Control prin învăţare (Learning Control)
Laborator 4: Planificarea online

Regulament

Regulament – acelaşi ca şi la laboratorul precedent, cu excepţia termenului limită pentru predarea soluţiei, care este **marţi 17 mai 2016**. Acesta este şi **termenul limită pentru orice soluţie întârziată**. Studenţii care la această dată nu au toate soluţiile predate nu vor fi eligibili pentru examen. Cum sesiunea de discuţii este programată în 19 mai, aceasta este din păcate cea mai târzie dată care mai permite şi corectarea laboratoarelor.

Introducere

Vom investiga comportamentul a trei algoritmi de planificare: *OPD*, planificarea optimistă pentru sisteme deterministe; *OSP*, planificarea optimistă cu un număr limitat de comutări, şi *SOPC*, planificarea optimistă simultană pentru acţiuni continuem, a se vedea partea 6 a cursului. Reamintim că planificarea are acces la model şi ca atare nu trebuie să înveţe.

Vom considera din nou problema pendulului inversat, vezi laboratorul 3 pentru o descriere. În funcţia de recompensă:

$$\rho(x, u) = -x^T Q_{\text{rew}} x - R_{\text{rew}} u^2$$

vom alege ponderile $Q_{\text{rew}} = \text{diag}(1, 0.01)$, $R_{\text{rew}} = 0.3$ pentru a optimiza şi consumul de energie pe lângă atingerea stării ţintă (originea). Această funcţie de recompensă este apoi normalizată în intervalul $[0, 1]$. Factorul de discount este ales $\gamma = 0.85$.

Codul ce implementează simulatorul 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`. Implementarea algoritmilor de planificare online este deja furnizată, iar fişierul `lab4_example` ilustrează cum se pot rula aceşti algoritmi şi cum se pot schimba parametrii lor.

Sistemul este vizualizat în timp ce algoritmii rulează, pentru a avea o impresie în "timp real" a performanţei lor, dar această vizualizare poate fi dezactivată dacă se doreşte ca algoritmii să ruleze mai repede. După ce un algoritm finalizează o traiectorie, aceasta este reprezentată grafic, oferind o imagine de ansamblu. Fişierul exemplu arată şi cum se poate afla returnul obţinut de-a lungul traiectoriei şi timpul mediu de execuţie al planificării (media este efectuată de-a lungul paşilor traiectoriei).

1 OPD

Pentru OPD, variaţi bugetul de expansiuni n alocat algoritmului la fiecare pas, folosind câteva valori între 50 şi 300. Înregistraţi returnurile şi timpii de execuţie pentru fiecare valoare a lui n, şi reprezentaţi grafic aceste cantităţi în funcţie de n. **[0.5p]**

Discutaţi influenţa lui n asupra calităţii soluţiei şi a timpului de execuţie. Analiza algoritmului presupune că timpul de execuţie este proporţional cu numărul de expansiuni; este această presupunere confirmată de rezultatele reale? De ce/de ce nu? **[1.5p]**

2 OSP

Pentru OSP, alegeţi o valoare intermediară pentru n şi păstraţi-o fixată, variind parametrul S – numărul de comutări permis în soluţiile explorate de algoritm – între 1 şi 5. Reprezentaţi grafic returnul în funcţie de S, comparându-l cu returnul obţinut de OPD pentru aceeaşi valoare a lui n. **[0.5p]**

Discutaţi influenţa lui S asupra calităţii soluţiei, comparativ cu OPD. **[1p]**

3 SOPC

Pentru SOPC, bugetul n este specificat ca un număr de tranziţii ale sistemului care pot fi simulate. Ţinând cont că expansiunea unui nod în OPD necesită mai multe tranziţii, configuraţi un set de valori ale lui n pentru SOPC în aşa fel încât să folosească acelaşi număr de tranziţii ca şi OPD mai sus. Rulaţi algoritmul pentru fiecare valoare, şi reprezentaţi grafic returnurile în funcţie de n. **[0.75p]**

Comparaţi aceste returnuri cu cele obţinute de OPD pentru aceleaşi bugete de tranziţii, discutând motivele pentru relaţia observată. **[1.25p]**

4 Studiul semnalului de comandă

Alegeţi o valoare a bugetului pentru care toţi algoritmii obţin soluţii bune (un singur swing; de ex. cel mai mare buget), şi studiaţi traiectoriile obţinute de fiecare dintre algoritmi, concentrându-vă pe forma semnalului de comandă u. Puteţi identifica diferenţele între semnalele obţinute de către algoritmii de planificare, şi motivele pentru care acestea apar, ţinând cont de natura celor trei algoritmi? **[1p]**

Datorită acţiunilor continue, SOPC ar trebui să aibă poată optimiza mai bine energia injectată în sistem pentru a-l controla. Pentru a infirma sau confirma acest lucru, vom studia pentru toate valorile lui n energia de comandă E depusă de-a lungul traiectoriei. Vom presupune că puterea electrică este proporţională cu pătratul voltajului $V^2 = u^2$; reamintiţi-vă de asemenea că energia este puterea consumată de-a lungul timpului. Reprezentaţi grafic E în funcţie de n pentru OPD şi SOPC. Discutaţi rezultatul. **[1.5p]**