

# **Simulazione dei Sistemi dinamici con Matlab-Simulink**

## **Modellazione Simulink – parte terza**

**Ing. Alessandro Pisano**  
**apisano@unica.it**

## Approccio vettoriale alla simulazione di sistemi dinamici

La modellazione di sistemi di ordine elevato è resa complicata dal fatto che sulla base della procedura usata fino ad ora si deve inserire un blocco integratore per ciascuna variabile dinamica del sistema.

**Rappresentando il modello matematico del sistema in termini vettoriali**, e sfruttando la possibilità di **applicare all'ingresso di un blocco integratore un segnale vettoriale**, la complessità dello schema Simulink può essere significativamente ridotta.

## Equazioni di Lotka-Volterra

Le equazioni di Lotka-Volterra, note anche come modello preda-predatore, sono un sistema di equazioni differenziali non lineari del primo ordine che forniscono un modello matematico in grado di descrivere la dinamica di un ecosistema in cui interagiscono due specie animali: una delle due come predatore, l'altra come la sua preda

$$\dot{x}(t) = Ax(t) - Bx(t)y(t)$$

$x(t)$  popolazione prede

$$\dot{y}(t) = -Cy(t) + Dy(t)x(t)$$

$y(t)$  popolazione predatori

$A$  = tasso di natalità delle prede (che si suppone abbiano riserve di cibo illimitate)

$B$  = tasso a cui le prede sono attaccate dai predatori

$$A, B, C, D > 0$$

$C$  = tasso di mortalità dei predatori

$D$  = tasso di crescita della popolazione dei predatori  
(in relazione alla disponibilità di prede di cui nutrirsi)

$$\dot{x}(t) = Ax(t) - Bx(t)y(t)$$

$$\dot{y}(t) = -Cy(t) + Dy(t)x(t)$$

Modello realizzato  
costruendo termine per  
termine le equazioni

X0=0.2;

Y0=0.5;

**File:**

LotkaVolterra\_01.slx

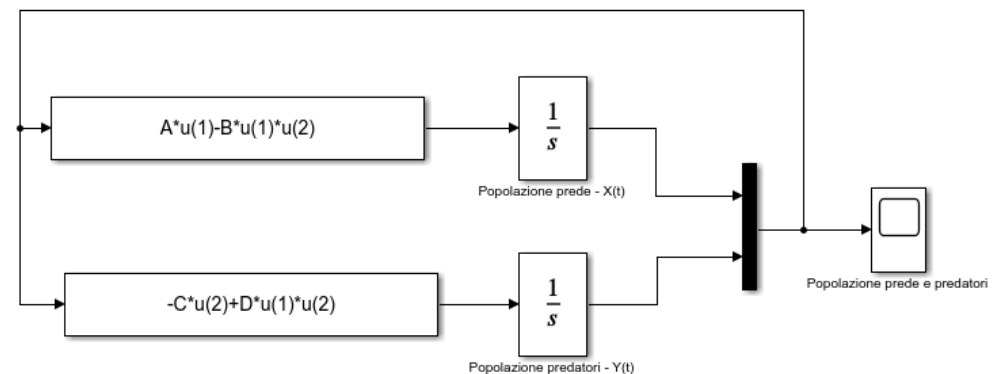
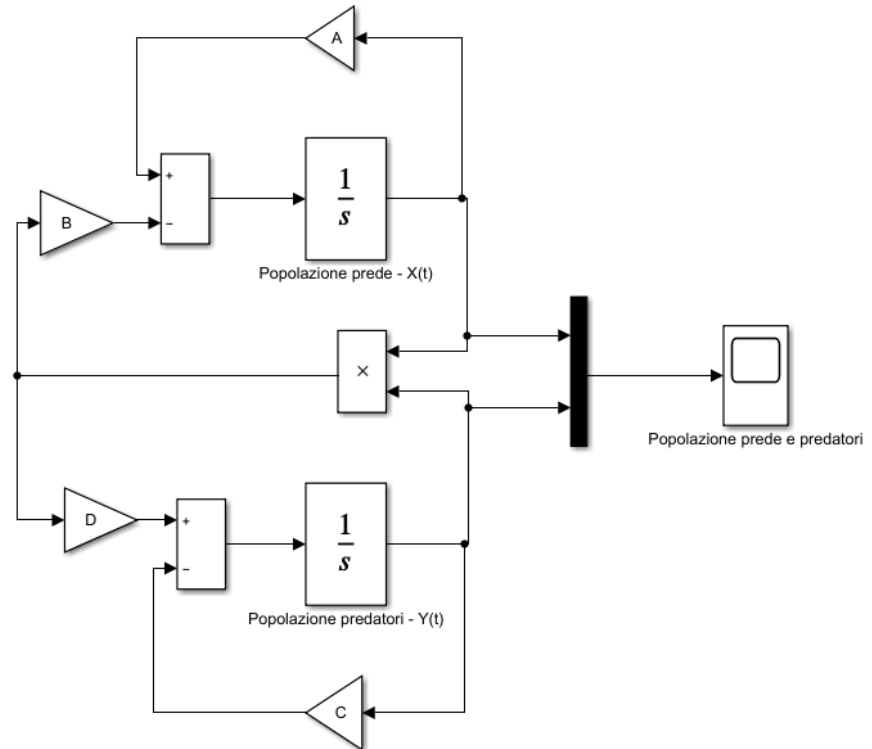
A=1;

B=1;

C=2;

D=1;

Modello realizzato  
impiegando blocchi Fcn



La riscrittura in forma matriciale del modello matematico consiste nel definire un vettore di segnali che contenga le variabili incognite

$$q(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}$$

e nel riscrivere il modello matematico in funzione di questo vettore

$$\dot{x}(t) = Ax(t) - Bx(t)y(t)$$

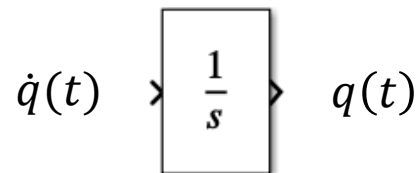
$$\dot{y}(t) = -Cy(t) + Dy(t)x(t)$$



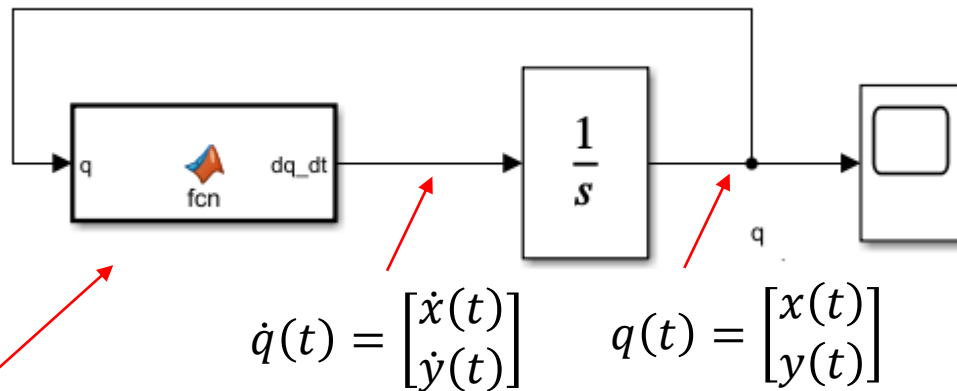
$$\dot{q} = M(q)$$

$$M(q) = \begin{bmatrix} Aq_1 - Bq_1q_2 \\ -Cq_2 + Dq_1q_2 \end{bmatrix}$$

A questo punto si realizza il modello Simulink con un unico blocco integratore facendo in modo che al terminale di uscita di quest'ultimo venga restituito l'intero vettore  $q(t)$



Generiamo l'ingresso dell'integratore per mezzo di un blocco Matlab Function



```
function dq_dt = fcn(q)
```

```
A=1; B=1;  
C=2; D=1;
```

```
x=q(1);  
y=q(2);
```

```
dq_dt=[ A*x-B*x*y ; -C*y+D*x*y ];
```

**File:**

LotkaVolterra.slx

Block Parameters: Integrator1

Integrator

Continuous-time integration of the input signal.

Parameters

External reset: none

Initial condition source: internal

Initial condition:

[0.2; 0.5]

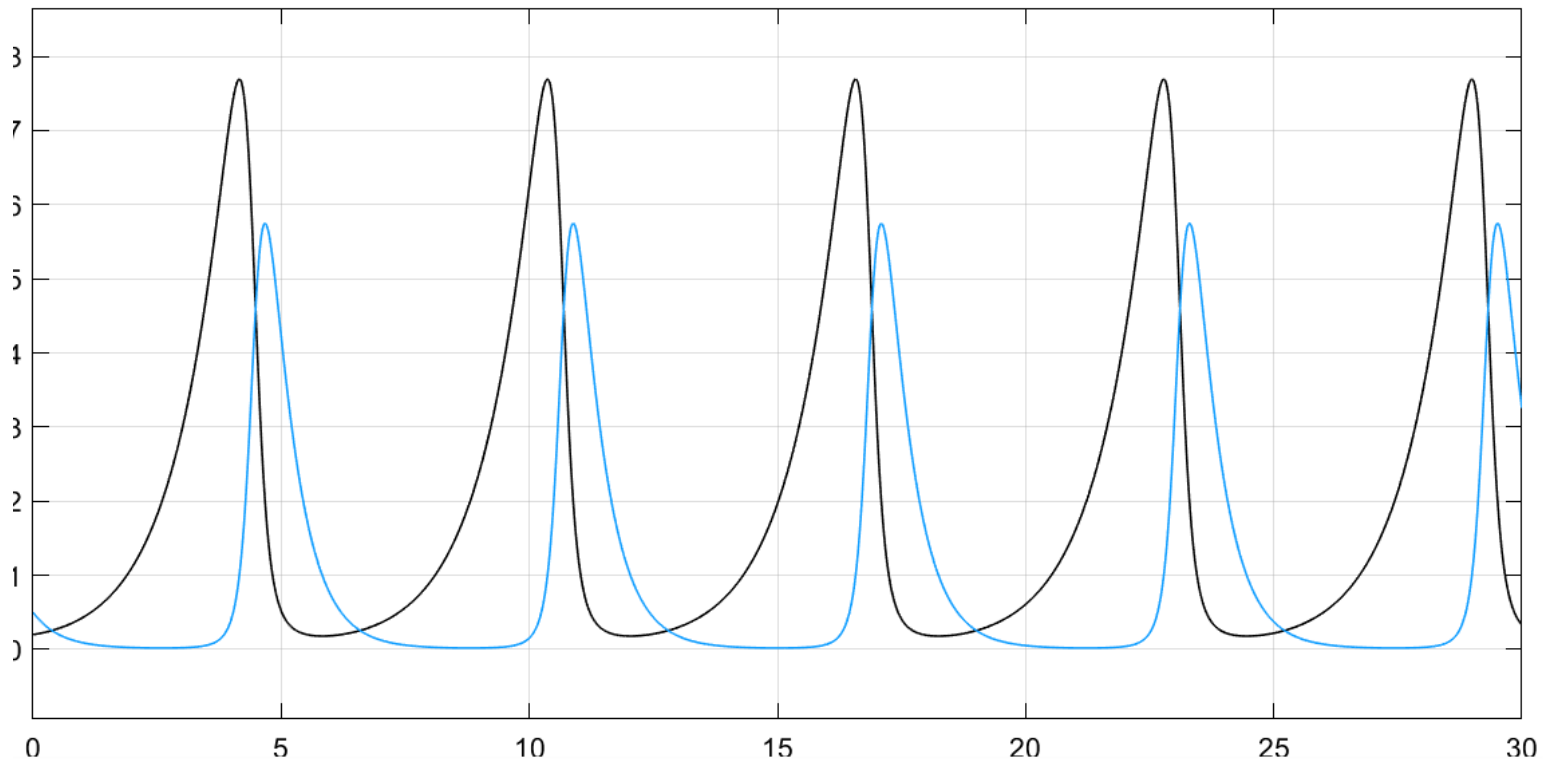
La condizione iniziale dell'integratore  
è ora un vettore colonna

Il blocco Matlab Function restituisce  
all'esterno un vettore colonna

## Evoluzioni temporali delle due popolazioni (ciclo limite)

$x(t)$  - prede

$y(t)$  - predatori



## SIMULAZIONE DI UN MANIPOLATORE ROBOTICO «SCARA»



Lo **SCARA**, acronimo di *Selective Compliance Assembly Robot Arm*, è un tipo di robot industriale, che muove vari "bracci" parallelamente aò piano orizzontale ed è dotato di una presa che può salire e scendere in quello verticale.

Modelliamo il movimento dei due bracci («link») di un robot come quello in figura.



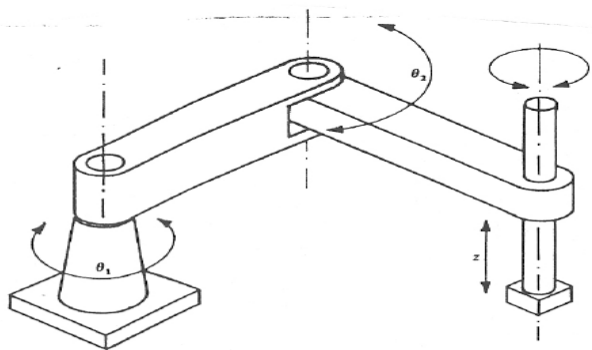


Fig. 26

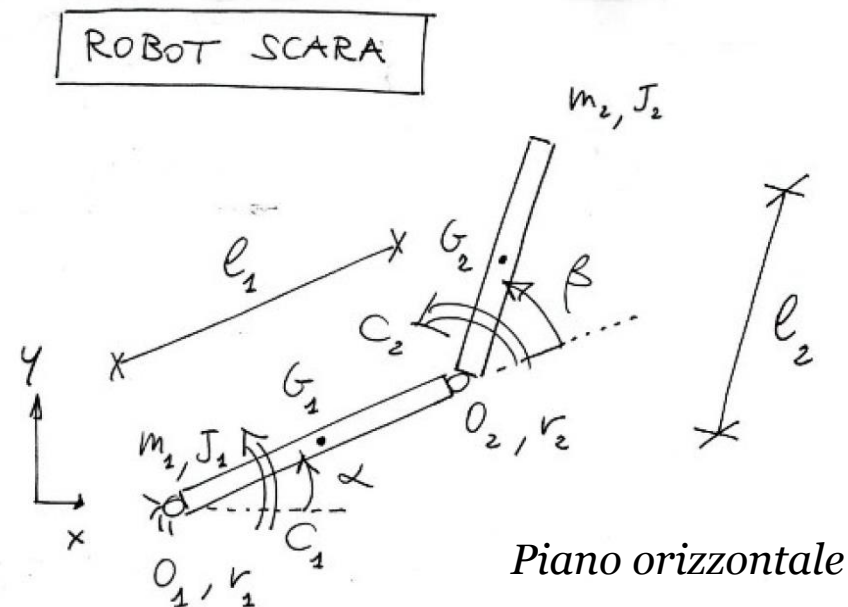
Ciascun link ruota intorno ad un asse ortogonale al piano orizzontale

## Variabili incognite

- $\alpha(t)$  Angolo fra il primo link ed una direzione fissa
- $\beta(t)$  Angolo del secondo link rispetto al primo

## Ingressi esterni

- $C_1(t), C_2(t)$  coppie applicate ai giunti



## Parametri

$m_1, m_2$  masse dei link

$l_1, l_2$  lunghezze dei link

$J_1, J_2$  momenti di inerzia dei link

$r_1, r_2$  coeff. di attrito viscoso dei giunti

## Modello matematico

$$\left[ \frac{1}{4} m_1 \ell_1^2 + J_1 + m_2 \ell_1^2 + J_2 + \frac{1}{4} m_2 \ell_2^2 + m_2 \ell_1 \ell_2 \cos \beta(t) \right] \ddot{\alpha}(t) + \left[ J_2 + \frac{1}{4} m_2 \ell_2^2 + \frac{1}{2} m_2 \ell_1 \ell_2 \cos \beta(t) \right] \ddot{\beta}(t) + r_1 \dot{\alpha}(t) - m_2 \ell_1 \ell_2 \left( \dot{\alpha}(t) \dot{\beta}(t) + \frac{1}{2} \dot{\beta}^2(t) \right) \sin \beta(t) = C_1(t)$$

$$\left[ J_2 + \frac{1}{4} m_2 \ell_2^2 + \frac{1}{2} m_2 \ell_1 \ell_2 \cos \beta(t) \right] \ddot{\alpha}(t) + \left[ J_2 + \frac{1}{4} m_2 \ell_2^2 \right] \ddot{\beta}(t) + r_2 \dot{\beta}(t) + \frac{1}{2} m_2 \ell_1 \ell_2 \dot{\alpha}^2(t) \sin \beta(t) = C_2(t)$$

La riscrittura del modello in forma esplicita volta a seguire l'approccio di modellazione Simulink fin qui adottato sarebbe oltremodo onerosa

Seguiamo **l'approccio vettoriale** per descrivere il modello del robot SCADA in forma più compatta e facilmente gestibile

$$q(t) = \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix} \quad \dot{q}(t) = \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} \quad \ddot{q}(t) = \begin{bmatrix} \ddot{\alpha}(t) \\ \ddot{\beta}(t) \end{bmatrix}$$

Definiamo alcune costanti dipendenti dai parametri fisici del robot

$$A_1 = \frac{1}{4} m_1 \ell_1^2 + J_1 + m_2 \ell_1^2 + J_2 + \frac{1}{4} m_2 \ell_2^2$$

$$A_2 = m_2 \ell_1 \ell_2$$

$$A_3 = J_2 + \frac{1}{4} m_2 \ell_2^2$$

$$[A_1 + A_2 \cos \beta(t)] \ddot{\alpha}(t) + \left[ A_3 + \frac{1}{2} A_2 \cos \beta(t) \right] \ddot{\beta}(t) + r_1 \dot{\alpha}(t) - A_2 \left( \dot{\alpha}(t) \dot{\beta}(t) + \frac{1}{2} \dot{\beta}^2(t) \right) \sin \beta(t) = C_1(t)$$

$$\left[ A_3 + \frac{1}{2} A_2 \cos \beta(t) \right] \ddot{\alpha}(t) + A_3 \ddot{\beta}(t) + r_2 \dot{\beta}(t) + \frac{1}{2} A_2 \dot{\alpha}^2(t) \sin \beta(t) = C_2(t)$$

Termini inerziali

Coppie di attrito

Coppie centripete  
e di Coriolis

Coppie esterne  
applicate

$$[A_1 + A_2 \cos \beta(t)]\ddot{\alpha}(t) + \left[A_3 + \frac{1}{2}A_2 \cos \beta(t)\right]\ddot{\beta}(t) + r_1\dot{\alpha}(t) - A_2\left(\dot{\alpha}(t)\dot{\beta}(t) + \frac{1}{2}\dot{\beta}^2(t)\right)\sin \beta(t) = C_1(t)$$

$$\left[A_3 + \frac{1}{2}A_2 \cos \beta(t)\right]\ddot{\alpha}(t) + A_3\ddot{\beta}(t) + r_2\dot{\beta}(t) + \frac{1}{2}A_2\dot{\alpha}^2(t)\sin \beta(t) = C_2(t)$$



$$\boxed{M(\alpha, \beta) \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + R \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + F_{CC}(\alpha, \beta, \dot{\alpha}, \dot{\beta}) = \begin{bmatrix} C_1(t) \\ C_2(t) \end{bmatrix}}$$

$$M(\alpha, \beta) = \begin{bmatrix} A_1 + A_2 \cos \beta & A_3 + \frac{1}{2}A_2 \cos \beta \\ A_3 + \frac{1}{2}A_2 \cos \beta & A_3 \end{bmatrix}$$

$$R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

$$F_{CC}(\alpha, \beta, \dot{\alpha}, \dot{\beta}) = \begin{bmatrix} -A_2\left(\dot{\alpha}(t)\dot{\beta}(t) + \frac{1}{2}\dot{\beta}^2(t)\right)\sin \beta(t) \\ \frac{1}{2}A_2\dot{\alpha}^2(t)\sin \beta(t) \end{bmatrix}$$

$$M(\alpha, \beta) \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + R \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + F_{CC}(\alpha, \beta, \dot{\alpha}, \dot{\beta}) = C(t)$$

$$M(\alpha, \beta) = \begin{bmatrix} A_1 + A_2 \cos \beta & A_3 + \frac{1}{2} A_2 \cos \beta \\ A_3 + \frac{1}{2} A_2 \cos \beta & A_3 \end{bmatrix}$$

Matrice di inerzia (sempre invertibile)

$$F_{CC}(\alpha, \beta, \dot{\alpha}, \dot{\beta}) = \begin{bmatrix} -A_2 \left( \dot{\alpha} \dot{\beta} + \frac{1}{2} \dot{\beta}^2 \right) \sin \beta \\ \frac{1}{2} A_2 \dot{\alpha}^2 \sin \beta \end{bmatrix}$$

Vettore dei termini centripeti e di Coriolis

$$R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

Matrice dei coefficienti di attrito viscoso

$$C(t) = \begin{bmatrix} C_1(t) \\ C_2(t) \end{bmatrix}$$

Vettore delle coppie esterne applicate

$$M(\alpha, \beta) \begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} + R \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} + F_{CC}(\alpha, \beta, \dot{\alpha}, \dot{\beta}) = C(t)$$



$$M(q)\ddot{q}(t) + R\dot{q}(t) + F_{CC}(q(t), \dot{q}(t)) = C(t)$$



**Modello matematico in forma esplicita**

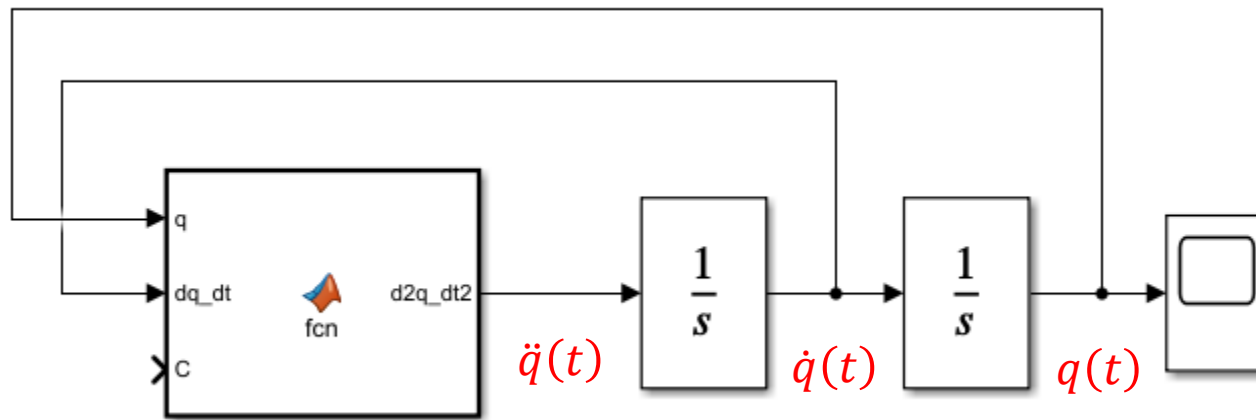
$$\ddot{q}(t) = M^{-1}(q) C(t) - M^{-1}(q) R\dot{q}(t) - M^{-1}(q) F_{CC}(q(t), \dot{q}(t))$$

Il modello matematico viene facilmente espresso in forma esplicita senza bisogno di sviluppare i conti, e sulla base di tale espressione risulta semplice predisporre un modello Simulink estremamente compatto del robot SCADA oggetto del presente esempio

$$\ddot{q}(t) = M^{-1}(q) C(t) - M^{-1}(q) R \dot{q}(t) - M^{-1}(q) F_{cc}(q(t), \dot{q}(t))$$

$$= M^{-1}(q) [C(t) - R \dot{q}(t) - F_{cc}(q(t), \dot{q}(t))]$$

Come se si dovesse modellare una equazione differenziale scalare di secondo grado, si importino nella pagina di lavoro **due blocchi integratori**, e li si dispongano in serie fra loro. Avendo come obiettivo quello di fare in modo che il vettore incognito  $q(t)$  venga prodotto al terminale di uscita dell'ultimo integratore, si realizzi il seguente schema a blocchi in cui l'ingresso del primo integratore (che deve coincidere con il vettore  $\ddot{q}(t)$ ) sarà generato mediante un blocco Matlab Function



```
function d2q_dt2 = fcn(q, dq_dt, C)
```

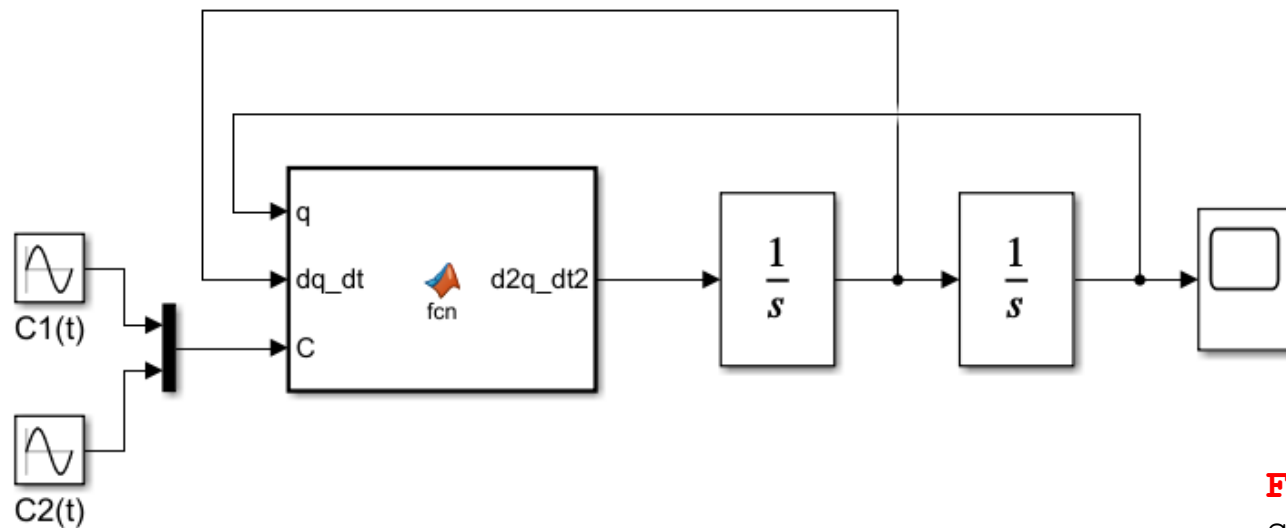
**Prima riga della funzione**

Inseriamo come condizioni iniziali degli integratori le variabili  $q_0$  e  $q_{punto0}$  che saranno definite in uno Script (dovranno entrambe essere dei **vettori colonna**).

Applichiamo in ingresso il vettore:

$$C(t) = \begin{bmatrix} C_1(t) \\ C_2(t) \end{bmatrix} = \begin{bmatrix} 10 \sin(t) \\ 6 \cos(3t) \end{bmatrix}$$

costruendolo secondo il seguente schema



**File:**  
SCARA.slx



Ora si predisponga il contenuto del blocco Matlab Function. Si ricordi che all'interno di un blocco Matlab Function non sono visibili le variabili del workspace. I parametri fisici del robot dovranno pertanto essere definiti all'interno del corpo della funzione

```
function d2q_dt2 = fcn(q,dq_dt,C)

m1=5; m2=5;
l1=2; l2=2;
J1=1; J2=2;
r1=5; r2=5;

A1=0.25*m1*l1^2+J1+m2*l1^2+J2+0.25*m2*l2^2;
A2=m2*l1*l2;
A3=J2+0.25*m2*l2^2;

alfa=q(1); beta=q(2);
alfadot=dq_dt(1);
betadot=dq_dt(2);

M=[A1+A2*cos(beta) A3+0.5*A2*cos(beta);
    A3+0.5*A2*cos(beta) A3];
R=[r1 0;0 r2];

Fcc=[-A2*(alfadot*betadot+0.5*betadot^2)*sin(beta);
    0.5*A2*alfadot^2*sin(beta)];

d2q_dt2 = inv(M)*(C-R*dq_dt-Fcc);
```

```
clear all
close all
clc
```

**File:**

SCARA\_dati.m

```
q0=[pi/2;0];
qpunto0=[0;0];
```

```
sim('SCARA')
```

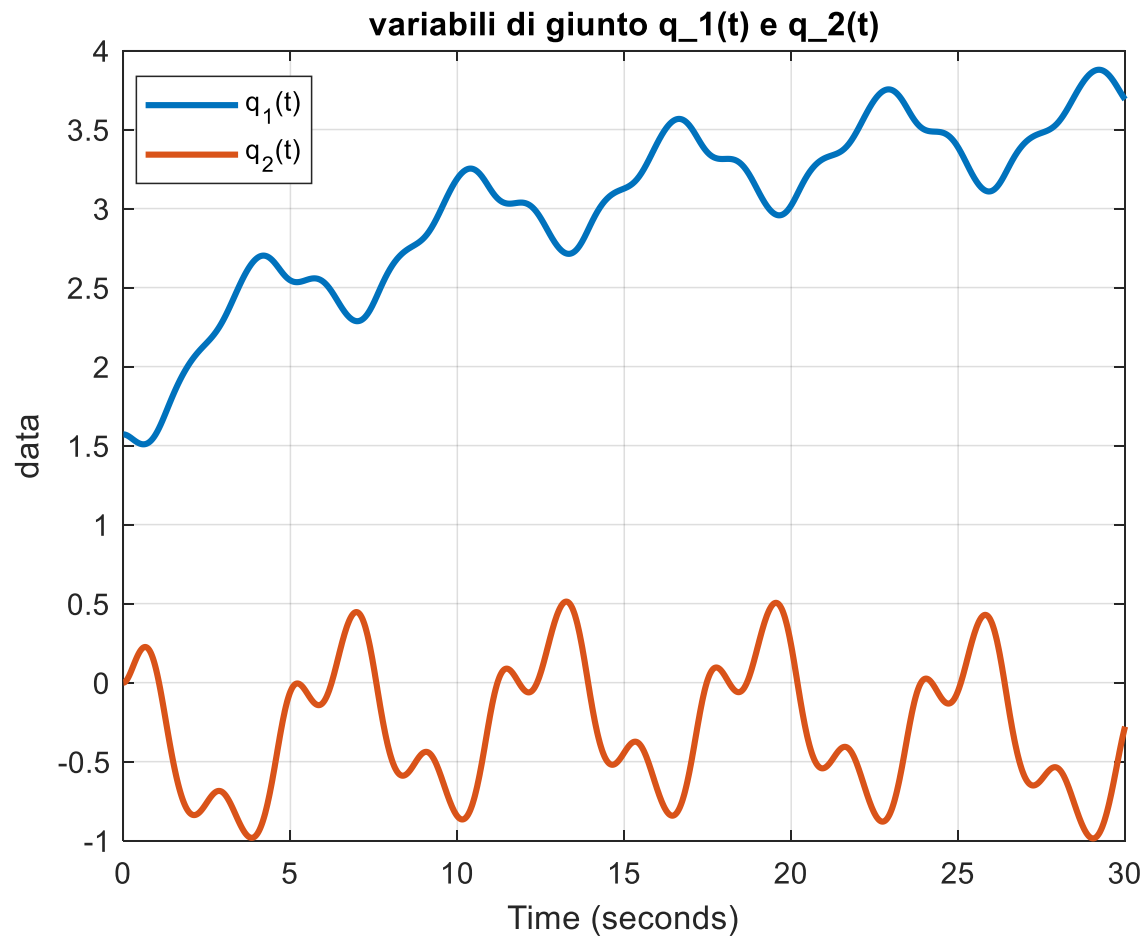
```
%% creazione grafici
```

```
figure(1)
plot(q, 'Linewidth', 2), grid
title('variabili di giunto q_1(t) e q_2(t)')
legend('q_1(t)', 'q_2(t)')
```

```
figure(2)
plot(q.Time, q.Data(:, 1), 'Linewidth', 2), grid
title('variabile di giunto q_1(t)')
xlabel('Tempo [sec]')
```

```
figure(3)
plot(q.Time, q.Data(:, 2), 'Linewidth', 2), grid
title('variabile di giunto q_2(t)')
xlabel('Tempo [sec]')
```

## Risultati



**Esercizio:** in analogia con quanto fatto per il modello del bioreattore, si includa nello schema un **sistema di controllo** in grado di posizionare gli assi del robot in un punto di lavoro desiderato

Siano  $\alpha^d$  e  $\beta^d$  le coordinate costanti di un punto di lavoro desiderato

Si applichino le seguenti coppie ai giunti (controllore **PD**)

$$C_1(t) = -K_d \dot{\alpha}(t) + K_p (\alpha^d - \alpha(t))$$

$$C_2(t) = -K_d \dot{\beta}(t) + K_p (\beta^d - \beta(t))$$

$K_d, K_p$  “guadagni” costanti

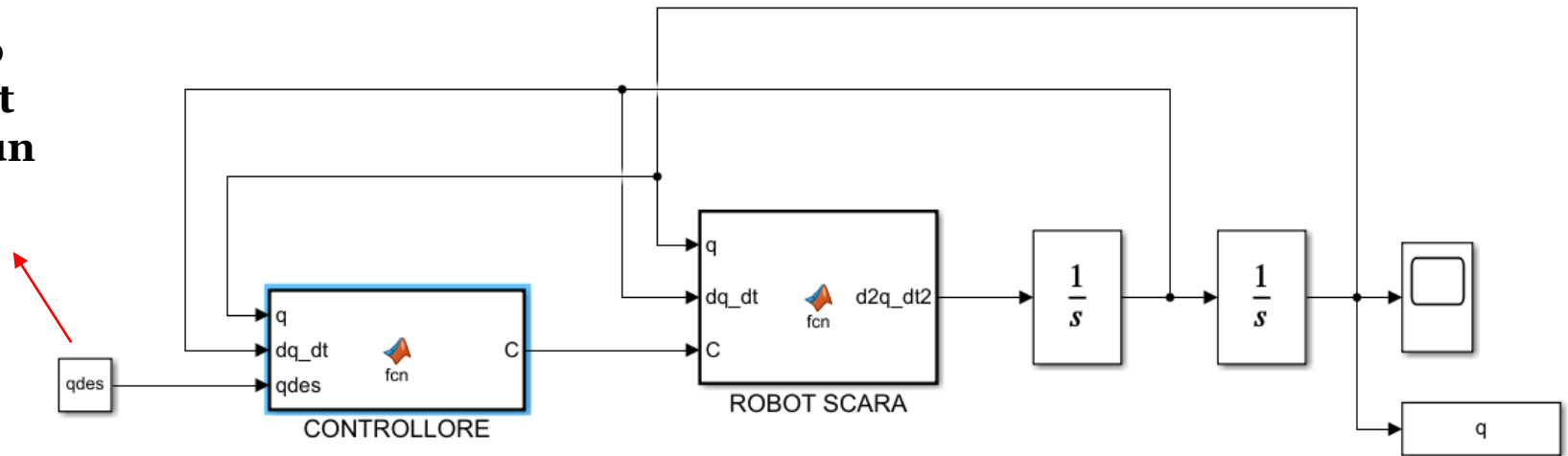


$$C(t) = -K_d \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} + K_p \left( \begin{bmatrix} \alpha^d \\ \beta^d \end{bmatrix} - \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix} \right)$$

$$K_p = 100$$

$$K_d = 20$$

**Il blocco  
Constant  
genera un  
vettore**



### Files:

SCARA\_feedback.slx  
SCARA\_feedback\_dati.m

```
function C = fcn(q,dq_dt,qdes)
```

```
Kp=100;
```

```
Kd=20;
```

```
C1 = -Kd*dq_dt(1)+Kp*(qdes(1)-q(1));
```

```
C2 = -Kd*dq_dt(2)+Kp*(qdes(2)-q(2));
```

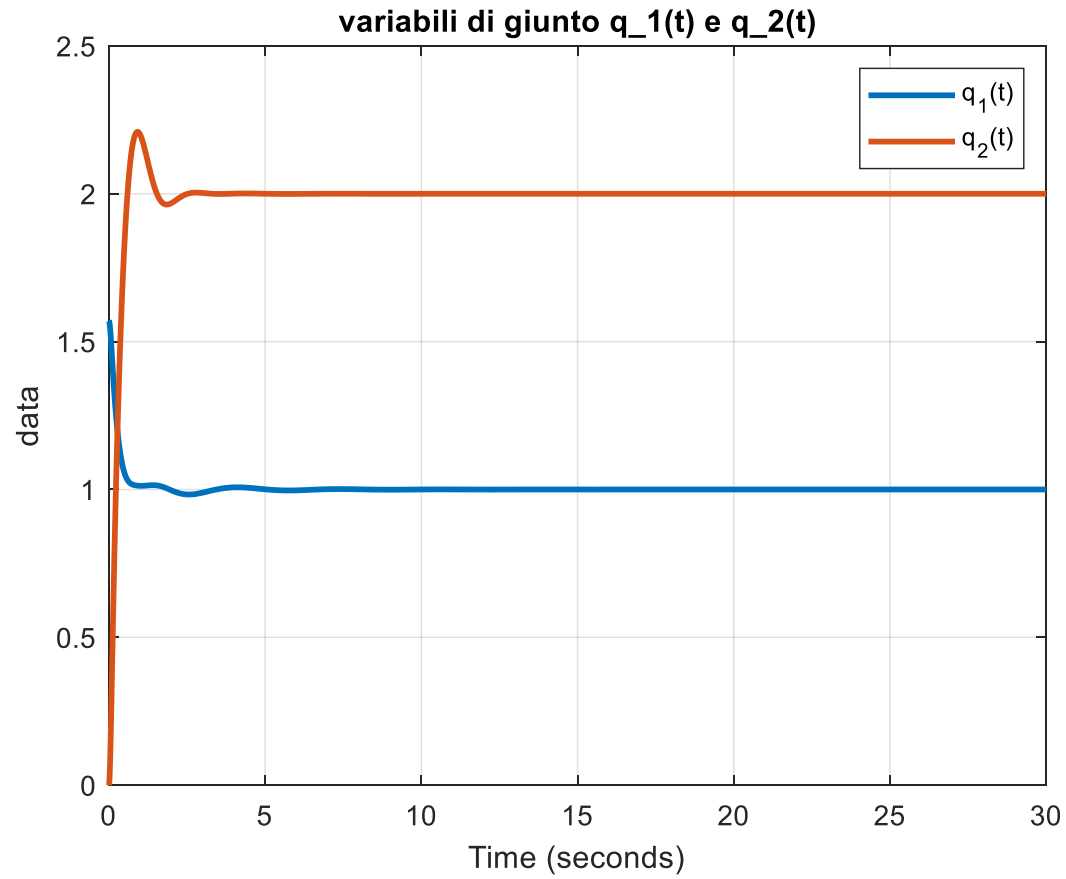
```
C=[C1;C2]
```

**Script:** clear all  
close all  
clc

```
q0=[pi/2;0];  
qpunto0=[0;0];  
qdes=[1;2];
```

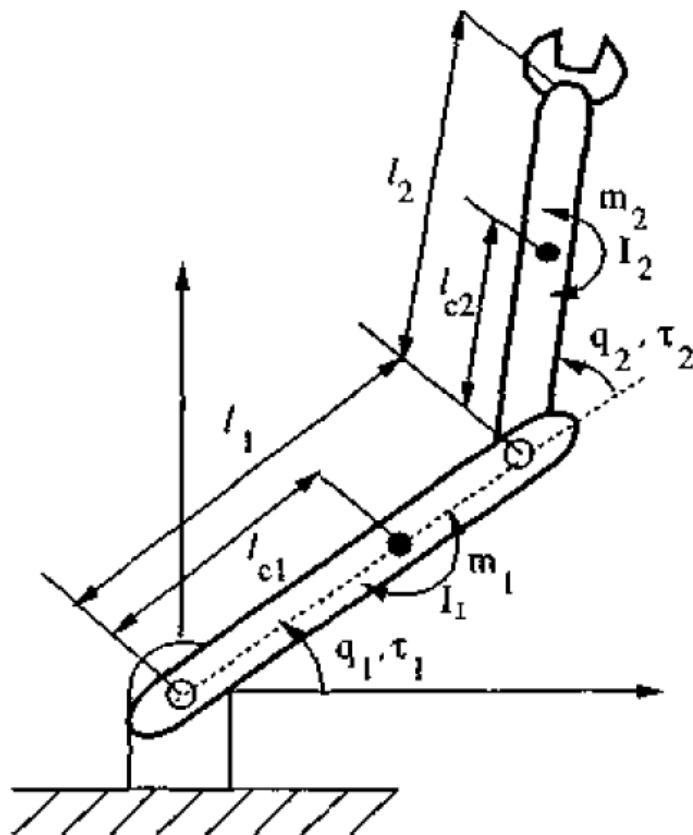
```
sim('SCARA_feedback')
```

## Risultati



# ESEMPI ADDIZIONALI CON TRACCIA DELLA SOLUZIONE

## SIMULAZIONE DI UN MANIPOLATORE PLANARE A 2 GDL



### Variabili incognite

$q_1(t)$  Angolo fra il primo link ed una direzione fissa

$q_2(t)$  Angolo del secondo link rispetto al primo

### Ingressi esterni

$\tau_1(t), \tau_2(t)$  coppie applicate ai giunti

### Parametri

$m_1, m_2$  masse dei link

$\ell_1, \ell_2$  lunghezze dei link

$\ell_{c1}, \ell_{c2}$  distanza del centro di massa dei link dal giunto a monte

$I_1, I_2$  momenti di inerzia dei link



## Modello matematico

$$\begin{aligned}
 &[m_1 \ell_{c1}^2 + I_1 + m_2(\ell_1^2 + \ell_{c2}^2 + 2\ell_1 \ell_{c2} \cos q_2) + I_2] \ddot{q}_1(t) + [m_2 \ell_1 \ell_{c2} \cos q_2 + m_2 \ell_{c2}^2 + I_2] \ddot{q}_2(t) \\
 &\quad - 2m_2 \ell_1 \ell_{c2} \sin q_2 \dot{q}_1 \dot{q}_2 - m_2 \ell_1 \ell_{c2} \sin q_2 \dot{q}_2^2 + m_1 \ell_{c1} g \cos q_1 \\
 &\quad + m_2 g [\ell_{c2} \cos(q_1 + q_2) + \ell_1 \cos q_1] = \tau_1(t) \\
 &[m_2 \ell_1 \ell_{c2} \cos q_2 + m_2 \ell_{c2}^2 + I_2] \ddot{q}_1(t) + [m_2 \ell_{c2}^2 + I_2] \ddot{q}_2(t) + m_2 \ell_1 \ell_{c2} \sin q_2 \dot{q}_1^2 + m_2 \ell_{c2} g \cos(q_1 + q_2) = \tau_2(t)
 \end{aligned}$$

## Dati per la simulazione

### Parametri

$$m_1 = m_2 = 5 \text{ kg}$$

$$\ell_1 = \ell_2 = 2 \text{ m}$$

$$\ell_{c1} = \ell_{c2} = 1 \text{ m}$$

$$I_1 = 1 \text{ kg m}^2 \quad I_2 = 2 \text{ kg m}^2$$

### Coppie applicate

$$\tau_1(t) = 8 \text{ Nm} = \text{cost.}$$

$$\tau_2(t) = 4 \text{ Nm} = \text{cost.}$$

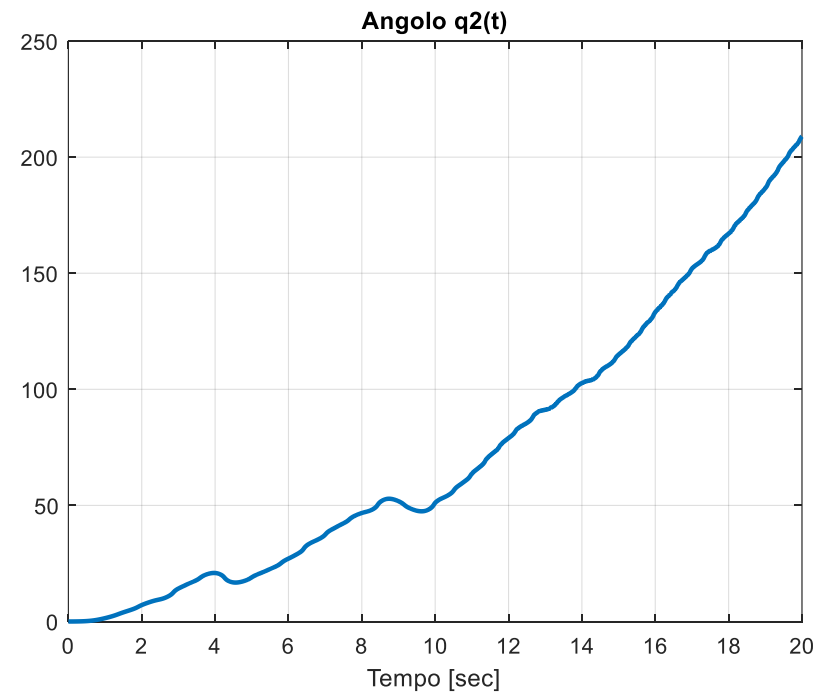
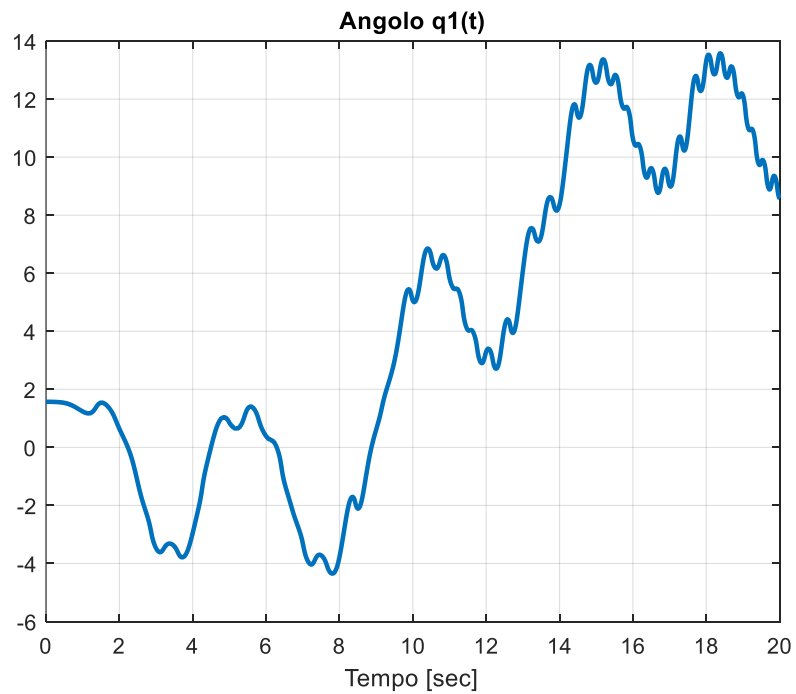
### Condizioni iniziali

$$q_1(0) = \pi/2$$

$$q_2(0) = 0$$

$$\dot{q}_1(0) = \dot{q}_2(0) = 0$$

## Risultati



## Modello matematico in forma vettoriale

$$q(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} \quad M(q)\ddot{q}(t) + F(q, \dot{q}) = \tau$$



**Forma esplicita**  $\ddot{q}(t) = M^{-1}(q)[\tau - F(q, \dot{q})]$

$$M(q) = \begin{bmatrix} m_1 \ell_{c1}^2 + I_1 + m_2(\ell_1^2 + \ell_{c2}^2 + 2\ell_1 \ell_{c2} \cos q_2) + I_2 & m_2 \ell_1 \ell_{c2} \cos q_2 + m_2 \ell_{c2}^2 + I_2 \\ m_2 \ell_1 \ell_{c2} \cos q_2 + m_2 \ell_{c2}^2 + I_2 & m_2 \ell_{c2}^2 + I_2 \end{bmatrix}$$

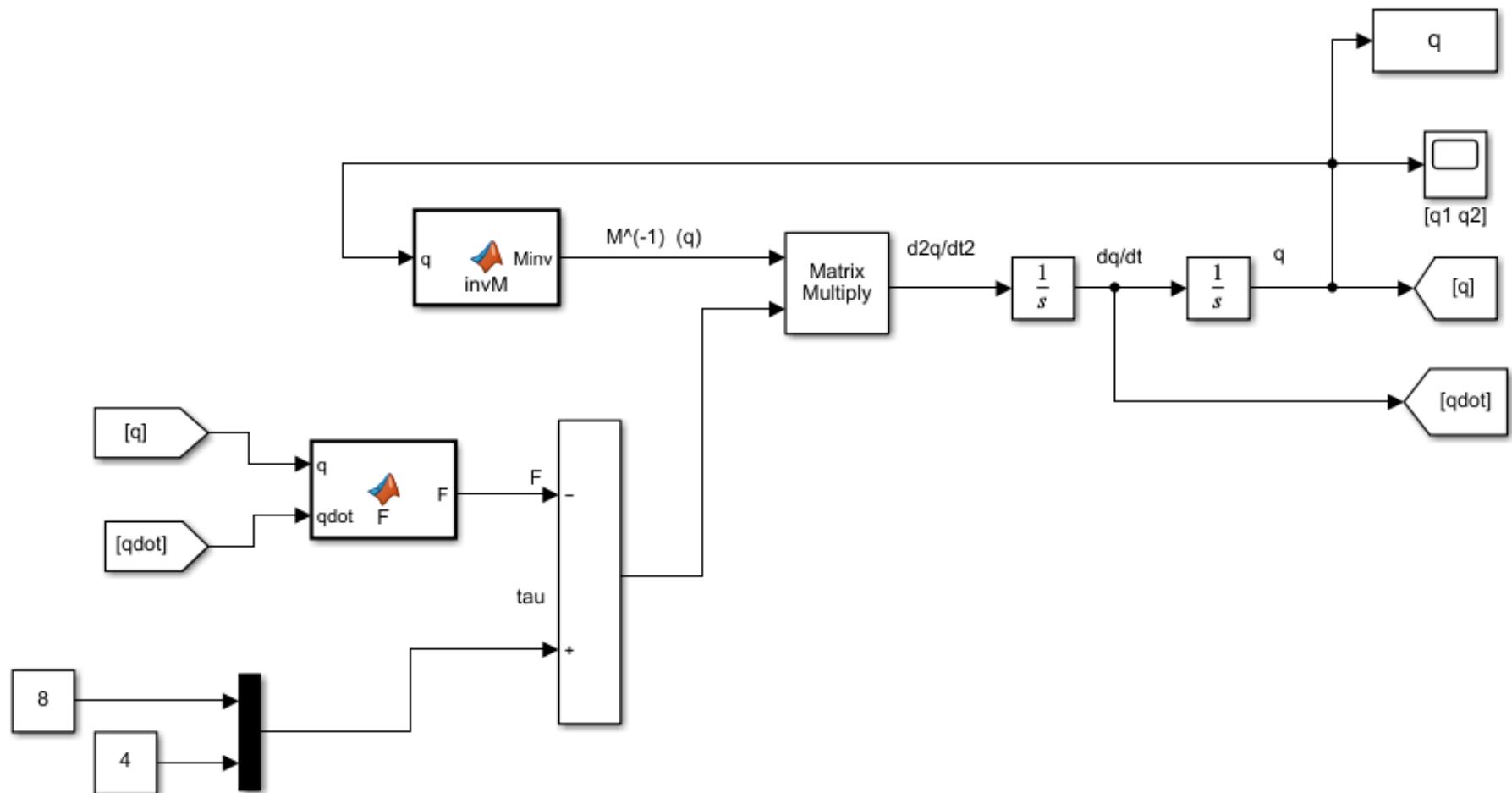
$$F(q, \dot{q}) = B(q, \dot{q})\dot{q}(t) + g(q)$$

$$B(q, \dot{q}) = m_2 \ell_1 \ell_{c2} \sin q_2 \begin{bmatrix} -\dot{q}_2 & -\dot{q}_1 - \dot{q}_2 \\ \dot{q}_1 & 0 \end{bmatrix}$$

**Forma equivalente:**

$$B(q, \dot{q}) = m_2 \ell_1 \ell_{c2} \sin q_2 \begin{bmatrix} -2\dot{q}_2 & -\dot{q}_2 \\ \dot{q}_1 & 0 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} m_1 \ell_{c1} g \cos q_1 + m_2 g [\ell_{c2} \cos(q_1 + q_2) + \ell_1 \cos q_1] \\ m_2 \ell_{c2} g \cos(q_1 + q_2) \end{bmatrix}$$



```
function Minv = invM(q)
```

```
m1=5;
```

```
m2=5;
```

```
l1=2;
```

```
lc1=1;
```

```
l2=2;
```

```
lc2=1;
```

```
I1=1;
```

```
I2=2;
```

```
q1=q(1);
```

```
q2=q(2);
```

```
M11= m1*lc1^2+I1+m2*(l1^2+lc2^2+2*l1*lc2*cos(q2))+I2;
```

```
M12= m2*l1*lc2*cos(q2)+m2*lc2^2+I2;
```

```
M21=M12;
```

```
M22=m2*lc2^2+I2;
```

```
M=[M11 M12;M21 M22];
```

```
Minv=inv(M);
```

```
function F = F(q,qdot)
```

```
m1=5;
```

```
m2=5;
```

```
l1=2;
```

```
lc1=1;
```

```
l2=2;
```

```
lc2=1;
```

```
I1=1;
```

```
I2=2;
```

```
g=9.8;
```

```
q1=q(1);
```

```
q2=q(2);
```

```
q1dot=qdot(1);
```

```
q2dot=qdot(2);
```

```
qdot=[q1dot;q2dot];
```

```
h=m2*l1*lc2*sin(q2);
```

```
B=[-h*q2dot -h*(q1dot+q2dot); h*q1dot 0];
```

```
grav=[m1*lc1*g*cos(q1)+m2*g*(lc2*cos(q1+q2)+l1*cos(q1));
```

```
m2*lc2*g*cos(q1+q2)];
```

```
F=B*qdot+grav;
```

## Integrazione nello schema di un sistema di controllo

### Coppie applicate

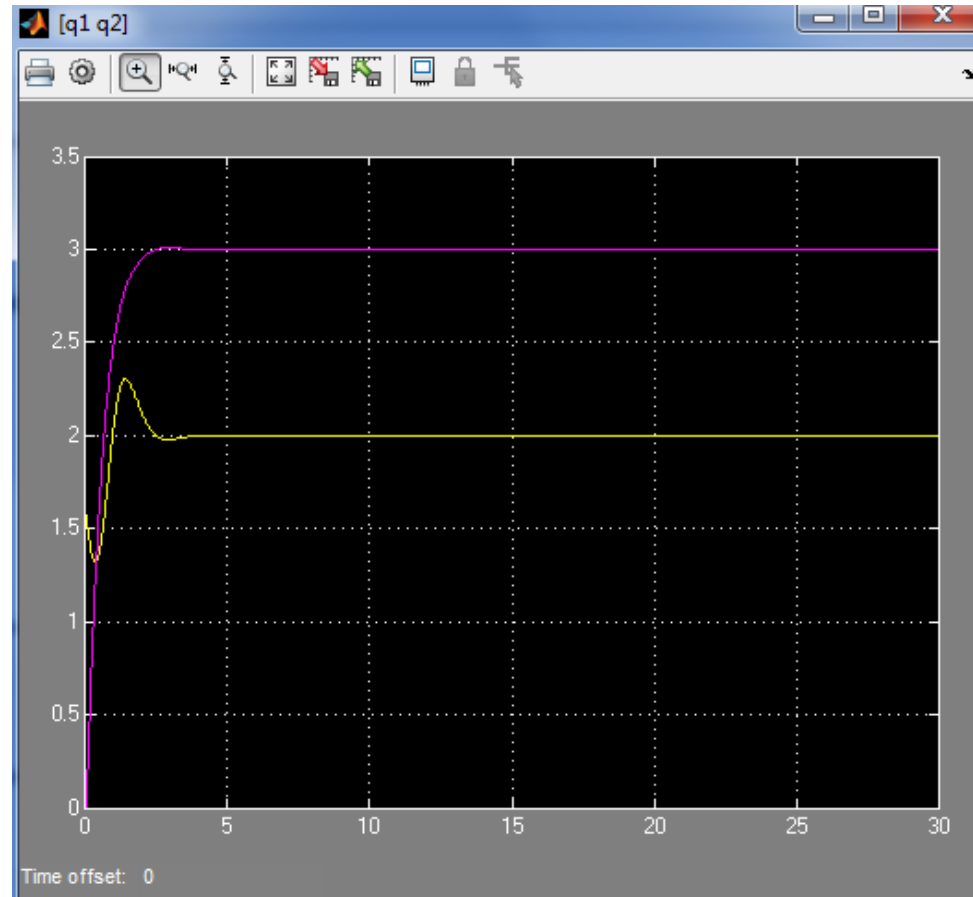
Siano  $q_1^d$  e  $q_2^d$  le coordinate costanti di un punto di lavoro desiderato

Si applichi il seguente vettore delle coppie ai giunti (controllore PD con compensazione diretta della gravità)

$$\tau(t) = g(q) + 100 \left( q^d - q(t) \right) - 50 \dot{q}(t) \qquad q^d = \begin{bmatrix} q_1^d \\ q_2^d \end{bmatrix}$$

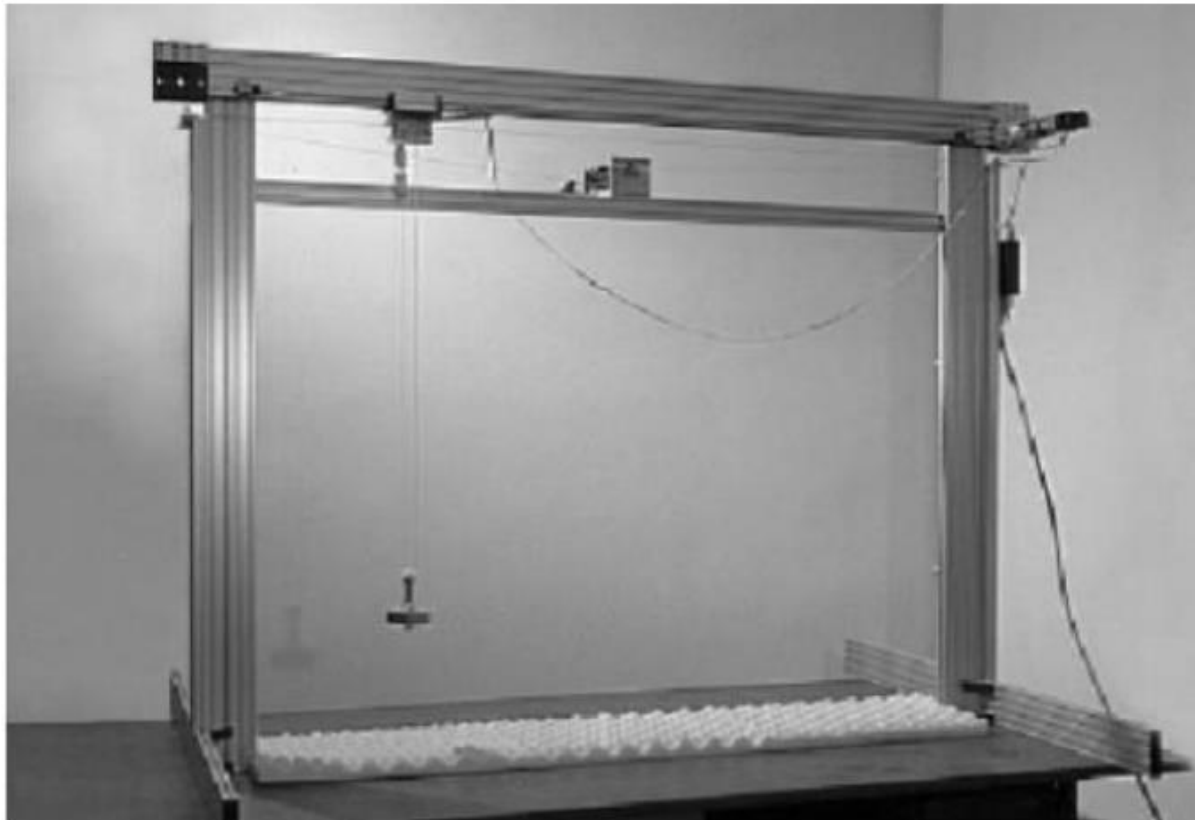
## Risultati

$$q^d = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$



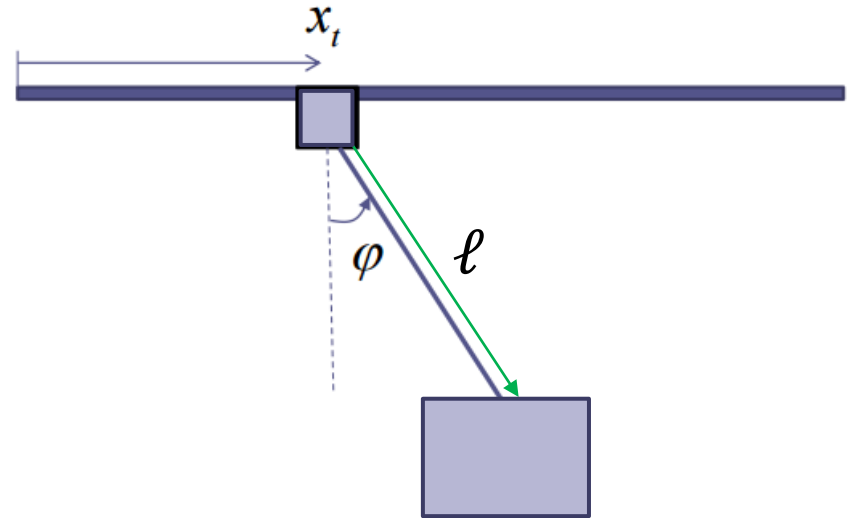


## SIMULAZIONE DI UNA GRU A PORTALE



## Variabili del sistema

- $x_t(t)$  Posizione del carrello  
 $\ell(t)$  Lunghezza della fune  
 $\varphi(t)$  Angolo fra la fune e la verticale  
 (positivo in senso antiorario)



## Ingressi esterni

- $F_1(t), F_2(t)$  Forze applicate al carrello ed alla fune

## Parametri

- $m, M$  masse del carrello e del container  
 $K_1, K_2$  coefficienti di attrito viscoso carrello e fune

## Ipotesi di progetto

La fune è sempre in tensione.

## Modello matematico

$$\begin{aligned}
 (M + m)\ddot{x}_t(t) + M\ddot{\ell}(t) \sin(\varphi(t)) + M\ell(t)\ddot{\varphi}(t) \cos(\varphi(t)) \\
 + K_1\dot{x}_t(t) + M(2\dot{\ell}(t)\dot{\varphi}(t) \cos(\varphi(t)) - \ell(t)\dot{\varphi}^2(t) \sin(\varphi(t))) = F_1(t) \\
 M\ddot{x}_t(t) \sin(\varphi(t)) + M\ddot{\ell}(t) + K_2\dot{\ell}(t) - M\ell(t)\dot{\varphi}^2(t) - Mg \cos(\varphi(t)) = F_2(t) \\
 \ddot{x}_t(t) \cos(\varphi(t)) + \ell(t)\ddot{\varphi}(t) + 2\dot{\ell}(t)\dot{\varphi}(t) + g \sin(\varphi(t)) = 0
 \end{aligned}$$

## Dati per la simulazione

### Parametri

$$m = 0.295 \text{ kg}$$

$$M = 1.8 \text{ kg}$$

$$K_1 = K_2 = 2 \text{ N s}$$

### Forze applicate (sistema di controllo in retroazione)

$$F(t) = \begin{bmatrix} F_1(t) \\ F_2(t) \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \cdot (x_t^d - x_t(t)) \\ 50 \cdot (\ell^d - \ell(t)) \\ 0 \end{bmatrix}$$

$$x_t^d = 5 \quad \ell^d = 2$$

**Condizioni iniziali tutte nulle eccetto  $\ell(0) = 1$**

## Modello matematico in forma vettoriale

$$q = \begin{bmatrix} x_t(t) \\ \ell(t) \\ \varphi(t) \end{bmatrix}$$

$$M(q)\ddot{q}(t) + H(q, \dot{q}) = F(t)$$

$$F(t) = \begin{bmatrix} F_1(t) \\ F_2(t) \\ 0 \end{bmatrix}$$



$$\ddot{q}(t) = M^{-1}(q)[F(t) - H(q, \dot{q})]$$

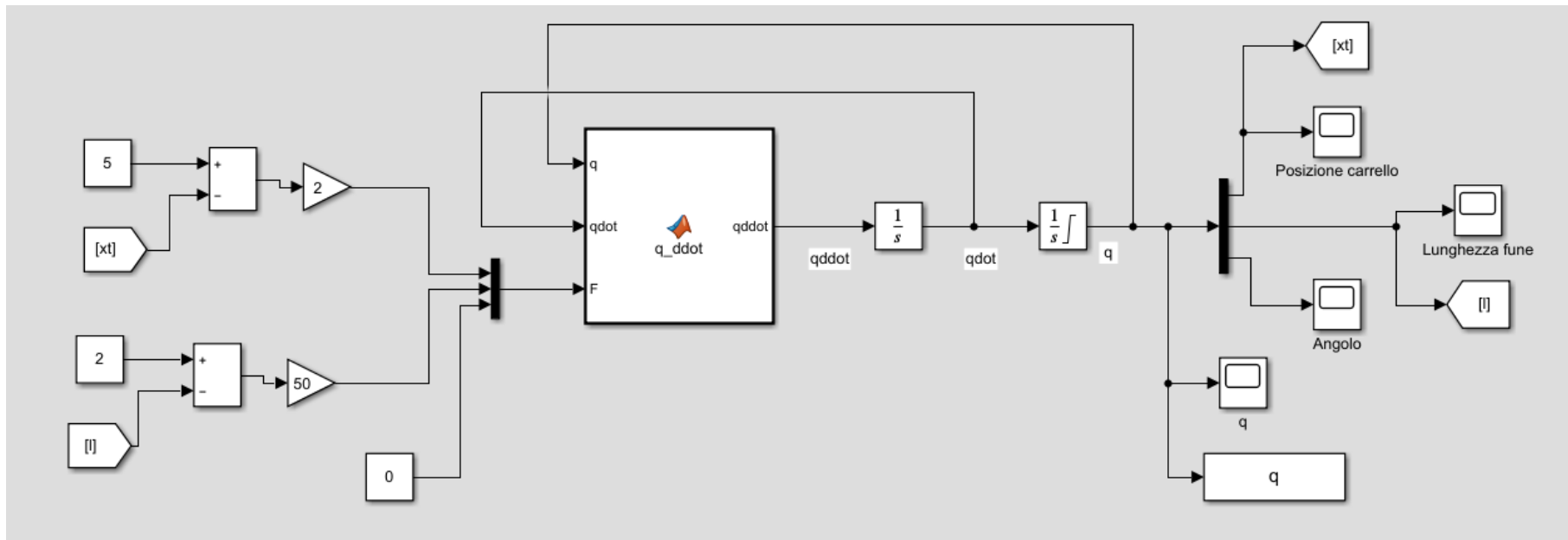
**Modello in forma  
esplicita**

$$M(q) = \begin{bmatrix} (M + m) & M \sin(\varphi(t)) & M\ell(t) \cos(\varphi(t)) \\ M \sin(\varphi(t)) & M & 0 \\ \cos(\varphi(t)) & 0 & \ell(t) \end{bmatrix}$$

Non singolare se  $\ell(t) > 0$

$$H(q, \dot{q}) = \begin{bmatrix} K_1 \dot{x}_t(t) + M(2 \dot{\ell}(t) \dot{\varphi}(t) \cos(\varphi(t)) - \ell(t) \dot{\varphi}^2(t) \sin(\varphi(t))) \\ K_2 \dot{\ell}(t) - M\ell(t) \dot{\varphi}^2(t) - Mg \cos(\varphi(t)) \\ 2\dot{\ell}(t) \dot{\varphi}(t) + g \sin(\varphi(t)) \end{bmatrix}$$

## Possibile realizzazione del modello Simulink



File: GruPortale2D\_2020.slx

```

function qddot = q_ddot(q,qdot,F)

M=1.8;
m=0.295;
k1=2;
k2=2;
g=9.81;


xt=q(1);
l=q(2);
fi=q(3);
xtdot=qdot(1);
ldot=qdot(2);
fidot=qdot(3);

A=[(M+m) M*sin(fi) M*l*cos(fi);
    M*sin(fi) M 0;
    cos(fi) 0 l];

H=[k1*xtdot+M*(2*ldot*fidot*cos(fi))+l*fidot^2*sin(fi);
    k2*ldot-M*l*fidot^2-M*g*cos(fi);
    2*ldot*fidot+g*sin(fi)];

qddot = inv(A)*(F-H);

```

 Block Parameters: Integrator ✕

Integrator

Continuous-time integration of the input signal.

Parameters

External reset:

Initial condition source:

Initial condition:

☒ Limit output

Upper saturation limit:

Lower saturation limit:

☐ Wrap state

☐ Show saturation port


☐ Show state port

Absolute tolerance:

☐ Ignore limit and reset when linearizing

☒ Enable zero-crossing detection

State Name: (e.g., 'position')

 OK Cancel Help Apply

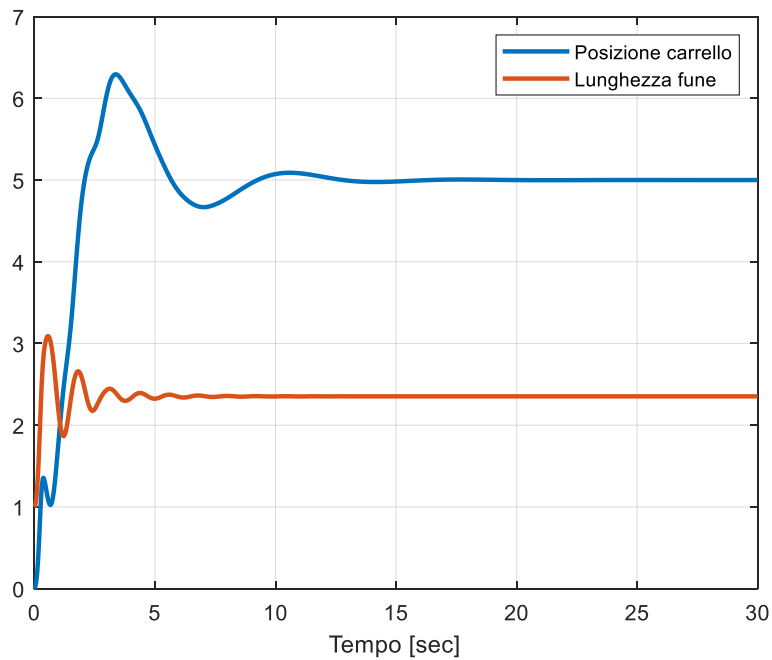
**Saturazione** dell'uscita  
dell'integratore

$$0 < \epsilon \leq \ell(t) \leq \infty$$

$$-\frac{\pi}{2} \leq \varphi(t) \leq \frac{\pi}{2}$$

## Risultati

Posizione del carrello e lunghezza della fune



Angolo

