CONTROLLO DI SISTEMI ROBOTICI

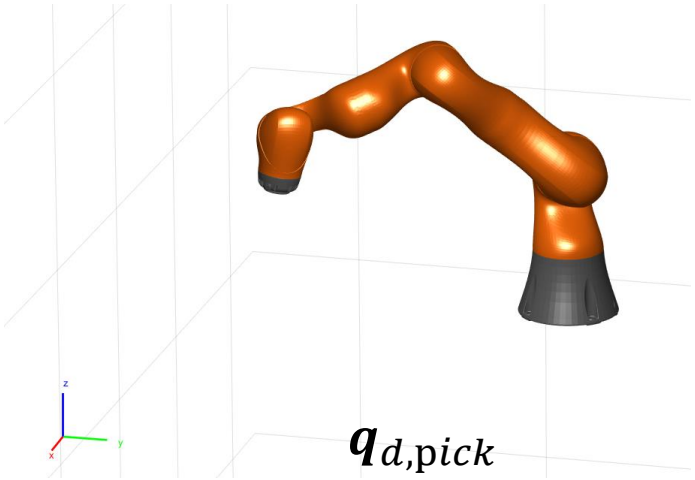
- Controllo di Robot Antropomorfi in ambiente Matlab/Simulink

Toolbox utilizzati

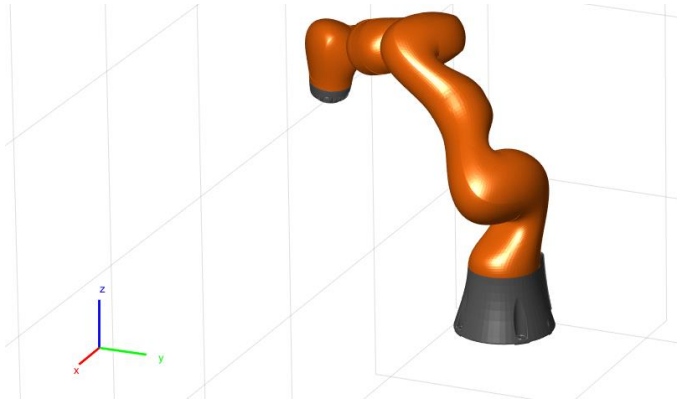
- Matlab Robotics System Toolbox
- Simulink 3D Animation

Robot Antropomorfo: Pick and Place

- Risolvere un problema di pick and place (regolazione) nello spazio di giunto per il manipolatore antropomorfo Kuka LBR 14 (robot ridondante 7 DoF)
- Tale problema può essere affrontato con la tecnica di controllo in Feedback Linearization PD + compensazione di gravità
- Il robot deve essere programmato per raggiungere la configurazione di picking $\boldsymbol{q}_{d,\mathrm{pick}}$ (dove avviene la presa dell'oggetto da movimentare) e in seguito la posizione di placing $\boldsymbol{q}_{d,\mathrm{place}}$ (dove avviene il rilascio)



$\boldsymbol{q}_{d,\mathrm{pick}}$



$\boldsymbol{q}_{d,\mathrm{place}}$

Modello dinamico del manipolatore

- Il Robotics System Toolbox di Matlab include le funzionalità per definire cinematica e dinamica di un manipolatore e alcuni modelli di Robot già definiti, tra i quali il manipolatore KUKA LBR 14
- Per importare il modello è necessario lanciare i seguenti comandi:

```
robot = importrobot('iiwa14.urdf');
robot.DataFormat = 'column';
robot.Gravity = [0;0;-9.81];
```

unified robot description format

specifica il vettore gravità

specifica il fatto che i vettori $\boldsymbol{q}, \dot{\boldsymbol{q}}$ … sono vettori colonna

- Nel Workspace di Matlab viene creata la variabile *robot,* di tipo *RigidBodyTree*

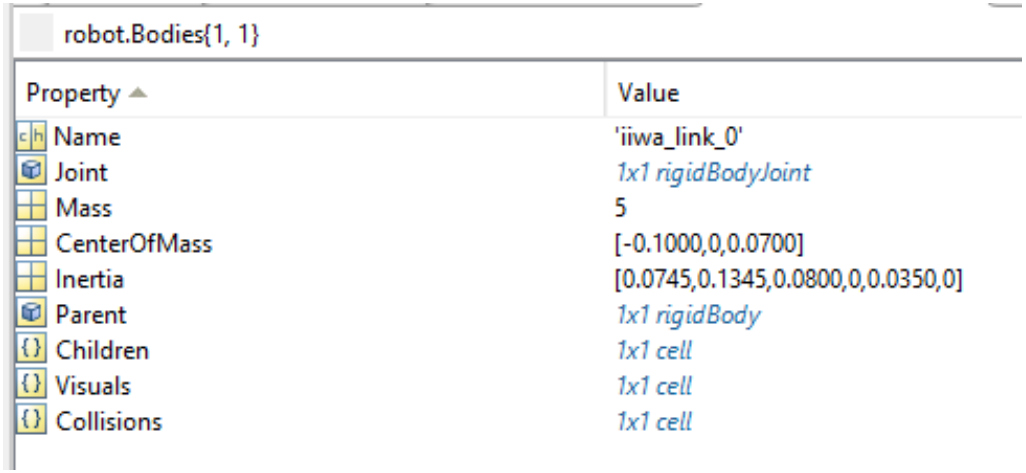
robot

1x1 rigidBodyTree

Property	Value
NumBodies	10
Bodies	*1x10 cell*
Base	*1x1 rigidBody*
BodyNames	*1x10 cell*
BaseName	'world'
Gravity	[0,0,-9.8100]
DataFormat	'column'

Modello dinamico del manipolatore

- Un RigidBodyTree è una catena cinematica di corpi rigidi (*Bodies*) a ognuno dei quali è associato un giunto, una posa, le proprietà inerziali (massa, baricentro, tensore d'inerzia) e una mesh 3D
- Tale variabile possiede dunque tutte le informazioni per calcolare il modello dinamico del manipolatore



Modello dinamico del manipolatore

- Il Robotics System Toolbox espone le funzioni che permettono di calcolare le varie componenti del modello dinamico del manipolatore, nella forma semplificata (con D,Fa = 0):

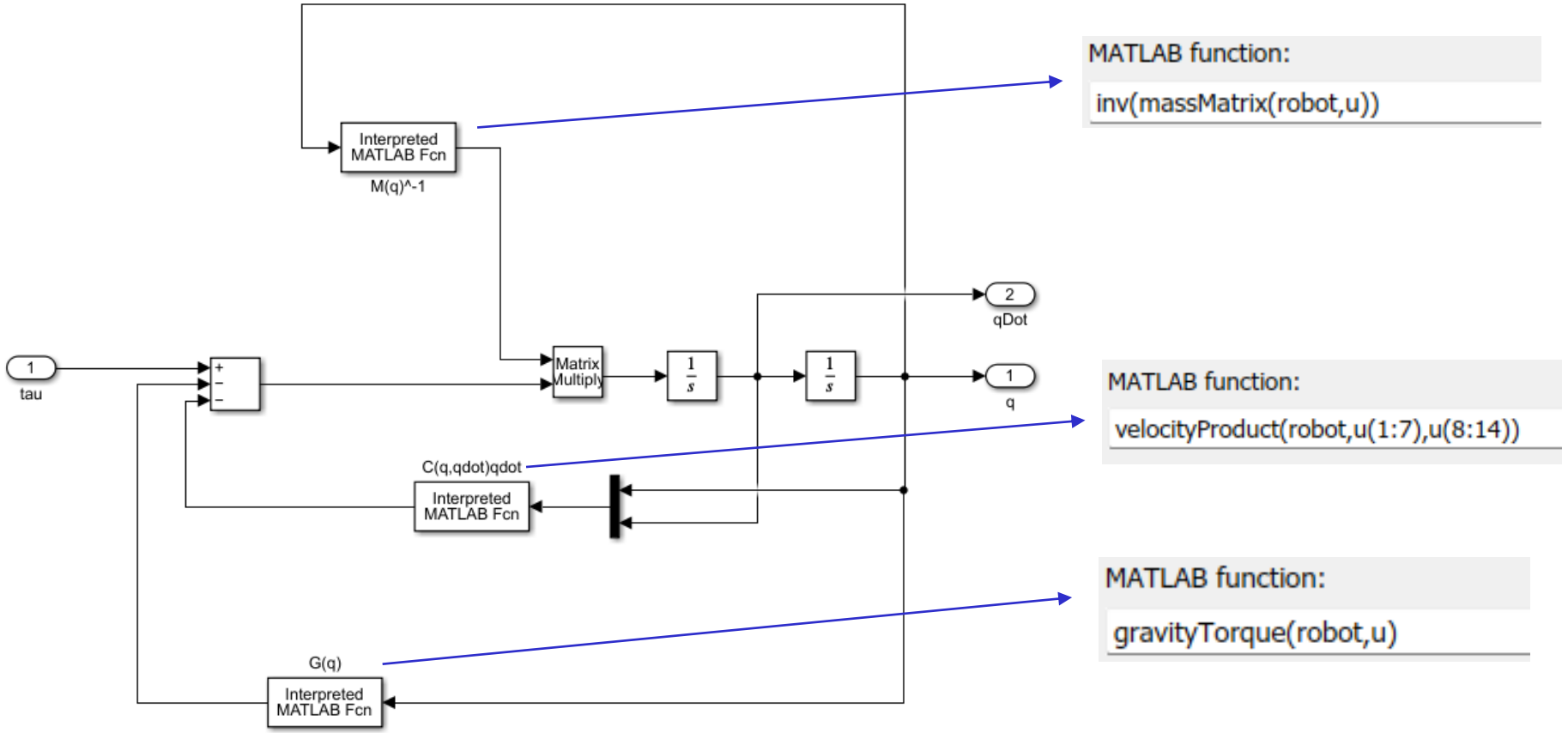
$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

- `M = massMatrix(robot,q)` calcola $\boldsymbol{M}(\boldsymbol{q})$
- `Cqdot = velocityProduct(robot,q,qdot)` calcola $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$
- `g = gravityTorque(robot,q)` calcola $\boldsymbol{g}(\boldsymbol{q})$

Script di inizializzazione

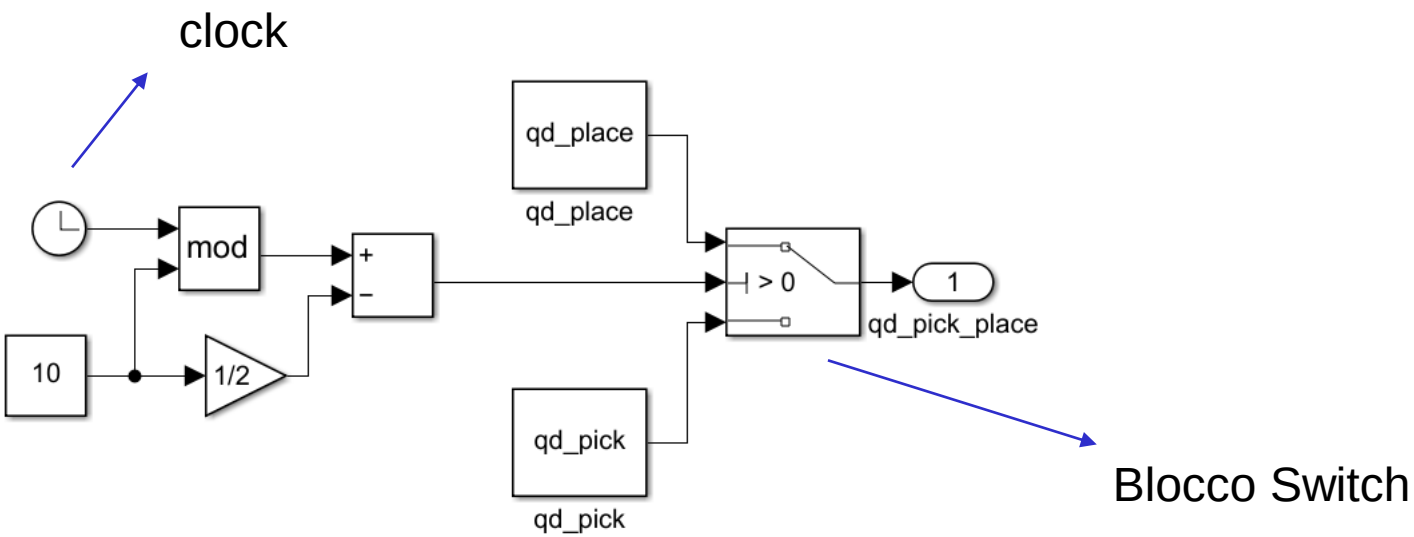
```
% import modello robot
robot = importrobot('iiwa14.urdf');
robot.DataFormat = 'column';
robot.Gravity = [0;0;-9.81];

% configurazione iniziale e target
q0 = [0;0;0;0;0;0;0];
qd_pick = [-60;50;0;-60;0;60;0]*pi/180;
qd_place = [-120;50;0;-60;0;60;0]*pi/180;

% parametri controllo
Kd = 50*eye(7);
Kp = 100*eye(7);
```

Modello Simulink del robot



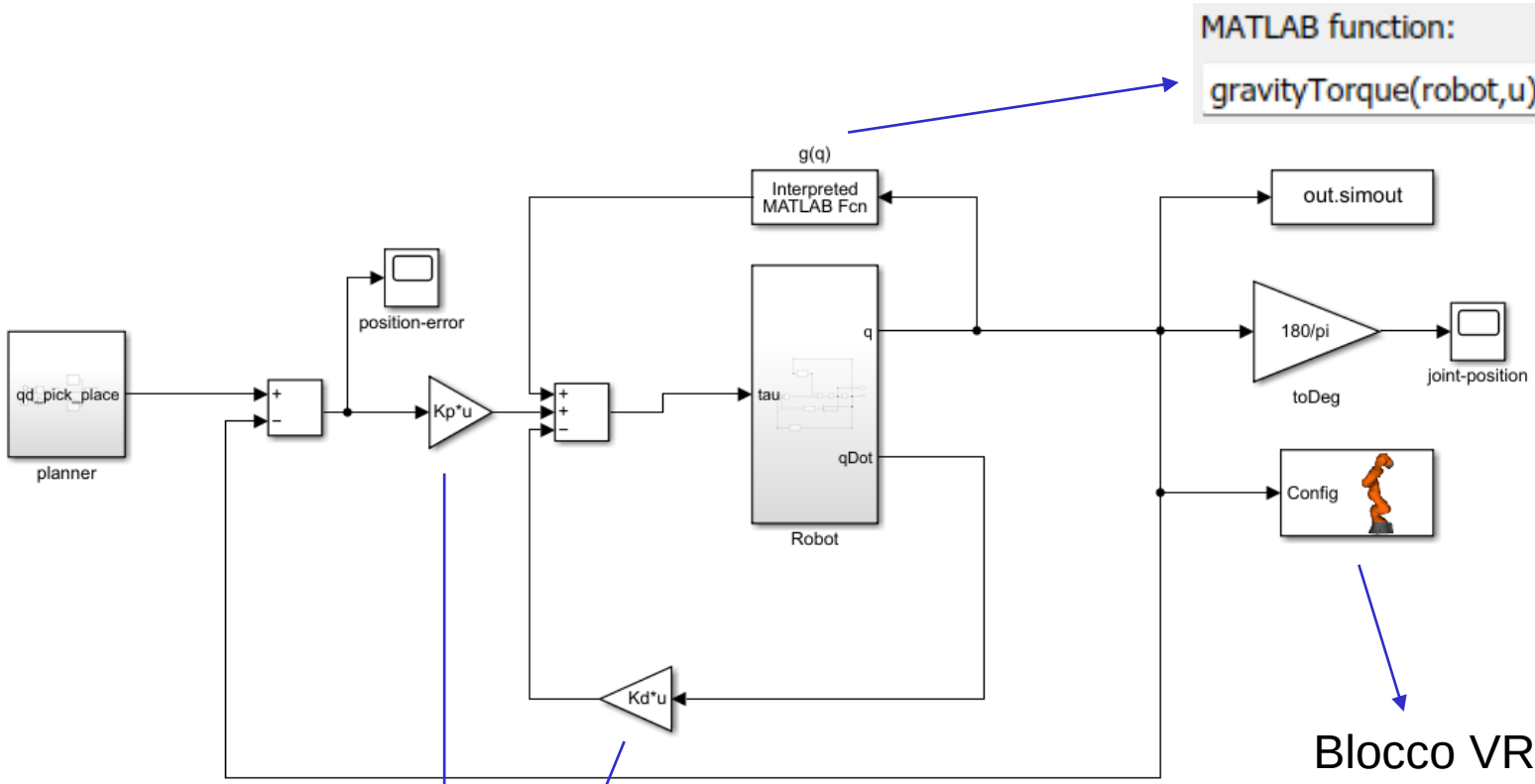
$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})(-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau})$$

Modello Simulink del pianificatore



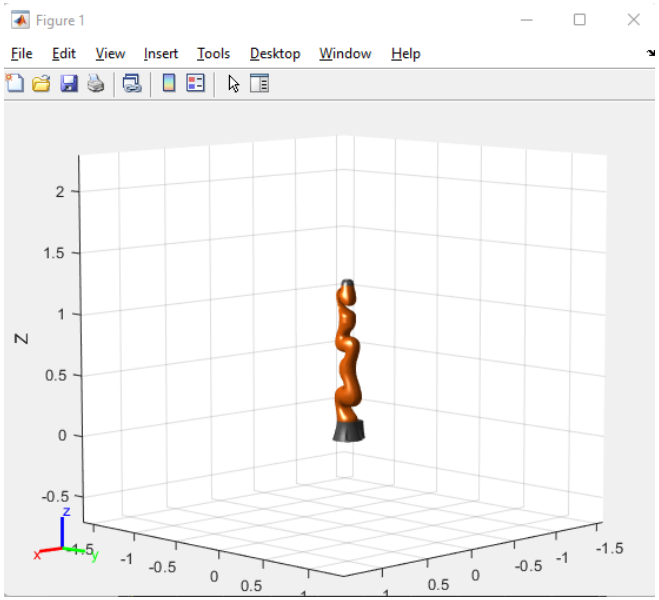
- Genera un ‘onda quadra’ che alterna qd_pick e qd_place ogni 5 secondi (tempo presumibilmente sufficiente per esaurire i transitori)

Modello Simulink del sistema controllato

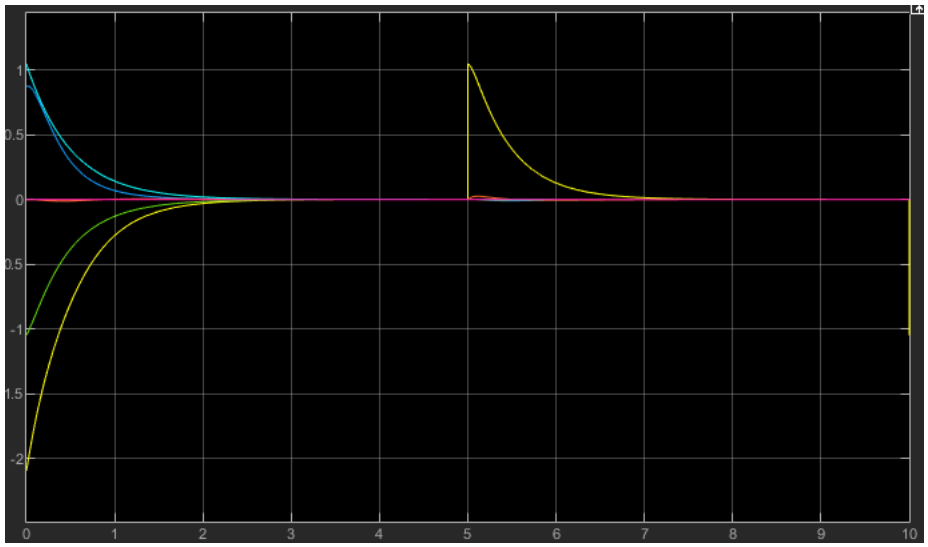


Script di visualizzazione risultati (post-simulazione)

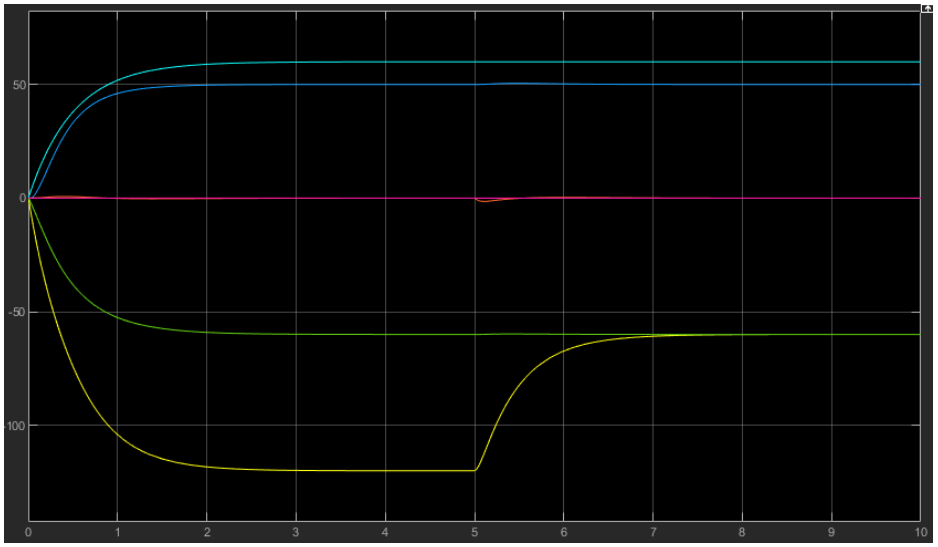
```
% visualizzazione 3D configurazioni assunte dal robot
n = length(out.tout);

for i=1:n

configNow = out.simout(i,:)';
show(robot,configNow,'PreservePlot',false,'Frames','off');
drawnow;

end
```



Risultati



- errore di posizione (rad)

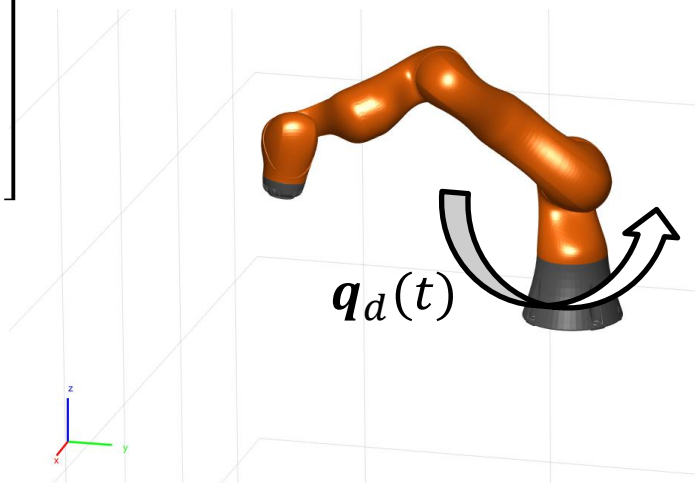


- posizione giunti (deg)

Robot Antropomorfo: Inseguimento di traiettoria

- Risolvere un problema di inseguimento di una traiettoria desiderata nello spazio di giunto per il manipolatore antropomorfo Kuka LBR 14 (robot ridondante 7 DoF)
- Tale problema può essere affrontato con la tecnica di controllo in Feedback Linearization denominata ‘controllo a dinamica inversa’
- In particolare si desidera inseguire la seguente traiettoria:

$$\boldsymbol{q_d(t)} = \begin{bmatrix} \sin(t) \\ 50\frac{\pi}{180} \\ 0 \\ -60\frac{\pi}{180} \\ 0 \\ 60\frac{\pi}{180} \\ 0 \end{bmatrix} \quad \boldsymbol{\dot{q}_d(t)} = \begin{bmatrix} \cos(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{\ddot{q}_d(t)} = \begin{bmatrix} -\sin(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Robot Antropomorfo: Inseguimento di traiettoria

- L'azione di controllo cancella le non-linearità tramite feedback dello stato:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \quad \Rightarrow \quad \ddot{\boldsymbol{q}} = \boldsymbol{v}$$

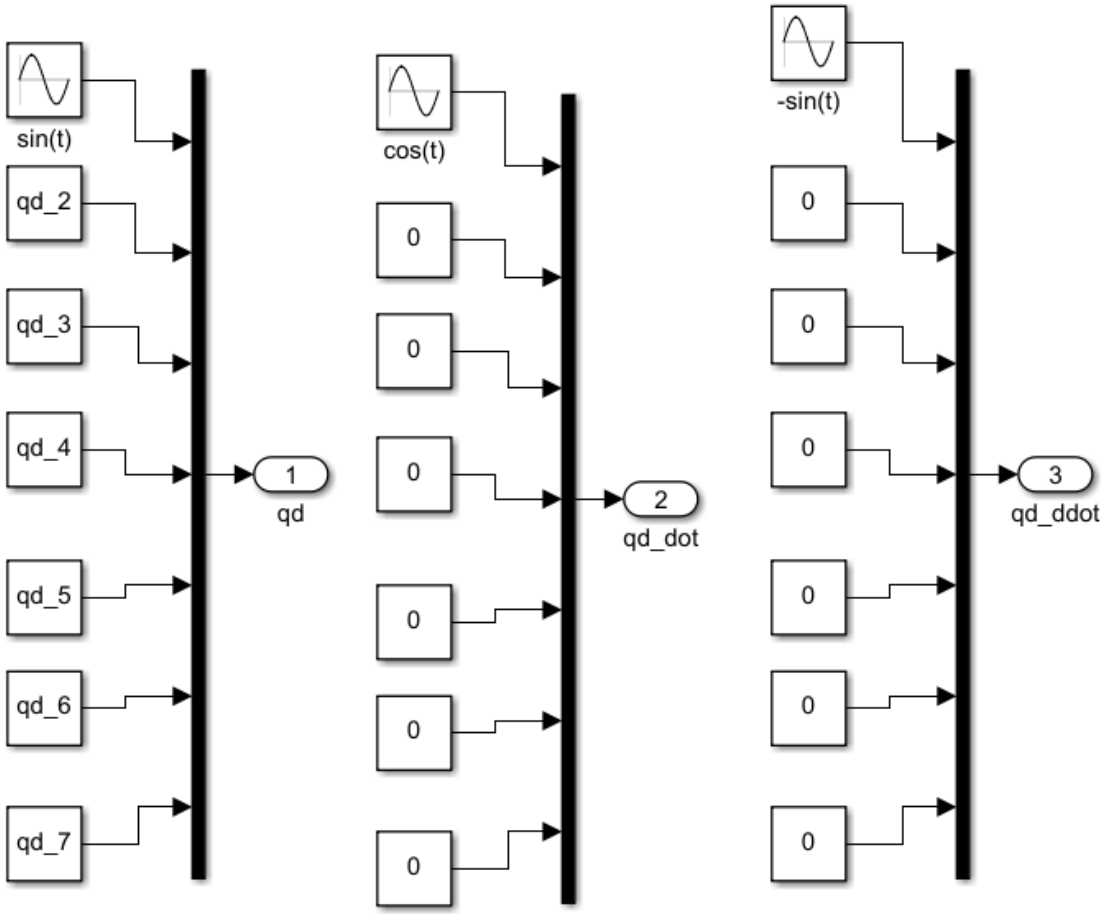
- La dinamica dell'errore di tracking viene resa asintoticamente stabile imponendo:

$$\boldsymbol{v} = \ddot{\boldsymbol{q}}_{\boldsymbol{d}} - \boldsymbol{K}_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\boldsymbol{d}}) - \boldsymbol{K}_p(\boldsymbol{q} - \boldsymbol{q}_{\boldsymbol{d}})$$

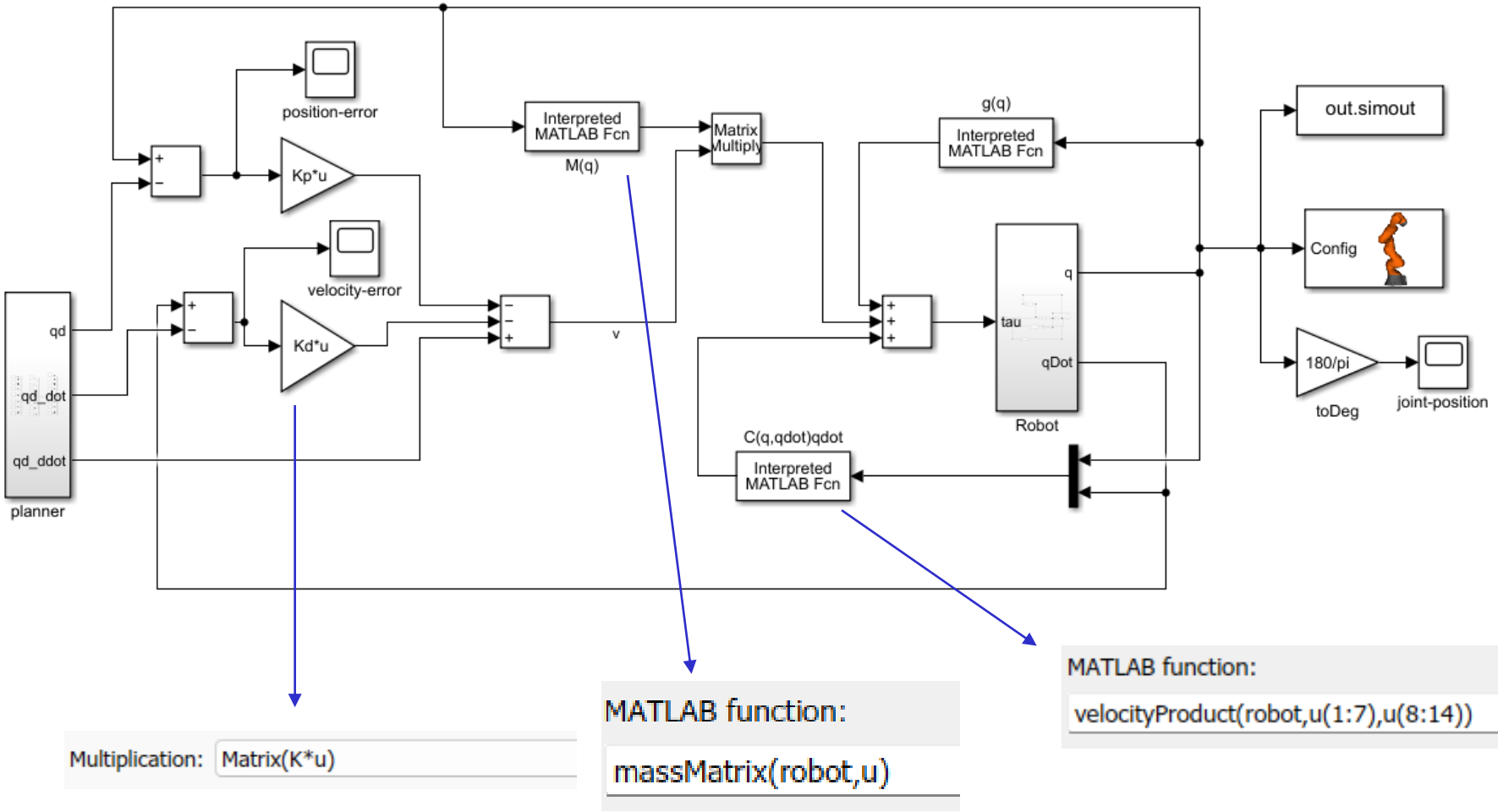
Script di inizializzazione

```
% import robot
robot = importrobot('iiwa14.urdf');
robot.DataFormat = 'column';
robot.Gravity = [0;0;-9.81];

% configurazione iniziale e target
(componenti 2->7)
q0 = [0;0;0;0;0;0;0];
qd_2 = 50*pi/180;
qd_3 = 0;
qd_4 = -60*pi/180;
qd_5 = 0;
qd_6 = 60*pi/180;
qd_7 = 0;

% controllo
Kd = 10*eye(7);
Kp = 10*eye(7);
```

Pianificatore traiettoria

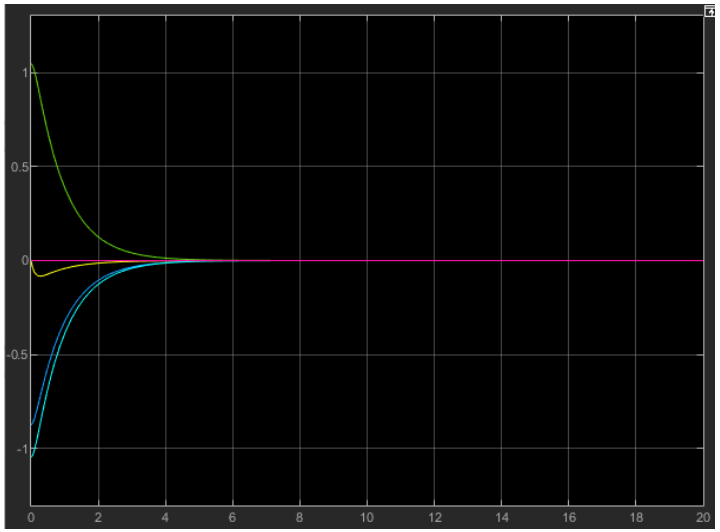


Schema di controllo

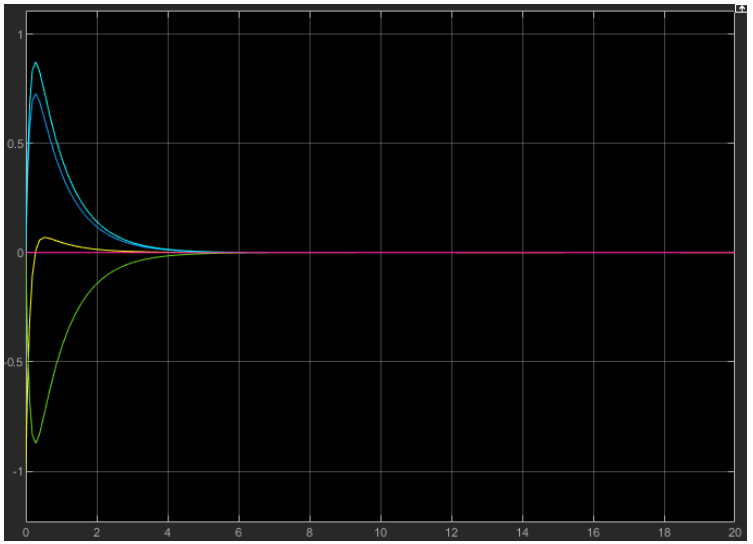


Risultati

- errore di posizione (rad)



- errore di velocità (rad)



- posizione giunti (deg)