

# Control prin învățare (Learning Control)

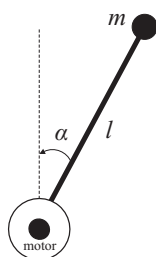
## Laborator 3: Metode offline cu aproximare

**Regulament** – același ca și la laboratorul precedent, cu excepția termenului limită pentru predarea soluției, care este **10 mai 2018**.

### Pendulul inversat

În acest laborator vom considera problema pendulului inversat. Acest sistem constă dintr-o masă  $m$  atașată unui braț ce se rotește în plan vertical și este acționat de un motor, vezi figura. Obiectivul este ca masa să fie adusă și stabilizată orientată în sus, pornind de la orice poziție și viteză inițială. Puterea motorului este însă insuficientă pentru a face acest lucru într-o singură rotație din toate stările inițiale. Din anumite stări, cum ar fi masa orientată în jos, pendulul trebuie acționat într-o mișcare oscilantă pentru a acumula suficientă energie, înainte de a fi împins în sus și stabilizat.

Modelul de timp continuu al pendulului este:



$$\ddot{\alpha} = 1/J \cdot [mgl \sin(\alpha) - b\dot{\alpha} - K^2\dot{\alpha}/R + Ku/R]$$

unde  $J = 1.91 \cdot 10^{-4} \text{ kgm}^2$ ,  $m = 0.055 \text{ kg}$ ,  $g = 9.81 \text{ m/s}^2$ ,  $l = 0.042 \text{ m}$ ,  $b = 3 \cdot 10^{-6} \text{ Nms/rad}$ ,  $K = 0.0536 \text{ Nm/A}$ ,  $R = 9.5 \Omega$ . Starea este  $x = [\alpha, \dot{\alpha}]^T$ . Unghiul  $\alpha$  variază în intervalul  $[-\pi, \pi]$  rad, cu  $\alpha = 0$  însemnând prin convenție că pendulul este orientat în sus. Tot prin convenție, unghiul se rotește în acest interval în așa fel încât, de exemplu, o rotație de  $3\pi/2$  corespunde la  $\alpha = -\pi/2$ . Viteza unghiulară  $\dot{\alpha}$  este restricționată prin saturație la intervalul

$[-15\pi, 15\pi]$  rad/s. Sistemul în timp discret este obținut integrând numeric dinamica continuă din ecuația de mai sus, de-a lungul eșantioanelor de timp. Acțiunea (voltajul motorului) este limitată la  $[-3, 3]$  V, insuficientă pentru a împinge pendulul în sus într-o singură rotație.

Obiectivul este stabilizarea pendulului în echilibrul instabil  $x = 0$  (orientat în sus), și este exprimat de funcția de recompensă:

$$r = -x^T Q_{\text{rew}} x - R_{\text{rew}} u^2, \quad \text{unde: } Q_{\text{rew}} = \text{diag}[5, 0.1], \quad R_{\text{rew}} = 1$$

Matricea  $Q_{\text{rew}}$  penalizează valorile diferite de zero ale celor două stări într-o măsură similară, date fiind amplitudinile lor diferite, iar  $R_{\text{rew}}$  penalizează consumul de energie, într-o măsură mai mică decât stările. Factorul de discount este  $\gamma = 0.98$ .

Codul ce implementează modelul de simulare al pendulului inversat și formează baza laboratorului poate fi descărcat de la adresa:

<http://busoniu.net/teaching/ci2018>

secțiunea "Practical assignments". Dezarhivați codul într-un director de pe calculatorul dvs., navigați din MATLAB în acest director și rulați scriptul `startup`. Codul poate fi acum folosit.

### Tema de laborator

Vom investiga comportamentul a doi algoritmi reprezentativi de programare dinamică aproximată: iterația fuzzy Q și iterația pe legea de control cu CMMP (LSPI). Implementarea ambilor algoritmi este deja furnizată.

## Partea 1. Acuratețea aproximatorului în iterația fuzzy Q

În această parte, vom investiga efectul unui element crucial în DP și RL cu aproximare: acuratețea aproximatorului. Fișierul `adp_example` ilustrează cum se poate rula iterația fuzzy Q folosind un interpolator pe o grilă echidistantă de  $N \times N$  puncte în spațiul stărilor, și o discretizare echidistantă cu  $M$  elemente în spațiul acțiunilor; precum și cum se poate extrage și studia soluția obținută, în forma unui aproximator generic. Acest aproximator poate fi folosit pentru a calcula valori Q și acțiuni, precum și pentru a reprezenta grafic funcția de valoare și legea de control. Fișierul ilustrează cum se poate simula o traiectorie controlată a sistemului folosind soluția calculată.

În plus, exemplul arată cum o soluție *aproape optimală* – funcție Q și lege de control – poate fi în mod similar extrasă dintr-un fișier de date deja furnizat, și inspectată. Pentru simplitate, vom nota această soluție cu  $Q^*$ ,  $h^*$  (chiar dacă nu este exact optimală).

**Cerințele sunt:**

1. Rulați iterația fuzzy Q pentru numărul  $N$  de puncte pe grila din spațiul stărilor variind în secvența 5, 11, 15, 21. Ceilalți parametri (numărul de acțiuni discrete  $M$ , pragul de convergență etc.) pot fi păstrați la valorile din exemplu. **[0.5p]**
2. Studiați soluția rezultantă: funcții de valoare, legi de control, și traiectorii controlate din starea inițială  $[-0.95\pi, 0]^T$  (pendulul orientat aproape în jos), pentru aceste valori ale lui  $N$ . Reprezentați grafic evoluția cu  $N$  a distanței între soluția  $\hat{Q}$  obținută de algoritm și  $Q^*$ , precum și între legea de control corespunzătoare  $\hat{h}$  și legea optimală  $h^*$ . Discutați rezultatele. **[2.5p]**

Pentru a calcula distanța între funcții Q, folosiți o grilă cu  $K \times K \times M$  puncte în spațiul stare-acțiune, unde  $K$  este o valoare liber aleasă (de exemplu în intervalul 20-50), iar  $M$  numărul de acțiuni discrete ales mai sus. De notat că aproximatorul nu poate calcula valori Q pentru acțiuni care nu fac parte din discretizare, așadar trebuie să ne limităm la discretizarea aleasă la pasul precedent. Distanța este media diferențelor pătratice între  $\hat{Q}$  și  $Q^*$  pentru punctele de pe grilă:

$$\frac{1}{K^2 M} \sum_{(x,u) \in \text{grilă}} |\hat{Q}(x,u) - Q^*(x,u)|^2$$

Distanța între legile de control poate fi calculată similar, pentru o grilă de  $K \times K$  stări (de notat că numărul de puncte pe această grilă 2D este mai mic decât pe grila 3D folosită pentru funcția Q).

3. Reprezentați grafic evoluția timpului de execuție cu  $N$ , și discutați rezultatul. **[1p]**
4. Repetați experimentul de la punctele 1–2, dar de această dată variați  $N$  mai fin, folosind valorile 15, 16, 17, ..., 20. Are evoluția calității soluției aceeași natură ca în cazul precedent? Puteți identifica motivele? **[1p]**

## Partea 2. Efectul setului de date asupra LSPI

Diferența esențială între LSPI și iterația fuzzy Q este că LSPI nu folosește modelul sistemului, ci este bazat pe date. Vom studia aici efectul numărului de eșantioane folosit pentru a învăța soluția cu LSPI. Același fișier ca și mai sus, `adp_example`, arată cum se rulează algoritmul cu o grilă echidistantă de  $N \times N$  funcții radiale de bază, și o discretizare echidistantă cu  $M$  elemente în spațiul acțiunilor; cum se

poate extrage aproximatorul rezultat; și cum se poate simula o traiectorie controlată a sistemului folosind soluția calculată. Setul de date este generat uniform aleator în spațiul stare-acțiune, unde stările sunt continue și acțiunile sunt discretizate, iar numărul de eșantioane poate fi configurat.

**Cerințele** sunt:

1. Rulați LSPI pentru numărul  $N_S$  de eșantioane variind în setul 2500, 5000, 7500, 10000. Ceilalți parametri (numărul de funcții de bază  $N$ , de acțiuni discrete  $M$ , pragul de convergență etc.) pot fi păstrați la valorile din exemplu. Pentru a ne asigura că algoritmul primește cel puțin aceleași informații când creștem numărul de eșantioane, este preferabil ca eșantioanele să fie generate în așa fel încât, de exemplu, setul de 5000 să includă setul de 2500. **[0.5p]**
2. Studiați soluția rezultantă: funcții de valoare, legi de control, și traiectorii controlate din starea inițială  $[-0.95\pi, 0]^T$ , pentru aceste valori ale lui  $N_S$ . Reprezentați grafic evoluția cu  $N_S$  a returnului obținut din această stare inițială, a numărului de iterații rulate, și a timpului de execuție. Discutați rezultatele. **[2p]**
3. Comparați cele mai bune soluții (care au cel mai mare return) obținute cu LSPI și iterația fuzzy Q, din punctul de vedere al calității și al timpului de execuție necesar. **[0.5p]**

De notat că LSPI poate intra în regim oscilatoriu sau să nu convergă. De obicei acest comportament se poate evita încercând generarea unui set de eșantioane mai informativ (fie mai mare, fie diferit).

În raport, descrieți pe scurt problemele pe care le-ați rezolvat, și mai în detaliu studiile cerute, incluzând grafice reprezentative / importante (nu toate graficele obținute). Nu uitați să includeți separat listingul codului dezvoltat.

(Pentru exemple suplimentare de algoritmi cu aproximare, puteți descărca toolboxul complet de DP și RL cu aproximare, de la adresa [busoniu.net/repository.php](http://busoniu.net/repository.php), și porni de la demonstrațiile din subdirectorul demo, în special din fișierul `invertedpendulum_demo`. Acest fișier conține demonstrațiile folosite la curs.)