Зборник радова Факултета Техничких наука, Нови Сад

**UDK: 4.8**
**DOI: https://doi.org/10.24867/30BE21Radojicic**

# ПРОЈЕКТОВАЊЕ АДАПТИВНИХ РЕГУЛАТОРА ПРИМЕНОМ ППО АЛГОРИТМА УПОТРЕБОМ ПРОГРАМСКОГ ПАКЕТА МАТЛАБ

# DESING OF ADAPTIVE CONTROLLERS BY MEANS OF PPO ALGORITHM USING MATLAB

Вељко Радојичић, *Факултет техничких наука, Нови Сад*

**Област** – **ЕЛЕКТРОТЕХНИКА И РАЧУ-НАРСТВО**

**Кратак садржај** – *У овом раду истражена је могућност употребе учења потркепљењем, кон-кретно Proximal Policy Optimization алгоритма, за проблеме управљања континуалним динамич-ким системима. Изабрано је неколико репрезен-тативних, линеарних и нелинеарних система са континуалном динамиком, а за решење пробле-ма управљања, трениран је одговарајући агент, и то користећи MATLAB-ово софтверско окружење Reinforcement Learning Designer. За приказ резул-тата и симулацију, коришћен је Simulink.*

**Кључне речи** *PPO, политика, агент, регулатор*

**Апстракт** - *This paper focuses on exploring of using Reinforcement learning's Proximal Policy Optimization algorithm for problems of control of continual dynamic systems. Reinforcement learning agent has been trained on relatively simple linear and non-linear exaples of systems, using MATLAB's Reinforcemet Learning Designer application, while for result and simulation, Simulink has been used.*

**Keywords:** *PPO, policy, agent, controller*

## 1. УВОД

*Proximal Policy Optimization* (у наставку *PPO*), као један од новијих алгоритама учења потрекљењем (енг. *reinforcement learning*), представља једно уна-прeђење *Policy gradient* алгоритама, као што су *Natural Policy Gradient*, *REINFORCE*, *Trust Region Policy Optimizaion* итд. *PPO* се, између осталог, из-двоја и због своје *actor-critic* структуре, која пружа баланс између ескплорације и експоатације. Делат-ник (енгл. *actor*) јесте агент или регулатор (контро-лер). То је ентитет који на основу мерења одређу-је наредну акцију коју треба извршити. Оценитељ (енгл. *critic*) је ентитет који оцењује текуће поноша-ње система у затвореној спрези и даје информације на основу којих регулатор ажурира политику одлу-чивања. С обзиром да је та терминологија уобича-јена у литератури на енглеском језику, у наставку овог рада наставићемо да називамо овакву архитек-туру термином *actor-critic*, и тако ћемо је и наводи-ти, у оригиналу, на енглеском језику, латиницом. Овакву структуру карактерише специфичан израз за апроксимацију градијента критеријума оптимал-ности

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{\substack{i=1..N \\ t=0..T}} \nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t}) \mathrm{adv}_{\pi,\phi,i,t}\,, \quad (1)$$

где смо "унапређење" (енгл. *advantage*), односно разлику очекиване добити након једне акције ($r(s_{i,t}, a_{i,t}) + \gamma\hat{v}_{\pi,\phi}(s_{i,t+1})$) и полазне очекиване до-бити ($\hat{v}_{\pi,\phi}(i, s_t)$) означили са

$$A_{\pi,\phi,i,t} = r(s_{i,t}, a_{i,t}) + \gamma\hat{v}_{\pi,\phi}(s_{i,t+1}) - \hat{v}_{\pi,\phi}(i, s_t)\,, \quad (2)$$

где су са $\theta$ и $\phi$ означени параметри одговарајућих неуронских мрежа делатника и оценитеља. Увође-њем новог, ограниченог критеријума оптималности, спречавају се нагле ажурирања полике, што може имати дестабилишући ефекат у процесу учења [2]. Напомињемо да се унапређење може посматрати као разлика апостериорне и априорне естимације добити. Политике се код овог алгоритма ажурира-ју на следећи начин

$$\Delta\theta^* = \arg\max_\theta \underset{s,a \sim \pi_{\theta_k}}{\mathrm{E}} [L(s, a, \theta_k, \theta)] \quad (3)$$

Критеријум оптималности дат је у облику

$$L(s, a, \theta_k, \theta) = \min\left( \rho_\theta(a|s) A_{\pi_{\theta_k}}(s, a),\ g(\epsilon, A_{\pi_{\theta_k}}(s, a)) \right), \quad (4)$$

при чему је

$$\rho_\theta(a|s) = \frac{\pi_\theta(a|s)}{\pi_{\theta+\Delta\theta}(a|s)}\,. \quad (5)$$

У претходним једначинама, са је означена функ-ција унапређења (*advantage*-функција), док је g де-финисано са

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A\,, & A \geq 0 \\ (1-\epsilon)A\,, & A < 0 \end{cases}. \quad (6)$$

Параметар $\epsilon$ стоји као ограничавајући параметар за ажурирање политике. У случају да је A-функција

**НАПОМЕНА: Овај рад је проистекао из мастер рада чији је ментор др Милан Рапајић, редовни професор.**

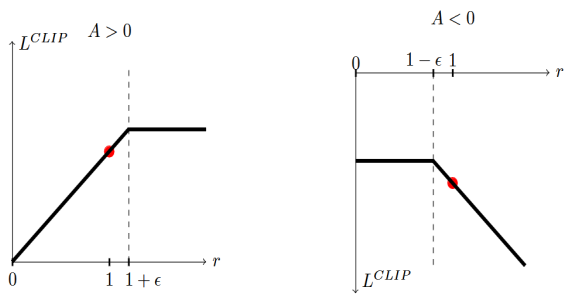
позитивна, критеријум оптималности своди се на облик

$$\mathcal{L}(s,a,\theta_k,\theta) = \min\left(\rho_\theta(a|s),(1+\epsilon)\right)A_{\pi_{\theta_k}}(s,a)\ , \tag{7}$$

и тада критеријум оптималности расте уколико $\pi_\theta(s|a)$ расте. Ипак због оператора min, онда када промена политике $\rho_\theta(s|a)$ достигне вредност $1+\epsilon$, $\mathcal{L}$ престаје да расте. Другим речима, ограничено је колико наредна политика може да одступа од претходне. У случају да A-функција има негативну вредност, критеријум оптималности постаје

$$\mathcal{L}(s,a,\theta_k,\theta) = \max\left(\rho_\theta(a|s),(1-\epsilon)\right)A_{\pi_{\theta_k}}(s,a)\ . \tag{8}$$

Критеријум оптималлности расте уколико и $\pi_\theta(s|a)$ опада, тј. уколико акција постаје мање вероватна. Оператор max спречава промену $\mathcal{L}$ након $1-\epsilon$, тако да промена политике такође ограничена. За фактор $\epsilon$, емпиријски је показано да се вредности блиске $\epsilon = 0.2$ дају најбоље резултате [4].




Слика 1: *Зависност критеријума оптималности од односа вероватноћа текуће и и наредне политике. Слика преузета са [5]*

## 2. ОПИС ПРОБЛЕМА

Истраживање могућности примене *PPO* алгоритма на проблеме управљана динамичким системима изискује потребу за адекватним софтверским окружењем, што због лакше имплементације и тренирања самог алгоритма, што због могућности симулације. *Simulink*, симулационо окружење програмског пакета *MATLAB*, заједно са интегрисаном *Reinforcement Learning Designer* апликацијом, пружа неопходне алате за овакав подухват. Конкретно, ограничили смо се на неколико репрезентативних динамичких система, а као циљ је постављено пројектовање *RL*-агента који би у класичној повратној спрези заменио неки од конвенционалних регулатора (нпр. ПИД регулатор), као и анализу добијених резултата.

### 2.1. Избор агента

Апликација омогућава олакшано прављење агента. У случају да корисник не жели да га имплементира кроз MATLAB, и прихвата подразумевану топологију неуронских мрежа за actor-a и critic-a, агент се прави секвенцијалним бирањем опција и хиперпараметара. Објекат агента такође се може направити коришћењем методе `rlPPOAgent`. Предност овакве имплементације је у томе што се агент може формирати независно од окружења, док је предуслов за прављење агента кроз апликацију постојање одговрајућег, претходно направљеног *enviroment*-а. Агент је у *simulink*-у представљен блоком *RL Agent*. Као улазни сигнали блока, дефинисани су вектор обсервација, сигнал награде и сигнал који означава да је излаазна величина изван опсега. На излазу блока је сигнал акције, управљачки сигнал.

### 2.2. Избор окружења

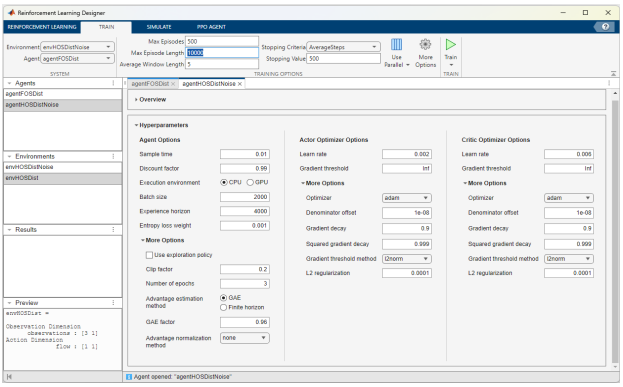
Корисник може одабрати неки од постојећих *enviroment*-а, а помоћу опције *Import*, уводи се тзв. *custom* окружење. За потребе овог рада, адекватно окружења имплеметирана је кроз *Simulink*, а потом, употребом метода *rlSimulinkEnv* и *createIntegratedEnv*. Излазни сигнал ограничен је на вредности из опсега $(-1000, 1000)$, а вектор обсервација сачињен је од сигнала грешке (разлика између сигнала референце и излазног сигнала) и интеграла сигнала грешке. Функција награде формирана је тако да строго "кажњава" излазак из опсега, велике вредности сигнала грешке и нагле промене управљачког сигнала. Рачунање награде врши се по формули

$$r = -e^2 - (u-u_p)^2\ , \tag{9}$$

где је $r$ сигнал награде, $e$ сигнал грешке, а $u$ и $u_p$ управљачки сигнали у текућем и претходном тренутку.

### 2.3. Обука

Тренирање агента у одговарајућем окружењу извршено је кроз *Reinforcement Learning Designer*, одабиром опције *Train* и подешавањем хиперпараметара.



Слика 2: *Тренирање агента*

Тренинг се може посматрати и кроз апликацију, где су приказане награде и вредности стања и усређене вредности стања, или кроз *Simulink*, где је

приказано понашање агента (регултора) током епизода.

### 2.4. Симулација

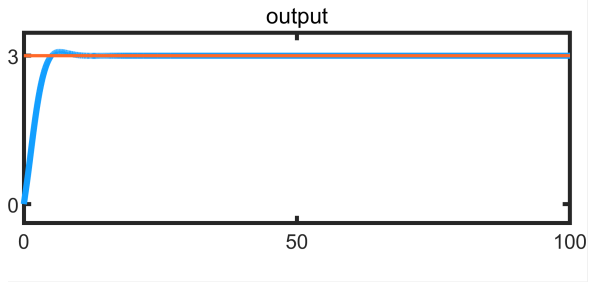
Симулирање агента (регулатора) такође се може извршити кроз апликацију, где је омогућено и чување добијених резултата, али без директног увида у понашање регулатора. Из тог разлога симулација је извршена је у оригиналном *Simulink* моделу. Име објекат агента се тада прослеђује уграђеном блоку *RL Agent* који се налази у *Simulink* моделу.

## 3. РЕЗУЛТАТИ

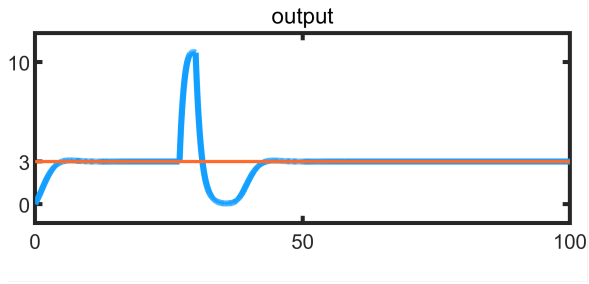
Агент (регулатор) намењен је за управљање у затвореној повратној спрези. Системи од интереса су систем првог реда описан функцијом преноса $\frac{1}{s+1}$, систем другог реда описаном функцијом преноса $\frac{1}{(s+1)^2}$, а за репрезент нелиенарног систем одабран је модел елктричног кола са потрошачем константне снаге.

### 3.1. Систем првог реда

Тренирање *PPO RL*-агента за почетак је извршено за управљање системом првог реда који је описан функцијом преноса $W(S) = \frac{1}{s+1}$. За референтни сигнал у току тренирања, одабрана је секвенца Хевисајдових сигнала променљиве вредности. Такође, агент је у трениран са присуством степ-поремећаја амплитуде 10 у трајању 3% времена симулације од 100 секунди, као и у случају када поремећај не постоји.



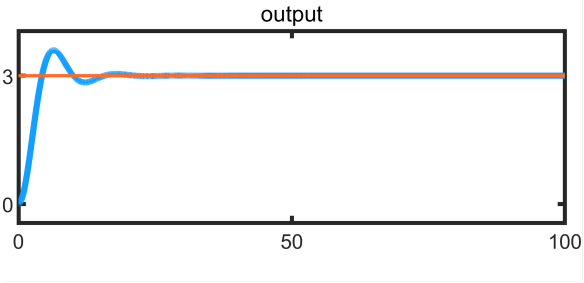

Слика 3: *Степ одзив система првог реда без присуства поремећаја*



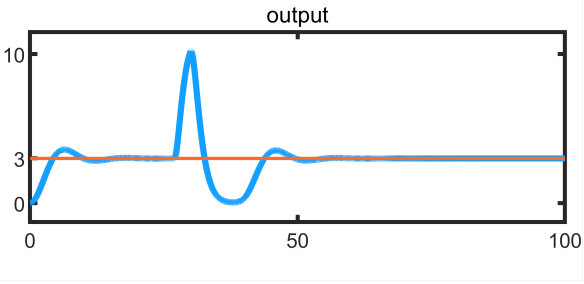

Слика 4: *Степ одзив система првог реда са присуством поремећаја*

### 3.2. Систем другог реда

Као репрезентативан динамички систем вишег реда, одабран је систем описан функцијом преноса $W(s) = \frac{1}{(s+1)^2}$. И у овом случају, агент је тренирам тако да је референца степ-сигнал променљиве висине. Поново, изложена су два засебна резултата симулације, један без присуства поремећаја, а други у присуству поремећаја истог типа као и случају система првог реда.




Слика 5: *Степ одзив система другог реда без присуства поремећаја*




Слика 6: *Степ одзив система другог реда са присуством поремећаја*

### 3.3. Нелинеарни систем

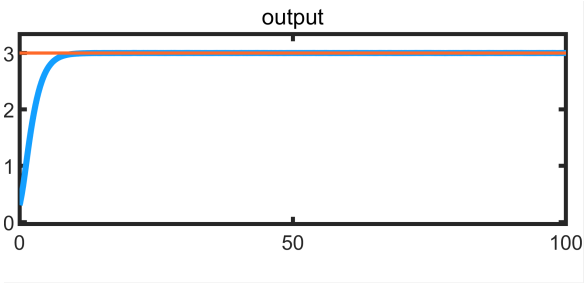
За пример нелинеарног система, узет је математички модел електричког кола са потрошачем константне снаге, где су са $v_c$ и $i_c$ означени напон на кондензатору и струја потрошача, респективно. Математички модел да је у облику

$$\begin{aligned} \dot{v}_c &= \frac{1}{C_f R_f}\left(u - v_c - R_f i_p\right) \\ \dot{i}_p &= \frac{1}{L_v}\left(v_c - R_v i_p - \frac{P}{i_p}\right) \\ y &= v_c \end{aligned} \tag{10}$$

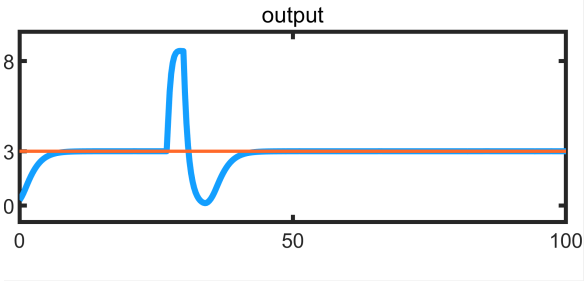
Имајући у виду да је акценат рада на понашању алгоритма, а не на самим моделима, али и због једноставности симулације и нумеричких ограничења, за све параметре кола узете су јединичне вредности, па се модел своди на

$$\begin{aligned} \dot{x}_1 &= x_1 - x_2 + u \\ \dot{x}_2 &= x_1 - x_2 - \frac{1}{x_2} \\ y &= x_1 \end{aligned} \tag{11}$$

Узлазним сигналом сматра се сигнал напонског генератора $u$, а излазним сингалом напон на кондензатору $v_c$, односно стање $x_1$ Почетне вредности стања су $x_1 = 0.3$ и $x_2 = 0.3$.




Слика 7: *Степ одзив нелинеарног система без присуства поремећаја*




Слика 8: *Степ одзив нелинеарног система са присуством поремећаја*

## 4. ЗАКЉУЧАК

У овом раду, приказана је употреба *Proximal Policy Optimization* алгоритма на проблеме управљања континуалним динамичким системима, на неколико репрезентативних примера. *RL*-агент, који у оваквој управљачкој шеми мења неки од стандардних регулатора (нпр. ПИД регулатор), показао се као могуће решење, макар на приказаним моделима. Перформансе овог и конценционалних регулатора, у различитим случајевима не разликују се значајно, барем не у покривеним случајевима. Ипак, треба имати у виду да понашање агената који су засновани на алгоритмима са овако комплексном структуром итекако зависи од подешавања хиперпараметара, што такође представља изазов, и може се сматрати засебним проблемом.

# Литература

[1] Richard S. Sutton, Andrew G. Barto "Reinforcement Learning: An Introduction, Second edition", Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 2014.-2015.

[2] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, "Proximal Policy Optimization Algorithms", `https://arxiv.org/abs/1707.06347`, 2017.

[3] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, Pieter Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation", ICLR, 2016.

[4] Nai-Chieh Huang, Ping-Chun Hsieh, Kuo-Hao Ho, I-Chen Wu "PPO-Clip Attains Global Optimality: Towards Deeper Understandings of Clippin", Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan, `https://arxiv.org/abs/2312.12065`, 2024.

[5] Wouter van Heeswijk, "Policy Gradients In Reinforcement Learning Explained" Medium, 2022.

**Кратка биографија:**

**Вељко Радојичић** рођен је у Новом Саду 2000. год. Дипломски рад на Факултету техничких наука из области Електротехнике и рачунарства на тему "Пројектовање струјне и напонске петље са спрежним филтром LCL тип" одбранио је у јулу 2023. год.
Контакт: radojicic.veljko@uns.ac.rs