

MATLAB - Esercizio 3 (Sezione 1)

Si consideri il sistema a tempo continuo descritto da

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad A = \begin{bmatrix} 0 & -1 & 2 \\ 2 & -3 & 12 \\ 10 & -10 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, \quad D = 0$$

1. Determinare l'equilibrio corrispondente a $u(t) = \bar{u} = 1$ e studiarne la stabilità.
2. Simulare il movimento libero del sistema con stato iniziale $x(0) = [0 \ 1 \ 1]^T$ su un opportuno intervallo di tempo.
3. Simulare il movimento forzato del sistema con ingresso $u(t) = \bar{u} = 1$.
4. Simulare il movimento del sistema con $u(t) = \bar{u} = 1$ e stato iniziale $x(0) = [0 \ 1 \ 1]^T$.
5. Verificare che stato e uscita convergono verso i valori di equilibrio.
6. Verificare che il movimento complessivo è la somma di movimento libero e forzato.

Traccia di soluzione

- Introdurre le matrici A, B, C, D e definire il sistema con la funzione `ss`
- Calcolare lo stato di equilibrio $\bar{x} = -A^{-1}b$ e l'uscita di equilibrio $\bar{y} = C\bar{x}$
- Calcolare la parte reale degli autovalori della matrice A con le funzioni `eig` e `real` e verificare il risultato calcolando le radici del polinomio caratteristico con le funzioni `poly` e `roots`
- Simulare il sistema con le funzioni `initial`, `step` e `lsim` dopo aver definito i vettori t , u , e x_0 (il movimento al punto 3. può essere simulato utilizzando il comando `step` in quanto l'ingresso è costante)
- Usando le funzioni `max` e `abs` calcolare il massimo scostamento tra y e la somma $y_l + y_f$
- Per rappresentare più curve sovrapposte sullo stesso grafico si può usare il comando `hold on` dopo il primo grafico per "bloccare" il grafico e poi ripetere il comando di `plot` quante volte si vuole. Al termine si "sblocca" il grafico con il comando `hold off`.

Soluzione

```
» A = [0 -1 2; 2 -3 12; 10 -10 -2]; B = [1 2 1]'; C = [1 0 1];
» sistc = ss(A,B,C,0);
» %%% Punto 1
» ubar = 1;
» xbar = -inv(A)*B;
» ybar = C*xbar;
» %%% Punto 2
» t = 0:0.01:5; u = ones(size(t)); x0 = [0 1 1]';
» [yl,tl,xl] = initial(sistc,x0,t);
» figure(1); plot(tl,xl); grid on
» figure(2); plot(tl,yl); grid on
» %%% Punto 3
» [yf,tf,xf] = step(sistc,t);
» figure(1); plot(tf,xf); grid on
» figure(2); plot(tf,yf); grid on
» %%% Punto 4
» [yt,tt,xt] = lsim(sistc,u,t,x0);
```

```

» figure(1); plot(tt,xt); grid on
» figure(2); plot(tt,yt); grid on
» %%% Punto 5
» figure(1)
» plot(t,yt)
» hold on
» plot(t,ybar*ones(size(t)),'k:')
» legend('y(t)','ybar')
» figure(2)
» plot(t,xt)
» plot(t,xbar(1)*ones(size(t)),'k:')
» plot(t,xbar(2)*ones(size(t)),'k:')
» plot(t,xbar(3)*ones(size(t)),'k:')
» %%% Punto 6
» plot(t,yt,'b',t,yl+yf,'r')
» errmax = max(abs(yt-(yl+yf)))

```