

# Uporaba ekstrapolacijskih metod pri računanju stacionarnega stanja električnih vezij

Borut Wagner, Árpád Búrmen, Janez Puhan, Iztok Fajfar in Tadej Tuma  
Fakulteta za elektrotehniko  
Univerza v Ljubljani  
Tržaška cesta 25, SI-1000 Ljubljana, Slovenija  
*borut.wagner@fe.uni-lj.si*

## Abstract

### Application of extrapolation methods for steady-state computation of electrical circuits

*The paper presents four extrapolation methods: epsilon, rho, theta and topological epsilon algorithm. The methods are used for accelerated computation of the steady-state response of circuits that must be simulated for hundreds or even thousands of periods before they reach the steady-state.*

*A brief description of the methods is given upon which implementation details are discussed. The algorithms are tested on two circuits. The computation time required by the direct approach is compared to the time required by the extrapolation methods.*

*The results show that the extrapolation methods are appropriate for the rapid evaluation of the steady-state response of circuits excited by a single periodic signal.*

## 1 Uvod

Računanje stacionarnega stanja električnih vezij je lahko v določenih primerih zelo dolgotrajno, npr. pri oscilatorjih, ozkopasovnih stih in ojačevalnikih, preklopnih napajalnikih itd. Električno vezje je treba simulirati tudi več 1000 period, da se prehodni pojavi iznihaajo in da vezje doseže stacionarno stanje.

Za pospešitev izračuna stacionarnega stanja vezja v časovnem prostoru so na voljo različne ekstrapolacijske metode [1, 2, 3, 4], s katerimi lahko tudi do nekajkrat pospešimo izračun.

V članku predstavljamo štiri ekstrapolacijske metode, ki so primerne za uporabo skupaj s simulatorjem električnih vezij SPICE OPUS [5, 6, 7]. Metode smo preizkusili na dveh vzorčnih vezjih. Prikazana je primerjava števila period za izračun stacionarnega stanja pri direktni metodi (običajna tranzientna analiza) in pri uporabi ekstrapolacijskih metod.

## 2 Ekstrapolacijske metode

Za pospešitev izračuna stacionarnega stanja vezja lahko uporabimo različne ekstrapolacijske metode. V članku smo uporabili štiri metode: epsilon algoritem [2], rho algoritem [4], theta algoritem [4] in topološki epsilon algoritem [3]. Pri izračunu stacionarnega stanja vezij s pomočjo teh metod vezje simuliramo prek manjšega števila period. Iz stanja vezja (vozliščne napetosti, vejski tokovi) nato izračunamo začetno stanje za novo simulacijo. Na kakšen način se izračuna novo začetno stanje, je odvisno od izbrane ekstrapolacijske metode.

Nelinearno električno vezje opišemo s sistemom navadnih diferencialnih enačb

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t). \quad (1)$$

Pri reševanju upoštevamo začetno stanje  $\mathbf{x}_0(0)$ . Vektor  $\mathbf{x}(t)$  predstavlja vozliščne napetosti in vejske tokove vezja. Rezultat simulacije vezja je  $\mathbf{x}(t)$ ,  $t > 0$ . Iz odziva vezja  $\mathbf{x}(t)$  izberemo tiste  $\mathbf{x}(t)$ , ki ustrezajo

$$\mathbf{x}^{(n)} = \mathbf{x}(t_{del} + nT), \quad n = 0, 1, 2, 3, \dots, \quad (2)$$

pri čemer je  $T$  osnovna perioda signalov v vezju,  $t_{del}$  pa zakasnitev za določitev  $\mathbf{x}^{(0)}$ . Z uporabo ekstrapolacijskih metod iz  $\mathbf{x}^{(n)}$  izračunamo začetno stanje  $\mathbf{x}_0(0)$  in ponovno simuliramo vezje (en. (1)).

### 2.1 Epsilon algoritem

Pri epsilon algoritmu novo začetno stanje izračunamo s pomočjo rekurzivne zveze

$$\begin{aligned} \epsilon_{-1}^{(n)} &= \mathbf{0}, & n &\in \mathbb{N} \\ \epsilon_0^{(n)} &= \mathbf{x}^{(n)}, & n &\in \mathbb{N}_0 \\ \epsilon_{k+1}^{(n)} &= \epsilon_{k-1}^{(n+1)} + (\epsilon_k^{(n+1)} - \epsilon_k^{(n)})^{-1}, & n, k &\in \mathbb{N}_0 \end{aligned} \quad (3)$$

Inverz vektorja  $\epsilon_k^{(n+1)} - \epsilon_k^{(n)}$  v en. (3) in v enačbah pri rho in theta algoritmu (en. (5) in (6)) je psevdoinverz in ga izračunamo kot

$$\mathbf{x}^{-1} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}. \quad (4)$$

Veljavni rezultat epsilon algoritma so samo elementi s sodim indeksom  $k$ . Zato za začetno stanje vezja za novo simulacijo vzamemo  $\mathbf{x}_0(0) = \epsilon_{2k}^{(0)}$ .

## 2.2 Rho algoritem

Začetno stanje  $\mathbf{x}_0(0)$  izračunamo z rho algoritmom

$$\begin{aligned} \rho_{-1}^{(n)} &= \mathbf{0}, & n &\in \mathbb{N} \\ \rho_0^{(n)} &= \mathbf{x}^{(n)}, & n &\in \mathbb{N}_0 \\ \rho_{k+1}^{(n)} &= \rho_{k-1}^{(n+1)} + (k+1) \left[ \rho_k^{(n+1)} - \rho_k^{(n)} \right]^{-1} & n, k &\in \mathbb{N}_0, \end{aligned} \quad (5)$$

pri čemer vzamemo za  $\mathbf{x}_0(0) = \rho_{2k}^{(0)}$ . Posebna inačica rho algoritma predpostavlja, da je zaporedje  $\{\|\mathbf{x}^{(n)}\|\}_{n=0,1,2,\dots}$  naraščajoče. V tem primeru lahko člen  $(k+1)$  v en. (5) zamenjamo z  $(\mathbf{x}_{n+k+1} - \mathbf{x}_n)$ . Pri vezjih zaporedje ni vedno naraščajoče, zato smo uporabili splošni rho algoritem, en. (5).

## 2.3 Theta algoritem

Theta algoritem je podoben epsilon algoritmu, le da se izračun členov za sode  $k$  razlikuje od izračuna za lihe  $k$ .

$$\begin{aligned} \vartheta_{-1}^{(n)} &= \mathbf{0}, & n &\in \mathbb{N} \\ \vartheta_0^{(n)} &= \mathbf{x}^{(n)}, & n &\in \mathbb{N}_0 \\ \vartheta_{2k+1}^{(n)} &= \vartheta_{2k-1}^{(n+1)} + (\Delta\vartheta_{2k}^{(n)})^{-1} & k, n &\in \mathbb{N}_0 \\ \vartheta_{2k+2}^{(n)} &= \vartheta_{2k}^{(n+1)} + \left[ \Delta\vartheta_{2k}^{(n+1)} \right] \left[ \Delta\vartheta_{2k+1}^{(n+1)} \right] \left( \Delta^2\vartheta_{2k+1}^{(n)} \right)^{-1} & k, n &\in \mathbb{N}_0 \end{aligned} \quad (6)$$

V en. (6) smo uporabili okrajšavi za  $\Delta\vartheta_k^{(n)} = \vartheta_k^{(n+1)} - \vartheta_k^{(n)}$  in  $\Delta^2\vartheta_k^{(n)} = \Delta\vartheta_k^{(n+1)} - \Delta\vartheta_k^{(n)}$ . Začetni približek za novo simulacijo vezja je spet vrednost izračunanega elementa pri sodem indeksu  $k$ , t.j.  $\mathbf{x}_0(0) = \vartheta_{2k}^{(0)}$

## 2.4 Topološki epsilon algoritem

Razlika med epsilon algoritmom, opisanim v 2.1, in topološkim epsilon algoritmom je v načinu izračuna inverza vektorja. Pri topološkem epsilon algoritmu inverz izračunamo tako, da v en. (4) zamenjamo kvadrat norme s skalarnim produktom  $(\mathbf{x}, \mathbf{x})$  ter  $\mathbf{x}$  v števcu in levi  $\mathbf{x}$  v imenovalcu z  $\mathbf{y}$ . Tako dobimo

$$\mathbf{x}^{-1} = \frac{\mathbf{y}}{(\mathbf{y}, \mathbf{x})}, \quad (7)$$

kar imenujemo *inverz vektorja  $\mathbf{x}$  glede na vektor  $\mathbf{y}$* . Topološki epsilon algoritem tako opišemo z naslednjimi enačbami:

$$\begin{aligned} \epsilon_{-1}^{(n)} &= \mathbf{0}, & n &\in \mathbb{N} \\ \epsilon_0^{(n)} &= \mathbf{x}^{(n)}, & n &\in \mathbb{N}_0 \\ \epsilon_{2k+1}^{(n)} &= \epsilon_{2k-1}^{(n)} + \mathbf{y}/(\mathbf{y}, \Delta\epsilon_{2k}^{(n)}) & n, k &\in \mathbb{N}_0 \\ \epsilon_{2k+2}^{(n)} &= \epsilon_{2k}^{(n)} + \Delta\epsilon_{2k}^{(n)} / (\Delta\epsilon_{2k}^{(n)}, \Delta\epsilon_{2k+1}^{(n)}) & n, k &\in \mathbb{N}_0 \end{aligned} \quad (8)$$

Pri sodih indeksih  $k$  računamo inverz vektorja glede na  $\Delta\epsilon_{2k}^{(n)}$ , pri lihah pa glede na  $\mathbf{y}$ , ki je za vse  $n$  pri določenem  $k$  enak in je lahko poljuben z omejitvijo, da vsi  $\epsilon_{2k+1}^{(n)}$  obstajajo.

## 3 Uporaba metod s programskim paketom SPICE

Opisane ekstrapolacijske metode smo preizkusili s simulatorjem električnih vezij SPICE OPUS [5, 6, 7]. Program najprej požene nekaj period simulacije vezja brez začetnega stanja oz. z  $\mathbf{x}_0 = \mathbf{0}$ . Nato določi vrednosti  $\mathbf{x}^{(n)}$  (en. (2)) in iz njih z uporabo ekstrapolacijskega algoritma izračuna novo začetno stanje  $\mathbf{x}_0$  za naslednjo iteracijo oz. za naslednjo simulacijo vezja. Postopek je zaključen, ko je izpolnjen absolutni in relativni konvergenčni pogoj

$$\begin{aligned} |x_i(t_{sim}) - x_i(t_{sim} - T)| &\leq \\ \leq \delta_a + \max[|x_i(t_{sim})|, |x_i(t_{sim} - T)|] \delta_r \end{aligned} \quad (9)$$

za vse komponente  $x_i$  vektorja  $\mathbf{x}$ . Pri tem je  $t_{sim}$  čas simulacije vezja,  $\delta_a$  in  $\delta_r$  pa absolutna in relativna natančnost, ki ju želimo pri izračunu stacionarnega stanja.

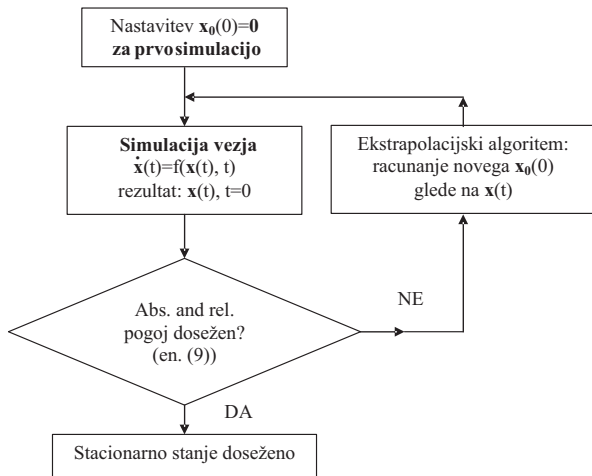
Iteracijski postopek je prikazan na sliki 1.

## 4 Primerjava metod

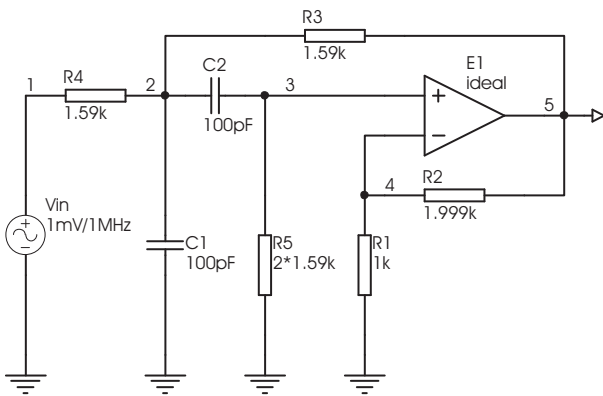
Opisane ekstrapolacijske algoritme smo preizkusili na dveh vzorčnih vezjih: ozkopasovnem situ in preklopnem napajalniku. Električni shemi vezij vidimo na slikah 2 in 4.

Kvaliteta ozkopasovnega filtra je  $Q = 100$ , resonančna frekvenca pa  $1MHz$ . Vhod v vezje je v vozlišču 1, izhod v vozlišču 5. Na vhodu v vezje je sinusni generator s frekvenco  $1MHz$  in amplitudo  $1mV$ . Odziv vezja v stacionarnem stanju je prikazan na sliki 3.

Na vhod v vozlišče 1 preklopnega napajalnika je priključen enosmerni napetostni vir z napetostjo  $20V$ . Na vrata tranzistorja  $M1$  je priključen generator pravokotnih napetostnih pulzov z enosmerno komponento  $10V$  in amplitudo  $10V$ . Izhod vezja je v vozlišču 4, kamor je priključeno še breme z upornostjo



Slika 1: Iteracijski postopek za izračun stacionarnega stanja električnega vezja z ekstrapolacijskimi metodami



Slika 2: Ozkopasovno sito

$1k\Omega$ .

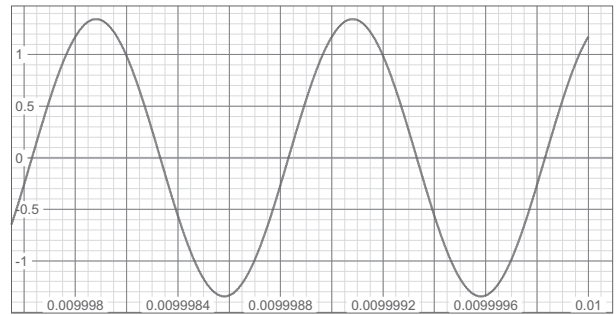
Odziv preklopnega napajalnika v stacionarnem stanju je prikazan na sliki 5.

Vezji smo najprej simulirali z običajno tranzientno analizo. Za doseg stacionarnega stanja je bilo potrebno simulirati  $N_{TRAN}$  period vhodnega signala. Nato smo izračunali stacionarno stanje obeh vezij še z epsilon, rho, theta in topološkim epsilon algoritmom. Skupno število period, potrebnih za doseg stacionarnega stanja, je označeno z  $N_e$ ,  $N_p$ ,  $N_\theta$  in  $N_{TEA}$ .

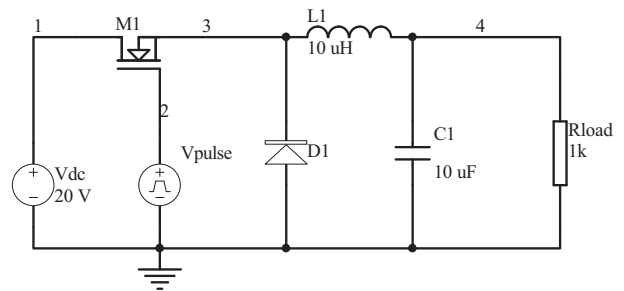
Pri ozkopasovnem situ sta bili izbrani vrednosti parametrov  $\delta_a = 10^{-7}$  in  $\delta_r = 10^{-6}$ , pri preklopnem napajalniku pa  $\delta_a = 10^{-6}$  in  $\delta_r = 10^{-7}$ .

Rezultati so prikazani v tabeli 1.

Pri vseh opisanih ekstrapolacijskih algoritmih je treba nastaviti še določene parametre, od katerih je odvisna hitrost konvergence in s tem število period simulacije vezja. Njihov opis in pomen je opisan v [8]. Vrednosti parametrov so podane v tabeli 2.



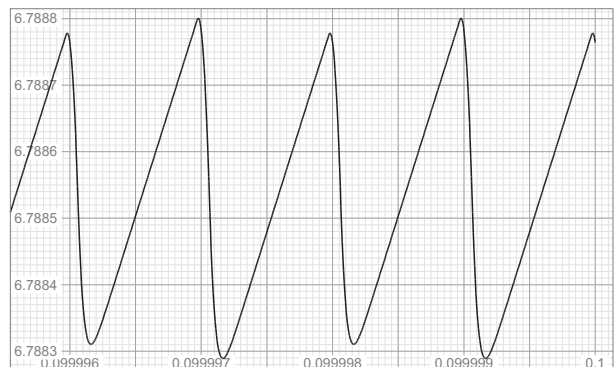
Slika 3: Stacionarni odziv ozkopasovnega sita (vozlische  $v_5(t)$ )



Slika 4: Preklopni napajalnik

Iz rezultatov v tabeli 1 je razvidno, da je za izračun stacionarnega stanja potrebno mnogo manj period simulacije vezja, če uporabljamo ekstrapolacijske metode, kot če vezje simuliramo z običajno tranzientno analizo toliko časa, da dosežemo stacionarno stanje. Faktor pohitritve izračuna stacionarnega stanja, ki je določen kot razmerje med številom period pri tranzientni analizi in pri ekstrapolacijski metodi, je za obe vezji in vse opisane ekstrapolacijske metode večji od 50, kar je zelo uporabno pri računalniškem načrtovanju vezij in pri optimizaciji [9], kjer je treba vezje velikokrat simulirati.

Od preizkušenih ekstrapolacijskih metod sta za uporabljeni vzorčni vezji najhitrejša epsilon in



Slika 5: Stacionarni odziv preklopnega napajalnika (vozlische  $v_4(t)$ )

	Ozkopasovno sito	Preklopni napajalnik
$N_{TRAN}$	10.000	10.000
$N_\epsilon$	12	24
$N_\rho$	94	30
$N_\vartheta$	182,25	75,7
$N_{TEA}$	12	24

Tabela 1: Primerjava števila period, ki jih potrebujemo, da dosežemo stacionarno stanje za obe vzorčni vezji pri običajni tranzientni analizi in pri uporabi ekstrapolacijskih metod (epsilon, rho, theta algoritem in topološki epsilon algoritem-TEA)

	Ozkopasovni filter	Preklopni napajalnik
Algoritem	$\epsilon/\rho/\vartheta/TEA$	$\epsilon/\rho/\vartheta/TEA$
tdel	1/1/0,25/1 per.	0/0/0,4/0 per.
tdelSec	1/1/10/1 per.	0/0/0,3/0 per.
NumPointMin	4/4/6/4	6/6/6/6
NumPointMax	16/16/6/16	6/6/6/6
PointStep	2/2/3/2	2/2/3/2
PeriodsPerPoint	1/1/1/1	1/1/1/1
MaxFreq	1 GHz	500 MHz
Frequency	1 MHz	1 MHz

Tabela 2: Vrednosti parametrov ekstrapolacijskih algoritmov, glej [8]

topološki epsilon algoritem. Hitrost konvergence posameznega algoritma je zelo odvisna od tipa vezja in hitrosti konvergence njegovega odziva. Vsak ekstrapolacijski algoritem je prilagojen za nek tip konvergence.

## 5 Zaključek

Računanje stacionarnega odziva električnega vezja lahko pospešimo z uporabo ekstrapolacijskih metod, ki na podlagi nekaj period odziva vezja določijo začetno stanje vezja za naslednjo simulacijo. Postopek ponavljamo, dokler vezje ne doseže stacionarnega stanja.

V članku so bile predstavljene štiri ekstrapolacijske metode: epsilon, rho, theta in topološki epsilon algoritem. Algoritmi so bili preizkušeni na dveh vzorčnih vezjih. Z uporabo ekstrapolacijskih algoritmov se je izračun stacionarnega stanja skrajšal za več desetkrat v primerjavi z običajno tranzientno analizo.

## 6 Zahvala

Raziskave je sofinancirala Agencija za raziskovalno dejavnost v okviru programa P2-0246 - Algoritmi in optimizacijski postopki v telekomunikacijah.

## Literatura

- [1] K. S. Kundert, J. K. White, A. Sangiovanni-Vincentelli, "Steady-state methods for simulation analog and microwave circuits", Kluwer Academic Publishers, 1990.
- [2] S. Skelboe, "Computation of the Periodic Steady-State Response of Nonlinear Networks by Extrapolation Methods", *IEEE Transactions on Circuits and Systems*, vol. CAS-27, no. 3, pp. 161-175, Marec 1980.
- [3] A. Sidi, W. F. Ford, D. A. Smith, "Acceleration of Convergence of Vector Sequences", *SIAM Journal on Numerical Analysis*, vol. 23, no. 1, pp. 178-196, Februar 1986.
- [4] E. J. Weniger, "Nonlinear Sequence Transformations: Computational Tools for the Acceleration of Convergence and the Summation of Divergent Series", URL: <http://arxiv.org/pdf/math.CA/0107080>
- [5] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, "SPICE3 Version 3f4 User's Manual", University of California, Berkeley, California, 1989.
- [6] SPICE OPUS circuit simulator homepage: URL: <http://www.fe.uni-lj.si/spice/> Faculty of Electrical Engineering, Electronic Design Automation Laboratory: URL: <http://www.fe.uni-lj.si/edalab/>
- [7] J. Puhon, T. Tuma, I. Fajfar, "SPICE for Windows 95/98/NT", *Electrotechnical Review*, vol. 65, no. 5, pp. 267-271, Ljubljana, Slovenija, 1998.
- [8] B. Wagner, Á. Bürmen, J. Puhon, I. Fajfar, T. Tuma, "Computing the steady-state response of nonlinear circuits by means of  $\epsilon$ -algorithm" *Elektrotehniški vestnik*, vol. 72 (2):??-??, Ljubljana, Slovenija, 2005.
- [9] J. Puhon, T. Tuma, "Optimization of analog circuits with SPICE 3f4", *Proceedings of the ECCTD '97*, vol. 1, pp. 177-180, 1997.