Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/11IH03Milic

PRIMENA DUBOKOG UČENJA SA PODSTICAJEM ZA PLANIRANJE KRETANJA ROBOTSKOG MANIPULATORA

DEEP REINFORCEMENT LEARNING FOR MOTION PLANNING OF ROBOTIC MANIPULATORS

Lazar Milić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA, ROBOTIKA I AUTOMATIZACIJA

Kratak sadržaj – *U ovom radu prikazana je primena dubokog učenja sa podsticajem za rešavanje problema planiranja kretanja robotskim manipulatorom. Ovakvim pristupom omogućava se generalizacija algoritama planiranja kretanja za industrijska ii neindustrijska okruženja. Takođe, pokazano je poređenje stihastičkih i determinističkih tipova algoritama za rešavanje problema planiranja kretanja.*

Ključne reči: *Duboko učenje sa podsticajem, planiranje kretanja, neuronske mreže, robotika*

Abstract – *In this paper we present Deep Reinforcement Learning framework as a way to solve problems of motion planning in robotics. With Deep Reinforcement Learning it is possible to generalize motion planning algorithms for both industrial and non-industrial enviroments. In the end, we compared both stochastic and deterministic algorithms for solving motion planning problem.*

Keywords: *Deep Reinforcement Learning, motion planning, neural networks, robotics*

1. UVOD

Manipulacija objektima jedan je od većih otvorenih problema u robotici, naročito ako struktura objekta nije poznata. Kako je korišćenje klasičnih metoda upravljanja robotima i robotskim manipulatorima za rešavanje problema manipulacije objektima samo po sebi težak zadatak, čija kompleksnost raste uz povećanje domena primene tog sistema, teži se ka generalizaciji rešenja ovog problema. Pored problema manipulacije objektima različitih struktura, javlja se problem primene algoritma upravljanja na različite robotske konfiguracije. Sl. 1 ilustruje postavku opisanog problema nestruktuirane raspodele objekata u prostoru.

U ovom radu će biti predložena metoda upravljanja robotskim sistemom zasnovana na učenja sa podsticajem (eng. *reinforcement learning* - RL), za manipulaciju objektima nepoznate strukture u nestruktuiranom okruženju. Cilj je razvoj upravljačkog algoritma opšte namene, koji će biti nezavisan od konfiguracije robota.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Mirko Raković, van. prof.

Slika 1. *Primer nestruktuiranog rasporeda objekata [1].*

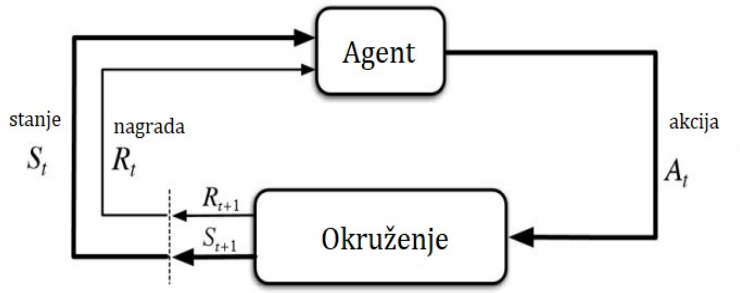
2. UČENJE SA PODSTICAJEM

Reinforcement learning (RL) je grana mašinskog učenja bazirana na sekvencijalnom odlučivanju [2]. Problem koji se posmatra u RL se sastoji od agenta i njegovog okruženja. RL agent, u daljem tekstu agent, vrši neke akcije u okruženju u kome se nalazi i za to dobija određene nagrade. Proces nagrađivanja predstalja povratnu informaciju agentu od strane okoline, govoreći koliko je izvršena akcija dobra za dati trenutak. Zadatak agenta jeste pronalaženje optimalne politike (eng. *policy*), odnosno mapiranje trenutnog stanja u akcije, tako da se maksimizuje očekivana ukupna nagrada.

Razlika između *reinforcement learning*-a i mašinskog učenja pod nadzorom (eng. *supervised learning*) je u tome što agent u datom trenutku ne zna koja od mogućih akcija je optimalna za dobijanje maksimalne očekivane sume nagrada. Proces učenja agenta se zasniva na principu pokušaja (eng. *trial and error*).

RL problem je definisan u sklopu optimalnog upravljanja za Markovljev proces odlučivanja (eng. *Markov Decision Process*, MDP).

Ostale komponente u RL učenju kao što su politika, vrednosna funkcija i Belmanov uslov optimalnosti će biti razmatrani u ovom poglavlju. Pored navedenih komponenti RL-a, u ovom poglavlju će biti opisani različiti tipovi pristupa u RL, to su algoritmi bazirani na modelu (eng. *model-based*) i bez modela (eng. *model-free)*. Na slici 2. ilustrovana je interakcija agenta sa njegovim okruženjem.



Slika 2. *Ilustracija procesa interakcije agenta sa okruženjem, izvor: autor.*

2.1. Markovljev proces odlučivanja

Markovljev proces odlučivanja je definisan sa:

- Skupom stanja, $s \in S$, gde je S prostor stanja
- Skupom akcija, $a \in A$, gde je A prostor akcija
- Skalarnom nagradom, r
- Verovatnoćom tranzicije iz jednog stanja u drugo

$$p(s,a) = P_r(s_{t+1} = s' | s_t = s, a_t = a)$$

- Funkcijom nagrade, $R: S \times S \times A \rightarrow \mathbb{R}$

$$R(s,a,s') = E(r_t | s_t = s, a_t = a, s_{t+1} = s')$$

- Faktorom odbitka (eng. *discount factor*), γ

Funkcija nagrade i verovatnoće prelaska definišu najvažnije aspekte MDP i uglavnom se smatraju nepoznatim u opštem slučaju u oblasti *reinforcement learning*-a.

Odabir akcija od strane agenta se modeluje preko funkcije mapiranja koja se naziva *policy*. *Policy* je definisana kao mapiranje stanja u akcije.

Policy može biti deterministička, u tom slučaju zavisi isključivo od trenutnog stanja, $\pi(s)$, ili stohastička, $\pi(a|s)$, tako da definiše distribuciju verovatnoća nad akcijama za dato stanje.

2.2. *Value* funkcija

Kako bi agent mogao da odluči koju akciju da odabere u određenom trenutku, važno je da agent zna koliko je dobro trenutno stanje u kome se agent nalazi. Način određivanja koliko je neko stanje dobro je preko vrednosne funkcije. Vrednosna funkcija je definisana kao očekivana suma nagrada koju bi agent dobio ako nastavi da prati *policy* π iz stanja s. Vrednosna funkcija, $V_\pi(s)$ za politiku π data je kao:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi(G_t | s_t = s) \\ &= \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \, | s_t = s\right) \end{aligned} \tag{1}$$

Slično tome, vrednosno-akciona funkcija (eng. *action value funcion*), poznatija kao Q-funkcija, se može definisati kao očekivana suma nagrada delujući akcijom a u stanju s, a zatim prateći politiku π. Vrednosno-akciona, $Q_\pi(s,a)$ je data kao:

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}(G_t | s_t = s, a_t = a) \\ &= \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a\right) \end{aligned} \tag{2}$$

2.3. Belmanove jednačine optimalnosti

Belmanove jednačine formulišu problem maksimizovanja očekivane sume nagrada upotrebom rekurzivne osobine vrednosne funkcije. Politika π se smatra boljom od neke druge politike π' ako je očekivana povratna vrednost te politike veća od π' za svako $s \in S$, što implicira $V^\pi(s) \geq V^{\pi'}(s)$ za svako $s \in S$. Belmanove jednačina optimalnosti za *value* i *action value* funkcije definisane su kao:

$$\begin{aligned} V_{*(s)} &= \max_a \mathbb{E}_{\pi^*}(G_t | s_t = s, a_t = a) \\ &= \max_a \mathbb{E}_{\pi^*}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a\right) \end{aligned} \tag{3}$$

$$\begin{aligned} Q_*(s,a) &= \mathbb{E}(r_t + \gamma \max_{a'} Q_*(s_{t+1}, a') | s_t = s, a_t = a) \\ &= \sum_{s'} p(s'|s,a)[R(s,a,s') + \gamma \max_{a'} Q_*(s',a')] \end{aligned} \tag{4}$$

3. DUBOKO UČENJE SA PODSTICAJEM

Popularnost primene dubokog učenja u oblasti *reinforcement learning*-a znatno je skočila prethodnih godina.
Termin *Deep Reinforcement Learning* (DRL) predstavlja primenu dubokih neuronskih mreža za aproksimaciju *value* ili *policy* funkcija u oblasti *reinforcement learning*-a.

DRL je u svojim prvim koracima razvoja postigao nivo uspešnosti rešavanja, odnosno igranja kompjuterskih Atari igara na nivou čoveka, kao i za kontrolu visokodimenzionih problema u 3D prostoru.

Činjenica da je primena dubokih neruonskih mreža za rešavanje RL problema veoma pogodna za upravljanje modelima visokih dimenzija observabilnosti i akcija, DRL je pronašao svoju primenu u robotici i kao takav je praktičan za upravljanje robotskim šakama, robotskim manipulatorima, autonomnim platformama i sl.

3.1. Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) [3] spada u grupu *actor-critic* algoritama. DDPG koristi teoremu o determinističkom gradijentu politike za ažuriranje *policy* funkcije.

Nestabilnost algoritma prilikom treninga, kao i kovarijansa između podataka prikupljenih tokom jedne epizode u okruženju se rešava korišćenjem *replay buffer*-a.

Svaka tranzicija koju agent izvrši u okruženju se skladišti u memoriju i tokom svakog optimizacionog koraka nasumično se uzorkuje skup tranzicija za obuku.

Funkcija koja se optimizuje u DDPG algoritmu je definisana kao:

$$L = \mathbb{E}_{s\sim\rho_\beta, a\sim\beta}\left(y - Q(s,a;\theta^Q)\right)^2 \tag{5}$$

3.2. Soft Actor Critic

Soft Actor-Critic (SAC) [4] pripada grupi *off-policy actor-critic* algoritama. Cilj kod SAC algoritma jeste maksimizovanje entropije pored optimizacije ukupne očekivane vrednosti nagrade. Funkcije koje se optimizuju u SAC algoritmu su:

$$J_Q(\theta) = E_{(s_t,a_t)\sim D}\left[\frac{1}{2}\Big(Q_\theta(s_t,a_t) - \big(r(s_t,a_t) + \gamma E_{s_{t+1}\sim p}[V_{\bar{\theta}}(s_{t+1})]\big)\Big)^2\right] \tag{6}$$

$$J_\pi(\phi) = E_{s_t\sim D}\Big[E_{a_t\sim\pi_\phi}\big[\alpha\, log\, log\, \left(\pi_\phi(s_t)\right) - Q_\theta(s_t,a_t)\big]\Big] \tag{7}$$

$$J(\alpha) = E_{a_t\sim\pi_t}[-\alpha\, log\, log\, \pi_t(s_t) - \alpha\bar{\mathcal{H}}] \tag{8}$$

4. OBUČAVANJE ALGORITAMA

U ovom poglavlju će biti opisano simulaciono okruženje u kome je izvršena obuka algoritama upotrebom dubokog učenja sa podsticajem.

4.1. Simulacija

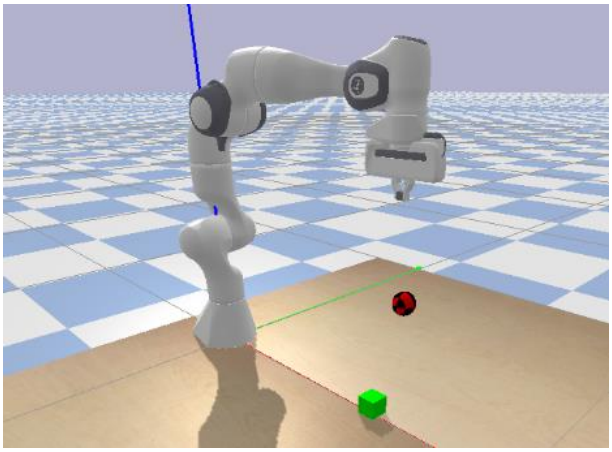
Pybullet je dodatak za *Python* programski jezik koji služi za simuliranje robotskih sistema, vizije i dr. Prednosti korišćenja simulacije su mogućnost bržeg izvršavanja komandi u odnosu na realne sisteme.

DRL algoritmi inicijalizovani nasumično, njihovo odlučivanje je nepredvidivo bez prethodnog obučavanja, što ih čini nepraktičnim za primenu na realnim sistemima, pošto može doći do oštećenja sistema ili do povrede čoveka u slučaju njegovog prisustva.
Iz tog razloga je pogodno koristiti simulacije za obučavanje algoritma.

Simulacija upotrebljena za obučavanje algoritama se sastoji od Frank Emika robota, čija baza se nalazi na stolu. Zadatak koji robot treba da ispune jeste manipulacija kockom, čija je inicijalna pozicija uniformno uzorkovana u granicama radnog prostora robota u ravni stola, u pozitivnom delu *x*-ose.

Simulacija radi brzinom od 240Hz, a zadavanje akcija od strane agenta se izvršava brzinom od 30Hz.

Na slici 3. prikazan je izgled simulacija Franka Emika robota, koji ima zadatak da pokupi kocku koja se nalazi u ravni stola i odnese je u poziciju naznačenu crveno-crnom sferom.



Slika 3. *Simulirano orkuženje u kome su obučavani algoritmi, izvor: autor.*

4.2. Detalji implementacije algoritama

Algoritmi čije performanse su poređene za planiranje kretanja robotskog manipulatora su DDPG i SAC. Da bi agent izvršio zadatak potrebno je da reši dva problema određenim redosledom.

Prvi problem jeste hvatanje kocke, nakon toga agent treba da kocku premesti na odgovarajuće mesto, zadato na početku epizode.

Iz tog razloga su algoritmi SAC i DDPG implementirani u kombinaciji sa HER tehnikom.

U tabelama 1. i 2. prikazani su detalji implementacije SAC i DDPG algoritma.

Tabela 1. *Detalji implementacije SAC algoritma*

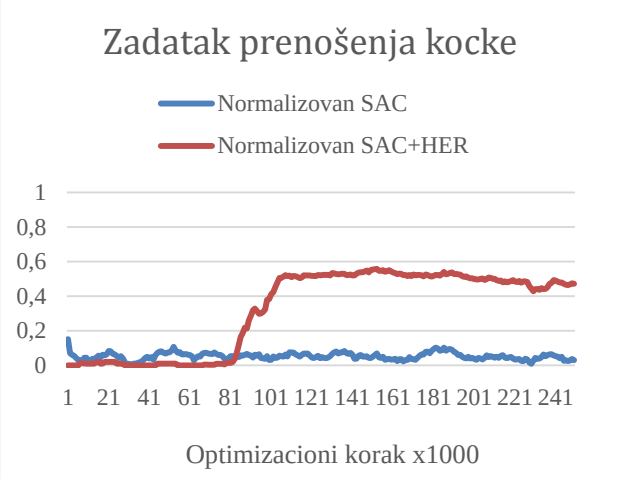
Broj skrivenih neurona za *actor* i *critic* mrežu	256
Broj skrivenih slojeva za *actor* i *critic* mrežu	2
Koeficijent brzine učenja za *actor* i *critic* mrežu	0.001
Gamma	0.99
Tau	0.005
Koeficijent brzine učenja α parametra	0.0003
Batch	256
Maksimalan broj tranzicija u memoriji	1,000,000

Tabela 2. *Detalji implementacije DDPG algoritma*

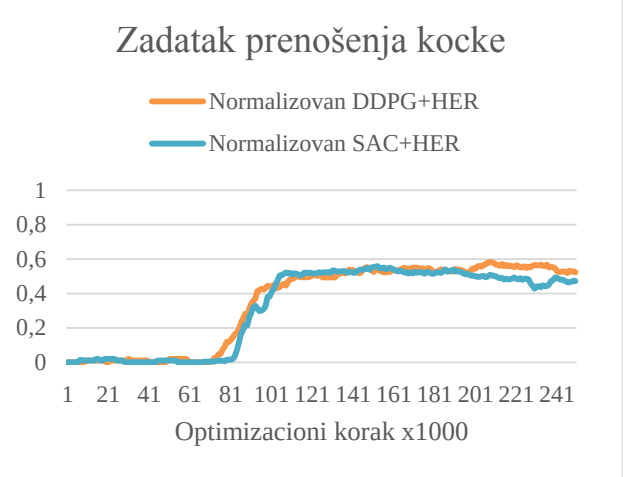
Broj skrivenih neurona za *actor* i *critic* mrežu	256
Broj skrivenih slojeva za *actor* i *critic* mrežu	3
Koeficijent brzine učenja za *actor* i *critic* mrežu	0.001
γ	0.98
τ	0.005
μ	0
σ	0.25
θ	0.15
Batch	256
Maksimalan broj tranzicija u memoriji	1,000,000

5. ZAKLJUČAK

Na osnovu dobijenih rezultata, pokazano je da se duboko učenje sa podsticajem može koristiti za planiranje kretanja. Na slici 4. prikazano je poređenje rezultata obuke SAC algoritma i SAC algoritma sa HER tehnikom. Slika 5. predstavlja poređenje rezultata obuke SAC i DDPG algoritma uz korišćenje HER tehnike za zadatak manipulacije kockom.



Slika 4. *Poređenje rezultata obuke SAC algoritma sa i bez upotrebe HER tehnike, izvor: autor.*



Slika 5. *Poređenje rezultata obuke SAC i DDPG algoritama, sa upotrebom HER tehnike, izvor: autor.*

6. LITERATURA

[1] https://www.roboticsbusinessreview.com/manufacturing/deep-learning-factory-automation/ (pristupljeno u martu 2020.)

[2] R. S. Sutton, A. G. Barto "*Reinforcement learning, an introduction*", MIT Press, Cambridge, MA, 2018.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continous Control with Deep Reinforcement Learning", ICLR, 2016..

[4] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor", ICML, 2018.

Kratka biografija:

Lazar Milić rođen je u Vrbasu, 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Mehatronike - Mehatronike, robotike i automatizacije je odbranio 2020. god.

kontakt: lazarmilic2@gmail.com