Modelare şi Simulare

Proiect 2023 – Modelele 1, 2, 3, 4 şi cerinţele asociate

Data ultimei actualizări: 12 octombrie 2023

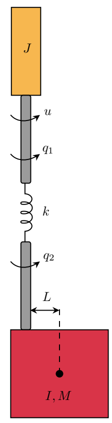
1 Alegerea modelului

Pentru realizarea proiectului de la MS şi a raportului final trebuie să alegeţi **un singur model** din cele prezentate în această secţiune. Alegeţi parametrii modelului arbitrar, ţinând cont că aceste constante trebuie să fie pozitive. Totodată, alegeţi factorul de amortizare $\zeta \in (0.6, 0.95)$.

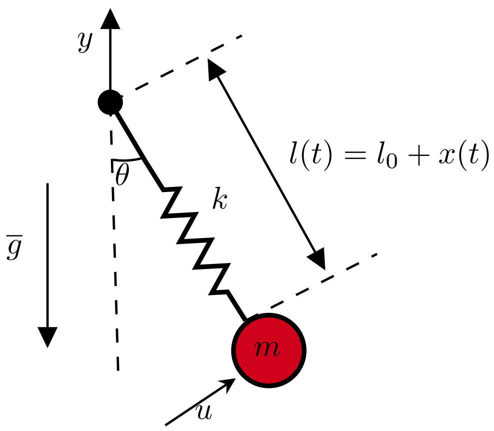
Modelul 1 – Manipulator robotic

Modelul matematic este descris de ecuaţiile:

$$J\ddot{q}_1(t) - k\cdot(q_2(t) - q_1(t)) + \zeta\dot{q}_1 = u(t) \quad (1a)$$
$$I\ddot{q}_2(t) + MgL\sin(q_2(t)) + k\cdot(q_2(t) - q_1(t)) + \zeta\dot{q}_2 = 0 \quad (1b)$$
$$y_1(t) = q_1(t) \quad (1c)$$
$$y_2(t) = q_2(t) \quad (1d)$$



Modelul 2 – Pendulul elastic



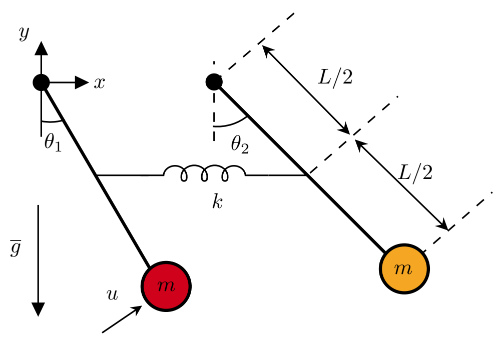
Modelul matematic este descris de ecuaţiile:

$$\ddot{x}(t) = (\ell_0 + x(t))\dot{\theta}^2(t) - \frac{k}{m}x(t) + g\cos\theta(t) - \zeta\dot{x}(t) \quad (2a)$$
$$\ddot{\theta}(t) = -\frac{g}{\ell_0 + x(t)}\sin\theta(t) - \frac{2\dot{x}(t)}{\ell_0 + x(t)}\dot{\theta}(t) - \zeta\dot{\theta}(t) + \quad (2b)$$
$$+\frac{1}{m(\ell_0 + x(t))^2}u(t)$$
$$y_1(t) = \theta(t) \quad (2c)$$
$$y_2(t) = x(t) \quad (2d)$$

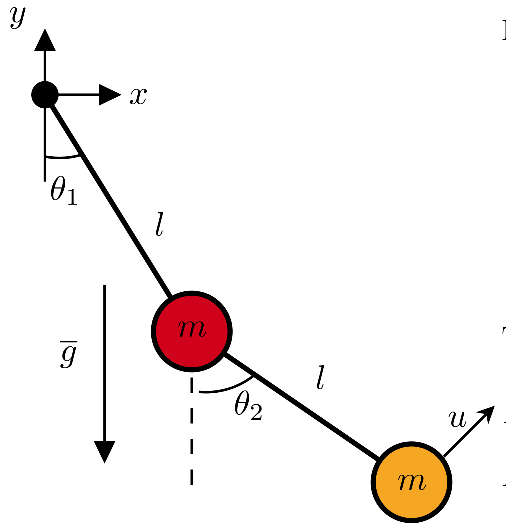
Model 3 – Pendule cuplate

Modelul matematic este descris de ecuaţiile:

$$u(t) = ML^2\ddot{\theta}_1(t) + MgL\sin\theta_1(t) \quad (3a)$$
$$+\frac{1}{4}kL^2\left(\sin\theta_1(t) - \sin\theta_2(t)\right)\cos\theta_1(t) + \zeta\dot{\theta}_1(t)$$
$$0 = ML^2\ddot{\theta}_2(t) + MgL\sin\theta_2(t) \quad (3b)$$
$$-\frac{1}{4}kL^2\left(\sin\theta_1(t) - \sin\theta_2(t)\right)\cos\theta_2(t) + \zeta\dot{\theta}_2(t)$$
$$y_1(t) = \theta_1(t) \quad (3c)$$
$$y_2(t) = \theta_2(t) \quad (3d)$$



Model 4 – Pendulul dublu



Modelul matematic este descris de ecuaţiile:

$$\ddot{\theta}_1(t) = \frac{B\cos\left(\theta_1(t) - \theta_2(t)\right) - A}{-\cos 2\left(\theta_1(t) - \theta_2(t)\right)} \tag{4a}$$

$$\ddot{\theta}_2(t) = \frac{2B - A\cos\left(\theta_1(t) - \theta_2(t)\right)}{\cos 2\left(\theta_1(t) - \theta_2(t)\right)} \tag{4b}$$

$$y_1(t) = \theta_1(t) \tag{4c}$$

$$y_2(t) = \theta_2(t) \tag{4d}$$

Termenii A şi B sunt:

$$A = -\dot{\theta}_2^2(t)\sin(\theta_1(t) - \theta_2(t)) - 2\frac{g}{\ell}\sin(\theta_1(t)) - \zeta\dot{\theta}_1(t) + \frac{1}{m\ell^2}u(t)$$

$$B = \dot{\theta}_1^2(t)\sin(\theta_1(t) - \theta_2(t)) - \frac{g}{\ell}\sin\theta_2(t) - \zeta\dot{\theta}_2(t).$$

2 Cerinţe

[3 p.] 1. Implementarea modelului matematic neliniar:

[1.5 p.] (a) Utilizați blocul `Matlab function` pentru a implementa modelul într-o schemă Simulink.

[1.5 p.] (b) Utilizați blocul `Fcn` pentru a implementa modelul într-o schemă Simulink.

În raport, atașați o captură de ecran cu modelul și includeți codul folosit în blocurile `Matlab function` și `Fcn`.

[0.5 p.] 2. Crearea unui Semnal de Treaptă:

- Într-un script `Matlab`, creați un semnal de treaptă pe un orizont de timp extins, asigurându-vă că sistemul ajunge în regimul staționar. Salvați semnalul într-o variabilă `timeseries` numită `usim`.
- Specificaţi în raport orizontul de timp ales (ex., $[0, 30]$ secunde).

[1 p.] 3. Simularea Modelului:

- Simulați atât modelul implementat cu `Matlab function`, cât și cel implementat cu blocul `Fcn`, folosind semnalul de treaptă creat anterior. Utilizați blocurile `To Workspace` și `From Workspace` pentru a gestiona datele. Salvați răspunsurile în `simout_mat1`, `simout_mat2`, `simout_fcn1`, și `simout_fcn2`.
- Ilustraţi în raport ieşirile.

[0.5 p.] 4. Evaluarea Erorii:

- Calculați norma 2 a diferenței dintre răspunsurile obținute și salvați rezultatele în variabilele `err1` și `err2`.
- În raport, comentați așteptările voastre și rezultatele obținute, explicând cauzele discrepanțelor, dacă există.

[4 p.] 5. Ilustrați caracteristica statică de funcționare a modelului matematic folosind **doar** blocul `Matlab function` parcurgând următorii pași:

[1 p.] (a) Generați **cel puțin** 10 intrări de tip treaptă (aveți grijă ca amplificările treptelor sa nu fie foarte apropiate). Stocați amplificările treptelor într-un vector denumit `ustar`.

[1 p.] (b) Simulați modelul pentru fiecare intrare și memorați valorile de regim staționar ale ieșirilor în vectorii `y1star` şi `y2star`.

[1 p.] (c) Folosind funcția `polyfit` calculaţi coeficienții polinomului care aproximează în sensul celor mai mici pătrate relația $y^\star = y^\star(u^\star)$; practic, pentru cele două ieșiri veți calcula două polinoame `p1` şi `p2`, corespunzătoare relațiilor $y_1^\star = y_1^\star(u^\star)$ și $y_2^\star = y_2^\star(u^\star)$.

[1 p.] (d) Ilustrați, pe același grafic, perechile de puncte $(u^\star, y^\star)$ obținute la pct. (b) și graficele celor două polinoame obținute la pct. (c).

[1 p.] 6. Aproximarea răspunsurilor:

- Alegeţi valori rezonabile pentru scalarii α, β şi γ (diferite de cele care apar în `ustar`).
- Folosind polinoamele `p1` și `p2`, și funcția `polyval`, aproximaţi răspunsurile în regim staționar pentru intrările $u_1^\star = \alpha,\ u_2^\star = \beta,\ u_3^\star = \gamma$.
- Notaţi în raport valorile scalarilor şi a răspunsurilor obţinute.

[1 p.] 7. Utilizând acelaşi model proiectat la punctul 1, construiţi un model Simulink cu blocuri `In` şi `Out`, luând în calcul numai ieşirea `y_1`.

[1.5 p.] 8. Folosind funcția `trim` şi modelul de la punctul anterior, determinați un punct static de funcționare (x^*, u^*, y^*) în jurul punctului[1] u_0, astfel încât eroarea $|u^* - u_0|$ să fie minimă[2]. Salvați valorile în variabilele `xstar`, `ustar`, `ystar`.

[1 p.] 9. Determinaţi sistemul de spaţiul stărilor, definit de cvadruplul matriceal, ca urmare a liniarizării în punctul static de funcționare (x^*, u^*, y^*). Salvați matricele rezultate în variabilele `A_lin`, `B_lin`, `C_lin` și `D_lin`.

[1 p.] 10. Verificați dacă sistemul liniarizat este stabil cu ajutorul funcției `eig` și salvați spectrul matricei `A_lin` în vectorul `vp`.

[1 p.] 11. În `Simulink`, calculați răspunsul în timp al sistemului liniarizat:

$$\dot{\Delta x}(t) = A_{lin}\Delta x(t) + B_{lin}\Delta u(t), \tag{5a}$$
$$\Delta y(t) = C_{lin}\Delta x(t) + D_{lin}\Delta u(t)\ , \tag{5b}$$

la intrarea treaptă[1] $r \cdot \mathbf{1}(t)$ şi salvaţi rezultatul în variabila `y_lin` de tip `timeseries`.

[1 p.] 12. Calculaţi răspunsul sistemului neliniar la intrarea treaptă $r \cdot \mathbf{1}(t)$ și salvați-l în variabila `y_nl` de tip `timeseries`, apoi determinați eroarea de liniarizare dintre `y_nl` şi `y_lin`. Stocaţi în `err` norma infinit a diferenței dintre răspunsul sistemului neliniar și cel al sistemului liniar, i.e., $\|y_{nl} - y_{lin}\|_\infty$.

[1 p.] 13. Discretizaţi modelul liniarizat folosind funcţia `c2d`. Justificaţi alegerea metodei de discretizare într-un comentariu în raport. Alegeţi o perioadă de eşantionare T_e care să respecte inegalitatea $0 < T_e \leq 0.1$, unde $T_e \in \mathbb{R}$.

[1.5 p.] 14. Scrieţi în raport ecuaţia cu diferenţe rezultată din aplicarea inversei transformatei Z pe sistemul discretizat şi apoi implementaţi ecuaţia într-un bloc `MATLAB function`.
Indicaţie: folosiţi variabile de tip `persistent` pe care să le iniţializaţi cu 0.

[1 p.] 15. Comparaţi grafic răspunsul sistemului neliniar cu cel al sistemului discret.

[1] Valorile u_0 şi r se pot alege din caracteristica statică a sistemului.
[2] În cazul în care funcţia `trim` nu returnează un rezultat satisfăcător, folosiţi o metodă alternativă de a determina punctul de echilibru.

Recomandări:

1. Etichetaţi corespunzător graficele:
   - folosiţi comanda `xlabel` pentru a seta denumirea axei *Ox* şi specificaţi unitatea de măsură (dacă există) – ex: `xlabel('Timp (s)');`
   - folosiţi comanda `ylabel` pentru a seta denumirea axei *Oy* şi specificaţi unitatea de măsură (dacă există) – ex: `ylabel('Theta (rad)');`
   - folosiţi comanda `title` pentru a seta titlul figurii;
   - adăugaţi legenda folosind comanda `legend`;

   Lipsa etichetelor aduce o depunctare de 10% puncte din punctajul asociat cerinţei.

2. Delimitaţi cerinţele în scriptul `Matlab` cu `%%Cerinţa x` pentru a crea secţiuni care pot fi executate independent.

3. Adăugaţi în raport graficele şi fragmentele de cod esenţiale cerute în fiecare exerciţiu. Oferiţi explicaţii scurte şi concrete acolo unde este necesar.

4. Salvaţi fişierele într-o arhivă care să respecte formatul: `Nume_Prenume_Grupa_Proiect.zip`.
   Exemplu: `Ionescu_Ion-Constantin_335AB_Proiect.zip`