

**POREĐENJE PERFORMANSI UPRAVLJAČKOG ALGORITMA REGULATORA PO STANJIMA SA ALGORITMOM ZASNOVANIM NA UČENJU SA POTKREPLJENJEM****PERFORMANCE COMPARISON OF THE STATE VARIABLE FEEDBACK REGULATOR ALGORITHM TO A REINFORCEMENT LEARNING BASED CONTROL ALGORITHM**Bojan Jorgovanović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – U ovom radu prikazan je jedan način primene učenja sa potkrepljenjem za upravljanje mehaničkim sistemom. Izvršeno je modelovanje dvostrukog matematičkog klatna i obuka agenta koji pomoću pomenutog algoritma dovodi klatno u željeni položaj. Zatim su rezultati ovog algoritma upoređeni sa rezultatima dobijenim primenom konvencionalne regulacije po stanjima. Na kraju su izmenjeni parametri modela sa kojim je agent obučen i za koji je regulator po stanjima projektovan i upoređeni su rezultati oba upravljačka algoritma primenjena na takav model.

**Ključne reči:** Regulator po stanjima, Mašinsko učenje, Učenje sa potkrepljenjem, dvostruko matematičko klatno

**Abstract** – This paper presents one way of utilising Reinforcement Learning (RL) to control a mechanical system. First, the modeling of a double pendulum and the training of an agent that would control the position of the pendulum was done. Then, the results of this algorithm were compared to the result of the state variable feedback controller. Finally, the parameters of the model with which the agent was trained and for which the state variable feedback controller was designed were changed and the results of both control algorithms applied to such a model were compared.

**Keywords:** State variable feedback controller, Machine learning, Reinforcement learning, double pendulum

**1. UVOD**

Klasična teorija upravljanja obezbeđuje pouzdan matematički aparat za upravljanje sistemom ukoliko je poznat model istog [1]. Na osnovu kvalitetnog modela moguće je projektovati upravljački algoritam koji ostvaruje željeno ponašanje sistema.

Potreba za modelom je upravo i najveći problem u ovoj oblasti, jer postupak formiranja modela može biti vrlo komplikovan, a često i nemoguć. Nasuprot tome, algoritmima zasnovanim na učenju sa potkrepljenjem nije nužno potrebno poznavanje modela sistema ili procesa kojim će upravljati [2]. Kao posebna klasa algoritama mašinskog učenja, učenje sa potkrepljenjem podrazumeva obučavanje agenta da na određen način deluje na

okolinu tako što primenjuje one akcije koje će obezbediti najveću nagradu.

U ovom radu prikazana je primena agenta čiji je princip rada zasnovan na učenju sa potkrepljenjem za upravljanje položajem dvostrukog matematičkog klatna. Rezultati ovog algoritma su upoređeni sa rezultatima koji su ostvareni primenom upravljačkog algoritma regulatora po stanjima. Na kraju je upoređena performansa prethodna dva upravljačka algoritma na modelu dvostrukog matematičkog klatna čiji su parametri izmenjeni tako da ne odgovaraju onom koji je korišćen za obuku agenta i projektovanje regulatora po stanjima.

**2. UČENJE SA POTKREPLJENJEM****2.1. Osnovni koncept**

Učenje sa potkrepljenjem (*eng. Reinforcement learning*) predstavlja jednu klasu algoritama mašinskog učenja u kom se definišu dva elementa, agent i okolina. Agent u datom diskretnom trenutku  $t$  poseduje informaciju o stanju okoline  $s_t$  i na osnovu toga primenjuje akciju  $a_t$ . Nakon primenjene akcije, okolina prelazi u stanje  $s_{t+1}$  i agentu pruža povratnu informaciju o kvalitetu primenjene akcije u vidu nagrade  $r_{t+1}$  [3]. Na osnovu nagrada, definiše se dobit  $R$  kao ponedirasa suma svih budućih nagrada.

$$R = \sum_{t=1}^{\infty} \gamma^{t-1} r_t \quad (1)$$

U (1) sa  $\gamma$  je označen faktor obezvređivanja, konstanta koja uzima vrednost iz opsega (0,1]. Pravilo na osnovu kog agent donosi odluku o tome koju akciju će primeniti u datom stanju  $s$  zove se politika odlučivanja i označava se sa  $\pi(s)$ . U zavisnosti od toga da li agent u svakom trenutku primenjuje akciju koja mu donosi najveću dobit ili ne, definišu se pohlepna politika odlučivanja (*eng. greedy decision policy*) i  $\varepsilon$ -pohlepna politika odlučivanja (*eng.  $\varepsilon$ -greedy decision policy*) [3]. Ukoliko primenjuje pohlepnu politiku odlučivanja, agent sa verovatnoćom 1 primenjuje akciju koja obezbeđuje najveću dobit. Ukoliko primenjuje  $\varepsilon$ -pohlepnu politiku odlučivanja, agent sa verovatnoćom  $1 - \varepsilon$  primenjuje akciju koja obezbeđuje najveću dobit, a sa verovatnoćom  $\varepsilon$  primenjuje neku drugu, nasumično odabranu akciju. Za datu politiku odlučivanja  $\pi$  definiše se vrednost akcije  $a$  u stanju  $s$  pri politici odlučivanja  $\pi$  kao veličina kojom agent ocenjuje

**NAPOMENA:**

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Rapačić, red. prof.

kvalitet primenjene akcije u datom stanju. Ova vrednost, koja se označava sa  $q_\pi(s, a)$ , se definiše kao dobit koju će agent ostvariti ukoliko u stanju  $s$  primeni akciju  $a$  i u narednim koracima nastavi da primenjuje akcije koje diktira politika odlučivanja  $\pi$ . Ukoliko je prostor stanja i akcija konačan, može se definisati matrica kvaliteta, odnosno Q-matrica (eng. *Q-Matrix*) koja sadrži vrednosti svih akcija u svim stanjima za datu politiku odlučivanja.

## 2.2. Q-learning

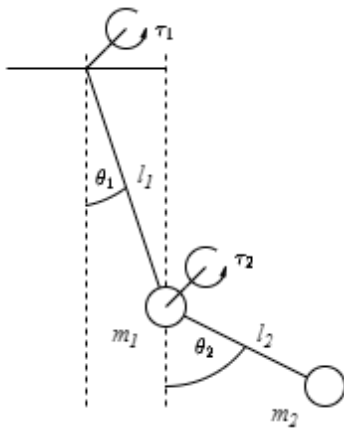
Cilj učenja sa potkrepljenjem je da agent na osnovu delovanja na okolinu nauči koju akciju je najbolje primeniti u datom stanju, odnosno da nauči optimalnu politiku odlučivanja  $\pi^*$ . Postoji više različitih klasa algoritama koji se koriste za određivanje optimalne politike odlučivanja  $\pi^*(s)$ . U ovom radu biće prikazan algoritam pod nazivom *Q-learning* koji iterativno vrši ažuriranje Q-matrice na osnovu svake primenjene akcije [3, 4].

Optimalna politika odlučivanja direktno sledi iz Q-matrice kada njena procena konvergira nakon dovoljnog broja primenjenih akcija i odgovarajućih ažuriranja. Agent pri definisanju nema nikakvo predznanje o okolini što se predstavlja nasumičnim postavljanjem inicijalnih vrednosti Q-matrice. Na osnovu definisane Q-matrice postavlja se politika odlučivanja tako što se agent u svakom stanju primenjuje akciju koja ima najveću vrednost. Kao što je već pomenuto, vrednost primenjene akcije se zatim ažurira na osnovu ostvarene nagrade po pravilu (2). U ovom izrazu sa  $Q^+(s_t, a_t)$  je označena nova vrednost Q-matrice nakon ažuriranja, a sa  $\alpha$  stopa učenja, konstanta u opsegu (0,1].

$$Q^+(s_t, a_t) = Q(s_t, a_t) + \alpha (r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \quad (2)$$

## 3. OPIS PROBLEMA

Za objekat upravljanja u ovom radu korišćeno je dvostruko matematičko klatno sa aktuatorima u oba zgloba, slika 1.



Slika 1. Dvostruko matematičko klatno

Zadatak upravljačkog algoritma jeste da dovede klatno u takav položaj da uglovi otklona  $\theta_1$  i  $\theta_2$  budu redom  $90^\circ$  i  $180^\circ$ , odnosno  $\frac{\pi}{2} \text{ rad}$  i  $\pi \text{ rad}$  u odnosu na vertikalu. Za upravljačke signale uzeti su momenti sile koje aktuatori generišu u zglobovima klatna,  $\tau_1$  i  $\tau_2$ . Matematički model

klatna prikazanog na slici 1 predstavljen je sa (3) [5] gde su:

- $m_1$  i  $m_2$  – mase prvog i drugog klatna
- $l_1$  i  $l_2$  – dužine prvog i drugog klatna
- $\theta_1$  i  $\theta_2$  – uglovi otklona oba klatna u odnosu na vertikalu
- $\tau_1$  i  $\tau_2$  – momenti sile koje aktuatori generišu u zglobovima prvog i drugog klatna redom
- $g$  – univerzalna gravitaciona konstanta

$$\begin{aligned} (m_1 + m_2)l_1^2 \frac{d^2\theta_1}{dt^2} + m_2l_1l_2 \frac{d^2\theta_2}{dt^2} \cos(\theta_1 - \theta_2) \\ + m_2l_1l_2 \left(\frac{d\theta_2}{dt}\right)^2 \sin(\theta_1 - \theta_2) \\ + (m_1 + m_2)l_1g \sin(\theta_1) = \tau_1 \\ m_2l_2^2 \frac{d^2\theta_2}{dt^2} + m_2l_1l_2 \frac{d^2\theta_1}{dt^2} \cos(\theta_1 - \theta_2) \\ - m_2l_1l_2 \left(\frac{d\theta_1}{dt}\right)^2 \sin(\theta_1 - \theta_2) \\ + m_2l_2g \sin(\theta_2) = \tau_2 \end{aligned} \quad (3)$$

Ukoliko se definišu vektori (4) i matrice (5), ovaj model moguće je predstaviti i u matricnoj formi (6).

$$\begin{aligned} \theta &= \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \\ \tau &= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \end{aligned} \quad (4)$$

$$\begin{aligned} M &= \begin{bmatrix} (m_1 + m_2)l_1^2 & m_2l_1l_2 \cos(\theta_1 - \theta_2) \\ m_2l_1l_2 \cos(\theta_1 - \theta_2) & m_2l_2^2 \end{bmatrix} \\ C &= \begin{bmatrix} 0 & -m_2l_1l_2 \sin(\theta_1 - \theta_2) \\ m_2l_1l_2 \sin(\theta_1 - \theta_2) & 0 \end{bmatrix} \\ G &= \begin{bmatrix} -(m_1 + m_2)l_1g & 0 \\ 0 & -m_2l_2g \end{bmatrix} \end{aligned} \quad (5)$$

$$M\ddot{\theta} = C\dot{\theta}^2 + G \sin(\theta) + \tau \quad (6)$$

## 4. METOD

U ovom poglavlju prikazana su dva algoritma koja su korišćena za upravljanje uglovima otklona  $\theta_1$  i  $\theta_2$ . Prvi je upravljački algoritam regulatora po stanjima, koji potiče iz teorije upravljanja [6], a drugi je baziran na učenju sa potkrepljenjem. U oba slučaja za projektovanje regulatora, odnosno za obuku agenta korišćene su vrednosti parametara kao i referentne vrednosti uglova otklona date sa (7).

$$\begin{aligned} m_1 = m_2 = 1 \text{ kg} \\ l_1 = 1 \text{ m}, l_2 = 0.7 \text{ m} \\ \theta_1^* = \frac{\pi}{2} \text{ rad}, \theta_2^* = \pi \text{ rad} \end{aligned} \quad (7)$$

### 4.1. Projektovanje regulatora po stanjima

Regulator po stanjima projektovan je na osnovu modela sistema, tako da dvostruko klatno dovede u željeno stanje bez izazivanja preskoka.

Pre nego što su se mogli podesiti parametri regulatora, model dvostrukog matematičkog klatna bilo je potrebno linearizovati. Ovo je učinjeno primenom *feedback*

linearizacije [7]. Najpre je model (6) transformisan u oblik (8).

$$\ddot{\theta} = M^{-1}(C\dot{\theta}^2 + G \sin(\theta) + \tau) \quad (8)$$

Zatim je izraz sa desne strane znaka jednakosti u (8) zamenjen virtuelnim upravljačkim signalom  $\mathbf{v}$ , te se dobija novi model (9).

$$\begin{aligned} \ddot{\theta} &= \mathbf{v} \\ \mathbf{v} &= \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \end{aligned} \quad (9)$$

Upravljački signali  $v_1$  i  $v_2$  određeni su pomoću regulatora po stanjima čiji je upravljački zakon dat sa (10).

$$\begin{aligned} v_1 &= k_{10}\theta_1^* - k_{11}\dot{\theta}_1 - k_{12}\ddot{\theta}_1 \\ v_2 &= k_{20}\theta_2^* - k_{21}\dot{\theta}_2 - k_{22}\ddot{\theta}_2 \end{aligned} \quad (10)$$

Parametri u (10) su određeni metodom postavljanja polova [6]. Svi polovi postavljeni su na vrednost  $-3$  tako da obezbede stabilan odziv bez preskoka. Za tako izabrane polove, parametri regulatora su postavljeni na vrednosti date sa (11).

$$\begin{aligned} k_{10} &= k_{11} = k_{20} = k_{21} = 9 \\ k_{12} &= k_{22} = 6 \end{aligned} \quad (11)$$

## 4.2. Projektovanje i obuka agenta zasnovanog na učenju sa potkrepljenjem

Primenom učenja sa potkrepljenjem obučeni su agenti čiji je zadatak bio da postave uglove otklona dvostrukog matematičkog klatna na željene vrednosti. Obučena su dva agenta, po jedan za svaku upravljačku veličinu. Pre obučavanja agenata bilo je potrebno izvršiti diskretizaciju modela kako po vremenu tako i po amplitudi da bi se mogao definisati skup sa konačnim brojem stanja i akcija. Na osnovu diskretizovanog modela, formiran je simulator dvostrukog klatna u kom se u svakom koraku vršila kvantizacija svih stanja. Uglovima otklona mogla je biti dodeljena vrednost iz skup sa 40 ekvidistantnih vrednosti u opsegu od  $0 \text{ rad}$  do  $2\pi \text{ rad}$ , dok su ugaone brzine mogle uzimati vrednosti iz skup sa 11 ekvidistantnih vrednosti u opsegu od  $-4 \frac{\text{rad}}{\text{s}}$  do  $4 \frac{\text{rad}}{\text{s}}$ . Agenti su mogli da primene po jednu od 20 ekvidistantnih akcija u opsegu od  $-20 \text{ Nm}$  do  $50 \text{ Nm}$ . Kako bi se omogućilo stanje mirovanja, u skup mogućih akcija dodata je još i 21. akcija primenom koje se na aktuatoru ostvaruje moment sile od  $0 \text{ Nm}$ . Akcije čijom primenom se ostvaruju momenti sile manji od  $0 \text{ Nm}$  služile su isključivo da pruže agentima mogućnost da izvrše korekciju ukoliko dođe do preskoka u odzivu, te je donja granica opsega po apsolutnoj vrednosti znatno manja u odnosu na gornju.

Perioda odabiranja za vremensku diskretizaciju modela  $T$  postavljena je na  $0.05\text{s}$ . Agenti su obučeni primenom  $Q$ -learning algoritma sa  $\varepsilon$ -pohlepnom politikom odlučivanja. Vrednost parametra  $\varepsilon$  je inicijalno postavljena na 1, a zatim je u svakom narednom koraku smanjivana na takav način da nakon ispunjenih 40% od ukupnog broja koraka dostigne vrednost 0.1 nakon čega više nije menjana. Stopa učenja  $\alpha$  postavljena je na 0.2, a faktor obezvređivanja  $\gamma$  na 0.95. Dimenzija svake  $Q$ -matrice je na osnovu broja mogućih stanja i akcija bila  $40 \times 11 \times 40 \times 11 \times 21$ . Ukupan broj elemenata svake  $Q$ -matrice je

stoga bio 4065600. Na osnovu ovih dimenzija  $Q$ -matrica, broj koraka za obuku agenata postavljen je na  $4.2 \times 10^8$  kao kompromis između potrebe da se simulacija završi u razumnom vremenskom roku i da se agentu omogući da primeni svaku moguću akciju u svakom stanju. Nagrade koje su agenti ostvarivali za svaku primenjenu akciju su definisane na osnovu trenutnog i željenog stanja sistema po ugledu na [8] po pravilu (12). Ideja na ovaj način definisane nagrade jeste da se agent „kazni“ ukoliko poveća grešku tako što mu se dodeli negativna nagrada.

Takođe, u slučaju porasta greške usled određene akcije, nagrada, odnosno kazna je skalirana sa po apsolutnoj vrednosti većim faktorom u cilju što efikasnijeg obeshrabrivanja agenta da ponovi tu akciju u istom stanju.

$$r_{t+1} = \begin{cases} -5(e_{t+1} - e_t), & e_{t+1} \geq e_t \\ 2(e_t - e_{t+1}), & e_{t+1} < e_t \end{cases} \quad (12)$$

U (12) sa  $e_t$  je označena greška odgovarajućeg ugla otklona u odnosu na željenu vrednost u trenutku pre primene akcije, dok je sa  $e_{t+1}$  označena ista veličina u trenutku nakon primene akcije. Izraz za računanje greške dat je sa (13).

$$e_t = \theta^* - \theta(t) \quad (13)$$

## 4.3. Testiranje upravljačkih algoritama

Nakon obuke upravljački algoritmi su testirani na modelima sa izmenjenim parametrima u odnosu na model koji je korišćen za projektovanje regulatora po stanjima, odnosno obuku agenta.

## 5. REZULTATI I DISKUSIJA

U ovom poglavlju prikazani su rezultati upravljanja dvostrukim klatnom primenom algoritama predstavljenih u poglavlju 4. Ispitana su 3 sučaja tako što su korišćena 3 različita modela:

1. Model čiji su parametri isti kao parametri modela koji je korišćen za projektovanje regulatora i obuku agenta
2. Model čiji su parametri  $m_1$  i  $m_2$  za 40% manji u odnosu na parametre modela koji je korišćen za projektovanje regulatora i obuku agenta
3. Model čiji su parametri  $m_1$  i  $m_2$  za 40% veći u odnosu na parametre modela koji je korišćen za projektovanje regulatora i obuku agenta

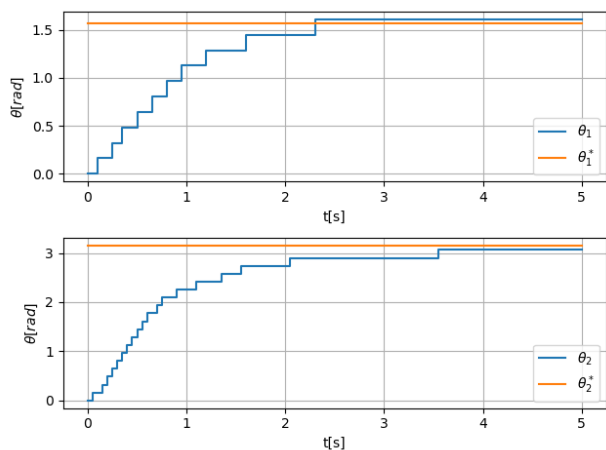
### 5.1. Rezultati slučaja 1

Na slici 2 prikazano je kretanje uglova otklona u vremenu kao posledica upravljanja primenom regulatora po stanjima u slučaju 1. Kao što je pomenuto u poglavlju 4.1, parametri regulatora po stanjima su postavljeni tako da odziv bude stabilan i bez preskoka što se upravo i vidi na slici 2. Greške u ustaljenom stanju u odnosu na željene vrednosti su u ovom slučaju posledica diskretnog i konačnog broja stanja, odnosno uglova otklona. Relativna greška oba ugla otklona je ispod 3%.

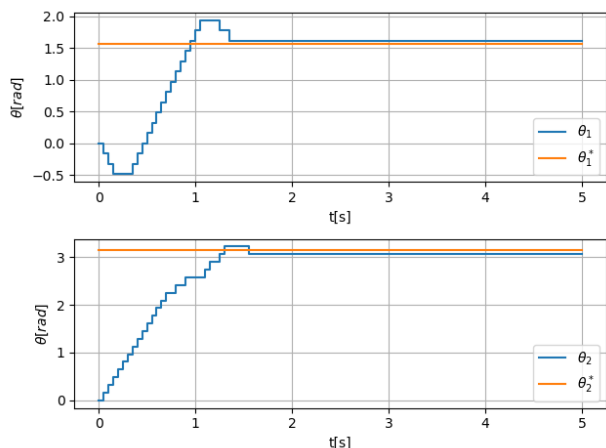
Na slici 3 prikazano je kretanje uglova otklona u vremenu kada je korišćeno upravljanje bazirano na učenju sa potkrepljenjem u slučaju 1. Prva razlika koja se može uočiti u odnosu na upravljanje koje je ostvareno primenom regulatora po stanjima jeste da se u ovom slučaju javlja preskok u odzivu.

Pri definisanju izraza za nagradu nije uveden penal za izazivanje preskoka, te su agenti naučili da će maksimalnu dobit ostvariti primenom ovakvog upravljanja.

Kao i u slučaju sa regulatorom po stanjima, i ovde je greška u ustaljenom stanju u odnosu na željenu vrednost posledica diskretnog i konačnog broja stanja. Relativna greška oba ugla otklona u ovom slučaju je takođe ispod 3%.



Slika 2. Kretanje uglova otklona usled upravljanja regulatorom po stanjima – slučaj 1



Slika 3. Kretanje uglova otklona usled upravljanja baziranom na učenju sa potkrepljenjem – slučaj 1

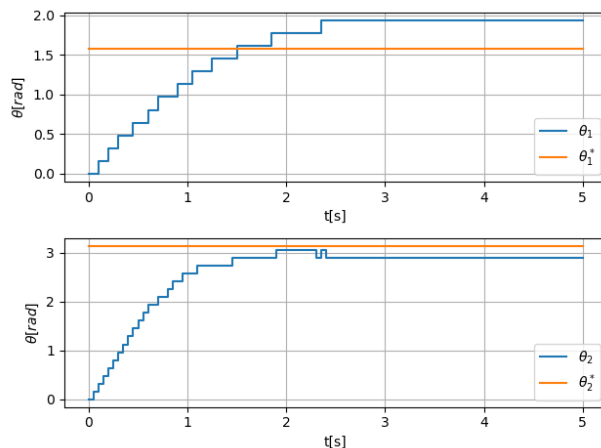
## 5.2. Rezultati slučaja 2

Na slici 4 prikazano je kretanje uglova otklona u vremenu kao posledica upravljanja primenom regulator po stanjima u slučaju 2. Zbog promene modela regulator u ovom slučaju nije uspeo da ostvari željeni položaj dvostrukog klatna kao što je bilo i očekivano. Relativna greška iznosila je približno 23% za ugao otklona  $\theta_1$  i približno 7.7% za ugao otklona  $\theta_2$ .

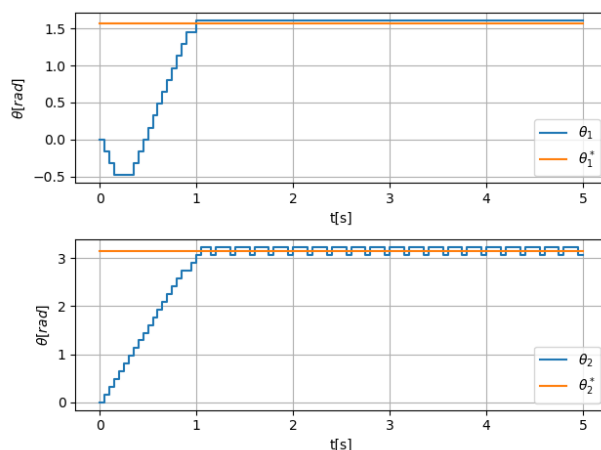
Na slici 5 prikazano je kretanje uglova otklona u vremenu kada je korišćeno upravljanje bazirano na učenju sa potkrepljenjem u slučaju 2. Za razliku od regulatora po stanjima, regulator baziran na učenju sa potkrepljenjem je uspeo da dovede dvostruko klatno u željeni položaj bez obzira što su parametri drugačiji u odnosu na parametre koji su korišćeni za obuku.

Potrebno je naglasiti da nije izvršena ponovna obuka agenata za ovaj slučaj. Relativna greška oba ugla otklona

je i u ovom slučaju ispod 3%. Oscilacije koje se javljaju u blizini željenog stanja ugla otklona  $\theta_2$  posledica su toga što iste akcije drugačije deluju na ovakav model kao i toga što zbog diskretizacije stanja model ne može savršeno da dostigne željenu vrednost.



Slika 4. Kretanje uglova otklona usled upravljanja regulatorom po stanjima – slučaj 2



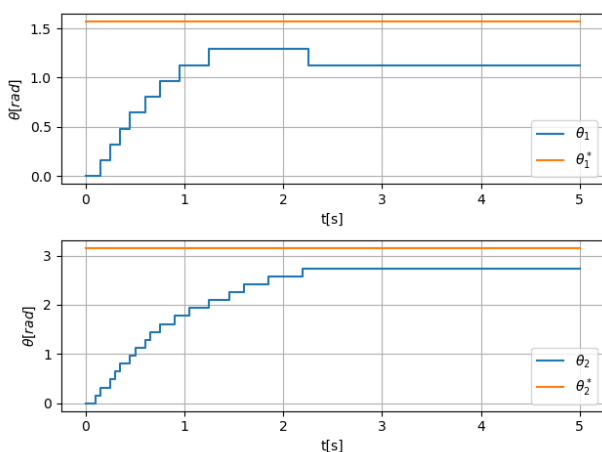
Slika 5. Kretanje uglova otklona usled upravljanja baziranom na učenju sa potkrepljenjem – slučaj 2

## 5.3. Rezultati slučaja 3

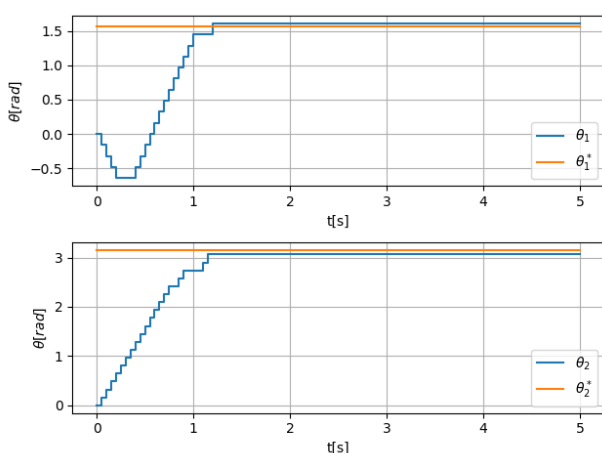
Na slici 6 prikazano je kretanje uglova otklona u vremenu kao posledica upravljanja primenom regulator po stanjima u slučaju 3. I u ovom slučaju zbog promene modela regulator nije uspeo da ostvari željeni položaj dvostrukog klatna. Relativna greška iznosila je približno 28.2% za ugao otklona  $\theta_1$  i približno 12.8% za ugao otklona  $\theta_2$ .

Na slici 7 prikazano je kretanje uglova otklona u vremenu kada je korišćeno upravljanje bazirano na učenju sa potkrepljenjem u slučaju 3. Ni u ovom slučaju nije vršena naknadna obuka agenata, te su sve akcije primenjivane na osnovu obuke opisane u poglavlju 4.

Regulator baziran na učenju sa potkrepljenjem uspeo je da dovede dvostruko klatno u željeni položaj, kao i u prethodnom slučaju. U ovom slučaju nije bilo oscilacija zbog veće inertnosti modela usled povećane mase. Relativna greška oba ugla otklona je i u ovom slučaju ispod 3%.



Slika 6. Kretanje uglova otklona usled upravljanja regulatorom po stanjima – slučaj 3



Slika 7. Kretanje uglova otklona usled upravljanja baziranom na učenju sa potkrepljenjem – slučaj 3

## 6. ZAKLJUČAK

U ovom radu prikazani su rezultati upravljanja dvostrukim matematičkim klatnom primenom regulatora po stanjima i algoritma zasnovanog na učenju sa potkrepljenjem. Primenom regulatora po stanjima pokazano je da se željeno upravljanje postiže samo u slučaju kada se upravlja sistemom čiji parametri odgovaraju parametrima koji su korišćeni za projektovanje regulatora.

Za drugačije parametre, bilo manje ili veće po vrednosti, ovaj regulator nije uspeo da dovede sistem u željeno stanje. Sa druge strane, algoritam baziran na učenju sa potkrepljenjem pokazao je dobre rezultate u sva tri slučaja.

## 7. LITERATURA

- [1] R. Hafner / M. Riedmiller, „Reinforcement learning in feedback control,“ *Machine Learning*, 2011.
- [2] M. Riedmiller, T. Gabel, R. Hafner / S. Lange, „Reinforcement learning for robot soccer,“ *Autonomous Robots*, 2009.
- [3] R. S. Sutton / A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA: MIT Press, 2018.
- [4] C. Watkins / P. Dayan, „Q-Learning,“ *Machine Learning*, t. 8, 1992.
- [5] A. Elbori, „Simulation of Double Pendulum,“ *Quest Journals, Journal of Software Engineering and Simulation*, 2017.
- [6] W. L. Brogan, Modern Control Theory, New Jersey: Prentice Hall, 1991.
- [7] H. K. Khalil, Nonlinear Systems, New Jersey: Prentice Hall, 2002.
- [8] A. Kuhnle, J.-P. Kaiser, F. Theiß, N. Stricker / G. Lanza, „Designing an adaptive production control system using reinforcement learning,“ *Journal of Intelligent Manufacturing*, 2020.

### Kratka biografija:



**Bojan Jorgovanović** rođen je u Novom Sadu 1998. godine. Diplomski rad na temu *Modelovanje sistema zasnovano na veštačkim neuronskim mrežama* iz oblasti Elektrotehnike i računarstva odbranio je na Fakultetu tehničkih nauka 2021. godine.

Kontakt: bojan.jorgovanovic@uns.ac.rs