

## DINAMIČKA REKONFIGURACIJA DISTRIBUTIVNE MREŽE PRIMENOM UČENJA POTKREPLJIVANJEM

### DEEP REINFORCEMENT LEARNING BASED DYNAMIC DISTRIBUTION NETWORK RECONFIGURATION

Milan Petković, Predrag Vidović, *Fakultet tehničkih nauka, Novi Sad*

#### Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

**Kratak sadržaj** – U radu je obrađena dinamička rekonfiguracija distributivne mreže primenom učenja potkrepljivanjem. Problem je modelovan višekriterijumskom funkcijom cilja koja minimizira gubitke energije i gubitke usled manipulacija prekidačima. Broj informacija neophodnih za izvršenje algoritma je redukovan, s obzirom da su topološke informacije i informacije o tokovima snaga smeštene u jedan skup promenljivih. Ovo smanjuje količinu telemetrisanih merenja potrebnih za potencijalno izvršenje algoritma na realnoj mreži i pogodno je sa aspekta treniranja, zbog toga što je smanjena potrebna veličina neuronske mreže.

**Ključne reči:** dinamička rekonfiguracija, učenje potkrepljivanjem

**Abstract** – This paper proposes DDNR based on Reinforcement Learning algorithm. It is multi-objective approach which minimizes cost of energy losses and switching manipulations. The amount of information needed for the algorithm execution is decreased, since the topology information and the information about the power flows in the network are compressed in the single set of variables. This reduces the amount of telemetered measurements needed for the potential real-world execution of the algorithm and it is also convenient from the algorithm training perspective, since the required size of the neural network is reduced.

**Keywords:** dynamic network reconfiguration, reinforcement learning

#### 1. UVOD

Rekonfiguracija distributivne mreže (DM) široko je korišćena energetska aplikacija za upravljanje DM. Pre svega, rekonfiguracija se koristi za minimizaciju gubitaka električne energije [1, 2] i odstupanja napona od nominalnih vrednosti [3]. Dodatno, koristi se i za balansiranje opterećenja po fiderima i transformatorima, smanjenje operativnih troškova, kao i u mnogim drugim funkcijama za upravljanje DM [1]. Generalno, postoje dva tipa rekonfiguracije DM, statička i dinamička [4]. Cilj statičke rekonfiguracije jeste optimizacija u tačno određenom vremenskom trenutku, dok se dinamička rekonfiguracija bavi optimizacijom u određenom vremenskom intervalu, npr. 24 časa.

#### NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Predrag Vidović, vanr. prof..

Iz tog razloga, dinamička rekonfiguracija pogodna je za aplikacije u realnom vremenu jer se potrošnje, proizvodnje i ostali uslovi u DM menjaju u vremenu. Dinamičkom rekonfiguracijom DM mogu se postići veći benefiti i bolje performanse mreže nego statičkom rekonfiguracijom [4]. Međutim, dinamička rekonfiguracija koristi se samo u potpuno automatizovanim DM.

#### 2. OPTIMALNA REKONFIGURACIJA DISTRIBUTIVNE MREŽE

Funkcija cilja dinamičke rekonfiguracije u ovom radu predstavlja minimizaciju ukupnih troškova usled gubitaka aktivne snage i prekidačkih manipulacija, dok su ograničenja bilansi aktivnih i reaktivnih snaga, naponi čvorova u dozvoljenim granicama, ograničenje struja po granama i radijalan pogon DM.

Funkcija cilja definisana je sledećom relacijom [4]:

$$\min \left\{ \sum_{t=1}^T C_{Loss} \cdot T^t \cdot P_{Loss}^t + C_{SWs} \cdot SW^t \right\} \quad (2.1)$$

gde je sa  $T^t$  naznačeno trajanje vremenskog intervala  $t$ , a sa  $P_{Loss}^t$  i  $SW^t$  gubici aktivne snage i prekidačke manipulacije:

$$P_{Loss}^t = \sum_{b=1}^{N_{Br}} \alpha_b^t R_b \frac{(P_b^t)^2 + (Q_b^t)^2}{(U_j^t)^2}, \quad (2.2)$$

$$SW^t = \sum_{s=1}^{N_{SW}} y_s^t, \quad (2.3)$$

gde su sa  $N_{Br}$  i  $N_{SW}$  naznačeni ukupan broj grana i ukupan broj prekidača, respektivno.

Sa  $\alpha_b^t$  i  $R_b$  naznačene su pomoćna promenljiva i rezistansa grane  $b$ , respektivno, a sa  $P_b^t$  i  $Q_b^t$  aktivne i reaktivne snage u vremenskom intervalu  $t$ . Njihove vrednosti definisane su relacijama:

$$\alpha_b^t = \begin{cases} x_s^t & \text{grana } b \text{ ima prekidač;} \\ 1 & \text{grana } b \text{ nema prekidač;} \end{cases} \quad (2.4)$$

$$x_s^t = \begin{cases} 1 & \text{prekidač } s \text{ je zatvoren u intervalu } t; \\ 0 & \text{prekidač } s \text{ je otvoren u intervalu } t, \end{cases} \quad (2.5)$$

$$P_b^t = \left\{ g_{jk} (U_j^t)^2 - U_j^t U_k^t [g_{jk} \cos(\theta_j^t - \theta_k^t) + b_{jk} \sin(\theta_j^t - \theta_k^t)] \right\} \quad (2.6)$$

$$Q_b^t = \left\{ -b_{jk} (U_j^t)^2 + U_j^t U_k^t [b_{jk} \cos(\theta_j^t - \theta_k^t) - g_{jk} \sin(\theta_j^t - \theta_k^t)] \right\} \quad (2.7)$$

Ograničenja su:

1. Ograničenje bilansa aktivnih i reaktivnih snaga ( $k=1, 2, \dots, N_N, b=j-k$ ):

$$P_j^t = U_j^t \sum_{k=1}^{N_N} U_k^t [g_{jk} \cos(\theta_j^t - \theta_k^t) + b_{jk} \sin(\theta_j^t - \theta_k^t)], \quad (2.8)$$

$$Q_j^t = U_j^t \sum_{k=1}^{N_N} U_k^t [g_{jk} \sin(\theta_j^t - \theta_k^t) - b_{jk} \cos(\theta_j^t - \theta_k^t)]. \quad (2.9)$$

gde je  $N_N$  broj čvorova distributivne mreže.

2. Naponsko ograničenje ( $j=1, 2, \dots, N_N$ ):

$$U_j^{\min} \leq U_j^t \leq U_j^{\max}, \quad (2.10)$$

gde su  $U_j^{\min}$  i  $U_j^{\max}$  minimalni i maksimalni napon čvora  $j$  izraženi u kV, respektivno.

3. Strujno ograničenje po granama ( $b=1, 2, \dots, N_{Br}$ ):

$$(P_b^t)^2 + (Q_b^t)^2 \leq (S_b^{\max})^2, \quad (2.11)$$

gde je  $S_b^{\max}$  maksimalna dozvoljena prividna snaga grane  $b$  izražena u VA.

4. Ograničenje radikalnosti:

$$\sum_{b=1}^{N_{Br}} \alpha_b^t = N_N - 1. \quad (2.12)$$

### 3. UČENJE POTKREPLJIVANJEM

Učenje potkrepljivanjem (eng. Reinforcement Learning, skr. RL), kao vid mašinskog učenja, bavi se načinom učenja agenta da preduzima akcije u okruženju iskustvom i istraživanjem, u cilju nalaženja optimalne strategije koja maksimizira dobijenu dugoročnu nagradu [5]. Agent predstavlja bilo koji autonomno ili polu-autonomno pogonjen sistem baziran na veštačkoj inteligenciji koji koristi duboko učenje da izvršava svoje zadatke i unapređuje način donošenja odluka u okruženju [6]. Nagrada predstavlja numerički signal koji agent dobija od okruženja. Nagradom je implicitno objašnjeno šta je dobro, a šta loše ponašanje. Politika opisuje način na koji se agent ponaša [7].

### 4. OPTIMALNA REKONFIGURACIJA PRIMENOM ALGORITMA UČENJA POTKREPLJIVANJEM

U ovom poglavlju opisan je način modelovanja dinamičke rekonfiguracije DM kao problem učenja potkrepljivanjem, kao i funkcija cilja i ograničenja. Takođe, detaljno je opisan algoritam za treniranje modela učenja potkrepljivanjem.

#### 4.1. Modelovanje dinamičke rekonfiguracije kao Markovljev proces odlučivanja

Razmena informacija između agenta i okruženja prilikom njihove interakcije prikazana je na slici 4.1.1.

Epizode se sastoje od 24 koraka i na početku svakog koraka zadaju se vrednosti aktivne i reaktivne snage u potrošačkim čvorovima za naredni sat.

Zatim se izvršava proračun tokova snaga, kako bi se kreirale promenljive stanja, a jedan skup promenljivih stanja sadrži broj koraka i prividne snage svih prekidača u mreži.

Biraju se prividne snage prekidača, a ne aktivne i reaktivne snage, kako bi broj promenljivih stanja bio manji. Dakle, ovakav način izbora promenljivih stanja smanjuje dimenziju prostora stanja kao i veličinu duboke neuronske mreže (eng. Deep Q Network, skr. DQN) zbog toga što su topološke informacije (statusi prekidača) i rezultati tokova snaga sačuvani u jedan skup promenljivih, iz kojeg agent može da kreira svoju reprezentaciju okruženja i poveže ih sa akcijama i dobijenim nagradama.

Prostor akcija sadrži sve kombinacije prekidača koje ne narušavaju radikalnost DM i gde su svi potrošači pod napajanjem.

Ove kombinacije prekidača su numerisane na jedinstven način tako da jednom izlaznom neuronu odgovara jedna kombinacija prekidača koja zadovoljava gorenavedene uslove. Ovakav način definisanja prostora akcija implicira da je radikalnost mreže uvek zadovoljena, što ubrzava proces učenja.



Slika 4.1.1. Interakcija agenta i okruženja za dinamičku rekonfiguraciju DM

#### 4.2. Algoritam za treniranje modela učenja potkrepljivanjem

Tokom treniranja agenta izvrši se  $N$  epizoda, a svaka epizoda sastoji se od predefinisano broja koraka  $T$ . Raznolikost scenarija prilikom treninga postiže se dodavanjem šuma na dnevne dijagrame opterećenja. Korišćeno je linearno opadanje metaparametra  $\epsilon$  do  $0.8 \cdot N$ -te epizode, a nakon toga, do kraja treninga, konstantna vrednost. Detaljan prikaz algoritma nalazi se na slici 4.2.1.

#### Algoritam: Treniranje duboke Q mreže

```

Inicijalizovati parametre duboke Q mreže  $Q(s | \theta^Q)$  nasumično
Inicijalizovati parametre ciljne duboke Q mreže  $Q'(s | \theta^{Q_{target}})$  koristeći
parametre originalne mreže
Inicijalizovati memoriju interakcija
for epizoda = 1, 2, ..., N do
  Nasumično uzeti uzorak dnevnih krivih opterećenja
  Inicijalizovati  $\epsilon$ 
  Inicijalizovati podskup dostupnih akcija na skup akcija
  Izvršiti inicijalni proračun tokova snaga
  Poslati inicijalno stanje  $s_1$  agentu
  for  $t = 1, 2, \dots, T$  do
    rand = nasumičan broj između 0 i 1
    if rand >  $\epsilon$  then
      Izabrati akciju  $a$  sa najvećom Q-vrednošću iz podskupa
      dostupnih akcija
    else
      Izabrati akciju  $a$  nasumično iz podskupa dostupnih akcija
    end
    Ažurirati podskup dostupnih akcija
    Ažurirati konfiguraciju mreže prema  $a_t$ 
    Izvršiti proračun tokova snaga
    Sačuvati neposrednu nagradu  $r_t$  i podatke za naredno stanje  $s_{t+1}$ 
    Skladištiti skup  $(s_t, a_t, r_t, s_{t+1})$  u memoriju interakcija
    Uzeti uzorak skupa za obučavanje (minibatch) iz memorije
    interakcija
    Kreirati oznake (label) za treniranje duboke Q mreže
    Ažurirati parametre duboke Q mreže minimizirajući funkciju
    greške
    Postaviti opterećenja za sledeći vremenski trenutak za svaki čvor
  end
  if ažurirati period preračunavanja parametara ciljne mreže then
     $\theta^{Q_{target}} = \theta^Q$ 
  Ažurirati  $\epsilon$ 
end

```

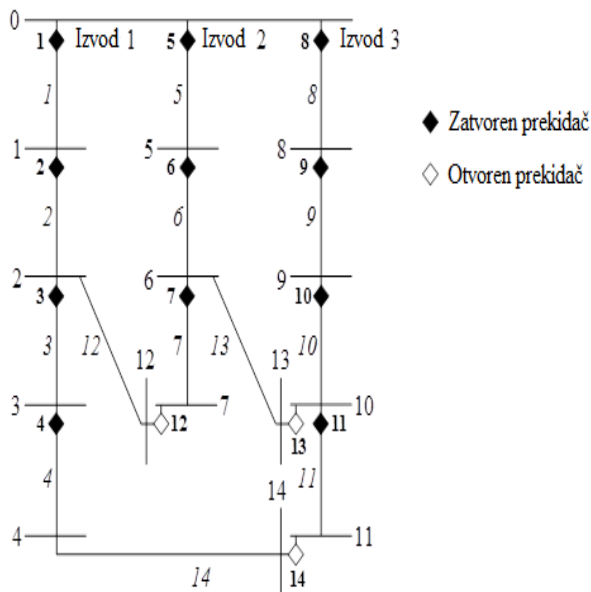
Slika 4.2.1. Algoritam za treniranje modela učenja potkrepljivanjem

## 5. VERIFIKACIJA MATEMATIČKOG MODELA OPTIMALNE REKONFIGURACIJE PRIMENOM ALGORITMA UČENJA POTKREPLJIVANJEM

U ovoj glavi prikazani su rezultati proračuna za test mrežu od 15 čvorova [4] kao i izbor metaparametara neuronske mreže i algoritma učenja potkrepljivanjem. Algoritam je napisan u programskom jeziku Python, duboka neuronska mreža modelovana je i trenirana korišćenjem programskog okvira za duboko učenje = PyTorch, dok je test mreža modelovana korišćenjem softvera za simulaciju elektrodistributivnih sistema = OpenDSSDirect [8], gde su vršeni i proračuni tokova snaga. Algoritam je izvršavan na 64-bitnom računaru sa operativnim sistemom Windows 10 i sledećom hardverskom konfiguracijom: AMD A8-6410 APU sa AMD Radeon R5 Graphics 2.00 GHz, 4 jezgra i 8GB RAM-a.

### Test mreža

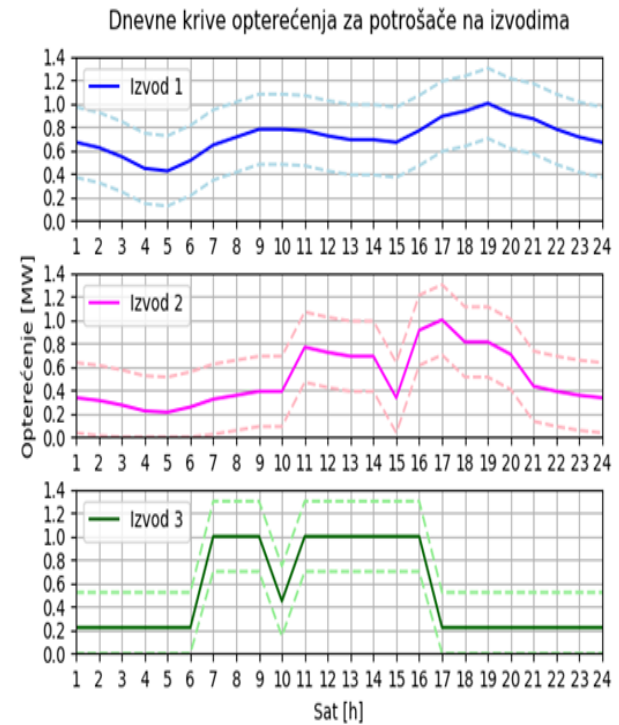
Slika 5.1 ilustruje test mrežu od 15 čvorova, gde je koren mreže označen sa 0, a preostalih 14 čvorova su tipa PQ. Potrošnje su definisane dnevnim hronološkim dijagramima opterećenja, slika 5.2, i maksimalnim opterećenjem (1MVA). Dužina svih grana je ista i iznosi 4.5 km. Sve grane su uravnotežene sa impedansom direktnog redosleda  $r + jx = (0.224 + j0.109) \Omega/\text{km}$ . Svaka grana poseduje po jedan prekidač, pri čemu je početno uklopno stanje prikazano na slici 5.1.



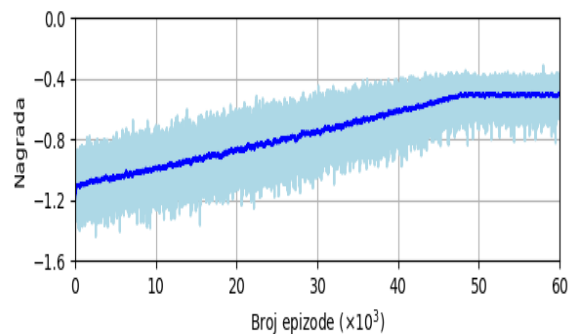
Slika 5.1. Jednopolna šema test mreže od 15 čvorova [4]

Algoritam je treniran kroz 60000 epizoda. Treniranje sa većim brojem epizoda od navedenog dalo je identične rezultate. Jedna epizoda sastoji se iz 24 vremenska koraka, a jedan korak definisan je uklopnim stanjem, odnosno normalno otvorenim prekidačima. Tokom treninga, opterećenja za svaki sat slučajno su birana iz intervala prikazanih na izlomljenim krivim na slici 5.2. Ciljna DQN ažurirana je na svakih 10 epizoda, a skupovi za obučavanje za DQN trening birani su iz memorije interakcija koja ima kapacitet od 1000000 uzoraka. Početna vrednost metaparametra  $\epsilon$  je 1 i opada linearno do vrednosti 0.1 u 48000. epizodi, a zatim je konstantna do kraja treninga. Ovim načinom opadanja metaparametra  $\epsilon$  ostvaruje se dobar balans između istraživanja okruženja

i iskorišćavanja naučenog tokom treninga pri biranju akcija. Vrednost metaparametra umanjena  $\gamma$  iznosi 0.99. Slika 5.3 prikazuje prosečnu vrednost funkcije greške po epizodi. Kako trening odmiče, DQN greška se smanjuje, što implicira da je Q-funkcija uspešno aproksimirana. Dodatno, neposredne nagrade po epizodi i njihova srednja vrednost prikazani su na slici 5.3. Približno linearno povećanje usrednjene neposredne nagrade posledica je linearnog smanjenja metaparametra  $\epsilon$  do 48000. epizode, iz razloga što se povećava verovatnoća izvršenja akcije predložene od DQN umesto biranja slučajne akcije.



Slika 5.2. Dnevni dijagrami opterećenja potrošača na izvodima 1, 2, 3



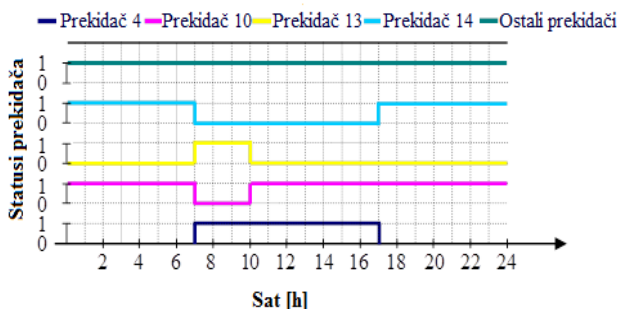
Slika 5.3. Prosečni gubici DQN po epizodi (gore) i ukupna nagrada po epizodi (dole)

U Tabeli 5.1 prikazani su ukupni gubici u kW, broj manipulacija prekidačima i ukupna cena u \$ za predloženi pristup dinamičkoj rekonfiguraciji, metod iz rada [9] i kad se ne vrše manipulacije prekidačima.

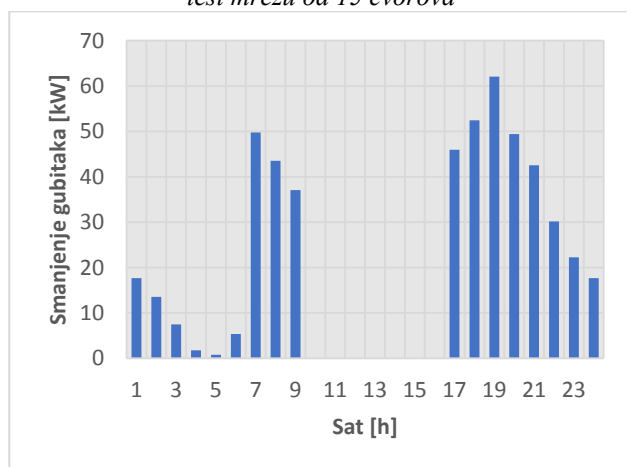
Tabela 5.1. Ukupni gubici, broj manipulacija prekidačima i ukupna cena za 24-časovni period optimizacije test mreže od 15 čvorova

	Predloženi pristup	Metod iz rada [9]	Bez manipulacija
Ukupni gubici [kW]	5978.48	5967.56	6477.81
Broj manipulacija prekidačima	10	22	0
Ukupna cena [\$]	402.3	413.6	425.1

Na osnovu Tabele 5.1 može se zaključiti da predloženi pristup dinamičkoj rekonfiguraciji daje bolje rešenje nego metod iz rada [9]. Takođe, predloženi pristup ima nešto malo veće ukupne gubitke aktivne snage, ali zato ima znatno manje manipulacija prekidačima i manje ukupne troškove. Trajanje izvršenja programa kada je algoritam istreniran iznosi 0.148s. Grafički prikaz manipulacija prekidačima prikazan je na slici 5.4, a smanjenje gubitaka aktivne snage tokom 24 časa, u odnosu na gubitke aktivne snage kad se ne vrše manipulacije prekidačima, prikazano je na slici 5.5.



Slika 5.4. Menjanje statusa prekidača tokom 24 časa za test mrežu od 15 čvorova



Slika 5.5. Smanjenje gubitaka aktivne snage korišćenjem dinamičke rekonfiguracije uz pomoć učenja potkrepljivanjem

## 6. ZAKLJUČAK

U ovom radu razmatran je problem dinamičke rekonfiguracije distributivne mreže sa višekriterijumskom

funkcijom cilja čiji je cilj minimizacija ukupnih troškova usled gubitaka aktivne snage i prekidačkih manipulacija. U radu je izložena dinamička rekonfiguracija uz pomoć učenja potkrepljivanjem.

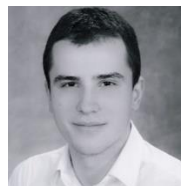
Na osnovu prikazanih rezultata, može se zaključiti:

1. algoritam učenja potkrepljivanjem se brzo izvršava
2. prikazani algoritam ima znatno manje manipulacija prekidačima nego algoritam baziran na statičkoj rekonfiguraciji [9]
3. ukupni troškovi usled gubitaka aktivne snage i prekidačkih manipulacija manji su nego u slučaju statičke rekonfiguracije

## 7. LITERATURA

- [1] G.Švenda: Specijalizovani softveri u elektroenergetici, skripta sa predavanja iz istoimenog predmeta na master studijama, Fakultet Tehničkih Nauka, Novi Sad, 2019.
- [2] Taleski R, Rajicic D. Distribution network reconfiguration for energy loss reduction. IEEE Transactions on Power Systems 1997; 1:398–406.
- [3] V.Fathi, H.Seyedi, B.M.Ivatloo: Reconfiguration of distribution systems in the presence of distributed generation considering protective constraints and uncertainties; Int. Trans Electr Energy Systems; February 2020;pp. 1-25; DOI: 10.1002/2050-7038.12346
- [4] Kovački NV, Vidović PM, Sarić AT. Scalable algorithm for the dynamic reconfiguration of the distribution network using the Lagrange relaxation approach. International Journal of Electrical Power & Energy Systems 2018; 94:188–202.
- [5] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press, 2018.
- [6] <https://searchenterpriseai.techtarget.com/definition/deep-learning-agent> 10.09.2020.
- [7] Nikolić M, Zečević A. Mašinsko učenje, Matematički fakultet, Univerzitet u Beogradu, 2019.
- [8] R. Dugan and T. McDermott. An open source platform for collaborating on smart grid research. Power and Energy Society General Meeting 2011 IEEE 2011; 1–7.
- [9] Ramos ER, Exposito AG, Santos JR, Iborra FL. Path-based distribution network modeling: application to reconfiguration for loss reduction. IEEE Transactions on Power Systems 2005; 2:556–564.

## Kratka biografija



*Milan Petković rođen je u Novom Sadu 1996. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi odbranio je 2020. godine.*



*Predrag Vidović rođen u Vlasenici, 1981. god. Diplomirao je, magistrirao i doktorirao na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektroenergetski sistemi 2005, 2008. i 2015. godine, respektivno.*