Modellistica e controllo del pendolo inverso

Lorenzo Nobili

December 2024

1 Introduzione

La stabilizzazione del pendolo inverso rappresenta un tipico problema di controllo a scopo didattico. In questa relazione daremo dapprima una descrizione generale del sistema, per poi passare ad una modellizzazione formale di esso.
Definito il modello, passeremo in ambiente MATLAB e Simulink per verificare la validità del modello e definire il controllore specifico per simulare il sistema e confrontarne i risultati con quelli desiderati.
Nel nostro caso, abbiamo utilizzato la tecnica standard del pole placement per mantenere il pendolo in posizione di equilibrio.
Dopo aver ottenuto una simulazione coerente con il sistema modellato, passeremo poi in ambiente C per implementare effettivamente il controllore all'interno del sistema reale.

2 Descrizione del problema

Il sistema è rappresentato in figura 1. Abbiamo una base che sostiene il braccio rotante del sistema, il quale, a sua volta, sostiene il pendolo.
Il braccio rotante è alimentato da un motore in corrente continua dove, una parte di coppia di uscita, considerando gli attriti del sistema,sarà trasferita sul braccio rotante.
Definiamo

- τ_1: coppia sul braccio che ne permette la rotazione
- τ_2: coppia sul pendolo che ne permette la deviazione dall'equilibrio che supporremo uguale a 0.
- θ_1: angolo tra la terna disposta sul centro di massa del braccio rotante e la terna inerziale di riferimento, posta alla base del braccio.
- θ_2: angolo tra la verticale e l'asta del pendolo.

L'obiettivo è quello di stabilizzare il sistema in modo che il pendolo resti in equilibrio. Descriviamo adesso in maniera più formale il problema, definendo il modello matematico del sistema completo.
Il modello sarà inevitabilmente non lineare. Dovremo quindi, visto che il pole placement è una tecnica di controllo lineare, linearizzare il sistema attorno al punto di equilibrio. Ciò significa che dobbiamo ottenere un modello lineare del sistema in maniera da esprimerlo nella forma di stato

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases}$$

Figure 1: Pendolo inverso

3 Modello dinamico

Per ricavare il modello fisico del sistema utilizziamo l'approccio energetico di tipo lagrangiano. Come variabili generalizzate scegliamo i due angoli del sistema: θ_1 e θ_2. Dobbiamo arrivare, secondo Lagrange, ad una formulazione di questo tipo:

$$\tau_k = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} \tag{1}$$

con $\tau_2 = 0$ In figura 2 è rappresentata la rappresentazione schematica del sistema, con un assegnazione delle terne non convenzionale.

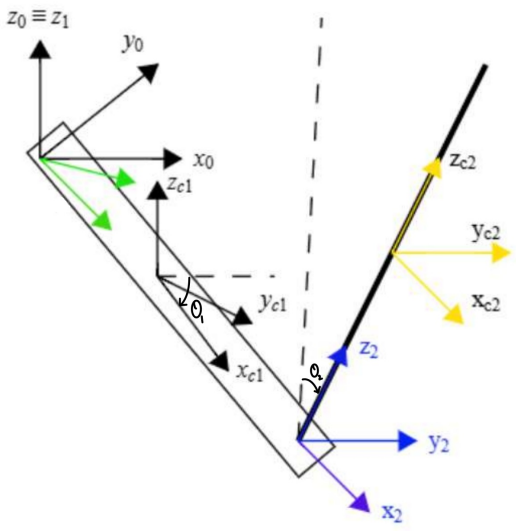

Figure 2: Rappresentazione schematica

3.0.1 Calcolo delle matrici di rotazione

Per definire il modello dobbiamo, come prima cosa, ricavare le matrici di rotazione che descrivono rispettivamente la terna 1 rispetto alla terna 0 e la terna 2 rispetto alla 1, ovvvero, formalmente, dobbiamo ricavare ${}^0_1\mathbf{R}$ e ${}^1_2\mathbf{R}$.
In particolare abbiamo:

$$ {}^0_1\mathbf{R} = \mathbf{R_z}(\alpha) \begin{pmatrix} \cos(\theta_1(t)) & -\sin(\theta_1(t)) & 0 \\ \sin(\theta_1(t)) & \cos(\theta_1(t)) & 0 \\ 0 & 0 & 1 \end{pmatrix} $$

$$ {}^1_2\mathbf{R} = \mathbf{R_x}(-\theta_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_2(t)) & \sin(\theta_2(t)) \\ 0 & -\sin(\theta_2(t)) & \cos(\theta_2(t)) \end{pmatrix} $$

3.0.2 Calcolo delle velocità lineari e angolari

La velocità angolare del braccio 1 è data da

$$\boldsymbol{\omega}_1 = \begin{bmatrix} 0 & 0 & \dot{\theta}_1 \end{bmatrix}^T$$

Considerando la terna 1 non baricentrica coincidente con la terna inerziale 0 abbiamo

$$\mathbf{v}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

La velocità lineare del centro di massa del braccio 1 ,applicando le regole di composizione delle velocita e ponendo ${}^1p_{c1,1} = \begin{bmatrix} l_1 & 0 & 0 \end{bmatrix}^T$ è quindi data da

$$\mathbf{v}_{1c} = \mathbf{v}_1 + \boldsymbol{\omega}_1 \times \begin{bmatrix} l_1 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & \dot{\theta}_1 l_1 & 0 \end{bmatrix}^T$$

La velocità angolare del pendolo rispetto alla terna inerziale è

$$\begin{aligned} \boldsymbol{\omega}_2 &= {}^1_2\mathbf{R}\boldsymbol{\omega}_1 + \begin{bmatrix} \dot{\theta}_2 & 0 & 0 \end{bmatrix}^T \\ &= \begin{bmatrix} \dot{\theta}_2 & \sin(\theta_2)\dot{\theta}_1 & \cos(\theta_2)\dot{\theta}_1 \end{bmatrix}^T \end{aligned}$$

La velocità del giunto alla base del pendolo rispetto alla terna 1 è data, secondo le regole di composizione delle velocità e definendo L_1 come la distanza tra la terna 0 e la terna 2, da

$${}^1v_2 = \boldsymbol{\omega}_1 \times \begin{bmatrix} L_1 & 0 & 0 \end{bmatrix}^T$$

Rispetto alla terna 2 è quindi

$${}^2v_2 = {}^1_2\mathbf{R}(\boldsymbol{\omega}_1 \times \begin{bmatrix} L_1 & 0 & 0 \end{bmatrix}^T) = \begin{bmatrix} 0, L_1\dot{\theta}_1\cos(\theta_2(t)), -L_1\dot{\theta}_1\sin(\theta_2(t)) \end{bmatrix}^T$$

Ponendo ${}^2p_{c2,2} = \begin{bmatrix} 0 & 0 & l_2 \end{bmatrix}^T$ abbiamo

$${}^{\mathbf{c_2}}\boldsymbol{v_{c2}} = {}^2\mathbf{v}_2 + \boldsymbol{\omega}_2 \times \begin{bmatrix} 0 & 0 & l_2 \end{bmatrix}^T = \begin{bmatrix} l_2\dot{\theta}_1\sin(\theta_2(t)) & L_1\dot{\theta}_1\cos(\theta_2(t)) - l_2\dot{\theta}_2 & -L_1\dot{\theta}_1\sin(\theta_2(t)) \end{bmatrix}^T$$

3.0.3 Calcolo della Lagrangiana e modello di stato

Considerando i tensori di inerzia dei due corpi nella seguente forma

$${}^{\mathbf{c_1}}\mathbf{I}_{c1} = \begin{bmatrix} I_{xx1} & 0 & -I_{xz1} \\ 0 & I_{yy1} & 0 \\ -I_{xz1} & 0 & I_{zz1} \end{bmatrix},$$

$${}^{\mathbf{c_2}}\mathbf{I}_{c2} = \begin{bmatrix} I_{xx2} & 0 & 0 \\ 0 & I_{yy2} & 0 \\ 0 & 0 & I_{zz2} \end{bmatrix}.$$

Abbiamo che l'energia cinetica per i due corpi è rispettivamente

$$E_{k1} = \frac{1}{2}\left(\mathbf{v}_{c1}^T m_1 \mathbf{v}_{c1} + \boldsymbol{\omega}_1^{T\,\mathbf{c1}}\mathbf{I}_{c1}\boldsymbol{\omega}_1\right) = \frac{1}{2}\dot{\theta}_1^2\left(m_1 l_1^2 + J_{1zz}\right)$$

$$\begin{aligned} E_{k2} =& \frac{1}{2}\left(\mathbf{v}_{c2}^T m_2 \mathbf{v}_{c2} + \boldsymbol{\omega}_2^{T\,\mathbf{c2}}\mathbf{I}_{c2}\boldsymbol{\omega}_2\right) \\ =& \frac{1}{2}\dot{\theta}_1^2\left(m_2 L_2^2 + \left(m_2 l_2^2 + J_{2yy}\right)\sin^2(\theta_2) + J_{2xx}\cos^2(\theta_2)\right) \\ &+ \frac{1}{2}\dot{\theta}_2^2\left(J_{2zz} + m_2 l_2^2\right) - m_2 L_1 l_2 \cos(\theta_2)\,\dot{\theta}_1\dot{\theta}_2. \end{aligned}$$

L'energia potenziale del primo corpo è nulla, quindi

$$E_{p0} = 0$$

Per il calcolo dell'energia potenziale del secondo corpo ci serve la posizione del baricentro rispetto alla terna inerziale, che è data da

$$p_{c2} = {}_1^0\mathbf{R}^1 p_{21} + {}_2^0\mathbf{R}^2 p_{c2,2} = \begin{bmatrix} l_2\sin(\theta_1)\sin(\theta_2) + L_1\cos(\theta_1) & -l_2\cos(\theta_1)\sin(\theta_2) + L_1\sin(\theta_1) & l_2\cos(\theta_2) \end{bmatrix}^T$$

Con cui ricaviamo l'energia potenziale del secondo corpo data da:

$$E_{p2} = -m_2 \cdot \mathbf{g} \cdot \mathbf{p}_{c2} = g l_2 m_2 \cos(\theta_2)$$

Utilizzando poi la Lagrangiana arriviamo al seguente modello

$$\begin{cases} L_1 l_2 m_2 {\dot{\theta}_2}^2 \sin(\theta_2(t)) - L_1 l_2 m_2 \ddot{\theta}_2 \cos(\theta_2(t)) + 2\left(l_2^2 m_2 + I_{yy_2} - I_{zz_2}\right)\dot{\theta}_1\dot{\theta}_2\cos(\theta_2(t))\sin(\theta_2(t)) \\ +\left(l_2^2 m_2 + I_{yy_2} - I_{zz_2}\right)\ddot{\theta}_1 \sin(\theta_2(t))^2 + \left(l_1^2 m_1 + L_1^2 m_2 + I_{zz_1} + I_{zz_2}\right)\ddot{\theta}_1 = \tau_1, \\ -L_1 l_2 m_2 \ddot{\theta}_1 \cos(\theta_2(t)) + \left(l_2^2 m_2 + I_{xx_2}\right)\ddot{\theta}_2 - \left(\left(l_2^2 m_2 + I_{yy_2} - I_{zz_2}\right){\dot{\theta}_1}^2 \cos(\theta_2(t)) + g l_2 m_2\right)\sin(\theta_2(t)) = 0 \end{cases}$$

Sapendo che la coppia di uscita τ_0 è

$$\tau_0 = -\frac{\left(k_g k_l k_m \dot{\theta}_1 - v_{in}\right) k_g k_l k_m}{R_{eq}}$$

e considerando gli attriti, avremo che la τ_1 sarà

$$\tau_1 = \tau_0 - \beta\dot{\theta}_1 - J_{eq}\ddot{\theta}_1$$

Ponendo come stato del sistema e ingresso del sistema

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

$$\boldsymbol{u} = \begin{bmatrix} v_{in} \end{bmatrix}$$

Possiamo, dopo aver risolto il sistema in funzione di $\ddot{\theta_1}$ e $\ddot{\theta_2}$ e linearizzato attorno allo stato nullo, porre il sistema nella seguente forma di stato

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

Sostituendo i valori numerici, abbiamo che le matrici A,B,C e D sono rispettivamente

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 35.7650 & -56.51923 & 0 \\ 0 & 36.43262 & -19.45591 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 47.54433 \\ 16.36643 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

3.1 Simulazione del modello

3.1.1 Controllo tramite pole placement

Prima di simulare il modello su Simulink, dobbiamo definire il controllo che ci permette di stabilizzare il sistema.
Il controllo che utilizzeremo è il **pole placement**. Tale tecnica consiste nel porre come ingresso del sistema lo stato moltiplicato per un opportuna matrice **K**. Tale matrice si può calcolare, dopo aver assegnato correttamente i poli del sistema, attraverso la funzione place di matlab.
Nel nostro caso, i poli selezionati sono: $-2.6 + 1.2i, -2.6 - 1.2i, -20, -25$.
I poli complessi coniugati influenzano la risposta oscillatoria del sistema, mentre quelli reali negativi influenzano la velocità con cui il sistema converge al regime stazionario.

3.1.2 Codice MATLAB

Nel codice MATLAB sottostante, si sono innanzitutto definite le matrici di stato del sistema. Successivamente, si sono scelti i poli per calcolare la matrice **K**.
Il nostro sistema di controllo reale è,ovviamente, a tempo discreto. Abbiamo quindi la necessità di discretizzare lo stato del sistema.
In particolare, deve essere discretizzati i filtri derivatori che mi permettono di calcolare $\dot{\theta_1}$ e $\dot{\theta_2}$.
Più precisamente dobbiamo discrettizare la seguente funzione di trasferimento che descrive un filtro derivatore generico

$$H(s) = \frac{as}{s+a}, a > 0$$

Come metodo di discretizzazione è stato usato il metodo delle differenze in avanti che consiste nel fare la sostituzione

$$s = \frac{z-1}{T_s}$$

Che risulta nella seguente funzione di trasferimento discreta

$$H(z) = \frac{az-a}{T_s a}$$

```
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];
```

Listing 1: codice MATLAB

3.1.3 Simulazione

Simuleremo prima il sistema con un setpoint di θ_1 fissato (nel nostro caso $\theta_1 = \frac{\pi}{4}$), per poi passare ad un setpoint variabile definito tramite un onda quadra. In figura 3 è rappresentato il modello Simulink del sistema. All'interno della MATLAB function è inserito il modello di stato non lineare del sistema, per calcolare le accelerazioni.In figura 4,invece, sono rappresentati i risultati della simulazione

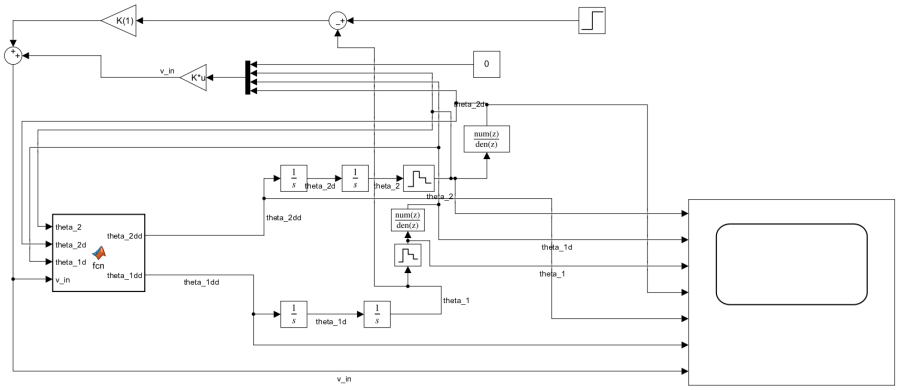

Figure 3: Modello Simulink

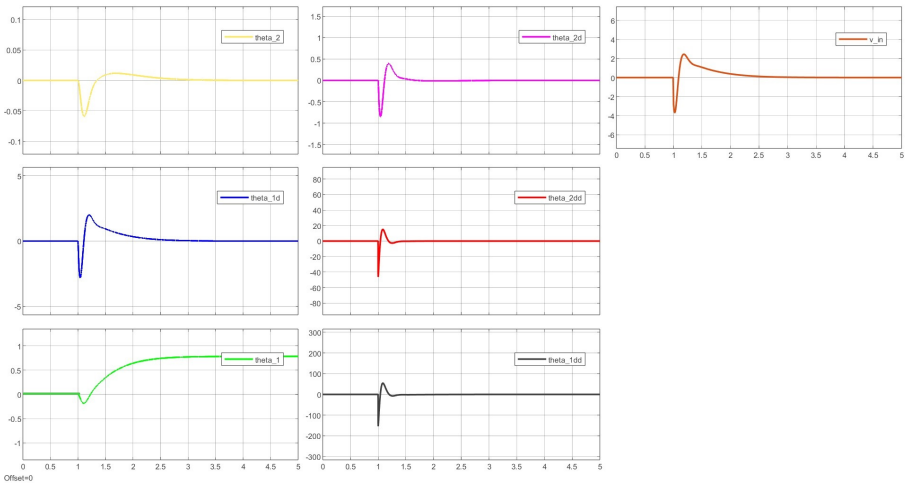

Figure 4: Risultati della simulazione

I risultati sono coerenti con quello che vogliamo ottenere. Il θ_1 converge,come desiderato, a $\frac{\pi}{4}$, mentre le velocità e le accelerazioni convergono correttamente a 0. In figura 5 è rappresentata la simulazione con un onda quadra di ampiezza $\frac{\pi}{4}$ e frequenza 0.5 rad/s.

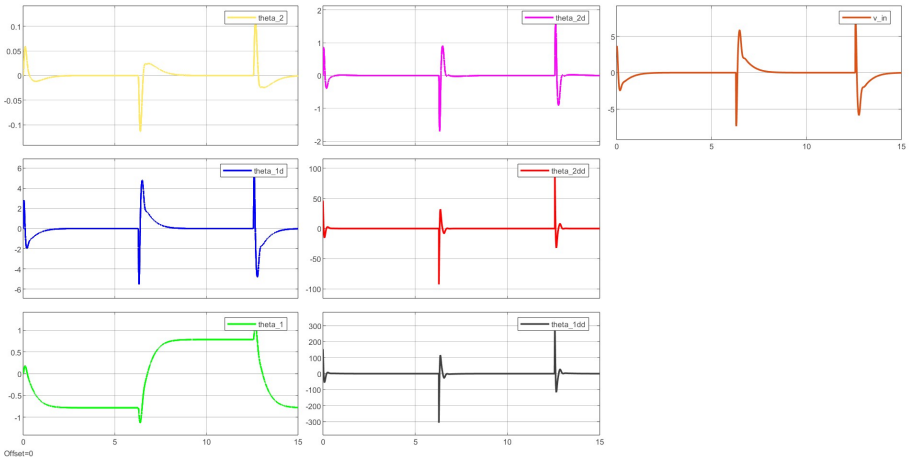

Figure 5: Simulazione con onda quadra

Anche qui i risultati sono coerenti con quando desiderato. Il sistema si stabilizza prima su $\frac{-\pi}{4}$ e poi su $\frac{\pi}{4}$.

4 Implementazione in C

Per implementare il codice in C, serve osservare lo stato corrente e aggiornare lo stato al passo precedente, calcolare $\dot{\theta_1}$ e $\dot{\theta_2}$ attraverso il filtro discretizzato e,infine, definire l'ingresso controllato attraverso i guadagni della matrice **K**.

```
            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;
```

Listing 2: Implementazione in C

5 Conclusioni

Simulando il sistema reale, il sistema si stabilizza correttamente sul setpoint definito. I risultati della simulazione Simulink sono approssimabili con la simulazione reale. L'unica differenza è che, nel simulink, non viene visualizzata la leggera oscillazione attorno all'equilibrio che è invece presente nel sistema reale, a causa dei poli complessi coniugati.