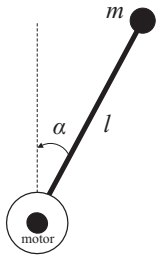
Control prin învăţare (Learning Control)
Laborator 3: Programarea dinamică cu aproximare

21 aprilie 2016

Regulament – acelaşi ca şi la laboratorul precedent, cu excepţia termenului limită pentru predarea soluţiei, care este **joi 5 mai 2016**.

Pendulul inversat

În acest laborator vom considera problema pendulului inversat. Acest sistem constă dintr-o masă m ataşată unui braţ ce se roteşte în plan vertical şi este acţionat de un motor, vezi figura. Obiectivul este ca masa să fie adusă şi stabilizată orientată în sus, pornind de la orice poziţie şi viteză iniţială. Puterea motorului este însă insuficientă pentru a face acest lucru într-o singură rotaţie din toate stările iniţiale. Din anumite stări, cum ar fi masa orientată în jos, pendulul trebuie acţionat într-o mişcare oscilantă pentru a acumula suficientă energie, înainte de a fi împins în sus şi 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 α variază în intervalul $[-\pi, \pi]$ rad, cu $\alpha = 0$ însemnând prin convenţie că pendulul este orientat în sus. Tot prin convenţie, unghiul se roteşte în acest interval în aşa fel încât, de exemplu, o rotaţie de $3\pi/2$ corespunde la $\alpha = -\pi/2$. Viteza unghiulară $\dot{\alpha}$ este restricţionată prin saturaţie la intervalul $[-15\pi, 15\pi]$ rad/s. Sistemul în timp discret este obţinut integrând numeric dinamica continuă din ecuaţia de mai sus, de-a lungul eşantioanelor de timp. Acţiunea (voltajul motorului) este limitată la $[-3, 3]$ V, insuficientă pentru a împinge pendulul în sus într-o singură rotaţie.

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

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

Matricea Q_{rew} penalizează valorile diferite de zero ale celor două stări într-o măsură similară, date fiind amplitudinile lor diferite, iar R_{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 şi formează baza laboratorului poate fi descărcat de la adresa:

`http://busoniu.net/teaching/ci2016`

secţiunea "Practical assignments". Dezarhivaţi codul într-un director de pe calculatorul dvs., navigaţi din MATLAB în acest director şi rulaţi scriptul `startup`. Codul poate fi acum folosit.

Tema de laborator

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

Partea 1. Acurateţea aproximatorului în iteraţia fuzzy Q

În această parte, vom investiga efectul unui element crucial în DP şi RL cu aproximare: acurateţea aproximatorului. Fişierul `adp_example` ilustrează cum se poate rula iteraţia fuzzy Q folosind un interpolator pe o grilă echidistantă de $N \times N$ puncte în spaţiul stărilor, şi o discretizare echidistantă cu M elemente în spaţiul acţiunilor; precum şi cum se poate extrage şi studia soluţia obţinută, în forma unui aproximator generic. Acest aproximator poate fi folosit pentru a calcula valori Q şi acţiuni, precum şi pentru a reprezenta grafic funcţia de valoare şi legea de control. Fişierul ilustrează cum se poate simula o traiectorie controlată a sistemului folosind soluţia calculată.

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

Cerinţele sunt:

1. Rulaţi iteraţia fuzzy Q pentru numărul N de puncte pe grila din spaţiul stărilor variind în secvenţa $5, 11, 15, 21$. Ceilalţi parametri (numărul de acţiuni discrete M, pragul de convergenţă etc.) pot fi păstraţi la valorile din exemplu. **[0.5p]**

2. Studiaţi soluţia rezultantă: funcţii de valoare, legi de control, şi traiectorii controlate din starea iniţială $[-0.95\pi, 0]^T$ (pendulul orientat aproape în jos), pentru aceste valori ale lui N. Reprezentaţi grafic evoluţia cu N a distanţei între soluţia $\hat{Q}$ obţinută de algoritm şi Q^*, precum şi între legea de control corespunzătoare $\hat{h}$ şi legea optimală h^*. Discutaţi rezultatele. **[2.5p]**

   Pentru a calcula distanţa între funcţii Q, folosiţi o grilă cu $K \times K \times M$ puncte în spaţiul stare-acţiune, unde K este o valoare liber aleasă (de exemplu în intervalul 20-50), iar M numărul de acţiuni discrete ales mai sus. De notat că aproximatorul nu poate calcula valori Q pentru acţiuni care nu fac parte din discretizare, aşadar trebuie să ne limităm la discretizarea aleasă la pasul precedent. Distanţa este media diferenţelor pătratice între $\hat{Q}$ şi Q^* pentru punctele de pe grilă:

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

   Distanţa 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 funcţia Q).

3. Reprezentaţi grafic evoluţia timpului de execuţie cu N, şi discutaţi rezultatul. **[1p]**

4. Repetaţi experimentul de la punctele 1–2, dar de această dată variaţi N mai fin, folosind valorile $15, 16, 17, ..., 20$. Are evoluţia calităţii soluţiei aceeaşi natură ca în cazul precedent? Puteţi identifica motivele? **[1p]**

Partea 2. Efectul setului de date asupra LSPI

Diferenţa esenţială între LSPI şi iteraţia fuzzy Q este că LSPI nu foloseşte modelul sistemului, ci este bazat pe date. Vom studia aici efectul numărului de eşantioane folosit pentru a învăţa soluţia cu LSPI. Acelaşi fişier ca şi mai sus, `adp_example`, arată cum se rulează algoritmul cu o grilă echidistantă de $N \times N$ funcţii radiale de bază, şi o discretizare echidistantă cu M elemente în spaţiul acţiunilor; cum se poate extrage aproximatorul rezultat; şi cum se poate simula o traiectorie controlată a sistemului folosind soluţia calculată. Setul de date este generat uniform aleator în spaţiul stare-acţiune, unde stările sunt continue şi acţiunile sunt discretizate, iar numărul de eşantioane poate fi configurat.

Cerinţele sunt:

1. Rulaţi LSPI pentru numărul N_S de eşantioane variind în setul $2500, 5000, 7500, 10000$. Ceilalţi parametri (numărul de funcţii de bază N, de acţiuni discrete M, pragul de convergenţă etc.) pot fi păstraţi la valorile din exemplu. Pentru a ne asigura că algoritmul primeşte cel puţin aceleaşi informaţii când creştem numărul de eşantioane, este preferabil ca eşantioanele să fie generate în aşa fel încât, de exemplu, setul de 5000 să includă setul de 2500. **[0.5p]**

2. Studiaţi soluţia rezultantă: funcţii de valoare, legi de control, şi traiectorii controlate din starea iniţială $[-0.95\pi, 0]^T$, pentru aceste valori ale lui N_S. Reprezentaţi grafic evoluţia cu N_S a returnului obţinut din această stare iniţială, a numărului de iteraţii rulate, şi a timpului de execuţie. Discutaţi rezultatele. **[2p]**

3. Comparaţi cele mai bune soluţii (care au cel mai mare return) obţinute cu LSPI şi iteraţia fuzzy Q, din punctul de vedere al calităţii şi al timpului de execuţie necesar. **[0.5p]**

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

În raport, descrieţi pe scurt problemele pe care le-aţi rezolvat, şi mai în detaliu studiile cerute, incluzând grafice reprezentative / importante (nu toate graficele obţinute). Nu uitaţi să includeţi separat listingul codului dezvoltat.

(Pentru exemple suplimentare de algoritmi cu aproximare, puteţi descărca toolboxul complet de DP şi RL cu aproximare, de la adresa `busoniu.net/repository.php`, şi porni de la demonstraţiile din subdirectorul `demo`, în special din fişierul `invertedpendulum_demo`. Acest fişier conţine demonstraţiile folosite la curs.)