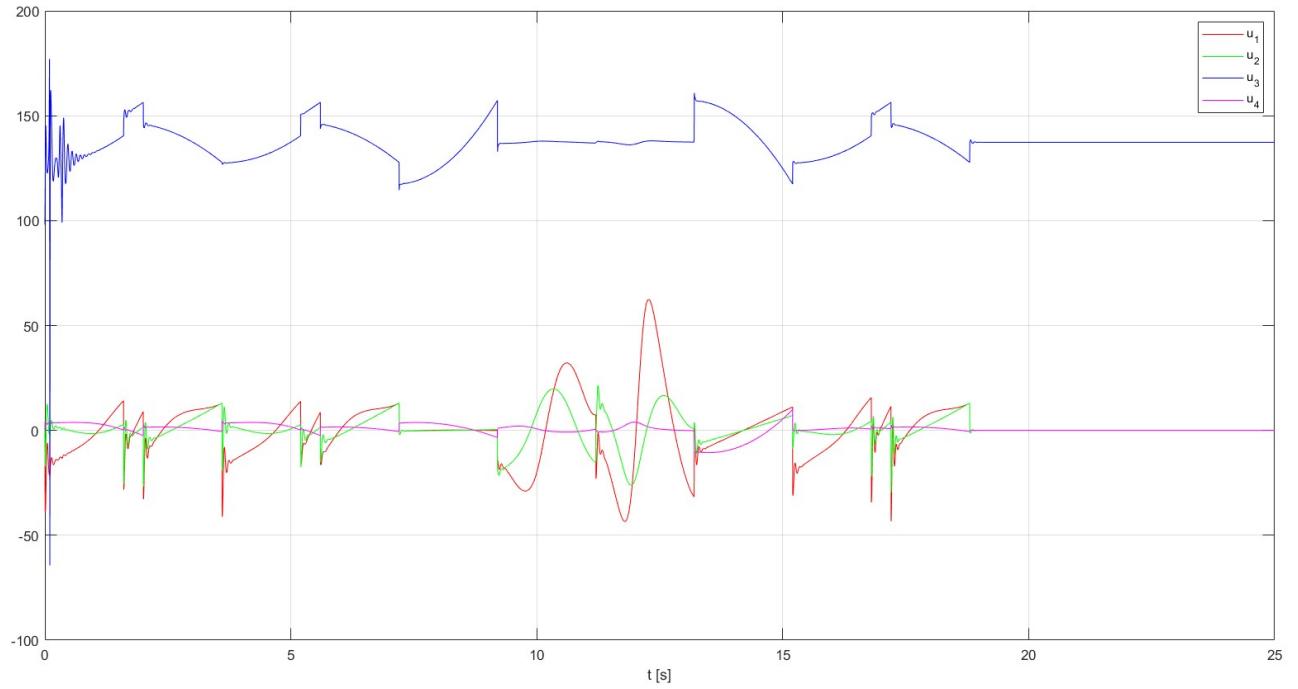


Fig. 5.3.3: Azioni di controllo ai giunti con algoritmo di controllo adattativo.

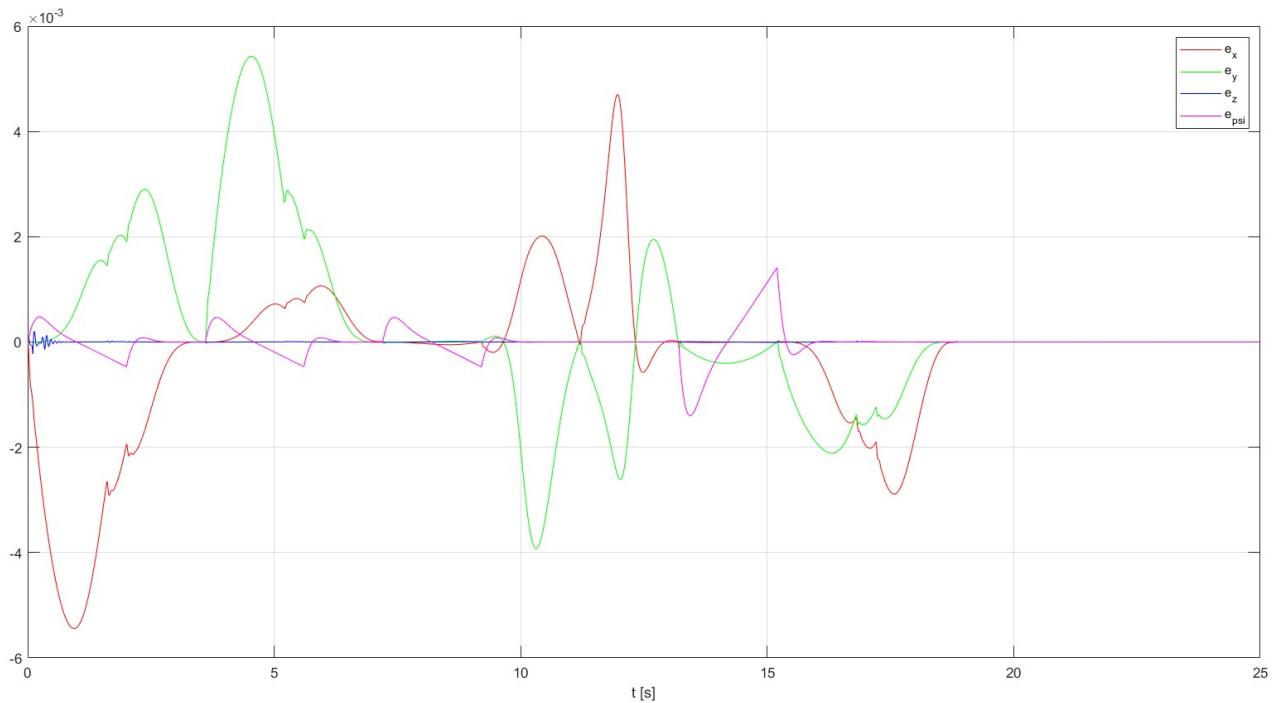


Fig. 5.3.4: Andamento dell'errore per la posa dell'EE con algoritmo di controllo adattativo.

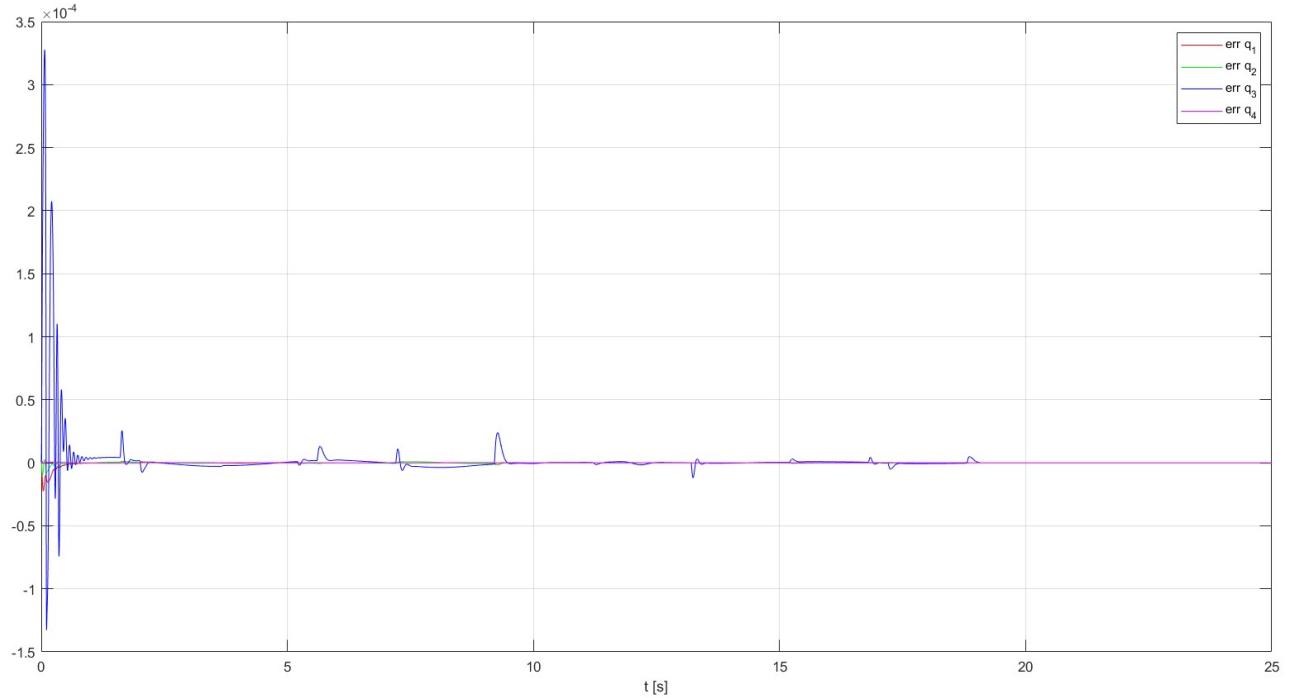


Fig. 5.3.5: Andamento dell'errore per q con algoritmo di controllo adattativo.

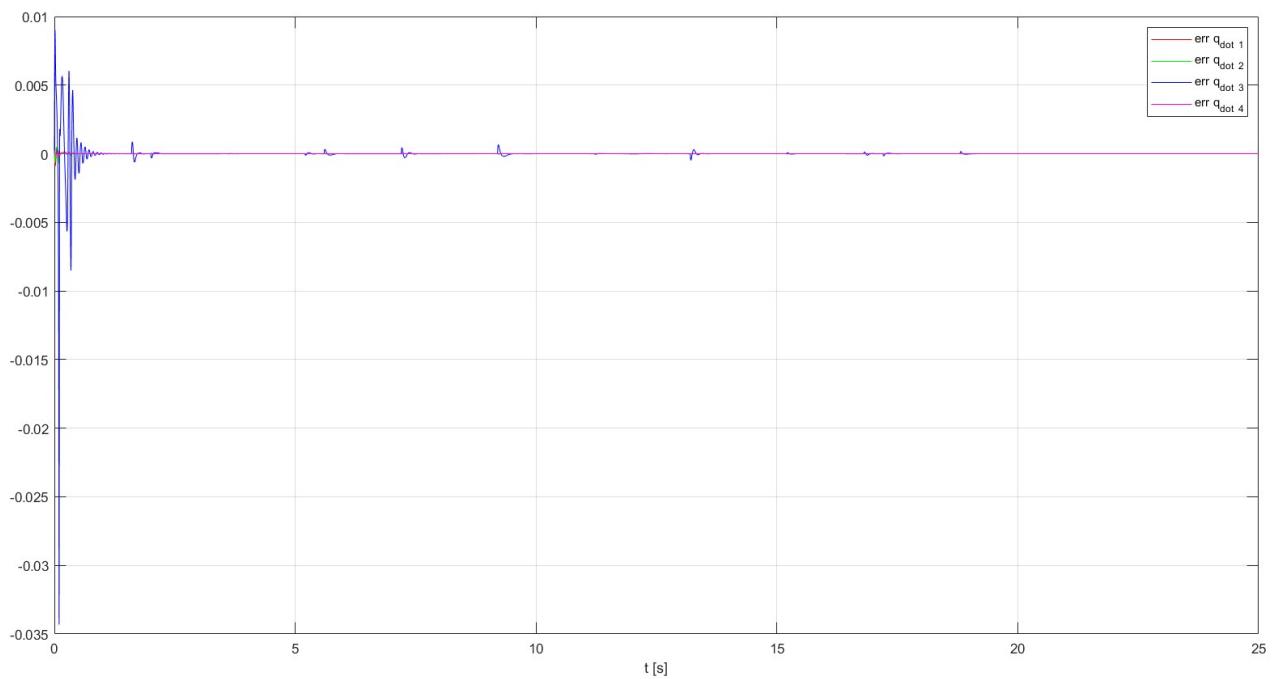


Fig. 5.3.6: Andamento dell'errore per \dot{q} con algoritmo di controllo adattativo.

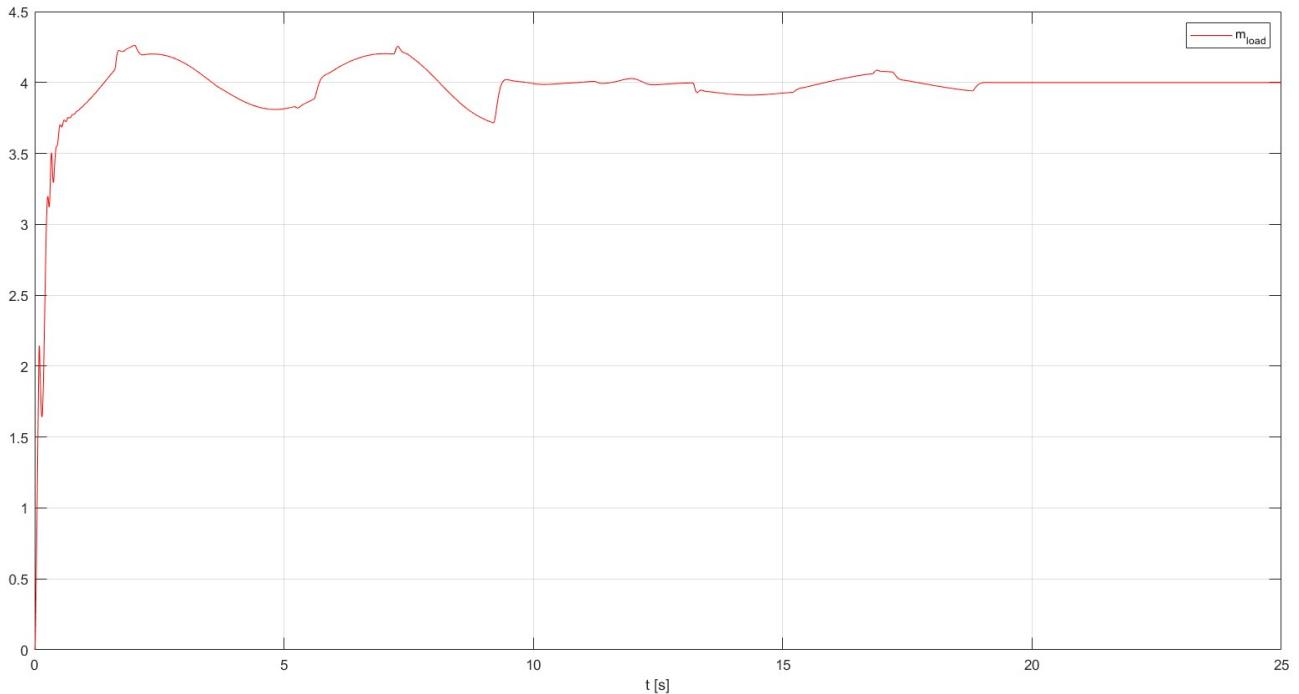


Fig. 5.3.7: Andamento del parametro m_{load} con algoritmo di controllo adattativo.

Gli errori di posizione e velocità ai giunti, a valle di oscillazioni iniziali dovuti all'incompleta conoscenza dei reali parametri dinamici del modello, risultano simili a quelli ottenuti adottando un controllo robusto, così come anche le coppie di attuazione ai giunti. Il parametro inizialmente non tarato m_{load} tende a convergere verso il suo valore reale di 4 kg: si tratta di un caso particolare, in cui la mancanza di conoscenza è relativa ad un unico parametro. Diversamente, la stima dei parametri non sarebbe coincisa con il loro vero valore, ma ciò non avrebbe influito significativamente sul controllo dell'errore a zero per posizione e velocità ai giunti, che sarebbe rimasto l'obiettivo di questo controllo.

5.4. Controllo a dinamica inversa nello spazio operativo

Gli schemi di controllo visti finora necessitavano di riferimenti nello spazio dei giunti. Il controllo a dinamica inversa che ci si appresta ad analizzare opera direttamente nello spazio operativo. Ciò è possibile adottando uno schema estremamente simile a quello dell'algoritmo CLIK del secondo ordine (figura 4.4.1), con variazioni minime.

L'ingresso di controllo è scelto pari a:

$$u = B(q)y + n(q, \dot{q}) \quad (5.4.1)$$

ove:

$$y = J_A^{-1}(q)(\ddot{x}_d - J_A(q, \dot{q})\dot{q} + K_D \dot{\tilde{x}} + K_P \tilde{x}) \quad (5.4.2)$$

In questo modo si giunge al sistema di errore:

$$\ddot{\tilde{x}} + K_D \dot{\tilde{x}} + K_P \tilde{x} = 0 \quad (5.4.3)$$

convergente a zero asintoticamente per K_D e K_P definite positive nell'ipotesi di conoscenza completa del modello. Tale ipotesi è tuttavia molto forte: una conoscenza inesatta del modello renderebbe impossibile giungere alla 5.4.3. Il controllo così come presentato risulta di conseguenza poco robusto: è necessario aggiungere un'azione integrale sulla variabile \tilde{x} .

Per il caso in esame, le matrici K_P e K_D sono state scelte come alle (4.4.3) e (4.4.4), e K_I è stata scelta pari a K_P . Si è assunta ignota la massa di carico in punta, con $m_{load} = 0$, di modo da dimostrare le potenzialità dell'azione integrale.

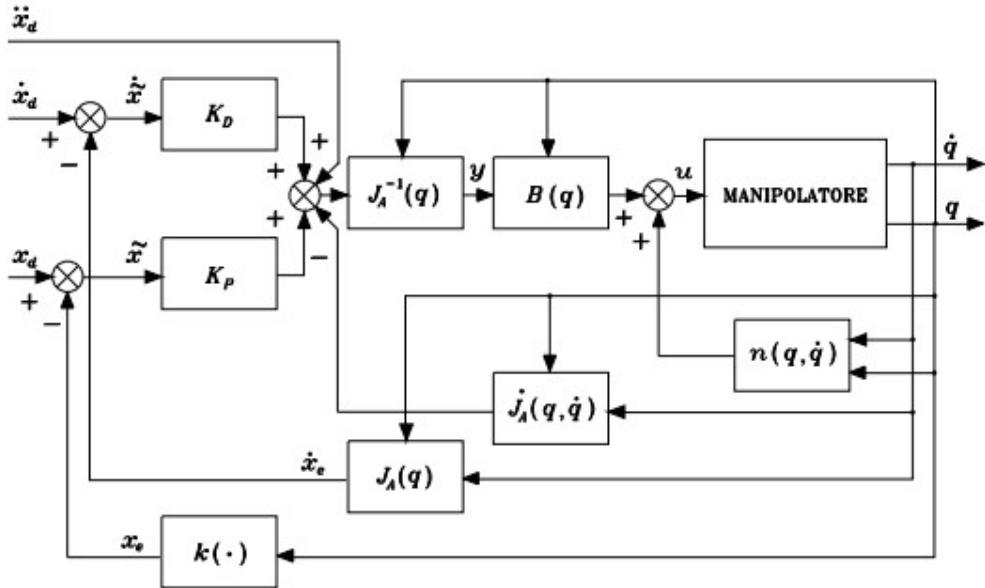


Fig. 5.4.1: Schema di principio dell'algoritmo di controllo a dinamica inversa nello spazio operativo.

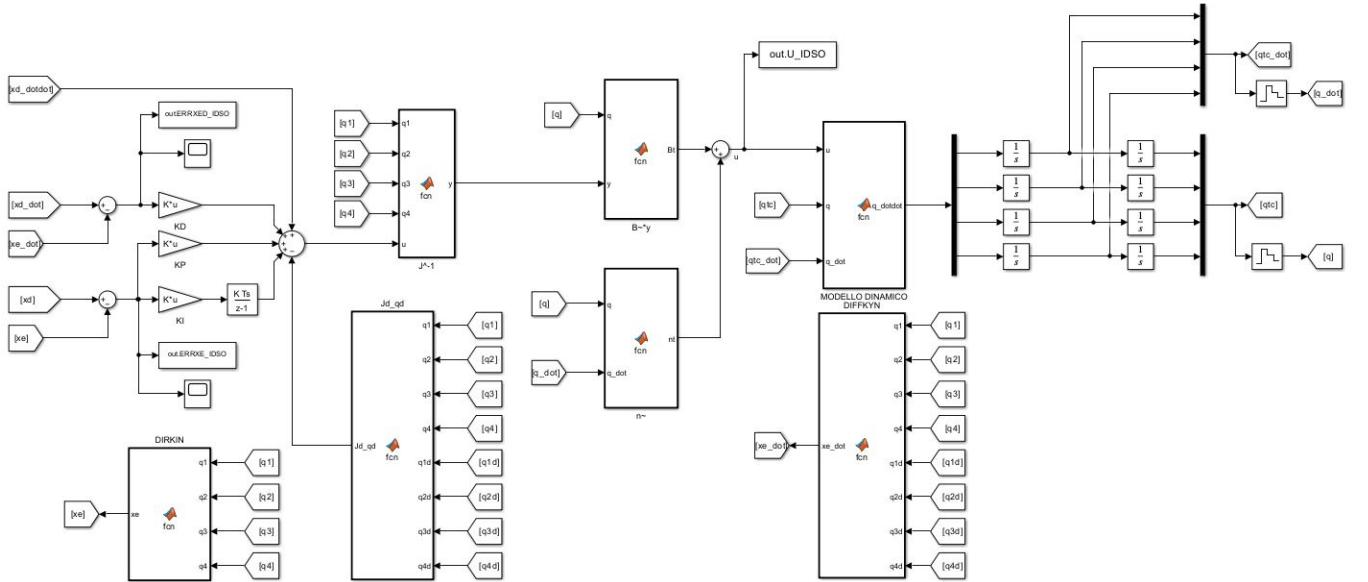


Fig. 5.4.2: Schema Simulink dell'algoritmo di controllo a dinamica inversa nello spazio operativo.

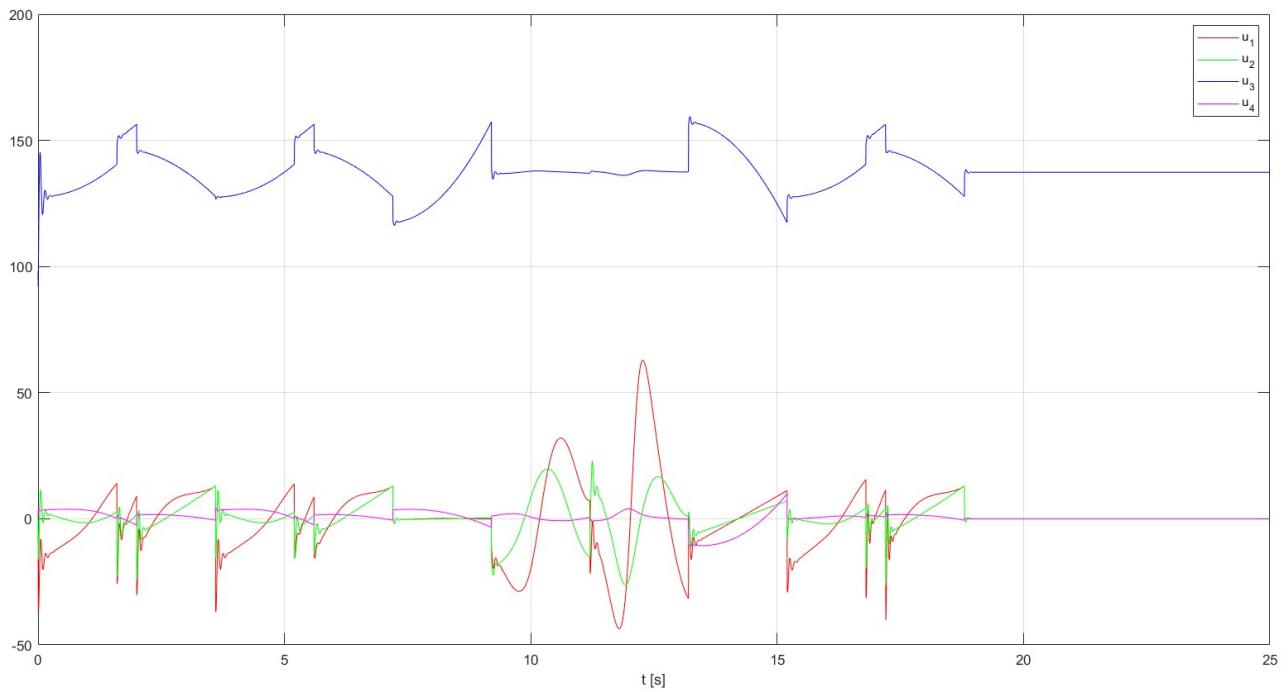


Fig. 5.4.3: Azioni di controllo ai giunti con algoritmo di controllo a dinamica inversa nello spazio operativo.

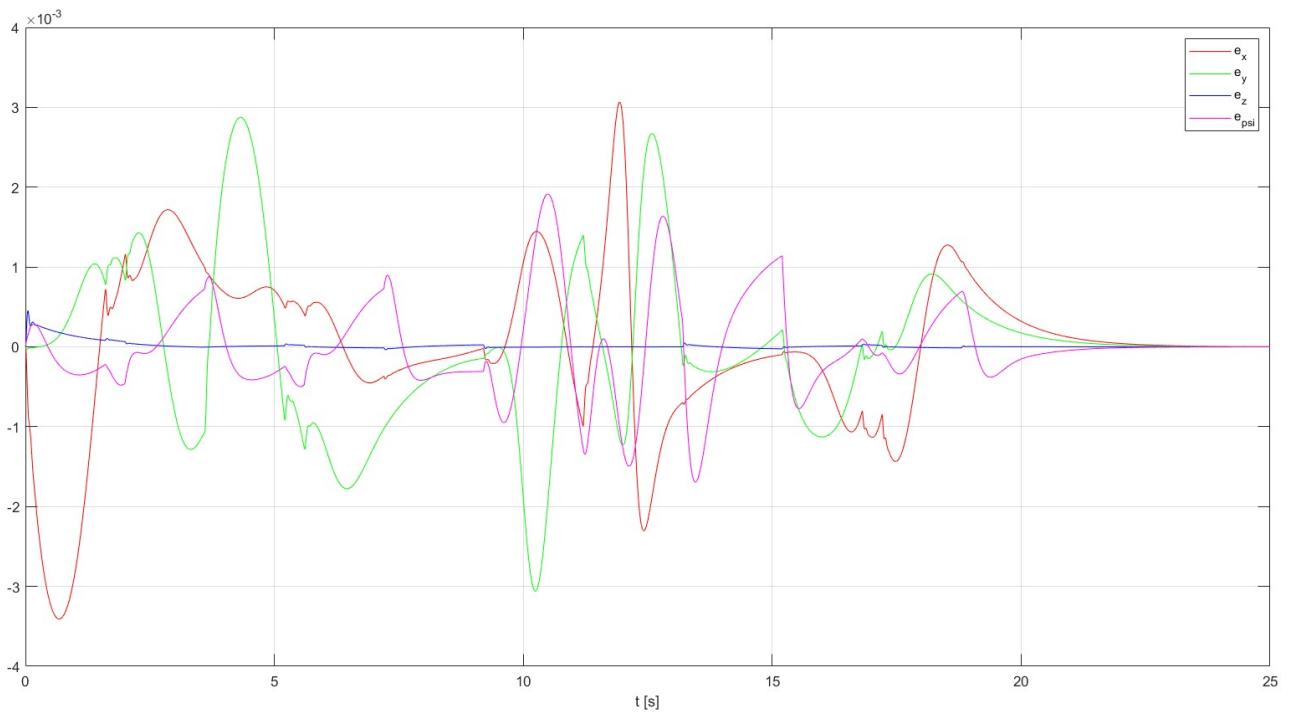


Fig. 5.4.4: Andamento dell'errore per la posa dell'EE con algoritmo di controllo a dinamica inversa nello spazio operativo. Si osservi la convergenza a zero dell'errore per effetto dell'azione integrale.

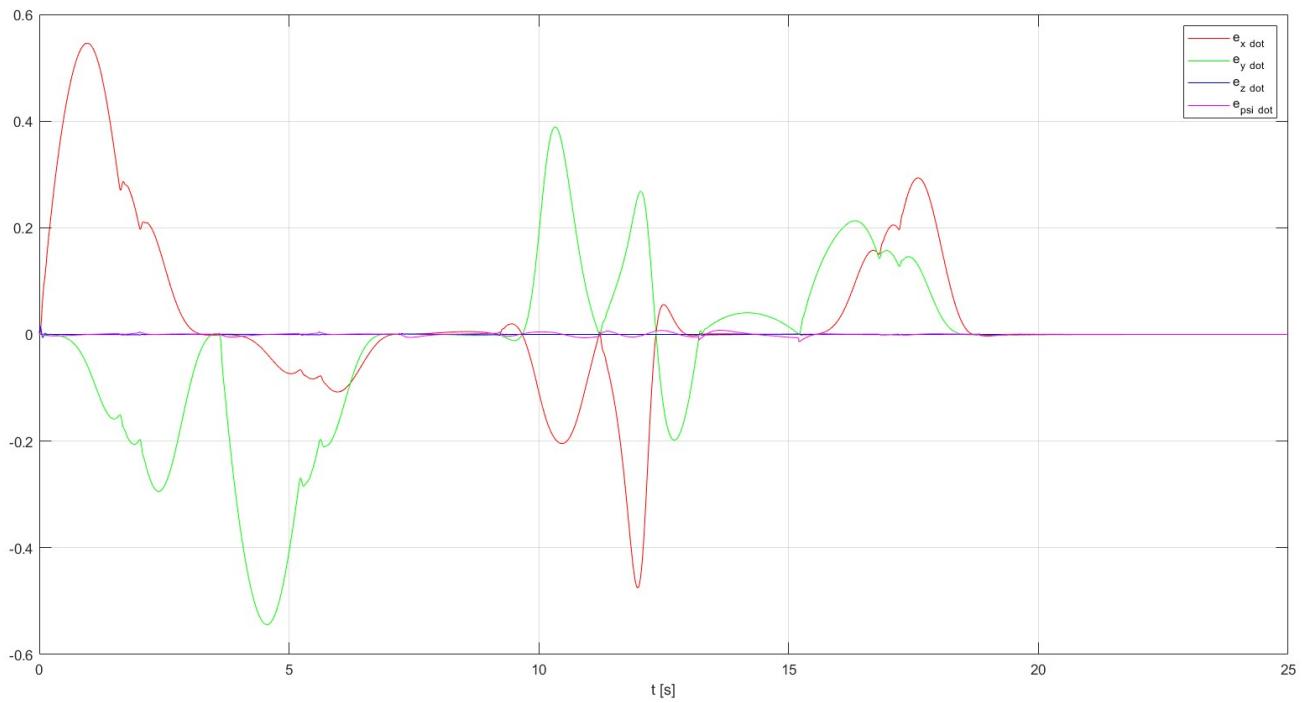


Fig. 5.4.5: Andamento dell'errore per la velocità dell'EE nello spazio operativo con algoritmo di controllo a dinamica inversa nello spazio operativo.

Le prestazioni di tale controllo sono soddisfacenti e non si discostano sensibilmente da quelle dei controlli visti in precedenza.