

# Modifikacija PBRC algoritma sa primenama u problemima adaptivne regresije

Vukan Turkulov, Stefana Jocić, Milan R. Rapaić, Zoran D. Jeličić

Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Departman za računarstvo i automatiku

Odsek za automatiku, geomatiku i upravljanje sistemima

vukan\_turkulov@uns.ac.rs, stefana.jocic@uns.ac.rs, rapaja@uns.ac.rs, jelicic@uns.ac.rs

*Sažetak* —U okviru ovog rada razmatran je jedan novi pristup problemu predikcije dinamičkih sistema. U predloženom formalizmu, faze obuke i eksploatacije modela nisu strogo razdvojene, već se model dodatno obučava na osnovu podataka raspoloživih u toku njegove eksploatacije. Predloženi pristup je prikladan u situacijama kada se vrši estimacija izlaza vremenski promenljivih (nestacionarnih) procesa kod kojih su merene vrednosti izlaznih veličine periodično, ali sporadično, dostupne, dok su ulazni podaci raspoloživi sa višom učestanošću odabiranja. Predloženo rešenje se zasniva na *Potential-Based Recursive Clustering* (PBRC) algoritmu adaptivne klasifikacije, te je u računskom smislu efikasan i pogodan za implementaciju u realnom vremenu. Efikasnost predloženog postupka ilustrovana je primerima.

*Cljučneriječi* - Regresija, veštačkaneuronskamreža, adaptacija, estimacija, PBRC

## I. UVOD

U okviru ovog rada bavimo se problemom procene (predikcije) izlaza statičkih i dinamičkih sistema na osnovu poznatih ulaznih podataka, tj. bavimo se problemom regresije. Uobičajen pristup rešavanju problema ovog tipa jeste da se na osnovu raspoloživih podataka o prethodnom ponašanju sistema formira (obuči) odgovarajući model, a potom se u eksploataciji odzivi tako dobijenog modela koriste umesto nedostajućih senzora. Postoji čitava lepeza različitih tipova modela koji se koriste u praksi. Među najjednostavnije, ali praktično široko korištene modele spadaju klasične regresije: linearna, polinomijalna, trigonometrijska i druge. Dobro svojstvo ovih modela jeste da se njihova obuka svodi na rešavanje konveksnog optimizacionog problema, što je problem koji sigurno poseduje globalno optimalno rešenje do koga se u računskom smislu po pravilu dolazi bez većih poteškoća [1]. Drugu grupu modela čine modeli zasnovani na principima računarske inteligencije, kao što su veštačke neuronske mreže, najčešće perceptroni (feed-forward neural networks) ili neuronske mreže sa radijalnim baznim funkcijama [2], modeli zasnovani na principima Zadehove fuzzy logike, kao što su Takagi-Sugeno-Kang modeli i drugi. Problem obuke ovih modela je posebno složen kada se ne može svesti na konveksni optimizacioni problem, već se mora rešavati algoritmima i postupcima nekoveksne optimizacije. U zavisnosti od složenosti algoritma, u praksi se koriste

gradijenti algoritam, kvazi-Njutnovi metodi, ali takođe i stohastičke metode, kao što su Genetski algoritam [3], PSO [4] i druge. Posebnu grupu modela čine i tzv. Algoritmi Potpornih Vektora (Support Vector Machines), koji su po strukturi slični RBF NN, ali se mogu obučavati kombinacijom metoda direktne pretrage (koje se primenjuju u pripremnj fazi unakrsne validacije) i konveksne optimizacije (konkretno kvadratnog programiranja) [2]. U tom smislu mana SVM algoritama leži u nemogućnosti neposredne kontrole složenosti algoritma (broja potpornih vektora). Posebno interesantan algoritam koji potpuno izbegava probleme obuke algoritma, ali na uštrb značajnog utroška memorijskih kapaciteta, jeste *General Regression Neural Network* (GRNN) algoritam [5].

Mogući nedostatak svih pomenutih metoda leži upravo u tome da je faza učenja odvojena od faze eksploatacije, te nije jednostavno izvršiti dodatnu obuku, odnosno prilagođenje jednom formiranih modela (izuzetak je GRNN algoritam, koji uopšte ne sadrži fazu učenja). U tom smislu, i u svom osnovnom obliku, svi navedeni modeli se „dobro ponašaju“ u slučaju regresione analize sistema sa nepromenljivim ili sporopromenljivim parametrima.

U okviru ovog rada razmatran je jedan alternativan pristup u kome faze obuke i eksploatacije modela nisu strogo razdvojene, već se model dodatno obučava na osnovu podataka raspoloživih u toku njegove eksploatacije. Predloženi pristup je prikladan u situacijama kada se vrši estimacija izlaza vremenski promenljivih (nestacionarnih) procesa kod kojih su merene vrednosti izlaznih veličina periodično, ali sporadično, dostupne, dok su ulazni podaci raspoloživi sa višom učestanošću odabiranja. Predloženo rešenje se zasniva na PBRC algoritmu adaptivne klasifikacije, te je u računskom smislu efikasan i pogodan za implementaciju u realnom vremenu.

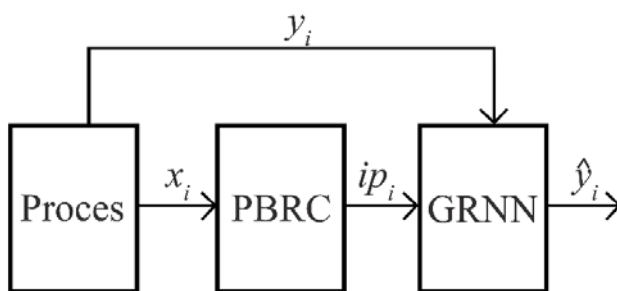
U nastavku je dat opis problema, dok se konkretna novopredložena rešenja nalaze u poglavljima koja slede. Posmatrajmo nepoznatu funkcionalnu zavisnost  $y = f(x, t)$ , gde je  $x$  vektor ulaznih, a  $y$  vektor izlaznih veličina. Pretpostavka

je da funkcija zavisi i od vremena, ali da su promene funkcije u vremenu nekoliko redova veličina sporije od dinamike ulaznih signala. Pretpostavićemo da su vrednosti ulaznih signala odbirkovane periodom  $T_x$ , a izlaznih signala periodom  $T_y$ , tako da je  $T_y > T_x$ . Cilj nam je da estimiramo vrednost izlazne veličine  $y$  u trenucima kada ta vrednost nije direktno dostupna.

## II. ALGORITAM

Predloženo rešenje se sastoji iz dve celine. Prvu celinu čini algoritam koji detektuje i prati režime rada procesa, te u svakom trenutku daje informacije o tome koliko trenutno ponašanje procesa liči na sve do sada viđene režime rada. Te informacije zovemo informacionim potencijalima, kao što je predloženo u [6]. Osim toga, prva celina vrši adaptaciju viđenih režima rada ukoliko se sami režimi promene u dozvoljenim granicama. Ovaj algoritam je realizovan kao modifikovana verzija PBRC algoritma opisanog u [7] i [8].

Zadatak druge celine je da na osnovu do sada poznatih vrednosti  $y$  iz sistema i trenutnih informacionih potencijala iz PBRC-a izračuna estimaciju veličine  $y$ . Ovu estimaciju obeležavaćemo sa  $\hat{y}$ . Ovaj postupak bi mogao biti implementiran veštačkom neuronskom mrežom. Međutim, ukoliko mrežu iznova obučavamo svaki put kada pristigne nova vrednost  $y$  iz procesa, obuka će se vršiti sa periodom  $T_y$ . Da bismo izbegli čestu obuku, možemo primeniti dva pristupa - da ponovno obučavanje vršimo tek kada prikupimo dovoljno veliki broj novih odbiraka  $y$ , ili da mrežu uopšte ne obučavamo nego da druga celina vrši regresiju između dosada viđenih odbiraka. U ovom radu korišćen je drugi pristup, koji polazi od *General Regression Neural Network* (GRNN) algoritma opisanog u [5]. Opšti pregled celog sistema prikazan je na slici 1.



Slika 1. Opšti pregled sistema

### A. PBRC

Angelovi Filevsu u [6] uveli pojam informacionog potencijala u cilju realizovanja metoda za klasifikaciju i identifikaciju. Rapačić i dr. su u [8] opisali ponderisani informacioni potencijal kao pogodniju veličinu za adaptivne metode koje treba da rade u realnom vremenu. Sada ćemo ukratko izložiti metode i pojmove opisane u radovima [7] i [8]. Cilj algoritma je pronalazak potencijala

vektora  $z$  u odnosu na sve prethodno viđene vektore koje obeležavamo skupom  $Z_k$ . Prvo se uvodi pojam otežane srednje kvadratne udaljenosti vektora  $z$  u odnosu na skup  $Z_k$

$$S_\lambda(z, Z_k) = (1 - \lambda) \sum_{i=0}^k \lambda^{k-i} \|z - z_k\|^2. \quad (1)$$

Parametar  $\lambda$  predstavlja faktor zaboravljanja. Što je ovaj parametar veći, PBRC prirodaje više značaja starijim odbircima. Potom se informacioni potencijal računa na sledeći način

$$P_\lambda(z, Z_k) = \frac{1}{1 + S_\lambda(z, Z_k)}. \quad (2)$$

Vrednost informacionog potencijala je u intervalu (0,1]. Što novi odbirak više liči na skorašnje ponašanje procesa, njegov informacioni potencijal je veći. Na sličan način se može odabrati fiksni vektor i tokom rada procesa se računa koliko odabrani vektor odgovara skorašnjem ponašanju procesa. Bitno je napomenuti da se račun za informacioni potencijal može vršiti rekurzivno, što je neophodno za efikasnu implementaciju i u realnom vremenu. Više detalja i svi dokazi iznetih teorema su izloženi u [7] i [8].

U PBRC-u postoji skup vektora koji su reprezentativni za postojeće režime rada procesa. Ove vektore nazivamo fokusima. Koristeći rekurzivne formule, prilikom svakog novopristiglog odbirka računamo trenutnu otežanu srednju kvadratnu udaljenost svakog fokusa u odnosu na dosadašnje ponašanje procesa:

$$S_k(r_i) = (1 - \lambda) \|x_k - r_i\| + \lambda S_{k-1}(r_i) \quad (3)$$

$S_k(r_i)$  je vrednost udaljenosti  $i$ -tog fokusa u trenutku  $k$ . Podrazumevano je da se udaljenost računa u odnosu na celokupni skup  $Z_k$ , te oznaku skupa izostavljamo radi preglednosti.  $x_k$  je ulazni vektor procesa u trenutku  $k$ . Informacioni potencijal svakog fokusa se onda ažurira po formuli (2). Detaljan algoritam je opisan u [7] i [8].

### B. GRNN

D. F. Specht je u [5] predložio GRNN kao rešenje za regresiju kontinualnih promenljivih. GRNN, za razliku od uobičajenih veštačkih neuronskih mreža, ne zahteva obuku. Za potrebe ovog rada GRNN kao ulazni vektor dobija trenutne vrednosti informacionih potencijala svih fokusa na osnovu čega treba estimirati veličine  $y_i$ . Periodično se iz procesa dobija i prava vrednost  $y_i$ , što predstavlja podatke na osnovu kojih GRNN vrši regresiju.

Vektor informacionih potencijala i odgovarajućih izlaza iz sistema koje je GRNN dobio za „obuku“ u prošlosti obeležavaćemo sa  $P_i$  i  $y_i$ . Ove vektore ćemo u nastavku rada zvati podacima za obuku iako se zapravo ne vrši nikakva obuka u uobičajenom smislu te reči. Na osnovu tih podataka, regresija novog vektora  $P$  se računa na sledeći način

$$\hat{y}(P) = \frac{\sum_{i=1}^n y_i \cdot \exp\left(-\frac{\sum_{j=1}^m (P_j - P_j^i)}{2\sigma^2}\right)}{\sum_{i=1}^n \exp\left(-\frac{\sum_{j=1}^m (P_j - P_j^i)}{2\sigma^2}\right)} \quad (4)$$

$n$  je broj dobijenih ulazno-izlaznih parova za obuku,  $m$  je broj fokusa u PBRC-u odnosno dimenzionalnost vektora  $P$ .  $\sigma$  je meta-parametar algoritma koji utiče na strminu regresorske površi. U matematičkom smislu  $\sigma$  je širina kernela za estimaciju koji je u ovom slučaju Gausov.  $P_j^i$  je  $j$ -ti informacijski potencijal iz  $i$ -tog podatka za obuku.

Suštinski, estimacija  $y(P)$  je parametrizovana srednja vrednost svih do sada dobijenih vrednosti  $y_j$ , gde težina svake pojedinačne vrednosti eksponencijalno opada sa razdaljinom  $P - P_j$ . Parametar  $\sigma$  utiče na oblik površi estimacije, kao i na otpornost na šum. Ukoliko je  $\sigma$  veomaveliko,  $y(P)$  će dobiti vrednost najbližeg odbirka čime se smanjuje otpornost na šum, a ukoliko je  $\sigma$  blisko nuli,  $y(P)$  će biti srednja vrednost svih do sada viđenih odbiraka  $y$ .

Naš sistem je namenjen radu u realnom vremenu ili blisko realnom vremenu, što znači da nije praktično čuvati sve prethodno dobijene odbirke za obuku. Ovaj problem je rešen grupisanjem (klasterovanjem) ulaznih odbiraka na način predložen u [5]. Svaka grupa ima svoj centar i dva pridružena parametra,  $A_i$  i  $B_i$ . Parametar  $A_i$  predstavlja zbir izlaza  $y$  svih odbiraka koji pripadaju  $i$ -toj grupi, a parametar  $B_i$  predstavlja broj odbiraka u  $i$ -toj grupi. Prilikom dobijanja novog odbirka za obuku, proverava se udaljenost tog odbirka od centra najbliže grupe. Ukoliko je udaljenost manja od unapred definisane granice  $r$ , ažuriraju se parametri  $A$  i  $B$  za tu grupu. Ukoliko je udaljenost veća od  $r$ , novi odbirak postaje centar nove grupe. Estimacija se onda vrši izmenjenom formulom

$$\hat{y}(P) = \frac{\sum_{i=1}^n A_i \cdot \exp\left(-\frac{\sum_{j=1}^m (P_j - P_j^i)}{2\sigma^2}\right)}{\sum_{i=1}^n B_i \exp\left(-\frac{\sum_{j=1}^m (P_j - P_j^i)}{2\sigma^2}\right)} \quad (5)$$

Najveći deo vremena izvršavanja algoritma predstavlja račun GRNN-a. Taj račun se može znatno ubrzati ukoliko arhitektura računara dozvoljava paralelizam, jer su formule (4) i (5) vrlo pogodne za paralelizaciju.

### C. Predložena hibridizacija

Ideja predloženog rešenja je korišćenje izlaznih vrednosti PBRC-a kao ulazne vrednosti GRNN-a. Drugim rečima, informacijski potencijali svih fokusa zajedno čine ulazni vektor za regresorsku neuronsku mrežu. Ovaj pristup zahteva nekoliko modifikacija na PBRC i GRNN algoritmima.

Prva modifikacija se odnosi na PBRC. Pošto informacijski potencijali fokusa učestvuju u kasnijoj regresiji GRNN-a, njihovo kretanje mora biti ograničeno. Ukoliko bi se fokus previše udaljio od svoje početne pozicije, njemu pridružene vrednosti  $y_i$  u GRNN-u bi postale neažurne. Zbog toga je uvedeno sledeće ponašanje: kada fokus dođe od novog položaja koji je izvan granice dozvoljenog kretanja, vraća se u prvobitni položaj. Potom se na položaj novog odbirka postavi obližnji fokus ukoliko takav postoji, ili napravi novi ukoliko ne postoji odgovarajući fokus. Najveća dozvoljena udaljenost od početne pozicije fokusa obeležava se sa  $d$  i predstavlja jedan od meta-parametara algoritma.

Druga modifikacija se odnosi na GRNN. Naime, za razliku od sistema opisanog u [5], ulazni vektor u GRNN u našem algoritmu menja dimenzionalnost tokom vremena. Svaka pojava novog fokusa u PBRC prouzrokuje povećanje broja dimenzija ulaznog vektora GRNN-a. Da bi GRNN i dalje mogao da računa udaljenost novih vektora u odnosu na stare, prilikom stvaranja novog fokusa u PBRC-u se odmah proširuju i svi vektori informacijskih potencijala u memoriji GRNN-a na odgovarajući broj dimenzija. Nove dimenzije se popunjavaju nulama.

### III. NUMERIČKI PRIMER

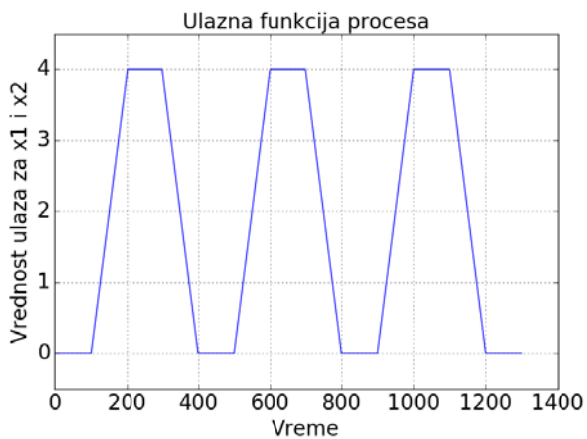
Na jednostavnom primeru ćemo pokazati ponašanje novopredloženog algoritma za predikciju dinamičkog sistema. Sistem koji pokušavamo da estimiramo je linearan, dvodimenzionalan i ima sledeću formu:

$$y = \alpha_1 \cdot x_1 + \alpha_2 \cdot x_2 \quad (5)$$

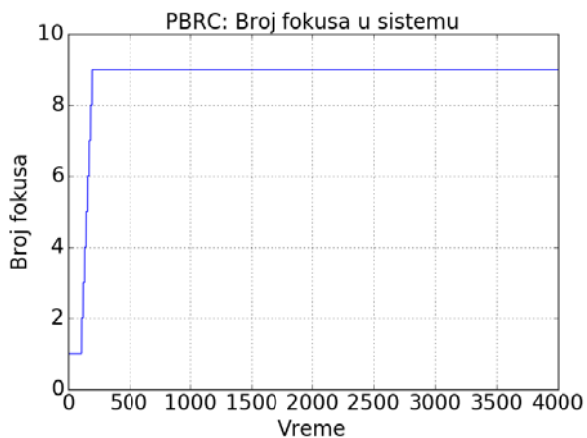
U nastavku primera parametri sistema su postavljeni na  $\alpha_1 = 10$  i  $\alpha_2 = 5$ . Obe ulazne veličine  $x_1$  i  $x_2$  će biti predstavljene istom periodičnom funkcijom prikazanom na slici 2.

Prvo ćemo prikazati ponašanje PBRC-a. Na slici 3 prikazan je broj fokusa u sistemu u zavisnosti od vremena. Pošto je ulazna funkcija periodična, novi fokusi se stvaraju jedino tokom prve periode. Posle toga ulazi se kreću po već postojećim fokusima, te broj fokusa ostaje isti. Na slici 4 su prikazani početni položaji svih 9 fokusa u dvodimenzionalnom prostoru. Fokusi su stvarani na položajima koji odgovaraju kretanju ulaznog vektora, što je i očekivano. Na slici 4 se vide samo početne pozicije fokusa, ali fokusi zapravo menjaju svoje pozicije u dozvoljenim granicama zbog adaptacije. Slika 5 prikazuje promene položaja fokusa tokom vremena. Na slici su zbog preglednosti prikazana samo kretanja koordinate  $x_1$ . Kretanja druge koordinate su slična.

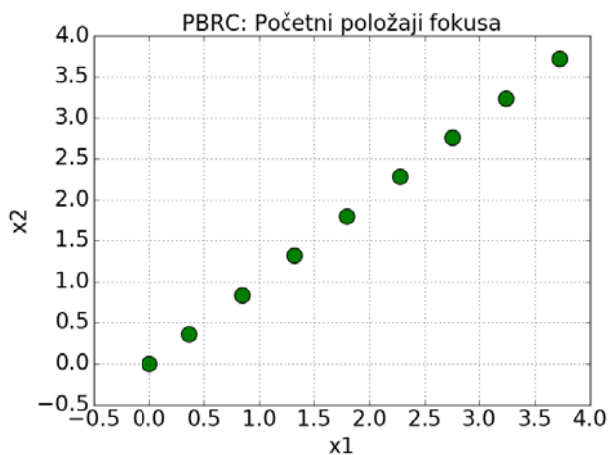
Interesantno je primetiti da se u svakom trenutku može promeniti položaj najviše dva fokusa - jedan koji se vraća na svoj prvobitni položaj i drugi koji zauzima njegov prethodni položaj.



Slika 2. Ulazni signal



Slika 3. Broj fokusa u sistemu

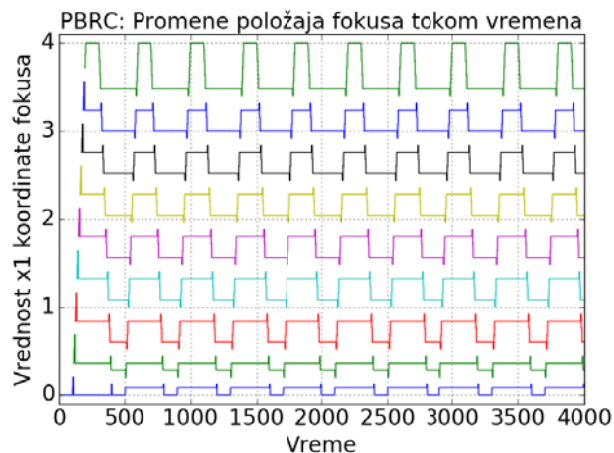


Slika 1. Početni položaji fokusa

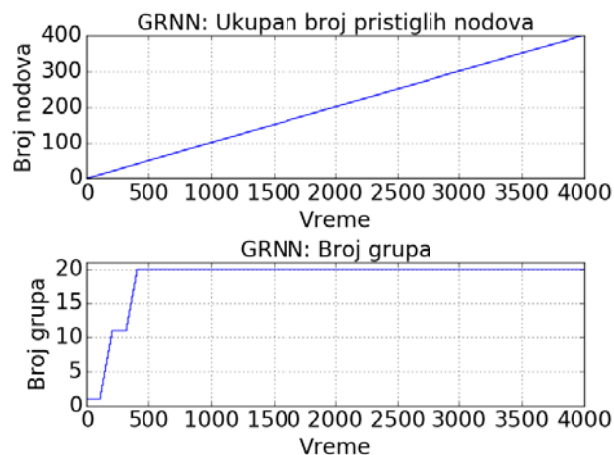
vremenom. Koristeći grupisanje, u našem primeru se efektivno koristi 20 nodova.

Preostalo je još uporediti izlaz GRNN-a (a ujedno i celog sistema) sa podacima koji se dobijaju iz samog procesa. Pošto se ti podaci dobijaju samo periodično, na slici 7 su predstavljeni tačkama, dok je izlaz GRNN-a prikazan linijom.

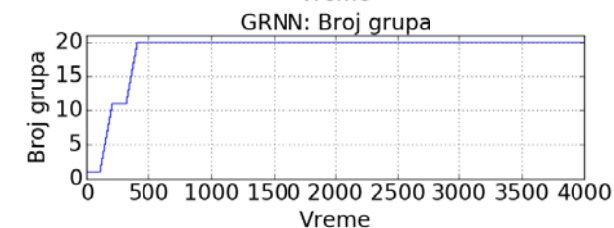
Rezultati pokazuju dobru estimaciju sistema. Ukoliko bi se parametar  $\sigma$  povećao, greška algoritma bi se smanjila, ali bi porasla osetljivost na šum.



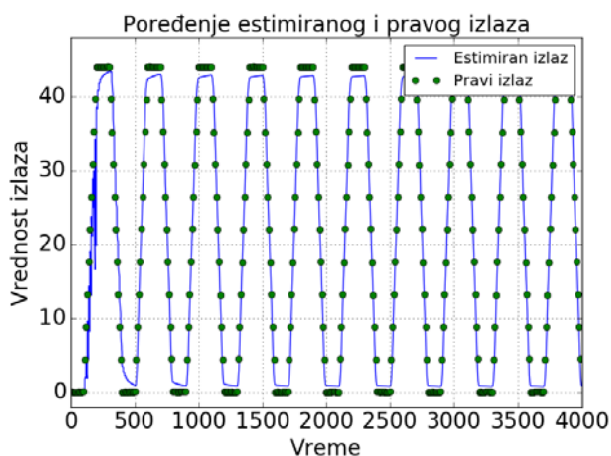
Slika 5. Pomeranje fokusa tokom vremena



Slika 6. Broj nodova u GRNN-u



Sada ćemo prikazati ponašanje GRNN-a. Od interesa nam je da proučimo da li je grupisanje nodova doprinosi održivosti sistema koji treba dugo da radi u realnom vremenu. Slika 6 uporedo prikazuje ukupan broj nodova koji bi postojao u GRNN-u bez grupisanja, i broj grupa u GRNN-u koji zamenjuje pojedinačne nodove. Kao što se vidi, sistem bez grupisanja bi bio neodrživ jer broj nodova linearno raste



Slika 6. Poređenje estimiranog i pravog izlaza

#### ZAKLJUČAK

U okviru ovog rada predloženo je jedno novo rešenje predikcije dinamičkih sistema. Zasnovano je na modifikovanim PBRC i GRNN algoritmima. PBRC je modifikovan tako da ne dozvoljava fokusima neograničeno udaljavanje od svog početnog položaja. GRNN je modifikovan tako da omogućuje povećanje dimenzionalnosti ulaznih vektora.

Količina podataka koji se čuvaju u memoriji raste jedino tokom pojavljivanja novih stanja procesa, što je retka pojava u ustaljenom režimu rada procesa. Takođe, sav račun u PBRC-u se može vršiti rekurzivno. Te dve činjenice omogućuju implementaciju algoritma u integrisanim sistemima koji rade u realnom vremenu. Takvi sistemi često imaju ograničene resurse, što zahteva adekvatne algoritme.

Postoje 4 meta-parametra u algoritmu koje treba podesiti - faktor zaboravljanja  $\lambda$ , maksimalno dozvoljeno kretanje fokusa  $d$ , širina regresorskog kernela  $\sigma$  i poluprečnik grupa GRNN-a  $r$ . Da bi se ovi parametri ispravno podesili, potrebno je

proučiti proces koji se estimira, vršiti simulacije i analizirati rezultate. Ukoliko je jednostavno testirati algoritam na pravom procesu, moguće je algoritamski pronaći meta-parametre koristeći neku metodu optimizacije poput direktne pretrage.

#### LITERATURA

- [1] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge: Cambridge University Press, 2009.
- [2] V. Kecman, *Learning and Soft Computing*. The MIT Press, 2016.
- [3] Ž. Kanović, Z. D. Jeličić and M. R. Rapaić, *Evolutivni optimizacioni algoritmi u inženjerskoj praksi*. Novi Sad: Fakultet tehničkih nauka, 2017.
- [4] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*. Berlin: Springer, 2011.
- [5] D. Specht, "A general regression neural network", *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568-576, 1991.
- [6] P. Angelov and D. Filev, "An Approach to Online Identification of Takagi-Sugeno Fuzzy Models", *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 484-498, 2004.
- [7] M. Petković, M. R. Rapaić, Z. D. Jeličić and A. Pisano, "On-line adaptive clustering for process monitoring and fault detection", *Expert Systems with Applications*, vol. 39, no. 11, pp. 10226-10235, 2012.
- [8] M. R. Rapaić, M. Petković, Z. D. Jeličić and A. Pisano, "An Adaptive Clustering Procedure with Applications to Fault Detection", *Electronics*, vol. 15, no. 2, 2011.

#### ABSTRACT

The output of a dynamic system is frequently estimated using learning-based algorithms which are trained on some previous data. However, acquiring such training data is sometimes costly in terms of time and other resources, making the training data periodically available. Moreover, many systems are time-variant and thus some form of adaptation is required. This paper presents an adaptive real-time estimation method based on sparse periodic training data.

#### MODIFICATION OF THE PBRC ALGORITHM WITH APPLICATIONS IN ADAPTIVE REGRESSION

Vukan Turkulov, Stefana Jocić, Milan R. Rapaić, Zoran D. Jeličić