Pendolo inverso

Modello matematico e controllo tramite pole placement

Lorenzo Nobili
Università Degli Studi Di Parma

January 13, 2025

Sommario

- Introduzione
- Modello matematico
  - Modello linearizzato
- Controllo del sistema
- Codice MATLAB e modello Simulink
- Risultati simulazione
- Implementazione in C
- Conclusioni

Introduzione

- **Strumentazione**
  - Base statica
  - Motore elettrico
  - Braccio rotante
  - Asta del pendolo
  - Sensori
- **Obiettivo**: definire un controllore che permetta di mantenere in equilibrio il pendolo, partendo da una posizione di equilibrio verticale.

Figure: Pendolo Inverso

Modello Matematico

- **Variabili**
  - θ_1: angolo tra base e braccio rotante
  - θ_2: angolo tra la verticale e l'asta del pendolo
  - τ_1: coppia sul braccio che ne permette la rotazione
  - τ_2: coppia sul pendolo che supporremo uguale a 0.
- **Obiettivo**
  - Trovare il modello del sistema (non lineare) e quello in forma di stato, necessario per il controllore

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

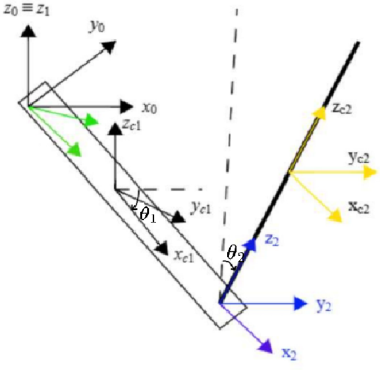


Figure: Disposizione delle terne

Modello Matematico

- **Derivazione del modello**
  - Approccio lagrangiano
    - Dobbiamo arrivare ad una formulazione di questo tipo

$$\tau_k = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k}$$

      con $q_1 = \theta_1,\ q_2 = \theta_2$ e $\tau_2 = 0$
    - Calcolare energia cinetica e potenziale per il braccio rotante e per il pendolo
    - Sommare i rispettivi contributi per ricavare l'energia cinetica e potenziale dell'intero sistema
    - Calcolo delll'equazione lagrangiana
    - Linearizzare il modello non lineare per ricavare la forma di stato

Modello Matematico

- Calcolo velocità angolari e lineari
  $\boldsymbol{\omega}_1 = \begin{bmatrix} 0 & 0 & \dot{\theta}_1 \end{bmatrix}^T$
  $\mathbf{v}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
  $\mathbf{v}_{1c} = \mathbf{v}_1 + \omega_1 \times \begin{bmatrix} l_1 & 0 & 0 \end{bmatrix}^T$
  $\boldsymbol{\omega}_2 = {}^1_2\mathbf{R}\omega_1 + \begin{bmatrix} \dot{\theta}_2 & 0 & 0 \end{bmatrix}^T$
  ${}^1v_2 = \boldsymbol{\omega}_1 \times \begin{bmatrix} L_1 & 0 & 0 \end{bmatrix}^T$
  ${}^2v_2 = {}^1_2\mathbf{R}\left(\boldsymbol{\omega}_1 \times \begin{bmatrix} L_1 & 0 & 0 \end{bmatrix}^T\right)$
  ${}^{c_2}v_{c2} = {}^2\mathbf{v}_2 + \omega_2 \times \begin{bmatrix} 0 & 0 & l_2 \end{bmatrix}^T$

- Calcolo energia cinetica e potenziale
  $E_{k1} = \frac{1}{2}\left(\mathbf{v}_{c1}^T m_1 \mathbf{v}_{c1} + \boldsymbol{\omega}_1^{T\,\mathbf{c1}}\mathbf{I}_{c1}\boldsymbol{\omega}_1\right)$
  $E_{k2} = \frac{1}{2}\left(\mathbf{v}_{c2}^T m_2 \mathbf{v}_{c2} + \boldsymbol{\omega}_2^{T\,\mathbf{c2}}\mathbf{I}_{c2}\boldsymbol{\omega}_2\right)$
  $E_{p1} = 0 \;\; E_{p2} = -m_2 \cdot \mathbf{g} \cdot \mathbf{p}_{c2}$
- Relazione tra τ_1 e v_{in}
  $\tau_1 = \tau_0 - \beta\dot{\theta}_1 - J_{eq}\ddot{\theta}_1$
  $\tau_0 = -\frac{\left(k_g k_l k_m \dot{\theta}_1 - v_{in}\right) k_g k_l k_m}{R_{eq}}$
- Modello di stato

$$\boldsymbol{x} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$\boldsymbol{u} = [v_{in}]$$

Modello linearizzato

- Dopo aver calcolato il modello non lineare, linearizziamo il sistema attorno allo stato di equilibrio $x_{eq} = [0, 0, 0, 0]$ utilizzando l'espansione di Taylor
- Abbiamo quindi il sistema nella seguente forma

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ B_{31} \\ B_{41} \end{bmatrix} v_{in}$$

- Con i valori numerici del sistema abbiamo

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 35.7650 & -56.51923 & 0 \\ 0 & 36.43262 & -19.45591 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 47.54433 \\ 16.36643 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Controllo del Sistema

- Per controllare il sistema, si è utilizzata la tecnica del **pole placement**
  - Assegnati i poli desiderati e supponendo che il sistema sia completamente controllabile,tale tecnica consiste nel porre come ingresso del sistema lo stato moltiplicato per un opportuna matrice **K**.
  - Tale matrice è calcolata attraverso la cosiddetta **formula di Ackermann** e permette di avere come poli del sistema quelli desiderati.

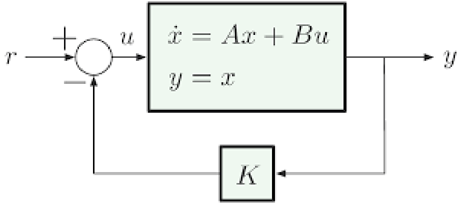
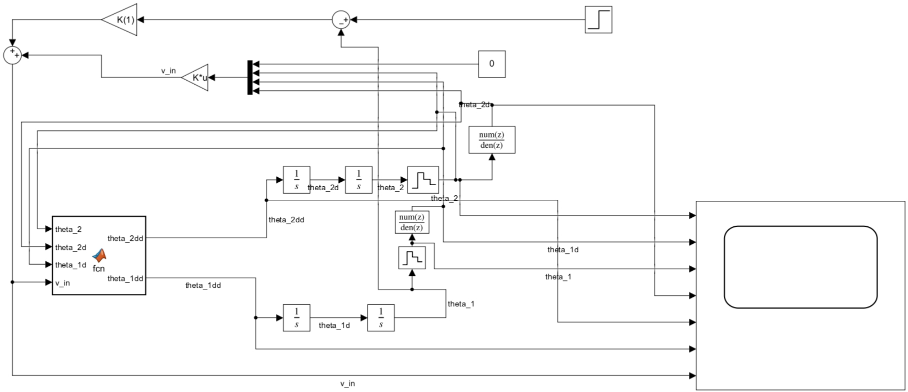


Figure: Pole placement

Codice MATLAB

```
clc;
clear;
close all;
a = 500; %filtro derivatore
A = [
    0, 0, 1, 0;
    0, 0, 0, 1;
    0, 35.7650, -56.51923, 0;
    0, 36.43262, -19.45591, 0
];
B = [
    0;
    0;
    47.54433;
    16.36643
];
C = [1 0 0 0;0 1 0 0];
D = [0,0]';
sys = ss(A,B,C,D);
theta_1sp = pi/4;
desired_poles = [-2.6+1.2*1i, -2.6-1.2*1i, -20, -25];
K = place(A, B, desired_poles);
% Discretizzazione del filtro derivatore
Ts = 0.0001;
num_d = [a -a];
den_d = [1 Ts*a-1];
```

Modello Simulink



Risultati simulazione

- Nella figura sottostante, sono rappresentati gli andamenti delle grandezze del sistema con un set point di θ_1 fissato a $\frac{\pi}{4}$.

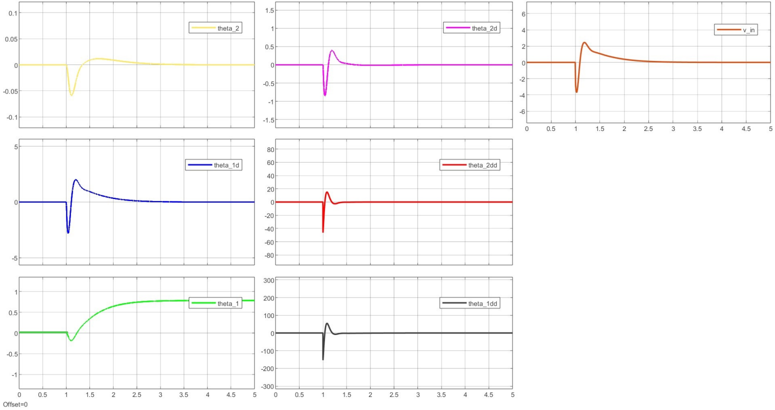


Figure: Risultati simulazione con $\theta_{1sp} = \frac{\pi}{4}$

Risultati simulazione

- Qui sono invece rappresentati i risultati della simulazione con un setpoint coincidente con un'onda quadra di ampiezza $\frac{\pi}{4}$

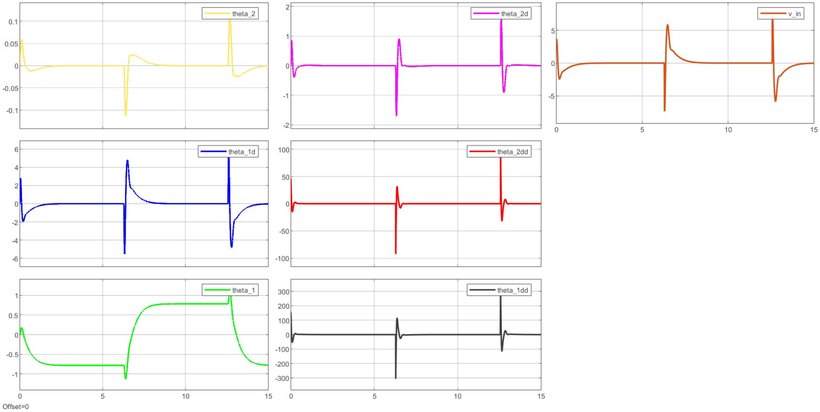


Figure: Risultati simulazione con θ_1 come onda quadra

Implementazione in C

- Di seguito l'implementazione in C del controllore che permette di stabilizzare il sistema

```
theta_1 = motor_absenc_pos_3;
    theta_1d = (1-Ts*a)*theta_1d_prev+a*theta_1-a*theta_1_prev;
    theta_1d_prev = theta_1d;
    theta_1_prev = theta_1;

    theta_2 = pendulum_absenc_pos_3;
    theta_2d = (1-Ts*a)*theta_2d_prev+a*theta_2-a*theta_2_prev;
    theta_2d_prev = theta_2d;
    theta_2_prev = theta_2;

    v_in = -K1*theta_1-K2*theta_2-K3*theta_1d-K4*theta_2d;
    volt_to_set = v_in;
```

Figure: Codice C del controllore

Conclusioni

- Il controllo tramite pole placement permette di stabilizzare efficacemente il sistema.
- Simulando il sistema reale, esso presenta un funzionamento coerente con le simulazioni, permettendo la stabilizzazione del pendolo lungo la verticale
- Nella simulazione reale rimangono piccole oscillazioni attorno all'equilibrio, dovuta alla natura dei poli complessi coniugati definiti.