UNIVERSITÀ DI PISA

Corso di Laurea Magistrale in INGEGNERIA ROBOTICA E DELL'AUTOMAZIONE
Robotica - Controllo dei robot

Controllo Dinamico per inseguimento di traiettoria di un manipolatore UR10e

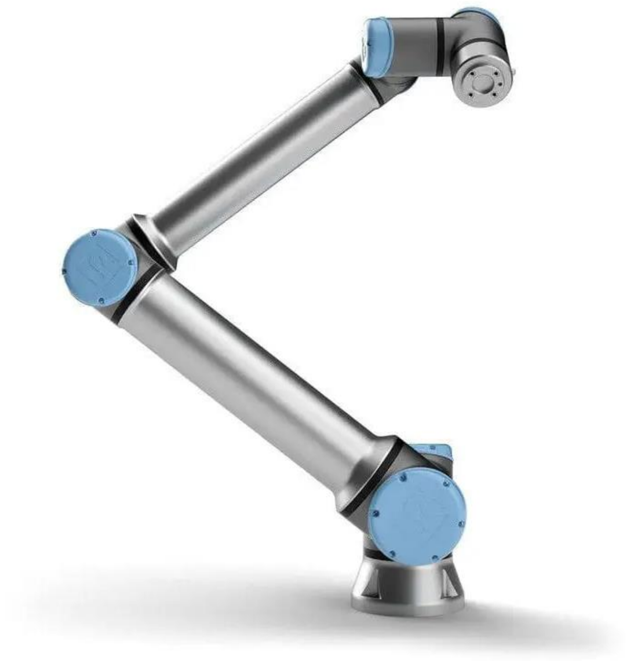
Docenti:
Antonio Bicchi
Giorgio Grioli

Supervisori del Progetto:
Lucia Pallottino
Paolo Salaris
Alessandro Palleschi

Studente:
Marco Borraccino

Contenuti

- UR10e: Caratterizzazione Cinematica e Dinamica
- Descrizione della traiettoria
- Manipolatore UR10e
  - Controllore PD con compensazione di gravità
  - Controllore Computed Torque
  - Controllore Backstepping (solo posizione)
  - Controllore Backstepping (posizione e orientazione)
- Manipolare RRR (primi tre link dell'UR10e)
  - Controllore PD con compensazione di gravità
  - Controllore Computed Torque
  - Controllore Backstepping
  - Controllore Adaptive Computed Torque
  - Controllore Adaptive Backstepping



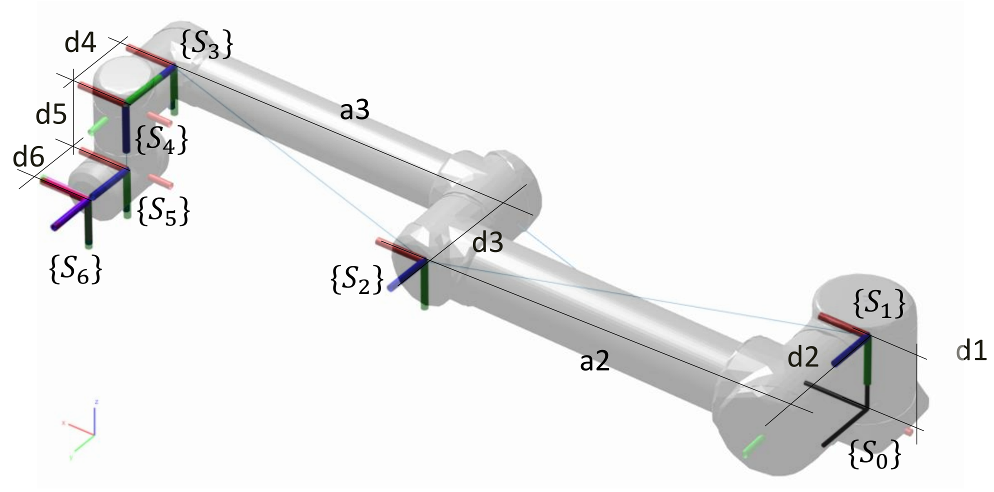
UR10e

Caratterizzazione Cinematica e Dinamica

Parametri DH

Parametri geometrici per parametrizzazione di Denavit – Hartemberg:

DH param.	$a[m]$	$\alpha[rad]$	$d[m]$	$\theta[rad]$
Joint 1	0	$-\pi/2$	0,1273	q_1
Joint 2	0,6120	0	0,2209	q_2
Joint 3	0,5723	0	0,1719	q_3
Joint 4	0	$-\pi/2$	0,1149	q_4
Joint 5	0	$\pi/2$	0,1157	q_5
Joint 6	0	0	0,0922	q_6

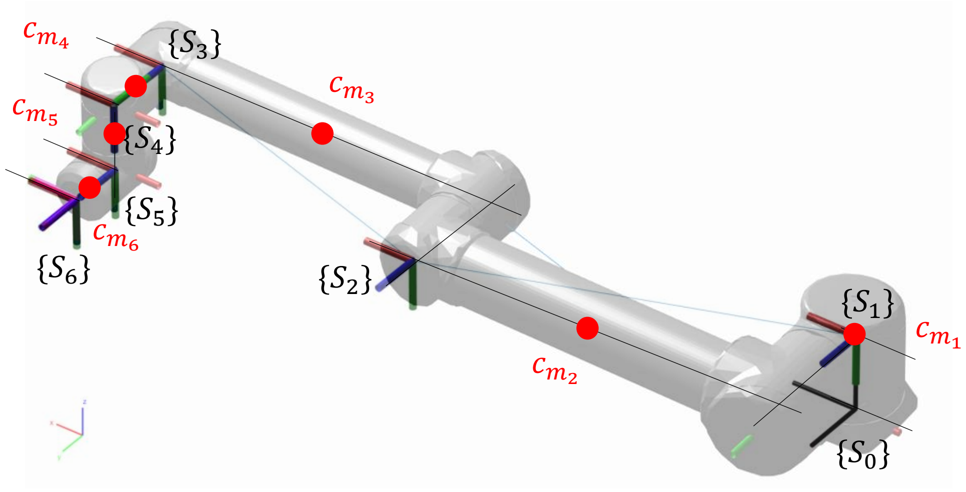


UR10e

Caratterizzazione Cinematica e Dinamica

Parametri dinamici

Dinamica	Mass $[kg]$	Centre of mass $[kg]$
Link 1	7,778	$[0 \quad 0 \quad 0]$
Link 2	12,93	$[-0{,}3060 \quad 0 \quad 0]$
Link 3	3,87	$[-0{,}28615 \quad 0 \quad 0]$
Link 4	1,96	$[0 \quad 0{,}1719 \quad 0]$
Link 5	1,96	$[0 \quad -0{,}1149 \quad 0]$
Link 6	0,202	$[0 \quad 0 \quad 0{,}0922]$



Le matrici dei momenti d’inerzia sono prese diagonali per semplicità, approssimando i corpi rigidi dei link come dei cilindri.

UR10e

Caratterizzazione Cinematica e Dinamica

Cinematica diretta e Jacobiano geometrico

A partire dalla parametrizzazione D-H si può calcolare la matrice omogenea che lega la posa dell'e-e in riferimento alla terna base, rispetto a tutti gli angoli di giunto.

$$q = (q_1, q_2, q_3, q_4, q_5, q_6)^T$$

$$\boldsymbol{T} = T_{06}(q) = T_{01}(q_1)T_{12}(q_2)T_{23}(q_3)T_{34}(q_4)T_{45}(q_5)T_{56}(q_6) = \begin{bmatrix} \boldsymbol{R_{06}} & \boldsymbol{d_{06}} \\ \boldsymbol{0^T} & \boldsymbol{1} \end{bmatrix}$$

Similmente si caratterizza il Jacobiano geometrico che lega le velocità dei giunti al twist dell'e-e.

$$J_{06}(q) = [\, J_{06}^1 \quad \dots \quad J_{06}^6 \,]$$

$$J_{06}^i = \begin{bmatrix} k_{i-1} \times (O_{i-1}O_6) \\ k_{i-1} \end{bmatrix} \qquad i = 1, \dots, 6$$

$$\longrightarrow \qquad \boldsymbol{\dot{\xi}_{06}} = \begin{bmatrix} \boldsymbol{v_{06}} \\ \boldsymbol{\omega_6} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{\dot{q}}$$

UR10e

Caratterizzazione Cinematica e Dinamica

Dinamica inversa

La Dinamica inversa invece propone di calcolare le coppie necessarie da dare al sistema per ottenere un movimento specificato in termini di accelerazioni $\ddot{q}(t)$, velocità $\dot{q}(t)$ e posizioni $q(t)$.

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}$$

$B(q)$ Matrice di inerzia

C$(q, \dot{q})$ Matrice delle forze centrifughe

$G(q)$ Matrice gravitazionale

Calcolate in simbolico tramite l'ausilio della libreria (*Wolfram Mathematica*):

ScrewCalculusPro [1]

[1] M. Gabiccini, "ScrewCalculus: a Mathematica Package for Robotics," DIMNP, University of Pisa," http://www2.ing.unipi.it/ »d11181/Personal/Downloads.htm, 2008.

Descrizione traiettoria

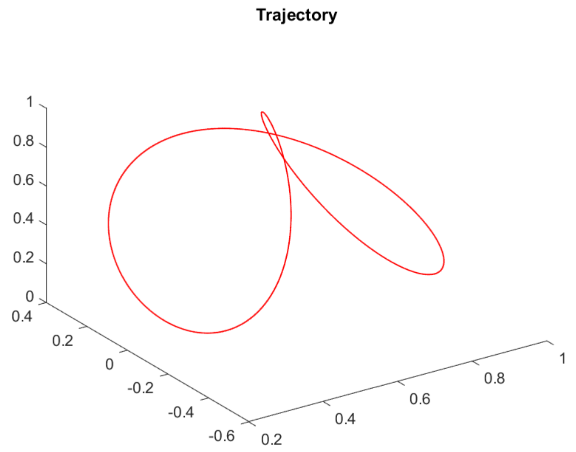
Con lo scopo di esplorare tutta la dinamica del sistema, si è scelta una traiettoria tridimensionale ispirandosi alle curve di *Lissajous* nello spazio.

$$x_{des} = \left[r_0 \sin\left(\frac{t}{2k}\right), \quad r_1 \sin\left(\frac{t}{k}\right), \quad r_3 \cos\left(\frac{t}{k}\right) \right] + p_0$$

$$p_0 = [0{,}6 \quad 0 \quad 0{,}5]$$

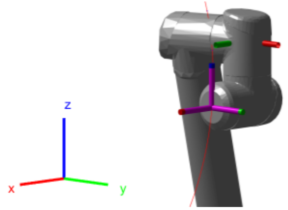
$parametro\ velocit$à: $k > 0$ (consigliato tra $5 - 30$)

$raggi\ r_0, r_1, r_2$



L'orientazione dell'end-effector è stata scelta coincidente con il *world frame* (comunque modificabile), codificata in un vettore di tre elementi secondo una parametrizzazione di Eulero XYZ.

$$\Phi = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \Phi_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



Manipolatore UR10e

Controllore PD con compensazione di gravità

Controllo più semplice possibile, ignora la conoscenza delle matrici dinamiche B e C del manipolatore e introduce solo una compensazione delle forze di gravità.

$$\boldsymbol{\tau} = \boldsymbol{K}_v \dot{\boldsymbol{e}} + \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{G}$$

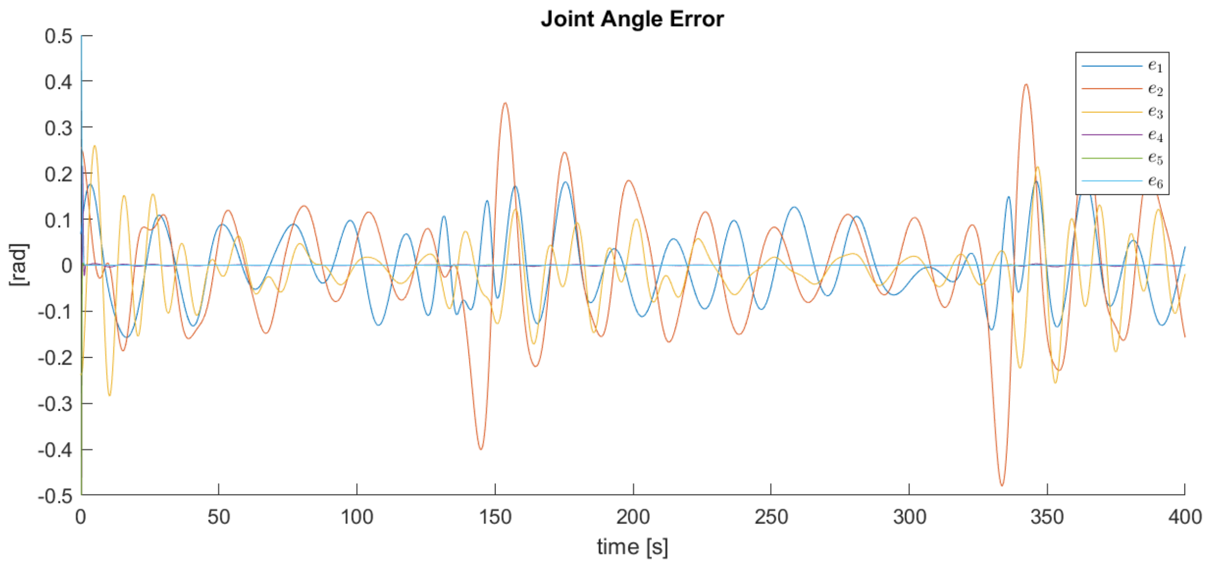
$$e(t) = q_d(t) - q(t)$$

$$K_v = \begin{bmatrix} 0{,}1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0{,}1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0{,}1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0{,}1 \end{bmatrix}$$

$$K_p = \begin{bmatrix} 0{,}5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0{,}5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0{,}5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0{,}5 \end{bmatrix}$$

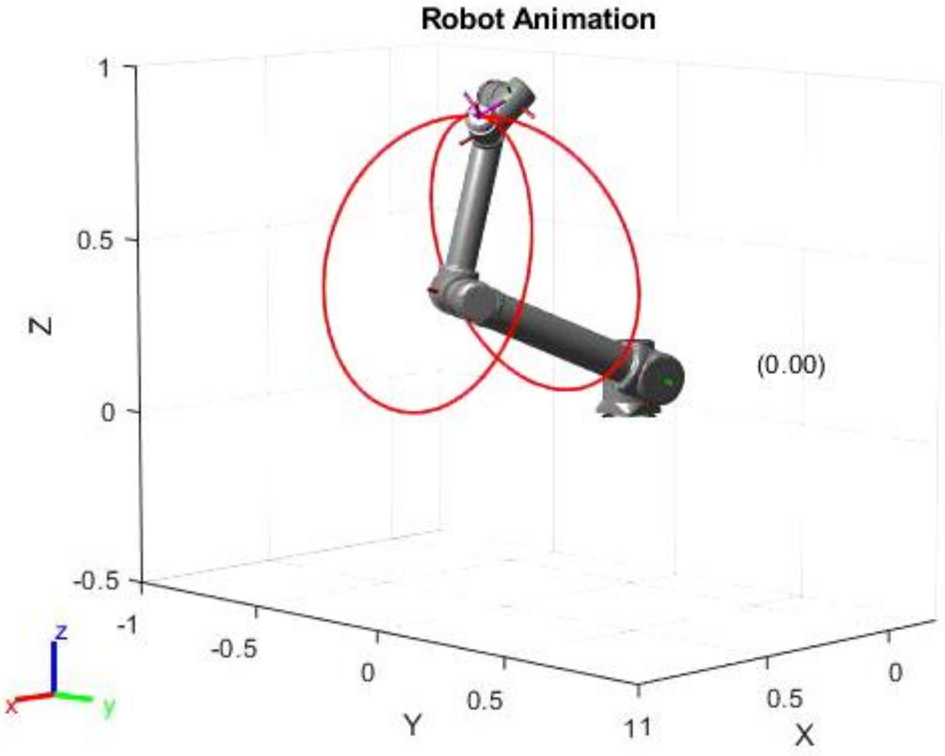
Manipolatore UR10e
Controllore PD con compensazione di gravità

$$\boldsymbol{\tau} = \boldsymbol{K}_v \dot{\boldsymbol{e}} + \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{G}$$



In un'applicazione reale questo errore è troppo significativo.

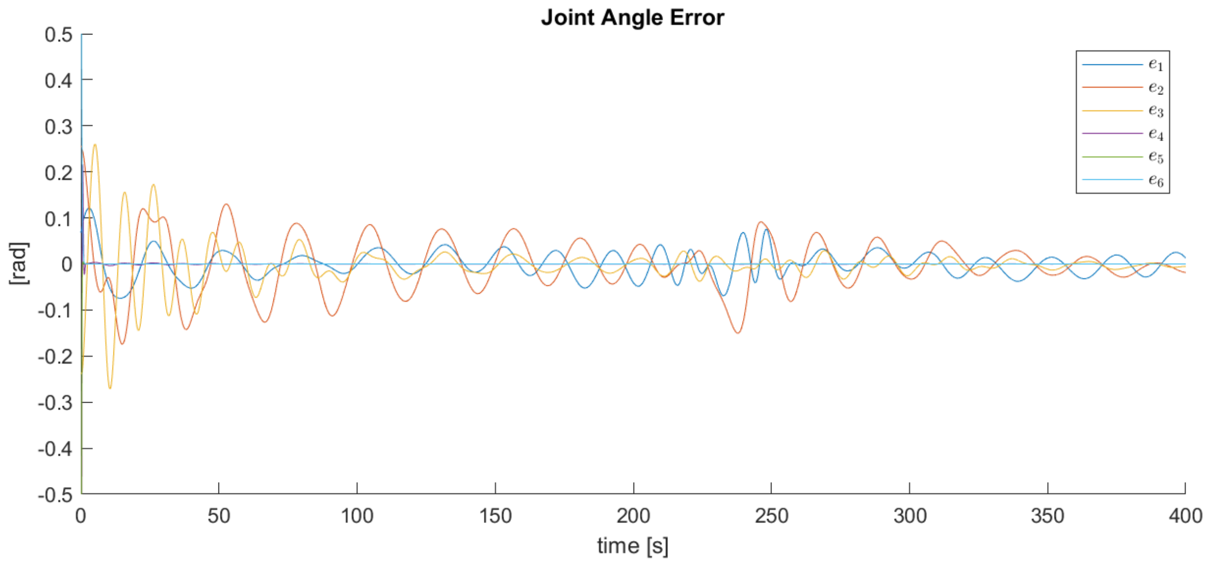
Traiettoria a *velocità standard* ($k = 15$)



Manipolatore UR10e

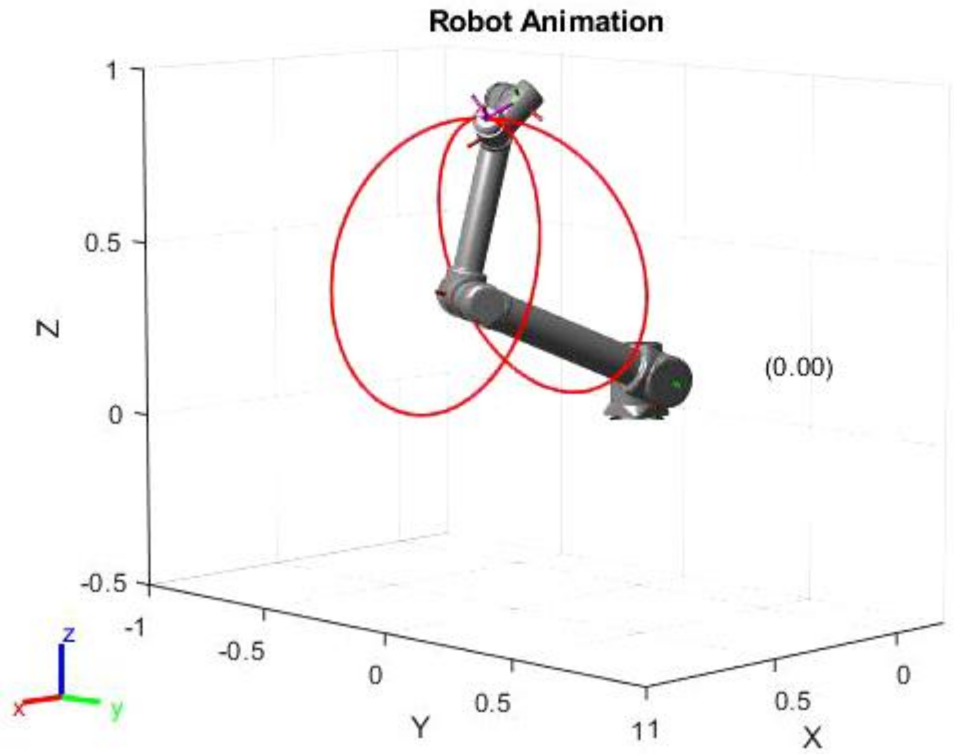
Controllore PD con compensazione di gravità

$$\boldsymbol{\tau} = \boldsymbol{K_v}\dot{\boldsymbol{e}} + \boldsymbol{K_p}\boldsymbol{e} + \boldsymbol{G}$$



Con una traiettoria più lenta l'errore tende a diminuire.

Traiettoria ***rallentata*** $(k = 25)$



Manipolatore UR10e
Controllore Computed Torque

Si basa sul principio del modello interno e inserisce la cancellazione della dinamica ipotizzando la sua **perfetta** conoscenza.

$$\boldsymbol{\tau} = \boldsymbol{B}\left(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{K}_{\boldsymbol{v}}\dot{\boldsymbol{e}} + \boldsymbol{K}_{\boldsymbol{p}}\boldsymbol{e}\right) + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G}$$

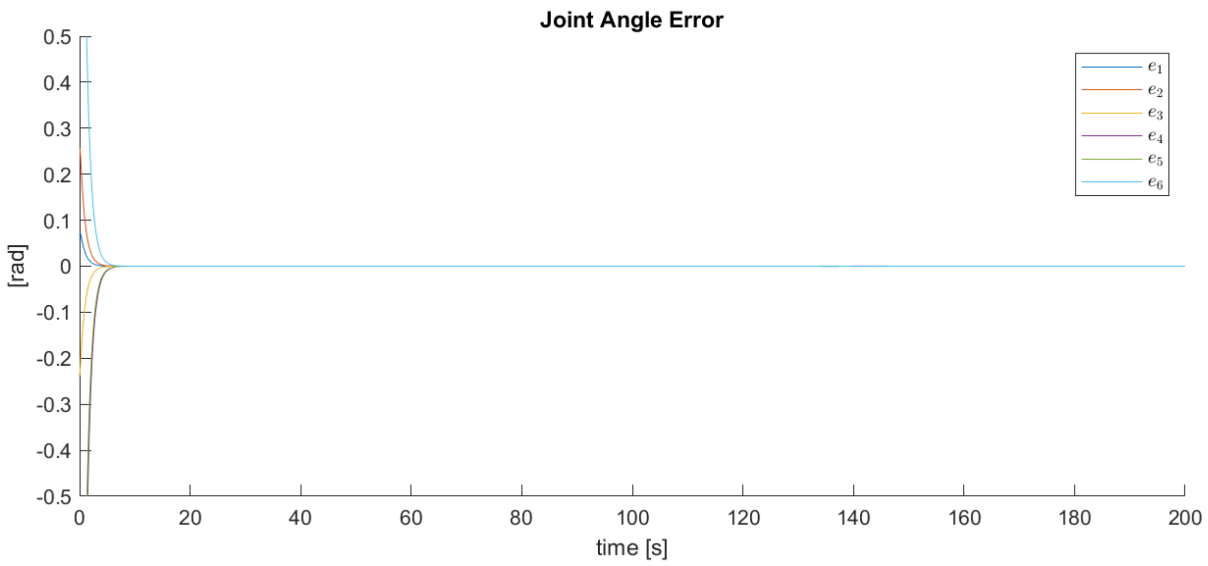
$$e(t) = q_d(t) - q(t)$$

$$K_v = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

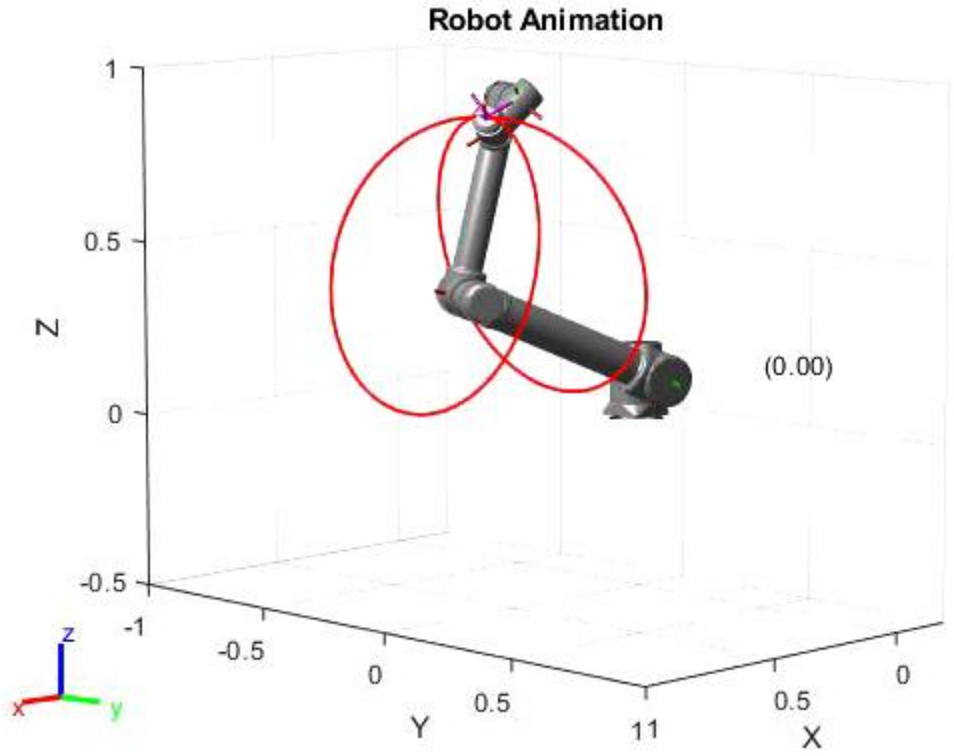
$$K_p = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

Manipolatore UR10e
Controllore Computed Torque

$$\boldsymbol{\tau} = \boldsymbol{B}\left(\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right) + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G}$$



Traiettoria a *velocità standard* ($k = 15$)



Manipolatore UR10e

Controllore Backstepping (solo posizione)

In questo caso semplificato, si controlla solo la posizione dell'end-effector.

$$\Lambda_\mathrm{p} = \begin{bmatrix} 0{,}25 & 0 & 0 \\ 0 & 0{,}25 & 0 \\ 0 & 0 & 0{,}25 \end{bmatrix} \quad K_s = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{bmatrix}$$

$$e_p = p_d - p = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\dot{q}_r = J_p^+ (\dot{p}_d + \Lambda_p e_p)$$

$$s = \dot{q}_r - \dot{q}$$

$$\boldsymbol{\tau} = \boldsymbol{B}\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{C}\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{G} - \boldsymbol{K}_{\boldsymbol{s}}\boldsymbol{s} - \boldsymbol{J}_{\boldsymbol{p}}^{\boldsymbol{T}}\boldsymbol{e}_{\boldsymbol{p}}$$

$$J \in \mathbb{R}^{6\times 6}$$
$$J_p, J_\omega \in \mathbb{R}^{3\times 6}$$

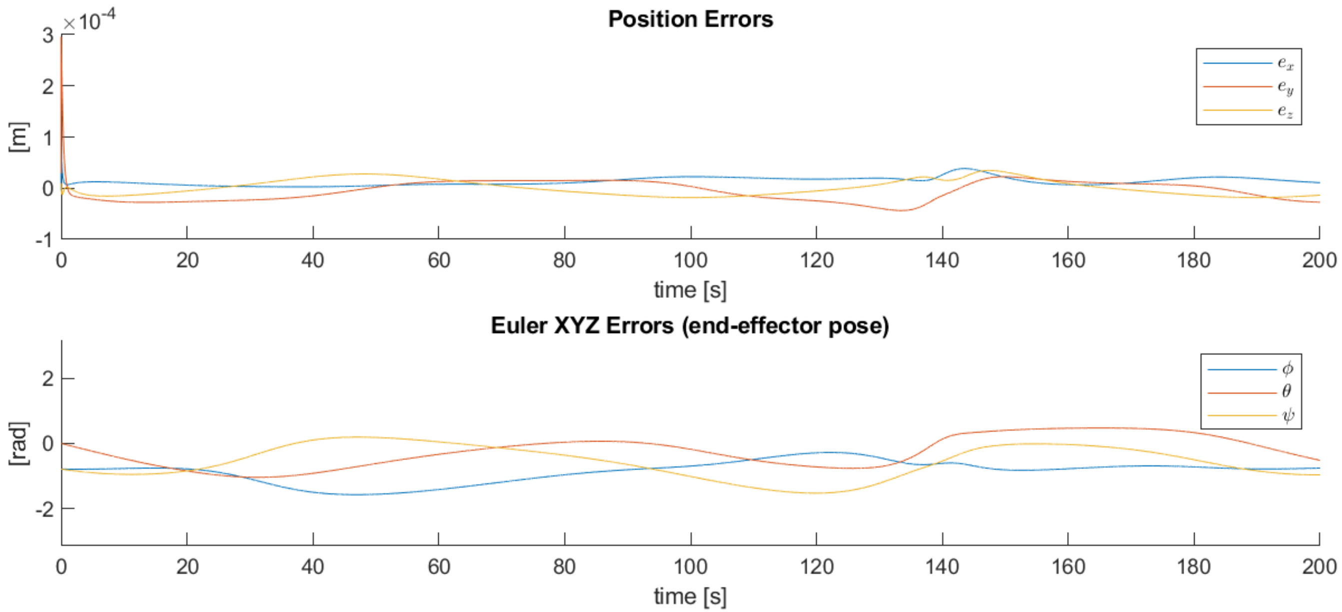
$$J = \begin{bmatrix} J_p \\ J_\omega \end{bmatrix}$$

Jacobiano di posizione

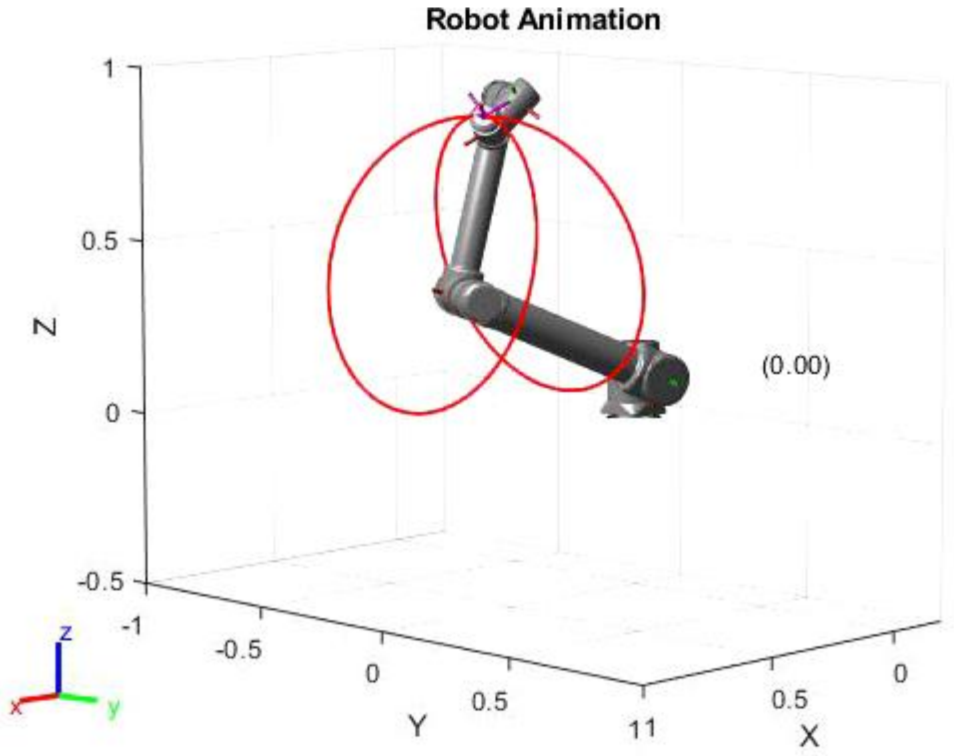
Jacobiano di orientazione

Manipolatore UR10e
Controllore Backstepping (solo posizione)

$$\boldsymbol{\tau} = \boldsymbol{B}\ddot{\boldsymbol{q}}_r + \boldsymbol{C}\dot{\boldsymbol{q}}_r + \boldsymbol{G} - \boldsymbol{K}_s\boldsymbol{s} - \boldsymbol{J}_p^T\boldsymbol{e}_p$$



Traiettoria a *velocità standard* ($k = 15$)



Naturalmente l'errore sull'orientazione non è trascurabile, dato che non la stiamo controllando in nessun modo.

Manipolatore UR10e
Controllore Backstepping Completo

Gestisco l'errore di orientazione passando alla descrizione dello stesso con i *quaternioni unitari*[2] , relativi alla posa desiderata e alla posa dell'ultimo frame del manipolatore.

$$\xi_d = \begin{bmatrix} x_d \\ y_d \\ z_d \\ \phi \\ \theta \\ \psi \end{bmatrix}$$

Dalla cinematica diretta $R_e(q)$ $\xrightarrow{\text{quaternioni}}$ $q_e = (q_{e1}, q_{e2}, q_{e3}, q_{e4}) = (\eta_e, \epsilon_e)$

$R_{XYZ}(\phi, \theta, \psi) = R_d$ $\longrightarrow$ $q_d = (q_{d1}, q_{d2}, q_{d3}, q_{d4}) = (\eta_d, \epsilon_d)$

$$e_p = p_d - p = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\dot{q}_r = J^{-1} \begin{pmatrix} \dot{p_d} + \Lambda_p e_p \\ \omega_d + \Lambda_o e_o \end{pmatrix}$$

$$\Lambda = \begin{bmatrix} \Lambda_p & 0 \\ 0 & \Lambda_o \end{bmatrix} = \begin{bmatrix} 0{,}25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0{,}25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0{,}25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0{,}25 \end{bmatrix}$$

$$\boldsymbol{e_o} = \boldsymbol{\eta_e}(\boldsymbol{q})\boldsymbol{\epsilon_d} - \boldsymbol{\eta_d}\boldsymbol{\epsilon_e}(\boldsymbol{q}) - \boldsymbol{S}(\boldsymbol{\epsilon_d})\boldsymbol{\epsilon_e}(\boldsymbol{q})$$

$$e = \begin{bmatrix} e_p \\ e_o \end{bmatrix}$$

$$s = \dot{q}_r - \dot{q}$$

$$K_s = \begin{bmatrix} 15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0{,}1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0{,}1 \end{bmatrix}$$

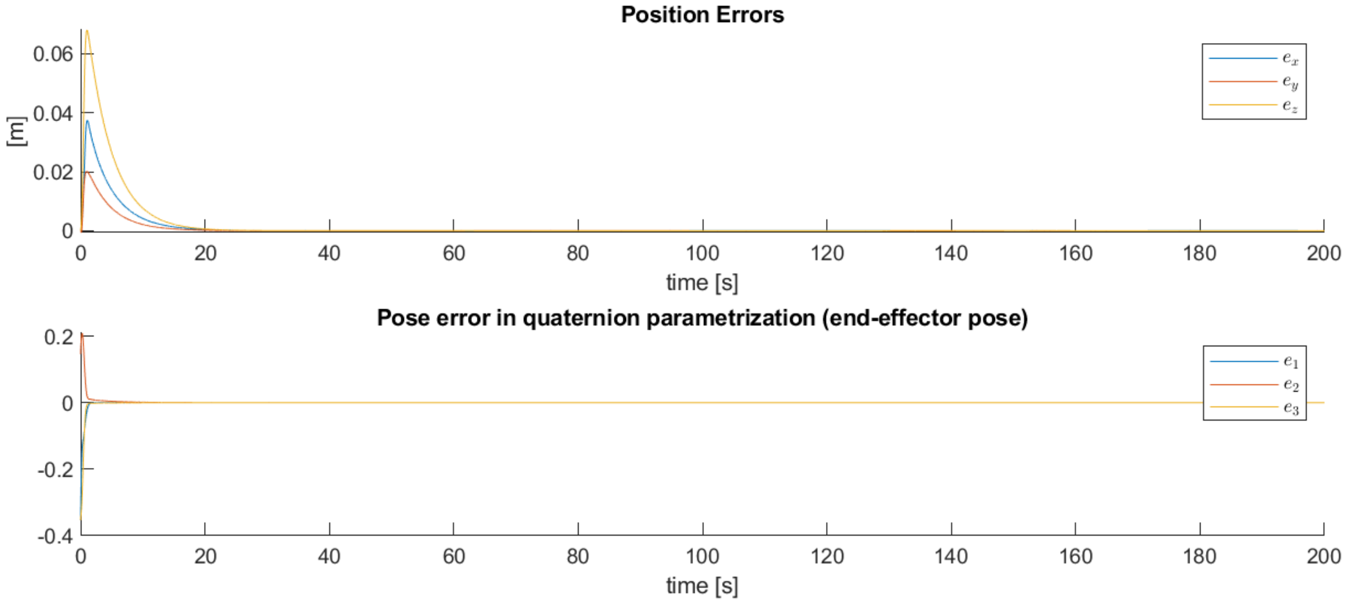
$$\boldsymbol{\tau} = \boldsymbol{B}\ddot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{C}\dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{G} - \boldsymbol{K_s s} - \boldsymbol{J^T e}$$

[2] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, «Robotics Modelling, Planning and Control»

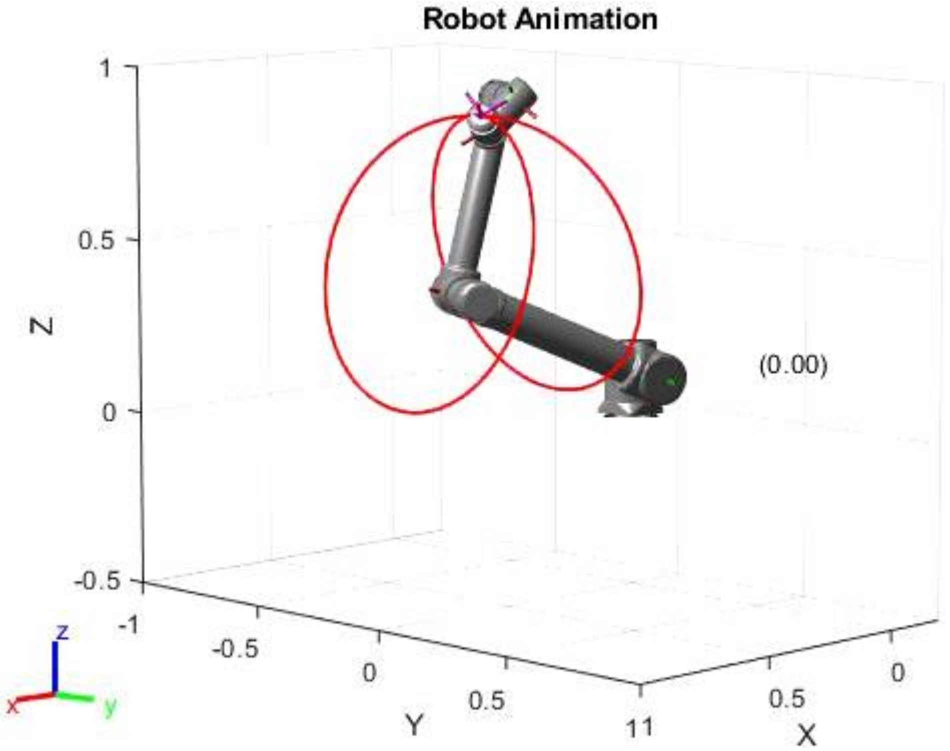
Manipolatore UR10e

Controllore Backstepping Completo

$$\boldsymbol{\tau} = \boldsymbol{B}\ddot{\boldsymbol{q}}_r + \boldsymbol{C}\dot{\boldsymbol{q}}_r + \boldsymbol{G} - \boldsymbol{K}_s\boldsymbol{s} - \boldsymbol{J}^T\boldsymbol{e}$$

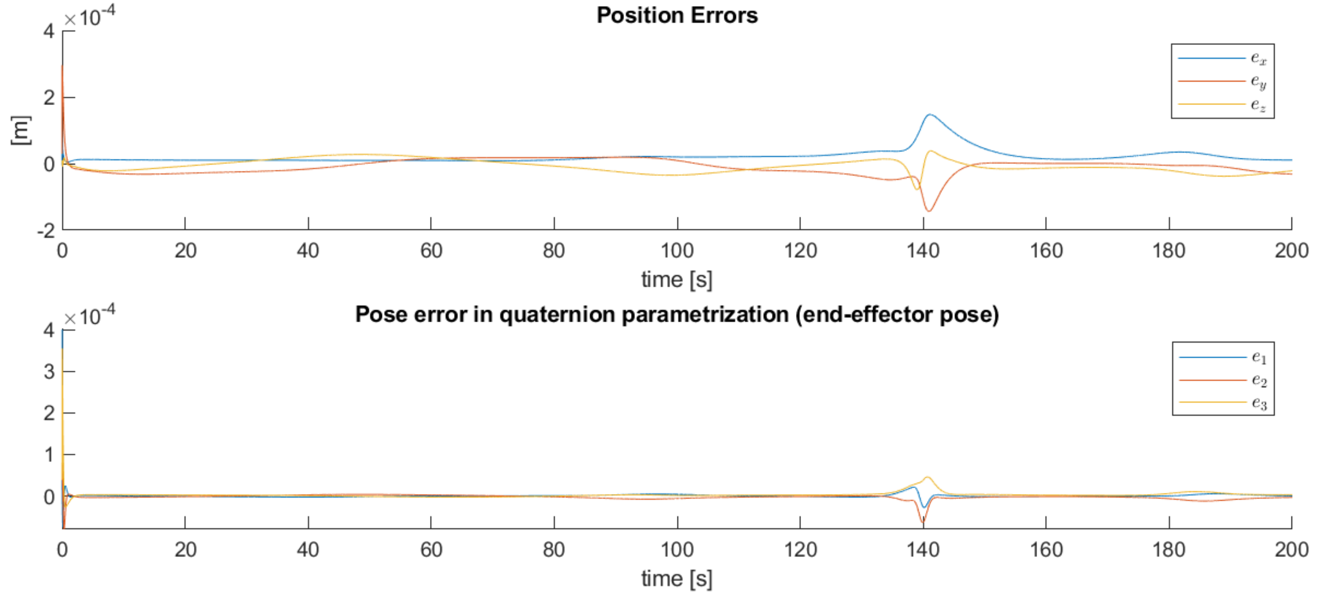


Traiettoria a *velocità standard* ($k = 15$)

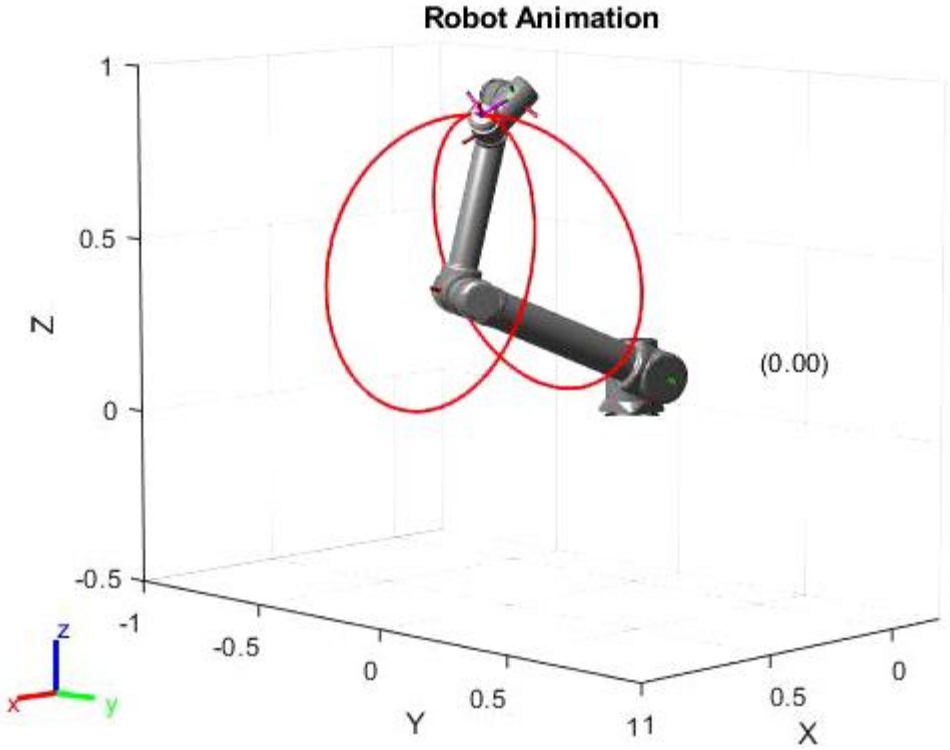


Manipolatore UR10e
Controllore Backstepping Completo

Orientazione desiderata (angoli Eulero XYZ) $\Phi_0 = [\pi/4 \quad 0 \quad \pi/4]$



Traiettoria a *velocità standard* ($k = 15$)

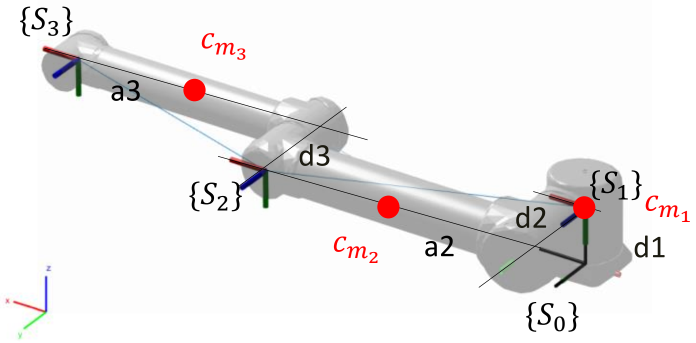


Manipolatore RRR
Caratterizzazione Cinematica e Dinamica

Creazione di un modello semplificato per i controllori adattivi.
Controllo **solo** la posizione del *wrist*.

DH param.	$a[m]$	$\alpha[rad]$	$d[m]$	$\theta[rad]$
Joint 1	0	$-\pi/2$	0,1273	q_1
Joint 2	0,6120	0	0,2209	q_2
Joint 3	0,5723	0	0,1719	q_3

Dinamica	Mass $[kg]$	Centre of mass $[kg]$
Link 1	7,778	$[0 \quad 0 \quad 0]$
Link 2	12,93	$[-0{,}3060 \quad 0 \quad 0]$
Link 3	3,87	$[-0{,}28615 \quad 0 \quad 0]$



Manipolatore RRR
Controllo Adattivo

L'idea è quella di partire dalla descrizione della dinamica del manipolatore e riportarla in forma lineare rispetto dei parametri ignoti $\hat{\pi}$:

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{Y}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q})\hat{\boldsymbol{\pi}}$$

Nel nostro caso abbiamo considerato ignoti solo i valori delle masse dei link, supponendo conosciuta e fissa la posizione dei centri di massa e la distribuzione delle masse stesse dei link.

Vettore dei parametri:

$$\hat{\boldsymbol{\pi}} = [\boldsymbol{m_1} \quad \boldsymbol{m_2} \quad \boldsymbol{m_3}]^{\boldsymbol{T}}$$

Esempio di tensore d'inerzia con l'approssimazione di *link cilindrico*:

$$I^1 = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} \frac{1}{12} m_1(h_1^2 + 3r_1^2) & 0 & 0 \\ 0 & \frac{1}{2} m_1(3r_1^2) & 0 \\ 0 & 0 & \frac{1}{12} m_1(h_1^2 + 3r_1^2) \end{bmatrix}$$

Manipolatore RRR
Controllo Adaptive Computed Torque

Con la notazione usata per il controllo computed torque il sistema diventa:

$$e = q_d - q \qquad \tilde{\pi} = \pi_d - \hat{\pi}$$

$$\tau = \widehat{M}(q)\left(\ddot{q}_d + K_v\dot{e} + K_p e\right) + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q)$$

$$\ddot{e} + K_v\dot{e} + K_p e = \widehat{M}(q)^{-1}Y(\ddot{q},\dot{q},q)\tilde{\pi}$$

$$\downarrow$$

$$\begin{cases} \begin{bmatrix}\dot{e}\\ \ddot{e}\end{bmatrix} = A\begin{bmatrix}e\\ \dot{e}\end{bmatrix} + B\widehat{M}(q)^{-1}Y(\ddot{q},\dot{q},q)\tilde{\pi} \\ \dot{\tilde{\pi}} = -R^{-1}Y^T(\ddot{q},\dot{q},q)\widehat{M}(q)^{-T}B^T P\begin{bmatrix}e\\ \dot{e}\end{bmatrix} \end{cases}$$

$$K_p = \begin{bmatrix}50 & 0 & 0\\ 0 & 200 & 0\\ 0 & 0 & 100\end{bmatrix} \qquad K_v = \begin{bmatrix}10 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 10\end{bmatrix}$$

$$R = I_{3\times3} \qquad B = \begin{bmatrix}0_{3\times3}\\ I_{3\times3}\end{bmatrix}$$

$$A = \begin{bmatrix}0_{3\times3} & I_{3\times3}\\ -Kp & -Kv\end{bmatrix} \qquad Q = I_{6\times6}$$

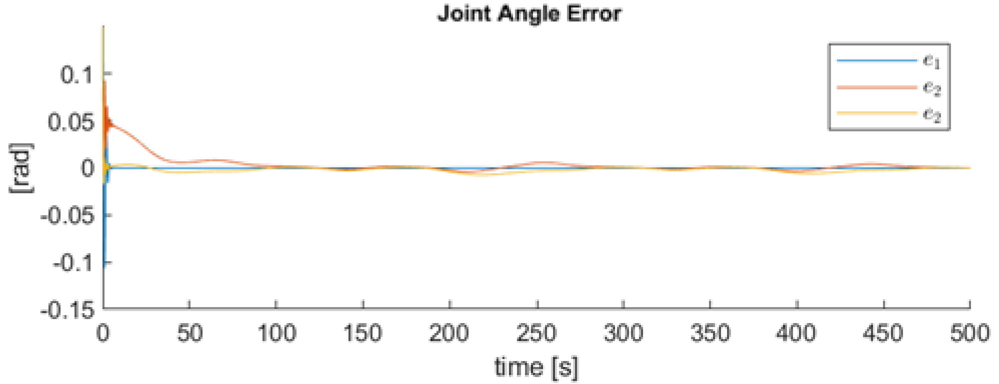
$$P \; tale \; che \; A^T P + PA = Q$$

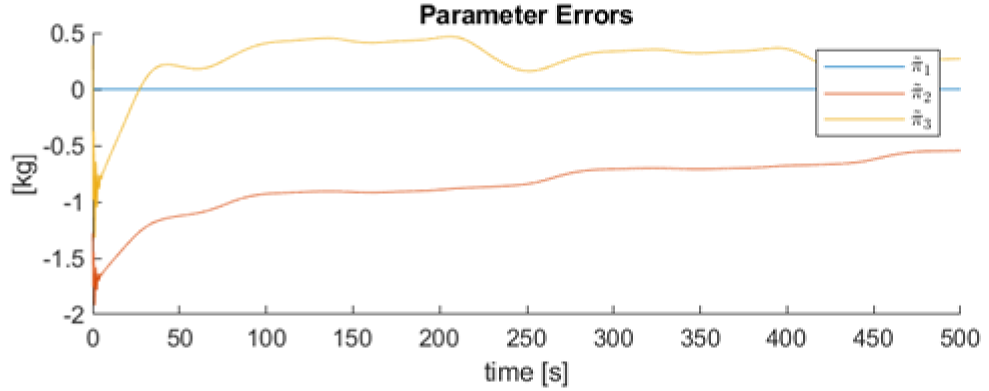
Il valori delle matrici K_p e K_v sono empirici.

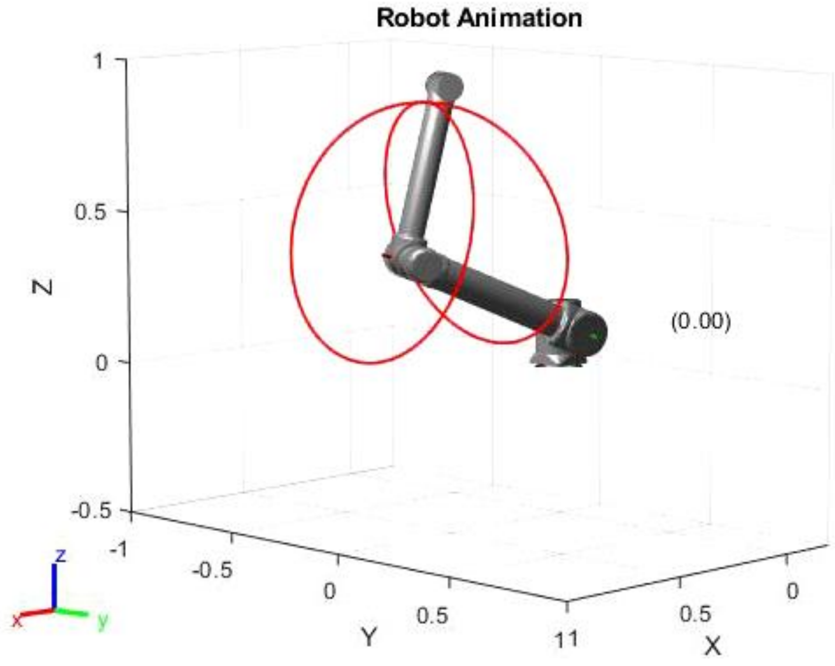
Manipolatore RRR
Controllo Adaptive Computer Torque

Errore iniziale sui parametri:

$$\widehat{m_1} = m_1$$
$$\widehat{m_2} = (1 + 10\%)m_2$$
$$\widehat{m_3} = (1 - 10\%)m_3$$







Lenta convergenza dell'errore dei parametri

Manipolatore RRR
Controllo Adaptive Backstepping

Similmente ricordando la notazione usata per il controllo *backstepping*, il sistema diventa:

$$e = \xi_d - \xi \qquad \dot{q}_r = J^{-1}(\dot{\xi}_d + \Lambda e)$$

$$s = \dot{q}_r - \dot{q}$$

$$\tau = \widehat{M}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{G}(q) - K_s s - J^T e$$

$$\tau = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_{\boldsymbol{r}},\ddot{\boldsymbol{q}}_{\boldsymbol{r}})\tilde{\pi} - K_s s - J^T e$$

$$\dot{\tilde{\pi}} = R^{-1}\boldsymbol{Y}^{\boldsymbol{T}}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_{\boldsymbol{r}},\ddot{\boldsymbol{q}}_{\boldsymbol{r}})s$$

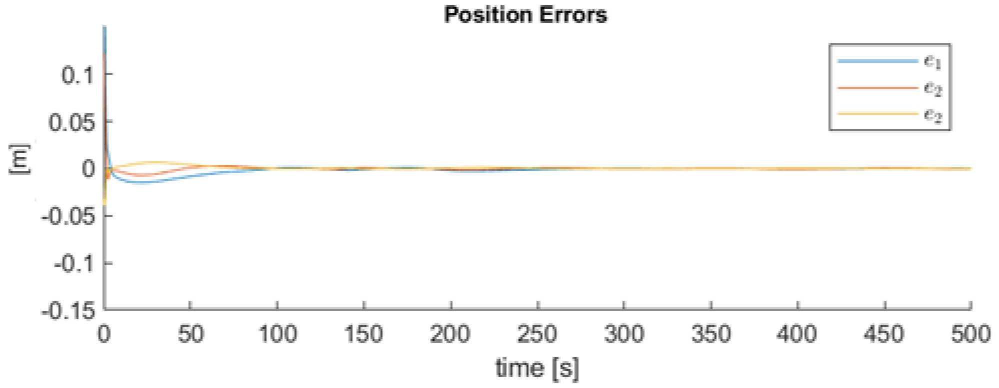
Il valori delle matrici Λ e K_v sono stati trovati empiricamente.

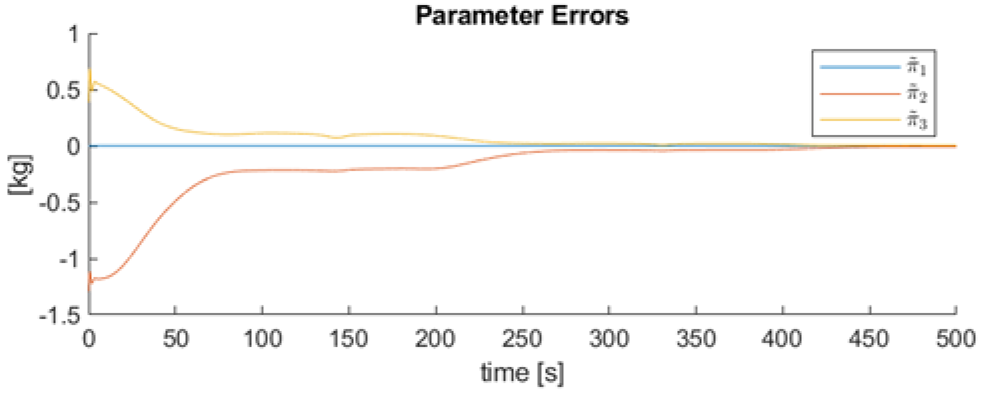
$$\Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad K_s = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{bmatrix}$$

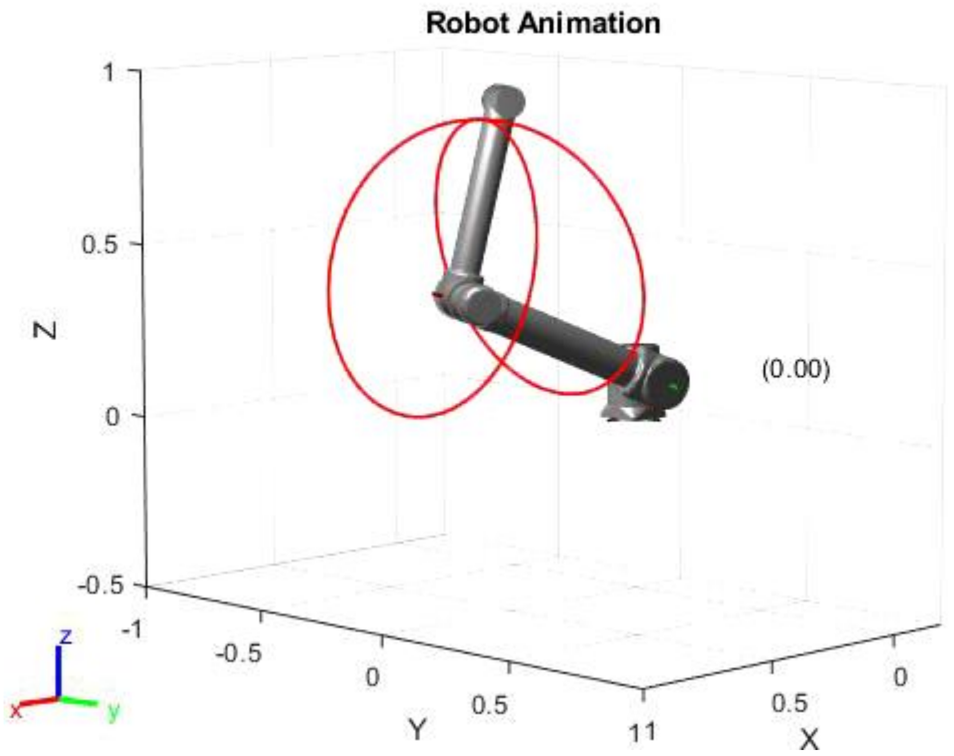
$$R = I_{3\times 3}$$

Manipolatore RRR
Controllo Adaptive Backstepping

Errore iniziale sui parametri:
$$\widehat{m_1} = m_1$$
$$\widehat{m_2} = (1 + 10\%)m_2$$
$$\widehat{m_3} = (1 - 10\%)m_3$$





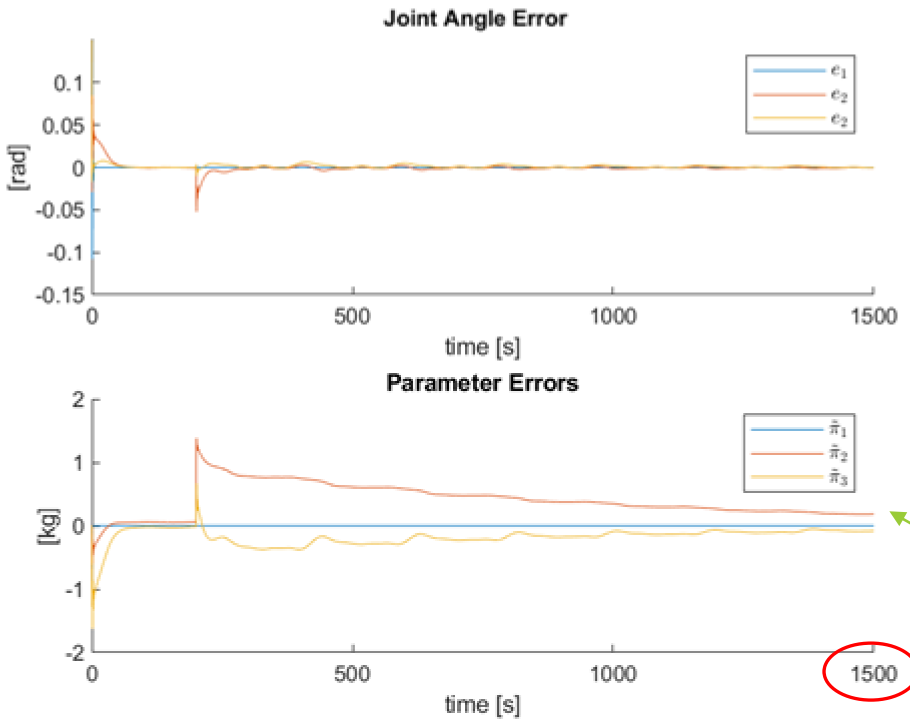


Manipolatore RRR

Controllo Ad. Computed Torque vs. Ad. Backstepping

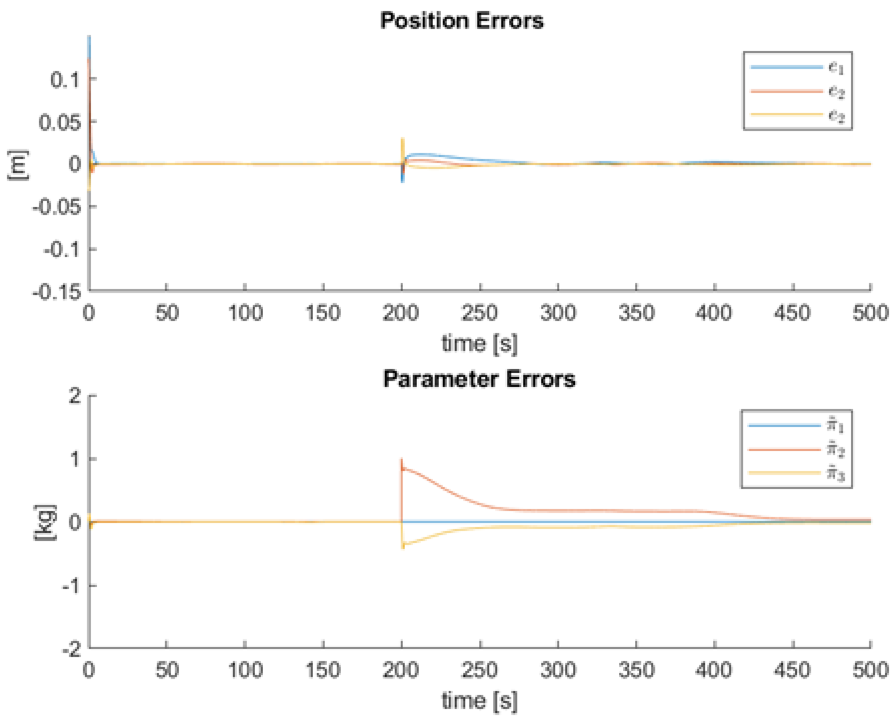
Ipotizzando noti i parametri al tempo iniziale con errore nullo, **al tempo $t = 200s$ su aggiunge $1\ kg$ a m_2.**

Adaptive Computed Torque



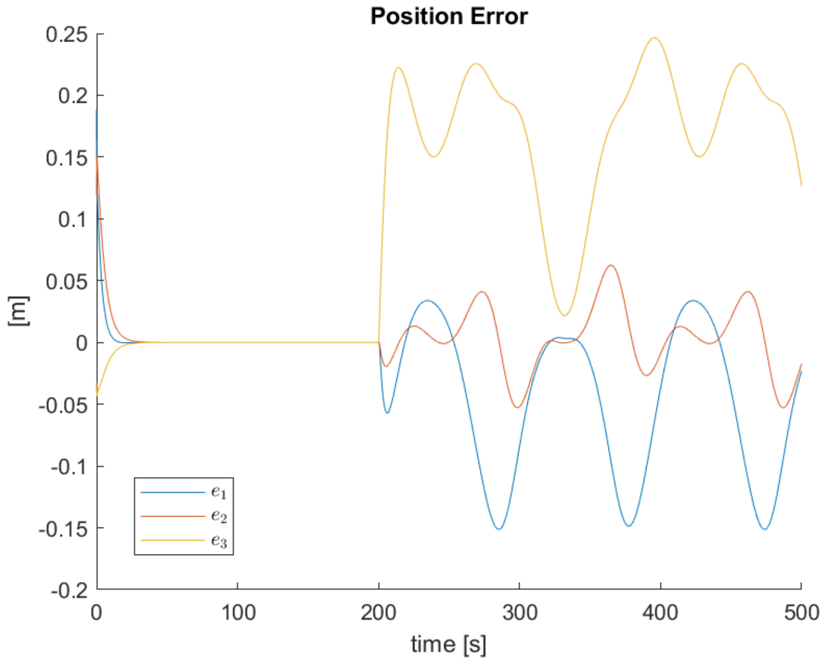
Converge più lentamente

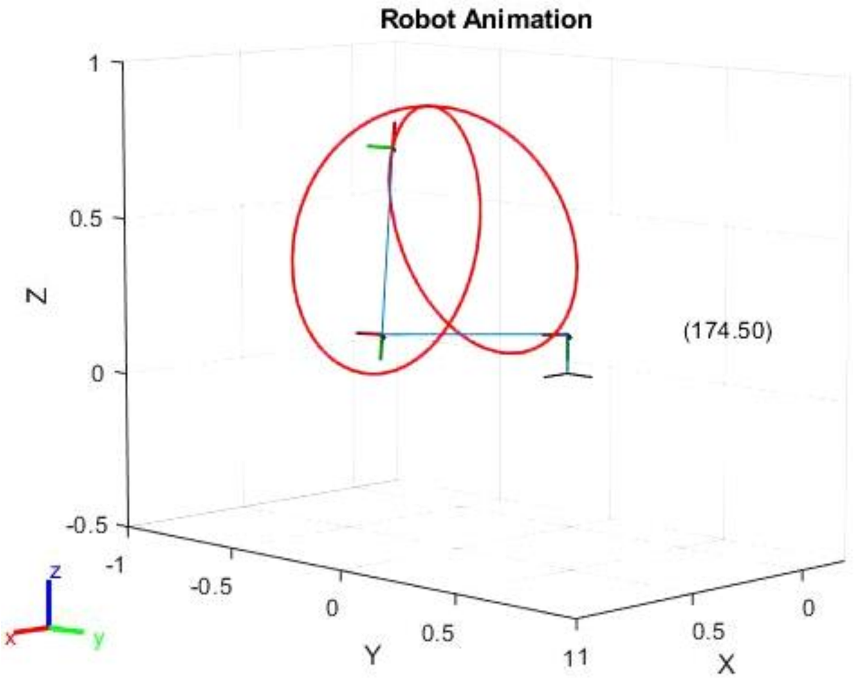
Adaptive Backstepping



Manipolatore RRR
Controllo Backstepping

Come si comporta un controllo non adattivo in questo caso? → Controllore Backstepping





All'errore si aggiunge una quantità dovuta alla differenza tra il modello vero e quello all'interno del controllore.
Con guadagni più alti sulla retroazione della posizione l'errore diminuisce.

Conclusioni

UR10e

I controllori Computed Torque e Backstepping risultano i più efficaci se la conoscenza della dinamica è esatta.

I tempi di convergenza all'errore nullo sono pressocché istantanei ad alti valori dei guadagni.

RRR

Il controllo Adaptive Backstepping risulta il più veloce e affidabile in termini di convergenza dei parametri dinamici da stimare.

Invece, l'entità dell'errore sulle variabili di giunto è confrontabile tra i due approcci.

Al variare della traiettoria e delle matrici dei guadagni, si possono ottenere risultati differenti.

Bibliografia

[1] M. Gabiccini, “ScrewCalculus: a Mathematica Package for Robotics,” DIMNP, University of Pisa,” http://www2.ing.unipi.it/ »d11181/Personal/Downloads.htm, 2008.

[2] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, «Robotics Modelling, Planning and Control»

[3] Appunti di Controllo dei Robot, A.Bicchi, P. Salaris, G. Grioli, M. Garabini, A. Settimi, A. Fagiolini et al., 2021

[4] Appunti di Meccanica dei Robot, M. Gabiccini, 2020-2021